Faculté des Sciences Appliquées
Département d'Electricité
Laboratoire de Microélectronique
Machine Learning Group

Probabilistic Models in Noisy Environments

And their Application to a Visual Prosthesis for the Blind

Cédric Archambeau

Thèse soutenue en vue de l'obtention du grade de
Docteur en Sciences Appliquées

Membres du jury:

Pr Michel Verleysen (Lab. de Microélectronique, UCL), promoteur
Dr Jean Delbeke (Lab. de Génie de la Réhabilitation Neurale, UCL), co-promoteur
Dr Tom Heskes (Radboud Universiteit Nijmegen)
Pr Léopold Simar (Institut de Statistique, UCL)
Pr Jean-Philippe Thiran (Ecole Polytechnique Fédérale de Lausanne)
Pr Joos Vandewalle (Katholieke Universiteit Leuven)
Pr Jean-Didier Legat (Lab. de Microélectronique, UCL), président

Septembre 2005

[. . .] tu vois, de bien regarder,
je crois que cela s'apprend.

Margeurite Duras

Pluralitas non est ponenda sine neccesitate

William of Ockham

Abstract

In recent years, probabilistic models have become fundamental techniques in machine learning. They are successfully applied in various engineering problems, such as robotics, biometrics, brain-computer interfaces or artificial vision, and will gain in importance in the near future. This work deals with the difficult, but common situation where the data is, either very noisy, or scarce compared to the complexity of the process to model. We focus on latent variable models, which can be formalized as probabilistic graphical models and learned by the expectation-maximization algorithm or its variants (e.g., variational Bayes).

After having carefully studied a non-exhaustive list of multivariate kernel density estimators, we established that in most applications locally adaptive estimators should be preferred. Unfortunately, these methods are usually sensitive to outliers and have often too many parameters to set. Therefore, we focus on finite mixture models, which do not suffer from these drawbacks provided some structural modifications.

Two questions are central in this dissertation: (i) how to make mixture models robust to noise, i.e. deal efficiently with outliers, and (ii) how to exploit side-channel information, i.e. additional information intrinsic to the data. In order to tackle the first question, we extent the training algorithms of the popular Gaussian mixture models to the Student-t mixture models. The Student-t distribution can be viewed as a heavy-tailed alternative to the Gaussian distribution, the robustness being tuned by an extra parameter, the degrees of freedom. Furthermore, we introduce a new variational Bayesian algorithm for learning Bayesian Student-t mixture models. This algorithm leads to very robust density estimators and clustering. To address the second question, we introduce manifold constrained mixture models. This new technique exploits the information that the data is living on a manifold of lower dimension than the dimension of the feature space. Taking the implicit geometrical data arrangement into account results in better generalization on unseen data.

Finally, we show that the latent variable framework used for learning mixture models can be extended to construct probabilistic regularization networks, such as the Relevance Vector Machines. Subsequently, we make use of these methods in the context of an optic nerve visual prosthesis to restore partial vision to blind people of whom the optic nerve is still functional. Although visual sensations

can be induced electrically in the blind's visual field, the coding scheme of the visual information along the visual pathways is poorly known. Therefore, we use probabilistic models to link the stimulation parameters to the features of the visual perceptions. Both black-box and grey-box models are considered. The grey-box models take advantage of the known neurophysiological information and are more instructive to medical doctors and psychologists.

Acknowledgements

I would like to thank Prof. Michel Verleysen for his guidance and for letting me explore my own research directions, with a total freedom. It was a real pleasure working with someone having so many human qualities and who is always available, although his very busy schedule.

I would also like to thank the members of the jury for having accepted to be part of my doctoral committee and for their constructive comments, as well as Prof. Charles Trullemans for having given me the opportunity to work on a challenging biomedical research project.

I am grateful to all the members of the UCL Machine Learning Group, past and present, for the stimulating discussions we had, but more importantly for their enthusiasm and friendship. Moreover, I sincerely appreciate the excellent work done by the system administration team of the Microelectronics laboratory.

Of course, I am very thankful to all the persons who spend many hours proof-reading and improving this text. In particular, I am very grateful to Vanessa Cannone for her careful revision of the English language.

My last thoughts are for my family and friends, who have given me the strength to complete this doctorate. Most of all, I would like to thank Debora Cannone for her love, encouragement and patience.

This work was supported by the European Commission (IST-2000-25145).

Contents

CHAPTER 1

Introduction

Machine learning and more generally artificial intelligence is about to play a crucial role in our modern society. Machine learning aims at teaching machines, either how to perform tasks in an autonomous fashion, or how to make reasonable and sound decisions, for example, in order to assist experts in many scientific domains or to support the latest technological advances.

Today's electronic devices fulfill a wide variety of duties. Among the most important ones, there is the control, the processing and the distribution of information. Since the late '60, the advances made in the field of microelectronics enabled engineers to build powerful communication systems in which computers interact with each other and with humans. However, at present time, human-computer interactions are mainly low level in the sense that a computer can be made to represent and to solve a problem or some aspect of it, provided it is correctly configured (by programming) and it is given appropriate input data. In other words, computers usually execute fastidious and repetitive operations they have been assigned to and which can be described in terms of simple logical and numerical expressions.

In the early '90, the advent of large scale communication systems created the need to organize and process more efficiently the enormous amount of information transmitted through man-made communication networks. As a result, information technology (IT) emerged. IT is concerned with all aspects of managing and processing information. Computers play a central role in these tasks, as they are used to convert, store, protect, process, transmit and retrieve information from anywhere, at anytime.

While IT led to an information revolution, making information universally available and accessible, nowadays we are facing a knowledge revolution. Machines no longer only manage and process information, but they are given computational intelligence in order to create *new* information, i.e. knowledge. For instance, machines extract and transform the information that is hidden in large databases or their environment to let *intelligent* systems adapt automatically to our needs and desiderata or just to provide us with meaningful information. In the near future, these systems will appear almost everywhere in people's everyday life. Currently, machine learning tools are already invaluable for efficient data mining and gain in importance in domains such as robotics, artificial vision, biometrics, speech processing, natural language processing,

brain-computer interfaces, haptics, bioinformatics, medical imaging, etc. However, in order to be capable of natural and seamless interaction with humans, some further advances in the fundamental techniques and their tailoring to various practical aspects are essential.

Among others, probabilistic and in particular Bayesian learning have emerged as approaches that are difficult to circumvent. In these approaches, knowledge is discovered based on the statistics of the observed data. The attractiveness of probabilistic models lies in their ability to bring powerful statistical tools into machine learning in order to represent uncertainty. This uncertainty may come from the noise on the data generation process or may be due to the fact that the number of data is small compared to their dimensionality or to the complexity of the process to model. The probabilistic approach allows us to deal with these sources of uncertainty in a principled way, by taking them into account explicitly when estimating the optimal model parameters.

Surprisingly, probabilistic models are based on only two fundamental rules: the sum rule and the product rule. For continuous random variables $\mathcal{X}$ and $\mathcal{Y}$, the sum rule states how to compute the marginal probability $p(\mathbf{x})$ based on the joint probability $p(\mathbf{x}, \mathbf{y})$:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y} \ . \tag{1.1}$$

The marginal probability $p(\mathbf{x})$ is thus obtained by integrating out $\mathbf{y}$, which can be viewed as a nuisance variable in this context. When dealing with discrete random variables, the integral is replaced by a sum.

The product rule says how to decompose the joint probability $p(\mathbf{x}, \mathbf{y})$. It is given by

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y}) p(\mathbf{y}) \ , \tag{1.2}$$

where $p(\mathbf{x}|\mathbf{y})$ is the conditional probability of $\mathbf{x}$ given $\mathbf{y}$. When $\mathcal{X}$ is independent from $\mathcal{Y}$, $p(\mathbf{x}|\mathbf{y})$ is equal to $p(\mathbf{x})$ and thus $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$.

From the product rule, it is straightforward to derive a third important rule: Bayes' rule. In analogy to (1.2), we have $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$. Equating both leads to the following expression:

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} \ , \tag{1.3}$$

where $p(\mathbf{y})$ and $p(\mathbf{y}|\mathbf{x})$ are respectively termed prior and posterior probability of $\mathbf{y}$. The normalizing constant $p(\mathbf{x})$ is nothing else than the marginal probability of $\mathbf{x}$, which is given by (1.1). The prior reflects usually an a priori belief on $\mathbf{y}$. Bayes' rule plays a very important role in Bayesian learning and in statistics in general as it allows updating the prior of $\mathbf{y}$ into its posterior, on the basis of the observation of $\mathbf{x}$.

Biomedical engineering also underwent a revolution in the past several years. Successful clinical systems stimulating electrically the nervous system have

emerged, including cochlear implants to restore hearing and deep brain stimulators to reduce symptoms of Parkinson's disease. Another complex system that is likely to emerge is the one that restores functional vision in profoundly blind individuals by electrical stimulation of the visual pathways. In particular, electrical stimulation of the optic nerve was proven to provide a viable solution when the blind's optic nerve is still functional, such as in retinitis pigmentosa and age-related macular degeneration.

In the frame of the European project OPTIVIP (*opti*mization of the *v*isual *i*mplantable *p*rosthesis), a blind female volunteer is chronically implanted with a complete optic nerve visual prosthesis. One of the main challenges is to induce meaningful visual sensations in his/her visual field, which requires to understand, decode and model the underlying neurophysiological process. Since very little is known about the coding scheme of the visual information along the visual pathways and since the underlying neurophysiological process is expected to be strongly nonlinear, nonlinear probabilistic techniques are suitable. Furthermore, the data gathered with the blind volunteer during exploratory stimulation sessions is expected to be very noisy since the degree of atrophy of her optic nerve is unknown and because of the high complexity of the data acquisition process.

The practical framework of this thesis is the optic nerve visual prosthesis, the ultimate goal being to link the stimulation parameters to the features of the visual sensations produced in the visual field of the volunteer. Therefore, we first study probabilistic models in the general case the underlying process is corrupted by lots of noise. In practice, very noisy environments are not uncommon, especially in medical and biomedical applications.

This work is organized as follows. In Chapter 2, we review nonparametric density estimation techniques and study their behavior in presence of noise. These techniques are fundamental statistical tools for data mining, Bayesian classification or statistical pattern recognition. A non-exhaustive list of multivariate kernel estimators is discussed and their performance is assessed on real data. The adequacy of several bandwidth selectors is investigated. Both methods with fixed smoothing and locally adaptive smoothing are considered. The popular leave-one-out cross-validation criterion for standard kernel density estimation is also extended to ordinary and weighted vector quantization-based kernel density estimation, as well as sample point kernel density estimation.

In Chapter 3, we study finite mixture models, which are the core of this work. Maximum likelihood, maximum a posteriori and Bayesian learning of Gaussian and Student-t mixtures are discussed in detail. The use of the student-t distribution is motivated by the fact that it is the robust counterpart of the Gaussian distribution. Since we use a latent variable formalism to describe the mixture models, their parameters can be learnt by means of the popular expectation-maximization algorithm and its extensions (e.g., variational Bayes). When viewing finite mixture models as a limiting case of the adaptive kernel density

estimators, they turn out to be a flexible and powerful alternative to nonparametric techniques. Variants improving the generalization capabilities of the mixture models and avoiding numerical instabilities in noisy environment are proposed. In particular, we introduce mixture models using the regularized Mahalanobis distance to determine the component's shape, as well as a practical maximum a posteriori framework. In addition, a new variational Bayesian learning algorithm is proposed for Student-t mixture models, which provides very robust density estimators and clustering tools. Furthermore, the algorithm leads to a tight variational lower bound, which can be used for automatic model selection. Finally, manifold constrained mixture models are introduced. They exploit the information that the data is embedded in a manifold of lower dimension than the dimension of the feature space. In practice, this leads to better generalization. Throughout this work, the emphasis is to analyze how these techniques perform in real applications. Whenever it is suitable, we give advice to the practitioners that are dealing with lots of noise and atypical observations, while the number of available data is limited.

In Chapter 4 we describe probabilistic regularization networks. In particular, the relevance vector machines, which are sparse Bayesian regularization networks, are discussed. We also show that the latent variable formalism studied in the previous chapter can be readily applied in this context.

In Chapter 5, the complete prototype of the optic nerve visual prosthesis is described and the probabilistic tools discussed in the previous chapters are used to model the neurophysiological process linking the stimulation parameters to the corresponding visual sensations generated in the visual field of the blind volunteer. Besides data mining, both classification and regression problems are involved. Entirely black-box models, as well as hybrid models are proposed. The hybrid models are grey-box models in the sense that they exploit as much as possible the known part of the neurophysiological process. Furthermore, they have a similar accuracy as their black-box counterpart.

Finally, the conclusions of this thesis are stated in Chapter 6. In this last chapter, we briefly discuss further research directions in Bayesian learning and the relevance of our results for the optic nerve visual prosthesis. We also point out which performance can be expected from this type of prothesis in the future, as well as the problems that are still unsolved.

To end this introduction, we summarize the contributions of this doctoral dissertation, which are three-fold:

Experimental contribution: Nonparametric kernel density estimators are assessed in the multivariate case, showing that adaptive estimators should be preferred in practice. Since these techniques are sensitive to noise and have often too many parameters to set, finite mixture models are more suitable in a similar context.

Theoretical contributions: Besides the extension of the leave-one-out cross-validation criterion to several adaptive kernel density estimators, the theoretical contributions focus on mixture modeling:

- Gaussian mixtures using the regularized Mahalanobis distance are introduced in order to avoid numerical instabilities and make them suitable for estimating arbitrary densities.
- A practical maximum a posteriori approach, which can be combined to the minimum message length principle to perform automatic model selection, is proposed.
- We extend all the training algorithms for Gaussian mixture models to the Student-t mixture models in order to obtain methods that are robust to atypical observations (outliers).
- We introduce a new variational Bayesian learning algorithm for Student-t mixture models, which leads to (i) robust density estimation, (ii) very robust clustering and (iii) robust automatic model selection.
- We also introduce manifold constrained mixture models in order to take advantage of the geometrical arrangement of the data when learning the parameters.
- Finally, we show that the latent variable formalism used for mixture models can be extended to probabilistic regularization networks.

Applicative contribution: The probabilistic models discussed in this work are used to model the neurophysiological process that is involved when inducing visual perceptions in the blind with an optic nerve visual prosthesis. The ultimate goal of these models is to reconstruct images such that they are meaningful for them and help medical doctors to better understand the underlying neurophysiology.

CHAPTER 2

A Review of Kernel Density Estimation

Probability density estimation is a fundamental concept in statistics (e.g., Izenman, 1991) and machine learning (e.g., Cheng and Titterington, 1994). It provides a solid basis to data mining, knowledge discovery, pattern recognition and unsupervised learning in general. Jain, Duin and Mao (2000) emphasize that, in the field of pattern recognition, the statistical approach is the most intensively used. Statistical pattern recognition (Fukunaga, 1972) was successfully applied to bioinformatics, industrial automation, remote sensing, medical diagnosis, speech processing or biometrics. It is the study of how machines observe the environment, learn to distinguish patterns of interest and make sound and reasonable decisions about categories.

Consider an unknown process described by a continuous random variable $\mathcal{X}$. This variable can be specified in a natural way by means of its probability density function (PDF). The PDF provides a very rich source of information of the underlying process, as it allows extracting key quantities such as the mean, the most probable value (mode), the dispersion around the mean (variance), the degree of asymmetry (skewness) and many other characteristic quantities. Besides, it enables us to determine in which portion of space the PDF exists, i.e. where $\mathcal{X}$ can take a certain value and with which probability. Unfortunately, in practice the true PDF is unknown. Only a finite and noisy realization of $\mathcal{X}$ is observed. Hence, estimating the PDF consists in describing the imperfect process by characterizing the behavior $\mathcal{X}$, based on the observations. When performing density estimation, two major families of methods can be considered: the parametric and the nonparametric ones. A third family, lying somewhat in between, are finite mixture models. They will be studied in detail in Chapter 3.

Parametric PDF estimation assumes the data is drawn from a specific density model. The unknown PDF is estimated by fitting an a priori chosen functional form (e.g., a Gaussian distribution) to the observed data. Of course, this a priori choice is too restrictive in most engineering and biomedical applications, as it might give a false representation of the underlying process. By contrast, in nonparametric density estimation (Silverman, 1986; Izenman, 1991; Scott, 1992; Wand and Jones, 1995; Härdle, Müller, Sperlich and Werwatz, 2004) the information embedded in the data is extracted by making as few assumptions as possible, that is to say letting the data "speak for themselves". Nonparametric

PDF estimation captures the underlying structure of the data without assuming any functional form. The estimate is taken to belong to a large enough family of densities so that it cannot be represented through a finite number of parameters. Nonparametric methods are therefore suitable to model any arbitrary density and are applicable in a much broader range of applications. Smoothness conditions are usually imposed on the estimate (and on its derivatives) and it should satisfy the following general constraints[1]:

$$\forall \mathbf{x} \in \mathbb{R}^d : p(\mathbf{x}|X, \mathcal{H}_M) \geq 0 \ , \quad \int p(\mathbf{x}|X, \mathcal{H}_M) d\mathbf{x} = 1 \ , \tag{2.1}$$

where d is the dimension of the feature space. Features are measurable characteristics by which the observations can be described and represented. The density $p(\mathbf{x}|X, \mathcal{H}_M)$ is a nonparametric estimate of the true density $p(\mathbf{x})$ having a model structure $\mathcal{H}_M$, its parameters being learned on the basis of a finite realization $X = \{\mathbf{x}_n\}_{n=1}^N$ of $\mathcal{X}$.

In the first part of this chapter, we present how to *learn* a density in a machine learning perspective. Statistical resampling techniques such as cross-validation and the bootstrap are recalled. Both are essential for model selection, especially when the data is limited and very noisy. As the standard error criteria are useless in unsupervised learning, it is proposed to use the average negative log-likelihood. This is motivated by the fact that this measure is closely related to the Kullback-Leibler divergence between the true density and its estimate. Even if it may be problematic in some particular cases, the average negative log-likelihood is a general criterion, which is viable and objective in practice, regardless of the method that is used.

In the second part, nonparametric kernel density estimators are reviewed. The key quantity in these methods is the amount of smoothing. We therefore discuss a non exhaustive list of data-driven smoothing selectors. Asymptotic criteria are also mentioned when appropriate. In particular, the approaches that can be used in multivariate PDF estimation problems are retained. Based on the smoothing selector, nonparametric estimators can be divided into two classes: the estimators with fixed smoothing and with locally adaptive smoothing. The first ones include the histogram and the popular Akaike-Parzen-Rosenblatt estimator. The second ones include the nearest neighbor, the adaptive or sample point kernel estimator, the vector quantization-based density estimator and the reduced set kernel density estimator. The problems faced with each method are extensively discussed.

At the end of this chapter, the quality of the estimators is assessed in presence of noise and according to the number of learning data. Two artificial benchmarks of increasing dimensionality are considered. Later on, the estimators are compared on real data. Univariate applications are briefly considered. Subsequently, we focus on multivariate problems, which are barely discussed in the literature. In most examples, the number of data is also limited.

[1] These constraints are general in the sense that they are satisfied for any density and thus imposed in parametric density estimation or finite mixture models as well.

2.1. Learning Densities

The aim of machine learning is to build a statistical model of the unknown process, which exhibits good generalization capabilities. It is not to fit perfectly the observed data generated by the process, but to predict well on new inputs. For a fixed size of the learning data set, the reliability of the estimators decreases when the number of parameters increases. The problem is even more severe in presence of noise. As a consequence, selecting the best model complexity, i.e. the optimal number of parameters, should be done carefully in practice, paying much attention to the amount of available data, the amount of noise and the complexity of the process to model.

2.1.1. Learning and Generalization

The methodology extensively used with artificial neural networks, such as multi-layer perceptrons (Rumelhart, Hinton and Williams, 1986; Bishop, 1995), radial basis function networks (Broomhead and Lowe, 1988; Moody and Darken, 1989) or the popular support vector machines (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000), is the hold-out method. It works in three successive steps: training, validation and test.

Consider the random variable $\mathcal{X}$ describing the unknown process of interest and let $X = \{\mathbf{x}_n\}_{n=1}^N$ be an identically and independently distributed (i.i.d.) sample of $\mathcal{X}$. Suppose a fixed model hypothesis $\mathcal{H}_M$ of complexity M and an a priori chosen error criterion E, which characterizes the prediction accuracy. In order to estimate the generalization capabilities of the model, the data is divided into three disjunct subsets: the training, the validation and the test set. First, the model parameters associated to the hypothesis $\mathcal{H}_M$ are computed by minimizing the training error, usually following an iterative scheme. As shown in Figure 2.1, the training error decreases as a function of the model complexity M. When M increases, the number of degrees of freedom increases as well, resulting in a more accurate description of the training data (if sufficient data are available). The training error is not a useful measure for selecting M, as it favors an ever increasing model complexity. Second, the validation error on the validation set is computed by using the optimal model parameters for hypothesis $\mathcal{H}_M$. In contrast with the training error, the validation error provides a measure for selecting M. Indeed, as the validation set is not used for training, the validation error is an estimate of the generalization error of this specific model. Looking to Figure 2.1, one can observe a minimum of the validation error, meaning that when the complexity is too high, the model performs worse, i.e. the model overfits the training data. Third, the generalization error is estimated on the test set for the optimal model complexity M_{opt}. This methodology can be applied to density estimation in a straightforward way, provided an adequate performance measure is given. Performance measures for PDF estimation will be further discussed in Section 2.1.3.

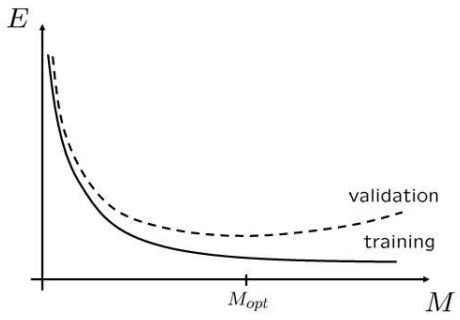


FIGURE 2.1. Evolution of the training and the validation error as a function of the model complexity M. While the training error decreases continually when M increases, the validation error first decreases, goes through a minimum and then increases due to overfitting. This phenomenon occurs when the model describes better the training data, but generalizes worse on new data (e.g., the validation data).

An approach related to the hold-out method and which is known as as early stopping, fixes the model complexity in advance (to a high value) and seeks for the optimal number of training iterations instead. In fact, a similar behavior of the errors is observed depending on the number of training iterations. While the training error continuously decreases, the validation error goes through a minimum. Good generalization is then obtained by stopping the training procedure at this point.

Curse of dimensionality

Although the actual definition of a density does not change as the dimensionality changes, there are subtle differences that are likely to make multivariate density estimation difficult. If we are forced to work with a limited number of data, as we are in practice, then increasing the dimensionality of the space rapidly leads to very sparse data, resulting in a very poor representation of the underlying density. The number of data required for a given accuracy grows exponentially with the number of features, which has been termed curse of dimensionality (Bellman, 1961), empty space phenomenon (Scott and Thompson, 1983) or peaking phenomenon (Jain et al., 2000). In addition, when moving to higher dimensions, regions of relatively low density, such as the distribution tails, can still be extremely important parts of the distribution (Silverman, 1986). Unfortunately, the tails are very difficult to model in practice.

2.1.2. Statistical Resampling Techniques

The problem with the hold-out method is that the data split is arbitrary and that it is wasteful of valuable data. This can possibly lead to a suboptimal

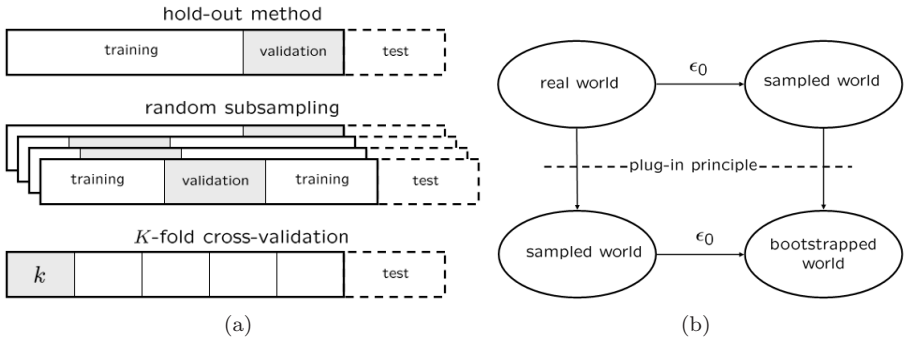


FIGURE 2.2. Examples of statistical resampling techniques. (a) shows the data partitions for the hold-out method, random subsampling and K-fold cross-validation. Although the validation sets are mutually exclusive in K-fold cross-validation, they are not in random subsampling. (b) illustrates the plug-in principle used in the bootstrap method. In this approach it is assumed that the optimism ϵ_0 is only due to the sampling process between the real world and the sampled world. It can therefore be simulated by subsampling the sampled world to construct the bootstrapped world.

choice of M, as it introduces a bias in the estimation of the generalization error. In practice, it is even more troublesome as the number of data can be relatively small and noisy. Therefore, more elaborate techniques are needed.

Random subsampling

A straightforward extension of the hold-out method that is less sensitive to the arbitrary split is random subsampling (Figure 2.2). The hold-out method is repeated K times and the generalization error is computed by averaging over the runs. However, the assumption of independency between instances of the validation (and learning) sets from successive runs is violated. Therefore, statistical resampling techniques (Efron and Tibshirani, 1993) such as cross-validation and bootstrapping are widely used instead. For a smaller number of runs, they allow to estimate the error with a greater reliability, especially when the sample size is limited. They are commonly used in machine learning for model selection and the optimization of the hyperparameters, i.e. other parameters that control the functional complexity of the resulting models.

Cross-validation

In K-fold cross-validation (Stone, 1974, 1975, 1977), the data is divided into K subsets of (approximately) equal size (Figure 2.2). The model is trained

K times, each time leaving out one of the subsets for validation. The generalization error is then estimated by computing the mean of the K validation errors:

$$E_{\text{cv}} = \frac{1}{K} \sum_{k=1}^{K} E_{X_{-k}, X_k} \ , \tag{2.2}$$

where X_{-k} and X_k denote respectively the complete data set X without subset k and subset k only. For two different data sets A_i and A_j, the error E_{A_i, A_j} is obtained by training the model with the data points of A_i and validating it on the data points of A_j. Confidence intervals are formed by using the standard deviation of E_{cv}, also known as standard error. In general, the variance of the mean of a finite population is equal to the variance of the population divided by its size. This leads to the following approximation for the standard deviation of E_{cv}:

$$\sigma(E_{\text{cv}}) \approx \frac{\hat{\sigma}(E_{X_{-k}, X_k})}{\sqrt{K}} \approx \frac{\hat{\sigma}(e_n)}{\sqrt{N}} \ , \tag{2.3}$$

where $\hat{\sigma}^2(\cdot)$ denotes the empirical variance and e_n is the validation error in $\mathbf{x}_n$. In (2.3), the equality $E_{\text{cv}} = \frac{1}{N} \sum_{n=1}^{N} e_n$ is used. It is important to realize that no unbiased estimator of the variance of K-fold cross-validation exists (Bengio and Grandvalet, 2004). In particular, for small sample sizes the bias incurred with respect to the variance may even be of the same order as the empirical variance itself.

When K equals the sample size N, the method is called leave-one-out cross-validation. Leave-one-out is nearly unbiased, but shows high variance and leads thus to unreliable estimates (Efron and Tibshirani, 1993). By contrast, K-fold cross-validation with moderate K reduces the variance while increasing the bias. In practice, 10-fold cross-validation seems to be a good compromise (Efron and Tibshirani, 1993; Kohavi, 1995). Further improvement can be obtained at an additional cost by Monte-Carlo simulations. Monte-Carlo cross-validation consists in repeating K-fold cross-validation multiple times and averaging E_{cv} over the Monte-Carlo runs.

Bootstrap

The bootstrap (Efron, 1979, 2003) is based on the plug-in principle. The generalization error is decomposed into two terms:

$$E_{\text{boot}} = E_{X,X} + \epsilon_0 \ , \tag{2.4}$$

where $E_{X,X}$ is the apparent error and ϵ_0 is the optimism. The first term is an "overtraining" error, and therefore optimistically biased. The second term is the correction between $E_{X,X}$ and the generalization error. In order to estimate the optimism, the plug-in principle states that this bias is only due to the sampling process of the real world (Figure 2.2). As a result, we may simulate this optimistic bias by first constructing a bootstrap sample, which is a subsample X^* of X with replacement and having the same size as X. Next, the optimism is considered as being the difference between the (over)training

error E_{X^*,X^*} and the validation error $E_{X^*,X}$ for a model trained with X^*. In order to be statistically reliable, the procedure is repeated B times:

$$\epsilon_0 = \frac{1}{B}\sum_{b=1}^{B}(E_{X_b^*,X} - E_{X_b^*,X_b^*}) \ . \tag{2.5}$$

As in K-fold cross-validation, the standard deviation of E_{boot} (or equivalently the standard deviation of ϵ_0) can be used to form confidence intervals.

The main problem with the standard bootstrap is that the optimism does not correct the bias sufficiently (Efron, 1983) and therefore does not lead to a good estimate of the generalization error. In the .632 bootstrap (Efron, 1983), the optimism $\epsilon_{.632}$ is estimated in a slightly different way, using only the points belonging to X and not to the bootstrap samples $\{X_b^*\}_{b=1}^B$:

$$\epsilon_{.632} = 0.632(\bar{E}_{X^*,X\setminus X^*} - E_{X,X}) \ , \tag{2.6}$$

where

$$\bar{E}_{X^*,X\setminus X^*} = \frac{1}{N}\sum_{n=1}^{N}\sum_{b=1}^{B}\frac{I(\mathbf{x}_n \in X \setminus X_b^*)e_n}{\sum_{b=1}^{B} I(\mathbf{x}_n \in X \setminus X_b^*)} \ . \tag{2.7}$$

The function $I(\cdot)$ is the indicator function. It is defined as follows:

$$I(x \in A) = \begin{cases} 1 & \text{if} \quad x \in A \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

The factor 0.632 in (2.6) can be motivated as follows. Since the data set X is sampled uniformly, the probability that a data point does not belong to a particular bootstrap sample X_b^* is $(1-1/N)^N \approx e^{-1} \approx 0.368$; the expected number of distinct instances from X appearing in X_b^* is thus $0.632N$. The .632 bootstrap estimate of the generalization error is then given by

$$E_{.632} = E_{X,X} + \epsilon_{.632} \tag{2.8}$$

$$= 0.368E_{X,X} + 0.632\bar{E}_{X^*,X\setminus X^*} \ . \tag{2.9}$$

According to Efron (1983), the .632 bootstrap is nearly unbiased. However, some ten years later it was reported by Kohavi (1995) that the .632 bootstrap may still have an important bias in some practical applications. More recently, Efron and Tibshirani (1997) proposed the .632+ bootstrap in order to correct the (small) pessimistic bias of .632 bootstrap. Unfortunately, this approach can only be applied in classification problems using the zero-one-loss (which counts the number of misclassifications) and will not be further discussed in this thesis.

The main drawback of statistical resampling techniques is that they are computationally very demanding. Other model selection methods include Akaike's information criterion (AIC) (Akaike, 1973) and Schwarz' Bayesian criterion (BIC) (Schwarz, 1978), but these asymptotic methods are often performing worse in practice, especially when the number of data is limited and in presence of noise and outliers. In Chapter 3, we will see how Bayesian techniques address this problem in a natural way by marginalizing over all the nuisance parameters.

2.1.3. Performance Measures

Typically, the L_2-norm of the prediction error, also known as the integrated squared error, is used in regression and the zero-one-loss in classification. While these standard error criteria are well suited in supervised learning, they are in general not in density estimation. Note however that the L_2-norm can be used in some particular asymptotic cases. Since density estimation is an unsupervised learning technique, the learning set does not contain input-output pairs, but only inputs. Nevertheless, when comparing different methods, it might be useful to assess them while knowing the target density in advance. Of course, this is not the case in practical applications.

Integrated Squared Error

Let $p(\mathbf{x})$ be the true density and $p(\mathbf{x}|X, \mathcal{H}_M)$ the density model of fixed structure $\mathcal{H}_M$, learnt with the observed data X. The integrated squared error (ISE) is commonly used to measure how well the entire curve $p(\mathbf{x}|X, \mathcal{H}_M)$ estimates $p(\mathbf{x})$. The goodness-of-fit is computed as follows:

$$\text{ISE} = \int \{p(\mathbf{x}|X, \mathcal{H}_M) - p(\mathbf{x})\}^2 d\mathbf{x} \ . \tag{2.10}$$

Depending on the realization X different estimators $p(\mathbf{x}|X, \mathcal{H}_M)$ are obtained. Taking the expectation with respect to the distribution of $p(\mathbf{x}|X, \mathcal{H}_M)$ at $\mathbf{x}$ gives the mean integrated squared error (MISE):

$$\text{MISE} = \text{E}\{\text{ISE}\} = \int \text{MSE}(\mathbf{x}) d\mathbf{x} \ , \tag{2.11}$$

where the mean square error (MSE) is defined at $\mathbf{x}$:

$$\begin{aligned} \text{MSE}(\mathbf{x}) &= \text{E}\left\{(p(\mathbf{x}|X, \mathcal{H}_M) - p(\mathbf{x}))^2\right\} & (2.12) \\ &= \text{E}\left\{\left(p(\mathbf{x}|X, \mathcal{H}_M) - \text{E}\{p(\mathbf{x}|X, \mathcal{H}_M)\}\right)^2\right\} \\ &\quad + \left(\text{E}\{p(\mathbf{x}|X, \mathcal{H}_M) - p(\mathbf{x})\}\right)^2 \ . & (2.13) \end{aligned}$$

The first term can be identified as the variance of the estimator at $\mathbf{x}$, while the second term is its squared bias at $\mathbf{x}$. The MISE is thus a global measure that corresponds to the integrated bias-variance trade-off of the estimator. Note that the empirical ISE divided by the number of observations is usually termed mean square error in the machine learning community and is often used in regression problems. This quantity approximates the expected squared error for a given model and a given X. In contrast to the MISE is concerned with the average over all possible data sets (Izenman, 1991).

Kullback-Leibler divergence

It is appealing to measure the dissimilarity between a target distribution $p(\mathbf{x})$ and its estimate $p(\mathbf{x}|X, \mathcal{H}_M)$ by the dispersion of their likelihood ratio with

respect to the target distribution, the likelihood ratio being given by

$$\phi = \frac{p(\mathbf{x}|X, \mathcal{H}_M)}{p(\mathbf{x})} \ . \tag{2.14}$$

This general class of divergence measures was formalized by Ali and Silvey (1966) and is known as F-divergence:

$$\mathrm{F}\big[p(\mathbf{x})\|p(\mathbf{x}|X, \mathcal{H}_M)\big] = g\left(\mathrm{E}\{f(\phi)\}\right) \ , \tag{2.15}$$

where $f(\cdot)$ is a convex function on $\mathbb{R}^+$ and $g(\cdot)$ is an increasing function on $\mathbb{R}$.

The Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) is a particular case of F-divergence. It measures the dissimilarity between the densities by posing $f(z) = -\log z$ and $g(z) = z$:

$$\mathrm{KL}\big[p(\mathbf{x})\|p(\mathbf{x}|X, \mathcal{H}_M)\big] = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{p(\mathbf{x}|X, \mathcal{H}_M)} d\mathbf{x} \geq 0 \ , \tag{2.16}$$

where log denotes the natural logarithm by convention. The KL divergence is minimum (and equal to zero) when both densities are identical and increases when the dissimilarity increases. It is not a distance, since the triangular inequality and the property of symmetry[2] are not respected, and it is sensitive to translation and scaling. In addition, the target distribution needs to be defined on the entire support of its estimate in order to be informative. Indeed, when it is not, the KL divergence tends to infinity. Related F-divergences include the Hellinger, the Bhattacharyya and the generalized Matustita dissimilarity measures. A non-exhaustive list can be found in Bassevile's survey (Basseville, 1989). In practice, they behave similarly as the KL divergence.

Average negative log-likelihood

Using the KL divergence or the ISE as generalization error in PDF estimation makes only sense when the target density is known, for example when using artificial generated data. In this thesis however, we are mainly interested in real applications, the target density being thus unknown. Therefore, the average negative log-likelihood is proposed as an alternative performance measure, as its computation does not require to know the target density, nor its support.

An important quantity in density estimation is the data likelihood. It is either used for characterizing the likelihood of a specific model hypothesis $\mathcal{H}_M$ having observed a particular data set X, or the likelihood of new observations. Let us here focus on the latter. Consider a fresh identically and i.i.d. sample $X' = \{\mathbf{x}_{n'}\}_{n'=1}^{N'}$ of $\mathcal{X}$. The likelihood of observing the new sample X' under the model hypothesis $\mathcal{H}_M$ and having learnt the model with sample X is the

[2]The KL divergence can be made symmetric by computing the following quantity: $\mathrm{KL}\big[p(\mathbf{x})\|p(\mathbf{x}|X, \mathcal{H}_M)\big] + \mathrm{KL}\big[p(\mathbf{x}|X, \mathcal{H}_M)\|p(\mathbf{x})\big]$. Nevertheless, this measure is not a distance either, as the triangular inequality does still not hold.

joint probability of X':

$$\mathcal{L}(X'|X,\mathcal{H}_M) \equiv p(X'|X,\mathcal{H}_M) = \prod_{n'=1}^{N'} p(\mathbf{x}_{n'}|X,\mathcal{H}_M) \ . \tag{2.17}$$

The likelihood measures the quality of the density model $p(\mathbf{x}|X,\mathcal{H}_M)$ with respect to the new observed data. In practice, it is convenient to take the negative logarithm of $\mathcal{L}(X'|X,\mathcal{H}_M)$ and to normalize it with respect to the number of data points, resulting in the average negative log-likelihood (ANLL) of X':

$$\mathrm{ANLL}_{X,X'} = -\frac{1}{N'} \sum_{n'=1}^{N'} \log p(\mathbf{x}_{n'}|X,\mathcal{H}_M) \ , \tag{2.18}$$

This performance measure can be regarded as an error function. It should be minimized and can be (reliably) estimated by statistical resampling techniques (see Section 2.1.2).

Let us now discuss the ANLL in more detail and show that this measure is well suited as error criterion in practical PDF estimation. Consider again definition (2.16) of the KL divergence. It can be decomposed as follows:

$$\begin{aligned} \mathrm{KL}\big[p(\mathbf{x})\|p(\mathbf{x}|X,\mathcal{H}_M)\big] &= \int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} - \int p(\mathbf{x}) \log p(\mathbf{x}|X,\mathcal{H}_M) & (2.19) \\ &= -\mathrm{H}(p(\mathbf{x})) - \mathrm{E}\{\log p(\mathbf{x}|X,\mathcal{H}_M)\} \ , & (2.20) \end{aligned}$$

where $\mathrm{H}(\cdot)$ is the differential entropy (Cover and Thomas, 1991), which is the extension of Shannon's entropy (Shannon and Weaver, 1963) to the continuous case. The second term in (2.20) is the expectation of the negative logarithm of the density model $p(\mathbf{x}|X,\mathcal{H}_M)$. It may be approximated by its empirical mean, which is nothing else than the average negative log-likelihood:

$$\mathrm{E}\{\log p(\mathbf{x}|X,\mathcal{H}_M)\} \approx -\frac{1}{N'} \sum_{n'=1}^{N'} \log p(\mathbf{x}_{n'}|X,\mathcal{H}_M) \ . \tag{2.21}$$

Furthermore, the entropy term is a constant that does not depend on the estimate $p(\mathbf{x}|X,\mathcal{H}_M)$ while the second term in (2.20) does. Minimizing the KL divergence consists in minimizing this second term, which is equivalent to minimizing the ANLL.

Remark that although this performance measure looks attractive for assessing unsupervised techniques, a serious problem may arise in practice. Assume $\mathbf{x}_{out} \in X'$ is isolated with respect to the learning data X, such that $p(\mathbf{x}_{out}|X,\mathcal{H}_M) = 0$. The logarithm of $p(\mathbf{x}_{out}|X,\mathcal{H}_M)$ is then equal to $-\infty$, resulting in an ANLL that is always equal to $+\infty$, regardless of the quality of the PDF in the other points of X'. We should therefore be careful when using blindly the ANLL in presence of strong outliers and when the densities have a limited support.

2.2. Kernel Density Estimators with Fixed Smoothing

In the previous section, the basic tools for learning densities given a particular model hypothesis $\mathcal{H}_M$ have been recalled. A simple, but general methodology is to use the ANLL as performance measure in conjunction with 10-fold cross-validation or the .632 bootstrap. This methodology can be applied for learning both the model complexity (i.e. the number of parameters) and selecting the optimal (hyper)parameters for this particular model complexity.

In the following, nonparametric kernel PDF estimators are reviewed and compared. Since the type and the amount of smoothing is crucial, we provide a non-exhaustive list of smoothing selectors, with a particular emphasis on approaches that are applicable in the multivariate case. In this section, the kernel estimators using a fixed smoothing parameter in the entire feature space are discussed. In the next section, kernel estimators using locally adaptive smoothing are considered at length.

2.2.1. Histogram

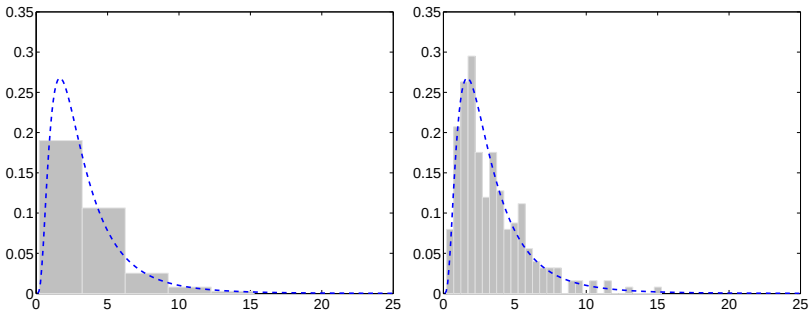
The simplest nonparametric density estimator is the histogram. Consider the random variable $\mathcal{X}$ describing an unknown process of interest and let $X = \{x_n\}_{n=1}^N$ be an i.i.d. sample of $\mathcal{X}$. Given these observations, the target PDF is approximated by dividing the real line in nonoverlapping bins $\{B_m\}_{m=1}^M$ of half bin width h and counting the number of data points falling into each of them, i.e. the frequency counts. The relative frequency associated to each bin is obtained by dividing its frequency count by the total number of observations N. In order to ensure that the integral of the estimate is equal to one, the relative frequency is also divided by the bin width $2h$, resulting in the following density model:

$$p(x|X, h, b_0) = \frac{1}{2Nh} \sum_{n=1}^{N} \sum_{m=1}^{M} I(x \in B_m) I(x_n \in B_m) \ , \tag{2.22}$$

where $B_m = [b_0 + 2h(m-1), b_0 + 2hm]$, b_0 is the first bin origin and $I(\cdot)$ is the indicator function. Even though the histogram is convenient for visualizing univariate data, it has many drawbacks. The estimate is discontinuous at the bin boundaries (and thus not differentiable), and may be zero outside a certain range. In addition, in order to construct a good estimator, one needs to carefully choose the first bin origin (starting point) and the half bin width h (or conversely the number of bins M).

Choice of the bin origin

The choice of the locations of the bin origins affects the quality of the estimate (Silverman, 1986; Härdle et al., 2004). One way to reduce this dependency is to use an averaged shifted histogram (Scott, 1985, 1992). In this approach, however, we need to choose additional parameters a priori, i.e. the number of shifts and the shift step. Besides, the kernel density estimator discussed



(a) Large h. (b) Small h.

FIGURE 2.3. Log-normal target distribution (dashed line) with mean equal to 1 and standard deviation to 0.7. The log-normal distribution is highly skewed. The effect of the choice of the half bin width h on the estimate is illustrated in (a) and (b). When h is too large the estimator is oversmoothed. By contrast, when h it is too small, the estimator does not reflect the true variations of the target PDF.

in Section 2.2.2 can be viewed as the limiting case of the averaged shifted histogram (Härdle et al., 2004).

Choice of the bin width

The histogram appears to be strongly dependent on the choice of h, as it regulates its smoothness. This is illustrated in Figure 2.3 on a toy example that will be used throughout this chapter for illustration purposes. The true underlying distribution is log-normal with mean equal to 1 and standard deviation equal to 0.7. In general, this distribution is difficult to estimate as it is highly skewed, i.e. asymmetric, and its support is equal to $\mathbb{R}^+$. The number of learning data is 250. When h is too small, the estimate is spiky and may thus not reflect the shape of the target PDF. By contrast, when it is too large, some important characteristics may be smoothed out. For instance, the decreasing character of the density when $\mathbf{x}$ tends to zero is not observed. Thus, using a fixed h is problematic as it may be locally unadapted (e.g., in the distribution tail). In general, problems occur when the dispersion of the data varies in different regions of the feature space.

Viewing histograms as finite mixture models

One advantage of histograms is that once they have been constructed, the data may be discarded. Only the bin locations and their amplitude need to be stored. By bin amplitude is meant the ratio between the relative frequency

and the bin width. As a result, (2.22) can be written in the following form:

$$p(x|\pi_1, \ldots, \pi_M, b_0) = \sum_{m=1}^{M} \pi_m \mathrm{I}(x \in B_m) \ , \tag{2.23}$$

where the bin amplitude π_m is defined as

$$\pi_m = \frac{1}{2Nh} \sum_{n=1}^{N} \mathrm{I}(x_n \in B_m) \ . \tag{2.24}$$

We refer the reader to Chapter 3 for a detailed discussion of finite mixture models.

Moving to higher dimensions

Last but not least, histograms cannot handle efficiently multivariate data. Due to the curse of dimensionality, moving to higher dimensions makes the number of bins increase exponentially with the dimension. Furthermore, additional parameters, such as the bin shape and orientation, need to be set. In practice, optimizing all the parameters becomes rapidly infeasible.

2.2.2. Kernel Density Estimator

The Akaike-Parzen-Rosenblatt kernel density estimator (KDE) (Akaike, 1954; Rosenblatt, 1956; Parzen, 1962) is a continuous estimator, which avoids the choice of the bin origin. Its multivariate extension was investigated by Calcoullos (1966) and Epanechnikov (1969). The target PDF is constructed by placing a well-defined kernel function on each data point of the learning set. The kernels are characterized by a width (or window), which is a common tuning parameter to all kernels. For a fixed value of this parameter, the PDF is estimated by making the sum of all the kernels over whole the domain and dividing it by a normalizing factor:

$$p(\mathbf{x}|X, \sigma) = \frac{1}{N\sigma^d} \sum_{n=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}_n}{\sigma}\right) \ , \tag{2.25}$$

where $K(\cdot) : \mathbb{R}^d \to \mathbb{R}$ and $\sigma > 0$ are respectively the kernel function and the kernel width. Usually, the kernel is chosen radially symmetric, integrates to one and is non-negative over its domain. As a result, the estimate automatically satisfies (2.1).

Whereas the computational effort for learning kernel estimators is limited to optimizing a single smoothing parameter, its model complexity scales linearly with the size of the data set. This leads rapidly to a prohibitive increase of memory usage. In Section 2.3.4, this problem is specifically addressed by optimally condensing the learning set. Moreover, when constructing kernel estimates that are locally adaptive, such as vector quantization-based estimators (Section 2.3.3), the model complexity is also kept small.

Choosing the kernel

In order to limit the number of parameters to set, it is convenient to sphere the data before estimating the PDF by removing the empirical mean and dividing by the empirical standard deviation. The isotropic Gaussian kernel can then be used for constructing the estimator (Hwang, Lay and Lippman, 1994):

$$K\left(\frac{\mathbf{x}-\mathbf{x}_n}{\sigma}\right) = \sigma^d \mathcal{N}(\mathbf{x}|\mathbf{x}_n, \sigma^{-2}\mathbf{I}) \ , \tag{2.26}$$

where the multivariate Gaussian distribution is defined as follows:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Lambda}|^{1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Lambda} (\mathbf{x}-\boldsymbol{\mu})\right) \ . \tag{2.27}$$

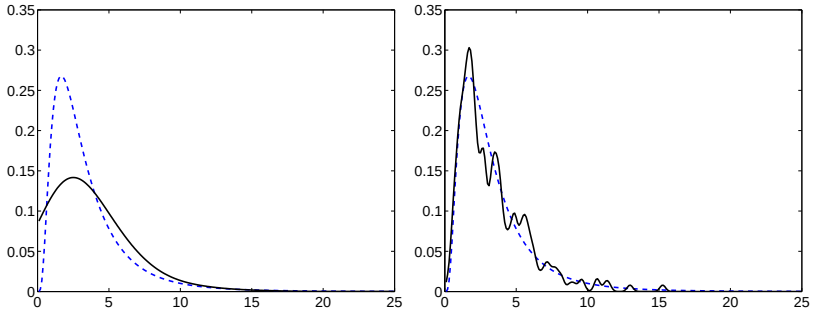
In this equation, $|\cdot|$ denotes the determinant, $\boldsymbol{\mu}$ is the kernel center and $\boldsymbol{\Lambda}$ is the kernel precision or inverse covariance matrix. Other kernels include the triangle, quartic or Epanechnikov kernels (Epanechnikov, 1969). However, for practical purposes the choice of the kernel function is almost irrelevant for the efficiency of the estimator (Härdle et al., 2004).

Choosing the kernel width

Similarly to histograms, which depend strongly on the value of the bin width h, the KDE depends strongly on the kernel width σ. Consider again the log-normal distribution with mean equal to 1 and standard deviation equal to 0.7. As noted before, this distribution is difficult to estimate, especially when using symmetric kernels having an unbounded support since the log-normal distribution is highly skewed and its support is equal to $\mathbb{R}^+$. Figure 2.4 shows the impact of the value of σ on the quality of Gaussian kernel density estimator. When σ is too large, the estimator is too flat. Large variation such as it is the case near the origin cannot be modeled. By contrast, when σ is too small, the large variation of the target PDF is easily modeled, but the estimate is spiky in the distribution tails, making the kernels clearly visible (cf. bumps). Those variations do not reflect the true underlying structure. Therefore, it is essential to optimize the kernel width carefully. In practice, the value of σ can be selected as the one that minimizes the ANLL. Of course, in order to avoid overfitting statistical resampling techniques are needed. Below, two popular alternatives to our methodology are presented.

The first method is based on the minimization of the asymptotic MISE ($N \to \infty$). It can be shown that, for an arbitrary radially symmetric kernel $K(\mathbf{t})$ with zero mean and finite variance, the asymptotic MISE can be approximated as follows (Silverman, 1986):

$$\mathrm{AMISE} \approx \frac{\sigma^4}{4}\left\{\int \mathbf{t}^2 K(\mathbf{t}) d\mathbf{t}\right\}^2 \int \left\{\nabla^2 p(\mathbf{x})\right\}^2 d\mathbf{x} + \frac{1}{N\sigma^d}\int K(\mathbf{t})^2 d\mathbf{t} \ , \tag{2.28}$$



(a) $\sigma = 2$. (b) $\sigma = 0.2$.

FIGURE 2.4. Effect of the choice of the kernel width on the Gaussian kernel density estimator of a log-normal distribution (dashed line). In (a) the kernel width is chosen too large and in (b) it is chosen too small. As a consequence, the estimator in (a) is oversmoothed (underfits), while the one in (b) is undersmoothed (overfits).

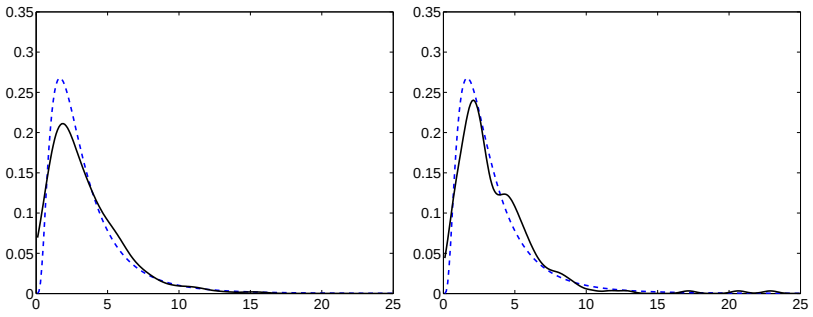
where ∇^2 is the Laplacian operator. The first term in this equation can be interpreted as the squared bias of the estimator and the second as its variance. The kernel width minimizing the AMISE, and thus achieving the best (asymptotic) bias-variance tradeoff, is:

$$(\sigma_{\text{AMISE}})^{d+4} = \frac{d \int K(\mathbf{t})^2 d\mathbf{t}}{N \left\{ \int \mathbf{t}^2 K(\mathbf{t}) d\mathbf{t} \right\}^2 \int \left\{ \nabla^2 p(\mathbf{x}) \right\}^2 d\mathbf{x}} \ . \tag{2.29}$$

The optimal kernel width cannot be computed in practice as it depends on the target density $p(\mathbf{x})$. However, using Gaussian isotropic kernels and plugging in a Gaussian distribution to compute $\nabla^2 p(\mathbf{x})$ leads to Scott's rule of thumb (Scott, 1992):

$$\hat{\sigma}_{\text{AMISE}} = \left(\frac{(d+2)N}{4} \right)^{-\frac{1}{d+4}} \hat{\sigma}_X \approx N^{-\frac{1}{d+4}} \hat{\sigma}_X \ , \tag{2.30}$$

where $\hat{\sigma}_X$ is the empirical standard deviation. The resulting estimator applied to the log-normal toy example is shown in Figure 2.5. While in this example the value provided by (2.30) leads to a fair estimate, the resulting performance is generally expected to be suboptimal. Indeed, the method selects the optimal σ according to an asymptotic criterion and uses a Gaussian approximation to compute the second order derivative of the target PDF. In practice, this value will only be valid in a limited number of applications. The optimal width strongly depends on the type of data we are dealing with, their number, the amount of noise they are corrupted by, and the dimension of the feature space. Besides, the Gaussian approximation leads to an oversmoothed estimate when the target is multi-modal or highly skewed. The solve-the-equation plug-in approach was proposed in order to find a better estimate of the kernel width.



(a) $\hat{\sigma}_{\text{AMISE}} = 0.85$. (b) $\hat{\sigma}_{\text{LOO}} = 0.5$.

FIGURE 2.5. Effect of the choice of the kernel width on the Gaussian kernel density estimator of a log-normal distribution (dashed line). Scott's rule of thumb and the leave-one-out cross-validation criterion are respectively used in (a) and (b). Both provide an intermediate kernel width with respect to the ones used in Figure 2.4. The resulting models generalize better, still the leave-one-out criterion seems to overfit slightly.

The method solves (2.29) iteratively, after having replaced $\nabla^2 p(\mathbf{x})$ by its non-parametric estimate $\nabla^2 p(\mathbf{x}|X, \mathcal{H}_M)$. A different kernel width is used however. Indeed, the optimal kernel width for $p(\mathbf{x})$ is sub-optimal for $\nabla^2 p(\mathbf{x})$. Luckily, both are linked, such that a fixed point of (2.29) can be found. In Jones, Marron and Sheather (1996), the approach was discussed in the one dimensional case only. Wand and Jones (1995) gave some clues to generalize the approach in the multivariate case, but many of the practical issues are still to be resolved. In addition, since this method does not resolve the problems linked to the use of a fixed smoothing parameter, it will not be further discussed.

The second method is an empirical one that is closely related to our methodology. Instead of minimizing the ANLL, it suggests to select σ by least squares cross-validation (Rudemo, 1982; Bowman, 1984). Consider again the ISE. Equation (2.10) can be decomposed as follows:

$$\text{ISE} = \int p(\mathbf{x}|X,\sigma)^2 d\mathbf{x} - 2\text{E}\{p(\mathbf{x}|X,\sigma)\} + \int p(\mathbf{x})^2 d\mathbf{x} \ . \tag{2.31}$$

When minimizing the ISE with respect to σ, the last term can be ignored as it only depends on the target distribution. Seeing that the second term can be approximated by its leave-one-out estimator, we may define the leave-one-out cross-validation criterion as

$$E_{\text{LOO}}(\sigma) = \int p(\mathbf{x}|X,\sigma)^2 d\mathbf{x} - 2\text{E}\{p(\mathbf{x}|X_{-n},\sigma)\} \tag{2.32}$$

$$\approx \int p(\mathbf{x}|X,\sigma)^2 d\mathbf{x} - \frac{2}{N}\sum_{n=1}^{N} p(\mathbf{x}_n|X_{-n},\sigma) \ , \tag{2.33}$$

where X_{-n} denotes the learning set without data point $\mathbf{x}_n$. Since the integral of a product of two Gaussian distributions is still a Gaussian distribution (depending only on the means), the following expression is obtained when using isotropic Gaussian kernels:

$$\hat{E}_{\text{LOO}}(\sigma) = \frac{1}{N^2} \sum_{n=1}^{N} \sum_{n'=1}^{N} \mathcal{N}(\mathbf{x}_n | \mathbf{x}_{n'}, (2\sigma^2)^{-1}\mathbf{I}) - \frac{2}{N(N-1)} \sum_{n=1}^{N} \sum_{\substack{n'=1 \\ n' \neq n}}^{N} \mathcal{N}(\mathbf{x}_n | \mathbf{x}_{n'}, \sigma^{-2}\mathbf{I}) \ . \tag{2.34}$$

The optimal kernel width is then found by exhaustive search:

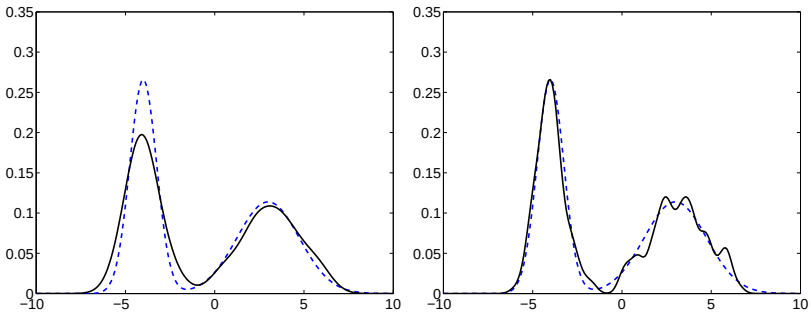
$$\hat{\sigma}_{\text{LOO}} = \underset{\sigma}{\text{argmin}} \ \hat{E}_{\text{LOO}}(\sigma). \tag{2.35}$$

This criterion is asymptotically unbiased (Stone, 1984). However, the first problem is that the approach is computationally very expensive. The second one is its high variance. A related approach using the (smoothed) bootstrap has been proposed by Taylor (1989) for minimizing the MISE, but better results were only reported for large data sets. From a practical point of view, the leave-one-out criterion tends to favor overcomplex models; it chooses a kernel width that is too small. This is illustrated in Figure 2.5 on the simple log-normal example.

The major drawback of kernel density estimation is that the width is fixed and identical for all the kernels, regardless of the local dispersion of the data in the feature space. As a consequence, either oscillations appear in the distribution tails, in regions of low-density or when dealing with multi-modal populations, or the estimator cannot accurately model high density regions. This is illustrated in Figure 2.6 for a bi-model 1D distribution. The target PDF is a mixture of two equally likely Gaussian distributions with different means and different standard deviations. Imposing an identical kernel width to all the kernels leads to locally mismatched kernel precisions. Since the value of the width is usually chosen as the one that minimizes a global error criterion, it is only well-founded in high-density regions. On the contrary, local mismatches occur in the low-density regions, because the kernel precisions are locally undersmoothed. This problem is addressed by kernel estimators with adaptive smoothing, which will be discussed in detail in Section 2.3.

Moving to higher dimensions

In contrast to the histogram, kernel density estimators do not need additional parameters to be set when the input dimension increases (if isotropic kernels are used). Of course, kernel estimators are also subject to the curse of dimensionality and need therefore an increasing amount of data when the dimension increases and/or an increasing amount of smoothing.



(a) $\sigma = 0.7$. (b) $\sigma = 0.3$.

FIGURE 2.6. Kernel density estimator of a bi-modal density (mixture of two equally likely Gaussian distributions). For a relatively large σ the flat Gaussian distribution is fairly estimated, while the peaky one is oversmoothed. For a relatively small σ strong oscillations appear in lower density regions. These mismatches are due to the varying dispersion of the data along the real axis. In practice, an intermediate σ is globally optimal. As a result, the peak is underestimated, while some oscillations are observed in the distribution tails of the flat component.

2.3. Kernel Density Estimators with Adaptive Smoothing

The density estimators described so far use a fixed kernel width. As shown on several simple examples, using estimators that are not locally adaptive leads to an oscillatory character in low density regions. The main reason is that the kernel width is selected according to a global criterion, favoring an accurate approximation of high density regions. In this section, kernel estimators that are adaptively smoothed are investigated. By contrast to the previous methods, they are quite sensitive to local irregularities in the data, such as sparseness or data clumping.

2.3.1. Nearest Neighbors Estimator

The M-nearest neighbor[3] (M-NN) estimator is a simple attempt to adapt locally the amount of smoothing (Loftsgaarden and Quesenberry, 1965). It has enjoyed a great success in pattern recognition and nonparametric discriminant analysis and was introduced quite early by Fix and Hodges (1951). The estimator is constructed by letting a hypervolume grow around $\mathbf{x}$ until it contains M

[3]In the literature, the term K-nearest neighbor is used instead of M-nearest neighbor. However, in order to avoid any confusion with the number of folds in K-fold cross-validation and as M denotes the complexity of the estimators throughout this thesis, a different notation is adopted.

data points of the learning set X. In general, the d-dimensional hypervolume is chosen to be the volume of the d-dimensional hypersphere:

$$V_d = c_d r^d = \frac{2\pi^{\frac{d}{2}} r^d}{\Gamma\left(\frac{d}{2}\right) d} \ , \tag{2.36}$$

where c_d is the hypervolume of the d-dimensional unit hypersphere, r is the radius of the hypersphere and $\Gamma(\cdot)$ is the gamma function. The resulting density estimator takes the following form:

$$p(\mathbf{x}|X, M) = \frac{M}{N V_d(\mathbf{x}|X, M)} \ , \tag{2.37}$$

where

$$V_d(\mathbf{x}|X, M) = c_d r(\mathbf{x}|X, M)^d \ . \tag{2.38}$$

In these equations, the probabilistic notation is abusively used to specify that the volume of the hypersphere and its radius depend conditionally on the learning set X and the number of neighbors M.

The M-NN can be viewed as a kernel estimator with kernel width $r(\mathbf{x}|X, M)$:

$$p(\mathbf{x}|X, M) = \frac{1}{N r(\mathbf{x}|X, M)^d} \sum_{n=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}_n}{r(\mathbf{x}|X, M)}\right) \ , \tag{2.39}$$

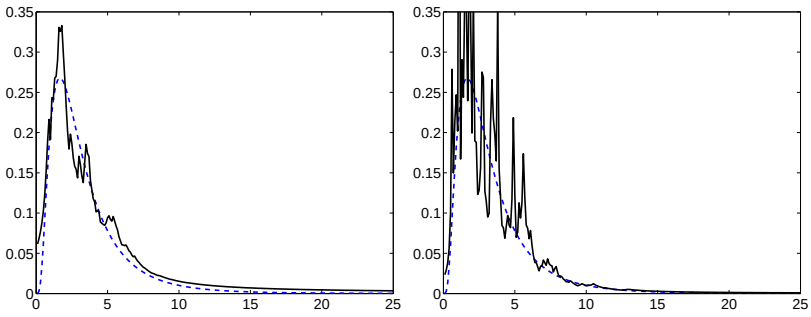
where the kernel is given by

$$K\left(\frac{\mathbf{x} - \mathbf{x}_n}{r(\mathbf{x}|X, M)}\right) = \begin{cases} {c_d}^{-1} & \text{if } \ (\mathbf{x} - \mathbf{x}_n)^{\mathrm{T}}(\mathbf{x} - \mathbf{x}_n) \leq r(\mathbf{x}|X, M) \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{2.40}$$

As for the KDE, the complete learning set need to be stored. Furthermore, the estimator is sensitive to local noise due to its adaptive character, shows discontinuities and has an infinite integral due to very heavy tails (Silverman, 1986). The kernel estimator undersmooths the tails, while M-NN overcompensates for this difficulty by smoothing them too much. As a result, its integral does not converge to one.

Choosing the number of neighbors

Figure 2.7 shows the effect of the number of neighbors on the M-NN estimators of the log-normal toy example that was already considered previously. First, we can clearly observe the discontinuities in the estimators. Second, the choice of M has a similar impact as the choice of σ in kernel density estimation: it regulates the radius of the hyperspheres. Therefore, it is essential to make a careful choice of M. Unfortunately, little has been done so far for selecting the optimal number of neighbors automatically. Silverman (1986) demonstrated that the optimal number of neighbors minimizing AMISE is approximately proportional to $N^{4/(d+4)}$. This result is not very helpful in practice as the constant of proportionality depends on $\mathbf{x}$. Nevertheless, M can still be selected by minimizing the ANLL.



(a) $M = 30$. (b) $M = 5$.

FIGURE 2.7. M-NN estimators of the log-normal distribution. The effect of the number of neighbors M on the quality of the estimate is illustrated in (a) and (b). The discontinuities in both cases can be observed, as well as the overestimated distribution tails. The dashed line shows the target distribution.

Moving to higher dimensions

According to Terrel and Scott (1992), the M-NN behave well only in higher dimensions. This result however is questionable. Beyer, Goldstein, Ramakrishnan and Shaft (1999) demonstrated that, for increasing dimensionality, the difference between the distance of a given data point to its nearest neighbor and its farthest neighbor does not increase as fast as its distance to its nearest points. This is already observed for dimensions as low as 10 to 15. For the Euclidean distance, this difference tends to zero (Hinneburg, Aggarwal and Keim, 2000). In other words, when the dimensionality increases, the relative contrast of the distances between different data points in the data set decreases. For M-NN, this suggests that the radius behaves similarly, tending to a unique value, regardless of the location of the reference point in the feature space. Mathematically, it can be formulated as follows:

$$\lim_{d \to +\infty} \frac{\mathrm{var}\{r(\mathbf{x}|X, M)\}}{\mathrm{E}\{r(\mathbf{x}|X, M)\}} = 0 \ , \tag{2.41}$$

where $\mathrm{var}\{\cdot\}$ is the variance with respect to $p(\mathbf{x})$. Figure 2.8 illustrates this phenomenon with a simple example. Consider a d-dimensional isotropic Gaussian distribution centered on the origin and with unit standard deviation. The proportion of neighbors is fixed in advance to 10%. Fifty M-NN estimators are constructed. When the dimension d increases, the relative contrast provided by the sphere radius decreases, because $\mathrm{E}\{r(\mathbf{x}|X, M)\}$ increases faster than $\mathrm{var}\{r(\mathbf{x}|X, M)\}$. As a result, the estimator tends to be flat regardless of M, the radius of the hyperspheres approaching the same value for increasing dimension.

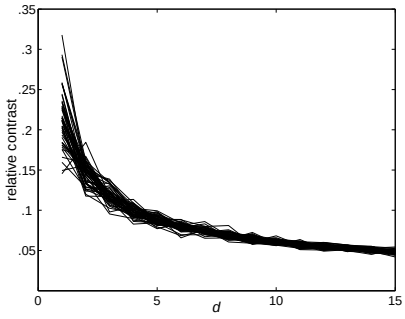


FIGURE 2.8. Evolution of the empirical relative contrast provided by the radius of the hypersphere in M-NN. The target distribution is a d-dimensional Gaussian distribution centered on the origin and having a unit standard deviation. Each curve corresponds to one out of the 50 estimators that were constructed.

As a matter of fact, this phenomenon is not only problematic for the M-NN estimator, but also for the standard and adaptive kernel estimators. All these techniques use mainly the Euclidean distance to determine the influence of neighboring data points and are thus prone to exhibit a flat character in very high dimensional spaces.

2.3.2. Sample Point Kernel Density Estimator

A global kernel width is only suitable when the data is homogenous. When the data statistics changes across the feature space, a local kernel width should be preferred. The sample point density estimator or adaptive kernel estimator (Breiman, Meisel and Purcell, 1977; Abramson, 1982; Silverman, 1986) is based on the common-sense notion that a natural way to deal with long-tailed distributions is to use a broader kernel in regions of low density. Thus, in order to build a locally adaptive density estimator, the mass of an observation in a low density region is smudged out over a wider range than in high density regions. Besides, an attractive property of the sample point estimator is that the approach provides a continuous estimate satisfying (2.1) automatically, unlike the M-NN estimator.

The sample point kernel density estimator (SKDE) works in three successive steps:

(1) Construct a pilot density estimator, which approximates (roughly) the true density and which is non-zero at each training datum:

$$\tilde{p}(\mathbf{x}_n|X,\mathcal{H}'_M) > 0 \ , \quad \forall n \ . \tag{2.42}$$

(2) Compute the data dependent kernel width factors:

$$\lambda_n = \left\{ \frac{\tilde{p}(\mathbf{x}_n|X, \mathcal{H}'_M)}{g} \right\}^{-\alpha} , \quad \forall n , \tag{2.43}$$

where α satisfies $0 \leq \alpha \leq 1$ and g is the geometric mean of $\tilde{p}(\mathbf{x}_n|X, \mathcal{H}'_M)$:

$$\log g = \frac{1}{N} \sum_{n=1}^{N} \log \tilde{p}(\mathbf{x}_n|X, \mathcal{H}'_M) . \tag{2.44}$$

(3) Construct the adaptive kernel estimator as follows:

$$p(\mathbf{x}|X, \sigma, \boldsymbol{\lambda}) = \frac{1}{N\sigma^d} \sum_{n=1}^{N} \frac{1}{\lambda_n{}^d} K\left(\frac{\mathbf{x} - \mathbf{x}_n}{\lambda_n \sigma} \right) , \tag{2.45}$$

where $\boldsymbol{\lambda} = \{\lambda_n\}_{n=1}^N$.

When the probability of $\mathbf{x}_n$ is high according to the pilot density, λ_n will be small, resulting in narrow kernels in high density regions. By contrast, when its probability is low, λ_n will be large, increasing the amount of smoothing locally. Parameter α controls the sensitivity of the SKDE to the local variations in the pilot estimator

It can be proven that the estimation bias decreases in comparison to the fixed kernel width estimators, while the covariance remains the same (Hall, Hui and Marron, 1995). However, unlike the approaches presented in the two following sections, the method suffers from the same drawback as the KDE regarding the computational burden of large size training sets: for each training data a term is added to (2.45).

Choice of the pilot density

The model structure $\mathcal{H}'_M$ of the pilot density is not necessarily the same as the one of the SKDE. Breiman et al. (1977) used the M-NN estimator as pilot density. A natural choice is rather to use the kernel estimator with fixed smoothing (Silverman, 1986; Hwang et al., 1994). As the resulting PDF estimator is insensitive to a fine detail of the pilot estimate, it is convenient to choose the kernel width by Scott's rule (2.30):

$$\tilde{p}(\mathbf{x}_n|X, \hat{\sigma}_{\text{AMISE}}) = \frac{1}{N(\hat{\sigma}_{\text{AMISE}})^d} \sum_{n=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}_n}{\hat{\sigma}_{\text{AMISE}}} \right) . \tag{2.46}$$

In any case, the use of the leave-one-out cross-validation criterion will not be rewarding due to its computational load.

Sain and Scott (1996) proposed the binned kernel density estimator that uses a piecewise constant kernel width function. A substantial improvement over the fixed kernel width estimator was reported. However, their discussion was limited to the one-dimensional case. In the multivariate case, the attractive

properties of binned kernel density estimators, such as for example the great computational savings, are lost (Holmström, 2000).

Choice of the sensitivity parameter

The larger the sensitivity parameter α is, the more sensitive will be the method to the pilot density. When α equals zero, the SKDE reduces to the standard kernel estimator with fixed smoothing. Abramson (1982) demonstrated both in the one-dimensional and the multi-dimensional cases that kernel width factors inversely proportional to the square root of the pilot density give an estimator whose bias is of a smaller order than that of the kernel estimator with fixed width. Furthermore, he showed that no other dependence of local kernel width on the pilot density will give this result. It is therefore common to choose $\alpha = 1/2$.

It was argued by Terrel and Scott (1992) that SKDE may have a non local behavior, that is, the estimate at a point may be significantly influenced by observations far away, leading mainly to lower convergence rates to the true PDF when $N \to \infty$. Nevertheless, good behaviors were still reported for small to moderate learning sets, which are mainly of interest in this work.

Choice of the kernel width

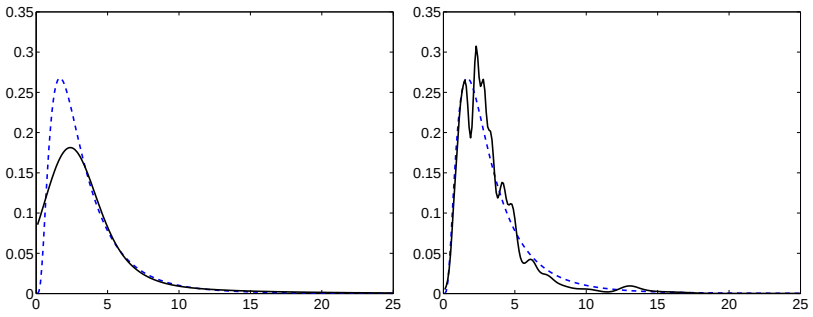
The choice of the kernel width is essential for constructing a good density estimator. Considering again the log-normal example in Figure 2.9, it is obvious that when σ is too large, the model underfits and when it is too small, overfitting occurs. The overall tendency of the SKDE is to increase locally the smoothness of the estimate. Therefore, for the same kernel width, either the overfitting is less than in the standard kernel estimator, or conversely, the underfitting is more important. In general, the optimal σ will be larger than for the kernel estimator with fixed smoothing.

The leave-one-out cross-validation criterion (minimizing ISE) can be extended in the case of the SKDE, the adaptive kernel factors being fixed. For Gaussian kernels, the following criterion is obtained:

$$\begin{aligned}\hat{E}_{\mathrm{LOO}}(\sigma) = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{n'=1}^{N}\mathcal{N}(\mathbf{x}_n|\mathbf{x}_{n'},(\lambda_n{}^2+\lambda_{n'}{}^2)^{-1}\sigma^{-2}\mathbf{I}) \\ -\frac{2}{N(N-1)}\sum_{n=1}^{N}\sum_{\substack{n'=1\\n'\neq n}}^{N}\mathcal{N}(\mathbf{x}_n|\mathbf{x}_{n'},(\lambda_{n'}\sigma)^{-2}\mathbf{I})\ . \end{aligned} \tag{2.47}$$

2.3.3. Vector Quantization-based Density Estimator

The main disadvantage of the KDE is the fixed kernel width and the high model complexity (equal to the number of data samples). A straightforward



(a) $\sigma = 2$. (b) $\sigma = 0.2$.

FIGURE 2.9. Effect of the choice of the kernel width on the SKDE of a log-normal distribution (dashed line) using Gaussian kernels. The sensitivity parameter is set to $1/2$ and Scott's rule is used to select the kernel width of the pilot density. In (a) the kernel width is chosen too large and in (b) too small.

way to avoid the local mismatch of the kernel precisions is to pre-process the data by vector quantization (VQ) (Holmström and Hämäläinen, 1993; Hwang et al., 1994; Voz, Verleysen and Comon, 1995). Interestingly, in contrast to the SKDE, which shares the same drawback as ordinary kernel density estimation regarding the model complexity, the VQ-based estimator provides a natural approach for reducing the size of the learning set at the same time.

Let A be the set of indices of the observed data $\{\mathbf{x}_n\}_{n=1}^N$ and B the set of indices of the prototypes $\{\boldsymbol{\mu}_m\}_{m=1}^M$ that minimizes an arbitrary reconstruction error R. Pre-processing the data by VQ results in applying the transformation $g_R(\cdot)$ on the indices:

$$\begin{gathered} g_R : A \subset \mathbb{N} \to B \subset \mathbb{N} \\ \text{s.t.} \\ \forall a \in A,\ \exists b \in B :\ g_R(a) = b \text{ and } |A| > |B| \ , \end{gathered}$$

where $|\cdot|$ denotes the cardinality of the sets. A wide variety of VQ schemes can be used to compute the kernel prototypes. Among the most popular ones, we have M-means (MacQueen, 1967), competitive learning (Grossberg, 1987; Ahalt, Krishnamurthy, Chen and Melton, 1990), neural-gas (Martinetz, Berkovich and Schulten, 1993) and Kohonen's self-organizing maps (Kohonen, 1995). In this work, we only consider competitive learning , as the other VQ methods lead to similar results.

The reconstruction error minimized by competitive learning is the mean square error:

$$R = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \boldsymbol{\mu}_{g_R(n)}\|^2 \ , \tag{2.48}$$

where $\|\cdot\|$ is the L_2-norm. In order to minimize R stochastically, the competitive learning algorithm proceeds as follows:

(1) Initialize the prototypes.
(2) Repeat until convergence:
  (a) For each datum $\mathbf{x}_n$, select the winner:

$$\boldsymbol{\mu}_{\text{win}} = \underset{\boldsymbol{\mu}_m}{\text{argmin}} \ \|\mathbf{x}_n - \boldsymbol{\mu}_m\|^2 \ . \tag{2.49}$$

  (b) Update the winner $\boldsymbol{\mu}_{\text{win}}$ according to:

$$\boldsymbol{\mu}_{\text{win}} \leftarrow \boldsymbol{\mu}_{\text{win}} + \alpha(\mathbf{x}_n - \boldsymbol{\mu}_{\text{win}}) \ . \tag{2.50}$$

In (2.50) α is the learning rate. Usually, α is chosen to decrease exponentially with the number of iterations on the training data set.

Once the VQ prototypes are computed, the underlying PDF is estimated as follows:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Mw^d} \sum_{m=1}^{M} \frac{1}{|\mathbf{S}_m|^{1/2}} K\left({\mathbf{S}_m}^{-1/2} \ \frac{\mathbf{x} - \boldsymbol{\mu}_m}{w} \right) \ , \tag{2.51}$$

where $\boldsymbol{\theta} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M, \mathbf{S}_1, \ldots, \mathbf{S}_M, w)$. Usually, the multivariate Gaussian kernel is used:

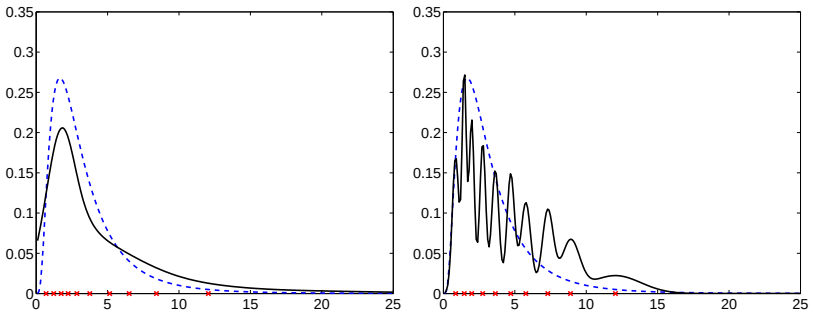
$$K\left({\mathbf{S}_m}^{-1/2} \ \frac{\mathbf{x} - \boldsymbol{\mu}_m}{w} \right) = w^d |\mathbf{S}_m|^{1/2} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, w^{-2} {\mathbf{S}_m}^{-1}) \ . \tag{2.52}$$

Parameter w is the width scaling factor. It regulates the overlap between the Gaussian kernels. In classical nonparametric density estimation, the precisions are identical for all kernels. In contrast, here they are locally data dependent through the empirical covariance matrix $\mathbf{S}_m$, associated to the Voronoi region of prototype $\boldsymbol{\mu}_m$. The Voronoi region of $\boldsymbol{\mu}_m$ is the region of the feature space mapped to $\boldsymbol{\mu}_m$, i.e. $\forall \mathbf{x} \in \mathbb{R}^d$ which is closer to $\boldsymbol{\mu}_m$ than to any other prototype.

The main drawback of competitive learning and VQ in general is that it involves an iterative nonlinear optimization scheme. As a result, the algorithm gets easily trapped into local minima of the reconstruction error surface. This can lead to a great variability in the generalization performance, depending on the initialization of the prototypes, the model complexity M and the learning rate α. In order to be less sensitive to the initial conditions, several runs with random initialization can be performed. The VQ with the lowest reconstruction error is therefore chosen. However, this procedure is relatively slow.

Choosing the width scaling factor and the model complexity

Given a model complexity M, VQ partitions the feature space into M Voronoi regions by associating the training data points to their closest prototype. As a result, the local dispersion of the data can be taken into account by determining the size and the orientation of each Voronoi region. This is computed by the kernel precision (inverse covariance matrix), which depends locally on the data.



(a) $w = 5$. (b) $w = 1$.

FIGURE 2.10. VQ-based kernel density estimators of the log-normal example. The number of prototypes is equal to 10. The crosses indicate their location. The impact of the width scaling factor is shown in (a) and (b). Here also, oscillations are observed when the value of the smoothing parameter is insufficient. The dashed line represents the target distribution.

The width scaling factor w is then optimized in order to enforce a smooth estimate. It plays thus a similar role as the kernel width σ in the KDE.

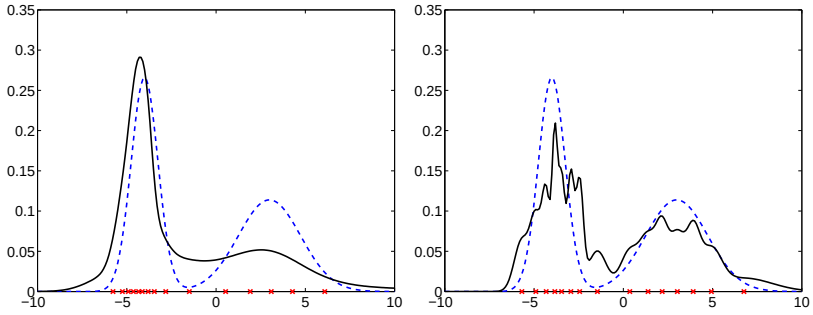
Being able to locally adapt to the data dispersion reduces the oscillations in the density estimates, which appear in low density regions of the target distribution. This can be observed in Figure 2.10 on the log-normal example. Again, we observe that the quality of the estimators are strongly affected by the choice of the smoothing factor. Parameter w needs to be chosen sufficiently large in order to ensure a smooth estimate. Nevertheless, even when w is too small, the adaptive width of the kernels can still be observed (cf. the wider bumps in the distribution tails).

Both the width scaling factor w and the number of prototypes M need to be optimized. As in the previous methods, w can be selected according to the leave-one-out cross-validation criterion:

$$\hat{E}_{\mathrm{LOO}}(w) = \frac{1}{M^2} \sum_{m=1}^{M} \sum_{m'=1}^{M} \mathcal{N}(\boldsymbol{\mu}_m | \boldsymbol{\mu}_{m'}, w^{-2}(\mathbf{S}_m + \mathbf{S}_{m'})^{-1}) - \frac{2}{M(M-1)} \sum_{m=1}^{M} \sum_{\substack{m'=1 \\ m' \neq m}}^{M} \mathcal{N}(\boldsymbol{\mu}_m | \boldsymbol{\mu}_{m'}, w^{-2}\mathbf{S}_{m'}{}^{-1}) \;, \tag{2.53}$$

where Gaussian kernels are used. The optimal width scaling factor is then found by exhaustive search:

$$\hat{w}_{\mathrm{LOO}} = \operatorname*{argmin}_{w} \hat{E}_{\mathrm{LOO}}(w) \;. \tag{2.54}$$



(a) $w = 5$ ($M = 15$). (b) $w = 1.5$ ($M = 15$).

FIGURE 2.11. VQ-based kernel estimator of a bi-modal density (mixture of two equally likely Gaussian distributions with different mean and standard deviation); the crosses indicate the position of the kernel prototypes after competitive learning. The resulting estimators globally overestimate the low density regions and underestimate the distribution modes both for (a) large and (b) small width scaling factors.

Remark that this criterion depends also on the meta-parameter M. It can therefore be used to select the optimal number of prototypes as well.

Magnification

It is reported by Hwang et al. (1994) that VQ-based kernel density estimators perform poorly in presence of outlying data. As a matter of fact, the problem is not limited to the presence of outliers. When using VQ methods, the original distribution is distorted according to the magnification factor:

$$p(\boldsymbol{\mu}) \propto p(\mathbf{x})^{\beta} \ , \tag{2.55}$$

where β is called the magnification. A magnification $\beta = 1$ corresponds to an information optimal coding of the observed data; unfortunately, β depends on the data dimension and the order of the minimized mean distortion error (Zador, 1982). Competitive learning for example minimizes the mean distortion error of order 2, which is equivalent to the mean square error (2.48). In this situation, we have $\beta < 1$ for competitive learning in general, leading to overestimated low-density regions (e.g., the distribution tails) and underestimated distribution modes, as illustrated in Figure 2.11. Thus, when estimating the underlying density, one should select the appropriate reconstruction error or conversely adapt the magnification to avoid additional distortion of the PDF. This means that an additional free parameter should be included (Bauer, Der and Herrmann, 1996), which is not feasible in practice. However, an alternative to lower this distortion consists in weighting the kernels as discussed next.

Weighted VQ-based kernel estimator

In order to reduce the effect of magnification, the kernels can be weighted according to the number of data points that are assigned to the kernel prototypes. Weighting the kernels has already been proposed by Babich and Camps (1996), but not motivated. The VQ-based estimator takes the following form:

$$p(\mathbf{x}|X,\boldsymbol{\theta}) = \frac{1}{Nw^d}\sum_{m=1}^{M}\frac{N_m}{|\mathbf{S}_m|^{1/2}}K\left(\mathbf{S}_m{}^{-1/2}\,\frac{\mathbf{x}-\boldsymbol{\mu}_m}{w}\right)\ , \tag{2.56}$$

where N_m is the number of data points assigned to prototype $\boldsymbol{\mu}_m$. Figure 2.12 shows the estimator of the bi-modal density considered in the previous paragraph. As usual, the choice of the smoothing factor is essential. However, the magnification is in this case clearly reduced and the functional form of the model is much closer to the target density, the model parameters being unchanged compared to Figure 2.11. Only non-uniform weights are introduced.

In contrast to what (2.56) may suggest, the complete learning set X does not need to be stored. As for the histograms, only the relative frequency associated to each cluster (or bin) is needed, allowing to view the weighted VQ-based estimator as a finite mixture model. Defining each kernel weight by

$$\pi_m = \frac{N_m}{Nw^d|\mathbf{S}_m|^{1/2}}\ , \tag{2.57}$$

equation (2.56) can be rewritten as follows:

$$p(\mathbf{x}|X,\boldsymbol{\theta}) = \sum_{m=1}^{M}\pi_m K\left(\mathbf{S}_m{}^{-1/2}\,\frac{\mathbf{x}-\boldsymbol{\mu}_m}{w}\right) = p(\mathbf{x}|\boldsymbol{\pi},\boldsymbol{\theta})\ , \tag{2.58}$$

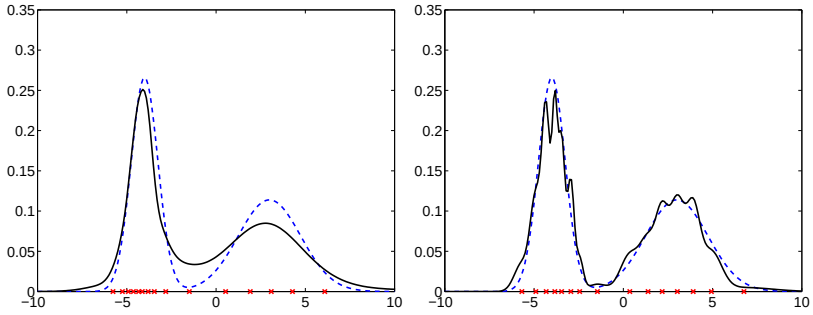
with $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{M}$.

Let us end this section by showing that the leave-one-out cross-validation is still applicable. In the case of Gaussian kernels the criterion becomes:

$$\begin{aligned}\hat{E}_{\mathrm{LOO}}(w) = &\sum_{m=1}^{M}\sum_{m'=1}^{M}\frac{N_m N_{m'}}{N^2}\mathcal{N}(\boldsymbol{\mu}_m|\boldsymbol{\mu}_{m'}, w^{-2}(\mathbf{S}_m+\mathbf{S}_{m'})^{-1})\\ &- 2\sum_{m=1}^{M}\sum_{\substack{m'=1\\ m'\neq m}}^{M}\frac{N_m N_{m'}}{N(N-N_m)}\mathcal{N}(\boldsymbol{\mu}_m|\boldsymbol{\mu}_{m'}, w^{-2}\mathbf{S}_{m'}{}^{-1})\ .\end{aligned} \tag{2.59}$$

2.3.4. Reduced Set Kernel Density Estimator

More recently, Girolami and He (2003) introduced the reduced set kernel density estimator (RSKDE). The underlying motivation of the method is to construct kernel density estimators based on an optimally condensed data set. This is only meaningful when the data scarcity is not an application constraint and the learning set is (very) large, leading to unacceptable memory usages. Furthermore, RSKDE can also be viewed as a kernel estimator which is locally



(a) $w = 5$ ($M = 15$). (b) $w = 1.5$ ($M = 15$).

FIGURE 2.12. Weighted VQ-based kernel estimator of the target bi-modal density of Figure 2.11; the crosses indicate the position of the kernel prototypes after competitive learning. The same prototypes are used as for the ordinary VQ-based estimators, as well as the same model complexity and width scaling factors. One can clearly observe from (a) and (b) that the magnification is compensated to a large extent compared to the Figures 2.11(a) and 2.11(b) when weighting the kernels.

adaptive. Indeed, if the multiplicative constant of each kernel is different, such as in Abramson's SKDE, this leads to different kernel widths.

Previous approaches for optimally condensing the data were based on vector quantization (see Section 2.3.3) or support vector machines (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000). The optimization of the support vector method for density estimation (Vapnik and Mukherjee, 1999) scales cubically with the size of learning set, whereas for RSKDE it scales only quadratically. In addition, the support vector method needs to set two parameters: the kernel width and the regularization parameter. The amount of regularization controls the tradeoff between sparsity and accuracy. By contrast, RSKDE does not require additional parameters to optimize, but the kernel width. As both techniques lead to a sparse representation of the original data and have a similar accuracy, the support vector method will not be further discussed.

Consider the kernel density estimator of the following form:

$$p(\mathbf{x}|X, \sigma, \boldsymbol{\pi}) = \frac{1}{\sigma^d} \sum_{n=1}^{N} \pi_n K\left(\frac{\mathbf{x} - \mathbf{x}_n}{\sigma}\right) , \tag{2.60}$$

where $\boldsymbol{\pi}$ denotes the set of weighting coefficients $\{\pi_n\}_{n=1}^N$. They are non-negative and must sum to one:

$$\forall n : \ \pi_n \geq 0 \ , \quad \sum_{n=1}^{N} \pi_n = 1 \ . \tag{2.61}$$

Subject to these constraints, it can be shown that the maximum likelihood estimator of the weights[4] results in the standard kernel density estimator (2.25) for a given kernel width σ:

$$\hat{\pi}_n = \underset{\pi_n}{\text{argmax}} \ \log \mathcal{L}(\boldsymbol{\theta}|X) \Rightarrow \hat{\pi}_n = \frac{1}{N} \ , \tag{2.62}$$

where parameter $\boldsymbol{\theta}$ denotes the model parameters $(\sigma, \pi_1, \ldots, \pi_N)$.

Instead, RSKDE minimizes the ISE. As shown below, this results in a sparse solution, as most of the weights are driven to zero. Note that this process can be viewed as a form of automatic model selection. Consider again the ISE as in KDE (2.31). Dropping the term that only depends on the target distribution leads to the following objective function:

$$E_{\text{RSKDE}} = \int p(\mathbf{x}|X, \boldsymbol{\theta})^2 d\mathbf{x} - 2\text{E}\{p(\mathbf{x}|X, \boldsymbol{\theta})\} \ . \tag{2.63}$$

As before, the first term in this equation can be computed exactly. The second term in contrast can be approximated as follows:

$$\text{E}\{p(\mathbf{x}|X, \boldsymbol{\theta})\} = \sum_{n=1}^{N} \pi_n \text{E}\left\{ \frac{1}{\sigma^d} K\left(\frac{\mathbf{x} - \mathbf{x}_n}{\sigma} \right) \right\} \approx \sum_{n=1}^{N} \pi_n p(\mathbf{x}_n|X, \sigma) \ . \tag{2.64}$$

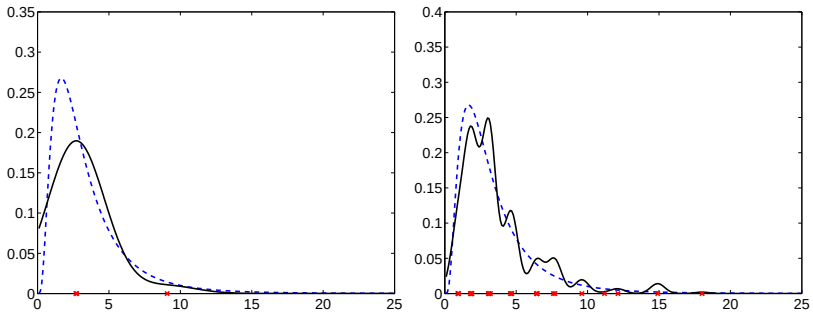
In this equation, $p(\mathbf{x}_n|X, \sigma)$ is the standard kernel density estimator as defined in (2.25) estimated at $\mathbf{x}_n$. If we assume isotropic Gaussian kernels, the objective function for a given kernel width becomes:

$$\begin{aligned} \hat{E}_{\text{RSKDE}}(\sigma) = &\sum_{n=1}^{N} \sum_{n'=1}^{N} \pi_n \pi_{n'} \mathcal{N}(\mathbf{x}_n|\mathbf{x}_{n'}, (2\sigma^2)^{-1}\mathbf{I}) \\ &- \frac{2}{N} \sum_{n=1}^{N} \sum_{n'=1}^{N} \pi_n \mathcal{N}(\mathbf{x}_n|\mathbf{x}_{n'}, \sigma^{-2}\mathbf{I}) \ . \end{aligned} \tag{2.65}$$

As discussed by Girolami and He (2003), the second term is sparsity inducing. Due to the summation constraint on the weights and since we maximize a convex combination of positive numbers, the second term is maximized by selecting a small number of points with small inter-point distance, i.e. in high density regions, and assinging them large weights. The minimum value of ISE is thus penalized by large inter-point distances in the kernel window. By contrast, the first term only causes the selection of points with high inter-point distances, as it has a constrained quadratic form. Therefore, the overall effect will be that points in regions of relatively high density will be selected to provide a smoothed density estimate.

Equation (2.65) can be written as a constrained quadratic optimization having simple positivity and equality constraints (Girolami and He, 2003). As a consequence, sequential minimal optimization (SMO) (Platt, 1999) can be used to

[4] Note that the maximum likelihood estimator of the weights corresponds to the minimal ANLL estimator.



(a) $\sigma = 2$. (b) $\sigma = 0.5$.

FIGURE 2.13. Effect of the choice of the kernel width on the optimally reduced Gaussian kernel density estimator of a log-normal distribution (dashed line). In (a) the kernel width is too large, in (b) it is too small. The data points having a non-zero weight are indicated by crosses.

optimize the weights. Recently, a multiplicative updating method for the non-negative quadratic programming of support vector machines was investigated (Sha, Saul and Lee, 2002). As for SMO, the updates have a simple closed form. However, the approach ensures a monotonic decrease of the weights at each iteration and all the quadratic programming variables can be adjusted in parallel, not just two at a time. While the multiplicative updating of the weights looks attractive for RSKDE, its convergence rate is much slower in practice.

Choosing the kernel width

The impact of the kernel width value on RSKDE is presented in Figure 2.13. Again, the choice of σ is crucial. In addition, for RSKDE the choice of σ has a direct impact on the number of weights that are non-zero, thus on the model complexity. Choosing its value based on (2.65) is delicate as the optimization procedure and its solution strongly depend on the specific data set used for learning. It is therefore advised to use another criterion to select σ, such as for example the ANLL.

2.4. Comparison of Kernel Density Estimators

In this section, the quality of the nonparametric PDF estimation techniques described so far are assessed. First, we study the impact of the amount of noise and the number of learning data on the estimation accuracy of the estimators. Artificially generated multivariate data are considered. Second, the performance of the PDF estimators is assessed on real data sets.

While the selection of the kernel width in the one dimensional case has extensively been discussed in the literature (see for example Park and Turlach, 1992; Cao, Cuevas and Manteiga, 1994; Farmen and Marron, 1999), very few was done in the multivariate case. One can mention the discussion of Hwang et al. (1994), who presented results on artificial generated data only. Therefore, we mainly focus on higher dimensional problems, as well as on real data. The performance of the methods is measured by computing the ANLL of the test set. In the experiments, we compare the following techniques:

(1) *Ordinary kernel density estimation* (KDE). The predictive distribution of KDE approximates the true density as follows:

$$p(\mathbf{x}) \approx p(\mathbf{x}|X, \sigma) \ . \tag{2.66}$$

The kernel width σ is selected by Scott's rule, the leave-one-out cross-validation criterion (minimizing the ISE), the 10-fold cross-validation criterion (minimizing the ANLL) or the .632 Bootstrap criterion (minimizing the ANLL).

(2) *Sample point kernel density estimation* (SKDE). Its predictive distribution is defined as

$$p(\mathbf{x}) \approx p(\mathbf{x}|X, \sigma, \boldsymbol{\lambda}) \ . \tag{2.67}$$

The optimal kernel width σ is selected by leave-one-out cross-validation, 10-fold cross-validation or .632 Bootstrap. Abramson's method is used, that is to say the sensitivity parameter α is set to $1/2$. The pilot density is constructed by KDE using Scott's rule for selecting its kernel width.

(3) *Weighted vector quantization-based kernel density estimation* (VQKDE). The predictive distribution is given by:

$$p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M, \mathbf{S}_1, \ldots, \mathbf{S}_M, w) \ . \tag{2.68}$$

The width scaling factor w is selected by the same techniques as for SKDE. For computational reasons, the number of prototypes is fixed in advance to 15% of the original learning set.

(4) *Reduced set kernel density estimation* (RSKDE). The predictive distribution is the following:

$$p(\mathbf{x}) \approx p(\mathbf{x}|X, \sigma, \boldsymbol{\pi}) \ . \tag{2.69}$$

Only 10-fold cross-validation and .632 Bootstrap criterion minimizing the ANLL are used to select the optimal kernel width σ. SMO is used for the optimization.

For all the above mentioned techniques, isotropic Gaussian kernels are used, even for VQKDE. In practice, the computation of the precisions associated to the Voronoi regions rapidly leads to numerical difficulties when the dimension of the feature space increases. Remark also that two methods are left aside: the histogram and M-NN. This is motivated by the fact that the first one is unpractical in high dimensional problems, while the second one does not satisfy

the basic constraints (2.1). As a result, the ANLL of the test set is meaningless in the case of the M-NN, as the estimator is not a true density.

2.4.1. Impact of the Amount of Noise

Considering Gaussian noise is of little interest as it results in estimators that are just flatter than the true density. This can be easily understood by noting that the PDF of the sum of two random variables is the convolution of the two PDFs. In this section, we rather analyze the impact of the proportion of atypical observation added to a training set of modest size (500 data points). In Figure 2.14, we report their effect on the quality of the estimators. The learning set was generated from a mixture of two multivariate Gaussian distributions:

$$p_{\mathcal{N}}(\mathbf{x}) = \pi_1 \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) + \pi_2 \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Lambda}_2) \ , \tag{2.70}$$

where $\pi_1 + \pi_2 = 1$. The mixture contains two overlapping components, resulting in a bi-modal distribution. The performance of the estimators in 2D, 5D and 7D are successively investigated. The parameters of the distributions are chosen as follows:

$$\begin{array}{ll} \pi_1 = 0.35 \ , & \pi_2 = 0.65 \ , \\ \boldsymbol{\mu}_1 = (0\ 0\ 0\ 0\ 0\ 0\ 0)^{\mathrm{T}} \ , & \boldsymbol{\mu}_2 = (2\ 2\ 2\ 2\ 2\ 2\ 2)^{\mathrm{T}} \ , \\ \boldsymbol{\Lambda}_1 = \mathrm{diag}(.4\ .7\ 1.5\ .9\ .65\ .8\ 1.2) \ , & \boldsymbol{\Lambda}_2 = \mathrm{diag}(.5\ 1.25\ .75\ 1\ 1.1\ .8\ .95) \ . \end{array}$$

In the 2D and 5D cases, we take respectively the first 2 and 5 elements of the parameters. The number of test points is 10,000. Due to their excessive computational cost, the leave-one-out cross-validation criteria are left out of the analysis. The atypical observations are generated from a uniform distribution ranging from -10 to 10 along each direction of the input space.

In 2D, all the methods perform similarly, except KDE using Scott's rule. As expected, the latter overestimates the kernel width when the distribution is multi-modal. Taking a closer look, we observe that ordinary KDE performs slightly worse than the adaptive techniques when the number of atypical observations is limited. Globally, RSKDE seems to perform the best.

In higher dimensions, the estimators behave differently. Clearly, VQKDE fails to provide good estimators. This is due to the fact that VQ methods are a type of least squares estimators, thus sensitive to outlying data. In addition, the curse of dimensionality increases this effect. The other methods can be ranked as follows RSKDE, SKDE and KDE (KDE using Scott's rule being again the worse).

In all cases and for all methods, the quality of the estimators decreases when the proportion of atypical observations increases, but the quality of the adaptive methods degrades slower. Besides, both 10-fold cross-validation and .632 bootstrap behaved similarly.

2.4.2. Effect of the Size of the Training Set

In this section, we consider multivariate mixtures of Gaussian and Cauchy distributions as in Hwang et al. (1994). As before, the mixtures contain two components, resulting in bi-modal distributions with overlapping components. The same parameters π_1, π_2, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ are used. A multivariate mixture of two Cauchy distributions is given by

$$p_{\mathcal{C}}(\mathbf{x}) = \pi_1 \mathcal{C}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) + \pi_2 \mathcal{C}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Lambda}_2) \ , \tag{2.71}$$

where $\pi_1 + \pi_2 = 1$. The multivariate Cauchy distribution has heavy tails compared to the Gaussian distribution and is defined as follows:

$$\mathcal{C}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \frac{\Gamma\left(\frac{d+1}{2}\right) |\boldsymbol{\Lambda}|^{1/2}}{\pi^{\frac{d+1}{2}}} \left[1 + (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Lambda} (\mathbf{x} - \boldsymbol{\mu})\right]^{-\frac{d+1}{2}} \ , \tag{2.72}$$

where $\Gamma(\cdot)$ denotes the gamma function. The Cauchy distribution is a particular case of the Student-t distribution, the degree of freedom being equal to 1. The ANLL of the test set (of size 10,000) versus the number of the learning data is reported in Figure 2.15.

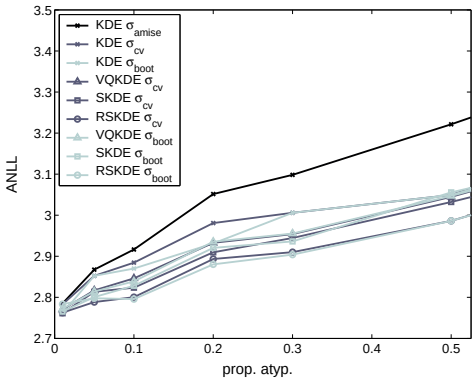
Let us first discuss the Gaussian mixture case. As expected, the quality of the estimates increases with the number of training data and degrades for increasing dimensionality. Again, it can be observed that VQKDE fails to provide good estimates in higher dimensions and that RSKDE slightly outperforms the other methods. For SKDE, the .632 bootstrap selects a smoothing factor that is too small, leading to poor generalization capabilities.

Next, consider the mixture of Cauchy distributions. Surprisingly, RSKDE performs very poorly. This is due to the fact that RSKDE provides very sparse solutions. As RSKDE is a weighted sum of Gaussian kernels, it performs well when the tails of the underlying distribution are not to thick. However, in general, it seems not to be capable of modeling accurately arbitrary densities. On the contrary, VQKDE and SKDE perform well. Overall, SKDE is the most flexible approach as it provides high quality estimates for both the mixture of Gaussian and Cauchy distributions.

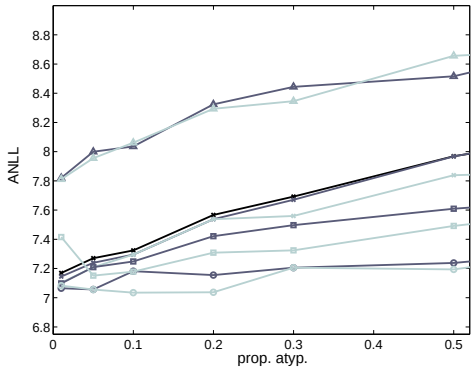
It is also worth to mention the comparative computational complexities of these techniques. Clearly, the iterative procedures (VQKDE and RSKDE) were found to be slower during the learning phase, while in the testing stage, SKDE is the slowest.

2.4.3. Assessing Real Data

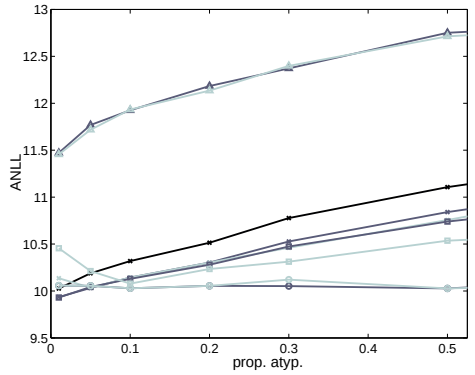
In this section, we assess the quality of the estimators on real data, the leave-one-out cross-validation criteria being also used. After briefly considering three univariate data sets, we focus on higher dimensional problems. The number of data is limited in all the examples. Whenever needed the data is first pre-processed by principal component analysis (PCA) (Jollife, 1986). Performing PCA consists in applying a linear transformation (rotation) to the coordinate



(a) Mixture of 2D Gaussians.

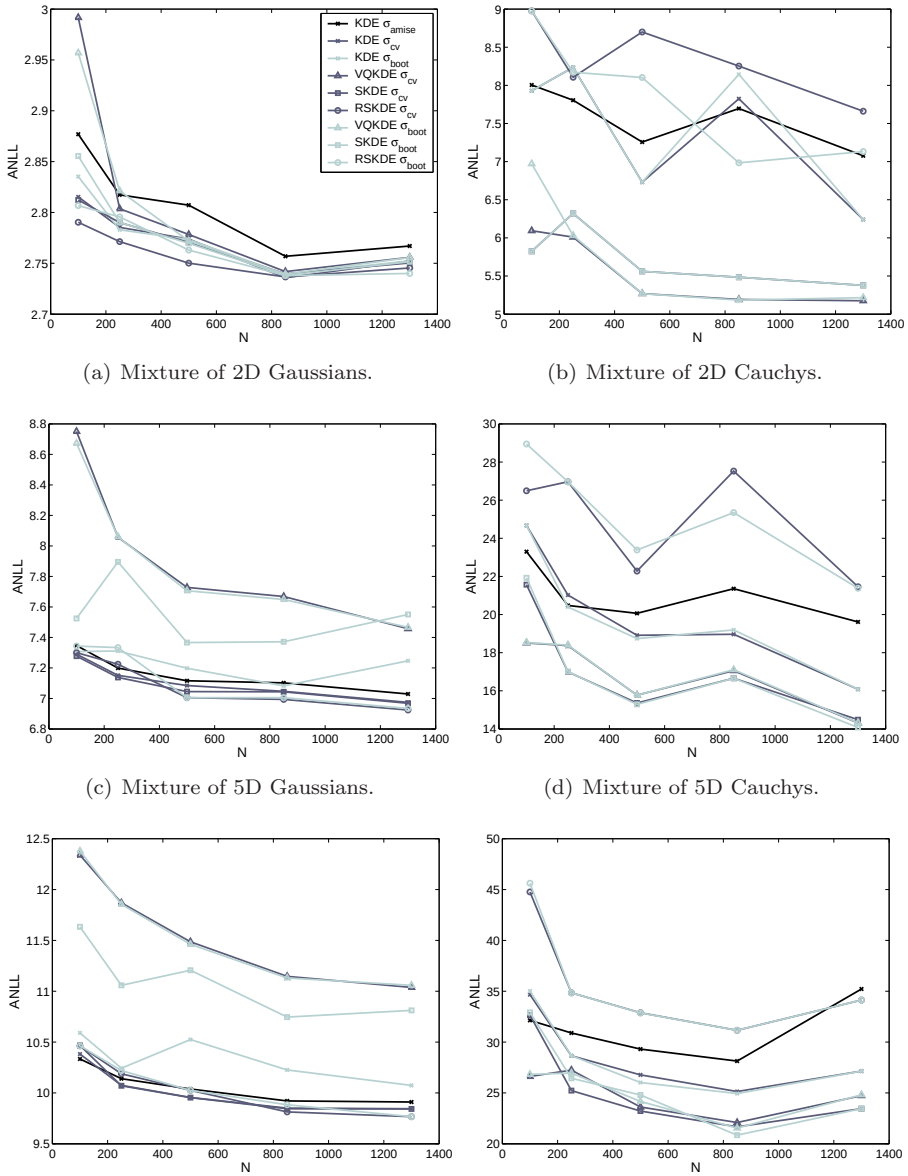


(b) Mixture of 5D Gaussians.



(c) Mixture of 7D Gaussians.

FIGURE 2.14. ANLL of the test set (10,000 data points) in presence of atypical observations. The number of learning data is 500. The proportion of atypical observations that is added ranges from 1% to 50% of the original size of the training set. See text for discussion.



(a) Mixture of 2D Gaussians.

(b) Mixture of 2D Cauchys.

(c) Mixture of 5D Gaussians.

(d) Mixture of 5D Cauchys.

(e) Mixture of 7D Gaussians.

(f) Mixture of 7D Cauchys.

FIGURE 2.15. ANLL of the test set (10,000 data points) in function of the number of learning data for mixture of Gaussian distribution and a mixture of Cauchy distribution, having the same parameters. See text for discussion.

axes, in order to find the principal directions of the data, i.e. the direction capturing maximal variance. Next, the directions capturing a sufficiently small portion of the total data variance can be discarded with a minimal loss of information. In addition, before constructing the models, the data is systematically sphered to avoid scaling problems. The data is first centered by subtracting the data mean and then divided by the data standard deviation.

The data sets that are considered are the univariate enzyme, acidity and galaxy data; the two dimensional ring, noisy spiral and the old faithful geyser data; the wine recognition data, of which the dimensionality can be reduced from 13D to 2D by PCA (99% of the total variance being kept); the NO^2 pollution data, of which the dimensionality can be reduced from 7D to 3D by PCA (99% of the variance being kept); the 4-dimensional iris plant data; the Boston housing data, of which the dimensionality can be reduced from 12D to 6D by PCA (99% of total variance being kept); the 6D liver disorder data; and the body fat data, of which the feature dimension can be reduced from 15D to 12D (again, 99% of the total variance being kept). In Appendix A, we further describe the data sets. Most of them are are available form the UCI Machine Learning repository (`http://www.ics.uci.edu/~mlearn`) or StatLib (`http://lib.stat.cmu.edu`).

In order to estimate how well the model represents the data, part of the learning data (20%) is kept aside for the test. The results are averaged over 20 runs, meaning that 20 random splits of the learning data into a training and a validation set are considered. The ANLL of the test set provides a natural criterion to compare the performance of the estimators, provided it is a true PDF. The optimal parameters are presented in Table 2.1 and the results are reported in Table 2.2 and 2.3.

First, we observe from Table 2.1 that the smoothing factor selected by Scott's rule ($\hat{\sigma}_{\text{AMISE}}$) is much larger than the ones selected by the other methods. This difference diminishes for increasing dimensionality. When the dimension increases the optimal kernel width increases as well, forcing more overlapping (note that the kernel widths in the tables are the ones after sphering the data). Remark also that the width scaling factors decrease. This parameter is thus only useful in feature spaces of modest dimension, suggesting that sufficiently overlap is enforced by weighting the kernel in higher dimensional feature spaces.

Next, consider Table 2.2 and 2.3. For each data set, the best estimator is underlined and the standard error is given. The following global performance index (PI) is used:

$$\text{PI}(i) = \frac{1}{J} \sum_{j=1}^{J} \frac{e_{ij}}{\min_{i'}(e_{i'j})} \geq 1 \ , \tag{2.73}$$

where $i \in \{1, \ldots, 12\}$ is the estimator label, $j \in \{1, \ldots, 12\}$ is the data set label and e_{ij} is either the corresponding test ANLL or its standard error. PI measures, on average, how estimator i performs compared to the optimal estimator on each data set. The closer PI is to 1, the better.

TABLE 2.1. The optimal model parameters. The values are averages over 20 runs. The data sets are ranked in ascending dimensionality.

	KDE				SKDE			VQKDE			RSKDE	
	$\hat{\sigma}_{\text{AMISE}}$	$\hat{\sigma}_{\text{LOO}}$	$\hat{\sigma}_{\text{CV}}$	$\hat{\sigma}_{\text{BOOT}}$	$\hat{\sigma}_{\text{LOO}}$	$\hat{\sigma}_{\text{CV}}$	$\hat{\sigma}_{\text{BOOT}}$	$\hat{w}_{\text{LOO}}$	$\hat{w}_{\text{CV}}$	$\hat{w}_{\text{BOOT}}$	$\hat{\sigma}_{\text{CV}}$	$\hat{\sigma}_{\text{BOOT}}$
Enzyme	0.37	0.06	0.12	0.12	0.07	0.09	0.10	10.5	6.6	7.7	0.12	0.11
Acidity	0.40	0.17	0.18	0.17	0.20	0.19	0.20	9.0	7.6	6.9	0.25	0.25
Galaxy	0.46	0.15	0.20	0.21	0.16	0.13	0.15	9.0	5.0	5.6	0.20	0.21
Ring	0.45	0.18	0.17	0.16	0.19	0.17	0.14	5.5	2.8	2.9	0.23	0.23
Spiral	0.41	0.07	0.07	0.06	0.08	0.07	0.06	5.0	2.0	2.4	0.08	0.08
Geyser	0.41	0.16	0.16	0.14	0.19	0.15	0.12	5.0	3.0	3.0	0.23	0.20
Wine	0.44	0.22	0.38	0.34	0.30	0.30	0.26	5.0	3.0	3.0	0.43	0.41
NO^2	0.41	0.22	0.27	0.23	0.26	0.25	0.18	4.4	3.0	3.0	0.37	0.33
Iris	0.52	0.17	0.23	0.20	0.16	0.21	0.17	3.0	1.1	1.4	0.20	0.24
Boston	0.51	0.11	0.28	0.23	0.12	0.22	0.17	3.0	1.0	1.0	0.15	0.35
Liver	0.53	0.31	0.56	0.45	0.26	0.56	0.37	3.0	1.1	1.1	0.38	0.39
Body Fat	0.66	0.68	0.67	0.63	0.68	0.78	0.52	3.0	1.0	1.0	0.78	0.73

TABLE 2.2. Average ANLL of the test set for KDE and SKDE for 20 runs. The standard errors are given between parentheses. The best of each line (out of Table 2.2 and 2.3) is underlined. The last line shows the performance indices. Again, the best of each line (across all estimators and smoothing selectors) is underlined. For each estimator, the best selector is bold as well.

	KDE				SKDE		
	$\hat{\sigma}_{\text{AMISE}}$	$\hat{\sigma}_{\text{LOO}}$	$\hat{\sigma}_{\text{CV}}$	$\hat{\sigma}_{\text{BOOT}}$	$\hat{\sigma}_{\text{LOO}}$	$\hat{\sigma}_{\text{CV}}$	$\hat{\sigma}_{\text{BOOT}}$
Enzyme	0.55 (.01)	0.47 (.11)	0.28 (.04)	0.27 (.04)	0.13 (.03)	0.14 (.03)	0.14 (.02)
Acidity	1.29 (.03)	1.25 (.05)	1.26 (.05)	1.30 (.08)	1.23 (.04)	1.26 (.05)	1.26 (.05)
Galaxy	2.67 (.03)	2.52 (.05)	2.55 (.06)	2.53 (.05)	2.55 (.05)	2.55 (.05)	2.55 (.05)
Ring	5.11 (.01)	4.85 (.03)	4.84 (.03)	4.86 (.04)	4.86 (.03)	4.84 (.03)	4.87 (.04)
Spiral	4.83 (.01)	3.67 (.01)	3.67 (.01)	3.69 (.02)	3.69 (.01)	3.69 (.01)	3.70 (.02)
Geyser	4.47 (.01)	4.19 (.02)	4.20 (.02)	4.21 (.02)	4.19 (.02)	4.18 (.02)	4.21 (.02)
Wine	2.65 (.03)	2.79 (.10)	2.64 (.04)	2.64 (.05)	2.60 (.04)	2.60 (.03)	2.62 (.04)
NO^2	3.93 (.01)	3.84 (.03)	3.82 (.02)	3.84 (.03)	3.80 (.02)	3.80 (.02)	3.85 (.03)
Iris	3.07 (.02)	2.41 (.17)	2.17 (.09)	2.20 (.10)	4.23 (1.18)	1.99 (.07)	2.09 (.10)
Boston	6.07 (.05)	11.04 (.98)	5.07 (.17)	5.15 (.24)	6.55 (.60)	4.34 (.14)	4.80 (.30)
Liver	22.16 (.09)	23.88 (.25)	22.15 (.08)	22.36 (.12)	25.03 (.38)	21.96 (.08)	22.68 (.20)
Body Fat	18.53 (.46)	18.38 (.42)	18.45 (.42)	18.93 (.54)	18.37 (.42)	18.03 (.34)	21.27 (.92)
	1.40 (1.78)	1.37 (5.88)	**1.12**(3.06)	**1.12**(3.76)	1.15 (8.80)	**1.01** (2.67)	1.04 (4.46)

TABLE 2.3. Average ANLL of the test set for VQKDE and RSKDE for 20 runs. The standard errors are given between parentheses. The best of each line (out of Table 2.2 and 2.3) is underlined. The last line shows the performance indices. Again, the best of each line (across all estimators and smoothing selectors) is underlined. For each estimator, the best selector is bold as well.

	VQKDE			RSKDE	
	$\hat{w}_{\text{LOO}}$	$\hat{w}_{\text{CV}}$	$\hat{w}_{\text{BOOT}}$	$\hat{\sigma}_{\text{CV}}$	$\hat{\sigma}_{\text{BOOT}}$
Enzyme	0.16 (.03)	0.16 (.03)	0.16 (.03)	0.25 (.05)	0.26 (.06)
Acidity	1.25 (.03)	1.24 (.03)	1.25 (.03)	1.27 (.06)	1.26 (.05)
Galaxy	2.67 (.04)	2.61 (.05)	2.60 (.05)	2.56 (.06)	2.56 (.07)
Ring	5.40 (.02)	5.06 (.03)	5.10 (.06)	4.92 (.04)	4.94 (.03)
Spiral	4.51 (.02)	4.05 (.03)	4.04 (.03)	3.77 (.02)	3.75 (.02)
Geyser	4.37 (.02)	4.21 (.02)	4.22 (.02)	4.23 (.03)	4.24 (.03)
Wine	2.79 (.02)	2.64 (.03)	2.64 (.03)	2.66 (.05)	2.66 (.05)
NO^2	4.32 (.04)	4.06 (.02)	4.05 (.01)	3.95 (.03)	3.94 (.04)
Iris	3.11 (.03)	2.58 (.15)	2.64 (.15)	2.45 (.15)	2.57 (.14)
Boston	6.28 (.04)	5.00 (.12)	4.98 (.11)	9.40 (.80)	13.08 (1.00)
Liver	22.66 (.04)	22.37 (.11)	22.29 (.10)	23.10 (.16)	23.20 (.17)
Body Fat	20.50 (.05)	18.02 (.32)	21.67 (1.90)	18.08 (.36)	18.18 (.36)
	1.17 (1.72)	**1.08** (3.03)	1.10 (5.83)	**1.21** (4.96)	1.29 (5.43)

As expected, Scott's rule performs the worse on average. This is not a surprising result as it is based on an (approximate) asymptotic rule, thus not taking the dispersion of the data into account. The leave-one-out (LOO) cross-validation is downward biased in many cases, meaning that it selects a smoothing factor which is in general too small. The resulting estimators have poor generalization capabilities. The standard errors are also much larger, reflecting the instability of the method. This is in agreement with previously published results (Park and Turlach, 1992; Cao et al., 1994).

For each estimator type, the 10-fold cross-validation and .632 bootstrap perform similarly. When compared to the LOO cross-validation criteria, they both perform better. In all situations, the PI of the standard error of 10-fold cross-validation is the smallest, suggesting that a smoothing selector based on this method is more attractive.

Now, comparing the estimators regardless of the resampling technique, one can see that VQKDE and especially SKDE perform well in practice. This emphasizes the importance of adapting the amount of smoothing through the feature space. RSKDE performs better than KDE using Scott's rule or the LOO cross-validation criterion, but worse than the 10-fold cross-validation and .632 bootstrap selectors. RSKDE was designed in the first place to reduce the size of the data set when performing nonparametric PDF estimation. In these examples, the number of data is limited. This may explain the poor results of RSKDE and in particular its relatively high standard error.

Recommendations for the practitioners

In practice, standard approaches should not be used blindly. For example, KDE using Scott's rule for selecting the kernel width often results in oversmoothing. When performing multivariate PDF estimation SKDE should be used, especially when the number of data is limited and in presence of outliers. The method clearly outperforms the other kernel estimators. In particular, using 10-fold cross-validation for selecting the amount of smoothing provides reliable results (low standard error), which are always close to the optimal in terms of the ANLL.

It was also shown that weighted VQKDE performs quite well in many cases, but it may be sensitive to (strong) outliers. A solution is to remove them in some way, before modeling the PDF. However, this would require to choose an additional parameter that is difficult to set in practice. The popular LOO cross-validation criteria do not provide reliable results and select a smoothing factor that leads to overfitting in many cases. In addition, we should underline the fact that the computational complexity of the methods becomes unacceptable for learning sets of more than 1,000 data points.

Finally, RSKDE performs similarly as ordinary KDE and should therefore only be used when the size of the learning set should be reduced. Note that training

RSKDE may be nevertheless time consuming in the beginning of the training phase, as many of the weights are not yet driven to zero.

2.5. Summary

In this chapter, a non-exhaustive list of nonparametric PDF estimators based on kernels is reviewed. More specifically, we discussed in detail the advantages and drawbacks of ordinary kernel density estimation (KDE), sample point kernel density estimation (SKDE), vector quantization-based kernel density estimation (VQKDE) and reduced set kernel density estimation (RSKDE). For each technique, several smoothing selectors were investigated. In particular, we focused on the ones that can be used for multivariate problems, which are hardly addressed in the literature. In this context, it was proposed to use the average negative log-likelihood as performance measure. It was shown that, when used with adequate statistical resampling techniques, this (conventional but) general methodology provides satisfactory results.

The well known leave-one-out cross-validation criterion for KDE was also extended to the adaptive SKDE and VQKDE. In addition, it was explained why the standard VQKDE does not work well due to the magnification, and how this effect can be reduced by weighting the kernels, thus providing an a posteriori justification of Babich and Camps' method (Babich and Camps, 1996). More importantly, the form of the weighted VQKDE motivates is very similar to finite mixture models. One cant therefore think of using the latter for estimating arbitrary densities. This will be extensively discussed in the next chapter.

The quality of the methods were assessed through extensive simulations. The main result of this comparative study is that adaptive estimators outperform the commonly used KDE with a fixed smoothing. In particular, SKDE is the method of predilection, especially when dealing with data sets of modest size. Its main drawback is its model complexity, which increases linearly with the size of the learning set. If memory resources are a limiting constraint, one should move to either weighted VQKDE, paying attention to outliers, or possibly to RSKDE.

Finally, when using the ANLL as performance measure for selecting the amount of smoothing, 10-fold cross-validation behaves well. On the one hand, it is a stable method (low standard errors) and, on the other hand, it is less biased (best generalization performance on average) than the other techniques. In addition, its computational complexity is smaller than the leave-one-out criteria and the .632 bootstrap.

CHAPTER 3

Finite Mixture Models

To be able to model arbitrary probability density functions is of common interest in many scientific domains. Density estimators are fundamental tools for extracting the information embedded in raw data. In the previous chapter, nonparametric kernel density estimators were reviewed. Starting with the ordinary kernel density estimator (KDE) with fixed smoothing, we moved on to variants allowing adaptive smoothing. In general, these techniques lead to models of higher quality or, at least, show a satisfactory accuracy for a much smaller model complexity. Unfortunately, they are also sensitive to outliers and have often many parameters to set.

An alternative to nonparametric methods are finite mixture models (Redner and Walker, 1984; McLachlan and Peel, 2000). As nonparametric techniques, they do not assume a priori the overall shape of the PDF to estimate. Mixture models are based on a divide-and-conquer approach, which means that subpopulations of the observed data are modeled by parametric distributions, while the resulting PDF is often far from any standard parametric form. Unlike the nonparametric methods, the complexity of the model is fixed in advance, avoiding a prohibitive increase of the number of parameters with the size of the data set.

In contrast to the traditional view of mixture models as being clustering tools, these techniques are also suitable for a more general purpose: nonparametric-like PDF estimation (Bishop, 1995). Even if subpopulations cannot be identified within the data, mixture models can still be used. As a matter of fact, we may consider finite mixture models as an extreme case of adaptively smoothed KDE. More specifically, they can be interpreted as the probabilistic version of the weighted vector quantization-based kernel density estimator. The frontier between nonparametric and finite mixture models is thus quite vague, especially when considering methods that use locally adaptive smoothing techniques. Based on the same considerations, Scott and Szewczyk (2001) proposed to build mixture models explicitly from nonparametric estimators. Another closely related approach is Priebe's adaptive mixtures (Priebe, 1994), which are based on the method of sieves.

In this chapter, finite mixture models are discussed in detail. Since they can be viewed as latent variable models, we first describe how to learn this type of models in general. Next, the methodology is applied to both finite Gaussian

mixture models and finite Student-t mixture models. The Student-t distribution provides a robust alternative to the Gaussian distribution and is particularly useful in noisy environment. Finally, manifold constrained mixture models are introduced. Whenever possible, this variant exploits the fact that the data manifold is of a lower dimension than the dimension of the feature space. Intuitively, one can picture a manifold as follows. Consider data points lying on a sheet of paper, which is folded in a 3D space. Even though the points are located in the 3D space, they are also lying on a 2D manifold (i.e., the sheet of paper). In practice, we can take advantage of this additional information to enhance the quality of the estimators.

3.1. Learning Latent Variable Models

In this section, we present how to learn the parameters of hidden or latent variables models, such as finite mixture models. Although they cannot be observed, latent variables may either interact through the model parameters in the data generation process, or are just mathematical artifacts that are introduced into the model in order to simplify it in some way. The expectation-maximization (EM) algorithm (Baum, Petrie, Soules and Weiss, 1970; Dempster, Laird and Rubin, 1977) and its extensions (McLachlan and Krishnan, 1997) are particularly suited for learning this type of models (see for example Titterington, 1984; McLachlan and Bashford, 1988; Jordan and Jacobs, 1994). In general, and more particularly in the Bayesian setting, it is convenient to formalize latent variable models as graphical models.

Probabilistic graphical models provide a general methodology for handling statistical problems involving (a large number of) random variables that are linked with each other in a complex way (Jordan, 2004). Probability distributions are defined in terms of directed or undirected graphs, called respectively Bayesian networks (Pearl, 1988) and Markov random fields (Kindermann and Snell, 1980). The nodes are identified with random variables and the joint probability distributions are defined by taking products of functions defined on connected subsets of nodes. For example, in directed acyclic graphs (Bayesian networks), an edge denotes the conditional dependency of the child node on its parent node and the joint probability of $Z = \{\mathbf{z}_n\}_{n=1}^N$ is defined as the product of the conditional probabilities of each variable given the set of its parents:

$$p(Z) = \prod_{n=1}^{N} p(\mathbf{z}_n | \mathbf{z}_{\mathbf{pa}(n)}) \ , \tag{3.1}$$

where $\mathbf{z}_{\mathbf{pa}(n)}$ is the set of parents of node $\mathbf{z}_n$. Each node is thus conditionally independent from its non-descendents (ancestors) given its parents. This conditional relationship allows us to represent the joint distribution more compactly. Consider for instance the Bayesian network shown in Figure 3.1. The joint distribution $p(z_\mathrm{M}, z_\mathrm{P}, z_\mathrm{R}, z_\mathrm{E})$ can be expanded using the product rule:

$$p(z_\mathrm{M}, z_\mathrm{P}, z_\mathrm{R}, z_\mathrm{E}) = p(z_\mathrm{M}|z_\mathrm{P}, z_\mathrm{R}, z_\mathrm{E}) p(z_\mathrm{P}|z_\mathrm{R}, z_\mathrm{E}) p(z_\mathrm{R}|z_\mathrm{E}) p(z_\mathrm{E}) \ . \tag{3.2}$$

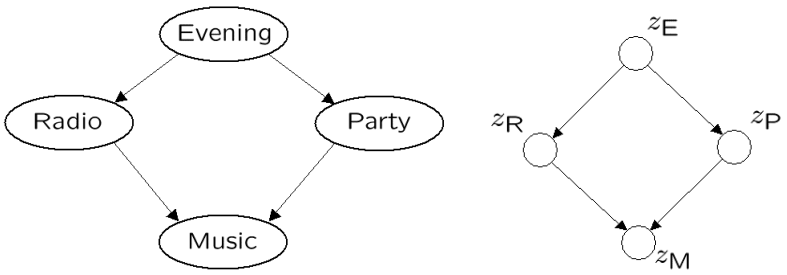


FIGURE 3.1. Example of a Bayesian network. The nodes are binary random variables (True/False). The arrows indicate conditional dependencies.

Using the conditional independence relationships, leads to the following expression:

$$p(z_M, z_P, z_R, z_E) = p(z_M|z_P, z_R)p(z_P|z_E)p(z_R|z_E)p(z_E) \ . \tag{3.3}$$

The first factor in (3.2) can be simplified as z_M is independent of z_E given its parents z_P and z_R. Similarly, the second one can be simplified as z_P is independent of z_R given its parent z_E.

The attractiveness of graphical models comes from their graph-theoretic representation, which provides general algorithms for computing marginal and conditional probabilities of interest. The three principal classes of Bayesian inference tools are first, exact algorithms, e.g. belief propagation (Pearl, 1986; Spiegelhalter, 1986); second, sampling algorithms, e.g. Markov Chain Monte-Carlo (Metropolis, Rosenbluth, Rosenbluth, Teller and Teller, 1953; Hastings, 1970), Gibbs sampling (Geman and Geman, 1984), rejection sampling (Gilks and Wild, 1992) and slice sampling (Neal, 2003); third, approximate algorithms, e.g. variational algorithms (Hinton and van Camp, 1993; Waterhouse, MacKay and Robinson, 1995; Jaakkola, 1997; Jordan, Ghahramani, Jaakkola and Saul, 1999; Beal, 2003) and loopy belief propagation (Heskes, 2002; Yedidia, Freeman and Weiss, 2003). Unlike the variational methods, which will be discussed in depth, the other graph-theoretic algorithms will not be further discussed in this work. For a detailed presentation of these methods we refer the interested reader to the books of Gilks, Richardson and Spiegelhalter (1996), Cowell, Dawid, Lauritzen and Spiegelhalter (1999) and Jordan (1999). Software implementations of these algorithms, such as WinBUGS (Lunn, Thomas, Best and Spiegelhalter, 2000) and VIBES (Winn and Bishop, 2005), are also available.

Maximum likelihood (ML) provides good estimators in large learning set settings, i.e. when asymptotic analyzes are meaningful. The underlying idea of ML is to maximize the joint probability (or likelihood) of the observations $X = \{\mathbf{x}_n\}_{n=1}^N$ in order to find the optimal model parameters. These parameters specify the specific model of which the functional form is assumed a priori. The probability of a new datum is then predicted on the basis of the parameters

$\boldsymbol{\theta}_{\mathrm{ML}}$, which are optimal in terms of likelihood:

$$p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\theta}_{\mathrm{ML}}) \ . \tag{3.4}$$

For small learning sets however, adding a penalty (or regularization) improves the ML estimate. The resulting estimate is termed maximum a posteriori (MAP) estimate. The goal in MAP is to penalize unrealistic values of the parameters by a prior on them. The likelihood is multiplied by the prior. Maximizing this new quantity, which corresponds to the posterior distribution of the parameters $p(\boldsymbol{\theta}|X)$ up to a normalizing constant, leads to the MAP estimate. The probability of a new datum is predicted as follows:

$$p(\mathbf{x}) \approx p(\mathbf{x}|\boldsymbol{\theta}_{\mathrm{MAP}}) \ , \tag{3.5}$$

where $\boldsymbol{\theta}_{\mathrm{MAP}}$ are the optimal parameters in terms of penalized likelihood.

In the Bayesian setting, the uncertainty on the model parameters is taken into account in a principled way. While ML or MAP only provide point-estimates of the model parameters, in the Bayesian framework, predictions are made by means of model averaging:

$$p(\mathbf{x}) \approx \int p(\mathbf{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|X) d\boldsymbol{\theta} \ , \tag{3.6}$$

where $p(\boldsymbol{\theta}|X)$ is the posterior distribution of the parameters given the observations. The predictions are thus made by a weighted sum of the predictions of all possible models (within the chosen family) and the weighting coefficients are given by the posterior distribution of the parameters. Unfortunately, model averaging involves usually the computation of intractable integrals and therefore approximate methods such as variational Bayes are needed.

In the remaining of this section, the general principle of ML, MAP and Bayesian learning is described. We focus on the particular case of latent variable models and discuss the EM algorithm and its variants, which provide elegant solutions to these learning problems.

3.1.1. Maximum Likelihood Learning

Consider a set of i.i.d. variables $X = \{\mathbf{x}_n\}_{n=1}^N$ that were generated using a set of hidden variables $Z = \{\mathbf{z}_n\}_{n=1}^N$. For a particular model $\mathcal{H}_M$ of complexity M, the data likelihood is defined as a function of the parameters $\boldsymbol{\theta}$ of $\mathcal{H}_M$:

$$\mathcal{L}(\boldsymbol{\theta}|X) \equiv p\left(X|\boldsymbol{\theta}\right) = \prod_{n=1}^{N} \int p\left(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}\right) d\mathbf{z}_n \ , \tag{3.7}$$

where the hidden variables are assumed to be continuous. The integration (marginalization) over the hidden variables is required to form the likelihood as a function of the observed data only. Note that when the hidden variables are discrete the integral is replaced by a sum.

Maximizing the data likelihood, or equivalently its logarithm, with respect to $\boldsymbol{\theta}$ results in the maximum likelihood parameters:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \log \mathcal{L}(\boldsymbol{\theta}|X) \ . \tag{3.8}$$

In the absence of hidden variables, maximizing this expression is straightforward. However, if some variables are hidden, the maximization problem becomes difficult as the integral appears inside the logarithm and is in many practical problems intractable:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \sum_{n=1}^{N} \log \int p\left(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}\right) d\mathbf{z}_n \ . \tag{3.9}$$

By introducing a distinct arbitrary auxiliary distribution $q_{\mathbf{z}_n}(\mathbf{z}_n)$ over each hidden variable, a lower bound on the log-likelihood is obtained using Jensen's inequality (Jensen, 1906):

$$\begin{aligned}
\log \mathcal{L}(\boldsymbol{\theta}|X) &= \log \int p(X, Z|\boldsymbol{\theta}) dZ && (3.10)\\
&= \log \int q_Z(Z) \tfrac{p(X,Z|\boldsymbol{\theta})}{q_Z(Z)} dZ && (3.11)\\
&\geq \int q_Z(Z) \log \tfrac{p(X,Z|\boldsymbol{\theta})}{q_Z(Z)} dZ && (3.12)\\
&\equiv \mathcal{F}\left(q_{\mathbf{z}_1}(\mathbf{z}_1), \ldots, q_{\mathbf{z}_N}(\mathbf{z}_N), \boldsymbol{\theta}\right) \ . && (3.13)
\end{aligned}$$

where

$$q_Z(Z) = \prod_{n=1}^{N} q_{\mathbf{z}_n}(\mathbf{z}_n) \ . \tag{3.14}$$

This equality follows from the fact that the data X are i.i.d.

The expectation-maximization (EM) algorithm, which was formalized by Dempster et al. (1977), can be understood as an iterative procedure for maximizing this lower bound. The bound $\mathcal{F}$ can be identified as Helmholtz' negative free-energy from statistical physics (Neal and Hinton, 1998). Defining the complete data likelihood

$$\mathcal{L}_c(\boldsymbol{\theta}|X, Z) \equiv p(X, Z|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}) \tag{3.15}$$

as opposed to the incomplete data likelihood $\mathcal{L}(\boldsymbol{\theta}|X)$, we can decompose (3.12) as follows:

$$\mathcal{F}\left(q_{\mathbf{z}_1}(\mathbf{z}_1), \ldots, q_{\mathbf{z}_N}(\mathbf{z}_N), \boldsymbol{\theta}\right) = \mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}|X, Z)\} + \mathrm{H}(q_Z(Z)) \ , \tag{3.16}$$

where the expectation is taken with respect to $q_Z(Z)$ and where $\mathrm{H}(\cdot)$ is Shannon's or the differential entropy for respectively discrete or continuous random variables (Cover and Thomas, 1991). Successively maximizing the lower bound

with respect to $q_Z(Z)$ while keeping $\boldsymbol{\theta}$ fixed, and then with respect to $\boldsymbol{\theta}$ while keeping $q_Z(Z)$ fixed, results in the EM update equations:

$$\textbf{E-step} \;: q_{\mathbf{z}_n}(\mathbf{z}_n) \leftarrow \operatorname*{argmax}_{q_{\mathbf{z}_n}(\mathbf{z}_n)} \mathcal{F}\left(q_{\mathbf{z}_1}(\mathbf{z}_1), \ldots, q_{\mathbf{z}_N}(\mathbf{z}_N), \boldsymbol{\theta}\right) \;, \quad \forall n \;. \tag{3.17}$$

$$\textbf{M-step} : \boldsymbol{\theta} \leftarrow \operatorname*{argmax}_{\boldsymbol{\theta}} \mathcal{F}\left(q_{\mathbf{z}_1}(\mathbf{z}_1), \ldots, q_{\mathbf{z}_N}(\mathbf{z}_N), \boldsymbol{\theta}\right) \;. \tag{3.18}$$

The functional maximization problem in the E-step is easily obtained (when tractable) by observing that the bound is made tight when equating each $q_{\mathbf{z}_n}(\mathbf{z}_n)$ to the posterior distribution of its corresponding latent variable:

$$\textbf{E-step} : q_{\mathbf{z}_n}(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta}) \;, \quad \forall n \;. \tag{3.19}$$

This result is straightforward when considering an alternative decomposition of (3.12):

$$\begin{aligned} &\mathcal{F}\left(q_{\mathbf{z}_1}(\mathbf{z}_1), \ldots, q_{\mathbf{z}_N}(\mathbf{z}_N), \boldsymbol{\theta}\right) \\ &\quad = \log \mathcal{L}(\boldsymbol{\theta}|X) - \mathrm{KL}[q_Z(Z) \| p(Z|X, \boldsymbol{\theta})] \qquad (3.20) \\ &\quad = \log \mathcal{L}(\boldsymbol{\theta}|X) - \textstyle\sum_{n=1}^{N} \mathrm{KL}[q_{\mathbf{z}_n}(\mathbf{z}_n) \| p(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta})] \;. \qquad (3.21) \end{aligned}$$

In these equations, KL denotes the Kullback-Leibler divergence (Kullback and Leibler, 1951). The E-step is illustrated in Figure 3.2. When the exact posterior is intractable, approximate EM is required; see for example Heskes, Zoeter and Wiegerinck (2003), where the exact free-energy is approximated by a Bethe-Kikuchi free-energy, leading to an approximate E-step.

In contrast to (3.9), the maximization problem in the M-step can be computed explicitly in many cases. As the logarithm appears inside the integral and the entropy term is independent of $\boldsymbol{\theta}$ we have:

$$\textbf{M-step} : \boldsymbol{\theta} \leftarrow \operatorname*{argmax}_{\boldsymbol{\theta}} \mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}|X, Z)\} \;. \tag{3.22}$$

The M-step is illustrated in Figure 3.2. When no closed form solution to the M-step exists, it is sufficient to chose the M-step in such a way that it ensures an increase in log-likelihood at each iteration rather than maximizing it. This is known as Generalized EM (Dempster et al., 1977).

Given initial parameters, applying successively the E- and the M-step provides an estimate of $\boldsymbol{\theta}_{\mathrm{ML}}$ corresponding to a local maximum of the likelihood surface. Let us denote this estimate by $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$. The predictive distribution in the maximum likelihood setting is then:

$$p(\mathbf{x}) \approx p(\mathbf{x}|\hat{\boldsymbol{\theta}}_{\mathrm{ML}}) \;. \tag{3.23}$$

The attractive property of the EM algorithm is its monotonic increase in likelihood at each iteration. During the E-step, the lower bound is made tight and in the M-step the expected energy is maximized with respect to the model parameters, keeping $q_Z(Z)$ fixed. However, as the likelihood is unbounded, its maximization may be ill-posed when the number of training data is small. Therefore, maximum penalized likelihood learning will be considered next.

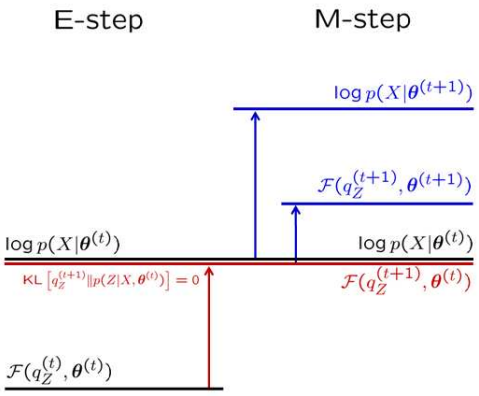


FIGURE 3.2. The EM algorithm maximizes iteratively the negative free-energy $\mathcal{F}$. In the E-step, the bound is made tight by equating $q_Z(Z)$ to the posterior probability of the latent variables. The current estimates of the parameters are fixed. In the M-step, the bound is maximized with respect to the parameters, while $q_Z(Z)$ is fixed. As a result, a new bound is obtained, as well as an updated incomplete data log-likelihood.

3.1.2. Maximum a Posteriori Learning

Regularization techniques (Tikhonov and Arsenin, 1977; Chen and Haykin, 2002) are powerful in making ill-posed problems well-posed, the penalized likelihood being a particular case among many others (Green, 1999). The use of the EM algorithm for maximum penalized likelihood or maximum a posteriori (MAP) estimation was investigated by Green (1990). The overall effect of regularization is to smooth the model, avoiding therefore overfitting. Moreover, adding a penalization term to the objective function (in this case the log-likelihood) usually makes it more concave.

Consider the prior distribution $p(\boldsymbol{\theta})$ on the model parameters $\boldsymbol{\theta}$ of $\mathcal{H}_M$, reflecting our prior knowledge on them. It is for instance common to have prior information on the range of $\boldsymbol{\theta}$. Applying Bayes' rule allows us to update our prior belief about the model parameters to a posterior belief (up to a normalizing constant) having observed the data X:

$$p(\boldsymbol{\theta}|X) \propto p(X|\boldsymbol{\theta})p(\boldsymbol{\theta}) \ . \tag{3.24}$$

The MAP estimate of the parameters is then obtained by maximizing the log-posterior distribution of the parameters:

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \ \log p(\boldsymbol{\theta}|X) \tag{3.25}$$

$$= \operatorname*{argmax}_{\boldsymbol{\theta}} \ \log \mathcal{L}(\boldsymbol{\theta}|X) + \log p(\boldsymbol{\theta}) \ . \tag{3.26}$$

Note that the quantity $\mathcal{L}(\boldsymbol{\theta}|X)$ is the incomplete data likelihood, which is defined in (3.7).

In the presence of latent variables, the same difficulty as in ML learning arises. Luckily, the EM algorithm is still applicable (Green, 1990). Since the penalty term only depends on $\boldsymbol{\theta}$, the E-step is unchanged. This can be easily understood by seeing that the incomplete log-posterior can still be lowerbounded using Jensen's inequality. In contrast, the M-step is augmented to:

$$\textbf{M-step} : \boldsymbol{\theta} \leftarrow \operatorname*{argmax}_{\boldsymbol{\theta}} \mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}|X, Z)\} + \log p(\boldsymbol{\theta}) \ . \tag{3.27}$$

Denoting the estimate of $\boldsymbol{\theta}_{\mathrm{MAP}}$ provided by the EM algorithm by $\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}$, the predictive distribution in MAP learning is:

$$p(\mathbf{x}) \approx p(\mathbf{x}|\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}) \ . \tag{3.28}$$

A practical MAP approach

Apart from the choice of the type of prior on the parameters, the main practical problem in MAP learning is the choice of the hyperparameters, i.e. the parameters of the priors. Choosing them a priori does not provide satisfactory results, as this may lead to significant biases in the estimators. A straightforward approach is to optimize these hyperparameters in a more conventional way, for example by means of resampling techniques. However, most priors depend on more than one parameter. As a result, the optimization procedure becomes rapidly infeasible for computational reasons. Instead, it is suggested to use the following practical approach.

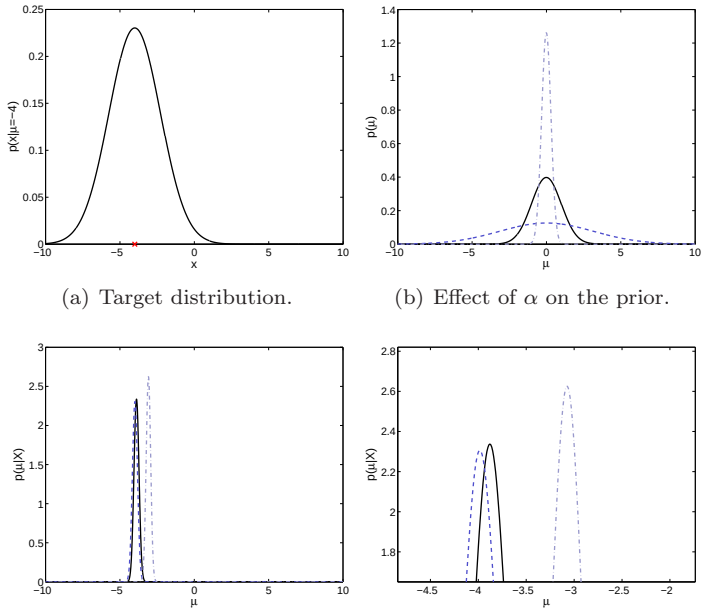
Let $\boldsymbol{\vartheta}$ be the hyperparameters. Introducing $\boldsymbol{\vartheta}$ in Bayes' rule leads to:

$$p(\boldsymbol{\theta}|X) \propto p(X|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\vartheta}) \ , \tag{3.29}$$

where X is assumed to be independent of $\boldsymbol{\vartheta}$ given $\boldsymbol{\theta}$. Next, instead of optimizing $\boldsymbol{\vartheta}$, we make an explicit choice $\boldsymbol{\vartheta}^*$ for $\boldsymbol{\vartheta}$ according to some prior belief on the problem. For instance, when imposing a prior on the covariance matrix of a Gaussian distribution, we may assume it should be (approximately) diagonal. However, as it is not known to which extent this belief is true, an additional learning parameter $\alpha > 0$ (to be optimized in a standard way) is introduced. The new prior is then defined as follows:

$$p(\boldsymbol{\theta}|\alpha) \propto p(\boldsymbol{\theta}|\boldsymbol{\vartheta}^*)^\alpha \ . \tag{3.30}$$

Adjusting α allows us to temper our prior belief whenever needed, as well as to reinforce it. Setting $\alpha < 1$ results in a prior that is flatter, thus less informative. By contrast, setting $\alpha > 1$ leads to a prior that is more peaked, strengthening our prior belief. This is illustrated in Figure 3.3(b) for a Gaussian prior on the mean of a specific target distribution. Subfigures (c) and (d) show how the choice of α affects the posterior on the mean. Clearly, adjusting α improves its quality.



(a) Target distribution. (b) Effect of α on the prior.

(c) Effect of α on the posterior. (d) Zoom on the posterior.

FIGURE 3.3. Effect of parameter α on the shape of the prior and the corresponding posterior. (a) shows the target distribution, its mean being equal to -4. The mean is assumed unknown. (b) shows a specific prior on the mean raised to the factor $\alpha = 0.1$ (dashed), $\alpha = 1$ (solid) and $\alpha = 10$ (dash-dotted). (c) shows the resulting posterior distributions and (d) is a zoom on the posteriors in the vicinity of the true mean.

Assuming a posterior distribution of the form (3.30) leads to the following modified maximum a posteriori estimate for the parameters:

$$\boldsymbol{\theta}_{\text{PMAP}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \ \log \mathcal{L}(\boldsymbol{\theta}|X) + \log p(\boldsymbol{\theta}|\alpha) \tag{3.31}$$

$$= \underset{\boldsymbol{\theta}}{\text{argmax}} \ \log \mathcal{L}(\boldsymbol{\theta}|X) + \alpha \log p(\boldsymbol{\theta}|\boldsymbol{\vartheta}^*) \ . \tag{3.32}$$

The form of this expression resembles much more the formulation of standard regularized models, which are commonly used in the field of machine learning (see for example Bishop, 1995).

The resulting M-step is defined as follows:

$$\textbf{M-step} : \boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}}{\text{argmax}} \ \text{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}|X, Z)\} + \alpha \log p(\boldsymbol{\theta}|\boldsymbol{\vartheta}^*) \ . \tag{3.33}$$

The modified MAP approach presented above is easily extended when the prior factorizes. Assume for example a prior of the following form:

$$p(\boldsymbol{\theta}|\boldsymbol{\vartheta}) = \prod_{k=1}^{K} p(\boldsymbol{\theta}_k|\boldsymbol{\vartheta}_k) \ , \tag{3.34}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ and $\boldsymbol{\vartheta} = (\boldsymbol{\vartheta}_1, \ldots, \boldsymbol{\vartheta}_K)$. When choosing particular values for $\boldsymbol{\vartheta}$ and introducing the learning vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ in order to adjust our prior belief, the following maximization problem is obtained:

$$\boldsymbol{\theta}_{\text{PMAP}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \ \log \mathcal{L}(\boldsymbol{\theta}|X) + \sum_{k=1}^{K} \alpha_k \log p(\boldsymbol{\theta}_k|\boldsymbol{\vartheta}_k^*) \ , \tag{3.35}$$

which can still be solved by means of the EM algorithm for a given $\boldsymbol{\alpha}$. Observe that since $\boldsymbol{\alpha}$ is optimized by means of resampling techniques, a moderate K is mandatory.

To conclude this section, note that an undesirable property of MAP learning is the dependance on the parametrization of the prior (Beal, 2003; Winn, 2003). In other words, the MAP approaches are basis-dependent, meaning that it is always possible to find a basis in which any particular $\boldsymbol{\theta}^*$ is the MAP solution (provided it has non-zero prior probability). This is not the case for Bayesian learning, which is described next.

3.1.3. Bayesian Learning

Although the posterior distribution of the parameters is used in MAP learning, predictions are still performed based on point-estimates. Therefore, MAP learning does not properly deal with the uncertainty on the parameters. In Bayesian learning, these parameters are treated as (latent) random variables. The uncertainty on the parameters is better taken into account by using their posterior distribution for constructing the predictive distribution:

$$p(\mathbf{x}) \approx p(\mathbf{x}|X) = \int p(\mathbf{x}|X, \boldsymbol{\theta}) p(\boldsymbol{\theta}|X) d\boldsymbol{\theta} = \int p(\mathbf{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|X) d\boldsymbol{\theta} \ , \tag{3.36}$$

where it is assumed that the prediction $\mathbf{x}$ is independent of X given $\boldsymbol{\theta}$. The parameters are viewed as nuisance parameters and are thus integrated out of the predictive distribution, i.e. a weighted average is performed according to the posterior distribution of the parameters. Therefore, Bayesian learning is also known as model averaging or ensemble learning (see for example MacKay, 1992b).

Bayesian learning consists of two stages: model fitting and model selection. In the maximum (penalized) likelihood setting, the structure $\mathcal{H}_M$ of the model cannot be inferred automatically. As the likelihood function is unbounded, both ML and MAP learning favor models of ever increasing complexity, and are thus incapable of performing model selection automatically. Still, we can follow a more conventional approach to model selection by splitting the learning

data in a training and validation set. The Bayesian setting does not have this drawback as the model complexity is included in the problem statement. As a result, Bayesian inference is not wasteful of valuable learning data.

During model fitting, it is assumed that the model structure $\mathcal{H}_M$ is fixed. The parameters are learnt given the observed data X. Applying Bayes' rule allows us to update our prior belief on $\boldsymbol{\theta}$ to a posterior belief given X:

$$\underbrace{p\left(\boldsymbol{\theta}|X,\mathcal{H}_M\right)}_{\text{posterior}} = \frac{\overbrace{p\left(X|\boldsymbol{\theta},\mathcal{H}_M\right)}^{\text{likelihood}}\overbrace{p(\boldsymbol{\theta}|\mathcal{H}_M)}^{\text{prior}}}{\underbrace{p\left(X|\mathcal{H}_M\right)}_{\text{evidence}}} \;, \tag{3.37}$$

where the dependency on $\mathcal{H}_M$ is introduced explicitly. In case of latent variable models, the likelihood is identical to the incomplete data likelihood defined in (3.7). The evidence is the probability of observing the data given a particular model $\mathcal{H}_M$. While this quantity is not important in this first level of inference, it plays a crucial role in the second level of inference, which is model selection.

During model selection, the posterior of the model having seen the data X is computed by Bayes' rule:

$$p\left(\mathcal{H}_M|X\right) \propto p\left(X|\mathcal{H}_M\right) p(\mathcal{H}_M) \;. \tag{3.38}$$

In practice, there is no reason to favor one model to another. Therefore, the prior $p(\mathcal{H}_M)$ is often chosen to be uniform, in which case the models can be ranked by their evidence $p\left(X|\mathcal{H}_M\right)$. To compute the evidence, however, we need to integrate out the model parameters:

$$p\left(X|\mathcal{H}_M\right) = \int p\left(X|\boldsymbol{\theta},\mathcal{H}_M\right) p\left(\boldsymbol{\theta}|\mathcal{H}_M\right) d\boldsymbol{\theta} \;. \tag{3.39}$$

Unfortunately, this integral is usually intractable. Next, this issue is addressed by means of variational inference.

Variational Bayes

Consider again the general case of latent variable models and let us treat the parameters $\boldsymbol{\theta}$ as latent variables as well. Following the same approach as in ML learning, for any auxiliary distribution $q(Z,\boldsymbol{\theta})$, the logarithm of the evidence can be lower bounded using Jensen's inequality:

$$\begin{aligned}
\log p\left(X|\mathcal{H}_M\right) &= \log \iint p(X,Z,\boldsymbol{\theta}|\mathcal{H}_M) dZ d\boldsymbol{\theta} && (3.40)\\
&= \log \iint q(Z,\boldsymbol{\theta}) \tfrac{p(X,Z,\boldsymbol{\theta}|\mathcal{H}_M)}{q(Z,\boldsymbol{\theta})} dZ d\boldsymbol{\theta} && (3.41)\\
&\geq \iint q(Z,\boldsymbol{\theta}) \log \tfrac{p(X,Z,\boldsymbol{\theta}|\mathcal{H}_M)}{q(Z,\boldsymbol{\theta})} dZ d\boldsymbol{\theta} && (3.42)\\
&\equiv \mathcal{F}'_{\mathcal{H}_M}(q(Z,\boldsymbol{\theta})) \;. && (3.43)
\end{aligned}$$

The bound can be made tight by equating the auxiliary distribution to the true joint posterior of the latent variables and the parameters:

$$q(Z, \boldsymbol{\theta}) = p(Z, \boldsymbol{\theta}|X, \mathcal{H}_M) \ . \tag{3.44}$$

This is easily verified by considering the following decomposition of the bound:

$$\mathcal{F}'_{\mathcal{H}_M}(q(Z, \boldsymbol{\theta})) = \log p\left(X|\mathcal{H}_M\right) - \mathrm{KL}[q(Z, \boldsymbol{\theta}) \| p(Z, \boldsymbol{\theta}|X, \mathcal{H}_M)] \ . \tag{3.45}$$

Unfortunately, this does not simplify the problem, unlike in the ML case. Evaluating the exact posterior distribution requires to know the evidence as well, which appears as the normalizing constant in Bayes' rule. Instead, in variational Bayes (VB), an approximate posterior is chosen in such a way that the lower bound becomes tractable. In fact, it is sufficient to constrain the variational posterior to have a factorized form:

$$q(Z, \boldsymbol{\theta}) = q_Z(Z) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \left(\prod_{n=1}^{N} q_{\mathbf{z}_n}(\mathbf{z}_n) \right) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \ , \tag{3.46}$$

where the last equation is a consequence of the data X being i.i.d. Thus, the variational approximation of the joint posterior assumes independency between the parameters and the latent variables given the observed data. In other words, the problem is converted into a simpler one by decoupling the degrees of freedom of the original problem.

Under this factorization, the lower bound on the log-evidence has the following form:

$$\log p\left(X|\mathcal{H}_M\right) \geq \iint q_Z(Z) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \log \tfrac{p(X, Z, \boldsymbol{\theta}|\mathcal{H}_M)}{q_Z(Z) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} dZ d\boldsymbol{\theta} \tag{3.47}$$

$$\equiv \mathcal{F}_{\mathcal{H}_M}\left(q_{\mathbf{z}_1}(\mathbf{z}_1), \ldots, q_{\mathbf{z}_N}(\mathbf{z}_N), q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\right) \ . \tag{3.48}$$

Maximizing this bound with respect to the free distributions $q_Z(Z)$ and $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ leads to the variational Bayesian EM update equations (Beal, 2003):

$$\textbf{VBE-step} \ : q_{\mathbf{z}_n}(\mathbf{z}_n) \propto \exp\left(\mathrm{E}_{\boldsymbol{\theta}}\{\log p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}, \mathcal{H}_M)\}\right) \ , \quad \forall n \ . \tag{3.49}$$

$$\textbf{VBM-step} : q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}|\mathcal{H}_m) \exp\left(\mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}|X, Z, \mathcal{H}_M)\}\right) \ . \tag{3.50}$$

In these equations, $\mathrm{E}_{\boldsymbol{\theta}}\{\cdot\}$ and $\mathrm{E}_Z\{\cdot\}$ denote respectively the expectation with respect to $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $q_Z(Z)$, and $\mathcal{L}_c(\boldsymbol{\theta}|X, Z, \mathcal{H}_M)$ is the complete data likelihood defined in (3.15). By construction, VBEM is guaranteed to monotonically increase.

Let us now demonstrate that the VBE- and VBM-step can be obtained without having to resolve a functional maximization problem. First, consider the VBE-step. Starting from (3.47) and denoting the entropy by $\mathrm{H}(\cdot)$, the lower bound

can be re-written as follows (omitting the dependency in $\mathcal{H}_M$):

$$\begin{aligned}
&\mathcal{F}_{\mathcal{H}_M}\left(q_{\mathbf{z}_1}(\mathbf{z}_1),\ldots,q_{\mathbf{z}_N}(\mathbf{z}_N),q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\right)\\
&= \int\!\!\int q_Z(Z)q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\log p(X,Z,\boldsymbol{\theta})dZd\boldsymbol{\theta} + \mathrm{H}(q_Z(Z)) + \mathrm{H}(q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) \quad (3.51)\\
&= \int q_Z(Z)\mathrm{E}_{\boldsymbol{\theta}}\{\log p(X,Z|\boldsymbol{\theta})\}dZ + \mathrm{E}_{\boldsymbol{\theta}}\{\log p(\boldsymbol{\theta})\} + \mathrm{H}(q_Z(Z)) + \mathrm{H}(q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) \quad (3.52)\\
&= \int q_Z(Z)\log\left[\frac{\exp\left(\mathrm{E}_{\boldsymbol{\theta}}\{\log p(X,Z|\boldsymbol{\theta})\}\right)}{q_Z(Z)}\right]dZ + c_1(\boldsymbol{\theta}) \quad (3.53)\\
&= -\mathrm{KL}\left[q_Z(Z)\middle\|\frac{1}{c_Z}\exp\left(\mathrm{E}_{\boldsymbol{\theta}}\{\log p(X,Z|\boldsymbol{\theta})\}\right)\right] + c_2(\boldsymbol{\theta},c_Z) \quad (3.54)\\
&= -\sum_{n=1}^{N}\mathrm{KL}\left[q_{\mathbf{z}_n}(\mathbf{z}_n)\middle\|\left(\frac{1}{c_Z}\right)^{\frac{1}{N}}\exp\left(\mathrm{E}_{\boldsymbol{\theta}}\{\log p(\mathbf{x}_n,\mathbf{z}_n|\boldsymbol{\theta})\}\right)\right] + c_2(\boldsymbol{\theta},c_Z). \quad (3.55)
\end{aligned}$$

In the last equation, we use the fact that the data are i.i.d. Constant $c_1(\cdot)$ is a function of $\boldsymbol{\theta}$ only and $c_2(\cdot)$ is a function of $\boldsymbol{\theta}$ and the normalizing constant c_Z. From (3.55), it can be seen that the VBE-step maximizes indeed the lower bound with respect to $q_{\mathbf{z}_n}(\mathbf{z}_n)$, $\forall n$.

Next, consider the VBM-step. It can be decomposed by analogy with the VBE-step (omitting again the dependency in $\mathcal{H}_M$):

$$\begin{aligned}
&\mathcal{F}_{\mathcal{H}_M}\left(q_{\mathbf{z}_1}(\mathbf{z}_1),\ldots,q_{\mathbf{z}_N}(\mathbf{z}_N),q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\right)\\
&\quad= \int\!\!\int q_Z(Z)q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\log p(X,Z,\boldsymbol{\theta})dZd\boldsymbol{\theta} + \mathrm{H}(q_Z(Z)) + \mathrm{H}(q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) \quad (3.56)\\
&\quad= \int q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\mathrm{E}_Z\{\log p(X,Z,\boldsymbol{\theta})\}d\boldsymbol{\theta} + \mathrm{H}(q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) + \mathrm{H}(q_Z(Z)) \quad (3.57)\\
&\quad= -\mathrm{KL}\left[q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\middle\|\frac{1}{c_{\boldsymbol{\theta}}}\exp\left(\mathrm{E}_Z\{\log p(X,Z,\boldsymbol{\theta})\}\right)\right] + c_3(Z,c_{\boldsymbol{\theta}})\ . \quad (3.58)
\end{aligned}$$

Constant $c_3(\cdot)$ is a function of Z and the normalizing constant $c_{\boldsymbol{\theta}}$. Since we have

$$\begin{aligned}
\exp\left(\mathrm{E}_Z\{\log p(X,Z,\boldsymbol{\theta}\}\right) &= p(\boldsymbol{\theta})\exp\left(\mathrm{E}_Z\{\log p(X,Z|\boldsymbol{\theta})\}\right) \quad (3.59)\\
&= p(\boldsymbol{\theta})\exp\left(\mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}|X,Z)\}\right)\ , \quad (3.60)
\end{aligned}$$

it can be seen from (3.58) that the VBM-step maximizes indeed the lower bound with respect to $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$.

One might think at first sight that the VBE- and the VBM-step only differ in the prior term on the parameters. However, the prior on the latent variables is included in $p(X,Z|\boldsymbol{\theta},\mathcal{H}_M)$. As a matter of fact, VBEM makes no distinction between the latent variables and the parameters, except that the number of hidden variables increases with the size of the data set, while the number of parameters is fixed. Both in the VBE- and the VBM-step, the lower bound on the log-evidence is maximized by minimizing the KL divergence between the factorized variational posterior and the true joint posterior of the latent

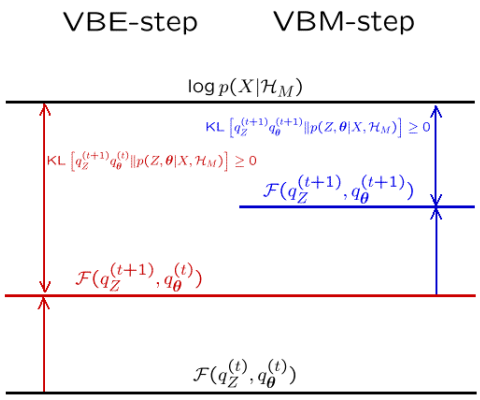


FIGURE 3.4. Variational Bayesian EM steps. In order to maximize the lower bound on the log-evidence, both the VBE- and the VBM-step minimize the KL divergence between the factorized variational posterior and the true joint posterior of the latent variables and the parameters.

variables and the parameters given X (see Figure 3.4):

$$\begin{aligned}\mathcal{F}_{\mathcal{H}_M}&\left(q_{\mathbf{z}_1}(\mathbf{z}_1), \ldots, q_{\mathbf{z}_N}(\mathbf{z}_N) q_{\boldsymbol{\theta}}(\boldsymbol{\theta})\right) \\ &= \log p\left(X|\mathcal{H}_M\right) - \mathrm{KL}[q_Z(Z) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \| p(Z, \boldsymbol{\theta}|X, \mathcal{H}_M)] \ . \end{aligned} \tag{3.61}$$

The factorized posterior is optimized such that, in terms of KL divergence, it is a good approximation of the true posterior, making the bound as tight as possible. However, the factorized posterior will have most of its mass in some region of the feature space where the true posterior has high probability, while it may have low probability in other high probability regions of the true posterior. In other words, the optimal $q(Z, \boldsymbol{\theta})$ will be generally more compact than the true posterior (Winn, 2003). This is a consequence of the KL divergence being asymmetric.

The VB framework requires to choose an initial prior distribution over the parameters. By repeatedly applying the VBE- and the VBM-step, the variational posterior is computed. In practice, it is convenient to choose the prior to be conjugate to the exponential family. The prior $p(\boldsymbol{\theta})$ is said to be conjugate to $r(\mathbf{x}|\boldsymbol{\theta})$ if the posterior $q(\boldsymbol{\theta}|\mathbf{x}) \propto r(\mathbf{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta})$ is of the same form as $p(\boldsymbol{\theta})$. Learning in the VB framework consists then simply in updating the parameters of the prior to the parameters of the posterior.

The true predictive distribution is approximated using the variational posterior of the parameters:

$$p(\mathbf{x}) \approx \int p(\mathbf{x}|\boldsymbol{\theta}) q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \ . \tag{3.62}$$

In case the integral is intractable, one may also compute the Bayes estimate of the parameters

$$\hat{\boldsymbol{\theta}}_{\text{Bayes}} = \int \boldsymbol{\theta} q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \ , \tag{3.63}$$

which can then be plugged into the original model by approximating the posterior of the parameters by a delta function:

$$p(\mathbf{x}) \approx \int p(\mathbf{x}|\boldsymbol{\theta}) \delta(\hat{\boldsymbol{\theta}}_{\text{Bayes}}) d\boldsymbol{\theta} = p(\mathbf{x}|\hat{\boldsymbol{\theta}}_{\text{Bayes}}) \ . \tag{3.64}$$

Note that ML and MAP approximate the predictive distribution in a similar way, but they use respectively $\delta(\hat{\boldsymbol{\theta}}_{\text{ML}})$ and $\delta(\hat{\boldsymbol{\theta}}_{\text{MAP}})$ instead of $\delta(\hat{\boldsymbol{\theta}}_{\text{Bayes}})$.

In the next sections, ML, MAP and VB learning are applied to Gaussian and Student-t mixture models.

3.2. Finite Gaussian Mixture Models

Early references on finite Gaussian mixtures include Sundberg (1972, 1974) and the excellent review of Redner and Walker (1984). During the past decade, Gaussian mixtures gained a renewed interest and are still an active field of research. See for example the book of McLachlan and Peel (2000) for a thorough discussion. Among others, the success of mixture models in general can be explained by their ability to model heterogenous data, which are frequent in a wide range of applications. For instance, Gaussian mixtures have been successfully applied to the segmentation of the brain tissues in magnetic resonance images (Gupta and Sortrakul, 1998; Schroeter, Vesin, Langenberger and Meuli, 1998; Ruan, Jaggi, Xue, Fadili and Bloyet, 2000; Bach-Cuadra, Platel, Solanas, Butz and Thiran, 2002), text desambiguation (de Marneffe, Archambeau, Dupont and Verleysen, 2004) or vision based fire detection (Liu and Ahuja, 2004).

A finite Gaussian mixture model (GMM) is defined as a linear combination of M multivariate Gaussian distributions:

$$p(\mathbf{x}|\boldsymbol{\theta}_{\mathcal{N}}) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) \ , \tag{3.65}$$

where $\boldsymbol{\theta}_{\mathcal{N}} = (\pi_1, \ldots, \pi_M, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M, \boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_M)$. The mixing proportions $\{\pi_m\}_{m=1}^M$ are non-negative and must sum to one. The definition of the multivariate Gaussian distribution is given in (2.27). Estimating the true density $p(\mathbf{x})$ by the approximate density $p(\mathbf{x}|\boldsymbol{\theta}_{\mathcal{N}})$ consists in computing the parameters $\{\boldsymbol{\mu}_m\}_{m=1}^M$, $\{\boldsymbol{\Lambda}_m\}_{m=1}^M$ and $\{\pi_m\}_{m=1}^M$ based on the learning data.

The GMM is commonly used for clustering tasks. We could however imagine to use it in a more general, nonparametric-like framework when the model complexity is selected arbitrarily. In this case, the goal would be not to decompose the observed data into distinct clusters, but to model the data locally. This is motivated by some attractive properties of the GMM:

(1) The GMM is locally data dependent and therefore able to deal with the local dispersion of the data points.
(2) The GMM is flexible due to the introduction of the mixture proportions $\{\pi_m\}_{m=1}^M$, resulting in a relatively low model complexity.
(3) The GMM provides smooth estimates that are expected to generalize better on new data, as oscillations in the density estimate are prevented.

In addition, the model complexity of the GMM does not depend on the size of the learning set. Therefore, an excessive use of memory resources is avoided. By contrast, the computational complexity during the training phase is rather large compared to ordinary kernel density estimation.

Subsequently, maximum likelihood, maximum a posteriori and Bayesian learning of the GMM are described. Variants are proposed either for improving the generalization capabilities or to ease the learning procedure. Finally, several solutions to the model selection problem are provided.

3.2.1. Maximum Likelihood Learning

Assume the observed data $X = \{\mathbf{x}_n\}_{n=1}^N$ are i.i.d. The data log-likelihood under the GMM density model is given by

$$\log \mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X) \equiv \log p(X|\boldsymbol{\theta}_{\mathcal{N}}) = \sum_{n=1}^{N} \log p(\mathbf{x}_n|\boldsymbol{\theta}_{\mathcal{N}}) \ . \tag{3.66}$$

Unfortunately, maximizing $\mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X)$ (or equivalently its log) subject to the constraint on the mixture proportions is intractable, unless we define a component dependent auxiliary variable associated to each data point:

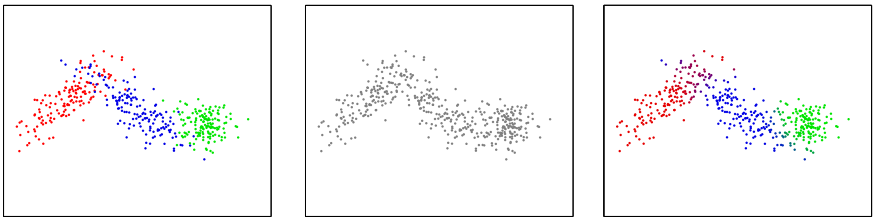
$$\bar{\rho}_{nm} = \frac{\pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)}{\sum_{m'=1}^M \pi_{m'} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{m'}, \boldsymbol{\Lambda}_{m'})} \ , \quad \forall n \ , \quad \forall m \ . \tag{3.67}$$

In this definition, each mixture proportion π_m can be interpreted as the (estimated) prior probability of having the m^{th} component of the mixture. Furthermore, the conditional probability $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$ is the probability of observing $\mathbf{x}_n$ given the component mean $\boldsymbol{\mu}_m$ and the component precision $\boldsymbol{\Lambda}_m$, i.e. assuming $\mathbf{x}_n$ is generated by the mixture component m. Recalling Bayes' rule, it can easily be seen that each auxiliary variable $\bar{\rho}_{nm}$ is nothing else than the posterior probability that $\mathbf{x}_n$ is generated by m, provided density model (3.65). The auxiliary variables are therefore called *responsibilities*.

Denoting the Lagrange multiplier by λ, the Lagrangian is constructed as follows:

$$\log \ell(\boldsymbol{\theta}_{\mathcal{N}}, \lambda) = \log \mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X) + \lambda \left(\sum_{m=1}^{M} \pi_m - 1 \right) \ . \tag{3.68}$$

When the responsibilities are fixed, we can maximize $\log \ell$ with respect to the model parameters. Rearranging leads to the following estimation formulas for



(a) Known labels. (b) Unknown labels. (c) Expected labels.

FIGURE 3.5. Example of a mixture of three Gaussian distributions. In (a) the data points are colored according to their true label. (b) shows the situation in practice: the label (color) is not observed. (c) shows the labels that are recovered by the EM algorithm. The data points are colored according to their responsibilities.

the component means, precisions and weights:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} \mathbf{x}_n}{\sum_{n=1}^{N} \bar{\rho}_{nm}} \ , \tag{3.69}$$

$$\boldsymbol{\Lambda}_m = \left\{ \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)^{\mathrm{T}}}{\sum_{n=1}^{N} \bar{\rho}_{nm}} \right\}^{-1} \ , \tag{3.70}$$

$$\pi_m = \frac{1}{N} \sum_{n=1}^{N} \bar{\rho}_{nm} \ . \tag{3.71}$$

Observe that (3.69) and (3.70) are weighted averages based on the responsibilities. These update equations turn out to be the EM update rules, which will be discussed shorthly. The procedure operates iteratively in two stages. In the E-step, the responsibilities (3.67) are computed, while the current model parameters $\boldsymbol{\theta}_{\mathcal{N}}$ are kept fixed. Subsequently, during the M-step, the model parameters are updated according to (3.69–3.71) using the responsibilities variables computed in the E-step.

Latent Variable Viewpoint

More formally, the GMM can be viewed as a latent variable model in the sense that the component label associated to each data point is unobserved. This is illustrated in Figure 3.5. Although the data generation process involves labeled data, the data labels are in practice unknown. In other words, we have no idea by which component a data point has been generated. By means of the EM algorithm, the expected labels can be recovered.

Consider the set of binary latent vectors $Z = \{\mathbf{z}_n\}_{n=1}^{N}$, with latent variables $z_{nm} \in \{0, 1\}$ indicating which component has generated $\mathbf{x}_n$. Variable z_{nm}

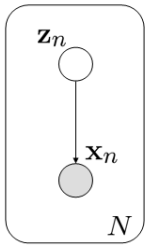


FIGURE 3.6. Graphical representation of the GMM. The nodes are random variables, the shaded ones being observed. The plate denotes the repetition of i.i.d. observations. The arrow indicates the conditional dependency of the nodes. We omit the dependency on the parameters since they are deterministic quantities.

is equal to 1 if $\mathbf{x}_n$ is generated by component m and equal to 0 otherwise. Therefore, the following constraint should be satisfied:

$$\sum_{m=1}^{M} z_{nm} = 1 \ , \quad \forall n \ . \tag{3.72}$$

The prior distribution of the latent vectors and the conditional distribution of observed data are then respectively:

$$p(\mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}) = \prod_{m=1}^{M} {\pi_m}^{z_{nm}} \ , \tag{3.73}$$

$$p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta}_{\mathcal{N}}) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)^{z_{nm}} \ . \tag{3.74}$$

Marginalizing over the latent variables results in (3.65):

$$p(\mathbf{x}_n|\boldsymbol{\theta}_{\mathcal{N}}) = \sum_{\mathbf{z}_n} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}) \tag{3.75}$$

$$= \sum_{\mathbf{z}_n} p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta}_{\mathcal{N}}) p(\mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}) \tag{3.76}$$

$$= \sum_{\mathbf{z}_n} \prod_{m=1}^{M} {\pi_m}^{z_{nm}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)^{z_{nm}} \tag{3.77}$$

$$= \sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) \ . \tag{3.78}$$

Figure 3.6 shows the GMM as a simple directed acyclic graph. The random variables in the model are the observations $X = \{\mathbf{x}_n\}_{n=1}^N$ and the latent vectors $Z = \{\mathbf{z}_n\}_{n=1}^N$. Each $\mathbf{x}_n$ depends conditionally on $\mathbf{z}_n$. Both the observations and the latent vectors are i.i.d. The plate indicates N copies. The parameters do not appear in the graph as they are fixed.

As discussed in Section 3.1.1, the EM algorithm seeks iteratively for a local maximum of the data log-likelihood by first computing the posterior probability of the latent variables (E-step) and then maximizing the expected complete data log-likelihood with respect to the model parameters (M-step). The expectation is taken with respect to the posterior distribution computed in the E-step as shown below.

Since the joint distribution $p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}})$ factorizes, the posterior probability of the indicator variables factorizes as well. Using Bayes' rule leads to the E-step:

$$p(z_{nm} = 1|\mathbf{x}_n, \boldsymbol{\theta}_{\mathcal{N}}) = \frac{\pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)}{p(\mathbf{x}_n|\boldsymbol{\theta}_{\mathcal{N}})} = \bar{\rho}_{nm} \ , \quad \forall n \ , \quad \forall m \ . \tag{3.79}$$

Next, let us detail how the M-step proceeds. The complete data log-likelihood of the GMM is given by

$$\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{N}}|X, Z) = \log \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}) \tag{3.80}$$

$$= \log \prod_{n=1}^{N} \prod_{m=1}^{M} {\pi_m}^{z_{nm}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)^{z_{nm}} \tag{3.81}$$

$$= \sum_{n=1}^{N} \sum_{m=1}^{M} z_{nm} \left\{\log \pi_m + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\right\} \ . \tag{3.82}$$

Observe the difference between this log-likelihood and the original log-likelihood, i.e. the incomplete data log-likelihood:

$$\log \mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X) = \log \prod_{n=1}^{N} p(\mathbf{x}_n|\boldsymbol{\theta}_{\mathcal{N}}) \tag{3.83}$$

$$= \sum_{n=1}^{N} \log \sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) \ . \tag{3.84}$$

In this expression, the logarithm appears outside the summation with respect to m. This reflects the fact that the incomplete log-likelihood is a marginal probability. As a result, there is no closed form solution to this maximization problem. In contrast, the complete log-likelihood is not a marginal probability, and thus the logarithm is inside the sum, leading to simple maximum likelihood formulas.

The expected complete data log-likelihood can be written as follows:

$$\mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{N}}|X, Z)\}$$

$$= \sum_{n=1}^{N} \sum_{m=1}^{M} \mathrm{E}_Z\{z_{nm}\} \left\{\log \pi_m + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\right\} \ . \tag{3.85}$$

Now, the conditional expectation of the latent variables with respect to their posterior distribution is equal to the responsibilities:

$$\mathrm{E}_Z\{z_{nm}\} = \sum_{z_{nm}} z_{nm} p(z_{nm}|\mathbf{x}_n, \boldsymbol{\theta}_{\mathcal{N}}) \tag{3.86}$$

$$= \sum_{z_{nm}} z_{nm} \frac{{\pi_m}^{z_{nm}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)^{z_{nm}}}{p(\mathbf{x}_n|\boldsymbol{\theta}_{\mathcal{N}})} \tag{3.87}$$

$$= \bar{\rho}_{nm} \ . \tag{3.88}$$

Maximizing (3.85) subject to the constraint on the mixture proportions leads to the update rules (3.69–3.71).

Initialization

Choosing adequate initial values for the component means, precisions and proportions is essential in order to attain a good local maximum of the log-likelihood surface. While a random assignment does not provide satisfactory results, a simple technique such as M-means[1] (MacQueen, 1967) provides already good starting values. Moreover, M-means is closely related to the GMM, making this technique even more attractive.

Recall we have a set of observations $X = \{\mathbf{x}_n\}_{n=1}^N$. Our goal in M-means is to cluster the data into M clusters. The basic idea is to assign each data point to its nearest cluster mean or centroid. Let $\{\boldsymbol{\mu}_m\}_{m=1}^M$ be the centroids and $Z = \{\mathbf{z}_n\}_{n=1}^N$ the discrete indicator vectors, such that $z_{nm} \in \{0, 1\}$ and $\sum_{m=1}^M z_{nm} = 1, \ \forall n$. Making an initial assignment for the centroids, the M-means algorithm alternates between the two following steps:

(1) Evaluate the indicator vectors:

$$z_{nm} = \begin{cases} 1 & \text{if } m = \arg\min_m \|\mathbf{x}_n - \boldsymbol{\mu}_m\|^2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.89}$$

(2) Compute the centroids:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^N z_{nm}\mathbf{x}_n}{\sum_{n=1}^N z_{nm}} \ . \tag{3.90}$$

As M-means depends on some initial assignment of the centroids as well, it is worth considering multiple initializations and keeping subsequently the best one (for example the one minimizing the reconstruction error). Of course, this is at the cost of additional computation time. For the same initialization of the centroids, M-means minimizes the same reconstruction error as competitive

[1] Similarly as for M-NN in Chapter 2, we use the term M-means instead of K-means in order to be consistent with our notations, M denoting the number of components.

learning (see Section 2.3.3):

$$R = \sum_{n=1}^{N} \sum_{m=1}^{M} z_{nm} \|\mathbf{x}_n - \boldsymbol{\mu}_m\|^2 \ . \tag{3.91}$$

Note, however, that competitive learning minimizes R stochastically.

Comparing (3.90) to (3.69) makes the link between M-means and the GMM explicit. While the first method performs a hard assignment of the data points to the cluster means, the second one performs a soft assignment. M-means converges after a finite number of iterations, since there are only a finite number of assignments for the discrete vectors and for each assignment there is a unique value for the cluster means. In practice, the algorithm converges rapidly. When it is terminated, the component precisions and proportions can be estimated as follows:

$$\boldsymbol{\Lambda}_m = \left\{ \frac{\sum_{n=1}^{N} z_{nm} (\mathbf{x}_n - \boldsymbol{\mu}_m)(\mathbf{x}_n - \boldsymbol{\mu}_m)^{\mathrm{T}}}{\sum_{n=1}^{N} z_{nm}} \right\}^{-1} \ , \tag{3.92}$$

$$\pi_m = \frac{1}{N} \sum_{n=1}^{N} z_{nm} \ . \tag{3.93}$$

More elaborate techniques for being less sensitive to the initial conditions of the GMM include stochastic EM (SEM) (Celeux and Diebolt, 1985; Celeux, Chaveau and Diebolt, 1996), deterministic annealing EM (DAEM) (Ueda and Nakano, 1998) and split-and-merge EM (Ueda, Nakano, Ghahramani and Hinton, 2000) (SMEM).

In SEM, a label vector $\mathbf{z}_n$ is effectively assigned to each observation $\mathbf{x}_n$ at each iteration step, according to the multinomial distribution with M categories and having the current responsibilities $\bar{\rho}_{nm}$ for parameters. Replacing the expected labels (i.e. the responsibilities) by a stochastic assignment allows SEM to escape from the current path of convergence followed by the EM algorithm. This is especially desirable when the algorithm is started from poor parameter values, but it is not when the process is close to convergence. It is therefore suggested to use ordinary EM in the latter stages of the iterative procedure.

DAEM acts also on the E-step. Here, the conditional expectation of the complete data log-likelihood is computed with respect to the current responsibilities raised to the power β:

$$\varrho_{nm} = \frac{(\bar{\rho}_{nm})^{\beta}}{\sum_{m'=1}^{M} (\bar{\rho}_{nm'})^{\beta}} \ , \quad \forall n \ , \quad \forall m \ . \tag{3.94}$$

The inverse of β is referenced as a temperature by analogy to statistical mechanics. It is suggested to start with a value of β close to zero and then increasing it after each iteration until β equals 1. When β is small, the adjusted responsibilities are close to $1/M$, producing component densities that overlap considerably. When β increases, the contribution of the data points are gradually localized. Therefore, the DAEM is able to recover from a poor

choice of the starting values by letting the components overlap considerably in the first iterations.

Ueda et al. (2000) proposed SMEM, which combines the EM algorithm to split-and-merge operations. After convergence of the EM algorithm, SMEM checks if the expected complete data log-likelihood can be improved by splitting one component into two, while merging two others. Next, two steps, called the partial and the full EM steps, are executed in turn. The partial EM is only applied to the above new three components and the full EM step to all components of the mixture, yielding a new set of parameters $\boldsymbol{\theta}^*_{\mathcal{N}'}$. Then, the new mixture is accepted if the following condition holds:

$$\mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}^*_{\mathcal{N}'}|X, Z\} > \mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{N}}|X, Z\} \ . \tag{3.95}$$

In fact, a set of three candidate components is generated and appropriately ranked. If none of the candidates yield an improvement, the algorithm is terminated.

SMEM allows to jump to regions in the parameter space being hopefully more attractive. However, it was recently demonstrated that SMEM is not fully compatible with maximum likelihood learning (Minagawa, Tagawa and Tanaka, 2002). The reason is that the expected complete log-likelihood is computed with respect to different posterior probabilities. Therefore, an increase in the expected complete log-likelihood does not correspond necessarily to an increase in likelihood, possibly leading to the rejection of the global optimum.

More recently, Verbeek, Vlassis and Kröse (2003) proposed an efficient greedy learning algorithm for the GMM, building on the work of Li and Barron (1999), and Vlassis and Likas (2002). In this approach, the components in the mixture are inserted one after the other according to a heuristic. A set of new candidate components are generated in a randomized manner. Then, by using partial EM searches (see Verbeek et al., 2003, for a detailed description), locally optimal candidates and their corresponding weights are computed. Subsequently, the optimal new component is selected as the one maximizing the resulting log-likelihood and is included in the mixture. Possibly ordinary EM is then applied until convergence. The greedy algorithm has a running time M times slower than standard EM. It is also reported to perform similarly as SMEM (while being faster) and to outperform ordinary EM initialized by M-means (see Verbeek et al., 2003).

Convergence

In the context of the GMM, the EM algorithm has been found to have the advantage to provide rather reliable estimators. A recent experimental study of the asymptotic properties (i.e., for $N \to \infty$) of the univariate GMM, showed that only (moderate) biases in the parameter estimates are observed when the component means are close to each other and the variances are considerably different (Nityasuddhi and Böhning, 2003). However, it was reported that

its convergence may be slow (Redner and Walker, 1984; Meng and van Dyk, 1997), especially when the component means are close. In addition, it is not guaranteed that EM provides a global maximum of the log-likelihood surface (Wu, 1983).

Xu and Jordan (1996) provided a careful study of the EM algorithm's convergence properties for the GMM, tempering the critics offered by Redner and Walker (1984). The authors linked the algorithm to gradient methods and demonstrated that, under appropriate conditions, it approximates superlinear methods (e.g., quasi-Newton). They concluded that the EM algorithm is particularly attractive in the case of the GMM, as it enjoys automatic satisfaction of probabilistic constraints, monotonic convergence, without the need to set a learning rate, and low computational overhead. Moreover, while EM can converge slowly for problems in which the mixture components are not well separated, the gradient-based algorithms (including Newton's method) are also likely to perform poorly due to a poorly conditioned Hessian. Finally, when one is concerned with the convergence in likelihood, EM generally performs well.

Choosing the number of components

The major drawback of the learning procedure of the GMM is that maximizing the likelihood is ill-posed. The numerical difficulties are for example often encountered when dealing with (multivariate) real data, which is due to the fact that the likelihood function is unbounded. This may result in putting infinite probability mass on a single data point, leading to a mixture component to collapse (see for example Duda and Hart, 1973; Abbas and Fahmy, 1994; McLachlan and Peel, 2000). Archambeau, Lee and Verleysen (2003) traced the collapsing mechanism in the case of isotropic Gaussian kernels and accredited this burden to the concept of relative isolation of some training data. By "relative isolated data point" it is meant that the point is either an outlier or abnormally repeated. The local character of Gaussian components combined to the presence of isolated data makes the EM possibly collapse. Actually, because of the exponentially decreasing shape of the components it is more likely that an outlier is generated by a highly improbable isolated component than by a component consistent with the database. The width of the component is then driven to zero and the corresponding mixture weight tends to $1/N$.

Since ML learning is an ill-posed problem, major problems arise when using resampling techniques. Maximizing the likelihood favors models of ever increasing complexity. As a result, neither the optimal number of components can be selected automatically, nor the model parameters can be estimated reliably. Model selection and parameter estimation on the basis of the data likelihood can thus only be carried out by learning and validating the model on separate data sets. However, due to numerical instabilities, this is impractical with the conventional GMM.

In order to avoid these numerical instabilities, the covariance matrices are often constrained, such that the likelihood is bounded. In particular, diagonal covariance matrices are usually enforced (Abbas and Fahmy, 1994). However, as shown in Section 3.2.2, diagonal GMM are incapable of modeling arbitrary densities due to their lack of flexibility. More recently, regularized versions of GMM were proposed (Borgelt and Kruse, 2004; Archambeau and Verleysen, 2003; Archambeau, Vrins and Verleysen, 2004), as well as the maximum a posteriori GMM (Ormoneit and Tresp, 1998) and variational GMM (Attias, 1999b). These methods are discussed in detail in the following sections.

3.2.2. Learning with the Regularized Mahalanobis Distance

Archambeau and Verleysen (2003) introduced a regularization scheme based on the regularized Mahalanobis distance (Mao and Jain, 1996). The approach is closely related to the work of Borgelt and Kruse (2004), who also proposed shape and size regularization of the mixture components. The main drawback of their method is that it requires to set many parameters.

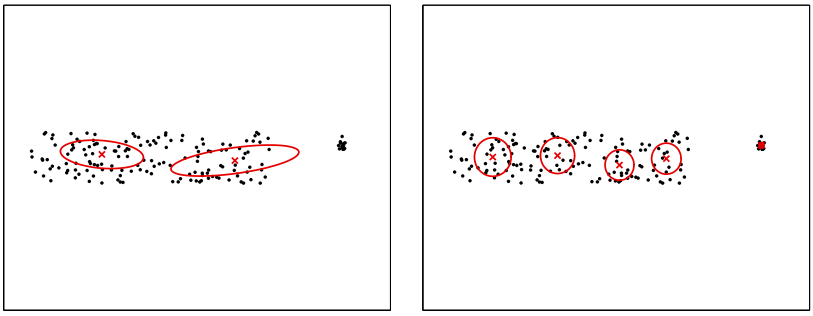
The multivariate Gaussian components determine their shape by means of the Mahalanobis distance. In order to improve the quality and the stability of the estimator, it is proposed to penalize the component shape by making a compromise between the Mahalanobis distance, which favors hyperellipsoidal components, and the Euclidian distance, which favors hyperspherical components. This is motivated by the prior belief that the shape of each component should not be too thin and that a component should include a sufficient number of data points in order to reliably estimate its covariance matrix. If this condition is not met, the covariance matrix should be further constrained. At the end of this section, the power and flexibility of the approach is validated by experimental results.

The regularized Mahalanobis distance

From (2.27), it can be seen that the multivariate Gaussian component m uses the Mahalanobis distance Δ to determine its shape:

$$\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) = (\mathbf{x} - \boldsymbol{\mu}_m)^{\mathrm{T}} \boldsymbol{\Lambda}_m (\mathbf{x} - \boldsymbol{\mu}_m) \ , \tag{3.96}$$

where a probabilistic notation is abusively used to denote the dependency on $\boldsymbol{\mu}_m$ and $\boldsymbol{\Lambda}_m$. When the number of data points contributing to the computation of the component covariance matrix (and thus also its precision) is small with respect to the square of the dimension d of the data points, it may be singular. Moreover, the use of $\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$ tends to produce hyperellipsoidal components, leading to unusually large and elongated densities. By contrast, when one considers the Euclidean distance $\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \mathbf{I})$, large data clusters need to split unnecessarily, as the component densities are constrained to be hyperspherical. This is illustrated in Figure 3.7.



(a) Unconstrained GMM. (b) Isotropic GMM.

FIGURE 3.7. Illustration of (a) the hyperellipsoidal character of the unconstrained GMM and (b) the hyperspherical character of the isotropic GMM. The former uses the Mahalanobis distance to determine its shape. The latter uses the Euclidean distance. On the one hand, the use of the Mahalanobis distance leads to elongated components, which possibly absorb small data clusters. On the other hand, the use of the Euclidean distance requires a high number of components to model elongated clusters.

Based on the hyperspherical character of $\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \mathbf{I})$ and the hyperellipsoidal character of $\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$, we can construct the regularized Mahalanobis distance as a convex combination of both distances:

$$\Delta'(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) = (1-\tau)\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) + \tau\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \mathbf{I}) \ , \tag{3.97}$$

where τ is in the interval $[0, 1]$. Parameter τ controls the trade-off between hyperspherical and hyperellipsoidal character of the components. It adjusts the effective number of parameters that determine the shape of the mixture components. Therefore, when the covariance matrices cannot be estimated reliably, a large value of τ should be used in order to enforce spherical components. Indeed, spherical components only require to estimate a single parameter. This will be illustrated experimentally below.

Modified M-step

Consider again the M-step (3.69–3.71) of the unconstrained GMM. Introducing the regularized Mahalanobis distance consists in adapting, at each iteration step, the precision $\boldsymbol{\Lambda}_m$ of each component density according to (3.97). Therefore, update rule (3.70) of the kernel precisions becomes:

$$\left\{ \begin{array}{rcl} \boldsymbol{\Sigma}_m & = & \frac{\sum_{n=1}^N \bar{\rho}_{nm} (\mathbf{x}_n - \boldsymbol{\mu}_m)(\mathbf{x}_n - \boldsymbol{\mu}_m)^{\mathrm{T}}}{\sum_{n=1}^N \bar{\rho}_{nm}} \ , \\ \boldsymbol{\Lambda}_m & = & (1-\tau) \left(\boldsymbol{\Sigma}_m + \epsilon \mathbf{I}\right)^{-1} + \tau\lambda\mathbf{I} \ . \end{array} \right. \tag{3.98}$$

Parameter ϵ is called the safety factor and λ the scaling factor. The role of ϵ is to stabilize the learning process when needed (especially when $\tau = 0$, i. e. no regularization), by converting a singular matrix to a non-singular one. As a result, using different values of ϵ does not make much difference as long as they are significantly smaller than the variance of the data points (see experimental results below).

The scaling factor λ takes the range of the data into account. It is computed according to a rule-of-thumb that reflects our prior belief about the expected precision of each kernel:

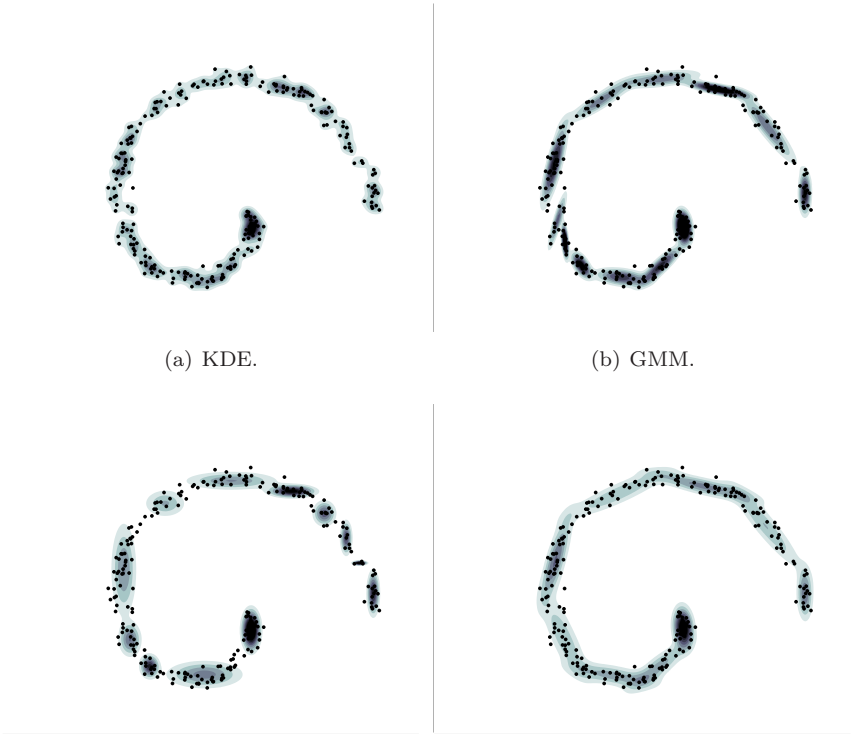
$$\lambda = \left(\frac{\hat{\sigma}_X}{\sqrt[2d]{M}} \right)^{-2} , \tag{3.99}$$

where $\hat{\sigma}_X$ is the empirical standard deviation of the observed data. Parameter λ is thus inversely proportional to the total variance of the data and proportional to the d^{th} root of the number of components. By including the dependency on the dimension d, more overlapping is enforced when moving to a higher dimension. However, a careful choice of λ does not make much difference either, as the amount of prior belief included in the model depends mainly on the value of τ.

Experimental validation

First, let us illustrate the difference between the GMM using unconstrained precisions, the GMM using diagonal constrained precisions (DGMM) and the GMM using the regularized Mahalanobis distance (RGMM). The data is sampled from the noisy spiral described in Appendix A. The number of components in the mixture is 12. Figure 3.8 shows the best estimators out of 20 runs. The ordinary kernel density estimator (KDE) is also constructed. Clearly, the DGMM fails to provide a good model for the data. Using diagonal precisions is thus not suitable in practice. Next, it can be observed that the standard GMM may be too sensitive to local variation in the data. It can for example be seen in the lower left corner of (b) that one of the components is not aligned along the spiral. This results from the fact that too few data points are assigned to the misaligned component. By contrast, it can be observed from (d) that this problem is avoided in RGMM. Comparing the RGMM estimator to the other estimators, it can be seen that the RGMM provides a much smoother estimator than the other techniques. In particular, the oscillations appearing in the estimator of the KDE are avoided.

Next, the quality of the RGMM estimator is compared to that of the KDE, the SKDE, the weighted VQKDE, the GMM and the DGMM. Three 2D toy examples are considered. The first one is a mixture of a Gaussian distribution and a Gaussian-Gamma distribution. By Gaussian-Gamma distribution is meant that the data is Gaussian distributed in one direction (horizontally) and Gamma distributed in the orthogonal direction (vertically). The Gamma distribution is defined as $\mathcal{G}(u|\alpha, \beta) = \beta^\alpha / \Gamma(\alpha) u^{\alpha-1} \exp(-\beta u)$, where $\Gamma(\cdot)$ is the



(a) KDE. (b) GMM.

(c) DGMM. (d) RGMM.

FIGURE 3.8. Density estimators obtained from the KDE, the GMM, the DGMM and the RGMM for the noisy spiral. The number of components in the mixtures is fixed to 12. The kernel width in KDE is set to $0.067\hat{\sigma}_X$ and the regularization parameter in RGMM to 0.26. Both were optimized by 10-fold cross-validation, the ANLL being used as performance measure.

gamma function. The target distribution is given by

$$p(\mathbf{x}) = 0.4\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) + 0.6\mathcal{N}(x_1|0, \lambda)\mathcal{G}(x_2|\alpha, \beta) \ , \tag{3.100}$$

where $\boldsymbol{\mu} = (2, -1)$, $\boldsymbol{\Lambda}^{-1} = \text{diag}\{1.25, 0.75\}$, $\alpha = 2$, $\beta = 0.7$ and $\lambda^{-1} = 2.25$. The training set for this first toy example contains 250 data points. The second and the third toy examples are respectively the ring and the spiral data. Both are described in Appendix A. In the three experiments 1,000 test points are used in order to obtain reliable results. The parameters of the estimators are optimized by 10-fold cross-validation. The performance measure is the average negative log-likelihood (ANLL) and the results are averaged over 20 training runs.

TABLE 3.1. Mixture of a Gaussian and a Gaussian-Gamma distribution. The ANLL is evaluated on the test set and averaged over 20 runs.

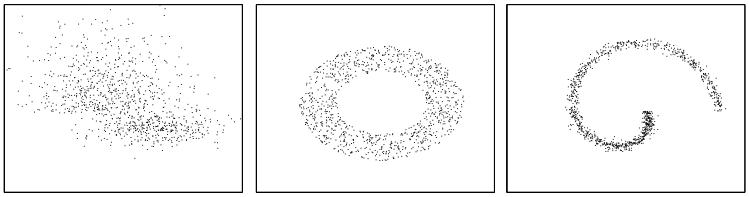
		M	ANLL	std. err.
KDE	$\sigma = 0.35$	250	4.15	0.001
SKDE	$\sigma = 0.29$	250	4.11	0.001
VQKDE	$w = 2.50$	38	4.12	0.002
GMM		3	4.14	0.001
DGMM		4	4.12	0.003
RGMM	$\tau = 0.10$	3	4.11	0.001

Figures 3.9 and 3.10 show the density estimators provided by each method. The darker the color, the higher the density is. For each example, the contour levels are identical across the methods. The grid size was chosen sufficiently small in order to avoid visual artifacts. When comparing the Gaussian mixture models, we can see that the DGMM (dramatically) fails to provide good estimates when the target densities are not aligned with the coordinate axes. It is also clear that the RGMM provides the smoothest estimators. The main drawback of the Gaussian mixture models compared to the kernel estimators is that, when no data clusters effectively exist, the arbitrary densities are approximated by a broken shape. In contrast, the kernel-based methods provide estimators with lots of oscillations.

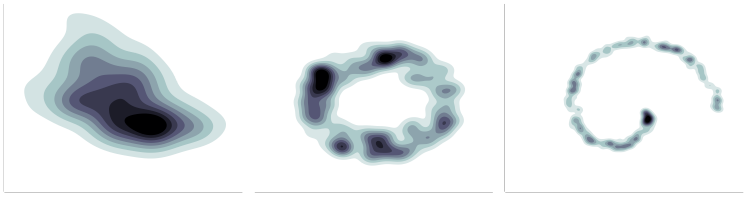
Tables 3.1, 3.2 and 3.3 show the optimal parameters for each method and the average ANLL of the test set, as well as the standard errors. The kernel density estimators perform similarly. As expected, the ordinary GMM and the DGMM perform worse when the target distribution is not a mixture. By contrast, the RGMM is competitive with the kernel methods, but has a much lower model complexity. Since the parameters are learnt by an iterative scheme, the method is sensitive to local maxima in the objective function, resulting in higher standard errors.

Influence of τ and ϵ

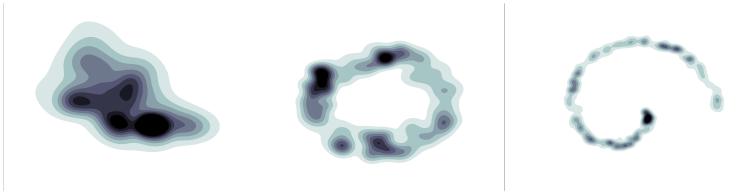
In this section, the influence of the value of parameters τ and ϵ is further discussed. The first column of Figure 3.11 shows the influence of an increasing number of components for each data set in the case of the RGMM. One can see that for each toy example, the higher the complexity, the higher the optimal value for τ is. The (slight) shift to the right of the minimum of the error curve expresses an increasing need for prior knowledge. When the number of components is high, compared to the number of learning data points or in presence of atypical observations, prior information is essential for obtaining reliable estimates of the covariance matrices.



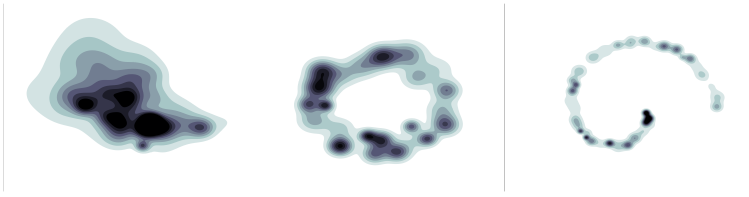
(a) Mixture. (b) Ring. (c) Spiral.



(d) KDE. (e) KDE. (f) KDE.



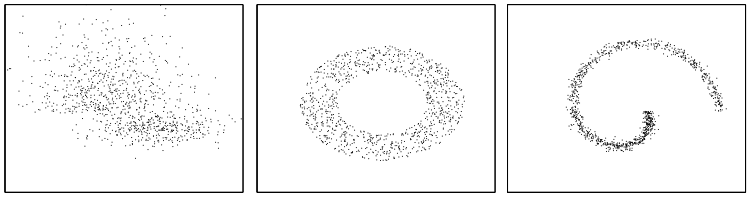
(g) SKDE. (h) SKDE. (i) SKDE.



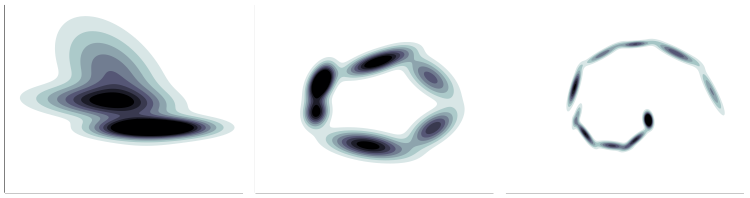
(j) VQKDE. (k) VQKDE. (l) VQKDE.

FIGURE 3.9. Optimal density estimators obtained for the KDE, the SKDE and the VQKDE. The first data set is a mixture of a bivariate Gaussian distribution and a Gaussian-Gamma distribution. The second and the third data sets are respectively the ring and the noisy spiral. The test set is shown on top.

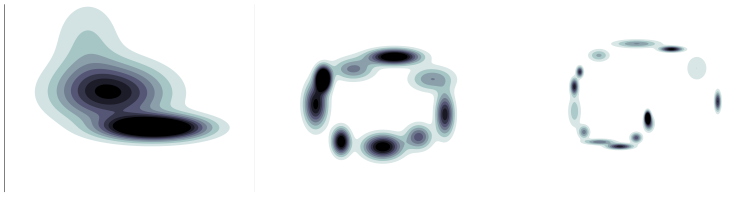
Finally, let us discuss the role of the safety factor ϵ. As suggested in Section 3.2.2, provided ϵ is chosen sufficiently small, it has no influence on the optimal performance of the RGMM. This is illustrated in the second column of Figure 3.11 for the three toy examples. When ϵ is sufficiently small, the iso-ANLL contours are independent of the value of ϵ, but only depend on the choice of τ. As a matter of fact, the safety factor just avoids the possible numerical



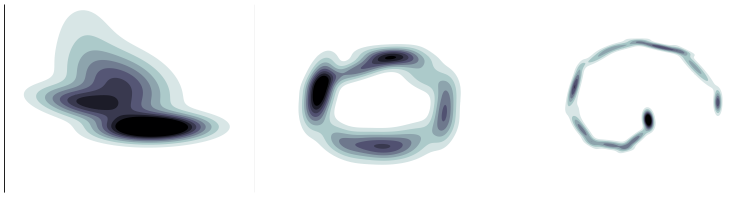
(a) Mixture. (b) Ring. (c) Spiral.



(d) GMM. (e) GMM. (f) GMM.



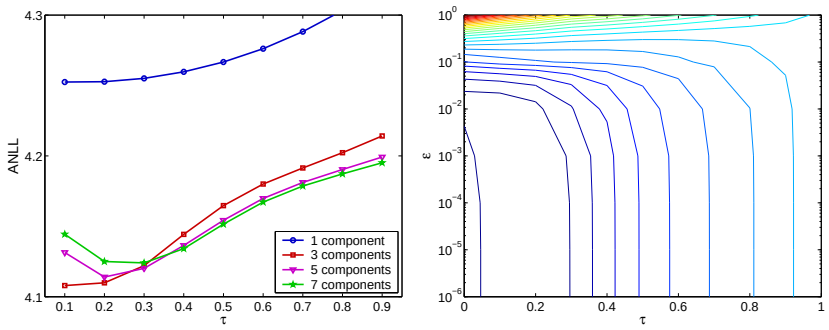
(g) DGMM. (h) DGMM. (i) DGMM.



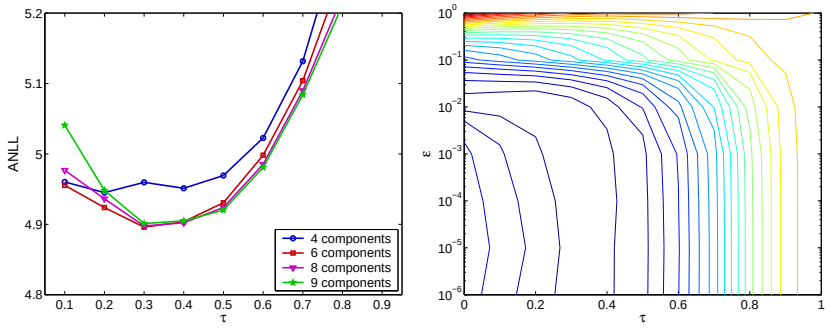
(j) RGMM. (k) RGMM. (l) RGMM.

FIGURE 3.10. Optimal density estimators obtained for the GMM, the DGMM and the RGMM. The first data set is a mixture of a bivariate Gaussian distribution and a Gaussian-Gamma distribution. The second and the third data sets are respectively the ring and the noisy spiral. The test set is shown on top.

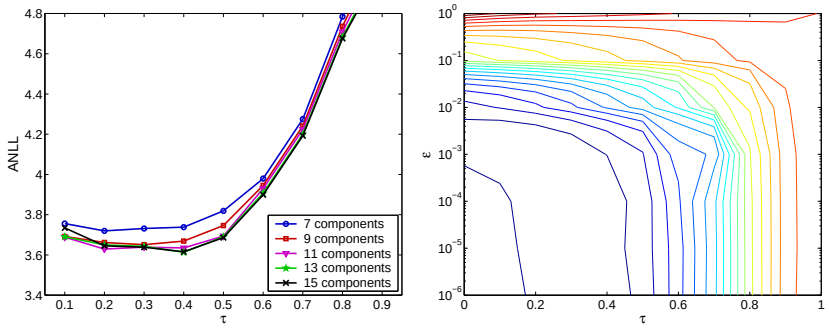
instabilities for a bad choice of τ. For example, a value of 0 corresponds to the classical multivariate GMM, which is known to be problematic in practice.



(a) Mixture. (b) Mixture.



(c) Ring. (d) Ring.



(e) Spiral. (f) Spiral.

FIGURE 3.11. For each experiment, the first column shows the expected ANLL (20 trials) of the test set with respect to τ, for different model complexities. The second column shows the iso-ANLL contours versus τ and ϵ. The model complexity is fixed to its optimal value.

TABLE 3.2. The ring data. The ANLL is evaluated on the test set and averaged over 20 runs.

		M	ANLL	std. err.
KDE	$\sigma = 0.16$	150	4.88	0.001
SKDE	$\sigma = 0.15$	150	4.89	0.002
VQKDE	$w = 1.60$	23	4.93	0.004
GMM		6	4.98	0.006
DGMM		5	5.03	0.002
RGMM	$\tau = 0.30$	6	4.89	0.005

TABLE 3.3. The noisy spiral data. The ANLL is evaluated on the test set and averaged over 20 runs.

		M	ANLL	std. err.
KDE	$\sigma = 0.07$	250	3.59	0.001
SKDE	$\sigma = 0.07$	250	3.60	0.002
VQKDE	$w = 1.50$	38	3.64	0.004
GMM		10	3.71	0.013
DGMM		13	3.88	0.007
RGMM	$\tau = 0.40$	13	3.61	0.010

Concluding remark

As the unconstrained GMM, the RGMM can model arbitrary densities, provided a sufficient number of components. The method can be used in conjunction with resampling or model averaging techniques (e.g., Breiman's bagging (Breiman, 1996)) in a practical and natural way, as the numerical difficulties encountered by EM are avoided. Besides, the approach provides high quality estimators as shown experimentally, as the amount of prior information needed to reliably estimate the kernel precisions is adapted by adjusting τ.

The RGMM also provides a flexible alternative compared to the DGMM, which are commonly used in practice. Diagonal constrained covariance matrices make GMM less sensitive and avoid components in the mixture to collapse, because their shape is determined by fewer parameters, but fail to provide high quality estimators.

Another regularization scheme is the maximum a posteriori GMM (Ormoneit and Tresp, 1998). In this approach, the covariance matrices are penalized according to a Wishart prior. Although the maximum a posteriori GMM have a stronger theoretical background, the main advantage of our approach is that, on the one hand, we only need to optimize one additional parameter τ, and

on the other hand, that its optimal value is in a closed interval. By contrast, the Wishart distribution is sensitive to two additional parameters (one of them being a d-dimensional matrix), which can range from zero to infinity. In the next section, the maximum a posteriori GMM is discussed in more detail and a variant is proposed in order to adjust the amount of penalization in a practical way.

3.2.3. Maximum a Posteriori Learning

A less heuristic way to deal with the problems encountered with ML learning and to obtain more consistent PDF estimators is to use a maximum a posteriori (MAP) approach. Maximizing the data likelihood by means of the EM algorithm does not necessarily correspond to computing the best possible model given the observed data. As the available data set is finite, it may be corrupted by noise and it is possibly sparse. In order to improve the quality of the estimators, MAP learning constraints the data likelihood according to some prior knowledge on the problem, and thus avoids poor local maxima of the unconstrained likelihood. As discussed in Section 3.2.5, it is essential in MAP learning to choose adequate priors for the model parameters. In general, they are chosen according to some belief on the form of a suitable solution.

Ormoneit and Tresp (1998) applied the MAP framework to the GMM by choosing the priors to be conjugate to the GMM. For a given density $p(\mathbf{x}|\boldsymbol{\theta})$, a prior $p(\boldsymbol{\theta})$ is said to be conjugate to $p(\mathbf{x}|\boldsymbol{\theta})$ if it gives rise to a posterior $p(\boldsymbol{\theta}|\mathbf{x})$ having the same functional form as $p(\boldsymbol{\theta})$. In other words, conjugacy is the property that the posterior distribution follows the same parametric form as the prior distribution. A nice property of conjugate priors is that they include non-informative priors as a limiting case. This property will be important in Section 3.2.5, where Bayesian learning of GMM is discussed.

The conjugate prior on the mixture proportions is a Dirichlet distribution (see for example Gelman, Carlin, Stern and Rubin, 1998):

$$\mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}) = c_{\mathcal{D}}(\boldsymbol{\kappa}) \prod_{m=1}^{M} {\pi_m}^{\kappa_m - 1} \;, \tag{3.101}$$

with $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{M}$ and $\boldsymbol{\kappa} = \{\kappa_m\}_{m=1}^{M}$. The normalizing constant $c_{\mathcal{D}}(\boldsymbol{\kappa})$ is defined as

$$c_{\mathcal{D}}\left(\boldsymbol{\kappa}\right) = \frac{\Gamma\left(\sum_{m=1}^{M} \kappa_m\right)}{\prod_{m=1}^{M} \Gamma(\kappa_m)} \;, \tag{3.102}$$

where $\Gamma(\cdot)$ is the Gamma function. In addition, $\boldsymbol{\kappa}$ satisfies the following constraints:

$$\forall m: \; \kappa_m \geq 0 \;, \quad \sum_{m=1}^{M} \kappa_m = N \;. \tag{3.103}$$

The conjugate prior on the mean and the precision of a single multivariate Gaussian component is the Gaussian-Wishart distribution (Gelman et al.,

1998):

$$\mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) = \mathcal{N}(\boldsymbol{\mu}_m|\mathbf{m}_m, \eta_m\boldsymbol{\Lambda}_m)\mathcal{W}(\boldsymbol{\Lambda}_m|\gamma_m, \mathbf{S}_m) \ , \tag{3.104}$$

with $\boldsymbol{\theta}_{\mathcal{NW}_m} = (\eta_m, \mathbf{m}_m, \gamma_m, \mathbf{S}_m)$. The Wishart distribution is given by

$$\mathcal{W}(\boldsymbol{\Lambda}_m|\gamma_m, \mathbf{S}_m) = c_{\mathcal{W}}\left(\gamma_m, \mathbf{S}_m\right)|\boldsymbol{\Lambda}_m|^{\frac{\gamma_m-d-1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\{\mathbf{S}_m\boldsymbol{\Lambda}_m\}\right) \ , \tag{3.105}$$

where $\gamma_m \geq d$, $\mathbf{S}_m$ is symmetric and positive definite, $\mathrm{tr}\{\cdot\}$ is the trace operator and $c_{\mathcal{W}}\left(\gamma_m, \mathbf{S}_m\right)$ is a normalizing constant. The normalizing constant $c_{\mathcal{W}}\left(\gamma_m, \mathbf{S}_m\right)$ is defined as

$$c_{\mathcal{W}}\left(\gamma_m, \mathbf{S}_m\right) = \frac{\pi^{\frac{-d(d-1)}{4}}|\mathbf{S}_m|^{\frac{\gamma_m}{2}}}{2^{\frac{\gamma_m d}{2}}\prod_{i=1}^{d}\Gamma\left(\frac{\gamma_m+1-i}{2}\right)} \ . \tag{3.106}$$

Based on this choice of priors, we can write the penalized data log-likelihood as follows:

$$\begin{aligned}\log\mathcal{L}_{\mathrm{MAP}}(\boldsymbol{\theta}_{\mathcal{N}}|X) &= \log p(X|\boldsymbol{\theta}_{\mathcal{N}}) + \log p(\boldsymbol{\theta}_{\mathcal{N}}) &(3.107)\\ &= \log\mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X) + \log\mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \\ &\quad + \textstyle\sum_{m=1}^{M}\log\mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) \ , &(3.108)\end{aligned}$$

where $p(\boldsymbol{\theta}_{\mathcal{N}})$ denotes the joint prior on the parameters of the GMM. This joint prior is given by

$$p(\boldsymbol{\theta}_{\mathcal{N}}) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa})\prod_{m=1}^{M}\mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) \ . \tag{3.109}$$

Similarly as in the ML case, the penalized log-likelihood cannot be maximized directly. However, defining the responsibilities as before,

$$\bar{\rho}_{nm} = \frac{\pi_m\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)}{\sum_{m'=1}^{M}\pi_{m'}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{m'}, \boldsymbol{\Lambda}_{m'})} \ , \quad \forall n \ , \quad \forall m \ , \tag{3.110}$$

and keeping these fixed during the maximization step leads to the following update rules for the component means, precisions and weights:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^{N}\bar{\rho}_{nm}\mathbf{x}_n + \eta_m\mathbf{m}_m}{\sum_{n=1}^{N}\bar{\rho}_{nm} + \eta_m} \ , \tag{3.111}$$

$$\begin{aligned}\boldsymbol{\Lambda}_m = \Bigg\{ &\frac{\sum_{n=1}^{N}\bar{\rho}_{nm}\left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)^{\mathrm{T}}}{\sum_{n=1}^{N}\bar{\rho}_{nm} + \gamma_m - d} \\ &+ \frac{\eta_m\left(\boldsymbol{\mu}_m - \mathbf{m}_m\right)\left(\boldsymbol{\mu}_m - \mathbf{m}_m\right)^{\mathrm{T}} + \mathbf{S}_m}{\sum_{n=1}^{N}\bar{\rho}_{nm} + \gamma_m - d}\Bigg\}^{-1} \ , &(3.112)\end{aligned}$$

$$\pi_m = \frac{\sum_{n=1}^{N}\bar{\rho}_{nm} + \kappa_m - 1}{N + \sum_{m'=1}^{M}\kappa_{m'} - M} \ . \tag{3.113}$$

MAP learning of the GMM consists thus in iteratively computing the responsibilities (E-step) and then maximizing the resulting penalized log-likelihood (M-step). Comparing the M-step of MAP (3.111–3.113) to the M-step of ML

(3.69–3.71), it can be observed that they are closely related. Obviously, the MAP step is a ML step that is penalized according to the hyperparameters of the priors. The choice of these parameters is therefore crucial in order to obtain a good model in practice.

Latent Variable Viewpoint

Consider again the latent variable model of the GMM:

$$p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}) = \prod_{m=1}^{M} {\pi_m}^{z_{nm}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)^{z_{nm}} \ . \tag{3.114}$$

As discussed in Section 3.1.2, it is straightforward to extend the EM algorithm to the MAP case, since the prior does not depend on the latent variables $Z = \{\mathbf{z}_n\}_{n=1}^N$. As a consequence, the E-step is unchanged and EM iteratively maximizes a penalized version of the expected complete data log-likelihood (subject to the constraint on the mixture proportions):

$$\begin{aligned} &\mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{N}}|X, Z)\} + \mathrm{E}_Z\{\log p(\boldsymbol{\theta}_{\mathcal{N}})\} \\ &\quad = \sum_{n=1}^{N}\sum_{m=1}^{M} \bar{\rho}_{nm} \left\{\log \pi_m + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\right\} + \log p(\boldsymbol{\theta}_{\mathcal{N}}) \ . \end{aligned} \tag{3.115}$$

3.2.4. Modified Maximum a Posteriori Learning

The main drawback in standard MAP learning is the prohibitive number of hyperparameters to set. For example for the GMM, up to $d(d+3)/2+1$ hyperparameters per component can be chosen. Usually, the same prior is imposed on the parameters of each component in the mixture. However, this still does not resolve the problem. Ormoneit and Tresp (1998) limited therefore their discussion to the effect of the Wishart prior on the precisions, the penalty terms due to the other priors being excluded from their simulations. A side effect of this analysis is that the power of the approach could not be fully appreciated, especially in the case of small learning sets.

Next, we propose to use the practical MAP framework presented in the second part of Section 3.1.2. In this approach, particular values are chosen for the hyperparameters of the joint prior (3.109) of the parameters. The amount of penalization is adjusted through the use of the regularization vector $\boldsymbol{\alpha}$.

Consider again the penalized log-likelihood defined in (3.108). Let $\boldsymbol{\vartheta}^*_{\mathcal{N}_m} = (\kappa^*_m, \boldsymbol{\theta}^*_{\mathcal{N}\mathcal{W}_m})$ be a particular choice of the hyperparameters for component m and the regularization vector $\boldsymbol{\alpha}$ be equal to $(\alpha_{\mathcal{D}}, \alpha_{\mathcal{N}\mathcal{W}})$, due to the factorized form of the joint prior. The modified MAP log-likelihood is then

$$\begin{aligned} \log \mathcal{L}_{\mathrm{PMAP}}(\boldsymbol{\theta}_{\mathcal{N}}|X) = \log \mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X) + \alpha_{\mathcal{D}} \log \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}^*) \\ + \alpha_{\mathcal{N}\mathcal{W}} \textstyle\sum_{m=1}^{M} \log \mathcal{N}\mathcal{W}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}^*_{\mathcal{N}\mathcal{W}_m}) \ . \end{aligned} \tag{3.116}$$

Applying the EM algorithm leaves the E-step unchanged, while the M-step becomes:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} \mathbf{x}_n + \alpha_{\mathcal{NW}} \eta_m^* \mathbf{m}_m^*}{\sum_{n=1}^{N} \bar{\rho}_{nm} + \alpha_{\mathcal{NW}} \eta_m^*} , \tag{3.117}$$

$$\boldsymbol{\Lambda}_m = \left\{ \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} (\mathbf{x}_n - \boldsymbol{\mu}_m)(\mathbf{x}_n - \boldsymbol{\mu}_m)^{\mathrm{T}}}{\sum_{n=1}^{N} \bar{\rho}_{nm} + \alpha_{\mathcal{NW}}(\gamma_m^* - d)} + \alpha_{\mathcal{NW}} \frac{\eta_m^* (\boldsymbol{\mu}_m - \mathbf{m}_m^*)(\boldsymbol{\mu}_m - \mathbf{m}_m^*)^{\mathrm{T}} + \mathbf{S}_m^*}{\sum_{n=1}^{N} \bar{\rho}_{nm} + \alpha_{\mathcal{NW}}(\gamma_m^* - d)} \right\}^{-1} , \tag{3.118}$$

$$\pi_m = \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} + \alpha_{\mathcal{D}}(\kappa_m^* - 1)}{N + \alpha_{\mathcal{D}}(\sum_{m'=1}^{M} \kappa_{m'}^* - M)} . \tag{3.119}$$

ML is recovered for $\boldsymbol{\alpha} = \mathbf{0}$ and the standard MAP is obtained for $\boldsymbol{\alpha} = \mathbf{1}$. When $\boldsymbol{\alpha}$ gets larger, the amount of penalization further increases until the prior information dominates. Thus, additional degrees of freedom are inserted in the estimation problem by means of $\boldsymbol{\alpha}$, such that the hyperparameters of the conjugate priors can be fixed according to some prior belief, while the amount of penalization can still be adjusted by learning the regularization vector in a classical way (e.g., resampling techniques). We discuss next how to choose these hyperparameters.

Prior on the mixture proportions

The Dirichlet distribution is the conjugate prior over the parameters of the multinomial distribution (see for example Gelman et al., 1998). The latter gives the probability of choosing a given collection of K items from a set of M items, with repetitions, and the probability of each item is given by $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{M}$. Therefore, we can easily see that for $K = N$ and since X is assumed i.i.d., the distribution of the mixture proportions is the multinomial distribution.

Parameter $\boldsymbol{\kappa}$ can be viewed as the vector of "prior observation counts" for events governed by $\boldsymbol{\pi}$. Therefore, an intuitive choice is to assign a priori the same number of data points to each component:

$$\forall m : \kappa_m^* = \frac{N}{M} . \tag{3.120}$$

This choice can be further motivated as follows. First, recall that $\sum_{m=1}^{M} \kappa_m^* = N$. The resulting expected value of the mixture proportions is given by

$$\mathrm{E}\{\pi_m\} = \frac{\kappa_m^*}{\sum_{m'=1}^{M} \kappa_{m'}^*} = \frac{1}{M} . \tag{3.121}$$

Second, the marginal distribution of each mixture proportion is the Beta distribution $\mathcal{B}(\pi_m | \kappa_m, \sum_{m=1}^{M} \kappa_m - \kappa_m)$ (Ferguson, 1973). Its mode is defined as

$$\mathrm{Mode}\{\pi_m\} = \frac{\kappa_m^* - 1}{\sum_{m'=1}^{M} \kappa_{m'}^* - 2} \approx \frac{1}{M} . \tag{3.122}$$

This approximation is valid as long as $M \ll N$, which is usually the case. Thus, imposing (3.120) allows us to incorporate the prior information that the components are equiprobable with the highest probability. In this way, the prior probability of each component matches its most likely value to its expected value. However, some probabilistic relaxation is still permitted around this particular value of the mixture proportions.

An interesting feature of this penalization scheme is that it indirectly acts on the location of the components by keeping them inside the data cloud. Our prior belief states that each Gaussian component is a priori equally likely. As a result, approximatively $1/M^{\text{th}}$ of the data points are associated to each component. Therefore, the approach prevents (to some extent) that infinite probability mass is put on a single datum and thus that a mixture proportion becomes too small. As discussed by Archambeau, Lee and Verleysen (2003), the collapsing mechanism is initiated when one of the component densities becomes highly improbable. By means of this penalization scheme on $\boldsymbol{\pi}$, we prevent the collapsing to happen.

Prior on the mixture means

In general, it is very delicate to introduce some prior belief in the GMM by choosing particular values for the hyperparameters $\{\mathbf{m}_m\}_{m=1}^M$. In fact, we usually have little prior information about the location of the mixture components. It is therefore recommended to use non-informative priors, i.e. broad priors, on the mixture means. This can be achieved by setting the parameters $\{\eta_m^*\}_{m=1}^M$ to a small value, e.g. 10^{-5}, and $\mathbf{m}_m^* \approx 0$, $\forall m$.

Prior on the mixture precisions

As mentioned in Section 3.2.2, the multivariate Gaussian distribution uses the Mahalanobis distance to determine its shape:

$$\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) = (\mathbf{x} - \boldsymbol{\mu}_m)^{\mathrm{T}} \boldsymbol{\Lambda}_m (\mathbf{x} - \boldsymbol{\mu}_m) \ . \tag{3.123}$$

In general, the problems faced with the unconstrained GMM are due to a singular covariance matrix when computing $\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$. This is likely to happen when the number of data points assigned to the corresponding component is less than or not considerably larger than the dimensionality of the feature space. As detailed above, this problem is indirectly addressed by introducing a penalization term on the mixture proportions. Nevertheless, penalizing the precisions directly is often more effective.

The penalization term on the precisions (or conversely on the covariance matrices) should achieve a tradeoff between hyperspherical and hyperellipsoidal component shapes (Mao and Jain, 1996; Archambeau and Verleysen, 2003). As already discussed in Section 3.2.2, the Euclidean distance favors hyperspherically shaped components of equal size. This leads to the undesirable effect of splitting large, as well as elongated data clouds unnecessarily. On the contrary,

the use of the Malahanobis distance causes components to absorb nearby small data clusters. This leads to unnatural large components, or forms unusually thin ones (for example when outliers exist in the database). According to this prior belief, regularity conditions are imposed on the shape of the components by means of the Wishart prior, resulting in a smoothness constraint on the estimator.

Consider again the Wishart prior on the component precisions. The following property holds:

$$\mathrm{E}\{{\boldsymbol{\Lambda}_m}^{-1}\} = \frac{\mathbf{S}_m}{\gamma_m - d - 1} \ . \tag{3.124}$$

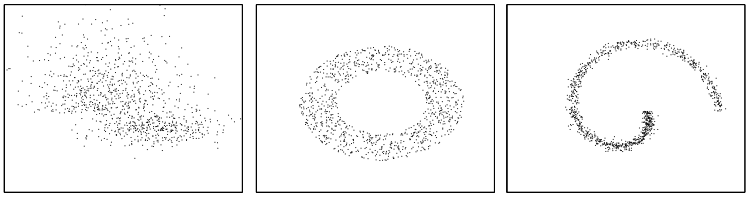
By choosing $\mathbf{S}_m$ and γ_m properly, the covariance matrix of each component can be penalized, such that it is unlikely that they are too elongated. Our prior belief suggests thus covariance matrices being diagonally shaped. In addition, for scaling purposes they should be proportional to the variance of the data and inversely proportional to the number of components. Furthermore, in order to achieve good generalization, they should sufficiently overlap. Accordingly, some dependency on the dimensionality of the data is included, and the following parameter values are proposed:

$$\forall m: \ \gamma_m^* = d + 2 \ , \quad \mathbf{S}_m^* = (\gamma_m^* - d - 1) \left(\frac{\hat{\sigma}_X}{\sqrt[2d]{M}} \right)^2 \mathbf{I} \ , \tag{3.125}$$

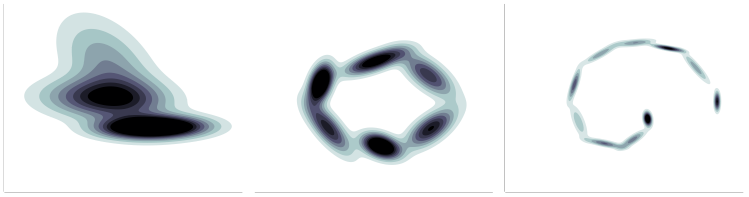
where $\mathbf{I}$ is the d-dimensional identity matrix and $\hat{\sigma}_X$ is the empirical standard deviation. The choice for γ_m^* corresponds to the less informative prior that is admissible (i.e. for which the expected value of the covariance matrix is positive). This is a natural choice as the goal of the modified MAP approach is to adjust the strength of the prior belief (and thus the penalization) by means of $\boldsymbol{\alpha}$.

Experimental validation

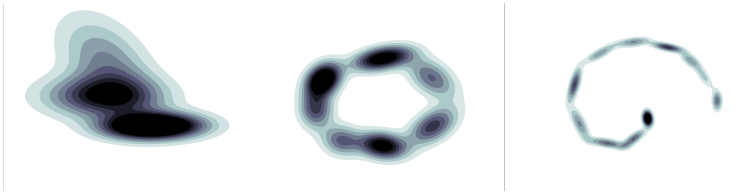
Consider the three toy examples used to assess the quality of the RGMM. Again, the aim is to do nonparametric-like density estimation. The estimators are shown in Figure 3.12. The test ANLL is shown in Tables 3.4, 3.5 and 3.6. In general, just imposing a prior on the mixture proportions prevents components to collapse. However, the quality of the estimator is similar to the quality of the estimator of the unconstrained GMM. By contrast, penalizing the precisions significantly improves the results. In fact, the resulting estimators are competitive with the standard kernel density estimators (KDE), while not making an intensive use of the memory resources. Note also that the results are slightly better than the estimators obtained with the RGMM. Finally, penalizing the mixture proportions and the precisions does not further improve the estimators.



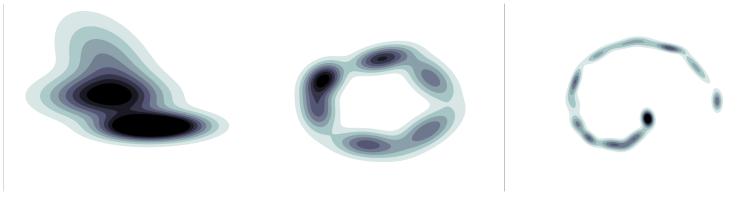
(a) Mixture. (b) Ring. (c) Spiral.



(d) $\alpha_{\mathcal{NW}} = 0$. (e) $\alpha_{\mathcal{NW}} = 0$. (f) $\alpha_{\mathcal{NW}} = 0$.



(g) $\alpha_{\mathcal{D}} = 0$. (h) $\alpha_{\mathcal{D}} = 0$. (i) $\alpha_{\mathcal{D}} = 0$.



(j) Full MAP. (k) Full MAP. (l) Full MAP.

FIGURE 3.12. Optimal density estimators obtained for the MAP GMM. Three variants are considered. First, the mixture proportions are penalized; second, the precisions are and third both are. The data sets are the mixture of a bivariate Gaussian distribution and a Gaussian-Gamma distribution, the ring data and the noisy spiral data. The test set is shown on top.

3.2.5. Variational Bayesian Learning

In Section 3.1.3, we have investigated how to learn latent variable models in the Bayesian framework. More specifically, it was shown how the variational Bayesian approach leads to EM-like update rules for the the posterior distribution of the latent variables and the model parameters by treating these as

TABLE 3.4. Mixture of a Gaussian and a Gaussian-Gamma distribution. The ANLL is evaluated on the test set and averaged over 20 runs.

		M	ANLL	std. err.
KDE	$\sigma = 0.35$	250	4.15	0.001
GMM		3	4.14	0.001
MAP GMM	$\alpha_{\mathcal{D}} = 50$, $\alpha_{\mathcal{NW}} = 0$	3	4.12	0.001
MAP GMM	$\alpha_{\mathcal{D}} = 0$, $\alpha_{\mathcal{NW}} = 0.1$	3	4.12	0.003
MAP GMM	$\alpha_{\mathcal{D}} = 50$, $\alpha_{\mathcal{NW}} = 0.1$	3	4.11	0.003

TABLE 3.5. The ring data. The ANLL is evaluated on the test set and averaged over 20 runs.

		M	ANLL	std. err.
KDE	$\sigma = 0.16$	150	4.88	0.001
GMM		6	4.98	0.006
MAP GMM	$\alpha_{\mathcal{D}} = 10$, $\alpha_{\mathcal{NW}} = 0$	5	4.97	0.010
MAP GMM	$\alpha_{\mathcal{D}} = 0$, $\alpha_{\mathcal{NW}} = 1$	7	4.87	0.003
MAP GMM	$\alpha_{\mathcal{D}} = 30$, $\alpha_{\mathcal{NW}} = 1$	7	4.88	0.009

TABLE 3.6. The noisy spiral data. The ANLL is evaluated on the test set and averaged over 20 runs.

		M	ANLL	std. err.
KDE	$\sigma = 0.07$	250	3.59	0.001
GMM		10	3.71	0.013
MAP GMM	$\alpha_{\mathcal{D}} = 20$, $\alpha_{\mathcal{NW}} = 0$	11	3.71	0.009
MAP GMM	$\alpha_{\mathcal{D}} = 0$, $\alpha_{\mathcal{NW}} = 0.1$	14	3.58	0.017
MAP GMM	$\alpha_{\mathcal{D}} = 20$, $\alpha_{\mathcal{NW}} = 0.1$	14	3.58	0.020

random latent variables as well. The main advantage of Bayesian inference is that the uncertainty on the model parameters is taken into account and that this approach allows to determine the optimal model complexity without having to resort to statistical resampling techniques. In this section, variational Bayes (VB) is applied to the GMM. As already mentioned, the GMM can be viewed as a latent variable model in the sense that we do not know by which component a data point is generated. The corresponding graphical model is shown in Figure 3.13. Since the model parameters are treated as random variables, they appear as nodes in the graph.

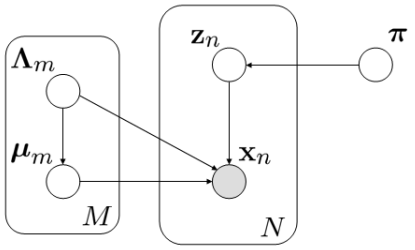


FIGURE 3.13. Graphical representation of the Bayesian GMM. In this model, it is assumed that the distribution on the mixture proportions and the joint distribution on the means and the precisions factorize, and that the means are conditionally dependent on the precisions.

In Bayesian learning, the quantity of interest is the incomplete data likelihood or model evidence. For a fixed model structure $\mathcal{H}_M$ of the GMM, this quantity is obtained by integrating out the latent variables Z and the parameters $\boldsymbol{\theta}_{\mathcal{N}}$:

$$p(X|\mathcal{H}_M) = \sum_Z \int p(X, Z, \boldsymbol{\theta}_{\mathcal{N}}|\mathcal{H}_M) d\boldsymbol{\theta}_{\mathcal{N}} \ . \tag{3.126}$$

For the GMM, this quantity is intractable. However, for any arbitrary density $q(Z, \boldsymbol{\theta}_{\mathcal{N}})$ a lower bound on the logarithm of the evidence can be found using Jensen's inequality:

$$\log p(X|\mathcal{H}_M) \geq \log p\left(X|\mathcal{H}_M\right) - \mathrm{KL}\left[q(Z, \boldsymbol{\theta}_{\mathcal{N}}) \| p\left(Z, \boldsymbol{\theta}_{\mathcal{N}}|X, \mathcal{H}_M\right)\right] \ . \tag{3.127}$$

The bound is made tight when $q(Z, \boldsymbol{\theta}_{\mathcal{N}})$ is equal to the joint posterior $p\left(Z, \boldsymbol{\theta}_{\mathcal{N}}|X, \mathcal{H}_M\right)$ of the latent variables and the parameters. In VB learning, the variational posterior approximates the joint posterior by assuming the latent variables and the parameters are independent:

$$q(Z, \boldsymbol{\theta}_{\mathcal{N}}) = q_Z(Z) q_{\boldsymbol{\theta}_{\mathcal{N}}}(\boldsymbol{\theta}_{\mathcal{N}}) \ . \tag{3.128}$$

Given this factorization, the lower bound on the log-evidence is tractable and the gap is minimized by minimizing the KL divergence between the true and the variational posterior. This is done iteratively by means of the VBEM algorithm (see Section 3.1.3):

$$\textbf{VBE-step} \ : q_{\mathbf{z}_n}(\mathbf{z}_n) \propto \exp\left(\mathrm{E}_{\boldsymbol{\theta}_{\mathcal{N}}}\{\log p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}, \mathcal{H}_M)\}\right) \ , \quad \forall n \ . \tag{3.129}$$

$$\textbf{VBM-step} : q_{\boldsymbol{\theta}_{\mathcal{N}}}(\boldsymbol{\theta}_{\mathcal{N}}) \propto p(\boldsymbol{\theta}_{\mathcal{N}}|\mathcal{H}_M) \exp\left(\mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{N}}|X, Z, \mathcal{H}_M)\}\right) \ . \tag{3.130}$$

In these equations, $\mathrm{E}_Z\{\cdot\}$ and $\mathrm{E}_{\boldsymbol{\theta}_{\mathcal{N}}}\{\cdot\}$ are respectively the expectation with respect to $q_Z(Z)$ and $q_{\boldsymbol{\theta}_{\mathcal{N}}}(\boldsymbol{\theta}_{\mathcal{N}})$. Remark that the posterior $q_Z(Z)$ factorizes, since X are i.i.d.

The complete data likelihood in the case of the GMM is given by

$$\mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{N}}|X, Z, \mathcal{H}_M) = \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}, \mathcal{H}_M) \ , \tag{3.131}$$

with

$$p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}, \mathcal{H}_M) = \prod_{m=1}^{M} {\pi_m}^{z_{nm}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)^{z_{nm}} \tag{3.132}$$

being the latent variable formulation of the GMM, in which the dependency on $\mathcal{H}_M$ is made explicit. Noting that $p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{N}}, \mathcal{H}_M)$ factorizes, it is likely that $\{q_{\mathbf{z}_n}(\mathbf{z}_n)\}_{n=1}^{N}$ factorize similarly:

$$q_{\mathbf{z}_n}(\mathbf{z}_n) = \prod_{m=1}^{M} q_{z_{nm}}(z_{nm})^{z_{nm}} \ , \quad \forall n \ . \tag{3.133}$$

Due to this factorized form, the VBE-step for the GMM simplifies to

$$q_{z_{nm}}(z_{nm} = 1) \propto \exp\left(\mathrm{E}_{\boldsymbol{\theta}_{\mathcal{N}}}\left\{\log \pi_m + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\right\}\right) \ . \tag{3.134}$$

These quantities correspond to the responsibilities in ML and MAP learning. Each of them is proportional to the posterior probability of having a component m when $\mathbf{x}_n$ is observed.

In order to compute the VBE-step, we need to know $q_{\boldsymbol{\theta}_{\mathcal{N}}}(\boldsymbol{\theta}_{\mathcal{N}})$. Looking at the VBM-step, one can see that taking the prior $p(\boldsymbol{\theta}_{\mathcal{N}}|\mathcal{H}_M)$ on the parameters as being conjugate to the exponential family is particularly attractive. In this case, the posterior and the prior have the same functional form. As a result, the VBM-step consists in simply updating the hyperparameters of the prior to the parameters of the posterior. As discussed in Section 3.2.3, the joint conjugate prior for the GMM is the product of a joint Dirichlet prior on the mixture proportions and Gaussian-Wishart distributions on the component means and precisions:

$$p(\boldsymbol{\theta}_{\mathcal{N}}|\mathcal{H}_M) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}_0) \prod_{m=1}^{M} \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_0}) \ , \tag{3.135}$$

where $\boldsymbol{\theta}_{\mathcal{NW}_0} = (\eta_0, \mathbf{m}_0, \gamma_0, \mathbf{S}_0)$ are particular values for the hyperparameters. In practice, they are chosen such that broad priors are obtained. Since the prior is a conjugate prior, the joint posterior has the same functional form and is thus also the product of a Dirichlet and Gaussian-Wishart distributions:

$$q_{\boldsymbol{\theta}_{\mathcal{N}}}(\boldsymbol{\theta}_{\mathcal{N}}) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \prod_{m=1}^{M} \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) \ , \tag{3.136}$$

where $\boldsymbol{\theta}_{\mathcal{NW}_m} = (\eta_m, \mathbf{m}_m, \gamma_m, \mathbf{S}_m)$. At this point, the expectation in the VBE-step can be computed since the form of the posterior is known. Recall that

$$\mathrm{E}\{(\mathbf{x} - \mathbf{m})^{\mathrm{T}}\mathbf{A}(\mathbf{x} - \mathbf{m})\} = (\boldsymbol{\mu} - \mathbf{m})^{\mathrm{T}}\mathbf{A}(\boldsymbol{\mu} - \mathbf{m}) + \mathrm{tr}\{\mathbf{A}\boldsymbol{\Lambda}^{-1}\} \ . \tag{3.137}$$

If $\mathbf{x} \sim \mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Lambda})$ and given that $\mathrm{E}_{\boldsymbol{\theta}_\mathcal{N}}\{\boldsymbol{\Lambda}_m\} = \gamma_m {\mathbf{S}_m}^{-1}$ under the Wishart prior, we have

$$\begin{aligned}&\mathrm{E}_{\boldsymbol{\theta}_\mathcal{N}}\left\{-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_m)^{\mathrm{T}}\boldsymbol{\Lambda}_m(\mathbf{x}_n - \boldsymbol{\mu}_m)\right\}\\ &\quad = -\frac{\gamma_m}{2}(\mathbf{x}_n - \mathbf{m}_m)^{\mathrm{T}}{\mathbf{S}_m}^{-1}(\mathbf{x}_n - \mathbf{m}_m) - \frac{d}{2\eta_m}\ .\end{aligned} \tag{3.138}$$

Substituting this result in (3.134) leads to the VBE-step for the GMM:

$$\begin{aligned}q_{z_{nm}}(z_{nm}=1) \propto\ &\tilde{\pi}_m(2\pi)^{-\frac{d}{2}}\tilde{\Lambda}_m{}^{\frac{1}{2}}\\ &\times\exp\left(-\frac{\gamma_m}{2}(\mathbf{x}_n - \mathbf{m}_m)^{\mathrm{T}}{\mathbf{S}_m}^{-1}(\mathbf{x}_n - \mathbf{m}_m) - \frac{d}{2\eta_m}\right)\ ,\end{aligned} \tag{3.139}$$

where the special quantities are defined as follows:

$$\log\tilde{\pi}_m \equiv \mathrm{E}_{\boldsymbol{\theta}_\mathcal{N}}\{\log\pi_m\} = \psi(\kappa_m) - \psi\left(\textstyle\sum_{m'=1}^{M}\kappa_{m'}\right)\ , \tag{3.140}$$

$$\log\tilde{\Lambda}_m \equiv \mathrm{E}_{\boldsymbol{\theta}_\mathcal{N}}\{\log|\boldsymbol{\Lambda}_m|\} = \sum_{i=1}^{d}\psi\left(\tfrac{\gamma_m+1-i}{2}\right) + d\log 2 - \log|\mathbf{S}_m|\ . \tag{3.141}$$

In these equations, $\psi(\cdot)$ is the digamma function. Taking into account the fact that $q_{\mathbf{z}_n}(\mathbf{z}_n)$ must be normalized for each data point $\mathbf{x}_n$ results in the responsibilities:

$$\bar{\rho}_{nm} = \frac{q_{z_{nm}}(z_{nm}=1)}{\sum_{m'=1}^{M} q_{z_{nm'}}(z_{nm'}=1)}\ ,\quad \forall n\ ,\quad \forall m\ . \tag{3.142}$$

Remark that the responsibilities have a very similar form as the quantities computed in the E-step (3.67) in ML learning.

Next, let us compute the VBM-step. Since

$$\begin{aligned}&\mathrm{E}_Z\{\log\mathcal{L}_c(\boldsymbol{\theta}_\mathcal{N}|X, Z, \mathcal{H}_M)\}\\ &\quad = \textstyle\sum_{n=1}^{N}\sum_{m=1}^{M}\bar{\rho}_{nm}\{\log\pi_m + \log\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\}\ ,\end{aligned} \tag{3.143}$$

we can identify from (3.130) the VBM update rules for the hyperparameters after some algebra:

$$\kappa_m = N\bar{\pi}_m + \kappa_0\ , \tag{3.144}$$

$$\eta_m = N\bar{\pi}_m + \eta_0\ , \tag{3.145}$$

$$\mathbf{m}_m = \frac{N\bar{\pi}_m\bar{\boldsymbol{\mu}}_m + \eta_0\mathbf{m}_0}{N\bar{\pi}_m + \eta_0}\ , \tag{3.146}$$

$$\gamma_m = N\bar{\pi}_m + \gamma_0\ , \tag{3.147}$$

$$\mathbf{S}_m = N\bar{\pi}_m\bar{\boldsymbol{\Sigma}}_m + \frac{N\bar{\pi}_m\eta_0}{\eta_m}(\bar{\boldsymbol{\mu}}_m - \mathbf{m}_0)(\bar{\boldsymbol{\mu}}_m - \mathbf{m}_0)^{\mathrm{T}} + \mathbf{S}_0\ , \tag{3.148}$$

where we have

$$\bar{\boldsymbol{\mu}}_m = \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} \mathbf{x}_n}{\sum_{n=1}^{N} \bar{\rho}_{nm}} \ , \tag{3.149}$$

$$\bar{\boldsymbol{\Sigma}}_m = \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} \left(\mathbf{x}_n - \bar{\boldsymbol{\mu}}_m\right) \left(\mathbf{x}_n - \bar{\boldsymbol{\mu}}_m\right)^{\mathrm{T}}}{\sum_{n=1}^{N} \bar{\rho}_{nm}}, \tag{3.150}$$

$$\bar{\pi}_m = \frac{1}{N} \sum_{n=1}^{N} \bar{\rho}_{nm} \ . \tag{3.151}$$

Note that $\boldsymbol{\Lambda}_m = {\boldsymbol{\Sigma}_m}^{-1}$, $\forall m$. The quantities (3.149–3.151) are the means of the posterior distributions and identical to the ML estimates of the parameters computed in the M-step of the ordinary EM algorithm. In fact, when $N \to \infty$, the posteriors collapse onto their means, and also $\tilde{\pi}_m = \bar{\pi}_m$, $\tilde{\Lambda}_m = |\boldsymbol{\Lambda}_m|$, $\forall m$. Thus in the limit, standard EM is recovered. Moreover, according to Attias (1999a), when the number of data points assigned to component m is 1 or less, i.e. $\bar{\pi}_m \leq 1/N$, VBEM sets $\bar{\pi}_m$ to zero, declaring the component non-existent. This property is important, as it protects the algorithm from putting infinite probability mass on a single data point, which is a well-known problem with ordinary EM. However, there will typically be multiple maxima in the variational bound, so different initializations may be beneficial in order to find a good maximum.

Predictive distribution

The variational predictive distribution is obtained by marginalizing the joint distribution $p(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{N}} | X, \mathcal{H}_M)$, using the variational posteriors instead of the true posteriors:

$$p(\mathbf{x}|X, \mathcal{H}_M) \approx \int p(\mathbf{x}|\boldsymbol{\theta}_{\mathcal{N}}, \mathcal{H}_M) q_{\boldsymbol{\theta}_{\mathcal{N}}}(\boldsymbol{\theta}_{\mathcal{N}}) d\boldsymbol{\theta}_{\mathcal{N}} \tag{3.152}$$

$$= \sum_{m=1}^{M} \breve{\pi}_m \mathcal{S}(\mathbf{x}|\mathbf{m}_m, \tfrac{\eta_m \nu_m}{\eta_m + 1} {\mathbf{S}_m}^{-1}, \nu_m) \ , \tag{3.153}$$

where $\nu_m = \gamma_m - d + 1$ and $\breve{\pi}_m = \kappa_m / \sum_{m'=1}^{M} \kappa_{m'}$. Parameter $\breve{\pi}$ is the expectation of π_m under the Dirichlet posterior. The distribution $\mathcal{S}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu)$ is the Student-t distribution with ν degrees of freedom, mean $\boldsymbol{\mu}$ and precision $\boldsymbol{\Lambda}$ (see Section 3.3 for a formal definition). Since in Bayesian learning overfitting is avoided by averaging over all models and weighting each model by its approximate posterior, it is rather intuitive that the predictive distribution is a mixture of Student-t distributions. Indeed, a Student-t distribution can be viewed as an infinite mixture of Gaussian distributions with the same mean and different precisions. When $N \to \infty$, the predictive distribution becomes a GMM.

Choosing the number of components

Before concluding this section, let us mention that the number of components in the mixture can be selected as the one maximizing the lower bound on the log-evidence $p(X|\mathcal{H}_M)$. This approach is appealing as it avoids the use of resampling techniques, which are wasteful of learning data.

Consider again the variational lower bound for the GMM:

$$\begin{aligned}\mathcal{F}_{\mathcal{H}_M}(q_Z(Z), q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N})) &= \sum_Z \int q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log \frac{p(X, Z, \boldsymbol{\theta}_\mathcal{N}|\mathcal{H}_M)}{q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N})} d\boldsymbol{\theta}_\mathcal{N} \\ &= \sum_Z \int q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log p(X|Z, \boldsymbol{\theta}_\mathcal{N}, \mathcal{H}_M) d\boldsymbol{\theta}_\mathcal{N} \\ &\quad + \sum_Z \int q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log p(Z|\boldsymbol{\theta}_\mathcal{N}, \mathcal{H}_M) d\boldsymbol{\theta}_\mathcal{N} \\ &\quad + \int q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log p(\boldsymbol{\theta}_\mathcal{N}|\mathcal{H}_M) d\boldsymbol{\theta}_\mathcal{N} \\ &\quad - \sum_Z q_Z(Z) \log q_Z(Z) \\ &\quad - \int q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) d\boldsymbol{\theta}_\mathcal{N} \ . \end{aligned} \tag{3.154}$$

The functional form of all the distributions appearing in this expression are known:

$$p(X|Z, \boldsymbol{\theta}_\mathcal{N}, \mathcal{H}_M) = \prod_{n=1}^N \prod_{m=1}^M \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)^{z_{nm}} \ , \tag{3.155}$$

$$p(Z|\boldsymbol{\theta}_\mathcal{N}, \mathcal{H}_M) = \prod_{n=1}^N \prod_{m=1}^M {\pi_m}^{z_{nm}} \ , \tag{3.156}$$

$$p(\boldsymbol{\theta}_\mathcal{N}|\mathcal{H}_M) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}_0) \prod_{m=1}^M \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_0}) \ , \tag{3.157}$$

$$q_Z(Z) = \prod_{n=1}^N \prod_{m=1}^M {\bar{\rho}_{nm}}^{z_{nm}} \ , \tag{3.158}$$

$$q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \prod_{m=1}^M \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) \ . \tag{3.159}$$

Each term of the lower bound can therefore be evaluated:

$$\begin{aligned}&\sum_Z \int q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log p(X|Z, \boldsymbol{\theta}_\mathcal{N}, \mathcal{H}_M) d\boldsymbol{\theta}_\mathcal{N} \\ &\quad = \textstyle\sum_{n=1}^N \sum_{m=1}^M \bar{\rho}_{nm} \Big\{ - \frac{d}{2} \log 2\pi + \frac{1}{2} \log \tilde{\Lambda}_m \\ &\qquad - \frac{\gamma_m}{2} (\mathbf{x}_n - \mathbf{m}_m)^\mathrm{T} {\mathbf{S}_m}^{-1} (\mathbf{x}_n - \mathbf{m}_m) - \frac{d}{2\eta_m} \Big\} \ , \end{aligned} \tag{3.160}$$

$$\sum_Z \int q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log p(Z|\boldsymbol{\theta}_\mathcal{N}, \mathcal{H}_M) \boldsymbol{\theta}_\mathcal{N}$$
$$= \sum_{n=1}^N \sum_{m=1}^M \bar{\rho}_{nm} \log \tilde{\pi}_m \ , \tag{3.161}$$

$$\int q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log p(\boldsymbol{\theta}_\mathcal{N}|\mathcal{H}_M) d\boldsymbol{\theta}_\mathcal{N}$$
$$= \log c_\mathcal{D}(\boldsymbol{\kappa}_0) + \sum_{m=1}^M (\kappa_0 - 1) \log \tilde{\pi}_m + \sum_{m=1}^M \Big\{ -\tfrac{d}{2} \log 2\pi$$
$$+ \tfrac{d}{2} \log \eta_0 - \tfrac{\gamma_m \eta_0}{2} (\mathbf{m}_m - \mathbf{m}_0)^\mathrm{T} {\mathbf{S}_m}^{-1} (\mathbf{m}_m - \mathbf{m}_0) - \tfrac{\eta_0 d}{2\eta_m}$$
$$+ \log c_{\mathcal{NW}}(\gamma_0, \mathbf{S}_0) + \tfrac{\gamma_0 - d}{2} \log \tilde{\Lambda}_m - \tfrac{\gamma_m}{2} \mathrm{tr}\{\mathbf{S}_0 {\mathbf{S}_m}^{-1}\} \Big\} \ , \tag{3.162}$$

$$\sum_Z q_Z(Z) \log q_Z(Z)$$
$$= \sum_{n=1}^N \sum_{m=1}^M \bar{\rho}_{nm} \log \bar{\rho}_{nm} \ , \tag{3.163}$$

$$\int q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) \log q_{\boldsymbol{\theta}_\mathcal{N}}(\boldsymbol{\theta}_\mathcal{N}) d\boldsymbol{\theta}_\mathcal{N}$$
$$= \log c_\mathcal{D}(\boldsymbol{\kappa}) + \sum_{m=1}^M (\kappa_m - 1) \log \tilde{\pi}_m + \sum_{m=1}^M \Big\{ -\tfrac{d}{2} \log 2\pi$$
$$+ \tfrac{d}{2} \log \eta_m - \tfrac{d}{2} + \log c_{\mathcal{NW}}(\gamma_m, \mathbf{S}_m) + \tfrac{\gamma_m - d}{2} \log \tilde{\Lambda}_m - \tfrac{\gamma_m d}{2} \Big\} \ . \tag{3.164}$$

The last two terms are the entropies of the variational distributions.

In order to illustrate the approach, the illustrative example shown in Figure 3.5 is considered. It is a mixture of three Gaussian distributions with different mean and different precisions. Hundred fifty data points are drawn from each component. The VBEM algorithm is run 10 times. The model complexity ranges from 1 to 5 components. Figure 3.14 shows the average lower bound on the log-evidence. One can observe that the number of components that maximizes the lower bound corresponds to the true number of components.

3.2.6. Related Approaches

An active field of research, yet unresolved, is the automatic selection of the number of components in the mixtures. For example, in VB learning, the number of components is selected according to the lower bound on the log-evidence (e.g., Attias, 1999b; Winn, 2003). However, the VB approximation leads in practice to favor too simple models as it tends to underestimate the variance of the true posterior. Moreover, VB assumes that the KL divergence between the variational posterior and the true posterior are the same for different model complexities. This is of course not true in practice.

A related approach was proposed by Corduneanu and Bishop (2001) in order to obtain sparse GMM. In this work, the mixture proportions are treated as parameters (thus not as latent variables). They are computed by maximizing the variational lower bound, which is conditioned on them. It is also assumed

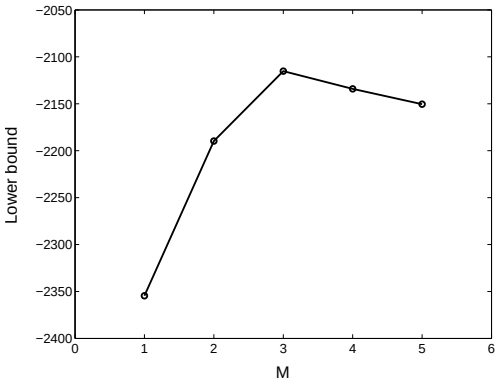


FIGURE 3.14. Lower bound on the log-evidence versus the number of components M. The curve shows the average over 10 trials. The maximum of the lower bound is obtained for 3 components, which is the true number of components from which the original data was drawn (see Figure 3.5).

that the joint prior on the means and the precisions, as well as their variational posterior further factorize. Note that the approach was only validated experimentally. In addition, it is worth mentioning that the standard variational GMM already prunes out excess components by setting the mixture weight of components having insufficient support equal to zero.

Apart from the VB approach, it was proposed to approximate the evidence, either directly in order to obtain model selection criteria (e.g., Roberts, Husmeier, Rezek and Penny, 1998), or by means of sampling techniques (e.g., Roeder and Wasserman, 1997). In particular,Green (1995) introduced reversible jump Markov chain Monte-Carlo (MCMC), which is capable of jumping between parameter spaces of different dimensionality. Applying this approach to mixture modeling (Richardson and Green, 1997) allows a fully Bayesian treatment of both, the parameters and the number of components, as the algorithm is able to jump between the parameter spaces of mixtures having a different number of components. However, it was argued that reversible jump MCMC, and sampling techniques in general, are rather slow (Figueiredo and Jain, 2002; Verbeek et al., 2003). In addition, it is difficult to assess convergence of MCMC and the posterior distribution is stored as a set of points, which can be inefficient.

A different approach was proposed by Figueiredo and Jain (2002). Rather than selecting one among a set of candidate models, the "best" model is directly selected in the entire set of available models on the basis of the minimum message length (MML) principle (Wallace and Freeman, 1987). In fact, the algorithm performs component annihilation such that excess components are pruned out of the mixture, not requiring multiple runs. The resulting objective

function takes the following form:

$$\log \mathcal{L}_{\text{MML}}(\boldsymbol{\theta}_{\mathcal{N}}|X) = \log \mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X) - \frac{k}{2} \sum_{\pi_m > 0} \log \frac{N\pi_m}{12} - \frac{M^*}{2} \log \frac{N}{12} - \frac{M^*(k+1)}{2} , \tag{3.165}$$

where $\mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X)$ is the incomplete likelihood defined in (3.66), M^* is the current number of components and k is the number of parameters specifying one Gaussian component. When using unconstrained precisions, k is equal to $d(d+3)/2$. Maximizing the MML objective function leads to update rules identical to the ones for standard EM, except for the mixture weights:

$$\pi_m = \frac{\max\{0, \sum_{n=1}^{N} \bar{\rho}_{nm} - \frac{k}{2}\}}{\sum_{m'=1}^{M} \max\{0, \sum_{n=1}^{N} \bar{\rho}_{nm'} - \frac{k}{2}\}} . \tag{3.166}$$

In practice, the algorithm is initialized with a large number of components. During training, the weights of the excess components are driven to zero. However, the component annihilation (3.166) does not take into account the additional increase in $\mathcal{L}_{\text{MML}}(\boldsymbol{\theta}_{\mathcal{N}}|X)$ caused by setting a component that is not annihilated to zero. Therefore, when a stable maximum of the objective function is attained, the least probable component is removed and the algorithm is rerun until convergence. This procedure is repeated until $M^* = 1$. The estimator is then chosen as the one that leads to the maximum value of $\mathcal{L}_{\text{MML}}(\boldsymbol{\theta}_{\mathcal{N}}|X)$. In practice, this approach may require an important amount of processing time.

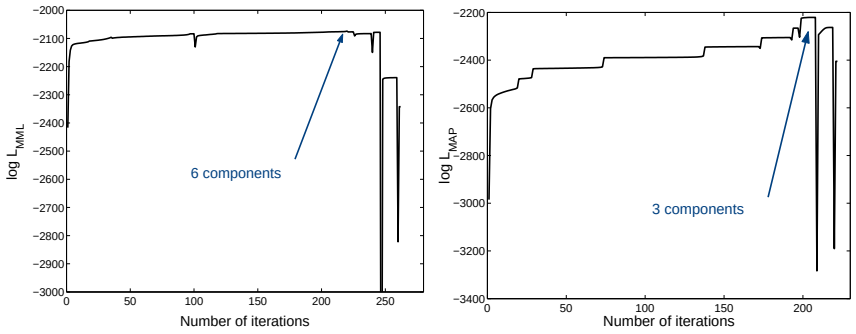
It is worth mentioning that the MML approach is related to the MAP as, for a fixed number of components M^*, it corresponds to imposing a flat prior on the means and precisions, and a Dirichlet-type prior (with negative parameters) on the weights:

$$p(\boldsymbol{\pi}) \propto \prod_{m=1}^{M} {\pi_m}^{-k/2} . \tag{3.167}$$

Interestingly, this framework can be extended to a MAP setting, i.e imposing informative priors on the means and the precisions. Consider a Gaussian-Wishart prior on these parameters. The resulting penalized log-likelihood takes the following form:

$$\log \mathcal{L}_{\text{MAP}}(\boldsymbol{\theta}_{\mathcal{N}}|X) = \log \mathcal{L}(\boldsymbol{\theta}_{\mathcal{N}}|X) - \frac{k}{2} \sum_{\pi_m > 0} \log \pi_m + \sum_{\pi_m > 0} \log \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) . \tag{3.168}$$

Maximizing this objective function leads to the same E-step and update rules for the means and the precisions as in MAP learning, while the update rule for the mixture proportion is identical to the one of the MML approach. Therefore, this MAP scheme is still sparsity inducing, for given hyperparameters, as most of the weights are driven to zero.



(a) MML GMM. (b) Sparse gen. MAP GMM.

FIGURE 3.15. Objective function for (a) MML and (b) sparse modified MAP versus the number of iterations. With the second method, the true number of components maximizes the objective function. The figure shows one trial. However, the algorithm was run 10 times. For each run, the true number of components was found. By contrast, the MML approach generally found 5 or 6 components.

Consider again the example of Figure 3.5 and let us apply the MML based GMM and the sparse modified MAP GMM. Figure 3.15 shows the corresponding objective functions as function of the number of iterations. It can be observed that when the number of data points is limited (in this example 150 per component), the MML approach fails to select the right number of components, which is 3. However, by using the MAP approach, the true number of components can be recovered. In this example, the regularization constant α is set to 3.

In a sense, the greedy EM of Verbeek et al. (2003) works opposite to the MML or sparse MAP approach. Indeed, the former starts with a small number of components and further builds the mixture component-wise. As a result, an interesting feature of the greedy approach is that it does not require to update a large number of parameters at the start of the algorithm.

Finally, note that a standard approach for choosing the number of mixture components is still applicable. The optimum is then chosen as the one minimizing a well-defined error criterion, e.g. the ANLL, which can be estimated by means of resampling techniques. Of course, this approach is wasteful of training data in some way.

3.3. Finite Student-*t* Mixture Models

A major limitation of the GMM is its lack of robustness to outliers. Providing robustness to outlying data is essential in many practical problems, since the

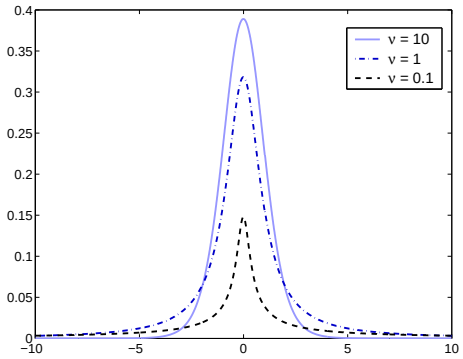


FIGURE 3.16. Univariate Student-t distribution with zero mean and unit precision. The robustness of the distribution increases for decreasing ν, i.e. the tails get heavier.

estimates of the means and precisions can be severely affected by atypical observations. In addition, in the case of the GMM, the presence of outliers or any other departure of the empirical distribution from Gaussianity can lead to selecting a false model complexity. More specifically, additional components are used (and needed) to capture the tails of the distribution.

Robustness can be introduced by embedding the Gaussian distribution in a wider class of elliptically symmetric distributions, called the Student-t distributions, which provide a heavy-tailed alternative to the Gaussian family:

$$\mathcal{S}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Lambda},\nu) = \frac{\Gamma\left(\frac{d+\nu}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)(\nu\pi)^{\frac{d}{2}}}|\boldsymbol{\Lambda}|^{\frac{1}{2}}\left[1+\frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Lambda}(\mathbf{x}-\boldsymbol{\mu})\right]^{-\frac{d+\nu}{2}} , \quad (3.169)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ are respectively the component mean and precision and $\Gamma(\cdot)$ denotes the gamma function. Parameter $\nu > 0$ are the degrees of freedom (df) and it can be viewed as a robustness tuning parameter. Its effect on the thickness of the distribution tails is shown in Figure 3.16. The smaller ν is, the heavier the tails are. When ν tends to infinity, the t-distribution tends to a Gaussian one.

A finite Student-t mixture model (SMM) is defined as follows:

$$p(\mathbf{x}|\boldsymbol{\theta}_{\mathcal{S}}) = \sum_{m=1}^{M}\pi_m\mathcal{S}(\mathbf{x}|\boldsymbol{\mu}_m,\boldsymbol{\Lambda}_m,\nu_m) , \quad (3.170)$$

where $\boldsymbol{\theta}_{\mathcal{S}} = (\pi_1,\ldots,\pi_M,\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_M,\boldsymbol{\Lambda}_1,\ldots,\boldsymbol{\Lambda}_M,\nu_1,\ldots,\nu_M)$. The mixing proportions $\{\pi_m\}_{m=1}^{M}$ are non-negative and must sum to 1.

3.3.1. Maximum Likelihood Learning

The SMM can be viewed as a latent variable model in the sense that the component label associated to each data point is unobserved. As for the GMM,

the set of indicator vectors are denoted by $Z = \{\mathbf{z}_n\}_{n=1}^N$, with $z_{nm} \in \{0,1\}$ and such that $\sum_{m=1}^M z_{nm} = 1$, $\forall n$. Furthermore, in the case of the SMM, the observed data X augmented by the indicator vectors Z is still incomplete. Indeed, the Student-t distribution can be written in the following form

$$\mathcal{S}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu) = \int_0^{+\infty} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, u\boldsymbol{\Lambda})\mathcal{G}(u|\tfrac{\nu}{2}, \tfrac{\nu}{2})du \ , \tag{3.171}$$

where $u > 0$ and the Gamma distribution is given by

$$\mathcal{G}(u|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} u^{\alpha-1} \exp(-\beta u) \ , \tag{3.172}$$

with $\alpha > 0$ and $\beta > 0$. Equation (3.171) can easily be verified by noting that the Gamma distribution is conjugate to the Gaussian distribution. Under this alternative representation, the Student-t distribution is thus an infinite mixture of Gaussian distributions with the same mean, but different precisions. The scaling factor u of the precisions is following a Gamma distribution with parameters depending only on ν. In contrast to the Gaussian distribution, there is no closed form solution for estimating the parameters of a single Student-t distribution based on the maximum likelihood principle. However, as discussed by Liu and Rubin (1995), the EM algorithm can be used to find an approximate ML solution by viewing u as an implicit latent variable, on which a Gamma prior is imposed. This result was extended to mixtures of Student-t distributions by Peel and McLachlan (2000). Here, for each data point $\mathbf{x}_n$ and for each component m, the scale variable u_{nm} given z_{nm} is unobserved. In the sequel, the set of scale vectors is denoted by $U = \{\mathbf{u}_n\}_{n=1}^N$.

The SMM is completely specified as follows:

$$p(\mathbf{z}_n|\boldsymbol{\theta}_\mathcal{S}) = \prod_{m=1}^M {\pi_m}^{z_{nm}} \ , \tag{3.173}$$

$$p(\mathbf{u}_n|\mathbf{z}_n, \boldsymbol{\theta}_\mathcal{S}) = \prod_{m=1}^M \mathcal{G}(u_{nm}|\tfrac{\nu_m}{2}, \tfrac{\nu_m}{2})^{z_{nm}} \ , \tag{3.174}$$

$$p(\mathbf{x}_n|\mathbf{u}_n, \mathbf{z}_n, \boldsymbol{\theta}_\mathcal{S}) = \prod_{m=1}^M \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, u_{nm}\boldsymbol{\Lambda}_m)^{z_{nm}} \ . \tag{3.175}$$

Marginalizing over the latent variables results indeed in (3.170):

$$p(\mathbf{x}_n|\boldsymbol{\theta}_\mathcal{S}) = \int \sum_{\mathbf{z}_n} p(\mathbf{x}_n|\mathbf{u}_n, \mathbf{z}_n, \boldsymbol{\theta}_\mathcal{S}) p(\mathbf{u}_n|\mathbf{z}_n, \boldsymbol{\theta}_\mathcal{S}) p(\mathbf{z}_n|\boldsymbol{\theta}_\mathcal{S}) d\mathbf{u}_n \tag{3.176}$$

$$= \int \sum_{\mathbf{z}_n} \prod_{m=1}^M \left\{\pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, u_{nm}\boldsymbol{\Lambda}_m)\mathcal{G}(u_{nm}|\tfrac{\nu_m}{2}, \tfrac{\nu_m}{2})\right\}^{z_{nm}} d\mathbf{u}_n \tag{3.177}$$

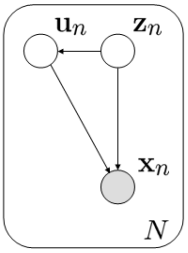


FIGURE 3.17. Graphical representation of the SMM. The shaded node is observed. The plate indicates N independent copies. The arrows represent conditional dependencies between the random variables.

$$= \int \sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, u_{nm}\boldsymbol{\Lambda}_m)\mathcal{G}(u_{nm}|\tfrac{\nu_m}{2}, \tfrac{\nu_m}{2})d\mathbf{u}_n \tag{3.178}$$

$$= \sum_{m=1}^{M} \pi_m \mathcal{S}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m, \nu_m) \ . \tag{3.179}$$

Figure 3.17 shows the directed acyclic graph of the SMM. Each observation $\mathbf{x}_n$ depends on the indicator vector $\mathbf{z}_n$ and the scale vector $\mathbf{u}_n$, which are both unobserved. The scale vectors are also conditionally dependent on the indicator variables.

As discussed in Section 3.1.1, the EM algorithm finds local ML estimates for the model parameters by alternating between an expectation and a maximization step. The E-step consists in computing the posterior distribution of the latent variables given the observations and the model parameters. The M-step maximizes the expected complete data log-likelihood with respect to the model parameters, the expectation being taken with respect to the posterior distributions computed in the E-step.

First, let us compute the posterior probability of the indicator variables. Since the marginal distribution $p(\mathbf{x}_n|z_{nm} = 1, \boldsymbol{\theta}_{\mathcal{S}})$ is equal to $\mathcal{S}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m, \nu_m)$, applying Bayes' rule leads to the posterior probability $P(z_{nm} = 1|\mathbf{x}_n, \boldsymbol{\theta}_{\mathcal{S}})$, termed responsibility:

$$\bar{\rho}_{nm} = \frac{\pi_m \mathcal{S}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m, \nu_m)}{\sum_{m=1}^{M} \pi_m \mathcal{S}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m, \nu_m)} \ , \quad \forall n, \quad \forall m \ . \tag{3.180}$$

These quantities correspond to the probability of having component m if $\mathbf{x}_n$ is observed.

Second, let us compute the posterior distribution of the scale variables. Using Bayes' rule we have:

$$p(u_{nm}|\mathbf{x}_n, z_{nm} = 1, \boldsymbol{\theta}_{\mathcal{S}}) \propto \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, u_{nm}\boldsymbol{\Lambda}_m)\mathcal{G}(u_{nm}|\tfrac{\nu_m}{2}, \tfrac{\nu_m}{2}) \ . \tag{3.181}$$

Since the Gamma distribution is conjugate to the Gaussian distribution, the posterior of each scale variable has the form of a Gamma distribution as well. It is then straightforward to show that

$$\begin{aligned} &p(u_{nm}|\mathbf{x}_n, z_{nm}=1, \boldsymbol{\theta}_{\mathcal{S}}) \\ &\quad = \mathcal{G}(u_{nm}|\tfrac{d+\nu_m}{2}, \tfrac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_m)^{\mathrm{T}}\boldsymbol{\Lambda}_m(\mathbf{x}_n-\boldsymbol{\mu}_m)+\tfrac{\nu_m}{2}) \ . \end{aligned} \tag{3.182}$$

The E-step for the scale variables consists thus in simply updating the parameters of the prior to the parameters of the posterior.

Next, let us compute the M-step. Given the latent variable formulation of the SMM, the complete data log-likelihood is given by

$$\begin{aligned} &\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{S}}|X,U,Z) \\ &\quad = \log \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n|\boldsymbol{\theta}_{\mathcal{S}}) \qquad (3.183) \\ &\quad = \log \ell_1(\boldsymbol{\pi}|Z) + \log \ell_2(\boldsymbol{\nu}|U,Z) + \log \ell_3(\boldsymbol{\theta}_{\mathcal{S}_1},\ldots,\boldsymbol{\theta}_{\mathcal{S}_M}|X,U,Z) \ , \qquad (3.184) \end{aligned}$$

where $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^M$, $\boldsymbol{\nu} = \{\nu_m\}_{m=1}^M$ and $\boldsymbol{\theta}_{\mathcal{S}_m} = (\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$, $\forall m$. The partial log-likelihood terms in (3.184) are defined as follows:

$$\log \ell_1(\boldsymbol{\pi}|Z) = \sum_{n=1}^{N}\sum_{m=1}^{M} z_{nm} \log \pi_m \ , \tag{3.185}$$

$$\log \ell_2(\boldsymbol{\nu}|U,Z) = \sum_{n=1}^{N}\sum_{m=1}^{M} z_{nm} \log \mathcal{G}(u_{nm}|\tfrac{\nu_m}{2}, \tfrac{\nu_m}{2}) \ , \tag{3.186}$$

$$\log \ell_3(\boldsymbol{\theta}_{\mathcal{S}_1},\ldots,\boldsymbol{\theta}_{\mathcal{S}_M}|X,U,Z) = \sum_{n=1}^{N}\sum_{m=1}^{M} z_{nm} \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, u_{nm}\boldsymbol{\Lambda}_m) \ , \tag{3.187}$$

Taking expectations with respect to the posterior distribution of the latent variables leads to:

$$\begin{aligned} &\mathrm{E}_{U,Z}\{\log \ell_1(\boldsymbol{\pi}|Z)\} \\ &\qquad = \textstyle\sum_{n=1}^{N}\sum_{m=1}^{M} \bar{\rho}_{nm} \log \pi_m \ , \qquad (3.188) \\ &\mathrm{E}_{U,Z}\{\log \ell_2(\boldsymbol{\nu}|U,Z)\} \\ &\qquad = \textstyle\sum_{n=1}^{N}\sum_{m=1}^{M} \bar{\rho}_{nm} \Big\{ \frac{\nu_m}{2}\log\frac{\nu_m}{2} - \log\Gamma\left(\frac{\nu_m}{2}\right) \\ &\qquad\quad + \left(\tfrac{\nu_m}{2}-1\right)\log\tilde{u}_{nm} - \tfrac{\nu_m}{2}\bar{u}_{nm} \Big\} \ , \qquad (3.189) \\ &\mathrm{E}_{U,Z}\{\log \ell_3(\boldsymbol{\theta}_{\mathcal{S}_1},\ldots,\boldsymbol{\theta}_{\mathcal{S}_M}|X,U,Z)\} \\ &\qquad = \textstyle\sum_{n=1}^{N}\sum_{m=1}^{M} \bar{\rho}_{nm} \Big\{ -\frac{d}{2}\log 2\pi + \frac{d}{2}\log\tilde{u}_{nm} \\ &\qquad\quad + \tfrac{1}{2}\log|\boldsymbol{\Lambda}_m| - \tfrac{\bar{u}_{nm}}{2}(\mathbf{x}_n-\boldsymbol{\mu}_m)^{\mathrm{T}}\boldsymbol{\Lambda}_m(\mathbf{x}_n-\boldsymbol{\mu}_m) \Big\} \ , \qquad (3.190) \end{aligned}$$

where we use the fact that $\mathrm{E}_Z\{z_{nm}\} = \bar{\rho}_{nm}$ and where the special quantities $\bar{u}_{nm}$ and $\log\tilde{u}_{nm}$ are respectively equal to $\mathrm{E}_U\{u_{nm}\}$ and $\mathrm{E}_U\{\log u_{nm}\}$. These

quantities can be computed using the properties of the Gamma distribution:

$$\bar{u}_{nm} = \frac{d + \nu_m}{(\mathbf{x}_n - \boldsymbol{\mu}_m)^{\mathrm{T}} \boldsymbol{\Lambda}_m (\mathbf{x}_n - \boldsymbol{\mu}_m) + \nu_m} \ , \tag{3.191}$$

$$\log \tilde{u}_{nm} = \psi\left(\frac{d + \nu_m}{2}\right) - \log\left\{\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_m)^{\mathrm{T}} \boldsymbol{\Lambda}_m (\mathbf{x}_n - \boldsymbol{\mu}_m) + \frac{\nu_m}{2}\right\} \ . \tag{3.192}$$

Finally, maximizing $\mathrm{E}_{U,Z}\{\mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{S}}|X, U, Z)\}$ with respect to $\boldsymbol{\theta}_{\mathcal{S}}$ and subject to the constraint on the mixture proportions results in the M-step for the SMM:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} \bar{u}_{nm} \mathbf{x}_n}{\sum_{n=1}^{N} \bar{\rho}_{nm} \bar{u}_{nm}} \ , \tag{3.193}$$

$$\boldsymbol{\Lambda}_m = \left\{\frac{\sum_{n=1}^{N} \bar{\rho}_{nm} \bar{u}_{nm} (\mathbf{x}_n - \boldsymbol{\mu}_m)(\mathbf{x}_n - \boldsymbol{\mu}_m)^{\mathrm{T}}}{\sum_{n=1}^{N} \bar{\rho}_{nm}}\right\}^{-1} \ , \tag{3.194}$$

$$\pi_m = \frac{1}{N} \sum_{n=1}^{N} \bar{\rho}_{nm} \ . \tag{3.195}$$

For the df however, there is no closed form solution. Therefore, we should seek, at each iteration and for each component, the root of the following equation:

$$\log \frac{\nu_m}{2} + 1 - \psi\left(\frac{\nu_m}{2}\right) + \frac{1}{N\pi_m} \sum_{n=1}^{N} \bar{\rho}_{nm} \{\log \tilde{u}_{nm} - \bar{u}_{nm}\} = 0 \ . \tag{3.196}$$

Liu and Rubin (1995) proposed to solve this equation by line search in the case of a single Student-t distribution, but noted that the EM algorithm converges slowly. In addition, the approach is computationally expensive. When it can be assumed that the df is identical for all the components, resampling techniques can be used. Shoham (2002), who discussed a deterministic annealing EM scheme for the SMM, proposed a heuristic for approximating (3.196) when using the same df for all components:

$$\nu \approx \frac{2}{y + \log y - 1} + 0.0416 \left\{1 + \mathrm{erf}\left(0.6594 \log \frac{2.1971}{y - \log y - 1}\right)\right\} \ , \tag{3.197}$$

where

$$y = -\frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} \bar{\rho}_{nm} \{\log \tilde{u}_{nm} - \bar{u}_{nm}\} \ . \tag{3.198}$$

In practice, this approximation turns out to be relatively accurate. Interestingly, the rule can be generalized to the case where the components have different df by simply replacing y with y_m:

$$y_m = -\frac{1}{N\pi_m} \sum_{n=1}^{N} \bar{\rho}_{nm} \{\log \tilde{u}_{nm} - \bar{u}_{nm}\} \ , \quad \forall m \ . \tag{3.199}$$

While this heuristic is more flexible in practice, it is also less robust in very noisy environments. This is due to the fact that fewer data points contribute to the computation of y_m than to the computation of y.

When looking at (3.193) and (3.195), the resemblance with the M-step of the GMM is obvious: the means and the precisions are computed by weighting the data points according to the responsibilities and when ν_m tends to infinity, the M-step of the GMM is recovered since

$$\lim_{\nu_m \to +\infty} \bar{u}_{nm} = 1 \ . \tag{3.200}$$

However, in contrast to the GMM, outliers are here downweighted due to the factor $\bar{u}_{nm}$. From (3.191) it can be seen that the downweighting (and thus the robustness) increases when ν_m decreases. A related approach was proposed by Markatou (2000), which is based on the weighted likelihood methodology (Green, 1984; Markatou, Basu and Lindsay, 1998). In this method, robustness is introduced by weighting the likelihood of each observation according to a weight function, which is defined in terms of the Pearson residuals. The approach was only established in the context of univariate mixture models.

To conclude, note that the convergence rate of the EM algorithm for the SMM can be improved. In case of a single Student-t distribution, Kent, Tyler and Vardi (1994) proposed to replace the normalizing constant in (3.194) by

$$\sum_{n=1}^{N} \bar{\rho}_{nm} \bar{u}_{nm} \ . \tag{3.201}$$

It was reported that the resulting EM steps converge faster (e.g., Kent et al., 1994; Meng and van Dyk, 1997). The approach was also used in the context of SMM by Peel and McLachlan (2000) and Shoham (2002).

3.3.2. Learning with the Regularized Mahalanobis distance

When approximating an unknown PDF by increasing the number of components arbitrarily, numerical difficulties might occur with the SMM as well. As in the case of the GMM, maximizing the data log-likelihood in the context of SMM is an ill-posed problem, since the width of a component may still tend to zero when it comes near an isolated data point (see for example Archambeau, Lee and Verleysen, 2003). With SMM, this only happens when a component is badly initialized or when the learning set contains lots of outliers. Yet, if sufficient data are available and the singularities of the likelihood function can be avoided, we may approximate the true PDF arbitrarily well. In order to recover from singular precisions, Archambeau, Vrins and Verleysen (2004) proposed to extend the use the regularized Mahalanobis distance in the frame of the SMM.

Modified M-step

As the Gaussian distribution, the Student-t distribution uses the Mahalanobis distance to determine its shape. For each component we have

$$\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) = (\mathbf{x} - \boldsymbol{\mu}_m)^{\mathrm{T}} \boldsymbol{\Lambda}_m (\mathbf{x} - \boldsymbol{\mu}_m) \ . \tag{3.202}$$

In Section 3.2.2, a regularized distance is constructed like the convex combination of the Euclidean distance $\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \mathbf{I})$, which favors hyperspherical components, and the Mahalanobis distance $\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$, which favors hyperellipsoidal components. A similar approach is used here:

$$\Delta'(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) = (1-\tau)\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) + \tau\Delta(\mathbf{x}|\boldsymbol{\mu}_m, \mathbf{I}) \ , \tag{3.203}$$

where $\tau \in [0,1]$ controls the trade-off between both distance measures. Next, consider the M-step of the unconstrained SMM. The regularized Mahalanobis distance is introduced in the M-step by adapting, at each iteration step, the precision $\boldsymbol{\Lambda}_m$ of each component according to (3.203). Therefore, the update rule of the component precisions becomes:

$$\left\{ \begin{array}{rcl} \boldsymbol{\Sigma}_m & = & \frac{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm}(\mathbf{x}_n - \boldsymbol{\mu}_m)(\mathbf{x}_n - \boldsymbol{\mu}_m)^{\mathrm{T}}}{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm}} \\ \boldsymbol{\Lambda}_m & = & (1-\tau)\left(\boldsymbol{\Sigma}_m + \epsilon\mathbf{I}\right)^{-1} + \tau\lambda\mathbf{I} \ . \end{array} \right. \ , \tag{3.204}$$

Parameter ϵ is the safety factor. The scaling factor λ takes the range of the data into account. This parameter can be computed according to (3.99), which is a rule-of-thumb reflecting our prior belief about the expected precision of each kernel.

3.3.3. Maximum a Posteriori Learning

As mentioned in the previous section, although the SMM is robust to outliers, it may still be attractive to constrain its parameters in order to improve the generalization capabilities of the resulting estimator. In fact, the SMM is successful when few atypical data occur in the data set, but their quality reduces when the number of atypical data increases or when the data set is sparse.

Alike the GMM, some prior information can be introduced when using the SMM. Besides, this is particularly suited when using SMM for nonparametric-like PDF estimation. Assuming a Dirichlet prior on the mixture proportions, Gaussian-Wishart priors on the component means and precisions and exponential priors on the df, the joint prior on the parameters takes the following form:

$$p(\boldsymbol{\theta}_{\mathcal{S}}) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \prod_{m=1}^{M} \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) \prod_{m=1}^{M} \mathcal{E}(\nu_m|\lambda_m) \ , \tag{3.205}$$

where $\boldsymbol{\kappa} = \{\kappa_m\}_{m=1}^M$ and $\boldsymbol{\theta}_{\mathcal{NW}_m} = (\eta_m, \mathbf{m}_m, \gamma_m, \mathbf{S}_m)$, $\forall m$. The exponential distribution is given by

$$\mathcal{E}(\nu|\lambda) = \lambda \exp(-\lambda\nu) \ , \tag{3.206}$$

with $\nu \geq 0$ and $\lambda > 0$.

When using the EM algorithm for MAP learning, it maximizes iteratively the expected complete data log-likelihood, augmented by a penalization term equal to the logarithm of the prior on the parameters:

$$\mathrm{E}_{U,Z}\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{S}}|X, U, Z)\} + \log p(\boldsymbol{\theta}_{\mathcal{S}}) \ . \tag{3.207}$$

Since this term does not depend on the latent variables U and Z, the E-step is unchanged. The M-step is obtained by maximizing this expression with respect to the parameters (and subject to the constraint on the mixture proportions). The MAP update rules for the SMM are:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm}\mathbf{x}_n + \eta_m \mathbf{m}_m}{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm} + \eta_m} \;, \tag{3.208}$$

$$\boldsymbol{\Lambda}_m = \left\{ \frac{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm} \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)^{\mathrm{T}}}{\sum_{n=1}^N \bar{\rho}_{nm} + \gamma_m - d} + \frac{\eta_m \left(\boldsymbol{\mu}_m - \mathbf{m}_m\right)\left(\boldsymbol{\mu}_m - \mathbf{m}_m\right)^{\mathrm{T}} + \mathbf{S}_m}{\sum_{n=1}^N \bar{\rho}_{nm} + \gamma_m - d} \right\}^{-1} \;, \tag{3.209}$$

$$\pi_m = \frac{\sum_{n=1}^N \bar{\rho}_{nm} + \kappa_m - 1}{N + \sum_{m'=1}^M \kappa_{m'} - M} \;. \tag{3.210}$$

The M-step for the df is given by

$$\log \frac{\nu_m}{2} + 1 - \psi\left(\frac{\nu_m}{2}\right) + \frac{1}{N\pi_m} \sum_{n=1}^N \bar{\rho}_{nm} \left\{\log \tilde{u}_{nm} - \bar{u}_{nm}\right\} - \frac{\lambda_m}{N\pi_m} = 0 \;. \tag{3.211}$$

3.3.4. Modified Maximum a Posteriori Learning

The main drawback in MAP learning is the prohibitive number of hyperparameters. Let us therefore handle the problem in a practical way by means of the modified MAP (see Section 3.1.2). In this approach, particular values are chosen for the hyperparameters of the joint prior of the parameters, the amount of penalization being adjusted by the regularization vector $\boldsymbol{\alpha}$.

Consider again the expected penalized complete data log-likelihood defined in (3.207). Let $\boldsymbol{\vartheta}^*_{\mathcal{S}_m} = (\kappa^*_m, \boldsymbol{\theta}^*_{\mathcal{NW}_m}, \lambda^*_m)$ be a particular choice of the hyperparameters for component m and the regularization vector $\boldsymbol{\alpha}$ be equal to $(\alpha_{\mathcal{D}}, \alpha_{\mathcal{NW}}, \alpha_{\mathcal{E}})$. The objective function for modified MAP is given by

$$\begin{aligned} &\mathrm{E}_{U,Z}\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{S}}|X,U,Z)\} + \alpha_{\mathcal{D}} \log \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}^*) \\ &+ \alpha_{\mathcal{NW}} \sum_{m=1}^M \log \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}^*_{\mathcal{NW}_m}) + \alpha_{\mathcal{E}} \sum_{m=1}^M \log \mathrm{E}(\nu_m|\lambda^*_m) \;. \end{aligned} \tag{3.212}$$

Applying the EM algorithm leads to a modified M-step:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm}\mathbf{x}_n + \alpha_{\mathcal{NW}}\eta^*_m \mathbf{m}^*_m}{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm} + \alpha_{\mathcal{NW}}\eta^*_m} \;, \tag{3.213}$$

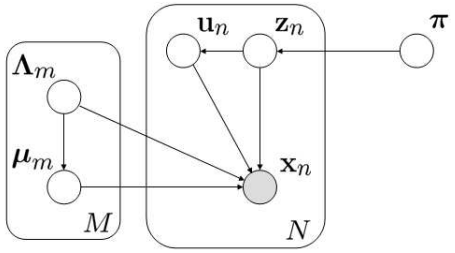


FIGURE 3.18. Graphical model of the Bayesian SMM.

$$\boldsymbol{\Lambda}_m = \left\{ \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} \bar{u}_{nm} \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)^{\mathrm{T}}}{\sum_{n=1}^{N} \bar{\rho}_{nm} + \alpha_{\mathcal{NW}}(\gamma_m^* - d)} + \alpha_{\mathcal{NW}} \frac{\eta_m^* \left(\boldsymbol{\mu}_m - \mathbf{m}_m^*\right) \left(\boldsymbol{\mu}_m - \mathbf{m}_m^*\right)^{\mathrm{T}} + \mathbf{S}_m^*}{\sum_{n=1}^{N} \bar{\rho}_{nm} + \alpha_{\mathcal{NW}}(\gamma_m^* - d)} \right\}^{-1} , \tag{3.214}$$

$$\pi_m = \frac{\sum_{n=1}^{N} \bar{\rho}_{nm} + \kappa_m^* - 1}{N + \sum_{m'=1}^{M} \kappa_{m'}^* - M} , \tag{3.215}$$

$$\log \frac{\nu_m}{2} + 1 - \psi\left(\frac{\nu_m}{2}\right) + \frac{1}{N\pi_m} \sum_{n=1}^{N} \bar{\rho}_{nm} \left\{\log \tilde{u}_{nm} - \bar{u}_{nm}\right\} - \alpha_{\mathcal{E}} \frac{\lambda_m^*}{N\pi_m} = 0 . \tag{3.216}$$

The corresponding ML and MAP steps are respectively recovered for $\boldsymbol{\alpha} = \mathbf{0}$ and $\boldsymbol{\alpha} = \mathbf{1}$. The choices proposed in Section 3.2.4 for κ_m^* and $\boldsymbol{\theta}_{\mathcal{NW}_m}^*$ can still be used for the SMM. For df, using $\alpha_{\mathcal{E}}$ does not simplify the problem, so it is adviced to optimize λ_m directly. In practice, the same choice is made for all the components.

3.3.5. Variational Bayesian Learning

The SMM is a latent variable model. Both the indicator variables and the scale variables are unobserved. In this section, we discuss how to estimate the parameters of the SMM in the Bayesian setting, and more specifically by means of the VBEM algorithm. In the Bayesian approach, the parameters are treated as latent random variables as well. The graphical model of the Bayesian SMM is shown in Figure 3.18. Note that the parameters appear as nodes in the graph. In contrast to the work of Svensén and Bishop (2004), it is not assumed that the scale variables are independent from the indicator variables. Therefore, the correlation between the indicator variables and the scale variables are not unnecessarily neglected, leading to different update rules for the variational distributions.

Recall that the aim in Bayesian learning is to compute (or approximate) the evidence. This quantity is obtained by integrating out all the latent variables.

For a fixed model structure $\mathcal{H}_M$ of the SMM, this quantity is given by

$$p(X|\mathcal{H}_M) = \sum_Z \int\int p(X, U, Z, \boldsymbol{\theta}_\mathcal{S}|\mathcal{H}_M) dU d\boldsymbol{\theta}_\mathcal{S} \ . \tag{3.217}$$

As in the case of the GMM, the evidence is intractable in practice. However, by assuming a factorized approximation of the joint posterior of the latent variables and the parameters, a tractable lower bound on the logarithm of the evidence can be constructed. Using Jensen's inequality we have

$$\log p(X|\mathcal{H}_M) \geq \log p\,(X|\mathcal{H}_M) - \mathrm{KL}\left[q(U, Z, \boldsymbol{\theta}_\mathcal{S}) \| p\,(U, Z, \boldsymbol{\theta}_\mathcal{S}|X, \mathcal{H}_M)\right] \ . \tag{3.218}$$

In VB learning, the arbitrary distribution $q(U, Z, \boldsymbol{\theta}_\mathcal{S})$ is chosen as a factorized approximation of the joint posterior $p\,(U, Z, \boldsymbol{\theta}_\mathcal{S}|X, \mathcal{H}_M)$. The resulting variational posterior is given by

$$q(U, Z, \boldsymbol{\theta}_\mathcal{S}) = q_{U,Z}(U, Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \ . \tag{3.219}$$

Given this factorized form, the lower bound on the log-evidence is tractable. Furthermore, since X are i.i.d., $q_{U,Z}(U, Z)$ factorizes as well. As discussed in Section 3.1.3, VBEM minimizes iteratively the KL divergence between the true and the variational posterior by alternating between the following two steps:

VBE-step :

$$q_{\mathbf{u}_n,\mathbf{z}_n}(\mathbf{u}_n, \mathbf{z}_n) \propto \exp\left(\mathrm{E}_{\boldsymbol{\theta}_\mathcal{S}}\{\log p(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n|\boldsymbol{\theta}_\mathcal{S}, \mathcal{H}_M)\}\right) \ , \quad \forall n \ . \tag{3.220}$$

VBM-step :

$$q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \propto p(\boldsymbol{\theta}_\mathcal{S}|\mathcal{H}_M) \exp\left(\mathrm{E}_{U,Z}\{\log \mathcal{L}_c(\boldsymbol{\theta}_\mathcal{S}|X, U, Z, \mathcal{H}_M)\}\right) \ . \tag{3.221}$$

In these equations, the expectations are taken with respect to the variational distributions. The complete data likelihood for the SMM is given by

$$\mathcal{L}_c(\boldsymbol{\theta}_\mathcal{S}|X, U, Z, \mathcal{H}_M) = \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n|\boldsymbol{\theta}_\mathcal{S}, \mathcal{H}_M) \ , \tag{3.222}$$

where we have

$$\begin{aligned} &p(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n|\mathcal{H}_M) \\ &\quad = \prod_{m=1}^{M} {\pi_m}^{z_{nm}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, u_{nm}\boldsymbol{\Lambda}_m)^{z_{nm}} \mathcal{G}(u_{nm}|\tfrac{\nu_m}{2}, \tfrac{\nu_m}{2})^{z_{nm}} \ . \end{aligned} \tag{3.223}$$

Due to the factorized form of $p(\mathbf{x}_n, \mathbf{u}_n, \mathbf{z}_n|\boldsymbol{\theta}_\mathcal{S}, \mathcal{H}_M)$, it is likely that $\{q_{\mathbf{z}_n}(\mathbf{z}_n)\}_{n=1}^N$ and $\{q_{\mathbf{u}_n}(\mathbf{u}_n)\}_{n=1}^N$ factorize similarly:

$$q_{\mathbf{z}_n}(\mathbf{z}_n) = \prod_{m=1}^{M} q_{z_{nm}}(z_{nm})^{z_{nm}} \ , \quad \forall n \ . \tag{3.224}$$

$$q_{\mathbf{u}_n}(\mathbf{u}_n|\mathbf{z}_n) = \prod_{m=1}^{M} q_{u_{nm}}(u_{nm})^{z_{nm}} \ , \quad \forall n \ . \tag{3.225}$$

Following the same approach as in the GMM case, the prior on the mixture proportions, the means and the precisions are chosen conjugate to the exponential family. The joint prior is thus the product of a Dirichlet distribution

and Gaussian-Wishart distributions. Since there is no conjugate prior for the set of df $\{\nu_m\}_{m=1}^M$, no prior is imposed on them. Instead, a ML estimate is used. The resulting joint prior is:

$$p(\boldsymbol{\theta}_{\mathcal{S}}|\mathcal{H}_M) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}_0) \prod_{m=1}^{M} \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_0}) \prod_{m=1}^{M} \delta(\nu_m) \;, \tag{3.226}$$

where $\delta(\cdot)$ denotes the Dirac pulse and $\boldsymbol{\theta}_{\mathcal{NW}_0} = (\eta_0, \mathbf{m}_0, \gamma_0, \mathbf{S}_0)$. These parameters are chosen such that they give broad priors. The joint posterior has the same functional form as the prior:

$$q_{\boldsymbol{\theta}_{\mathcal{S}}}(\boldsymbol{\theta}_{\mathcal{S}}) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \prod_{m=1}^{M} \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) \prod_{m=1}^{M} \delta(\nu_m) \;, \tag{3.227}$$

where $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_m)$ and $\boldsymbol{\theta}_{\mathcal{NW}_m} = (\eta_m, \mathbf{m}_m, \gamma_m, \mathbf{S}_m)$.

Given these specific choices for the priors and the posteriors, the VBE-step can be computed. Taking expectations with respect to the posterior distribution of the parameters leads to:

$$\begin{aligned}
&\mathrm{E}_{\boldsymbol{\theta}_{\mathcal{S}}}\{\log p(\mathbf{x}_n, u_{nm}, z_{nm}|\mathcal{H}_M)\} \\
&\quad = z_{nm}\Big\{ \log\tilde{\pi}_m - \frac{d}{2}\log 2\pi + \frac{d}{2}\log u_{nm} + \frac{1}{2}\log\tilde{\Lambda}_m \\
&\qquad - \frac{u_{nm}\gamma_m}{2}(\mathbf{x}_n - \mathbf{m}_m)^{\mathrm{T}}{\mathbf{S}_m}^{-1}(\mathbf{x}_n - \mathbf{m}_m) - \frac{u_{nm}d}{2\eta_m} \\
&\qquad + \frac{\nu_m}{2}\log\frac{\nu_m}{2} - \log\Gamma\left(\frac{\nu_m}{2}\right) + \left(\frac{\nu_m}{2} - 1\right)\log u_{nm} - \frac{\nu_m}{2}u_{nm}\Big\} \;,
\end{aligned} \tag{3.228}$$

where

$$\log\tilde{\pi}_m \equiv \mathrm{E}_{\boldsymbol{\theta}_{\mathcal{S}}}\{\log\pi_m\} = \psi(\kappa_m) - \psi\left(\textstyle\sum_{m'=1}^{M}\kappa_{m'}\right) \;, \tag{3.229}$$

$$\log\tilde{\Lambda}_m \equiv \mathrm{E}_{\boldsymbol{\theta}_{\mathcal{S}}}\{\log|\boldsymbol{\Lambda}_m|\} = \sum_{i=1}^{d}\psi\left(\tfrac{\gamma_m+1-i}{2}\right) + d\log 2 - \log|\mathbf{S}_m| \;. \tag{3.230}$$

On the one hand, substituting (3.228) in (3.220) and integrating out the scale variable leads to the VBE-step for the indicator variables:

$$\begin{aligned}
q_{z_{nm}}(z_{nm}=1) \propto \tilde{\pi}_m \frac{\Gamma\left(\frac{d+\nu_m}{2}\right)}{\Gamma\left(\frac{\nu_m}{2}\right)(\nu_m\pi)^{\frac{d}{2}}}\tilde{\Lambda}_m{}^{\frac{1}{2}} \\
\times\left[1 + \frac{\gamma_m}{\nu_m}(\mathbf{x}_n - \mathbf{m}_m)^{\mathrm{T}}{\mathbf{S}_m}^{-1}(\mathbf{x}_n - \mathbf{m}_m) + \frac{d}{\nu_m\eta_m}\right]^{-\frac{d+\nu_m}{2}} \;.
\end{aligned} \tag{3.231}$$

Since the distribution $q_{\mathbf{z}_n}(\mathbf{z}_n)$ must be normalized for each data point $\mathbf{x}_n$, we have

$$\bar{\rho}_{nm} = \frac{q_{z_{nm}}(z_{nm}=1)}{\sum_{m'=1}^{M} q_{z_{nm'}}(z_{nm'}=1)} \;, \quad \forall n \;, \quad \forall m \;. \tag{3.232}$$

These quantities are termed responsibilities and are very similar to the E-step of the indicator variables in ML learning.

On the other hand, it can be seen from (3.228) that the variational posterior on the scale variables $q_{u_{nm}}(u_{nm}|z_{nm}=1)$ has the form of the following Gamma distribution:

$$q_{u_{nm}}(u_{nm}|z_{nm}=1) = \mathcal{G}(u_{nm}|\alpha_{nm},\beta_{nm}) \ , \tag{3.233}$$

with

$$\alpha_{nm} = \frac{d+\nu_m}{2} \ , \tag{3.234}$$

$$\beta_{nm} = \frac{\gamma_m}{2}(\mathbf{x}_n-\mathbf{m}_m)^{\mathrm{T}}{\mathbf{S}_m}^{-1}(\mathbf{x}_n-\mathbf{m}_m) + \frac{d}{2\eta_m} + \frac{\nu_m}{2} \ . \tag{3.235}$$

Again, the VBE-step of the scale variables shows a striking similarity to the corresponding E-step in ML learning. Moreover, this step simply consists in updating the hyperparameters $\{\alpha_{n,m}\}_{n,m=1}^{N,M}$ and $\{\beta_{nm}\}_{n,m=1}^{N,M}$.

Next, let us compute the VBM-step. Using (3.188–3.188), the expected complete data log-likelihood is given by

$$\begin{aligned}
&\mathrm{E}_{U,Z}\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{S}}|X,U,Z)\} \\
&\quad= \sum_{n=1}^{N}\sum_{m=1}^{M} \bar{\rho}_{nm}\Big\{ \log\pi_m - \frac{d}{2}\log 2\pi + \frac{d}{2}\log\tilde{u}_{nm} + \frac{1}{2}\log|\boldsymbol{\Lambda}_m| \\
&\qquad - \frac{\bar{u}_{nm}}{2}(\mathbf{x}_n-\boldsymbol{\mu}_m)^{\mathrm{T}}\boldsymbol{\Lambda}_m(\mathbf{x}_n-\boldsymbol{\mu}_m) + \frac{\nu_m}{2}\log\frac{\nu_m}{2} \\
&\qquad - \log\Gamma\left(\frac{\nu_m}{2}\right) + \left(\frac{\nu_m}{2}-1\right)\log\tilde{u}_{nm} - \frac{\nu_m}{2}\bar{u}_{nm}\Big\} \ ,
\end{aligned} \tag{3.236}$$

where $\boldsymbol{\pi} = \{\pi_m\}_{m=1}^{M}$, $\boldsymbol{\nu} = \{\nu_m\}_{m=1}^{M}$ and $\boldsymbol{\theta}_{\mathcal{S}_m} = (\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)$, $\forall m$. In this equation, we use the fact that $\mathrm{E}_Z\{z_{nm}\} = \bar{\rho}_{nm}$ and equate the special quantities $\bar{u}_{nm}$ and $\log\tilde{u}_{nm}$ respectively to $\mathrm{E}_U\{u_{nm}\}$ and $\mathrm{E}_U\{\log u_{nm}\}$. These quantities can be computed using the properties of the Gamma distribution:

$$\bar{u}_{nm} = \frac{\alpha_{nm}}{\beta_{nm}} \ , \tag{3.237}$$

$$\log\tilde{u}_{nm} = \psi(\alpha_{nm}) - \log\beta_{nm} \ . \tag{3.238}$$

Using these results, the VBM update rules for the hyperparameters can be identified from (3.221) after some algebra:

$$\kappa_m = N\bar{\pi}_m + \kappa_0 \ , \tag{3.239}$$

$$\eta_m = N\bar{\omega}_m + \eta_0 \ , \tag{3.240}$$

$$\mathbf{m}_m = \frac{N\bar{\omega}_m\bar{\boldsymbol{\mu}}_m + \eta_0\mathbf{m}_0}{N\bar{\omega}_m + \eta_0} \ , \tag{3.241}$$

$$\gamma_m = N\bar{\pi}_m + \gamma_0 \ , \tag{3.242}$$

$$\mathbf{S}_m = N\bar{\omega}_m\bar{\boldsymbol{\Sigma}}_m + \frac{N\bar{\omega}_m\eta_0}{\eta_m}(\bar{\boldsymbol{\mu}}_m-\mathbf{m}_0)(\bar{\boldsymbol{\mu}}_m-\mathbf{m}_0)^{\mathrm{T}} + \mathbf{S}_0 \ , \tag{3.243}$$

where

$$\bar{\boldsymbol{\mu}}_m = \frac{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm}\mathbf{x}_n}{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm}} \ , \tag{3.244}$$

$$\bar{\boldsymbol{\Sigma}}_m = \frac{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm}\left(\mathbf{x}_n - \bar{\boldsymbol{\mu}}_m\right)\left(\mathbf{x}_n - \bar{\boldsymbol{\mu}}_m\right)^{\mathrm{T}}}{\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm}} \ , \tag{3.245}$$

$$\bar{\pi}_m = \frac{1}{N}\sum_{n=1}^N \bar{\rho}_{nm} \ , \tag{3.246}$$

$$\bar{\omega}_m = \frac{1}{N}\sum_{n=1}^N \bar{\rho}_{nm}\bar{u}_{nm} \ . \tag{3.247}$$

These quantities are weighted averages. All, except the last one are identical to the ML parameter estimates computed in the M-step of the EM algorithm. Note that the normalizing factor of the covariance matrices corresponds to the one proposed by Kent et al. (1994) in ML learning, in order to accelerate the convergence of the algorithm.

Finally, maximizing (3.236) according to the df leads to the same update rule as in ML learning:

$$\log\frac{\nu_m}{2} + 1 - \psi\left(\frac{\nu_m}{2}\right) + \frac{1}{N\bar{\pi}_m}\sum_{n=1}^N \bar{\rho}_{nm}\left\{\log\tilde{u}_{nm} - \bar{u}_{nm}\right\} = 0 \ . \tag{3.248}$$

Predictive distribution

For the SMM, the predictive distribution based on the variational posterior of the model parameters is still intractable. Therefore, the predictive distribution is approximated as follows:

$$p(\mathbf{x}|X,\mathcal{H}_M) \approx p(\mathbf{x}|\hat{\boldsymbol{\theta}}_{\mathcal{S}}) \ , \tag{3.249}$$

where

$$\hat{\boldsymbol{\theta}}_{\mathcal{S}} = \int \boldsymbol{\theta}_{\mathcal{S}} q_{\boldsymbol{\theta}_{\mathcal{S}}}(\boldsymbol{\theta}_{\mathcal{S}}) d\boldsymbol{\theta}_{\mathcal{S}} \ . \tag{3.250}$$

The resulting predictive distribution is given by

$$p(\mathbf{x}|\hat{\boldsymbol{\theta}}_{\mathcal{S}}) = \sum_{m=1}^M \breve{\pi}_m \mathcal{S}(\mathbf{x}|\mathbf{m}_m, \gamma_m \mathbf{S}_m{}^{-1}, \nu_m) \ , \tag{3.251}$$

where $\breve{\pi}_m = \kappa_m / \sum_{m'=1}^M \kappa_{m'}$.

Choosing the number of components

Let us conclude this discussion of the SMM by indicating how the optimal model complexity can be chosen on the basis of the lower bound on the log-evidence $p(X|\mathcal{H}_M)$.

Consider the variational lower bound for the SMM:

$$\begin{aligned}
&\mathcal{F}_{\mathcal{H}_M}(q_{U,Z}(U,Z), q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S})) \\
&\quad = \sum_Z \iint q_U(U|Z) q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log \frac{p(X,U,Z,\boldsymbol{\theta}_\mathcal{S}|\mathcal{H}_M)}{q_U(U|Z) q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S})} dU\, d\boldsymbol{\theta}_\mathcal{S} \\
&\quad = \sum_Z \iint q_U(U|Z) q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log p(X|U,Z,\boldsymbol{\theta}_\mathcal{S},\mathcal{H}_M) dU\, d\boldsymbol{\theta}_\mathcal{S} \\
&\qquad + \sum_Z \iint q_U(U|Z) q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log p(U|Z,\boldsymbol{\theta}_\mathcal{S},\mathcal{H}_M) dU\, d\boldsymbol{\theta}_\mathcal{S} \\
&\qquad + \sum_Z \int q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log p(Z|\boldsymbol{\theta}_\mathcal{S},\mathcal{H}_M) d\boldsymbol{\theta}_\mathcal{S} \\
&\qquad + \int q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log p(\boldsymbol{\theta}_\mathcal{S}|\mathcal{H}_M) d\boldsymbol{\theta}_\mathcal{S} \\
&\qquad - \sum_Z \int q_U(U|Z) q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log q_U(U|Z) dU \\
&\qquad - \sum_Z q_Z(Z) \log q_Z(Z) \\
&\qquad - \int q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) d\boldsymbol{\theta}_\mathcal{S} \ . \qquad (3.252)
\end{aligned}$$

All the distributions appearing in this expression are known:

$$p(X|U,Z,\boldsymbol{\theta}_\mathcal{S},\mathcal{H}_M) = \prod_{n=1}^N \prod_{m=1}^M \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, u_{nm}\boldsymbol{\Lambda}_m)^{z_{nm}} \ , \qquad (3.253)$$

$$p(U|Z,\boldsymbol{\theta}_\mathcal{S},\mathcal{H}_M) = \prod_{n=1}^N \prod_{m=1}^M \mathcal{G}(u_{nm}|\tfrac{\nu_m}{2}, \tfrac{\nu_m}{2})^{z_{nm}} \ , \qquad (3.254)$$

$$p(Z|\boldsymbol{\theta}_\mathcal{S},\mathcal{H}_M) = \prod_{n=1}^N \prod_{m=1}^M {\pi_m}^{z_{nm}} \ , \qquad (3.255)$$

$$p(\boldsymbol{\theta}_\mathcal{S}|\mathcal{H}_M) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}_0) \prod_{m=1}^M \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_0}) \prod_{m=1}^M \delta(\nu_m) \ , \qquad (3.256)$$

$$q_U(U|Z) = \prod_{n=1}^N \prod_{m=1}^M \mathcal{G}(u_{nm}|\alpha_{nm}, \beta_{nm})^{z_{nm}} \ , \qquad (3.257)$$

$$q_Z(Z) = \prod_{n=1}^N \prod_{m=1}^M {\bar{\rho}_{nm}}^{z_{nm}} \ , \qquad (3.258)$$

$$q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) = \mathcal{D}(\boldsymbol{\pi}|\boldsymbol{\kappa}) \prod_{m=1}^M \mathcal{NW}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m|\boldsymbol{\theta}_{\mathcal{NW}_m}) \prod_{m=1}^M \delta(\nu_m) \ . \qquad (3.259)$$

Therefore, each term of the variational bound can be computed as follows:

$$\sum_Z \iint q_U(U|Z) q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log p(X|U, Z, \boldsymbol{\theta}_\mathcal{S}, \mathcal{H}_M) dU d\boldsymbol{\theta}_\mathcal{S}$$
$$= \sum_{n=1}^N \sum_{m=1}^M \bar{\rho}_{nm} \Big\{ -\tfrac{d}{2}\log 2\pi + \tfrac{d}{2}\log \tilde{u}_{nm} + \tfrac{1}{2}\log \tilde{\Lambda}_m$$
$$- \tfrac{\bar{u}_{nm}\gamma_m}{2}(\mathbf{x}_n - \mathbf{m}_m)^\mathrm{T} {\mathbf{S}_m}^{-1}(\mathbf{x}_n - \mathbf{m}_m) - \tfrac{\bar{u}_{nm} d}{2\eta_m} \Big\} , \tag{3.260}$$

$$\sum_Z \iint q_U(U|Z) q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log p(U|Z, \boldsymbol{\theta}_\mathcal{S}, \mathcal{H}_M) dU d\boldsymbol{\theta}_\mathcal{S}$$
$$= \sum_{n=1}^N \sum_{m=1}^M \bar{\rho}_{nm} \Big\{ \tfrac{\nu_m}{2}\log\tfrac{\nu_m}{2} - \log\Gamma\left(\tfrac{\nu_m}{2}\right)$$
$$+ \left(\tfrac{\nu_m}{2} - 1\right)\log \tilde{u}_{nm} - \tfrac{\nu_m}{2}\bar{u}_{nm} \Big\} , \tag{3.261}$$

$$\sum_Z \int q_Z(Z) q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log p(Z|\boldsymbol{\theta}_\mathcal{S}, \mathcal{H}_M) \boldsymbol{\theta}_\mathcal{S}$$
$$= \sum_{n=1}^N \sum_{m=1}^M \bar{\rho}_{nm} \log \tilde{\pi}_m , \tag{3.262}$$

$$\int q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log p(\boldsymbol{\theta}_\mathcal{S}|\mathcal{H}_M) d\boldsymbol{\theta}_\mathcal{S}$$
$$= \log c_\mathcal{D}(\boldsymbol{\kappa}_0) + \sum_{m=1}^M (\kappa_0 - 1)\log\tilde{\pi}_m + \sum_{m=1}^M \Big\{ -\tfrac{d}{2}\log 2\pi$$
$$+ \tfrac{d}{2}\log\eta_0 - \tfrac{\gamma_m \eta_0}{2}(\mathbf{m}_m - \mathbf{m}_0)^\mathrm{T} {\mathbf{S}_m}^{-1}(\mathbf{m}_m - \mathbf{m}_0) - \tfrac{\eta_0 d}{2\eta_m}$$
$$+ \log c_{\mathcal{NW}}(\gamma_0, \mathbf{S}_0) + \tfrac{\gamma_0 - d}{2}\log\tilde{\Lambda}_m - \tfrac{\gamma_m}{2}\mathrm{tr}\{\mathbf{S}_0 {\mathbf{S}_m}^{-1}\} \Big\} , \tag{3.263}$$

$$\sum_Z \int q_U(U|Z) q_Z(Z) \log q_U(U|Z) dU$$
$$= \sum_{n=1}^N \sum_{m=1}^M \bar{\rho}_{nm} \big\{ -\log\Gamma(\alpha_{nm}) + (\alpha_{nm} - 1)\psi(\alpha_{nm})$$
$$+ \log\beta_{nm} - \alpha_{nm} \big\} , \tag{3.264}$$

$$\sum_Z q_Z(Z) \log q_Z(Z)$$
$$= \sum_{n=1}^N \sum_{m=1}^M \bar{\rho}_{nm} \log \bar{\rho}_{nm} , \tag{3.265}$$

$$\int q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) \log q_{\boldsymbol{\theta}_\mathcal{S}}(\boldsymbol{\theta}_\mathcal{S}) d\boldsymbol{\theta}_\mathcal{S}$$
$$= \log c_\mathcal{D}(\boldsymbol{\kappa}) + \sum_{m=1}^M (\kappa_m - 1)\log\tilde{\pi}_m + \sum_{m=1}^M \Big\{ -\tfrac{d}{2}\log 2\pi$$
$$+ \tfrac{d}{2}\log\eta_m - \tfrac{d}{2} + \log c_{\mathcal{NW}}(\gamma_m, \mathbf{S}_m) + \tfrac{\gamma_m - d}{2}\log\tilde{\Lambda}_m - \tfrac{\gamma_m d}{2} \Big\} . \tag{3.266}$$

The approach is illustrated on the same example as the one used for the GMM, now corrupted by 25% of atypical observations (uniform random noise). The data are shown in Figure 3.19. Hundred fifty data points are drawn from each component. The VBEM algorithm is run 10 times. The model complexity

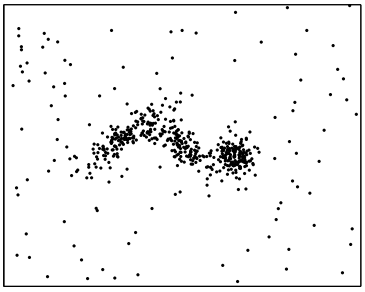
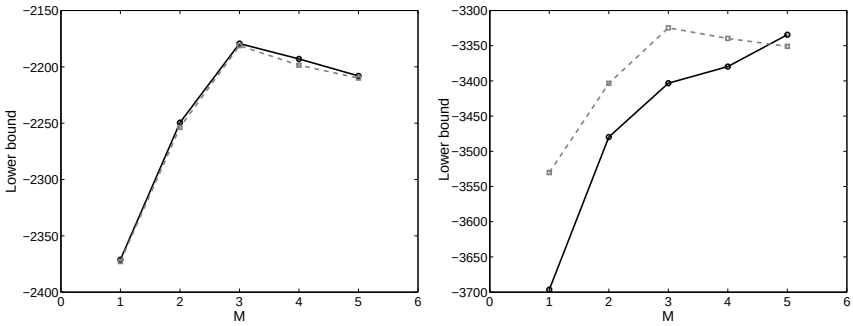


FIGURE 3.19. Training set. The data consists in a mixture of three Gaussian distribution with different mean and precision. The data is corrupted by 25% of atypical observations.



(a) Without atypical observations. (b) With atypical observations.

FIGURE 3.20. Lower bound on the log-evidence versus the number of components M. The solid and the dashed lines correspond respectively to the GMM and the SMM. The curves show the average on 10 trials. The model complexity is selected according to the maximum of the lower bound.

ranges from 1 to 5 components. Figure 3.20 shows the average lower bound on the log-evidence, both in absence and presence of atypical observations. When there are no atypical observations, the GMM and the SMM perform similarly. Both methods select the correct number of components. In contrast, when there are atypical observations only the SMM selects the right number of components.

Effect of the factorization of the latent variables' posterior

As already mentioned, Svensén and Bishop (2004) also applied variational Bayesian inference to Student-t mixture models. In particular, they assume that the variational posterior on the latent indicator variables and the latent

scale variables factorize. However, we have shown in Section 3.3.5 that this factorization is not necessary. Taking into account the correlations between the indicator and the scale variables leads to a model with an increased robustness to atypical observations. Furthermore, it is expected that the lower bound on the log-evidence is tighter. This is important, since the lower bound is used as a model selection criterion. However, by doing so, it is assumed that the gap between the log-evidence and the variational lower bound is identical, after convergence, for models of different complexity. Of course, this is not true in practice. Usually, variational Bayesian inference tends to overpenalize complex models, as the factorized approximations lead to a posterior that is more compact than the true posterior. Therefore, when we want to perform model selection based on the lower bound, it is essential to avoid any unnecessary approximations. Next, we give some experimental evidence to illustrate these remarks.

Let us consider the Old Faithful Geyser data (see Appendix A). The data is normalized and then corrupted by a certain amount of outliers. These are simulated by uniform random noise on the interval $[-10, 10]$ in each direction of the feature space. Figure 3.21 shows the variational lower bound for the variational GMM, the variational type-I SMM, which assumes that the variational posterior on the indicator variables and the scale variables factorize, and the variational type-II SMM, which does not make this assumption. The number of components vary from 1 to 6. For each model complexity 20 runs are considered. Note that in some cases components are automatically pruned out when they do not have sufficient support. In absence of outliers, the bound of the three methods is maximal for two components. In presence of 2% outliers the type-I SMM has solutions for both two and three components with almost identical values of the bound. This was also observed by Svensén and Bishop (2004). For the type-II SMM, the bound is still maximal for two components. The GMM however favors 3 components. When the amount of noise further increases (25%), only the type-II SMM selects 2 components. As a matter of fact, the value of the bound seems almost not affected by an increase of the noise. Thus, not neglecting the correlation between the indicator variables and the scale variables clearly increases the robustness. This can easily be verified when looking to Figure 3.22, where it is shown how the outliers affect the quality of both SMM.

In Figure 3.23, the typical variational posterior of a single data point is shown. It can be observed that the type-I SMM assigns the probability mass almost exclusively to one component (here to component 2) and that the posterior for that component is more peaked than the posterior of the type-II SMM. This suggests that the empirical variance is (even more) underestimated when assuming that the scale variables are independent from the indicator variables. Since the uncertainty is underestimated, the robustness of the model is reduced. In general, the variational posterior is more compact than the true posterior. This can be understood by seeing that maximizing the lower bound is done by minimizing the KL divergence between the variational posterior and the true

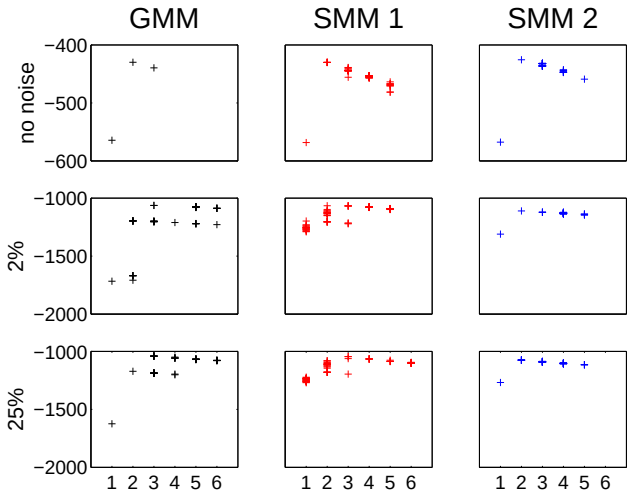
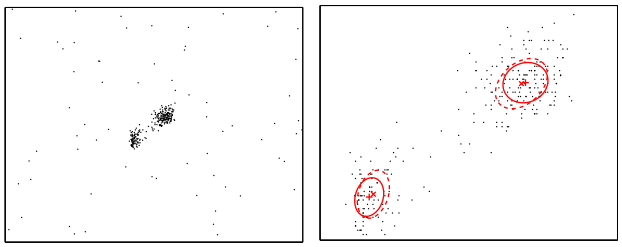


FIGURE 3.21. Old Faithful Geyser data. The lower bound on the log-evidence for the variational GMM and the variational type-I and type-II SMM for each run. Twenty runs are considered. The number of components vary from 1 to 6. An increasing number of outliers is successively added to the training set.

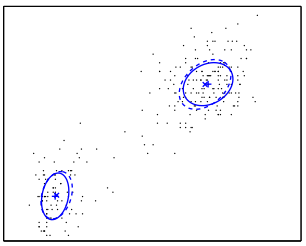
posterior. However, the KL divergence is taken with respect to the support of the variational distribution and not with respect to the support of the true posterior.

In order to further assess the robustness of the type-II SMM, consider the 3-component bivariate mixture of Gaussian distributions from Ueda and Nakano (1998). The mixture proportions are all equal to $1/3$, the mean vectors are $(0, -2)^{\mathrm{T}}$, $(0, 0)^{\mathrm{T}}$ and $(0, 2)^{\mathrm{T}}$, and the covariance matrix of each component is equal to $\mathrm{diag}\{2, 0.2\}$. The labeling (colors) of the data points are presented in Figure 3.24. Two situations are investigated. In presence of a small proportion of outliers (2%), both variational SMM perform similarly. However, note that the type-II SMM assigns a blue label to all outliers, while the type-I SMM partitions the feature space in three parts. In presence of lots of outliers (25%) only the type-II SMM provides a satisfactory solution. Still all outliers are assigned a blue label, i.e. the blue component has very heavy tails.

In conclusion, we have shown that the alternative variational update rules that we derived for a Bayesian mixture of Student-t distributions lead to a model that has a higher robustness against outliers. Our derivation is based on a different formulation of the latent variable model. As a result, it is possible to avoid the use of a factorized variational posterior on the indicator and the scale variables. Taking the correlation between these latent variables into account leads to a variational posterior that is less compact than the one obtained



(a) Noisy training data. (b) Zoom on type-I SMM



(c) Zoom on type-II SMM

FIGURE 3.22. Old Faithful Geyser data. The dashed curves correspond to the model in absence of outliers. The solid curves are obtained when 25% of outliers are added to the training set. Clearly, the type-II SMM (b) is less affected by the outliers than the type-I SMM (a). The models are constructed with 2 components.

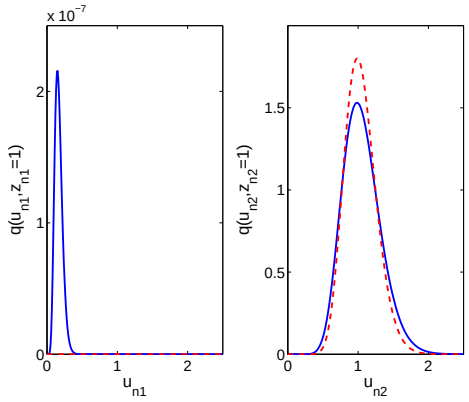
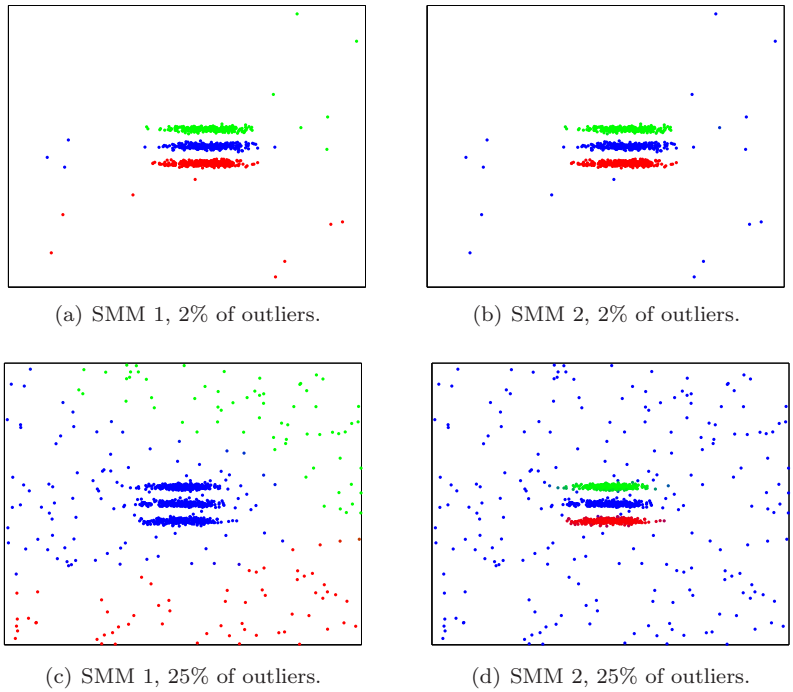


FIGURE 3.23. The typical joint variational posterior of the indicator and the scale variable for a single data point. The mixture has two components. The data is the Old Faithful Geyser data. The solid curve does not neglect the correlation between both latent variables (type-II SMM), while the dashed curve does (type-I SMM).



(a) SMM 1, 2% of outliers. (b) SMM 2, 2% of outliers.

(c) SMM 1, 25% of outliers. (d) SMM 2, 25% of outliers.

FIGURE 3.24. Label (color) assigned to the data points by the variational type-I and type-II SMM, using 3 components. (a) and (b) are the models obtained when 2% of outliers is added to the training set, while (c) and (d) are the ones obtained in presence of 25% of outliers.

in previous approaches; therefore it underestimates less the uncertainty in the latent variables. Although the resulting lower bound does not seem tighter, the correct model complexity is selected in a more consistent way and the model is less sensitive to local maxima.

3.4. Manifold Constrained Mixture Models

In many machine learning applications, the data is living in a high dimensional space. Due to the curse of dimensionality, this can lead to serious problems in practice. Fortunately, the data is in many cases also concentrated on an implicit manifold, of lower dimension than the dimension of the feature space. Subsequently, we show how to take advantage of the intrinsic geometrical arrangement of the data during the training of the models.

Recently, manifold kernel density estimation (Vincent and Bengio, 2002) was introduced in order to improve standard nonparametric kernel density estimation (KDE) in this context. Since the true density mass in the vicinity of a

data point is oriented along the manifold rather than along all the directions in the input space, estimating the unknown density by conventional techniques is suboptimal. It tends to give too much probability mass to irrelevant directions of space (i.e. perpendicular to the local manifold orientation) and too little along the manifold. Instead of placing a spherical kernel on each data point as in KDE, Vincent and Bengio (2002) compute for each of them a local covariance matrix based on the closest neighbors. Furthermore, in order to obtain a more compact (i.e. a lower dimensional) representation of the Gaussian kernels, only the eigenvectors associated to the largest eigenvalues are kept. As a result, the density mass is oriented along the principal directions of the data in the vicinity of each data point.

In this section, a related technique for finite mixture models is introduced. Both ML/MAP and Bayesian mixtures are considered (Archambeau and Verleysen, 2005a,b). When the data manifold is of lower dimension than the dimension of the feature space, it is proposed to take this additional information into account during training. In this perspective, the responsibilities computed in the E-step are penalized according to some prior belief on the discrepancy between the Euclidian and the geodesic distance. The latter is measured along the manifold and not through the embedding space. Here also, the key idea is to favor the directions along the manifold when estimating the unknown density, rather than wasting valuable density mass in directions perpendicular to the manifold orientation. How to achieve this in the case of ML and variational Gaussian mixtures is explained below. It is straightforward to extend these results to other mixture models.

3.4.1. Constructing the Data Manifold

The basic principle of nonlinear data projection techniques, such as ISOMAP (Tenenbaum, de Silva and Langford, 2000), Locally Linear Embedding (LLE) (Roweis and Saul, 2000) or Curvilinear Distance Analysis (CDA) (Lee, Lendasse and Verleysen, 2003), is to find the lower dimensional data manifold (if any) embedded in the input space and unfold it. An essential building block for constructing this manifold is the geodesic distance. This metric is measured along the manifold and not through the feature or embedding space, akin to the Euclidean distance. As a result, the geodesic distance less depends on the curvature of the manifold and takes thus the intrinsic geometrical structure of the data into account. This is illustrated in Figure 3.25.

Geodesic distances

Consider two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ of the p-dimensional manifold $\mathcal{M}$ of lower dimensionality than the embedding space. The manifold $\mathcal{M}$ is parameterized as follows:

$$\mathbf{m} : \mathbb{R}^p \rightarrow \mathcal{M} \subset \mathbb{R}^d : \mathbf{y} \mapsto \mathbf{x} = \mathbf{m}(\mathbf{y}) \ ,$$

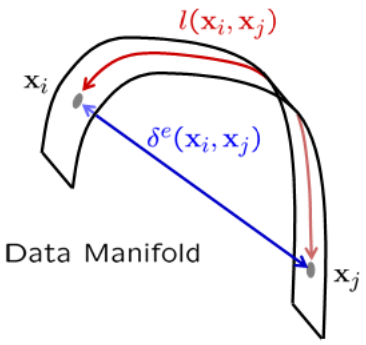


FIGURE 3.25. The data is located on a 1D manifold. In contrast, the dimension of the feature space is 2. The Euclidean and the geodesic distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ are respectively denoted by $\delta^e(\mathbf{x}_i, \mathbf{x}_j)$ and $l(\mathbf{x}_i, \mathbf{x}_j)$. The geodesic distance is measured along the manifold and is therefore larger than the Euclidean distance.

where d is the dimension of the embedding space. Different paths may go from point $\mathbf{x}_i$ to point $\mathbf{x}_j$. Each of them is described by a 1D submanifold $\mathcal{P}_{i,j}^{(k)}$ of the multidimensional manifold $\mathcal{M}$ with parametric equations:

$$\mathbf{p}_k : \mathbb{R} \rightarrow \mathcal{P}_{i,j}^{(k)} \subset \mathbb{R}^p : t \mapsto \mathbf{y} = \mathbf{p}_k(t) \ .$$

The geodesic distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is then defined as the minimal arc length connecting both data points:

$$l(\mathbf{x}_i, \mathbf{x}_j) = \min_{\mathbf{p}_k(t)} \int_{t_i}^{t_j} \|\mathbf{J}_t(\mathbf{m}(\mathbf{p}_k(t)))\| dt \ ,$$

where $\|\cdot\|$ is the L_2-norm and $\mathbf{J}_t(\cdot)$ is the Jacobian with respect to t. In practice, this minimization is difficult and often intractable since it is a functional minimization and the parametric equations of the submanifolds are generally unknown; only noisy observations of points on $\mathcal{M}$ are available.

Graph distances

Even though geodesic distances cannot be computed in practice, they can easily be approximated by minimum graph distances (Bernstein, de Silva, Langford and Tenenbaum, 2000). The problem of minimizing the arc length between two data points on the manifold $\mathcal{M}$ reduces to the problem of minimizing the length of path (i.e., broken line) between these data points, while passing through a certain number of other data points of $\mathcal{M}$. In order to follow the manifold, only the smallest jumps between successive points will be permitted. This can be achieved by using, either the K-rule, or the ϵ-rule. The former allows jumping to the K nearest neighbors, K being a constant. The latter allows jumping to points lying inside a ball of pre-determined radius ϵ. In the following, we only

consider the K-rule as the choice for ϵ is more difficult in practice than the one for K (Lee, 2004).

The data and the set of allowed jumps constitute a weighted graph $G(V_N, E)$, the vertices V_N being the N data points, the edges E the allowed jumps and the edge labels (or weights) the Euclidean distance between the corresponding vertices. This graph is called the neighborhood graph. The Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$\delta^e(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| \ . \tag{3.267}$$

A path in the graph G is an ordered subset of vertices of V_N such that the edges linking these vertices belong to E. The path k between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as follows:

$$\begin{gathered}\hat{\mathcal{P}}_{i,j}^{(k)} = \{\mathbf{x}_i, \mathbf{x}_{i'}, \mathbf{x}_{i''}, \ldots, \mathbf{x}_{j'}, \mathbf{x}_j\} \subseteq V_N \\ \text{s.t.} \\ (\mathbf{x}_i, \mathbf{x}_{i'}), (\mathbf{x}_{i'}, \mathbf{x}_{i''}), \ldots, (\mathbf{x}_{j'}, \mathbf{x}_j) \in E \ .\end{gathered}$$

The path length is then found by adding the edge weights, corresponding to the length of the successive jumps in the path:

$$\text{length}\hat{\mathcal{P}}_{i,j}^{(k)} = \|\mathbf{x}_i - \mathbf{x}_{i'}\| + \|\mathbf{x}_{i'} - \mathbf{x}_{i''}\| + \ldots + \|\mathbf{x}_{j'} - \mathbf{x}_j\| \ . \tag{3.268}$$

In order to be a distance, the path length must satisfy the properties of non-negativity, symmetry and triangular inequality. The first and the third property are satisfied by construction. Symmetry is ensured when the graph is undirected. In the case of the K-rule, this is gained by adding some missing edges: if $\mathbf{x}_j$ belongs to the K nearest neighbors of $\mathbf{x}_i$, but $\mathbf{x}_i$ is not a neighbor of $\mathbf{x}_j$ then the corresponding edge is added. The graph distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, which approximates the corresponding geodesic distance, is then defined as the minimum path length between these points:

$$l(\mathbf{x}_i, \mathbf{x}_j) \approx \delta^g(\mathbf{x}_i, \mathbf{x}_j) = \min_{\hat{\mathcal{P}}_{i,j}^{(k)}} \text{length}\hat{\mathcal{P}}_{i,j}^{(k)} \ . \tag{3.269}$$

When necessary, extra edges are added to the graph in order to avoid disconnected parts. For this purpose, a minimum spanning tree is used. A minimum spanning tree (MST) of a graph $G'(V_N, E')$ is an undirected, acyclic and connected subgraph of G' containing all the vertices V_N and having the minimal total weight. As a result, there is only one path in the subgraph connecting each pair of vertices and the sum of all the weights of the edges is minimal. Minimum spanning trees are commonly constructed by using either Prim's, or Kruskal's algorithm (see for example West, 1996). Prim's algorithm builds the MST by adding one vertex at a time. Starting at any vertex of G', the algorithm picks the vertex connected to the start vertex with minimal edge weight. Next, it finds the least costly vertex connection to one of these two vertices without creating a cycle. The procedure continues until all the vertices of G' are connected without any cycles. In contrast, Kruskal's algorithm is a greedy algorithm that keeps adding any edge of G' with the least weight, while avoiding the creation of cycles. Suppose this graph has N vertices. The iterative procedure stops when $(N-1)$ edges have been added.

At this point, the data manifold can be described through the distance matrix of the weighted undirected graph. This matrix is symmetric, of size $N \times N$ for a graph with N vertices and contains the distances, i.e. the length of the shortest paths, between all pairs of vertices in this graph. The shortest paths between all data points are generally computed by repeatedly applying Dijkstra's algorithm (Dijkstra, 1959). Dijkstra's procedure computes the shortest path between a source vertex and all other vertices in a weighted graph, provided the edge labels are non-negative (which is the case here). The algorithm begins at a specific vertex and extends outward within the graph, until all vertices are reached. The total minimum cost, i.e. the minimal sum of the edge weights, from the source vertex to the current vertices is stored during the procedure. This means that Dijkstra's algorithm ends up with the minimum cost or shortest path to all vertices.

3.4.2. Manifold Constrained E-step

In this section, it is shown how to constrain the E-step in ML or MAP learning according to the implicit information of the data manifold (Archambeau and Verleysen, 2005a). The idea is to downweight the contribution of the data points which are lying far away from the component centers on the manifold. The approach is applied to the GMM as a particular case, but its extension to the SMM is straightforward.

Let us respectively denote the Euclidian and the graph distance between point $\mathbf{x}_n$ and component mean $\boldsymbol{\mu}_m$ by $\delta^e(\mathbf{x}_n, \boldsymbol{\mu}_m)$ and $\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)$. The graph distance $\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)$ approximates the corresponding geodesic distance $l(\mathbf{x}_n, \boldsymbol{\mu}_m)$.

Consider the exponential distribution with location parameter ν and scale parameter λ:

$$\mathcal{E}(y|\nu, \lambda) = \lambda \exp(-\lambda(y - \nu)) \ , \tag{3.270}$$

where $y \geq \nu$ and $\lambda > 0$. Figure 3.26 shows the shape of the exponential distribution for different values of the location parameter. Setting ν to $\delta^e(\mathbf{x}_n, \boldsymbol{\mu}_m)^2$ and y to $\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)^2$ provides an appropriate measure of the mismatch between both distances since $\delta^e(\mathbf{x}_n, \boldsymbol{\mu}_m) \leq \delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)$. If the component means are held fixed during the E-step, we can bias the posterior distribution of the latent variables as follows:

$$\begin{aligned} &p'(z_{nm} = 1|\mathbf{x}_n, \boldsymbol{\theta}_{\mathcal{N}}) \\ &\quad \propto \pi_m \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) \mathcal{E}(\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m)^2 | \delta^e(\mathbf{x}_n, \boldsymbol{\mu}_m)^2, \lambda_m) \ , \end{aligned} \tag{3.271}$$

where $\boldsymbol{\theta}_{\mathcal{N}} = (\pi_1, \ldots, \pi_M, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M, \boldsymbol{\Lambda}_m, \ldots, \boldsymbol{\Lambda}_M)$. As before, the posterior distribution of the latent variables needs to be normalized as for each $\mathbf{x}_n$ the posterior probabilities must sum to one. The responsibilities are thus given by

$$\bar{\rho}'_{nm} = \frac{p'(z_{nm} = 1|\mathbf{x}_n, \boldsymbol{\theta}_{\mathcal{N}})}{\sum_{m'=1}^{M} p'(z_{nm'} = 1|\mathbf{x}_n, \boldsymbol{\theta}_{\mathcal{N}})} \ . \tag{3.272}$$

Choosing λ_m equal to 1 leaves the posterior probability unchanged when both distances are identical. However, when the discrepancy between the distances

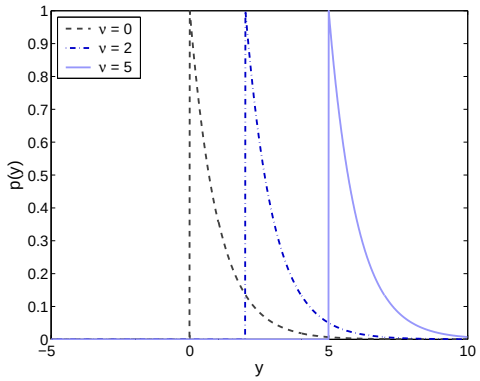


FIGURE 3.26. Exponential distribution with different location parameters and a scale parameter equal to 1.

increases, the posterior $p'(z_{nm} = 1|\mathbf{x}_n, \boldsymbol{\theta}_{\mathcal{N}})$ decreases. This means that it is less likely that data point $\mathbf{x}_n$ was generated by component m because its graph (and thus geodesic) distance to $\boldsymbol{\mu}_m$ is large, compared to its Euclidean distance to $\boldsymbol{\mu}_m$. This results in weaker responsibilities. As a consequence, data points lying far from the component means in terms of geodesic distance (i.e. along the manifold) will contribute less to both the update of the means and the precisions of the corresponding component during the M-step.

As discussed in Section 3.1.1, the EM algorithm maximizes iteratively the expected complete data log-likelihood. Recall that the expectation of the indicator variables is equal to their responsibilities; the expected complete data log-likelihood is given by

$$\mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{N}}|X, Z)\} = \sum_{n=1}^{N}\sum_{m=1}^{M} \bar{\rho}'_{nm} \left\{\log \pi_m + \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m)\right\} \ . \tag{3.273}$$

Maximizing this quantity subject to the constraint on the mixture proportions leads to the following M-step:

$$\boldsymbol{\mu}_m = \frac{\sum_{n=1}^{N} \bar{\rho}'_{nm}\mathbf{x}_n}{\sum_{n=1}^{N} \bar{\rho}'_{nm}} \ , \tag{3.274}$$

$$\boldsymbol{\Lambda}_m = \left\{ \frac{\sum_{n=1}^{N} \bar{\rho}'_{nm} \left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_m\right)^{\mathrm{T}}}{\sum_{n=1}^{N} \bar{\rho}'_{nm}} \right\}^{-1} \ , \tag{3.275}$$

$$\pi_m = \frac{1}{N}\sum_{n=1}^{N} \bar{\rho}'_{nm} \ . \tag{3.276}$$

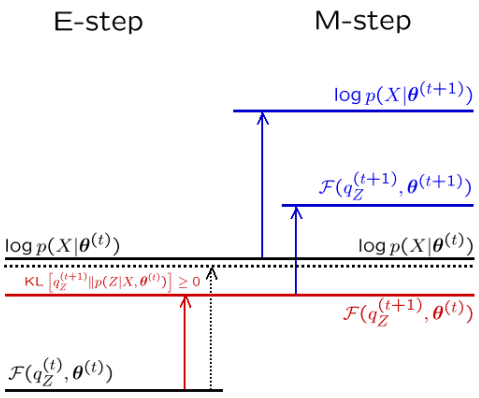


FIGURE 3.27. Constrained EM algorithm. In the unconstrained E-step the bound is made tight (dotted line). In the constrained E-step it is not due to the fact that the posterior distribution of the latent variable is biased according to the prior belief that data points lying far from a component mean on the manifold should contribute less to its update.

These update rules are identical to (3.69–3.71), except that when a data point is far away from a particular component mean on the manifold, the contribution of that point to the parameters of the corresponding component is downweighted.

The interpretation of the constrained EM algorithm is shown in Figure 3.27. Recall that the EM algorithm finds a local maximum of the incomplete data log-likelihood iteratively by alternating between the E-step and the M-step. In the E-step, the lower bound on the incomplete data log-likelihood is made tight by equating the arbitrary distribution of the latent variables $q_Z(Z)$ to their true posterior distribution, while the parameters are held fixed. In the M-step, the bound is maximized with respect to the parameters, keeping the posterior distribution of the latent variable fixed. In the constrained EM algorithm, however, the bound is not made tight, but it is biased according to some prior belief on the discrepancy between the Euclidian and the geodesic distance. Subsequently, in the M-step the lower bound (which is not tight) is maximized. By construction, the bound is still guaranteed to monotonically increase at each iteration.

Learning algorithm

The learning procedure for manifold constrained finite Gaussian mixtures (MGMM) can be summarized as follows:

(1) Construct the training manifold (i.e. the neighborhood graph of X) by the K-rule and compute the associated distance matrix $\delta^g(\mathbf{x}_i, \mathbf{x}_j)$ by Dijkstra's shortest path algorithm.

(2) Repeat until convergence:

Update the distance matrix of the component means:
Find for each $\boldsymbol{\mu}_m$ the K nearest training points $\{\mathbf{x}_k\}_{k=1}^{K}$ and compute its graph distance to all training data by

$$\delta^g(\mathbf{x}_n, \boldsymbol{\mu}_m) = \min_k \{\delta^g(\mathbf{x}_n, \mathbf{x}_k) + \delta^e(\mathbf{x}_k, \boldsymbol{\mu}_m)\} \ . \tag{3.277}$$

E-step: Compute the manifold constrained responsibilities by (3.272) using the current parameter estimates.

M-step: Update the model parameters by (3.274–3.276) using the manifold constrained responsibilities of the E-step.

End.

Remark that the increase of the computational cost at each iteration step is limited with respect to the conventional GMM. Indeed, the computational overhead due to the computation of the distance matrix of the component means does not require to recompute the data manifold, nor to re-apply Dijkstra's algorithm. However, additional computational effort is required for constructing the training manifold and the computation of its distance matrix; both are performed only once (in step 1), but can nevertheless be time consuming.

Experimental results

In this subsection, the quality of the MGMM density estimators is assessed on three 2D artificial data sets. The MGMM is compared to the ordinary GMM and the KDE. The performance measure that we use is the average negative log-likelihood of the test set.

The first distribution is distributed along a cross. The data points are generated from a uniform distribution $\mathcal{U}(-0.5, +0.5)$ in horizontal or vertical direction with probability 1/2. Gaussian noise with zero mean and standard deviation $\sigma_n = 0.03$ is added in the transversal direction. The training set and the validation set both contain 100 points, and the test set 500 points. For comparison purposes, M is fixed a priori to 4 for both mixture models. The density estimators using the optimal kernel width for the KDE ($\sigma_{opt} = 0.03$) and the optimal number of neighbors for the MGMM ($K_{opt} = 3$), as well as the ANLL are shown in Figure 3.28.

The second data set is a noisy spiral (generated similarly as the one described in Appendix A). The data are generated as follows:

$$\begin{cases} \mathbf{x}_1 = 0.04t \sin t + \epsilon_1 \ , \\ \mathbf{x}_2 = -0.04t \cos t + \epsilon_2 \ , \end{cases} \tag{3.278}$$

where $t \sim \mathcal{U}(3, 15)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 1/{\sigma_n}^2\mathbf{I})$ is zero-mean Gaussian noise. A training set of 300 points, a validation set of 300 points and a test of 1,000 points were generated. The standard deviation of the Gaussian noise σ_n is equal to 0.025. The number of components in the mixtures is fixed to 10, the optimal kernel width for the KDE is 0.025 and the optimal number of

neighbors for constructing the learning manifold is 4. The results are shown in Figure 3.28.

The third distribution has an S-shape. A training , validation and test set of respectively 100, 100 and 1,000 points are generated from the following distribution:

$$\begin{cases} \mathbf{x}_1 = 3\cos(t) + 3(-1)^z + \epsilon_1 \ , \\ \mathbf{x}_2 = 10\sin(t)(-1)^{1-z} + \epsilon_2 \ , \end{cases} \tag{3.279}$$

where $t \sim \mathcal{U}(0, \pi)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 1/0.25\mathbf{I})$ and $z \sim \mathcal{B}r(0.5)$ (Bernouilli distribution with parameter 0.5). The results for $M = 6$, $\sigma_{\text{opt}} = 0.5$ and $K_{\text{opt}} = 10$ are shown in Figure 3.28.

Visually, the MGMM gives the best results for the three experiments, the grid step being chosen sufficiently small to avoid visual artifacts. First, the MGMM provides smoother estimates than the KDE. Second, the geometric arrangement of the data is better respected with the MGMM than with the conventional GMM. In the case of the spiral, the GMM completely fails to provide a good estimate, as one component mixes two branches. Numerically, the MGMM generalizes better than the GMM in the three examples, as we observe a lower ANLL on the test set (see Fig. 3.28). Note also that the MGMM is not (too) sensitive to few unhappy edges in the learning manifold, e.g. the S-shape.

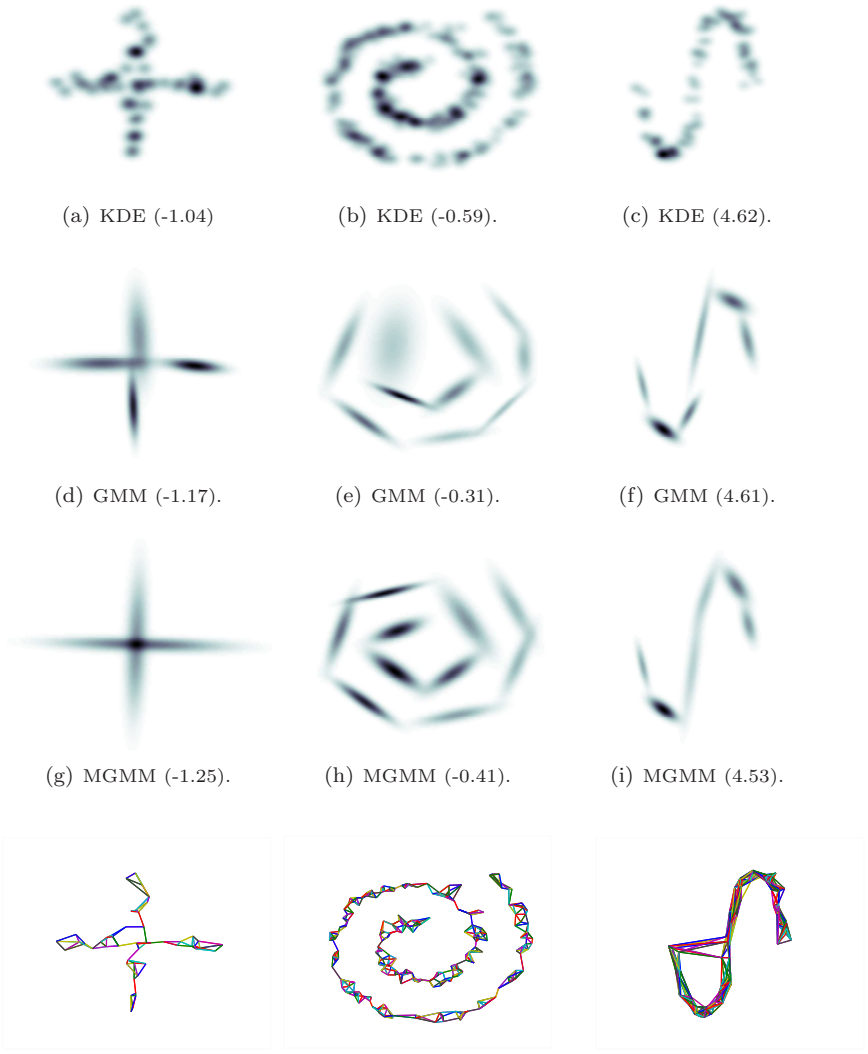
Figure 3.29 shows the evolution of the lower bound for the noisy spiral as a function of the number of training iterations. Both the unconstrained GMM and the MGMM are considered. In both cases the lower bound monotonically increases at each iteration. A similar behavior was observed for the noisy cross and the S-shape.

3.4.3. Manifold Constrained VBE-step

A similar approach can be used in the context of Bayesian mixtures, and more specifically variational mixtures (Archambeau and Verleysen, 2005b). Again, the GMM is considered as a particular case, but the approach can be readily extended to the SMM.

The exponential distribution is also used as a measure of the discrepancy between the Euclidean and the geodesic distance. However, in the Bayesian framework the model parameters are viewed as random variables. Therefore, the expectation of the component mean $\mathrm{E}\{\boldsymbol{\mu}_m\} = \mathbf{m}_m$ is used to test whether a data point is far from the corresponding component on the manifold or not. Let us denote the Euclidean and the graph distance (i.e. approximate geodesic distance) between sample $\mathbf{x}_n$ and the expected component mean $\mathbf{m}_m$ respectively by $\delta^e(\mathbf{x}_n, \mathbf{m}_m)$ and $\delta^g(\mathbf{x}_n, \mathbf{m}_m)$. The biased variational posterior can be constructed as follows:

$$q'_{z_{nm}}(z_{nm} = 1) \propto q_{z_{nm}}(z_{nm} = 1)\mathcal{E}(\delta^g(\mathbf{x}_n, \mathbf{m}_m)^2 | \delta^e(\mathbf{x}_n, \mathbf{m}_m)^2, \lambda_m) \ , \tag{3.280}$$



(a) KDE (-1.04) (b) KDE (-0.59). (c) KDE (4.62).

(d) GMM (-1.17). (e) GMM (-0.31). (f) GMM (4.61).

(g) MGMM (-1.25). (h) MGMM (-0.41). (i) MGMM (4.53).

(j) Learn. Manif. (k) Learn. Manif. (l) Learn. Manif.

FIGURE 3.28. The density estimators for the noisy cross, the noisy spiral and the S-shape. Each column shows successively the estimators of the KDE, the GMM and the MGMM. The last line is the training manifold. For each model, the ANLL of the test set is in parentheses.

where $q_{z_{nm}}(z_{nm} = 1)$ is given by (3.139). Since the responsibilities should sum to one for each $\mathbf{x}_n$, they are normalized:

$$\bar{\rho}'_{nm} = \frac{q'_{z_{nm}}(z_{nm} = 1)}{\sum_{m'=1}^{M} q'_{z_{nm'}}(z_{nm'} = 1)} \,. \tag{3.281}$$

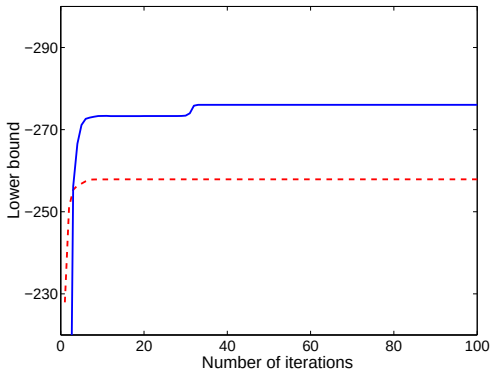


FIGURE 3.29. Lower bound on the incomplete data log-likelihood for the GMM (solid) and the MGMM (dashed) as a function of the number of training iterations.

For all components, choosing λ_m equal to 1 leaves the posterior distribution unchanged when both distances are identical. However, when the mismatch increases, $q'(z_{nm})$ decreases, which means that it is less likely that $\mathbf{x}_n$ was generated by m. This results in a weaker responsibility, reducing the influence of $\mathbf{x}_n$ when updating the variational posterior of the parameters of m in the VBM step.

Using the biased responsibilities, the expected complete data log-likelihood is

$$\begin{aligned}&\mathrm{E}_Z\{\log \mathcal{L}_c(\boldsymbol{\theta}_{\mathcal{N}}|X,Z,\mathcal{H}_M)\}\\&\quad=\textstyle\sum_{n=1}^{N}\sum_{m=1}^{M}\bar{\rho}'_{nm}\{\log\pi_m+\log\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m,\boldsymbol{\Lambda}_m)\}\ .\end{aligned}\tag{3.282}$$

The VBM update rules for the hyperparameters of the variational distributions are still given by (3.144-3.148). The intermediate quantities however take the following forms:

$$\bar{\boldsymbol{\mu}}_m=\frac{\sum_{n=1}^{N}\bar{\rho}'_{nm}\mathbf{x}_n}{\sum_{n=1}^{N}\bar{\rho}'_{nm}}\ ,\tag{3.283}$$

$$\bar{\boldsymbol{\Sigma}}_m=\frac{\sum_{n=1}^{N}\bar{\rho}'_{nm}\left(\mathbf{x}_n-\bar{\boldsymbol{\mu}}_m\right)\left(\mathbf{x}_n-\bar{\boldsymbol{\mu}}_m\right)^{\mathrm{T}}}{\sum_{n=1}^{N}\bar{\rho}'_{nm}},\tag{3.284}$$

$$\bar{\pi}_m=\frac{1}{N}\sum_{n=1}^{N}\bar{\rho}'_{nm}\ .\tag{3.285}$$

Learning algorithm

The training procedure for manifold constrained variational Gaussian mixtures (MVBGMM) can be summarized as follows:

(1) Construct the training manifold, i.e. the neighborhood graph of X, by the K-rule and compute the associated distance matrix $\delta^g(\mathbf{x}_i, \mathbf{x}_j)$ by Dijkstra's shortest path algorithm.
(2) Repeat until convergence:
Update the distance matrix of the expected component means: Find for each $\mathbf{m}_m$ the K nearest training samples $\{\mathbf{x}_k\}_{k=1}^K$ and compute its graph distance to all training data as follows:

$$\delta^g(\mathbf{x}_n, \mathbf{m}_m) = \min_k \{\delta^g(\mathbf{x}_n, \mathbf{x}_k) + \delta^e(\mathbf{x}_k, \mathbf{m}_m)\} \ . \tag{3.286}$$

VBE-step: Compute the manifold constrained responsibilities using (3.281).
VBM-step: Update the variational posteriors by first computing $\{\bar{\boldsymbol{\mu}}_m\}_{m=1}^M$, $\{\bar{\boldsymbol{\Sigma}}_m\}_{m=1}^M$ and $\{\bar{\pi}_m\}_{m=1}^M$. Next, update the hyperparameters of the variational posteriors given by (3.144-3.148).
End.

The computational overhead at each iteration step is limited with respect to the standard variational GMM, as the number of components in the mixture is usually small and updating $\delta^g(\mathbf{x}_n, \mathbf{m}_m)$ does not require to recompute the distance matrix of the manifold $\delta^e(\mathbf{x}_i, \mathbf{x}_j)$. Note however that computing $\delta^e(\mathbf{x}_i, \mathbf{x}_j)$ can be time consuming when the training set is large.

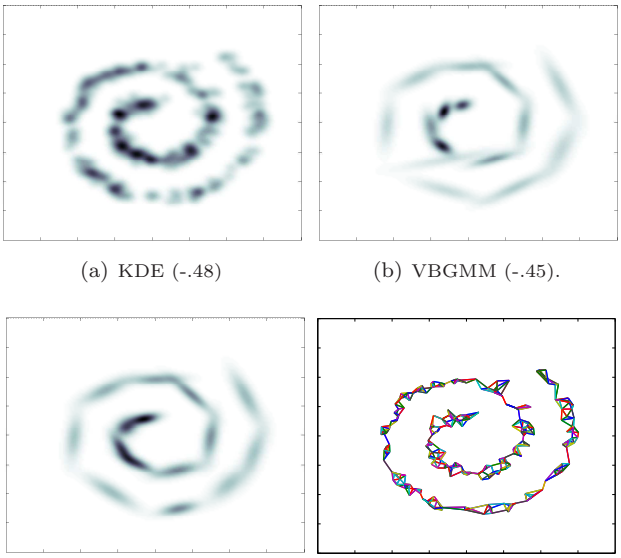
Experimental results

We end this section by briefly assessing the quality of the density estimators. The ANLL of the test set is used as performance measure. In the following, the MVBGMM is compared to the standard VBGMM and nonparametric kernel density estimation (KDE) on artificial and real data.

The first example is presented for illustrative purposes. The data are generated from a 2D noisy spiral:

$$\begin{cases} \mathbf{x}_1 = 0.04t \sin t + \epsilon_1 \ , \\ \mathbf{x}_2 = -0.04t \cos t + \epsilon_2 \ , \end{cases} \tag{3.287}$$

where $t \sim \mathcal{U}(3, 15)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 1/0.03\mathbf{I})$ is zero-mean Gaussian noise. The training, validation and test sets have respectively 300, 300 and 10,000 data points. The optimal parameters are $M = 15$ and $K = 5$. The estimators are shown in Figure 3.30. On the one hand, the MVBGMM avoids manifold related local minima in which the standard VBGMM may get trapped into. This is achieved by forcing the expected component centers to move through the training manifold and the covariance matrices to be oriented along it. On the other hand, the MVBGMM clearly produces smoother estimators than the KDE.



(a) KDE (-.48) (b) VBGMM (-.45).

(c) MVBGMM (-.50). (d) Learn. Manif.

FIGURE 3.30. Training manifold of a noisy spiral, as well as the MVBGMM, the standard VBGMM and the KDE. For each one, the ANLL of the test set is in parentheses (and the best is underlined).

In order to assess the performance of the MVBGMM on a real data set, the density of the Abalone data[2] is estimated after normalization. Note that the information regarding the sex is not used. The available data is divided in 2,500 training, 500 validation, and 1,177 test points. The optimal parameters are $M = 7$ and $K = 20$. The optimal width of the Gaussian kernel for the KDE is 0.17. The ANLL of test set for the KDE, the VBGMM and the MVBGMM are respectively 2.49, 0.84 and 0.37. The improvement of the MVBGMM compared to the VBGMM is statistically significant (the standard error of the ANLL is 0.025).

3.4.4. Related Approaches

Related approaches include mixture of probabilistic principal component analyzers (Tipping and Bishop, 1999; Bishop, 1999) and mixture of factor analyzers (Ghahramani and Beal, 1999), which were for example used for character and digit recognition. In these approaches, the data is assumed to be generated in a low dimensional latent space and then embedded in the high dimensional feature space. However, due to noise, the observed data deviate from the embedded linear subspace. In order to formalize the latent variable model, it

[2]The Abalone data is available from the UCI Machine Learning repository: http://www.ics.uci.edu/~mlearn.

is convenient to introduce a set of additional latent variables $Y = \{\mathbf{y}_n\}_{n=1}^N$, which represent the p-dimensional latent coordinate vector associated to the data vectors. Denoting the offset by $\boldsymbol{\mu}_m$ and the factor loading matrix by $\boldsymbol{\Omega}_m$, which is of size $d \times p$ with $p < d$, results in the following latent variable model:

$$p(\mathbf{y}_n) = \mathcal{N}(\mathbf{y}_n|0, \mathbf{I}) \ , \tag{3.288}$$

$$p(\mathbf{x}_n|\mathbf{y}_n, \boldsymbol{\mu}_m, \boldsymbol{\Omega_m}, \boldsymbol{\Psi}) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m + \boldsymbol{\Omega_m}\mathbf{y}_n, \boldsymbol{\Psi}^{-1}) \ , \tag{3.289}$$

where $\boldsymbol{\Psi}$ is the covariance matrix of the noise. Integrating out the latent variables leads to the standard expression for the Gaussian mixture components, where the precisions are constrained to have a particular form:

$$p(\mathbf{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_m, (\boldsymbol{\Omega_m}\boldsymbol{\Omega_m}^{\mathrm{T}} + \boldsymbol{\Psi})^{-1}) \ , \tag{3.290}$$

where $\boldsymbol{\Lambda}_m \equiv (\boldsymbol{\Omega_m}\boldsymbol{\Omega_m}^{\mathrm{T}} + \boldsymbol{\Psi})^{-1}$, $\forall\ m$. Under this standard form, the model is termed mixture of factor analyzers. When the noise covariance matrix is further constrained to be isotropic, we obtain mixtures of principal component analyzers. The parameters of both models can be estimated by means of the EM algorithm or its extensions.

Unlike the approach that we proposed in the previous section, which defines a global coordinate system associated to the manifold, the main drawback of these techniques is that the local latent spaces are not necessarily compatible with each other, meaning that the neighboring coordinate systems may have different dimensionalities or may be differently oriented. This is problematic when one wants to predict new data points as it requires to move from one factor or principal component analyzer to the other. This problem was also addressed by Verbeek, Vlassis and Kröse (2002) by forcing the successive subspaces to agree with respect to a global coordinate system. In this approach, the level of agreement is implemented by means of a penalized log-likelihood optimization problem (see for example Roweis, Saul and Hinton, 2001).

3.5. Summary

In this chapter, we first presented a unified methodology for learning latent variable models. The EM algorithm for maximum likelihood and maximum a posteriori learning was described. A modified MAP approach was also introduced in order to ease the learning procedure in practice. Next, Bayesian learning was discussed. More specifically, we showed how the variational Bayesian framework leads to an EM-like algorithm in the case of latent variable models.

Subsequently, we applied these approaches to finite Gaussian mixture models. We proposed to use the regularized Mahalanobis distance instead of the ordinary Mahalanobis distance in the maximum likelihood framework, providing an alternative to standard MAP. In general, regularization is essential when the learning set is very noisy and limited in size. The MAP approach was described in detail, emphasizing on practical solutions and discussing all aspects of the various regularization possibilities. In this context, an MML approach was also proposed for selecting the model complexity automatically. When combined to

the MAP scheme, this approach is particularly powerful. Variational Bayesian mixture models were also described. Since Bayesian methods take the uncertainty of the parameters into account, numerical difficulties encountered in maximum likelihood are avoided. Furthermore, the model complexity can be inferred without having to split the learning data in a training and a validation set.

Next, we focused on Student-t mixture models. Although Gaussian mixture models are used in many applications, they are sensitive to outliers. By contrast, finite Student-t mixture models are robust to those atypical observations. It was shown at length that most of the techniques used with Gaussian mixtures can be extended to Student-t mixtures. Furthermore, we proposed a new variational Bayesian EM learning algorithm for Bayesian Student-t mixtures, which does not assume that the posterior on the indictor variables and the scale variables factorize. In practice, the method leads to better and more robust estimators.

Finally, we introduced manifold constrained (Bayesian) mixture models. It was shown that the knowledge that the data is lying on a manifold of lower dimension than the dimension of the embedding space can be exploited when learning mixture models. By penalizing the posterior distribution of the latent indicator variables, the responsibilities are biased according to a discrepancy measure between the Euclidean and the geodesic distance. Experimentally, the resulting estimators are superior to standard variational approaches, as unacceptable local maxima of the log-likelihood function are avoided.

Through our discussion, we emphasized that finite mixture models can be used in a nonparametric-like framework. More specifically, they can be viewed as a limiting case of adaptive kernel density estimators. At various occasions, it was shown that mixture models are competitive with the most elaborate nonparametric density estimation techniques (in terms of likelihood), and are thus a powerful alternative. Furthermore, they have a much lower model complexity and can thus handle much larger data sets in practice.

CHAPTER 4

Regularization Networks

The primary aim of this chapter is to provide a comprehensive probabilistic view of regularization networks (RN) for regression. These techniques will be used in the next chapter in order to predict the characteristics of the visual sensations generated electrically in the visual field of blind patients.

The goal in regression is to infer the parameters $\mathbf{w}$ of a specific model $y(\mathbf{x};\mathbf{w})$ from a set of real valued input-target pairs $\{\mathbf{x}_n, t_n\}_{n=1}^N$ in order to generalize well on new data. RN are commonly used for this purpose (Haykin, 1999; Evgeniou, Pontil and Poggio, 2000). This regressor family includes the well-known radial basis function networks (Broomhead and Lowe, 1988; Moody and Darken, 1989), the popular support vector machines (SVM) (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000) and many related methods. The predictions are expressed as a weighted sum of nonlinear basis functions $\{\phi_m(\cdot)\}_{m=1}^M$ centered on learning prototypes (e.g., the input data):

$$y(\mathbf{x};\mathbf{w}) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x}) + w_0 = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) \ , \tag{4.1}$$

where w_0 is a bias term, $\mathbf{w} = (w_0, ..., w_M)^{\mathrm{T}}$ and $\boldsymbol{\phi}(\mathbf{x}) = (1, \phi_1(\mathbf{x}), ..., \phi_M(\mathbf{x}))^{\mathrm{T}}$. In practice, a wide variety of basis functions can be used. For example, in support vector machines, the kernels should satisfy Mercer's condition. In this work, we only consider basis functions (or kernels) having a Gaussian shape:

$$\phi_m(\mathbf{x}) = \exp\left(-\frac{\lambda_m}{2}\|\mathbf{x} - \mathbf{x}_m\|^2\right) \ , \tag{4.2}$$

where $\{\mathbf{x}_m\}_{m=1}^M$ is the set of learning prototypes and $\{\lambda_m\}_{m=1}^M$ determine the widths of the basis functions.

In the first part of the chapter, we discuss the well-known radial basis function networks. Subsequently, we make the link with the probabilistic formulation of the RN, introducing maximum likelihood and maximum a posteriori learning for regression. Next, we move on to a hierarchical Bayesian formulation. This leads to the relevance vector machines (Tipping, 1999), which are sparse Bayesian regressors. Two learning algorithms are considered. The first one is based on the evidence framework (MacKay, 1992a,b), also known as type-II maximum likelihood (Berger, 1985). The second one uses variational Bayes, which was extensively discussed in the previous chapter in the context of latent variable models. As a matter of fact, the probabilistic techniques used to learn

the parameters in regression problems are closely related to the ones discussed in Section 3.1. Finally, we end the chapter by discussing an alternative sparsity inducing algorithm, which is also based on the hierarchical Bayesian approach.

The main advantage of probabilistic regression is that the more elaborate techniques provide highly sparse approximators, which are competitive with the state-of-the-art SVM, but have fewer parameters to set. Furthermore, in the variational Bayesian formalism, we propose to select the width of the Gaussian kernels on the basis the the variational lower bound. This allows us to optimize the kernel width, which greatly influences the quality of the regressors, in a single data run. It is thus not necessary to use computationally intensive resampling techniques such as cross-validation or the bootstrap. Another advantage of the Bayesian techniques is that they provide a confidence measure, expressed as error bars in regression, for the prediction they make. This is very important in practice, especially when humans are involved such as in (bio-)medical applications.

4.1. Radial Basis Function Networks

Radial basis function networks (RBFN) have their origins in techniques for performing exact interpolation of a set of data points, which are called interpolation networks (Micchelli, 1986; Powell, 1987). Since these networks are prone to overfit, they require some form of regularization (Poggio and Girosi, 1990). Regularization techniques allow controlling the smoothness properties of the mapping function. Next, we will present how to learn the parameters of the RBFN and then discuss two types of regularization schemes.

Let us define the prediction error as the sum-of-squares:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(\mathbf{x}_n;\mathbf{w}) - t_n\}^2 \ . \tag{4.3}$$

Since this error function is quadratic with the parameters, its (unique) minimum can be found in terms of the solution of a set of linear equations. Minimizing this expression with respect to the parameters leads to

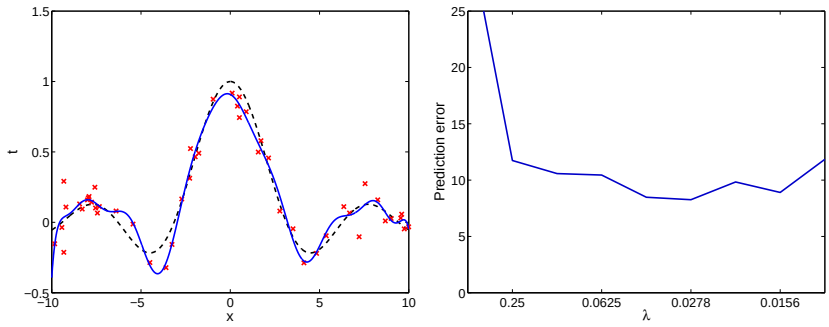
$$\sum_{n=1}^{N}\left\{t_n - \sum_{m'=0}^{M} w_{m'}\phi_{m'}(\mathbf{x}_n)\right\}\phi_m(\mathbf{x}_n) = 0 \ , \quad \forall m \ , \tag{4.4}$$

which can be written in matrix notation as

$$(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})\mathbf{w} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \ . \tag{4.5}$$

Matrix $\mathbf{\Phi}$ of size $N \times (M+1)$ with lines $\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}$ is the design matrix. Vector $\mathbf{t}$ is the vector of targets $(t_1, \ldots, t_N)^{\mathrm{T}}$. Provided the square matrix $\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}$ is non singular, it can be inverted. This leads to the least squares solution for the parameters:

$$\mathbf{w}_{\mathrm{LS}} = \mathbf{\Phi}^{\dagger}\mathbf{t} \ , \tag{4.6}$$



(a) Least squares approximator. (b) Error function.

FIGURE 4.1. (a) shows the least squares approximator (solid) obtained for the sinc function (dashed). The number of training data (crosses) is 50. The standard deviation of the Gaussian noise is 0.1. The precision of the kernels is set to 1/36. (b) shows the prediction error on the validation set as a function of the kernel precision.

where $\mathbf{\Phi}^{\dagger} \equiv (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}$ is the pseudo-inverse of $\mathbf{\Phi}$. In practice, the equations defined by (4.5) are rather solved by singular value decomposition in order to avoid problems due to a possibly ill-conditioned matrix $\mathbf{\Phi}$.

If we associate a basis function to each training data, the linear equation system (4.5) is underdetermined. As a consequence, the least squares solution does not lead to an exact interpolator, but to an approximator that has an oscillatory character. In other words, the approximator overfits the training data. From a machine learning perspective, this is undesirable as the model generalizes poorly on new data. Figure 4.1 illustrates the unregularized RBFN. The regression target is the sinc function: $f(x) = \sin(x)/x$, $x \in [-10, 10]$. The kernel precision is set to 1/36, which corresponds to the minimum of the prediction error on a validation set of 1,000 points. Remark that slightly better approximations are found for very small values of the kernel precision (or conversely very large values of the kernel standard deviation). These solutions are unacceptable however, as they lead to extreme values for the parameters. The local character of the approximator is lost and the solutions are very sensitive to numerical instabilities (as we are summing and subtracting very large numbers). Nevertheless, by introducing a number of modifications to this unregularized RBFN smooth approximators are obtained. This can be done by either constraining the parameters or reducing the number of prototypes.

4.1.1. Regularized Radial Basis Function Network

One of the simplest ways to obtain a smooth approximation function is to penalize the prediction error by the sum of squares of the parameters:

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{\eta}{2} \sum_{m=0}^{M} {w_m}^2 \ . \tag{4.7}$$

Parameter η regulates the amount of penalization. This simple regularizer is commonly known as ridge regression (Hoerl and Kennard, 1970) or weight decay. The approach penalizes large values of the parameters, which are to be avoided in practice. Indeed, it was observed experimentally that large parameter values correspond to large curvatures, which mainly occur when the approximation function overfits the data.

Minimizing (4.7) leads to the following set of linear equations:

$$\sum_{n=1}^{N} \left\{ t_n - \sum_{m'=0}^{M} w_{m'} \phi_{m'}(\mathbf{x}_n) \right\} \phi_m(\mathbf{x}_n) + \eta w_m = 0 \ , \quad \forall m \ . \tag{4.8}$$

In matrix notation, the global minimum of the prediction error is then given by

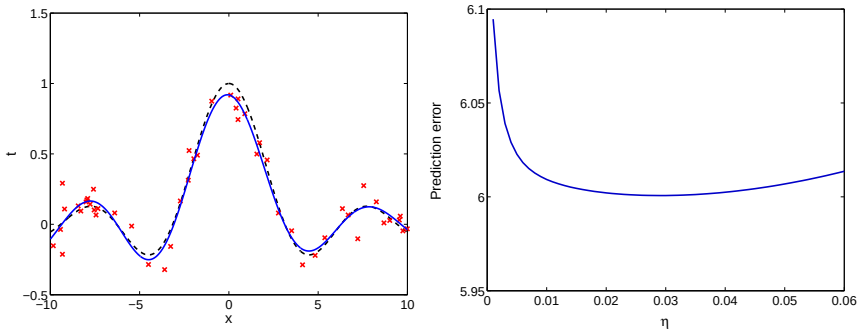
$$\mathbf{w}_{\mathrm{WD}} = (\mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi} + \eta \mathbf{I})^{-1} \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \ . \tag{4.9}$$

The approximator for the sinc function is illustrated in Figure 4.2. Weight decay is used to control the effective complexity of the approximator. Clearly, the resulting model has a higher generalization capability than the unregularized RBFN. The kernel precision is set to the same value as before. The regularization parameter is selected as the one that minimizes the prediction error on the same validation set.

Although weight decay allows tuning of the effective complexity of the regression model, the size of the matrix to invert increases linearly with the number of learning prototypes (which are often chosen to be the training data). As a result, computing the optimal parameters may be very costly for large databases. In practice, it is therefore advised to limit the number of prototypes as discussed below.

4.1.2. Vector quantization-based Radial Basis Function Network

A simple method for reducing the number of prototypes is to select a random subset of the training data. Of course, this leads to a suboptimal choice. Yet, another approach is to select the subset based on orthogonal least squares (Chen, Cowan and Grant, 1991). The basic principle of the approach is to select successively the basis function associated to the training datum which gives rise to the smallest residual prediction error. In order to be efficient, the sequential addition of basis functions is done by constructing a set of orthogonal vectors in the space spanned by the N-dimensional vectors associated to each



(a) Regularized RBFN. (b) Error function.

FIGURE 4.2. (a) shows the weight decay RBFN approximator (solid) obtained for the sinc function (dashed). The number of training data (crosses) is 50. The standard deviation of the Gaussian noise is 0.1. The precision of the kernels is set to 1/36. The optimal value for the regularization parameter is 0.029. (b) shows the prediction error on the validation set as a function of this hyperparameter.

basis function. At some point, the procedure should be stopped in order to avoid overfitting. For further details we refer to (Chen et al., 1991).

Instead of choosing a subset of the training data, vector quantization (VQ) can be used (Moody and Darken, 1989) in order to find a set of prototypes that better reflects the distribution of the training set. Among the most popular ones, we have M-means[1] (MacQueen, 1967), discussed in Section 3.2, competitive learning (Grossberg, 1987; Ahalt et al., 1990), discussed in Section 2.3.3, and neural-gas (Martinetz et al., 1993). Other unsupervised techniques that can be used for this purpose include Kohonen's self-organizing maps (Kohonen, 1995) and finite Gaussian mixture models. The latter were extensively discussed in the previous chapter. Note that in the case of the Gaussian mixture models, once the component means and precisions are estimated, the mixture proportions can be discarded as they are no longer needed for regression.

The training algorithm of the VQ-based RBFN is split into two steps:

(1) The kernel centers, i.e. the training prototypes, and the kernel precisions are estimated by vector quantization techniques.
(2) The model parameters are computed by (4.6).

The first step is unsupervised, meaning that the kernel parameters are adjusted without taking the values of the training targets $\{t_n\}_{n=1}^N$ into account. In practice, vector quantization often minimizes a reconstruction error. By contrast,

[1]Recall that we denote the complexity by M rather than by K.

the second step is supervised. The parameters minimizing the prediction error are computed. In this step, the parameters of the kernels are fixed.

Two alternatives are typically considered for the estimation of the kernel precisions. The first one consists in taking the precisions equal to a constant for all basis functions (see for example Park and Sandberg, 1991; Haykin, 1999). Haykin (1999) sets the precisions as follows:

$$\lambda_m = \left(\frac{d_{\max}}{\sqrt{2M}} \right)^{-2} , \quad \forall m , \tag{4.10}$$

where $d_{\max}$ is the maximum distance between the prototypes. Such a procedure fixes the degree of overlapping of the Gaussian basis functions a priori. It allows finding a compromise between locality and smoothness of the approximator. This choice would be close to the optimal solution if the data were uniformly distributed in the input space, leading to a uniform distribution of the prototypes. Unfortunately, most real-life problems show non-uniform data distributions. The method is thus inadequate in practice and an identical precision for all the kernels should be avoided. The precisions should depend on the position of the prototypes, which in turn depends on the data distribution in the input space.

The second option consists in estimating independently the precisions of the Gaussian basis functions to take the variations in the distribution of the data into account. This can be done by simply computing the inverse variance of the distances between the data and their closest prototype. Verleysen and Hlavackova (1996) suggested an iterative procedure for estimating this standard deviation. Moody and Darken (1989), in contrast, proposed to compute the precisions by the r nearest neighbors heuristic:

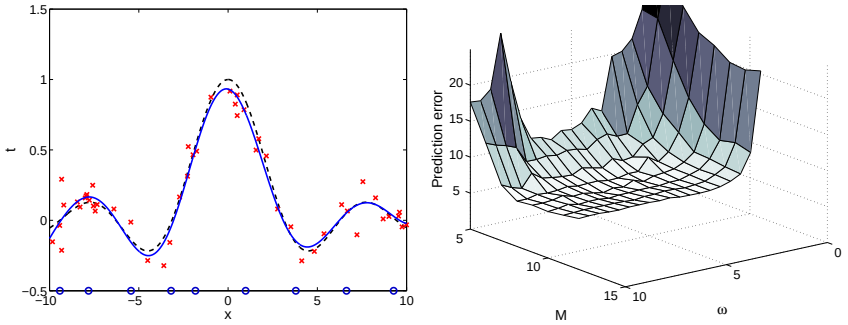
$$\lambda_m = \left(\frac{1}{r} \sqrt{\sum_{i=1}^{r} \left\| \mathbf{x}_i^{(m)} - \mathbf{x}_m \right\|^2} \right)^{-2} , \quad \forall m , \tag{4.11}$$

where $\{\mathbf{x}_i^{(m)}\}_{i=1}^{r}$ are the r-nearest neighbors of prototype $\mathbf{x}_m$. In general, these methods provide indeed locally adjusted precisions, but in some cases the overlap of the basis functions is not sufficient, possibly leading to poor generalization.

More recently, it was proposed to combine the advantages of both approaches in a more principled way (Benoudjit, Archambeau, Lendasse, Lee and Verleysen, 2002). After having computed the M empirical standard deviations of the distances between the data points and their closest prototype, these quantities are multiplied by a common scaling factor ω, which is termed width scaling factor. The resulting precisions are given by

$$\lambda_m = (\omega \hat{\sigma}_m)^{-2} , \quad \forall m . \tag{4.12}$$

In this equation, $\{\hat{\sigma}_m\}_{m=1}^{M}$ are the standard deviations associated to the prototypes. A different precision is thus used for each prototype, but the amount of smoothing, i.e. the overlap of the basis functions, is further controlled by



(a) VQ-based RBFN. (b) Error surface.

FIGURE 4.3. (a) shows the VQ-based RBFN approximator (solid) obtained for the sinc function (dashed). The prototypes are found by competitive learning. Their location is indicated by circles at the bottom of the figure. The number of training data (crosses) is 50. The standard deviation of the Gaussian noise is 0.1. The number of prototypes is 9 and the width scaling factor is set to 6.5. (b) shows the prediction error on the test set (1,000 points) as a function of the number of basis functions and the width scaling factor. The curve is an average on 20 training runs.

means of a common tuning parameter. This parameter can be optimized by resampling techniques, such as cross-validation, in order to have the lowest residual prediction error.

Figure 4.3 shows the approximator obtained for the VQ-based RBFN. The target is the sinc function. Both the number of prototypes and the width scaling factor are selected as values that minimize the prediction error (4.3) on the validation set. It can be observed that although the number of basis functions is much smaller than for weight decay, the approximator has a comparable accuracy.

4.2. Probabilistic View of Regularization Networks

In the regularization networks (RN) presented in the previous section, the number of basis function is either equal to the number of training data, or is selected by resampling techniques. Furthermore, these approaches do not provide any measure of the confidence of the prediction they make. In this section, a probabilistic view of RN is presented. In particular, probabilistic RN include unconstrained RBFN and ridge regression as special cases. However, probabilistic RN also capture the uncertainty of the predictions they make in the form of error bars. In addition, when considering the Bayesian framework, sparse solutions can be obtained, meaning that parameters that are irrelevant

are driven to zero during the training procedure. This is important, as the degree of sparseness, which is also a key idea behind support vector machines, improves the generalization abilities of the approximation functions (Cristianini and Shawe-Taylor, 2000).

4.2.1. Maximum Likelihood Learning

Following a standard probabilistic formulation, the noisy targets can be decomposed as follows:

$$t_n = y(\mathbf{x}_n; \mathbf{w}) + \epsilon_n \ , \tag{4.13}$$

where $y(\mathbf{x}; \mathbf{w})$ is the regression model defined in (4.1). The error terms $\{\epsilon_n\}_{n=1}^N$ are assumed to be independently drawn from a Gaussian distribution $\mathcal{N}(\epsilon_n|0, \tau)$. Note that τ is the noise precision or inverse variance. Using this noise model and assuming the input-target pairs are i.i.d., the joint distribution or (target) likelihood is given by

$$\mathcal{L}(\mathbf{w}, \tau|\mathbf{t}) \equiv p(\mathbf{t}|\mathbf{w}, \tau) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(\mathbf{x}_n; \mathbf{w}), \tau) \ . \tag{4.14}$$

Remark that the conditional dependency on the input data $X = \{\mathbf{x}_n\}_{n=1}^N$ is omitted for the sake of simplicity and that M is equal to N.

When the width of the basis functions is fixed a priori, maximizing the likelihood (or equivalently the log-likelihood) w.r.t. the parameters $\mathbf{w}$ and the noise precision τ leads to the following equations:

$$\frac{\partial \log \mathcal{L}}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w}_{\mathrm{ML}} = \mathbf{\Phi}^{\dagger}\mathbf{t} \ , \tag{4.15}$$

$$\frac{\partial \log \mathcal{L}}{\partial \tau} = 0 \quad \Rightarrow \quad \tau_{\mathrm{ML}} = \left\{ \frac{\|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2}{N} \right\}^{-1} \ . \tag{4.16}$$

The predictive distribution is then given by

$$p(t|\mathbf{t}) \approx p(t|\mathbf{w}_{\mathrm{ML}}, \tau_{\mathrm{ML}}) = \mathcal{N}(t|y(\mathbf{x}; \mathbf{w}_{\mathrm{ML}}), \tau_{\mathrm{ML}}) \ . \tag{4.17}$$

From (4.6), we note that the maximum likelihood (ML) solution for the parameters is the same as the one obtained in the case of the unregularized RBFN. Observe also that the estimated noise variance is nothing else than the average prediction residual or unexplained variance, since $\|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2 = \sum_{n=1}^{N} \{t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)\}^2$.

As shown experimentally in the previous section, using unconstrained approximators leads to severe overfitting. As a result, maximizing the likelihood $\mathcal{L}(\mathbf{w}, \tau|\mathbf{t})$ does not provide an acceptable solution, unless an additional regularization term is introduced to constrain the model towards a simpler form. This is addressed in the next section.

4.2.2. Maximum a Posteriori Learning

In order to take the uncertainty of the parameters into account, a prior probability distribution is imposed on them. Here, we consider a Gaussian prior on the parameters. Furthermore, a Gamma prior is imposed on the noise precision. This leads to the following priors:

$$p(\tau|a_0, b_0) = \mathcal{G}(\tau|a_0, b_0) \ , \tag{4.18}$$

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{m=0}^{M} \mathcal{N}(w_m|0, \alpha_m) \ , \tag{4.19}$$

where the hyperparameters are a_0, b_0 and $\boldsymbol{\alpha} = (\alpha_0, ..., \alpha_M)^{\mathrm{T}}$.

The maximum a posteriori (MAP) likelihood, which is proportional to the posterior distribution of the parameters $p(\mathbf{w}|\mathbf{t})$, is obtained by applying Bayes' rule. Taking the logarithm leads to

$$\log \mathcal{L}_{\mathrm{MAP}}(\mathbf{w}, \tau|\mathbf{t}) = \log p(\mathbf{t}|\mathbf{w}, \tau) + \log p(\tau|a_0, b_0) + \log p(\mathbf{w}|\boldsymbol{\alpha}) \tag{4.20}$$

$$= \log \mathcal{L}(\mathbf{w}, \tau|\mathbf{t}) + \log p(\tau|a_0, b_0) + \log p(\mathbf{w}|\boldsymbol{\alpha}) \ . \tag{4.21}$$

For a fixed kernel width and fixed hyperparameters, a set of coupled equations is obtained by maximizing the penalized log-likelihood (or log-posterior) w.r.t. the parameters $\mathbf{w}$ and the noise precision τ:

$$\frac{\partial \log \mathcal{L}_{\mathrm{MAP}}}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} \leftarrow \tau \left(\tau \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} + \mathbf{A} \right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \ , \tag{4.22}$$

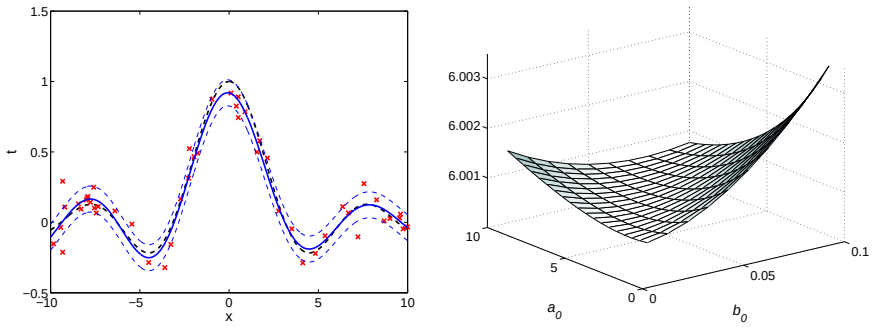
$$\frac{\partial \log \mathcal{L}_{\mathrm{MAP}}}{\partial \tau} = 0 \quad \Rightarrow \quad \tau \leftarrow \left\{ \frac{\|\mathbf{t} - \boldsymbol{\Phi}\mathbf{w}\|^2 + 2b_0}{N + 2(a_0 - 1)} \right\}^{-1} \ . \tag{4.23}$$

Matrix $\mathbf{A}$ is diagonal, its non-zero elements being equal to the vector of hyperparameters $\boldsymbol{\alpha}$. Applying both equations alternatively converges to the MAP solution $\mathbf{w}_{\mathrm{MAP}}$ and τ_{MAP}. Plugging these estimates in (4.17) leads to the MAP predictive distribution. When $\{\alpha_m\}_{m=0}^{M}$ are constrained to be equal, the well known weight decay regularizer for the RBFN is found. The regularization constant η is then equal to $\alpha/\tau_{\mathrm{MAP}}$.

Figure 4.4 shows the approximator for the sinc function, as well as the prediction surface versus a_0 and b_0. The algorithm converges after few iteration steps (less than 10). For appropriate values of the hyperparameters, a smooth approximation function is found. In addition, the amount of noise can be estimated. The true noise value is underestimated in this example.

The main drawback of this method is the high number of hyperparameters to optimize, so all elements of $\boldsymbol{\alpha}$ are usually set to the same value. Moreover, instead of optimizing a_0 and b_0, they are rather set to small values in order to express our ignorance about the scale of the noise precision. The resulting prior is flat and it approaches Jeffrey's noninformative prior (see for example Berger, 1985):

$$p(\tau) \propto \frac{1}{\tau} \ . \tag{4.24}$$



(a) MAP RN. (b) Error function.

FIGURE 4.4. (a) shows the MAP RN approximator (solid) obtained for the sinc function (dashed). The true noise standard deviation (0.1) is underestimated (0.085). The noise tube (one standard deviation) is also shown with thin dashed lines. The number of training data (crosses) is 50. The precision of the kernels is set to 6 and each regularization parameter to 3.4, which is the optimal value in terms of prediction error. (b) shows the error surface of the test set (1,000 points) as a function of a_0 and b_0. These parameters are respectively set to 2 and 0.008.

Note that this prior is improper, meaning that it does not integrate to one. Interesting facts about this prior is that it is parameter-free and scale invariant. Using Jeffrey's prior on the noise precision instead of a Gamma prior leads to the following update formula for the noise precision:

$$\frac{\partial \log \mathcal{L}_{\text{MAP}}}{\partial \tau} = 0 \quad \Rightarrow \quad \tau \leftarrow \left\{ \frac{\|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2}{N-2} \right\}^{-1} . \tag{4.25}$$

Observe that the estimate of the noise variance is biased downwards, especially for small data sets.

4.2.3. Bayesian Learning: the Relevance Vector Machine

In MAP learning, we do not deal properly with the uncertainty of the parameters. Predictions are made on the basis of point-estimates, rather than using the posterior distribution of the parameters. Furthermore, in order to deal more efficiently with the uncertainty of the hyperparameter vector $\boldsymbol{\alpha}$, which plays a crucial role in the quality of the approximator, and address the problem of model selection, a hierarchical Bayesian approach is used. Since the hyperparameters are scale variables, a Gamma hyperprior is imposed on them:

$$p(\boldsymbol{\alpha}) = \prod_{m=0}^{M} \mathcal{G}(\alpha_m | c_{m0}, d_{m0}) . \tag{4.26}$$

When using a different hyperparameter α_m for each w_m, sparsity is achieved because the posterior distribution of most parameters is sharply peaked around zero (Tipping, 2001). The predictive distribution is obtained by integrating out all the parameters and the hyperparameters:

$$p(t|\mathbf{t}) = \iiint p(t|\mathbf{w}, \boldsymbol{\alpha}, \tau, \mathbf{t}) p(\mathbf{w}, \boldsymbol{\alpha}, \tau|\mathbf{t}) d\mathbf{w} d\boldsymbol{\alpha} d\tau \tag{4.27}$$

$$= \iiint p(t|\mathbf{w}, \tau) p(\mathbf{w}, \boldsymbol{\alpha}, \tau|\mathbf{t}) d\mathbf{w} d\boldsymbol{\alpha} d\tau \ , \tag{4.28}$$

where it is assumed that the prediction t is independent of $\mathbf{t}$ and $\boldsymbol{\alpha}$ given $\mathbf{w}$ and τ. Unfortunately, this integral is intractable in practice. This can be understood by looking to the second factor on the right-hand-side in (4.28), which is the posterior distribution of the parameters and the hyperparameters. This distribution can be decomposed by the product rule as follows:

$$p(\mathbf{w}, \boldsymbol{\alpha}, \tau|\mathbf{t}) = p(\mathbf{w}|\boldsymbol{\alpha}, \tau, \mathbf{t}) p(\boldsymbol{\alpha}, \tau|\mathbf{t}) \ . \tag{4.29}$$

First, the posterior distribution of the parameters $p(\mathbf{w}|\boldsymbol{\alpha}, \tau, \mathbf{t})$ can be computed analytically, as the normalizing integral is a convolution of two Gaussian distributions:

$$p(\mathbf{w}|\boldsymbol{\alpha}, \tau, \mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{w}, \tau) p(\mathbf{w}|\boldsymbol{\alpha})}{\int p(\mathbf{t}|\mathbf{w}, \tau) p(\mathbf{w}|\boldsymbol{\alpha}) d\mathbf{w}} = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}) \ , \tag{4.30}$$

where $p(\mathbf{t}|\mathbf{w}, \tau) = \mathcal{N}(\mathbf{t}|\boldsymbol{\Phi}\mathbf{w}, \tau\mathbf{I})$ and $p(\mathbf{w}|\boldsymbol{\alpha})$ is defined in (4.19). The posterior mean $\boldsymbol{\mu}$ and the posterior covariance matrix $\boldsymbol{\Sigma}$ are respectively given by

$$\boldsymbol{\mu} = \tau \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \ , \tag{4.31}$$

$$\boldsymbol{\Sigma} = \left(\tau \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} + \mathbf{A} \right)^{-1} \ , \tag{4.32}$$

where $\mathbf{A} = \mathrm{diag}\{\alpha_0, \ldots, \alpha_M\}$. Note that the form of the posterior mean is identical to the form of the MAP estimate of the parameters in (4.22).

Second, using Bayes' rule we may write the posterior of the hyperparameters and the noise precision as follows:

$$p(\boldsymbol{\alpha}, \tau|\mathbf{t}) = \frac{p(\mathbf{t}|\boldsymbol{\alpha}, \tau) p(\boldsymbol{\alpha}) p(\tau)}{p(\mathbf{t})} \ . \tag{4.33}$$

This distribution cannot be computed analytically as the integral involved in the normalizing factor is untractable. Therefore, we need to make some approximations. Below we investigate two approaches. The first method is based on evidence maximization (MacKay, 1992a), which is also known as type-II maximum likelihood (Berger, 1985), and the second one is a variational Bayesian approach, of which the general principle was discussed in detail in Section 3.1.3.

Evidence maximization

Since we cannot compute the exact posterior $p(\boldsymbol{\alpha}, \tau|\mathbf{t})$, we seek its mode. By doing so, we assume that the corresponding point-estimates of $\boldsymbol{\alpha}$ and τ are representative of the posterior, in the sense that the approximator using these

values is nearly identical as the one using the full posterior. It is important to realize that this does not require the entire mass of the posterior be accurately approximated by the corresponding delta function. According to Tipping (2001), this approximation is effective in practice.

The marginal likelihood or evidence $p(\mathbf{t}|\boldsymbol{\alpha}, \tau)$, which is also the normalizing constant in (4.30), is given by

$$p(\mathbf{t}|\boldsymbol{\alpha}, \tau) = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{S}^{-1}) \ , \tag{4.34}$$

where

$$\mathbf{S} = \tau^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{\mathrm{T}} \ . \tag{4.35}$$

Following a MAP approach, we maximize $p(\boldsymbol{\alpha}, \tau|\mathbf{t}) \propto p(\mathbf{t}|\boldsymbol{\alpha}, \tau)p(\boldsymbol{\alpha})p(\tau)$ w.r.t. $\boldsymbol{\alpha}$ and τ. This leads to a set of coupled update rules:

$$\frac{\partial \log p(\boldsymbol{\alpha}, \tau|\mathbf{t})}{\partial \alpha_m} = 0 \quad \Rightarrow \quad \alpha_m \leftarrow \left\{ \frac{{\mu_m}^2 + \Sigma_{mm} + 2d_{m0}}{1 + 2(c_{m0} - 1)} \right\}^{-1} \ , \tag{4.36}$$

$$\frac{\partial \log p(\boldsymbol{\alpha}, \tau|\mathbf{t})}{\partial \tau} = 0 \quad \Rightarrow \quad \tau \leftarrow \left\{ \frac{\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 + \mathrm{tr}\{\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\} + 2b_0}{N + 2(a_0 - 1)} \right\}^{-1} \ , \tag{4.37}$$

where $\mathrm{tr}\{\cdot\}$ is the trace operator and $\{\Sigma_{mm}\}_{m=0}^{M}$ are the diagonal elements of $\boldsymbol{\Sigma}$. Successively applying (4.36) and (4.37), while updating the posterior mean (4.31) and the posterior covariance matrix (4.32) leads to a highly sparse Bayesian approximator: the relevance vector machine (RVM) (Tipping, 1999). Note that in contrast to the approach followed here, Tipping (2001) takes the derivatives with respect to the logarithm of the scale variables $\{\alpha_m\}_{m=0}^{M}$ and τ, as he assumes uniform hyperpriors over the logarithmic scale. As a result, the independent term in the denominator of (4.36) and (4.37) vanishes. The update equations are then identical to the ones obtained in variational RVM, which will be discussed shortly.

An attractive property of the training algorithm for RVM is that it is guaranteed to maximize locally (4.33), as it is equivalent to apply the EM algorithm. Indeed, if we treat $\mathbf{w}$ as a "hidden" variable, the quantity on the right-hand-side in (4.33) can be considered as a penalized incomplete likelihood. The E-step consists then in computing the posterior distribution $p(\mathbf{w}|\boldsymbol{\alpha}, \tau, \mathbf{t})$. This is done by updating the posterior mean (4.31) and the posterior covariance matrix (4.32). In the M-step, the expected complete log-posterior $\mathrm{E}_{\mathbf{w}}\{\log p(\mathbf{t}|\mathbf{w}, \tau)p(\mathbf{w}|\alpha)p(\boldsymbol{\alpha})p(\tau)\}$ is maximized with respect to $\boldsymbol{\alpha}$ and τ. This leads to (4.36) and (4.37). The graphical representation of the RVM is shown in Figure 4.5.

The main disadvantage of RVM is the computational complexity of the learning algorithm. Following MacKay (1992a) in defining the quantities $\gamma_m \equiv 1 -$

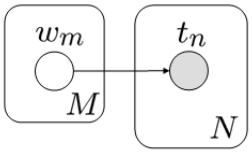


FIGURE 4.5. Graphical model of the RVM. Note that $n \in \{1, \ldots, N\}$ and $m \in \{0, \ldots, M\}$.

$\alpha_m \Sigma_{mm}$, the updates (4.36) and (4.37) can be rewritten as follows:

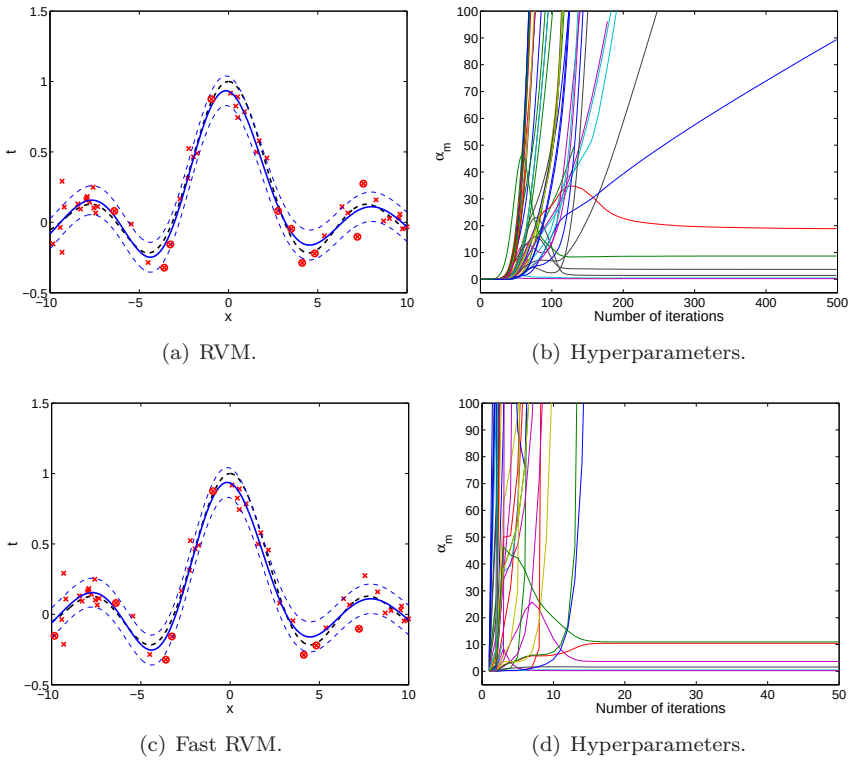
$$\alpha_m \leftarrow \left\{ \frac{{\mu_m}^2 + 2d_{m0}}{\gamma_m + 2(c_{m0} - 1)} \right\}^{-1}, \tag{4.38}$$

$$\tau \leftarrow \left\{ \frac{\|\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu}\|^2 + 2b_0}{N - \sum_{m=0}^{M} \gamma_m + 2(a_0 - 1)} \right\}^{-1}. \tag{4.39}$$

Although these updates do not guarantee a local maximization of the penalized log-likelihood, they were observed to lead to much faster convergence (Tipping, 2001). Furthermore, each γ_m can be interpreted as a measure of how well-determined its corresponding w_m is by the data (Gull, 1989). This can be understood by looking at (4.32) and seeing that when α_m is large, the posterior of w_m is highly constrained by the prior, such that $\Sigma_{mm} \approx {\alpha_m}^{-1}$ and consequently $\gamma_m \approx 0$. Conversely, when α_m is small, the posterior w_m fits to the data and $\gamma_m \approx 1$. More recently, a greedy variant was also proposed in order to accelerate the learning algorithm when dealing with large data sets (Tipping and Faul, 2003).

As mentioned before, the central idea behind RVM is to associate a different hyperparameter α_m to each parameter w_m. When imposing a Gamma prior on each α_m, the marginal prior $p(w_m)$ is a Student-t distribution. This prior resembles the Laplacian prior, which is equivalent to the L_1-regularizer when taking the logarithm. As the Laplacian distribution, the Student-t distribution is symmetric and has heavier tails than the Gaussian one. It is well known that the zero-mean Laplacian prior induces sparsity (see for example Williams, 1995; Tibshirani, 1996). In the RVM context, this prior is thus approximated by adopting a hierarchical Bayesian approach. As a result, the marginal posterior distribution of each parameter $p(w_m|\alpha_m, \tau)$ is highly peaked around zero. The corresponding hyperparameter is therefore driven to infinity (or rather very large values). Note that since the expected value of the parameters is non-zero (but very small), sparsity is realized in practice through thresholding.

The approximators for the sinc function is shown in Figure 4.6. The ordinary and the fast learning algorithm is used. The second one tends to provide sparser solutions. The estimate of the hyperparameters as a function of the number of training iterations is also shown. It can be observed that the modified learning algorithm converges much faster than the ordinary one, while the quality of



(a) RVM. (b) Hyperparameters.

(c) Fast RVM. (d) Hyperparameters.

FIGURE 4.6. (a) and (c) show respectively the RVM approximator and the one obtained with the fast learning rule. The solid curves indicate the approximators and the dashed curves the target function. The true noise standard deviation (0.1) is only slightly underestimated (0.098) by both algorithms. The noise tube (one standard deviation) is also shown with thin dashed lines. The number of training data (crosses) is 50. The precision of the kernels is set to 1/6. The hyperparameters $\{a_0, b_0\}$ and $\{c_{m0}, d_{m0}\}_{m=0}^{M}$ are set to small values, resulting in broad priors. The number of relevance vector (circles) found by the algorithm is respectively 10 and 8, the threshold being set to 10^{-3}. (b) and (d) show the evolution of $\{\alpha_m\}_{m=0}^{M}$ as a function of the number of training iterations. Most of them are driven to infinity during the training process.

the solutions that are obtained are very close in terms of log-evidence (see Figure 4.7).

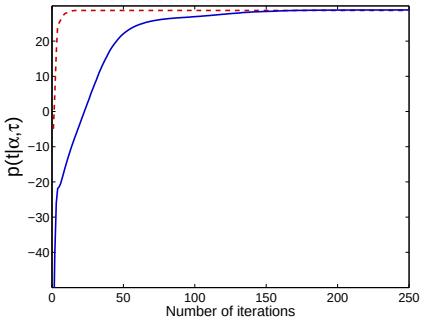


FIGURE 4.7. Log-evidence of the hyperparameters and the noise precision as a function of the number of training iterations. The RVM learnt by the ordinary and the modified updates correspond respectively to the solid and the dashed curve.

Finally, the predictive distribution of an unseen target on a new data point is obtained by marginalizing over the parameters:

$$p(t|\mathbf{t}) \approx p(t|\boldsymbol{\alpha}^*, \tau^*, \mathbf{t}) = \int p(t|\mathbf{w}, \tau^*) p(\mathbf{w}|\boldsymbol{\alpha}^*, \tau^*, \mathbf{t}) d\mathbf{w} \ , \tag{4.40}$$

where $\boldsymbol{\alpha}^*$ and τ^* denote respectively the values of the hyperparameter vector and the noise precision that maximize the evidence. The first distribution in this expression is the likelihood. The second is obtained by plugging $\boldsymbol{\alpha}^*$ and τ^* in the posterior distribution of the parameters. Since both are Gaussian distributions, the integral is tractable:

$$p(t|\boldsymbol{\alpha}^*, \tau^*, \mathbf{t}) = \mathcal{N}(t|\mu_t, \lambda_t) \ , \tag{4.41}$$

with

$$\mu_t = y(\mathbf{x}; \boldsymbol{\mu}) \ , \tag{4.42}$$

$$\lambda_t = \left\{ \frac{1}{\tau^*} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{\phi}(\mathbf{x}) \right\}^{-1} \ . \tag{4.43}$$

The predictive mean μ_t is thus a prediction based on the posterior mean $\boldsymbol{\mu}$ of the parameters. The predictive variance $1/\lambda_t$ is called error bar. It provides a confidence measure in each prediction the model makes. The error bars contain two terms. The first is an estimate of the noise variance. It corresponds to the unexplained variance by the model as in MAP learning. The second is an uncertainty measure on the parameters $\mathbf{w}$.

Variational Bayes

An alternative to evidence maximization for learning the RVM is the variational Bayesian approach (Bishop and Tipping, 2000). For a detailed discussion of the variational Bayesian framework, we refer to Section 3.1.3. Figure 4.8 shows

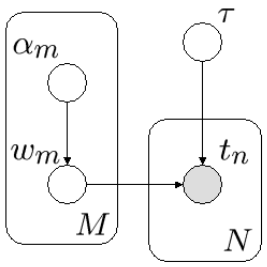


FIGURE 4.8. Graphical model of the variational RVM.

the graphical representation of the variational RVM. In this approach, the hyperparameters and the noise precision are considered as latent variables as well.

Consider again the marginal likelihood $p(\mathbf{t})$, which is the normalizing constant in (4.33). This quantity is in practice intractable. However, for any distribution $q(\mathbf{w}, \boldsymbol{\alpha}, \tau)$ the logarithm of $p(\mathbf{t})$ can be lowerbounded using Jensen's inequality (Jensen, 1906):

$$\log p(\mathbf{t}) \geq \iiint q(\mathbf{w}, \boldsymbol{\alpha}, \tau) \log \frac{p(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha}, \tau)}{q(\mathbf{w}, \boldsymbol{\alpha}, \tau)} d\mathbf{w} d\boldsymbol{\alpha} d\tau \ . \tag{4.44}$$

The bound is made tight when equating $q(\mathbf{w}, \boldsymbol{\alpha}, \tau)$ to the joint posterior distribution $p(\mathbf{w}, \boldsymbol{\alpha}, \tau|\mathbf{t})$. In the variational Bayesian setting, this bound is iteratively maximized through a factorized approximation of the joint posterior distribution of the parameters:

$$q(\mathbf{w}, \boldsymbol{\alpha}, \tau) = q_{\mathbf{w}}(\mathbf{w}) q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) q_{\tau}(\tau) \ . \tag{4.45}$$

Treating the parameters as the hidden variables leads to the following variational update equations:

$$\textbf{VBE-step} \ : \ q_{\mathbf{w}}(\mathbf{w}) \propto \exp\left(\mathrm{E}_{\boldsymbol{\alpha},\tau}\{\log p(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha}, \tau)\}\right) \ . \tag{4.46}$$

$$\textbf{VBM-step} : \ q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \propto p(\boldsymbol{\alpha}) \exp\left(\mathrm{E}_{\mathbf{w},\tau}\{\log p(\mathbf{t}, \mathbf{w}, \tau|\boldsymbol{\alpha})\}\right) \ , \tag{4.47}$$

$$q_{\tau}(\tau) \propto p(\tau) \exp\left(\mathrm{E}_{\mathbf{w},\boldsymbol{\alpha}}\{\log p(\mathbf{t}, \mathbf{w}, \boldsymbol{\alpha}|\tau)\}\right) \ . \tag{4.48}$$

In these equations, the expectations are taken with respect to the variational distributions.

The VBE-step leads to a variational posterior of the parameters which is in agreement with the exact posterior (4.30):

$$q_{\mathbf{w}}(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}) \ . \tag{4.49}$$

The posterior mean and the posterior covariance matrix are given by

$$\boldsymbol{\mu} = \bar{\tau} \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \ , \tag{4.50}$$

$$\boldsymbol{\Sigma} = \left(\bar{\tau} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} + \bar{\mathbf{A}}\right)^{-1} \ , \tag{4.51}$$

where the special quantities are $\bar{\tau} \equiv \mathrm{E}_\tau\{\tau\} = a/b$ and $\bar{\mathbf{A}} = \mathrm{diag}\{\bar{\alpha}_0, \ldots, \bar{\alpha}_M\}$ with $\bar{\alpha}_m \equiv \mathrm{E}_{\boldsymbol{\alpha}}\{\alpha_m\} = c_m/d_m$, $\forall m$. The values that are used to compute the parameters of the posterior distribution are thus the mean of the noise precision and the means of the hyperparameters. In contrast, the values that are plugged into the true posterior in the maximum evidence framework correspond to the mode of the evidence $p(\boldsymbol{\alpha}, \tau|\mathbf{t})$.

Since the priors $p(\boldsymbol{\alpha})$ and $p(\tau)$ are conjugate to the exponential family, computing their posterior consists in updating their parameters. The resulting VBM-step is then given by

$$a = a_0 + \frac{N}{2} \ , \quad b = b_0 + \frac{\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 + \mathrm{tr}\{\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\}}{2} \ , \tag{4.52}$$

$$c_m = c_{m0} + \frac{1}{2} \ , \quad d_m = d_{m0} + \frac{{\mu_m}^2 + \Sigma_{mm}}{2} \ . \tag{4.53}$$

Figure 4.9 shows the approximator for the sinc function. The RVM learnt by the variational Bayes is very close to the one learnt by maximum evidence. Note that when considering different training sets, both algorithms select on average 7 to 8 relevance vectors. By contrast, it was reported by Bishop and Tipping (2000) that the standard SVM selects on average 28 support vectors on this example for a similar accuracy. The RVM provides thus sparser solutions than the standard SVM.

To conclude, we approximate the predictive distribution by replacing the true posterior by its variational approximation:

$$p(t|\mathbf{t}) \approx \int p(t|\mathbf{w}, \bar{\tau}) q_{\mathbf{w}}(\mathbf{w}) d\mathbf{w} \ . \tag{4.54}$$

Note that we have used the fact that $q_\tau(\tau)$ is highly peaked around its mean value. This is indeed the case for large training sets. Since $p(t|\mathbf{w}, \bar{\tau})$ and $q_{\mathbf{w}}(\mathbf{w})$ are both Gaussian distributions, the integral is tractable:

$$p(t|\bar{\boldsymbol{\alpha}}, \bar{\tau}, \mathbf{t}) = \mathcal{N}(t|\mu_t, \lambda_t) \ , \tag{4.55}$$

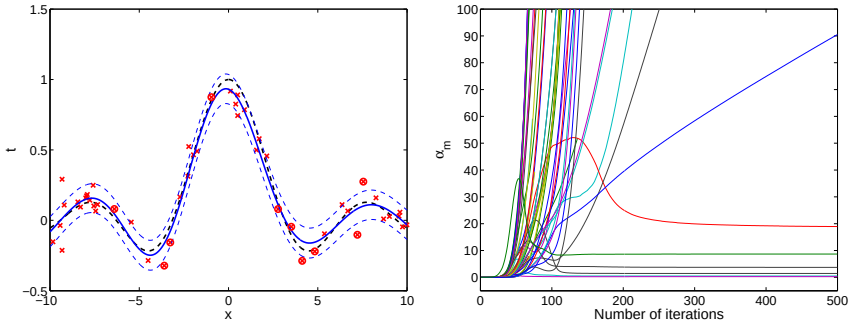
where

$$\mu_t = y(\mathbf{x}; \boldsymbol{\mu}) \ , \tag{4.56}$$

$$\lambda_t = \left\{ \frac{1}{\bar{\tau}} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{\phi}(\mathbf{x}) \right\}^{-1} \ . \tag{4.57}$$

4.2.4. Bayesian selection of the basis functions' precision

When using the RVM or the variational RVM, one problem remains: the optimization of the basis functions precisions. The precision greatly influences the quality of the approximator. Unfortunately, no simple reestimation formula exists as for the other hyperparameters. Since isotropic basis functions are used in practice, it is common to select the precision by resampling techniques, such as cross-validation or the bootstrap. In the Bayesian context, however, we



(a) Variational RVM. (b) Mean of the hyperparameters.

FIGURE 4.9. (a) shows the variational Bayesian RVM approximator (solid) obtained for the sinc function (dashed). The true noise standard deviation (0.1) is only slightly underestimated (0.098). The noise tube (one standard deviation) is also shown with thin dashed lines. The number of training data (crosses) is 50. The precision of the kernels is set to 1/6. The hyperparameters $\{a_0, b_0\}$ and $\{c_{m0}, d_{m0}\}_{m=0}^{M}$ of the priors are set to small values in order to non-informative. The number of relevance vector (circles) found by the algorithm is 10 for a threshold set to 10^{-3}. (b) shows the evolution of $\{\bar{\alpha}_m\}_{m=0}^{M}$ as a function of the number of training iterations. Most of them are driven to infinity during the training process.

could think of an additional level of inference. Indeed, making the dependency on the precision λ explicit, we have

$$p(\lambda|\mathbf{t}) \propto p(\mathbf{t}|\lambda)p(\lambda) \ . \tag{4.58}$$

The first factor on the right-hand-side of this equation is the marginal likelihood or evidence of λ. This quantity is defined as follows:

$$p(\mathbf{t}|\lambda) = \int\int p(\mathbf{t}|\boldsymbol{\alpha}, \tau, \lambda)p(\boldsymbol{\alpha}, \tau|\mathbf{t}, \lambda)d\boldsymbol{\alpha}d\tau \ . \tag{4.59}$$

If we assume a flat prior $p(\lambda)$, the value of the precision λ can be selected as the one that maximizes $p(\mathbf{t}|\lambda)$. Unfortunately, this requires to approximate the integrals in (4.59), which are intractable. This approach was already used in Bayesian support vector regression (Law and Kwok, 2001), which is a Bayesian version of support vector machines. A similar technique was also applied by MacKay (1992b) for selecting the number of hidden units in multi-layer perceptrons. Here, we propose to rather use the variational lower bound for selecting the optimal precision.

Let us denote the variational lower bound, which is given in (4.44), by $\mathcal{F}(q_{\mathbf{w}}(\mathbf{w}), q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}), q_{\tau}(\tau))$. Since we know the functional form of the variational

posteriors, the bound can be computed:

$$\begin{aligned}\mathcal{F}(q_{\mathbf{w}}(\mathbf{w}), q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}), q_\tau(\tau)) = &\iint q_{\mathbf{w}}(\mathbf{w}) q_\tau(\tau) \log p(\mathbf{t}|\mathbf{w}, \tau) d\mathbf{w} d\tau \\ &+ \iint q_{\mathbf{w}}(\mathbf{w}) q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \log p(\mathbf{w}|\boldsymbol{\alpha}) d\mathbf{w} d\boldsymbol{\alpha} \\ &+ \int q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \log p(\boldsymbol{\alpha}) d\boldsymbol{\alpha} \\ &+ \int q_\tau(\tau) \log p(\tau) d\tau \\ &- \int q_{\mathbf{w}}(\mathbf{w}) \log q_{\mathbf{w}}(\mathbf{w}) d\mathbf{w} \\ &- \int q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \log q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) d\boldsymbol{\alpha} \\ &- \int q_\tau(\tau) \log q_\tau(\tau) d\tau \ , \end{aligned} \tag{4.60}$$

where the individual terms are given by

$$\begin{aligned}&\iint q_{\mathbf{w}}(\mathbf{w}) q_\tau(\tau) \log p(\mathbf{t}|\mathbf{w}, \tau) d\mathbf{w} d\tau \\ &\qquad = -\frac{N}{2} \log 2\pi + \frac{N}{2} \log \tilde{\tau} - \frac{\bar{\tau}}{2} \left(\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu})\|^2 + \mathrm{tr}\{\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\} \right) \ , \end{aligned} \tag{4.61}$$

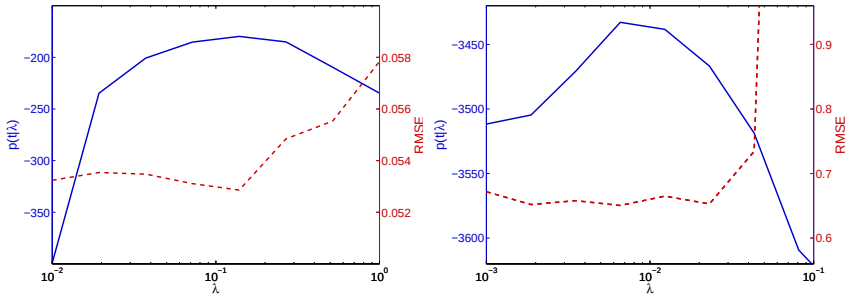
$$\begin{aligned}&\iint q_{\mathbf{w}}(\mathbf{w}) q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \log p(\mathbf{w}|\boldsymbol{\alpha}) d\mathbf{w} d\boldsymbol{\alpha} \\ &\qquad = -\frac{M+1}{2} \log 2\pi + \frac{1}{2} \sum_{m=0}^{M} \{\log \tilde{\alpha}_m - \bar{\alpha}_m \left({\mu_m}^2 + \Sigma_{mm} \right)\} \ , \end{aligned} \tag{4.62}$$

$$\begin{aligned}&\int q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \log p(\boldsymbol{\alpha}) d\boldsymbol{\alpha} \\ &\qquad = \sum_{m=0}^{M} \{c_{m0} \log d_{m0} - \log \Gamma(c_{m0}) + (c_{m0} - 1) \log \tilde{\alpha}_m - d_{m0} \bar{\alpha}_m\} \ , \end{aligned} \tag{4.63}$$

$$\begin{aligned}&\int q_\tau(\tau) \log p(\tau) d\tau \\ &\qquad = a_0 \log b_0 - \log \Gamma(a_0) + (a_0 - 1) \log \tilde{\tau} - b_0 \bar{\tau} \ , \end{aligned} \tag{4.64}$$

$$\begin{aligned}&\int q_{\mathbf{w}}(\mathbf{w}) \log q_{\mathbf{w}}(\mathbf{w}) d\mathbf{w} \\ &\qquad = -\frac{M+1}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}| - (M+1) \ , \end{aligned} \tag{4.65}$$

$$\begin{aligned}&\int q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \log q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) d\boldsymbol{\alpha} \\ &\qquad = \sum_{m=0}^{M} \{c_m \log d_m - \log \Gamma(c_m) + (c_m - 1) \log \tilde{\alpha}_m - d_m \bar{\alpha}_m\} \ , \end{aligned} \tag{4.66}$$



(a) Sinc function. (b) Abalone.

FIGURE 4.10. Variational lower bound (solid) and root mean square error estimated by 10-fold cross-validation (dashed) versus the kernel precision for (a) the sinc function and (b) the Abalone data. The results are averaged over 10 runs. For the sinc function, random data sets are generated, while for the Abalone data, 10 random splits are performed (2,133 learning and 1,044 test data).

$$\int q_\tau(\tau) \log q_\tau(\tau) d\tau \\ = a \log b - \log \Gamma(a) + (a-1) \log \tilde{\tau} - b\bar{\tau} \; . \tag{4.67}$$

The special quantities in these equations are $\log \tilde{\tau} \equiv \mathrm{E}_\tau\{\log \tau\} = \psi(a) - \log(b)$ and $\log \tilde{\alpha} \equiv \mathrm{E}_{\boldsymbol{\alpha}}\{\log \alpha_m\} = \psi(c_m) - \log(d_m)$.

Figure 4.10 illustrates the approach on the sinc function and the Abalone data[2]. In both cases, the variational bound is correlated to the estimated root mean square error and the optimal values for the precision of the basis functions are in agreement. The results obtained for the sinc function are comparable to the ones obtained by Law and Kwok (2001). The Abalone data is normalized component-wise. The gender, which is encoded by $\{m, i, f\}$ and stands for male, infant and female, is respectively mapped onto $\{(100), (010), (001)\}$. The objective is to predict the age of the abalone from physical measurements. The average number of relevance vectors is 13.3. In order to reduce the computational effort, the training set is first reduced by vector quantization to 150 points. The results obtained here (MSE=0.423 ± 0.008) are slightly better than the ones reported in a recent study of Bayesian support vector regression (Chu, Keerthi and Ong, 2004, MSE=0.441 ± 0.021 or MSE=0.428 ± 0.022 depending on the method).

The attractive property of the variational Bayesian approach is that computationally intensive resampling techniques are not needed. We should however be aware that minimizing the prediction error does not necessarily correspond

[2]The Abalone data is available form the UCI Machine Learning repository: http://www.ics.uci.edu/~mlearn

to maximizing the evidence (MacKay, 1992b; Bishop, 1995). For instance, the Bayesian approach seeks for the most probable model among a particular family of models given the data. This implicitly assumes that the true model is within this model family, which may be a false assumption. When the models are poorly matched, ranking them according to their evidence may be misleading. The test error, by contrast, is evaluated on a finite data set and is thus a noisy quantity.

4.2.5. Related Approach

As discussed in Section 4.2.3, the learning algorithm for the ordinary RVM has an EM formulation. By viewing the parameter vector $\mathbf{w}$ as unobserved, the incomplete log-posterior $p(\boldsymbol{\alpha}, \tau|\mathbf{t}) \propto p(\mathbf{t}|\boldsymbol{\alpha}, \tau)p(\boldsymbol{\alpha})p(\tau)$ can be maximized iteratively by the EM algorithm. Recently, Figueiredo (2003) proposed an alternative supervised learning algorithm which induces sparsity. Under the same model assumption, i.e. Gaussian noise on the targets, a Laplacian prior is imposed on the parameters $\mathbf{w}$ instead of a Gaussian one and the hyperparemeter vector $\boldsymbol{\alpha}$ is considered as being a hidden variable. As shown below, the incomplete log-posterior $p(\mathbf{w}, \tau|\mathbf{t})$ can then be maximized by the EM algorithm.

The zero-mean Laplacian prior induces sparsity. This was already mentioned in our discussion of the RVM. By adopting a hierarchical Bayesian approach, a Laplacian prior on $\mathbf{w}$ can be obtained. Consider a Gaussian prior $\mathcal{N}(w_m|0, \alpha_m)$ on each parameter as in RVM and let us denote each variance $1/\alpha_m$ by β_m, such that $p(w_m|\beta_m) = \mathcal{N}(w_m|0, {\beta_m}^{-1})$. Now, we impose a zero-mean exponential hyperprior on β_m:

$$p(\beta_m|c_m) = \mathcal{E}(\beta_m|0, c_m) = c_m \exp(-c_m\beta_m) \ , \tag{4.68}$$

with $c_m \geq 0$. Integrating out β_m yields a zero-mean Laplacian prior on the parameters:

$$p(w_m|c_m) = \int_0^\infty p(w_m|\beta_m)p(\beta_m|c_m)d\beta_m = \frac{\sqrt{2c_m}}{2} \exp(-\sqrt{2c_m}|w_m|) \ . \tag{4.69}$$

Next, regarding the set of variances $\boldsymbol{\beta} = \{\beta_m\}_{m=0}^M$ as unobserved, we want to maximize the log-posterior $p(\mathbf{w}, \tau|\mathbf{t})$. Therefore, the E-step consists in computing the posterior distribution of the latent variables:

$$p(\beta_m|\mathbf{t}, w_m, \tau) = p(\beta_m|w_m) = \frac{p(w_m|\beta_m)p(\beta_m|c_m)}{p(w_m|c_m)} \ , \tag{4.70}$$

where in the first equality we have used the fact that β_m is independent of $\mathbf{t}$ and τ given w_m. Subsequently, the expected complete log-posterior is maximized with respect to $\mathbf{w}$ and τ. The complete log-posterior is given by

$$\log p(\mathbf{t}|\boldsymbol{\beta}, \mathbf{w}, \tau)p(\boldsymbol{\beta}|\mathbf{w})p(\mathbf{w})p(\tau) = \log p(\mathbf{t}|\mathbf{w}, \tau)p(\mathbf{w}|\boldsymbol{\beta})p(\boldsymbol{\beta})p(\tau) \ , \tag{4.71}$$

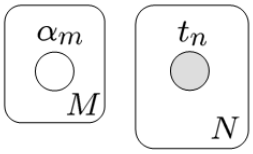


FIGURE 4.11. Graphical model of Figueiredo's (2003) sparse approximator. Note that $\alpha_m = 1/\beta_m$, $\forall m$. The variance vector $\boldsymbol{\beta}$, which is unobserved (thus neither $\boldsymbol{\alpha}$), is independent of $\mathbf{t}$ given $\mathbf{w}$. The parameter vector $\mathbf{w}$ is a deterministic quantity and therefore does not appear in the graph.

where we use the fact that $\mathbf{t}$ is independent of $\boldsymbol{\beta}$ given $\mathbf{w}$ and

$$p(\mathbf{w}|\boldsymbol{\beta}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A}) = \prod_{m=0}^{M} p(w_m|\beta_m) \ , \tag{4.72}$$

$$p(\boldsymbol{\beta}) = \prod_{m=0}^{M} p(\beta_m|c_m) \ . \tag{4.73}$$

Matrix $\mathbf{A}$ is equal to $\text{diag}\{{\beta_0}^{-1}, \ldots, {\beta_M}^{-1}\}$. Since $\bar{\beta}_m \equiv \text{E}_{\boldsymbol{\beta}}\{\beta_m\} = (1/2c_m + |w_m|/\sqrt{2c_m})$ and $\bar{\alpha}_m \equiv \text{E}_{\boldsymbol{\beta}}\{{\beta_m}^{-1}\} = \sqrt{2c_m}/|w_m|$, taking expectations and then maximizing (4.71) w.r.t. the parameters $\mathbf{w}$ and the noise precision τ results in the following M-step:

$$\mathbf{w} \leftarrow \tau \left(\tau \boldsymbol{\Phi}^{\text{T}} \boldsymbol{\Phi} + \bar{\mathbf{A}}\right)^{-1} \boldsymbol{\Phi}^{\text{T}} \mathbf{t} \ , \tag{4.74}$$

$$\tau \leftarrow \left\{ \frac{\|\mathbf{t} - \boldsymbol{\Phi}\mathbf{w}\|^2 + 2b_0}{N + 2(a_0 - 1)} \right\}^{-1} \ . \tag{4.75}$$

Following this approach leads thus to the same update rules as in MAP learning, except for $\mathbf{A}$, which is replaced by $\bar{\mathbf{A}} = \text{diag}\{\bar{\alpha}_0, \ldots, \bar{\alpha}_M\}$. The graphical model associated to this formulation of the sparse approximator is shown in Figure 4.11.

In order to get rid of the hyperparameters $\{c_m\}_{m=0}^{M}$, which control the degree of sparseness, Jeffrey's non-informative prior can be used instead of the exponential one:

$$p(\beta_m) \propto \frac{1}{\beta_m} \ , \quad \forall m \ . \tag{4.76}$$

The resulting marginal distribution for $\{w_m\}_{m=0}^{M}$ is no longer the Laplacian prior, but is equal to the following (improper) distribution:

$$p(w_m) = \frac{1}{|w_m|} \ , \quad \forall m. \tag{4.77}$$

This prior strongly induces sparseness (Figueiredo, 2003). In fact, as Jeffrey's prior is a limiting case of the Gamma distribution, the resulting prior on the parameters is a limiting case of the zero-mean Student-t distribution, which in turn is sharply peaked around zero like the Laplacian prior. This is illustrated

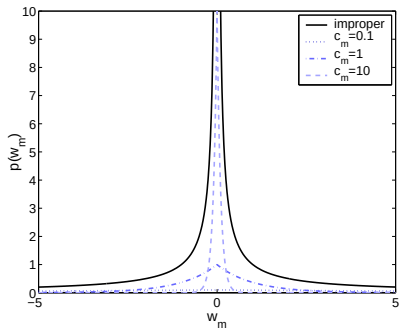


FIGURE 4.12. Comparison of the improper prior $p(w_m)$ (solid), which is obtained by integrating out the nuisance parameter β_m, with the corresponding Laplacian priors for different values of the hyperparameter c_m. The improper prior is sharply peaked around zero and it has very heavy distribution tails.

in Figure 4.12. Using the improper prior leads to the same M-step, except that now $\bar{\alpha}_m \equiv \mathrm{E}_{\boldsymbol{\beta}}\{{\beta_m}^{-1}\} = 1/|w_m|^2$. However, since most of the parameters are driven to zero during learning, it is convenient to use the following equivalent M-step for $\mathbf{w}$ in practice:

$$\mathbf{w} \leftarrow \tau \mathbf{W} \left(\tau \mathbf{W} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} \mathbf{W} + \mathbf{I} \right)^{-1} \mathbf{W} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \ , \tag{4.78}$$

where $\mathbf{W} = \mathrm{diag}\{|w_0|, \ldots, |w_M|\}$. This avoids having to deal with arbitrarily large numbers and allows solving the corresponding linear system by singular value decomposition.

Figure 4.13 shows the sparse approximator for the sinc function. In general, the training algorithm leads to very sparse solutions, which are expected to exhibit very good generalization abilities. In addition, unlike most other techniques, the method does not require to set additional parameters, such as regularization constants or a threshold. Nevertheless, the algorithm is sensitive to the initialization of $\mathbf{w}$ and the problem of choosing the precision of the basis functions remains. As usual, the value of the precision may have a significant influence on the quality of the approximator and needs to be optimized by resampling techniques. Finally, as the algorithm provides point estimates of $\mathbf{w}$ and τ, no local error bars can be constructed, which is of course a drawback compared to the RVM.

4.3. Summary

In this chapter, we put regularization networks for regression into a probabilistic perspective. The link with the standard radial basis function network was highlighted and a vector quantization-based variant was introduced. The

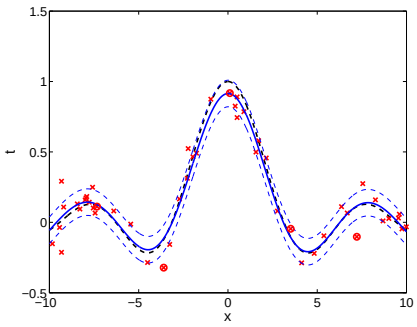


FIGURE 4.13. Figueiredo's (2003) sparse approximator (solid) obtained for the sinc function (dashed). The true noise standard deviation (0.1) is only slightly underestimated (0.098). The noise tube corresponds to one standard deviation. The number of training data (crosses) is 50. The precision of the kernels is set to 1/6. The number of relevance vector (circles) found by the algorithm is 5.

latter adjusts locally the precision of the basis functions, while the amount of smoothing is controlled by a common width scaling factor. The core of the chapter discusses several probabilistic approaches to regularization networks. In particular, the Bayesian approaches are attractive as they allow us to obtain very sparse solutions, which are expected to generalize well. Finally, we showed that the variational framework is especially appealing as it allows to determine the precision of the basis functions based on the variational lower bound.

CHAPTER 5

Probabilistic Models of the Electrical Stimulation of the Human Optic Nerve

Since the early eighties cochlear implants are an active field of research in biomedical engineering, mainly to rehabilitate patients with hearing loss for whom there is no other potential treatment. In recent years, most patients with modern cochlear implant systems can understand speech using the device alone, at least in favorable listening conditions. For example, these implants allow deaf patients to hear and even talk over the phone (Clark, McAnally, Black and Shepherd, 1995; Cray, Allen, Stuart, Hudson, Layman and Givens, 2004). The hearing quality reached by the existing devices justifies their use in less severely affected patients and is even advocated as a treatment to prevent language deficit in pre-lingual deaf children (Gstoettner, Hamzavi, Egelierler and Baumgartner, 2000). Currently, an increasing research effort has also been directed towards implant users' perception of nonspeech sounds, especially music (see McDermott (2004) and references therein).

Further to this success, several multidisciplinary teams were established during the past decade with the goal to restore partial vision to the blind. The human visual system, which extracts relevant visual information from images of the environment that are projected on the retina, is a far more complex information processing system than the auditory system. Despite promising results in animal experiments, there are still several major obstacles to overcome before visual prostheses can be used clinically (Zrenner, 2002).

Blindness can result from damage to any processing step in the visual pathways. First, the retina is a thin layer of cells at the back of the eyeball, which converts light into nervous signals. The light enters the eye through the pupil and is focused by the lens on the retina (see Figure 5.1). Millions of photoreceptor cells, called rods and cones, are excited by the local luminance and color. The cones respond to bright light and mediate high-resolution and color vision. The rods respond to dim light and mediate lower-resolution, black-and-white night vision. Both these photoreceptors transform the visual information into electrical and chemical signals and activate the retinal neurons: horizontal, bipolar, amacrine and ganglion cells (see Figure 5.2). Second, after compression of the visual information, the corresponding electrical signals are carried by the optic nerve, which bundles the axons of the ganglion cells (and few amacrine cells). So far, the compression mechanism is not fully understood.

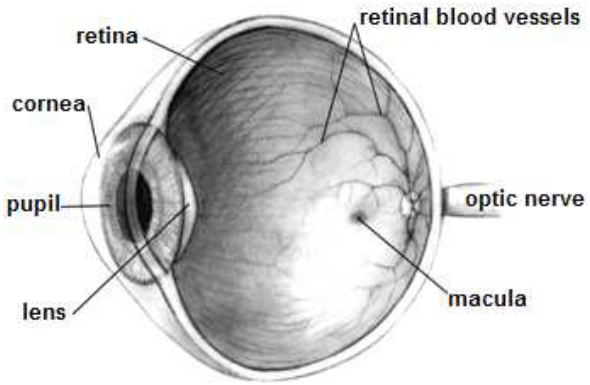
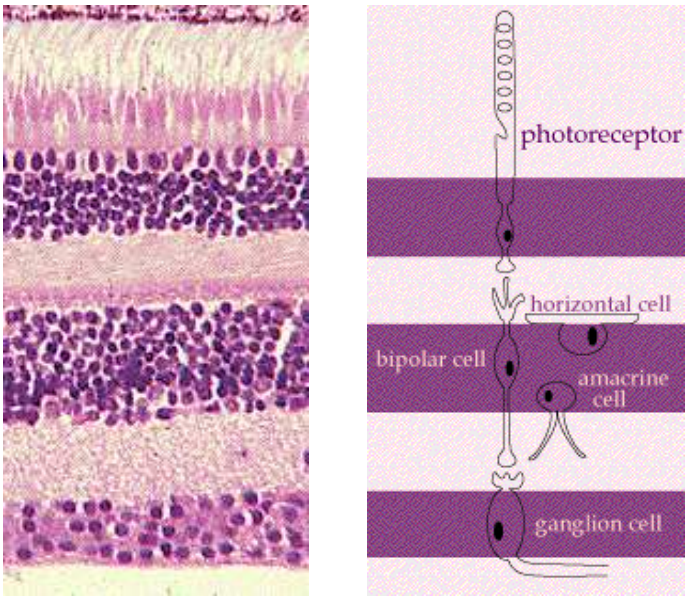


FIGURE 5.1. Schematic cross-section of the human eye. Light enters the eye through the pupil and is projected on the retina. The eyeball is filled with the vitreous fluid. (Modified from the website of the U.S. National Eye Institute: http://www.nei.nih.gov/health/macularhole)

However, one should realize that the amount of compression is enormous since the number of photoreceptors is roughly 100 million, while there are only approximatively 1 million axons (Meister and Berry, 1999). Third, the visual information is transmitted to the brain (primary visual cortex) via the lateral geniculate nucleus.

From the basic physiology of the visual pathways, artificial vision can be envisioned based on the following facts: (i) most causes of blindness do not lead to a destruction of the entire visual system, (ii) electrical stimuli (electrons) can be substituted to light stimuli (photons) to create visual perception, and (iii) the retinotopy, i.e. the spatial organization of the visual information along the visual system, tells us how to arrange electrical stimulations to produce rational visual sensations. At present, this means that a low-resolution artificial vision can be expected after extensive training. It is thus important to realize that it would be unreasonable to expect from visual prostheses to fully restore vision. However, it is hoped that they will help the blind patient with simple tasks such as object recognition, spatial localization, obstacle avoidance and that they will improve the quality of his/her everyday life. This very last point is worth some caution. Indeed, the acceptability and attractiveness of visual prostheses must not be taken for granted. As a matter of fact, some profoundly blind people have developed excellent strategies for coping with their condition and may not look with favor on a prosthesis unless it is quite safe, readily affordable and provides a useful visual sense.

The current visual prostheses are based on the neuronal electrical stimulation at different locations along the visual pathways, within the central nervous



(a) Retina. (b) Schematic view.

FIGURE 5.2. Axial organization of (a) the retina and (b) its schematic view. The light comes from below. The top layer (outer retina) contains the rods and the cones. The axons of the ganglion cells (bottom layer of inner retina) come together in the optic nerve. (Modified from the online neuroscience tutorial of the Washington University School of Medicine: `http://thalamus.wustl.edu/course`)

system. According to this location, the different prostheses are called cortical, optic nerve or retinal ones. For a comprehensive introduction to visual prostheses, we refer to recent review papers by Maynard (2001) and by Margalit, Maia, Weiland, Greenberg, Fujii, Torres, Piyathaisere, O'Hearn, Liu, Lazzi, Dagnelie, Scribner, de Juan and Humayun (2002). The general principle consists in implanting a neural prosthesis, either intracranially or intraoccularly, and bypass neurons that have become non-functional by electrical stimulation. The very first attempt to create a light perception in the visual field comparable to the perception resulting from a light stimulus was made by Foerster (1929). He noted that the electrical stimulation of the visual cortex caused his subject to see a spot light. This electrically induced visual sensation is termed phosphene. It is convenient to view phosphenes as pixels, since they are usually perceived as white, round or oval points of light, which can have different sizes and show short persistence. In recent experiments, other shapes and colors were also reported (Veraart, Raftopoulos, Mortimer, Delbeke, Pins, Michaux, Vanlierde, Parrini and Wanet-Defalque, 1998). Foerster's work already demonstrates that

a small area of the neuronal tissue can be stimulated in order to get a light perception (Foerster, 1929); however, this perception is not comparable to the light stimulation the sighted know. As a result, the following questions arise:

(1) Can we reconstruct a whole visual scene by stimulating many small areas of the neuronal tissue, i.e. activating many pixels (or phosphenes)?
(2) How many pixels should we use and how should we combine them?
(3) What are the electrical stimulus parameters (amplitude, duration, shape, etc.) needed for each pixel to make it safe and effective?

These questions are critical, as little is known about the coding scheme of the visual information along the visual pathways. The present knowledge of the visual system remains limited and only crude models of the bypassed parts can be considered. Induction of visual perception using these models remains therefore questionable and this is even more the case when a substantial part of the visual system is being bypassed. As a consequence, we will use non-linear statistical tools instead of crude neurophysiological models and try to answer the questions mentioned above, at least partially. We will make extensive use of the probabilistic techniques discussed in the previous chapters.

In this chapter, we first review the different types of visual prostheses that are currently under development and focus on the optic nerve visual prosthesis. Next, we present both neurophysiological and probabilistic models predicting the characteristics of the visual perceptions based on the parameters of the electrical stimuli. Subsequently, we describe techniques for classifying these perceptions based on their location in the visual field. Finally, these building blocks are put together and an efficient stimulation strategy is proposed.

5.1. Visual Prostheses

To date three types of visual implants exist: cortical, retinal and optic nerve implants. Each one of them is discussed below.

5.1.1. Cortical Prosthesis

Cortically-based prosthetic vision is based upon the retinotopic organization of the visual neural system. The concept of retinotopy says that neighboring cells in the retina transmit information to (more or less) neighboring cells in the visual cortex, meaning that the retinal output is mapped directly onto the visual cortex. However, due to the nonuniform distribution of photoreceptors across the retina, magnification occurs. In other words, the central part of the visual field is represented to a far greater extent in the cortex than in the peripheral retina (Hubel and Wiesel, 1974; Horton and Hoyt, 1991).

The earliest visual implants, which used surface cortical electrodes, are due to Brindley and Lewin (1968) and were further studied by Dobelle and Mladejovsky (1974). The experiments showed that chronical electrical stimulation

was possible, that phosphenes are stable over time, that repeated stimulation of the same location of the visual cortex leads to a phosphene at the same location in the visual field and that the amount of current required to obtain a light perception is also fairly stable. Moreover, stimulating several points on the cortex caused the subject to see a set of phosphenes. However, as these early experiments included interactions between phosphenes and inconsistency of phosphenes, as well the use of high currents, the development of intracortical, i.e. penetrating, electrodes came about (Bak, Girvin, Hambrecht, Kufta, Loeb and Schmidt, 1990; Schmidt, Bak, Hambrecht, Kufta, ORourke and Vallabhanath, 1996; Normann, Maynard, Guillory and Warren, 1996). These electrodes are much smaller and close to the target neurons. As a result, the current thresholds are lower and localized stimulations are possible. Schmidt et al. (1996) produced visual perceptions in a 42-year-old woman, who had been totally blind for 22 years secondary to glaucoma. It was reported that the brightness of the phosphenes could be modified by adjusting the amplitude, the frequency and pulse duration of the electrical pulses. Usually, the phosphenes did not flicker. Near stimulation threshold, the phosphenes were often reported to have colors. The duration of the perception could be increased by interrupting a long stimulation train with brief pauses in stimulation. In addition, intracortical microelectrodes spaced 500μm apart generated separate phosphenes, while microelectrodes spaced 250μm did not. Finally, most of the phosphenes were located within a relatively small area of visual space.

Recent studies demonstrate that the traditional view of retinotopy is only valid at a very coarse level. The relationship between the photoreceptors and the corresponding locations on the visual cortex is extremely complex, i.e. highly nonlinear and non-conformal (Warren, Fernandez and Normann, 2001). This result has implications for the design of cortical prostheses, as it requires to remap the visual space to accommodate the scatter in the phosphene locations. In addition, individual neurons encode many specific features of the visual image. For example, cortical neurons respond best to particular colors and shapes, manifest eye-related dominance and may be sensitive to particular orientations due to their receptive fields. Finally, serious drawbacks to cortical implants are the surgical risks of an intracranial procedure. In particular, surgical complications can have devastating results, including death, on healthy subjects.

5.1.2. Retinal Prostheses

The main advantage of cortical prostheses is that they are able to treat blindness secondary to retinal or optic nerve diseases. Nevertheless, the approach needs to deal with the complex geometry of the brain and requires to perform an intracranial surgical procedure with high risks. By contrast, ocular prostheses avoid these risks, but they can only be applied in cases where the optic nerve is still functional and would thus not be helpful in diseases such as glaucoma.

In the industrial countries, the leading cause of inherited blindness is retinitis pigmentosa (RP). About 1.5 million people are affected by RP worldwide (Margalit et al., 2002). Another common cause of visual loss in the western countries is age-related macular degeneration (AMD), which is the most common form of blindness in the elderly. Both diseases are due to a degeneration of the outer retina, i.e. the photoreceptors die off. This means that the capacity of the retina to transduce light into biologic signals is diminished. Morphometric analyzes (Stone, Barlow, Humayun, de Juan and Milam, 1992; Santos, Humayun, de Juan, Greenberg, Marsh, Klock and Milam, 1997; Kim, Sadda, Pearlman, Humayun, de Juan E, Melia and Green, 2002) showed however that the inner retinal layers are still functional and can be stimulated electrically. Therefore, a viable alternative to cortical implants are retinal ones.

An implantation at the level of the retina has the advantage to benefit from the fine retinotopic organization of the retina. More importantly, it allows exploiting the natural processing of the rest of the visual pathways. Retinal prostheses stimulate the inner retina. They are either subretinal (Chow and Chow, 1997; Zrenner, Stett, Weiss, Aramant, Guenther, Kohler, Miliczek, Seiler and Haemmerle, 1999; Zrenner, 2002) or epiretinal (Humayun, Propst, de Juan E, McCormick and Hickingbotham, 1994; Humayun, de Juan, Dagnelie, Greenberg, Prost and Phillips, 1996; Wyatt and Rizzo, 1996; Eckmiller, 1997; Rizzo and Wyatt, 1997), according to the location of fixation of the stimulation array.

In the first approach, high-density microphotodiodes arrays are implanted behind the retina in order to replace the lost photosensitive cells, i.e. the outer retina, by an artificial one. The adjacent retinal neurons are then stimulated through multi-site injection of photocurrents generated by locally absorbed light (Zrenner et al., 1999). In this approach, the optics of the eye need to be intact. Although it was demonstrated that the retinal neurons could be electrically stimulated using this method, the required retinal illuminance (between 10 and 100kLux) to stimulate the inner retina is far above the ones naturally occurring (approximately 8Lux). Therefore, it is likely that active electronics, and thus an external power supply is needed. Furthermore, histological evaluation has shown that there is an ongoing degenerative process of the inner retina under the device. So far, it is not fully understood why this occurs, but according to Zrenner et al. (1999), this may be due to the fact that the microphotodiode array is flat, rigid and not perforated. As a result, the inner retina could be mechanically damaged and the transport of nutrients to it could be reduced.

In the second approach, the stimulating device is placed intraocularly on the inner retina, while most of the electronics is located off the retinal surface in the vitreous body. Since this is a fluid filled cavity, it helps dissipating heat. The implanted retinal microchip receives information from outside the body via a telemetry link (Liu, Vichienchom, Clements, DeMarco, Hughes, McGucken, MS, de Juan, Weiland and Greenberg, 2000). The carrier signal is either radio frequency or laser modulated. The external part of the system consists in a

camera and an electronic image-processing unit. Experiments demonstrated that the electrical stimulation of the retinal surface elicits visual sensations in blind individuals (Humayun, de Juan, Weiland, Dagnelie, Katona, Greenberg and Suzuki, 1999) and that the location of the visual perceptions is directly related to the retinal area that is stimulated. In addition, it was reported that the stimulation threshold depends on the targeted retinal area of the subjects. Because of the rotational ocular movements, the main disadvantage of epiretinal prostheses is the way the stimulation array is fixed, such that it remains in place for a prolonged period of time without damaging the retina. Another concern is the viability of the tissues under the implant (Maynard, 2001) and possible activation of unwanted axons, which passes nearby the activation sites.

5.1.3. The Optic Nerve Visual Prosthesis

In case of outer retina pathologies such as RP and AMD, the electrical stimulation of the peripheral visual system can be considered at two different locations: the retina, as described in the previous section, or the optic nerve (Veraart et al., 1998). As noted by Maynard (2001), one issue when electrically stimulating the retina, as well as the visual cortex, is that the visual field is represented over a relatively large area, making coverage of the entire visual field nearly impossible with current electrode array technologies. By contrast, the optic nerve is one place where the entire visual field is represented in a relatively small area. Unfortunately, as with cortical implants, there we have to deal with the fact that the retinotopic organization of the human optic nerve is possibly not fully respected (Ding and Marotte, 1997). Furthermore, it might be difficult to achieve focal stimulation, and therefore detailed perception.

The optic nerve contains roughly one million fibres, which are clustered into bundles and are surrounded by connective tissue. It can be accessed either intracranially, near the optic chiasm, or directly behind the eye, via the eye cavity after having carefully removed the eye. After dissection of the dura, a spiral cuff electrode, as the ones used in neuromuscular stimulation (Naples, Mortimer, Scheiner and Sweeney, 1988; Veraart, Grill and Mortimer, 1993), can be wrapped around the optic nerve. In fact, by using a multi-contact electrode, subsets of axons can be stimulated selectively by complex patterns of electrical stimulation (Parrini, Delbeke, Legat and Veraart, 2000).

The MIVIP (*mi*crosytems based *vi*sual *p*rosthesis) and OPTIVIP (*opti*mization of the *v*isual *i*mplantable *p*rosthesis) projects funded by the European Commission aim to investigate the feasibility and the prospects of an optic nerve based visual prosthesis. A 59-year old female was selected among six totally blind candidates in order to assess the electrical excitability of the visual pathways and the viability of the optic nerve by using surface electrodes (Delbeke, Pins, Michaux, Wanet-Defalque, Parrini and Veraart, 2001). The volunteer,

who gave her informed consent[1], has been chronically implanted with a self-sizing spiral cuff electrode around her right optic nerve in February 1998. The blind patient was affected by RP. She was left with a mere light perception at the age of 40 and diagnosed totally blind at 57. It was demonstrated experimentally that the optic nerve can be safely interfaced with a four-contact electrode and that the elicited phosphenes, of various shapes and colors, were broadly distributed in the visual field (Veraart et al., 1998). After training, the patient could recognize different shapes, line orientations, and even letters (Veraart, Wanet-Delfalque, Gerard, Vanlierde and Delbeke, 2003). Furthermore, no acute or chronic side effect was noted.

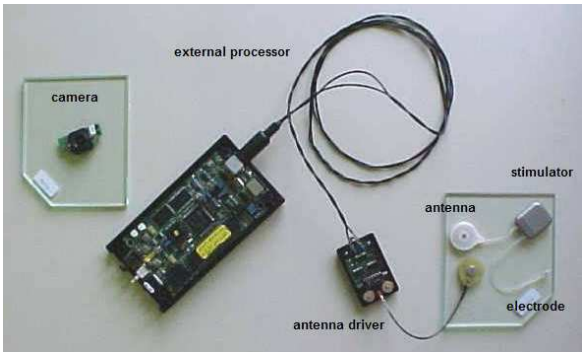
Prosthetic device

The three levels of hierarchy in the sensory systems, i.e. receptor organ, sensory pathways and perception, suggest a similar architecture for artificial and prosthetic sensory systems. Accordingly, complete artificial systems should include a transducer corresponding to the receptor organ, an encoder corresponding to the sensory processing system, and an interpreter corresponding to perceptual functions (Margalit et al., 2002). In the case of the optic nerve prosthesis, the transducer is bypassed, as well as a substantial part of the encoder. Therefore, the prosthetic device should compensate for this such that the electrical signals arriving in the visual cortex are meaningful.

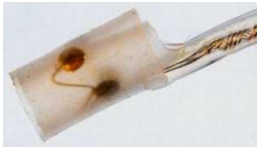
Figure 5.3 shows a picture of the microelectronic prototype of the prosthesis (Doguet, Mevel, Verleysen, Troosters and Trullemans, 2000). Images of the environment are captured by means of a small camera, which is fixed on the branch of a pair of glasses. These images are processed by the portable external processor, which extracts the relevant information and encodes it into a restricted data stream, which in turn is sent to the implanted stimulator. The transcutaneous antenna and its driver are used for telemetry and power supply. Of course, the amount of power needed for the implant should be kept as small as possible. The antenna is an inductive link and therefore avoids wires through the skin, which may be a source of infection. The stimulator decodes the data streams of 3Mbit/s and transforms them into adequate electrical stimuli to be applied to the optic nerve via the self-sizing cuff electrode.

Figure 5.4 shows X-rays of the head to the blind volunteer after implantation. The stimulator is located behind the right ear. As it is chronically implanted, biocompatibility of the implanted material and hermetic sealing from the corrosive biological fluid are of major concern. In particular, the connectors are the most vulnerable leakage points of the system. The stimulator is connected on one side to the electrode and on the other side to the secondary coil of the antenna. Both coils are kept in place by means of a little magnet.

[1]Both MIVIP and OPTIVIP projects fully comply with the declaration of Helsinki and have been approved by the ethical committee (comité hospitalo-facultaire) of St-Luc University Hospital, Université catholique de Louvain, Brussels, Belgium.



(a) Prototype of the prosthesis.



(b) Spiral cuff electrode.

FIGURE 5.3. (a) shows the prototype of the optic nerve visual prosthesis. The camera, for example fixed on spectacles, captures an image of the environment and sends it to the portable external signal processor. The processor extracts the information to be transmitted to the stimulator via the inductive link (antenna). Finally, the stimulator sends electrical pulses to the four-contact self-sizing cuff electrode wrapped around the optic nerve of the blind volunteer. (b) is a zoom of the electrode.

Electrical stimulation principle

The visual information captured by the camera has to be translated by the external processor into a spatiotemporal stimulation pattern of electrical impulses that can be understood by the brain's visual cortex. More specifically, the stimulation principle relies on the selective response of the human optic nerve to adequately chosen electrical stimuli. In other words, phosphenes with the desired features will be generated by tuning the stimulation parameters.

The electrical stimulation of neurons elicits an electro-chemical response, called action potential, according to an all-or-one mechanism (Lapique, 1907; Hodgkin and Huxley, 1952). This means that neural excitation only occurs when the minimum excitation threshold, depending on the shape, the amplitude and the duration of the electrical stimulus, is exceeded. In fact, it is not directly the amplitude and the duration of the current pulse that matters, but the charge that is injected. For example, if the pulse duration decreases, the threshold

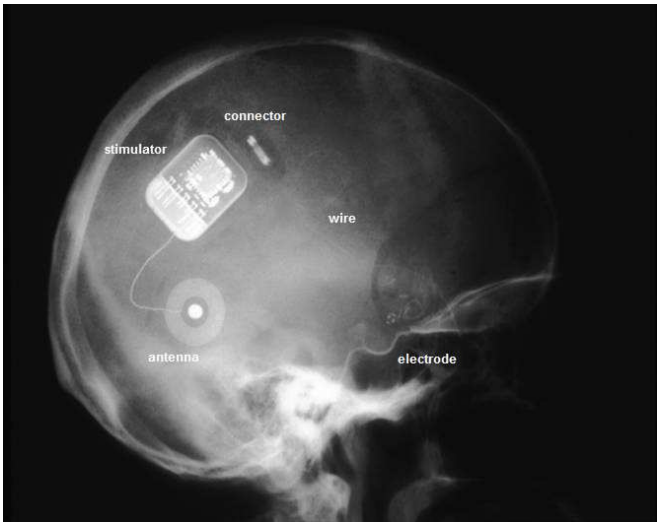


FIGURE 5.4. X-rays of the implanted part of the prosthetic device. One can see the secondary coil of the antenna, the stimulator and the connector of the wires to the electrode. When taking a closer look, the wire running from the stimulator to the electrode can also be observed.

current will increase. The relationship between stimulus amplitude and duration is described by the well-known strength-duration curve (Hill, 1936). An example of this curve is shown in Figure 5.5 and its mathematical form is given by

$$I_{\mathrm{th}}(D) = \frac{I_{\mathrm{r}}}{1 - \exp\left(-\frac{D \log 2}{D_{\mathrm{c}}}\right)} \ . \tag{5.1}$$

In this equation, I_{r} and D_{c} are respectively the rheobase and the chronaxie. The rheobase is the minimum current amplitude required to trigger the neuron when the stimulus is a square pulse of infinite duration. The chronaxie is the minimum duration required to trigger the neuron when the pulse has an amplitude equal to $2I_{\mathrm{r}}$. In order to excite the axons of the optic nerve, one should thus use, for a given stimulation duration, appropriate and safe current amplitudes. It was also reported that the rate of stimulation affects the threshold as well (Bak et al., 1990; Veraart et al., 1998).

The electrode of the prototype has four contacts (see Figure 5.3). Each of them is driven by an independent current source of the stimulator. These current sources send biphasic square pulses, i.e. with charge recovery. For safety reasons, it is important to use balanced electrical stimuli, such that the net charge supply after each stimulation is zero. Therefore, any net DC current, which would lead over time to irreversible electrolyte reactions in the neuronal tissue,

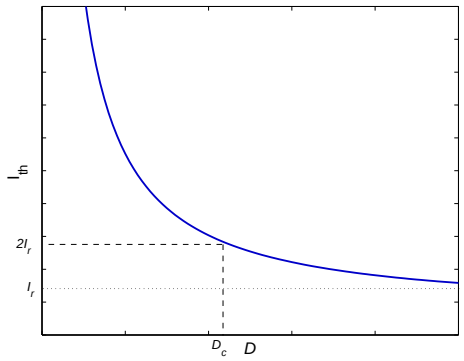


FIGURE 5.5. Example of a strength-duration curve. The rheobase and the chronaxie are respectively I_{r} and D_{c}. For a given pulse duration D, a current amplitude below I_{th} will not trigger the neuron and thus not generate an action potential.

is avoided. The time resolution of the pulses is 21.3μs and the current amplitude ranges from 10μA to 3mA (with a nonlinear resolution). Besides, since higher currents are required to reach threshold with anodic stimulation, the preferred polarity is the cathodic one. In conclusion, the stimulation parameters characterizing the electrical pulse trains and thus controlling the features of the phosphenes are the current amplitude I, the duration of the pulse D, the frequency of the pulse train f and the number of pulses in the train N.

5.2. Prediction of Phosphenes

In order to establish the features of the visual perceptions, a large number of experiments have been conducted with the blind volunteer (Veraart et al., 2003). Before and during stimulation, the subject's head is stabilized in front of a hemispheric surface, her right eye being located at the center. Meridians and parallels are traced on the hemisphere. When ready for stimulus, she is asked to gaze right in front of her, which is important as the apparent location of the perceptions depend on the gaze direction. Of course, she does not know the kind of stimulation she might expect. When single or short current pulses are delivered to the optic nerve, phosphenes light up in the black visual field of the patient. These perceptions are (quasi) instantaneous and comparable to a flash. Right after stimulation, she is asked to point the location where she perceived the phosphene, which is then drawn by an operator on a grid with azimuth and elevation coordinates. Subsequently, the perceived phosphenes are documented in terms of brightness, color, size, texture and motion following the subject's description.

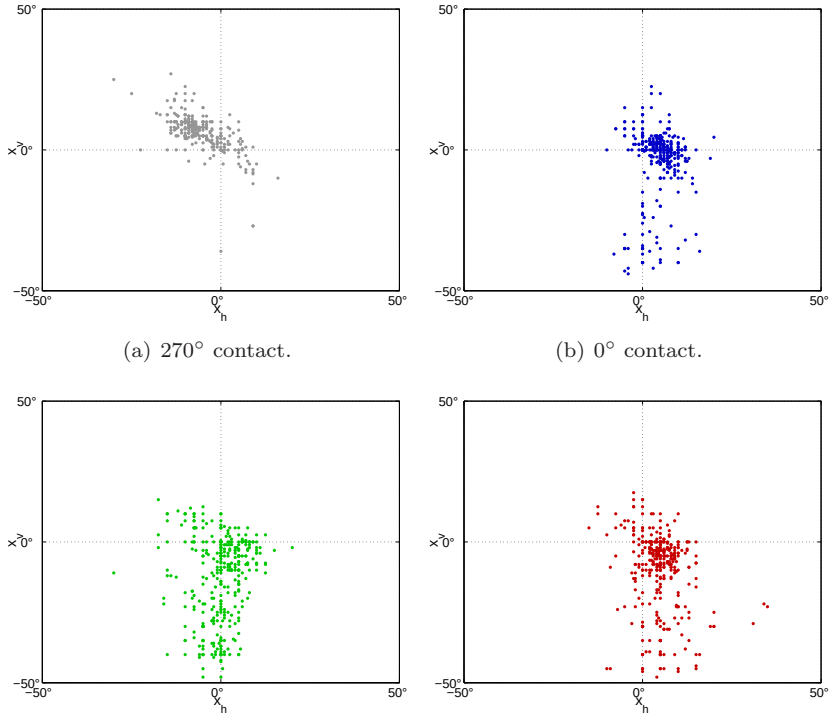
The light perceptions are spatially organized in the volunteer's visual field according to a coarse retinotopic map. This means that each contact around

the optic nerve activates fibres located in a certain area of the optic nerve cross-section, which in turn corresponds to a well-defined area in the visual field of the blind. Figure 5.6 shows the location of the phosphenes in the visual field of the patient. We consider single electrode contact stimulations, meaning that only one contact is activated in each stimulation. The contacts are identified by their angular position around the optic nerve. Although there is some overlap, it can be observed from the figure that the phosphenes are roughly elicited in one quadrant of the visual field, which depends on the electrode contact. We can also see that the region in which the phosphenes are perceived ranges from –30° to +30° horizontally and from –50° to +30° vertically.

The complexity of the neurophysiological process, whereby the electrical pulses applied to the optic nerve generate phosphenes, makes it difficult to study the entire process at a biological level. For instance, some unknown parameters might influence it to a large extent. Furthermore, it must be stressed that the optic nerve of the patient is probably damaged to some unknown degree due to RP. Finally, the characteristics of the phosphenes are a description of subjective perceptions by a human being, leading inevitably to inaccuracies and even errors. As a consequence, the collected data set can be expected to be very noisy. For these reasons, even if partial decoding can be achieved by using neurophysiological knowledge, a mathematical identification of the undecoded part of the process might be very helpful.

One of the main building blocks of a meaningful stimulation strategy are patient-dependent models that can reliably and accurately predict the characteristics of the phosphenes. Obviously, correctly predicting the location of the phosphenes is of utmost importance for reconstructing visual scenes and we will therefore focus exclusively on this problem. By “correctly” is meant that the position of the phosphene is understood in a natural way by the blind. For example, this information can be used to reconstruct contours in the visual field of the subject. Besides, having effective and flexible prediction tools is very helpful for further guiding the data acquisition and better understanding the underlying neurophysiology.

The prediction (or regression) problem is summarized in Figure 5.7. The target is the unknown neurophysiological process which links the stimulation parameters $\{I, D, N, f\}$ to the position (x_h, x_v) of the corresponding phosphene. In the remaining of this chapter, the azimuth x_h and the elevation x_v will be examined separately. Once the neurophysiological process is modeled with a satisfactory accuracy, it may be reversed, for example by means of a simple lookup table implemented in the external processor. In other words, when we want a light perception at a certain location in the visual field, it is sufficient to select the appropriate stimulation parameters in the table (as well as the appropriate electrode contact). Note however that several sets of stimulation parameters may lead to the same (or a very similar) phosphene and would thus be equally suitable. In this case, additional criteria such as reliability or safety can be used to select specific parameters.



(a) 270° contact. (b) 0° contact.

(c) 180° contact. (d) 90° contact.

FIGURE 5.6. Location of the phosphenes recorded during the experiments in the visual field of the blind patient. The posphenes that were elicited by a stimulation strength smaller than -100 or larger than $1,000$ are discarded. For a formal definition of this quantity, we refer the reader to Section 5.2.1. The azimuth and elevation coordinates are denoted by (x_h, x_v). The resolution of the measurements' grid is approximately 1°. In each panel a different electrode contact is activated. The electrode contacts are named by their position around the optic nerve.

Next, three predictive models are presented. The first one, which is due to Delbeke, Oozeer and Veraart (2003) is a neurophysiological model. The second one follows a standard machine learning approach and is therefore purely of the black-box type (Archambeau, Delbeke, Veraart and Verleysen, 2004). Finally, a hybrid predictive model is proposed. This model tries to combine the advantages of both previous approaches in order to obtain high quality predictive models.

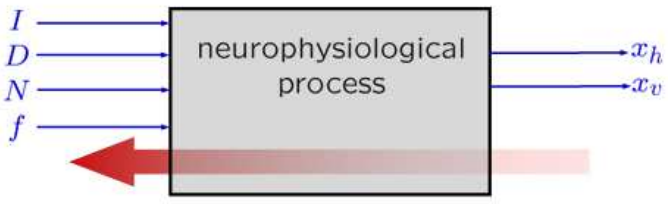


FIGURE 5.7. Prediction problem. The primary aim is to model the unknown neurophysiological problem, linking the stimulation parameters to the position of the phosphenes. In addition, unknown parameters may significantly influence this process. The ultimate goal is to reverse the predictive model in order to select the stimulation parameters corresponding to the desired visual sensation.

5.2.1. Neurophysiological Predictive Model

The neurophysiological model is depicted in Figure 5.8. It assumes that the stimulation strength S is the key quantity for predicting the features of the phosphenes, and in particular their position. The stimulation strength is defined as the useful proportion of the stimulation current to generate a visual sensation:

$$S = \frac{I - I_{\text{th}}}{I_{\text{th}}} \ . \tag{5.2}$$

The perception threshold I_{th} is the minimum current required to elicit a phosphene. In practice, this quantity is estimated experimentally, using a two-staircase limit method (see for example Delbeke et al., 2001). The strength-duration relationship (5.1) is only valid for the activation of individual fibres. Here however, we do not focus on the fibre activation directly, but rather on the perception of phosphenes which result from the activation of one or more fibres in the optic nerve. Delbeke, Parrini, Michaux, Vanlierde and Veraart (2000) showed experimentally that the perception threshold obeys in form to the classical strength-duration curve. The main difference resides in the fact that the rheobase now depends on the frequency f of the pulse train and on the number of pulses N:

$$I_{\text{th}}(D, N, f) = \frac{I_{\text{r}}(N, f)}{1 - \exp\left(-\frac{D \log 2}{D_{\text{c}}}\right)} \ . \tag{5.3}$$

In practice, it was noted that the perception threshold decreases when either the number of pulses or the frequency are increased. These experimental results suggest the existence of a synaptic-like temporal summation mechanism, i.e. that the fibres have some sort of memory. Moreover, the phenomenon occurs although biphasic pulses prohibit electric charge accumulation. Therefore, a central integration mechanism has been hypothesized (Delbeke et al., 2000).

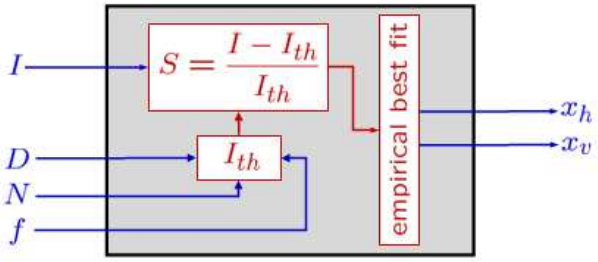


FIGURE 5.8. Neurophysiological predictive model. The stimulation parameters specify the current pulse train sent to the electrodes, which produces a visual sensation. The duration D of the pulses, their number N and the frequency f of the pulse train determine the perception threshold I_{th}. The stimulation strength S, which is the useful proportion of the stimulation current, is then used to predict the phosphene's position in the visual field of the blind.

Based on synapse electrophysiology, such a behavior can be roughly mimicked by a summation of equal sized decreasing exponential curves.

Let us denote the proportion of fibres activated by a single pulse at perception threshold by P_S and the proportion of fibres activated by N identical pulses at perception rheobase by P_N. If we assume that both stimuli produce an hypothetical integrating neuron to fire such that a perception is induced, it is likely that this neuron acts as if the same proportion of fibres is activated. This yields the following relationship:

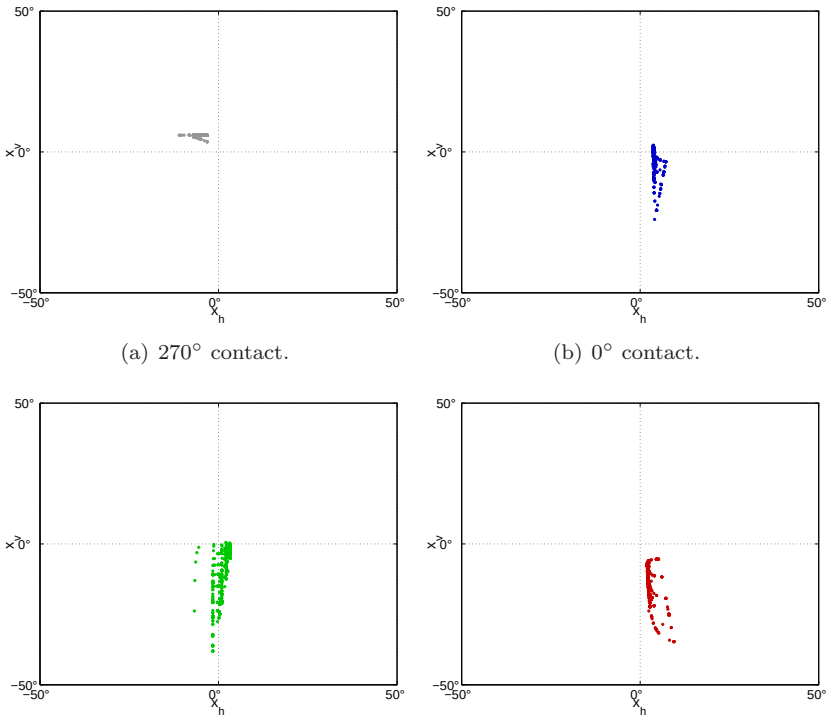
$$P_S = P_N \sum_{i=1}^{N} \exp\left(\frac{i-N}{\tau f}\right) , \tag{5.4}$$

where τ is a time constant. As discussed by Delbeke et al. (2003), the proportion P_N is directly related to the perception rheobase $I_{\text{r}}(N, f)$:

$$I_{\text{r}}(N, f) = \frac{I_1 + (I_{P_S/2} - 2I_1)P_N}{1 - P_N} , \tag{5.5}$$

where I_1 and $I_{P_S/2}$ are respectively the current for which a single axon is activated and the current for which half the population is.

Finally, the neurophysiological model postulates migration of the phosphenes on the basis of further experimental evidence. By "migration", it is meant that the phosphenes follow L-shape trajectories towards the center of the visual field when the stimulation strength is increased. While the origins of these trajectories depend on P_S/P_N, the end points only depend on the electrode contacts. The predictions made by the neurophysiological model are shown in Figure 5.9. These predictions should be compared to the recorded locations shown in Figure 5.6. Although the areas of the visual field in which each electrode generates



(a) 270° contact.

(b) 0° contact.

(c) 180° contact.

(d) 90° contact.

FIGURE 5.9. Phosphene locations predicted by the neurophysiological model. Each panel corresponds to a different electrode contact. Each location corresponds to one of the locations shown in Figure 5.6.

phosphenes are in accordance with each other, the model only accounts partially for the initial dispersion of the phosphenes. This is easily verified by standard statistical tools. In order to assess the quality of the neurophysiological model, we construct a linear regressor between the predicted positions (in each direction) and the observed ones. Table 5.1 shows the coefficients of determination and their level of significance. The coefficient of determination r^2 is defined as the explained variance by the predictive model divided by the total variance (see Appendix B for further details on linear regression). As a consequence, when $r^2 = 1$, the predictive model is perfect, while when $r^2 = 0$, it has no predictive power. One can thus observe from the table, that the neurophysiological model has only limited predictive power, except for the 90° and 180° electrode contacts (in elevation only).

In the next section, we tackle the problem from a machine learning perspective. It is our hope that the resulting black-box models will have higher predictive power. Using nonlinear machine learning tools is also motivated by the fact

TABLE 5.1. Coefficient of determination r^2 and its p-value obtained by means of the F-test. In most cases, the results are highly significant ($p < .01$).

	0°		90°		180°		270°	
	x_h	x_v	x_h	x_v	x_h	x_v	x_h	x_v
Neurophys. model	.03 (.00)	.04 (.00)	.01 (.03)	.44 (.00)	.01 (.01)	.42 (.00)	.03 (.00)	.02 (.01)
Linear model	.03 (.00)	.07 (.00)	.02 (.00)	.28 (.00)	.05 (.00)	.18 (.00)	.06 (.00)	.03 (.00)
VQ RBFN	.32 (.01)	.46 (.01)	.35 (.01)	.58 (.01)	.32 (.01)	.59 (.01)	.42 (.01)	.37 (.01)
MAP RN	.49 (.00)	.58 (.00)	.52 (.00)	.69 (.00)	.43 (.00)	.66 (.00)	.45 (.00)	.45 (.00)
RVM	.26 (.00)	.47 (.00)	.38 (.00)	.66 (.00)	.34 (.00)	.62 (.00)	.40 (.00)	.40 (.00)
Hybrid RN	.43 (.00)	.56 (.00)	.45 (.00)	.68 (.00)	.42 (.00)	.65 (.00)	.44 (.00)	.45 (.00)

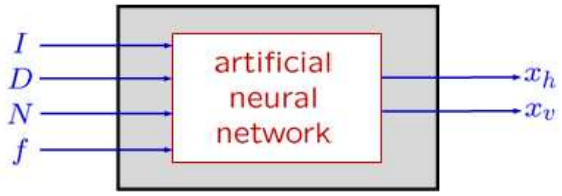


FIGURE 5.10. Black-box predictive model. The artificial neural network is implemented by either linear regression, the vector quantization-based radial basis function network, the maximum a posteriori regularization network or the relevance vector machine.

that the migration of the location of the phosphenes is described in the neurophysiological model by a functional form based only on intuition. In addition, it requires to set many parameters, which are currently estimated by fitting the model directly to the data.

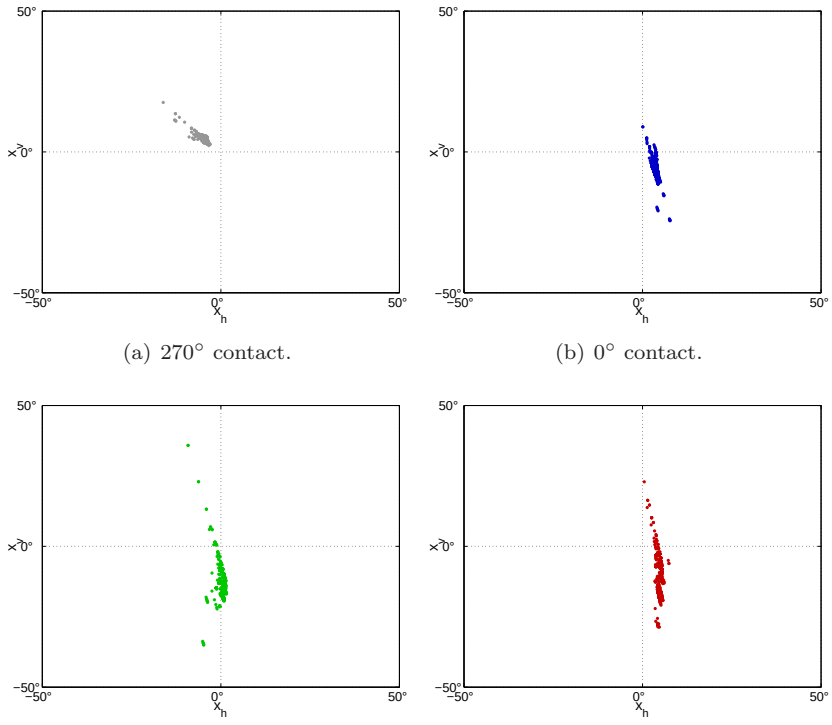
5.2.2. Black-box Predictive Models

As discussed in the previous section, the neurophysiological process involved during the electrical stimulation of the optic nerve is largely unknown. It is therefore proposed to use nonlinear statistical tools (i.e. machine learning techniques or artificial neural networks) in order to build more accurate predictive models. Preliminary results on this matter were reported by Archambeau, Lendasse, Trullemans, Veraart, Delbeke and Verleysen (2001).

Statistical methods are black-box approaches, meaning that they link any input-output relationship based on a set of examples (i.e., the learning set) and are able to generalize on new data points. These methods can thus model any underlying process, provided a sufficient number of data is available. The black-box predictive model used in the case of the optic nerve visual prosthesis is illustrated in Figure 5.10. The main advantage of the approach is that any unknown process can be captured, without assuming the form of its functional form. Unfortunately, the price we have to pay is that, in general, it does not provide any (neurophysiological) interpretation. Next, we investigate both linear and nonlinear techniques.

Linear model

Before investigating the performance of nonlinear tools, we first consider multiple linear regression, which will be later used as reference model. This model can be formalized as a particular case of the radial basis function network (RBFN), which is studied in Section 4.1. In contrast to the RBFN, the design matrix $\mathbf{\Phi}$ is here defined by linear kernels instead of nonlinear ones. In other



(a) 270° contact. (b) 0° contact.

(c) 180° contact. (d) 90° contact.

FIGURE 5.11. Phosphene locations predicted by the linear model. Each panel corresponds to a different electrode contact. Each location corresponds to one of the locations shown in Figure 5.6.

words, the lines of $\mathbf{\Phi}$ are given by $\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = (1, \mathbf{x}_1, \ldots, \mathbf{x}_N)$. The model parameters $\mathbf{w}$ can then be computed by using (4.6), which leads to the well-known least-squares solution from statistics.

The predictions made by the linear model are shown in Figure 5.11. Obviously, the linear model cannot reproduce the dispersion of the recorded data. When looking at Table 5.1, it is clear that the linear model has very few predictive power. However, this result suggests that the underlying neurophysiological process is rather nonlinear, and thus that nonlinear tools are more suitable as shown below.

Nonlinear models

Archambeau, Delbeke, Veraart and Verleysen (2004) investigated predictive models based on the multi-layer perceptron (MLP) and the vector quantization-based RBFN. The model parameters were optimized by 10-fold cross-validation

or the bootstrap. Furthermore, due to numerical instabilities, it was advised to construct the final predictors by model aggregating, such as averaging or bagging (Breiman, 1996). The results with the vector quantization-based RBFN are reported in Table 5.1 for comparison purposes. The MLP is performing very similarly and is therefore not further discussed.

In this section, we follow a different approach. We use maximum a posteriori (MAP) regularization networks (RN) and relevance vector machines (RVM). The MAP RN is expected to provide stable solutions as a basis function is placed on each learning data. By contrast, relevance vector machines (RVM) seek a sparse regressor and may thus be more sensitive to the training set. Both regressors are described in detail in Chapter 4. The results are reported in Table 5.1. The kernel precision is optimized by 10-fold cross-validation. Globally, the RVM performs similarly as the vector quantization-based RBFN, but only requires to optimize a single parameter: the kernel width. Although the RVM is able to better capture the dispersion of the data, it can be seen from Figure 5.12 that it is still unsatisfactory. For example, when considering the 270° electrode contact, it can be observed that the predictions are mainly located on a very thin cross, suggesting some form of overfitting; note also the relatively high number of relevance vectors compared to the number of training data (see Table 5.2).

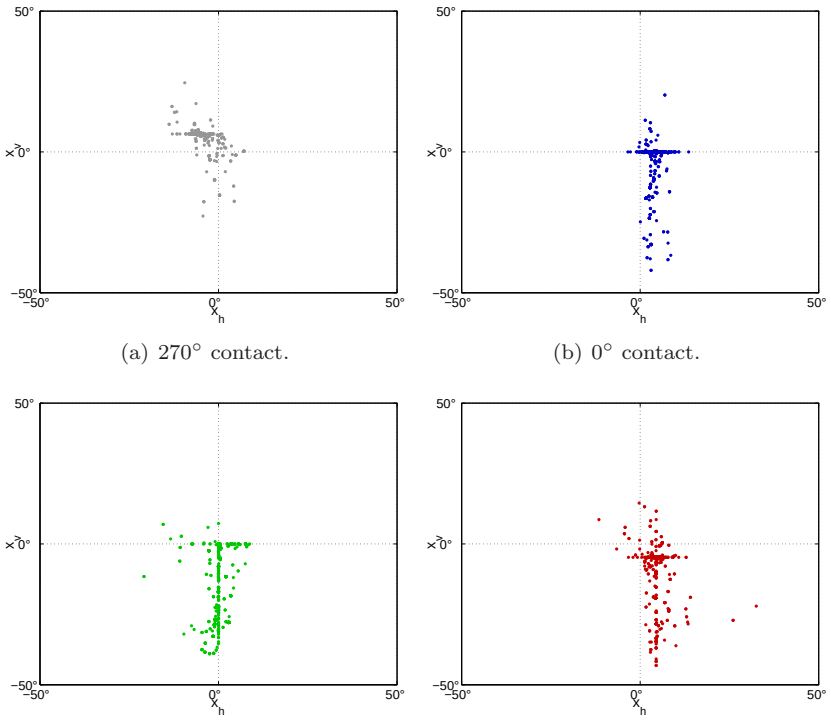
By contrast, the MAP RN provides a much more satisfactory predictive model. This is confirmed visually by Figure 5.13 and numerically by Table 5.1. Note that the training procedure of the MAP RN is slower, since the regularization constant α also needs to be optimized in addition to the kernel precision. However, this is not a problem in practice as the data acquisition is much more expensive.

Table 5.2 shows the mean square error (MSE) estimated by 10-fold cross-validation, as well as the number of basis functions used by the predictive model and the noise estimate. The values obtained for the MSE confirm that the MAP RN performs better than the RVM in this context. The error bars for the RVM are approximately constant, meaning that most of the uncertainty is due to the noise on the data, rather than the uncertainty in the predictions that are made. The RVM and the MAP RN are basically in agreement regarding the amount of noise, which is close to the actual precision that can be expected from the recordings during the experiments. It can also be observed that the noise is larger in terms of elevation for all electrode contacts.

The main drawback of black-box predictive models is that they are only of little help for understanding the underlying neurophysiological process. By contrast, a hybrid predictive model, which combines neurophysiological knowledge and nonlinear statistical tools, might be more instructive. This model is discussed in the next section.

TABLE 5.2. Mean square error estimated by 10-fold cross-validation, number of basis functions and expected noise standard deviation (which is estimated on the training set). These quantities are given for the models in azimuth (h) and elevation (v).

	0°			90°			180°			270°		
	MSE_h	M_h	σ_h	MSE_h	M_h	σ_h	MSE_h	M_h	σ_h	MSE_h	M_h	σ_h
MAP RN	15.8	665	3.2	20.2	640	3.5	29.4	683	4.6	24.9	557	4.4
RVM	16.8	67	3.9	20.8	77	4.1	31.6	121	5.1	26.1	51	4.8
Hybrid RN	16.7	665	3.3	21.2	640	3.8	30.2	683	4.6	25.0	557	4.5
	MSE_v	M_v	σ_v	MSE_v	M_v	σ_v	MSE_v	M_v	σ_v	MSE_v	M_v	σ_v
MAP RN	120.7	665	9.4	78.0	640	7.6	117.8	683	9.2	59.5	557	6.7
RVM	137.6	103	11.0	97.9	104	8.4	120.0	90	9.9	63.2	71	7.2
Hybrid RN	131.9	665	9.6	82.3	640	7.7	114.8	683	9.3	58.1	557	6.7



(a) 270° contact.

(b) 0° contact.

(c) 180° contact.

(d) 90° contact.

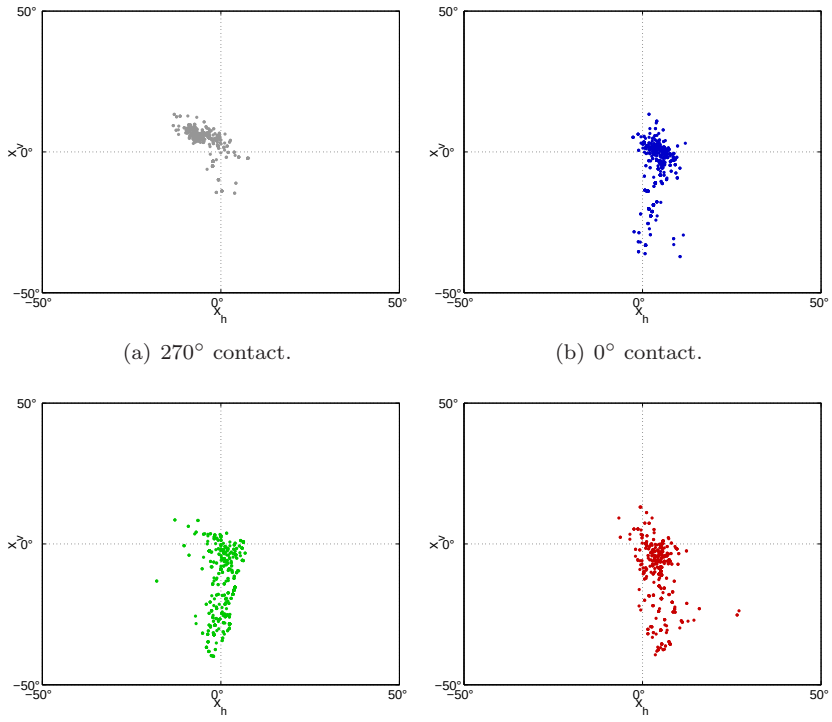
FIGURE 5.12. Phosphene locations predicted by the RVM model. Each panel corresponds to a different electrode contact. Each location corresponds to one of the locations shown in Figure 5.6.

5.2.3. Hybrid Predictive Model

Hybrid predictive models use the most reasonable physiological knowledge at disposal, as well as (nonlinear) statistical tools to learn unexpected or ill-characterized relationships through the use of the data. Therefore, we first extract the most reliable neurophysiological information, before using the black-box models.

First, consider again the perception threshold I_{th}. This quantity is linked to the rheobase I_r through the strength-duration curve which is given by equation (5.3). When considering the ratio between I_{th} and I_r, we obtain a quantity that only depends on the pulse duration D and the chronaxie D_c. This constant can easily and reliably be fitted to the data (see Delbeke et al., 2000).

Second, the rheobase depends on the number of pulses N in the pulse train and the frequency f by means of the proportions of fibres P_N activated by a



(a) 270° contact. (b) 0° contact.

(c) 180° contact. (d) 90° contact.

FIGURE 5.13. Phosphenes' locations predicted by the MAP RN model. Each panel corresponds to a different electrode contact. Each location corresponds to one of the locations shown in Figure 5.6.

single pulse of the pulse train. Using (5.4), we may rewrite (5.5) as follows:

$$I_{\mathrm{r}}(N,f) = \frac{I_1 \sum_{i=1}^{N} \exp\left(\frac{i-N}{\tau f}\right) + (I_{P_S/2} - 2I_1)P_S}{\sum_{i=1}^{N} \exp\left(\frac{i-N}{\tau f}\right) - P_S}\,, \tag{5.6}$$

Since P_S is the minimal proportion of fibres to be activated in order to elicit a visual perception (and is thus a constant for each electrode contact), the informative part of the rheobase is given by the ratio P_S/P_N. In addition, according to the neurophysiological model, this ratio determines the origin of the L-shape trajectories of the phosphenes when the current amplitude increases and the other stimulation parameters are fixed. Note that the time constant τ can be fitted to the data by using the volume-conductor model of the optic nerve (Parrini et al., 2000).

A diagram of the resulting hybrid predictive model is shown in Figure 5.14. The neural network is implemented by a MAP RN, as this model yields the

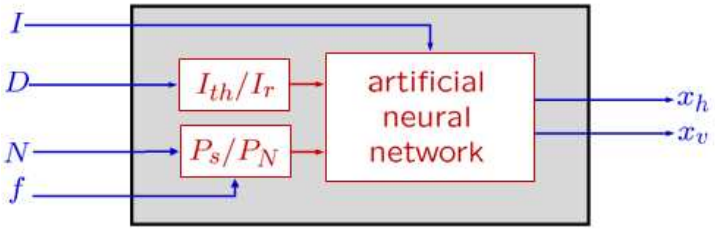


FIGURE 5.14. Hybrid predictive model. The artificial neural network is implemented by the maximum a posteriori regularization network.

best results in the case of the black-box predictive models. Table 5.1 shows that the hybrid model has a very similar predictive capability as its black-box equivalent. Only a slight loss of performance can be noted in terms of elevation. This is also observed when looking at the MSE in Table 5.2. However, a major advantage of this model is that 3 inputs are feeded to the neural network instead of 4. This is very important when planning future data acquisition, as the number of experiences to conduct in order to cover the entire feature space grows exponentially with the dimension. Remember that the stimulation sessions are very time consuming and tiring for both the blind volunteer and the persons conducting the experiments.

5.3. Classification of Phosphenes

In the previous section, predictive models have been developed in order to predict the location of a single phosphene. A phosphene is elicited by activating a single electrode contact at a time. An electrode contact is said to be activated when it is sent an electrical pulse train. However, it was observed experimentally that in most cases, when several contacts are activated in a short time period, the same number of phosphenes is perceived. This phenomenon is attractive in practice as it allows us to increase the amount of visual information transmitted to the volunteer within this time period.

Let us denote the starting times of two pulse trains respectively by t_1 and t_2 and the total duration of the first (reference) pulse train by T_1. The electrode contacts can be combined in three ways, leading to the following stimulations:

$$\begin{cases} |t_1 - t_2| = 0 & : \text{ synchronous stimulation;} \\ |t_1 - t_2| < T_1 & : \text{ interlaced stimulation;} \\ |t_1 - t_2| > T_1 & : \text{ sequential stimulation.} \end{cases}$$

It was established that in 80% to 90% of the experiments, depending on the type of stimulation combination, the number of perceived phosphenes is equal to the number of activated electrode contacts. Furthermore, it was found that in the case of single contact activations, for each contact, a restricted, but dissimilar area of the visual field is accessible (see Figure 5.6). In practice, it is

likely that the phosphenes still light up in the corresponding areas of the visual field when considering combinations of electrode contacts. Based on these experimental results, it seems reasonable to assume spatial superposition when electrode contacts are combined. In other words, when a set of pulse trains is sent to several electrode contacts within a sufficiently short time period, we may assume that the resulting visual perceptions are a superposition of all the single visual perceptions. In practice however, a slight influence has been noticed on the exact phosphene location, but the effect is limited and localized inside the specific area associated to each electrode contact.

Subsequently, we characterize more accurately the activation areas associated to each electrode contact. Besides, this enables us to tackle the problem of analyzing the data resulting from experiments involving electrode combinations. Indeed, based on the probability that a phosphene was generated by a particular electrode contact, we may classify the induced visual perceptions and assign them to the most probable activated contact. This problem was already partially investigated by Archambeau, Delbeke and Verleysen (2003).

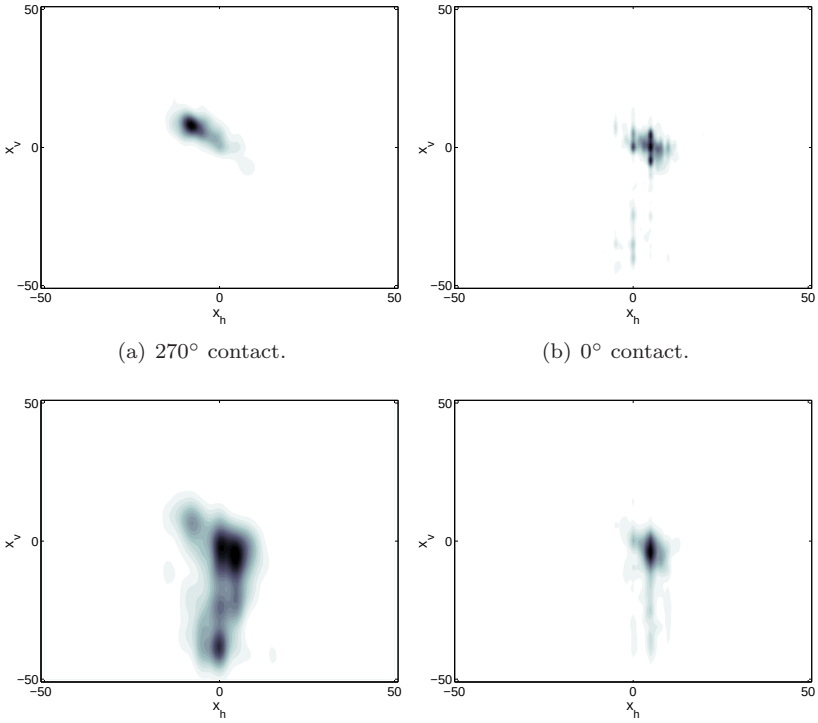
5.3.1. Activation areas

Describing the activation areas associated to each electrode contact by means of their probability density function is particularly appealing. It allows determining the most suitable contact to activate in order to generate a specific visual perception and with what confidence we may generate this perception. This information is very important for the setup of an efficient stimulation strategy.

Figure 5.15 shows the estimated density for each contact when using the ordinary kernel density estimator (KDE). The darker the color, the higher the density. The method is described in Section 2.2.2. The kernel precision is optimized for each contact separately by 10-fold cross-validation. The optimal value is selected as the one minimizing the average negative log-likelihood (ANLL) of the validation sets.

It can be observed that the density of electrode contact 180° is less localized than the other ones. Furthermore, one can notice very high peaks in the estimators (cf. very dark spots), especially for contact 0°, suggesting some overfitting. This result is surprising as we use statistical resampling techniques to avoid this kind of problem. However, when taking a closer look to the data, this can be explained as follows. Due to the sampling process and the experimental setup, the data base contains occasionally data points which are repeated a large number of times. As a result, these data points bias the estimation of the ANLL, leading to an overestimated kernel precision. Note that the problem was even more severe when using adaptive KDE, such as the sample point kernel density estimator (see Section 2.3.2).

In order to obtain more reliable estimators, we use variational Gaussian mixture models (GMM). As discussed in Section 3.2.5, the variational GMM avoid numerical instabilities in contrast to the standard GMM. This is important



(a) 270° contact. (b) 0° contact.

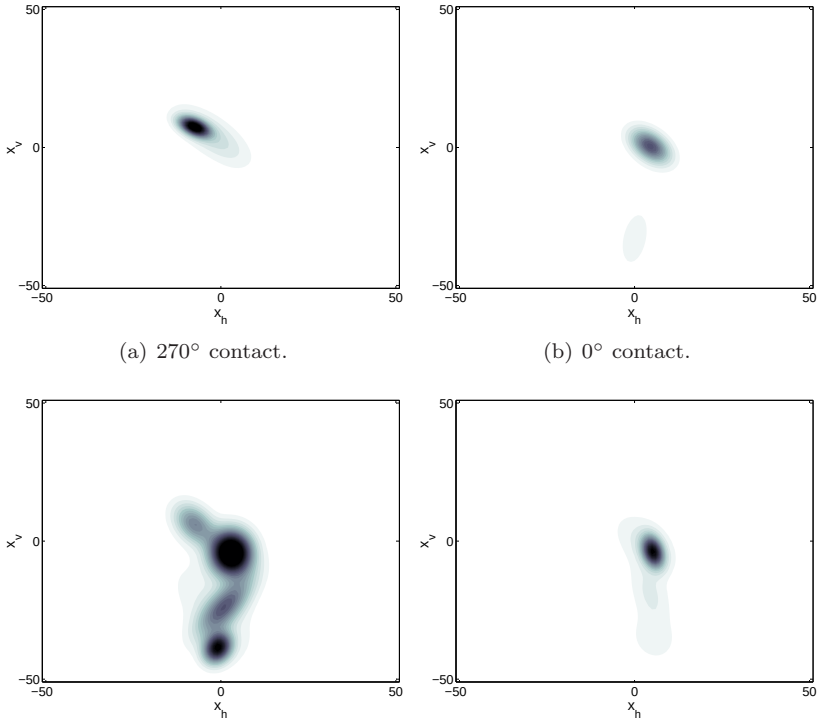
(c) 180° contact. (d) 90° contact.

FIGURE 5.15. Nonparametric probability density estimation of the phosphene location for each electrode contact. The estimators are constructed by the kernel density estimator. The kernel precision is chosen as the one minimizing the average negative log-likelihood, which is estimated by 10-fold cross-validation. The data is shown in Figure 5.6.

in this context as the GMM failed to provide consistent estimators due to numerical instabilities caused by the repetitions in the data. The estimators provided by the variational GMM are shown in Figure 5.16. Obviously, they are smoother and thus more intuitive than the ones obtained with KDE. The number of components in the mixture is optimized for each electrode contact on the basis of the variational lower bound on the log-evidence. Again, this is attractive as the model complexity can be optimized in a single run, without having to split the data in a training and validation set (see Section 3.2.5).

5.3.2. Classification model

The phosphenes can be classified based on the probability of eliciting a phosphene by a particular electrode contact. The classification problem can



(a) 270° contact. (b) 0° contact.

(c) 180° contact. (d) 90° contact.

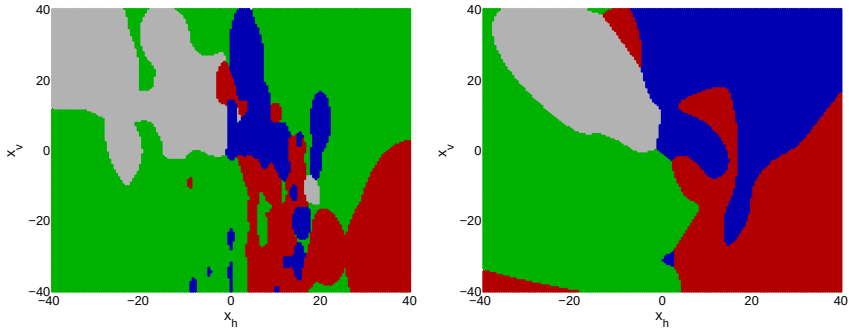
FIGURE 5.16. Probability density estimator of the phosphene locations for each electrode contact. The models are constructed with the variational Gaussian mixture model. The number of components is selected as the one maximizing the variational lower bound. The data is shown in Figure 5.6.

be stated as follows. For each perceived phosphene, we would like to find the most probable electrode contact among the activated ones. The ultimate goal of this classification problem is to decompose the perceptions induced by stimulation combinations, in order to further analyze the data and verify if the predictive models discussed in the previous section are still applicable.

Once the density is estimated, we may perform Bayesian classification using Bayes' rule:

$$P(C|\mathbf{x}) = \frac{p(\mathbf{x}|C)P(C)}{p(\mathbf{x})} \,, \tag{5.7}$$

where $P(C)$ is class prior and $p(\mathbf{x}|C)$ is the probability of $\mathbf{x} = (x_h, x_v)$ when assuming it was generated by class C. The probability $p(C|\mathbf{x})$ is thus the posterior probability of having class C when $\mathbf{x}$ is observed. The normalizing



(a) KDE. (b) Variational GMM.

FIGURE 5.17. Phosphene classes obtained for (a) the kernel density estimator and (b) the variational Gaussian mixture model. The 4 electrode contacts are activated, which are indicated by colors (blue for 0°; red for 90°; green for 180°; grey for 270°).

constant is given by

$$p(\mathbf{x}) = \sum_C p(\mathbf{x}|C)P(C) \ . \tag{5.8}$$

The classification results based on the density estimators from Figures 5.15 and 5.16 are shown in Figure 5.17. We assume that the 4 electrode contacts are activated, which is the worst case as the overlap between the different classes is maximal. To each color corresponds a winning electrode contact. As expected, the KDE provides rather noisy classification boundaries. By contrast, the classification result provided by the variational GMM is much more reassuring. In particular, it confirms the hypothesized (coarse) retinotopic structure of the optic nerve. Nevertheless, it can be observed that some areas associated to the electrode contact 90° are doubtful (e.g., lower left corner), but fortunately, in practice no phosphenes have ever been elicited in those areas.

Table 5.3 takes a quantitative look at the classification results shown in Figure 5.17. The table represents the empirical confusion matrix when classifying the phosphenes associated to the single contact stimulation (see Figure 5.6). The density models are built with the variational GMM. The confusion matrix counts, for each class, the number of data that are correctly classified and the number of misclassifications. Each line represents the target (or true) class and each column the class the data is assigned to. Therefore, the sum of the proportions of each line is equal to 1. At first sight, the classification results are poor. This is mainly due to the fact that when the 4 contacts are activated, the classes are strongly overlapping. However, in practice, most of the experiments involve 3, and usually only 2 contacts. Of course, the resulting classification performances increase considerably. Two examples are shown in

TABLE 5.3. Empirical confusion matrix for the variational GMM estimators when the 4 electrode contacts are activated. The proportions of correct classifications are on the diagonal. The true class labels are indicated in italics, while the assigned class labels are straight.

	0°	90°	180°	270°
0°	**0.37**	0.29	0.16	0.18
90°	0.18	**0.34**	0.26	0.22
180°	0.17	0.16	**0.56**	0.11
270°	0.09	0.14	0.13	**0.64**

TABLE 5.4. Empirical confusion matrix for the variational GMM estimators when the electrode contacts 90°, 180° and 270° or 0° and 270° are activated. The proportions of correct classifications are on the diagonal. The true class labels are indicated in italics, while the assigned class labels are straight.

	90°	180°	270°
90°	**0.44**	0.34	0.22
180°	0.19	**0.68**	0.13
270°	0.18	0.15	**0.67**

	0°	270°
0°	**0.74**	0.26
270°	0.24	**0.76**

Table 5.4. The confusion matrices for the other contact combinations are given in Appendix C.

As a final remark, one may object that the generic approach that we follow is suboptimal, since we do not solve the classification problem directly as would be the case with discriminative techniques (e.g., support vector machines). However, one realize that the generic approach is attractive in this biomedical application, as it provides us with additional information. On the one hand, the densities tell us where an activated electrode contact is the most likely to induce a phosphene. On the other hand, using a Bayesian classifier provides us with a confidence measure on the classification results. Furthermore, we do not need to recompute a new model for each of the many electrode combinations, which is very attractive in practice.

5.4. Stimulation Strategy

Currently, a limited number of phosphenes is used during the stimulation sessions. However, it was recently reported by Brelén, Duret, Gérard, Delebeke and Veraart (2005) that a better performance is observed in terms of object or pattern recognition when the number of phosphenes used by the stimulation

algorithm is increased. Note that in order to achieve these better results, a longer training period of the blind volunteer is required. In this context, it is also important to have relevant selection criteria for determining the most informative phosphenes.

Based on the predictive and the classification models described in the previous sections, one can think of an enhanced stimulation algorithm, which will supply meaningful visual information to the blind. The proposed stimulation algorithm is depicted in Figure 5.18. In order to reconstruct shapes in the visual field of the blind, we can proceed as follows. After having recorded an image and having performed some form of edge detection in the external processor, the set of phosphenes to generate is identified. This is done by superposing the object contours with a predifined phosphene map. The most suitable electrode contact to generate each of the selected phosphenes can then be identified by means of the classification models. In addition, we may rank these phosphenes according to their probability, as it is more likely that we can generate accurate predictions in high density regions. Next, using lookup tables implementing the inverse predictive models, we determine the adequate stimulation parameters for each visual perception to induce. The corresponding pulse trains are then sent via the stimulator to the optic nerve. Finally, we remove the phosphenes that have already been generated from the list to provide new information to the blind. Note however that after a certain time period, it should be possible to generate them again.

In conclusion, this new stimulation strategy is particularly useful when one wants to increase the number of phosphenes to work with. On the one hand, the classification models provide automatic and appealing selection rules for the visual perceptions to elicit. On the other hand, the predictive models and in particular the black-box or hybrid models, predict the phosphene locations with a sufficient accuracy, such that the corresponding inverse models are expected to provide adequate stimulation parameters.

5.5. Summary

The probabilistic models described in the previous chapters are here applied to two modeling problems related to the optic nerve visual prosthesis.

First, predictive models are introduced to model the neurophysiological process linking the stimulation parameters to the visual perceptions induced in the visual field of the blind. These visual perceptions are produced by stimulating electrically the optic nerve. We showed that the black-box models have a much greater predictive power than the neurophysiological models developed so far. A hybrid model, combining reasonable neurophysiological knowledge and black-box models to model the unknown part of the underlying process, was also proposed. This model has the advantage of reducing the dimension of the feature space. This is particularly useful when planning future experiments.

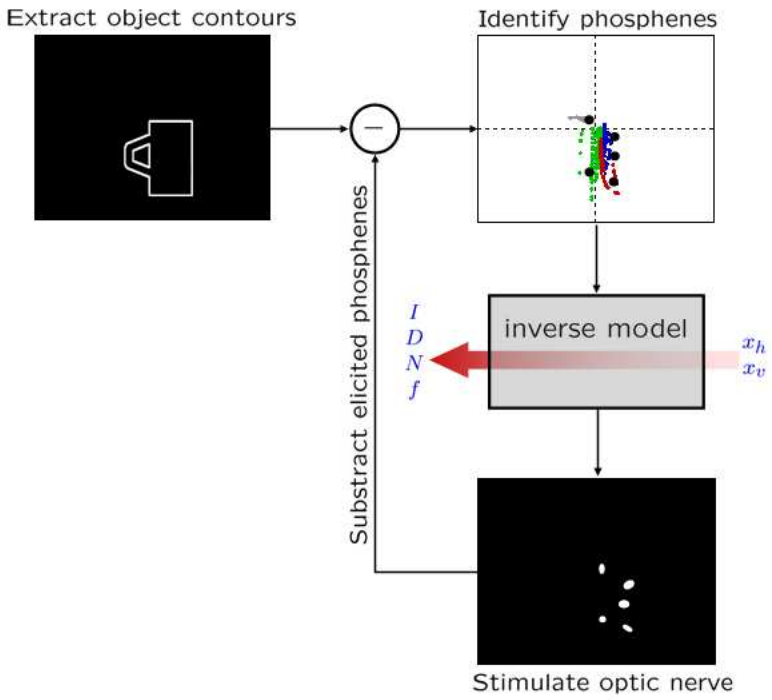


FIGURE 5.18. Stimulation strategy based on the predictive and classification models of the phosphenes.

Second, classification tools are discussed. These techniques enable us to associate parts of the visual field to the electrode contacts. In other words, each electrode contact produces visual perceptions in different areas of the visual field. On the one hand, the classification models allow us to determine the most suitable phosphenes to elicit in order to create meaningful visual information for the blind. On the other hand, they enable us to decompose the perceptions associated to stimulation combinations. This is required for further analyzing these experiments in a principled way.

Finally, the predictive and the classification models were combined in order to build an advanced stimulation algorithm. However, this algorithm needs to be validated experimentally in the near future.

CHAPTER 6

Conclusion

The primary goal of this thesis was to put probabilistic models into a practical perspective. More specifically, nonparametric kernel density estimators, finite mixture models and probabilistic regressors were investigated in the difficult, but common situation where the data is very noisy and scarce. Most techniques perform very well on toy examples. However, many of them often fail to provide meaningful solutions when tackling challenging applications.

First, well-known techniques in nonparametric density estimation were reviewed and extensively compared on benchmark data sets. We considered both the effect of the size of the training set and the amount of noise. We studied mainly multivariate data. It was shown at length how gradually, in a time span of approximately two decades, statisticians moved from methods having fixed parameters across the feature space to locally adaptive ones. As expected, adaptive methods are more accurate, but they are also less robust to inconsistencies and inhomogeneities in the data. In practice, this can lead to problematic situations.

Modern nonparametric approaches are closely related to finite mixture models and therefore motivate the use of the latter in a more general framework, which we termed nonparametric-like density estimation. Unfortunately, mixture models have a serious drawback when they are used for this purpose. Their parameters are estimated iteratively by maximizing the likelihood of the observed data, which is known to be an ill-posed problem. This is a consequence of the likelihood function being unbounded. As a result, some modifications of the models are required.

The most widely used finite mixture model is the finite Gaussian mixture model (GMM). The GMM uses the Mahalanobis distance to determine the shape of its components. A simple modification to avoid numerical instabilities and increase the GMM's generalization abilities is to constrain the shape of the components through some form of regularization. For instance, this can be achieved by using the regularized Mahalanobis distance, which is computed as a weighted average of the Euclidean distance and the ordinary Mahalanobis distance. Another, yet more elaborate technique is to estimate the maximum a posteriori parameters instead of the maximum likelihood parameters. However, maximum a posteriori learning requires to set many hyperparameters, which are usually difficult to optimize. Therefore, we proposed a practical maximum

a posteriori learning scheme. In this approach, the hyperparameters do not need to be optimized directly, but are set according to a prior belief. The effect of the prior is then tempered or amplified by means of an additional parameter, which is optimized in a classical way (i.e., by resampling techniques).

The most recent advances in Bayesian learning made it possible to increase the robustness of the GMM dramatically. This is due to the fact that in the Bayesian approach the uncertainty on the parameters is taken into account in a principled way. Furthermore, one gets for free a lower bound on the log-evidence, which allows us to determine the model complexity automatically. Note however that a fully Bayesian approach cannot be used unless one makes additional assumptions on the form of the solution. As a consequence, too simple models are favored. A future research direction would be to relax these assumptions in order to find a tighter lower bound, which in turn is expected to be more reliable for model selection.

A robust alternative to the GMM is the finite Student-t mixture model (SMM). The Student-t distribution has an additional parameter, the degrees of freedom, which regulates the robustness of the distribution to atypical observations. In this work, we showed that all the approaches used in the frame of the GMM, could be extended to the SMM. Furthermore, a new variational algorithm was introduced for learning Bayesian SMM. This algorithm is particularly attractive in noisy environments as it leads to very robust models. The reason for this is that, in contrast to previous approaches, unnecessary approximations are avoided, leading for example to a tighter variational lower bound.

Although the feature space is in many applications high dimensional, the data are often living on a lower dimensional manifold. This particular geometric arrangement can be used in the frame of mixture models by means of a constrained expectation step. In practice, this corresponds to lower the contribution of a data point to the parameter update of a particular component, when this point is lying far away on the manifold from that component. It was shown experimentally that the manifold constrained mixture models are attractive as they avoid local maxima of the likelihood function the ordinary mixture models may get trapped into. A possible future research direction is to extend this approach to other graphical models than finite mixture models. For example, one could think of variants of the relevant vector machines, which would exploit this additional information in regression problems.

After having discussed probabilistic regularization networks, which fit the same latent variable framework as finite mixture models, we applied them to a modeling problem related to the optic nerve visual prosthesis. More specifically, visual perceptions can be elicited in the visual field of the blind by electrical stimulation of his/her optic nerve. However, it is unclear how these electrical pulses induce visual perceptions at the neurophysiological level. These perceptions are therefore difficult to predict. Instead of constructing a neurophysiological model where all the input quantities have a neurophysiological interpretation, it was suggested to approach the problem from a machine learning

perspective. Machine learning tools are powerful (nonlinear) statistical tools, which can recover any relationship between the input and the target data. We showed in this thesis that black-box prediction models with sufficient accuracy can be constructed in order to link the stimulation parameters to the corresponding visual sensations, the ultimate goal being to inverse these models in order to determine which stimulation parameters should be used to generate the desired visual perception. A hybrid (or grey-box) model having a comparable accuracy was also proposed. In this model, some neurophysiological quantities are explicitly used. This approach should be further investigated in the future in order to obtain predictive models that are more instructive to medical doctors and psychologists. Another important research direction is the development of patient dependent models. Being able to easily tune the predictive models such that they are meaningful to other patients is important in practice. By "easily" is meant that the models can be adjusted without having to collect a large amount of data.

Another major concern is to increase the amount of visual information transmitted to the blind within a given time period. In this context, Bayesian classification models were developed. The aim of these models is to identify, when combinations of electrode contacts are used for stimulation, by which one the corresponding visual perceptions were generated. The classification models provide useful information regarding the unknown neurophysiological process as they indicate where phosphenes might be induced and with which probability. In addition, they are an important building block for efficient stimulation algorithms. It was shown experimentally that combined stimulations elicit phosphenes in the same areas of the visual field as simple stimulations. However, it is still unclear how they interact with each other. A careful study of the related data will definitely help to understand the underlying neurophysiological mechanisms.

Although it is currently an active field of research, very little is known about the coding scheme of the visual information in the optic nerve or in the visual pathways in general. A deeper insight into the ways visual information is encoded will be extremely valuable in the design of future visual prostheses. Furthermore, sending meaningful information to the blind involves high level image processing in order to extract the relevant information and send it to the stimulator. Answering this question will involve psychophysics to determine what information is the most relevant, as well as engineering tasks to extract this information in an automated way. In conclusion, one should realize that there are still many open questions regarding the design of visual prostheses and their use in practice, including the long term viability of the implanted system in the human body or the precision visual perceptions can be actually induced with. Hopefully, some practical clinical systems will appear in the near future, but there is still a long way to go before these systems will be made widely available and there are still a lot of obstacles to be overcome...

APPENDIX A

Benchmarks

The data sets used in Chapter 2 are briefly described in this appendix. Some of them are used in subsequent chapters as well. Most are available form the UCI Machine Learning repository (`http://www.ics.uci.edu/~mlearn`) or StatLib (`http://lib.stat.cmu.edu`).

Enzyme data

The first data set concerns the distribution of enzymatic activity in the blood, for an enzyme involved in the metabolism of carcinogenic substances. The data was collected on a group of 245 unrelated individuals. The aim is to identify subgroups of slow or fast metabolizers as a marker of genetic polymorphism in the general population. This data set was first analyzed by Bechtel, Bonaïti-Piellé, Poisson, Magnette and Bechtel (1993), who identified a mixture of 2 skewed distributions using maximum likelihood techniques.

Acidity data

The second data set concerns an acidity index that is measured in a sample of 155 lakes in the Northeastern United States. It was analyzed as a mixture of Gaussian distributions on the log scale by Crawford, Groot, Kadane and Small (1992).

Galaxy data

The third univariate data is the galaxy data, which was first described by Roeder (1990). It consists of the velocities of 82 distant galaxies, diverging from our own galaxy.

Ring data

The ring data is a toy example and was artificially generated. This 2D data set are used several times in this thesis. They are uniformly distributed in an annular region centered on the origin. The mid-radius of the ring equals 5 and its width 2. The data set contains 150 data points.

Noisy spiral data

Another 2D toy data set is the noisy spiral. The data are distributed along a spiral of Archimedes. The radius is defined as follows:

$$r = a\theta \ , \tag{A.1}$$

where a is a constant (chosen equal to 1) and θ is the radius angle. The angular position θ along the spiral follows the uniform distribution $\mathcal{U}(\theta|\frac{\pi}{2}, \frac{5\pi}{2})$ and the spiral width w has the Gaussian distribution $\mathcal{N}(w|0, 2)$. The data set contains 250 data points.

Old faithful geyser data

A popular 2D data set is the old faithful geyser data. It consists in the waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.

Wine recognition data

The wine recognition data are the results of a chemical analysis of wines grown in the same region in Italy, but derived from three different cultivars. The analysis determined the quantities of 13 constituents (alcohol, magnesium, acidity, ash, etc.) found in each of the three types of wines. The data set contains 178 data points.

NO^2 pollution data

The data set contains 500 observations. It originates from a study of air pollution near a road, relating it to the traffic volume and several meteorological variables. It was collected by the Norwegian Public Roads Administration. The predictive quantity consists of the hourly values of the logarithm of the concentration of NO^2 particles, measured at Alnabru in Oslo, Norway, between October 2001 and August 2003. The features are the logarithm of the number of cars per hour, the temperature 2 meter above ground, the wind speed, the temperature difference between 25 and 2 meters above ground, the wind direction, the hour of the day, and the day number from October 2001.

Iris plant data

The iris plant data is probably the best known database in the field of Pattern Recognition (Duda and Hart, 1973). It contains 3 classes of 50 instances each, where each class refers to a type of iris plant: Iris Setosa, Iris Versicolour, Iris Virginica. One class is linearly separable from the others. The latter are not. The features of the plant type are the sepal length, the sepal width, the petal length and the petal width, all being measured in centimeters.

Boston housing data

The features of the Boston house-price data (Harrison and Rubinfeld, 1978) are the following: the crime rate per capita by town, the proportion of residential land zoned for lots over 25,000 sq.ft., the proportion of non-retail business acres, the nitric oxides concentration (parts per 10 million), the average number of rooms per dwelling, the proportion of owner-occupied units built prior to 1940, the weighted distances to five Boston employment centers, the full-value property-tax rate, pupil-teacher ratio, the proportion of black men and women by town and the proportion of lower status of the population. The index of accessibility to radial highways and the dummy variable indicating if the house is close to Charles river are not used. The predictive quantity is the median value of owner-occupied homes. The data contains 506 data points.

Liver disorder data

The Liver Disorder data was gathered by BUPA Medical Research. The data constitutes the record of 345 male individuals. Five features are blood tests thought to be sensitive to liver disorders that might arise from excessive alcohol consumption: the mean corpuscular volume, alkaline phosphotase, alamine aminotransferase, aspartate aminotransferase and gamma-glutamyl transpeptidase. The sixth feature is the number of half-pint equivalents of alcoholic beverages drunk per day.

Body fat data

The body fat data lists estimates of the percentage of body fat determined by underwater weighing and various body circumference measurements for 252 men. The percentage of body fat for an individual can be estimated once the body density has been determined. Usually, one assumes that the body consists of two components: lean body tissue and fat tissue. The volume, and hence the body density, can be accurately measured in a variety of ways. The technique of underwater weighing computes body volume as the difference between body weight measured in air and weight measured during water submersion. In other words, body volume is equal to the loss of weight in water with the appropriate temperature correction for the water's density. The data features are the following: the density determined from underwater weighing, the age, the weight, the height and the circumference of the neck, the chest, the abdomen, the hip, the thigh, the knee, the ankle, the biceps, the forearm and wrist. Finally, the predictive quantity is the percent of body fat.

APPENDIX B

Linear Regression

In this appendix, we introduce linear regression for two scalar variables. This standard statistical tool is particularly suited for assessing the quality of the predictive models in Chapter 5. The idea is to test how informative the predictions made by the models are with respect to the recorded data. The approach is used for the azimuth and elevation coordinates independently as separate models are used for each direction.

Consider the independent and the dependent continuous random variables $\mathcal{X}$ and $\mathcal{Y}$. Given sets of observations $X = \{x_n\}_{n=1}^N$ and $Y = \{y_n\}_{n=1}^N$, we would like to know if the following linear model can explain the relationship between both random variables:

$$y_n = \beta x_n + \alpha + \epsilon_n \ , \quad \forall n \ . \tag{B.1}$$

The parameters α and β are respectively the intercept and the slope of the linear model. The errors $\{\epsilon_n\}_{n=1}^N$ take the departure from linearity into account and are assumed to have zero mean and variance σ^2.

A standard statistical approach for estimating the parameters α and β is to minimize the sum of squared errors:

$$\sum_{n=1}^{N} {\epsilon_n}^2 = \sum_{n=1}^{N} \{y_n - \beta x_n - \alpha\}^2 \ . \tag{B.2}$$

Minimizing this expression leads to the following estimators for the parameters (see for example Härdle and Simar, 2003):

$$\hat{\beta} = \frac{s_{XY}}{s_{XX}} \ , \tag{B.3}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x} \ . \tag{B.4}$$

The special quantities in these equations are the empirical means and (co)variances:

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n \ , \quad s_{XX} = \frac{1}{N}\sum_{n=1}^{N} (x - \bar{x})^2 \ , \tag{B.5}$$

$$\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n \ , \quad s_{XY} = \frac{1}{N}\sum_{n=1}^{N} (y - \bar{y})(x - \bar{x}) \ . \tag{B.6}$$

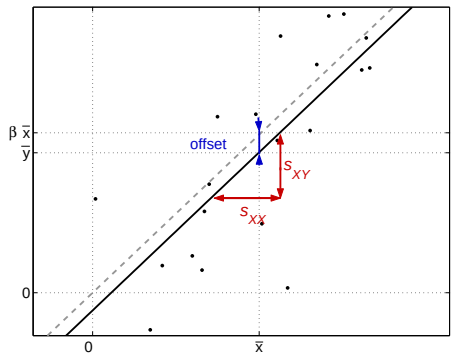


FIGURE B.1. Graphical illustration of the parameter estimators in a linear regression model. The estimated slope $\hat{\beta}$ is given by s_{XY}/s_{XX} and the offset $\hat{\alpha}$ by $\bar{y} - \hat{\beta}\bar{x}$.

As shown in Figure B.1, the solutions (B.3) and (B.4) have an intuitive graphical interpretation. The estimator of β corresponds indeed to a slope as it measures the empirical variation of $\mathcal{Y}$ with respect to $\mathcal{X}$, normalized by the empirical variation of $\mathcal{X}$. The estimator of α measures the empirical offset once the slope is estimated.

Next, we would like to evaluate the goodness-of-fit of the linear model $\hat{y} = \hat{\beta}x + \hat{\alpha}$. First, let us consider the observed total variation of the dependent variable $\mathcal{Y}$:

$$\sum_{n=1}^{N}(y_n - \bar{y}_n)^2 \ . \tag{B.7}$$

Second, the variation explained by the linear model is given by

$$\sum_{n=1}^{N}(\hat{y}_n - \bar{y}_n)^2 \ . \tag{B.8}$$

Third, the total and the explained variation can be linked to the unexplained variation by using elementary statistics:

$$\sum_{n=1}^{N}(y_n - \hat{y}_n)^2 = \sum_{n=1}^{N}(y_n - \bar{y}_n)^2 - \sum_{n=1}^{N}(\hat{y}_n - \bar{y}_n)^2 \ . \tag{B.9}$$

Using these expressions the quality of the linear model and the confidence we have in it can be readily assessed.

In order to measure the quality of the model, it is appealing to compute the ratio of the explained variation and the total variation, which is called the coefficient of determination:

$$r^2 = \frac{\sum_{n=1}^{N}(\hat{y}_n - \bar{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y}_n)^2} = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y}_n)^2} \ . \tag{B.10}$$

This quantity is in the closed interval $[0, 1]$. When $r^2 = 1$, all the variation is explained by the linear model. When $r^2 = 0$, it can be concluded that the relationship (if there is one) between both variable is not linear.

In order to evaluate the confidence we may have in the linear model, we compare it to a prediction by the mean. In other words, we test if the linear model, which uses the observations of $\mathcal{X}$, is more informative than just predicting $\mathcal{Y}$ by its empirical mean. Let us term the linear model the "full model" and the prediction by the mean the "reduced model". The residual variations, i.e. the variations that are not explained by the models, correspond respectively to the unexplained (B.9) and the total variation (B.7). The $\mathcal{F}$-statistic is commonly used in this context to test how significantly the variation is reduced when predicting the data with the full model rather than with the reduced one:

$$\mathcal{F} = \frac{\dfrac{\sum_{n=1}^{N}(y_n - \bar{y}_n)^2 - \sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{df_{\text{red}} - df_{\text{ful}}}}{\dfrac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{df_{\text{ful}}}} = \frac{r^2/(df_{\text{red}} - df_{\text{ful}})}{(1 - r^2)/df_{\text{ful}}} \,, \tag{B.11}$$

where df_{red} and df_{ful} are the degrees of freedom of each model. They are essential as they define the shape of the $\mathcal{F}$-distribution and have a simple interpretation: the degrees of freedom are equal to the number of observations (here N) minus the number of parameters (here 1 and 2 for the reduced and the full model respectively). Note also that $r^2/(1 - r^2)$ is nothing else than the ratio of the explained variation and the unexplained variation. The $\mathcal{F}$-statistic tests thus whether the explained variation is significantly higher than the unexplained variation (for given degrees of freedom).

APPENDIX C

Phosphene Classification Results

In this appendix, the confusion matrices resulting from the classification of the phosphenes for different combinations of the electrode contacts are reported. The density models are all constructed with variational GMM estimators.

TABLE C.1. Empirical confusion matrix when the 4 electrode contacts are activated.

	0°	90°	180°	270°
0°	**0.37**	0.29	0.16	0.18
90°	0.18	**0.34**	0.26	0.22
180°	0.17	0.16	**0.56**	0.11
270°	0.09	0.14	0.13	**0.64**

TABLE C.2. Empirical confusion matrices when 3 electrode contacts are activated.

	0°	90°	180°
0°	**0.51**	0.31	0.18
90°	0.27	**0.44**	0.29
180°	0.20	0.21	**0.59**

	0°	90°	270°
0°	**0.56**	0.20	0.24
90°	0.25	**0.62**	0.13
270°	0.17	0.14	**0.69**

	0°	180°	270°
0°	**0.42**	0.38	0.20
180°	0.23	**0.53**	0.24
270°	0.10	0.19	**0.71**

	90°	180°	270°
90°	**0.44**	0.34	0.22
180°	0.19	**0.68**	0.13
270°	0.18	0.15	**0.67**

TABLE C.3. Empirical confusion matrices when 2 electrode contacts are activated.

	0°	90°
0°	**0.58**	0.42
90°	0.31	**0.69**

	0°	180°
0°	**0.77**	0.23
180°	0.33	**0.67**

	0°	270°
0°	**0.74**	0.26
270°	0.24	**0.76**

	90°	180°
90°	**0.64**	0.36
180°	0.29	**0.71**

	90°	270°
90°	**0.74**	0.26
270°	0.25	**0.75**

	180°	270°
180°	**0.83**	0.17
270°	0.25	**0.75**

Bibliography

Abbas, H. M. and Fahmy, M. M. (1994). Neural networks for maximum likelihood clustering, *Signal Processing* **36**(1): 111–126. 3.2.1

Abramson, I. (1982). On bandwidth variation in kernel estimates - a square root law, *Annals of Statistics* **10**: 1217–1223. 2.3.2, 2.3.2

Ahalt, S. C., Krishnamurthy, A. K., Chen, P. K. and Melton, D. E. (1990). Competitive learning algorithms for vector quantization, *Neural Networks* **3**(3): 277–290. 2.3.3, 4.1.2

Akaike, H. (1954). An approximation to the density function, *Annals of the Institute of Statistical Mathematics* pp. 127–132. 2.2.2

Akaike, H. (1973). Information theory as an extension of the maximum likelihood principle, *in* B. N. Petrov and F. Csaki (eds), *International Symposium on Information Theory*, pp. 267–281. 2.1.2

Ali, S. M. and Silvey, D. (1966). A general class of coefficients of divergence of one distribution from another, *Journal of the Royal Statistical Society B* **28**(1): 131–142. 2.1.3

Archambeau, C., Delbeke, J., Veraart, C. and Verleysen, M. (2004). Prediction of visual perceptions with artificial neural networks in a visual prosthesis for the blind, *Artificial Intelligence in Medicine* **32**(3): 183–194. 5.2, 5.2.2

Archambeau, C., Delbeke, J. and Verleysen, M. (2003). Classification of visual sensations generated electrically in the visual field of the blind, *in* D. D. Feng and E. R. Carson (eds), *Fifth IFAC symposium on Modelling and Control in Biomedical Systems*, pp. 223–228. 5.3

Archambeau, C., Lee, J. A. and Verleysen, M. (2003). On the convergence problems of the EM algorithm for finite Gaussian mixtures, *Eleventh European Symposium on Artificial Neural Networks*, pp. 99–106. 3.2.1, 3.2.4, 3.3.2

Archambeau, C., Lendasse, A., Trullemans, C., Veraart, C., Delbeke, J. and Verleysen, M. (2001). Phosphene evaluation in a visual prosthesis with artificial neural networks, *First European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems*, pp. 509–515. 5.2.2

Archambeau, C. and Verleysen, M. (2003). Fully nonparametric probability density function estimation with finite Gaussian mixture models, *in* D. P. Mukherjee and S. Pal (eds), *Fifth International Conference on Advances in Pattern Recognition*, pp. 81–84. 3.2.1, 3.2.2, 3.2.4

Archambeau, C. and Verleysen, M. (2005a). Manifold constrained finite Gaussian mixtures, *in* J. Cabestany, A. Prieto and F. Sandoval (eds), *Computational Intelligence and Bioinspired Systems*, Vol. 3512 of *Lecture Notes in Compter Science*, Springer, pp. 820–828. 3.4, 3.4.2

Archambeau, C. and Verleysen, M. (2005b). Manifold constrained variational mixtures, *in* W. Duch, J. Kacprzyk, E. Oja and S. Zadrozny (eds), *Artificial Neural Networks: Formal Models and Their Applications*, Vol. 3697 of *Lecture Notes in Compter Science*, Springer, pp. 279–284. 3.4, 3.4.3

Archambeau, C., Vrins, F. and Verleysen, M. (2004). Flexible and robust Bayesian classification by finite mixture models, *Twelfth European Symposium on Artificial Neural Networks*, pp. 75–80. 3.2.1, 3.3.2

Attias, H. (1999a). Inferring parameters and structure of latent variable models by variational Bayes, *in* K. B. Laskey and H. Prade (eds), *Fifteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kauffman, pp. 21–30. 3.2.5

Attias, H. (1999b). A variational Bayesian framework for graphical models, *in* S. A. Solla, T. K. Leen and K.-R. Müller (eds), *Advances in Neural Information Processing Systems 12*, MIT Press, pp. 209–215. 3.2.1, 3.2.6

Babich, G. A. and Camps, O. I. (1996). Weighted Parzen windows for pattern classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(5): 567–570. 2.3.3, 2.5

Bach-Cuadra, M., Platel, B., Solanas, E., Butz, T. and Thiran, J. (2002). Validation of tissue modelization and classification techniques in t1-weighted MR brain images, *Fifth International Conference on Medical Image Computing and Computer Assisted Intervention*, Vol. 2489 of *Lecture Notes in Computer Science*, Springer, pp. 290–297. 3.2

Bak, M., Girvin, J. P., Hambrecht, F. T., Kufta, C. V., Loeb, G. E. and Schmidt, E. M. (1990). Visual sensations produced by intracortical microstimulation of the human occipital cortex, *Medical and Biological Engineering and Computing* **28**(3): 257–259. 5.1.1, 5.1.3

Basseville, M. (1989). Distance measures for signal processing and pattern recognition, *Signal Processing* **18**: 349–369. 2.1.3

Bauer, H. U., Der, R. and Herrmann, M. (1996). Controlling the magnification factor of self-organizing feature maps, *Neural Computation* **8**(4): 757–771. 2.3.3

Baum, L. E., Petrie, T., Soules, G. and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Annals of Mathematical Statistics* **41**: 164–171. 3.1

Beal, M. J. (2003). *Variational Algorithms for Approximate Bayesian Inference*, PhD thesis, Gatsby Computational Neuroscience Unit, University College London. 3.1, 3.1.2, 3.1.3

Bechtel, Y. C., Bonaïti-Pellié, C., Poisson, N., Magnette, J. and Bechtel, P. R. (1993). A population of family study of N-acetyltransferase using caffeine urinary metabolites, *Clinical Pharmacology and Therapeutics* **54**: 134–141. A

Bellman, R. (1961). *Adaptive control processes: a guided tour*, Princeton University Press, New Jersey. 2.1.1

Bengio, Y. and Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation, *Journal of Machine Learning Research* **5**: 1089–1105. 2.1.2

Benoudjit, N., Archambeau, C., Lendasse, A., Lee, J. A. and Verleysen, M. (2002). Width optimization of the Gaussian kernels in radial basis function networks, *Tenth European Symposium on Artificial Neural Networks*, pp. 425–432. 4.1.2

Berger, J. (1985). *Statistical Decision Theory and Bayesian Analysis.*, Springer-Verlag, New York. 4, 4.2.2, 4.2.3

Bernstein, M., de Silva, V., Langford, J. and Tenenbaum, J. (2000). Graph approximations to geodesics on embedded manifolds, *Technical report*, Stanford University. 3.4.1

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbor" meaningful?, *in* C. Beeri and P. Buneman (eds), *Seventh International Conference Database Theory*, Vol. 1540 of *Lecture Notes in Computer Science*, Springer, pp. 217–235. 2.3.1

Bishop, C. M. (1995). *Neural networks for pattern recognition*, Oxford university press. 2.1.1, 3, 3.1.2, 4.2.4

Bishop, C. M. (1999). Variational principal components, *Ninth International Conference on Artificial Neural Networks*, Vol. 1, IEE, pp. 509–514. 3.4.4

Bishop, C. M. and Tipping, M. E. (2000). Variational relevance vector machines, *in* C. Boutilier and M. Goldszmidt (eds), *Sixth Conference on Uncertainty in Artificial Intelligence*, Morgan Kauffmann, pp. 46–53. 4.2.3, 4.2.3

Borgelt, C. and Kruse, R. (2004). Shape and size regularization in expectation maximization and fuzzy clustering, *in* J.-F. Boulicaut, F. Esposito, F. Giannotti and D. Pedreschi (eds), *Knowledge Discovery in Databases*, Vol. 3202 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 52–65. 3.2.1, 3.2.2

Bowman, A. W. (1984). An alternative method of cross-validation for the smoothing of density estimates, *Biometrika* **71**: 153–176. 2.2.2

Breiman, L. (1996). Bagging predictors, *Machine Learning* **24**(2): 123–140. 3.2.2, 5.2.2

Breiman, L., Meisel, W. and Purcell, E. (1977). Variable kernel estimates of multivariate densities, *Technometrics* **19**: 135–144. 2.3.2, 2.3.2

Brelén, M. E., Duret, F., Gérard, B., Delebeke, J. and Veraart, C. (2005). Creating a meaningful visual perception in blind volunteers by optic nerve stimulation, *Journal of Neural Engineering* **2**: S22–S28. 5.4

Brindley, G. S. and Lewin, W. S. (1968). The sensations produced by electrical stimulation of the visual cortex, *Journal of Physiology* **196**: 479–493. 5.1.1

Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks, *Complex Systems* **2**: 321–355. 2.1.1, 4

Calcoullos, T. (1966). Estimation of a multivariate density, *Annals of the Institute of Statistical Mathematics* **18**: 179–189. 2.2.2

Cao, R., Cuevas, A. and Manteiga, W. G. (1994). A comparative study of several smoothing methods in density estimation, *Computational Statistics and Data Analysis* **17**: 153–176. 2.4, 2.4.3

Celeux, G., Chaveau, D. and Diebolt, J. (1996). Stochastic versions of the EM algorithm: an experimental study in the mixture case, *Journal of Statistical Computation and Simulation* **55**: 287–314. 3.2.1

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem, *Computational Statistics Quarterly* **2**: 73–82. 3.2.1

Chen, S., Cowan, C. F. N. and Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Transactions on Neural Networks* **2**(2): 302–309. 4.1.2

Chen, Z. and Haykin, S. (2002). On different facets of regularization theory, *Neural Computation* **14**: 2791–2846. 3.1.2

Cheng, B. and Titterington, D. M. (1994). Neural networks - a review from a statistical perspective, *Statistical Science* **9**(1): 2–30. 2

Chow, A. Y. and Chow, V. Y. (1997). Subretinal electrical stimulation of the rabbit retina, *Neuroscience letters* **28**(1): 13–16. 5.1.2

Chu, W., Keerthi, S. S. and Ong, C. J. (2004). Bayesian support vector regression using a unified loss function, *IEEE Transactions on Neural Networks* **15**(1): 29–44. 4.2.4

Clark, G. M., McAnally, K. I., Black, R. C. and Shepherd, R. K. (1995). Electrical stimulation of residual hearing in the implanted cochlea, *The Annals of otology, rhinology, and laryngology (Suppl.)* **166**: 111–113. 5

Corduneanu, A. and Bishop, C. M. (2001). Variational Bayesian model selection for mixture distributions, *in* T. Jaakkola and T. Richardson (eds), *Artificial Intelligence and Statistics 8*, Morgan Kaufmann, pp. 27–34. 3.2.6

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*, John Wiley and Sons, New York. 2.1.3, 3.1.1

Cowell, R. G., Dawid, A. P., Lauritzen, S. L. and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*, Springer, New York. 3.1

Crawford, S. L., Groot, M. H. D., Kadane, J. B. and Small, M. J. (1992). Modeling lake chemistry distributions: approximate Bayesian methods for estimating a finite mixture model, *Technometrics* **34**: 441–453. A

Cray, J. W., Allen, R. L., Stuart, A., Hudson, S., Layman, E. and Givens, G. D. (2004). An investigation of telephone use among cochlear implant recipients, *American journal of Audiology* **13**(2): 200–212. 5

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*, Cambridge University Press, Cambridge. 2.1.1, 2.3.4, 4, 4.2

de Marneffe, M.-C., Archambeau, C., Dupont, P. and Verleysen, M. (2004). Local vector-based models for sense discrimination, *in* H. Bunt, J. Geertzen and E. Thijsse (eds), *Sixth International Workshop on Computational Semantics*, pp. 163–174. 3.2

Delbeke, J., Oozeer, M. and Veraart, C. (2003). Position, size and luminosity of phosphenes generated by direct optic nerve stimulation, *Vision Research*

43(9): 1091–1102. 5.2, 5.2.1

Delbeke, J., Parrini, S., Michaux, G., Vanlierde, A. and Veraart, C. (2000). Perception threshold changes in phosphenes generated by direct stimulation of a human optic nerve, *Fifth Annual Conference of the International Functional Electrical Stimulation Society*, pp. 152–155. 5.2.1, 5.2.1, 5.2.3

Delbeke, J., Pins, D., Michaux, G., Wanet-Defalque, M. C., Parrini, S. and Veraart, C. (2001). Electrical stimulation of anterior visual pathways in retinitis pigmentosa, *Investigative Ophthalmology and Visual Science* **42**(1): 291–297. 5.1.3, 5.2.1

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion), *Journal of the Royal Statisitcal Society B* **39**: 1–38. 3.1, 3.1.1, 3.1.1

Dijkstra, E. W. (1959). A note on two problems in connection with graphs, *Numerical Mathematics* **1**: 269–271. 3.4.1

Ding, Y. and Marotte, L. R. (1997). Retinotopic order in the optic nerve and superior colliculus during development of the retinocollicular projection in the wallaby (Macropus eugenii), *Anatomy and Embryology* **196**(2): 141–158. 5.1.3

Dobelle, W. H. and Mladejovsky, M. G. (1974). Phosphenes produced by electrical stimulation of human occipital cortex, and their application to the development of a prosthesis for the blind, *Journal of Physiology* **243**: 533–576. 5.1.1

Doguet, P., Mevel, H., Verleysen, M., Troosters, M. and Trullemans, C. (2000). An integrated circuit for the electrical stimulation of the optic nerve, *Fifth Annual Conference of the International Functional Electrical Stimulation Society*, pp. 18–20. 5.1.3

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*, John Wiley and Sons. 3.2.1, A

Eckmiller, R. (1997). Learning retina implants with epiretinal contacts, *Ophthalmic Research* **29**(5): 281–289. 5.1.2

Efron, B. (1979). Bootstrap methods: another look at the Jackknife, *Annals of Statistics* **7**: 1–26. 2.1.2

Efron, B. (1983). Estimation of the error rate of a prediction rule: improvement on cross-validation, *Journal of the American Statistical Association* **78**(382): 316–331. 2.1.2, 2.1.2

Efron, B. (2003). Second thoughts on the bootstrap, *Statistical Science* **18**(2): 135–140. 2.1.2

Efron, B. and Tibshirani, R. (1997). Improvements on cross-validation: the .632+ bootstrap method, *Journal of the American Statistical Association* **92**(438): 548–560. 2.1.2

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*, Chapman and Hall, London. 2.1.2, 2.1.2

Epanechnikov, V. A. (1969). Nonparametric estimation of a multivariate probability density, *Theory of Probability and Its Applications* **14**: 153–158. 2.2.2, 2.2.2

Evgeniou, T., Pontil, M. and Poggio, T. (2000). Regularization networks and support vector machines, *Advances in Computational Mathematics* **13**: 1–50. 4

Farmen, M. and Marron, J. S. (1999). An assessment of finite sample performance of adaptive methods in density estimation, *Computational Statistics and Data Analysis* **30**: 143–168. 2.4

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems, *Annals of Statistics* **1**(2): 209–230. 3.2.4

Figueiredo, M. A. T. and Jain, A. K. (2002). Unsupervised learning of finite mixture models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(3): 381–396. 3.2.6

Figueiredo, M. T. (2003). Adaptive sparseness for supervised learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **9**(25): 1150–1159. 4.2.5, 4.2.5

Fix, E. and Hodges, J. L. (1951). Discriminatory analysis, nonparametric estimation: consistency properties, *Technical report*, Project 21-49-004 (Report no. 4), Randolph Field, Texas: USAF School of Aviation Medicine. 2.3.1

Foerster, O. (1929). Beitrage zur pathophysiolgie der sehbahn und der spehphare, *Journal of Psychology and Neurology* pp. 435–463. 5

Fukunaga, K. (1972). *Introduction to Statistical Pattern Recognition*, Academic Press. 2

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1998). *Bayesian Data Analysis*, Chapman and Hall, London. 3.2.3, 3.2.3, 3.2.4

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(1): 721–741. 3.1

Ghahramani, Z. and Beal, M. J. (1999). Variational inference for Bayesian mixtures of factor analysers, *in* S. A. Solla, T. K. Leen and K.-R. Müller (eds), *Advances in Neural Information Processing Systems 12*, MIT Press, pp. 449–455. 3.4.4

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*, Chapman and Hall. 3.1

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling, *Applied Statistics* **41**(2): 337–348. 3.1

Girolami, M. and He, C. (2003). Probability density estimation from optimally condensed data samples, *IEEE Transactions Pattern Analysis and Machine Intelligence* **25**(10): 1253–1264. 2.3.4, 2.3.4

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives, *Journal of the Royal Statistical Society B* **46**: 149–192. 3.3.1

Green, P. J. (1990). On the use of the EM algorithm for penalized likelihood estimation, *Journal of the Royal Statisitcal Society B* **52**(3): 443–452. 3.1.2, 3.1.2

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, *Biometrika* **82**: 711–732. 3.2.6

Green, P. J. (1999). Penalized likelihood, *Encyclopaedia of Statistical Sciences* **3**: 578–586. 3.1.2

Grossberg, S. (1987). Competitive learning - From interactive activation to adaptive resonance, *Cognitive Science* **11**(1): 23–63. 2.3.3, 4.1.2

Gstoettner, W. K., Hamzavi, J., Egelierler, B. and Baumgartner, W. D. (2000). Speech perception performance in prelingually deaf children with cochlear implants, *Acta Oto-Laryngologica* **120**(2): 209–213. 5

Gull, S. F. (1989). Developments in maximum entropy data analysis, *in* J. Skillilng (ed.), *Maximum Entropy and Baysian Methods 9*, pp. 53–57. 4.2.3

Gupta, L. and Sortrakul, T. (1998). A Gaussian-mixture-based image segmentation algorithm, *Pattern recognition* **31**(3): 315–325. 3.2

Hall, P., Hui, T. and Marron, J. (1995). Improved variable window kernel estimates of probability densities, *Annals of Statistics* **23**(1): 1–10. 2.3.2

Härdle, W., Müller, M., Sperlich, S. and Werwatz, A. (2004). *Nonparametric and semi-parametric models*, Springer, New York. 2, 2.2.1, 2.2.2

Härdle, W. and Simar, L. (2003). *Applied Multivariate Statistical Analysis*, Springer, New York. B

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic prices and the demand for clean air, *Journal of Environmental Economics and Management* **5**: 81–102. A

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**: 97–109. 3.1

Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation*, Prentice-Hall. 4, 4.1.2

Heskes, T. (2002). Stable fixed points of loopy belief propagation are local minima of the Bethe free energy, *in* S. T. S. Becker and K. Obermayer (eds), *Advances in Neural Information Processing Systems 15*, MIT Press, pp. 343–350. 3.1

Heskes, T., Zoeter, O. and Wiegerinck, W. (2003). Approximate expectation maximization, *in* S. Thrun, L. Saul and B. Schölkopf (eds), *Advances in Neural Information Processing Systems 16*, MIT Press. 3.1.1

Hill, A. V. (1936). The strength-duration relation for electric excitation of medullated nerve, *Proceedings of the Royal Society of London, Series B* (815): 440–453. 5.1.3

Hinneburg, A., Aggarwal, C. C. and Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces?, *in* A. E. Abbadi, M. L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter and K.-Y. Whang (eds), *Twentysixth International Conference on Very Large Data Bases*, Morgan Kaufmann, pp. 506–515. 2.3.1

Hinton, G. E. and van Camp, D. (1993). Keeping neural networks simple by minimizing the description length of the weights, *Sixth Annual Conference on Computational Learning Theory*, pp. 5–13. 3.1

Hodgkin, A. and Huxley, A. (1952). A quantitative description off mebrane current and its application to conduction and excitation in nerve, *Journal of Physiology* pp. 500–544. 5.1.3

Hoerl, A. and Kennard, R. (1970). Ridge regression: biased estimation for nonorthogonal problems, *Technometrics* **12**: 55–67. 4.1.1

Holmström, L. (2000). The error and the computational complexity of a multivariate binned kernel density estimator, *Journal of Multivariate Analysis* **72**(2): 264–309. 2.3.2

Holmström, L. and Hämäläinen, A. (1993). The self-organizing reduced kernel density estimator, *in* J. E. Gentle (ed.), *IEEE International Conference on Neural Networks*, IEEE press, pp. 417–421. 2.3.3

Horton, J. C. and Hoyt, W. F. (1991). The representation of the visual-field in human striate cortex - a revision of the classic Holmes map, *Archives of Ophthalmology* **109**(6): 816–824. 5.1.1

Hubel, D. H. and Wiesel, T. N. (1974). Uniformity of monkey striate cortex: a parallel relationship between field size, scatter, and magnification factor, *Journal of Comparative Neurology* **158**: 295–305. 5.1.1

Humayun, M. S., de Juan, E., Dagnelie, G., Greenberg, R. J., Prost, R. H. and Phillips, P. H. (1996). Visual perception elicited by electrical stimulation of the retina in blind humans, *Archives of Ophthalmology* **114**(1): 40–46. 5.1.2

Humayun, M. S., de Juan, E., Weiland, J. D., Dagnelie, G., Katona, S., Greenberg, R. J. and Suzuki, S. (1999). Pattern electrical stimulation of the human retina, *Vision Research* **39**(15): 2569–2576. 5.1.2

Humayun, M. S., Propst, R., de Juan E, E., McCormick, K. and Hickingbotham, D. (1994). Bipolar surface electrical stimulation of the vertebrate retina, *Archives of Ophthalmology* **112**(1): 110–116. 5.1.2

Hwang, J.-N., Lay, S.-R. and Lippman, A. (1994). Nonparametric multivariate density estimation: a comparative study, *IEEE Transactions on Signal Processing* **42**(10): 2795–2810. 2.2.2, 2.3.2, 2.3.3, 2.3.3, 2.4, 2.4.2

Izenman, A. L. (1991). Recent developments in nonparametric density estimation, *Journal of the American Statistical Association* **86**(413): 205–224. 2, 2.1.3

Jaakkola, T. (1997). *Variational Methods for Inference and Estimation in Graphical Models*, PhD thesis, Massachusetts Institute of Technology. 3.1

Jain, A. K., Duin, R. P. W. and Mao, J. C. (2000). Statistical pattern recognition: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(1): 4–37. 2, 2.1.1

Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inegalités entre les valeurs moyennes, *Acta Mathematica* pp. 175–193. 3.1.1, 4.2.3

Jollife, I. T. (1986). *Principal Component Analysis*, Springer-Verlag, New York. 2.4.3

Jones, M. C., Marron, J. S. and Sheather, S. J. (1996). A brief survey of bandwidth selection for density estimation, *Journal of the American Statistical Association* **91**(433): 401–407. 2.2.2

Jordan, M. I. (2004). Graphical models, *Statistical Science* **19**: 140–155. 3.1

Jordan, M. I. (ed.) (1999). *Learning in Graphical Models*, MIT Press, Cambridge. 3.1

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. and Saul, L. K. (1999). An introduction to variational methods for graphical methods, *Machine Learning* **37**(2): 183–233. 3.1

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm, *Neural Computation* **6**: 181–214. 3.1

Kent, J. T., Tyler, D. E. and Vardi, Y. (1994). A curious likelihood identity for the multivariate t-distribution, *Communications in Statistics – Simulation and Computation* **23**(2): 441–453. 3.3.1, 3.3.1, 3.3.5

Kim, S. Y., Sadda, S., Pearlman, J., Humayun, M. S., de Juan E, E., Melia, B. M. and Green, W. R. (2002). Morphometric analysis of the macula in eyes with disciform age-related macular degeneration, *Retina* **22**(4): 471–477. 5.1.2

Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and their Applications*, American Mathematical Society, Providence. 3.1

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection, *International Joint Conference on Artificial Intelligence*, pp. 1137–1145. 2.1.2, 2.1.2

Kohonen, T. (1995). *Self-organizing maps*, Springer, Berlin. 2.3.3, 4.1.2

Kullback, S. and Leibler, R. (1951). On information and sufficiency, *Annals of Mathematical Statistics* (1): 79–86. 2.1.3, 3.1.1

Lapique, L. (1907). Recherches quantitatives sur l'excitation électrique des nerfs, traitée comme un polarisation, *Journal of Physiology* pp. 622–635. 5.1.3

Law, M. H. and Kwok, J. (2001). Bayesian support vector regression, *in* T. Jaakkola and T. Richardson (eds), *Artificial Intelligence and Statistics 8*, Morgan Kaufmann, pp. 239–244. 4.2.4, 4.2.4

Lee, J. A. (2004). *From Prinicpal Component Analysis to Non-linear Dimensionality Reduction and Blind Source Separation*, PhD thesis, Electrical Engineering Department, Université catholique de Louvain. 3.4.1

Lee, J. A., Lendasse, A. and Verleysen, M. (2003). Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis, *Neurocomputing* **57**: 49–76. 3.4.1

Li, J. Q. and Barron, A. R. (1999). Mixture density estimation, *in* S. A. Solla, T. K. Leen and K.-R. Müller (eds), *Advances in Neural Information Processing Systems 12*, MIT Press. 3.2.1

Liu, C.-B. and Ahuja, N. (2004). Vision based fire detection, *Seventeenth Intenational Conference on Pattern Recognition*, pp. 414–417. 3.2

Liu, C. and Rubin, D. B. (1995). ML estimation of the t distribution using EM and its extensions, ECM and ECME, *Statistica Sinica* **5**: 19–39. 3.3.1, 3.3.1

Liu, W. T., Vichienchom, K., Clements, M., DeMarco, S. C., Hughes, C., McGucken, E., MS, M. S. H., de Juan, E., Weiland, J. D. and Greenberg, R. J. (2000). A neurostimulus chip with telemetry unit for retinal prosthetic device, *IEEE Journal of Solid-State Circuits* **35**(10): 1487–1497. 5.1.2

Loftsgaarden, D. O. and Quesenberry, C. P. (1965). A nonparametric estimate of a multivariate density function, *Annals of Mathematical Statistics* **36**: 1065–1076. 2.3.1

Lunn, D. J., Thomas, A., Best, N. G. and Spiegelhalter, D. J. (2000). Winbugs - a Bayesian modelling framework: concepts, structure and extensibility, *Statistics*

and Computing **10**: 321–333. 3.1

MacKay, D. J. C. (1992a). Bayesian interpolation, *Neural Computation* **4**(3): 415–447. 4, 4.2.3, 4.2.3

MacKay, D. J. C. (1992b). A practical Bayesian framework for backpropagation networks, *Neural Computation* **4**(3): 448–472. 3.1.3, 4, 4.2.4, 4.2.4

MacQueen, J. (1967). Some methods of classification and analysis of multivariate observations, *in* L. LeCam and J. Neyman (eds), *Fifth Berkeley Symposium on Mathematical Statistics and Probabilities*, pp. 281–297. 2.3.3, 3.2.1, 4.1.2

Mao, J. and Jain, A. K. (1996). A self-organizing network for hyperellipsoidal clustering (HEC), *IEEE Transactions on Neural Networks* **7**(1): 16–29. 3.2.2, 3.2.4

Margalit, E., Maia, M., Weiland, J. D., Greenberg, R. J., Fujii, G. Y., Torres, G., Piyathaisere, D. V., O'Hearn, T. M., Liu, W. T., Lazzi, G., Dagnelie, G., Scribner, D. A., de Juan, E. and Humayun, M. S. (2002). Retinal prosthesis for the blind, *Survey of Ophthalmology* **47**(4): 335–356. 5, 5.1.2, 5.1.3

Markatou, M. (2000). Mixture models, robustness, and the weighted likelihood methodology, *Biometrics* **56**: 483–486. 3.3.1

Markatou, M., Basu, A. and Lindsay, B. G. (1998). Weighted likelihood estimating equations with a bootstrap root search, *Journal of the American Statistical Association* **93**: 740–750. 3.3.1

Martinetz, T. M., Berkovich, S. G. and Schulten, K. J. (1993). Neural-gas network for vector quantization and its application to time-series prediction, *IEEE Transactions on Neural Networks* **4**(4): 558–569. 2.3.3, 4.1.2

Maynard, E. M. (2001). Visual prostheses, *Annual Reviews in Biomedical Engineering* **3**: 145–168. 5, 5.1.2, 5.1.3

McDermott, H. J. (2004). Music perception with cochlear implants: a review, *Trends in amplification* **8**(2): 49–82. 5

McLachlan, G. J. and Bashford, K. E. (1988). *Mixture Models: Inference and Applications to Clustering*, M. Dekker, New York. 3.1

McLachlan, G. J. and Krishnan, T. (1997). *The EM Algorithm and Extensions*, Wiley, New York. 3.1

McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*, John Willey and Sons, New York. 3, 3.2, 3.2.1

Meister, M. and Berry, M. J. (1999). The neural code of the retina, *Neuron* **22**: 435–450. 5

Meng, X.-L. and van Dyk, D. (1997). The EM algorithm - an old folk-song sung to a fast new tune, *Journal of the Royal Statistical Society B* **59**(3): 511–567. 3.2.1, 3.3.1

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equations of state calculations by fast computing machines, *Journal of Chemical Physics* pp. 1087–1092. 3.1

Micchelli, C. A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constructive Approximations* **2**: 11–22. 4.1

Minagawa, A., Tagawa, N. and Tanaka, T. (2002). SMEM algorithm is not fully compatible with maximum-likelihood framework, *Neural Computation* **14**: 1261–1266. 3.2.1

Moody, J. E. and Darken, C. (1989). Fast learning in networks of locally-tuned processing units, *Neural Computation* **1**: 281–294. 2.1.1, 4, 4.1.2, 4.1.2

Naples, G. G., Mortimer, J. T., Scheiner, A. and Sweeney, J. D. (1988). A spiral nerve cuff electrode for peripheral nerve stimulation, *IEEE Transactions on Biomedical Engineering* **35**(11): 905–916. 5.1.3

Neal, R. M. (2003). Slice sampling (with discussion), *Annals of Statistics* **31**: 705–767. 3.1

Neal, R. M. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants, *in* M. I. Jordan (ed.), *Learning in Graphical Models*, Kluwer Academic Publishers, pp. 355–368. 3.1.1

Nityasuddhi, D. and Böhning, D. (2003). Asymptotic properties of the EM algorithm estimate for normal mixture models with component specific variances, *Computational Statistics and Data Analysis* **41**: 591–601. 3.2.1

Normann, R. A., Maynard, E. M., Guillory, K. S. and Warren, D. J. (1996). Cortical implants for the blind, *IEEE Spectrum* **33**(5): 54–59. 5.1.1

Ormoneit, D. and Tresp, V. (1998). Averaging, maximum penalized likelihood and Bayesian estimation for improving Gaussian mixture probability density estimates, *IEEE Transactions on Neural Networks* **9**(4): 639–649. 3.2.1, 3.2.2, 3.2.3, 3.2.4

Park, B. U. and Turlach, B. A. (1992). Practical performance of several data driven bandwidth selectors, *Computational Statistics* **7**: 251–270. 2.4, 2.4.3

Park, J. and Sandberg, I. (1991). Universal approximation using radial basis function networks, *Neural Computation* **3**: 246–257. 4.1.2

Parrini, S., Delbeke, J., Legat, V. and Veraart, C. (2000). Modelling analysis of human optic nerve fibre excitation based on experimental data, *Medical and Biolgical Engineering and Computing* **38**: 454–464. 5.1.3, 5.2.3

Parzen, E. (1962). On estimation of a probability density function and mode, *Annals of Mathematical Statistics* **33**: 1065–1076. 2.2.2

Pearl, J. (1986). Fusion, propagation and structuring in belief networks, *Artifcial Intelligence* **29**: 241–288. 3.1

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, San Francisco. 3.1

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution, *Statistics and Computing* **10**: 339–348. 3.3.1, 3.3.1

Platt, J. (1999). *Advances in Kernel Methods - Support Vector Learning*, MIT Press, Cambridge, chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208. 2.3.4

Poggio, T. and Girosi, F. (1990). Regularization algorithms for learning that are equivalent to multilayer networks, *Science* **317**: 314–319. 4.1

Powell, M. J. D. (1987). *Algorithms for Approximations*, Clarendon Press, Oxford, chapter Radial basis functions for multivariable interpolation: a review, pp. 143–167. 4.1

Priebe, C. E. (1994). Adaptive mixtures, *Journal of the American Statistical Association* **89**(427): 796–806. 3

Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood, and the EM algorithm, *Siam Review* **26**: 195–239. 3, 3.2, 3.2.1

Richardson, S. and Green, P. (1997). On Bayesian analysis of mixtures with unknown number of components, *Journal of the Royal Statistical Society B* **59**: 731–792. 3.2.6

Rizzo, J. F. and Wyatt, J. (1997). Prospects for a visual prosthesis, *The Neuroscientist* **3**(4): 251–262. 5.1.2

Roberts, S., Husmeier, D., Rezek, I. and Penny, W. (1998). Bayesian approaches to Gaussian mixture modeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(11): 1133–1142. 3.2.6

Roeder, K. (1990). Density estimation with confidence sets exemplified by superclusters and voids in the galaxies, *Journal of the American Statistical Association* **85**: 617–624. A

Roeder, K. and Wasserman, L. (1997). Practical Bayesian density estimation using mixtures of normals, *Journal of the American Statistical Association* **92**: 894–902. 3.2.6

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density, *Annals of Mathematical Statistics* pp. 832–837. 2.2.2

Roweis, S., Saul, L. and Hinton, G. (2001). Global coordination of local linear models., *in* T. Dietterich, S. Becker and Z. Ghahramani (eds), *Advances in Neural Information Processing Systems 14*, MIT Press, pp. 889–896. 3.4.4

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding, *Science* **290**: 1323–2326. 3.4.1

Ruan, S., Jaggi, C., Xue, J.-H., Fadili, M.-J. and Bloyet, D. (2000). Brain tissue classification of magnetic resonance images using partial volume modeling, *IEEE Transactions on Medical Imaging* **19**(12): 1179–1187. 3.2

Rudemo, M. (1982). Empirical choice of histograms and kernel density estimators, *Scandinavian Journal of Statistics* **9**: 65–78. 2.2.2

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *Nature* **323**: 533–536. 2.1.1

Sain, S. R. and Scott, D. W. (1996). On locally adaptive density estimation, *Journal of the American Statistical Association* **91**: 1525–1534. 2.3.2

Santos, A., Humayun, M. S., de Juan, E., Greenberg, R. J., Marsh, M. J., Klock, I. B. and Milam, A. H. (1997). Preservation of the inner retina in retinitis pigmentosa - a morphometric analysis, *Archives of Ophthalmology* **115**(4): 511–515. 5.1.2

Schmidt, E. M., Bak, M. J., Hambrecht, F. T., Kufta, C. V., ORourke, D. and Vallabhanath, P. (1996). Feasibility of a visual prosthesis for the blind based on intracortical microstimulation of the visual cortex, *Brain, Part 2* **119**: 507–522. 5.1.1

Schroeter, P., Vesin, J.-M., Langenberger, T. and Meuli, R. (1998). Robust parameter estimation of intensity distribution for brain magnetic resonance imaging, *IEEE Transactions on Medical Imaging* **17**(2): 172–186. 3.2

Schwarz, G. (1978). Estimating the dimension of a model, *Annals of Statistics* **6**: 461–464. 2.1.2

Scott, D. W. (1985). Average shifted histograms: effective nonparametric density estimators in several dimensions, *Annals of Statistics* **13**: 1024–1040. 2.2.1

Scott, D. W. (1992). *Multivariate density estimation: theory, practice, and visualization*, John Wiley and Sons, New York. 2, 2.2.1, 2.2.2

Scott, D. W. and Szewczyk, W. F. (2001). From kernels to mixtures, *Technometrics* **43**: 323–335. 3

Scott, D. W. and Thompson, J. R. (1983). Probability density estimation in higher dimensions, *in* J. E. Gentle (ed.), *Computer Science and Statistics: Proceedings of the fifteenth Symposium on the Interface*, pp. 267–281. 2.1.1

Sha, F., Saul, L. K. and Lee, D. D. (2002). Multiplicative updates for nonnegative quadratic programming in support vector machines, *in* S. Becker, S. Thrun and K. Obermayer (eds), *Advances in Neural Information Processing Systems 15*, MIT Press, pp. 1041–1048. 2.3.4

Shannon, C. E. and Weaver, W. (1963). *Mathematical Theory of Communication*, University of Illinois Press, Urbana. 2.1.3

Shoham, S. (2002). Robust clustering by deterministic agglomeration EM of mixtures of multivariate t-distributions., *Pattern Recognition* **35**(5): 1127–1142. 3.3.1, 3.3.1

Silverman, B. W. (1986). *Density Estimation*, Chapman and Hall, London. 2, 2.1.1, 2.2.1, 2.2.2, 2.3.1, 2.3.1, 2.3.2, 2.3.2

Spiegelhalter, D. J. (1986). Probabilistic reasoning in predictive expert systems, *in* L. N. Kanal and J. F. Lemmer (eds), *Third conference on Uncertainty in Artifcial Intelligence*, pp. 47–68. 3.1

Stone, C. J. (1984). An asymptotically optimal window selection rule for kernel density estimates, *Annals of Statistics* **12**: 1285–1297. 2.2.2

Stone, J. L., Barlow, W. E., Humayun, M. S., de Juan, E. and Milam, A. H. (1992). Morphometric analysis of macular photoreceptors and ganglion cells in retinas with retinitis pigmentosa, *Archives of Ophthalmology* **110**(11): 1634–1639. 5.1.2

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society B* **36**: 111–147. 2.1.2

Stone, M. (1975). The predictive sample reuse method with applications, *Journal of the American Statistical Association* **70**: 320–328. 2.1.2

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion, *Journal of the Royal Statisitcal Society B* **38**: 44–47. 2.1.2

Sundberg, R. (1972). *Maximum Likelihood Theory and Applications for Distributions Generated when Observing a Function of an Exponential Family*, PhD thesis, Institute of Mathematics and Statistics, Stockholm University. 3.2

Sundberg, R. (1974). Maximum likelihood theory for incomplete data from an exponential family, **1**: 49–58. 3.2

Svensén, M. and Bishop, C. M. (2004). Robust Bayesian mixture modelling, *Neurocomputing* **64**: 235–252. 3.3.5, 3.3.5

Taylor, C. C. (1989). Bootstrap choice of the smoothing parameter in kernel density estimation, *Biometrika* **76**: 705–712. 2.2.2

Tenenbaum, J. B., de Silva, V. and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction, *Science* **290**: 2319–2323. 3.4.1

Terrel, G. R. and Scott, D. W. (1992). Variable kernel density estimation, *Annals of Statistics* **20**(3): 1236–1365. 2.3.1, 2.3.2

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO, *Journal of the Royal Statistical Association B* **58**(1): 267–288. 4.2.3

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of Ill-Posed Problems*, V. H. Winston, Washington DC. 3.1.2

Tipping, M. E. (1999). The relevance vector machine, *in* S. A. Solla, T. K. Leen and K.-R. Müller (eds), *Advances in Neural Information Processing Systems 12*, MIT Press, pp. 652–658. 4, 4.2.3

Tipping, M. E. (2001). Sparse Bayesian learning and the Relevance Vector Machines, *Journal of Machine Learning Research* **1**: 211–244. 4.2.3, 4.2.3, 4.2.3, 4.2.3

Tipping, M. E. and Bishop, C. M. (1999). Mixtures of probabilistic principal component analyzers, *Neural Computation* **11**(2): 443–482. 3.4.4

Tipping, M. E. and Faul, A. C. (2003). Fast marginal likelihood maximisation for sparse Bayesian models, *in* C. M. Bishop and B. J. Frey (eds), *Ninth International Workshop on Artificial Intelligence and Statistics*, MIT Press. 4.2.3

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data, *Journal of the Royal Statistical Association B* **46**(2): 257–267. 3.1

Ueda, N. and Nakano, R. (1998). Deterministic annealing EM algorithm, *Neural Networks* **11**: 271–282. 3.2.1, 3.3.5

Ueda, N., Nakano, R., Ghahramani, Z. and Hinton, G. E. (2000). SMEM algorithm for mixture models, *Neural Computation* **12**: 2109–2128. 3.2.1, 3.2.1

Vapnik, V. and Mukherjee, S. (1999). Support vector method for multivariate density estimation., *in* S. A. Solla, T. K. Leen and K.-R. Müller (eds), *Advances in Neural Information Processing Systems 12*, The MIT Press, pp. 659–665. 2.3.4

Vapnik, V. N. (1998). *Statistical learning theory*, John Wiley and Sons. 2.1.1, 2.3.4, 4

Veraart, C., Grill, W. M. and Mortimer, J. T. (1993). Selective control of muscle activation with a multipolar nerve cuff electrode, *IEEE Transactions on Biomedical Engineering* **40**(7): 640–653. 5.1.3

Veraart, C., Raftopoulos, C., Mortimer, J., Delbeke, J., Pins, D., Michaux, G., Vanlierde, A., Parrini, S. and Wanet-Defalque, M. (1998). Visual sensations produced by optic nerve stimulation using an implanted self-sizing spiral cuff electrode, *Brain Research* **813**: 181–186. 5, 5.1.3, 5.1.3

Veraart, C., Wanet-Delfalque, M.-C., Gerard, B., Vanlierde, A. and Delbeke, J. (2003). Pattern recognition with the optic nerve visual prosthesis, *Artificial Organs* **27**(11): 996–1004. 5.1.3, 5.2

Verbeek, J. J., Vlassis, N. A. and Kröse, B. J. A. (2002). Coordinating principal component analyzers., *in* J. R. Dorronsoro (ed.), *Thirteenth International Conference on Artificial Neural Networks*, Vol. 2415 of *Lecture Notes in Computer Science*, Springer, pp. 914–919. 3.4.4

Verbeek, J. J., Vlassis, N. and Kröse, B. (2003). Efficient greedy learning of Gaussian mixture models, *Neural Computation* **15**: 469–485. 3.2.1, 3.2.6, 3.2.6

Verleysen, M. and Hlavackova, K. (1996). Learning in RBF networks, *IEEE International Conference on Neural Networks*, IEEE press, pp. 199–204. 4.1.2

Vincent, P. and Bengio, Y. (2002). Manifold Parzen windows, *in* S. Becker, S. Thrun and K. Obermayer (eds), *Advances in Neural Information Processing Systems 15*, MIT Press, pp. 825–832. 3.4

Vlassis, N. and Likas, A. (2002). A greedy EM algorithm for Gaussian mixture learning, *Neural Processing Letters* **15**(1): 77–87. 3.2.1

Voz, J.-L., Verleysen, M. and Comon, P. (1995). A practical view of suboptimal Bayesian classification with radial Gaussian kernels, *in* J.Mira and F.Sandoval (eds), *From Natural to Artificial Neural Computation*, Vol. 930 of *Lecture Notes in Computer Science*, Springer, pp. 404–411. 2.3.3

Wallace, C. and Freeman, P. (1987). Estimation and inference via compact coding, *Journal of the Royal Statistical Society B* **49**(3): 241–252. 3.2.6

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*, Chapman and Hall, London. 2, 2.2.2

Warren, D. J., Fernandez, E. and Normann, R. A. (2001). High-resolution two-dimensional spatial mapping of cat striate cortex using a 100-microelectrode array, *Neuroscience* **105**(1): 19–31. 5.1.1

Waterhouse, S. R., MacKay, D. J. and Robinson, A. J. (1995). Bayesian methods for mixtures of experts, *in* D. S. Touretzky, M. Mozer and M. E. Hasselmo (eds), *Advances in Neural Information Processing Systems 8*, MIT Press, pp. 351–357. 3.1

West, D. B. (1996). *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River. 3.4.1

Williams, P. M. (1995). Bayesian regularization and pruning using a Laplace prior, *Neural computation* **7**(1): 117–143. 4.2.3

Winn, J. (2003). *Variational Message Passing and its Applications*, PhD thesis, Department of Physics, University of Cambridge. 3.1.2, 3.1.3, 3.2.6

Winn, J. and Bishop, C. (2005). Variational message passing, *Journal of Machine Learning Research* pp. 661–694. 3.1

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm, *Annals of Statistics* **11**: 95–103. 3.2.1

Wyatt, J. and Rizzo, J. F. (1996). Ocular implants for the blind, *IEEE Spectrum* **112**: 47–53. 5.1.2

Xu, L. and Jordan, M. I. (1996). On convergence properties of the EM algorithm for Gaussian mixtures, *Neural Computation* **8**: 129–151. 3.2.1

Yedidia, J., Freeman, W. and Weiss, Y. (2003). *Exploring Artifcial Intelligence in the New Millenium*, Morgan Kaufmann, chapter Understanding belief propagation and its generalizations. 3.1

Zador, P. L. (1982). Asymptotic quantization error of continuous signals and the quantization dimension, *IEEE Transactions on Information Theory* **28**: 139–149. 2.3.3

Zrenner, E. (2002). Will retinal implants restore vision?, *Science* **295**(5557): 1022–10025. 5, 5.1.2

Zrenner, E., Stett, A., Weiss, S., Aramant, R. B., Guenther, E., Kohler, K., Miliczek, K. D., Seiler, M. J. and Haemmerle, H. (1999). Can subretinal microphotodiodes successfully replace degenerated photoreceptors?, *Vision Research* **39**(15): 2555–2567. 5.1.2