Graph-based fraud detection with the free energy distance

Sylvain Courtain[1], Bertrand Lebichot[1,3], Ilkka Kivimäki[4], and Marco Saerens[1,2]

[1] LOURIM, Université catholique de Louvain, Belgium
[2] ICTEAM, Université catholique de Louvain, Belgium
[3] MLG, Université Libre de Bruxelles, Belgium
[4] Department of Computer Science, Aalto University, Finland

Abstract. This paper investigates a real-world application of the *free energy distance* between nodes of a graph [14, 20] by proposing an improved extension of the existing *Fraud Detection System* named APATE [36]. It relies on a new way of computing the free energy distance based on paths of increasing length, and scaling on large, sparse, graphs. This new approach is assessed on a real-world large-scale e-commerce payment transactions dataset obtained from a major Belgian credit card issuer. Our results show that the free-energy based approach reduces the computation time by one half while maintaining state-of-the art performance in term of Precision@100 on fraudulent card prediction.

Keywords: Credit card fraud detection, Network science, Network data analysis, Free energy distance, Semi-supervised learning.

1 Introduction

With the emergence of e-commerce systems, the number of online credit card transactions has skyrocketed. However, not all of these transactions are legitimate – worldwide card fraud losses in 2017 reached 24.26 billion US dollars, an increase of 6.4% from 2016, and the forecast for the following years goes in the same direction [8]. This huge amount of loss has led to the development of a series of countermeasures to limit the number of frauds. Among these countermeasures, Fraud Detection System (FDS) aims to identify perpetrated fraud as soon as possible [4].

The credit card fraud detection domain presents a number of challenging issues [1, 7]. Firstly, there are millions of credit card transactions processed each day, creating a massive stream of data. That is why data mining and machine learning play an important role in FDS, as they are often applied to extract and uncover the hidden truth behind very large quantities of data [26]. Secondly, the data are unbalanced: there is (fortunately) a more prevalent number of genuine transactions than fraudulent ones. The main risk with unbalanced classes is that the classifier tends to be overwhelmed by the majority class and to ignore the minority class [22]. Thirdly, the data are exposed to a concept drift as the habits and behaviors of the consumers and fraudsters change over time [9]. Finally, the FDS needs to process the acceptance check of an online credit-card transaction within a few seconds to decide whether to pursue the transaction or not [36].

This work focuses on automatically detecting fraudulent e-commerce transactions using network-related features and free energy distance [20]. Our work is based on a recent paper [21] which introduced several improvements to an existing collective inference algorithm called APATE [36]. More precisely, this algorithm starts from a defined number of known frauds and propagates the fraudulent influence through a graph to obtain a risk score, quantifying the fraudulent behavior for each transaction, cardholder, and merchant [36]. In short, the main contributions of this paper are:

- a new way of computing the free energy distance [20] scaling on large, sparse, graphs;
- an application of this method to the fraud detection field through the adaptation of the existing FDS APATE [36];
- an experimental comparison between this method and others on a large real-life e-commerce credit card transaction dataset obtained from Worldline SA/NV.

The remainder of the paper is organised as follows. Section 2 contains the related work. Section 3 introduces the proposed contributions. In Section 4, we present the experimental comparisons and analyse the results. Finally, Section 5 concludes the paper.

2 Related work

Over the past few years, credit card fraud detection has generated a lot of interest and a wide range of techniques has been suggested. However, the number of publications available is only the tip of the iceberg. Indeed, credit card issuers protect the sharing of data and most algorithms are produced in-house, concealing the details of the models [36].

Credit card fraud detection techniques can be seen as a classification problem. Therefore, it can be categorized into three broad types of learning: supervised (SL), unsupervised (USL) and semi-supervised (SSL). The most widespread approach is the supervised one which uses the information content in the labels, i.e. 'fraud' and 'genuine' in our case, to build a classification model. Common supervised methods are logistic regression [30], decision trees [30], Bayes minimum risk [2], support vector machines [10], meta-learning [7], case-based reasoning [38], genetic algorithms [12], hidden Markov models [3, 33], association rules [29], random forest [10] and, the most prevalent one for the moment, artificial neural networks [10, 18, 40]. Unlike supervised techniques, unsupervised learning does not use the class label to build the model, but simply extracts clusters of similar observations while maximizing the difference between these clusters. Common unsupervised methods include standard clustering methods, self-organizing map [39] and peer group analysis [37]. The interested reader is advised to consult [10, 26, 41] for more information.

The last category of classification techniques is semi-supervised learning which lies between supervised and unsupervised techniques, since it constructs predictive models using labeled samples together with a usually larger amount of unlabeled samples [13]. Some common semi-supervised methods are graph-based

approaches, which consist in the creation of a graph model that reflects the relations included in the data and then transfers the labels on the graph to build a classification model [41]. Compared to the two other categories presented before, there are few publications about semi-supervised methods applied to card fraud detection. Ramaki et al. [28] proposed a model on a semantic connection between data stored for every transaction fulfilled by a user, then represent it by ontology graph and finally store them in patterns databases. Cao et al. [6] suggested HitFraud, a collective fraud detection algorithm that captures the inter-transaction dependency based on a heterogeneous information network. Molloy et al. [24] presented a new approach to cross channel fraud detection based on feature extraction techniques applied to a graph of transactions. Finally, Lebichot et al. [21] proposed several improvements to an existing FDS, APATE, which spreads fraudulence influence through a graph by using a limited set of confirmed fraudulent transactions [36]. Our FDS is of this type.

As mentioned earlier, we base our approach on the previous work of Lebichot et al. [21] and on the methodology of APATE [36]. The rest of this section summarizes these two works to make this paper self-contained.

APATE starts by building a real tripartite symmetric transaction/cardholder/merchant adjacency matrix $\mathbf{A}^{\text{tri}} = (a_{ij})$ based on a list of time stamped, labeled transactions where each cardholder and merchant is known,

$$\mathbf{A}^{\text{tri}} = \begin{bmatrix} \mathbf{0}_{t\times t} & \mathbf{A}_{t\times c} & \mathbf{A}_{t\times m} \\ \mathbf{A}_{c\times t} & \mathbf{0}_{c\times c} & \mathbf{0}_{c\times m} \\ \mathbf{A}_{m\times t} & \mathbf{0}_{m\times c} & \mathbf{0}_{m\times m} \end{bmatrix}$$

where $\mathbf{A}_{c\times t}$ is a biadjacency matrix where cardholders are linked with their corresponding transactions, $\mathbf{A}_{m\times t}$ is a biadjacency matrix where merchants are linked with their corresponding transactions and $\mathbf{0}_{\dots\times\dots}$ is a correctly sized matrix full of zeros. Moreover, a column vector $\mathbf{z}_0^{\text{tri}} = [\mathbf{z}_0^{\text{Trx}}; \mathbf{z}_0^{\text{CH}}; \mathbf{z}_0^{\text{Mer}}]$, containing the risk score of each transaction (Trx), cardholder (CH) and merchant (Mer) is created and initialized with zeroes, except for known fraudulent transactions (Trx) which are set to one.

APATE integrates also a time decay factor in order to address the dynamic behavior of fraud. Interested readers can consult the original APATE paper [36] for more information as it is not crucial to understand the framework in detail here. At the end, we obtain four pairs of $\mathbf{A}^{\text{tri}}$ and $\mathbf{z}_0^{\text{tri}}$ corresponding to four different time windows: no decay, day decay, or short term (ST), week decay, or medium term (MT) and monthly decay, or long term (LT).

Then for each of the four time window, in order to spread fraudulence influence through the tripartite graph, the vector $\mathbf{z}_0^{\text{tri}}$ is updated following an iterative procedure similar to the PageRank algorithm [27], namely the random walk with restart (RWWR) [35]:

$$\mathbf{z}_k^{\text{tri}} = \alpha\, \mathbf{P}^{\text{T}} \mathbf{z}_{k-1}^{\text{tri}} + (1-\alpha)\, \mathbf{z}_0^{\text{tri}} \tag{1}$$

where k is the iteration number, $\mathbf{P} = (p_{ij}) = (\frac{a_{ij}}{a_{i\bullet}})$ is the transition probability matrix [13] associated to $\mathbf{A}^{\text{tri}}$, α is the probability to continue the walk, and

symmetrically $(1 - \alpha)$ is the probability to restart the walk from a fraudulent transaction. Eq. 1 is iterated until convergence, to reach $\mathbf{z}_{k^*}^{\text{tri}}$ (where k^* stands for k at convergence) from which three new feature vectors can be extracted, $\mathbf{z}_{k^*}^{\text{Trx}}$, $\mathbf{z}_{k^*}^{\text{CH}}$ and $\mathbf{z}_{k^*}^{\text{Mer}}$. These three features correspond to a risk measure for each transaction, cardholder and merchant respectively. Therefore, for each transaction, there are 12 new graph based features created.

As this procedure cannot be computed in a few seconds, the scores for each transaction, cardholder and merchant are only re-estimated once a day or once per hour, in order to analyse transactions that will occur during the day. In cases where new merchants or cardholders appear, their scores are set to zero as nothing can be inferred from the past graph data. The risk score of a new transaction that did not yet occur in the past can be approximated using an update formula presented in [36].

Finally, APATE combines those 12 graph-based features with the transaction-related features initially present in the dataset (see [36] for details) , and use these as input of a random forest classifier.

While APATE is showing good performance, according to Lebichot et al. [21], it can be improved in three ways by: dealing with hubs, introducing a time gap and including investigators feedback.

The first way of improvement consists in dealing with hubs, which are nodes having a high degree, i.e. a large number of links with other nodes. Due to their connections to a lot of transactions, hubs tend to accumulate a high risk score. A simple solution to counterbalance this accumulation is to divide the risk score by the node degree after convergence of Eq. 1. Lebichot et al. [21] make the link between this solution and the Regularized Commute Time Kernel (RCTK) [23] in the sense that the elements of this kernel have the same interpretation as for the RWWR used in APATE. Therefore, they recommended the use of RCTK to deal with the problem of hubs.

Their second proposal is to introduce a time gap between the training set and the test set. Lebichot et al. [21] explain that, in most real FDS, the model cannot be based on the past few days, as is proposed in APATE, for two reasons. The first reason is that, in a real setting, the fraudulent transaction tags cannot be known without human investigator feedback. However, this feedback usually takes several days, mainly because it is often the cardholders that report undetected fraud. The second reason is that the strategy of the fraudsters changes over time and so it is less reliable to build the model on old data.

The third way of improvement consists in including feedback from the investigators on the predictions of the previous days. Even if it appears clear, in view of the second proposal, that it is impossible to know all fraud tags for the gap set, it is still conceivable that a fraction of previous alerts have been confirmed or overturned by human investigators (which is indeed the case in practice). We will refer later to this option as FB (for feedback) in Section 4.

3 The free energy distance

As discussed previously, the main contributions of this work are three-fold, (1) to propose a way of computing the free energy distance scaling on large, sparse,

graphs and (2) to incorporate the free energy framework into a FDS and (3) evaluate its performance on a real-world large-scale fraud detection problem. In this section, we start by providing a short account of the free energy distance and its properties, before discussing its implementation in the FDS. Note that this distance measure between nodes obtained very good results in a number of semi-supervised classification and clustering tasks [14, 16, 31, 32].

3.1 Background

The free energy distance [20], also know as the bag-of-paths potential distance [14], is a distance measure between nodes of a directed, strongly connected, graph based on the bag-of-paths framework. It is usually introduced by considering a statistical physics framework where it corresponds to the minimized free energy[1] of the bag-of-paths system connecting two nodes, but we will consider here a more intuitive explanation.

We already introduced the random walk on the graph whose transition probability matrix is $\mathbf{P}$, see Eq. 1. Recall that its elements are nonnegative and each of its rows sums to one. The free energy distance will be computed to some nodes of interest $\mathcal{A}$ (in our application, the fraudulent nodes), called *target nodes*. These nodes are made killing and absorbing by setting the corresponding rows in the transition probability matrix to zero. Note that if the original graph is not strongly connected, we used a common trick, namely to add a new absorbing, killing, (sink) node connected to the set of target nodes $\mathcal{A}$ with a directed link. In addition, we also assume that there is a nonnegative cost $c_{ij} \geq 0$ associated to each edge (i, j) of the graph with $\mathbf{C} = (c_{ij})$ being the cost matrix. The cost on an edge is assigned depending on the application and quantifies, in the model, the difficulty of following this edge in the random walk [13]. In our application, we fixed the cost to $c_{ij} = 1/a_{ij}$.

Then, a new matrix $\mathbf{W} = \mathbf{P} \circ \exp[-\theta \mathbf{C}]$ is introduced, where $\circ$ is the elementwise (Hadamard) product and θ is a positive parameter (the inverse temperature). This matrix is substochastic because each of its row sums is less or equal to 1 and at least one row sum is strictly less than 1 (for example the killing, absorbing, nodes whose row sum is equal to zero). In fact, this matrix defines a transition probability matrix of a *killed random walk* on the graph, because at each time step, when visiting a node i, the random walker has a probability $0 \leq (1 - \sum_{j \in \mathcal{S}ucc(i)} w_{ij}) \leq 1$, where $\mathcal{S}ucc(i)$ is the set of successor nodes of node i, of giving up the walk – we then say that the walker is killed. The larger the cost to successors, the larger the probability of being killed.

In this context, it can be shown that the *directed free energy dissimilarity* $\phi_{i\mathcal{A}}$ between any node $s = i$ (starting node) and the absorbing target nodes in $\mathcal{A}$ is simply $-\frac{1}{\theta} \log \mathrm{P}(\text{reaching}(\mathcal{A}) | s = i)$, that is, minus the logarithm of the probability of surviving during the killed random walk, i.e., of reaching an absorbing node without being killed during the walk [14]. Let us now explain how it can be computed.

[1] Expected total cost of the paths plus scaled relative entropy of the probability distribution of following these paths (see [20] for details).

The free energy distance between two nodes i and j is obtained by $\Delta_{ij} = \frac{\phi_{ij}+\phi_{ji}}{2}$. Besides being a distance measure, it has many interesting properties. One of those is the fact that it interpolates between two widely used distances, the shortest path distance and the commute cost distance (which is proportional to the effective resistance also called resistance distance [13]). Indeed, if the parameter θ approaches ∞, the free energy distance converges to the shortest path distance [13]. Conversely, if θ approaches 0^+, we recover the commute cost distance [13]. The details and proofs of these properties are available in [14].

The free energy distance between all pairs of nodes can be computed by performing a matrix inversion [20]. However, Françoisse et al. [14] showed that the directed free energy distance to a unique, fixed, target node t (the set of absorbing nodes reduces to node t) can also be computed thanks to an extension of the Bellman-Ford formula:

$$\phi_{it}(\tau+1) = \begin{cases} -\frac{1}{\theta}\,\log\Big[\sum\limits_{j\in\mathcal{S}ucc(i)} p_{ij}\,\exp\big[-\theta(c_{ij}+\phi_{jt}(\tau))\big]\Big] & \text{if } i \neq t \\ 0 & \text{if } i = t \end{cases} \tag{2}$$

where τ is the iteration number[2]. This expression uses the softmin, or log-sum-exp, operator [25], $\text{softmin}_{\theta,\mathbf{q}}(\mathbf{x}) = -\frac{1}{\theta}\log\sum_{j=1}^{n} q_j \exp[-\theta x_j]$ (with $q_j \geq 0$ and $\sum_{j=1}^{n} q_j = 1$) where, in the present context of Eq. 2, $q_j = p_{ij}$ and $x_j = (c_{ij}+\phi_{jt}(\tau))$. This operator interpolates between the minimum and the weighted average of the values x_j with weights q_j. In fact, Eq. 2 is nothing else than the Bellman-Ford formula (based on dynamic programming; see, e.g., [15]) where the minimum operator is replaced by the soft minimum operator. Eq. 2 can be iterated until convergence to the directed free energy distances. The main advantage of this formulation is that it can be applied on large, sparse, graphs thanks to some specific techniques which are explained below. After convergence, ϕ_{it} contains $-\frac{1}{\theta}\log$ of the probability of surviving during a killed random walk from i to t with transition matrix $\mathbf{W}$ [14].

3.2 Computing the directed free energy distance on large graphs

In order to scale the computation of the free energy, we will use Eq. 2 and rely on two different ideas: (i) the *log-sum-exp* trick and (ii) to *bound the length* of the set of paths on which the distance is computed. This second point brings another benefit: it allows to tune the length of the walks, which has been shown to improve the performance in some situations (see, e.g., [5, 23]).

The log-sum-exp trick [17, 19, 25] aims at pre-computing $x^* = \min_{j\in\{1\ldots n\}}(x_j)$, leading to the form $\text{softmin}_{\theta,\mathbf{q}}(\mathbf{x}) = x^* - \frac{1}{\theta}\log\sum_{j=1}^{n} q_j \exp[-\theta(x_j - x^*)]$ and then neglecting the terms in the summation for which $\theta(x_j - x^*)$ is too large (exceeds a certain threshold). This is a kind of pruning and has two benefits: it reduces significantly the number of terms to be computed and it avoids numerical underflow problems.

[2] Notice that the usual free energy distance (not directed) is defined by symmetrization of ϕ_{ij} (Eq. 2, so that the resulting distance is symmetric [14, 20]), but this quantity will not be used in this work.

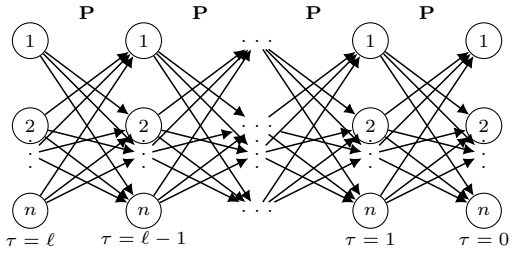

Fig. 1. The directed lattice derived from the original graph. It only considers walks of length up to ℓ.

Whereas the standard bag-of-paths framework is based on paths of unbounded length (from 0 to ∞), our second technique considers only walks bounded by a given length ℓ [5, 23]. This is done by defining a directed lattice L unfolding the original graph G in terms of increasing walk lengths. More precisely, this lattice is made of the graph nodes repeated at walk lengths $\tau = 0, 1, \ldots, \ell$, usually with $\ell \ll n$ [23] (see Fig. 1). Then, transitions are only allowed from nodes at walk length τ to successor nodes at length $\tau - 1$ by means of the transition matrix $\mathbf{P}$ of the killed random walk associated to the graph. This lattice represents a bounded random walk on the graph G where walks' lengths are in the interval $[0, \ell]$. The combination of these two tricks allows to scale on large, sparse, graphs – many edges are pruned and the computation occurs on a (usually small) grid.

In this context, on lattice L and considering now a set of target nodes $\mathcal{A}$, Eq. 2 becomes, for the initialization of the distances at $\tau = 0$, corresponding to zero-length walks,

$$\phi_{i\mathcal{A}}(0) = \begin{cases} \infty & \text{if } i \notin \mathcal{A} \\ 0 & \text{if } i \in \mathcal{A} \end{cases} \tag{3}$$

Moreover, $\phi_{i\mathcal{A}}(\tau)$ contains the directed free energy distance from node i to the absorbing nodes in $\mathcal{A}$ when considering walks up to length τ. A resulting distance of ∞ means that no walk of length up to τ exists between node i and a node in $\mathcal{A}$ – absorbing nodes cannot be reached in τ steps. Then, for walk length $\tau > 0$,

$$\phi_{i\mathcal{A}}(\tau + 1) = \begin{cases} -\frac{1}{\theta} \log \Big[\sum\limits_{j \in \mathcal{S}ucc(i)} p_{ij} \exp \big[-\theta(c_{ij} + \phi_{j\mathcal{A}}(\tau)) \big] \Big] & \text{if } i \notin \mathcal{A} \\ 0 & \text{if } i \in \mathcal{A} \end{cases} \tag{4}$$

This recurrence relation defines the directed free energy distance to target nodes $\mathcal{A}$ that will be used in our fraud detection application.

3.3 Application to the fraud detection problem

In order to incorporate the free energy (FE) framework into FDS APATE to create the FDS called FraudsFree (FF), we introduce some other modifications.

The first modification is related to the computation of the risk score vector $\mathbf{z}^{\text{tri}}_{k*} = \boldsymbol{\phi}^{\text{tri}}_{k*}$, presented in Eq. 1. For that, we use Eq. 4 by considering each

known fraud (Trx) as an absorbing node $a \in \mathcal{A}$. We iterate this equation until convergence to obtain the distance between all the nodes i and the set $\mathcal{A}$ which corresponds to the risk score of each transaction, cardholder and merchant. At this point, a score near 0 represents a fraud and a high value represents a genuine transaction. In order to keep the same interpretation of the risk score as in APATE[3], we apply the following transformation

$$\boldsymbol{\phi}^{\text{tri}}_{k*} = \max(\boldsymbol{\phi}^{\text{tri}}_{k*}) - \boldsymbol{\phi}^{\text{tri}}_{k*} \tag{5}$$

As for APATE [36] (see Section 2), we set the score of a new transaction j that did not yet occur in the past between a cardholder k and a merchant i via Eq. 2, which provides the following expression

$$\begin{aligned} \text{score}(Trx_j) = &- \tfrac{1}{\theta}\log\Big[p_{ji}\exp\big[-\theta(c_{ji} + \text{score}(Mer_i))\big]\Big] \\ &- \tfrac{1}{\theta}\log\Big[p_{jk}\exp\big[-\theta(c_{jk} + \text{score}(CH_k))\big]\Big] \end{aligned} \tag{6}$$

where $p_{ji} = p_{jk} = 0.5$ because a transaction is linked by construction with one merchant and one cardholder and we fixed $c_{ji} = c_{jk} = 1$ as the transaction appends now (no decay), but this is still a degree of freedom of our method that we left for further work.

4 Experimental comparisons and discussion

To evaluate our approach following the methodology of [21], we perform a comparison between the different versions of our FraudsFree (FF) model and the other variations of APATE (Random Walk With Restart (RWWR) and Regularized Commute Time Kernel (RCTK)), in supervised (SL) and semi-supervised learning (SSL) with feedback (FB), on the same real-life e-commerce credit card transaction dataset as [21]. The dataset contains 16 socio-demographic features on 25,445,744 e-commerce transactions gathered during 139 days. The data are highly imbalanced, with only 78,119 frauds among the transactions ($< 0.31\%$). The average size of $\mathbf{A}^{\text{tri}}$ is 3,910,783. This dataset does not focus on a certain type of card fraud but contains all reported fraudulent transactions in the investigated time period [21]. Besides the 16 original features, a set of 12 graph-based features per node, as described in Section 2 and 3, is created for each method. A small sample of this dataset is available on www.kaggle.com/mlg-ulb/creditcardfraud but the data are anonymised and the transactions are not presented day by day. Finally, all these features are fed into a class-rebalanced random forest with 400 trees. Each tree is built based on a random selection of 4 features of the original dataset and 4 graph-based features.

In order to assess the performance of each method, we select two measures. In accordance with field experts, we chose the Precision@100 in terms of card (Card Pr@100) [21, 34] (which is the most realistic setting). More precisely, we select the more fraudulent transactions according to the model until we screen

[3] So that a score near 0 represents a genuine transaction and a high value represents a fraud. The higher the score, the higher the risk.

Classifier	Hubs	Learning	Feedback	Card Pr@100	Time	Best parameter
RWWR SL = APATE	No	SL	No	18.19	155.40	0.1
RWWR SSL+FB	No	SSL	Yes	20.90	273.16	0.9
RCTK SL	Yes	SL	No	21.48	195.43	0.9
RCTK SSL+FB	Yes	SSL	Yes	24.03	294.07	0.7
RCTK SSL+FB 5 ITER	Yes	SSL	Yes	22.82	122.60	0.9
FF SL	No	SL	No	16.13	204.48	5
FF SSL+FB	No	SSL	Yes	24.20	489.43	0.5
FF SSL+FB 5 ITER	No	SSL	Yes	24.01	155.40	0.5

Table 1. Mean Card Pr@100 between Day 41 and Day 139 for each of the 8 methods (see Section 2,3 and 4 for acronyms). The average time in seconds required to create the tripartite graph and extract the 12 graph features between Day 41 and Day 139 is also reported for each of the 8 methods.

100 cards. As a second measure, we use the average time in seconds required to create the tripartite graph and extract the 12 graph features between Day 41 and Day 139.

Concerning the hyperparameters, the RWWR and RCTK methods consider tuning values of $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.85, 0.9\}$ and, for the FF methods $\theta = \{0.1, 0.5, 1, 5, 10\}$. For each method, we tuned its parameter based on the mean Card Pr@100 between Day 30 and Day 40. The results for the best parameter is presented in Table 1[4]. We obtained these results by applying a sliding window technique in accordance with expert knowledge [21]. We set 15 days in the training set, 7 days in the gapset (see Section 2 and [21]), 1 day in the test set and we shifted the sliding window day by day. Furthermore, in order to exploit all the properties of the bounded FE, we select the number of iterations (the walk length ℓ) for the best FF-based method based on a reasonable trade-off Card Pr@100/time. For the sake of comparisons, we also limited the number of RCTK iterations to the same number as in FF-based method.

To analyse the results with Card Pr@100 of Table 1, we use a nonparametric Friedman-Nemenyi statistical test and a Wilcoxon signed-ranks tests [11]. We perform all statistical tests at a level of confidence of 95%, which amounts to taking an α of 0.05. From the results of the Nemenyi test illustrated in Fig. 2, four methods perform equivalently, in terms of Card Pr@100, to the best one: the FE SSL+FB, the FF SSL+FB 5 ITER, the RCTK SSL+FB and the RCTK SSL+FB 5 ITER. However, the Wilcoxon tests report that RCTK SSL+FB 5 ITER is significantly inferior to the three other methods in one-to-one comparisons (with respective p-values of 0.0055, 0.0252 and 0.0030). Even if we cannot ensure a statistical difference between the top three, we still observe that there are some differences in terms of their computation times. The FF SSL+FB 5 ITER method is the fastest of the top three with a reduction of 47.16% in computation time compared to the best method proposed by Lebichot et al. [21], RCTK SSL+FB. All results were obtained with Matlab (version R2017a) running on an Intel Xeon with 2×8 3.6Ghz processors and 128 GB of RAM.

[4] Numerical values differ from [21] because the dataset was further curated : some obvious fraud cases were removed.

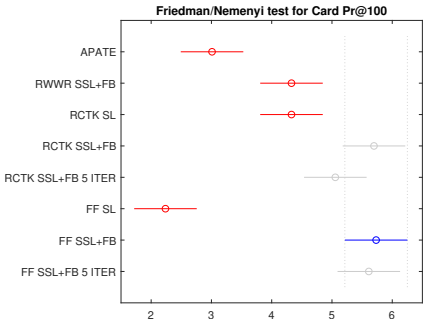

Fig. 2. Mean ranks and 95% Nemenyi confidence intervals for the 8 methods (see Table 1) based on the Card Pr@100. Two methods are considered as significantly different if their confidence intervals do not overlap.

5 Conclusion

In this paper, we investigate a version of the free energy distance that scales on large, sparse graphs. It is used in the existing Fraud Detection System APATE in order to extract features from the graph of transactions. Thanks to the properties of the free energy, we manage to reduce the computational time and improve the scalability of the Fraud Detection System. The Fraud Detection System based on the free energy distance, FraudsFree, is competitive as it obtains a Pr@100 score on fraudulent card prediction comparable to the previous work of Lebichot et al. [21] with a significant speed-up in computation. This shows that the free energy distance can be used on real-word applications involving large graphs. One considered further work is to deal with the hub nodes by modifying directly the cost matrix, as it has shown good results in other studies [21]. Another avenue that could be explored is to determine if our approach is complementary, or just redundant, to existing fraud defence lines of our industrial partner.

Acknowledgements. This work was partially supported by the Immediate funded by Wallon Region project and by the Defeatfrauds project funded by Innoviris. We thank these institutions for giving us the opportunity to conduct both fundamental and applied research. We also thank Worldline SA/NV, R&D, for providing us the data and expertise.

References

1. Abdallah, A., Maarof, M.A., Zainal, A.: Fraud detection system: A survey. Journal of Network and Computer Applications 68, 90–113 (2016)
2. Bahnsen, A.C., Stojanovic, A., Aouada, D., Ottersten, B.: Cost sensitive credit card fraud detection using bayes minimum risk. In: 2013 12th international conference on machine learning and applications. vol. 1, pp. 333–338. IEEE (2013)
3. Bhusari, V., Patil, S.: Study of hidden markov model in credit card fraudulent detection. International Journal of Computer Applications 20(5), 33–36 (2011)

4. Bolton, R.J., Hand, D.J.: Statistical fraud detection: A review. Statistical science pp. 235–249 (2002)
5. Callut, J., Francoisse, K., Saerens, M., Dupont, P.: Semi-supervised classification from discriminative random walks. In: Daelemans, W., Morik, K. (eds.) Proceedings of the 19th European Conference on Machine Learning (ECML '08). Lecture Notes in Artificial Intelligence, vol. 5211, pp. 162–177. Springer (2008)
6. Cao, B., Mao, M., Viidu, S., Yu, P.: Collective fraud detection capturing inter-transaction dependency. In: KDD 2017 Workshop on Anomaly Detection in Finance. pp. 66–75 (2018)
7. Chan, P.K., Fan, W., Prodromidis, A.L., Stolfo, S.J.: Distributed data mining in credit card fraud detection. IEEE Intelligent systems 14(6), 67–74 (1999)
8. Consultants, H.: The nilson report issue 1142 (2018), https://nilsonreport.com
9. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE transactions on neural networks and learning systems 29(8), 3784–3797 (2018)
10. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.A., Waterschoot, S., Bontempi, G.: Learned lessons in credit card fraud detection from a practitioner perspective. Expert systems with applications 41(10), 4915–4928 (2014)
11. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research 7, 1–30 (2006)
12. Duman, E., Elikucuk, I.: Solving credit card fraud detection problem by the new metaheuristics migrating birds optimization. In: International Work-Conference on Artificial Neural Networks. pp. 62–71. Springer (2013)
13. Fouss, F., Saerens, M., Shimbo, M.: Algorithms and models for network data and link analysis. Cambridge University Press (2016)
14. Françoisse, K., Kivimäki, I., Mantrach, A., Rossi, F., Saerens, M.: A bag-of-paths framework for network data analysis. Neural Networks 90, 90–111 (2017)
15. Gondran, M., Minoux, M.: Graphs and algorithms. Wiley (1984)
16. Guex, G., Courtain, S., Saerens, M.: Covariance and correlation kernels on a graph in the generalized bag-of-paths formalism. arXiv preprint arXiv:1902.03002 (2019)
17. Huang, X., Ariki, Y., Jack, M.: Hidden Markov models for speech recognition. Edinburgh University Press (1990)
18. Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.E., He-Guelton, L., Caelen, O.: Sequence classification for credit-card fraud detection. Expert Systems with Applications 100, 234–245 (2018)
19. Kivimäki, I.: Distances, centralities and model estimation methods based on randomized shortest paths for network data analysis. Ph.D. thesis, UCL-Université Catholique de Louvain (2018)
20. Kivimäki, I., Shimbo, M., Saerens, M.: Developments in the theory of randomized shortest paths with a comparison of graph node distances. Physica A: Statistical Mechanics and its Applications 393, 600–616 (2014)
21. Lebichot, B., Braun, F., Caelen, O., Saerens, M.: A graph-based, semi-supervised, credit card fraud detection system. In: International Workshop on Complex Networks and their Applications. pp. 721–733. Springer (2016)
22. Liu, Q., Wu, Y.: Supervised learning. Encyclopedia of the Sciences of Learning pp. 3243–3245 (2012)
23. Mantrach, A., Van Zeebroeck, N., Francq, P., Shimbo, M., Bersini, H., Saerens, M.: Semi-supervised classification and betweenness computation on large, sparse, directed graphs. Pattern recognition 44(6), 1212–1224 (2011)

24. Molloy, I., Chari, S., Finkler, U., Wiggerman, M., Jonker, C., Habeck, T., Park, Y., Jordens, F., van Schaik, R.: Graph analytics for real-time scoring of cross-channel transactional fraud. In: International Conference on Financial Cryptography and Data Security. pp. 22–40. Springer (2016)
25. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
26. Ngai, E.W., Hu, Y., Wong, Y.H., Chen, Y., Sun, X.: The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision support systems 50(3), 559–569 (2011)
27. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
28. Ramaki, A.A., Asgari, R., Atani, R.E.: Credit card fraud detection based on ontology graph. International Journal of Security, Privacy and Trust Management (IJSPTM) 1(5), 1–12 (2012)
29. Sánchez, D., Vila, M., Cerda, L., Serrano, J.M.: Association rules applied to credit card fraud detection. Expert systems with applications 36(2), 3630–3640 (2009)
30. Shen, A., Tong, R., Deng, Y.: Application of classification models on credit card fraud detection. In: 2007 International conference on service systems and service management. pp. 1–4. IEEE (2007)
31. Sommer, F., Fouss, F., Saerens, M.: Comparison of graph node distances on clustering tasks. Artificial Neural Networks and Machine Learning – Proceedings of ICANN 2016. Lecture Notes in Computer Science 9886, 192–201 (2016), springer
32. Sommer, F., Fouss, F., Saerens, M.: Modularity-driven kernel k-means for community detection. Artificial Neural Networks and Machine Learning (Proceedings of ICANN 2016. Lecture Notes in Computer Science 10614, 423–433 (2017), springer
33. Srivastava, A., Kundu, A., Sural, S., Majumdar, A.: Credit card fraud detection using hidden markov model. IEEE Transactions on dependable and secure computing 5(1), 37–48 (2008)
34. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Fourth Edition. Academic Press, Inc., 4th edn. (2008)
35. Tong, H., Faloutsos, C., Pan, J.Y.: Fast random walk with restart and its applications. In: Sixth International Conference on Data Mining (ICDM'06). pp. 613–622. IEEE (2006)
36. Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akogu, L., Snoeck, M., Baesens, B.: Apate : A novel approach for automated credit card transaction fraud detection using network-based extensions. Decision Support Systems 75, 38–48 (2015)
37. Weston, D.J., Hand, D.J., Adams, N.M., Whitrow, C., Juszczak, P.: Plastic card fraud detection using peer group analysis. Advances in Data Analysis and Classification 2(1), 45–62 (2008)
38. Wheeler, R., Aitken, S.: Multiple algorithms for fraud detection. In: Applications and Innovations in Intelligent Systems VII, pp. 219–231. Springer (2000)
39. Zaslavsky, V., Strizhak, A.: Credit card fraud detection using self-organizing maps. Information and Security 18, 48 (2006)
40. Zhang, Z., Zhou, X., Zhang, X., Wang, L., Wang, P.: A model based on convolutional neural network for online transaction fraud detection. Security and Communication Networks 2018 (2018)
41. Zhou, X., Cheng, S., Zhu, M., Guo, C., Zhou, S., Xu, P., Xue, Z., Zhang, W.: A state of the art survey of data mining-based fraud detection and credit scoring. In: MATEC Web of Conferences. vol. 189. EDP Sciences (2018)