2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Graph mining assisted semi-supervised learning for fraudulent cash-out detection

Yuan Li
Dept. of Statistics
Northwestern University
Evanston, USA
Email: yuanli2012@u.northwestern.edu

Yiheng Sun
JD.com
Beijing, China
Email: sunyiheng1@jd.com

Noshir Contractor
Dept. of Industrial Engineering
& Management Sciences
Northwestern University
Evanston, USA
Email: nosh@northwestern.edu

***Abstract*—Fraudulent cash-out is an increasingly serious problem in China, which costs financial facilities billions of dollars. Unlike most of the well-studied credit card fraud, where only one party illicitly seeks financial gain, fraudulent cash-out involves both parties of the transaction. When prior information, such as credit score and reputation score, about the majority of consumers and shops is available, the phenomenon can be readily analyzed by using the Markov random field models. In this paper, we investigate the detection of fraudulent cash-out under the circumstance where no prior information but only the labels of a small set of consumers and shops are available. The novelty of this work is building a semi-supervised learning algorithm that automatically tunes the prior and parameters in Markov random field while inferring labels for every node in the graph. We evaluate our algorithm with data from JD Finance.**

***Index Terms*—graph mining, Markov Random Field, semi-supervised learning, Bayesian optimization**

I. Introduction

Financial fraud has been increasing with the prevalence of modern technologies, resulting in hundreds of billions of dollars of loss each year [1], [2] There are many types of financial frauds and credit card fraud alone costs financial facilities billions of dollars of lost revenue annually [3].

Fraudulent cash-out is a new type of credit card fraud appearing in China, which involves the use of credit cards at point-of-sales (POS) machines and third-party online payment systems. Unlike most credit card fraud, in this case both the cardholder and merchant collude in fraudulent cash-outs. In a typical fraudulent transaction, a merchant with POS machines fabricates fictitious transactions for a cardholder, for example for the sale of goods. Rather than receiving goods in the transaction, the cardholder receives cash directly from the merchant instead. During the process, the merchant usually takes a small portion of the transaction price as commission fee, while the cardholder enjoys an interest free "loan" for a period of up to 56 days while avoiding to pay high interest payments on legal cash advances on their credit card. Moreover, by engaging in an fraudulent cash-out, a cardholder can obtain funds up to his/her credit limit, unlike cash advances, which typically have lower ceilings. A schematic diagram of fraudulent cash-out is shown in Fig.1. In our paper, we are especially interested in finding fraudulent merchants so that financial facilities can regulate these merchants directly.

Fraudulent cash out has wide reaching consequences in financial facilities and cardholders. It costs financial facilities billions of dollars annually and harms cardholders credit score. Traditional detection approaches rely on manual techniques which are inefficient and not scalable. Data mining based fraud detection algorithms, by recognizing patterns in transactions, have been proven to be useful [4] in many real-world cases. However, fraudulent activities also have been evolved to game the detection algorithms [5]. As such, the detection methods must improve accordingly.

There are many obstacles to these improvements and innovations to fraudulent detection algorithms. First, there is a dearth of scholarly publications on credit card fraud [6] due to the difficulties for academicians to obtain credit card transaction data. Without abundance of literatures, it makes exchanging ideas among academicians difficult and innovation slow. Second, the transaction data are complex by its nature. Even though fraud detection can be considered as a classification problem in machine learning, there is an imbalanced number of fraudulent and legitimate transactions, and different costs for misclassification. Another difficulty with analysis of the transaction dataset is that perpetrators, both the cardholder and merchant, usually carry more than one fraudulent cash-out fraud [7]. Instead of analyzing these frauds independently, a successful method should integrate the information.

Previous studies on data mining-based fraud detection can be categorized into four types [8]. Supervised learning methods, such as logistic regression, SVM, as well as neural networks [9], [10], [11] are applied on labeled data. Later more sophisticated approaches are developed by combining popular supervised learning methods in sequential fashion [12], [13]. When the available data are partially labeled, semi-supervised learning methods are popular[14], [15]. Graph mining and

ASONAM '17, July 31 - August 03, 2017, Sydney, Australia

ACM ISBN 978-1-4503-4993-2/17/07?/$15.00
http://dx.doi.org/10.1145/3110025.3110099

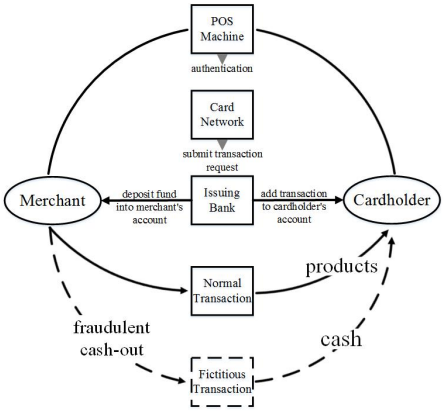

Fig. 1. The schematic diagram of fraudulent cash-out. Solid line represents the process of a normal transaction and dash line represents the process of fraudulent cash-out.

link analysis are popular unsupervised learning algorithms that have proven successful in detecting anomalies in unlabeled data [16]. However, these techniques are under-rated in fraud detection research [17].

In our paper, we modeled the detection problem within the Markov random field (MRF) framework and discuss the use of topology information on the bipartite transaction network as well as the use of transaction patterns to detect fraudulent merchants. With the presence of some labeled data, we hybridize Belief Propagation (BP), which works well in solving inference problem in unlabeled networks, and Bayesian optimization to develop a robust semi-supervised learning method. In previous studies, MRFs are applied to modify the reputation scores obtained either by some heuristic measures based on domain knowledges or by existing fraud detection algorithms. More specifically, the reputation score is first calculated independently of the assumption of the MRF model and then used to construct node potential in MRF. However reputation scores of consumers and shops are not always easily available in real-world problem. In previous study, parameters in the MRF are predetermined by the researchers' domain knowledges instead of being estimated by a data-driven method. To overcome these drawbacks, our algorithm tunes the prior for nodes and estimate parameters of edge potential in MRF by applying Bayesian optimization under the semi-supervised learning settings, therefore the algorithm infers the label of unknown nodes without requiring any prior knowledge or reputation scores of nodes. The algorithm complements existing fraud detection algorithms when any prior knowledge of node or reputation score is available. Our main contributions are as followings:

- Formulating the fraudulent cash-out detection problem as a graph mining and semi-supervised learning problem, where transaction information is embedded in edge potential.
- Using both labeled and unlabeled data to develop a robust algorithm. Bayesian optimization is applied in the algorithm to tune the parameters in Markov random field. Our method leverages the information of the network and the information of the labeled data therefore performs well.
- Evaluating our algorithm on JingDong (JD) Finance dataset. The performance shows that our algorithm is efficient, effective and scalable.

II. Data

The performance of the model is evaluated with real-world data from JD Finance. JD is one of the largest business to consumer (B2C) platforms in China with 1.6 billion transactions and 222.6 million active users in 2016. The data are stored and analyzed on JDs server. All sensitive fields in the data are encrypted and no personal identifiable information is accessible. A summary of the experiment data are shown in Table 1. The degree distribution of transactions for consumers and shops are shown in Fig. 1 a and Fig. 1 b correspondingly. The log-log plot suggests that the number of transactions has a heavy-tailed distribution. Like many other real-world networks, the degree distribution for shops exhibits power law property. A summary of descriptive statistics of the data is shown in Table 1. Some sensitive statistics are marked as NA.

A. Transaction

JD provides purchase-on-credit service for its consumers since Feb, 2014. The credit-card-like service enables consumers to purchase products on JD without instant payment and to repay the bill later. We use the terms card-holder and consumer interchangeably in different contexts. We obtained data on a sample of 2.91 million of offline purchase-on-credit transactions of JD users. Data on the transaction contains userID, merchantID, transaction amount, and trade status (succeeded or rejected). These transactions were made by $230,238$ users at $201,289$ shops. The data and results we present in this paper is only from a small proportion of all JDs transactions. In the practical application of the model, JD Finance will use its complete dataset.

B. labeled data

Users in the dataset are marked as fraudulent or unknown, while merchants are labeled as good, fraudulent or unknown. Both the fraudulent consumers and the fraudulent merchants were confirmed and marked by JD's agents manually. The agents are trained professionally to identify suspicious transactions and make phone calls to check. The marked fraudulent users usually have suspicious online behavior. Notably, no users are marked as good users. This is because in China,

TABLE I
Descriptive Statistics of the experiment data

	labeled	**Unknown**	**Sum**
user	NA	NA	230238
Mercahnt	7582	193707	201289
Transaction	0	2913471	2913471

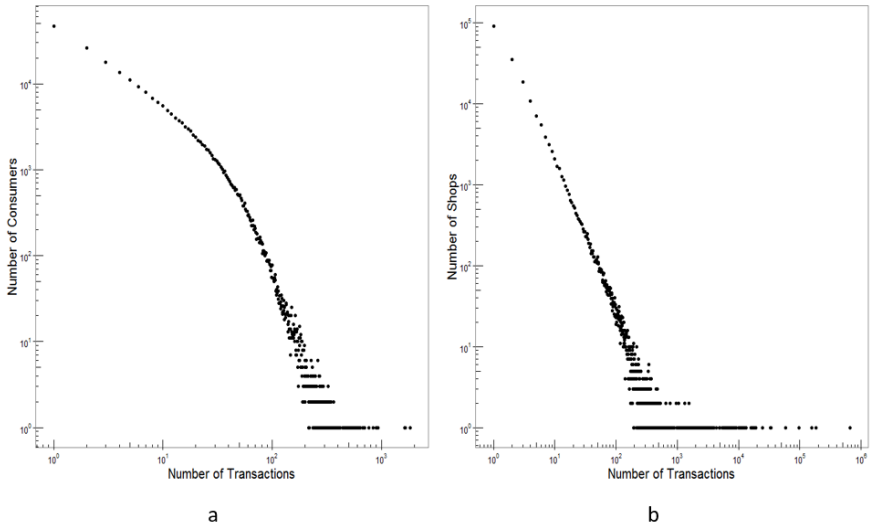

Fig. 2. (a) Distribution of number of transactions for consumer (log-log). (b) Distribution of number of transactions for shop

the credit score system has not been well developed yet and the cost of delinquency for card-holder is relatively low. As a consequence, it is hard to tell whether a good consumer will turn into a fraudulent one in the near future. However on the other hand, the cost of making fraudulent transactions for a shop with a good reputation is much higher. Therefore we are confident in labeling shops with good reputation as good in the sampled data.

III. Method

In this section, we formalize the fraudulent cash-out detection problem into a semi-supervised network learning problem and discuss the methodology.

A. Problem statement

After introducing our goal and dataset, we formally define our problem as follows. Given:

- An undirected bipartite Graph $G = (V_c, V_s, E)$ where vertices i_c in the set V_c, represents consumers and vertices j_s in set V_s represents shops, and E correspond to the transactions among V_c and V_s.
- The binary label $X \in \{-1, 1\}$ observed over a subset V_s^l of V_s and label $X = 1$ over a subset V_c^l of V_c, where $X = 1$ corresponds to fraudulent status
- The frequency of transactions between vertices i_c and vertices j_s and the amount associated with the transactions.

Output: marginal probability $P(X_{j_s} = 1)$ for vertices j_s in V_s, or the probability of a shop involved in fraudulent cash-out transaction.

In general, the task of labeling vertices in a graph is NP-hard. Markov Random Fields provide an attractive theoretical model for this problem. In details, we can model the joint probability of vertices as

$$P\{X\} = \frac{1}{Z} \prod_{j_s \in V_s} \phi(X_{j_s}) \prod_{i_c \in V_c} \phi(X_{i_c}) \prod_{i,j \in E} \psi_{i_c j_s}(X_{i_c}, X_{j_s}) \tag{1}$$

where the compatibility function $\psi_{i_c j_s}$ is the edge potential, function ϕ is the node potential and Z is a normalization constant. More specifically, node potential $\phi(X_{i_c})$ and $\phi(X_{j_s})$ reflect our prior knowledge about consumer i_c and shop j_s.

The inference problem is still NP-hard [16] even under the assumption of the MRF model. However recent developments of Belief Propagation algorithm can be used to solve inference problem on graph in several different domains [17][18][19], including our context, where we are able to label vertices by passing messengers along the edges. Mathematically, the messenger is updated by the following rules. The belief passed from a consumer to a shop takes the following form:

$$m_{i_c j_s}(X_{j_s}) = \sum_{X_{i_c}} \phi(X_{i_c}) \psi_{i_c j_s}(X_{i_c}, X_{j_s}) \prod_{k_s \in \partial i_c \backslash j_s} m_{k_s i_c}(X_{i_c}) \tag{2}$$

where $\partial i_c \backslash j_s$ represents the neighbors of consumer i_c except shop j_s and the messenger can be understood as consumer i_c's belief of what state shop j_s should be. Similarly, the belief passed from a shop to a consumer takes the form of:

$$m_{j_s i_c}(X_{i_c}) = \sum_{X_{j_s}} \phi(X_{j_s}) \psi_{j_s i_c}(X_{j_s}, X_{i_c}) \prod_{k_c \in \partial j_s \backslash i_c} m_{k_c j_s}(X_{j_s}) \tag{3}$$

Then our belief of consumer i_c is updated as:

$$b_{i_c}(X_{i_c}) = K_{i_c} \phi(X_{i_c}) \prod_{j_s \in \partial i_c} m_{j_s i_c}(X_{i_c}) \tag{4}$$

and our belief of shop j_s is updated as:

$$b_{j_s}(X_{j_s}) = K_{j_s}\phi(X_{j_s}) \prod_{i_c \in \partial j_s} m_{i_c j_s}(X_{j_s}) \tag{5}$$

where K_{i_c}, K_{j_s} are normalizing constants.

In our problem, several modifications on Belief Propagation algorithm are needed to incorporate extra information carried by transaction and our knowledge of observed labels. The adaption is a non-trivial process since BP is originally designed for unsupervised learning, while our problem is semi-supervised.

Several studies [22], [23] tried to address the semi-supervised learning problem in generative approach. One commonly adopted method is to redefine the overall log likelihood function where labeled data and unlabeled data have different weights in the redefined function, thus the algorithm that maximizes this redefined function is robust against incorrect model assumptions. However, in Markov random fields, due to the compatibility function $\phi_{i_c j_s}$, it seems unclear about how to assign different weights to labeled and unlabeled data in the log likelihood function. Some researches [24], [25] proposed methods that directly absorb the information of labeled nodes into the Markov random field model , but their approach suffers when the generative model is not accurate.

In our paper, we use the labeled data in a different way. To the best of our knowledge, under the Markov random model assumption, all the algorithms for fraud detection or anomaly detection choose the parameters in potential functions arbitrarily or by some domain knowledge. However with the presence of partially labeled data, we develop a method that achieves better performance by estimating parameters in potential functions from the labeled data. In the next part, we discuss how to relax our model assumption to make our method more robust and how to apply Bayesian optimization to tune the parameters in node potentials and edge potentials.

B. The adaption of belief propagation algorithm

This section details how to incorporate transaction information and observed labels to achieve our objective of detecting fraudulent cash-out.

1) Transaction information: Transaction between consumers and shops are categorized into different types based on their amount. Then we model the edge potential between i_c and j_s in the Markov Random Field as following:

$$\psi_{i_c j_s}(X_{i_c}, X_{j_s}) = \frac{1}{1 + e^{\sum_1^p \alpha_{k X_{i_c} X_{j_s}} m_{k X_{i_c} X_{j_s}}}} \tag{6}$$

where p is the number of all possible types of transactions and $m_{k X_{i_c} X_{j_s}}$ is the number of k^{th} type transactions between vertices i_c and j_s and $\alpha_{k X_{i_c} X_{j_s}}$ is the parameter that indicates hemophilic relation among shops and consumers for the k^{th} type of transaction.

2) Consumers label: In the Markov Random Field model, the known label can be directly formulated into the generative model. More specifically, for known-fraudulent consumers, we freeze their node potential $\phi(X_{i_c} = 1)$ to be 1, so the message passed from a known-fraudulent consumer i_c to a shop j_s takes the following form:

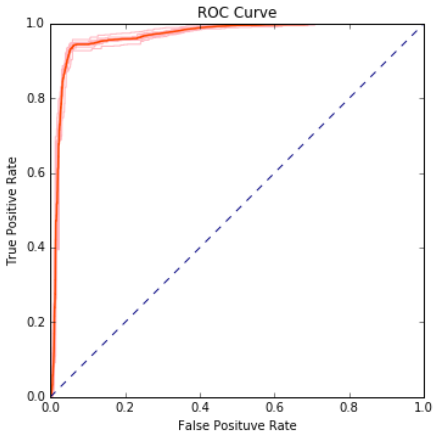

Fig. 3. ROC curve for shops. Dark red line is the average ROC curve over 10 experiments and light red lines are ROC curves for each experiment.

$$m_{i_c j_s}(X_{j_s}) = \psi_{i_c j_s}(X_{i_c} = 1, X_{j_s}) \prod_{k_s \in \partial i_c \backslash j_s} m_{k_s i_c}(X_{i_c} = 1) \tag{7}$$

where ∂i_c represents the neighbors of consumer i_c. Then the marginal probability for a known fraudulent consumer i_c, by applying BP, is 1. In practice, it is more desirable to relax the model and set node potential for labeled consumer as:

$$\phi(i_c \in V_c^l) = \begin{cases} \beta_c^l, & \text{for } X_{i_c} = 1 \quad (8a) \\ 1 - \beta_c^l, & \text{for } X_{i_c} = -1 \quad (8b) \end{cases}$$

and node potential for unlabeled consumer as:

$$\phi(i_c \in V_c \backslash V_c^l) = \begin{cases} \beta_c^u, & \text{for } X_{i_c} = 1 \quad (9a) \\ 1 - \beta_c^u, & \text{for } X_{i_c} = -1 \quad (9b) \end{cases}$$

where $\beta_c^u < \beta_c^l < 1$. These parameters are estimated by applying Bayesian Optimization.

3) Shops label: Labeled shops are used to estimate parameters $\alpha_{k X_{i_c} X_{j_s}}$ for edge potentials and parameters (β_c^u, β_c^l) for consumer node potentials. Both the potentials of unlabeled shops and labeled shops are set to be 0.5. Note here we do not estimate parameters for shop node potentials, the reason for which is discussed in section 4. Next we minimize the loss function over labeled shops by tuning edge potentials and consumer potentials with Bayesian optimization [26]. More details are discussed in the next section. This offers two advantages: first, by tuning parameter in the Markov Random Field, our algorithm efficiently uses the information carried by different types of transactions thus achieves good performance; second, instead of putting extreme value 0 or 1 for labeled shops and consumers, the shop potentials and consumer potentials are trained relatively neutral to avoid the undesirable chain reaction that changes beliefs dramatically.

C. estimation of parameters

In our algorithm, parameters in edge potential and node potential are estimated by the following procedures.

- First, given a set of parameters $(\alpha_{kX_{i_c}X_{j_s}}, \beta_c^u, \beta_c^l)$, by applying BP, the marginal probability of a shop j_s being fraudulent is obtained.
- Then we calculate the value of a loss function L over all labeled shops based on the obtained marginal probability. The choice of the loss function L is discussed in details later.
- Last, Bayesian optimization is used to find the optimal solution to the following optimization problem:

$$(\alpha_{kX_{i_c}X_{j_s}}, \beta_c^u, \beta_c^l) = \underset{\alpha_{kX_{i_c}X_{j_s}}, \beta_c^u, \beta_c^l}{\operatorname{argmin}} L(j_s | j_s \in V_s^l) \quad (10)$$

Note that after applying BP, no explicit expression of the loss function L can be obtained in terms of $(\alpha_{kX_{i_c}X_{j_s}}, \beta_c^u, \beta_c^l)$, therefore Bayesian optimization [23] is used to find the optimal solution. It seems plausible to estimate parameters for shop node potentials as well however this approach results in an unstable algorithm. More details are discussed in section 4.

An alternative approach is expectation-maximization (EM) algorithm [27]. More specifically, we try to maximize the marginal likelihood of observed labels:

$$\begin{aligned} &P\{X_{i_c}, X_{j_s} | i_c \in V_c^l, j_s \in V_s^l\} \\ =&\sum_{\{S\}} \frac{1}{Z} \prod_{j_s \in V_s} \phi(X_{j_s}) \prod_{i_c \in V_c} \phi(X_{i_c}) \prod_{i,j \in E} \psi_{i_c j_s}(X_{i_c}, X_{j_s}) \end{aligned} \quad (11)$$

with respect to $\{S\}, \alpha, \beta$, where $\{S\}$ is the set of all possible joint states of unlabeled consumer $i_c \in V_c \backslash V_c^l$ and unlabeled shop $j_s \in V_s \backslash V_s^l$, α is the set of parameters in edge potential and β is the set of parameters in node potential. In E step, by applying BP, we maximize $P\{X_{i_c}, X_{j_s} | i_c \in V_c^l, j_s \in V_s^l\}$ with respect to s and calculate the optimal $q\{S\}$,where $q\{S\}$ is the distribution for $\{S\}$, and in M step, holding $q\{S\}$ constant, we maximize $P\{X_{i_c}, X_{j_s} | i_c \in V_c^l, j_s \in V_s^l\}$ with respect to α, β. We iterate these two steps until the parameters converge. EM algorithm is designed to find parameters corresponding to a local maximal of likelihood function, however when the likelihood function is not correctly specified, the good performance of EM algorithm is not guaranteed. In our paper, we prefer the more robust algorithm that maximizes a goal oriented loss function L by applying Bayesian optimization to the EM algorithm.

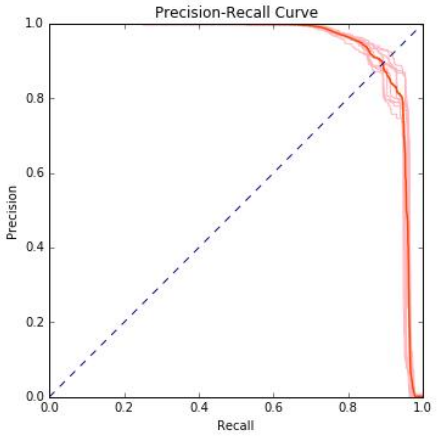

Fig. 4. Precision-Recall curve for shops. Dark red line is the average Precision-Recall curve over 10 experiments and light red lines are Precision-Recall curves for each experiment.

IV. Experiments and results

In this section, we evaluate our algorithm with bipartite consumer-shop network. The raw data are collected from JD Finance. The network consists of all $230,238$ consumers and $201,289$ shops, and 2.91 million transactions among them. We show that our algorithm effectively detects fraudulent shops by passing beliefs along the bipartite network and estimating edge potentials iteratively. We also evaluate the influence of multiple factors, including the parameter settings for edge potentials and node potentials, number of sampled nodes and choice of loss functions. One-fourth of the labeled vertices were used for testing, and the rest were used as training data. Without specification, all true positive rates (TPR) provided in this paper were measured at 5% false positive rate (FPR).

A. Experiment setup

In the basic experimental setup, multiple initial guesses for the parameters are generated to prevent local optimal solutions. Bayesian optimization is conducted for each initial guess and returns a respective set of estimated parameters. The set of parameters leading to the smallest loss function is chosen as the optimal solution. The algorithm attains an average TPR of 92.47% over 10 random 4-fold cross validations as shown in Fig.3. To create smooth ROC curve, $10,000$ of threshold values are generated such that vertices with higher posterior probability than the thresholds are classified as fraudulent shops. The ROC curve, AUC of ROC, precision-recall curve and TPR are used to measure the performance of the algorithm. Fig.4 shows the precision-recall curve for shops achieved by our algorithm. F1-score can be calculated corresponding to the different choice of threshold value. The highest F1-score achieved in Fig.4 is 0.8955, where the corresponding precision is 0.8962 and the corresponding recall is 0.8947.

B. Comparison between different loss function

In this section, we discuss the choice of loss function in eq(10). In the context of fraud detection, goal driven approaches are sometimes desirable, therefore we tune the parameters in the MRF by either maximizing TPR or AUC of ROC or minimizing deviance.

Fig.5 shows the performance of our algorithm with different choices of loss function. Interestingly, the algorithm converges to the same set of parameters for all three loss functions in all 10 experiments when allowing Bayesian optimization to run sufficient number of iterations. In each experiment, we randomly choose three-fourth of nodes as training data and remaining one-fourth as testing data. The performance

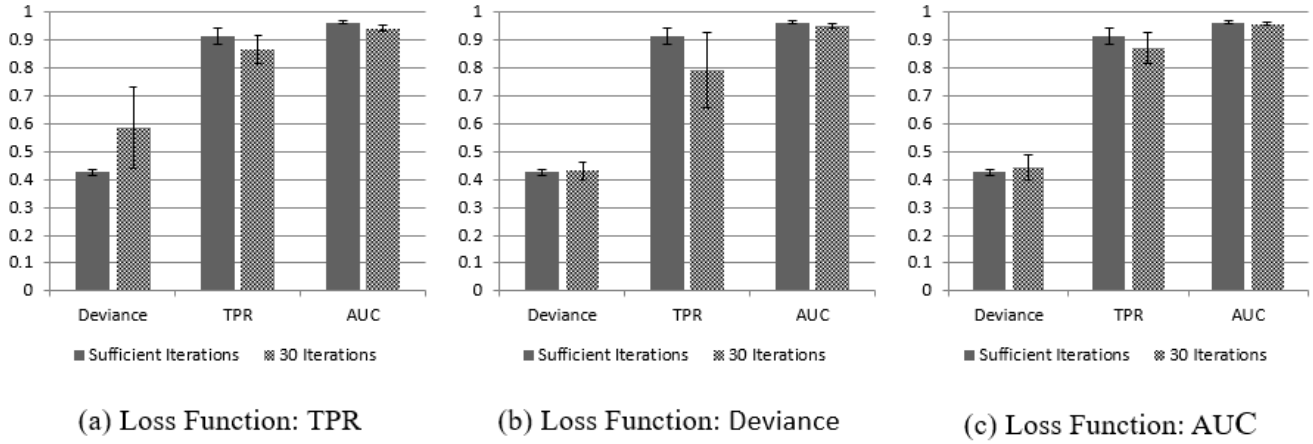

Fig. 5. A comparison of different loss function. Dark bars represent the performances of the algorithms after running sufficient number of iterations of Bayesian optimization, and light bars represent the performances of the algorithms after running 30 iterations of Bayesian optimization. The performances are measured in Deviance, TPR and AUC. (a) The performance of algorithm that maximizes TPR; (b) The performance of algorithm that minimizes TPR; (c) The performance of algorithm that maximizes AUC

of the algorithm with these optimal parameters is represented by dark bars in Fig.5. One possible reason is that when conducting Bayesian optimization, parameters are restricted and the optimal solution obtained by Bayesian optimization is on the boundary. Relaxation of some restrictions could lead to different optimal parameters for different loss function; but since the algorithm has already achieved a good accuracy, it is not our primary interests to run Bayesian optimization over a larger parameter space. On the other hand, although the algorithm converges to the same set of parameters, it converges to the set of optimal parameters with different rates under different loss functions. In all of the 10 experiments, the algorithm that maximizes AUC is always the fastest to converge to the optimal parameters. In real world application, we would prefer to limit the maximal number of iterations in the Bayesian optimization. If we limit our optimization to 30 iteration, using AUC as the loss function (shown in Fig. 5c) achieves good average performance among all three different measures and the variances are small. When TPR (shown in Fig. 5a) is chosen as the loss function, the performance of the algorithm is relatively unstable with respect to the deviance of the results. When deviance has been minimized, the algorithm has poor performance with respect to TPR. Hence in our algorithm, we maximize AUC over the labeled shops to tune the parameters in Markov Random Fields

C. impact of edge potentials

Most prior research on fraud detection, malware detection and Sybil detection model edge potential as a function of node labels. This parsimonious way of modeling ignores information carried by different types of edges and is therefore limited when prior information of node potentials is not available. In our algorithm, edge potentials are modeled in more sophisticated way as shown in eq(4). Fig.6 shows the performance of the algorithm under different edge potential models. The more sophisticated model outperforms the parsimonious one in all three measurements. The results indicate that frequency of transaction and type of transaction carry extra information that should be included into the edge potentials. Another advantage of our model is that by modeling different types of transactions, we can understand which type of transaction is more likely to be fraudulent, and financial facilities would use this information to regulate fraudulent merchants.

D. impact of the node potentials

As discussed in the previous section, to make our algorithm more robust, the priors for labeled fraudulent consumers and unlabeled consumers are set to be β_c^u, β_c^l and determined by Bayesian optimization. When sampling different numbers of labeled nodes to train the model, the optimal β_c^l falls in the range $[0.6, 0.7]$, which is in contrast with the choice of assigning known-fraudulent node a prior equal to 1 in Markov random fields model. We hypothesized that this is because that a small portion of the labeled nodes can't be modeled accurately by Markov random field. When assigning these nodes priors based on Markov random fields model, the neighbors of which are influenced by the strong priors and therefore wrongly labeled. Therefore, by tuning node potentials, our algorithm avoids this undesirable chain reaction and achieves better TPR.

E. impact of the number of labeled nodes

We run a series of experiments to test the impact of the number of labeled nodes. In each experiment, we randomly select $0\%, 10\%, 25\%, 50\%$ or 100% labeled consumers and $10\%, 25\%, 50\%$ or 100% labeled shops as labeled nodes and

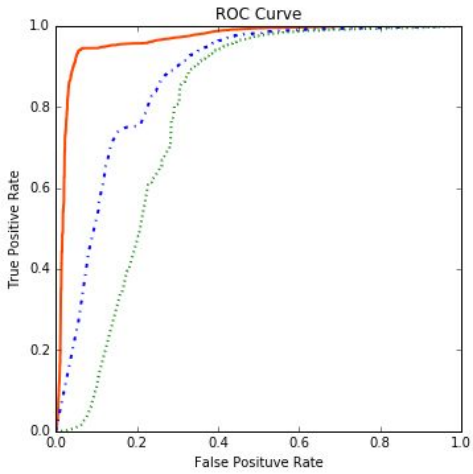

Fig. 6. ROC curves of the algorithms under different edge potential models. Red line corresponds to our model. Dark blue and light blue lines correspond to two parsimonious models used in previous studies [25], [28].

treat the rest nodes as unknowns. Table 2 shows that our algorithm is robust to the number of labeled nodes.

When only a small fraction of the labeled data are sampled and used as input, our algorithm still performs decently. For ground truth nodes, our algorithm recovers the labels of fraudulent shops with 91.14% accuracy when controlling TPR at 5%, given only 10% labeled shops as input. We hypothesize that even though only a small fraction of nodes is labeled, the average geodesic distance between a node and its nearest labeled node is small. For example, in a random graph whose average degree is 10, when given 1% labeled data, the average geodesic distance between a node and its nearest labeled node is around 2 by some simple calculation. This fact suggests that a small fraction of labeled nodes would provide more information than we thought. Belief propagation will efficiently use the information of network topologies therefore results in an effective algorithm. However as [22] pointed out, there should be at least one labeled node in each local community; otherwise belief propagation is unable to infer the nodes label.

TABLE II
IMPACT OF THE NUMBER OF LABELED NODES WHEN SHOP POTENTIALS ARE SET TO BE 0.5

	$P_m = 10\%$	$P_m = 25\%$	$P_m = 50\%$	$P_m = 100\%$
$P_c = 0\%$	0.9114	0.9033	0.8967	0.9127
$P_c = 10\%$	0.9156	0.9036	0.8965	0.9099
$P_c = 25\%$	0.9237	0.9116	0.9086	0.9288
$P_c = 50\%$	0.9250	0.9123	0.9196	0.9148
$P_c = 100\%$	0.9012	0.9008	0.9071	0.9248

TABLE III
IMPACT OF THE NUMBER OF LABELED NODES WHEN SHOP POTENTIALS ARE ESTIMATED

	$P_m = 10\%$	$P_m = 25\%$	$P_m = 50\%$	$P_m = 100\%$
$P_c = 0\%$	0.7960	0.8815	0.9196	0.9306
$P_c = 10\%$	0.8055	0.9195	0.9108	0.9206
$P_c = 25\%$	0.9163	0.9227	0.9226	0.9271
$P_c = 50\%$	0.8362	0.9047	0.9225	0.9305
$P_c = 100\%$	0.8570	0.9092	0.9313	0.9348

F. impact of the estimation of parameters for shop node potentials

Our algorithm does not estimate parameters for shop node potentials, instead, shop node potentials are set to be 0.5. It might sound plausible to estimate the parameters for shop node potentials, however a direct application of Bayesian optimization always yields trivial degenerate solutions where prior for fraudulent shop is 1 and prior for good shop is 0. This is because the loss function is defined over the labeled shops. To overcome this problem, we need an extra procedure to estimate those parameters. More specifically, instead of conducting 4-fold cross validation, we have to spare another one-fourth of the data to determine the parameters. We divide the data into four quarters; the first one-half of the data are used as training data and posterior distributions for the rest of the nodes are calculated, another one-fourth of the data are used for calculating the loss function and the last one-fourth of the data are used for cross-validation. The parameters that minimizes the loss function are estimated by Bayesian optimization. When building a less biased model, less data are available to estimate the parameters. The selection of algorithm reflects the trade-off between bias and variance. As shown in Table 3, when using all the labeled data as input, the algorithm that estimates extra parameters for shop node potentials outperforms the original algorithm, however its performance deteriorates sharply as the number of labeled nodes decreases. To obtain a more robust algorithm, we choose not to estimate the parameters for shop node potentials.

V. CONCLUSION AND FUTURE WORK

In this study, we proposed an algorithm that infers the network by graph mining and semi-supervised learning. We carefully use the nodes label in the bipartite network and combine the information of transaction details into our model. We evaluate the efficiency of our algorithm with JD data set.

A. Conclusion

We have the following observations:

- Our algorithm is efficient. We achieve 92% TPR while controlling FPR at 5% level in JD dataset. The algorithm is scalable. In a sparse network, the total complexity of Belief propagation is $O(n)$ and since our parameter space is relatively small, the total complexity after applying Bayesian optimization is still $O(n)$.
- Our algorithm sheds light on regulation for the fraudulent merchants. It is often the case that the fraudulent merchants conduct fraudulent transactions as well as legal

business. By eliminating high risk transactions and keeping safe transactions, financial facilities can maximize their capital gains.

- Our algorithm is robust even if only a small number of nodes are labeled. In real world, ground truth is hard to obtained. Our algorithm provides an attractive way to use the limited observed labels.

B. Future work

- In the current model, edge potential isn't a function of node degree. When the degree distribution follows power law, which is often the case in real world network, it might be more desirable to correct the edge potential with node degree.
- When parameter space grows, tuning parameters for edge potential and node potential could be computationally expensive by simply applying Bayesian optimization. A fast optimization algorithm would be needed.
- In practice, how to allocate the budget of labeling nodes in a network is an important question. The nave strategy to randomly samples nodes from the network and labels them is inefficient. Better strategy should take advantage of the network structure.
- This algorithm can be developed into an ensemble approach. In our framework, it is possible to incorporate the information collected by other existing fraud detection algorithms into the node potentials. However, the optimal way to incorporate this information

ACKNOWLEDGMENT

The authors would like to thank Jianbo Chen, Xiao Pan, and Jiangxiao Jiang for useful discussions regrading to JD dataset. The authors would also like to thank professor Wenxin Jiang and Tian Lu for useful discussions regrading to modeling and algorithm.

REFERENCES

[1] J. West, and M. Bhattacharya, "Intelligent financial fraud detection: a comprehensive review," Computers & Security, vol. 57, pp.47-66, 2016.

[2] E. Kirkos, C. Spathis, and Y. Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements," Expert Systems with Applications, vol. 32(4), pp.995-1003, 2007.

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel,and J.C. Westland, "Data mining for credit card fraud: a comparative study," Decision Support Systems, vol. 50(3), pp.602-613, 2011.

[4] E.W.T. Ngai, Y. Hu, Y.H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature," Decision Support Systems, vol. 50(3), pp.559-569, 2011.

[5] R.J. Bolton,and D.J. Hand, "Unsupervised profiling methods for fraud detection," Credit Scoring and Credit Control VII, pp.235-255, 2001

[6] R.J. Bolton, and D.J. Hand, "Statistical fraud detection: a review," Statistical Science, pp.235-249, 2002.

[7] S. Jha, M. Guillen, and J.C. Westland, "Employing transaction aggregation strategy to detect credit card fraud," Expert Systems with Applications, vol. 39(16), pp.12650-12657, 2012.

[8] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," arXiv preprint arXiv:1009.6119, 2010.

[9] T.S. Lim, W.Y. Loh, and Y.S. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms" Machine Learning, vol. 40(3), pp.203-228, 2000.

[10] M.J. Kim, and T.S. Kim, "A neural classifier with fraud density map for effective credit card fraud detection," in International Conference on Intelligent Data Engineering and Automated Learning, pp.378-383, Springer Berlin Heidelberg, 2002, August.

[11] M. Syeda, Y.Q. Zhang, and Y. Pan, "Parallel granular neural networks for fast credit card fraud detection," in Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference, vol. 1, pp.572-577, 2002.

[12] C. Phua, D. Alahakoon, and V. Lee, "Minority report in fraud detection: classification of skewed data," ACM SIGKDD Explorations Newsletter, vol. 6(1), pp.50-59, 2004.

[13] T. Ormerod, N. Morley, L. Ball, C. Langley, and C. Spenser, "Using ethnography to design a Mass Detection Tool (MDT) for the early discovery of insurance fraud," in CHI'03 Extended Abstracts on Human Factors in Computing Systems, pp. 650-651, 2003, April.

[14] H.C. Kim, S. Pang, H.M. Je, D. Kim, and S.Y. Bang, "Constructing support vector machine ensemble," Pattern Recognition, vol. 36(12), pp.2757-2767, 2003.

[15] S. Rosset, U. Murad, E. Neumann, Y. Idan, and G. Pinkas, "Discovery of fraud rules for telecommunicationschallenges and solutions," in Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 409-413, 1999, August.

[16] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," Data Mining and Knowledge Discovery, vol. 29(3), pp.626-688, 2015.

[17] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," arXiv preprint arXiv:1009.6119, 2010.

[18] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Exploring Artificial Intelligence in the New Millennium, vol. 8, pp.236-239, 2003.

[19] K.P. Murphy, Y. Weiss, and M.I. Jordan, "Loopy belief propagation for approximate inference: an empirical study," in Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence , pp.467-475, 1999, July.

[20] P.F. Felzenszwalb, and D.P. Huttenlocher, "Efficient belief propagation for early vision," International Journal of Computer Vision, vol. 70(1), pp.41-54, 2006.

[21] J. Sun, N.N. Zheng, and H.Y. Shum, "Stereo matching using belief propagation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25(7), pp.787-800, 2003.

[22] E. Eaton, and R. Mansbach, "A spin-glass model for semi-supervised community detection," in AAAI, pp. 900-906, 2012, July.

[23] X. Zhu, "Semi-supervised learning tutorial," in International Conference on Machine Learning (ICML) ,pp. 1-135, 2007, June.

[24] N.Z. Gong, M. Frank, and P. Mittal, "Sybilbelief: a semi-supervised learning approach for structure-based sybil detection," IEEE Transactions on Information Forensics and Security, vol. 9(6), pp.976-987, 2014.

[25] D.H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos, "Polonium: Tera-scale graph mining for malware detection," in ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2010, July.

[26] J. Snoek, H. Larochelle, and R.P. Adams, "Practical Bayesian optimization of machine learning algorithms," In Advances in Neural Information Processing Systems, pp.2951-2959, 2012.

[27] S.S. Saquib, C.A. Bouman, and K. Sauer, "ML parameter estimation for Markov random fields with applications to Bayesian tomography," IEEE Transactions on Image Processing, vol. 7(7), pp.1029-1044, 1988.

[28] S. Pandit, D.H. Chau, S. Wang, and C. Faloutsos, "Netprobe: a fast and scalable system for fraud detection in online auction networks' in Proceedings of the 16th International Conference on World Wide Web, pp.201-210, 2007, May.