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

Yuan Li Yiheng Sun Noshir Contractor

Aug 2, 2017

NORTHWESTERN UNIVERSITY

Outline

- Introduction
- Method
- Experiments and Results
- Conculsion and Future work

What is fraudulent cash-out?

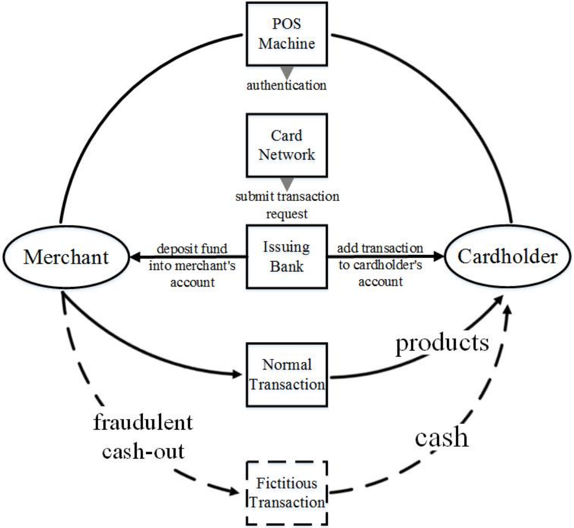


Figure: The schematic diagram of fraudulent cash-out.

Problem Setting

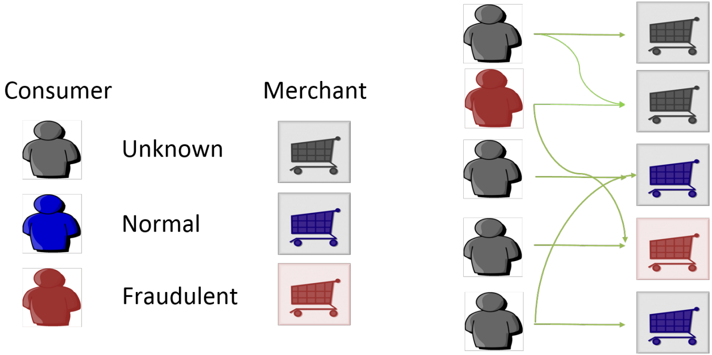


Figure: The schematic diagram of the problem setting.

Approaches for fraud detection

- Supervised learning methods, such as logistic regression, SVM, as well as neural networks

C.Phua, V.Lee, K.Smith, and R.Gayler. arXiv preprint arXiv:1009.6119, 2010.

Approaches for fraud detection

- Supervised learning methods, such as logistic regression, SVM, as well as neural networks
- Supervised learning hybrid

C.Phua, V.Lee, K.Smith, and R.Gayler. arXiv preprint arXiv:1009.6119, 2010.

Approaches for fraud detection

- Supervised learning methods, such as logistic regression, SVM, as well as neural networks
- Supervised learning hybrid
- Semi-supervised learning approach with clustering algorithm

C.Phua, V.Lee, K.Smith, and R.Gayler. arXiv preprint arXiv:1009.6119, 2010.

Approaches for fraud detection

- Supervised learning methods, such as logistic regression, SVM, as well as neural networks
- Supervised learning hybrid
- Semi-supervised learning approach with clustering algorithm
- Graph mining

C.Phua, V.Lee, K.Smith, and R.Gayler. arXiv preprint arXiv:1009.6119, 2010.

If we could estimate user reputation, then

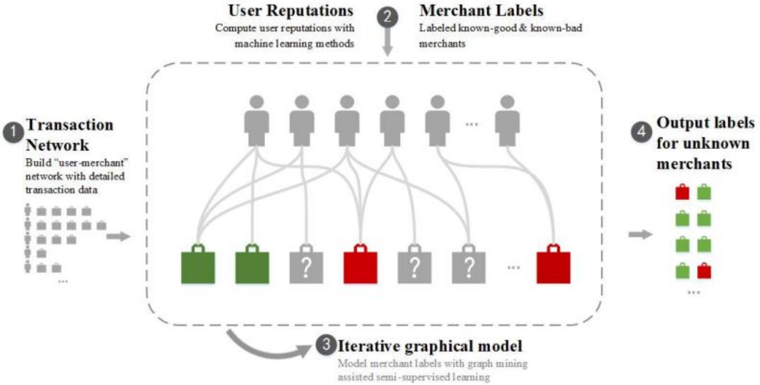


Figure: The schematic diagram of modeling fraudulent cash-out detection problem in supervised learning and graph mining hybrid approach.

D.H.Chau etc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2010

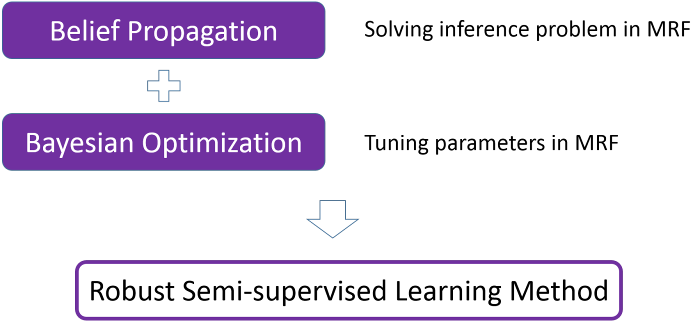
However when reputation score is not available, we need to

- Model edge potential more carefully
- Tune the parameters in Markov random field

However when reputation score is not available, we need to

- Model edge potential more carefully
- Tune the parameters in Markov random field

Our approach:



Problem statement

Given:

- An undirected bipartite graph $G = (V_c, V_s, E)$
  V_c: the set of consumer nodes
  V_s: the set of merchant nodes
  E: the edge set corresponding to the transactions among V_c and V_s.

Problem statement

Given:

- An undirected bipartite graph $G = (V_c, V_s, E)$
  V_c: the set of consumer nodes
  V_s: the set of merchant nodes
  E: the edge set corresponding to the transactions among V_c and V_s.
- The binary variable $X \in \{-1, 1\}$ observed over a subset V_s^l of V_s and $X = 1$ over a subset V_c^l of V_c, where $X = 1$ corresponds to fraudulent status.

Problem statement

Given:

- An undirected bipartite graph $G = (V_c, V_s, E)$
  V_c: the set of consumer nodes
  V_s: the set of merchant nodes
  E: the edge set corresponding to the transactions among V_c and V_s.
- The binary variable $X \in \{-1, 1\}$ observed over a subset V_s^l of V_s and $X = 1$ over a subset V_c^l of V_c, where $X = 1$ corresponds to fraudulent status.
- The frequency of transactions between $i_c \in V_c$ and $j_s \in V_s$ and the amount associated with the transactions.

Output:

- $P(X_{j_s} = 1)$ for $j_s \in V_s$: probability of a shop involved in fraudulent cash-out transaction.

Modeling

Markov random field:

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

Given node potential $\phi(X_{j_s}), \phi(X_{i_c})$ and edge potential $\psi_{i_c j_s}(X_{i_c}, X_{j_s})$, the marginal probability $P(X_{j_s} = 1)$ for vertices j_s can be calculated with Belief Propagation algorithm.

Edge Potential

- Transaction between consumers and shops are categorized into different types based on their amount.

- Edge potential is modeled as:

$$\psi_{i_c j_s}(X_{i_c}, X_{j_s}) = \frac{1}{1 + e^{\sum_1^p \alpha_{k x_{i_c} x_{j_s}} m_{k x_{i_c} x_{j_s}}}} \quad (2)$$

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

Node Potential

- Consumer node potential:

$$\phi(i_c \in V_c^l) = \begin{cases} \beta_c^l, & \text{for } X_{i_c} = 1 \quad (3a) \\ 1 - \beta_c^l, & \text{for } X_{i_c} = -1 \quad (3b) \end{cases}$$

$$\phi(i_c \in V_c \backslash V_c^l) = \begin{cases} \beta_c^u, & \text{for } X_{i_c} = 1 \quad (4a) \\ 1 - \beta_c^u, & \text{for } X_{i_c} = -1 \quad (4b) \end{cases}$$

Node Potential

- Consumer node potential:

$$\phi(i_c \in V_c^l) = \begin{cases} \beta_c^l, & \text{for } X_{i_c} = 1 \quad (3a) \\ 1 - \beta_c^l, & \text{for } X_{i_c} = -1 \quad (3b) \end{cases}$$

$$\phi(i_c \in V_c \backslash V_c^l) = \begin{cases} \beta_c^u, & \text{for } X_{i_c} = 1 \quad (4a) \\ 1 - \beta_c^u, & \text{for } X_{i_c} = -1 \quad (4b) \end{cases}$$

- Shop node potential:
  - Labeled shops are used to estimate parameters
  - Both the potentials of unlabeled shops and labeled shops are set to be 0.5.

Parameter Estimation

- (1) Given a set of parameters $(\alpha_{k x_{i_c} x_{j_s}}, \beta_c^u, \beta_c^l)$, by applying BP, the marginal probability of a shop j_s being fraudulent is calculated.

Parameter Estimation

- (1) Given a set of parameters $(\alpha_{k x_{i_c} x_{j_s}}, \beta_c^u, \beta_c^l)$, by applying BP, the marginal probability of a shop j_s being fraudulent is calculated.
- (2) The value of a loss function L defined over all labeled shops are calculated.

Parameter Estimation

- (1) Given a set of parameters $(\alpha_k x_{i_c} x_{j_s}, \beta_c^u, \beta_c^l)$, by applying BP, the marginal probability of a shop j_s being fraudulent is calculated.
- (2) The value of a loss function L defined over all labeled shops are calculated.
- (3) Bayesian optimization is used to find the optimal solution to the following optimization problem:

$$(\alpha_k x_{i_c} x_{j_s}, \beta_c^u, \beta_c^l) = \underset{\alpha_k x_{i_c} x_{j_s}, \beta_c^u, \beta_c^l}{\operatorname{argmin}} L(j_s | j_s \in V_s^l) \tag{5}$$

Data

The performance of the model is evaluated with real-world data from JD Finance.

Table: Descriptive Statistics of the experiment data

	Labeled	Unknown	Sum
Consumer	NA	NA	230238
Merchant	7582	193707	201289
Transaction	0	2913471	2913471

Number of nodes vs Number of transactions

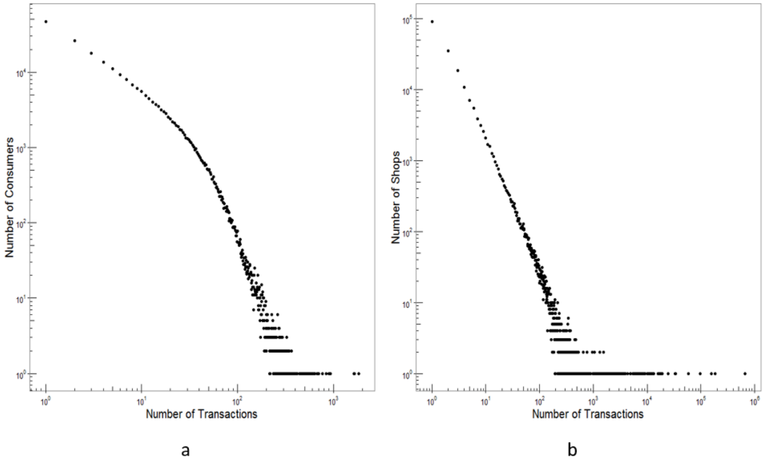


Figure: node degree distribution (log-log)

Experiment Setup

- 10 random 4-fold cross validations

Experiment Setup

- 10 random 4-fold cross validations
- Multiple initial guesses for the parameters are generated to prevent local optimal solutions.

Performance of our algorithm

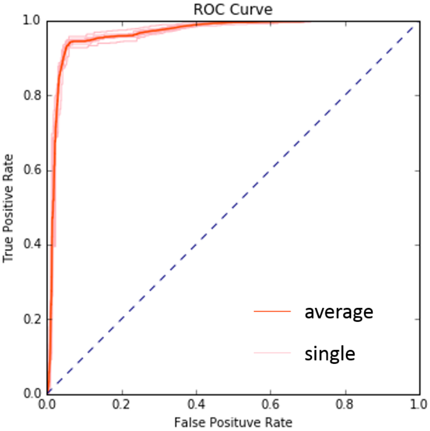
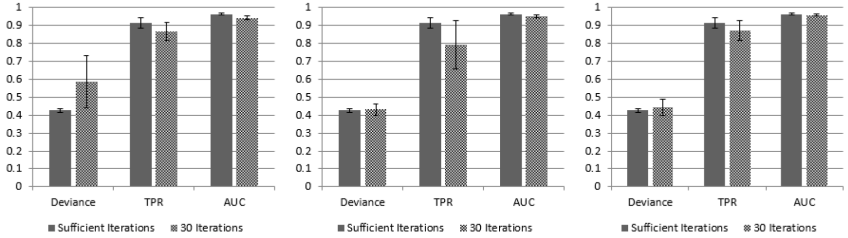


Figure: ROC curve for shops. Dark red line is the average ROC curve over 10 experiments and light red lines are ROC curves for each experiment.

Choice of Loss function



(a) Loss Function: TPR (b) Loss Function: Deviance (c) Loss Function: AUC

Figure: 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.

Edge Potential

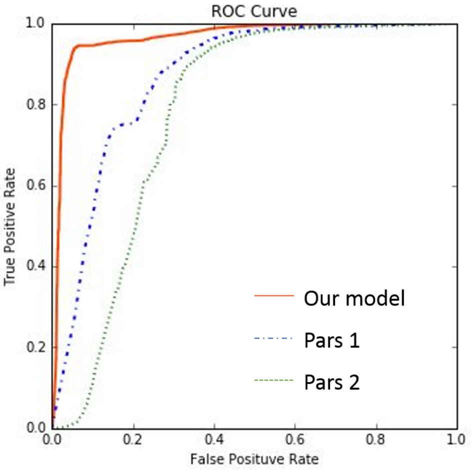


Figure: ROC curves of the algorithms under different edge potential models. Red line corresponds to our model. Dark blue and light blue lines correspond to two parsimonious models used in previous studies.

Node Potential

Table: 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

Node Potential

Table: 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

Conclusion

- Our algorithm is efficient and scalable. We achieve 92% TPR while controlling FPR at 5% level in JD dataset. The algorithm is scalable.
- Our algorithm sheds light on regulation for the fraudulent merchants.
- 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.

Future work

- Including node degree into the model
- Allocating the budget of labeling nodes in a network
- Developing an ensemble approach.

Thank you for your attention!