Spectrum-based Deep Neural Networks for Fraud Detection

Shuhan Yuan
Tongji University
4e66@tongji.edu.cn

Xintao Wu
University of Arkansas
xintaowu@uark.edu

Jun Li
University of Oregon
lijun@cs.uoregon.edu

Aidong Lu
University of North Carolina at Charlotte
aidong.lu@uncc.edu

ABSTRACT

In this paper, we focus on fraud detection on a signed graph with only a small set of labeled training data. We propose a novel framework that combines deep neural networks and spectral graph analysis. In particular, we use the node projection (called as spectral coordinate) in the low dimensional spectral space of the graph's adjacency matrix as the input of deep neural networks. Spectral coordinates in the spectral space capture the most useful topology information of the network. Due to the small dimension of spectral coordinates (compared with the dimension of the adjacency matrix derived from a graph), training deep neural networks becomes feasible. We develop and evaluate two neural networks, deep autoencoder and convolutional neural network, in our fraud detection framework. Experimental results on a real signed graph show that our spectrum based deep neural networks are effective in fraud detection.

KEYWORDS

fraud detection; spectrum; deep neural networks

1 INTRODUCTION

Online social networks (OSNs) have become popular social services for linking people together. Unfortunately, due to the openness of OSNs, fraudsters can also easily register themselves, inject fake contents, or take fraudulent activities, imposing severe security threads to OSNs and their legitimate participants. Many fraud detection techniques have been developed in recent years [1, 3, 6, 9], including content-based approaches and graph-based approaches. Different from content-based approaches that extract content features, (i.e., text), from user activities on social networks [3], graph-based approaches identify frauds based on network topologies. Often based on unsupervised learning, the graph-based approaches consider fraud as anomalies and extract various graph features associated with nodes, edges, ego-net, or communities from the graph [1].

In practice, a small set of labeled users are often available and hence supervised learning based detection approaches could be developed. In this paper, we introduce deep neural network models for detecting frauds in signed graphs. Deep neural networks have achieved remarkable results in computer vision, natural language processing, and speech recognition areas [8]. A deep neural network can learn different levels of representations on different layers of neural network [4]. However, one challenge of applying deep neural networks for fraud detection is lack of sufficient labeled data. When deep neural networks with a high dimensional input have a large number of parameters, the deep neural networks need to be trained with a large training dataset [8]. Hence it is often infeasible to use the adjacency matrix of the underlying graph as inputs of deep neural network models because of the high dimension of the adjacency matrix and the small number of labeled users.

We propose a novel framework that combines spectral graph analysis with the deep neural networks. In particular, we first project a graph to its spectral space formed by the principal eigenvectors of its adjacency matrix. The spectral space captures the main topological information of the graph. Each node is then mapped to a low dimensional point (called spectral coordinate) in the spectral space. We then use each node's spectral coordinate together with the aggregate information of its neighbor nodes' spectral coordinates as the input of two deep neural network models, deep autoencoder and convolutional neural network.

The advantages of our framework over past efforts are as follows. First, using both spectral graph analysis and deep neural networks, we can avoid defining graph metrics (features) to identify the difference between fraudsters and regular users. Second, the low-dimensional spectral space contains the most useful topology information of a graph. Comparing with the adjacency matrix, the dimension of spectral coordinates of nodes is much lower. Thus, using the node spectral coordinates as inputs to deep neural networks is suitable for real cases where the labeled users are limited. Moreover, most of the existing works for fraud detection focus on unsigned graphs in which there are only one type of links, while our framework covers signed networks. In order to capture both positive and negative edge information of a node in the signed graph, inputs of the two deep neural networks are composed by combining spectral coordinates of the node and its positive/negative-connected neighbors.

2 MODELS

2.1 Framework

Given a signed undirected graph G, each node in G indicates either a regular user or fraudster. The signed graph G can be represented as a symmetric adjacency matrix $\mathbf{A}_{n*n}$, where n is the number of nodes. In $\mathbf{A}_{n*n}$, $a_{ij} = 1$ ($a_{ij} = -1$) indicates there is a positive (negative)

CIKM'17, November 6–10, 2017, Singapore.
 ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: https://doi.org/10.1145/3132847.3133139

edge between nodes i and j and $a_{ij} = 0$ indicates no edge. $\mathbf{A}$ has n real eigenvalues. Let λ_i be the i-th largest eigenvalues of $\mathbf{A}$ with eigenvector $\mathbf{v}_i$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. The spectral decomposition of $\mathbf{A}$ is $\mathbf{A} = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^T$ (shown in Figure 1). There usually exist k leading eigenvalues that are significantly greater than the rest ones for networks. The row vector $\boldsymbol{\alpha}_u = (v_{1u}, v_{2u}, \ldots, v_{ku})$ is the spectral coordinate of node u in the k-dimensional subspace spanned by $(\mathbf{v}_1, \ldots, \mathbf{v}_k)$.

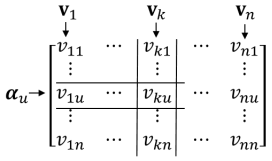

Figure 1: Spectral decomposition of the adjacency matrix A

In this work, we adopt deep autoencoder and convolutional neural network to identify fraudsters. Instead of using adjacency matrix, we adopt spectral coordinates to represent nodes as inputs to the two deep neural networks since spectral coordinates of nodes preserve the most useful structure information about nodes. Meanwhile, given a node u in a graph G, it has neighbors in s-steps. For example, when $s = 1$, the 1-step neighbors indicate the neighbors are one step away from the node u. Spectral coordinate of u's neighbors in s-steps represent broader topological information about the node u. For example, spectral coordinates of 1-step neighbors have been successfully used to detect random link attacks from unsigned graphs [9]. Thus, we further adopt the spectral coordinates of node neighbors in s-steps. In a signed graph, neighbors of node u can be divided into 2 categories based on their edge types, i.e., neighbors connected by positive edges and neighbors connected by negative edges. We compute the mean vector of s-step neighbors' spectral coordinates for each category, denoted as $\boldsymbol{\beta}_u^{s^+}$ and $\boldsymbol{\beta}_u^{s^-}$, where s indicates the s-step neighbors. Then, given a node u, to capture a broad structure information of u, the final inputs of two deep neural networks combine the spectral coordinates of node u and its two categories of neighbors in s-steps. We use a small part of labeled nodes to train the deep autoencoder and convolutional neural network. After training, the deep neural networks are able to identify fraudsters of the rest nodes in the signed graph.

2.2 Using deep autoencoder (DAE) for fraud detection

DAE stacks multiple basic autoencoder blocks hierarchically, which can capture multiple levels of representations of the input data [2]. We adopt spectral coordinates of nodes in a signed graph as inputs to DAE. DAE can preserve the hidden knowledge about the node from its spectral coordinate. Training DAE for fraud detection contains two phases: the pre-training phase and training phase. In the pre-training phase, DAE trains the model in an unsupervised manner. In the training phase, DAE trains the classifier and fine-tunes the whole model to predict the labels of nodes. Given the spectral coordinates of node u and its two categories of neighbors in s-steps, the input $\mathbf{x}_u \in \mathbb{R}^{(2s+1)k}$ of DAE is defined as:

$$\mathbf{x}_u = [\boldsymbol{\alpha}_u \ \boldsymbol{\beta}_u^{1^+} \ \boldsymbol{\beta}_u^{1^-} \ \ldots \ \boldsymbol{\beta}_u^{s^+} \ \boldsymbol{\beta}_u^{s^-}]. \tag{1}$$

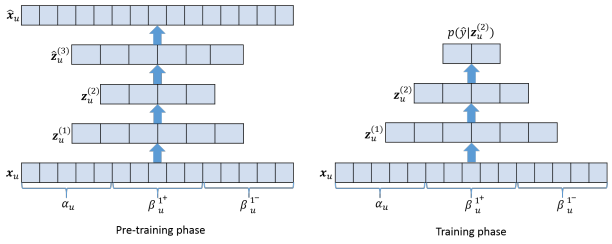

Figure 2: Architecture of DAE with spectral coordinates of node u and its 1-step neighbors for fraud detection

Given a DAE, there are L encoders to compute the hidden representations $\mathbf{z}_u^{(1)}, \mathbf{z}_u^{(2)}, \ldots, \mathbf{z}_u^{(L)}$ of $\mathbf{x}_u$ layer by layer. The input of the $(l+1)$-th encoder is the output of the l-th encoder. Specially, the input of the first encoder is $\mathbf{x}_u$. Then, there are another L decoders to compute the reconstructed input $\hat{\mathbf{x}}$. The equations of encoder and decoder are shown in Equation 2 and 3, respectively.

$$\mathbf{z}_u^{(l)} = \sigma(\mathbf{W}^{(l)} \mathbf{z}_u^{(l-1)} + \mathbf{b}^{(l)}), \tag{2}$$

$$\hat{\mathbf{z}}_u^{(L+l)} = \sigma(\mathbf{W}^{(L+l)} \hat{\mathbf{z}}_u^{(L+l-1)} + \mathbf{b}^{(L+l)}), \tag{3}$$

where $\mathbf{W}$, $\mathbf{b}$ are the parameters of the encoder; σ is a nonlinear activation function.

The objective function of pre-training is to make the reconstructed input $\hat{\mathbf{x}}$ to be close to the original input $\mathbf{x}$ by minimizing the reconstruction squared error,

$$\boldsymbol{L}(\mathbf{x}, \hat{\mathbf{x}}) = |\hat{\mathbf{x}} - \mathbf{x}|^2. \tag{4}$$

After pretraining DAE, we stack L encoders layer by layer to generate the hidden representation $\mathbf{z}_u^{(L)}$. $\mathbf{z}_u^{(L)}$ captures the hidden information of $\mathbf{x}_u$ since it can be used to reconstruct the input. Then, a softmax classifier is applied on top of the $\mathbf{z}_u^{(L)}$ to predict the label of node u.

$$P(\hat{y} = c | \mathbf{z}_u^{(L)}) = \frac{\exp(\mathbf{u}_c^T \mathbf{z}_u^{(L)} + b_c)}{\sum_{c'=1}^{C} \exp(\mathbf{u}_{c'}^T \mathbf{z}_u^{(L)} + b_{c'})}, \tag{5}$$

where C is number of classes, $\mathbf{u}_c$ and b_c are the parameters of softmax function for the c-th class. The parameters of softmax and deep autoencoder are trained and fine-tuned by minimizing the cross entropy loss function,

$$\boldsymbol{L} = -\frac{1}{N} \sum_{i=1}^{N} y_i * \log(P(\hat{y}_i)), \tag{6}$$

where y_i is the true class of the i-th input, and N is the number of training data. Figure 2 shows the architecture of DAE with spectral coordinate of node u and its 1-step neighbors.

2.3 Using convolutional neural network (CNN) for fraud detection

A basic convolutional neural network is composed by a convolution operation and a pooling operation. Given a node u, the input of CNN $\mathbf{X}_u \in \mathbb{R}^{(2s+1)*k}$ is represented as

$$\mathbf{X}_u = [\boldsymbol{\alpha}_u; \boldsymbol{\beta}_u^{1^+}; \boldsymbol{\beta}_u^{1^-}; \ldots; \boldsymbol{\beta}_u^{s^+}; \boldsymbol{\beta}_u^{s^-}], \tag{7}$$

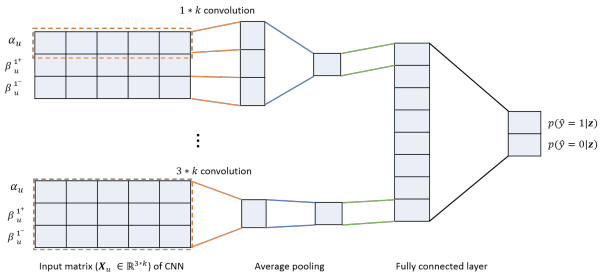

Figure 3: Architecture of CNN with spectral coordinates of node u and its 1-step neighbors for fraud detection

where ; indicates the vertical concatenation of two vectors. A convolution operation which involves a filter $\mathbf{W} \in \mathbb{R}^{m*k}$ is applied on a sub-matrix of $\mathbf{X}_u$ with m continuous rows to generate a hidden feature:

$$h_j = \sigma((\mathbf{W} * \mathbf{X}_u)_{j:j+m-1} + b), \tag{8}$$

where σ indicates a nonlinear function; $*$ is a convolution operation; b is a bias parameter; and $m \leq 2s+1$. Initially, the filter is partially connected to the input matrix $\mathbf{X}_u$. Then, the filter slides through the whole input matrix $\mathbf{X}_u$ and generates a feature vector $\mathbf{h} = [h_1, \cdots, h_{2s-m+2}]$. After that, an average pooling operation, which calculates the average value of the feature vector $\mathbf{h}$, is used to capture all discriminative features of $\mathbf{X}_u$. The average pooling operation is defined as $z = mean(\mathbf{h})$.

For example, when the filter size m is 1, the convolution operation generates a feature vector $\mathbf{h} \in \mathbb{R}^{2s+1}$ which contains the hidden information of node u and its all positive/negative-connected neighbors in s-steps. Then, an average pooling operation is applied on $\mathbf{h}$ to get the hidden feature z by calculating the average value of $\mathbf{h}$ as the output of current filter. When the filter size $m > 1$, the model follows the similar procedure to generate hidden features from $\mathbf{X}_u$.

Meanwhile, because one hidden feature vector $\mathbf{h}$ is generated by one filter, the feature vector detects the same feature from different locations of $\mathbf{X}_u$. To identify different aspects of features from $\mathbf{X}_u$, the model applies multiple filters with different sizes of m to generate different feature vectors. After applying the pooling operation on each feature vector, the model encodes the input $\mathbf{X}_u$ to a representation vector $\mathbf{z} = [z_1, z_2, \cdots, z_q]$, where q is the number of feature vectors generated by q different filters. For each filter size m, CNN generates the same number of feature vectors. We then apply a softmax classifier on $\mathbf{z}$ to predict the node label $P(\hat{y} = k|\mathbf{z})$ by Equation 5. The whole model is trained by minimizing the loss function shown in Equation 6. The architecture of CNN with spectral coordinates of node u and its 1-step neighbors is shown in Figure 3.

3 EXPERIMENTS

To evaluate the effectiveness and efficiency of our approach, we conduct experiments on a signed graph for fraud detection.

3.1 Experimental Setup

Datasets. We conduct our evaluation on a signed network, *WikiEditor*, which is extracted from the UMD Wikipedia dataset[7]. The dataset is composed by 17015 vandals and 17015 benign users who edited the Wikipedia pages from Jan 2013 to July 2014. Different from benign users, vandals edit articles in a deliberate attempt to damage Wikipedia. One edit may be reverted by bots or editors. Hence, each edit can belong to either revert or no-revert category. WikiEditor is built based on the co-edit relations. In particular, a positive (negative) edge between users i and j is added if the majority of their co-edits are from the same category (different categories). We remove those edits on meta pages (i.e., with titles containing "User:", "Talk:", "User talk:", "Wikipedia:") because the editings on those pages are not reverted by bot or administrators. We further remove from our signed network those users who do not have any co-edit relations with others. Finally, there are 18992 users in WikiEditor, including 6086 benign users and 12906 vandals. Meanwhile, there are 81316 edges, including 52139 positive edges and 29177 negative edges.

Experimental settings. In our experiments, after projecting the signed graph to the spectral space, we first normalize the spectral coordinates of nodes. We then combine nodes and their 1-step neighbors' spectral coordinates as inputs to deep neural networks. DAE contains two encoders. The dimensions of two encoders are 128 and 64. In CNN, the filter size $m \in [1, 2, 3]$ and the number of filters q is 300. The training epochs of DAE and CNN models are 30 with early stopping. We randomly sample different percentages of nodes for training and use the rest of the nodes for testing. Since the labeled fraudsters in real cases are usually small, we only adopt a small percentage of nodes as training data. We use the accuracy to evaluate the performance of different approaches for vandal detection. We report the mean values of 10 different runs by sampling different training data.

Baselines. We compare deep neural networks with two classical classifiers, k-NN and SVM. In k-NN, we set $k = 3$. In SVM, we adopt the RBF kernel and set the regularization parameter $c = 1$. The inputs of k-NN and SVM are the same as DAE.

Table 1: Accuracy of vandal detection with various sizes of training dataset

Input	Algorithm	Ratio of the training dataset			
		5%	10%	15%	20%
A	k-NN	66.16%	68.82%	69.66%	74.00%
	SVM	67.81%	67.82%	67.88%	67.92%
	DAE	76.31%	78.55%	79.56%	80.59%
	CNN	**76.70%**	**78.95%**	**80.09%**	**81.33%**
$\mathbf{x}_u$	k-NN	76.60%	77.38%	77.83%	78.19%
	SVM	71.60%	80.40%	80.82%	81.15%
	DAE	**80.89%**	81.13%	81.45%	81.92%
	CNN	80.57%	**81.40%**	**82.02%**	**82.61%**

3.2 Results

We first compare each model with different sizes of training datasets using spectral coordinates and the adjacency matrix. The default dimension of spectral coordinate k is 30. When using the adjacency matrix $\mathbf{A}$, the input of each algorithm is each row of $\mathbf{A}$. In this scenario, CNN has only one size filter $\mathbf{w} \in^{1*n}$.

Deep neural networks vs. Baselines. Table 1 shows the accuracy of vandal detection using the adjacency matrix $\mathbf{A}$ and spectral coordinates $\mathbf{x}_u$. We can observe that DAE and CNN models outperform the baselines significantly in all settings at the 10^{-8} level with a t-test. In the spectral space, the performance of SVM has a big jump when the percentage of training data increasing from 5% to 10%. In the adjacency matrix space, the performances of SVM do not improve while increasing the size of training data. It indicates that the SVM cannot be well-trained with a small training dataset, especially when the input has high dimensions. The accuracies of k-NN increase steadily while increasing the training data, but the accuracies of k-NN are also much worse than our DAE and CNN.

Spectrum vs. Adjacency matrix. In Table 1, we further observe that using the spectral coordinates as inputs ($\mathbf{x}_u$) achieves significantly better performance than using the adjacency matrix, especially when the percentages of training data are 5% and 10%. Meanwhile, in the spectral space, when the percentage of training dataset is 5%, DAE performs better than CNN since DAE pre-trains the model first. The pre-training phase of DAE encodes the information of nodes into hidden layers, which make the classifier predict the node labels with small training data. On the contrary, when the percentage of training dataset is larger than 5%, CNN outperforms DAE in the spectral space. It indicates CNN has better performance with enough training data. In the adjacency matrix space, the performances of DAE are worse than CNN with various sizes of training data. This is because DAE has much more parameters than CNN when using the adjacency matrix. DAE cannot be well-trained in a high-dimensional space with a small training dataset.

Effect of the dimension of spectral coordinate k. Table 2 compares the deep neural networks with k-NN and SVM on varying the dimension of spectral coordinate k. In this experiment, we use 20% of nodes as training data and the rest of nodes as testing data. When inputs of algorithms are $\mathbf{x}_u$, we can observe that DAE and CNN models outperform classical classifiers with various dimensions of spectral coordinates. We can further discover that DAE and CNN also achieve the most stable accuracy with various dimensions of spectral coordinates. However, the performance of SVM significantly drops while increasing the dimension of the spectral coordinate. This is because both DAE and CNN learn the hidden representations of nodes from their spectral coordinates. The hidden representations of nodes are useful for predicting the labels.

Neighbor inclusion vs. Neighbor exclusion. In our experiment, the inputs of DAE and CNN combine spectral coordinates of nodes and their 1-step neighbors. We further compare the performance of algorithms that adopt the node spectral coordinate with and without combining the 1-step neighbors' spectral coordinates as inputs in Table 2. We can observe that when using the information of the 1-step neighbors' spectral coordinates, the accuracies of all algorithms achieve around 1%-2% improvement. Therefore, combining the information of node neighbors can improve the performance of fraud detection.

Execution time. We also compare the execution time of deep neural networks using spectral coordinates and the adjacency matrix. The DAE and CNN models are trained on a Nvidia Tesla K20 GPU. We observe that when the ratio of training data is 20%, the training time of each epoch for CNN and DAE models with adjacency matrix is 2 seconds. On the contrary, the training time of each epoch for CNN and DAE models with spectral coordinates is less than 1 second. Therefore, using the spectral coordinates with low dimension is also more efficient than using the adjacency matrix.

Table 2: The accuracy of vandal detection with various dimensions of spectral coordinate k when 20% of nodes are used as the training dataset. We further compare algorithms using node spectral coordinate with/without combining neighbor spectral coordinates as inputs to algorithms.

Input	Algorithm	Dimension of spectral coordinate k				
		10	20	30	40	50
$\mathbf{x}_u$	k-NN	79.90%	78.85%	78.19%	78.33%	77.52%
	SVM	81.00%	81.40%	81.15%	80.70%	76.28%
	DAE	81.86%	81.65%	81.92%	81.87%	81.61%
	CNN	**82.24%**	**82.26%**	**82.61%**	**82.42%**	**82.47%**
$\boldsymbol{\alpha}_u$	k-NN	78.00%	77.49%	76.75%	76.55%	76.92%
	SVM	79.79%	79.65%	80.19%	80.31%	77.94%
	DAE	80.47%	81.10%	81.17%	81.20%	**81.40%**
	CNN	**80.52%**	**81.22%**	**81.48%**	**81.44%**	**81.39%**

4 CONCLUSIONS

We have presented a novel framework that applies deep neural networks on the spectral space of a signed graph to identify frauds. In particular, we first conduct graph spectral projections on a signed graph to obtain node spectral coordinates. The node and its s-step neighbors' spectral coordinates are combined together as inputs to the deep autoencoder and convolutional neural network models for fraud detection. The experiment results show that both deep neural networks achieve promising results on fraud detection. Our empirical evaluation further shows that combining the information of node neighbors can improve the effectiveness of deep neural networks on fraud detection.

ACKNOWLEDGMENTS

The authors acknowledge the support from the 973 Program of China (2014CB340404) to Shuhan Yuan and from National Science Foundation to Xintao Wu (1564250), Jun Li (1564348) and Aidong Lu (1564039). This research was conducted while Shuhan Yuan visited University of Arkansas.

REFERENCES

[1] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *DMKD* 29, 3 (2015), 626–688.
[2] Pierre Baldi. 2012. Autoencoders , Unsupervised Learning , and Deep Architectures. In *Workshop on Unsupervised and Transfer Learning*. 37–50.
[3] FabrÃŋcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and VirgÃŋlio Almeida. 2010. Detecting spammers on twitter. In *CEAS*.
[4] Y. Bengio, A. Courville, and P. Vincent. 2013. Representation Learning: A Review and New Perspectives. *TPAMI* 35, 8 (2013), 1798–1828.
[5] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering Large Groups of Active Malicious Accounts in Online Social Networks. In *CCS*.
[6] Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, and Shiqiang Yang. 2014. CatchSync: Catching Synchronized Behavior in Large Directed Graphs. In *KDD*.
[7] Srijan Kumar, Francesca Spezzano, and V.S. Subrahmanian. 2015. VEWS: A Wikipedia Vandal Early Warning System. In *KDD*.
[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
[9] X. Ying, X. Wu, and D. Barbará. 2011. Spectrum based fraud detection in social networks. In *ICDE*. 912–923.