Detecting Fraudulent Personalities in Networks of Online Auctioneers

Duen Horng Chau, Shashank Pandit, and Christos Faloutsos

School of Computer Science, Carnegie Mellon University
{dchau, shashank, christos}@cs.cmu.edu

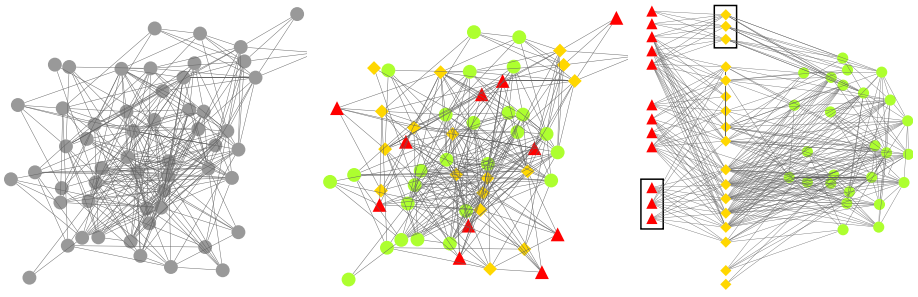
Abstract. Online auctions have gained immense popularity by creating an accessible environment for exchanging goods at reasonable prices. Not surprisingly, malevolent auction users try to abuse them by cheating others. In this paper we propose a novel method, *2-Level Fraud Spotting* (*2LFS*), to model the techniques that fraudsters typically use to carry out fraudulent activities, and to detect fraudsters preemptively. Our key contributions are: (a) we mine *user level features* (e.g., number of transactions, average price of goods exchanged, etc.) to get an initial belief for spotting fraudsters, (b) we introduce *network level features* which capture the interactions between different users, and (c) we show how to combine both these features using a *Belief Propagation* algorithm over a *Markov Random Field*, and use it to detect suspicious patterns (e.g., unnaturally close-nit groups of people that trade mainly among themselves). Our algorithm scales linearly with the number of graph edges. Moreover, we illustrate the effectiveness of our algorithm on a real dataset collected from a large online auction site.

1 Introduction

Given a set of transactions among online auction users, how do we spot fraudsters? Suppose we want to transact with a user *u*, and we want to know how honest he is. Suppose we also have a lot of historical information (product names, amounts sold for, feedbacks from other users, timestamps, etc.), and that we also have a list of user IDs, who have committed frauds in the past. Currently, users of online auction sites can view the past feedbacks of a user *u*, which may very well be fabricated. How can we include the vast amount of historical information about the user and his trading partners, to spot fraud more effectively? In this paper we present *2LFS*, the first systematic approach to attack auction fraud.

Online auctions have gained immense popularity. For example, eBay, the world's largest auction site, had over 192.9 million registered users at the end of Q1 2006, a 31% increase over the previous year [4]. Unfortunately, auction frauds happen, and they are by far the most serious problems that auction sites face today. In 2005, the Internet Crime Complaint Center (IC3) received 231,493 complaints, 62.7% of which were auction frauds. 41% of the victims reported monetary loss with an average loss of $385 [7]. In some elaborate fraud schemes, the total incurred loss was in the order of millions [10].

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): PKDD 2006, LNAI 4213, pp. 103 – 114, 2006.



(a) Initial (b) Labeled (c) Manually labeled

Fig. 1. *2LFS* in action: (a) given graph (b) after labeling by *2LFS*: fraud (red triangles), honest (green circles), "accomplices" (yellow diamonds) (c) after manual rearrangement, to highlight the "bipartite cores". The nodes in the two black rectangles are confirmed fraudsters.

The goal of our work is to treat the auction fraud problem systematically, using data mining and machine learning techniques to spot unnatural patterns in auctions. We propose the *2LFS* algorithm, and illustrate its effectiveness on real, public data from a large auction site. Figure 1(a) illustrates a small graph from the large auction site, in which it is difficult to spot any suspicious patterns. The result of labeling by *2LFS* is shown in Figure 1(b). Fraudsters are the red triangles and honest users are the green circles. The yellow diamonds correspond to *accomplices*, which we will discuss in detail later in the paper. Figure 1(c) shows the same graph after manual rearrangement so that nodes with the same label are grouped together. Now we can clearly observe the existence of a *bipartite core* between the fraudsters and accomplices. As we will explain later, such bipartite cores are a tell-tale sign of a popular fraud scheme. In fact, the nodes in the two rectangles in Figure 1(c) are confirmed fraudsters, who have received many negative feedbacks from buyers who had paid for items that never got delivered.

The rest of the paper is organized as follows. Section 2 provides an overview of related work. Section 3 describes the auction fraud detection problem. Section 4 describes in detail the *2LFS* algorithm. Section 5 provides empirical evidence for the effectiveness, robustness and scalability of our method. Section 6 discusses some observations on how easily we can generalize our method to other fraud detection problems. Finally, we present a brief summary of our results in Section 7 with pointers for future work.

2 Related Work

To the best of our knowledge, this is the first work that uses a systematic approach to analyze and detect electronic auction frauds. We survey earlier attempts to detect such frauds, as well as literature related to trust propagation.

Auction Frauds and Reputation Systems. Reputation systems are extensively used by electronic auctions to prevent frauds. Although helpful, these systems are very simple and can be easily foiled. To study the effectiveness of today's reputation systems,

Melnik et al. [9] and Resnick et al. [14] conducted empirical studies which showed that selling prices of goods are positively affected by the seller's reputation. In an overview, Resnick et al. [13] summarized that today's reputation systems face many challenges which include the difficulty to elicit honest feedback and to show faithful representations of users' reputation. Common-sense approaches to avoid frauds can be easily found on Web sites [5] and in news articles [15]. However, they require people to invest substantial amount of time and to constantly maintain a high level of vigilance, something that the average person cannot afford to do. Some researchers [3] have categorized auction fraud into different types, but they have not suggested any formalized methods to deal with them.

Trust and Authority Propagation. Authority propagation, an area highly related to fraud detection, has been studied by milestone papers and systems, and specifically by PageRank [1] and HITS [8] which treat a Web page as "important" if other important pages point to it, thus propagating the importance of pages over the Web graph. However, none of them explicitly focus on fraud detection. Trust propagation was used by TrustRank [6] to detect Web spam, and their goal was to distinguish between "good" sites and "bad" sites (like phishers, adult-content sites, etc). Also related is the work by Neville et al. [11, 12], where the goal is to aggregate features from neighboring nodes, to do classification, in movie and stock databases.

None of the above techniques focuses on a systematic way to do online auction fraud detection, which is the focus of our work.

3 Problem Description

We define our *Auction Fraud Detection Problem* as follows: **given** (a) a user of interest u (b) the historical information of user u, as well as of many other users and (c) the fact that some of them are known fraudsters, **find** whether user u is a potential fraudster.

We focus on describing the setup of eBay as other auction sites work similarly. A new user begins by registering an ID (also called "handle") with the site. The user may then buy or sell items through bidding and auctioning. All auctions are time-stamped and detailed information about auctions occurring in the last six months is usually available on the site. After a transaction, the site allows the buyer and the seller to rate each other on a scale of positive, neutral and negative (1, 0, -1) and leave a brief comment (e.g., "Great buyer! Prompt payment."). These ratings are added up to form the feedback score of a user. Other users can see the score of a given user before they choose to transact with him. The key idea of the feedback system is to provide an estimate of trustworthiness for each user, that future dealers can consult.

The most prevalent auction fraud is the non-delivery fraud, where the fraudster receives a payment for an item but never delivers it. To be able to commit such a fraud, fraudsters try to devise methods to trick the reputation system and boost their feedback score (we will see how this is done later in the paper.) Other types of frauds include selling faulty, counterfeit, or stolen goods.

4 2-Level Fraud Spotting (*2LFS*) Algorithm

We now present the *2LFS* algorithm, which tackles the fraud detection problem in two steps: (1) it examines *user level features*, i.e., information intrinsic to individual users (e.g., "age" of the user, the number and prices of items sold/bought, the burstiness of the transaction times, etc.), and (2) it examines *network level features* to detect suspicious patterns in the network of transactions between users.

4.1 User Level Features

Auction sites keep records of their users. For each user, we can divide the stored information into two parts, *profile* and *past transactions*. To determine a set of user level features that distinguish fraudsters from honest users, we begin by learning from frauds that were widely publicized in newspaper articles and examining the involved fraudsters. Our observations indicate that fraudsters tend to be short-lived, they exhibit bursty trading patterns (many, fake sales, on a single day) and a bi-modal distribution of prices (cheap items to real, honest users; fictitious, expensive items to their alter-egos).

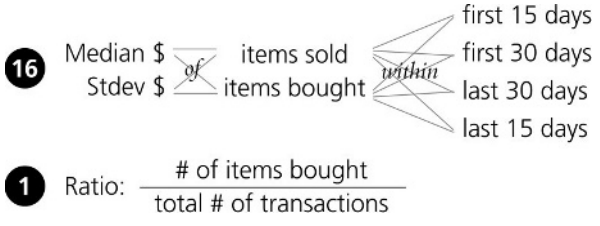


Fig. 2. 17 user level features

We believe that the trends (medians) and fluctuations (standard deviations) in the prices of items traded over time (first 15 days, first 30 days, etc) are the most important features that we should use for classification, as they have direct relevance to fraudsters' investments, costs and profits. The final set of 17 features is summarized in Figure 2. For example, one of the features is *the standard deviation of prices of items sold within the first 15 days since the user registered*. These features were previously evaluated to achieve a precision of 82% and a recall of 83% on some real eBay test data [2].

The feature values can be extracted from the profiles and transaction history of users, available from the Web. The class labels (fraud/honest) are derived manually, by inspecting users with many negative feedback scores. We train a decision tree with the C5.0 classification system, and use this tree to classify other user nodes. These class labels are then fed into the network level detection algorithm described in the next section.

4.2 Network Level Features

Transactions between users can be modeled as a graph, with a node for each user and an edge for all the transactions between two users. As is the case with hyperlinks on

the Web, an edge between two nodes can be assigned a definite semantics, and can be used to propagate properties from one node to its neighbors. For instance, an edge between two nodes can be interpreted as an indication of similarity in their behavior, since honest users will interact more often with other honest users, while fraudsters will interact in small cliques of their own. This semantics is very similar in spirit to the one used by TrustRank[6]. However, preliminary experiments with our dataset, as described in Section 5, suggest that fraudsters do not directly interact with other fraudsters, as this could cause them to suffer extensive "loss" relatively easily – suppose one of the fraudulent account involved in a fraud is exposed, the auction site may easily identify and void other fraudulent accounts in the clique, which would destroy the "infrastructure" that the fraudster had invested in for carrying out the fraud. To carry out another fraud, the fraudster will have to re-invest efforts in building a new clique.

A bit of manual inspection of the data unveiled an alternate way in which fraudsters behave to build a reusable infrastructure for their fraudulent activities. They create several identities and arbitrarily split them into two categories – *fraud* and *accomplice*. The fraud identities are eventually used to carry out the actual fraud, while the accomplices exist only to help the fraudsters by boosting their feedback rating. Accomplices themselves behave like perfectly legitimate users and interact with other honest users to achieve high feedback ratings. On the other hand, they also interact with the fraud identities to form near-bipartite cores, which helps the fraud identities gain high feedback ratings. Once a fraud is committed, the fraud identities get voided by the auction site, but the accomplices manage to beat contemporary fraud detection schemes (owing to their interactions with honest users) and linger around for reuse by new fraudsters. Through *2LFS*, we propose a systematic way to model the network level interactions between users and identify suspicious graph patterns. In a nutshell, each node is given three scores (fraud-, accomplice-, honest-scores), and we update these scores to be in harmony with the neighbors' scores.

The Markov Random Field Model. Markov Random Fields (MRFs) are a class of graphical models particularly suited for solving inference problems with uncertainty in observed data. The data is modeled as a graph with two types of nodes – *hidden* and *observed*. Observed nodes correspond to values that are actually observed in the data. For each observed node, there is a hidden node which represents the true state underlying the observed value. The state of a hidden node depends on the value of its corresponding observed node as well as the states of its neighboring hidden nodes. These dependencies are captured via an *edge compatibility function* $\psi(\sigma,\sigma')$ and a *node compatibility function* $\phi(\sigma,\omega)$. $\psi(\sigma,\sigma')$ gives the probability of a hidden node being in state σ' given that it has a neighboring hidden node in state σ. $\phi(\sigma,\omega)$ gives the probability of a node being in state σ given that its corresponding observation was ω.

We model the auction users and their mutual transactions as a MRF. We create a hidden node for each user, which can be in any of three states – fraud, accomplice, and honest. Let us denote this set of possible states by *S*. A transaction between two users is represented by an edge between their corresponding hidden nodes. With each hidden node *n*, we associate a belief vector $\mathbf{b}_n$, such that $\mathbf{b}_n(\sigma)$ equals the probability of

node *n* being in state σ (which we call the *belief* of node *n* in state σ). Further, each hidden node is also associated with an observed node, which corresponds to our initial (and possibly noisy) observation of its state.

	Fraud	Accomplice	Honest
Fraud	ε_p	$1-2\varepsilon_p$	ε_p
Accomplice	0.5	$2\varepsilon_p$	$0.5-2\varepsilon_p$
Honest	ε_p	$(1-2\varepsilon_p)/2$	$(1-2\varepsilon_p)/2$

Fig. 3. The Propagation Matrix for an edge. Entry (*i*, *j*) gives the conditional probability that the destination node is at state *j*, when the source node is at state *i*.

	Fraud	Honest
Fraud	$1-\varepsilon_o$	ε_o
Accomplice	0	0
Honest	ε_o	$1-\varepsilon_o$

Fig. 4. The Observation Matrix for an edge. Entry (*i*, *j*) gives the *observed* probability that the destination node is at state *j*, when the source node is at state *i*.

To completely define the MRF, we need to instantiate the compatibility functions ψ and ϕ. For now, let us assume that we do not have an initial observation about the states of any of the nodes, and choose ϕ such that $\phi(\sigma,\omega)=1/|S|, \forall\sigma,\omega$. The edge compatibility function can be viewed as a matrix (which we call the *Propagation Matrix*) of dimension $|S|\times|S|$. Figure 3 shows a sample instantiation of the propagation matrix based on the following intuition: a fraudster heavily links to accomplices but not to other fraudsters; an accomplice links to both fraudsters and honest nodes, with a higher affinity for fraudsters; a honest node links to other honest nodes as well as accomplices (since an accomplice effectively appears to be honest to the innocent user.) In our experiments, we set ε_p to 0.05. We would like to set ε_p to zero, but this would create numerical problems with multiplications. Thus we set it to a small value, to denote the fact that it is highly unlikely that a fraudster will have a transaction with another fraudster. Ideally, we would "learn" the value of ε_p, as well as the form of the propagation matrix itself, if we had a large training set.

The Belief Propagation Algorithm. The Belief Propagation (BP) algorithm has been successfully applied to a variety of disciplines (bayesian networks, MRFs, error-correcting codes, etc.) In all of its applications, BP takes as input some form of a network of nodes, each of which can be in a finite number of states. Some encoding of how the state of a node influences its neighbors is also known beforehand. The BP algorithm then infers the posterior state probabilities of all nodes in the network given the observed states of some of the network nodes. We refer the reader to [16] for an excellent discussion on the BP algorithm and its generalizations to various problems.

Here, we present the version of BP suitable for MRFs. The algorithm functions via iterative *message passing* between the different nodes in the network. Let $\mathbf{m}_{ij}$ denote the message that *i* passes to *j*. The message $\mathbf{m}_{ij}$ is a vector with 3 values (fraud-, accomplice- and honest-score), and it represents *i*'s opinion about the belief of *j*. At every iteration, each node *i* computes its belief based on messages received from its

neighbors, and uses the propagation matrix to transform its belief into messages for its neighbors. Mathematically,

$$\mathbf{m}_{ij}(\sigma) \leftarrow \sum_{\sigma'} \psi(\sigma', \sigma) \prod_{n \in N(i) \setminus j} \mathbf{m}_{ni}(\sigma); \qquad \mathbf{b}_i(\sigma) = k \prod_{j \in N(i)} \mathbf{m}_{ji}(\sigma) \tag{1}$$

where k is a normalization constant to make the beliefs sum up to 1. Initially, a suitable prior on the beliefs of the nodes is assumed. The algorithm then proceeds by iteratively passing messages between nodes based on previous beliefs, and then updating beliefs based on the passed messages. The purpose of iteration is to reach a fixed point (equilibrium), that is, status-assignments to nodes that are as compatible with their neighbors as possible. The iteration is stopped as soon as the beliefs converge, or a maximum limit for the number of iterations is exceeded. Theoretically, convergence is not guaranteed, although in practice, BP has been found to converge quickly to reasonably accurate solutions.

4.3 Merging the Two Levels – *2LFS*

We now present the *2LFS* algorithm, which combines the user level features (Section 4.1) with the network level features (Section 4.2) to detect suspicious patterns in a graph of transactions between online auction users.

We treat the user level features as noisy observations of the states of users, and use them to instantiate the observed values of nodes in the MRF model for the network level features. We believe that such a combination would yield the following benefits: (a) suitable prior knowledge will help the belief propagation to converge to a more accurate solution in less time, and (b) incorrect inference at the user level can be corrected by the network level propagation of features.

To combine these observations with the belief propagation, we need to modify the previously stated instantiation of the node compatibility function ϕ. Recall that $\phi(\sigma, \omega)$ gives the probability of a hidden node being in state σ given that the corresponding observation was ω. Let the domain of observed values be Ω. Then, the function ϕ can be encoded as a matrix (which we call the *Observation Matrix*) of dimension $|S| \times |\Omega|$. In our case, the user level features classify users into only two categories – fraud and honest. A sample instantiation of ϕ is shown in Figure 4. ε_o can be interpreted as the uncertainty in observation at the user level. In our experiments, we set $\varepsilon_o = 0.2$.

Let ω_i denote the observed value for node i. To incorporate the effect of the observed values, the update rules in Equation 1 can be extended as follows:

$$\mathbf{m}_{ij}(\sigma) \leftarrow \sum_{\sigma'} \phi(\sigma, \omega_i) \psi(\sigma', \sigma) \prod_{n \in N(i) \setminus j} \mathbf{m}_{ni}(\sigma); \quad \mathbf{b}_i(\sigma) = k \phi(\sigma, \omega_i) \prod_{j \in N(i)} \mathbf{m}_{ji}(\sigma) \tag{2}$$

These equations together constitute the *2LFS* algorithm. The pseudo code for the same is provided in Figure 5.

```
Input:  A: Adjacency matrix, N: Number of nodes, O: Initial observations
        S: Number of states, ψ: Propagation matrix, φ: Observation matrix
Output: B: Belief matrix

NETWORK-LFS()
 1  B = NEWMATRIX(N, S); M = NEWMATRIX(N, N)
 2  for i in 1 to N
 3    M[i][i] = message for observation O[i] /* Initialize priors from user level observations */
 4  while (not converged)
 5   for i in 1 to N
 6     m = NEWMESSAGE() /* Create an empty message */
 7     for k in NEIGBORS(i)
 8       m = MULTIPLYMESSAGE(m, M[k][i]) /* Accumulate messages from all neighbors */
 9     for j in NEIGHBORS(i)
10       m_j = DIVIDEMESSAGE(m, M[j][i]) /* ignore message sent by the neighbor itself */
11       M[i][j] = PROPAGATEMESSAGE(m_j, ψ) /* propagate message to the neighbor */
12   /* Now compute the normalized beliefs for each node */
13  for i in 1 to N
14    m = NEWMESSAGE() /* Create an empty message */
15    for j in NEIGHBORS(i)
16      m = MULTIPLYMESSAGE(m, M[j][i]) /* Accumulate messages from all neighbors */
17    m = NORMALIZEMESSAGE(m) /* Normalize the beliefs */
18    for s in 1 to S
19      B[i][s] = m[s] /* Store beliefs in the belief matrix */
20  return B
```

Fig. 5. Pseudo code for network-LFS

5 Experiments

Here we describe the experiments we did, on real and synthetic datasets. Our goal was to answer the following questions:

1. *Robustness*: how well does *2LFS* work for fraud detection when the topology of the auction graph deviates from the ideal bipartite core setting?
2. *Effectiveness*: how effective is *2LFS* in focusing our attention on suspicious bipartite cores occurring in real auction data?
3. *Scalability*: how well does the network level belief propagation scale with graph size?

The algorithm was implemented in C++ and the experiments were performed on a desktop running Red Hat Linux 9 on a Intel P4 3.00GHz processor, with 1GB RAM, 25GB disk space, and a 100Mbps internet connection.

Robustness of *2LFS*. Graphs observed in practice will almost never have exact bipartite cores; there will always be some "missing" edges. To successfully identify suspicious

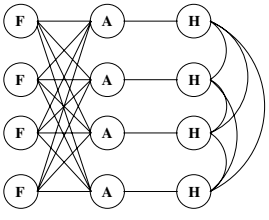


Fig. 6. $G_{3\times4}$ with labels assigned by *2LFS*

nodes in such settings, *2LFS* should be robust in nature and be able to tolerate minor deviations from the ideal scenario. In this section, we describe an experiment we designed to systematically test the robustness of *2LFS*.

We generated synthetic graphs which mimic ideal fraudulent behavior, and then randomly deleted a few edges from them. We started with the graph shown in Figure 6 (called $G_{3\times4}$) which contains 12 nodes – 4 fraud, 4 accomplice and 4 honest. This graph closely agrees with the propagation matrix shown in Figure 3. *2LFS* when run on top of $G_{3\times4}$, converges in 3 iterations and assigns correct labels to all the nodes. Similar results were observed for $G_{3\times x}$ with x varying from 5 to 20.

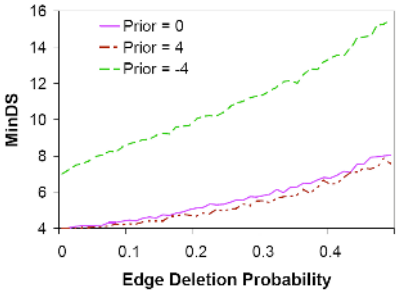


Fig. 7. Min Detection Size vs noise - lower is better: *2LFS* is robust to minor deviations in graph structure, and even to wrong priors

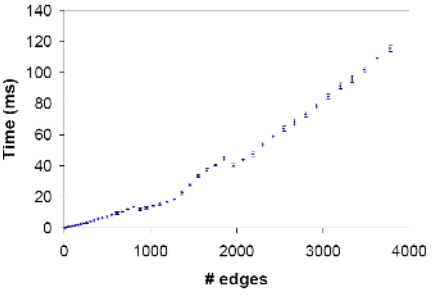


Fig. 8. *2LFS* scales almost linearly with the number of edges

Next, we deleted edges in $G_{3\times x}$ with a fixed probability p (called the *Edge Deletion Probability*). The lowest value of x for which *2LFS* produces the correct labeling is called the *Minimum Detection Size* (MinDS) for the given edge deletion probability. Further, to understand how critically *2LFS* depends on the user level features, we introduced prior observations for some of the nodes in $G_{3\times x}$. Our observations are summarized in Figure 7. Each curve in this figure corresponds to a specific prior value. A prior of 0 means all the nodes were initialized with unbiased priors, a positive prior of z means z nodes were initialized with the correct prior observation, while a negative prior of $-z$ means z nodes were initialized with an incorrect prior. With unbiased priors and edge deletion probabilities below 0.5, the MinDS is 9, which indicates that for large real-world graphs *2LFS* can be expected to robustly tolerate deviations from the ideal $G_{3\times x}$ scenario. Further, minor changes in the prior do not seem to significantly affect the stability of *2LFS*. In case the prior knowledge is correct, performance is improved, while (interestingly) in case the prior knowledge is incorrect, the network level features are able to offset the error in the prior and *2LFS* still converges to the correct solution.

Effectiveness of *2LFS*. To test the effectiveness of *2LFS* we decided to use a dataset crawled from eBay, the world's largest auction site. EBay allows public access to the profiles and feedbacks of (almost) all its users. The feedbacks of a user tell us about

other users with whom he has interacted in the past. The pricing information of the items exchanged is also available only for transactions over the last six months.

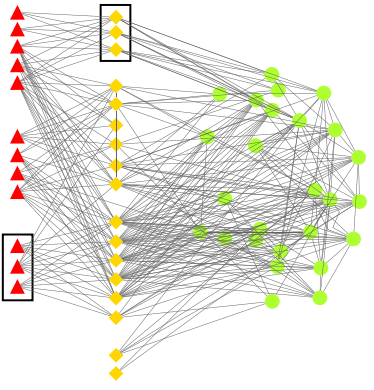


Fig. 9. Labeling of eBay users output by *2LFS*

To propagate the network level features, we rely on a complete and accurate description of the graph. However, user level features (i.e., pricing information) are available only for the last six months. Since the utility of our chosen user level features is evident from [2], we focused on evaluating the effectiveness of the network level features alone, and we set all belief scores to 1/3, for all nodes.

The data was collected by a breadth-first crawl starting from 5 users. The resulting graph had 55 nodes and 620 edges. We then manually observed the feedbacks for each of the 55 users and found six of them to be confirmed fraudsters. Next, we ran *2LFS* over this graph and recorded the labeling assigned to each node in the graph. The entire graph arranged as per the labellings produced by *2LFS* is shown in Figure 9 (same as Figure 1(c)). The nodes labeled as fraud and accomplice form a near bipartite core, one end of which is disconnected from the rest of the graph. As mentioned earlier, the existence of such disconnected bipartite cores is unnatural and suspicious. Moreover, all the six confirmed fraudsters were found to be a part of this core. Thus, *2LFS* clearly succeeds in drawing our attention to suspicious patterns in the graph. Such cores, once identified, can be used to predict which users are likely to commit frauds in the future or are serving to boost the feedback ratings of fraudsters. We believe this aspect of *2LFS* is invaluable in the context of fraud detection.

Scalability of *2LFS*. To test the scalability of our algorithm, we chose to generate synthetic graphs, since we are able to systematically control their sizes and structures. We measured the time taken by *2LFS* to execute over $G_{3\times x}$ for various values of x (averaged over 100 runs.) The results are shown in Figure 8. We observe that the absolute amount of time taken is very small (less than 0.15 seconds for $G_{3\times 3500}$.) Moreover, the running time appears to grow linearly with the number of edges, which indicates that *2LFS* can easily scale to very large real-world graphs.

6 Discussion

We would like to emphasize some important observations. The first is the generality of our method, and the second is its potential for heavy impact on fraudsters.

	Fraud	**Honest**
Fraud	$1-\varepsilon$	ε
Honest	ε'	$1-\varepsilon'$

Fig. 10. Propagation matrix for clique detection ($\varepsilon << \varepsilon'$)

Generality. Thanks to the propagation matrix, *2LFS* is general in applicability. The propagation matrix of Figure 3 can spot bipartite cores. With different instantiations of the propagation matrix, we might be able to spot a broader variety of graph patterns. For example, near cliques could be spotted by the propagation matrix shown in Figure 10.

Making fraud unprofitable. Not only does *2LFS* find confirmed fraudsters, it also spots the accomplices, who help the fraudsters and can themselves commit frauds in the future. Accomplices are valuable to fraudsters, because they take time and effort to build, and they provide a reusable infrastructure for fraudsters to build positive feedbacks quickly. Spotting accomplices can be a hard blow to fraudsters, because accomplices take much more time, money and effort to create and manage. In response, the fraudsters might resort to more sophisticated schemes to hide their evils. However, these schemes will require more effort and cost, thus making fraud increasingly unprofitable for them.

7 Conclusions

We have shown how to use data mining, machine learning and trust propagation methods to address the problem of fraud detection in the complex settings of auction sites. Users and their respective transactions form a rich social network, with much more information than just nodes and links – feedbacks, timestamps, the prices and types of items sold, and more. To handle the complexity of the problem and to exploit useful pieces of information hidden in the social network of auction users, we propose *2LFS*, a novel, two-step algorithm that merges user level and network level information to detect fraudulent users. Our main contributions include:

- The careful extraction of user level features
- The use of belief propagation and MRFs to combine user level and network level features
- Experiments on synthetic and real data, proving the robustness, scalability, and effectiveness of *2LFS*.

Future research directions include the generalization of *2LFS*, so that it can automatically learn the propagation matrix from data, and the inclusion of game theory, to

anticipate and guard against new fraud schemes, in addition to Fraud-Accomplice bipartite cores.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grants No. IIS-0209107 SENSOR-0329549 IIS-0534205. Additional support was provided by by the Pennsylvania Infrastructure Technology Alliance (PITA), by Intel, NTT and Hewlett-Packard. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

References

1. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In Seventh international conference on World Wide Web 7 (1998) 107–117
2. D. H. Chau and C. Faloutsos. Fraud detection in electronic auction. In European Web Mining Forum at ECML/PKDD (2005).
3. C. Chua and J. Wareham. Fighting internet auction fraud: An assessment and proposal. In Computer, Vol. 37 no. 10 (2004) 31–37,
4. eBbay 2006 1Q financial results. http://investor.ebay.com/releases.cfm (2006)
5. Federal trade commission: Internet auctions: A guide for buyers and sellers. http://www.ftc.gov/bcp/conline/pubs/online/auctions.htm (2004)
6. Z. Gyongyi, H. G. Molina, and J. Pedersen. Combating web spam with TrustRank. In VLDB (2004) 576–587
7. IC3 2004 internet fraud - crime report. http://www.ifccfbi.gov/strategy/statistics.asp (2005)
8. J. Kleinberg. Authoritative sources in a hyperlinked environment. ACM (JACM), Vol. 46 (1999) 604–632
9. M. Melnik and J. Alm. Does a seller's ecommerce reputation matter? Evidence from eBay auctions. Industrial Economics, Vol. 50 (2002) 337–49
10. Msnbc: Man arrested in huge ebay fraud. http://msnbc.msn.com/id/3078461/ (2003)
11. J. Neville and D. Jensen. Collective classification with relational dependency networks. In 2nd Multi-Relational Data Mining Workshop, 9th ACM SIGKDD (2003) 77–91
12. J. Neville, . Simsek, D. Jensen, J. Komoroske, K. Palmer, and H. Goldberg. Using relational knowledge discovery to prevent securities fraud. In 11th ACM SIGKDD (2005) 449–458
13. P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems. Communications of the ACM, Vol. 43 (2000) 45–48
14. P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on eBay: A controlled experiment. (2003)
15. USA Today: How to avoid online auction fraud. http://www.usatoday.com/tech/columnist/2002/05/07/yaukey.htm (2002)
16. J. S. Yedidia,W. T. Freeman, and Y.Weiss. Understanding belief propagation and its generalizations. Exploring artificial intelligence in the new millennium (2003) 239–269