Ontology-Based Fraud Detection

Liu Fang, Ming Cai, Hao Fu, and Jinxiang Dong

College of Computer Science and Technology, Zhejiang University,
Hangzhou 310027, China P.R.

fangliuqq@gmail.com, cm@zju.edu.cn,
zjufuhao@gmail.com, djx@zju.edu.cn

Abstract. One method for detecting fraud is to check for suspicious changes in user behavior. This paper proposes a novel method, built upon ontology and ontology instance similarity. Ontology is now widely used to enable knowledge sharing and reuse, so some personality ontologies can be easily used to present user behavior. By measure the similarity of ontology instances, we can determine whether an account is defrauded. This method lows the data model cost and make the system very adaptive to different applications.

Keywords: Fraud detection, ontology, ontology instance similarity, behavior, personality.

1 Introduction

According to Miniwatts Marketing Group [1], there are more than 1 billion Internet users now. Many wet sites ask suffers to online registration at the first visit and logon every other visit to offer complete service, such as recommendations, resource access control, especially in e-commerce system.

Many systems send the logon information to users to let them check whether someone others have logged in the system with their accounts, but we find that most of the users discard these messages. For complex system, this mechanism is not enough. During an invalid access, an invalid user may damage the system, although most of the system has a back up mechanism, its important to detect these invalid accesses as soon as possible.

People have their own usual practice when suffering. When a user logins to the system, his/her activities are logged and these activities are compared to those before, if the results are quite different, we can constrain his/her access or privileges, such as access prohibition or message post prohibition, which can be configured by system administrator.

In [2], the authors have discussed some fraudulent behavior, such as in areas of insurance, credit card, and telecommunications. And a brief description of the fraud detection method and applications is given.

The rest of the paper is organized into the following sections. Section 2 discusses related work, and section 3 presents the overview of our system. In section 4, we provide a discussion of how user's activity is colleted, classified, and presented. We give a method to measure the similarity of ontology instances in section 5. And finally, we show our experimental result and discuss some future work in section 6.

2 Related Works

One method for dealing with cloning fraud is to check for suspicious changes in user behavior. [3] describes the automatic design of user profiling methods for the purpose of fraud detection, using a series of data mining techniques. Specifically, they use a rule-learning program to uncover indicators of fraudulent behavior from a large database of customer transactions. Then the indicators are used to create a set of monitors, which profile legitimate customer behavior and indicate anomalies. Finally, the outputs of the monitors are used as features in a system that learns to combine evidence to generate high-confidence alarms. [4] presents a network intrusion detection models. They encode and compare the frequent patterns mined from the normal and intrusion datasets, and automatically construct statistical and temporal features that describe the anatomy and invariant behavior of the attacks.

In [5], they use activity monitoring to term a problem class which involves monitoring the behavior of a large population of entities for interesting events requiring action. They differentiate activity monitoring problems and present a framework within which each of the individual problems has a natural expression, as well as a methodology for evaluating performance of activity monitoring techniques.

From [2, 6], we can know the main idea of fraud detection now used is classification of feature patterns. But the classifier must be trained with examples and it may take many efforts to create these examples, for examples, in [3], they use thirty-day's record as the examples. Another problem is that the figures collected by the system may difficult for people to understand.

To prevention and early detection of fraudulent activity is an increasingly important goal for the EU and its Member States [7]. The FF POIROT (Financial Fraud Prevention Oriented Information Resources using Ontology Technology) project has developed legal core ontology, financial ontology and fraud ontology. They are at their beginning and focus on ontology building. For web applications, such as e-commerce, we can build the user behavior ontology using technology described in recommendation system [8]. In recommendation system, the personality of suffers can be extracted from users' behaviors and used to help the recommendation. We use the personality information to detect fraudulent activity. In the early stage of the application, we use the personality similarity as the judgments. We find that the result is quite attractive. Once there are enough examples to train the classifier, classifiers can be used to improve the fraud detection accuracy.

3 System Overview

Figure 1 presents framework of our detector. The users' activities will be tracked by the web server and then stored into the activity database. The activity database is an ontology instance database where each account has its according activities. Once there are certain new activities in the database for an account, the trigger in the database will start the ontology instance similarity computation component. If new ontology is added, first use similarity of ontology instances for detecting, after collecting enough examples, classifies will be started. So the system is very flexible to different applications. Now many applications have already building useful ontology which we can use for activity monitoring, and this is important for applications to use the fraud detection.

Fig. 1. System Architecture

If one ontology is little useful for detection, it will be easily removed from the system, and other ontology also can be added. In [3], they use rule-learning program to uncover indicator of fraudulent activities. If activities are added, the rule-learning program must be run again to generate the rules, because rules must be found from the all activities. For activity ontology, the activities are already clearly classified when the ontology is modeled. Some features will be removed from the ontology if they appear no usefulness to fraud detection.

The newly added ontology doesn't immediately be outputted to classifier, we should determine the newly added ontology is useful for fraud detection before adding the new ontology to the classifiers. So instance similarity computation component can help classifier to reduce noises.

4 Activity Ontology

4.1 Data Preparation

The Semantic Squirrel Special Interest Group (SSSIG) [9] is a group of researchers based at the University of Southampton who aim to automate the process of logging available raw data, (or 'nuts'), that can describe aspects of one's personal experience. A number of squirrels have been developed in this process, and an ethos of the group is to preserve this raw data in order to retain any unforeseen potentials for exploitation and transcend issues pertaining to platform and application restrictions. Applications can surely add their own data to the system.

4.2 Data Classification

This raw data forms the basis of the knowledge acquisition phase and then is parsed into RDF representations. Based on work of [10], we use following types of preliminary data:

■ Time Data:

Time Data includes the time when user logins to, leave the web site, such as MORNING, AFTERNOON, TWILIGHT, EVENING and NIGHT, and the time intervals, such as time a user spends on a web pages, the reading speeds and so on.

■ Geographical Data

Through users' IP information, we can get the geographical data of users. Although many IPs can only indicate which city it belongs to, IP information is helpful for fraud detection if the user's IP is not dynamically allocated.

■ Navigation Habit Data

The Navigation Habit Data contains information about how user use bookmarks, how user use navigation history, and how user use back and forward button of the browser and so on. By virtue of its cross-platform nature, we have selected the Mozilla Firefox as our web-browser of choice. Firefox exposes the download information in RDF form and thus can be easily imported to the system. Scripts have been developed to parse the bookmarks and history data into RDF. The RDF model uses two namespaces taken from the mozilla developers centre.

■ Web Page Data

Web Page Data contains information about which types of web pages the user wants to browse, such as entertainment, political, sports and so on. We also extract some key words from text using natural language processing technology. Other types of data are optional, such as email, music playcount statistics, and file system information.

4.3 Data Presentation

All the data we collected are presented in RDF. Each time the user login to the system, we create new ontology instances or update the statistical information of some ontology instances periodically. We add a new property {rev, acc, per, tim} for each RDF triples. The meaning of the 3 parameters is:

rev is the relevance value. It is a real number ranging from 0 to 1. It gives a measure of how much a concept characterizes a web resource be accessed by a user. A bigger value denotes higher relevance. For example, if a user accesses a web page and the rev is 0.9 to a concept "political news", we know that the web page is very relevant to political news.

acc is the access times for a web resource be accessed by a user in a certain time. It is a positive real number.

per is the time period in which the acc should to be updated, such as 1 minute, 10 minutes, half an hour and so on.

tim denotes the average time a user linger on a web resource. When computing it, we should take the content of the resource into consideration. For web pages, we divide the time by the number of words of the text.

Figure 2 is an example:

Fig. 2. Example of ontology instance

From the figure, we can know that a user called Caucasus has viewed 5 sports news in 10 minutes, and the average time of each news he linger on is 3 minutes.

5 Ontology Instance Similarity

Once a user login to a system, his/her new data is colleted and compared to the data stored in the database. If the similarity value between the new behaviors and the behaviors before for this account is below a threshold, we can say that the user may be defrauded.

In a RDF graph (ontology instance), the leaf node is a node that doesn't have an output edge. Since a RDF graph is not a tree, so it may be more than one path from the root node to a leaf node.

For each edge created in the new session (the data collected from the login), the similarity value with the corresponding edges (be stored in the database) is defined as following:

sim_edg = (1-(rev_new-rev_old))*per/(acc_new*tim_new-acc_old*tim_old) (1)

*_new and *_old denotes the newly colleted data and data in the database respectively. The time period should be the same for the two corresponding edges.

For a path from the root node to a leaf node, the similarity is defined as following:

sim_path = (Σ(sim_edg_n / ((rev_new+rev_old)/2))) / n (2)

where th1 <= sim_edg / ((rev_new+rev_old)/2), and th1 is a threshold value. The edges with its sim_edg / ((rev_new+rev_old)/2) too little means that the edge can't well characterize a user's personality, so it won't be used for the fraud detection. For example, some events may change user's suffer interest, such as the holding of the Olympic Games, and the value of acc and tim may be much larger than the normal one.

The total similarity value of the two ontology instances is defined as following:

$$\text{sim_inst} = (\sum(\text{sim_path}_n))/n \quad (3)$$

the similarity value between the new instance and all the instances before:

$$\text{sim_onto} = (\sum(\text{sim_inst}_n))/n \quad (4)$$

and the similarity between the user's new behavior and before is:

$$\text{sim} = (\sum(w_n * \text{sim_onto}_n))/n \quad (5)$$

w_n is the weight for an ontology, and this is an empirical value. A larger weight value means the ontology can better characterize the behavior of a user. If the value of sim is smaller than a threshold, then a warning may be delivered and some action may be taken, such as inhibiting the access or inhibiting the user from uploading files.

When using the classifier, the classifier may be trained by certain ontology instances and the new ontology instance is classified by the classifier.

6 Experiments and Future Work

We have applied our method to a BBS for experimental test. In the BBS, normal users can read and post articles or pictures, send messages or mails to each other, upload little attachment ($\leq 1\text{M}$), and so on. Board manager has some other privileges, such as modifying or deleting posts. In BBS, there is a friend list and user can add himself to the list, we have disabled this function for the users attend the test, we don't want the user to know whether they are be frauded so they won't change their activities.

40 students have been invited to do the test, among them, 5 are board managers. They have been asked to use the BBS as they do before and 6 random chosen users are asked to be the faker for some time. We use 3 ontologies for preliminary data (described in section 4.2) and a picture ontology. The geographical ontology is too helpful to be used in our experiment because most of the students only use computer when they are in laboratory or in their dorms, so we can easily detect the frauded accounts, but in other applications, the geographical data of a user may change frequently.

With a 30 days test, we have colleted these data, user use BBS 3 valid hours per day (many students stay in the BBS much longer than this, but the time period in which there are too few activities are omitted), and we created about 28800 ontology instances, including temporal instances, after selecting and merging, 8200 instances were stored in the database, (about 200 instances for each account, but not or the instances are used when computing similarity). Among them, 723 instances were created from the random chosen fakers, 674 instances have correctly indicated the fraud, and the accuracy is 93.2%. From the figure 3 of following, we can see, at the time of 2 minutes after the user' login, the accuracy for fraud detection is 70.60 (this

Fig. 3. Fraud detection accuracy

is for experiment, and not counted in former statistics), at the time of 5 minutes, the accuracy is 87.7%, at this time, an alarm may be delivered if fraud detected, after 30 minutes, the accuracy are mostly the same.

Compared to other experiments [2, 3], the accuracy has not be improved, we will explore more large scale test using better classifiers, and this will be one of our future work. Our contribution here is using the ontology to present the data, using the ontology instance similarity as the indicator of fraud detection for early stage of applications, when there are not enough examples to train the classifies. Ontology is now widely used and more and more domain will use ontology for the for the purpose of enabling knowledge sharing and reuse. Our system can easily be adapted to different applications. As far as we know, other methods for fraud detection should model the data for each applications, this will take a lots of work.

Our future work may take 2 directions, one is to choose or design better classifiers for ontology instances. And the other is to apply ontology extraction method to our system, because many ontologies are designed for other purpose and are not suit fraud detection very well.

References

1. <http://www.internetworldstats.com/stats.htm>
2. Clifton Phua, Daminda Alahakoon, and Vincent Lee: Minority Report in Fraud Detection: Classification of Skewed Data. ACM SIGKDD Explorations Newsletter. Volume 6, issue 1 (2004). 50-59.
3. Tom Fawcett, and Foster Provost: Adaptive fraud detection. Data Mining and Knowledge Discovery, (1997), 291-316.
4. Wenke Lee, Salvatore J. Stolfo, Kui W. Mok: Mining in a Data-flow Environment: Experience in Network Intrusion Detection. Data Mining and Knowledge Discovery, (1999), 114-124.
5. Tom Fawcett, and Foster Provost: Activity Monitoring: Noticing interesting changes in behavior. Data Mining and Knowledge Discovery, (1999), 53-62.

6. Balaji Padmanabhan and Alexander Tuzhilin: On Characterization and Discovery of Minimal Unexpected Patterns in Rule Discovery. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 2, February 2006
7. John Kingston, Burkhard Schafer, and Wim Vandenberghe: No Model Behavior: Ontologies for Fraud Detection. *Law and the Semantic Web*, LNCS 3369, (2005), 233-247.
8. Shuk Ying HO: An Exploratory Study of Using a User Remote Tracker to Examine Web users' Personality Traits. *ICEC'05*, 659-665.
9. <http://www.semantic-squirrel.org/>
10. Mischa M Tuffield, Antonis Loizou, David Dupplaw: The Semantic Logger: Supporting Service Building from Personal Context. *CARPE'06*, 55-63.