DISTRIBUTED DATAMINING

IN

CREDIT CARD FRAUD DETECTION

ABSTRACT

Credit card transactions continue to grow in number, taking a larger share of the US payment system, and have led to a higher rate of stolen account numbers and subsequent losses by banks. Hence, improved fraud detection has become essential to maintain the viability of the US payment system. Banks have been fielding early fraud warning systems for some years. We seek to improve upon the state-of-the-art in commercial practice via large scale data mining. Scalable techniques to analyze massive amounts of transaction data to compute efficient fraud detectors in a timely manner are an important problem, especially for e-commerce. Besides scalability and efficiency, the fraud detection task exhibits technical problems that include skewed distributions of training data and non-uniform cost per error, both of which have not been widely studied in the knowledge discovery and data mining community. Our proposed methods of combining multiple learned fraud detectors under a "cost model" are general and demonstrably useful; our empirical results demonstrate that we can significantly reduce loss due to fraud through distributed data mining of fraud models.

INTRODUCTION

Credit card transactions continue to grow in number, taking an ever larger share in the US payment system and leading to the higher rate of stolen account numbers and subsequent losses by banks. Improved fraud detection thus has become essential to maintain the viability of the US-payment system. Banks has used early fraud warning systems for some years.

Large-scale data-mining techniques can improve the state of art in commercial practice. Scalable techniques to analyze a massive amounts of transaction data efficiently compute fraud-detection tasks in a timely manner is an important problem, especially for e-commerce. Beside scalability

and efficiency, the fraud-detection tasks exhibits technical problems that include screwed distributions of training data and non uniform cost per error, both of which had not been widely studied in the knowledge-discovery and data mining community.

Our proposed methods of combining multiple learned fraud detectors under a "cost model" are general and demonstrate that we can significantly reduce loss due to fraud through distributed data-mining of fraud models.

Our Approach

In today's increasingly electronic society and with rapid advances of electronic commerce on the internet, the use of credit cards for purchases has became convenient and necessary. Credit card transactions has became the de facto standard for internet and web-based e-commerce. The growing number of credit card transactions provides more opportunity for thief's to steal credit card numbers and subsequently commit fraud. When banks loose money of credit card fraud, card holders pay for all that loss through higher interest rates, higher fees, and reduced benefits. Hence it is in both the banks and the card holders' interest to reduce illegitimate use of credit cards by early fraud detection. For many years the credit card industry has studied computing models for automated detection systems; recently, these models have been the subject of academic research, especially with respect to e-commerce.

The credit card fraud-detection domain presents a number of challenging issues for data mining.

- There are millions of credit card transactions processed each day. Mining such massive amounts of data requires highly efficient technique that scale.
- The data are highly screwed-many more transactions are legitimate than fraudulent. Typical accuracy based mining techniques can generate high accurate fraud detectors by simply predicting that all transactions are legitimate, although this is equivalent to not detecting fraud at all.
- Each transaction record has a different amount and thus has variable potential loss, rather than fixed misclassification cost per error-type, as is commonly-assumed in a cost based mining technique.

Our approach addresses the efficiency and scalability issues in several ways. We divide a large data set of labeled transactions into smaller subsets, apply mining techniques to generate classifiers in parallel, and combine the resultant base models by meta learning from the classifiers behavior to generate meta classifiers. Our approach treats the classifier as black boxes so that we can employ a variety of learning algorithms. Beside extensibility, combining multiple models computed over all available data produces meta classifiers that can offset the loss of predictive performance that usually occurs when mining from data subsets or sampling. Furthermore, when we use the learned classifiers, the base classifiers can execute in parallel, with the meta classifier then combining their results. So our approach is highly efficient in generating these models and also relatively efficient in applying them.

Another parallel approach focuses on parallelizing a particular algorithm on a particular parallel architecture. However a new algorithm or architecture requires a substantial amount of parallel-programming work. Although our architecture and algorithm independent approach is not efficient as some fine grained parallelization approaches, it lets users plug different off-the-shelf learning programs into a parallel and distributed environment with relative ease and eliminates the need for expensive parallel hardware.

Furthermore our approach can generate a potentially large no of classifiers from the currently processed data subsets, and therefore potentially require more computational resources during detection, we investigate pruning methods that identify redundant classifiers and remove them from the ensemble with out significantly degrading the predictive performance. This pruning technique increases learned detectors' computational performance and throughput.

The issue of screwed distributions has not been studied widely because many of the datasets used in this research do not exhibit this characteristic. We address skewness by partitioning the data into subsets with a desired distribution. Applying mining techniques to the subsets, and combining the mined classifiers by meta learning. Other researchers attempt to remove instances from the majority classes-instances that are in the border line region are candidates for removal. In contrast our approach keeps all the data for mining and doesn't change the underlying mining algorithms.

We address the issue of non uniform cost by developing the appropriate cost model for the credit card fraud domain and biasing our methods toward reducing cost. This cost model determines the desired distribution just mentioned. Ada Cost relies on the cost model for updating weights in the training distribution. Naturally, this cost model also defines the primary evaluation criterion for our techniques. Furthermore we investigate techniques to improve the cost performance of bank's fraud detector by importing remote classifiers from other banks and combining this remotely learned knowledge with locally stored classifiers. The law and competitive concerns restrict banks from sharing information about their customers with other banks. However they may share black box fraud detection models. Our distributed data mining approach provides a direct and efficient solution to sharing knowledge with out sharing data. We also address possible incompatibility of data schemata among different banks.

We designed and developed an agent based distribution environment to demonstrate our distributed and parallel data mining technique.

Credit Card Data and Cost Models:

The records of the transactions consist of about 30 attributes including binary class label (fraudulent/ legitimate transaction). Some fields are numeric and categorical. Because account identification is not present in data, we cannot group transactions into accounts. Therefore, instead of learning behavioral models of individual customer accounts, we build overall models that try to differentiate legitimate transactions from fraudulent ones. Our models are customer independent and can serve as a second line of defense, the first being customer dependent models.

Most machine learning literature concentrates on model accuracy This domain provides considerably different metric to evaluate the learned model's performance models are evaluated and rated by a cost model. Due to different amounts of each credit card transaction and other factors, the cost failing to detect the fraud varies with each transaction. Hence the cost model for this domain relies on sum and average of loss caused by fraud. We define

$$CumulativeCost = \sum_{i}^{n} Cost(i)$$

and

$$AverageCost = \frac{CumulativeCost}{n}$$

Where *Cost(i)* is the cost associated with transaction i, and n is the total no of transactions.

Each investigation incurs an overhead, other related costs-for example, the operational resources needed for the fraud detection system-are consolidated into overhead. So if the amount of transaction is smaller than the overhead investigating the transaction is not worth while, even it is suspicious.

Cost model assuming fixed overhead

Outcome	Cost
Miss (false negative – FN)	Tranamt
False Alarm(false positive –FP)	Overhead if tranamt> overhead or 0 if tranamt ≤ overhead
Hit (true positive – TP)	Overhead if tranamt>overhead or tranamt if tranamt ≤ overhead
Normal (true negative – TN)	0

Here for each Transaction, where tranamt is the amount of credit card transaction. The overhead threshold for obvious reasons is closely guarded secret and varies over time the range of values used here are probably reasonable levels as bound for this data set, but are probably significantly lower. We evaluated all our empirical studies using this cost model.

Skewed Distributions

Given skewed distribution, we would like to generate training set of labeled transactions with a desired distribution with out removing any data, which maximizes classifier performance. In this

domain we found that determining the desired distributions is an experimental art and requires exclusive empirical tests to find the most effective training distribution.

In our approach we first create data subsets with the desired distribution (determined by extensive sampling experiments). Then we generate classifiers from these subsets and combine them by metalearning from their classification behavior. For example the skewness distribution is 20:80 and desired distribution for generating the best models is 50:50, we randomly divide the majority instances into four partitions and from four data subsets by merging the minority instances with each of the four partitions from four data subsets to generate desired 50:50 distribution for each distributed training set.

For concreteness. Let N be the size of the data set with a distribution of xy (x is the percentage of minority class) and u:v be the desired distribution. The no of minority instances is N*x, and desired number of majority instances in a subset is Nx * v/u. The number of subsets is the number of majority instances (N * y) divided by the no of desired majority instances in each subset, which is Ny divided by the Nxv/u or y/x * u/v. so we have y/x * u/v subsets, each of which has Nx minority instances and Nxv/u majority instances.

The next step is to apply a learning algorithm or algorithm to each subset. Because the learning processes on the subsets are independent, the subsets can be distributed to different processors and each learning process can run in parallel. For massive amounts of data, our approach can substantially improve speed for superliner time learning algorithms. The generated classifiers as combined by meta learning from their classification behavior.

Class-Combiner strategy composes a meta level training set by using the base classifiers predictions on a validations set as attribute values and the actual classification as the class label. This training set then serves for training a meta classifier. For integrating subsets the class combiner strategy is more effective than the voting-based techniques. When the learned models are used during online fraud detection, transactions fed into the learned base classifiers and the meta classifier then combines their predictions. Again the base classifiers are independent and can execute in parallel on different processors. In addition our approach can purne redundant

base classifier with out effecting the cost performance, making it relatively efficient the credit card authorization process.

C4.5, CART , Ripper and Bayes. Bayes the matalearner for all the applications, CART is used to generate classifiers. We calculate the R the ratio of the overhead amount to the average cost.

R = Overhear/ Average Cost. Our approach is significantly more effective than the deployed COTS when R<6. Both methods are not effective when R>24. So under a reasonable cost model with a fixed overhead cost in challenging transactions as potentially fraudulent, when the number of fraudulent transactions is very small percentage of the total, is very un desirable to detect fraud. The loss due to fraud is yet another cost of conducting business.

However- filtering out easy or low risk transaction can reduce a high overhead –to-loss ratio. The filtering process can use fraud detectors that are built based on the individual customer profiles, which are now use by many credit card companies. These individual profiles characterize the customers' purchasing behavior. For example, if customer regularly buys groceries from a particular super market or has setup a monthly payment for telephone bills, these transactions are close to no risk; hence purchases of similar characteristics can be safely authorized with out further checking. Reducing the overhead through streamlining business operations and increased operation will also lower the ratio.

Knowledge Sharing through Bridging

Much of the prior work on the combining multiple models assumes that all models originate from different subsets of single data set as means to increase accuracy and not as a means to integrate distributed information. Although the JAM system addresses the later problem by employing meta learning techniques, integrating classification models derived from distinct distributed databases might not always be feasible.

In all cases considered so far, all classification models are assumed to originate from databases from identical schemata because classifiers depends directly on the underlying data's format, minor differences in the schemata between databases derive in compatible classifiers – That is a

classifier that cannot be applied on data of different formats. Yet these classifiers may target the same concept. We seek the bridge these disparate classifiers in same principled fashion.

The banks seek to be able to exchange their classifiers and thus incorporate useful information in their system that would otherwise be inaccessible to both. Indeed for each credit card transaction, both institutions record similar information, however they also include specific fields containing important information that each has acquired separately and that provides predictive value in determining fraudulent transaction patterns. To facilitate the exchange of knowledge and take advantage of in compactable and other wise useless classifiers, we need to devise methods that bridge the differences imposed by the different schemata.

Pruning

An ensemble of classifiers can be unnecessarily complex, meaning that many classifiers might be redundant, wasting resources and reducing system throughput. We study the efficiency of metaclassifiers by investigating the effect of pruning on their performance. Determining optimal set of classifiers for metalearning is a combinatorial problem. Hence the objective of pruning is to utilized the heuristic methods to search for partially grown meta classifiers that are more efficient and scalable and at the same time archieve comparable or better predictive performance results than fully grown metaclassifiers. There are two stages for pruning meta classifiers; The pre training and post training pruning stages.

Pre training pruning refers to the filtering of classifiers before they are combined. Instead of combining classifiers in brute force manner, we introduce preliminary stage for analyzing the available classifiers and qualifying them from inclusion in a combined meta classifier. Only those classifiers that appear to be most promising participate in the final metaclassifier.

Post training pruning denotes the evaluation and pruning of constituent base classifiers after a complete metaclassifiers has been constructed.