A modeling approach for credit card fraud detection in electronic payment services

Gabriel Preti Santiago
Universo Online Inc.
Dept. of R & D
São Paulo, SP, Brazil
gabrielpreti@gmail.com

Adriano C. M. Pereira
Federal Univ. of Minas Gerais
Dept. of Computer Science
Belo Horizonte, MG, Brazil
adrianoc@dcc.ufmg.br

Roberto Hirata Jr.
Univ. of São Paulo - IME
Dept. of Computer Science
São Paulo, SP, Brazil
hirata@ime.usp.br

ABSTRACT

There is an increase in the volume of electronic transactions in the Web in recent years, mainly due to the significant growth that has been observed in e-commerce. This scenario makes the frauds in electronic transactions a matter of high importance. This work proposes a comprehensive approach to address the problem of fraud in the emerging market of on-line payment services. We present a model for this problem, based on the history of the main entities involved in a transaction and we retrieve features to classify whether the transaction is a fraud or not. In order to validate our results, we use real data provided by a large Latin American on-line payment service company.

Categories and Subject Descriptors

K.4.4 [**Computers and Society**]: Electronic Commerce—*e-markets*

General Terms

Experimentation, Management, Security

Keywords

trust management, fraud detection, e-commerce

1. INTRODUCTION

The Internet has been facing an increase in the volume of electronic transactions due to e-commerce in the Web. As a consequence, there is an increasing usage of credit cards as a payment method for on-line transactions and, consequently, an increase of individuals with the only intention of taking advantage to obtain illicit financial benefits.

As a result of this fact, there is a high incidence of frauds with billions of dollars in losses. According to [7], in 2009, with the quick expansion of e-commerce, half of all credit card frauds occurred in the the Web. According to [6], losses due to fraud in the e-commerce averaged 0.9% of the total revenue of the North American companies in 2012, leading to a total loss of $3.5 billion and an increase of $100 million compared to 2011.

Facing this scenario, emerged in the Web the so called on-line payment services companies, which provides on-line payment solutions with the purpose of making e-commerce transactions safer. These companies need effective strategies for the prevention and detection of fraudulent transactions on the Web. An effective strategy has to be based on the application of state of the art theories and computational techniques that help in an efficient combat against frauds and at an economically viable cost.

Fraud detection is a complex problem. Fraudulent transactions are hidden among a huge amount of legitimate transactions and one of the biggest challenges is its heterogeneity. The profile of fraudsters and legitimate users are quite diversified and that makes the problem of finding patterns even harder. Besides that, user profiles changes over time, in particular the fraudsters profile: they try to learn how the fraud detection systems work and keep evolving its behavior to tease these systems [7, 3, 8, 1, 4]. Therefore it is difficult to have a single system or a single approach able to detect every kind of fraud.

In this work we propose a comprehensive approach to automatically generate fraud alarms in on-line payment systems. We present a model based on the history of the main entities involved in an on-line transaction and we retrieve features to classify whether the transaction is a fraud or not. In order to validate our approach, we adopt a huge dataset with millions of real transactions provided by a large Latin American on-line payment services company.

2. MODELING FRAUD

Given the complexity involved in the problem, it is really important to have an efficient modeling of the problem before the application of computational techniques.

Table 1 presents the features in each transaction of our dataset. We were careful to use only features that are general enough to be applied to any company in the same kind of business. By doing that, we keep the generality of the work and avoid problems with the disclosure of confidential information.

A transaction in an on-line payment service is an interaction between four entities: the buyer, the seller, the credit card and the credit card holder. Note that the product, or the service involved in the transaction, is not listed here because its complete information is not available to the on-line payment service.

SAC'15 April 13-17, 2015, Salamanca, Spain.

http://dx.doi.org/10.1145/2695664.2695990

Transaction hour of the day	Buyer code
Buyer IP	Buyer geolocation code
Buyer phone number	Buyer residential zip code
Buyer delivery zip code	Seller code
Seller category	Credit card identifier
Credit card holder code	Holder born date
Holder age	Credit card acquirer
Transaction value range	Payment method
Installments quantity	Transaction products quantity
Transaction distinct products quantity	

Table 1: Features in the dataset

A given entity takes part in multiple transactions over time. Assuming that, in a given moment, each entity has a history of transactions previously made. As the entities of a transaction are independent among them, each one of their history will be a set of transactions that are independent on the other entities in the same transaction. We may have shared transactions in the history of the entities of a given transaction, but the history itself is independent.

Given that each entity takes part in many transactions over time, a reasonable assumption is that each entity tends to keep an almost constant behavior pattern in its legitimate transactions. Therefore a way to estimate the risk of a transaction is to calculate how discrepant the transaction is from the history of legitimate transactions of every one of its entities. Besides, as each entity is independent from the others, this estimation can be made in a independent way for each one of the entities.

Whenever we talk about an entity history, we are talking about the history of a particular entity in a particular transaction, comprising only transactions that occurred before the transaction being analyzed. So, an important definition is the size of the historical window (in this work we fixed the size of the window in ninety days).

Let us define $H_et = \{t_1, t_2...t_n\}$ as the set of transactions in the history of an entity $e \in E_t$, being
$E_t = \{buyer_t, seller_t, card_t, holder_t\}$ the set of the four entities involved in a transaction t. Let us also define $X = \{x_1, ..., x_{19}\}$ as the set of the nineteen features of a transaction, and v_i the value of a feature $x_i \in X$ in a transaction $t = [v_1, ..., v_{19}]$. To calculate the score of a feature, we compare it with information of the transactions in the history of the entities of t, resulting in some quantitative measure. Our approach to this comparison consists in counting how many times the value v_i has already been used previously in the set H_et for each one of the entities $e \in E_t$, obtaining a specific score for each feature in each entity of transaction t. The function that calculates the score of a feature x_i for an entity $e \in E_t$ in a transaction t is defined as:

$$SCa(x_i, e, t) = count(v_i, H_et) \quad (1)$$

where $count(v_i, H_et)$ is the number of times in which the value v_i appears for the feature x_i in H_et.

At this point, it is important to note that, in Equation 1, we consider just the effective and legitimate transactions from H_et. An effective transaction is the order accepted or approved by the fraud analysis sector of the company. Moreover, it is important to note that a past transaction may have not been approved by the fraud analysis procedure of the company, but we have no way to tell if that transaction is really fraudulent or not. The general idea is to calculate how similar a transaction is in relation to past legitimate transactions of its entities, and so we consider only transactions that are proven legitimate from H_et in Equation 1.

For a transaction being analyzed, we calculate Equation 1 for every feature x_i once for each entity $e \in E_t$, obtaining a score $SCa(x_i, e, t)$ for each feature in each one of the entities. Something we noted when analyzing our data is that the entity Seller has a very distinct behavior when compared to the other three entities: sellers in general have larger transaction history and they tend to have more heterogeneous set of transactions in their history. This characteristic makes it more difficult to model the behavior of sellers in the same way we do for the other three entities. We performed experiments using entity Seller, however the obtained results were much worse than the ones we achieved excluding this entity. Therefore, we are not going to use the scores of the seller features.

Another important concept that is considered in the classification procedure is the weight of an entity in a transaction. We call $p(e)$ of an entity $e \in E_t$ for a transaction t the size of the history of e, excluding the transactions identified as fraud. So, $p(e)$ stands for the number of effective and legitimate transactions plus the number of non-effective transactions in the history of e.

Finally we have the concept of the general weight of the transaction, which is the biggest weight among the entities considered in the classification task.

3. CLASSIFICATION

After computing the score and the weight for each transaction, we have a vector of features that, in conjunction with its classification (fraud or not), can be used to train and test a classifier. However, before being used in the classifier, the values of all the features are normalized to have mean zero and variance one.

Classification is performed using support vector machines with a radial kernel - *SVM* [9, 5] (other methods have been tested in our dataset, but *SVM* was the most robust with higher dimension data [2]). *SVMs* are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. A *SVM* model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In addition to performing linear classification, *SVM* can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

4. EXPERIMENTS

The entities history size and, consequently, the transaction general weight, have a significant impact in our model. Thus, it is important to consider the evolution of the transaction general weight in the results of our experiments. To do that, we will generate subsets of transactions with certain general weights, train the classifier and execute the classification task independently in each subset. This approach is important to assess the impact of the general weight in the results and, also, to mitigate *SVM* complexity issues. The subsets defined are:

1. *Subset 1 to 4*: transactions with general weight from

1 to 4, respectively;

2. *Subset 5*: transactions with general weight greater or equal to 5 and less than 10;

3. *Subset 6*: transactions with general weight greater or equal to 10 and less than 15;

4. *Subset 7*: transactions with general weight greater or equal to 15 and less than 20;

5. *Subset 8*: transactions with general weight greater or equal to 20 and less than 30; and

6. *Subset 9*: transactions with general weight greater or equal to 30;

As our model is essentially based in the history of the entities involved, it is important to define the limits in a way that we have only transaction with enough information, which are the transactions with general weight greater than zero, meaning that transactions with general weight of zero will not be considered in our experiments.

We executed isolated experiments for each month in the period we are evaluating (from January 2011 to May 2012), i.e, for each month we have a different training and testing sample. For each month, our training sample consists of the credit card transactions from the third month before that, and the testing sample consists of all the transactions from the respective month. Due to this fact, the first month we will analyze is April 2011. Another important point is that when preparing the training samples, we selected only credit card transactions that were effective for the class of legitimate transactions.

For the classification task, we build a separate model for each one of the subsets defined, so we train and execute the classification task in an isolated way in each subset. After executing the classification for every subset, we consolidate the results summing up the numbers of hits and misses in each subset, resulting in a single measure for the whole classification procedure in all subsets.

Something important to be considered in the classification task is the high unbalance level between the two classes, which brings us serious problems in the generation of the model from the training sample. To deal with this, we do an artificial softening in the unbalance degree in the training sample: we select every fraudulent transaction available for our training sample and we randomly select from the legitimate transactions twice the number of fraudulent transactions. As the legitimate transactions are chosen randomly, we are introducing an aleatory component in our experiments, and so we executed every experiment ten times. The results that we show are the arithmetic mean of all executions.

Due to the unbalance level between classes, traditional measures used in classification tasks (e.g., accuracy and f-measure) are not the most suited to our problem, thus we are going to present separate hit rates for each class. Tables 2, 3 and 4 show the frauds detection rate for each subset, and Figure 1 shows the results of each subset in a incremental way (results from Subset 1 represents the amount of fraud detected in the Subset 1 in relation to the total of fraud from all subsets; the results from Subset 2 represents the amount of fraud detected in Subset 1 and Subset 2 in relation to the total of fraud from all subsets, and so on).

Subset	04/11	05/11	06/11	07/11	08/11
1	20.72/1.42	21.26/1.69	23.8/1.16	24.43/5.8	22.37/1.27
2	25.59/3.33	31.19/2.87	44.92/4.94	28.85/3.56	40.45/5.05
3	38.24/7.79	44.4/5.14	49.71/3.53	41.14/3.82	48.88/2.43
4	37.49/3.11	45.83/2.61	55.89/1.77	47.75/4.4	49.38/2.47
5	42.81/4.2	53.95/3	59.94/1.36	54.83/1.31	57.72/1.57
6	46.27/4.59	64.91/3.04	55.42/1.98	61.54/1.73	51.84/2.37
7	48.88/3.97	50.98/5.43	58.82/2.91	64.28/1.64	65.25/2.62
8	31.93/4.69	55.38/2.85	62.95/2.51	71.54/2.24	73.49/1.24
9	21.96/2.34	27.18/1.19	24.04/5.6	9.39/6.47	63.57/3.06

Table 2: Frauds detection rate for each subset with the respective standard deviation.

Subset	09/11	10/11	11/11	12/11	01/12
1	24.02/0.78	28.28/8.58	27.63/3.76	21.19/1.2	25.27/1.03
2	41.2/3.1	37.43/5.02	44.92/8.41	37.39/5.56	31.02/5.48
3	52.01/4.91	51.99/4.77	46.25/3.08	48.88/3.02	41.58/2.64
4	52.95/4.29	55.96/4.14	48.11/4.23	48.47/3.47	51.48/3.72
5	59.91/1.75	60.86/1.52	62.11/1.14	57.21/1.75	55.63/1.71
6	58.93/2.25	70.1/3.01	66.51/3.17	59.04/1.9	59.01/1.86
7	64.34/2.58	60.77/4.33	61.47/2.54	63.78/2.89	73.51/4.14
8	65.18/3.12	73.25/2.99	73.87/2.28	56.42/2.51	63.52/3.45
9	42.44/1.59	49.81/2.84	53.1/2.55	54.85/2.56	48.93/1.76

Table 3: Frauds detection rate for each subset with the respective standard deviation.

Subset	02/12	03/12	04/12	05/12
1	24.5/0.78	22.01/0.95	19.15/2.98	23.14/3.03
2	43.63/2.61	44.9/1.69	46.24/3	44.49/2.09
3	39.4/3.56	60/2.28	40.34/4.96	51.69/2.59
4	44.1/3.03	49.09/6.66	48.46/3.31	40.53/8.86
5	57.84/2.16	50.22/0.98	60/1.6	56.4/2.09
6	55.87/1.68	53.34/2.58	62.26/1.99	54.33/1.95
7	56.81/2.6	60.15/2.42	61.87/2.92	43.27/1.54
8	64.26/1.32	41.18/2.38	54.32/4.98	50.23/3.18
9	54.64/2.92	39.14/3.2	57.51/4.63	50.66/4.55

Table 4: Frauds detection rate for each subset with the respective standard deviation.

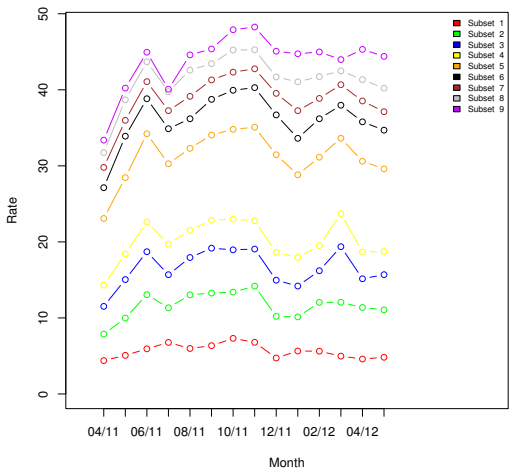

Figure 1: Incremental fraud detection rate for each subset, by month.

Let us look now to the false alarm rate obtained with the classification procedure. We call false alarms, legitimate transactions identified as fraud by the classifier. Tables 5, 6 and 7 show the results for each subset and Figure 2 shows the results in an incremental way.

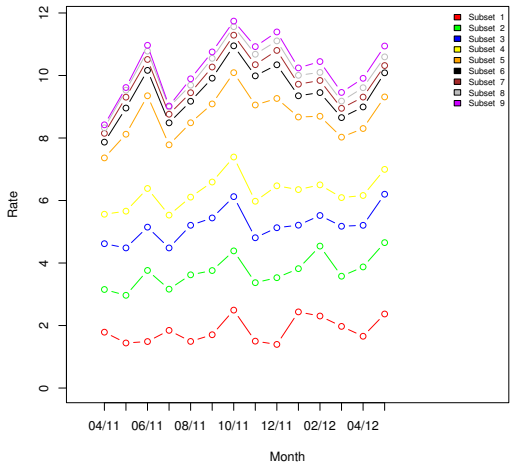

Figure 2: Incremental false alarm rate for each subset, by month.

Subset	04/11	05/11	06/11	07/11	08/11
1	7.2/0.92	5.33/0.98	5.51/0.6	6.6/2.5	5.28/0.52
2	8.44/1.6	8.98/1.5	13.35/2.78	7.54/2.12	12.02/2.5
3	12.84/3.28	13.17/2.34	11.93/1.55	11.14/1.57	13.41/0.78
4	11.28/1.55	14.2/1.4	15.02/0.99	12.54/2.25	10.84/1.14
5	8.5/1.02	12.3/0.74	14.73/0.41	11.36/0.41	12.15/0.82
6	6.22/1.12	11.58/0.76	11.22/0.81	10.37/0.47	10.25/0.93
7	7.37/1.34	10.11/1.25	10.28/0.74	9.01/0.76	9.16/1.52
8	5.36/0.97	7.64/1.01	9.49/0.72	9.12/0.67	9.62/0.67
9	3.42/0.26	3.33/0.3	7.28/0.83	1.53/1.47	9.78/0.59

Table 5: False alarm rate for each subset with the respective standard deviation.

Subset	09/11	10/11	11/11	12/11	01/12
1	6.07/0.43	9.1/5.12	5.29/0.98	4.74/0.45	7.68/0.56
2	11.59/1.8	10.77/3.05	10.64/3.72	11.93/3.71	7.44/1.9
3	14.19/2.07	14.57/2.42	12.27/1.38	13.8/1	11.86/1.27
4	13.66/1.39	14.97/2.08	14.01/1.61	16.58/1.97	14.28/1.6
5	12.68/0.5	13.49/0.48	15.68/0.8	14.81/1.11	13.1/0.55
6	12.23/1.07	12.49/1.32	13.89/0.96	16.38/0.91	11.84/1.05
7	12.03/1.32	11.06/1.44	11.9/1.25	15.84/1.08	14.84/1.71
8	11.68/0.72	10.89/1.29	13.45/0.65	12.32/1.26	13.67/1.6
9	9.69/0.81	8.2/0.81	11.3/1.26	13.06/1.27	12.18/0.69

Table 6: False alarm rate for each subset with the respective standard deviation.

Subset	02/12	03/12	04/12	05/12
1	7.11/0.3	6.08/0.48	5.3/0.98	7.38/1.35
2	12.12/2.31	8.63/1.04	12.23/0.84	12.32/0.7
3	8.41/1.38	13.61/1.11	11.59/1.52	13.38/1.49
4	12.5/1.06	11.68/1.01	12.29/0.84	10.24/2.27
5	12.8/0.43	11.29/0.52	12.46/0.71	13.57/0.82
6	13.45/0.49	11.29/0.98	11.89/0.46	14.16/1.2
7	15.48/1.04	12.29/1.23	11.62/0.76	9.25/0.88
8	12.34/0.83	11.25/0.98	11.67/1.01	12.46/1.23
9	15.22/0.92	12.48/1.04	9.89/0.9	12.56/1.05

Table 7: False alarm rate for each subset with the respective standard deviation.

5. CONCLUSION

In face of the growth of e-commerce that we have seen in the recent years, and the consequent rise of the on-line intermediation services market, associated to the financial losses caused by frauds, this work presents a modeling and classification approach to be applied to the fraud detection problem. We present a modeling approach based on the four main entities involved in a transaction: the Buyer, the Seller, the Card and the Holder. This model was used in a historical analysis based on the legitimate transactions of each entity. We also saw that the Seller entity has a very peculiar behavior when compared to the other three, and so it was not considered in our experiments. In the experimental phase, we used a *SVM* classifier to classify the transactions as fraud or legitimate. Our results showed that we identified between forty and fifty percent of the frauds in most months, while keeping a false alarm rate between ten and twelve percent.

It is important to say here that the fraudulent transactions that we had in the data used in the experiments were the frauds that could not be detected by the already existent fraud detection procedure in the company which collaborated with our work. So, we can say that this set of frauds is the most difficult to be detected.

An interesting source of future research is the peculiar behavior of sellers: we think it could be promising to model it in a way that we could also consider this entity in the experiments.

Acknowledgment

This research was supported by the Brazilian National Institute of Science and Technology for the Web (CNPq grant numbers 573871/2008-6 and 477709/2012-5), CAPES, CNPq, Finep, Fapemig and Universo On Line Inc. (UOL).

6. REFERENCES

[1] R. J. Bolton and D. J. Hand. Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control VII*, 2001.

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2:1–43, 1998.

[3] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, pages 67–74, 1999.

[4] P. K. Chan and S. J. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 164–168, 1998.

[5] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[6] CyberSource. Online fraud report. Technical report, 2013.

[7] L. Delamaire, H. Abdou, and J. Pointon. Credit card fraud and detection techniques: a review. *Banks and Bank Systems*, 4(2):57–68, 2009.

[8] C. Phua, V. Lee, K. Smith, and R. Gayler. A comprehensive survey of data mining-based fraud detection research. In *International Conference on Intelligent Computation Technology and Automation (ICICTA)*, volume 1, pages 50–53, 2010.

[9] V. Vapnik. The nature of statistical learning theory. *Springer*, 1995.