E-commerce Transaction Anomaly Classification

Minyong Lee
Statistics Department
Stanford University
minyong@stanford.edu

Seunghee Ham
Statistics Department
Stanford University
sham12@stanford.edu

Qiyi Jiang
Statistics Department
Stanford University
qjiang@stanford.edu

I. Introduction

Due to the increasing popularity of e-commerce in our daily lives, credit card usages have dramatically increased over the years. As credit card being the primary method of payment in online transactions, credit card frauds have also been observed to surge as the number of online transactions have increased. Because of the significant financial losses that credit card fraudulent incidents can cause, credit card fraud detection drew our interest into its investigation. Specifically, we examined anomaly detection classification algorithms that can be applied to detect fraudulent credit card transactions. In addition, we proposed a new type of sampling method called Oversampling via Randomly Imputed Features (ORIF) and details of ORIF is given in section IV. Results of incorporating ORIF to each basic classification algorithms are also provided in section V.

II. Data and Preprocessing

A. Data

The data is from the 2009 UC San Diego Data Mining Contest and the goal of the data is to detect anomalous online credit card transactions with 1 being anomalous and 0 being normal. The dataset consists of 19 features with a few continuous variables and some categorical variables such as a state and a web address that correspond to each transaction. The most striking characteristic of the data is that it is highly imbalanced; about only 2 % of the transactions are anomalous, which represents the real world scenario. The problem is that standard classifiers tend to bias in favor of the larger class since, by doing so, it can reach high classification accuracy.

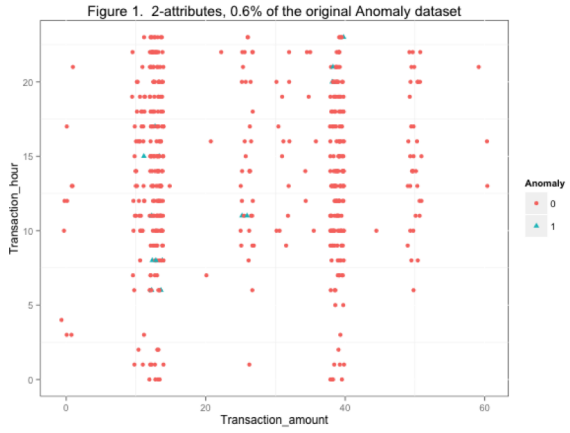


Fig. 1: E-commerce data.

B. Preprocessing

First of all, the data (94,682 observations) was split into training data (80%) and validation data (20%). Due to the issue of highly imbalanced data we are facing, we took a special treatment of sampling 80% of the anomalous (i.e., positive) responses into the training set and the rest 20% was assigned into the validation set. As a result, our training data consist of 80% of total anomalous responses and 80% of total non-anomalous responses. For validation set, it contains the rest 20% of total anomalous and non-anomalous responses. Although we are provided with the original validation dataset with 36,019 observations, we cannot use it to test the performance of our learning algorithms because the dataset does not have the outcome variable. Therefore, we again divided the training data obtained above into training and validation data with the ratio of 80% and 20%. For all the algorithms we tried, we first ran and tested on the second layer of training and validation data to find the best parameter(s) of a specific algorithm, and then applied it to the first layer of validation data and reported the results.

Initially, we did data visualization to detect patterns of each feature and correlations among them, in which we found $amount$ and $total$ are highly correlated and so are $hour1$ and $hour2$. Therefore, we decided to drop $total$ and $hour2$ in later analysis to avoid a multicollinearity problem. In addition, we detected some skew-ness in some of the features that needed variable transformation to normalize them and applied the Box-Cox transformation to address the issue. Meanwhile, outliers have been found for $flag5$ and $field5$. We adopted capping approach for outlier treatment, which is to cap the outliers that exceed 99% quantile of its distribution and imputed 99% quantile value for the outliers.

The features we had the most difficulty with in terms of coming up with creative transformations were $hour1$, $state1$ and $domain1$. For $hour1$, which indicates the time of transaction, showed a wavy trend over 24 hours. So, we proposed the idea of applying a sine and a cosine function on $hour1$ and created two features $coshour$ and $sinhour$ to preserve continuity. We believe the linear combination of these two will capture the wavy and repeating trend of $hour1$. For $state1$, we categorized them into 9 groups according to both the frequency of transactions occurred in each state and geographical segmentations of the United States. We set the top four states with the highest number of transactions as four individual groups ?A, FL, TX, and NY, followed by the West, Midwest, South, and Northeast groups of states. Lastly, we grouped the military addresses AP and AE into the 9th category. For $domain1$, we categorized them into 6

categories based on both the frequency of a domain appearing in the transactions and an industry that the domain belongs to. Shopping websites, news websites, and email sites are the examples of three groups that we assigned domains into.

After data preprocessing, our final training dataset contained 17 features with 12 of them being categorical variables and 5 continuous variables. The same preprocessing steps were applied to the validation dataset.

III. Oversampling via Randomly Imputed Features

In figure 1, it presents our imbalanced data that has unequally proportioned positive and negative classes. In order to deal with the issue that imbalanced data creates, we proposed a new sampling method, called Oversampling via Randomly Imputed Features (ORIF). What ORIF does is to generate artificial instances for minority classes. In details, we imputed the feature values for a new minority class observation by simple random sampling from the existing feature values that corresponding to minority classes. The intuition behind this method was that by looking at our dataset from figure 1, we observed that most of the minority data points are overlapped with each other, which suggests the minority classes are sharing similar feature values. A toy example is displayed in figure 2. We arranged the top four observations as the 2% positive classes in the whole 200 observations dataset. In order to create half more of the existing positive classes, which is 2 artificial positive classes in this example, we imputed each feature values for new1 and new2 by randomly select corresponding feature values from the positive classes.

The main advantage of ORIF is that it does not impose any structure to the data. Compared to Synthetic Minority Oversampling Technique (SMOTE) which adds artificial neighbors to the minority class (Chawla et al., 2002), ORIF does not require the distance metric on the feature space which is hard to define when it is a mixture of numerical and categorical variables. Also, it is very simple and fast to implement on any dataset.

We expect that ORIF will increase the performance of a classification rule especially when the features are independently affecting the response. ORIF breaks the interaction of features by randomly choosing the values from the whole positive observations, so it might not work well if there are interactions of features which are significant. If we have a prior information about the interaction among features, we can apply grouped ORIF. For the anomalies in our data, we believe that the features themselves represent the characteristics of anomalies more than the interactions, and therefore ORIF could be a good oversampling method.

IV. Evaluation for classification and parameter selection

In an imbalanced dataset, classifying all the observations as negatives usually minimize the classification error. But in anomaly detection, we want to penalize false negatives more while not classifying all the observations as positives. Therefore, the F-measure

$$F - measure = \frac{2 \times precision \times recall}{precisionrecall}$$

Example of imbalanced data with 2% positive classes

	feature1	feature2	feature3	Y
obs1	111	2	0.09	1
obs2	121	3	0.14	1
obs3	109	5	0.11	1
obs4	115	4	0.2	1
obs5	120	1	0.15	0
obs6	110	2	0.16	0
...	...	...	...	...
obs199	124	2	0.24	0
obs200	108	3	0.07	0
ORIF	feature1	feature2	feature3	Y
new1	109	3	0.09	1
new2	121	2	0.11	1

Fig. 2: ORIF

can give us a good performance measure by balancing the tradeoff between $precision = \frac{true\ positives}{predicted\ positives}$ and $recall = \frac{true\ positives}{actual\ positives}$. A weighted harmonic mean could also be a good criterion if $recall$ is regarded more important than $precision$.

Most of the basic classification rules minimize misclassification error rate. When there are not many positives, the rules will classify all the observations as negatives to minimize misclassification error rate. Therefore the parameters in each classification rule must be carefully chosen to maximize $F - measure$. Here we use validation set approach to find the best parameters. Without touching the original validation data, we divide our training data into training-training data and validation-training data and train with a specific parameter on the training-training data and evaluate on the validation-training data.

V. Classification Methods

Our goal is to find a good algorithm to classify anomalous transaction in a reliable manner. As a preliminary learning process, we tried various basic classification algorithms involving different parameters and followed by employing ORIF to each algorithm.

A. Logistic regression

We first fit logistic regression, which can output probability of an event appearing. Since our dataset is highly imbalanced, the probabilities of classifying in a positive class produced by the learning algorithm are very small. Thus, instead of looking at the probabilities, it makes more sense to look at the quantiles of probabilities. By experimenting with different quantiles on training set, we found by assigning true positives among top 1.7% to be the positive likely transactions achieved highest f-measure. The f-measure and recall computed for validation set are 26.5% and 23.4% respectively. After applying logistic regression with ORIF, f-measure has increased by 0.3% and 0.2% for recall.

B. Linear Discriminant Analysis

The f-measure and recall results we obtained from LDA are 28.8% and 28.6%. However, applying ORIF to LDA does not change the performance of the f-measure and recall. Although it is a bit debatable, one study found that there is no reliable empirical evidence that imbalanced dataset has a negative effect on the performance of LDA and that the improvement in the performance of LDA from re-balancing is very marginal (Xue and Titterington, 2008).

C. Decision Trees

Since the competition did not provide definition of each feature in the dataset, we practiced decision tree classification method to get a sense of what are the most significant features in predicting anomalies. We found out three features appeared in the tree classification and they are transaction amount, time of the transaction and type of transaction method. In general, decision tree algorithms have moderate to high variances (Dietterich and Kong, 1995), the f-measure and recall we obtained are a lot lower than the previous two methods with 17.5% and 10.0% respectively. However, by using decision tree with ORIF, both measures increased significantly to 25.1% for f-measure and 15.8% for recall.

D. K-nearest neighbors

By experimenting with different parameters of k, we learned that as k increases, both f-measure and recall decrease. So we chose k=1 as an optimal parameter, tested on the original dataset and achieved 36.2% for f-measure and 30.1% for recall. Among all other algorithms, KNN turned out to have the highest f-measure. With ORIF, f-measure went down a little while recall went up slightly, but the change was not significant.

E. Naive Bayes

We applied Naive Bayes on the smaller training set by starting without Laplace smoothing then trying with different smoothing values to achieve the best f-measure result with Laplace smoothing of 1. Then following by testing on the validation dataset, it returned a f-measure of 4.8% and recall of 71.1%. Even though the recall is high in this setting, when we look at precision of 2.5%, it reveals a high false positives have been made, thereby a low f-measure shows a poor performance by Naive Bayes classification. By applying ORIF to Naive Bayes, we only see an improvement of 0.1% for f-measure.

F. Random Forest and Weight Random Forest

First, we tried Random Forest with different number of trees from 100 to 500. With more than 500, the algorithm ran out of memory. Overall performance improved only slightly with increasing running time as the number of trees increased. The best result was achieved with 400 trees; 31.1% and 19.1% for f-measure and recall, respectively. When ORIF was applied, f-measure and recall both improved by about 5%.

To weigh in the issue of imbalanced data, we tried Weighted Random Forest, where a larger weight is assigned to the minority class (anomalies) when it was misclassified (Chen, Liaw and Breiman, 2004). Similar to Random Forest, the improvement was only marginal as the number of trees increases, but achieves its highest f-measure with 400 trees. With the number of trees at 400, we applied different class weights and found that the bigger the difference in the class weight between misclassifying positive and negative classes, the better results we get. Therefore, with 400 trees and weights at 0.7 for positive and 0.1for negative classes on the original dataset, f-measure was 21.0% and recall was 12.0%.

G. Support vector machine

Before we implemented Support Vector Machine with Different Costs (SDC), we wanted to look at how SVM performs as a starting point. Without any cost parameter, the performance was very low and with an increasing number of costs, the performance improved significantly although the running time increased accordingly. Since the original data has almost 80,000 observations, cost higher than 10 took an enormous amount of time. So, we decided to report the result with cost=10, which is 1.9% and 0.9% for f-measure and recall, respectively. This performance was the lowest of all and we do not think it is surprising given the fact that we had a lot of categorical variables and we forced them into numeric variables, which gave them orderings when there should not be any.

H. Support vector machine with different costs

Support vector machine with different costs (SDC) penalizes misclassified positives more. The primal problem can be written as

$$\min.\frac{1}{2}||w||^2 + C^+ \sum_{y^{(i)}=1} \xi_i + C^- \sum_{y^{(j)}=-1} \theta_j$$

subject to $w^T\phi(x^{(i)}) + b \geq 1 - \xi_i$, $w^T\phi(x^{(j)}) + b \leq -1 + \theta_j$, $\xi_i \geq 0,\ \theta_j \geq 0$. The cost parameters $C^+,\ C^-$ and a feature mapping should be determined. In a highly imbalanced setting, we put $C^+ \gg\ C^-$ to penalizes positives more. Figure 3a explains the effect of SDC. From now on, i represents an index of positive labels and j represents an index of negative labels. Introduce the Lagrangian

$$G = \frac{1}{2}w^T w + C^+ \sum_i \xi_i + C^- \sum_j \theta_j - \sum_i \alpha_i [w^T\phi(x^{(i)}) + b - 1 + \xi_i] - \sum_j \beta_j [-w^T\phi(x^{(j)}) - b - 1 + \theta_j] - \sum_i \gamma_i \xi_i - \sum_j \delta_j \theta_j$$

$$\frac{\partial G}{\partial w} = w - \sum_i \alpha_i \phi(x^{(i)}) + \sum_j \beta_j \phi(x^{(j)}) = 0$$
$$\frac{\partial G}{\partial b} = -\sum_i \alpha_i + \sum_j \beta_j = 0$$
$$\frac{\partial G}{\partial \xi_i} = C^+ - \alpha_i - \gamma_i = 0$$
$$\frac{\partial G}{\partial \theta_j} = C^- - \beta_j - \delta_j = 0$$

Substituting and simplifying, we obtain the dual form of the problem

$$\max.\mathcal{L} = \sum_i \alpha_i + \sum_j \beta_j - \frac{1}{2}[\sum_{i,i'} \alpha_i \alpha_{i'} K(x^{(i)}, x^{(i')}) + \sum_{j,j'} \beta_j \beta_{j'} K(x^{(j)}, x^{(j')}) - 2\sum_{i,j} \alpha_i \beta_j K(x^{(i)}, x^{(j)})]$$

subject to

$$0 \leq \alpha_i \leq C^+,\ 0 \leq \beta_j \leq C^-,\ -\sum_i \alpha_i + \sum_j \beta_j = 0$$

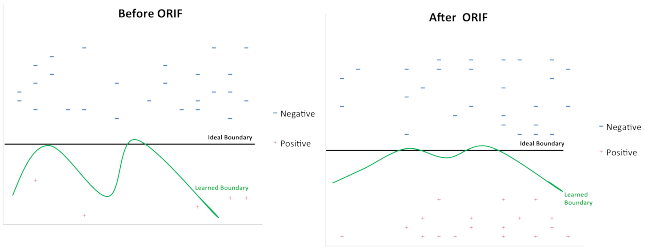
Let the solution of this optimization problem α^*, β^*. Then

$$w^* = \sum_i \alpha_i^* \phi(x^{(i)}) - \sum_j \beta_j^* \phi(x^{(j)})$$

and b^* can be derived from the KKT conditions for an α_l^* such that $0 < \alpha_l^* < C^+$: $\gamma_l^* \neq 0$, thus $\xi_l = 0$ and then $b^* = 1 - w^{*T}\phi(x^{(l)})$. β^* can also be used find b^*.
The classification rule is that we label a new observation as positive ($y = 1$) if and only if

$$w^{*T}\phi(x) + b^* = \sum_i \alpha_i^* K(x^{(i)}, x) - \sum_j \beta_j^* K(x^{(j)}, x) + b^* > 0.$$

Notice that when solving the dual problem and classification, we only need the kernel $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ instead of $\phi(x_i)$s. For our data, we tried linear kernel $K(x_i, x_j) = x_i^T x_j$and gaussian kernel $K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$.



(a) SDC before ORIF (b) SDC-ORIF

Fig. 3: SDC and SDC-ORIF.

Figure 3a and Figure 3b illustrate the effect of ORIF with SDC. The idea is inspired by SMOTE (Chawla et al., 2002), but ORIF can be applied without a distance metric. Non-zero errors on positive support vectors will have larger costs while non-zero errors on negative support vectors will have smaller costs. The net effect of this is that the boundary can be pushed more towards the negative instances. However, as a consequence, SVM becomes more sensitive to the positive instances. Therefore, if the positive instances are sparse, as in imbalanced datasets, the boundary may not have the proper shape in the input space. After using ORIF, the positive instances are now more densely distributed and the learned boundary is much well defined.

Classification Result		
Classification methods	F-measure	Recall
Linear Discriminant Analysis	0.2881	0.2864
Linear Discriminant Analysis-ORIF	0.2885	0.2792
Logistic Regression	0.2648	0.2338
Logistic Regression-ORIF	0.2675	0.2362
Decision Tree	0.1753	0.1002
Decision Tree-ORIF	0.2514	0.1575
K-Nearest Neighbors	0.3620	0.3007
K-Nearest Neighbors-ORIF	0.3403	0.3078
Naive Bayes	0.0486	0.7422
Naive Bayes-ORIF	0.0484	0.7422
Support Vector Machine	0.0185	0.0095
Random Forest	0.3112	0.1909
Random Forest-ORIF	0.2885	0.2792
Weighted Random Forest	0.2096	0.1193
SDC-Linear	0.3062	0.4224
SDC-ORIF-Linear	0.3062	0.4224
SDC-Gaussian	0.3235	0.2888
SDC-ORIF-Gaussian	0.3249	0.2745

TABLE I: Performance Result

VI. Results

We applied 18 classification methods with carefully chosen parameters to maximize F-measure. F-measures varied from 0 to 0.36, and corresponding recall values are in Table I. From Figure 4, we observed that F-measure of all the methods are increased by ORIF except K-nearest neighbors. Figure 5 shows the methods with largest F-measures among all the classification methods, and the recall of SDC-ORIF with linear kernel was relatively larger than others.

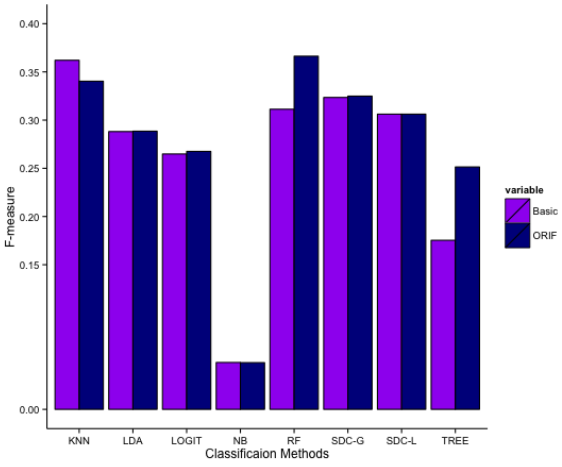


Fig. 4: F-measure comparison between basic classification methods and ORIF-applied classification methods

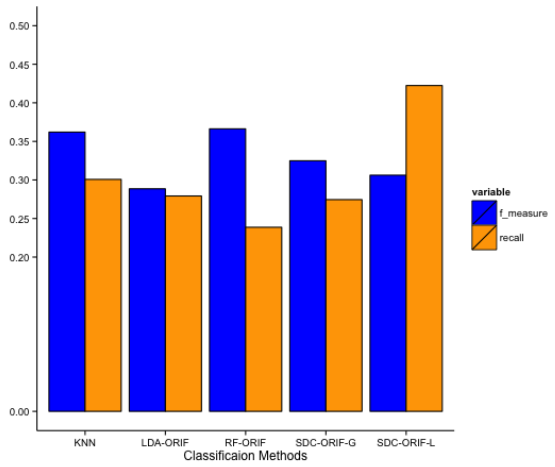


Fig. 5: F-measure and Recall of Classification methods

VII. Conclusion

We successfully applied a new oversampling technique on a highly imbalanced dataset. Our results suggest that ORIF could be effective especially when features are unknown. With ORIF, we were able to improve classification performance by searching appropriate parameters in each classification method. We found that K-nearest neighbors, Random Forest and SDC performs better than others. This means the preprocessing process including scaling were reasonable for the distance measure used in K-nearest neighbors, and the importance of variables in Random forest can be used to analyze significant features in transaction anomalies. In addition, noting that SDC can be applied with various kernels which represent similarity in observations, SDC-ORIF can be improved once we know more about the structure of the data. The results can also be enhanced by ensemble learning.

References

[1] Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. In *Machine Learning: ECML 2004* (pp. 39-50). Springer Berlin Heidelberg.

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence and Research, 16,* 2002, pp. 321-357.

[3] Chen, C., Liaw, A., and Breiman, L. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley.*

[4] Dietterich, T. G. and Kong, E. B. (1995). Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. *Machine Learning,* 255, 0-13.

[5] Imam, T., Ting, K. M., and Kamruzzaman, J. (2006). z-SVM: An SVM for improved classification of imbalanced data. In *AI 2006: Advances in Artificial Intelligence* (pp. 264-273). Springer Berlin Heidelberg.

[6] Phung, S. L., Bouzerdoum, A., and Nguyen, G. H. (2009). Learning pattern classification tasks with imbalanced data sets. Available from: http://www.intechopen.com/books/pattern-recognition/learning-pattern-classification- tasks-with-imbalanced-data-sets

[7] Tang, Y., Zhang, Y. Q., Chawla, N. V., and Krasser, S. (2009). SVMs modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39* (1), 281-288.

[8] Weng, C. G., and Poon, J. (2008, November). A new evaluation measure for imbalanced datasets. In *Proceedings of the 7th Australasian Data Mining Conference-Volume 87* (pp. 27-32). Australian Computer Society, Inc..

[9] Xue, J. H., and Titterington, D. M. (2008). Do unbalanced data have a negative effect on LDA?. *Pattern Recognition, 41* (5), 1558-1571.

[10] Yang, H. and King, I., Ensemble Learning for Imbalanced E-commerce Transaction Anomaly Classification. In *Neural Information Processing* (pp. 886-874). Springer Berlin Heidelberg.