International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
www.ijset.com, Volume No.1, Issue No.3, pg : 194-198 01 July 2012

A Machine Learning Approach for Detection of Fraud based on SVM

Gajendra Singh[1], Ravindra Gupta[1], Ashish Rastogi[1], Mahiraj D. S. Chandel[1], A. Riyaz[2]
[1]Department of Computer Science and Information technology, SSSIST Sehore M.P.
[2]Department of Physics, Aligarh Muslim University, Aligarh

Abstract: ***the growth of e-commerce increases the money transaction via electronic network which is designed for hassle free fast & easy money transaction but the facility involves greater risk of misuse of facility for fraud one of them is credit card fraud it can be happened by many types as by stolen card, by internet hackers who can hack your system & get important information about your card, or by information leakage during the transaction, although many person has proposed their work for credit card fraud detection by characterizing the user spending profile, but in this paper we are proposing the SVM(support vector machine) based method with multiple kernel involvement also including several fields of user profile instead of only spending profile & the simulation result shows improvement in TP(true positive),TN(true negative) rate, it also decreases the FP(false positive) & FN(false negative) rate.***

Keywords: Fraud detection, kernels, SVM (support vector machine).

1. Introduction

Growth in communication network, increased internet speed, easy wireless connectivity & lack of time causes the people to buy through electronic network. Here are some statistics and projections of the Indian credit card industry (http://hubpages.com/hub/Indian-Credit-card-Industry) to show importance of the topic.

1. India is currently the fastest growing Mobile Market in the world and is also among the fastest growing credit card markets in the world.

2. India has a total approx.75 million cards under circulation (25 million credit and 50 million debit) and a 30% year-on-year growth.

3. With 87% of all transactions in plastic money happening through credit cards, debit cards in India continue to be used largely for cash withdrawals.

4. Though Visa, which accounts for 70% of the total card industry is the market leader in India; MasterCard is fast catching up.

5. Every transaction involves payment of an interchange charge to MasterCard or Visa for settlement, which amounted to about $50 million during the year.

6. Internal estimates of Barclaycard have pegged the Indian market with potential to grow to at least 55million credit cards by 2010-11.

The above statistics shows the money involved in transaction through cards & it is required to insure the security of money for both the Bank & for customer.

2. Related Work

As we stated before that many persons has proposed their work on same field some of which we have studied & we think most relevant to our topics are, the work done by Abhinav Srivastava, Amlan Kundu, Shamik Sural, Arun K. Majumdar **[1]** have proposed the probabilistic model based on HMM(Hidden Markov Model) they consider the spending history of card holder & characterize the spending pattern by dividing the transaction amount in three category shows the TP rate of 0.65 & FP rate of 0.05.another paper published by Wen-Fang YU & Na Wang **[2]** who proposed the distance based method This method judge whether it is outlier or not according to the nearest neighbors of data objects. They only showed the highest accuracy about 89.4 percent but not talked about FP & FN.a neural network based approach is presented by Sushmito Ghosh and Douglas L. Reilly **[3]** in their paper they selected large set of 50 field & after proper relation it is reduced to set of 20 features which is used for training neural network. The neural network used in this fraud detection feasibility study is the P-RCE neural network**.** The P-RCE is a member of the family of radial-basis function networks that have been developed for application to pattem recognition. The P-RCE is a three-layer, feed-forward network that is distinguished by its use of only two training passes through the data set. Same work is also done by using regression techniques & compared against neural & decision tree methods **[4]** this work is done by Aihua Shen, Rencheng Tong, Yaochen Deng.their simulation shows that neural networks model provides higher lift(Lift table and lift chart

were used to describe the usefulness of the model to create the scored data set. "Lift" is probably the most commonly used metric to measure the performance of targeting models in classification applications.) than a logistic regression and decision tree on the same data, while neural networks slightly better than logistic regression. This provides a key factor in choosing the models. A similar coefficient sum based model analysis explained by Chun Hua Ju & Na wang **[5]** they analyze type I & type II error rate with highest rate of TP up to 89 percent.

3. SVM (Support Vector Machine)

Support Vector Machines (SVMs) have developed from Statistical Learning Theory **[6]**. They have been widely applied to fields such as character, handwriting digit and text recognition, and more recently to satellite image classification. SVMs, like ANN and other nonparametric classifiers have a reputation for being robust. SVMs function by nonlinearly projecting the training data in the input space to a feature space of higher dimension by use of a kernel function. This results in a linearly separable dataset that can be separated by a linear classifier. This process enables the classification of datasets which are usually nonlinearly separable in the input space. The functions used to project the data from input space to feature space are called kernels (or kernel machines), examples of which include polynomial, Gaussian (more commonly referred to as radial basis functions) and quadratic functions. Each function has unique parameters which have to be determined prior to classification and they are also usually determined through a cross validation process. A deeper mathematical treatise of SVMs can be found in **[7]**.

By their nature SVMs are intrinsically binary classifiers however there exist strategies by which they can be adapted to multiclass tasks. But in our case we not need multiclass classification.

3.1 SVM classification

Let $x_i \in R^m$ be a feature vector or a set of input variables and let $y_i \in \{+1, -1\}$ be a corresponding class label, where m is the dimension of the feature vector. In linearly separable cases a separating hyperplane satisfies **[8]**.

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, \quad i = 1, \ldots, n, \qquad (1)$$

Where the hyperplane is denoted by a vector of weights w and a bias term b. The optimal separating hyperplane, when classes have equal loss-functions, maximizes the margin between the hyperplane and the closest samples of classes. The margin is given by

$$d(\mathbf{w}, b) = \min_{\{\mathbf{x}_i, y_i = 1\}} \frac{|\langle \mathbf{w} \cdot x_i \rangle + b|}{\|\mathbf{w}\|} + \min_{\{\mathbf{x}_j, y_j = -1\}} \frac{|\langle \mathbf{w} \cdot x_j \rangle + b|}{\|\mathbf{w}\|} \qquad (2)$$

$$= \frac{2}{\|\mathbf{w}\|}. \qquad (3)$$

The optimal separating hyperplane can now be solved by maximizing (3) subject to (1). The solution can be found using the method of Lagrange multipliers. The objective is now to minimize the Lagrangian

$$L_p(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) + \sum_{i=1}^{l} \alpha_i, \qquad (4)$$

and requires that the partial derivatives of w and b be zero. In (4), α_i are nonnegative Lagrange multipliers. Partial derivatives propagate to constraints $w = \sum_i \alpha_i y_i x_i$ and $\sum_i \alpha_i y_i = 0$. Substituting w into (4) gives the dual form

$$L_d(\mathbf{w}, b, \alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle, \qquad (5)$$

which is not anymore an explicit function of $\mathbf{w}$ or b. The optimal hyperplane can be found by maximizing (5) subject to $\sum_i \alpha_i y_i = 0$ and all Lagrange multipliers are nonnegative. However, in most real world situations classes are not linearly separable and it is not possible to find a linear hyperplane that would satisfy (1) for all $i = 1. . . n.$ In these cases a classification problem can be made linearly separable by using a nonlinear mapping into the feature space where classes are linearly separable. The condition for perfect classification can now be written as

$$y_i(\langle \mathbf{w} \cdot \Phi(\mathbf{x}_i) \rangle + b) \geq 1, \quad i = 1, \ldots, n, \qquad (6)$$

where Φ is the mapping into the feature space. Note that the feature mapping may change the dimension of the feature vector. The problem now is how to find a suitable mapping Φ to the space where classes are linearly separable. It turns out that it is not required to know the mapping explicitly as can be seen by writing (6) in the dual form

$$y_i \left(\sum_{j=1}^{l} \alpha_j y_j \langle \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i) \rangle \right) + b \geq 1, \quad i = 1, \ldots, n, \qquad (7)$$

and replacing the inner product in (7) with a suitable kernel function $K(xj , xi) = (\Phi(xj) \cdot \Phi(xi))$. This form arises from the same procedure as was done in the linearly separable case that is, writing the Lagrangian of (6), solving partial derivatives, and substituting them back into the Lagrangian. Using a kernel trick, we can remove the explicit calculation of the mapping Φ and need to only solve the Lagrangian (5) in dual form, where the inner product $(xi \cdot xj)$ has been transposed with the kernel function in nonlinearly separable cases. In the solution of the

Lagrangian, all data points with nonzero (and nonnegative) Lagrange multipliers are called support vectors (SV).
Often the hyperplane that separates the training data perfectly would be very complex and would not generalize well to external data since data generally includes some noise and outliers. Therefore, we should allow some violation in (1) and (6). This is done with the nonnegative slack variable ζ_i

$$y_i(\langle \mathbf{w} \cdot \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \zeta_i, \quad i = 1, \ldots, n. \qquad (8)$$

The slack variable is adjusted by the regularization constant ***C***, which determines the tradeoff between complexity and the generalization properties of the classifier. This limits the Lagrange multipliers in the dual objective function (5) to the range $0 \leq \alpha_i \leq C$. ***Any function that is derived from mappings to the feature space satisfies the conditions for the kernel function.***

The choice of a Kernel depends on the problem at hand because it depends on what we are trying to model. A polynomial kernel, for example, allows us to model feature conjunctions up to the order of the polynomial. Radial basis functions allows to pick out circles (or hyper spheres) - in contrast with the Linear kernel, which allows only to pick out lines (or hyper planes).

Linear Kernel: The Linear kernel is the simplest kernel function. It is given by the inner product <x,y> plus an optional constant c. Kernel algorithms using a linear kernel are often equivalent to their non-kernel counterparts, i.e. KPCA with linear kernel is the same as standard PCA.

$$k(x, y) = x^T y + c$$

Polynomial Kernel: The Polynomial kernel is a non-stationary kernel. Polynomial kernels are well suited for problems where all the training data is normalized.

$$k(x, y) = (\alpha x^T y + c)^d$$

Adjustable parameters are the slope alpha, the constant term c and the polynomial degree d.

Gaussian Kernel: The Gaussian kernel is an example of radial basis function kernel.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

Alternatively, it could also be implemented using

$$k(x, y) = \exp\left(-\gamma \|x - y\|^2\right)$$

The adjustable parameter sigma plays a major role in the performance of the kernel, and should be carefully tuned to the problem at hand. If overestimated, the exponential will behave almost linearly and the higher-dimensional projection will starts to lose its non-linear power. In the other hand, if underestimated, the function will lack regularization and the decision boundary will be highly sensitive to noisy training data.

The SVM gives the following advantages over neural networks or other AI methods (link for more details http://www.svms.org).

SVM training always finds a global minimum, and their simple geometric interpretation provides fertile ground for further investigation.

Most often Gaussian kernels are used, when the resulted SVM corresponds to an RBF network with Gaussian radial basis functions. As the SVM approach "automatically" solves the network complexity problem, the size of the hidden layer is obtained as the result of the QP procedure. Hidden neurons and support vectors correspond to each other, so the center problems of the RBF network is also solved, as the support vectors serve as the basis function centers.

Classical learning systems like neural networks suffer from their theoretical weakness, e.g. back-propagation usually converges only to locally optimal solutions. Here SVMs can provide a significant improvement.

The absence of local minima from the above algorithms marks a major departure from traditional systems such as neural networks.

SVMs have been developed in the reverse order to the development of neural networks (NNs). SVMs evolved from the sound theory to the implementation and experiments, while the NNs followed more heuristic path, from applications and extensive experimentation to the theory.

4. Proposed Algorithm

Here we detail the proposed algorithm for classification of Fraud Transactions.
Step 1: Read the given data.
Step 2: Re-categorize the data into five groups as transaction month, date, day, amount of transaction & difference between successive transaction amounts.
Step 3: Make each transaction data as vector of five fields.
Step 4: Make two separate groups of data named True & False transaction group (if false transaction data is not available add randomly generate data in this group).

Step 5: Select one of three kernels (Linear, Quadratic, and RBF).
Step 6: Train SVM.
Step 7: Save the classifier.
Step 8: Read the current Transaction.
Step 9: Repeat the process from **step1** to **step3** for current transaction data only.
Step 10: Place the saved classifier & currently generated vector in classifier.
Step 11: Take the generated decision from the classifier.

5. IMPLEMENTATION

Since there is no real data is available because of privacy maintained by banks. Hence for testing of implementation of our algorithm we generated the data of true & false Transaction using different mean & variance & then mixed them with different probability. We used the MATLAB for the implementation of the algorithm because of its rich sets of mathematical functions and also supporting the inbuilt functions for SVM.

6. RESULTS

The results are simulated for five different Fraud probabilities from 0.3 to 0.5 & changing the training data size from 30 to 100, then according to outputs of program the following tables are drawn which shows

TPR = True Positive Rate
TNR = True Negative Rate
FPR = False Positive Rate
FNR = False Negative Rate

Complete details of these parameters are discussed in (http://en.wikipedia.org/wiki/Rec eiver _ operating_characteristic).

Kernel Type: Linear

Total DATA	Fraud Prob.	TPR	TNR	FPR	FNR	Accu--racy
30	0.30	0.90	0.72	0.15	0.18	0.83
30	0.40	0.61	0.59	0.38	0.41	0.60
30	0.50	0.26	0.77	0.33	0.50	0.56
60	0.30	0.98	0.22	0.38	0.03	0.72
60	0.40	0.77	0.61	0.32	0.26	0.70
60	0.50	0.70	0.75	0.29	0.24	0.73
100	0.30	0.89	0.27	0.39	0.20	0.67
100	0.40	0.65	0.43	0.51	0.38	0.54
100	0.50	0.81	0.48	0.38	0.24	0.67

Kernel Type: Quadratic

Total DATA	Fraud Prob.	TPR	TNR	FPR	FNR	Accu--racy
30	0.30	0.96	0.93	0.03	0.06	0.95
30	0.40	0.95	0.81	0.08	0.09	0.91
30	0.50	0.91	0.88	0.08	0.11	0.90
60	0.30	0.98	0.75	0.06	0.08	0.93
60	0.40	0.91	0.67	0.25	0.10	0.81
60	0.50	0.89	0.75	0.18	0.14	0.83
100	0.30	0.92	0.40	0.27	0.16	0.76
100	0.40	0.87	0.54	0.26	0.21	0.75
100	0.50	0.64	0.69	0.35	0.30	0.67

Kernel Type: RBF

Total DATA	Fraud Prob.	TPR	TNR	FPR	FNR	Accu--racy
30	0.30	0.98	0.92	0.03	0.01	0.97
30	0.40	0.98	0.94	0.03	0.01	0.97
30	0.50	0.99	0.94	0.05	0.01	0.97
60	0.30	0.98	0.89	0.06	0.03	0.94
60	0.40	0.94	0.93	0.07	0.04	0.93
60	0.50	0.97	0.94	0.05	0.03	0.96
100	0.30	0.98	0.90	0.05	0.02	0.95
100	0.40	0.97	0.98	0.02	0.04	0.97
100	0.50	0.95	0.93	0.05	0.06	0.94

This shows that the RBF kernel outperform to Linear & quadratic kernel in all fields of comparison it has maximum accuracy up to 97%, maximum TPR(99%),maximum TNR(98%) & maximum FPR(7%),maximum FNR(6%), it also behaves almost same for all type of data set generated(having very low fraud data & high fraud data)

7. CONCLUSION

Referring to results we can say that proposed algorithm with RBF kernel gives the better results in comparison with the previous papers we have discussed before & hence can be used for automatic Credit card Fraud detection with excellent accuracy & minimum false alarm.

We can enhance this model for dynamic improvements in training of classifiers using different SVM models like incremental SVM detrimental SVM, evolutionary SVM etc. but we leave this job for future.

REFERENCES

[1] Credit Card Fraud Detection Using Hidden Markov Model Abhinav Srivastava, Amlan Kundu, Shamik Sural, Senior Member, IEEE, and Arun K. Majumdar, Senior Member, IEEE Transaction on Dependable and Secure computing Vol. 5, No. 1, Jan.-March 2008.

[2] Research on Credit Card Fraud Detection Model Based on Distance Sum Wen-Fang YU, Na Wang 2009 International Joint Conference on Artificial Intelligence.

[3] Credit Card Fraud Detection with a Neural-Network Sushmito Ghosh and Douglas L. Reilly Nestor, Inc. Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences, 1994.

[4] Application of Classification Models on Credit Card Fraud Detection Aihua Shen, Rencheng Tong, Yaochen Deng School of Management, Graduate University of the Chinese Academy of Sciences, Beijing, 100084, China, 2007 IEEE.

[5] Research on Credit Card Fraud Detection Model Based on Similar Coefficient Sum Chun-Hua JU, Na Wang 2009 First International Workshop on Database Technology and Applications.

[6] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[7] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neutral Networks and Fuzzy Logic Models*. Cambridge, MA: MIT Press, 2001.

[8] *Research Article* Bird Species Recognition Using Support VectorMachines Hindawi Publishing Corporation EURASIP Journal on Advances in Signal Processing Volume 2007, Article ID 8637,8pages doi : 10.1155 / 2007 / 38637.