CREDIT CARD FRAUD DETECTION USING HIDDEN MARKOV MODEL

Divya.Iyer,Arti Mohanpurkar,Sneha Janardhan,Dhanashree Rathod,Amruta Sardeshmukh
Department of Computer engineering and Information Technology
MMIT
Pune ,India
divi2006@rediffmail.com, yasharti@gmail.com, sneha.janardhan20@gmail.com, dhanashreerathod4@gmail.com, amrutasardeshmukh@gmail.com

***Abstract*— Since past few years there is tremendous advancement in electronic commerce technology, and the use of credit cards has dramatically increased. As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. In this paper we present the necessary theory to detect fraud in credit card transaction processing using a Hidden Markov Model (HMM). An HMM is initially trained with the normal behavior of a cardholder. If an incoming credit card transaction is not accepted by the trained HMM with sufficiently high probability, it is considered to be fraudulent. At the same time, we try to ensure that genuine transactions are not rejected by using an enhancement to it(Hybrid model).In further sections we compare different methods for fraud detection and prove that why HMM is more preferred method than other methods.**

Keywords-Credit card , fraud , Hidden Markov Model ,Hybrid model

Introduction

An e-commerce payment system facilitates the acceptance of electronic payment for online transactions. Also known as a sample of Electronic Data Interchange (EDI), e-commerce payment systems have become increasingly popular due to the widespread use of the internet-based shopping and banking. In the early years of B2C transactions, many consumers were apprehensive of using their credit and debit cards over the internet because of the perceived increased risk of fraud. There are numerous different payments systems available for online merchants. These include the traditional credit, debit and charge card but also new technologies such as digital wallets, e-cash, mobile payment and e-checks. Another form of payment system is allowing a 3rd party to complete the online transaction for you. These companies are called Payment Service Providers (PSP).

The Credit Card is a small plastic card used to buy the goods and services on the cardholder's promise to pay for it later. It's estimated that credit card fraud events causes loses of about $2 billion a year globally to the credit card industry [1]. Unfortunately credit card fraud is a fact-of-life for all merchants who accept credit card payments as part of their business operation. With the increasing transition to online merchandising via the Internet, online credit card fraud is a serious issue. A business requires a sound order-confirmation-system if we want to avoid getting 'ripped-off', being subject to bank 'charge-backs' and/or constantly arranging refunds for fraudulent transactions.

The credit card fraud-detection domain presents a number of challenging issues [3]:

- There are millions of credit card transactions processed each day. With Customer-independent model, it is very difficult to process the massive amounts of data efficiently.
- The data are highly skewed—many more transactions are legitimate than fraudulent. Typical accuracy based mining techniques can generate highly accurate fraud detectors by simply predicting that all transactions are legitimate, although this is equivalent to not detecting fraud at all. To overcome this problem, some researchers have developed methods to generate training sets of labeled transactions with a desired distribution by replicating transaction records labeled fraud.
- Some factors used in some detection models, such as age and income of cardholder, which are significant in the detection models, are not available or unreliable in practice.

To address above issues, we design a behavior-based fraud detection model. In this model, no demographic information is needed, and also data used to establish the model are transaction records of a single credit card instead of ones from many different credit cards. Instead of supervised learning algorithms, unsupervised learning algorithms are employed to detect outlier patterns from normal ones. Behavior-based fraud detection model means that the data used in the model are from the transactional behavior of cardholder directly or derived from them. And personal data such as age, gender used in other models [2, 3, 4] are not used in our behavior-based model. There are many kinds of credit card transactions such as on-line commercial transactions or face to- face ones.

In this paper, we model the sequence of operations in credit card transaction processing using a Hidden Markov Model (HMM) and show how it can be used for the detection of frauds. An HMM is initially trained with the normal behavior of a cardholder. If an incoming credit card transaction is not accepted by the trained HMM with

978-1-4673-0126-8/11/$26.00 ©2011 IEEE

sufficiently high probability, it is considered to be fraudulent. At the same time, we try to ensure that genuine transactions are not rejected. We present detailed experimental results to show the effectiveness of our approach and compare it with other techniques available in the literature [2].

The first section gives a brief idea of necessity for credit card fraud detection also giving an overview of HMM model as a solution to it.The next section deals with the different research works done on credit card fraud detection which helps us to decide the most efficient method for it.Finally we conclude with the result based on various analysis done.

I. RELATED WORK

To detect credit card fraud there are multiple approaches like-

1. A Fusion Approach Using Dempster-Shafer Theory And Bayesian Learning.
2. Blast-Ssaha Hybridization.
3. Hidden Markov Model.
4. Fuzzy Darwinian Detection.
5. Bayesian And Neural Network.

Different Algorithms are there for each Model.

As mentioned in [4]First approach i.e. DEMPSTER-SHAFER THEORY basically proposes Fraud Detection System using information fusion and Bayesian learning in which evidences from current aswell as past behavior are combined together and depending on certain type shopping behavior establishes an activity profile for every cardholder. It has advantages like: - high accuracy, processing speed, reduces false alarm, improves detection rate, applicable in E-commerce. But one disadvantage of this approach is that it is highly expensive.Refer table 1.

As the name suggest BLAST-SSAHA HYBIDIZATION is a hybridization of BLAST and SSAHA algorithms which is referred as BLAH-FDS algorithm. This algorithm is basically the efficient two-stage sequence alignment algorithm which is used for analyzing spending behavior of customers. The performance of this algorithm is good, also accuracy is high. It is useful in telecommunication and banking fraud detection and processing speed is also high. But disadvantage is that it does not detect cloning of credit cards. [4]

In the third approach Baum Welch algorithm is used for training purpose and K-means algorithm for clustering.HMM sores data in the form of clusters depending on three price value ranges low, medium and high. The probabilities of initial set of transaction have chosen and FDS checks whether transaction is genuine or fraudulent. Since HMM maintains a log for transactions it reduces tedious work of employee but produces high false alarm as well as high false positive. Refer table 2

Fuzzy Darwinian Detection is Evolutionary-Fuzzy system which uses genetic programming for evolving fuzzy logic rules. It classifies the transactions into suspicious and non-suspicious. It comprises of Genetic Programming (GP) search algorithm and a fuzzy expert system. This approach has very high accuracy and produces a low false alarm. But it is not applicable in online transactions. Also it is highly expensive and processing speed is low.[4] Refer table 3.

Bayesian and Neural network approach is automatic credit card fraud detection system and type of artificial intelligence programming which is based on variety of methods including machine learning approach, supervised and data mining for reasoning under uncertainty. The advantage of neural network is that it learns and does not need to be reprogrammed. Its processing speed is higher than Bayesian neural networks but it needs high processing time for large neural networks. Whereas Bayesian neural networks provide good accuracy but needs training of data to operate and requires high processing speed.[4]. Refer table 4.

The Hidden Markov Model (HMM) is a variant of a finite state machine having a set of hidden states, Q, an output alphabet (observations), O, transition probabilities, A, output (emission) probabilities, B, and initial state probabilities, Π. The current state is not observable. Instead, each state produces an output with a certain probability (B). Usually the states, Q, and outputs, O, are understood, so an HMM is said to be a triple, (A, B, Π).

In our work we will be considering the following parameters as explained in [2]:

1. N is the number of states in the model. We denote the set of states S :S1; S2; . . . SN , where Si, I=1; 2; . . .; N is an individual state. The state at time instant t is denoted by qt.
2. Observations (symbols) O = { o_k }, k = 1, . . . , M . In our case we will be using the 3 price ranges high, medium and low. Hence, O={l,m,h} so M=3.
3. Transition probabilities A = {a_{ij} = P(q_j at t +1 | q_i at t)}, where P(a | b) is the conditional probability of a given b, t = 1, . . . , T is time, and q_i in Q. Informally, A is the probability that the next state is q_j given that the current state is q_i.
4. Emission probabilities B = { b_{ik} = $b_i(o_k)$ = P(o_k | q_i) }, where o_k in O. Informally, B is the probability that the output is o_k given that the current state is q_i.
5. Initial state probabilities Π = {p_i = P(q_i at t = 1)}.

There are 3 canonical problems to solve with **HMM**s as described in[6]:

1. Given the model parameters, compute the probability of a particular output sequence. This problem is solved by the Forward and Backward algorithms .
2. Given the model parameters, find the most likely sequence of (hidden) states which could have generated a given output sequence. Solved by the Viterbi algorithm and Posterior decoding.
3. Given an output sequence, find the most likely set of state transition and output probabilities. Solved by the Baum-Welch algorithm

Forward Algorithm

Let $\alpha_t(i)$ be the probability of the partial observation sequence $O_t = \{o(1), o(2), \ldots , o(t)\}$ to be produced by all possible state sequences that end at the i-th state.

$$\alpha_t(i) = P(o(1), o(2), \ldots , o(t) \mid q(t) = q_i).$$

Then the unconditional probability of the partial observation sequence is the sum of $\alpha_t(i)$ over all N states.

The Forward Algorithm is a recursive algorithm for calculating $\alpha_t(i)$ for the observation sequence of increasing length t . First, the probabilities for the single-symbol sequence are calculated as a product of initial i-th state probability and emission probability of the given symbol o(1) in the i-th state. Then the recursive formula is applied. Assume we have calculated $\alpha_t(i)$ for some t. To calculate $\alpha_{t+1}(j)$, we multiply every $\alpha_t(i)$ by the corresponding transition probability from the i-th state to the j-th state, sum the products over all states, and then multiply the result by the emission probability of the symbol o(t+1). Iterating the process, we can eventually calculate $\alpha_T(i)$, and then summing them over all states, we can obtain the required probability.

Backward Algorithm

In a similar manner, we can introduce a symmetrical backward variable $\beta_t(i)$ as the conditional probability of the partial observation sequence from o(t+1) to the end to be produced by all state sequences that start at i-th state (3.13).

$$\beta_t(i) = P(o(t+1), o(t+2), \ldots , o(T) \mid q(t) = q_i).$$

The Backward Algorithm calculates recursively backward variables going backward along the observation sequence. The Forward Algorithm is typically used for calculating the probability of an observation sequence to be emitted by an HMM, but, as we shall see later, both procedures are heavily used for finding the optimal state sequence and estimating the HMM parameters.

Posterior decoding

There are several possible criteria for finding the most likely sequence of hidden states. One is to choose states that are individually most likely at the time when a symbol is emitted. This approach is called posterior decoding.

Let $\lambda_t(i)$ be the probability of the model to emit the symbol o(t) being in the i-th state for the given observation sequence O.

$$\lambda_t(i) = P(q(t) = q_i \mid O).$$

It is easy to derive that

$$\lambda_t(i) = \alpha_t(i)\, \beta_t(i) / P(O) , i =1, \ldots , N , t =1, \ldots , T$$

Then at each time we can select the state q(t) that maximizes $\lambda_t(i)$.

$$q(t) = \arg\max \{\lambda_t(i)\}$$

Posterior decoding works fine in the case when HMM is ergodic, i.e. there is transition from any state to any other state. If applied to an HMM of another architecture, this approach could give a sequence that may not be a legitimate path because some transitions are not permitted.

Viterbi algorithm

The Viterbi algorithm chooses the best state sequence that maximizes the likelihood of the state sequence for the given observation sequence.

Let $\delta_t(i)$ be the maximal probability of state sequences of the length t that end in state i and produce the t first observations for the given model.

$$\delta_t(i) = \max\{P(q(1), q(2), ..., q(t-1) ; o(1), o(2), \ldots , o(t) \mid q(t) = q_i).\}$$

The Viterbi algorithm is a dynamic programming algorithm that uses the same schema as the Forward algorithm except for two differences:

1. It uses maximization in place of summation at the recursion and termination steps.
2. It keeps track of the arguments that maximize $\delta_t(i)$ for each t and i, storing them in the N by T matrix $\boldsymbol{\psi}$. This matrix is used to retrieve the optimal state sequence at the backtracking step.

Baum-Welch algorithm

Let us define $\xi_t(i, j)$, the joint probability of being in state q_i at time t and state q_j at time $t+1$, given the model and the observed sequence:

$$\xi_t(i, j) = P(q(t) = q_i, q(t+1) = q_j \mid O, \Lambda)$$

Therefore we get

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j(o(t+1))\beta_{t+1}(j)}{P(O \mid \Lambda)}$$

The probability of output sequence can be expressed as

$$P(O \mid \Lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij}\, b_j(o(t+1))\beta_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i)\, \beta_t(i)$$

The probability of being in state q_i at time t:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = \frac{\alpha_t(i)\, \beta_t(i)}{P(O \mid \Lambda)}$$

Estimates

Initial probabilities:

$$\bar{p}_i = \gamma_1(i)$$

Transition probabilities:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

Emission probabilities:

$$\bar{b}_{jk} = \frac{\sum_{t}^{*} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

In the above equation Σ^* denotes the sum over t so that $o(t) = o_k$.

In this way we will apply these algorithms to train the HMM .It includes clustering algorithm also.

For performance evaluation of the tree most popular clustering techniques K-Mean clustering, Hierarchical clustering and Farthest first clustering, we have taken three datasets containing nominal attributes type that is all these datasets contains the continuous attributes. As suggested by Pallavi , Sunila Godara et.al[7] the result analysis shows that K-means algorithm performs well without inserting the principle component analysis filter compared to the Hierarchical clustering algorithm and Farthest first clustering since it have less instances of incorrectly clustered objects on the basis of class clustering.

K-Means Clustering:-

The K-Means Clustering consists of basic steps. In this algorithm we initially determine the number of clusters present ,assume it to be K and we also assume the center or centroid of these clusters. Now we can consider an random objects as the initial centroids or we can also consider the sequence of first K objects as the centroids. Later the K-Means algorithm will carry out the iteration of below stated 3 steps till the convergence.

Step1: Determine the centroid coordinate

Step 2. Determine the distance of each object to the centroids

Step 3.Group the object based on minimum distance (find the closest centroid).

II. SUMMARY:-

A highly efficient and accurate credit card fraud detection system is the need of the hour as millions of credit card transactions are being carried out every day. As a result a large amount of research is being carried out in this domain and a number of techniques are proposed overcome credit card fraud. As seen in the previous section our study shows that HMM model is the best alternative to detect online credit card frauds. The HMM uses the K-Means algorithm for clustering purpose which is based on finding the centroids of the clusters and then grouping the minimum distance objects from the centroid into one group. The HMM uses the Baum Welch algorithm for training purpose. It produces high false alarms and the performance is low as compared to the other methods[4]. Though it produces high false alarm as well as high false positive this drawback can be overcome by using Hybrid model as mentioned in [5].

REFERENCES

[1] JERMY QUITTNER."AVOIDING CREDIT CARD FRAUD". http://abcnews.go.com/business/financialSecurity/Story?id=89746&page=12004

[2] Abhinav Srivastava, Amlan Kundu, Shamik Sural, Arun K. Majumdar."Credit Card Fraud Detection using Hidden Markov Model" IEEE transactions on depedable and secure computing.

[3] Aihua Shen; Rencheng Tong; Yaochen Deng."Application of Classification Models on Credit Card Fraud Detection".Service Systems and Service Management, 2007 International Conference on Volume , Issue , 9-11 June 2007 Page(s):1 - 4.

[4] S. Benson Edwin Raj, A. Annie Portia "Analysis on Credit Card Fraud Detection Methods" International Conference on Computer, Communication and Electrical Technology – ICCCET2011, 18th & 19th March, 2011

[5] SandeepPratap Singh, Shiv Shankar P.Shukla,Nitin Rakesh and Vipin Tyagi "Problem Reduction In Online Payment System Using Hybrid Model" International Journal of Managing Information Technology (IJMIT) Vol.3, No.3, August 2011

[6] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE, vol. 77, no. 2, pp. 257-286, 1989.

[7] Pallavi , Sunila Godara "A Comparative Performance Analysis of Clustering Algorithms" International Journal of Engineering Research and Applications (IJERA) Vol. 1, Issue 3, pp.441-445

[8] Amlan Kundu, Suvasini Panigrahi, Shamik Sural and Arun K. Majumdar, "Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning," Special Issue on Information Fusion in Computer Security, Vol. 10, Issue no 4, pp.354- 363, October 2009

[9] Peter J. Bentley, Jungwon Kim, Gil-Ho Jung and Jong-Uk Choi, "Fuzzy Darwinian Detection of Credit Card Fraud," In the 14th Annual Fall Symposium of the Korean Information Processing Society, 14th October 2000.

[10] Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, Bernard Manderick,"Credit card fraud detection using Bayesian and neural networks,"Interactive image-guided neurosurgery, pp.261-270, 1993.

Table 1 Reference[8]

θ_{UT}	θ_{LT}			
	0.20	0.25	0.30	0.35
0.70	83/8	81/7	81/4	78/4
0.75	82.5/7	80/6.5	79/3.5	76/3
0.80	80/6.5	78/6	78/3	75/2.5
0.85	77/5	75/4	73/3	71/2

Variation of mean TP/mean FP (%) with θ_{LT} and θ_{UT}.

Table 2.Reference[2]

Threshold (%)	TP averaged over all the 6 states for different sequence lengths					FP averaged over all the 6 states for different sequence lengths				
	5	10	15	20	25	5	10	15	20	25
30	0.52	0.56	**0.64**	0.58	0.6	0.05	0.05	**0.05**	0.05	0.05
50	0.54	0.54	**0.63**	0.57	0.6	**0.03**	0.05	0.04	0.05	0.05
70	0.50	0.60	0.60	**0.61**	0.59	**0.04**	**0.04**	0.05	0.05	0.05
90	0.42	0.52	**0.59**	0.58	0.57	**0.02**	0.04	0.05	00.05	0.05

Variation of TP and FP with different Sequence Lengths

Table 3 Reference[9].

	[A] Fuzzy Logic with non-overlappingMFs					[B] Fuzzy Logic with overlapping MFs					[C] MP-Fuzzy Logic with overlapping MFs					[D] MP-Fuzzy Logic with smooth MFs				
	R	Training		Test		R	Training		Test		R	Training		Test		R	Training		Test	
		TP%	FN%	TP%	FN%		TP%	FN%	TP%	FN%		TP%	FN%	TP%	FN%		TP%	FN%	TP%	FN%
1	3	6.09	3.81	10.4	3.35	2	100	0	100	85.1	16	10.9	5.79	100	100	5	48.6	5.79	42.5	10.3
2	2	44.1	5.79	47.8	9.45	3	100	1.67	99.7	6.38	3	1.37	5.64	99.7	100	10	41.6	5.79	47.6	12.5
3	3	46.8	5.18	46.9	6.09	3	100	5.78	100	5.79	4	1.67	5.64	86.9	100	16	42.7	5.94	42.9	6.40

Intelligibility (number of rules) and accuracy (number of correct classifications of "suspicious" items) of rule sets for test and training data.
R shows the number of rules in the generated rule set and **TP** and **FN** is represented in %.

Table 4.Reference[10]

experiment	±10% false pos	±15% false pos
ANN-fig 2(a)	60% true pos	70% true pos
ANN-fig 2(a)	47% true pos	58% true pos
ANN-fig 2(c)	60% true pos	70% true pos
BBN-fig 2(e)	68% true pos	74% true pos
BBN-fig 2(g)	68% true pos	74% true pos

Results with ANN and BNN