CREDIT CARD FRAUD DETECTION USING HIDDEN MARKOV MODEL

Term Project

Semester-II, 2020-21

IME625A

Introduction to Stochastic Processes and Their Applications

Submitted

By

Group 7

Sk Raju (20114021)

Harsh Jain (20114007)

Shruti (180742)

May 2020

Department of Industrial & Management Engineering

Indian Institute of Technology Kanpur

Contents

1 Summary

The use of credit cards has risen rapidly because of brisk advancements in electronic commerce technologies. As credit cards become the most common payment method for both online and offline purchases, cases of credit card fraud are hiking as well. We used a Hidden Markov Model (HMM) to model the sequence of operations in credit card transaction processing and demonstrate how to use it to identify fraud. An HMM is initially conditioned on a cardholder's normal behaviour using the KMeans algorithm for clustering. HMM calculated the probability of sequence, and when a qualified HMM does not authorize a credit card purchase with a high enough likelihood, it is classified as 'Fraudulent.' This approach comes out to be efficient, scalable and quite accurate in real-time predictions.

2 Credit card and fraudulent transaction

Online shopping and cashless payments are much popular nowadays. The credit card allows the user to purchase now and pay later, which may be why a rapid increase in credit card usage. The credit card users in India touched 52 million by 2019. (Research and Markets, 2020) Fraudulent transactions may happen if the attacker got the card or the security information anyhow. Every customer has its spending pattern and tends to exhibit specific behaviourist profiles. The bank has the customer's detail. Every cardholder represents a set of practices containing information about the type of purchase, amount of purchase, time from last purchase, etc. Deviation from this trend can be a threat to the system. When the customer is going for the next purchase, the fraud detection system matches the spending amount and other information with the profile by transition probabilistic calculation based on Hidden Markov Model. The system will decide accordingly whether the transaction is genuine or fraudulent.

3 Markov model

A Markov chain is a discrete-time discrete-state stochastic process. In Markov Model, the prediction of the following observation in a sequence will depend only on the value of the immediately preceding observations, and the knowledge of the past will be irrelevant. (Ross, 2019)

Product rule for the joint distribution of a sequence:

$$p(x_1, \dots, x_n) = \prod_{n=1}^{N} p(x_n | x_1, \dots, x_{n-1})$$

In the first-order Markov Model, the conditional distribution on the right-hand side of the equation is independent of all previous observations except the most recent one. In the case of the second-order Markov Model, the same depends on the last two observations. The phenomenon can be generalized in a similar fashion to the Mth order Markov Model. (Bishop, 2006)

From first-order Markov Model

$$p(x_n | x_1, \dots, x_{n-1}) = p(x_n | x_{n-1})$$

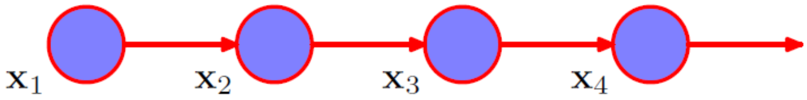


Figure: A first order Markov Model

4 Hidden Markov Model

4.1 Introduction to HMM

A Hidden Markov model is a double embedded stochastic process with two hierarchy levels (Abhinav Srivastava, 2008). It has finite states and transition probabilities. In the case of HMM, the underlying Markov chain states are unobserved or hidden, but the sequence of signals is visible. The following figure shows a graphical structure of an HMM. The latent variables are discrete multinomial variables s_n and the corresponding visible signal at that observation is v_n.

HMM is significantly used in speech recognition, natural language modelling, online handwriting recognition, biological sequence analysis etc. (Bishop, 2006).

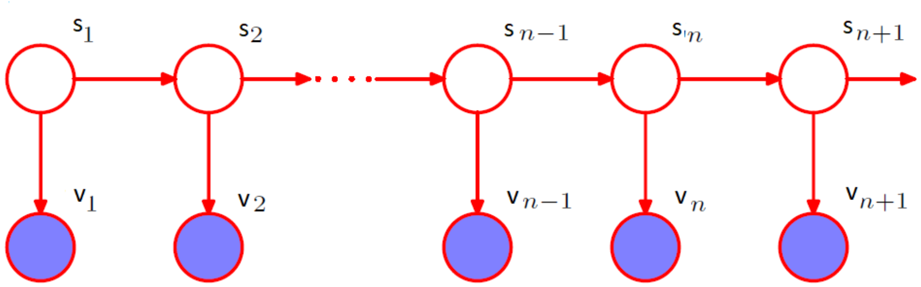


Figure: Graphical structure of an HMM

The probability distribution of s_n depends on the previous hidden variable s_{n-1} through a conditional probability distribution $p(s_n|s_{n-1})$. (Bishop, 2006)

4.2 Characterization of a discrete HMM

- S is the set of hidden states of the model.

$$S = \{s_1, s_2, \dots, s_N\}$$

Where N is the number of hidden states and q_t is the state at time t, $q_t \in S$

- V the set of observable symbols of the model.

$$V = \{v_1, v_2, \dots, v_M\}$$

Where M denotes the distinct observation symbols per state and o_t is the observation at time t, $o_t \in V$

- $A = \{a_{ij}\}$: the transition probability from s_i to s_j. A is the transition probability matrix. (Daniel Jurafsky, 2020)

$$A = a_{11} \dots a_{ij} \dots a_{NN}$$

$$a_{ij} = \mathrm{P}(q_{t+1} = s_j | q_t = s_i)$$

$$\sum_{j=1}^{N} a_{ij} = 1 \; \forall i \quad 1 \le i \le N, 1 \le j \le N$$

- $B = \{b_j(k)\}$: the probability of emitting a visible signal x_k from hidden state z_j. B is the set of emission probabilities b_{jk}.

$$B = b_1(1) \dots b_j(k) \dots b_N(M)$$

$$b_i(k) = \mathrm{P}(v_k \; at \; t | q_t = s_j)$$

$$\sum_{k=1}^{M} b_i(k) = 1 \ \forall j \quad 1 \le j \le N, 1 \le k \le M$$

- $\pi = \{\pi_i\}$: distribution of initial state.

$$\pi_i = P(q_1 = s_i) \quad 1 \le i \le N$$

Conventionally an HMM is typically written as $\lambda = \{A, B, \pi\}$

4.3 An example of HMM

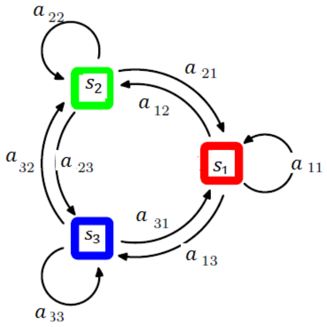


Figure: Graphical representation of HMM example

4.4 Generation of HMM observations

1. Initialization: Choosing of the initial state, $q_1 = s_i$, from the initial state distribution π.
2. For $t = 1 \; to \; T$:
   - Choosing $o_t = v_k$, from the emission probability distribution in the state $s_i, \; b_i(k)$.
   - Transition to a new state $q_{t+1} = s_j$, as per the state transition probability distribution for state s_i, a_{ij}.
3. Increment t by 1, return to step 2 if $t \le T$; else, terminate.

4.5 Three basics HMM problems

Three fundamental problems of interest must be solved for the model to be helpful in real-world applications.

4.5.1 Scoring

An observation sequence $O = \{o_1, o_2, \dots, o_T\}$ and a model $\lambda = \{A, B, \pi\}$ are given. We need to compute the probability of the observation sequence $P(O|\lambda)$.

4.5.1.1 Evaluation of $P(O|\lambda)$ by crude method

$$P(O|\lambda) = \sum_{for\ all\ Q} P(O, Q|\lambda) \quad where\ a\ state\ sequence\ Q = q_1, q_2, \dots, q_T$$

$$P(O, Q|\lambda) = P(O|Q, \lambda)P(Q|\lambda)$$

Considering a fixed state sequence: $Q = q_1 q_2 \dots q_T$

$$P(O, Q|\lambda) = b_{q_1}(o_1) b_{q_2}(o_2) \dots b_{q_T}(o_T)$$

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$

Therefore:

$$P(O|\lambda) = \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(o_1)\, a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T)$$

The complexity of this process is $O(TN^T)$Requires very high computation power to process. The computation cost can be less in recursive methods, which is depicted in the following.

4.5.1.2 The Forward Algorithm

$\alpha_t(i)$ is defined as the probability of the partial observation sequence up to time t and state s_i at time t in the given model. It is also known as forward variable.

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = s_i|\lambda)$$

For the initial state

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N$$

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

By induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), \quad 1 \le t \le T-1, 1 \le j \le N$$

The complexity of this process is of $O(TN^2)$

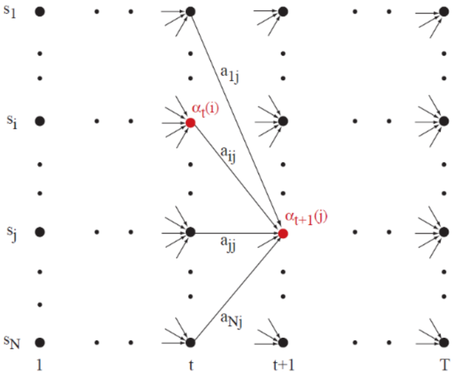


Figure: Forward Algorithm pictorial representation

4.5.2 Matching

An observation sequence $O = \{o_1, o_2, \dots, o_T\}$ is given for a model $\lambda = \{A, B, \pi\}$. We need to choose a somewhat optimum state sequence $Q = \{q_1, q_2, \dots, q_T\}$.

4.5.2.1 The Backward Algorithm

$\beta_t(i)$ is defined as the probability of the partial observation sequence from time t+1 to the end, given state s_i at time t in the given model, also known as backward variable.

$$\beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T | q_t = s_i, \lambda)$$

At time $t = T$,

$$\beta_t(i) = 1\,, \quad 1 \leq i \leq N$$

And,

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1)\beta_1(i)$$

By induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1})\beta_{t+1}(j)\ , \quad t = T-1, T-2, \dots, 1 \ \ ; \ 1 \leq i \leq N$$

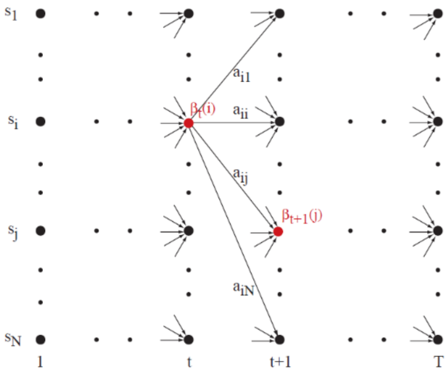


Figure: Backward Algorithm pictorial representation

4.5.2.2 Optimal State Sequences

$\gamma_t(i)$ is defined as the probability of being in the state s_i at time t, provided the observation sequence and the model

$$\gamma_t(i) = P(q_t = \ s_i|O, \lambda) \qquad \sum_{i=1}^{N} \gamma_t(i) = 1, \qquad \forall t$$

Then the individually most likely state, q_t, at time t is:

$$q_t = \underset{1 \leq i \leq N}{argmax}\, \gamma_t(i) \quad , \qquad 1 \leq t \leq T$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}$$

4.5.3 Training

We need to adjust the model parameters $\lambda = \ \{A, B, \pi\}$ to maximize $P(O|\lambda)$

4.5.3.1 Baum-Welch Re-estimation (The Forward Backward Algorithm)

$$\xi_t(i,j) = P\big(q_t = s_i, q_{t+1} = s_j \big| O, \lambda\big)$$

Then:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$

After summing up we get,

$$\sum_{t=1}^{T-1} \gamma_t(i) = expected\ number\ of\ transitions\ from\ s_i$$

$$\sum_{t=1}^{T-1} \xi_t(i) = expected\ number\ of\ transitions\ from\ s_i\ to\ s_j$$

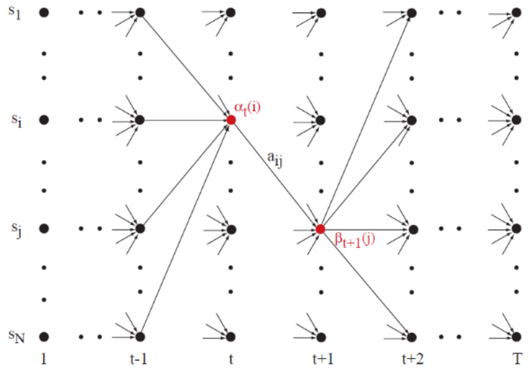


Figure: Baum-Welch re-estimation pictorial representation

Formulas:

$$\bar{\pi} = expected\ number\ of\ times\ in\ state\ s_i\ at\ (t = 1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{expected\ number\ of\ transitions\ from\ state\ s_i\ to\ s_j}{expected\ number\ of\ transitions\ from\ state\ s_i} = \frac{\sum_{t=1}^{T-1} \xi_t\,(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{expected\ number\ of\ times\ in\ state\ s_i\ with\ symbol\ v_k}{expected\ number\ of\ times\ in\ state\ s} = \frac{\sum_{\substack{t=1 \\ O_t=v_k}}^{T-1} \gamma_t\,(j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

If $\lambda = (A, B, \pi)$ is the initial model and $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ is the re-estimated model, then it can be shown that:

- Either the initial model, λ, defines a critical point of the likelihood function, in which case $\bar{\lambda} = \lambda$, or
- Model $\bar{\lambda}$ is more likely than λ in the sense that $P(O|\bar{\lambda}) > P(O|\lambda)$, i.e., a new model $\bar{\lambda}$ was found from which the observation sequence is more likely to have been produced.
- Thus, we can improve the probability of O being observed from the model if we iteratively use $\bar{\lambda}$ in place of λ and repeat the re-estimation until some limiting point is reached. The resulting model is called the maximum likelihood HMM. (James Glass, 2003) (Biswas, 2014).
- The forward-backward algorithm directs to local maxima only, and that needs to be taken care of. The optimization surface is very complex and has many local maxima. (Rabiner, 1989)
- The re-estimation formulas can be derived directly by maximizing Baum's auxiliary function over $\bar{\lambda}$.

$$Q(\lambda|\bar{\lambda}) = \sum_{Q} P(Q|O,\lambda) log[P(O,Q|\lambda)]$$

- Maximization of $Q(\lambda|\bar{\lambda})$ leads to an increased likelihood

$$\underset{\bar{\lambda}}{max}\,[Q(\lambda|\bar{\lambda})] \Rightarrow P(O|\bar{\lambda}) \geq P(O|\bar{\lambda})$$

Eventually, the critical function converges to a crucial point. (Rabiner, 1989)

5 Implementation of HMM in credit card fraud detection

5.1 Introduction to Modeling

We can observe the spending amount at any purchase in a credit card fraud transaction and categorize it under different price ranges. Since the amount of purchase depends upon the type of purchase which is linked to the line of business of an individual and to determine the deviation in the transaction type of purchase is more stable than the amount of purchase, but this data is hidden from the fraud detection system. We can model this as hidden states in HMM. We took a dataset of 193 transactions out of which we used 143 for training and rest 50 for testing purpose.

5.2 Defining observable symbols

The price range has been determined using KMeans clustering based on the customer's earlier transactions. As per the past spending behaviour, three clusters were built and modelled as three observable symbols. Whenever any new spending occurs, we add it to the cluster according to the minimum absolute deviation of spending amount with respect to the centroid (average of all transaction in that cluster) of the clusters.

Observational symbols	low	medium	high
Centroids	15.236	147	899.25
Fraction total transaction	131/143	8/143	4/143

5.2.1 Spending profile of the card holder

According to the distribution, this cardholder belongs to a low spending group, hence corresponding to a low spending profile. These Fraction of total transaction data from the table was used to initialize our model in the subsequent process. (Abhinav Srivastava, 2008)

5.3 Defining hidden states

Marchants' all possible lines of businesses and other essential information were modelled as hidden states and we assume there are three hidden states altogether. (Abhinav Srivastava, 2008)

5.4 Model initialization

We used Baum-Welch algorithm to fit the data into the HMM model and predicted probabilities for fraudulent and genuine transaction. To better efficiency of the algorithm and make the initial guess of observational emission probabilities more accurate, we used the earlier mentioned cluster distribution fractions as emission probabilities. We assumed the initial guesses to be uniform since there is no a priori knowledge about the state transition probabilities,. (Abhinav Srivastava, 2008)

5.5 Model training

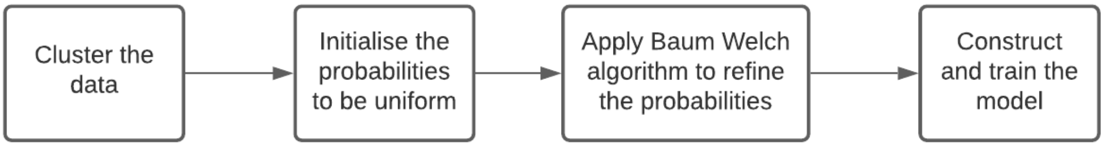
1. Initializing the model with HMM parameters as described earlier.
2. Baum-Welch Re-estimation (The Forward Backward Algorithm) is used for training.
3. The forward algorithm estimated the probability of upcoming observation.

5.6 Model testing or fraud detection

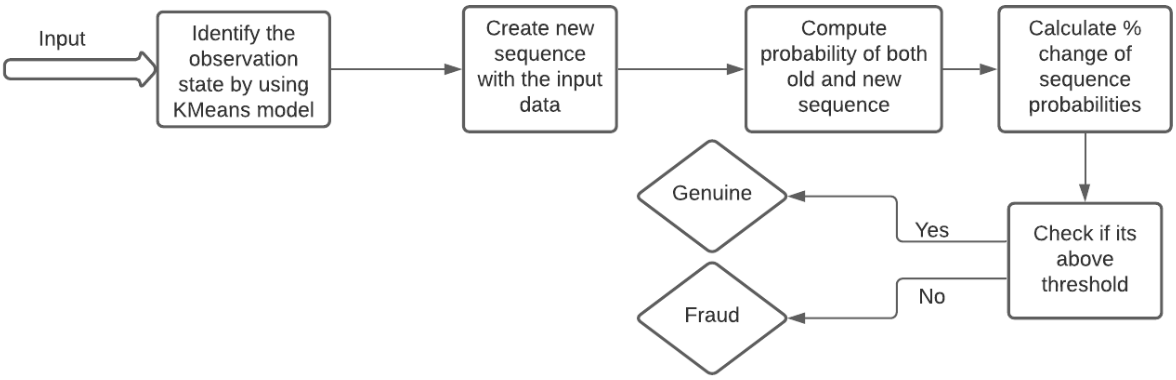
1. After learning the model parameters from HMM training data, an initial sequence of symbols formed. Let $o_1, o_2, \ldots o_R$ be one such sequence of length R, which is recorded till time *t*. By inputting this sequence to the HMM acceptance probability was computed, which is shown in the following.

$$Acceptance\_probability_1 = P(o_1, o_2, o_3, \ldots o_R | \lambda)$$

2. Let o_{R+1} be the symbol emitted by a new transaction at time $t+1$. Then another sequence will be formed with length R by dropping o_1 and appending o_{R+1}. Again, acceptance probability was computed in the same way.
$$Acceptance\_probability_2 = P(o_2, o_3, o_4, \dots o_{R+1}|\lambda)$$
3. The difference between these two acceptance probabilities is calculated to determine the fraudulent transaction.
$$\Delta Acceptance\_probability \ = Acceptance\_probaility_1 - Acceptance\_probability_2$$
4. If the percentage change in acceptance probability is above a threshold, then the last transaction is classified as fraudulent, that is,
$$\frac{\Delta Acceptance\_probability}{Acceptance\_probaility_1} \geq Threshold$$
5. The threshold can be calculated empirically but, in our case, we guessed it as 10%.
6. Test result: Out of 50 test samples 8 were classified as 'Fraudulent' and 42 were classified as 'Genuine'.


Cluster the data
Initialise the probabilities to be uniform
Apply Baum Welch algorithm to refine the probabilities
Construct and train the model

Training schema


Input
Identify the observation state by using KMeans model
Create new sequence with the input data
Compute probability of both old and new sequence
Calculate % change of sequence probabilities
Check if its above threshold
Yes
No
Genuine
Fraud

6 Conclusion

In this project fundamentals of HMM and its application in detection credit card fraudulent transaction is shown. As one of many advantages of HMM, it does not require labelled data. Due to the efficient algorithms, it can give result in seconds, which is a much-needed feature during live transactions. The model is also scalable to huge dataset.

7 Codes

Clustering and some pre-processing parts are done using python and the HMM model is done in R. The codes are attached as separate files.

8 References

Abhinav Srivastava, A. K. (2008). Credit Card fraud Detection Using Hidden Markov Model. *IEEE, 5*(1).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Biswas, P. K. (2014). *Pattern Recognition and Application*. Retrieved from NPTEL: https://drive.google.com/file/d/1IWnrZea42onenx_lT7Dp_FR5RJWxHPh-/view

Daniel Jurafsky, J. H. (2020, December). *Speech and Language Processing.* Retrieved from web.standford.edu.

James Glass, V. Z. (2003). *Automatic Speech Recognition.* Retrieved from MITOPENCOURSEWARE: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-345-automatic-speech-recognition-spring-2003/lecture-notes/lecture10.pdf

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE*, 257-286.

Ross, S. M. (2019). *Introduction to Probability Models.* Los Angeles: Academic Press.