Modeling the Dynamic Trust of Online Service Providers using HMM

Xiaoming Zheng
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
xiaoming.zheng@mq.edu.au

Yan Wang
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
yan.wang@mq.edu.au

Mehmet A. Orgun
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
mehmet.orgun@mq.edu.au

***Abstract*—Online trading takes place in a very complex environment full of uncertainty in which deceitful service providers or sellers may strategically change their behaviors to maximize their profits. The proliferation of deception cases makes it essential to model the dynamics of a service provider and predict the trustworthiness of the service provider in transactions. Recently, probabilistic trust models have been used to assist decision making in computing environments. Although the typical Hidden Markov Model (HMM) has been used to model a provider's behavior dynamics, existing approaches focus only on the outcomes or ignore the hidden characteristics of the HMM model. In this paper, we model the dynamic trust of service providers concerning a forthcoming transaction in light of as much information as we can consider, including the static features, such as the provider's reputation and item price, and the dynamic features, such as the latest profile changes of a service provider and price changes. Based on a service provider's historical transactions, we predict the trustworthiness of the service provider in a forthcoming transaction. In addition, the Mutual Information theories and the Principle Component Analysis method are leveraged to eliminate redundant information and combine essential features to form lower dimensional feature vectors. Furthermore, by adopting Vector Quantization techniques, we apply the discrete HMM in a more powerful way, in which all the features extracted from both contextual information and the rating of each transaction are treated as observations of HMM. We evaluate our approach empirically in order to study its performance. The experiment results illustrate that our approach significantly outperforms the state-of-the-art probabilistic trust methods in accuracy in the cases with complex changes.**

***Keywords*-E-commerce; E-service; Trust prediction**

I. Introduction

In recent years, the rapid development of the Internet has greatly changed people's traditional behaviors. People are increasingly active in various large, open and dynamic network systems including social networks, Peer-to-Peer systems, e-commerce and e-service [20]. Due to the nature of virtual communities, people including service providers and service consumers do not meet or interact physically. In such an environment with uncertainty, the prediction of the dynamic trust about a service provider in online service applications has been growing in importance [10][21][19]. Trust, by definition, is a commitment to a future action based on a belief that it will lead to a good outcome [7]. Without trust, prudent business operators and clients may leave the Internet market and revert back to traditional business [13]. In human society, trust depends on a host of factors like the past experience with a person, opinions, motivations etc. [3]. In electronic commerce and service environments, consumers cannot directly interact with products and workers, and the credibility of online information may be doubtful [16]. As a result, trust mainly relies on the past experience with a service provider or seller, the description of services, the provider's reputation etc. Thus how to establish the trust of a service provider becomes a major challenge to ensure that every forthcoming transaction is reliable for honest participants. As the issue of trust exists in both e-commerce and e-service environments, in this paper, we use the terms "seller" and "service provider" interchangeably.

A number of techniques have been proposed for establishing trust online. These techniques fall under two main categories: security based solutions and social control based solutions [12]. The former techniques include authentication, access control, public key infrastructure etc.; the latter techniques mainly focus on trust recommendations and reputation. So, trust emerges as the most popular concept to manage the uncertainty of service providers' behaviors online [18]. In addition, probabilistic trust can broadly be characterized as aiming to build probabilistic models upon which the future behavior could be predicted [5].

In online e-commerce and e-service environments, such as eBay and Amazon, the system maintains the past transaction information for a certain period which offers the possibility to infer a service provider's future action and could give useful advice to customers. A number of approaches have been proposed to model the behavior of a service provider. The Beta model is an early static model in which the behavior of any service provider is assumed to be represented by a fixed probability distribution over outcomes [8]. The Beta model with a decay factor introduces an exponential decay factor to control the weight of each outcome according the time of occurrence [6]. Although this approach shows success in certain scenarios, it is not effective in other scenarios where the provider's behavior is highly dynamic. There are many factors that affect a seller's behavior [4]. For instance, in e-commerce web sites like eBay[1] and Taobao[2],

[1] http://www.ebay.com
[2] http://www.taobao.com

a seller's behavior, say honest or cheating, may vary unwittingly or consciously change according to different items, different buyers etc. It is more likely to trade imprudently in the afternoon just before the closing time or in peak time [25][24][23]. So, the trust of a seller is dynamically changing.

To deal with the dynamics of a service provider's trust, probabilistic models are the most promising tools to deal with uncertainty. One of the most typical and powerful probabilistic models is Hidden Markov Model (HMM). HMM is a stochastic model appropriate for nonstationary stochastic sequences whose statistical properties undergo distinct random transitions among a set of, say k, different stationary processes. In other words, HMMs are used to model piecewise stationary processes whose statistical properties do not change with time themselves [17]. HMM treats the list of transactions as a Markov chain and assumes the service provider's behavior has finite salient states which determine the distribution of the outcomes of transactions. A given state, which determines the distribution of observations, only depends on its previous state; thus, the trustworthiness of the next transaction could be inferred from the historically recorded list in the system.

The application of HMM for trust prediction has led to many approaches with different efficiencies and precisions. ElSalamouny et al. [6] uses HMM to predict the outcome of the future transaction based on only the list of outcomes happened in the past achieving better performance than the Beta Model based approaches with a forgetting factor [8]. However, a transaction includes contextual information and an outcome/rating. The contextual information, which characterizes all the details of a transaction, may contain more clues leading to the outcome and can be utilized to predict the status of the future transaction. From contextual information, Liu and Datta [11] extract useful features, as observations, to construct an HMM to model the dynamic trust of a seller. However, they directly treat outcomes of transactions as hidden states, which eliminates the hidden characteristic of HMM and limits the ability of HMM to model a service provider' dynamics in trust/reputation.

In this paper, we discuss the advantages and disadvantages of different approaches and propose a new approach as shown in Figure 1. We firstly analyze what features of transactions can affect the outcomes resulting in a more comprehensive characterization of contextual information. Based on the contextual information, we extract not only the static features but also the dynamic changes as features. For instance, some sellers may change their profile before committing deception. Secondly, we boost the execution effectiveness through three steps: (1) We select the most powerful features as observations using information theories [9]. (2) We use the Principal Component Analysis (PCA) algorithm [1] to combine the most powerful features into relatively lower dimensions. (3) We apply Vector Quantization (VQ) techniques [9] to project final feature vectors into discrete values. Lastly, we propose a Discrete Hidden Markov Model (DHMM) to model the trust trend. We treat all the contextual features and outcomes/ratings as HMM observations and predict the most possible rating of the observation of the forthcoming transaction.

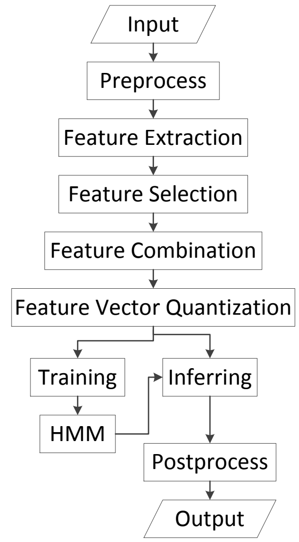

Figure 1. Flowchart of Our Approach

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 describes how we extract useful features as HMM observation sequences. In Section 4, we first introduce the basic knowledge of HMM needed in this paper. Then we describe the HMM based trust approach and discuss the basic method in training HMM. Section 5 discusses the evaluation process and the experimental results respectively. Finally Section 6 gives a summary and discusses future work.

II. Related Work

Modeling a service provider's dynamic behavior is a challenging task. Jøsang proposed the Beta reputation method [8] to combine feedback and derive reputation ratings based on beta probability density functions. The advantage of the Beta reputation system is its flexibility and simplicity as well as its foundation on the theory of statistics. Later on, the Beta model is improved by adding a decay factor to control the weight of data [6], which makes the model focus more on recent data. Zhang and Cohen [26] model agent behaviors with a time window, in which the numbers of successful and failed transactions are aggregated with a forgetting rate and the trustworthiness is adjusted accordingly.

Recently, several Hidden Markov Model based trust methods have been proposed to deal with service provider dynamics. Several studies [6][14] proposed the HMM approach in modeling transaction results. The model is trained on the outcomes of each past transaction, say failure or success. The results of experiments demonstrate that HMM works much better than Beta Model based methods in detecting a service provider's changes and is more suitable for modeling dynamics. However, such an HMM based approach utilizes only the outcomes of the past transactions as the observation sequence but ignores the contextual information about each transaction and makes the prediction a full guess which works well only when service providers' behaviors alternate without any regular rules.

To solve this problem, Liu and Datta [11] propose a contextual information-based Markov model, which extracts features from transaction contextual information as the HMM observation sequences and treats the outcomes directly as the hidden states of the models. Liu and Datta [11] also apply information theories and Multiple Discriminant Analysis to reduce the feature space. This approach utilizes the contextual information and speeds up the calculation as it simplifies the training of HMM to the statistics of the past records. However, it reveals the hidden states and the authors also assume a series of transactions occurring between a seller and the same customer, which can hardly be true in most actual scenarios. In addition, there could be more features to be taken into account, such as price changes, in addition to static features in the contextual information.

Our approach utilizes information theories to select contextual features but we extend them and project feature vectors to discrete observations by VQ after combined by PCA. And we treat both contextual information and ratings as observations to train a Discrete HMM (DHMM), which, unlike previous methods, uses HMM in a different and more effective way. Our method restores the hidden states of HMM by moving ratings to part of the observation vectors. The details will be discussed in the following sections.

III. Producing Features

It is shown by Liu and Datta [11] that a service provider's interaction behavior can be estimated by contextual information. Based on the features extracted from contextual information of transactions, we analyze the dynamic characters of the sellers in online trading web sites and build our own HMM trust model.

We propose four main steps to produce the most effective features from contextual information. Firstly, we conceptually choose as much information related to the behaviors of a seller as possible. Secondly, based on the data, we use information theories like Mutual Entropy to reveal the most powerful features and eliminate the least powerful ones. Thirdly, we use PCA to combine the feature vectors. In other words, the features are projected into another coordinate space with lower dimensionality. Lastly, Vector Quantization techniques are used to project the combined feature vector into different discrete observations that are provided to the Discrete Hidden Markov Model as the input.

A. Feature Extraction

Contextual information refers to all the information that characterizes the transactions and from which classifying features could be extracted. The state-of-the-art features used in HMM models usually can be divided into two main types—static features and dynamic features which also occur in the seller's behaviors. Here following Liu and Datta's method [11], the static features are extracted from three aspects as described below.

1) About the service provider/seller: the contextual information includes the features about the provider who offers service. Taking eBay as an example, the features are mainly collected from the provider profiles including seller's system age, detailed connection information, actual age, gender, location, number of items sold already, reputation value, average delivery time.
2) About the service: the contextual information contains item price, category average price, comments, the number of items in stock, the number of buyers etc.
3) About the social relationships: these kinds are the features concerning the relationships between the service provider and others, such as family, colleague, friend, acquaintance ties, community, organization, trust networks and so on.

In order to accurately model the dynamics of a service provider, the dynamics must be captured precisely to represent the changes of providers. For instance, some online sellers may change their profiles or prices before committing a cheating action, which can lead to an essential caution for the buyers.

In this step, we extract features which have the probability to distinguish the service providers' behaviors. However, not all of them may be essential. The next step is to refine the extracted features.

B. Feature Selection

To reduce the computation time and eliminate the less powerful features, we choose the Mutual Entropy to select and maintain the K most powerful features or the ones over a threshold as our final observation in the HMM model.

Entropy usually has two views: the lower bound on the average number of bits to encode our feature values or the measure of the uncertainty about the feature values [9]. Here the latter meaning is more suitable to our case.

Features are usually dependent on the exact e-commerce, e-service or auction web site. The potential features extracted are represented as $\Omega = \{w_1, w_2, ...\}$, where $w_r \in \Omega$ corresponds to an exact feature. Suppose we are given the past transaction list of a seller $\Theta = \{\theta^1, \theta^2, ...\}$, and all the transaction outcomes/ratings coming from a set of discrete quantitative variables in a certain range denoted by $L = \{l_1, l_2, ..., l_l\}$ with the probability distribution $P = \{p_1, p_2, ..., p_l\}$. According to the definition of entropy, the entropy of a seller's all past transactions Θ is:

$$H_P(\Theta) = E_P[log\frac{1}{p_j}] = \sum_{j=1}^{l} P(\theta)log\frac{1}{p_j}$$

In the extremely simple situation, when all the feedback of a service provider's transaction history is positive, then according to the equation, the entropy of the feedback is 0. If there is 1/3 positive, 1/3 neutral and 1/3 negative feedback in this provider's history, the entropy is 1.585. So it is clear that entropy describes the distribution of data.

For each feature $w_r \in \Omega$, we use $\Upsilon(w_r)$ to denote its value set. For each value $v \in \Upsilon(w_r)$, we represent all the past transactions associated with v for feature w_r by Θ^v. Then the mutual information between Θ and Θ^v represents the extent to which the knowledge of Θ^v reduces our uncertainty about Θ, which is:

$$\begin{aligned} I_p(\Theta; w_r) &= H_p(\Theta) - H_p(\Theta|w_r) \\ &= H_p(\Theta) - \sum_{v \in \Upsilon(w_r)} \frac{|\Theta^v|}{|\Theta|} H_p(\Theta^v) \end{aligned}$$

The Mutual Entropy actually calculates a certain feature's effect to the probability distribution of the entire feature data. So it is clear that the higher the mutual information is, the lower the corresponding entropy becomes, and as a result, the better the classification is [11]. Then the top K features or the features above a certain threshold are selected according to mutual information.

C. Feature Combination

So far it is not guaranteed that the feature vectors are best presented in the current orthogonal coordinate system. In other words, the variance of the vectors along the axes is not maximized. If it is, we could select the principal components and reduce the vector dimension. So, in order to further reduce the computation time and maintain only the essential information, we apply Principal Component Analysis (PCA) [1] to simplify the features selected from mutual information.

Mathematically PCA is a procedure to determine an orthogonal transformation of the coordinate system to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables such that in the new coordinate system, the variance of the transformed data along the new axes has been maximized [1]. PCA performs the transformation using the statistical nature of the information so that the number of principal components is less than or equal to the number of original variables keeping the information of the data set lossless or only with a little loss.

After applying PCA, we can find the key components and structure of data and eliminate noise and redundancy resulting in the reduction of the number of dimensions by calculating eigenvalues and eigenvectors. For each transaction, we denote the key features as a vector: E_i. After including the contextual information, the rating l_i is absorbed into the key features. The optimized feature vector together with the rating element forms the full feature vector, denoted by $c_i = [E_i, l_i]$, describing each transaction's contextual information with little redundancy.

D. Feature Quantization

With the final feature vectors, we can either quantize them to discrete values and apply the Discrete Hidden Markov Model or directly apply continuous HMM. Due to the fact that some information is discrete and the consideration of computation time, here we choose DHMM.

Vector Quantization (VQ) is an area that has close affinity with clustering. This technique is mainly used for data compression which is a prerequisite for achieving better computer storage utilization and better bandwidth utilization in especially communications [17].

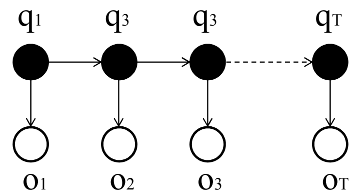

Figure 2. First Order Hidden Markov Chain

Let T be the set of all feature vectors for our prediction problem. We separate T into M distinct regions $\{R_j\}$ that exhaust T and represent each of them with a code vector v_j (here in our problem that is simplified as a value). The major goal in VQ is to define the regions $\{R_j\}$ and their representatives $\{v_j\}$ so that the information loss (called distortion) is minimized. The discrete representatives $\{v_j\}$, projected from the feature vector $\{c_j\}$, are the observations of DHMM.

IV. Markov Trust Model

Although statistical methods of Markov source or Hidden Markov Model was first introduced and studied in late 1960s, it is still popular in the pattern recognition area because of its richness in mathematical structure and good performance in practical applications [15]. It is very powerful for predicting the future trend based on sequential datasets. In this study, we modify HMM and make it suitable for e-commerce and e-service scenarios.

A. Basic knowledge of HMM

A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states which control the mixture components to be selected for each observation [15]. So, each HMM has a hidden state sequence from a finite state set and its corresponding observation sequence. The basic structure is shown in Figure 2. It also obeys the Markov chain hypothesis: each state is only determined by the previous one; the distribution of observations is only dependent on its corresponding state and independent with other observations; the number of states is finite [9].

Suppose in a sequential data we have T observations $O = \{o_1, o_2, ..., o_T\}$, $o_i \in V = \{v_1, v_2, ..., v_M\}$ and the states we infer from the observation are $Q = \{q_1, q_2, ..., q_T\}$, $q_i \in S = \{s_1, s_2, ..., s_N\}$, where M is the number of distinct observation symbols and N is the number of states in the model. Then the HMM can be represented as [15]:
$\lambda = (V, S, A, B, \pi)$.
A is the state transition probability distribution:
$A = \{a_{ij}\}_{N \times N}$, where $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$, $1 \leq i, j \leq N$, $1 \leq t \leq T$;
B is the emission probability distribution:
$B = \{b_j(k)\}_{N \times M}$, where $b_j(k) = P(v_k @ t | q_t = S_j)$, $1 \leq j \leq N$, $1 \leq k \leq M$, $1 \leq t \leq T$;
π is the initial state distribution:
$\pi = \{\pi_i\}_{N \times 1}$, where $\pi_i = P(q_1 = s_i), 1 \leq i \leq N$.

In general, there are three basic HMM problems of interest to be solved for the model to be useful in real-world applications described as follows:

- **Problem 1:** Given the observation sequence O and a model λ, how to efficiently compute the probability of the observation sequence $P(O|\lambda)$?
- **Problem 2:** Given the observation sequence O and a model λ, how to choose a corresponding state sequence Q?
- **Problem 3:** How to adjust the model parameters λ?

In the application for the calculation of trustworthiness, we only refer to the basic Problem 1 and Problem 3. Our goal is to calculate the probability: $P(O_{t+1} = v_i|O_{1:t}, \lambda)$ which means the probability of the case that the next observation will be v_i if we have already known the past observations from time 1 to time t.

In the following sections, we present, in more detail, how to predict the outcome by HMM in the most powerful way.

B. Our Proposed HMM based Approach

As stated before, the outcome-based HMM achieves better performance than the Beta model with a decay factor, and the Markov Model (MM) based on contextual information with visualized states is even better than the outcome-based HMM in accuracy and efficiency. We therefore propose an HMM algorithm based on both contextual information and outcomes to model the service providers' behaviors with hidden states.

To further explain the use of HMM, we give a simple example. A malicious provider honestly sold cheap but good-quality items in 80 transactions to accumulate good reputation and then started to deceive in the next 20 transactions. The history of this provider can be considered consisting of two states—honest and dishonest which are not visible. But when the provider was in the honest state in the first 80 transactions, we had very high probabilities (not 100% necessarily) to observe relative low prices, reputation rises and positive feedbacks. However, when the provider was in the dishonest state in the last 20 transactions, we were more likely to observe higher prices, reputation declines and negative feedbacks. Each transaction can be treated as one time point in HMM. The observations are the prices, reputation changes and feedbacks, all of which are determined by the hidden states (honest or dishonest). Therefore, the states are hidden but they can be predicted by the observations. Then the whole transaction history can be modelled by HMM.

Suppose that the past transactions of a service provider are denoted by $\Theta = \{\theta^1, \theta^2, ...\}$. We also assume that the outcomes are discrete quantities such as rating numbers from 1 to 5. Using the HMM described above and the same symbols to model the transaction list which describes the behavior of the service provider, the service provider's behavior can be denoted by $\lambda = (V, S, A, B, \pi)$ (or $\lambda = (A, B, \pi)$ for short), where there are N possible hidden states,denoted by S. The hidden states can be understood as the provider's working periods/states which determine the distribution of observations. M extracted features/observations are denoted by V. A and B are the transition probability matrix and the emission probability matrix respectively as described above. From Θ, we extract observation sequence $O = \{o_1, o_2, ..., o_T\}$ and train the model parameters λ with a finite number of hidden states which maximizes the expectation that the HMM λ could emit the observation sequence (the list of past transactions), where $O_i \in V$ and $q_i \in S$, $1 \leq i \leq T$. The observation and the state of the next transaction are denoted by O_{T+1} and q_{T+1} respectively.

To predict the next transaction behavior based on the past knowledge of a service provider, we need to calculate the probability distribution of the rating in the next transaction at time $T+1$ given the observation sequence for the past time $[1, T]$. In our model, a rating to a service provider is part of the observation. So, the probability of each possible observation in the forthcoming transaction can be computed by:

$$\begin{aligned} P(o_{T+1} = v_j|O_{1:T},\lambda) &= \frac{P(o_{T+1} = v_j, O_{1:T}, |\lambda)}{P(O_{1:T}, |\lambda)} \\ &= \frac{P(o_{T+1} = v_j, O_{1:T}|\lambda)}{\sum_{j=1}^{N} P(o_{T+1} = v_j, O_{1:T}, |\lambda)} \end{aligned} \tag{1}$$

The numerator is the joint probability that observations $O_{1:T}$ are observed in the first T transactions and at next transaction $T+1$ the feature v_j is observed as the next observation. The denominator represents the sum of all the possible o_{T+1} together with previous observations $O_{1:T}$ given the model λ.

Following the structure in Figure 2 to calculate the probability of the observation sequence $P(O|\lambda)$, the most straightforward way is to enumerate every possible state sequence of length T (the number of observations) [15].

$$\begin{aligned} P(O_{1:T}|\lambda) &= \sum_{allQ_{1:T}} P(O_{1:T}|Q_{1:T}, \lambda)P(Q_{1:T}|\lambda) \\ &= \sum_{q_1, q_2, ..., q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \\ &\quad ...a_{q_{T-1} q_T} b_{q_T}(o_T) \end{aligned} \tag{2}$$

where $Q_{1:T} = \{q_1, q_2, ..., q_T\}$ is a fixed state sequence which emits the observation sequence; $P(Q_{1:T}|\lambda)$ is the probability of the state sequence Q given the model λ; $P(O_{1:T}|Q_{1:T}, \lambda)$ is the probability of the observation sequence $O_{1:T}$ for the state sequence of $Q_{1:T}$ given the model λ.

According to Equation (2), the calculation of $P(O_{1:T}|\lambda)$ has a complexity of the order of $2T \times N^T$. Clearly a more efficient method is required. Fortunately, the Forward-Backward procedure [2] can solve the calculation problem.

The Forward algorithm could calculate $P(O_{1:T+1}|\lambda)$ avoiding an exponentially growing calculation. For $t = 1, 2, ..., T+1$ and $i, j = 1, 2, ..., N$, we define $\alpha_t(i) = P(O_{1:t}, q_t = s_i|\lambda)$, that is, the probability of the partial observation sequence $o_1, o_2, ..., o_t$ and state s_j at time t, given the model λ. The calculation of $\alpha_t(i)$ can be solved inductively as follows:

1) Initialization: $\alpha_1(i) = \pi_i B_i(1)$
2) Induction: $\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{i,j}] b_j(t)$
3) Termination: $P(O_{1:T}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

Then, we have $P(o_{T+1} = v_j, O_{1:T}|\lambda) = \sum_{i=1}^{N} \alpha_{T+1}(i)$

For the next transaction at time $T+1$, usually we have already known the contextual information so we can extract the feature vector E_{t+1} and limit the possible outcomes to a subset of the whole observation set $V' = \{v'_j\} \in V$, where the subset V' contains all the observations projected from the same feature vector together with different possible ratings. Finally, the most possible predicted outcome of the next transaction is the rating in the observation with the highest probability:

$$o_{T+1} = \arg\max_{v'_j \in V'} [P(o_{T+1} = v'_j, O_{1:T}|\lambda)].$$

C. HMM Training

Given the output sequences: $\lambda = \arg\max_{\lambda} P(O_{1:T}|\lambda)$, the training procedure of HMM is to find the best state transition probability and observation emission probability. The task is usually carried out by the Baum-Welch algorithm [22] using the Forward-Backward algorithm. The Baum-Welch algorithm is a particular case of the Expectation Maximization (EM) algorithm. It could reestimate the parameters of HMM given only emissions/observations as training data by maximizing Baum's auxiliary function. The details can be found in [15][22].

We first define the three variables:

- $\beta_t(i) = P(O_{t+1:T}|q_t = s_i, \lambda)$: the probability of the partial observation sequence from $t+1$ to the end, given state s_i at time t and the model λ;
- $\gamma_t(i) = P(q_t = s_i|O, \lambda)$: the probability of being in state s_i at time t, given the observation sequence O and the model λ; and
- $\xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j|O, \lambda)$: the probability of being in state s_i at time t and state s_j at time $t+1$.

Then the iterative reestimating procedure could be conducted as follows:

$$\bar{\pi} = \gamma_1(i);$$
$$\bar{a}_{ij} = \sum_{t=1}^{T-1} \xi_t(i,j) / \sum_{t=1}^{T-1} \gamma_t(i);$$
$$\bar{b}_j(k) = \sum_{t=1, s.t. o_t = v_k}^{T} \gamma_t(j) / \sum_{t=1}^{T} \gamma_t(j).$$

where $\bar{\pi}$ is the expected frequency in state s_i at time $t = 1$; $\bar{a}_{ij}$ is the ratio of the expected number of transitions from state s_i to state s_j to the expected number of transitions from state s_j; $\bar{b}_j(k)$ is the ratio of the expected number of times in state j and observation symbol v_k to the expected number of times in state j.

$\xi_t(i,j)$ and $\gamma_t(i)$ can be calculated by the Forward-Backward algorithm. We do not discuss the details here. Usually, there are lots of local minima and the data surface is complicated. The EM algorithm leads to local minima only. Fortunately, the local minima are usually adequate models of the data [15]. EM does not estimate the number of states. So, that must be given by experience or other algorithms such as standard gradient techniques. The initialization of HMM does affect performance but it can be optimized by other standard optimization algorithms. Usually, the reestimating procedure could be conducted several times to lessen the effect of random initialization.

V. Evaluation

A. Experimental Methodology

With the APIs released by eBay, we have developed a program using PHP to extract real datasets from the popular e-commerce website eBay (http://www.ebay.com.au/). The data we obtained from eBay contains the records of transactions of sellers within 3 months. In order to evaluate the performance of our modified model and to compare it with the state-of-the-art models, we generate synthetic datasets following the scenarios in the real data. Each dataset has 1000 sellers, each of which has finished 100 transactions for the same item.

For the complexity issue, we assume binary values for each property of each transaction and select only typical features to evaluate the approach. We treat any transaction with 5 points (highest points in eBay) positive feedback as successful transactions. Otherwise, we handle them as unsuccessful transaction records. We use the category ID to differentiate items. Within the same category, the features we used contain static features (item price, the time each transaction occurred, reputation and quantity, all of which are calculated as the percentage of the averages in the same category) and dynamic features (price changes, reputation changes). To investigate the performance of feature extraction, we set the time one transaction occurred as random noise.

From the analysis of the real datasets, we consider two scenarios of sellers' behaviors:

- **Scenario 1:** The sellers conducted a number of transactions with very good attitudes and quality obtaining successful transaction records. Then they become imprudent completing several unsuccessful forthcoming transactions.
- **Scenario 2:** The sellers change their behaviors randomly but most of the transactions are still successful.

We perform the feature extraction process as described in Section 3. Based on the extracted features, we compare our approach with the outcome-based HMM [6] and the contextual information-based Markov Model [11] on the performance of the rates of correct predictions.

B. Results

We compare the rates of correct predictions of our model with the outcome-based HMM (OHMM) [6] and the contextual information-based Markov Model (CIMM) [11] on the synthetic datasets in the above two scenarios.

Results of Scenario 1: The sellers all keep prudent at the beginning and then start to be imprudent. In this scenario,

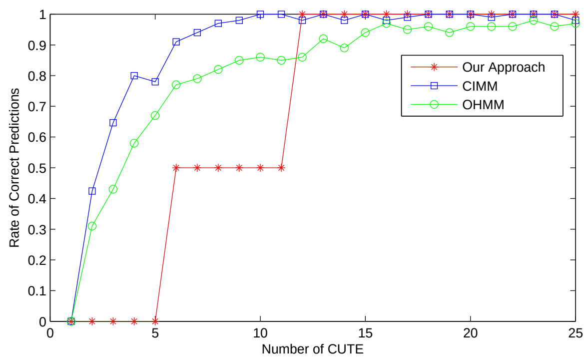

Figure 3. Rates of Correct Predictions in Scenario 1

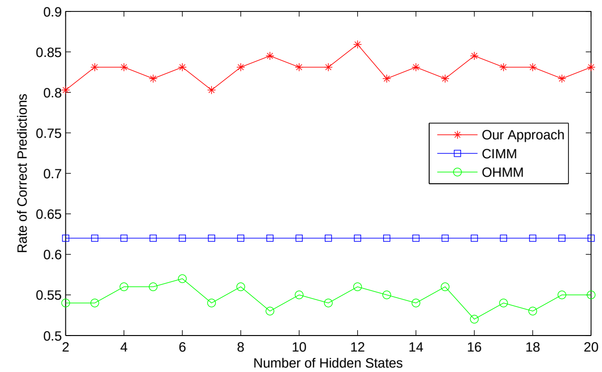

Figure 4. The Effect of Numbers of Hidden States in Scenario 2

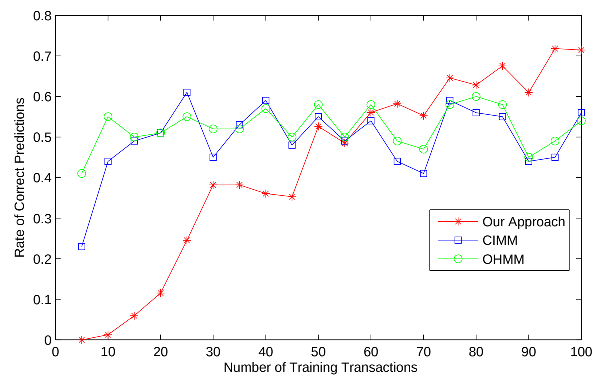

Figure 5. Comparison with Various Sizes of Training Data in Scenario 2

we mainly compare the performance of the three approaches on different percentages of unsuccessful transactions. We synthesize 25 groups of seller transaction datasets and each group has 1000 sellers. For each seller, there are 100 records of recent transactions. The difference between the groups is that the sellers in the first group have unsuccessful outcomes only on the last transaction, the sellers in the second group are unsuccessful on the last 2 transactions, and so on.

For each approach, we use the first 99 transactions to train the models and use the last transaction to evaluate the rates of correct predictions. From the experimental results, we notice that in the first group, none of the three approaches can predict correctly with a zero correct rate, because there is no unsuccessful transaction in the training data. As the percentage of unsuccessful transactions goes higher, there are more unsuccessful transactions in training data. As a result, the rates of correct predictions delivered by the three approaches all improve.

Figure 3 shows the performance of each approach with different numbers of continuous unsuccessful transactions in the end (CUTE). In the results of the first several groups, our approach is not as good as the other models. But the performance of all the three approaches improve and the correct rates of our approach go up to 100%, equal to or better than the others while the number of unsuccessful transactions accumulates to more than 13. This is because the number of observations is larger when treating both features and outcomes as observations and the data is relatively less dynamic in this scenario.

Results of Scenario 2: To compare the three approaches in Scenario 2, we also synthesize a dataset of 1000 sellers. Each seller maintains 100 historical records of transactions. But in this scenario, the ratings of sellers are randomly generated following the rule that most of transactions are successful. The distributions of contextual information at different ratings are different, which is the characteristic used to evaluate different prediction models. Figure 4 shows the performance of the three approaches with different hidden state numbers, where we use the first 99 transactions to train the three models and use the last transaction to examine the prediction result. Again, the performance of our approach and OHMM changes according to different numbers of hidden states while CIMM could not change this parameter which is also plotted in the figure for comparison.

From Figure 4, we can see that the performance of our approach and OHMM fluctuates with different number of hidden states. On average, our approach achieves 21% higher than CIMM and 27% higher than OHMM on the rates of correct predictions. Thus, our approach significantly outperforms both OHMM and CIMM approaches in this scenario.

In addition, we also compare the three approaches with different sizes of training data. That is, for each seller, we use first x percent of the previous transactions to train the three models separately and predict the next transaction. The results of experiments presented in Figure 5 demonstrate that the size of training data affects the prediction performance of all the three approaches. Our approach outperforms the others on the rates of correct predictions when the number of training transactions is more than 55 out of 100. As the number of training transactions goes higher, our approach becomes much better than the others. The gap reaches the peak which is 16% better than the others when 99 transactions are used as training data. This is because there are more observations in our approach.

VI. Conclusion

This paper presents a HMM based trust prediction approach which is different from outcome-based HMM and the contextual information-based Markov Model using outcomes as hidden states. Our approach utilizes both of the contextual information and the outcome of each transaction to build an HMM and predict the future result. The Information theories, PCA and QV help us extract effective features instead of directly using contextual information which significantly reduces calculation time, because these strategies preserve the essentials of the contextual information and reduce the noise in presenting the service provider's behaviors. The experimental results on synthetic datasets demonstrate that HMM based on both contextual information and outcomes needs more training data than the others but is more effective in predicting the future results of a service provider in complex dynamics.

In the future, we plan to mine semantic features, which are more concerned by prudent potential customers, from the feedback of customers. In addition, we want to make the model more sensitive to recent records of transactions by adding a forgetting factor into the HMM-based approaches and utilize other optimization algorithms to minimize the impact of HMM's initialization on performance.

References

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2:433–459, 2010.

[2] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Am. Math. Soc.*, 73:360–363, 1967.

[3] H. Billhardt, R. Hermoso, S. Ossowski, and R. Centeno. Trust-based service provider selection in open environments. In *SAC*, pages 1375–1380, 2007.

[4] S. Dorri Nogoorani and R. Jalili. Uncertainty in probabilistic trust models. In *IEEE 26th International Conference on Advanced Information Networking and Applications (AINA)*, pages 511 –517, march 2012.

[5] B. Elizabeth, R. Aaishwarya, P. Kiruthika, M. Shrada, A. Prakash, and V. Uthariaraj. Bayesian based confidence model for trust inference in MANETs. In *International Conference on Recent Trends in Information Technology (ICRTIT), 2011*, pages 402 –406, june 2011.

[6] E. ElSalamouny, V. Sassone, and M. Nielsen. Hmm-based trust model. In *International Workshop on Formal Aspects in Security and Trust (FAST2009)*, volume 5983, pages 21–35. LNCS, Springer, July 2009.

[7] J. Golbeck and J. A. Hendler. Inferring binary trust relationships in web-based social networks. *ACM Trans. Internet Techn.*, 6(4):497–529, 2006.

[8] A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, Jun 2002.

[9] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[10] L. Li and Y. Wang. A trust vector approach to service-oriented applications. In *IEEE International Conference on Web Services, 2008. ICWS'08*, pages 270–277. IEEE, 2008.

[11] X. Liu and A. Datta. Modeling context aware dynamic trust using hidden markov model. In *AAAI*, pages 1938–1944, 2012.

[12] Z. Malik, I. Akbar, and A. Bouguettaya. Web services reputation assessment using a hidden markov model. In L. Baresi, C.-H. Chi, and J. Suzuki, editors, *Service-Oriented Computing*, volume 5900 of *Lecture Notes in Computer Science*, pages 576–591. Springer Berlin Heidelberg, 2009.

[13] R. Marchany and J. Tront. E-commerce security issues. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002. HICSS.*, pages 2500 – 2508, Jan. 2002.

[14] M. E. G. Moe, B. E. Helvik, and S. J. Knapskog. TSR: trust-based secure manet routing using hmms. In *Q2SWinet*, pages 83–90, 2008.

[15] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257 –286, Feb 1989.

[16] J. Riegelsberger, M. A. Sasse, and J. D. Mccarthy. Shiny happy people building trust? photos on e-commerce websites and consumer trust. In *Proceedings of CHI 2003*, pages 121–128. ACM Press, 2003.

[17] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras. *Introduction to Pattern Recognition: A Matlab Approach*. Academic Press, 2010.

[18] X. Wang, Y. Chen, and B. Xu. Trust service selection in pervasive computing. In *International Conference on Multimedia Information Networking and Security, 2009. MINES '09.*, volume 1, pages 173 –176, Nov. 2009.

[19] Y. Wang and E. P. Lim. The evaluation of situational transaction trust in e-service environments. In *IEEE International Conference on e-Business Engineering, 2008. ICEBE'08*, pages 265–272. IEEE, 2008.

[20] Y. Wang and F. Lin. Trust and risk evaluation of transactions with different amounts in peer-to-peer e-commerce environments. In *IEEE International Conference on e-Business Engineering, 2006. ICEBE'06*, pages 102–109. IEEE, 2006.

[21] Y. Wang and K. J. Lin. Reputation-oriented trustworthy computing in e-commerce environments. *Internet Computing, IEEE*, 12(4):55–59, 2008.

[22] L. R. Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4), Dec. 2003.

[23] H. Zhang, Y. Wang, and X. Zhang. Transaction similarity-based contextual trust evaluation in e-commerce and e-service environments. In *2011 IEEE International Conference on Web Services (ICWS)*, pages 500–507, 2011.

[24] H. Zhang, Y. Wang, and X. Zhang. Efficient contextual transaction trust computation in e-commerce environments. In *TrustCom*, pages 318–325, 2012.

[25] H. Zhang, Y. Wang, and X. Zhang. A trust vector approach to transaction context-aware trust evaluation in e-commerce and e-service environments. In *SOCA*, pages 1–8, 2012.

[26] J. Zhang and R. Cohen. Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications*, 7(3):330–340, 2008.