Trust Modelling in Dynamic Environments

Yue Xia, Alexandre Parmentier, and Robin Cohen

Computer Science; University of Waterloo; Waterloo, Canada
{yue.xia, aparment, rcohen}@uwaterloo.ca

Abstract. Recent work has shown that Hidden Markov Models (HMM) are an effective formalism for modelling dynamic multi-agent trust systems. A significant application is modeling the trustworthiness of vendors in e-marketplaces. In this paper we survey two HMM trust models which approach this problem from significantly different angles. Then, we propose a new system which incorporates design choices from both. Our system has a number of improvements over the existing models: it incorporates trust modeling and context into its predictions, models user feedback variability as a guard against coordinated cheating efforts, and makes progress towards solving the problem of data sparsity in existing models.

1 Introduction

The open Internet has made it possible for nearly anyone to participate in growing online marketplaces. The size and accessibility of these marketplaces has created a challenge for users and marketplace administrators: how can honest users maximize their chances of dealing with reputable and honest vendors? Trust management is a popular way to provide decision support by indicating the trustworthiness of interaction partners.

Many approaches have been proposed to estimate trust based on a target agent’s past behavior. For example, one can use buyers’ average ratings of a seller’s past transactions [1, 2] or a beta probability density function [3] to measure trustworthiness. However, these approaches assume a relatively static agent behavior, and thus are not really focused on modelling an agent’s dynamic trustworthiness very well (e.g. honesty changing over time). One step forward is to adapt a forgetting factor [3] to provide less weight to old transactions. But the forgetting factor needs to be chosen carefully in order for the model to both accurately evaluate the seller’s trustworthiness and react fast enough when the seller’s behaviour changes. A more serious problem with those approaches is that they can be easily exploited by malicious sellers. For instance, a smart cheater may act honestly in selling many cheap items to have a high system rating and then cheat in selling an expensive item [15].

In this paper we review approaches to this problem using Hidden Markov Models and propose a new hybrid solution. Our proposed solution incorporates both seller trust modeling and contextual queues in order to recommend sellers. Our aim is to offer a high degree of flexibility and to take steps towards solving the issue of data sparsity that current systems struggle with.
2 Existing Work

In this section, we briefly review the parameters of a Hidden Markov Model and outline two approaches to trust modeling in an e-marketplace using this formalism.

2.1 Hidden Markov Models and Trust Modeling

A Hidden Markov Model can be characterized by a finite set of hidden states, \( S = \{s_1, ..., s_N\} \), a set of transition probabilities between states, \( A = \{a_{11}, a_{12}, ..., a_{nn}\} \), where \( a_{ij} \) is the probability of transitioning from state \( i \) to state \( j \), a set of possible observations, \( V = \{v_1, ..., v_T\} \), an emission probability matrix, \( B = \{b_{11}, b_{12}, ..., b_{nT}\} \), where \( b_{it} \) is the probability of observing \( v_t \) when in state \( s_i \), and an initial probability distribution over states \( \pi(s_i) \). Time is typically considered to advance in discrete steps, with the hidden state and observation at time \( k \) denoted by \( x_k \) and \( y_k \) respectively. For emphasis, note that when modeling a process in this way we only have access to the observations: the underlying state remains hidden. When modeling some process with an HMM, typical tasks include: determining the sequence of hidden states which would be most likely to produce a sequence of observations and learning the values of \( A \) and \( B \) which make the observed sequence most likely. This basic formulation can be extended in numerous ways, including by considering state transition matrices which change with time and removing the restriction on modeling time discretely. See [6] for a more extensive introduction to HMMs.

HMMs are convenient for modeling sequences of interactions between agents where trust is a factor. Two reasons for this are: 1) HMMs are computationally lightweight, making it feasible to represent many relationships between agents with HMMs and 2) the Markovian assumption allows for quick reactions when trust is broken, as the model will rapidly adjust to the new state of the world.

2.2 A Review of Two Models

In [4], Moe et al. propose a continuous time hidden Markov model be used to model the relationship between each buyer and seller in a marketplace. In this formulation, the hidden states of the model correspond to the trustworthiness of the seller. (e.g. there might be three states: not-trustworthy, neutral, trustworthy). The set of possible observations corresponds to the rating that a buyer gives to a transaction once it is completed (e.g. 1-5 stars). The state transition matrix (i.e. the probability that a seller will modify their behaviour from one state of trustworthiness to another) and emission matrix (i.e. the probability that a seller will receive a certain rating from a user given their trustworthiness) are learned.

---

1 The details of continuous time HMMs are not necessary to explain to broadly understand this model (summarized in Appendix I). What is key is that the values of the state transition matrix are assumed to be a function of time that has passed between samplings (e.g. change of state may be more likely if time has passed).
offline on a historical data set using the Baum-Welch algorithm [6]. Once these parameters are learned, the model can be applied to a live e-marketplace and set to the task of estimating seller trustworthiness based on new transactions and ratings that are created. This estimation task is very efficient, as it corresponds to calculating the most likely state of each buyer-seller HMM, which can be efficiently re-estimated after every transaction using the forward algorithm [4]. These estimates of seller trustworthiness are then used as inputs to a reinforcement learning algorithm for recommending buyer policies.

We have two main critiques of this model. First, while the model can quickly react to changes in seller behaviour, it only incorporates minimal information from the environment into its predictions (i.e. the ratings users submit). Complex fraud strategies (smart cheaters) may not be easily detectable, given such a small set of information [15]. Second, this model is defined over continuous time, where the odds of seller trustworthiness transitions are a function of time, but the difference between time intervals may not be the deciding factor. For instance, a seller who acts honestly out of moral considerations is unlikely to become more likely of fraud over time. Conversely, a seller who is implementing a complex strategy of fraud may be altering their behaviour in response to queues such as changes in market conditions or the demographic makeup of a buyer, but it is unclear how these queues map onto time. For reputation lag (building reputation and then cheating), the transition may also be independent of time intervals.

This model is substantially improved upon in [5]. Here, Liu and Datta propose a discrete HMM be used to model the relationship of every seller-buyer pair. Rather than model seller trustworthiness as the underlying set of hidden states, they treat the final rating of transactions by buyers as a state. The vocabulary of observations is drawn from the context of transactions - for example, the type of item being sold, the price of the sale, etc. By treating transaction outcome as hidden state and transaction context as observable, the task of predicting the most likely state given a sequence of observations is equivalent to trying to predict a buyer’s rating of a transaction before it is completed.

The parameters of this model are significantly easier to calculate than in Original HMM, as the state transition probabilities (e.g. the odds of a low rated transaction following a high rated transaction) and the emission probabilities (e.g. the odds of a highly rated transaction occurring in the context where the item being sold is a laptop and its price is > 1000$) can be directly calculated from historical data via MLE [5].

Liu and Datta tested this solution on simulations of an e-marketplace where sellers were engaged in a number of complex cheating strategies. These strategies, as described in [11] and [12], are designed to be challenging for trust modeling systems to detect. Selling agents could either (a) behave honestly in several transactions and then cheat (b) change their behavior half way through the simulation (from honesty to dishonesty); (c) change behavior randomly. Their solution, which models the context of transactions, performed significantly better than the solution described in [4], with improvements on both the false positive and false negative rates (for detecting cheating behaviour).
We take these results as an indication that context is indeed an important indicator of transaction outcome, and may be helpful for detecting complex cheating patterns. Further, we note that the adoption of a simpler discrete time model still leads to an increase in model performance, perhaps indicating that modeling transaction context lessens the need for a continuous time model.

A final advantage of the approach in [5] is that it directly estimates the most likely rating of a transaction by a buyer before the transaction is completed. Therefore, it is not necessary to process the results of the model any further in order to recommend a policy for a buyer (they can simply choose the seller who maximizes expected transaction outcome). A slightly awkward feature of this model is that, by modeling transaction outcome as state and context as observation, the model implies that the (future) outcome of a transaction is the basis for the context of the transaction. This is obviously backwards, yet this formulation is mathematically convenient and still reasonable so long as transaction outcome correlates with context.

Before proceeding to our proposed solution, we note two important factors of this problem that neither model has addressed. The first is that while both models rely on buyer ratings as an input, neither models the variability in user rating behaviour. For instance, some users submit consistently high ratings, while other are consistently negative. Without a user model for buyers which captures this variability, all ratings are given the same weighting. This issue is sidestepped by giving each buyer and seller pair a unique HMM. This leads to the second issue: constructing an HMM to model every buyer-seller relationship will naturally lead to sparsity issues (insufficient basis for trust modeling), as most buyers will have a small or non-existent transaction history with most buyers. In our model, we will attempt to address both these issues by constructing a single HMM for each seller and implementing a model of user rating variability.

3 Proposed Methods

The performance of the HMM in [5] has confirmed the importance of contextual information. Thus we will also use the contextual information in our proposed method. However, a great disadvantage of the approach in [5] is that the model parameters cannot be accurately learned from too few transactions. This situation can happen extremely often because a model only considers transactions between a seller and a fixed buyer. An important factor that causes the restriction is that the states in the HMM are the buyer’s ratings. For this reason, the ratings must always correctly measure the seller’s trustworthiness, which may be true if they all come from the same buyer (they now measure the seller’s trustworthiness from the specific buyer’s perspective). A possible solution is to restore the structure of the HMM in [4] such that states are again the seller’s trustworthiness. Then to reflect the difference among buyers, each buyer is associated with a distinct rating observation matrix which determines how they rate a transaction. Then a model can flexibly consider transactions with various buyers.
3.1 Modified Context Aware HMM

**Trust Modelling.** Our proposed model combines properties of the two models described above into a novel discrete time\(^2\) HMM. We model the trustworthiness of sellers as hidden states, and transaction outcomes (e.g. buyer ratings of seller behaviour) as observations. In addition, we model the context of transactions, and the tendency for individual users to rate transactions more or less harshly. Finally, our model combines the feedback of all buyers into a single model of the trustworthiness for each seller, rather than modeling the individual relationship between every pair of buyer and seller.

The next step is to decide where to put the contextual information. Clearly, the seller’s behavior (his trustworthiness in a transaction) depends on the context of the transaction. But as described in [5], this dependency would introduce a large amount of probabilities to be learned, as the contextual information may contain a huge dimension of features. Thus, the trustworthiness with a smaller dimension is set to be the cause and we may assume the conditional independence of contextual features. So the context is used as another observation in the HMM beside the buyer’s rating. Thus the hidden states are the seller’s trustworthiness and the observations are the buyer’s ratings plus the contexts of the transactions.

We then define the following notations for our HMM: \(\mathbf{S} = \{s_1, s_2, \ldots\}\) is the set of all possible hidden states, each value \(s_i\) can correspond to a level of trustworthiness. \(\mathbf{L} = \{l_1, l_2, \ldots\}\) is the set of all possible ratings; \(\mathbf{\Omega}\) is the set of all possible feature representation of the contextual information. Here the contexts are represented in the same way as in [5]. The same feature selection process is applied; \(T_k^u\) denotes the seller \(u\)’s \(k\)-th transaction; \(x_T \in \mathbf{S}\) is the state of seller at his transaction \(T\); \(y_T \in \mathbf{L}\) is the buyer’s rating of his transaction \(T\); \(F_T \in \mathbf{\Omega}\) is the contextual feature representation associated with the transaction \(T\).

The previous two HMM based methods model the relationships between every buyer-seller pair. However, the prediction would be very inaccurate when there are few transactions between a buyer and seller. Our proposed method tackles this problem by constructing a model for each seller and the transactions in the Markov chain can be between the seller and any buyer. In our model, each seller \(u\) is associated with a state transition matrix \(P^u = (p^u_{ij})\) with \(p^u_{ij} = P(x_{T_{u,+1}} = s_j | x_{T_u} = s_i)\), the conditional probability distribution of the seller’s state in his next transaction \(T_{u,+1}\) given his state in the current transaction \(T_u\); and a context observation matrix \(C^u = (c^u_{ij}(F))\). Then \(c^u_{ij}(F) = P(F_T = F | x_T = s_j)\) indicates the probability that the contextual feature representation of transaction \(T\) is \(F\) given that the seller \(u\) of the transaction is at state \(s_j\). For example, \(c^u_{i1}(\{\text{item} = "laptop", \text{price} = 500\})\) is the probability that seller \(u\) sells a laptop at price $500 given that his trustworthiness in this transaction is \(s_1\).

Different buyers are assigned with different conditional probability distribution of ratings given the seller’s behavior (to account for varying buyer preferences/standards). Let \(B^v = (b^v_j(y))\) denote the rating observation matrix associated with the buyer \(v\), \(b^v_j(y) = P(y_T = y | x_T = s_j)\) is the probability that buyer

---

\(^2\) Each time stamp corresponds to one transaction; one transition between transactions.
\( v \) gives rating \( y \) to transaction \( T \), given that the seller of the transaction is at state \( s_j \). So for different buyers \( v_1 \) and \( v_2 \), the rating observation matrices \( B^{v_1} \), \( B^{v_2} \) may be different. In summary, the context observation matrix is associated with the seller’s decision whether or not to cheat based on the context of a transaction (item price, easiness of cheating, etc.), while the rating observation matrix is based on a buyer’s behaviour when rating transactions.

![Fig. 1: The proposed Markov model.](image)

Figure 1 shows that the seller \( u \)'s state transitions after every his transaction according to his state transition matrix \( P^u \). Those transactions can be with different buyers. The distribution of context at each transaction given the state is determined by his context observation matrix \( C^u \). However, the buyers’ ratings are associated with the buyers’ rating observation matrices \( B^{v_i} \). For transactions with different buyers, different rating observation matrices will be used. For the first transaction \( T_1 \), it is with buyer \( v_1 \) and the rating is determined by \( v_1 \)'s rating observation matrix \( B^{v_1} \). The second transaction is associated with another buyer \( v_2 \), and \( v_2 \)'s matrix \( B^{v_2} \) is used.

**Updating State Probability Distribution.** Like in [4], the state probability distribution is updated after each transaction completes. Let \( \gamma_T = (\gamma_T(i)) \) denote the state probability distribution at transaction \( T \) given all observations received until when transaction \( T \) happens, \( \gamma_T(i) = P(x_T = s_i|y_T, F_T) \) where \( y_T = y_{T_1}, y_{T_2}, ..., y_{T_k} \) and \( F_T = F_{T_1}, F_{T_2}, ..., F_{T_k} \) for \( T = T_k \). Algorithm 1 updates the state probability distribution \( \gamma \) after the completion of transaction \( T = T_k \) between seller \( u \) and buyer \( v \) based on the following inputs: \( k \) the index, \( \gamma = \gamma_{T_{k-1}} \) the previous state distribution, \( y = y_T \) the rating of the current transaction and \( F = F_T \) the feature representation of the current transaction’s contextual information. In addition to the dynamic variables listed above, the algorithm requires the following model parameters: the initial state distribution \( \pi \), the transition matrix \( P^u \), and the two observation matrices \( C^u \) and \( B^v \).

Different from the normal forward algorithm, Algorithm 1 uses two observation variables: the buyer’s rating \( y \) and the context \( F \). For transactions between the seller and different buyers, different rating observation matrices \( B^v \) are used. \( \alpha(i) \) in the algorithm represents the probability \( P(y_T, F_T, x_T = s_i) \). It is also known as the forward variable. By storing \( \alpha(i) \), This algorithm can achieve \( O(N^2) \) time for an update.
backward variable $\beta$ transactions with different buyers. The modified Baum-Welch algorithm uses formation of the transactions. And the rating observation matrix is different for cationally re-estimated after a fixed number of transactions.

\[ P \]

\[ Require: \ k, \gamma, y, F \]

if $k = 1$ then
  for $i = 1$ to $N$ do
    $\alpha(i) \leftarrow b_i^u(y)c_i^u(F)\pi_i$
    $\gamma(i) \leftarrow \sum_{j=1}^{N} b_j^u(y)c_j^u(F)s_j$
  end for
else
  for $i = 1$ to $N$ do
    $\alpha(i) \leftarrow b_i^u(y)c_i^u(F)\sum_{j=1}^{N} \gamma(j)p_{ji}^u$
  end for
  for $i = 1$ to $N$ do
    $\gamma(i) \leftarrow \frac{\alpha(i)}{\sum_{j=1}^{N} \alpha(j)}$
  end for
end if
return $\gamma$

\[ Algorithm 1 \]

\[ Algorithm 2 \]

Re-estimating Model Parameters. Similar to the approach in [4], the model parameters $P^u$, $C^u$ and $B^u$ are default values in the beginning, but are periodically re-estimated after a fixed number of transactions.

Let’s consider seller $u$. The model parameters associated with $u$ are learned from all their past transactions from $T_1^u$ to $T_K^u$. The Baum-Welch algorithm is also used. But now we have an additional type of observation, the contextual information of the transactions. And the rating observation matrix is different for transactions with different buyers. The modified Baum-Welch algorithm uses backward variable $\beta_k(i) = P(y_{T_k+1}^u, ..., y_{T_T}^u, F_{T_{k+1}}^u, ..., F_{T_T}^u)$. It is calculated by setting $\beta_K(i) = 1$ and then recursively calculating $\beta_k(i) = \sum_{j=1}^{N} P_{ji}^u b_j^u(y_{T_{k+1}}^u)c_j^u(F_{T_{k+1}}^u)\beta_{k+1}(j)$, where $v$ is the buyer in transaction $T_k^u$. Then the joint probability of being in state $s_i$ at time $k$, and state $s_j$ at $k+1$ given all observation: $\xi_k^u(i,j) = P(x_{T_k+1}^u = s_i, x_{T_{k+1}}^u = s_j, y, F)$ is calculated by:

\[\xi_k^u(i,j) = \frac{\alpha_k(i)P_{ji}^u b_j^u(y_{T_{k+1}}^u)c_j^u(F_{T_{k+1}}^u)\beta_{k+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_k(i)P_{ji}^u b_j^u(y_{T_{k+1}}^u)c_j^u(F_{T_{k+1}}^u)\beta_{k+1}(j)}\]  (1)

where $v$ is the buyer in transaction $T_{k+1}^u$. The formula represents probability of seeing state $i$ at time $k$, state $j$ at $k+1$, and all observations divided by a normalization factor. Then according to Baum-Welch[6], the state transition probability is calculated similarly by:

\[ p_{ij}^u = \frac{\sum_{k=1}^{K-1} \xi_k^u(i,j)}{\sum_{k=1}^{K-1} \gamma_k^u(i,j)}\]  (2)
And the context observation probability associated with this seller is:

\[
c_j^u(F) = \frac{\sum_{k=1}^K \gamma_{k,T_u^j} \gamma_{k,T_x^j}(j)}{\sum_{k=1}^K \gamma_{k,T_x^j}(j)}
\] (3)

Algorithm 2 implements those calculations. It outputs the transition matrix and context observation matrix associated with a seller \( u \). It takes as input all the \( K \) different \( \alpha_k \) and \( \gamma_T \) vectors, which are calculated by Algorithm 1, as well as the model parameters \( P \) and \( F \) to be re-estimated. It also requires the set \( \{ B^v \} \) consisting of the rating observation matrices of all buyers who had transactions with the seller.

The re-estimation of \( C \) in Algorithm 2 would be very inefficient since we compute a probability for every possible feature representation \( \omega \in \Omega \). And the estimation of those probabilities would be inaccurate given limited number of examples. Thus we may assume conditional independence of different features in the feature representation. i.e. let \( \Omega = (\Omega_1, \Omega_2, ... , \Omega_m) \) be a collection of ordered features and \( \Omega_i \) corresponds to the \( i \)-th feature, then \( \Omega_i \mid x_T \) is independent of \( \Omega_j \mid x_T \) for \( i \neq j \). Let \( \text{values}(\Omega_i) \) denote the set of all possible values of feature \( \Omega_i \) and let \( F_i^T \in \Omega_i \) denote the value of the \( i \)-th feature in the feature representation of transaction \( T \). Then the re-estimation of \( C \) can be calculated by Algorithm 3 if we assume that the features are conditionally independent. Note that this assumption may not be valid but can greatly simplify the model.

**Algorithm 3** Re-estimation of \( C^u \)

```plaintext
for i = 1 to m do
  for all \( \omega_i \in \text{values}(\Omega_i) \) do
    for j = 1 to N do
      \( c_{j,i}^u(\omega_i) \leftarrow \frac{\sum_{k=1}^K \gamma_{k,T_u^j} \gamma_{k,T_x^j}(j)}{\sum_{k=1}^K \gamma_{k,T_x^j}(j)} \)
    end for
  end for
end for
return \( C \)
```

\( c_{j,i}^u(\omega_i) \) is the probability that the \( i \)-th feature of the transaction context is \( \omega_i \) given that the seller \( u \) is at state \( s_j \), i.e. \( c_{j,i}^u(\omega_i) = P(F_i^T = \omega_i \mid x_T = s_j) \). \( c_{j,i}^u(\omega_i) \) will be stored instead of \( c_j^u(\omega) \). For any feature representation \( \omega = (\omega_1, ... , \omega_m) \), \( c_j^u(\omega) \) can be calculated from \( c_{j,i}^u(\omega_i) \):

\[
c_j^u(\omega) = \prod_{i=1}^m c_{j,i}^u(\omega_i)
\] (4)

as \( P(F_1^T, ... , F_m^T \mid x_T) = \prod_{i=1}^m P(F_i^T \mid x_T) \) when \( F_i^T \)'s are conditionally independent.

After updating the model parameters associated with the sellers, the buyers’ rating observation matrices also need to be re-estimated. The calculations are similar to the re-estimation of \( C \) and they are implemented by Algorithm 4.

**Algorithm 4** Re-estimation of model parameters for buyer \( v \)

```plaintext
for l \in L do
  for j = 1 to N do
    \( b_j^v(l) \leftarrow \frac{\sum_{\text{all transaction } T \text{ of } v \mid s_T = l} \gamma_{T_x^j}(j)}{\sum_{\text{all transaction } T \text{ of } v} \gamma_{T_x^j}(j)} \)
  end for
end for
return \( B \)
```
Buyer’s Policy. Similar to the approach in [5], the model would output the predicted buyer’s rating as a reference for the buyer to decide whether to make the transaction. And the predicted rating would be from the specific buyer’s perspective who will make the transaction. Thus the buyer can directly use the output in their decision making without considering how his personal preferences are different from others’. For a target future transaction $T = T_{k+1}$ between seller $u$ and buyer $v$, we will first predict the seller’s state in the transaction based on the current state distribution of the seller $\gamma = \gamma_{T_k}$, and the context of that future transaction $F = F_T$. The seller’s state probability distribution in the future transaction $T$ given all observations $\gamma'(i) = P(x_{T_{k+1}} = s_i|y_{T_k}, F_{T_{k+1}})$ would be calculated by:

$$\gamma'(i) = \frac{c^u_i(F) \sum_{j=1}^{N} \gamma(j)p_{ij}^u}{\sum_{k=1}^{N} c^u_k(F) \sum_{j=1}^{N} \gamma(j)p_{jk}^u}$$ (5)

Then the buyer’s rating can be predicted from $\gamma'(i)$ using the buyer’s rating observation matrix. The probability that the buyer $v$ would give a rating $y$ is:

$$P(y_{T_{k+1}} = y|y_{T_k}, F_{T_{k+1}}) = \frac{\sum_{i=1}^{N} \gamma'(i)b_v^i(y)}{\sum_{l \in L} \sum_{i=1}^{N} \gamma'(i)b_v^i(l)}$$ (6)

With the predicted distribution of rating, the buyer $v$ can decide whether to make the transaction.

Conclusion. The proposed approach also integrates the contextual information associated with transactions into the HMM. And we expect the proposed method to work well and be able to identify complex patterns (like the smart cheater’s) with the use of contextual information. However, our HMM has a different structure from that in [5]. Instead, it is more similar to that in [4]. The dynamic trustworthiness of the seller is preserved in the model as the hidden states, while the observations are the contextual information as well as the buyer’s ratings. It remains to be tested whether this configuration is better than that in [4]. However, this structure allows the construction of a model for each seller based on all the seller’s transactions with different buyers. Thus an HMM can be trained with more transactions and become more accurate. By focusing on each seller and its transactions (and not modelling all seller-buyer pairs), we overcome data sparsity. In terms of performance, the update of a seller’s state distribution and the prediction of buyer’s rating are very efficient and can be done on-line. However, since the states at past transactions are hidden, the re-estimation of model parameters requires the Baum-Welch algorithm and can be computationally intensive. Though the proposed method uses more complex HMMs, it is expected to have fewer total model parameters than the HMM in [5]. Both reviewed models build an HMM for every pair of seller and buyer. This requires a transition matrix and an observation matrix for each seller-buyer pair. Our approach constructs only one model for each seller. Thus those two matrices to be learned are associated with sellers whose total number is much
smaller than that of seller-buyer pairs. The proposed method also requires the rating observation matrices associated with each buyer. But their number and dimension are small compared with the other two matrices. One concern with the proposed method is the estimation of rating observation matrices associated with each buyer. Though sellers’ models may be trained with sufficient number of transactions, a buyer may not have made enough transactions to accurately estimate his rating observation matrix.

### 3.2 Variation: Clustering Buyers

A problem with the proposed method is that a buyer may not have made enough transactions to accurately estimate his rating observation matrix in Algorithm 4. A possible solution is to cluster the buyers and let the buyers in the same cluster share the same rating observation matrix learned from transactions of all buyers in the cluster. In this way, the rating observation matrix may be calculated with sufficient number of transactions. The clustering of buyers can be done using existing clustering algorithm, e.g. k-means[13]. They can be clustered based on the features extracted from their profile. For instance, the buyer’s age, gender, location, system age, number of items purchased in each category, average rating, etc. The buyers should be clustered such that (1) the buyers in the same cluster should give similar ratings on the same transaction; (2) each cluster contains sufficient number of transactions to estimate its rating observation matrix. The first condition is required because the buyer’s individual observation matrix is approximated by the cluster’s observation matrix in the calculations. And thus we want the cluster’s observation matrix to be close to the observation matrices of all buyers represented by that cluster. So the cluster’s rating observation matrix should be learned from buyers who will give similar ratings. In order to satisfy this condition, the buyers’ features as well as their weights should be chosen very carefully. An important feature may be the buyer’s average rating. But experiments should be conducted to find the best feature representation. (1) can be easily satisfied when all buyers are assigned to different clusters. But we also want the cluster parameters to be estimated accurately with sufficient number of transactions. Thus condition (2) is required. It can be achieved by testing different $k$, the number of clusters. However, the optimal size of a cluster remains to be determined, we may need to tune the clustering method to maximize the HMM’s prediction accuracy of buyers’ ratings.

### 4 Conclusion

In this paper, we compare two different existing ways to model an agent’s dynamic trustworthiness (one an HMM [4] and another an HMM using contextual information [5]) and propose an approach based on both these methods. It allows a model to be trained from transactions between a seller and different buyers while using the contextual information. A variation of the proposed method that clusters the buyers is also suggested. But the detailed optimization steps are not
covered in this paper. Regarding future works, testings are required to identify the weaknesses of the proposed method. And its variation can be further explored by conducting various experiments. Those experiments can test different clustering algorithms, feature selections, etc. More complicated situations are not considered in the existing and proposed methods, for example, the trustworthiness of the evaluator. Some research has been done on those topics[14] and they can be combined with the proposed method.

References

Appendix I: the HMM model in [4]

Trust Modelling. A continuous time hidden Markov model (HMM) is built for each pair of seller and buyer. The HMM consists of a finite set of \( N \) hidden states \( S = \{s_1, \ldots, s_N\} \) which represent different levels of trustworthiness of the seller. The state of the monitored seller at his \( k \)-th transaction is represented by \( x_k \in S \). \( P_k = \{p_{ij}^k\} \) is the set of state transition probabilities, \( p_{ij}^k = P(x_{k+1} = s_j|x_k = s_i) \), where \( x_k \) is the current state of the system. \( \pi = \{\pi_i\} \) is the initial state distribution, where \( \pi_i = P(x_1 = s_i) \). The buyer’s ratings are classified into a set of observation symbols \( V = \{v_1, \ldots, v_M\} \). Let \( y_k \in V \) denote the observation made at sampling instant \( k \). The HMM consists of two stochastic processes: the hidden process \( x_k \), and the observable process \( y_k \) that depends on \( x_k \). The relation between \( x_k \) and \( y_k \) is described by the probability distribution matrix \( B = b_j(m) \), where \( b_j(m) = P(y_k = v_m|x_k = s_j) \). See [6] for a more extensive introduction to HMMs. In this HMM modelling the seller’s trustworthiness, the system could make zero, one or more transitions during the time between two successive transactions. Thus the transition probabilities \( p_{ij}^k \) from the \( k \)-th transaction to the next transaction depend on the time between two transactions. By Kolmogorov’s equations[7], a transition rate matrix \( \Lambda \) is defined and the transition probability matrix \( P_k \) can be calculated from \( \Lambda \) and the time between transactions.

Updating State Distribution and Re-estimating Model Parameters. The probability distribution of the hidden state, the seller’s trustworthiness, is initially set to \( \pi \) and is updated using the forward algorithm[4] every time when a new transaction is observed. This algorithm is very efficient. A similar version of the forward algorithm is explained in detail in our proposed method. The forward algorithm requires two model parameters: the transition rate matrix \( \Lambda \) and the observation matrix \( B \). When a buyer encounters a seller for the first time, those parameters are set to default values, and the model might not properly predict the system dynamics. As the buyer has more transactions with the seller, the HMM parameters should be updated in order to improve the prediction of the model. The state transition rates \( \Lambda \) and the observation probabilities \( B \) of the HMM are updated after a predetermined number of transactions. Such number should be chosen carefully. A smaller number can achieve higher prediction accuracy but cost more computations. The re-estimation of model parameters is done by the Baum-Welch algorithm[6] which takes much longer time than the forward algorithm. This algorithm is also integrated into our proposed method.

Buyer’s Policy. Given the model’s output the buyer should be able to decide his optimal trust-related behavior which may be difficult. Thus the authors apply reinforcement learning (RL) (Q-learning) to help users decide. Given the output from the model, this algorithm computes the best action the buyer should take. The detail of the RL algorithm is beyond the scope of this paper.