A Trust Model based on Markov Model Driven Gaussian Process Prediction

Sarangthem Ibotombi Singh
Department of Computer Science & Engineering,Tezpur University
Napam, Tezpur-784028, Assam, India

Smriti Kumar Sinha
Department of Computer Science & Engineering,Tezpur University
Napam, Tezpur-784028, Assam, India

ABSTRACT
In a completely open Web service environment, where identities cannot be directly checked, only hard security mechanisms are incapable to guarantee fair interactions among the service providers and service consumers. Trust and reputation modeling and management based on social approach is proved to provide the necessary safeguards against malicious interacting partners. In the heart of any trust modeling and management mechanism, predicting trust values for making a decision for interaction at future time is a key part. Trust prediction is a method of predicting potentially unknown trust of a target partner using its previously observed behaviour and also the recommendations received from other peers. In this paper, a trust prediction model based on detection of behavior pattern that may prevail at future time point using a Markov model is proposed. The trust value is obtained from a Gaussian process using the detected pattern.

General Terms
Clustering, Time series, Machine Learning, Trust and Reputation, Web Service.

Keywords
Trust, Reputation, clustering, Gaussian process, regression, Markov model.

1. INTRODUCTION
Web services have emerged as mechanism for an efficient and loosely-coupled cross organizational business-to-business integration. Web services are commonly operated under three main architectures [4,5,6] : Single – where only a single service is providing service to its users; Composite - where many services have assembled their capabilities together to provide a more complex function(s) that otherwise cannot be fulfilled by the individual service alone; Community–where many services providing the same functionalities form a community. One thing that is common to these three architectures is *"selection of the right service that can best meet a given criteria."* Given the possibility that a service may act in a malicious manner, the task of selection becomes an issue related to security. The usual security mechanism based on cryptographic measures alone cannot handle the problem of selection. These measures stop at the periphery of verifying credentials and checking the identities but fail to foretell how well a service will behave while delivering its functionalities. Trust and reputation mechanisms based on social approaches have found its ground in supplementing the cryptographic based approaches. Using trust and reputation modeling, service users are enabled to distinguish good services from bad ones. Practically, any provider has the freedom to publish bad quality, expensive, and even harmful services; which makes wise selection of great importance.

According to [1], service behaviour and quality of service (QoS) parameters can vary over time due to several sources listed below:

- The service provider, depending on load being experienced at a particular time, may provide different quality of service.
- A possible change in management or policy in the provisioning of the service by the service provider may lead to different quality of service of the provided service over time.
- Possibility of slow deterioration in the quality of the provided service over time.

Based on these observations, trust and reputation of a service provider cannot be treated as static. They are *dynamic, context-sensitive, transferable, and history based*. Context-sensitive nature of trust and reputation makes a player trustworthy in one context and untrustworthy in another context. Transferability means that one infer trust and reputation in a context from their values in other related context(s). For example, one can infer that an agent's good reputation as a politician is likely to mean that he is probably a good speaker also. However, his reputation as a driver cannot be inferred from the fact that he is a good politician. So, transference is allowed only between related contexts. Dynamic nature of trust and reputation makes their values time dependent. This may be attributed to a number of reasons which are discussed in detail in [1, 2]. Over a period of time, trust and reputation of an agent may either increase or decrease. Finally, history-based property emphasizes that current trust and reputation of an entity can be predicted based on their previous values [2]. Any trust and reputation modelling approach will be more complete if above mentioned characteristics are given due consideration.

A key element of modelling and managing trust is being able to accurately predict future trust values [1, 2, 3]. This is of particular importance when a service user needs to make a decision about the service provider at a future point in time. The basic goal of prediction or forecasting is, first, to generate a model of the process under observation, and then to use the model to predict values that have not yet been measured. The concept of *predicting* or *forecasting* values is not new. Different models and theories such as the Markov Model [7], Kalman Filter Theory [8], Holt-Winter forecasting method [9], Neural Networks [10], Bayesian Networks [11] etc, have been proposed in the literature for forecasting and been developed for various application domains such as energy forecasting, weather forecasting, stock market forecasting etc These models can be a *Global model* or a *Local model.* In a Global model, the relationship between the input and the output values is described by a single analytical function over

the whole input domain. On the other hand, local modelling only creates a specific model that describes the systems behaviour for a given input. The input for which the prediction has to be performed is only known at the prediction time. Again prediction can be *one-step-ahead prediction* or *k-step-ahead prediction*.

In this paper, a prediction based on Markov model and Gaussian Process Regression for time series forecasting is proposed. Repeatedly appearing similar sub sequences in the trust time series constructed from history of direct interactions or recommended trust values collected from intermediaries over a sequence of time slots are clustered into regimes. A transition network of these regimes is learned by a Markov process. Finally a Gaussian process is used to predict the trust value one step-ahead in the future.

2. RELATED WORK

A number of trust and reputation models have been published in the literature [12,13,14]. In [15] a trust model for autonomous agents in multi-agent environments based on Hidden Markov models (HMM) and reinforcement learning is proposed. Trusting agent in the system rates all other agents after an interaction and uses an HMM per agent to decide and predict whether or not the agent is malicious. The HMM is updated from observations which come in the form of ratings after direct experiences or recommendations requested from other intermediaries. The authors, in [22], provided an idea of a trust model based on HMM to cope with the inability of probabilistic trust models to capture the dynamic aspect of trust making over time. A HMM based approach to measuring an agent's reputation as a recommender is proposed in [23]. They model the chained recommendation events as an HMM. The main features of the model are (1) no explicit requirement of chained recommended reputations, (2) a flexible recommendation network with presence of loops, and (3) integration of learning speed into trust evaluation reliability. In [24], an HMM and digital signatures based architecture for trust management in ubiquitous environments is proposed. Here the HMM is used to infer about user presence from incomplete sensor signals. The model in [6] uses a Markov chain constructed from a reputation time series to model the dynamic nature of trustee's trustworthiness. The current state vector show the repute value of the trustee at a time slot. The Markov matrix of the trustee denotes the probability that it will transit from one trustworthiness level to another trustworthiness level based on its past behaviour as captured by the Markov chain. The future state vector is determined by multiplying the current state vector with the Markov matrix. We propose a different approach in which different segments in the whole series representing same volatility are groped to form a state of trusting behaviour of the trustee. After discovering all such states, the dynamics of changing between regimes over time is modelled by a Markov Model. The model, similar to [6], predicts the future regime and using the behaviour patterns in this regime are used to predict the future trust value. Our prediction model is a Gaussian process regression.[17]

3. THE PROPOSED MODEL

3.1 Mathematical Background

3.1.1 Gaussian Process

Our predictor is a Gaussian Process predictor. A brief introduction to Gaussian Process (GP) is presented here. For a comprehensive report on GP, kindly refer [17]. Formally, a Gaussian Process can be defined as:

Definition: A Gaussian Process is a collection of random variables, any subset of which has a joint normal distribution.

A Gaussian process is completely defined by its mean function and covariance function.

$$f(x) \sim GP(m(x), \Sigma) \quad (1)$$

Given a set of data $D = \{x_{i,} y_i\}_{i=1}^{n}$, where $x_i \in R^d$ and $y_i = f(x_i) + \varepsilon, \in R$, the input-output relationship is modelled by using a Gaussian process with mean function $m(x)$ and covariance function Σ. In most of the applications, the mean function is set to zero, and any covariance function generating a positive definite covariance matrix is used. In order to make prediction about a new input $x_* \in R^d$, the joint distribution of the training outputs f and the test output f_* (Eq. 2) is conditioned on the observations and the expected value is obtained according to Eq. 3 and variance of the prediction according to Eq. 4

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X)+\sigma_n^2 I & K(X,x_*) \\ K(x_*,X) & k(x_*,x_*) \end{bmatrix}\right) \quad (2)$$

$$\mathrm{E}[f_* \mid X, x_*, f] = K(x_*,X)\left[K(X,X)+\sigma_n^2 I\right]^{-1} f \quad (3)$$

$$\mathrm{cov}(f_*) = k(x_*,x_*) - K(x_*,X)\left[K(X,X)+\sigma_n^2 I\right]K(X,x_*) \quad (4)$$

where X represents the matrix of training inputs, K denotes the covariance matrix which is obtained by pairwise evaluation , $\Sigma_{ij} = \mathrm{cov}(y_i, y_j) = \mathrm{cov}(f(x_i), f(x_j) = k(x_i, x_j)$ of the covariance function for the given inputs. Writing in shorthand form, the predicted value of the process at the new input and its prediction variance are

$$\bar{f}_* = k_*^T (K + \sigma_n^2 I)^{-1} y = k_*^T \alpha \quad (4)$$

$$\mathrm{cov}(f_*) = k(x_*,x_*) - k_*^T\left[K + \sigma_n^2 I\right]k(x_*) \quad (5)$$

here $k_* = K(X, x_*)$, y is the vector of training function outputs, and σ_n^2 is the variance of the Gaussian noise $\in$. The covariance function chosen is not completely free from parameters and they are called *hyper-parameters*. Let us accumulate these parameters in θ. Parameters θ can be learned from the training data by marginal likelihood optimization. The log-marginal likelihood is defined as

$$\log(y \mid X, \theta) = -\frac{1}{2} y^T K^{-1} y - \frac{1}{2}\log\left|K + \sigma_n^2 I\right| - \frac{n}{2}\log 2\pi \quad (6)$$

The gradient of the marginal likelihood with respect to θ can be computed by a gradient optimization technique to minimize the objective function in Eg. 6.

3.2 Model Elements

3.2.1 Definitions

Time Horizon: *Total time duration in the past over which the service user will analyse the trustworthiness of a service provider in order to make a trust-based decision for future interaction.*

The time horizon is a positive value representing years or months or days or seconds depending on the user. The idea is that for making a trust-based decision for a future interaction, a service user may like to analyse the behaviour of the service provider over previous years or months etc.

Time Slot: *A finite duration of time in the time horizon over which the direct trust value or recommended trust values collected from the direct experiences or from other intermediaries are aggregated into a single value for analysis of its dynamic nature of trustworthiness of the service provider.*

Time slot allows us to divide the time horizon into equidistance intervals. For example, a time horizon of one year, can be divided it into days, giving a sequence of 365(6) equidistance intervals. For each interval, using an aggregation method, a single value of trust value can be generated. This process will generate a time series of trust values.

Time Point: *The time of interaction between a service user and the service provider and at which the trustworthiness value based on the outcome(s) of the interaction is recorded by the service user.*

Time point will help to identify which past interaction(s) falls under a given time slot.

Direct Trust: *A measure of trustworthiness of the service provider in a given context and at a given time point established by a service user from the previous interactions with this provider.*

Recommended trust: *A measure quantifying the trustworthiness of the service provider in a given context and at a given time point as communicated by an intermediary.*

Reputation Trust: *A numerical value representing the truthfulness of the recommended trust provided by an intermediary.*

3.2.2 Trust Intermediaries

A system based on the knowledge of trustees' past behaviour could sustain trust and a consistent degree of trustworthiness provided that information was sufficiently reliable. Following argument in [23], it is crucial to understand the role of trust intermediaries who have positions and interests either analogous or different from the trustors. When positions and interests are aligned, trustors are expected to seriously consider the opinion of the intermediaries so that their decisions will reflect reputational information available. Intermediaries may be either an advisor or a guarantor. Trustor trusts the advisor's judgment which leads him to place trust on the potential trustee. So always there is an element of risk involved while taking the recommendations from advisors. The trustor, however places trust on a guarantor's performance and integrity just as the later does in that of the potential trustee. So we claim the following in our model:

- A guarantor intermediary is one whom the trustor has already established a recommendation trust relationship and from whom the opinion of the trustee's behaviour can be elicited.
- An advisor intermediary one whom the trustor has not established any recommendation trust relationship earlier, yet it can provide an opinion of the trustee's behaviour.

Our opinion is that a guarantor is already know to the trustor from their past exchanges of recommendations while an advisor is an unknown one. Again if a guarantor is within the reputation trust range of the trustor i.e. its reputation trust value is above a threshold, then it is called as a *known and trusted guarantor*.

3.2.3 Information sources

According to [19], the placement of trust on a trustee is essentially based on the information available to a trustor from three sources:

1. Trustor's assessment of trustee's performance. (direct source)
2. Recommendations from other intermediaries who have a position similar to the trustor's and similar interest on the placement of trust.
3. Recommendations from other intermediaries who do not have a position similar to the trustor's and do not have the similar interest.

Source (1) passing through no intermediaries at all, will be most likely to lead to a correct assessment. Source (2) often leads to the decision about trust as made by other intermediaries whose judgment was trusted. Finally source (3), provides the independent evidence of the decision.

In our prediction model, "position and interest" similarity is decided from the context information. It is explained in following with an example.

Let c_i, c_j be two possible contexts of interaction with a service providor. Let *A* be a service user who wants to interaction with the service provider in future and let c_i be its context. Let *B* be another service user who already has interacted with the same provider in the context *cj* and has established an opinion about the provider. Let $ConSim(c_i, c_j) \in [0, 1]$ be an operator which evaluates the similarity of any two contexts with a value of 1 meaning exact match and 0 meaning exact mismatch. So, $ConSim(c_i, c_j) = 1$, means that the *A*'s present position and interest is analogous to that of *B* 's past experince. *A* can directly utilze *B*'s opinion in its analysis of provider's past behaviour. By $0 < ConSim(c_i, c_j) < 1$, it is meant that *A*'s position and interest is not aligned completely to that of *B*. In this case, the opinion of *B* can still be utilized in *A*'s analysis following the *transferrability* property of trust and reputation. With this explaination, trust intermediaries in the proposed model are shown in Table 1.

Table 1. Categorization of Trust Intermediaries

Name	Know & Trusted	Position & Interest	Type Source
Type I Guarantor	yes	$ConSim(c_i, c_j) = 1$	2
Type II Guarantor	yes	$0 < ConSim(c_i, c_j) < 1$	3
Type I Advisor	no	$ConSim(c_i, c_j) = 1$	2
Type II Advisor	no	$0 < ConSim(c_i, c_j) < 1$	3

3.3 Interaction Architecture & Database

In our model, a reasonable size network of service users' and a single service provider (Figure 1.) is assumed. Services from the service provider are accessed by the service users. They record the trustworthiness values of the service provider they have previously interacted with. Service users also communicate to each other about (1) their experience with the service provider and (2) their experience with other service users in soliciting recommendation trust. This communications serve as source of third party feedbacks. The second communication is an important one because using the

information from this channel one can validate the trustworthiness of the third source of information i.e. *advisor* used in our model.

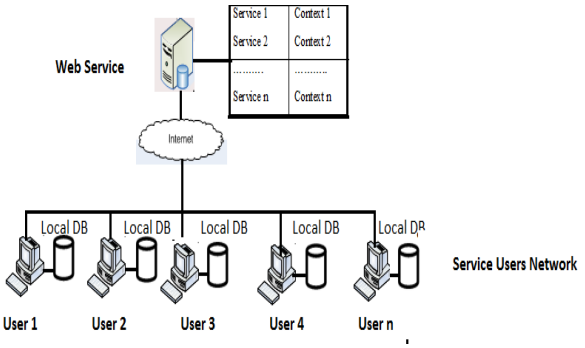

Figure 1: Single Service Access Architecture

Each service user in the network maintains the following information in its local database.

- Direct Trust Table: It stores the trustworthiness values of all service providers that a service user has interacted with in the past.
- Reputation Trust Table: It stores the reputation trust values of all other service users from whom recommended trust values of service providers have been collected.

The structures of these tables are given in the following Figure 2.

Provider Id	Trust Value	Context	Time

(a) Direct Trust Table Entry

Service User Id	Reputation Trust	Time

(b) Reputation Trust Table Entry

Figure 1: Table Entry format

Using entries in the Direct Trust table, a service user can do the prediction of trustworthiness of a service user in a future time slot. It is referred as *direct trust prediction.* Secondly, service user also can answer a reputation query from other service users using this table. Each user is associated with an individual level of trustworthiness determined by the quality of its answer to such query. Reputation trust value in the reputation trust table reflects this level of trustworthiness. Its numeric value can be used by a service user to decide whether other witness service users are within its reputation range. Reputation trust value of a witness service user is calculated from the difference between:

- Recommended trust value communicated by the witness agent about the target service provider.
- The actual trustworthiness value obtained on interaction with the target service provider.

The following simple mechanism for reputation trust calculation is used.

- If actual trustworthiness value is greater than or equal to the recommended trust value, reputation trust is set to 1; otherwise it is set to -1.

Context information in direct trust table and reputation trust value in the reputation trust table will enable us to categorize a service user as one of the intermediary types given in Table 1.

3.4 Trust Equations

3.4.1 Final trust

During the analysis of behavior of a service provider, a service user calculates provider's final trust based on the history of direct interactions and/or recommended trust values obtained from other intermediaries. So the final trust of a service provider A over a context c_i at a future time point t is evaluated from the following relation.

$$T(A,c_i,t) = T_d(A,c_i,t\ ) + T_r(A,c_i,t) \qquad (7)$$

where $T_d(A,c_i,t)$ is the direct trust value obtained from direct interaction history stored in the direct trust table, $T_r(A,c_i,t)$ is the indirect trust estimated from the recommended trust values obtained from other intermediaries, and α is the weighting factor to decide the importance between these two trust values.

3.4.2 Direct Trust: $T_d(A,c_i,t)$

To calculate the direct trust value, the service user must decide the time slot over which all its direct interactions with the service provider must be examined. For example, if the future time point of interaction is next month, then the time slot can be the current year. If the service user already had directly interacted with the service provider in the identified time slot, $T_d(A,c_i,t)$ is evaluated as

$$T_d(A,c_i,t) = \frac{1}{n}\sum_{i=1}^{n} e^{-\lambda(t-t')} * conSim(c_i,c_j) * T_{-d}(A,c_j,t') \quad (8)$$

where $T_{-d}(A,c_j,t')$ is the direct trust value recorded in the direct trust table from a previous interaction with the provider at time point t' and context c_j, $conSim(c_i,c_j)$ allows the transfer of trust from similar context, $\lambda \in [0,1]$ takes care of the trust dynamics over time, n is the number of all previous direct interactions recorded.

While using Eq. 8, it may happen that the service user has no previous interactions in the identified time slot. This is true from two possibilities: (1) all previous interactions were in the previous time slots, *for example in the previous years*, and (2) the provider is a complete stranger. In this scenario, our service user have to choose one or both of the following:

Choice 1: Calculate the final direct trust value $T(A,c_i,t)$ only from $T_r(A,c_i,t)$ component of Eq. 7.

Choice 2: Use the prediction mechanism presented in the next section to get an estimate $\hat{T}_d(A,c_i,t)$ by considering a larger time horizon.

The procedure for direct trust calculation is summarized in the following algorithm.

Algorithm: Direct Trust

1. Decide the time point t and context c_i of future interaction.

2. Decide the time slot over which the analysis of behavior of the service provider is to be done.
3. If previous interactions are available in the identified time slot then

Calculate $T_d(A,c_i,t)$ from Eq. 8.

4. else
5. Case based on Choice
6. Choice is 1: Set $T_d(A,c_i,t)$ to 0.
7. Choice is 2: Estimate $\hat{T}_d(A,c_i,t)$.
8. End Case
9. End If

3.4.3 Indirect Trust : $T_r(A,c_i,t)$

The service user first seeks the recommendation trust from the intermediaries by submitting a recommendation trust query specifying time slot. All the intermediaries, who have previously interacted with the service provider, in *some context* and within the specified time slot, reply back to querying service user with a recommended trust value for the target service provider. After receiving all the replies, our service user will categorized the values as coming from the types of intermediaries mentioned in *Section 3.2.3*. The *Beta* distribution is used to select guarantors and also to weight the feedback from advisor. A quantity called *reputation range measure* of every replying intermediary is calculated as

$$R = \frac{\#p}{\#n + \#p} \tag{9}$$

where $n\#$ and $\#p$ is the number of negative entries and positive entries, against each replying user, in the reputation trust table of our service user. If $R \succ R_{th}$ then a replying intermediary is said to be within the reputation query range of the querying service user and hence it is termed as *guarantor*. All other intermediaries for which there is no entry in the reputation trust table are considered as unknown and hence they are termed as *advisor*s. To solicit feedback from them, our querying agent must measure their reputation trust indirectly. First a reputation trust query must be forwarded to all its guarantors, identified in the previous step, by specifying the name of the advisor. Each guarantor, if they know, the targeted advisor will return a pair $(\#n,\#p)$ from their reputation trust tables. From these pairs, a final reputation range measure of the targeted advisor will be calculated as

$$Tot(\#n) = \sum_{i=1}^{ng} (\#n)_i \tag{10}$$

$$Tot(\#p) = \sum_{i}^{ng} (\#p)_i \tag{11}$$

$$R_a = \frac{Tot(\#p)}{Tot(\#p) + Tot(\#n)} \tag{12}$$

where ng is the number of guarantors identified. If $R_a \succ R_{ath}$ then the feedback from the targeted advisor will be solicited.

Having devised the mechanism for identification and selection of intermediaries, the calculation of $T_r(A,c_i,t)$ is done as follows

$$T_{rg\_tot}(A,c_i,t) =$$

$$\frac{1}{ng}\sum_{i=1}^{ng} R_i * e^{-\lambda(t-t')} * conSim(c_i,c_j) * T_{rg_i}(A,c_j,t') \tag{13}$$

$$T_{ra\_tot}(A,c_i,t) =$$

$$\frac{1}{na}\sum_{i=1}^{na} R_{a_i} * e^{-\lambda(t-t')} * conSim(c_i,c_j) * T_{ra_i}(A,c_j,t') \tag{14}$$

$$T_r(A,c_i,t) = 0.5*[T_{rg\_tot}(A,c_i,t) + T_{ag\_tot}(A,c_i,t)] \tag{15}$$

where na is the total number of identified advisors. The indirect trust calculation procedure is summarized in the following algorithm.

Algorithm: Indirect Trust

1. Decide the time point t and context c_i of future interaction.
2. Decide the time slot over which the analysis of behavior of the service provider is to be done.
3. Decide the thresholds R_{th}, R_{ath}
4. Issue a reputation query specifying the time slot.
5. For each replying intermediary
6. Begin
7. if there exist a record in the reputation trust table
8. Calculate R_i using Eg. 9
9. if $R \succ R_{th}$
10. Mark the intermediary as a guarantor.
11. End If
12. else
13. Mark the intermediary as advisor.
14. End If
15. End Begin
16. End For
17. From all marked guarantors calculate $T_{rg\_tot}(A,c_i,t)$ using Eq. 13.
18. For each advisor
19. Begin
20. Issue reputation trust query to marked guarantors.
21. Calculate R_a using Eq. 10-12.
22. if $R_a \succ R_{ath}$
23. Mark the advisor as useful.
24. End Begin
25. From all useful advisors calculate $T_{ra\_tot}(A,c_i,t)$ using Eq 14.

26. Calculate $T_r(A,c_i,t)$ from Eq. 15

It may so happen that our service user is unable to obtain the recommended trust values from the intermediaries because none of the intermediaries has interacted with the provider during specified time slot. In such scenario, the analysis is to be done in a larger time horizon to get an estimate $\hat{T}_r(A,c_i,t)$.

3.5 Trust Prediction

When our service user cannot measure $T_d(A,c_i,t)$ and/or $T_r(A,c_i,t)$, analysis is to be done in a wider time horizon. A prediction mechanism using non-linear time series mechanism and Gaussian Process Regression [18] is used. The prediction mechanism is explained in the following subsections.

3.5.1 Prediction Steps

Step 1: Parameters Selection.

- Select the time horizon T_h over which the trustworthiness of the service provider is to be analyzed.
- Select duration T of time slot to divide the time horizon into N equal intervals of length $\frac{T_h}{T}$ each.
- Select time spot T_p at which a trust based decision of whether or not to interact with the service provider in a given context is to be made. This is considered to fall in the immediate next time slot.

Step 2: Data Exploration and Time series formation.

Case 1: Direct trust data.

Service user determines if it has the context specific trust information of the service provider for the specified time horizon in its direct trust table. Available data are then distributed into time slot intervals over the time horizon using the time point of each record. Then in each slot, aggregation method in Eq. 8 is applied to generate a single direct trust value. This value is tagged with the interval number. Here it is assumed that all intervals will receive at least one data point. This can be done by adjusting the slot length.

Case 2: Recommendation trust data.
Service user issues a recommendation trust query to all other service users by specifying the time horizon and the service provider's Id in the query. The replies are collected and distributed into the time intervals over the horizon. Using Eq. 9-15, the data points falling in each interval are converted into a single recommendation trust value. Each value is tagged with the respective interval number.

The time series generated in either case is termed as $Y=\{y_t\}_{t=1}^{N}$ where N is the number of intervals in the time horizon and y_t is the direct trust or recommended trust value of each slot. Thus *Y* becomes a time series sampled with time equal to slot duration.

Step 3: Course graining of the Time series.

From the field of non-linear time series analysis, there exists a mapping f in the state space satisfying:

$$y_t = f(y_{t-1}, y_{t-2}, ..., y_{t-m}) \tag{16}$$

The prediction of the of the time series is the regression of the trajectory over the observed samples and generating the future value from the reconstructed state space

$$X_t = [y_{t-1}, ..., y_{t-m}] \tag{17}$$

The segmentation of the time series is implemented in the space R^{m+1} of the joint velocity of the state vector and the corresponding output as $[\Delta X_t, \Delta y_t] = \left[X_t - X_{t-1}, y_t - y_{t-1} \right]$. The velocity vector of the sate vector is

$$D_t = \left[y_t - y_{t-1}, y_{t-1} - y_{t-2}, ..., y_{t-m} - y_{t-(m+1)} \right] \tag{18}$$

The segmentation is achieved by using fuzzy-C mean clustering [18]. The clustering mechanism calculates the cluster membership degree μ_i as the degree to which D belongs to cluster i and updates the cluster centers V_i iteratively to minimize the objective function

$$O = \sum_{i}^{M} \sum_{j}^{N} (\mu_i(t))^2 \left\| D_t - V_i \right\| \tag{19}$$

where M is the number of clusters. Please note that each time value t has a corresponding pair $x_* = X_i$ $[X_t, y_t]$ assigned to it. Therefore, the original series can be visualized as a series with $[X_t, y_t]$ as its observed value. This series is called as course grained series of the trust series. Using the membership degree obtained from the clustering algorithm, each $[X_t, y_t]$ is assigned to cluster for which membership degree is maximum. Each cluster now contains vectors representing same volatility region of the trust time series and it is called as a *regime*. Then whole time series is regarded as the evolution of the regimes over time. In this manner, one can look into the trust series at a course level. The principle behind the approach is that each regime represents the same behavior pattern over time. So must be extracted to learn a local model.

Step 4: Markov Model of the course grained series.

Now each observed trust value y_t is associated with a vector $z_t = \left[X_t, y_t \right]$ from a particular regime, so one can model the original trust value sequence as a regime transition network using a Markov chain.

The Markov transition matrix $A_{M \times M}$ is to be constructed to find the probability of changing from one regime to another regime between any two consecutive time slots. This is a way of finding the change in the behavior pattern of our service provider. First, define the following:

Inter-regime transition: $V_i \to V_j$: A transition $V_i \to V_j$ is said to occur from time t to time $t+1$ if $z_t \in V_i$ implies $z_{t+1} \in V_j$

Intra-regime transition: $V_i \to V_i$: A transition $V_i \to V_i$ is said to occur from time t to time $t+1$ if $z_t \in V_i$ implies $z_{t+1} \in V_i$

Now define the regime transition probability of the Markov matrix as

$$a_{ii} = p(V_i \rightarrow V_i) = \frac{TotOf(V_i \rightarrow V_i)}{\sum_{k=1}^{M} TotOf(V_i \rightarrow V_k)} \quad (20)$$

$$a_{ij} = p(V_i \rightarrow V_j) = \frac{TotOf(V_i \rightarrow V_j)}{\sum_{k=1}^{M} TotOf(V_i \rightarrow V_k)} \quad (21)$$

where the *TotOf* operator counts the total number of transitions from cluster to cluster.

Step 5: Prediction.

Once the matrix is constructed, prediction for trust value for future time point is done in the following steps.

- **Construction of current state vector (S_c):** Current state vector is a vector of size $1 \times M$. First for the last time spot N, the associated vector $z_{t=N}$ is identified and its owner cluster V_i is decided. Then the *ith* entry of the state vector is set to 1 and all other entries to 0.
- **Finding of Next state vector (S_f):** Future state vector is generated by $S_f = S_c * A$.
- **Selection of state vectors:** Using the future state vector, S_f the next regime(s) are identified to select the training data for the predictor GP. Selection detail is explained in the prediction step.
- **Prediction procedure:**
  The prediction is performed by using a mean zero Gaussian Process. For this purpose, first the kernel of GP is selected as the square exponential kernel.

$$k(x, x') = \sigma_f^2 \exp(-\frac{(x - x')^2}{2l^2}) \quad (22)$$

This kernel is a stationary kernel with the hyper-parameter *l* representing its length scale. The full set of hyper parameters of our GP is accumulated in $\theta = \{l, \sigma_f, \sigma_n\}$. These hyper parameters are learnt by gradient descend optimization technique to minimize the objective function in Eg. 6. The Matlab source code available at http://www.gaussianprocess.org/gpml is used.

For the training of the GP to learn the hyperparameters, data selection is done based on the following observation.

Observation 1: The current regime vector $z_{t=N}$ and the next regime vector $z_{t=N+1}$ in the course time series will have $(m-1)$ values in common.

This can be clarified from the Figure 3. This observation helps in filtering the training set of our GP.

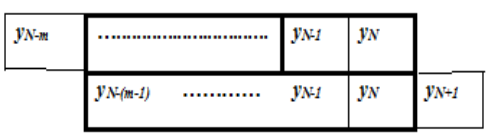

Figure 3 : Common values between adjacent regime vectors

Please not that due to windowing effect in the generation of course grained series, the substring $[y_{N-(m-1)}, ...y_N]$ of X_N is the state vector for next time spot $N+1$. The value y_{N+1} is our target of the prediction. From the future state vector f_s, one or all the regimes (clusters) that immediately follow the regime at time N can be found. Either only the regime with the highest transition probability value in f_s can be taken as the next regime or all regimes with none zero transition probability value can be considered. Former can be considered as *winner takes all* policy of lazy learning while the latter in *ensemble* approach. The proposed mechanism used the ensemble approach but with a *filtering* procedure.

The training set in is generated in the following manner.

- Current regime vector $z_n = [X_N, y_N]$ is identified as the last regime vector in the course time series. Its owning regime as given by current state vector S_c is called V_N.
- All the regimes, whose transition probability in future state vector S_f is nonzero, are identified. They represent the groups of possible behavior patterns of the service provider in the future time slot.
- The regime vectors of the form $[X_i, y_i]$ belonging to these clusters are filtered further by examining their time slots. Only the vectors whose time slot comes next to the time slot of any of the vector in V_N are retained in a group. Before applying this filtering, z_n from V_N is to be removed.
- For each regime vector, $z_i = [X_i, y_i]$ in the group, a modified form of normalized cross correlation between X_N and X_i of z_i is calculated as

$$xC(X_N, X_i) = \frac{\sum_{j=1}^{m} \left[y_{N-j} - \bar{Y} \right] \left[y_{i-j} - \bar{Y} \right]}{\sqrt{\sum_{j=1}^{m} \left[y_{N-j} - \bar{Y} \right]^2 * \sum_{j=1}^{m-1} \left[y_{i-j} - \bar{Y} \right]^2}} \quad (23)$$

  where $\bar{Y}$ is the mean of all observations in the trust time series $Y = \{y_t\}_{t=1}^{N}$. For this calculation please refer to Eq. 17. $\bar{Y}$ is used because when the either state vector X_i or X_N is a flat pattern, $xC(X_N, X_i)$ is undefined if the mean of X_i or X_N is used in calculation of Eq. 23.
- Only the regime vectors whose $xC(X_N, X_i)$ value is above a given threshold xC_{th} are selected from the group and are retained in the final training data set of our GP. This process like finding the nearest neighbour of X_N.
- Now, $x_* = X_N$ is taken as the new point for the prediction of $\bar{f}_*$ from the GP using Eq. 4.

The prediction model is a local GP model because the regime vectors are selected not from all clusters. To check the efficiency of our model, a global GP trained by using all the

regime vectors of the course grained trust time series is used. The query point x_* of this global model is selected from the training set of the local GP model. The selection approach is as given below.

- For every pair of regime vectors (z_i, z_j) in the local training set formed above, Eq. 23 is extended to calculate correlation between them by including $[y_i, y_j]$ in the calculation. The required equation is given in Eq. 24. After calculating the pairwise distances, the regime vector which average similarity is the maximum is selected. Let it be z_i . Average similarity is calculated by Eq. 25

$$xC(z_i, z_j) = \frac{\sum_{k=0}^{m}\left[y_{i-k} - \bar{Y}\right]\left[y_{j-k} - \bar{Y}\right]}{\sqrt{\sum_{k=0}^{m}\left[y_{i-k} - \bar{Y}\right]^2 * \sum_{k=0}^{m-1}\left[y_{j-k} - \bar{Y}\right]^2}} \quad (24)$$

$$\gamma_{z_i} = \sum_{j=1}^{K} xC(z_i, z_j) \quad (25)$$

where K is the total number of regime vectors in the training set. By using the average similarity, in a way it is trying to find the regime vector which has occurred most frequently in the past. This is the most likely pattern to follow the current regime z_N .Then $x_* = X_i$ is use as the test point to predict $\bar{f}_*$ from the global GP using Eq. 4.

4. IMPLEMENTATION

We have tested the accuracy of our GP predictor on a synthetic data series. We generated a series of 500 data points. The next value in the series changes randomly with a factor ϕ (next value/current value). We assume a wide range [0.6, 1.4] for the factor ϕ so that the GP model can be tested in a difficult situation. Moreover, the minimum and maximum values of the series are set to be 0.1 and 1, respectively. Out of these 500 data point 480 points are used as the direct trust or reputation trust series. The model order i.e. lag value *m* was decided by FNN method [20]. The number of clusters i.e. M=5 was chosen for clustering. A rolling prediction is performed by allowing the next value of the series to enter into the training set sequentially. The predicted result is shown in the figure 4.

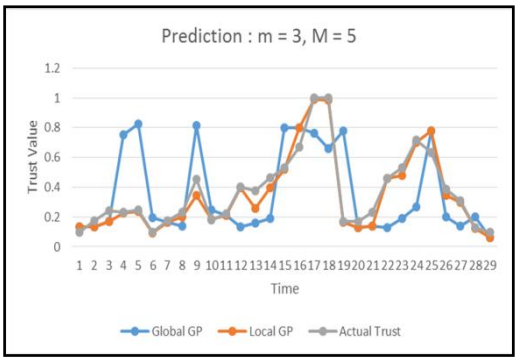

Figure 4: Prediction from Local and Global GP using square exponential kernel.

5. CONTRIBUTIONS

We have proposed a trust model using the machine learning approach of Gaussian Process. To the best of our knowledge, this is the first model using GP in the computational trust model.

6. CONCLUSIONS

Our future work will be to test the proposed model for its efficiency in a more elaborate manner. It has provided a framework of using a GP in trust forecasting. Kernels of a GP describe the input data pattern. So an extension can be to use different GPs with different kernels and use a model selection approach to choose the best model at a given time. A comparative study of the proposed model with the existing Web service trust prediction models is another future work.

7. REFERENCES

[1] Chang, E., Dillon, T., and Hussain, F.K, 2005 Trust and Reputation for Service-Oriented Environments : Technologies for Building Business Intelligent and Consumer Confidence, John Wiley and Sons, U.K, ,

[2] Farookh K. Hussain, Elizabeth Chang and Hussain, O. , 2008 A Robust Methodology for prediction of Trust and Reputation Values, Proceedings of the ACM workshop on Secure web services, Alexandria, Virginia, USA.

[3] Hussain, F., Chang, E., Dillon, T., 2005 Markov model for managing dynamic trust, INDIN, 3rd International Conference on Industrial Informatics, Frontier Technologies for the Future of Industry and Business, IEEE.

[4] Alonso, G., Casati, F., Kuno, H., Machiraju, V. 2004 Web Services: Concepts, Architectures and Applications, 1st edition Springer Publishing Company

[5] Maamar, Z., Subramanian, S., Thiran, P., Benslimane,V., Bentahar, J. 2009 An approach to engineer communities of web services: concepts, architecture, operation, and deployment, International Journal of E-Business Research 5 (4).

[6] Pathak, J., Basu, S., Honavar, V., Assembling composite web services from autonomous components, 2007 Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering, IOS Press.

[7] Hassan, M.R., Nath, B., 2005 Stock Market forecasting using hidden Markov model: a new approach. In: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, Wroclaw, Poland.

[8] Kalman, R.E., 1960 A new approach to linear filtering and prediction problems, Trans. ASME J. Basic Eng. 82(Series D).

[9] Holt, C.C.: Forecasting seasonal and trends by exponentially weighted moving averages. Int. J. Forecast.20, 5–10 (2004)

[10] Drossu, R., Obradovic, Z., 1996. Rapid design of neural networks for time series prediction. IEEE Comput. Sci. Eng. 3(2).

[11] Hou, L., Wang, L., Yang, J., 2008. Evolutionary prediction of online keywords bidding. E-Commerce and Web Technologies, Springer, Berlin/Heidelberg

[12] Jordi Sabater and Cales Siera, Review on Computational Trust and Reputation Models. 2005 Artificial Intelligence Review.

[13] Josang, A., Ismail, R., Boyd, C., 2007. A Survey of Trust and Reputation Systems for Online Service provisioning, Decision Support Systems

[14] Yao Wang and J. Vassileva, Towards Trust and Reputaion Based Web Service Selection: A Survey, [Online]. Available: ttp://www.caip.rutgers.edu/TASSL/Fall05Reading/Viraj_webservicescomp.pdf

[15] Moe, Marie Elisabeth Gaup, Tavakolifard, Mozhgan, Knapskog, Svein Johan, 2008. Learning Trust in Dynamic Multiagent Environments using HMMs, Proceedings of The 13th Nordic Workshop on SecureIT Systems. Copenhagen, Denmark

[16] Sassone,V, Krukow, K., Nielsen, M., Towrds a FormalFramework for Computational Trust, 2006. Proceedings of the 5th International Symposium on Formal Methods for Components and Objects (FMCO 2006), vol. LNCS 4709, Springer.

[17] Carl Edward , Gaussian Processes for Machine Learning, Rasmussen and Chris Williams, the MIT Press, 2006

[18] Bezdek, J 1981 Pattern Recognition with Fuzzy Objective Function Algorithms, New York, Plenum Press

[19] Coleman, J. S. (1990). Foundations of Social Theory. Harvard University Press, Harvard.

[20] Kennel, M, Brown, R, Abarbanel,H. 1992 Determining embedding dimension for phase-space reconstruction using a geometrical construction. Physical Review A 45(6), 3403-3411

[21] Song, W., Phoda, V.V., X. Xu., 2004. The HMM-based Model for evaluating Recommender's Reputation, Proceddings of the IEEE International Conference on Ecommerce Technology for Dynamic EBusiness (CEC-East'04), IEEE.

[22] Noda, J., Takashashi, M., Hosomi, I., Mouri,V, Takata, Y., Seki, H., 2006. Integrating presence inference into trust management for ubiquitous systems, Proceedings of the eleventh ACM symposium on Access Control models and technologies, ACM Press, New York,

[23] Song, W., Phoda, V.V., Xu, X., 2004.The HMM-based Model for evaluating Recommender's Reputation, Proceedings of the IEEE International Conference on Ecommerce Technology for Dynamic EBusiness, IEEE

[24] Noda, J., Takashashi, M., Hosomi, I., Mouri, H., Takata, Y. Seki, H., 2006. Integrating presence inference into trust management for ubiquitous systems, Proceedings of the eleventh ACM symposium on Access Control models and technologies, ACM Press, New York.