Trust Assessment-Based Multiple Linear Regression for Processing Big Data Over Diverse Clouds

Hadeel El-Kassabi[1,2(✉)], Mohamed Adel Serhani[2], Chafik Bouhaddioui[2], and Rachida Dssouli[1]

[1] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada
h_elkass@encs.concordia.ca, rachida.dssouli@concordia.ca
[2] College of Information Technology, UAE University, Al Ain, United Arab Emirates
{serhanim, chafikb, htalaat}@uaeu.ac.ae

Abstract. Assessing trust of cloud providers is considered to be a key factor to discriminate between them, especially once dealing with Big Data. In this paper, we apply Multiple Linear Regression (MLR) to develop a trust model for processing Big Data over diverse Clouds. The model relies on MLR to predict trust score of different cloud service providers. Therefore, support selection of the trustworthiness provider. Trust is evaluated not only on evidenced information collected about cloud resources availability, but also on past experiences with the cloud provider, and the reputation collected from other users experienced with the same cloud services. We use cross validation to test the consistency of the estimated regression equation, and we found that the model can perfectly be used to predict the response variable trust. We also, use bootstrap scheme to evaluate the confidence intervals for each pair of variables used in building our trust model.

Keywords: Trust · Multiple Linear Regression · Cloud · Big Data · Community management

1 Introduction

With the abundance of cloud services sharing the market space, it becomes challenging to select the appropriate, and trustworthy cloud providers that guarantee user's quality preferences and ensure continuity of service provisioning especially when dealing with Big Data. Big Data processing requires trustworthy cloud provider who ensures service delivery with high QoS guarantee. The dynamic nature of cloud makes it hard to evaluate the trust of cloud providers to process Big Data as it is dynamic in nature and can be subject of continuous resource availability, high dependability, and fault tolerance. Previous trust models are non-dynamic and lack of real-time adaptability, which makes them unsuitable in the context of Cloud and Big Data. Building trust only based reputation can be irrelevant if the users are untrustworthy or subjective. Also,

F. Belqasmi et al. (Eds.): AFRICATEK 2017, LNICST 206, pp. 99–109, 2018.
https://doi.org/10.1007/978-3-319-67837-5_10

trust models have used local trust and recommendation trust using weights that are not necessarily dynamic and suitable to the user's preferences. Therefore, we need trust to be dynamic and relies on evidenced information collected about cloud resources availability, past experiences with the cloud provider, and the reputation calculated from other users experienced with the same cloud services. The trust model we aim to develop in this paper will fulfill the following requirements: (1) Supports dynamic trust score calculation and update, (2) Provides credibility validation through community management system, and (3) Collects reputation information dynamically using reputation request messages broadcasted to community members.

In this paper, we first describe trust approaches in Cloud. We then, formalize trust evaluation of cloud providers using Multiple Linear regression. Afterwards, we describe our trust prediction scheme, and we evaluate it using data generated from a simulator we have developed for this purpose.

2 Background and Related Work

2.1 Properties of Trust

Few research initiatives described various properties of trust some of which are Subjectivity, Dynamicity, and Context Dependency [1]. Trust by nature is subjective because it depends on user's opinion and it is based on personal perspective and preference. However, assessing trust objectively depends on real evidenced measurements, which make it challenging to achieve due mainly to two factors: incompleteness and uncertainty. Subjective assessment is usually studied using probability set and fuzzy set techniques [2]. Another property of trust is dynamicity, where the trust is subject to time elapse, amount of interaction, external factors like authority control and contract rules, and even physical resource capabilities decay over time. This necessitates the periodic refreshment of trust evaluation. Trust also is context depended because an entity can be trusted in a service domain but not in another. This property is modeled in various works in the literature as in [2–4].

2.2 Trust Model Approaches

Many classifications of Trust model for clouds were proposed in the literature. Authors in [5] described four main categories: self-managed case-based, SLA-based, broker-based, and reputation-based approaches. Other classification schemes relied either on the user or provider perspectives, or both, in building trust model such as in [6]. While in [7] trust models were classified into policy, reputation, recommendation, and prediction, the necessity of prediction models arise when there is no previous communication with the cloud service provider. Additionally, a reputation-based trust model is based on the opinions of other users towards service providers. We further classify reputation-based models into service quality-based and resource quality-based models. The service quality-based model performs trust assessment based on the Quality of Service of the cloud. However, the resource quality-based model relies on the cloud resources quality and availability to evaluate the trust.

Many trust model approaches relied on previous experience with the service provider. Authors in [8] built trust score evaluation based on historical records were the Last-K algorithm is adopted. Nevertheless, this method could decrease accuracy because of the limited number of used quality attributes. Other approaches adopted game theory to evaluate trust like in [9–11].

Trust model based on service quality reputation has been proposed in [12], where kept information about service providers are kept in a registry using a discovery system. The credibility of a service provider was evaluated using the ratio of the period of time over which a service is provided to the number of times the service is offered to evaluate. In [13], the trust score of a cloud resource is evaluated based on multiple QoCs attributes. The weights were manually and evenly distributed, so it was inflexible to user quality preferences for services. In the context of Big Data and cloud computing, authors in [3] suggested a category-based context-aware and recommendation incentive-based reputation mechanism to enhance the accuracy and protect data against attacks. Authors in [14] suggested a trust framework for cloud service selection that uses QoCs monitoring and feedback ratings in trust assessment.

Prediction-based trust models typically use statistical techniques for trustworthiness evaluation and prediction. In [7], they study the capabilities and the historical reputation of the service provider and predict its future behavior. These approaches use Fuzzy logic, Bayesian inference, or regression models to estimate the trust of service providers calculated as the probability of providing satisfactory QoCS to users [15]. These models are usually used when there is no previous historical interaction with the cloud service provider. They are also resilient to false reputation attacks especially the logistic regression models that are known to detect outlier values [16]. Bayesian inference is widely used as it considers trust as a probability distribution and is simple with strong statistical basis. However, the belief discounting technique is resilient to false attacks [15]. The fuzzy logic uses approximation to evaluation trust based on ranges between 0 and 1 rather than binary sets. It is widely used despite it incur some implementation complexity and low malicious behavior detection [2].

2.3 Trust Score Computation Approaches

A simple way to evaluate reputation scores is to calculate the difference between the number of positive ratings and the number of negative ratings, which was used in eBay's reputation forum [17]. Yet, this approach might give weak results. A refined method was proposed by some commercial websites such as Epinions and Amazon, where they compute the average of all the ratings. Other approach suggested using a weighted average of all the ratings based on the rater's age, credibility, and difference of the rate value to existing ratings. Similar approach was also used in [18]. Other computational reputation models include Bayesian Systems, Regression Analysis, Belief Models, Fuzzy Models, and Flow Models. However, not all of the aforementioned approaches are used for cloud provider trust evaluation because of unsuitability or simply untried.

The different computation methods are also associated with how the trust scores are scaled. The different scales for trust that are represented in literature include binary, discrete, nominal scale, and continuous values [1]. One problem with several trust

score evaluation methods is that they are based on sophisticated and time-consuming mathematical models. This is unsuitable for a Big Data environment with its own special characteristics (multi-Vs). Most of the aforementioned trust models are non-dynamic in nature and unsuitable for Big Data and the cloud environment. Some base their trust only on reputation, which can be misleading especially if the users are untrustworthy or subjective. Other trust models have used local trust and recommendation trust using weights that are not necessarily dynamic.

3 Trust Evaluation Model

3.1 Problem Definition

In this section, we describe the trust evaluation problem in competing cloud environment as follows: a user wants to select a Cloud Service Provider (CP) to execute some Big Data processing task. Given a history of previous service interactions received from members of community, the user will predict whether CP_i is trustworthy or not. We define a trustworthy CP as being able to satisfy a set of QoCSs. The goal is to reach a high prediction accuracy.

For each service interaction with CP_i at time t, a record containing the observed quality level of this service y^t_k by user k with respect to a set of quality attributes a^t_{ki} that is a real value [0,1]; where:

$$\mathrm{CP} = \{cp_i | i = 1, 2, 3, \ldots n\} \tag{1}$$

$$\mathrm{A} = \{a_j | j = 1, 2, 3, \ldots m\} \tag{2}$$

$$\mathrm{P^t} = \{p_1, p_2, p_3 \ldots p_m\} \tag{3}$$

where t is the time stamp of the observed service transaction, cp_1, cp_2 … cp_n are the possible n alternative cloud service providers CPs available to the user k, a_1, a_2,…, a_m represent QoCS attributes (criteria) such as reliability, availability, and throughput. p_1, p_2,…, p_m represent the performance level of a_1, a_2,…, a_m respectively.

Then, *trust* is the score that CP_i will achieve according to set of QoCS at time t described by p^t vector. Let $y^t_i = y^t_{ki} \cup \{y^t_{ui}, k \neq u\}$ where y^t_{ui} is an observation of neighbor u about a prior service experience with CP_i provided to user k. The observation record is in the form of $\{P^t, y^t\}$ specifying the performance of each quality attribute at time t. Let $y_i = \{y^t_i, t = 1, \ldots, N\}$ represent the set of observations gathered by a user k which includes both self-experience and collected observations from neighbors in [0, N]. And, let $\boldsymbol{p} = \{P^t, t = 1, \ldots, N\}$ be the corresponding performance level of the quality attributes in [0, N].

We suggest to use Multiple Linear Regression (MLR) to solve this problem and model the relationship between the trust score which we consider the dependent variable y and some explanatory (also named independent) variables p using a linear function of the independent variables [19].

$$E\left[y_i^t | p_i\right] = \beta_0 + \sum_{i=1}^{m} \beta_i P_i + \varepsilon \tag{4}$$

where $\beta i = [\beta_i,\ i = 1, 2, 3, \ldots m]$ is a column vector of coefficients that are estimated values from the available data, and ε is the 'noise' which is a random variable having an independent normal distribution with mean equals to zero and unknown constant standard deviation σ.

We estimate the values for β_i coefficients by minimizing the sum of squares of differences between the predicted values and the observed values in the data given by:

$$\sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_m x_{im})^2 \tag{5}$$

Let the ordinary least squares (OLS) $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_m$ be the optimized coefficients that minimize Eq. 5. Then we substitute the computed values in the linear regression model in Eq. 4 to predict the trust score for one CP according to the following:

$$\hat{y} = \hat{\beta}_0 + \sum_{i=1}^{m} \hat{\beta}_i P_i \tag{6}$$

To summarize, history experience $\{p, y_i\}$ is a collection of self-experience QoS performance of CP_i and reputation provided by neighbors upon their experience dealing with CP_i. We perform the multiple linear regression processing for each CP calculating the expected $\hat{y}$. The selected CP would be the one with the highest $\hat{y}_i$ value, i.e. the one with highest predicted trust score, which means the highest probability of providing satisfactory QoS performance. The algorithm shown in Fig. 1 describes the CP selection process according to trust score prediction using MLR algorithm. A trust score is predicted for each CP_i. The algorithm then recommends a CP_i having the highest score.

Algorithm 1 Multiple Linear Regression for CSPs Trust Score Prediction

Input: *CSPList* //List of CSPs
CSPServiceLog //Service Log of all CSPs
ReqAttrVals //list of Required QoS attributes

Output: CSP with Highest Predicted Trust Score

```
1: procedure PredictCPTrust(CSPList, CSPServiceLog,ReqAttrVals)
2:     for all csp ∈ CSPList do
3:         attScore ← 0
4:         Evaluate Bs coefficients according to Eq.5 and Eq.6
5:         for attLabel ← 1, nAttributes do //in ReqAttrVals
6:             attScore ← attScore + ReqAttrVals[attLabel] * B[attLabel]
7:         end for
8:         CSPListScore[csp] ← attScore
9:     end for
10:    return max(CSPListScore)
11: end procedure=0
```

Fig. 1. MLR algorithm for cloud service provider (CP) selection

3.2 Community Management

Our trust evaluation scheme depends on the CP's reputation within the community neighborhood. Initially it requires establishing a degree of trust towards information providers, and then the neighbors need to be motivated and willing to offer reputation information. Hence, we propose a community management system to facilitate the aforementioned requirements. Community is defined in the Oxford dictionary as "the condition of sharing or having certain attitudes and interests in common". With this viewpoint, the community members dealt with the possibility of acquiring CP reputation information from other community members. Many Community management initiatives were presented in the literature as in [20, 21]. An important issue in community management would be the adaptability to the cloud environment dynamic changes and being robust against false information or malicious attacks. We propose a third party entity to maintain a database of community members' information. A user will send request to join the community, and when accepted, the new member is provided with an identification number.

4 Trust Prediction Scheme

Figure 2 describes the set of entities involved in our trust prediction system.

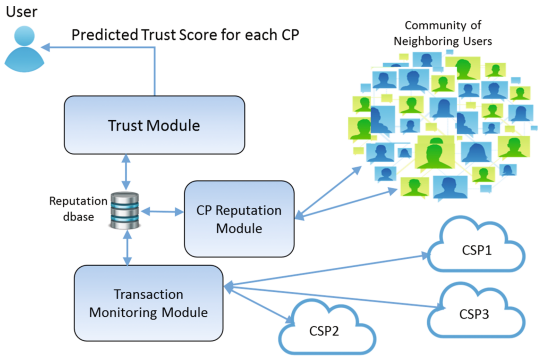

Fig. 2. Trust prediction scheme

Trust Module: It is responsible for analyzing the reputation database to predict the trust score for each cloud provider, and producing a selection decision to the user using the algorithm explained in Sect. 3.1. It generates a predicted trust score for each cloud service provider (CSP) from the logs containing QoS performance and trust score generated by neighbors. The user will choose the CSP having the highest probability of giving a satisfactory QoS.

Transaction Monitoring Module: It monitors all communications with other cloud providers and logs the performance information. A record for each communication transaction exchanged between the user and the cloud provider is logged to the

database called reputation database. This record contains QoS evidence that can help to evaluate a cloud provider's trust score. This information might contain for example, the invocation time, data size, response time, cost and distance between the user and cloud and success status (success or fail).

CP Reputation Module: This module is responsible for sending requests to other neighboring users asking for their own previous experience with other CSPs. In addition, it handles replies received. The request message contains the list of the CSPs to be evaluated. Each reply message contains a list of cloud providers; their QoS performance and their trust scores calculated by the neighboring users. It also analyzes all the reply messages and stores this information in the *Reputation* database and is eventually communicated to the ***Trust module*** for the final trust evaluation. The request messages are sent periodically and the reply messages are collected during this time period. The reputation database is updated whenever a reply message is received.

Reputation Database: It is a local database containing the self-experience and the collected logs from neighbors who were asked to provide their own historical experience with each CSP. Each log contains QoS performance values and the trust score. For scalability reasons, we keep the most recent transaction logs.

5 Implementation and Experimentations

In this section, we describe the experimentations we have conducted to evaluate our proposed trust model. We explain experimental setup, and then we describe the simulator system including all modules.

5.1 Trust Prediction Implementations

The following is the implementation details of the main components involved in our trust prediction model which we have developed in Java to test our proposed trust model. Our simulator implemented all modules described in Sect. 4 including user modules, which are the trust module, CP reputation manager, transaction monitoring module, cloud provider's components, as well as neighbor components (e.g., other users). The simulation generates database logs that are analyzed using Weka MLR to predict the trust scores for each CP.

We considered the following default simulation parameters: Number of cloud providers: 1 to 50, number of nodes within each cloud: 1 to 100, cloud provider's properties: proximity, average node performance and unit storage price, node properties: available resources, memory, disk space, processing power, round trip delay (RT) and bandwidth, QoCS attributes: data size, distance, cost, response time, availability and confidence, and number of community members: 3 to 100 neighbors.

All statistical results were obtained using R language and the packages MASS, DAAG and RELIMPO.

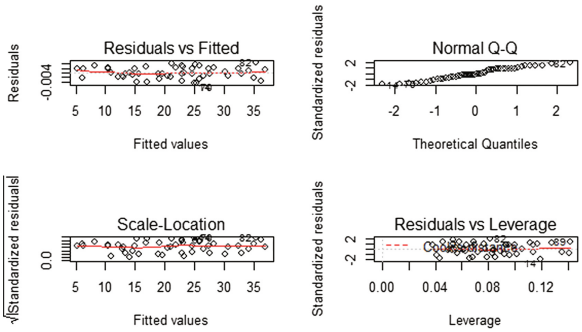

Fig. 3. The regression residuals plot

5.2 Experiments

In this experiment, we generate 50 observations from one provider of the dependent variable Trust denoted by Y and *six* explanatory variables data size (X_1), distance (X_2), availability (X_3), response time (X_4), confidence (X_5) and cost (X_6). First, the variable cost can't be included in the model generated by one provider. It can only be used to compare between different providers. We tested the correlation between the explanatory variables and the response variable, and we can clearly conclude that the correlations are significant with all independent variables except the confidence variable (X_5). Also, we note that the data size and the response time are highly correlated ($r = 1$). Therefore, the estimated regression equation is expressed by

$$\hat{y} = 0.00631 + 0.0243X_1 + 0.0165X_2 + 0.0194X_3.$$

The three variables have a significant positive effect (all p-values are close to zero). This means that the trust will increase with the increase of each of these explanatory variables. As depicted in Fig. 3, the residuals satisfy the assumptions of normality (p-value for Shapiro-Wilk normality test is 0.1689), constant variance and independence. Using the cross-validation procedure to evaluate the consistency of the estimated regression equation, using three folds, we found that the model can perfectly be used to predict the response variable *trust* as depicted in Fig. 4.

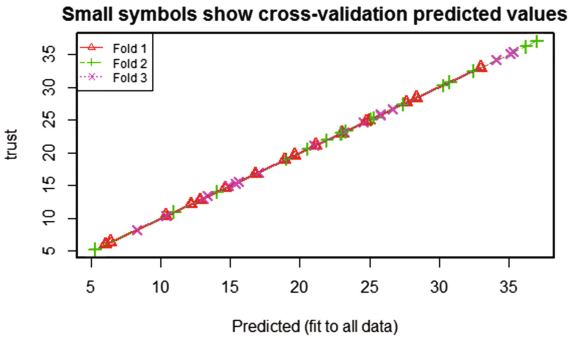

Fig. 4. Cross-validation for predicted values.

By calculating the relative importance for each explanatory variable, we found that the *data size* has the most relative importance for explaining the *trust* variable, roughly more than 62% followed by the distance variable, which has 25% of importance. The three variables explained 100% of the variability of the *trust* variable. To evaluate if the difference between the relative importance for trust is significant, we used the bootstrap procedure to calculate the confidence intervals of the difference between the relative importance of each pair of variables, see Fig. 5.

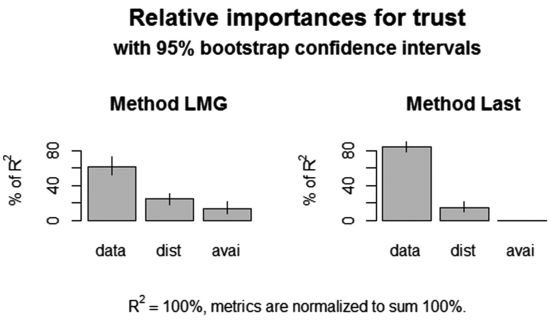

Fig. 5. 95% bootstrap confidence interval of relative importance for the *trust*.

Using the LMG metric, the 95% bootstrap confidence interval (BCI) of the relative importance of data size variable is (51.43%, 71.56%) while using the LAST metric; we note that the coefficient of determination is explained only by the data size and distance variables. In this case, the 95% BCI of the relative importance of data size variable is (78.29%, 89.32%).

6 Conclusion

In this paper, we proposed a Trust model for processing Big Data over different clouds. The model applies the MLR to predict trust scores for different cloud providers. Trust is evaluated based on evidenced information collected about cloud resources availability, past experiences with the cloud provider, and the reputation collected from other users experienced with the same cloud services. The trust model we have developed supports dynamic trust score calculation and update, provides credibility validation through community management system, and retrieves dynamically reputation scores. The model has been evaluated with few experiments and the results we have achieved prove that our Trust model exhibits high prediction accuracy. To evaluate the prediction accuracy, the consistency of the estimated regression equation, and the trust significance, we used the cross-validation method. As a result, we found that the model can perfectly be used to predict the response variable trust. Finally, we estimated and compared the relative importance of each explanatory variable in the model using the bootstrap confidence intervals for the difference between the relative importance of each pair of variables. We found that the data size variable explains the largest relative importance in the proposed trust model followed by the distance variable.

References

1. Cho, J.-H., Chan, K., Adali, S.: A survey on trust modeling. ACM Comput. Surv. (CSUR) **48**(2), 28 (2015)
2. Kanwal, A., Masood, R., Shibl, M.A.: Taxonomy for trust models in cloud computing. Comput. J. **58**, 601–626 (2014)
3. Lin, H., Hu, J., Liu, J., Xu, L., Wu, Y.: A context aware reputation mechanism for enhancing Big Data veracity in mobile cloud computing. In: IEEE International Conference on Computer and Information Technology, Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing, Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM) (2015)
4. Malacka, O., Samek, J., Zboril, F.: Event driven multi-context trust model. In: 2010 Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (2010)
5. Hussain, W., Hussain, F.K., Hussain, O.K.: Maintaining trust in cloud computing through SLA monitoring. In: International Conference on Neural Information Processing (2014)
6. Wu, J.-B.: A trust evaluation model for web service with domain distinction. Int. J. Granul. Comput. Rough Sets Intell. Syst. **2**(4), 273–280 (2012)
7. Noor, T.H., Sheng, Q.Z., Zeadally, S., Yu, J.: Trust management of services in cloud environments: obstacles and solutions. ACM Comput. Surv. (CSUR) **46**(1), 12 (2013)
8. Qi, L., Dou, W., Zhou, Y., Yu, J., Hu, C.: A context-aware service evaluation approach over Big Data for cloud applications. IEEE Trans. Cloud Comput. **PP**(99), 1
9. Gokulnath, K., Uthariaraj, R.: Game theory based trust model for cloud environment. Sci. World J. (2015)
10. Yahyaoui, H.: A trust-based game theoretical model for web services collaboration. Knowl. Based Syst. **27**, 162–169 (2012)
11. Hassan, M.M., Abdullah-Al-Wadu, M., Almogren, A., Rahman, S.K., Alelaiwi, A., Alamri, A., Hamid, M.: QoS and trust-aware coalition formation game in data-intensive cloud federations. Concurr. Comput. Pract. Exp. **28**, 2889–2905 (2015)
12. Muchahari, M.K., Sinha, S.K.: A new trust management architecture for cloud computing environment. In: IEEE International Symposium on Cloud and Services Computing (ISCOS) (2012)
13. Kim, H., Lee, H., Kim, W., Kim, Y.: A trust evaluation model for QoS guarantee in cloud systems. Int. J. Grid Distrib. Comput. **3**(1), 1–10 (2010)
14. Tang, M., Dai, X., Liu, J., Chen, J.: Towards a trust evaluation middleware for cloud service selection. Future Gener. Comput. Syst. **74**, 302–312 (2016)
15. Guo, J., Chen, R.: A classification of trust computation models for service-oriented internet of things systems. In: 2015 IEEE International Conference on Services Computing (SCC) (2015)
16. Wang, Y., Lu, Y.-C., Chen, I.-R., Cho, J.-H., Swami, A., Lu, C.-T.: LogitTrust: a logit regression-based trust model for mobile ad hoc networks (2015)
17. Resnick, P., Zeckhauser, R.: Trust among strangers in internet transactions: empirical analysis of eBay's reputation system. Econ. Internet E-commer. **11**(2), 23–25 (2002)
18. Nitti, M., Girau, R., Atzori, L.: Trustworthiness management in the social internet of things. IEEE Trans. Knowl. Data Eng. **26**(5), 1253–1266 (2014)

19. Freedman, D.A.: Statistical Models: Theory and Practice. Cambridge University Press, Cambridge (2009)
20. He, D., Peng, Z., Hong, L., Zhang, Y.: A social reputation management for web communities. In: International Conference on Web-Age Information Management (2011)
21. Gutowska, A., Sloane, A.: Evaluation of reputation metric for the B2C e-Commerce reputation system. In: WEBIST (2009)