Using Statistical Tests in a Trust Model

Francisco Javier Nieto
ATOS Research and Innovation
Atos Origin
Bilbao, Spain
Francisco.nieto@atosresearch.eu

Abstract—**Nowadays, it is widely recognized that trust is a key aspect of services when integrating them in enterprise environments. This paper presents a solution which takes advantage of the data available by carrying out statistical hypothesis tests in order to evaluate some aspects which are directly related to a service trust. As a result, the application of some tests improves the accuracy and increases the robustness of trust models.**

Keywords-trust; services; statistical tests; model

I. Introduction

As services have become the key for enabling interoperability between enterprises and systems in general, it is necessary to guarantee that interactions are performed properly. It is easy to imagine a scenario where a cluster of companies work together in order to obtain a final product (i.e., suppliers and providers interacting in the automotive domain for producing cars, where many external services are used, such as currency converters or even traffic information providers for logistics).

Services may become very important pieces integrated in business processes or in systems which require that third parties provide functionalities in a professional way or, at least, fulfilling a minimum quality levels in a secure way. For this reason, it is necessary to include mechanisms which guarantee that services provide functionalities as expected and that they deal internally with the information in a confidential and secure way.

The analysis performed in [1] already revealed that there are two kinds of mechanisms: hard security mechanisms and soft security mechanisms. In this case, this paper is focused on a soft security mechanism for calculating trust, understanding it as the belief in the reliability, truth and capability of the service. Such a solution could be used by service discovery tools, as a way to filter services to be listed as candidates.

Despite of the existence of several solutions and the availability of a lot of data about services, no formal analysis is performed about the data gathered as a mean to improve the accuracy and robustness of trust models.

Information about a service can be provided from users who invoked it, but nowadays there are other sources such as monitoring tools (gathering information each invocation, like response time or availability) and platforms collaborating in federations. The presented approach aims at exploiting all this information in a trust model by applying several statistical hypothesis tests. Depending on the data available and the test, it is possible to determine an evaluation of concrete aspects which have great importance upon the global trust of a service, so these evaluations will be aggregated later to other aspects in a higher level model in order to obtain an accurate enough measure of the trust.

The paper is structured as follows: Section 2 provides a vision of the related work in the area, while Section 3 describes the main objectives on which the approach is based. Section 4 will describe the usage of tests for checking agreements fulfillment, Section 5 presents those tests used for evaluating the successfulness of a new release and Section 6 is focused on how to apply the tests when aggregating opinions. Finally, Section 7 presents a set of conclusions and future work.

II. State of the Art

There are multiple initiatives which have analyzed the best way to aggregate users' opinions in order to determine the trust and the reputation of services and systems. Each solution depends on two factors: the kind of information to be used (a general opinion or if there are some aspects to be aggregated) and the way to calculate the aggregation of results.

Usually, reputation is used as the main (or unique) measure for trust, although more aspects can be taken into account and so it is in some of the solutions proposed. Some of the analyzed models are ServiceTrust [2], RateWeb [3] and a model defined in the COIN project [4]. There are also other former models but which propose interesting ideas such as NICE [5], REGRET [6] or Afras [7].

While Afras is a system based on fuzzy logic representing the degree of satisfaction (which aggregates opinions depending on weights, giving more importance to the latest ones), NICE is more oriented to peer-to-peer (P2P) networks, where each node keeps a cookie (with value between 0 and 1) with information about performed transactions. When a node has no direct information about another one, it infers it using others' cookies by means of an oriented weighted graph. In the case of REGRET, a weighted average is performed based on the time and it uses concepts semantically related for calculating the trust.

Latest models are more complex in which refers to the calculation. While ServiceTrust aggregates users' opinions using the function obtained from probabilistic distribution (based on a normal distribution), RateWeb is focused on a weighted average taking into account users' credibility and using a Hidden Markov Model as a way for inferring expected opinions when the number of opinions received is low, since the result would not be very accurate.

These models lack the usage of more information sources (not only opinions, but more aspects) and their robustness is limited to the ability to filter some of the inputs received, without performing a deep analysis of the data being interchanged by all the parties involved. The main challenge is to improve the accuracy of the trust calculation while the robustness of the model is increased at the data level, in order to resist any malicious attack trying to alter the trust calculation.

As a result, [4] proposes a complex and wide model for services, using several heterogeneous aspects which have to be calculated in different ways (especial averages, exponential smoothing, fuzzy logic, etc…). It uses a fuzzy set as the mean to represent the aspects values (see Figure 1) and the main aspects are aggregated by using a weighted average which is based on a weight given by the model administrator (as a way to customize it), time and semantic relationships between aspects (if an aspect is directly related with another one updated recently, it is expected to be more important and represent the actual behavior of the service).

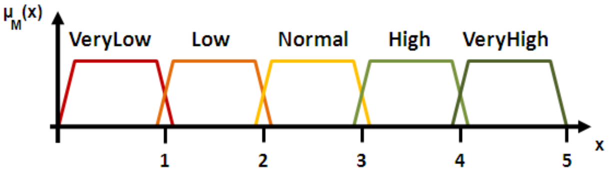


Figure 1. Linguistic terms and their membership function.

While [4] presented a global vision of the trust model and global aggregation functions, this paper presents some of the concrete calculations (closer to the implementation level) performed for some of the aspects, which are based on statistical hypothesis tests, as a tool to exploit the data available from monitoring tools, from external platforms and from users.

III. Main Objectives

As it is possible to obtain a lot of information from different sources, there is the opportunity to exploit it in an adequate way in order to determine the concrete value for some aspects which affect the global trust of a service. These aspects will be key enablers for the improvement of accuracy in trust calculation.

There are two main objectives which are fulfilled by using statistical tests in the trust model:

- Calculate aspects which are not available in other models, as a way to have a better idea of the behavior of the service and its provider, and as a way to evaluate aspects which are important for users but which are very difficult to control and monitor.
- Improve the robustness of the model at the data level, as tests give a good idea of what is going on at a general level with the service, taking many data from several sources, improving the accuracy of the model at the same time.

In order to achieve these principles, next sections propose the way to exploit the information available, which will be provided from a monitoring tool (available in the implementation), from users (gathered through simple forms) and from external platforms, which may be organized in federations as well.

IV. Agreements Fulfillment

The model presented in [4] proposes an aspect which compares monitored values with those agreements made, in order to determine whether a service is behaving as expected, according to contractual commitments.

The idea is to evaluate whether those agreements related to Quality of Service (QoS) parameters (such as response time, availability, etc.) and Trust Level Agreements (TLAs) are fulfilled as expected when users interact with the service. The principle of the aspect is that a service provider cannot be trusted if the agreements are not being fulfilled and the services provided are not stable enough, as expected in business environments.

There are two main parameters to take into account, as defined in the previously mentioned model:

- The stability of the service – Sometimes, the service may behave very well but others it may be behaving wrong (because of technical problems or a wrong development), so it is necessary to guarantee that the service is stable in general, with respect to the agreed QoS and published non-functional properties.
- The fulfillment of agreements – It is mandatory to compare the measured values, obtained from monitoring tools, with the agreed ones. Not only the last value measured will be used, but some measures as well, as a way to give a general view of the service behavior.

There are concrete statistical hypothesis tests which are designed to analyze the variance and average of a set of measured values. Moreover, the inputs of these tests are continuous variables, instead of categories, a fact which facilitates their usage in the presented context.

While the first test is applied for confirming the stability in the variance (a requirement for applying the second test), the second one compares the average of measured values with the expected value for each parameter.

A. Stability Analysis

One of the questions to be answered is whether the service uses to behave in a similar way each time it is invoked. As a service in normal operation is expected to behave always in similar conditions (if no external factors alter its status), the measures obtained from monitoring tools are expected to represent a 'population' following a normal distribution. Consequently, it is possible to perform a Chi-square test in order to determine whether the variance of the measures taken has a pre-determined value, by taking into account each parameter monitored during the service usage.

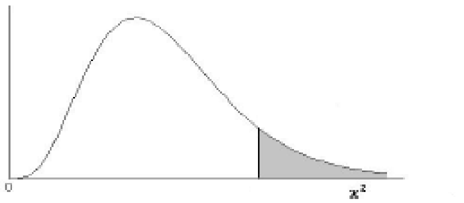


Figure 2. Representation of Chi-square test (Chi-square distribution).

It is possible to determine if the variance is too high by using the null hypothesis H_0: $\delta^2 \leq \delta_0^2$, being δ_0^2 the 5% of the agreed value for the parameter, which means that a variance of 5% in the measured values is acceptable. On the contrary, the alternate hypothesis would represent a variance higher than the 5%. This is a case of unilateral contrast with one degree of freedom.

If the null hypothesis is accepted, then the variance is good, otherwise it is considered bad. One value will be obtained for each parameter in the Service Level Agreements (SLAs) and TLAs with the following equation for one-sample Chi-square test (see [8]).

$$\chi^2 = \frac{(n-1)s^2}{\sigma_0^2} \tag{1}$$

The inputs needed are the expected values for each aspect (for calculating δ value) and the last measures taken by the monitoring tools from each aspect to be evaluated (for obtaining s value). The variable n represents the number of measures taken and used in the test.

B. *Agreements Fulfillment Analysis*

A key factor for determining the trust in a service is the evaluation of agreements fulfillment by the service and the service provider. As there are contractual commitments about the QoS to be offered (by means of agreed SLAs), it is possible to use information from monitoring tools to determine whether the contracts are being fulfilled.

A way for checking the fulfillment of these agreements is to compare the average of the measured parameters with the agreed value. This is done with a contrast 2-tailed Z-test (see [8]), using the agreed value as the expected mean μ_0.

$$z = \frac{\overline{x} - \mu_0}{(\sigma/\sqrt{n})} \tag{2}$$

One-sample Z-test is a statistical test applicable to populations with well known nuisance parameters, such as variance (which can be easily calculated in this case), and which are expected to vary in a normal distribution.

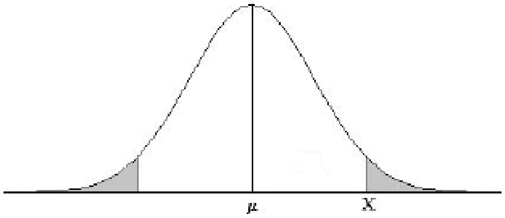


Figure 3. Representation of Z-test (normal distribution).

In this case, the null hypothesis defined is H_0: $\mu = \mu_0$, being μ_0 the agreed value for the parameter, δ will be the variance calculated, and n will depend on the amount of gathered data. The significance level will be 0.05 and the standard deviation will be calculated directly from the measured values during monitoring. This means that the monitored values are expected to have an average of μ_0 with a small variance and very small error which would be accepted. The alternative hypothesis is an average different from μ_0 (H_1: $\mu \neq \mu_0$).

As the Z-test will be calculated once for each parameter under evaluation, there will be a count of not fulfilled parameters, a count of fulfilled parameters and a count of improved parameters (for those values which are better than expected and agreed).

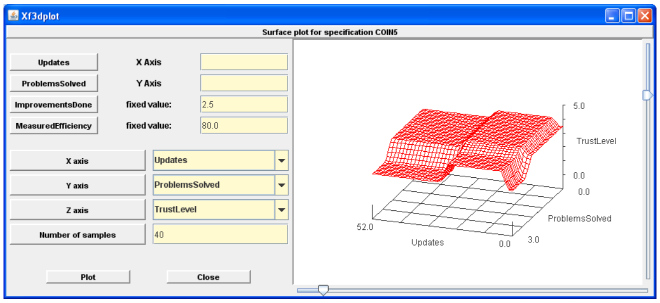
Finally, these counts are taken into account in order to determine how good the service is. In case there are parameters which were not fulfilled, the trust will always be 'low' or 'very low' (if the variance is too high). In case the agreed values for QoS are fulfilled (or better), the result will be 'high' or 'very high' (if the service is stable).

V. Release Improvement

According to the model in [4], the aspect 'Release' is used for determining how good the maintenance of the service is and how effective new releases are. Some key parameters are taken into account:

- Releases periodicity – The usual time between two releases. A periodicity in releases is good, as it means that the service is maintained, although too many releases could be considered bad.
- Releases successfulness – Measure of how better the service is after the last release, according to the parameters measured. It is represented by the number of improvements in non functional aspects which can be measured.
- Problem solved – Number of problems solved in new releases in comparison to previous releases.
- Functionalities added/improved – Number of functionalities improved or added to the service with the new release.

Although these parameters are aggregated by using a fuzzy model which provides the trust value (one of the simulations is showed in Figure 4), it is very useful to apply a statistical test for performing the comparison between previous and current release, according to measured values through monitoring (represented by the parameter "Release successfulness"). This requires letting the service be used for some time before the test can be performed, as it is necessary to gather some data by monitoring the service behavior during some invocations.



Xf3dplot

Surface plot for specification COIN5

Updates — X Axis:

ProblemsSolved — Y Axis:

ImprovementsDone — fixed value: 2.5

MeasuredEfficiency — fixed value: 80.0

X axis: Updates

Y axis: ProblemsSolved

Z axis: TrustLevel

Number of samples: 40

Figure 4. Fuzzy simulation for the 'Release' aspect.

A. Release Successfulness Analysis

This parameter requires confirming that the service has been improved by checking improvements in parameters such as response time, availability and robustness. This can be measured by comparing the averages of measured data for each parameter under evaluation. If measures have improved for, at least, most of the parameters, it can be considered that the release really was successful in the sense that the improvements in the service are clearly observable.

The way to compare averages is performing a Student's t test (to be more concrete, an unpaired t-test, as there are no correspondences between requests) between old monitored measures and new ones (taken after the deployment of the new release).

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{X_1X_2} \cdot \sqrt{\frac{2}{n}}} \tag{3}$$

For the implementation, it is necessary to calculate average values for each group (the group with old values and the one with new values, for X_1 and X_2 averages) and the standard deviation for S. As old values can be filtered using only those closer to the release, the size of the samples will be the same, represented by n. The significance level, which will delimit the critical region, will be again 0.05 (as usually recommended).

As the measures come from the same monitoring tools and their nature and source is the same, it is expected that the variance in both cases will be similar, fulfilling the requirements for performing the test.

In this case, the null hypothesis for each test is H_0: "There are no significant differences between the average values of the compared groups" and the alternate hypothesis would be the contrary (H_1: "There are significant differences between the average values of the compared groups"). The first case means that there are not concrete improvements, while the second case requires checking whether the measures taken are better or worse in order to determine the degree of improvement thanks to the new release.

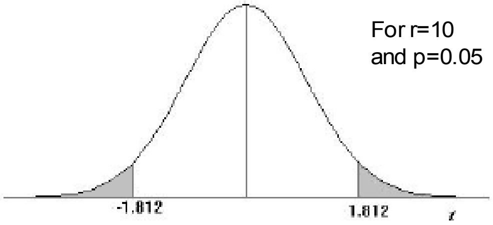


Figure 5. Representation of Student's t test (T Student distribution).

After the test is solved for each measurable non-functional property, the percentage of improved parameters is obtained. For instance, 80% would mean that most of the parameters have been observed to be better than in the previous release. This will be the input for the fuzzy model showed in Figure 4.

B. Non-Functional Properties Checking

The model presented in [4] describes an area about service claims regarding functionalities and non-functional properties which are relevant when the service is new and when there is a new release of the service. Although not included currently in the model, there is the possibility to include an analysis about how much the service fulfills the non-functional properties claimed by the service provider.

Mc Nemar's X^2 (see [8]) can be used for comparing claimed Non-Functional Properties (NFPs) and measured NFPs before and after the release. Columns will be 'fulfilled NFPs' and 'not fulfilled NFPs' after the release, while rows will be the same before the release. As usual, the significance level recommended is 0.05.

This means that the null hypothesis would be H_0: "The number of correct claims about NFPs is invariable before and after the release", while the alternate hypothesis would be represented by H_1: "The number of correct claims about NFPs has varied after the release".

In order to calculate it, information contained in the service description can be used as input, as well as the measures taken by the monitoring tools.

As the number of NFPs is not very high and it is not expected to vary, this calculation may not be very significant in the model proposed in [4], but it could be useful in more complex models where there are claims about many parameters.

VI. Opinions and Feedback

One of the essential inputs in any trust model is the usage of information provided by third parties about the element under evaluation. In the case of the model defined in [4], there are two kind of external inputs used: users' feedback and platforms feedback.

In the case of users' feedback, while users execute a business process, they are requested to evaluate a small set of concrete characteristics of the services used in the process. They just need to fill in a very simple form (an eBay like form) and submit it to the Trust Manager.

On the contrary, in the case of platforms feedback, a web service interface is used for requesting information about a service, with the support of an ontology, which is used for mapping aspects of two different platforms. The external platform may send one or more aspects calculated for the service, depending on the information in its trust model.

The approach presented in [4] is based on a weighted average in which received values are more or less important depending on the credibility of the platform or user who provided the value. This credibility is based on two parameters: coincidence in measures and affiliation with the current platform. Both of them will be multiplied for obtaining the final value of credibility.

Cohen's Kappa (see [9]) will be calculated for checking coincidences in measures, comparing measures of one platform and measures of other platform. Depending on the result, it is possible to determine the credibility of an external platform. This is because, as a platform has direct measures about the behavior of a service, if external entities are giving very different feedback it means that different models are being used or malicious information is being received, although it is possible that our monitoring system has some kind of problem. So the weight will be decreased, but the evaluation received will not be ignored at all.

For each parameter measured by both parties, one table is created. Columns will represent results observed by one platform (how many calls were categorized as 'VeryHigh', 'High', 'Medium', 'Low', 'VeryLow') and rows will represent results observed by the platform requesting for information. Then, kappa index will be calculated.

$$k = \frac{p_a - p_r}{1 - p_r} \qquad (4)$$

By applying the formulas for p values defined in [9] to the values in the matrix, the kappa value is obtained. Values provided by each platform are directly requested using the ontology of the trust model in [4].

According to Landis and Koch [10], the level of agreement in measures can be categorized as follows:

TABLE I. AGREEMENT LEVELS

Kappa	Agreement
< 0	Disagreement
0 – 0,2	Slight agreement
0,2 – 0,4	Fair agreement
0,4 – 0,6	Moderate agreement
0,6 – 0,8	Good agreement
0,8 - 1	Very good agreement

The value received from the kappa calculation is directly used for modifying the weight of the data received, by multiplying it to the affiliation (a number between 0 and 1 which represents the historical relationship between the platform and the rater which provided an evaluation). This means that all the weights will be between 0 and 1.

VII. Conclusions and Future Work

Although there are several models defined for calculating the trust associated to a service, most of them are based only in opinions received from users. Moreover, this feedback is used normally by applying weighted measures and, sometimes, probability distributions, but in no cases statistical analysis is exploited for obtaining the trust value.

Used in the proper way, the statistical tests are an interesting tool in order to determine which values are more important or have more sense, supporting as well the robustness against malicious attacks.

But the more interesting usage of statistical tests comes from their utility for calculating other aspects which enrich the trust model with more information, such as those aspects mentioned in the paper (agreements fulfillment and release analysis). These aspects are not included in other models and statistical tests provide the means to study and predict the service behavior thanks to the information gathered during services invocation.

As the model were the mentioned aspects are included is based on a common fuzzy set (representing data in categories) and as values obtaining for monitoring are data of continuous nature, the number of test to be applied is not restricted, by performing adequate normalization.

Next actions are to analyze how to use other statistical tests in the model as a way to improve it. For instance, in the case of feedback analysis, Fleiss' Kappa could be used for providing a more accurate idea about the trustworthiness and weight of service raters (users or external platforms), detecting whether there is some consensus in the behavior of a service.

Finally, as the model defined in [4] uses fuzzy rules in one of its rounds, it is planned to perform some analysis which will provide information about correlations and relationships between different aspects in the model. This way, rules for robustness will be built according to the results obtained, as a way to provide a more robust model.

Acknowledgment

This research was supported by the COIN project (http://www.coin-ip.eu/) and has been partly funded by the European Commission's IST priority of the 7th Framework Programme under contract number 216256.

References

[1] Jøsang, A., Ismail, R., and Boyd, C.: A survey of trust and reputation systems for online service provision. Decis. Support Syst. 43, 2, pp. 618-644. Mar. 2007

[2] He, Q., Yan, J., Jin, H., and Yang, Y. “ServiceTrust: Supporting Reputation-Oriented Service Selection”. In Proceedings of the 7th international Joint Conference on Service-Oriented Computing (Stockholm, November 24 - 27, 2009). L. Baresi, C. Chi, and J. Suzuki, Eds. Lecture Notes In Computer Science, vol. 5900. Springer-Verlag, Berlin, Heidelberg, 269-284.

[3] Malik, Z., Akbar, I., and Bouguettaya, A. “ Web Services Reputation Assessment Using a Hidden Markov Model”. In Proceedings of the 7th international Joint Conference on Service-Oriented Computing (Stockholm, November 24 - 27, 2009). L. Baresi, C. Chi, and J. Suzuki, Eds. Lecture Notes In Computer Science, vol. 5900. Springer-Verlag, Berlin, Heidelberg, 576-591.

[4] Nieto, F. J. “A Trust Model for Services in Federated Platforms”, Vol. 76 of Lecture Notes in Business Information Processing, Springer Berlin Heidelberg, 2011, pp. 118–131.

[5] S. Lee, R. Sherwood, and B. Bhattacharjee. “Cooperative Peer Groups in NICE”. In IEEE Infocom, San Francisco, CA, Apr. 2003.

[6] Sabater, J. and Sierra, C. “REGRET: A reputation model for gregarious societies”. In Proceedings of the 4th Int. Workshop on Deception, Fraud and Trust in Agent Societies, in the 5th Int. Conference on Autonomous Agents (AGENTS’01), pages 61-69, Montreal, Canada, 2001.

[7] Carbo, J., Molina, J., and Davila, J. “Comparing predictions of SPORAS vs. a Fuzzy Reputation Agent System”. In: 3rd International Conference on Fuzzy Sets and Fuzzy Systems, Interlaken, 2002. pp. 147—153.

[8] “NIST/SEMATECH e-Handbook of Statistical Methods”, http://www.itl.nist.gov/div898/handbook/, 2010 (accessed June 2011).

[9] Sim, J.; Wright, C. C.; "The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements". Physical Therapy 85 (3): 257–268 (2005).

[10] Landis, J. R. and Koch, G. G. "The measurement of observer agreement for categorical data" in Biometrics. Vol. 33, pp. 159–174 (1977).