Reputation System Aggregation and Ageing Factor Selection using Subjective Opinions Classification

Abdelmageed Algamdi
Alexei Lisitsa and Frans Coenen

Department of Computer Science
University of Liverpool,
Liverpool, UK

July 01-02, GSCIT' 2017

Outline

- Cloud Computing
- Service Level Agreement - SLA
- Trust
- Reputation System
- Cloud Trust Protocol - CTP
- Author's Previous work
- Research Objectives
- Assessment System
- Proposed opinion rating classifier
- Updating the trust
  - Using aggregation only
  - Using aggregation with ageing
- Summary
- Future work

Cloud Computing

Cloud Computing

It allows users to remotely share the various resources over the internet.With its low cost, it provides customers with huge memory space and robust computation capability.

Service Level Agreement (SLA)

An SLA can be defined as a contract between the service user and the service provider that contains a description of the agreed-upon service.

Trust and Reputation Systems

Trust in our life

It is a belief that someone or something is reliable, good, honest, effective, etc or to have confidence in someone or something.

Trust in CS and Cloud Computing

- relies upon diverse attributes such as information security, compliance and data governance etc.
- Trust management monitors the quality of service (QoS) and the service level agreement between the user and the service provider and affect a reputation system.eg.Trust in Ubiquitous Computing(Ries S., 2009)

Trust and Reputation Systems

- Reputation
  - is a general belief about a person, such as his or her character. In general, reputation can be used as a source of trust. It is comprised of opinions (a collective picture) about an entity or person.eg.Trust and Reputation Systems(Jsang A, Ismail R, Boyd C., 2007)
  - It calculates the trustworthiness of the service provider based on a self-assessment done by the provider.
  - There are several contributions in order to design a reliable reputation system.

Cloud Trust Protocol - CTP

- The Cloud Trust Protocol (CTP) is a protocol which was proposed by the late Ron Knode in 2010 and licensed for use in 2011 by the Cloud Security Alliance.
- It provides a way for the user to request evidence or certificates from the cloud service provider regarding the operation of a specific service.
- This information gives the user a whole picture about what he should expect while running the service.

Author's Previous work

- In (Algamdi, Abdelmageed et. all , (ICCAT'17)).
  1: Use CTP to get information about the services .
  2: Provide infrastructure to support CTP/assessments requests.
  3: Collecting user/provider opinions from assessment questionnaires.

Research Objectives

Main Idea

Classifying cloud user's opinion so that it can update the initial/latest trust value.

How?

- By proposing a classifier based on barycentric coordinates.
- Updating the trust using aggregation only or aggregation with ageing.

Assessment System

- We assume that the trustworthiness for any service is being represented by a scalar value between 0 and 100. That value increases if the service is satisfied by the SLA agreement and hence satisfies the users demands and decreases if it violates the SLA agreement.
- The case if the SLA agreement is not violated and the user feedback shows that he is unhappy, this mean that the user did the assessment in a wrong way and his opinion now shouldn't affect the old trustworthiness value –untrusted agent. (**Future Work**)

Assessment System

- We use the approach in (Habib, et. all ,2014) to get the cloud provider initial trust value using the CAIQ assessment.
- We use the approach in (Algamdi, Abdelmageed et. all , (ICCAT'17)) to get the consumer opinion using the Smals ICT research group questionnaire.

User Opinion

As shown in (Algamdi, Abdelmageed et. all , (ICCAT'17)), the user binomial opinion obtained from the MCQ questionnaire can be expressed as

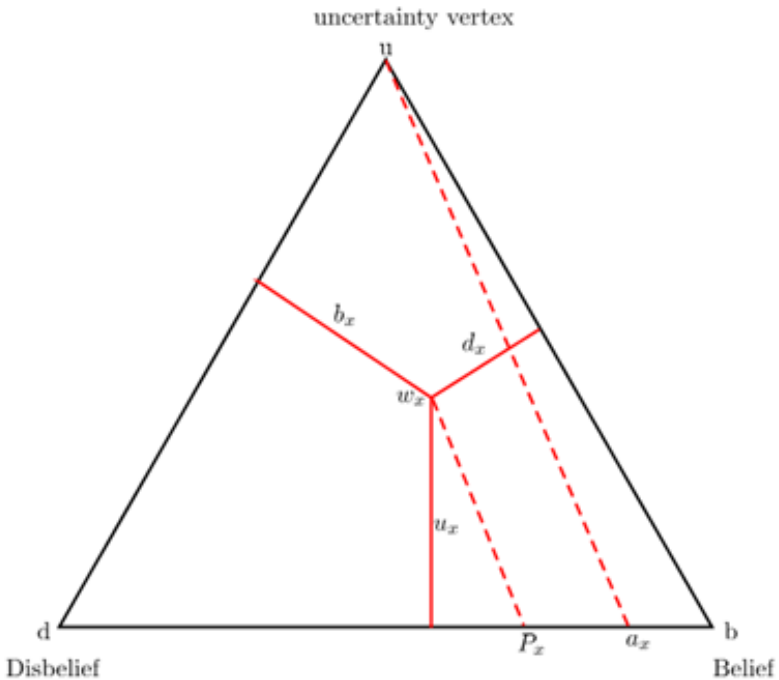
$$\omega_x = (b_x, d_x, u_x, a_x)$$

where:

- b_x : is the belief based on the Yes answers.
- d_x : is the disbelief based on the No answers.
- u_x : is the uncertainty based on the Unknown answers.
- a_x : is the base rate which is $1/2$ in our model.

Opinion visualisation

- The users opinion $\omega_x = (b_x, d_x, u_x, a_x)$ where $b_x + d_x + u_x = 1$ can be visualized using Barycentric coordinates.



Opinion Classification

- Classify the user opinion into one of seven fuzzy rating classes (Very good, Good, Low good, Very bad, Bad, Low bad, and Uncertain).

Opinion Classification

Region	Belief	Disbelief	Uncertainty
Very Good, Certain	$b_x \geq 0.5$	$d_x < 0.5$	$u_x < 0.5$
Good, Certain	$0.25 < b_x < 0.5$	$d_x < 0.25$	$u_x < 0.5$
Very Bad, Certain	$b_x < 0.5$	$d_x \geq 0.5$	$u_x < 0.5$
Bad, Certain	$b_x < 0.25$	$0.25 < d_x < 0.5$	$u_x < 0.5$
Un-named, Certain	$0.25 \leq b_x < 0.5$	$0.25 \leq d_x < 0.5$	$u_x < 0.5$
Very Uncertain	—	—	$u_x \geq 0.5$

Table: Opinion Classification

Opinion Classification

Converting the fuzzy rating classes into scalar factors k:
The value of k is determined as follow and depends on the rating class for the consumer opinion:

- For very good and certain class $k = 1$.
- For good and certain class $k = \frac{1}{2}$.
- For very bad and certain class $k = -1$.
- For bad and certain class class $k = \frac{-1}{2}$.
- For unnamed and certain class $k = \frac{1}{4}$ if $P_x \geq 0.5$ and $K = \frac{-1}{4}$ if $P_x < 0.5$).
- For very uncertain class $k = 0$.

Updating the trust - aggregation only

Let's define:

- $R_{x,(t=0)}$ is the initial rating value (only provider) generated from the provider self-assessment for service x at the beginning ($t = 0$).
- $R^{A}_{x,(t\neq 0)}$ is the old rating value (provider and user A) over time t for service x.
- $R^{A}_{x,(t+1)}$ represents the overall (provider and user A) new accumulated rating value after time period $t + 1$ for service x.
- $R_{x,(t+1)}$ represents the overall (provider and all users) new accumulated rating value after time period $t + 1$ for service x.

Updating the trust - aggregation only

- Let's define aggregation constant $\lambda \in [0, 1]$.
- For any assessment, the update is a k-ratio our of the λ value.

Updating the trust - aggregation only

Assume that the value of previous opinion rating class for those agents that do their first assessment is $k_t = 0$. The new accumulated rating $R^A_{x,(t+1)}$ after time period $t+1$ can be expressed as:

- For the first user assessment:

$$R^A_{x,(t+1)} = \lambda' + R_{x,(t=0)}$$

where:

$$\lambda' = (k_{t+1} - k_t)\lambda,\ 0 \leq \lambda \leq 1$$

- For any user assessment except the first one:

$$R^A_{x,(t+1)} = \lambda' + R^A_{x,(t\neq 0)}$$

where:

$$\lambda' = (k_{t+1} - k_t)\lambda,\ 0 \leq \lambda \leq 1$$

Updating the trust - aggregation only

Let that the symbol $\mathbb{A}$ represents the set of all users did the assessments for the service x. The overall reputation (rating) generated from all users is simply generated from the average overall users' ratings as follows:

$$R_{x,(t+1)} = \frac{\sum_{A \in \mathbb{A}} R^{A}_{x,(t+1)}}{|\mathbb{A}|}$$

We assume that the overall reputation $R_{x,(t+1)}$ has lower bound of 0 and higher bound of 100. If the calculated overall reputation lies out of the boundaries we modify it to lie on the boundaries 0 if smaller and 100 if bigger.

Aggregation Constant Estimation

Main Idea

- We use randomly generated answers (values) for the user questionnaire.
- Use the same approach in our previous work to collect the overall random user opinion and visualize it inside the Barycentric triangle.
- The visualization window will be outputted to the user with the rating classes shown also on the same figure and ask the tester to give a guess for the overall trust value.
- The tester will not be informed with the k-values for each rating class as we want to compare the suggested approach with the human feelings.

Aggregation Constant Estimation

Main Idea

- The tester will be informed with the latest trust as well as the aggregation constant value before giving his estimation.
- We show the visualisation of $n > 0$ random opinions corresponding to n different users assessed the same service and ask the tester for a guess of the new trust value.

Aggregation Constant Estimation

Experimentation steps

1. Generate n random subjective opinions $\omega_{\mathbb{A}} = \{\omega_{A_1}, \omega_{A_2}, ..., \omega_{A_n}\}$ that act as the overall opinions generated from the assessments done by the set of users $\mathbb{A} = \{A_1, A_2, ..., A_n\}$ assessments for a given service x.
2. Find the set of opinions' ratings $K = \{k_1, k_2, , k_n\}$.
3. Visualize the set of random opinions $\omega_{\mathbb{A}}$
4. Choose an initial value of $\lambda \in [0, 1]$.

Aggregation Constant Estimation

Experimentation steps

5. Ask the human tester to give estimation to the final trust value $R^*_{x,(t+1)}$ after showing him the visualization picture, the value of λ and the latest trust just before the process.
6. Let the program calculate the updated trust value using the following equation.

$$R_{x,(t+1)} = \sum_{i \leq n} k_i \times \lambda + R_(x, t)$$

Where,

$R_{x,(t+1)}$ is the updated trust value of the service y

$R_{x,t}$ is the latest trust value of service y before this process

7. Find the absolute error $e = |R_{x,(t+1)} - R_{x,(t+1)} * |$ and store the values (λ, e).

Aggregation Constant Estimation

Experimentation steps

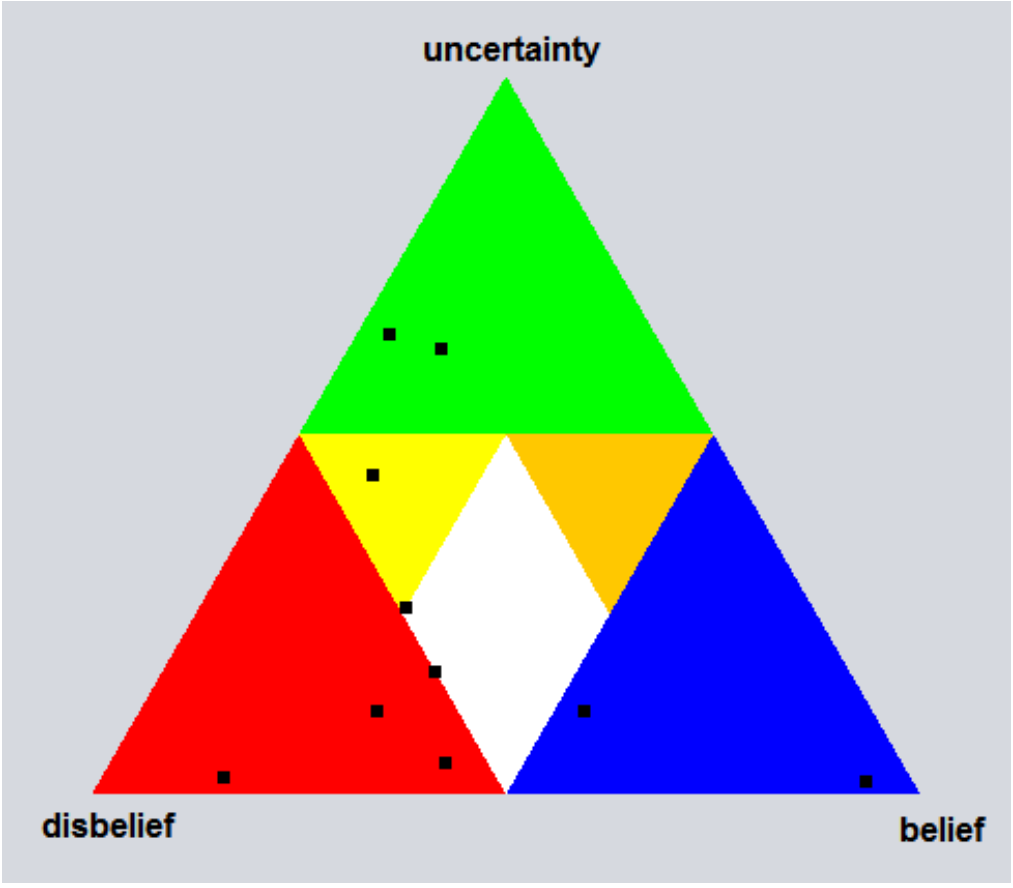
8. Repeat steps 1 to 7 with different values of n and λ.
9. Repeat the whole process from 1 to 8 with $m \geq 1$ testers.
10. We can do this procedure for the values $n \in \{1, 5, 10, 15, 20\} \text{and} \lambda \in \{0.1, 0.2, 0.4, 0.5, 0.8, 1\}$.
11. For every λ find the average absolute error

$$\overline{e_\lambda} = \frac{\sum e_\lambda}{m \times n}$$

1. Select the best λ value that minimizes the average absolute error.

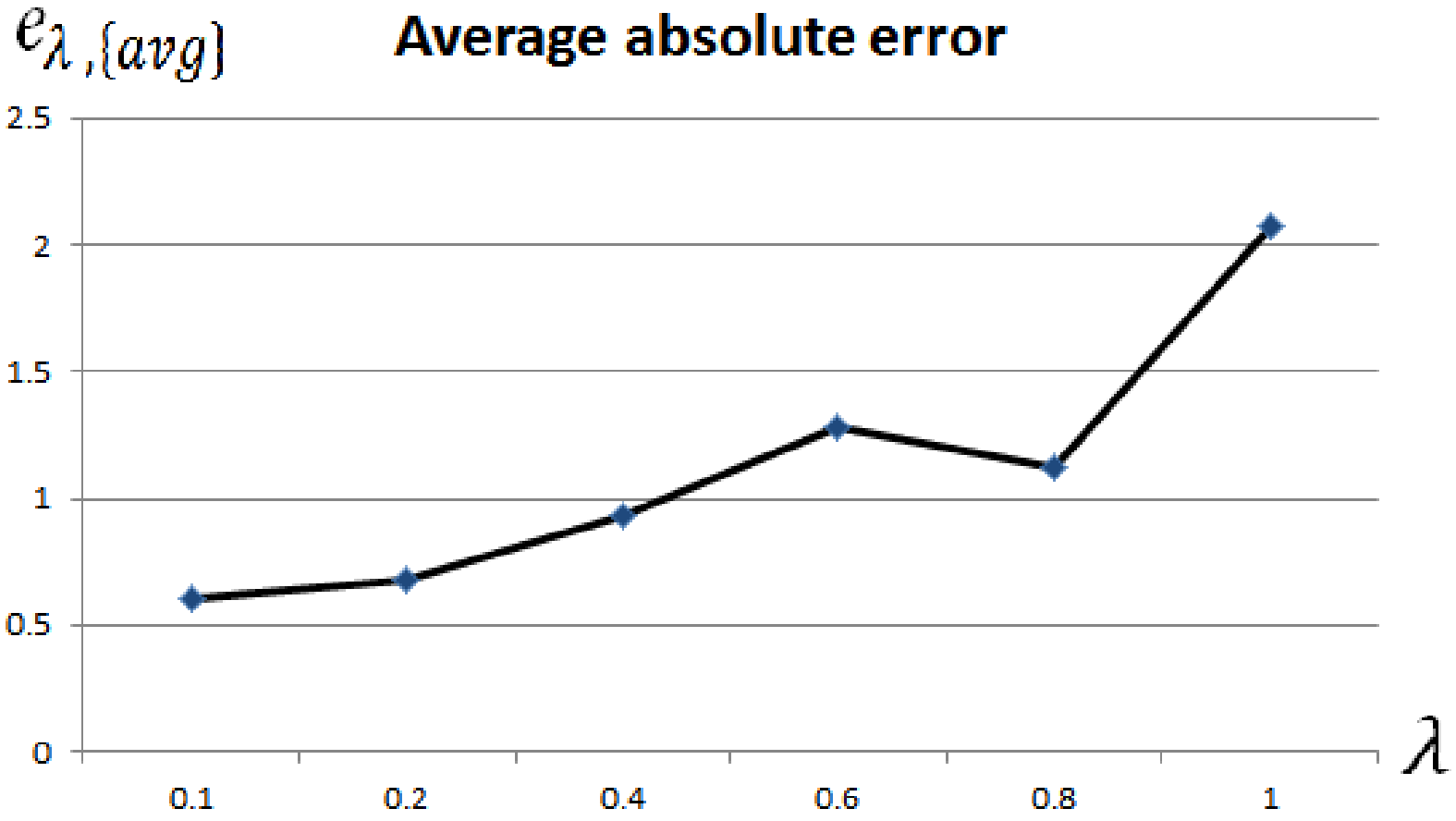
Aggregation Constant Estimation
Example

- Example of 10 different users assessing the same service.



Aggregation Constant Estimation
Results

- The lowest λ value, the minimal average error (best value at $\lambda = 0.1$).



Updating the trust - aggregation with ageing

Let's define:

- Ageing factor $\Lambda \in [0, 1]$.
- For any user assessment, the update is the aggregation between the most recent assessment and aged factor of the user's previous assessments to the same service.
- $\delta^{A}_{x,t+1}$ is the most recent update effect at time $t + 1$ due to user A assessment to the service x.
- $\delta^{A}_{x,t}$ is the update effect for all the history assessments in the time period $[0, t]$ for every user A assessed the service x

Updating the trust - aggregation with ageing

- For the first user assessment:
  There is no assessment history for the user A towards the service x. So, there is no need for doing any form of ageing here in the first user assessment where λ is the aggregation constant.

$$\delta^{A}_{x,t+1} = k_t \lambda$$

Updating the trust - aggregation with ageing

- For any assessment except the first one:
  Now, we have assessment history for the user A so, we should use the ageing factor now.

$$\delta^{A}_{x,t+1} = k_t\lambda + \Lambda\delta^{A}_{x,t}$$

  - For decreasing the effect of the history we use $\Lambda = 0.01$ (very close to 0).
  - For increasing the contribution of the history in the calculation of the current reputation value we use $\Lambda = 0.99$ (very close to 1).

Updating the trust - aggregation with ageing

The overall reputation (rating) generated from all the users $A \in \mathbb{A}$towards the service x where $\mathbb{A}$ is the set of all users did the assessments is the sum of all the updates done by all the users plus the initial trust $R_{x,(t=0)}$ generated from the provider self-assessment. This is shown as follow

$$R_{x,(t+1)} = \sum_{A \in \mathbb{A}} \delta^{A}_{x,t+1} + R_{x,(t=0)}$$

We assume that the overall reputation $R_{x,(t+1)}$ has lower bound of 0 and higher bound of 100. If the calculated overall reputation lies out of the boundaries we modify it to lie on the boundaries 0 if smaller and 100 if bigger.

Ageing Factor Estimation

Main Idea

- Use human testers like the previous experiment.
- Visualise n random opinions to the tester corresponding n different opinions to the same user at different time samples.
- Ask the tester for guessing the update or the new trust value.
- The tester will be informed with the aggregation constant λ as well as the ageing factor Λ but not the k values.

Ageing Factor Estimation
Expiremental Steps

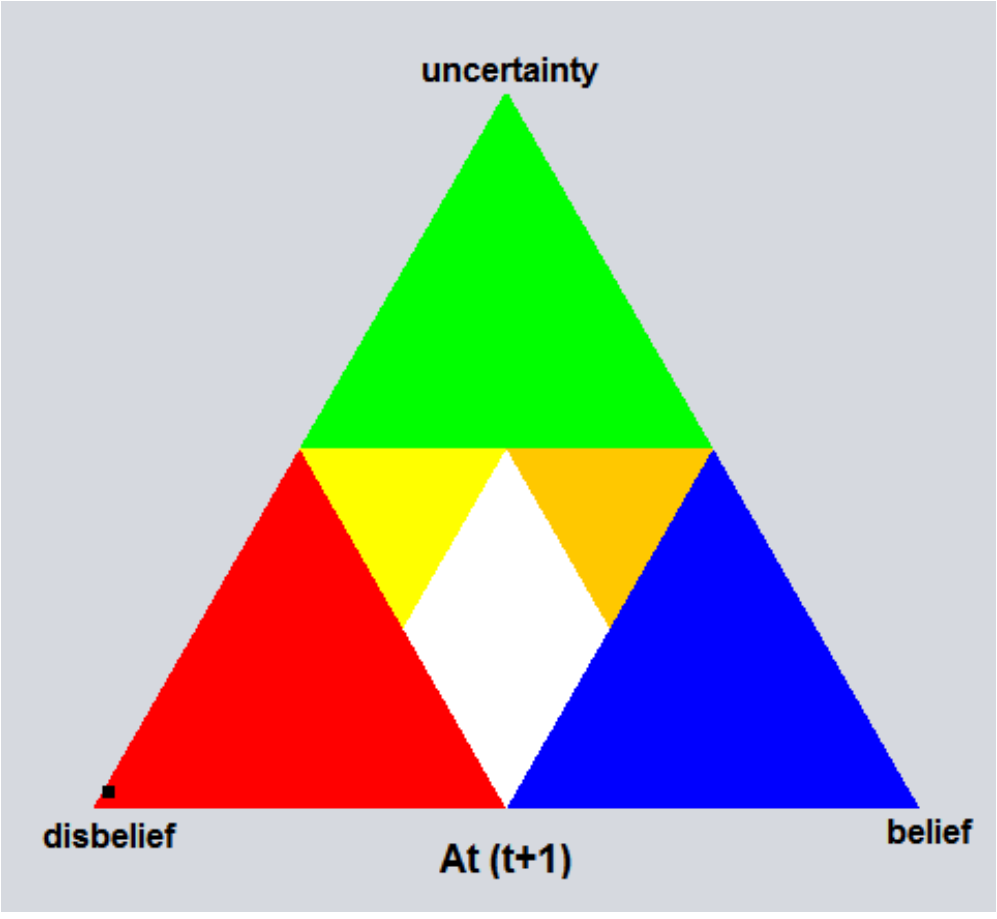
1. Assume that the aggregation constant always $\lambda = 0.1$.
2. We are going to do the following steps for the testers $\{t_1, t_2, , t_m\}$, $m \in \mathbb{N}_{>0}$.
3. We do the following steps for various ageing factor $\Lambda \in \{0.01, 0.2, 0.4, 0.6, 0.8, 0.99\}$.
4. For every tester t_i , we ask him to do n random test trials with every single value of Λ.
5. For every trial, we ask the tester to select positive random integer I which represents how many random opinions will be generated. Each opinion represents a different time frame.

Ageing Factor Estimation
Expiremental Steps

6. We visualize each opinion alone and ask the tester to give estimated value about the updated trust value giving him the latest trust value (the past).
7. We let the program calculate the overall updated trust value.
8. Calculate the absolute error e_Λ between the estimated value and the calculated value.
9. Repeat steps from 5 to 8 for the rest ($l - 1$) trials.
10. Repeat steps from 5 to 9 with all values of Λ.
11. Repeat steps from 5 to 10 with all the other ($m - 1$) testers.
12. After finishing all the testers we calculate for every Λ the average absolute error $\overline{e_\Lambda} = \frac{\sum e_\Lambda}{l \times m}$.
13. Select the best ageing factor Λ with the minimum average absolute error $\overline{e_\Lambda}$.

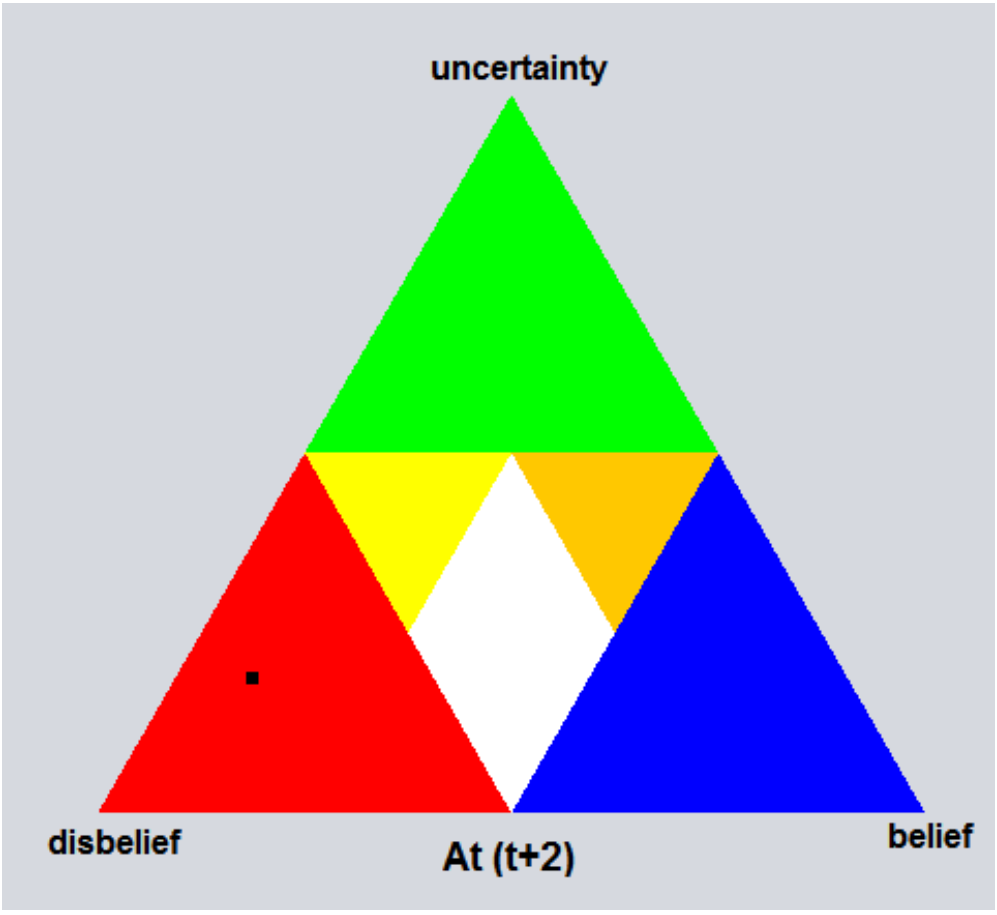
Ageing Factor Estimation
Example

- Example of 3 different user opinions at three different time samples for the same service.



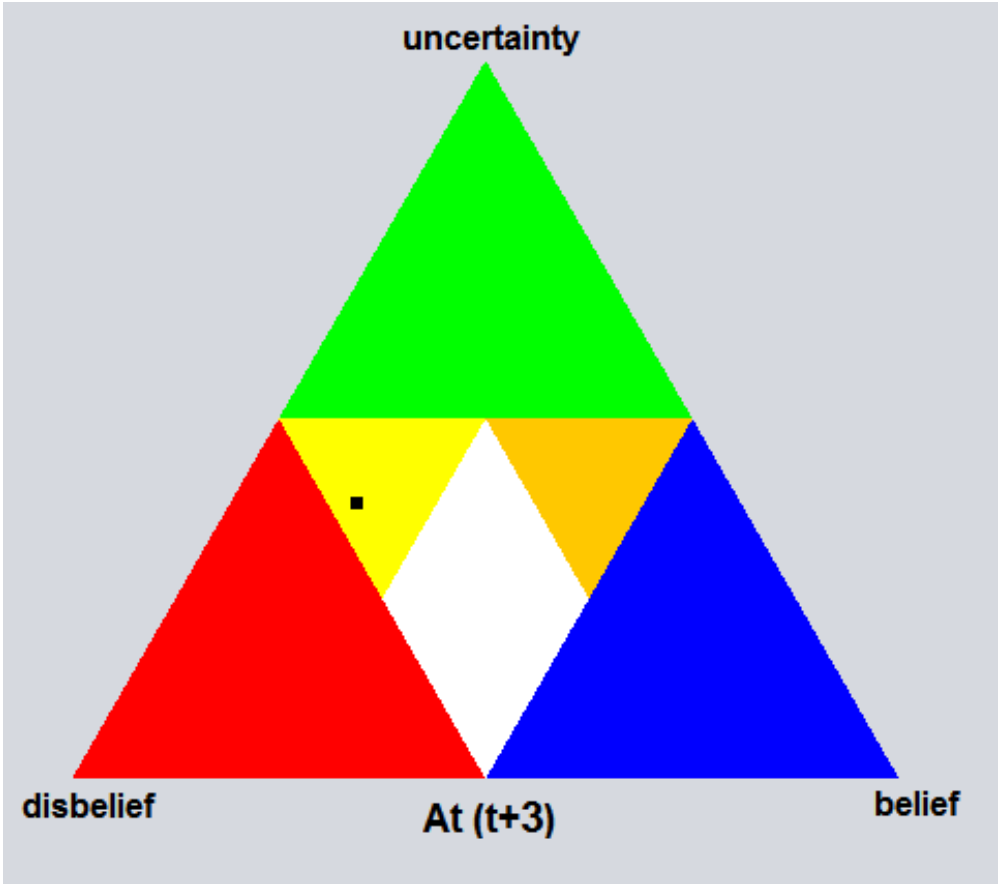
Ageing Factor Estimation
Example

- Example of 3 different user opinions at three different time samples for the same service.



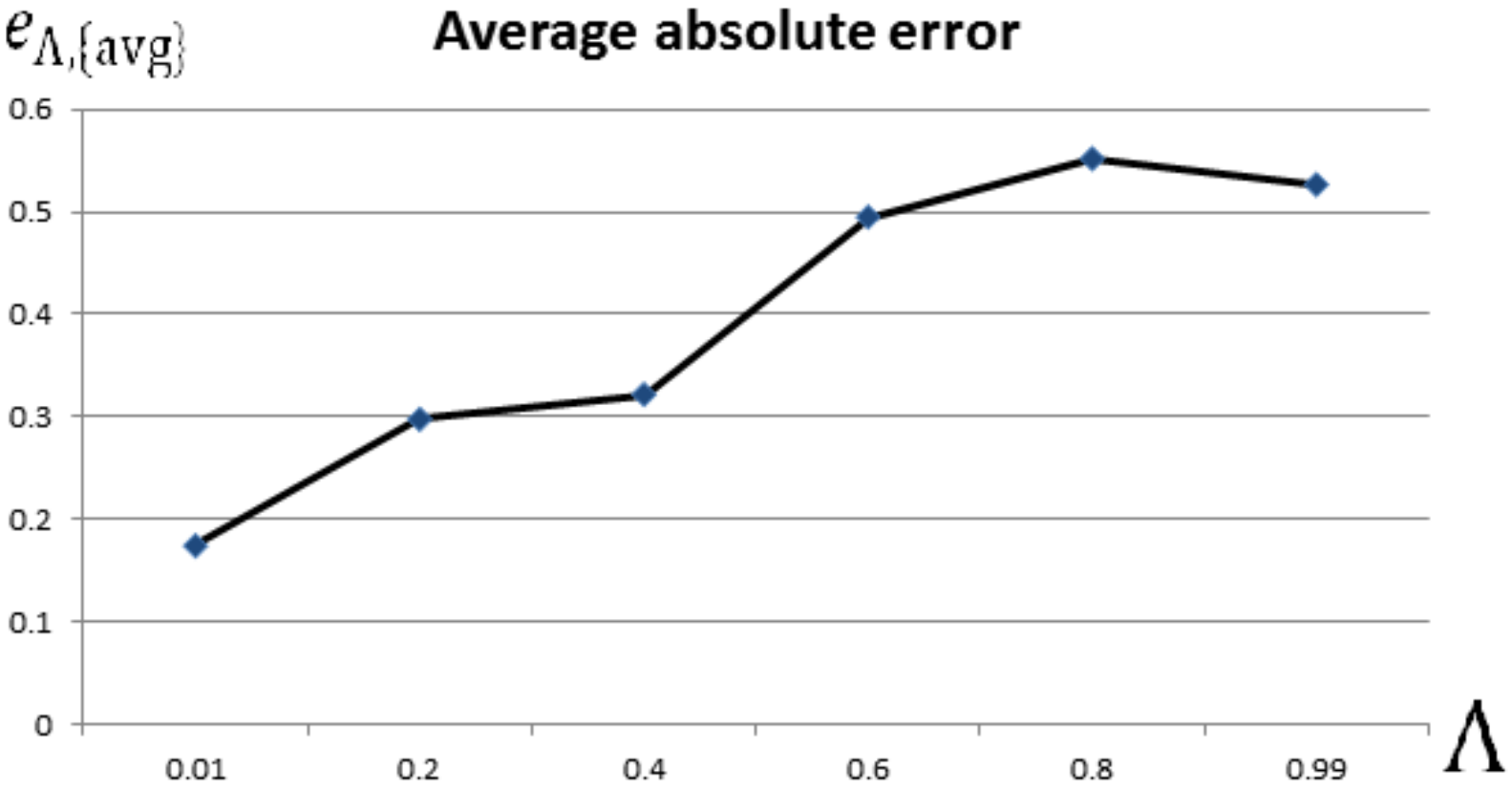
Ageing Factor Estimation
Example

- Example of 3 different user opinions at three different time samples for the same service.



Ageing Factor Estimation
Results

- The lowest Λ value, the minimal average error (best value at $\Lambda = 0.01$).



Summary

1. Subjective binomial opinions can be classified into rating values using the proposed barycentric classifier.
2. For updating the trust using aggregation only, choose small value for the aggregation constant (in our settings, $\lambda = 0.1$).
3. For updating the trust using aggregation with ageing, choose a very small value for the ageing factor (in our settings, $\Lambda = 0.01$).

Future Work

- Solving the problem of removing the untrusted agents before doing the assessments.
- Solving the problem of securing the approach against the malicious users attacks done by hackers.

The End

Thank you for your attention