$$t_\epsilon^{k+1} = C^T t_\epsilon^k \tag{6}$$

until $\|t_\epsilon^{k+1} - t_\epsilon^k\| < \eta$, where η is a specified threshold to determine that we have reached a fix point. As in the eigentrust algorithm, the trust vector t_ϵ converges after a certain amount of iterations. In this way, the trust that ϵ has on i is built aggregating the direct trust distributions between community members and peer i weighted by the trust (initially ignorance) that ϵ has on each community member. The product between matrix C^T and t_ϵ^k is defined, recalling previous definitions, as follows:

$$t_{\epsilon,j}^{k+1} = \bigoplus_{0<i\leq n} \mathbb{T}_{i,j} \otimes \mathbb{T}_{\epsilon,i}^k \tag{7}$$

Finally, if a direct trust distribution is already built between ϵ and j, $\mathbb{T}_{i,j}$, then after each step of the algorithm, $t_{\epsilon,j}^{k+1}$ is overwritten with $\mathbb{T}_{i,j}$, since we prefer to preserve direct trust distributions, which are built from the history of assessments.

3.2.3 Information Decay

An important notion in our proposal is the *decay* of information. We say the integrity of information decreases with time. In other words, the information provided by a trust probability distribution should lose its value over time and decay towards a default value. We refer to this default value as the *decay limit distribution* $\mathbb{D}$. For instance, $\mathbb{D}$ may be the ignorance distribution, which would mean that trust information learned from past experiences tends to ignorance over time.

Information in a probability distribution $\mathbb{T}$ decays from t to t' (where $t' > t$) as follows:

$$\mathbb{T}^{t \rightsquigarrow t'} = \Lambda(\mathbb{D}, \mathbb{T}^t) \tag{8}$$

where Λ is the *decay function* satisfying the property: $\lim_{t'\to\infty} \mathbb{T}^{t \rightsquigarrow t'} = \mathbb{D}$. One possible definition for Λ could be:

$$\mathbb{T}^{t \rightsquigarrow t'} = \nu^{\Delta_{t,t'}} \cdot \mathbb{T}^t + (1 - \nu^{\Delta_{t,t'}})\mathbb{D} \tag{9}$$

where ν is the decay rate, and:

$$\Delta_{t,t'} = \begin{cases} 0 & \text{, if } t' - t < \omega \\ 1 + \dfrac{t'-t}{t_{max}} & \text{, otherwise} \end{cases}$$

The definition of $\Delta_{t,t'}$ above serves the purpose of establishing a minimum *grace* period, determined by the parameter ω, during which the information does not decay, and that once reached the information starts decaying. The parameter t_{max}, which may be defined in terms of multiples of ω, controls the *pace of decay*. The main idea behind this is that after the grace period, the decay happens very slowly; in other words, $\Delta_{t,t'}$ decreases very slowly.

To implement such a decay mechanism in our model, we need to:

1. Record all evaluations $e_\mu^\alpha \in \mathcal{H}$ made at time t with a timestamp t, noted $e_\mu^{\alpha^t}$.
2. Record all direct trust distributions $\mathbb{T}_{i,j}$ with a timestamp t, noted $\mathbb{T}_{i,j}^t$, where t is the timestamp of the last evaluation that modified the trust distribution (recall that direct trust distributions may be modified when a new assessment occurs). The first time $\mathbb{T}_{i,j}$ is modified, t is the timestamp of the evaluation involved in the modification. Then, every time a new evaluation with timestamp $t' > t$ is considered to update $\mathbb{T}_{i,j}^t$, $\mathbb{T}_{i,j}^t$ is first decayed from t to t' before the distribution is modified.
3. Record all indirect trust distributions $\mathbb{T}_{i,j}$ with a timestamp t, noted $\mathbb{T}_{i,j}^t$. Every time $\mathbb{T}_{i,j}$ is calculated, all probability distributions involved in this calculation will first be decayed to the time of calculation t, which will be the resulting timestamp of $\mathbb{T}_{i,j}$.

3.3 Step 2: What to believe when a peer gives an opinion?

Given a peer assessment e_μ^α, the question now is how to compute the probability distribution of ϵ's evaluation. In other words, what is the probability that ϵ's evaluation of α is x given that μ evaluated α with e_μ^α. As illustrated earlier, this is expressed as the conditional probability:

$$\mathbb{P}(X^\alpha{=}x \mid e_\mu^\alpha)$$

To calculate this conditional probability, the intuition is that ϵ would tend to agree with μ's evaluation if his trust on μ is high (that is, the expected evaluation difference between their assessments is close to 0). Otherwise, ϵ's evaluation would probably be different. We perform then a sort of analogical reasoning: if in the past μ gave assessments with a certain evaluation difference with respect to ϵ, then this will probably happen again now.

We thus calculate the above conditional probability simply as:

$$p(X^\alpha{=}x \mid e_\mu^\alpha) = \begin{cases} \displaystyle\sum_{y \leq \mathit{diff}(x, e_\mu^\alpha)} \mathbb{T}_{\epsilon,\mu}(y) & \mathit{if}\ x = 0 \\ \displaystyle\sum_{y \geq \mathit{diff}(x, e_\mu^\alpha)} \mathbb{T}_{\epsilon,\mu}(y) & \mathit{if}\ x = b \\ \mathbb{T}_{\epsilon,\mu}(\mathit{diff}(x, e_\mu^\alpha)) & \mathit{otherwise} \end{cases} \tag{10}$$

Observe that in two cases the probabilities are computed as the summation of the probability mass of $\mathbb{T}_{\epsilon,\mu}$ for points below or over the difference between the new opinion and the point x under consideration. This is done to cope with the fact that we cannot under rate or over rate more as we are at the extremes already and consider that for instance past cases where we under rated more should be taken into account when we are determining the probability that the leader gives a 0 in the assessment. Similarly for b. For example, assume μ's assessment is 2 when the maximum mark is 3, we are calculating the probability of ϵ's assessment, and ϵ usually over rates μ by 2 marks. The probability of ϵ's assessment being 2 will essentially be $\mathbb{T}(0)$ (since the difference $2 - 2 = 0$). However, the probability of ϵ's assessment being 3, cannot simply be $\mathbb{T}(1)$ (since the difference $3 - 2 = 1$), because it is the maximum value of the evaluation space and so it also needs to consider all the over rating possibilities described by $\mathbb{T}(2)$ and $\mathbb{T}(3)$ as well. As such, the probability of ϵ's assessment being 3 aggregates $\mathbb{T}(1)$, $\mathbb{T}(2)$, and $\mathbb{T}(3)$.

3.4 Step 3: What to believe when many give opinions?

In the previous section we computed $\mathbb{P}(X^\alpha \mid e_\mu^\alpha)$. That is, the probability distribution of ϵ's evaluation on α given the evaluation of a peer μ on α. But what does ϵ do when there is more than one peer assessing α?

Given the set of opinions $\mathcal{O}^\alpha = \{e_{\mu_1}^\alpha, e_{\mu_2}^\alpha, \ldots, e_{\mu_n}^\alpha\}$ of a group of peers over the object α, we define the probability of ϵ's assessment being x as follows:

$$p(X^\alpha{=}x \mid \mathcal{O}^\alpha) = \begin{cases} \bigvee_{i=1}^{n} (\mathbb{I}(\mathbb{T}_{\epsilon,\mu_i}) \cdot p(X^\alpha{=}x \mid e^\alpha_{\mu_i})) & \sum_{i=1}^{n} \mathbb{I}(\mathbb{T}_{\epsilon,\mu_i}) > \delta \\ 1/n & \text{otherwise} \end{cases} \tag{11}$$

where $\vee$ is an operator that combines probabilities assuming the sources are independent:[2] $a \vee b = a + b - a * b$, and $\mathbb{I}(\mathbb{T}_{\epsilon,\mu})$ measures the information content of the probability distribution $\mathbb{T}_{\epsilon,\mu}$ as the earth mover's distance to the ignorance distribution (the uniform distribution $\mathbb{F}$). In other words, the probability of ϵ's assessment being x given the set of opinions $\mathcal{O}^\alpha$ is a disjunction of the probabilities of ϵ's assessment being x given each evaluation $e^\alpha_{\mu_i} \in \mathcal{O}^\alpha$ and diminished by the information content of the evaluation distributions $\mathbb{I}(\mathbb{T}_{\epsilon,\mu_i})$. We diminish the probability derived from a particular opinion when that opinion is actually not very informative and thus very close to ignorance. In the case that most opinions are close to ignorance, $\sum_{i=1}^{n} \mathbb{I}(\mathbb{T}_{\epsilon,\mu_i}) \leq \delta$, the result of such combination might be too close to zero (for a small δ) and thus we prefer to assume ignorance, $1/n$, for the probability value.

Finally, for several purposes (give a mark to a student, rank objects to purchase, ...) it is practical to 'summarise' distributions $\mathbb{P}(X^\alpha \mid \mathcal{O}^\alpha)$ into a number. From the several methods that can be used (centre of gravity, mean, median, ...) in the experiments we use the mode value of the distribution.

3.5 Step 4: What should be evaluated next?

The previous three steps allow to compute assessments of objects that have not been assessed by ϵ, based on peers opinions. The level of uncertainty of the assessments so generated by our method can be calculated as the uncertainty of the probability distribution $\mathbb{P}(X^\alpha \mid \mathcal{O}^\alpha)$. A classical method to measure this uncertainty is the the distribution's entropy:

$$\mathbb{H}(\mathbb{P}(X^\alpha \mid \mathcal{O}^\alpha)) = \sum_{x \in X^\alpha} p(X^\alpha{=}x \mid \mathcal{O}^\alpha) \cdot \ln p(X^\alpha{=}x \mid \mathcal{O}^\alpha) \tag{12}$$

We will explore in the experiments a heuristic that aims at reducing the number of assessments made by the leader. In other words, what object should be assessed next by ϵ in order to maximally decrease the overall uncertainty? For example, what assignments and in which order should a tutor evaluate so that the uncertainty of the computed assessments, i.e. the uncertainty on the students' marks, becomes *acceptable*. The heuristic is simple: we suggest that ϵ evaluates objects by decreasing value of the entropy of their assessment distribution, that is the next object α that the leader should assess is:

$$\alpha = arg \max_\alpha \mathbb{H}(\mathbb{P}(X^\alpha \mid \mathcal{O}^\alpha)$$

4. ALGORITHM

In this section we provide the pseudo-code of PAAS, which is a straightforward implementation from the equations defined in Section 3.

Algorithm 1 defines the method to apply when a new assessment is performed. In lines 1-14 direct trust distributions are updated in matrix C and vector t_ϵ, as discussed in subsection 3.2.1. In lines 15-22, indirect trust distributions are updated using the adapted eigentrust method, as discussed in subsection 3.2.2. Algorithm 2 is the method that updates direct trust distributions given a new opinion. Line 1 decays the distribution from time stamp t to t'. Line 2 updates the value in the distribution for the point representing the distance in the observation. Line 3 normalizes this distribution by computing the distribution with minimum relative entropy with respect to the distribution before the observation and that respects the updated value. Algorithm 3 deduces the overall assessment values (i.e. $\mathbb{P}(X^\alpha \mid \mathcal{O}^\alpha)$ after a number of assessments have been made.

Algorithm 1 $newAssessment(e_i^{\alpha^t})$

Require: $\mathcal{H} = \{\}$ ▷ This is the history of assessments
Require: $\mathbb{F}$ ▷ This is a trust probability distribution describing ignorance
Require: t'_ϵ ▷ This is a vector where ϵ's direct trust distributions are stored
1: **for all** $e_j^\alpha \in \mathcal{H}$ **do** ▷ Ordered by their timestamps
2: $diff_{i,j} = e_i^\alpha - e_j^\alpha$
3: $diff_{j,i} = e_j^\alpha - e_i^\alpha$
4: **if** $i = \epsilon$ **then**
5: $updateDirectDistribution(t_\epsilon[j], t, diff_{i,j})$
6: $t'_\epsilon = t'_\epsilon \cup t_\epsilon[j]$
7: **else if** $j = \epsilon$ **then**
8: $updateDirectDistribution(t_\epsilon[j], t, diff_{j,i})$
9: $t'_\epsilon = t'_\epsilon \cup t_\epsilon[j]$
10: **else**
11: $updateDirectDistribution(C_{i,j}, t, diff_{i,j})$
12: $updateDirectDistribution(C_{j,i}, t, diff_{j,i})$
13: **end if**
14: **end for**
15: $t_\epsilon^0 = \mathbb{F}$
16: **repeat**
17: $t_\epsilon^{k+1} = C^T t_\epsilon^k$ ▷ Equations 6 and 7
18: $error = \|t_\epsilon^{k+1} - t_\epsilon^k\|$
19: $t_\epsilon^{k+1} = t_\epsilon^k$
20: $t_\epsilon^{k+1} \leftarrow t'_\epsilon$ ▷ Overwrite distributions for those peers with direct trust
21: **until** $error < \eta$
22: $\mathcal{H} = \mathcal{H} \cup \{e_i^{\alpha^t}\}$

Algorithm 2 $updateDirectDistribution(\mathbb{T}^{t'}, t, x)$

Require: Λ ▷ This is the decay function
Require: $\mathbb{D}$ ▷ This is the default distribution
1: $\mathbb{T}_{i,j}^{t' \rightsquigarrow t} = \Lambda(\mathbb{D}, \mathbb{T}_{i,j}^t)$ ▷ Equations 8 and 9
2: $\mathbb{T}(X{=}x) = \mathbb{T}(X{=}x) + \gamma \cdot (1 - \mathbb{T}(X{=}x))$
3: $\mathbb{T}(X) = \arg\min_{\mathbb{P}'(X)} \sum_{x'} p(X{=}x') \log \frac{p(X{=}x')}{p'(X{=}x')}$ such that $\{p(X{=}x) = p'(X{=}x)\}$

5. EVALUATION

We present experiments performed over real data coming from two English language classrooms (30 14-years old students). Two different tasks were given to the classroom: an English composition task and a song vocabulary task. A total of 71 assignments were submitted by the students and marked by the teacher (our leader).

Students assessed their fellow students during a 1 hour period. A total of 168 student assessments were completed by the students (each student assessed on average 2.4 assignments). Marks vary from 1 (very bad) to 4 (very good). Students evaluated different criteria from the assignments: *focus, coherence, grammar* in the composition task and *in-time submission, requirements, lyrics* in the song vocabulary task.

[2] This assumption is not very restrictive for the scenarios we are considering: peer assessments in online education or e-commerce as opinions are expressed by people that do not know each other.

Algorithm 3 $calculateAssessments()$

Require: $\mathcal{I}$ ▷ This is the set of objects to be assessed
Require: $\mathcal{H}$ ▷ This is the history of assessments
$result = \{\}$
for all $\alpha \in \mathcal{I}$ **do**
  if $e^{\alpha}_{\epsilon} \in \mathcal{H}$ **then**
    $result = result \cup e^{\alpha}_{\epsilon}$
  else
    $\mathcal{O}^{\alpha} = \{e^{\alpha}_{\mu} | e^{\alpha}_{\mu} \in \mathcal{H}\}$
    $e^{\alpha}_{-} = \{x | \mathbb{P}(X^{\alpha} = x \mid O^{\alpha})$ is maximum$\}$ ▷ Equations 11 and 10
    $result = result \cup e^{\alpha}_{-}$
  end if
end for
return $result$

Thus, $\mathcal{E} = \{1, 2, 3, 4\}$, and the x-axis of our trust distributions is $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ (which are the possible evaluation distance values between peers in this setting). We calculate the error of the generated assessments, noted as e^{α}_{-}, as the average difference between them and the tutor assessments, that is:

$$error = \frac{\sum_{\alpha \in \mathcal{I}} \|e^{\alpha}_{-} - e^{\alpha}_{\epsilon}\|}{|\mathcal{I}|}$$

In addition to the error, we are also interested in plotting the number of deduced assessments. We note that when there is no peer or tutor assessment for a particular assignment, an automated mark for that assignments can not be generated.

In the first experiment where we compare our model with the well known Collaborative Filtering (CF) algorithm [9]. As discussed in Section 2, CF is a social information filtering algorithm that recommends content to users based on their previous preferences. CF biases the final computation towards a particular member: the person being recommended, as our algorithm does.

In this experiment, we randomly select a subset of 6 teacher assessments to use as the leader's opinion in both PAAS and CF (this subset represents 8.4% of the total number of assessments, the rest of teacher assessments are used to calculate the error). Then, several iterations are performed, one for each student assessment. On each iteration:

- One student assessment is selected randomly from the set of student assessments and added to PAAS and CF
- Automated assessments are generated by PAAS and CF and the error is calculated. To calculate the error, our groundtruth is the set of all tutor assessments.

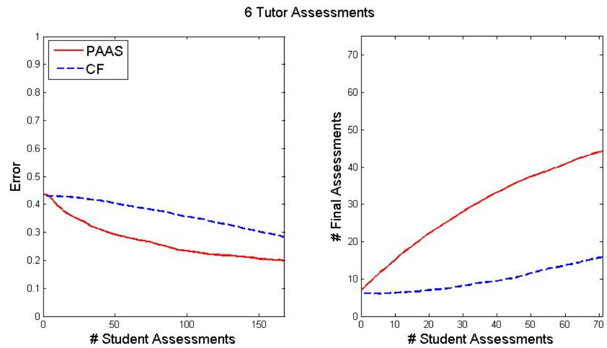
Results are averaged over 50 executions. When an assessment for a particular assignment could not be deduced, a default mark (ignorance) 2 is given, since this value is situated more or less in the middle of the evaluation space. Default marks are used in both PAAS and CF error calculations.

Figure 1 shows the results of PAAS and CF on three cases. As the assignments are different with different evaluation criteria we choose a criterion per group, necessarily different, so that we can have a larger number of assignments in the experiments. Three such pairings of criteria are shown in the figure. It is clear in the three cases the remarkable improvement of PAAS over CF considering the number of final marks generated (see the right column of graphics in the figure). PAAS has an added capability with respect to CF in using indirect trust measures to generate assessments. In CF the opinion of someone without any similarity in her profile with the leader (in our case, without any common assignment being assessed) cannot be used to suggest a recommendation (an assessement). Thus, PAAS is capable of generating many more assessments, specially once the graph of indirect trust relationships becomes more and more connected. This highlights PAAS's first point of strength: *PAAS increases the number of assessments that can be calculated.* On the left, we show the improvement of PAAS over CF in terms of the error with respect to the ground truth that we know (the actual teacher assessments). The error is calculated over the entire set of assignments, including assignments that receive the default mark. This highlights PAAS's second point of strength in outperforming CF: *PAAS decreases the error of the assessments calculated.* We note that when the number of peer assessments increases PAAS and CF's error get closer because the effect of indirect trust diminishes. However, we are much better than CF for a small effort per peer (for instance, think of 5 or 6 assessments per peer instead of hundreds).

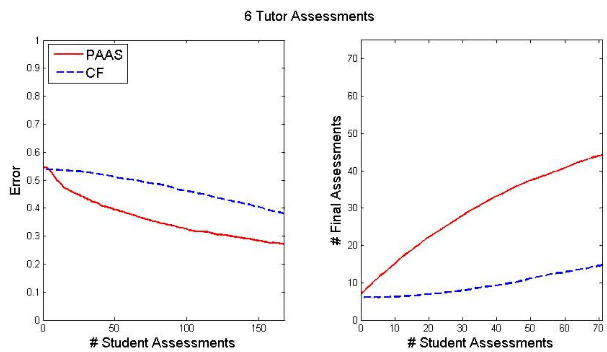
We perform a second experiment where we assess the impact of using the heuristic that informs the teacher of which assignment to select next to assess, see section 3.5 for details. In this case, we designed an experiment where we simulate a classroom of 200 students with 200 submitted assignments, where each assignment is evaluated by 5 students (1000 peer assessments performed). To show a critical case, we simulate that half of the assignments are evaluated accurately by half of the students (that is, those students provided the same mark as the tutor) and the other half of the assignments are evaluated poorly (that is, randomly) by the other half of the students. In the simulation, we have two instances of the PAAS model: PAAS Random and PAAS Ranking. First, all the student assessments are added to both instances of the PAAS model. Then, several iterations are performed, one for each tutor assessment. On each iteration:

- We randomly select a tutor assessment for an assignment that has not been assessed yet, and we add this tutor assessment to PAAS Random.
- We select a tutor assessment for an assignment not yet assessed following the suggestion of the entropy heuristic, and we add this tutor assessment to PAAS Ranking.
- Automated assessments are generated by PAAS Random and PAAS Ranking and the error is calculated.

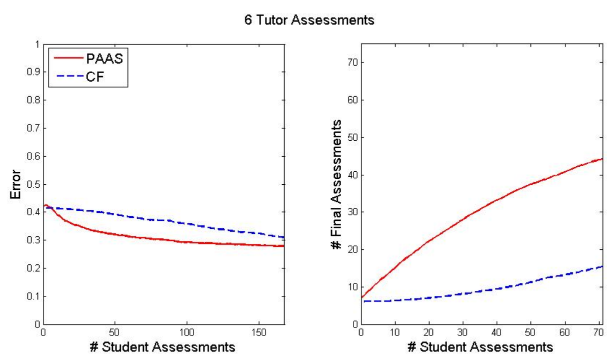
Figure 2 (a) shows the error which of course decreases with every new tutor assessment. We also see how ranking the assessments with the entropy heuristic decreases the error faster. Figure 2 (b) shows the same experiments but with the real data. In this case, there is no clear advantage in ranking assessments over simply assessing randomly. This is an indicator that the students from these two groups where closely aligned with the tutor's opinion. In other words, all the assignments where performed with more or less the same quality (in contrast with the scenario presented in (a) with simulated data). Figure 2 (c) shows the same experiment presented in (b) but in this case the assessments of half of the assignments were overwritten providing a random mark. Such noise introduced, even in this rough manner, produces that the ranking strategy becomes slightly more effective. We also highlight the fact that the error of the PAAS model does not change drastically when noise is introduced, since PAAS is able to distinguish which assessments are trustworthy and which are not very quickly. We conclude from this second experiment that although in some cases, e.g. when the students are good 'recommenders', the heuristic may not be needed, in general it can improve the results when such recommendation quality is missing.



(a) focus and in-time submission criteria



(b) coherence and requirements criteria



(c) grammar and lyrics criteria

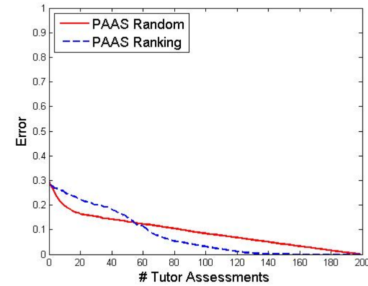
Figure 1: Experiments with Real Data: PAAS vs CF. We show the results for opinions on two criteria, one for each assignment. For instance in (a) we combine opinions on focus in the composition task and submission on time in the song vocabulary task.

6. CONCLUSIONS AND FUTURE WORK

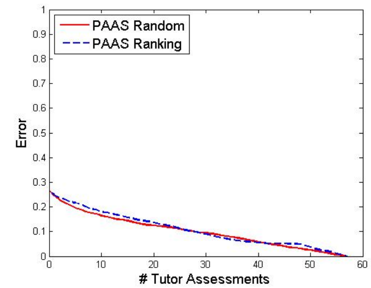
In this paper we have presented the Personalised Automated ASsessments model (PAAS), a trust-based assessment service that helps compute automated assessments from the perspective of a specific community member: a community leader. This computation essentially aggregates peer assessments, giving more weight to those peers that are trusted by the leader. How much the leader trusts a peer is based on the similarity between her (past) assessments and the peer's (past) assessments over the same objects. The application of this model is specially useful in the context of online communities, where community members interact providing feedback or when the number of objects to be assessed is so large that it would be very costly to assess them on an individual basis.

We have experimentally shown that the algorithm works well in a real setting, and outperforms the well-known CF algorithm in two different ways: (1) by remarkably increasing the number of assessments that can be calculated, and (2) by remarkably decreasing the error of the assessments calculated.

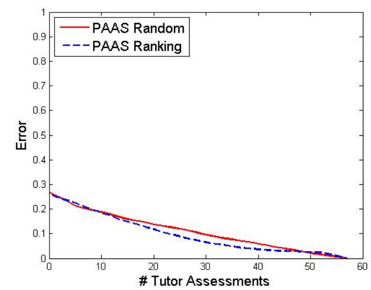
Plans for future work include: 1) evaluating the model with more extensive real datasets that are currently being collected; 2) testing the model in real settings by a company specialised in online learning solutions; and 3) applying the model to a domain other than online learning, where the direct and indirect trust relations can help community members decide who to trust in a given context. Another question to study is how the results would change when other similarity measures for the differences between peers are used.



(a) Synthetic data



(b) Real data



(c) Randomised Real data

Figure 2: Experiments considering the entropy heuristic

Acknowledgments

This work is supported by the CollectiveMind project (funded by the Spanish Ministry of Economy and Competitiveness, under grant number TEC2013-49430-EXP).

REFERENCES

[1] L. de Alfaro and M. Shavlovsky. Crowdgrader: Crowdsourcing the evaluation of homework assignments. *Thech. Report 1308.5273, arXiv.org*, 2013.

[2] J. Debenham and C. Sierra. Trust and honour in information-based agency. *Proceedings of Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 1225–1232, 2006.

[3] P. Gutierrez, N. Osman, and C. Sierra. Trust-based community assessment. *Pattern Recognition Letters*, submitted for publication.

[4] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. *Proceedings of the 12th international conference on World Wide Web*, pages 640–651, 2003.

[5] N. Osman, C. Sierra, F. McNeill, J. Pane, and J. K. Debenham. Trust and matching algorithms for selecting suitable agents. *ACM TIST*, 5(1):16, 2013.

[6] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. Tuned models of peer assessment in moocs. *Proc. of the 6th International Conference on Educational Data Mining (EDM 2013)*, 2013.

[7] K. Regan, P. Poupart, and R. Cohen. Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. *Proceedings of 21st national conference on Artificial intelligence*, pages 1206–1212, 2006.

[8] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV 1998)*, ICCV '98, pages 59–, Washington, DC, USA, 1998. IEEE Computer Society.

[9] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". pages 210–217. ACM Press, 1995.

[10] C. Sierra and J. Debenham. An information-based model for trust. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 497–504, New York, NY, USA, 2005. ACM.

[11] C. Sierra and J. K. Debenham. Information-based agency. In M. M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 1513–1518, 2007.

[12] W. T. L. Teacy, M. Luck, A. Rogers, and N. R. Jennings. An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artificial Intelligence*, pages 149–185, 2012.

[13] T. Walsh. The peerrank method for peer assessment. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 909–914. IOS Press, 2014.

[14] J. Wu, F. Chiclana, and E. Herrera-Viedma. Trust based consensus model for social network in an incompletelinguistic information context. *Applied Soft Computing*, 2015.