Personalised Automated Assessments

Patricia Gutierrez[1], Nardine Osman[1], Carme Roig[2], Carles Sierra[1]
[1]IIIA-CSIC, Campus de la UAB, Barcelona, Spain
{patricia,nardine,sierra}@iiia.csic.es
[2]INS Torras i Bages, L'Hospitalet de Llobregat, Spain
mroig112@xtec.cat

ABSTRACT

Consider a person who needs to assess a large amount of information. For instance, think of a teacher of a massive open online course with thousands of enrolled students, or a senior program committee member in a large conference who needs to decide what are the final marks of reviewed papers, or a buyer in an e-commerce scenario who needs to build up her opinion about products. When assessing a large number of objects, sometimes it is simply unfeasible to evaluate them all and very often one needs to rely on the opinions of others. In this paper, we provide a model that uses peer assessments (assessments made by others) in an online community to approximate the assessments that a particular member of the community would generate given the occasion to do so (e.g. the tutor, the SPC member or the buyer—we refer to this person as the *leader*). Furthermore, we provide a measure of the uncertainty of the computed assessments and a ranking of the objects that should be assessed next. The model, although inspired by human societies is thought to be used in the organisation of agent communities.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Theory

Keywords

Trust and reputation, Collective intelligence, Community Assessment, Educational Applications

1. INTRODUCTION

Consider a person who needs to assess a large amount of information. For instance, think of a teacher of a massive open online course with thousands of enrolled students, or a senior program committee member in a large conference who needs to decide what are the final marks of reviewed papers, or a buyer in an e-commerce scenario who needs to build up her opinion about products. When assessing a large number of objects, sometimes it is simply unfeasible to evaluate them all and very often one needs to rely on the opinions of others. In the process of building up one's opinion, there are some questions that need to be answered, such as: How much should I trust the opinion of a peer? What should I believe given a peer's opinion? What should I believe when many peers give divergent opinions? Which objects should I assess next, such that the certainty of my beliefs improves? This paper seeks answers to these questions through the *P*ersonalised Automated ASsessment model (PAAS).

PAAS uses peer assessments (assessments made by others) in an online community to approximate the assessments that a particular member of the community would generate given the occasion to do so (e.g. the tutor, the SPC member or the buyer). The computed peer-based assessment of objects aims at matching the perspective/view/opinion of a *s*pecific community member. We will call this person the *community leader* or simply the *leader*. PAAS aggregates peer assessments giving more weight to those peers that are trusted by the leader. How much a leader trusts a peer is based on the similarity between her (past) assessments and the peer's (past) assessments over the same objects.[1] When such a trust measure between the leader and a peer cannot be obtained—that is, when no object was assessed by both of them—, an indirect trust measure is built based on the reputation of the peer within the community, where this reputation is computed from the perspective of the leader. That is, the closer the peer's opinions to the leader's opinions the higher her reputation. As such, our objective, differently from most previous approaches, is not consensus building.

Finally, we are also able to provide a measure of the uncertainty of PAAS's computed assessments and a ranking of the objects that should be assessed next by the leader in order to decrease the overall uncertainty of those calculated assessments.

This paper is based on the work carried out in [3]. We take from [3] the idea of biasing evaluations with respect to a leader. However, the method described in this paper is completely new. In particular, our improvements with respect to [3] include: (1) the use of information theory: trust and assessments are represented as probability distributions and not as just ordinal numbers; (2) the use of the Eigentrust algorithm to calculate indirect trust instead of a graph search; (3) the increased efficiency as we do not need to maintain and update a graph of trust relations; and (4) the computation of the uncertainty of the automated assessments generated, helping suggesting which objects should be evaluated next.

This paper is structured as follows. In Section 2 we do a quick review of previous work. In Section 3 we present the model with associated algorithms in Section 4. We experimentally evaluate the model in Section 5. Finally, we conclude in Section 6.

[1]We could extend the approach by using semantic similarity and thus using past experiences evaluating *similar* objects. This has been explored in previous works [5, 11] and PAAS could be easily extended to cover this case.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), John Thangarajah, Karl Tuyls, Stacy Marsella, Catholijn Jonker (eds.), May 9–13, 2016, Singapore.*

2. RELATED WORK

Previous works have proposed different methods for generating assessments based on peer assessments.

CrowdGrader [1] is a framework which defines a crowdsourcing algorithm for peer evaluation. The authors claim that, when performing evaluations, relying on a single person is often impractical and can be perceived as unfair. Their method aggregates the assessments of an assignment made by several students into an overall assessment for the assignment, relying on a reputation system. The reputation of each student (or their *accuracy degree* as they call it) is measured by comparing the student's assessments with the assessments of their fellow students for the same assignments. In other words, the reputation of a student describes how far are her assessments from those of her fellow students. The overall assessment (consensus grade) is calculated by aggregating all student assessments weighted by the reputation of the students providing them. The algorithm executes a fixed number of iterations using the consensus grade to estimate the reputation (or accuracy degree) of students, and in turn uses the updated student's reputation to compute more precise suggested assessments.

PeerRank [13] is based on the idea that the grade of an agent is constructed from the grades it receives from other agents, and the grade an agent gives to another agent is weighted by the grading agent's own grade. Thus, the grade of each agent α is calculated as a weighted average of the grades of the agents evaluating α, and thus the grades of α's evaluators are themselves weighted averages of the grades of other agents evaluating them, and so on. The final grades are defined as a fixed point of an equation, similar to the PageRank method where web-pages are ranked according to the ranks of the web-pages that link to them.

Piech et al [6] proposes a method to estimate student reliability and to correct student biases in an online learning scenario, presenting results over two Coursera courses. They assume the existence of a true score for every assignment, which is unobserved and to be estimated. Every grader is associated with a bias, which reflects the grader's tendency to inflate or deflate her assessments with respect to the true score. Also, graders are associated with a reliability which reflects how close the grader's assessments tend to land near the corresponding true score, after having them corrected for bias. Authors infer the values of these unobserved variables using known approximated inference methods such as Gibbs sampling. The model proposed is therefore probabilistic and is compared to the grade estimation algorithm used on Coursera's platform, which does not take into account individual biases and reliability.

Wu et al [14] investigates consensus building between a group of experts in a trust network. New trust relationships are derived from the trust network and the trust scores of such relationships are calculated using an averaging operator that aggregates trust/distrust values from multiple trust paths in the network. The trust score is used to distinguish the most trusted expert from the group and, ultimately, to drive the aggregation of the individual opinions in order to arrive at a group consensual decision making solution. This work also includes a visual consensus model to identify discordant opinions, to produce recommendations to those experts that are furthest from the group, and to show future consensus status if experts are to follow the recommendations.

Collaborative Filtering (CF) [9] is a classical social information filtering algorithm that recommends content to users based on their previous ratings, exploiting the similarities between the tastes of different users. The basic idea is as follows:

1. The system maintains a user profile, which is a record of the user ratings over specific items.
2. Then, the system computes a similarity measure among users' profiles.
3. Finally, the system recommends items to users with a rating that is a weighted average of the ratings on that item given by other users. The weights are the similarity measures between the profiles of users rating the item and the profile of the user receiving the recommendation.

What is fundamentally different between our PAAS model and these works is that the computation of our automated assessments is tuned to the perspective of a specific community member, a leader. We clarify that our target is not consensus building to provide assessments, but to accurately estimate those unknown assessments *from the leader's point of view*. Furthermore, PAAS aggregates peer assessments giving more weight to those peers that are trusted by the leader. Such trust metrics are built, as we will see shortly, using probability distributions based on the history of past assessments between the leader and his/her peers, rather than using aggregations.

Some models allow agents to adapt the opinions expressed by others when aggregating those to compute an overall trust value. For instance, BLADE [7] is a Bayesian model that enables a buyer agent to interpret the ratings of other buyers (aka advisors) over sellers. Each buyer uses the advisors' ratings to build a model of the evaluation function of the advisor. Then, it adjusts the ratings by comparing the advisor's evaluation function and its own evaluation function. Similarly, HABIT [12] is another Bayesian trust model which builds a two-layer hierarchical model in which the opinions of different reputation sources are represented at the bottom level, and their correlation with the actual trustee's behaviour is computed at the top level. Then, information sources with a low correlation with the actual behaviour are considered unreliable and filtered out from the reputation computation. Similarly to these models, we reinterpret the opinions given by peers. However, in our case, the reinterpretation is made based on the relation between the opinions and a gold standard provided by an opinion leader in the community and not by the observation of actual behaviour. Also, differently from them, we use an approach based on information theory and entropy measures to build trust, where with every new observation the uncertainty of the model decreases.

In the experimental evaluation of our system, we compare PAAS to CF since CF is the only one that biases the final computation towards the opinion of a particular member of the community. Furthermore, CF has been widely adopted by the industry. Typical recommendation services, as the ones provided by Amazon, Youtube or Last.fm, are based on the CF algorithm.

3. THE PAAS MODEL

3.1 Notation and Problem Definition

Let ϵ represent the leader who needs to assess a large set of objects $\mathcal{I}$, and let $\mathcal{P}$ be a set of peers able to assess objects in $\mathcal{I}$.

We define a peer assessment e_μ^α (also referred to as evaluation or opinion) as an element from an ordered discrete evaluation space $\mathcal{E}$, where $\alpha \in \mathcal{I}$ is the object being evaluated and $\mu \in \{\epsilon\} \cup \mathcal{P}$ is the evaluating peer. We define an automated assessment $\bar{e}^\alpha$ for object α as a metric (which could be the mean, the median, the maximum, etc.) built from a probability distribution $\mathbb{P}$ over the evaluation space $\mathcal{E}$. We say $\mathbb{P}=\{x_1 \mapsto v_1, \ldots, x_n \mapsto v_n\}$, where $\{x_1, \ldots, x_n\}=\mathcal{E}$ and $v_i \in [0,1]$ represents the value assigned to each element $x_i \in \mathcal{E}$, with the condition that $\sum_{0<i\leq|\mathcal{E}|} v_i = 1$.

For example, one can define the evaluation space of the quality of an English classroom homework as $\mathcal{E}=\{poor, good, excellent\}$.

The distribution $\{poor \mapsto 0, good \mapsto 0, excellent \mapsto 1\}$ would represent the best possible assessment, whereas the distribution $\{poor \mapsto 0, good \mapsto 1/2, excellent \mapsto 1/2\}$ would represent that the quality of the homework is most probably between good and excellent, and so on.

Finally, we define $\mathcal{H}$ as the history of all assessments performed, and $\mathcal{O}^\alpha \subset \mathcal{H}$ as the set of past peer assessments over the object α.

The ultimate goal of our model is to compute the probability distribution of ϵ's evaluation over a certain object α, given the evaluations of several peers over that same object α. In other words, what is the probability that ϵ's evaluation of α is x given the set of peers' evaluations $\mathcal{O}^\alpha$? Such expectation can be formalized with the following conditional probability: $p(X^\alpha{=}x \mid \mathcal{O}^\alpha)$. To calculate the above conditional probability, we take into account every particular evaluation in $\mathcal{O}^\alpha$. In other words, expectations (or probabilities) are calculated for *each* individual evaluation in $\mathcal{O}^\alpha$, before those expectations are aggregated into $p(X^\alpha{=}x \mid \mathcal{O}^\alpha)$. The probability that ϵ's assessment is x given a particular evaluation $e^\alpha_\mu \in \mathcal{O}^\alpha$ is $p(X^\alpha{=}x \mid e^\alpha_\mu)$ and then, $p(X^\alpha{=}x \mid \mathcal{O}^\alpha)$ is defined as an aggregation of these individual probabilities, where our particular aggregation is presented in Section 3.4.

We base the computation of the individual conditional probabilities on the notion of *trust* between peers built from previous experiences, where trust is understood as the similarity between the assessments made by those peers for the same objects. In other words, our intuition is that we expect ϵ will tend to agree with μ's assessment on an object if her trust on μ is high. Otherwise, ϵ's evaluation will probably be different. To build a trust measure between ϵ and μ we perform a sort of analogical reasoning: if in the past μ gave opinions that were similar to ϵ's opinions to a certain degree (trust), then ϵ is likely to coincide with μ's opinion again now to the same degree.

The remainder of this section is divided as follows. We first describe in detail how the measure of trust between peers is calculated (Section 3.2). Then, we illustrate how to calculate ϵ's assessment of an object α given μ's assessment of α and ϵ's trust on μ's assessments (Section 3.3), that is, we present our approach to calculate the individual probability $p(X^\alpha{=}x \mid e^\alpha_\mu)$. We then illustrate how to combine these probabilities to build the probability distribution of ϵ's assessments given the assessments of several peers (Section 3.4), that is, we present an approach for calculating the probability $p(X^\alpha{=}x \mid \mathcal{O}^\alpha)$. Finally, we provide a measure of the uncertainty of the computed assessments and a ranking of the objects that should be assessed next by ϵ in order to decrease that uncertainty (Section 3.5).

3.2 Step 1. How much should I trust a peer?

ϵ needs to decide how much can she trust the assessment of a peer μ. We base this trust measure on two intuitions. First, if ϵ and μ have both assessed the same object in the past, then the similarity of their assessments for that object can give a hint on how close their judgments/thinking are. When there are no objects evaluated by both ϵ and μ, ϵ would not know how much to trust μ's assessments. Second, and to cover this latter situation, we approximate the unknown trust between ϵ and μ by transitivity over the path with direct trust links between ϵ and μ In the following, we make these two intuitions concrete through two different types of trust relations: *direct trust* and *indirect trust*.

3.2.1 Direct Trust

Direct trust is the trust relation that emerges between two people or two agents that have already assessed the same objects in the past. Our approach is to compute such relation as an aggregation of their evaluations' similarity. For instance, let the set $A_{i,j} = \{\alpha \mid e^\alpha_i, e^\alpha_j \in \mathcal{H}\}$ be the set of objects that have been assessed by agents i and j. Then, different definitions for the direct trust between i and j can be adopted based on different similarity functions over objects in $A_{i,j}$, such as:

- The average of the similarities: $\dfrac{\sum_{\alpha \in A_{i,j}} sim(e^\alpha_i, e^\alpha_j)}{|A_{i,j}|}$
- The conjunction of the similarities: $\bigwedge_{\alpha \in A_{i,j}} sim(e^\alpha_i, e^\alpha_j)$
- The linear correlation between i and j:
$$\frac{\sum_{\alpha \in A_{i,j}} sim(e^\alpha_i, \bar{e_i}) \cdot sim(e^\alpha_j, \bar{e_j})}{\sqrt{\sum_{\alpha \in A_{i,j}} sim(e^\alpha_i, \bar{e_i})^2} \sqrt{\sum_{\alpha \in A_{i,j}} sim(e^\alpha_j, \bar{e_j})^2}},$$
where $\bar{e_i}$, $\bar{e_j}$ are the means of the evaluations performed over the set $A_{i,j}$ by i and j, respectively.

However, when we calculate such aggregations we loose relevant information. For instance, we are not able to tell if j usually under rates with respect to i, if it usually over rates, or neither. We are also not able to tell if the dissimilarities between i and j's evaluations are highly variable or not.

To cope with such loss of information, we will define the direct trust between two peers i and j as a probability distribution $\mathbb{T}_{i,j}$ over evaluation differences built from the historical data of previous evaluations performed by i and j.

DEFINITION 3.1. *We define the* evaluation difference *between two assessments performed by i and j as:*

$$\mathit{diff}(e^\alpha_i, e^\alpha_j) = e^\alpha_i - e^\alpha_j \tag{1}$$

We use the euclidean distance between assessments as the measure of dissimilarity, as it is the most used distance in the literature on similarity in metric spaces. If $\mathit{diff}(e^\alpha_i, e^\alpha_j){=}0$, it means that i and j provide the same evaluation for α. If $\mathit{diff}(e^\alpha_i, e^\alpha_j) > 0$, it means that i *over rates* α with respect to j, if $\mathit{diff}(e^\alpha_i, e^\alpha_j) < 0$, it means that i *under rates* α with respect to j. Note that $\mathit{diff}(e^\alpha_i, e^\alpha_j) \neq \mathit{diff}(e^\alpha_j, e^\alpha_i)$.

When defining $\mathbb{T}_{i,j}$, we are interested in maintaining information about whether a peer under rates or over rates with respect to another peer. As such, the support of the distribution representing i's direct trust on j (i.e. the x-axis of $\mathbb{T}_{i,j}$) consists of the possible evaluation difference values between i and j. Trust distribution $\mathbb{T}_{i,j}(x)$ then describes the probability that i and j would assess an object with an evaluation difference x. Therefore, the distribution $\mathbb{T}_{i,j}(0) = 1$ represents a trust distribution where i *fully* trusts on j's opinion, since the probability that their assessments are the same is 1.

DEFINITION 3.2. *Given a numeric evaluation space $\mathcal{E} = [0, b]$, a* Trust Distribution *is any probability distribution over the differences in $\mathcal{E}$, that is over the interval $[-b, b]$.*

In what follows, we explain how we build direct trust distributions computationally, based on previous experiences. We use an information theory approach where the behavior of the studied phenomenon is modeled by probability distributions which are updated with every new observation. This approach is inspired by [2].

Initially, the direct trust distribution between any two peers i and j is the distribution describing ignorance (i.e. the uniform distribution). Then, whenever j evaluates an object α that was already evaluated by i we update $\mathbb{T}_{i,j}$ as follows:

1. We find the element x in $\mathbb{T}_{i,j}$'s support whose probability needs to be adjusted: x=*diff* (e_i^α, e_j^α).

2. We increase the probability of x in $\mathbb{T}_{i,j}$ as follows:

$$p(X^\alpha{=}x) = p(X^\alpha{=}x) + \gamma \cdot (1 - p(X^\alpha{=}x)) \quad (2)$$

The update is based on increasing the current probability $p(X^\alpha = x)$ by a fraction $\gamma \in [0, 1]$ of the total potential increase $(1 - p(X^\alpha{=}x))$. For instance, if the probability of x is 0.6 and γ is 0.1, then the new probability of x becomes $0.6{+}0.1{\cdot}(1{-}0.6) = 0.64$. We note that the ideal value of γ should be closer to 0 than to 1 so that one single experience does not result in considerable changes in the distribution. In other words, a *single* assessment cannot result in a *significant* change in the probability distribution.

3. We normalize $\mathbb{T}_{i,j}$ by following the entropy based approach of [10]. The entropy-based approach updates $\mathbb{T}_{i,j}$ such that: (1) the value $p(X^\alpha{=}x)$ is maintained and (2) the resulting distribution has a minimal relative entropy with respect to the previous one. In other words, we look for a distribution that contains the updated probability value $p(X^\alpha{=}x)$ and that is at a minimal distance from the previous $\mathbb{T}_{i,j}$:

$$\mathbb{T}_{i,j}(X) = \underset{\mathbb{P}'(X)}{\arg\min} \sum_{x'} p(X^\alpha{=}x') \log \frac{p(X^\alpha{=}x')}{p'(X^\alpha{=}x')} \quad (3)$$
$$\text{such that} \quad \{p(X^\alpha{=}x) = p'(X^\alpha{=}x)\}$$

where $p(X^\alpha{=}x')$ is a probability value in the original distribution, $p'(X^\alpha{=}x')$ is a probability value in the potential new distribution $\mathbb{P}'$, and $\{p(X^\alpha{=}x) = p'(X^\alpha{=}x)\}$ specifies the constraint that needs to be satisfied by the resulting distribution.

3.2.2 Indirect Trust

Indirect trust is the trust relation that is deduced between peers when they have not assessed any objects in common and thus a direct trust relation cannot be computed. The notion of indirect trust is inspired in the eigentrust algorithm for reputation management [4]. Eigentrust proposes a reputation system that aggregates the local trust values of users in a peer-to-peer network, based on the notion of transitive trust, that is: a peer i will have a high opinion of those peers who have had a trustworthy behaviour. If they are honest in their behaviour they are also likely to be honest in reporting their local trust values, so peer i is likely to trust them. Local trust values are reported to i by community members. Then, the trust that i places on a peer k is based on the trust values reported by the community members and weighted by the trust i has on each community member.

In eigentrust the transitivity in trust is based on products and additions of positive real numbers. However, in our case we need to define operators to compute the transitive trust distribution from two distributions. That is, what is $\mathbb{T}_{i,k}$ given $\mathbb{T}_{i,j}$ and $\mathbb{T}_{j,k}$. The idea is that differences have to be combined in an additive way. We define this next.

If we want to compute the distance distribution between the leader and a peer α via an intermediary peer β we need to combine the probability distribution representing the evaluation difference between β and α with the probability distribution representing the evaluation difference between the leader and β. Thus, for a difference in opinion x between peers β and α and a difference in opinion y between β and the leader, the overall difference between the leader and α is $z = x + y$, as we are in an ordinal space. When we move to probabilities, we then say that $P(z) = P(x) * P(y)$, as we assume independence between opinions. Following this intuition, we define the combined distance distribution between two peers as follows.

DEFINITION 3.3. *Given Trust Distributions $\mathbb{P}$ and $\mathbb{Q}$ over the numeric interval $[-b, b]$ we define their* Combined Distance Distribution*, noted $\mathbb{R} = \mathbb{P} \otimes \mathbb{Q}$, as:*

$$r(X = x) = \begin{cases} \sum\limits_{x_1+x_2=x} p(X = x_1) * q(X = x_2) & \text{if } x \in (-b, b) \\ \sum\limits_{x_1+x_2\leq -b} p(X = x_1) * q(X = x_2) & \text{if } x \leq -b \\ \sum\limits_{x_1+x_2\geq b} p(X = x_1) * q(X = x_2) & \text{if } x \geq b \end{cases} \quad (4)$$

This operation can be nicely applied to our case of evaluation differences as the transitive trust is nothing else than the aggregation (addition) of the combined probability (product) of given evaluation differences happening.

The $\leq$ and $\geq$ are used to maintain the range of the evaluation distance within the $[-b, b]$ limits. For example, assume $\mathbb{P} = \{0, 0, 1\}$ and $\mathbb{Q} = \{1, 0, 0\}$, over the support (x-axis) [-1,1]. Now assume we need to calculate $\mathbb{R}(-1)$. We say $\mathbb{R}(-1)$ should aggregate the product of the probabilities of $\mathbb{P}(-1)$ and $\mathbb{Q}(0)$ (since $(-1) + 0 = -1$), the product of the probabilities of $\mathbb{P}(0)$ and $\mathbb{Q}(-1)$ (since $0 + (-1) = -1$), as well as the product of the probabilities of $\mathbb{P}(-1)$ and $\mathbb{Q}(-1)$ (since $(-1) + (-1) = -2$, and -2 is outside the limits of the numeric interval of the evaluation distance).

Note that this operator, $\otimes$, is commutative. Its neutral element is the distribution $\mathbb{O}$ for the ideal (or optimal) distribution where the probability that the evaluation difference between two peers is 0 is equal to 1, that is $p(X = 0) = 1$.

The other important operation needed is how to aggregate combined distances calculated from different sources (different peers). In this case, from several distance distributions, we select the one that is closer to $\mathbb{O}$, that is, the one that makes the leader and the student closer in their judgments. In the eigentrust algorithm, this would be equivalent to selecting the maximum combination (modeled as the product of the values in the links) instead of the used weighted sum of all the combinations. We note that other operators could be used here, for instance selecting the distribution with minimum entropy. In the following we define this operator.

DEFINITION 3.4. *Given probability distributions $\mathbb{P}$ and $\mathbb{Q}$ over the numeric interval $[a, b]$ we define $\mathbb{P} \oplus \mathbb{Q}$, as:*

$$\mathbb{P} \oplus \mathbb{Q} = \arg \min_{\mathbb{T}\in\{\mathbb{P},\mathbb{Q}\}} (emd(\mathbb{T}, \mathbb{O})) \quad (5)$$

with emd standing for the earth mover's distance [8].

Note that this operator, $\oplus$, is commutative and associative so the order in which we combine the trust distributions is irrelevant.

Next, we show how we use these operators following a similar approach to eigentrust. First, we store the direct trust distributions between ϵ's peers in a matrix C, where at the position (i, j) we store the current probability distribution between peers i and j: $\mathbb{T}_{i,j}$. We

store the indirect trust distributions between the leader ϵ and each community member in a vector t_ϵ, where at each position i we have $\mathbb{T}_{\epsilon,i}$. Initially, t_ϵ contains the probability distributions describing ignorance (i.e. the flat equiprobable distribution $\mathbb{F}$) in all rows. Let us call this initial vector t^0_ϵ. The t_ϵ vector is updated as follows:

$$t^{k+1}_\epsilon = C^T t^k_\epsilon \tag{6}$$

until $\|t^{k+1}_\epsilon - t^k_\epsilon\| < \eta$, 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^k_ϵ is defined, recalling previous definitions, as follows:

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

Finally, if a direct trust distribution is already built between ϵ and j, $\mathbb{T}_{i,j}$, then after each step of the algorithm, $t^{k+1}_{\epsilon,j}$ 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 $e^\alpha_\mu \in \mathcal{H}$ made at time t with a timestamp t, noted ${e^\alpha_\mu}^t$.
2. Record all direct trust distributions $\mathbb{T}_{i,j}$ with a timestamp t, noted $\mathbb{T}^t_{i,j}$, 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 $\mathbb{T}^t_{i,j}$, $\mathbb{T}^t_{i,j}$ 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 $\mathbb{T}^t_{i,j}$. 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^α_μ, 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^α_μ. As illustrated earlier, this is expressed as the conditional probability:

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

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:

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

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 $\mathbb{P}(X^\alpha \mid e^\alpha_\mu)$. 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 $\mathcal{O}^\alpha = \{e^\alpha_{\mu_1}, e^\alpha_{\mu_2}, \ldots, e^\alpha_{\mu_n}\}$ of a group of peers over the object α, we define the probability of ϵ's assessment being x as follows:

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

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:

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

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.

[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 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** $e^\alpha_j \in \mathcal{H}$ **do** ▷ Ordered by their timestamps
2: $diff_{i,j} = e^\alpha_i - e^\alpha_j$
3: $diff_{j,i} = e^\alpha_j - e^\alpha_i$
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: $t^0_\epsilon = \mathbb{F}$
16: **repeat**
17: $t^{k+1}_\epsilon = C^T t^k_\epsilon$ ▷ Equations 6 and 7
18: $error = \|t^{k+1}_\epsilon - t^k_\epsilon\|$
19: $t^{k+1}_\epsilon = t^k_\epsilon$
20: $t^{k+1}_\epsilon \leftarrow t'_\epsilon$ ▷ 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: $\mathbb{T}^{t' \rightsquigarrow t}_{i,j} = \Lambda(\mathbb{D}, \mathbb{T}^t_{i,j})$ ▷ 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

Algorithm 3 $calculateAssessments()$

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

criteria from the assignments: *focus, coherence, grammar* in the composition task and *in-time submission, requirements, lyrics* in the song vocabulary task.

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 $e_{\_}^{\alpha}$, as the average difference between them and the tutor assessments, that is:

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

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.

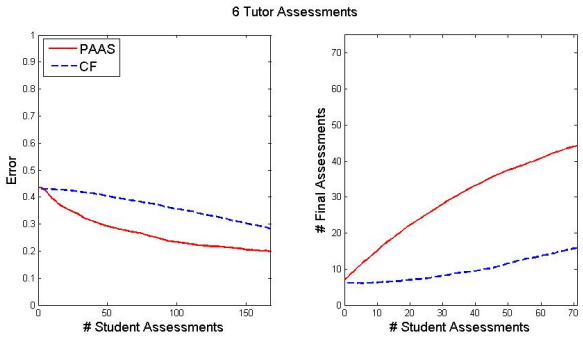
Figure 1 shows the results of PAAS and CF on three cases. As the assignments are different with different evaluation criteria we choose a criterion per group, necessarily different, so that we can have a larger number of assignments in the experiments. Three such pairings of criteria are shown in the 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:

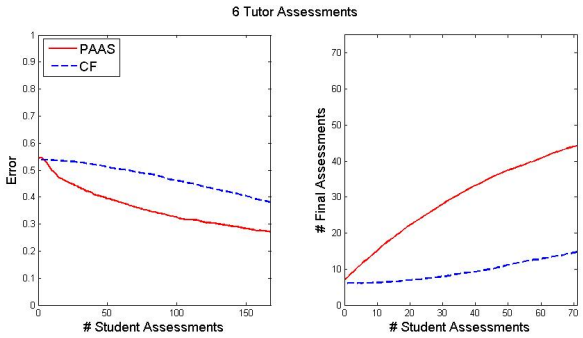
- We randomly select a tutor assessment for an assignment that has not been assessed yet, and we add this tutor assessment to PAAS Random.
- We select a tutor assessment for an assignment not yet assessed following the suggestion of the entropy heuristic, and we add this tutor assessment to PAAS Ranking.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

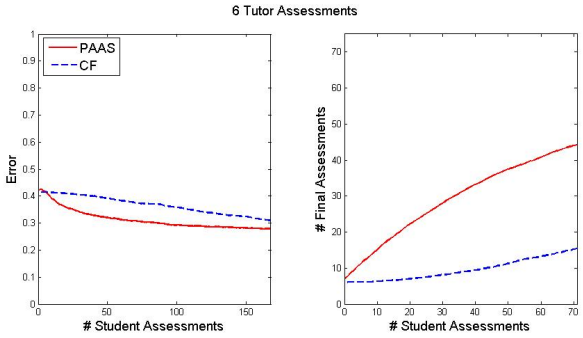
e.g. when the students are good 'recommenders', the heuristic may not be needed, in general it can improve the results when such recommendation quality is missing.



(a) focus and in-time submission criteria



(b) coherence and requirements criteria

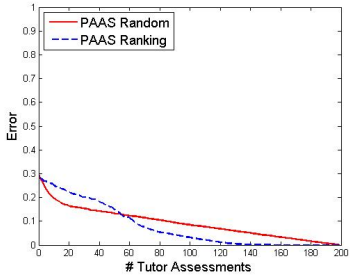


(c) grammar and lyrics criteria

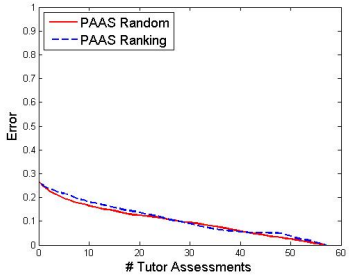
Figure 1: Experiments with Real Data: PAAS vs CF. We show the results for opinions on two criteria, one for each assignment. For instance in (a) we combine opinions on focus in the composition task and submission on time in the song vocabulary task.

6. CONCLUSIONS AND FUTURE WORK

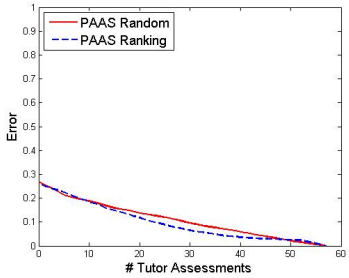
In this paper we have presented the Personalised Automated ASsessments model (PAAS), a trust-based assessment service that helps compute automated assessments from the perspective of a specific community member: a community leader. This computation essentially aggregates peer assessments, giving more 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



(a) Synthetic data



(b) Real data



(c) Randomised Real data

Figure 2: Experiments considering the entropy heuristic

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.

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.