# Collaborative assessments in on-line classrooms

Nardine Osman[1], Ewa Andrejczuk[1,2], Juan A. Rodriguez-Aguilar[1], and Carles Sierra[1]

[1] Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain
[2] Change Management Tool S.L., Barcelona, Spain
{nardine,ewa,jar,sierra}@iiia.csic.es

**Abstract.** Consider a teacher who needs to assess a large number of assignments. With massive open on-line courses (MOOCs) gaining momentum, it is now common for thousands of students to enrol in the same course, and hence manual assessment by teachers is simply unfeasible. Peer assessments is one way to go when auto-scoring approaches are not possible. Current on-line courses usually use a simple aggregation of peer assessments, but these suffer from two main pitfalls. First, simple aggregation does not take into consideration how reliable each peer assessment is. Second, simple aggregation calculates what the students think of an assignment as opposed to what the teacher (the far more important assessment source) thinks of it. This paper proposes two different models to address these two different pitfalls. These models lay the foundation for future work, where we intend to combine both models into a single one that addresses both pitfalls at once.

## 1 Introduction

Self and peer assessment have clear pedagogical advantages. Students increase their responsibility and autonomy, get a deeper understanding of the subject, become more active and reflect on their role in group learning, and improve their judgement skills. However, in this paper, we are interested in relying on peer assessments for reducing the marking load of teachers. This is specially critical when teachers face the challenge of marking large quantities of students as needed in the increasingly popular Massive Open Online Courses (MOOC). Existing platforms for on-line courses, like Coursera (coursera.org), apply a simple assessment aggregation method. In this paper, we propose two different models to address two different pitfalls of the simple aggregation method. The first pitfall is that a simple aggregation does not take into consideration how good is each peer assessment.[3] The academic publishing field has attempted to address the pitfall of the simple aggregation method when aggregating reviews by asking each reviewer to specify their confidence level. This confidence level is then used to weigh the review. In this paper, we propose the Collaborative Judgements (CJ) model, our first proposed model, where we go even further by stating that

[3] Some MOOCS are using the mean or median of the students peer assessments.

the confidence level provided by a reviewer is not sufficient as it completely relies on how much objective is that reviewer in assessing himself. As such, our CJ model proposes that peers judge each others assessments (or reviews). As such, the weight used when aggregating assessments is then based on the judgements that this assessment has received. In other words, instead of relying on one's self-confidence, we rely on how other peers judge each other's assessments.

The second pitfall of the simple aggregation method of on-line courses is that they solely aggregate student assessments, that is they aggregate what students think of their peers assignment results. In classrooms, we believe the teacher's assessment of an assignment is far more important (and credible) than the student assessments. As such, we propose the personalised automated assessment service (PAAS), which modifies peer assessments to approximate the unknown teacher assessments. It basically predicts how the teacher will assess an assignment, given how the fellow peers have assessed it. In other words, the aggregation is tuned to the point of view of the teacher.

While each of the models addresses a different problem of the simple aggregation method used by current on-line courses, these models lay the foundation for future work, where we intend to combine both models into a single one that addresses both pitfalls at once.

The rest of this paper is divided as follows. Section 2 opens with an overview of the related work; Sections 3 and 4 follow with the two models, Collaborative Judgements and PAAS; Section 5 presents our experimental setting that we have setup to evaluate our models, whereas our evaluation results are presented next in Section 6; and finally, Section 7 closes with a brief summary of our work and our plans for future work.

## 2 Background

Previous works have proposed several methods to generate student assessments based on peer-student assessments. Table 1 categorises the related work, including our CJ and PAAS models, with respect to whether the model aggregates peer opinions by tuning them to the teacher's view or not, and whether they weigh assessments by their reliability. We briefly present these models next.

| | **WbR** | **¬ WbR** |
|---|---|---|
| **T2T** | Future Work | **PAAS**, LocPat [5], Collaborative Filtering [11] |
| **¬ T2T** | **CJ**, CrowdGrader [1], PeerRank [13], Piech et al. [9] | Simple aggregation (mean or median) |

**Table 1.** Categorisation of related work w.r.t. whether the model aggregates peer opinions by tuning them to the teacher's view (Tuned to Teacher – T2T) or not, and whether they weigh assessments by their reliabiltiy (Weighed by Reliability – WbR)

CrowdGrader [1] is a framework which defines a crowdsourcing algorithm for peer assessments. The authors claim that, when performing assessments, 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 then 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 $\alpha$ is calculated as a weighted average of the grades of the agents evaluating $\alpha$, and thus the grades of $\alpha$'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 PageRank, where web-pages are ranked according to the ranks of the web-pages that link to them.

Piech et al. [9] propose 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 (mean of assessments), which does not take into account individual biases and reliability.

Next, we present relevant recommender systems, as recommender systems tune their results to the point of view of a specific person (as in PAAS). One relevant system is LocPat [5], a generalised framework for personalised recommendations in agent networks. LocPat builds trust measures based on mining the graph of an agent network. For instance, trustworthy relationships are discovered by studying the link structure (e.g., the number of common neighbours). Then, it suggests to a specific requester (who requests a recommendation in the agent network) a list of trustworthy agents for the requester to interact with.

Collaborative Filtering [11] is a classical social information filtering algorithm that recommends content to users based on their previous ratings, exploiting similarities between the tastes of different users. In summary:

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.

## 3 CJ: Collaborative Judgements Model

Recall that the collaborative judgements model (CJ) aggregates peer assessments by weighing each assessment with respect to its reliability, where reliability in this model is referred to as the peer's reputation, and it is based on the judgements that this peer has received. In this section we detail our CJ algorithm, but first, we introduce the notation, which we will use in the rest of this section.

### 3.1 Notation

We say an *appraisal* is a tuple $\langle P, R, E, o, v\rangle$, where $P = \{p_i\}_{i\in\mathcal{P}}$ is a set of works (that is students' solutions for the assignment); $R = \{r_j\}_{j\in\mathcal{R}}$ is a set of peers (or students) evaluating someone' works; $E = \{e_i\}_{i\in\mathcal{E}} \cup \{\bot\}$ is a totally ordered evaluation space, where $e_i \in \mathbb{N}$ and $e_i < e_j$ iff $i < j$ and $\bot$ stands for the absence of evaluation; $o : R \times P \to E$ is a function giving the opinions of peers on someone's work; and $v : R \times R \times P \to E$ is a function giving the judgements of peers over opinions on someone' works.

In general we might have different dimensions of evaluation, that is a number of $E$ spaces over which to express opinions and judgements. For instance, originality, soundness, etc. Nonetheless, here for simplicity reasons we will assume that the evaluation of a work is made over a single dimension. Actually, the 'overall' opinion is what is aggregated in many real systems.

### 3.2 The CJ Algorithm

The steps of the CJ algorithm applied over an appraisal $\langle P, R, E, o, v\rangle$ are:

**Step 1.** Compute the *agreement level* between each pair of peers $r_i$ and $r_j$ as a function $a : R \times R \to [0, 1] \cup \{\bot\}$. This computation involves the set of works jointly assessed by peers $r_i$ and $r_j$, which we will formally define as $P_{ij} = \{p_k \in P | o(r_i, p_k) \neq \bot, o(r_j, p_k) \neq \bot\}$. If two peers jointly reviewed works, then their agreement level is based on the similarities of their opinions on common works as well as on their judgements. Formally, we say:

$$a(r_i, r_j) = \begin{cases} \frac{\sum_{p_k \in P_{ij}} s(r_i, r_j, p_k)}{|P_{ij}| \cdot d} & \text{if } P_{ij} \neq \emptyset \\ \bot & \text{otherwise} \end{cases} \quad (1)$$

where $d$ is the maximum distance in the evaluation space and:

$$s(r_i, r_j, p_k) = \begin{cases} v(r_i, r_j, p_k) & \text{if } P_{ij} \neq \emptyset \text{ and } v(r_i, r_j, p_k) \neq \perp \\ Sim(o(r_i, p_k), o(r_j, p_k)) & \text{if } P_{ij} \neq \emptyset \text{ and } v(r_i, r_j, p_k) = \perp \\ \perp & \text{otherwise} \end{cases} \quad (2)$$

$Sim$ stands for an appropriate similarity measure. When no explicit judgements are given, we use the similarity between opinions as a heuristic. This is based on the following assumption: the more similar an opinion is to my opinion, the better I am bound to judge that opinion.

**Step 2.** Compute a complete *Trust Graph* as an adjacency function matrix $C = \{c_{ij}\}_{i,j \in R}$ such that:

$$c(r_i, r_j) = \begin{cases} a(r_i, r_j) & \text{if } a(r_i, r_j) \neq \perp \\ \max\limits_{h \in chains(r_i, r_j)} \prod\limits_{(k,k') \in h} a(r_k, r_{k'}) & \text{otherwise} \end{cases} \quad (3)$$

where $chains(r_i, r_j)$ is the set of sequences of peer indexes connecting $i$ and $j$. Formally, a chain $h$ between peers $i$ and $j$ is a sequence $\langle l_1, \ldots, l_{n_h} \rangle$ such that $l_1 = i, l_{n_h} = j$, and $a(r_k, r_{k+1}) \neq \perp$ for each pair $(k, k{+}1)$ of consecutive values in the sequence. To compute this step we use a version of Dijkstra's algorithm that instead of looking for the shortest path (using + and min as mathematical operations), it looks for the path with the largest edge product (using $\cdot$ and max as mathematical operators).

**Step 3.** Compute a *reputation* for each peer in $R$, $\{t_i\}_{i \in R}$. We follow the notion of transitive trust: If a peer $i$ trusts any peer $j$, it would also trust the peers trusted by $j$. Since this principle is employed by the Eigentrust algorithm [7], we use it to compute peer reputations. The use of Eigentrust allows us to obtain a global trust value for each peer by the repeated and iterative multiplication and aggregation of reputation values until the trust grades for all peers converge to stable values. Note that the trust graph generated in step 2 is aperiodic and strongly connected as required by the Eigentrust algorithm. Furthermore, we normalise the powers of the matrix $C$ at each step to ensure its convergence. In vectorial notation, the trust vector is assessed as $\bar{t} = \lim_{k \to \infty} \bar{t}^{k+1}$ with $\bar{t}^{k+1} = C^T \bar{t}^k$ and $\bar{t}^0 = \bar{e}$ being $\bar{e}_i = 1/|\bar{e}|$.

**Step 4.** Compute the *collective opinion* on each work as a weighted average of the opinions of those that expressed an opinion on the work. In other words, given a work $p_j$, we only consider the opinions of those peers that reviewed $p_j$, which we formally define as $R_j \subseteq R, R_j = \{r \in R | o(r, p_j) \neq \perp\}$. We can then compute the collective opinion on a work $p_j$ as a weighted average of the opinions of the peers in $R_j$ using as weights the peers' reputations. Finally, the collective opinion computed by our collaborative judgements algorithm for a work $p_j$, noted as $o_{CJ}(p_j)$, is:

$$o_{CJ}(p_j) = \frac{\sum_{r \in R_j} \bar{t}_r \cdot o(r, p_j)}{\sum_{r \in R_j} \bar{t}_r} \quad (4)$$

where $\bar{t}_r$ stands for the reputation value of peer $r$ as computed in Step 3.

**Step 5.** Generate a partial ranking based on the set of collective opinions $O_{CJ}(P)$. CJ sorts works in descending order by the collective opinion values. Thus, the work with the highest value of collective opinion gets the first ranking position. Works with equal collective opinion receive the same ranking number, and the work(s) on the next position receive the immediately following ranking number (i.e. bucket index). The procedure continues until CJ assigns bucket indexes to all works.

### 3.3 Comparing CJ to Related Literature

As illustrated by Table 1, several approaches have been proposed to aggregate peer assessments by weighing each with respect to its reliability. The main difference between these approaches and CJ is the usage of judgement information over such assessments. Here, we focus on comparing CJ to those models.

The reliability of a student in the Crowd Grader model is measured by comparing the student's assessments with the assessments of their fellow students for the same assignments. In other words, when one student's assessments are similar to his friends, then he is considered reliable, and vice versa. We believe our proposed CJ model is more accurate that the Crowd Grader model as one's assessment need not be similar to others, but needs to be highly viewed by others. For instance, think of the clever student who always makes excellent observations that have gone unnoticed by others.

PeerRank is another model that takes into consideration the reliability of an assessment as the reliability of the student providing that assessment. And the reliability of a student is calculated following an approach similar to Google's PageRank, where the assessment of each student is calculated as a weighted average of the assessments of the students evaluating the student in question. In other words, PeerRank assumes that if you are assessed highly by reliable friends, then your own assessments will be more reliable. CJ, on the other hand, differentiates between one performing well in an assignment and one who provides good assessments. For instance, one might fail at solving a difficult problem, but might still be exceedingly capable of spotting out the mistakes of others. Unlike PeerRank, CJ allows for that, because one's assessments are judged by his peers independently of his own performance in an assignment.

The model by Piech et al. [9] also uses reliability of assessments, where reliability is defined as the distance between one's score and the true score of an assignment. They assume the existence of a true score for every assignment, and they then use Gibbs sampling to infer these values. CJ on the other hand does not assume any true scores exist and reliability simply depends on how others judge one's assessment.

## 4 The PAAS Model

After presenting our CJ model that introduces judgements as a way for calculating the reliability of an assessment, we now present our PAAS model that tunes assessments to the point of view of the teacher.

### 4.1 Notation and Problem Definition

Let $\epsilon$ represent the teacher who needs to assess a large set of students' works $\mathcal{I}$, and let $\mathcal{P}$ be a set of peers able to assess works in $\mathcal{I}$.

We define a peer assessment $e_\mu^\alpha$ (also referred to as evaluation or opinion) as an element from a numerically ordered evaluation space $\mathcal{E}$, where $\alpha \in \mathcal{I}$ is the work being evaluated and $\mu \in \{\epsilon\} \cup \mathcal{P}$ is the evaluating peer. We define an automated assessment $e_{\_}^\alpha$ for student work $\alpha$ 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 $\mathcal{E} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}$. The distribution $\{\mathbf{1} \mapsto 0, \mathbf{2} \mapsto 0, \mathbf{3} \mapsto 0, \mathbf{4} \mapsto 1\}$ would represent the best possible assessment, whereas the distribution $\{\mathbf{1} \mapsto 0, \mathbf{2} \mapsto 1/2, \mathbf{3} \mapsto 1/2, \mathbf{4} \mapsto 0\}$ would represent that the quality of the homework is most probably average, 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 work $\alpha$.

### 4.2 The PAAS Algorithm

The ultimate goal of our model is to compute the probability distribution of $\epsilon$'s assessment over a certain work $\alpha$, given the assessments of several peers over that same work $\alpha$. In other words, what is the probability that $\epsilon$'s assessment of $\alpha$ is $x$ given the set of peers' assessments $\mathcal{O}^\alpha$?

We base the computation 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 works. In other words, our intuition is that we expect $\epsilon$ will tend to agree with $\mu$'s assessment on a work if her trust on $\mu$ is high. Otherwise, $\epsilon$'s assessment will probably be different. To build a trust measure between $\epsilon$ and $\mu$ we perform a sort of analogical reasoning: if in the past $\mu$ gave opinions that were similar to $\epsilon$'s opinions to a certain degree (trust), then $\epsilon$ is likely to coincide with $\mu$'s opinion again now to the same degree. The steps needed for calculating the teacher's assessment are presented next.

**Step 1. How much should I trust a peer?** $\epsilon$ needs to decide how much can she trust the assessment of a peer $\mu$. We base this trust measure on two intuitions. First, if $\epsilon$ and $\mu$ have both assessed the same work in the past, then the similarity of their assessments for that work can give a hint on how close their judgements/thinking are. When there are no works evaluated by both $\epsilon$ and $\mu$, $\epsilon$ would not know how much to trust $\mu$'s assessments. Second, and to cover this latter situation, we approximate the unknown trust between $\epsilon$ and $\mu$ by transitivity over the path with direct trust links between $\epsilon$ and $\mu$. In the following, we make these two intuitions concrete through two different types of trust relations: *direct trust* and *indirect trust*.

*Direct Trust.* Direct trust is the trust relation that emerges between two people or two agents that have already assessed the same works in the past.

We will define the direct trust between two peers $i$ and $j$ as a probability distribution $\mathbb{T}_{i,j}$ over assessment differences built from the historical data of previous evaluations performed by $i$ and $j$. First, we define the *evaluation difference* between two assessments performed by $i$ and $j$ as:

$$\mathit{diff}\,(e_i^\alpha, e_j^\alpha) = e_i^\alpha - e_j^\alpha \tag{5}$$

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 $\mathit{diff}\,(e_i^\alpha, e_j^\alpha) = 0$, it means that $i$ and $j$ provide the same assessment for $\alpha$. If $\mathit{diff}\,(e_i^\alpha, e_j^\alpha) > 0$, it means that $i$ *over rates* $\alpha$ with respect to $j$, if $\mathit{diff}\,(e_i^\alpha, e_j^\alpha) < 0$, it means that $i$ *under rates* $\alpha$ with respect to $j$. Note that $\mathit{diff}\,(e_i^\alpha, e_j^\alpha) \neq \mathit{diff}\,(e_j^\alpha, e_i^\alpha)$.

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 a work 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 1.** *Given a numeric evaluation space $\mathcal{E} = [0, b]$, a* Trust Distribution *is any probability distribution over the differences in $\mathcal{E}$, that is the interval $[-b, b]$.*

We now explain how we build direct trust distributions computationally, based on previous experiences. We use an information theory approach where the behaviour of the studied phenomenon is modelled 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 a work $\alpha$ 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: $x = \mathit{diff}\,(e_i^\alpha, e_j^\alpha)$.
2. We increase the probability of $x$ in $\mathbb{T}_{i,j}$ $(p(X = x))$ as follows:

$$p(X = x) = p(X = x) + \gamma \cdot (1 - p(X = x)) \tag{6}$$

The update is based on increasing the current probability $p(X = x)$ by a fraction $\gamma \in [0, 1]$ of the total potential increase $(1 - p(X = x))$. For instance, if the probability of $x$ is 0.6 and $\gamma$ 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 $\gamma$ 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 [12]. The entropy-based approach updates $\mathbb{T}_{i,j}$ such that: (1) the value $p(X=x)$ 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 $p(X=x)$ and that is at a minimal distance from the previous $\mathbb{T}_{i,j}$:

$$\mathbb{T}_{i,j}(X) = \underset{\mathbb{P}'(X)}{\arg\min} \sum_{x'} p(X=x') \log \frac{p(X=x')}{p'(X=x')} \quad \text{such that} \quad \{p(X=x) = p'(X=x)\} \tag{7}$$

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

*Indirect Trust.* Indirect trust is the trust relation that is deduced between peers when they have not assessed any works 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 [7]. In Eigentrust the transitivity in trust is based on products and additions of positive real numbers. For example, for a difference in opinion $x$ between peers $\beta$ and $\alpha$ and a difference in opinion $y$ between $\beta$ and the teacher, the overall difference between the teacher and $\alpha$ is $z = x + y$, when we are in an ordinal space. However, in our case we need to define operators to compute the transitive trust distribution from two *distributions*. When we move to probabilities, we then say that $\mathbb{P}(z) = \mathbb{P}(x) * \mathbb{P}(y)$, as we assume independence between opinions. Following this intuition, we define the combined distance distribution between two peers as follows.

**Definition 2.** *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:*

$$r(X=x) = \begin{cases} \sum\limits_{x_1+x_2=x} p(X=x_1) * q(X=x_2) & \text{if } x \in (-b, b) \\ \sum\limits_{x_1+x_2\leq -b} p(X=x_1) * q(X=x_2) & \text{if } x = -b \\ \sum\limits_{x_1+x_2\geq b} p(X=x_1) * q(X=x_2) & \text{if } x = b \end{cases} \tag{8}$$

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 $\geq$ (in cases $x = b$ and $x = -b$) 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 next problem to tackle 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 teacher and the student closer in their judgements. In the Eigentrust algorithm, this would be equivalent to selecting the maximum combination (modelled 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.** *Given probability distributions* $\mathbb{P}$ *and* $\mathbb{Q}$ *over the numeric interval* $[a, b]$ *we define* $\mathbb{P} \oplus \mathbb{Q}$*, as:*

$$\mathbb{P} \oplus \mathbb{Q} = \arg \min_{\mathbb{T} \in \{\mathbb{P}, \mathbb{Q}\}} (emd(\mathbb{T}, \mathbb{O})) \tag{9}$$

*with emd standing for the earth mover's distance [10].*

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 $\epsilon$'s peers in a matrix $C^T$, 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 teacher $\epsilon$ and each community member in a vector $t_\epsilon$, where at each position $i$ we have $\mathbb{T}_{\epsilon,i}$. Initially, $t_\epsilon$ contains the probability distributions describing ignorance (i.e. the uniform distribution) in all rows. Let us call this initial vector $t^0_\epsilon$. In Eigentrust, the $t_\epsilon$ vector is updated as follows:

$$t^{k+1}_\epsilon = C^T t^k_\epsilon \tag{10}$$

until $\|t^{k+1}_\epsilon - t^k_\epsilon\| < \eta$, where $\eta$ is a specified threshold to determine that we have reached a fix point. The trust vector $t_\epsilon$ then converges after a certain amount of iterations. In this way, the trust that $\epsilon$ has on $i$ is built aggregating the direct trust distributions between community members and peer $i$ weighted by the trust (initially ignorance) that $\epsilon$ has on each community member. In our model, however, the product between matrix $C^T$ and $t^k_\epsilon$ is defined using the previous definitions of $\otimes$ and $\oplus$, resulting in:

$$t^{k+1}_{\epsilon,j} = \bigoplus_{0<i\leq n} \mathbb{T}_{i,j} \otimes \mathbb{T}^k_{\epsilon,i} \tag{11}$$

Finally, if a direct trust distribution is already built between $\epsilon$ 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.

*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:

$$\mathbb{T}^{t \rightsquigarrow t'} = \Lambda(\mathbb{D}, \mathbb{T}^t) \tag{12}$$

where $\Lambda$ is the *decay function* satisfying the property: $\lim_{t' \to \infty} \mathbb{T}^{t \rightsquigarrow t'} = \mathbb{D}$. In our implementation, we adopt the decay function of [8].

Naturally, to implement such a decay mechanism in our model, we will need to add a timestamp $t$ to every assessment ($e_\mu^{\alpha^t}$) and every trust distribution, direct or indirect, ($\mathbb{T}_{i,j}^t$). Note that while the timestamp of an assessment is fixed, the timestamp of a trust distribution should refer to the timestamp of the latest assessment that modified this distribution.

**Step 2: What to believe when a peer gives an opinion?** Given a peer assessment $e_\mu^\alpha$, the question now is how to compute the probability distribution of $\epsilon$'s assessment. In other words, what is the probability that $\epsilon$'s assessment of $\alpha$ is $x$ given that $\mu$ evaluated $\alpha$ with $e_\mu^\alpha$. 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 $\epsilon$ would tend to agree with $\mu$'s assessment if his trust on $\mu$ is high (that is, the expected assessment difference between their assessments is close to 0). Otherwise, $\epsilon$'s assessment would probably be different. We perform then a sort of analogical reasoning: if in the past $\mu$ gave assessments with a certain evaluation difference with respect to $\epsilon$, then this will probably happen again now.

We thus calculate the above conditional probability simply as:

$$p(X^\alpha = x \mid e_\mu^\alpha) = \begin{cases} \sum_{y \leq \mathit{diff}(x, e_\mu^\alpha)} \mathbb{T}_{\epsilon,\mu}(y) & \text{if } x = 0 \\ \sum_{y \geq \mathit{diff}(x, e_\mu^\alpha)} \mathbb{T}_{\epsilon,\mu}(y) & \text{if } x = b \\ \mathbb{T}_{\epsilon,\mu}(\mathit{diff}(x, e_\mu^\alpha)) & \text{otherwise} \end{cases} \tag{13}$$

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 teacher gives a 0 in the assessment. Similarly for $b$. For example, assume $\mu$'s assessment is 2 when the maximum mark is 3, we are calculating the probability of $\epsilon$'s assessment, and $\epsilon$ usually over rates $\mu$ by 2 marks. The probability of $\epsilon$'s assessment being 2 will essentially be $\mathbb{T}(0)$ (since the difference $2-2=0$). However, the probability of $\epsilon$'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 $\epsilon$'s assessment being 3 aggregates $\mathbb{T}(1)$, $\mathbb{T}(2)$, and $\mathbb{T}(3)$.

**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 $\epsilon$'s assessment on $\alpha$ given the assessment of a peer $\mu$ on $\alpha$. But what does $\epsilon$ do when there is more than one peer assessing $\alpha$?

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 work $\alpha$, we define the probability of $\epsilon$'s assessment being $x$ as follows:

$$p(X^\alpha = x \mid \mathcal{O}^\alpha) = \begin{cases} \bigvee_{i=1}^{n} (\mathbb{I}(\mathbb{T}_{\epsilon,\mu_i}) \cdot p(X^\alpha = x \mid e^\alpha_{\mu_i})) & \sum_{i=1}^{n} \mathbb{I}(\mathbb{T}_{\epsilon,\mu_i}) > \delta \\ 1/n & \text{otherwise} \end{cases} \tag{14}$$

where $\vee$ is an operator that combines probabilities assuming the sources are independent:[4] $a \vee b = a + b - a * b$, 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 $\epsilon$'s assessment being $x$ given the set of opinions $\mathcal{O}^\alpha$ is a disjunction of the probabilities of $\epsilon$'s assessment being $x$ given each assessment $e^\alpha_{\mu_i} \in \mathcal{O}^\alpha$ and diminished by the information content of the assessment 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 $\delta$) 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.

[4] 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.

**Step 4: What should be evaluated next?** The previous three steps allow to compute assessments of students' works that have not been assessed by $\epsilon$, 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:

$$\mathbb{H}(\mathbb{P}(X^\alpha \mid \mathcal{O}^\alpha)) = \sum_{x \in X^\alpha} p(X^\alpha = x \mid \mathcal{O}^\alpha) \cdot \ln p(X^\alpha = x \mid \mathcal{O}^\alpha) \tag{15}$$

We will explore in the experiments a heuristic that aims at reducing the number of assessments made by the teacher. In other words, what work should be assessed next by $\epsilon$ in order to maximally decrease the overall uncertainty? For example, what students' works 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 $\epsilon$ evaluates works by decreasing value of the entropy of their assessment distribution, that is the next work $\alpha$ that the teacher should assess is:

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

### 4.3 Comparing PAAS to Related Literature

What is fundamentally different between our PAAS model and the related work presented earlier is that the computation of our automated assessments is tuned to the perspective of a specific community member, a teacher. We clarify that our target is to accurately estimate those unknown assessments *from the teacher's point of view*. PAAS aggregates peer assessments giving more weight to those peers that are trusted by the teacher. Such trust metrics are built, as we will see shortly, using probability distributions based on the history of past assessments between the teacher and his/her peers, rather than using aggregations.

Unlike LocPat, PAAS bases trust measures on the similarity of assessments. If the student's assessments are similar to the teacher, then the student will be considered trustworthy by the teacher. LocPat, on the other hand, bases trustworthiness on characteristics of the social graph, which we believe are not as important as the similarity of assessments in the specific domain of automatic assessment calculation for online courses.

In the experimental evaluation of our system, we compare PAAS to Collaborative Filtering (CF), since CF (like PAAS) 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.

## 5 The English Classroom Experiment

### 5.1 Experimental Setting

In this section, 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. 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.

We calculate the error of the generated assessments, noted as $e^{\alpha}_{\_}$, as the average difference between them and the tutor assessments, that is:

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

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 this experiment, we compare PAAS with the well known Collaborative Filtering (CF) algorithm [11]. (Please note that the CJ model is evaluated in the following section.) 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 teacher'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. At each iteration: (1) one student assessment is selected randomly from the set of student assessments and added to PAAS and CF; and (2) 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.

### 5.2 Results

Figure 1 shows the results of comparing PAAS tp CF. 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. Figure 1 presents the results of one such pairing of criteria, although the results of other pairings (not presented here) are very similar. It is clear 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 teacher (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).

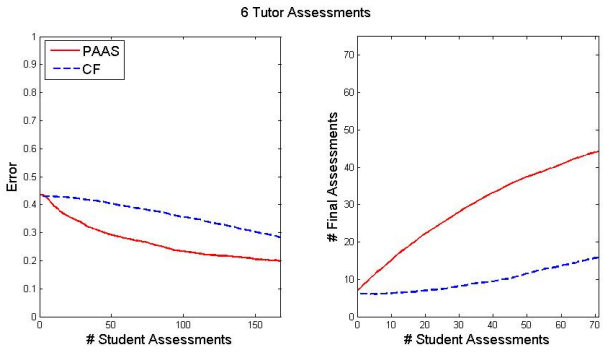


**Fig. 1.** Results of the English classroom experiment, focusing on the two criteria "focus" and "in-time submission"

## 6 The Simulation-Based Experiment

Students failed to provide sufficient judgements to allow us to properly evaluate our CJ model. As such, the data obtained from the real experiment of Section 5 was used to evaluate PAAS, whereas we simulated data for evaluating CJ. This section presents our simulation.

### 6.1 Experimental Setting

We assume a set $P = \{p_1, \ldots, p_n\}$ of students' works (assignments) and a function for their true quality in a range $[0, 1]$,[5] $q : P \to [0, 1]$. We use the following evaluation space $E = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$.

We use beta distributions to model students' opinions and judgements as it is an appropriate distribution to simulate a behaviour that is subject to random variation and is limited on both extremes, i.e. represents processes with natural lower and upper boundaries [6].

We model two types of student assessors: good and bad, with the following behaviour:

- *Good student assessor.* She provides fair opinions and fair judgements. Her opinion on any work $p_k$ is always close to its true quality $q(p_k)$. We assume that the absolute value of the difference between the opinion of a student assessor and the true quality of a work (as a percent) follows a beta distribution, $Beta(\alpha, \beta)$, very positively skewed, for instance with $\alpha = 1$ and $\beta = 30$. For each work $p_k$ reviewed by a good student assessor, we sample the assessor's associated beta distribution for a percentage difference, apply it to the work quality $q(p_k)$ (up or down randomly) and round the result to fit an element in $E$. Her judgements on someone's opinion are close to 0 if that opinion is far from the true quality of the work, and close to 1 otherwise. We implement this as the following function:

  $$v(r_i, r_j, p_k) = 1 - |o(r_j, p_k) - q(p_k)|$$

  and self-judgements from $Beta(5, 2)$, slightly negatively skewed.
  We assume that when a good student assessor judges a bad student assessor she samples a value in $E$ from a beta distribution rather positively skewed: $Beta(2, 40)$. The intuition is that good student assessor poorly mark bad student assessor.
- *Bad student assessor.* She provides unfair opinions, because she is incompetent, but provides reasonable judgements as she can interpret the opinions of others as being informative or not. Thus, we sample opinions from $Beta(20, 12)$ (rather central with a slight positive skew), judgements for good assessments (or opinions) and self-judgements from $Beta(5, 2)$ as for good student assessors (negatively skewed), and judgements on bad reviews from $Beta(2, 5)$ (slightly positively skewed). The overall idea is that bad student assessors stay mostly in the central area of the evaluation space.

We use $Sim(x, y) = (|E| - 1 - |\tau(x) - \tau(y)|)/(|E| - 1)$ as a simple linear similarity function where $\tau$ is a function that gives the position of an element in the ordered set $E$.

[5] Assessing the true quality of an object may be difficult and it is certainly a domain dependent issue.

### 6.2 Simulation Results

**Analysing the accuracy of opinions.** Here, we consider the accuracy of a collective opinion on a work as the difference between that opinion and the true quality of that work. We compare CJ to the algorithm that weighs opinions with the assessors' self-assessments. This is because the models presented in Section 2 that consider the reliability of an assessment have not been adopted yet, and as such, we compare to one that is commonly used in academia (specifically in online conference management systems, such as Confmaster or Easychair). There, each reviewer states his self-confidence when it comes to the assessment he has given, and the aggregation of assessments uses this measure as a weight describing the reliability of assessments. We will call this simple algorithm the *Self-Assessment Weighted Algorithm* (SAWA).

We compare the accuracy of the opinions computed by CJ and SAWA as the percentage of good student assessors increases. We compute the accuracy of both CJ and SAWA as the mean absolute error of their opinions with respect to the true qualities using the following expressions:

$$MAE_{CJ} = \frac{\sum_{p \in P} |o_{CJ}(p) - q(p)|}{|P|} \qquad MAE_{SAWA} = \frac{\sum_{p \in P} |o_{SAWA}(p) - q(p)|}{|P|}$$

where $q$ is a function that yields the true quality of each work. Figure 2 plots the percentage error reduction of CJ with respect to SAWA (computed as $(1 - \frac{MAE_{CJ}}{MAE_{SAWA}}) \cdot 100$) by aggregating the values obtained from 30 runs of each algorithm (each run samples all the distributions and thus generates different collective assessments). Note that CJ outperforms SAWA, as it is much more resilient to bad student assessors. As a matter of fact, as opposed to SAWA that treats all student assessors equally, CJ is designed to detect bad student assessors and diminish the importance of their opinions by the usage of the reputation measure. We observe that CJ's gains become larger than 20% and statistically significant for percentages of good student assessors between 20% and 80%.

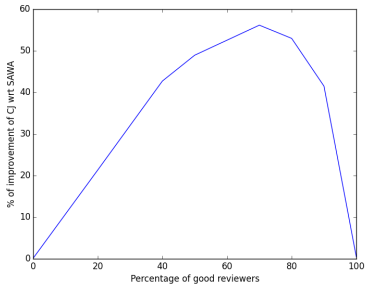


**Fig. 2.** Accuracy of opinions: percentage of error improvement of CJ over SAWA

**Analysing the accuracy of rankings.** Now we compare the accuracy of the rankings produced by CJ and SAWA with respect to the ranking resulting from the true quality of works. In order to compare two partial rankings, we largely rely on the work of [3], which provides sound mathematical principles to compare partial rankings. In particular, we use one of the four metrics presented in [3], the so-called *Kendall distance with penalty parameter p*, and we employ the normalised Kendall distance [4] with penalty factor $p = 0.5$.

We employed the partial rankings resulting from 30 runs of CJ and SAWA. We note by $\sigma_1^{CJ}, \ldots, \sigma_{30}^{CJ}$ the partial rankings produced by CJ by $\sigma_1^{SAWA}, \ldots, \sigma_{30}^{SAWA}$ the partial rankings produced by SAWA, and by $\sigma^q$ the true ranking. Then, for each partial ranking computed by CJ and SAWA, we compute its normalised Kendall distance with respect to the true ranking. On the one hand, we assess the average Kendall distance of the rankings produced by CJ as $K_{CJ} = \frac{\sum_{i=1}^{30} \tilde{K}^{(0.5)}(\sigma_i^{CJ}, \sigma^q)}{30}$. On the other hand, we assess the average Kendall distance of the rankings produced by SAWA as $K_{SAWA} = \frac{\sum_{i=1}^{30} \tilde{K}^{(0.5)}(\sigma_i^{SAWA}, \sigma^q)}{30}$.

Figure 3 (left) plots the average Kendall distance of the rankings produced by CJ with respect to the true ranking, namely $K_{CJ}$, as the number of good student assessors increases. We observe that the distance between CJ rankings and the true ranking quickly decreases as the number of good student assessors increases. Notice that beyond 50% of good student assessors the distance drops below 0.1. That means that CJ can produce rather accurate rankings despite the presence of a large ratio of bad student assessors.

Figure 3 (right) shows the accuracy gain of CJ with respect to SAWA. We calculate such accuracy gain as $\frac{K_{SAWA} - K_{CJ}}{K_{SAWA}} \cdot 100$. We observe that the accuracy gain yield by CJ as the number of good student assessors grows, going beyond a 40% gain with 80% good student assessors. Similarly to experiment 6.2, the graph clearly shows that CJ performs significantly better even when the number of bad student assessors is high. We see that CJ has been able to discriminate poor assessments, while SAWA treats all student assessors equally. We observe also that CJ benefits larger from good student assessors than SAWA.

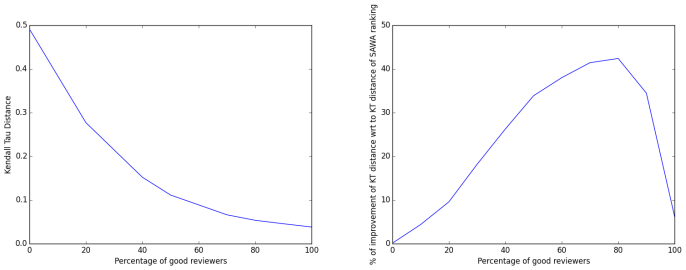


**Fig. 3.** (Left) Normalised Kendall Ranking distance calculated for CJ ranking and true ranking of the works. (Right) Percentage of error decrease measured as a Kendall distance between rankings produced by CJ and SAWA and true ranking of works for increasing percentages of good student assessors.

**Analysing the robustness against bad student assessors.** As mentioned before, we model the opinion of good student assessors with a $Beta(\alpha, \beta)$ very positively skewed from which we sample the difference between the student assessor's opinion and the true quality. With $\alpha = 1$ and $\beta > 30$ the expert is frequently telling the true quality in her opinions (specially because we discretise the sampled values into our evaluation space —i.e. almost all the distribution mass is rounded to a distance of 0 with respect to the true quality). In figure 4 we plot the improvement of CJ with respect to SAWA for $\alpha = 1$ and increasing values of $\beta$ (better student assessor behaviour). We observe that the algorithm outperforms SAWA by 10% when the student assessor is frequently mistaken ($\beta = 5$). This shows that even when good student assessors give frequently inaccurate opinions, CJ is still able to capture them and increases the importance of their assessments. The improvement asymptotically grows to 51% with increasing quality of the student assessor behaviour.

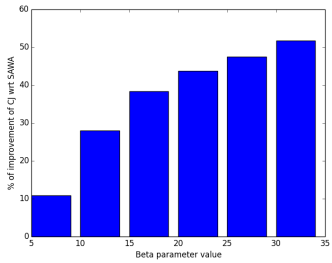


**Fig. 4.** Improvement of CJ over SAWA as assessors' quality increases (with $\alpha = 1$ and increasing $\beta$ values, and a population with 50% good and 50% bad student assessors).

## 7 Conclusion

This paper proposes two different models for tackling two different problems of aggregating peer assessments in online classrooms. The Collaborative Judgements (CJ) model, our first proposed model, requires that peers judge each other's assessments. This helps assess the reliability of student assessments: the weight used when aggregating assessments is based on the judgements that this assessment has received. The personalised automated assessment service (PAAS), our second model, modifies peer assessments to approximate the unknown teacher assessments. It basically predicts how the teacher will assess an assignment, given how the fellow peers have assessed it (as opposed to calculating what the students think of this assignment). These two models lay the foundation for our future work, where we intend to combine both models into a single one that takes into consideration judgements when weighing student assessments, as well as tune assessments to the point of view of the teacher. The aim is to build an automated assessment system that results from the collaboration of both students and teachers.

## 8 Acknowledgements

The second author is supported by an Industrial PhD scholarship from the Generalitat de Catalunya. Furthermore, this work is supported by the Gencat 2014 SGR 118 project (funded by the Generalitat de Catalunya), and the CollectiveMind and Collectiveware projects (funded by the Spanish Ministry of Economy and Competitiveness, under grant numbers TEC2013-49430-EXP and TIN2015-66863-C2-1-R, respectively).

## References

1. de Alfaro, L., Shavlovsky, M.: Crowdgrader: Crowdsourcing the evaluation of homework assignments. Thech. Report 1308.5273, arXiv.org (2013)
2. Debenham, J., Sierra, C.: Trust and honour in information-based agency. In: Proceedings of the Fiftth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS '06). pp. 1225–1232. ACM (2006)
3. Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., Vee, E.: Comparing and aggregating rankings with ties. In: Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. pp. 47–58. PODS '04, ACM, New York, NY, USA (2004)
4. Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., Vee, E.: Comparing partial rankings. SIAM Journal on Discrete Mathematics 20(3), 628–648 (2006)
5. Hang, C.W., Singh, M.P.: Generalized framework for personalized recommendations in agent networks. Autonomous Agents and Multi-Agent Systems (JAAMAS) 25(3), 475–498 (2012)
6. Hill, T., P., L.: Statistics: Methods and Applications. StatSoft, Inc. (2005)
7. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web (WWW '03). pp. 640–651. ACM (2003)
8. Osman, N., Sierra, C., McNeill, F., Pane, J., Debenham, J.: Trust and matching algorithms for selecting suitable agents. ACM Transactions on Intelligent Systems and Technology 5(1), 16:1–16:39 (Jan 2014)
9. Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., Koller, D.: Tuned models of peer assessment in moocs. In: Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013). International Educational Data Mining Society (2013)
10. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Proceedings of the Sixth International Conference on Computer Vision (ICCV '98). pp. 59–66. IEEE Computer Society (1998)
11. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating “word of mouth”. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95). pp. 210–217. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1995)
12. Sierra, C., Debenham, J.: An information-based model for trust. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05). pp. 497–504. ACM, New York, NY, USA (2005)
13. Walsh, T.: The peerrank method for peer assessment. In: Schaub, T., Friedrich, G., O'Sullivan, B. (eds.) Proceedings of the 21st European Conference on Artificial Intelligence (ECAI '14). Frontiers in Artificial Intelligence and Applications, vol. 263, pp. 909–914. IOS Press (2014)