Collaborative Assessment

Patricia Gutierrez
IIIA-CSIC
Campus de la UAB
Barcelona, Spain
patricia@iiia.csic.es

Nardine Osman
IIIA-CSIC
Campus de la UAB
Barcelona, Spain
nardine@iiia.csic.es

Carles Sierra
IIIA-CSIC
Campus de la UAB
Barcelona, Spain
sierra@iiia.csic.es

ABSTRACT

In this paper we introduce an automated assessment service for online learning support in the context of communities of learners. The goal is to introduce automatic tools to support the task of assessing massive number of students as needed in Massive Open Online Courses (MOOC). The final assessments are a combination of tutor's assessment and peer assessment. We build a trust graph over the referees and use it to compute weights for the assesments aggregations. The model proposed intends to be a support for intelligent online learning applications that encourage student's interactions within communities of learners and benefits from their feedback to build trust measures and provide automatic marks.

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 in the learning process, reflect on their role in group learning, and improve their judgement skills. Also, it may have the positive side effect of reducing the marking load of tutors. This is specially critical when tutors face the challenge of marking large quantities of students as needed in the increasingly popular Massive Open Online Courses (MOOC).

Online learning communities encourage different types of peer-to-peer interactions along the learning process. These interactions permit students to get more feedback, to be more motivated to improve, and to compare their own work with other students accomplishments. Tutors, on the other hand, benefit from these interactions as they get a clearer perception of the student engagement and learning process.

Previous works have proposed different methods of peer assessment as part of the learning process with the added advantage of helping tutors in the sometimes dauting task of marking large quantities of students [7, 3].

The authors of [7] propose methods to estimate peer reliability and correct peer biases. They present results over real world data from 63,000 peer assessments of two Coursera courses. The models proposed are probabilistic and they are compared to the grade estimation algorithm used on Coursera's platform, which does not take into account individual biases and reliabilities. Differently from them, we place more trust in students who grade like the tutor and do not consider student's biases. When a student is biased its trust measure will be very low and his/her opinion will have a moderate impact over the final marks.

[3] proposes the CrowdGrader framework, which defines a crowdsourcing algorithm for peer evaluation. The accuracy degree (i.e. reputation) of each student is measured as the distance between his/her self assesment and the aggregated opinion of the peers weighted by their accuracy degrees. The algorithm thus implements a reputation system for students, where higher accuracy leads to higher influence on the consensus grades. Differently from this work, we give more weight to those peers that have similar opinions to those of the tutor.

In this paper, and differently from previous works, we want to study the *reliability* of student assessments when compared with tutor assessments. Although part of the learning process is that students participate in the definition of the evalution criteria, tutors want to be certain that the scoring of the students' works is fair and as close as possible to his/her expert opinion.

Our inspiration comes from a use case explored in the EU-funded project PRAISE [1]. PRAISE enables online virtual communities of students with shared interests and goals to come together and share their music practice with each other so the process of learning becomes social. It provides tools for giving and receiving feedback, as feedback is considered an essential part of the learning process. Tutors define *lesson plans* as pedagogical workflows of activities, such as uploading recorded songs, automatic performance analysis, peer feedback, or reflexive pedagogy analysis. The goal of any lesson plan is to improve student skills, for instance, the performance speed competence or the interpretation maturity level. Assessments of students' performances have to evaluate the achievement of these skills. Once a lesson plan is defined, PRAISE's interface tools allow students to navigate through the activities, to upload assignments, to practice, to assess each other, and so on. The tools allow tutors to monitor what students have done and to assess them. In this work we concentrate on the development of a service that can be included as part of a lesson plan and helps tutors in the overall task of assessing the students participating in the lesson plan. This assessment is based on aggregating students' assessments, taking into consideration the trust that tutors have on the students' individual capabilities in judging each others work.

To achieve our objective we propose in this paper an automated assessment method (Section 2) based on *tutor assessments*, aggregations of *peer assessments* and on *trust measures* derived from peer interactions. We experimentaly evaluate (Section 3) the accuracy of the method over different topologies of student interactions (i.e. different types of student grouping). The results obtained are based on simulated data, leaving the validation with real data for future work. We then conclude with a discussion of the results (Section 4).

2. COLLABORATIVE ASSESSMENT

In this section we introduce the formal model of the method and the algorithms for collaborative assessment.

2.1 Notation and preliminaries

We say an online course has a tutor τ, a set of peer students $\mathcal{S}$, and a set of assignments $\mathcal{A}$ that need to be marked by the tutor and/or students with respect to a given set of criteria $\mathcal{C}$.

The automated assessment state S is then defined as the tuple:

$$S = \langle R, \mathcal{A}, \mathcal{C}, \mathcal{L} \rangle$$

$R = \{\tau\} \cup \mathcal{S}$ defines the set of possible referees (or markers), where a referee could either be the tutor τ or some student $s \in S$. $\mathcal{A}$ is the set of submitted assignments that need to be marked and $\mathcal{C} = \langle c_1, \dots, c_n \rangle$ is the set of criteria that assignments are marked upon. $\mathcal{L}$ is the set of marks (or assessments) made by referees, such that $\mathcal{L} : R \times A \to [0, \lambda]^n$ (we assume marks to be real numbers between 0 and some maximum value λ). In other words, we define a single assessment as: $\mu^\rho_\alpha = \vec{M}$, where $\alpha \in \mathcal{A}$, $\rho \in R$, and $\vec{M} = \langle m_1, \dots, m_n \rangle$ describes the marks provided by the referee on the n criteria of $\mathcal{C}$, $m_i \in [0, \lambda]$.

Similarity between marks. We define a similarity function $sim : [0, \lambda]^n \times [0, \lambda]^n \to [0, 1]$ to determine how close two assesments μ^ρ_α and μ^η_α are. We calculate the similarity between assessments $\mu^\rho_\alpha = \{m_1, \dots, m_n\}$ and $\mu^\eta_\alpha = \{m'_1, \dots, m'_n\}$ as follows:

$$sim(\mu^\rho_\alpha, \mu^\eta_\alpha) = 1 - \frac{\displaystyle\sum_{i=1}^{n} |m_i - m'_i|}{\displaystyle\sum_{i=1}^{n} \lambda}$$

This measure satisfies the basic properties of a fuzzy similarity [6]. Other similarity measures could be used.

Trust relations between referees. Tutors need to decide up to which point they can believe on the assessments made by peers. We use two different intuitions to make up this belief. First, if the tutor and the student have both assessed some assigments, their similarity gives a hint of how close the judegements of the student and the tutor are. Similarly, we can define the judgement closeness of any two students by looking into the assignments evaluated by both of them. In case there are no assigments evaluated by the tutor and one particular student we could simply not take that student's opinion into account because the tutor would not know how much to trust the judgement of this student, or, as we do in this paper, we approximate that unknown trust by lookig into the chain of trust between the tutor and the student through other students. To model this we define two different types of trust relations:

- *Direct trust*: This is the trust between referees $\rho, \eta \in R$ that have at least one assignement assessed in common. The trust value is the average of similarities on the assessments over the same peers. Let the set $A_{\rho,\eta}$ be the set of all assignments that have been assessed by both referees. That is, $A_{\rho,\eta} = \{\alpha \mid \mu^\rho_\alpha \in \mathcal{L}$ and $\mu^\eta_\alpha \in \mathcal{L}\}$. Then,

  $$T_D(\rho, \eta) = \frac{\sum_{\alpha \in A_{\rho,\eta}} sim(\mu^\rho_\alpha, \mu^\eta_\alpha)}{|A_{\rho,\eta}|}$$

  We could also define direct trust as the conjunction of the similarities for all common assignments as:

  $$T_D(\rho, \eta) = \bigwedge_{\alpha \in A_{\rho,\eta}} sim(\mu^\rho_\alpha, \mu^\eta_\alpha)$$

  However, this would not be practical, as a significant difference in just one assessment of those assessed by two referees would make their mutual trust very low.

- *Indirect trust*: This is the trust between referees $\rho, \eta \in R$ without any assignement assessed by both of them. We compute this trust as a transitive measure over chains of referees for which we have pair-wise direct trust values. We define a trust chain as a sequence of referees $q_j = \langle \rho_i, ..., \rho_i, \rho_{i+1}, \dots, \rho_{m_j} \rangle$ where $\rho_i \in R$, $\rho_1 = \rho$ and $\rho_{m_j} = \eta$ and $T_D(\rho_i, \rho_{i+1})$ is defined for all pairs (ρ_i, ρ_{i+1}) with $i \in [1, m_j - 1]$. We note by $Q(\rho, \eta)$ the set of all trust chains between ρ and η. Thus, indirect trust is defined as a aggregation of the direct trust values over these chains as follows:

  $$T_I(\rho, \eta) = \max_{q_j \in Q(\rho,\eta)} \prod_{i \in [1, m_j - 1]} T_D(\rho_i, \rho_{i+1})$$

  Hence, indirect trust is based in the notion of transitivity.[1]

[1] T_I is based on a fuzzy-based similarity relation sim presented before and fulfilling the $\otimes$-Transitivity property: $sim(u, v) \otimes sim(v, w) \leq sim(u, w)$, $\forall u, v, w \in V$, where $\otimes$ is a t-norm [6].

Ideally, we would like to not overrate the trust of a tutor on a student, that is, we would like that $T_D(a,b) \geq T_I(a,b)$ in all cases. Guaranteeing this in all cases is impossible, but we can decrease the number of overtrusted students by selecting an operator that gives low values to T_I. In particular, we prefer to use the product $\prod$ operator, because this is the t-norm that gives the smallest possible values. Other opertors could be used, for instance the *min* function.

Trust Graph. To provide automated assessments, our proposed method agregates the assessments on a given assignment taking into consideration how much trusted is each marker/referee from the point of view of the tutor (i.e. taking into consideration the trust of the tutor on the referee in marking assignments). The algorithm that computes the student final assessment is based on a graph defined as follows:

$$G = \langle R, E, w \rangle$$

where the set of nodes R is the set of referees in S, $E \subseteq R \times R$ are edges between referees with direct or indirect trust relations, and $w : E \to [0,1]$ provides the trust value. We note by $D \subset E$ the set of edges that link referees with direct trust. That is, $D = \{e \in E | T_D(e) \neq \bot\}$. An similarly, $I \subset E$ for indirect trust, $I = \{e \in E | T_I(e) \neq \bot\} \setminus D$. The w values will be used as weights to combine peer assessments and are defined as:

$$w(e) = \begin{cases} T_D(e) & \text{, if } e \in D \\ T_I(e) & \text{, if } e \in I \end{cases}$$

Figure 1 shows examples of trust graphs with $e \in D$ (in black) and $e \in I$ (in red —light gray) for different sets of assessments $\mathcal{L}$.

2.2 Computing collaborative assessments

Algorithm 1 implements the collaborative assessment method. We keep the notation (ρ, η) to refer to the edge connecting nodes ρ and η in the trust graph and $Q(\rho, \eta)$ to refer the set of trust chains between ρ and η.

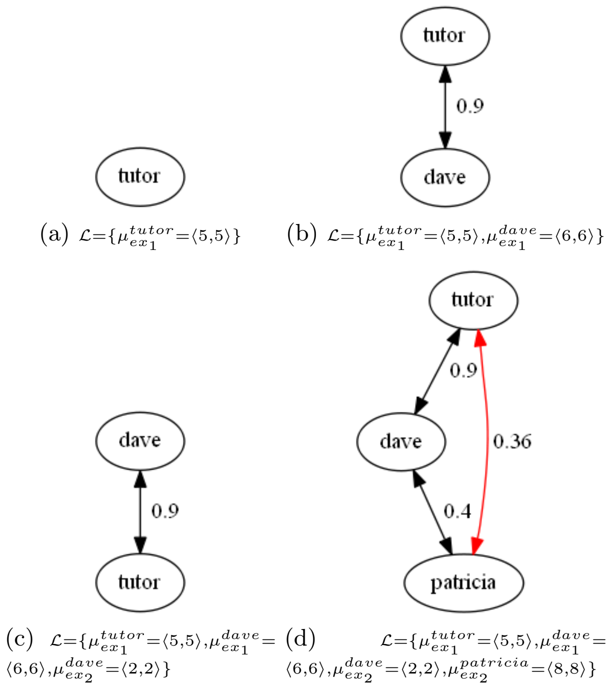
The first thing the algorithm does is to build a trust graph from $\mathcal{L}$. Then, the final assessments are computed as follows. If the tutor marks an assignment, then the tutor mark is considered the final mark. Otherwise, a weighted average (μ_α) of the marks of student peers is calculated for this assignment, where the weight of each peer is the trust value between the tutor and that peer. Other forms of aggregation could be considered to calculate μ_α, for instance a peer assessment may be discarded if it is very far from the rest of assessments, or if the referee's trust falls below a certain threshold.

Algorithm 1: collaborativeAssessments($S = \langle R, \mathcal{A}, \mathcal{C}, \mathcal{L} \rangle$)

```
                                ▷ Initial trust between referees is zero
$D = I = \emptyset$;
for $\rho, \eta \in \mathcal{R}, \rho \neq \eta$ do
    $w(\rho, \eta) = 0$;
end
                                ▷ Update direct trust and edges
for $\rho, \eta \in \mathcal{R}, \rho \neq \eta$ do
    $A_{\rho,\eta} = \{\beta \mid \mu_\beta^\rho \in \mathcal{L}$ and $\mu_\beta^\eta \in \mathcal{L}\}$;
    if $|A_{\rho,\eta}| > \mathit{0}$ then
        $D = D \cup (\rho, \eta)$;
        $w(\rho, \eta) = T_D(\rho, \eta)$;
    end
end
        ▷ Update indirect trust and edges between tutor & students
for $\rho \in \mathcal{R}$ do
    if $(\tau, \rho) \notin D$ and $Q(\tau, \rho) \neq \emptyset$ then
        $I = I \cup (\rho, \eta)$;
        $w(\rho, \eta) = T_I(\tau, \eta)$;
    end
end
                                ▷ Calculate automated assessments
$assessments = \{\}$;
for $\alpha \in A$ do
    if $\mu_\alpha^\tau \in \mathcal{L}$ then
                                ▷ Tutor assessments are preserved
        $assessments = assessments \cup (\alpha, \mu_\alpha^\tau)$
    else
                                ▷ Generate automated assessments
        $R' = \{\rho | \mu_\alpha^\rho \in \mathcal{L}\}$;
        if $|R'| > 0$ then
            $\mu_\alpha = \frac{\sum_{\rho \in R'} \mu_\alpha^\rho * w(\tau, \rho)}{\sum_{\rho \in R'} w(\tau, \rho)}$;
            $assessments = assessments \cup (\alpha, \mu_\alpha)$;
        end
    end
end
return $assessments$;
```

Figure 1 shows four trust graphs built from four assessments histories that corresponds to a chronological sequence of assessments made. The criteria $\mathcal{C}$ in this example are *speed* and *maturity* and the maximum mark value is $\lambda = 10$. For simplicity we only represent those referees that have made assessments in $\mathcal{L}$. In Figure 1(a) there is one node representing the tutor who has made the only assessment over the assignment ex_1 and there are no links to other nodes as no one else has assessed anything. In (b) student Dave assesses the same exercise as the tutor and thus a link is created between them. The trust value $w(tutor, Dave) = T_D(tutor, Dave)$ is high since their marks were similar. In (c) a new assessment by Dave is added to $\mathcal{L}$ with no consequences in the graph construction. In (d) student Patricia adds an assessment on ex_2 that allows to build a direct trust between Dave and Patricia and an indirect trust between the tutor and Patricia, through Dave. The automated assessments generated in case (d) are: $\langle 5, 5 \rangle$ for exercise 1 (which preserves the tutor's assessment) and $\langle 3.7, 3.7 \rangle$ for exercise 2 (which uses a weighted aggregation of the peers' assessments).

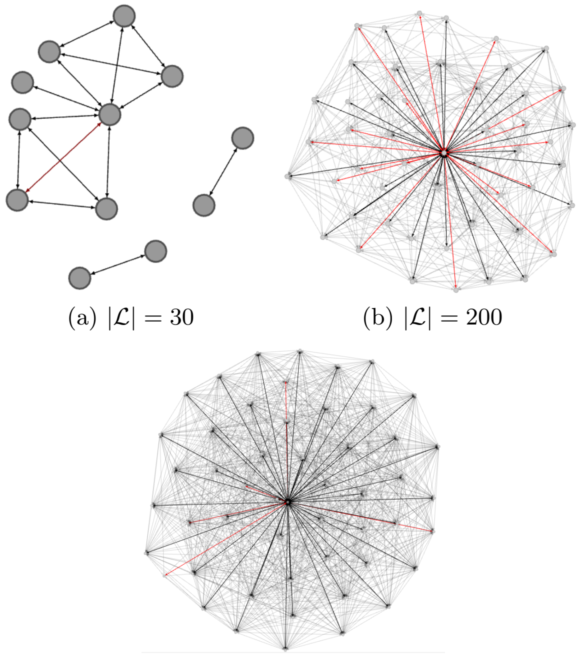
Note that the trust graph built from $\mathcal{L}$ is not necessarily connected. A tutor wants to reach a point in which the graph is totally connected because that means that the collaborative assessment algorithm generates an assessment for every assignment. Figure 2 shows an example of a trust graph of a particular learning community involving 50 peer students and a tutor. When S has a history of 5 tutor assessments and 25 student assessments ($|\mathcal{L}| = 30$) we observe that not all nodes are connected. As the number of assessments in-



(a) $\mathcal{L}$=$\{\mu_{ex_1}^{tutor}$=$\langle 5{,}5\rangle\}$ (b) $\mathcal{L}$=$\{\mu_{ex_1}^{tutor}$=$\langle 5{,}5\rangle$,$\mu_{ex_1}^{dave}$=$\langle 6{,}6\rangle\}$

(c) $\mathcal{L}$=$\{\mu_{ex_1}^{tutor}$=$\langle 5{,}5\rangle$,$\mu_{ex_1}^{dave}$=$\langle 6{,}6\rangle$,$\mu_{ex_2}^{dave}$=$\langle 2{,}2\rangle\}$ (d) $\mathcal{L}$=$\{\mu_{ex_1}^{tutor}$=$\langle 5{,}5\rangle$,$\mu_{ex_1}^{dave}$=$\langle 6{,}6\rangle$,$\mu_{ex_2}^{dave}$=$\langle 2{,}2\rangle$,$\mu_{ex_2}^{patricia}$=$\langle 8{,}8\rangle\}$

Figure 1: Trust graph example 1.



(a) $|\mathcal{L}| = 30$ (b) $|\mathcal{L}| = 200$

(c) $|\mathcal{L}| = 400$

Figure 2: Trust graph example 2

creases, the trust graph becomes denser and eventually it gets completely connected. In (b) and (c) we see a complete graph.

3. EXPERIMENTAL PLATFORM AND EVALUATION

In this Section we describe how we generate simulated social networks, describe our experimental platform, define our benchmarks and discuss experimental results.

3.1 Social Network Generation

Several models for social network generation have been proposed reflecting different characteristics present in real social communities. Topological and structural features of such networks have been explored in order to understand wich generating model resembles best the structure of real communities [5].

A social network can be defined as a graph $\mathcal{N}$ where the set of nodes represent the individuals of the network and the set of edges represent connections or social ties among those individuals. In our case, individuals are the members of the learning community: the tutor and students. Connections represent the social ties and they are usually the result of interactions in the learning community. For instance a social relation will be born between two students if they interact with each other, say by collaboratively working on a project together. In our experimentation, we rely on the social network in order to simulate which student will assess the assignment of which other student. We assume students will assess the assignments of students they know, as opposed to picking random assignments. As such, we clarify that social networks are different from the trust graph of Section 2. While the nodes of both graphs are the same, edges of the social network represent social ties, whereas edges in the trust graph represent how much does one referee trust another in judging others work.

To model social networks where relations represent social ties, we follow three different approaches: the Erdős-Rényi model for random networks [4], the Barabási-Albert model for power law networks[2] and a hierarchical model for cluster networks.

3.1.1 Random Networks

The Erdős-Rényi model for random networks consists of a graph containing n nodes connected randomly. Each possible edge between two vertices may be included in the graph with probability p and may not be included with probability $(1 - p)$. In addition, in our case there is always an edge between the node representing the tutor and the rest of nodes, as the tutor knows all of its students (and may eventually mark any of those students).

The degree distribution of random graphs follows a Poisson distribution. Figure 3(a) shows an example of a random graph with 51 nodes and $p = 0.5$ and its degree distribution. Note that the point with degree 50 represents the tutor node while the rest of the nodes degree fit a Poisson distribution.

3.1.2 Power Law Networks

The Barabási-Albert model for power law networks base their graph generation on the notions of *growth* and *preferential attachment*. The generation scheme is as follows. Nodes are added one at a time. Starting with a small number of initial nodes, at each time step we add a new node with m edges linked to nodes already part of the network. In our experiments, we start with $m + 1$ initial nodes. The edges are not placed uniformly at random but preferentially in proportion to the degree of the network nodes. The probability p that the new node is connected to a node i already in the network depends on the degree k_i of node i, such that: $p = k_i / \sum_{j=1}^{n} k_j$. As above, there is also always an edge between the node representing the tutor and the rest of nodes.

The degree distribution of this network follows a Power Law distribution. Figure 3(b) shows an example of a power law graph with 51 nodes and $m = 16$ and its degree distribution. The point with degree 50 describes the tutor node while the rest of the nodes closely resemble a power law distribution. Recent empirical results on large real-world networks often show, among other features, their degree distribution following a power law [5].

3.1.3 Cluster Networks

As our focus is on learning communities, we also experiment with a third type of social network: the cluster network which is based on the notions of *groups* and *hierarchy.* Such networks consists of a graph composed of a number of fully connected clusters (where we believe clusters may represent classrooms or similar pedagogical entities). Additionally, as above, all the nodes are connected with the tutor node. Figure 3(c) shows an example of a cluster graph with 51 nodes, 5 clusters of 10 nodes each and its degree distribution. The point with degree 50 describes the tutor while the rest of the nodes have degree 10, since every student is fully connected with the rest of the classroom.

3.2 Experimental Platform

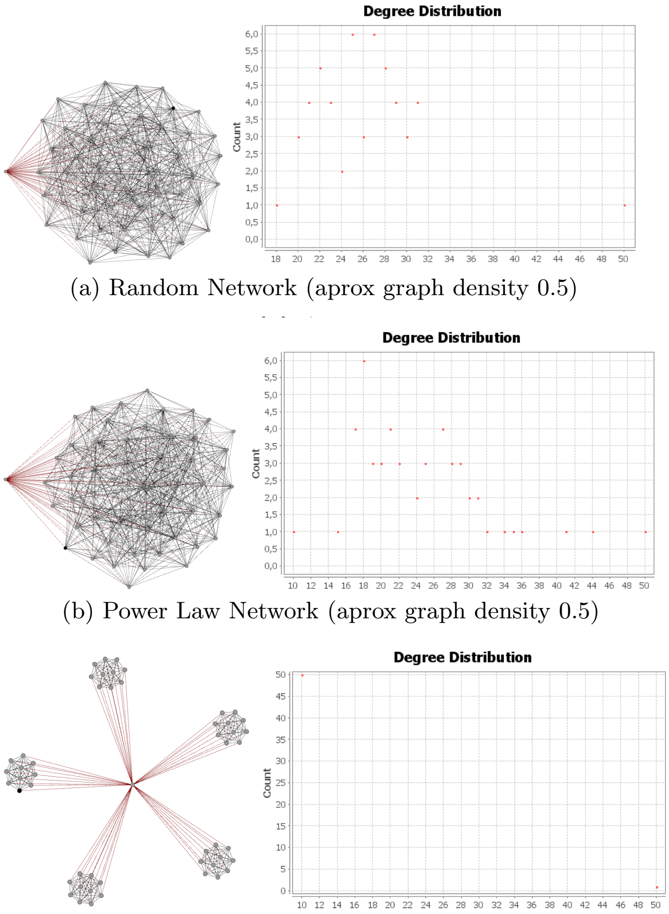
In our experimentation, given an initial automated assessment state $S = \langle R, \mathcal{A}, \mathcal{C}, \mathcal{L} \rangle$ with an empty set of assessments $\mathcal{L} = \{\}$, we want to simulate tutor and peer assessments so that the collaborative assessment method can eventually generate a reliable and definitive set of assessments for all assignments.

To simulate assessments, we say each students is defined by its profile that describes how good its assessments are. The profile is essentially defined by the measure, or distance, $d_\rho \in [0, 1]$ that specifies how close are the student's assessments to that of the tutor.

We then assume the simulator knows how the tutor and each student would assess an assignment. This becomes necessary in our simulation, since we generate student assessments in terms of their distance to that of the tutor's, even if the tutor does not choose to actually assess the assignment in question. This simulator's knowledge of the values of all possible assessments is generated accordingly:

- For every assignment $\alpha \in \mathcal{A}$, we calculate the tutor's assessment, which is randomly generated according to the function $f_\tau : \mathcal{A} \to [0, \lambda]^n$. This assessment essentially describes what mark would the tutor give α, if it decided to assess it.

- For every assignment $\alpha \in \mathcal{A}$, we also calculate the assessment of each student $\rho \in \mathcal{S}$. This is calculated according to the function $f_\rho : A \to [0, \lambda]^n$, such that: $sim(f_\rho(\alpha), f_\tau(\alpha)) \geq d_\rho$ We note that we only need to calculate ρ's assessment of α if the student who submitted the assignment α is a neighbour of ρ in $\mathcal{N}$.



(a) Random Network (aprox graph density 0.5)

(b) Power Law Network (aprox graph density 0.5)

(c) Cluster Network (aprox graph density 0.2)

Figure 3: Social Network generation examples

We note that the above only calculates what the assessments would be, if referees where to assess assignments.

3.3 Benchmark

Given an initial automated assessment state $S = \langle R, \mathcal{A}, \mathcal{C}, \mathcal{L} \rangle$ with an empty set of assessments $\mathcal{L} = \{\}$, a set of student profiles $Pr = \{d_s\}_{\forall s \in \mathcal{S}}$, and a social network $\mathcal{N}$ (whose nodes is the set R), we simulate individual tutor and students' assessments. When does a referee in R assess an assignment in $\mathcal{A}$ is explained shortly. However we note here that the value of each generated assessment is equivalent to that calculated for the simulator's knowledge (see Section 3.2 above).

In our benchmark, we consider the three types of social networks introduced earlier: random social networks (with 51 nodes, $p = 0.5$, and approximate density of 0.5), power law networks (with 51 nodes, $m = 16$, and approximate density of 0.5), and cluster networks (with 51 nodes, 5 clusters of 10 nodes each, and approximate density of 0.2). Examples of these generated networks are shown in Figure 3.

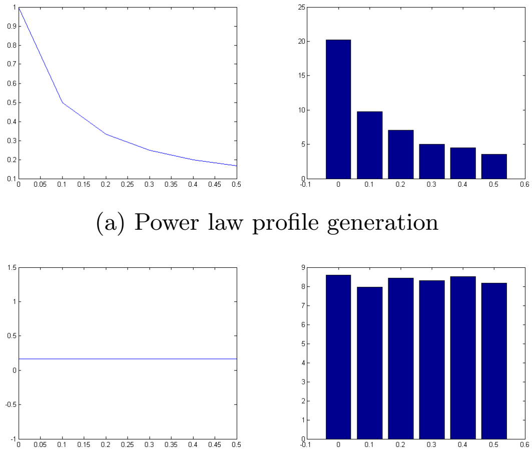
We say one assignment is submitted by each student, resulting in $|\mathcal{S}| = 50$ and $|\mathcal{A}| = 50$. The range that a referee (tutor or student) may mark a given assignment with respect to a given criteria is [0,10]. And the set of criteria is $\mathcal{C} = \langle speed, maturity\rangle$. The criteria essentially measure the *speed* of playing a musical piece, and the *maturity level* of the student's performance.

An assessment profile is generated for each student ρ at the beginning of the execution, resulting in a set of student profiles $Pr = \{d_s\}_{\forall s \in \mathcal{S}}$, where $d \in [0, 0.5]$. We consider here two cases for generating the set of student profiles Pr. A first case where d is picked randomly following a power law distribution (Figure 4(a)) and a second case where d is picked randomly following a uniform distribution (Figure 4(b)).

With simulated individual assessments, we then run the collaborative assessment method in order to compute an automated assessment. We also compute the 'error' of the collaborative assessment method, whose range is $[0, 1]$, over the set of assignments $\mathcal{A}$ accordingly:

$$\frac{\sum\limits_{\alpha \in \mathcal{A}} sim(f_\tau(\alpha), \phi(\alpha))}{|\mathcal{A}|}$$

, where $\phi(\alpha)$ describes the automated assessment for a given assignment $\alpha \in \mathcal{A}$



(a) Power law profile generation

(b) Uniform profile generation

Figure 4: Example of the profile distributions (left) and of d counting averaged over 50 instances (right)

With the settings presented above, we run two different experiments. The results presented are an average over 50 executions. The two experiments are presented next.

In experiment 1, students provide their assessments before the tutor. Each student ρ provides assessments for a randomly chosen a_ρ number of peer assigments (of course, where assignments are those of their neighboring peers in $\mathcal{N}$). We run the experiment for 5 different values of $a_\rho = \{3, 4, 5, 6, 7\}$. After the students provide their assessments, the tutor starts assessing assignments incrementally. After every tutor assessment, the error over the set of automated assessment is calculated. Notice that the collaborative assessment method takes the tutor assessment, when it exists, to be the final assessment. As such, the number of automated assessments calculated based on aggregating students' assessments is reduced over time. Finally, when the tutor has assessed all 50 students, the resulting error is 0.

In experiment 2, the tutor provides its assessments before the students. The tutor in this experiment will assess a randomly chosen number of assignments, where this number is based on the percentage a_τ of the total number of assignments. We run the experiment for 4 different values of $a_\tau = \{5, 10, 15, 20\}$. After the tutor provides their assessments, students' assessments are performed. In every iteration, a student ρ randomly selects a neighbor in $\mathcal{N}$ and assesses his assignment (in case it has not been assessed before by ρ, otherwise another connected peer is chosen). We note that in the case of random and power law networks (denser networks), a total number of 1000 student assessments are performed. Whereas in the case of cluster networks (looser network), a total of 400 student assessments are performed. We note that initially, the trust graph is not fully connected, so the service is not able to provide automated assessments for all assignments. When the grap gets fully connected, the service generates automated assessments for all assignments and we start measuring the error after every new iteration.

3.4 Evaluation

In experiment 1, we observe (Figure 5) that the error decreases when the number of tutor assessments increase, as expected, until it reaches 0 when the tutor has assessed all 50 students. This decrement is quite stable and we do not observe abrupt error variations or important error increments from one iteration to the next. More variations are observed in the initial iterations since the service has only a few assessments to deduce the weights of the trust graph and to calculate the final outcome.

In the case of experiment 2 (Figure 6), the error diminishes slowly as the number of student assessments increase, although it never reaches 0. Since the number of tutor assessments is fixed in this experiment, we have an error threshold (a lower bound) which is linked to the students' assessment profile: the closest to the tutor's the lower this threshold will be. In fact, in both experiments we observe that when using a power law distribution profile (Figure 4(a)) the automated assessment error is lower than when using a uniform distribution profile (Figure 4(b)). This is because when using a power law distribution, more student profiles are generated whose assessments are closer to the tutors'.

In general, the error trend observed in all experiments comparing different social network scenarios (random, cluster or power law) show a similar behavior. Taking a closer look at experiment 2, cluster social graphs have the lowest error and we observe that assessments on all assignments are achieved earlier (this is, the trust graph gets connected earlier). We attribute this to the topology of the fully connected clusters which favors the generations of indirect edges earlier in the graph between the tutor and the nodes of each cluster. Power law social graphs have lower error than random networks in most cases. This can be attributed to the criteria of preferential attachment in their network generation,

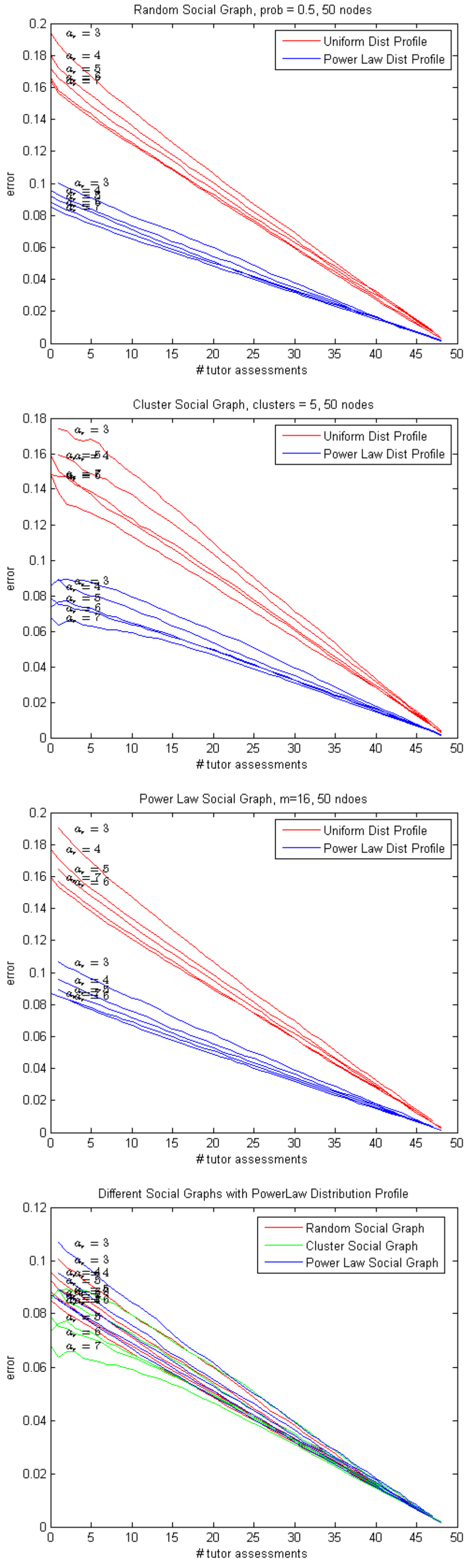


Figure 5: Eperiment 1

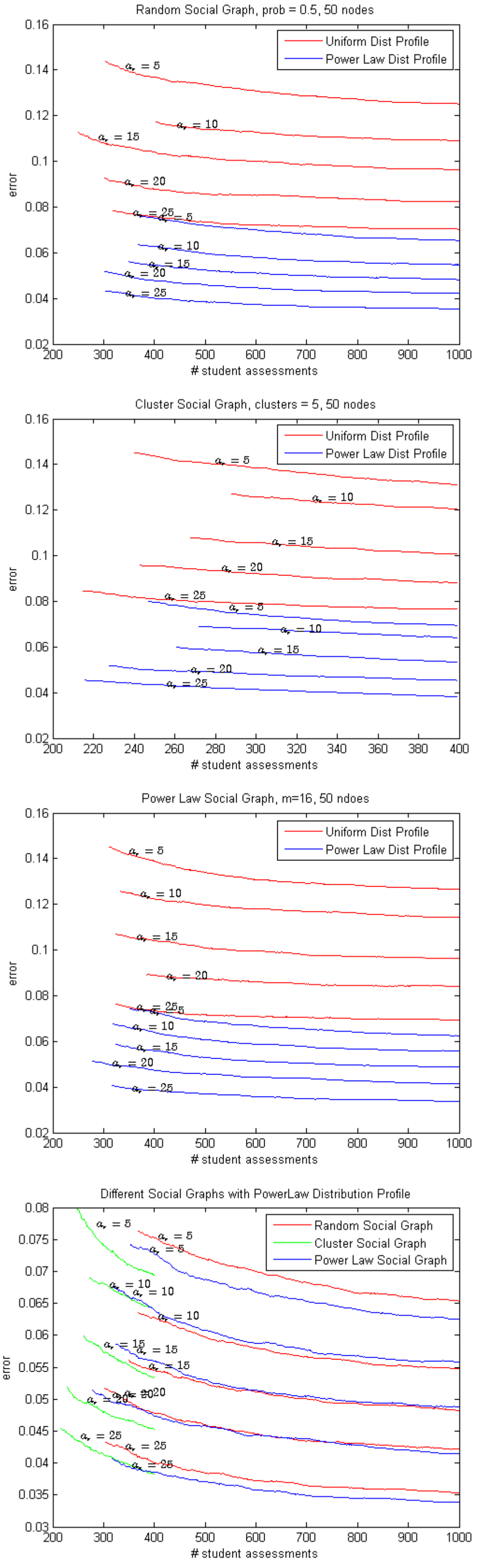


Figure 6: Experiment 2

which favors the creation of some highly connected nodes. Such nodes are likely to be assessed more frequently since more peers are connected to them. Then, the automated assessments of these higly connected peers are performed with more available information which could lead to more accurate outcomes.

4. DISCUSSION

The collaborative assessment model proposed in this paper is thought of as a support in the creation of intelligent online learning applications that encourage student interactions within communities of learners. It goes beyond current tutor-student online learning tools by making students participate in the learning process of the whole group, providing mutual assessment and making the overall learning process much more collaborative.

The use of AI techniques is key for the future of online learning communities. The application presented in this paper is specially useful in the context of MOOC: with a low number of tutor assessments and encouraging students to interact and provide assessments among each other, direct and indirect trust measures can be calculated among peers and automated assessments can be generated.

Several error indicators can be designed and displayed to the tutor managing the course, which we leave for future work. For example the error indicators may inform the tutor which assignments have not received any assessments yet, or which deduced marks are considered unreliable. For example, a deduced mark on a given assignment may be considered unreliable if all the peer assessments that have been provided for that assignment are considered not to be trusted by the tutor as they fall below a preselected acceptable trust threshold. Alternatively, a reliability measure may also be assigned to the computed trust measure T_D. For instance, if there is only one assignment that has been assessed by τ and ρ, then the computed $T_D(\tau, \rho)$ will not be as reliable as having a number of assignments assessed by τ and ρ. As such, some reliability threshold may be used that defines what is the minimum number of assignments that both τ and ρ need to assess for $T_D(\tau, \rho)$ to be considered reliable. Observing such error indicators, the tutor can decide to assess more assignments and as a result the error may improve or the set of deduced assessments may increase. Finally, if the error reaches a level of acceptance, the tutor can decide to endorse and publish the marks generated by the collaborative assessment method.

Another interesting question for future work is presented next. Missing connections might be detected in the trust graph that would improve its connectivity or maximize the number of direct edges. The question that follows then is, what assignments should be suggested to which peers such that the trust graph and the overall assessment outcome would improve?

Additionally, future work may also study different approaches for calculating the indirect trust value between two referees. In this paper, we use the product operator. We suggest to study a number of operators, and run an experiment to test which is most suitable. To do such a test, we may calculate the indirect trust values for edges that do have a direct trust measure, and then see which approach for calculating indirect trust gets closest to the direct trust measures.

Acknowledgements

This work is supported by the Agreement Technologies project (CONSOLIDER CSD2007-0022, INGENIO 2010) and the PRAISE project (EU FP7 grant number 388770).

5. REFERENCES

[1] Praise project: http://www.iiia.csic.es/praise/.

[2] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 1999.

[3] L. de Alfaro and M. Shavlovsky. Thecnical report 1308.5273, arxiv.org. *Crowdgrader: Crowdsourcing the evaluation of homework assignments*, 2013.

[4] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 1959.

[5] E. Ferrara and G. Fiumara. Topological features of online social networks. *Communications in Applied and Industrial Mathematics*, 2011.

[6] L. Godo and R. Rodríguez. Logical approaches to fuzzy similarity-based reasoning: an overview. *Preferences and Similarities*, 2008.

[7] 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.