Collaborative Assessment

Patricia Gutierrez a,1, Nardine Osman a,2 and Carles Sierra a,3

a IIIA-CSIC, Spain

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 assessments aggregations.

Keywords. Automated Assessment, Trust Model, Online Learning

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. 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. The authors of [6] 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. 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. The authors of [2] 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 assessment and the aggregated opinion of the peers weighted by their accuracy degrees.

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

1Corresponding Author: E-mail: patricia@iiia.csic.es.
2Corresponding Author: E-mail: nardine@iiia.csic.es.
3Corresponding Author: E-mail: sierra@iiia.csic.es.
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 experimentally 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 and algorithm for collaborative assessment.

2.1. Notation and preliminaries

We say an online course has a tutor \( \tau \), a set of peer students \( S \), and a set of assignments \( A \) that need to be marked by the tutor and/or students with respect to a given set of criteria \( C \). The automated assessment state \( X \) is then defined as the tuple:

\[
X = (R, A, C, L)
\]

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

Similarity between marks We can define a similarity function \( sim: [0, \lambda]^n \times [0, \lambda]^n \rightarrow [0, 1] \) to determine how close two assessments \( \mu_\alpha^\rho \) and \( \mu_\eta^\rho \) are. We calculate the similarity between assessments \( \mu_\alpha^\rho = \{m_1, \ldots, m_n\} \) and \( \mu_\eta^\rho = \{m_1', \ldots, m_n'\} \) as follows:

\[
sim(\mu_\alpha^\rho, \mu_\eta^\rho) = 1 - \frac{\sum_{i=1}^{n} |m_i - m_i'|}{\sum_{i=1}^{n} \lambda}.\]

This measure satisfies the basic properties of a fuzzy similarity [5]. We note that other similarity measures can 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 assignments, their similarity gives a hint of how close the judgements of the student and the tutor are. Similarly, we can define the judgement closeness of any two students by looking into the assign-
ments evaluated by both of them. In case there are no assignments 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 looking 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 assignment assessed in common. The trust value is the average of similarities on their assessments over the same assignments. Let the set $A_{\rho, \eta}$ be the set of all assignments that have been assessed by both referees. That is, $A_{\rho, \eta} = \{ \alpha \mid \mu^\rho_\alpha \in L \text{ and } \mu^\eta_\alpha \in L \}$. Then,

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

We could also define direct trust as the conjunction of the similarities for all common assignments as: $T_D(\rho, \eta) = \bigwedge_{\alpha \in A_{\rho, \eta}} \text{sim}(\mu^\rho_\alpha, \mu^\eta_\alpha)$. In this case $T_D$ would be a similarity measure satisfying $\otimes$-transitivity for any T-norm. However, this would not be practical, as a significant difference in just one assessment of two referees would make their mutual trust very low.

- **Indirect trust**: This is the trust between referees $\rho, \eta \in R$ without any assignment 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 $c_j = (\rho_1, \ldots, \rho_i, \ldots, \rho_{m_j})$ 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 $(\rho_i, \rho_{i+1})$ with $i \in [1, m_j - 1]$. We note by $C(\rho, \eta)$ the set of all trust chains between $\rho$ and $\eta$. Thus, indirect trust is defined as a aggregation of the direct trust values over these chains as follows:

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

Hence, indirect trust is based in the notion of transitivity.$^4$

Ideally, we would like to not over rate the trust of a tutor on a student, that is, we would like that $T_D(\alpha, b) \geq T_I(\alpha, b)$ 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 operators could be used, for instance the $\min$ function.

---

$^4$ $T_I$ is based on a fuzzy-based similarity relation $\text{sim}$ presented before and fulfilling the $\otimes$-Transitivity property: $\text{sim}(u, v) \otimes \text{sim}(v, w) \leq \text{sim}(u, w)$, $\forall u, v, w \in V$, where $\otimes$ is a t-norm [5].
Trust Graph  We use a graph data structure to store and represent the trust relations between referees. The trust graph is defined as: $G = (R, E, w)$, where the set of nodes $R$ is the set of referees, $E \subseteq R \times R$ 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, $D = \{e \in E | T_D(e) \neq \perp\}$. An similarly, $I \subset E$ for indirect trust, $I = \{e \in E | T_I(e) \neq \perp\} \setminus D$. The $w$ values 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 $L$.

2.2. Computing collaborative assessments

Algorithm 1 implements the collaborative assessment method. We keep the notation $(\rho, \eta)$ to refer to the edge connecting nodes $\rho$ and $\eta$ in the trust graph and $\mathcal{C}(\rho, \eta)$ to refer the set of trust chains between $\rho$ and $\eta$. First, the trust graph is built 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 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.

Figure 1 shows the evolution of a trust graph built from 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 $ex_1$. In (b) student Dave assesses the same exercise as the tutor and thus a link is created between them. The direct trust value
Algorithm 1: collaborativeAssessments(S = (R, A, C, L))

Initial trust between referees is zero
\[ D = I = ∅; \]

for \(\rho, \eta \in R, \rho \neq \eta\) do
\[ w(\rho, \eta) = 0; \]
end

Update direct trust and edges
for \(\rho, \eta \in R, \rho \neq \eta\) do
\[ A_{\rho, \eta} = \{ \beta \mid \mu_{\rho}^\beta \in L \text{ and } \mu_{\eta}^\beta \in L \}; \]
if \(|A_{\rho, \eta}| > 0\) then
\[ D = D \cup (\rho, \eta); \]
\[ w(\rho, \eta) = T_D(\rho, \eta); \]
end
end

Update indirect trust and edges between tutor & students
for \(\rho \in R\) do
if \((\tau, \rho) \notin D \text{ and } C(\tau, \rho) \neq ∅\) 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 \(\nu_{\alpha}^\tau \in L\) then
\[ assessments = assessments \cup (\alpha, \mu_{\alpha}^\tau); \]
else
\[ R' = (\rho | \nu_{\alpha}^\rho \in L); \]
if \(|R'| > 0\) then
\[ \mu_{\alpha} = \frac{\sum_{\rho \in R'} \mu_{\rho}^\alpha \cdot w(\tau, \rho)}{\sum_{\rho \in R'} w(\tau, \rho)}; \]
\[ assessments = assessments \cup (\alpha, \mu_{\alpha}); \]
end
end
end
return \(assessments;\)

\(w(\text{tutor, Dave})\) is high since their marks were similar. In (c) a new assessment by Dave is added to \(L\) with no consequences in the graph construction. In (d) student Patricia adds an assessment on \(e_2\) that allows to build a direct trust between Dave and Patricia and an indirect trust between the tutor and Patricia, through Dave.

Note that the trust graph built from \(L\) is not necessarily connected. A tutor wants to reach a point in which the graph is connected because that means a trust measure between the tutor and every student can be deduced. Figure 2 shows an example of a trust graph of a learning community involving 50 peer students and a tutor. When the history of assessments is small (\(|L| = 30\)) we observe that not all nodes are connected. As the number of assessments increases, the trust graph becomes denser and eventually gets 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 which generating model resembles best the structure of real communities [4].

A social network can be defined as a graph $N$ 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. 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 [3], the Barabási-Albert model for power law networks [1] 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 them). The degree distribution of random graphs follows 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 distribu-
tion of this network follows a Power Law distribution. Recent empirical results on large real-world networks often show their degree distribution following a power law [4].

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.

3.2. Experimental Platform

In our experimentation, given an initial automated assessment state \( X = \langle R, A, C, \mathcal{L} \rangle \) 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 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, even if they do not actually assess it in the simulation. Then:

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

3.3. Benchmark

We consider 50 students and one assignment is submitted by each student, resulting in \( |S| = 50 \) and \( |A| = 50 \). Three types of social networks are used: 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).

Two cases are considered for generating the set of student profiles \( P_r = \{d_\alpha \}_{\alpha \in S} \), where \( d \in [0, 0.5] \). A first case where \( d \) is picked randomly following a power law distribution and a second case where \( d \) is picked randomly following a uniform distribution. The set of criteria is \( C = \langle \text{speed}, \text{maturity} \rangle \) and the maximum mark value is \( \lambda = 10 \).

We also compute the ‘error’ of the collaborative assessment method, whose range is \([0, 1]\), over the set of assignments \( A \) accordingly: \( \sum_{\alpha \in A} \frac{\text{sim}(f_\tau(\alpha), \phi(\alpha))}{|A|} \), where \( \phi(\alpha) \) describes the automated assessment for a given assignment \( \alpha \in A \).

With the settings presented above, we run two different experiments, each with 50 executions. The results presented are an average of the 50 executions.
In experiment 1, students provide their assessments before the tutor. Each student \( \rho \) provides assessments for a randomly chosen \( a_{\rho} \) number of peer assignments. 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 \( a_{\tau} \) number of assignments. We run the experiment for 5 different values of \( a_{\tau} = \{5, 10, 15, 20, 25\} \). After the tutor provides their assessments, students assessments are performed. In every iteration, a student \( \rho \) randomly selects a neighbor in \( N \) and assesses his assignment. 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 graph 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 3) 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. 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 4), 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 the automated assessment error is lower than when using a uniform distribution profile. 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, 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 highly connected peers are performed with more available information which could lead to more accurate outcomes.

4. Conclusion and Future Work

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. Alternatively, a reliability measure may also be assigned to the computed trust measure $T_D$. 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. When 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 in the trust graph might 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 those connections appear and 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.

References