Using Collective Trust for Group Formation

Thomas Largillier and Julita Vassileva

MADMUC Lab
University of Saskatchewan
Saskatoon, Canada
firstname.lastname@usask.ca

Abstract. Group formation is a difficult task that arises in many different contexts. It is either done manually or using methods based on individual users' criteria. Users may not be willing to fill a profile or their profile may evolve with time without users updating it. A collaboration may also fail for personal reasons between users with compatible profiles as it may be a success between antagonist users that may start a productive conflict inside a team. Existing methods do not take into account previous successful or unsuccessful collaborations to forge new ones. The authors introduce a new model of collaborative trust to help select the "best" fitted group for a task. This paper also presents one heuristic to find the best possible group since in practice considering all the possibilities is hardly an option.

1 Introduction

There are many situations where people have to collaborate. It is an important job to make sure that the group gathered to accomplish a given task will perform efficiently. In learning context, it is often required that students perform exercises or projects as groups.

Several trust mechanisms have been developed over the years [1]. Most of those systems concern the trust a user A has in another user/product B. For example, Wang *et al.* in [11] define formally trust as "a peer's belief in another peer's capabilities, honesty and reliability *based on its own direct experiences*". They build a trust and reputation mechanism using Bayesian networks for file providers selection in peer-to-peer systems. Their approach helps users to find "better" peers in the system as well as even the load between file providers.

Gummadi *et al.* in [4] introduce a group to group trust value in peer-to-peer networks. However, their method forces all groups to be disjoint and the group-to-group trust between groups A and B is simply the average trust members of groups A have in members of group B. Therefore, this notion is simply an aggregation of trust collected in pairwise interactions.

Many virtual interactions nowadays are not between only two people, so there is a need to redefine trust metrics, since most of the existing ones always characterize the trust some user a has in one user/product b. Simply aggregating the pairwise trust will not help the user know which groups of users/products

she can trust. When interactions are group based, it is not enough to know that a peer is trustworthy, the user needs to know who he is trustworthy with and more importantly who he is not trustworthy with. A pairwise trust metric does not carry enough information. This paper introduces a collective trust mechanism together with an algorithm to compute the estimated trust of any group a user had no previous interaction with. This mechanism is then used to solve the group formation problem.

2 Group Formation

Several people addressed the problem of defining groups to perform a task. Most of the relevant work has been done in the field of collaborative learning and how to optimize the group formation phase so students will learn faster and better.

All the following approaches use individual characteristics of users to gather them into efficient groups.

Oakley *et al.* present in [10] their system to group students. Their team formation method aims at grouping together people with diverse ability levels with common blocks of time to meet outside classes. The groups are assembled by the teacher based on forms filled by the student. Each team member is also assigned a designed role inside the team. The roles change over time so that each student can see several aspects of team work. Since in this approach groups last at least a semester, the authors provide several guidelines on how to deal with problems like free riders in a group. The authors' scheme authorizes groups to be reshaped if a group wants to fire one of its members or if a group member wishes to leave her coworkers.

Martin *et al.* propose in [8] to use the Felder-Silverman [3] classification to adapt learning material to students as well as to group students in e-learning. The authors' idea is to gather both active and reflexive students inside groups to make the groups more efficient. Their idea, as well as the latent jigsaw method were used in class and described by Deibel in [2]. The feedback from the students was really positive as they say the groups help them to learn more efficiently and confronted them with new ideas.

Wessner *et al.* present in [13] a tool to group e-learning students. They introduce the Intended Points of Cooperation and how they can be used to form appropriate groups for a task. The grouping is done by hand by the teacher or can be done automatically to regroup people that have reached the same learning stage.

Inaba *et al.* in [5] propose to identify and describe users' personal objective using ontologies and to group people having similar objectives for collaboration to be more efficient. The collaborative learning ontology is developped further in [6] to provide a framework for group formation and designing collaborative learning sessions.

Muehlenbrock in [9] proposes severals ways to regroup people for efficient collaboration in learning. His system takes into account the users' availability

detected automatically and also stores a static as well as a dynamic event profile for its users.

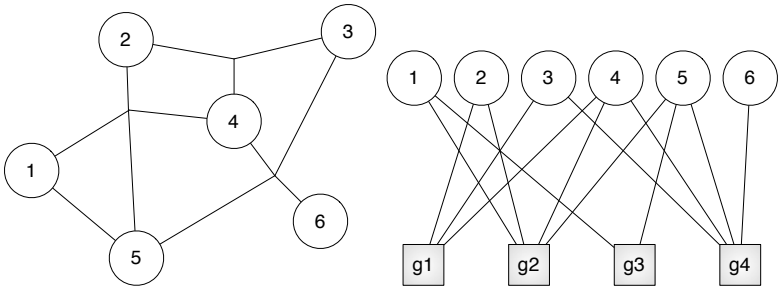
Wang *et al* in [12] propose a trust-based community formation method to recommend scientific papers. Users regroup around common interests and communities are built between users having a reciprocal high trust. Their method is based on a pairwise notion of trust and the trust a user has in a community is simply the average trust she has in its members.

All these methods use individual characteristics of the users and none of them uses the results from previous non-pairwise collaborations that may be really helpful in capturing all the complexity of human interactions. In the following section of this paper, we introduce a method based on the notion of collective trust and on the idea that groups that performed well in the past should perform well in the future. This method is orthogonal to all the methods presented in this section and can therefore be used to enhance the results provided by those as well as used alone.

3 Proposed Method

To overcome the limitations of the existing approaches, described in the previous section, we introduce the notion of collective trust. Having a non-pairwise trust metric allows to capture the interaction between users inside a group. For example, two users trusted independently can be untrustworthy when collaborating together and a small group can be really efficient while disappointing when integrated into a bigger structure. In reality personal factors may affect professional collaborations even if two people have compatible profiles. These notions are close to impossible to capture using individual profiles and have to be acquired with experience.

3.1 Collective Trust



(a) Hypergraph (b) bipartite graph

Fig. 1: Graph representations for collective trust

This notion of collective trust is exactly the same as the trust defined in [11] except that it applies to groups of people/products instead of just one entity. It is based on the interactions someone has with a group of users/products.

Let $\mathcal{U}$ be a set of users, for each $g \in 2^{\mathcal{U}}, Tr(g) \in [a,b] \cup \{\perp\}$. $Tr(g) = \perp$ means that g has never done a task and therefore has no trust value yet. Then after each interaction involving the group g, $Tr(g)$ is adjusted using the following formula:

$$Tr(g) = \begin{cases} (1-\alpha) \cdot \frac{b-a}{2} + \alpha \cdot e & \text{if } Tr(g) = \perp \\ (1-\alpha) \cdot Tr(g) + \alpha \cdot e & \text{otherwise} \end{cases} \tag{1}$$

where $\alpha \in [0,1]$ is the learning rate and $e \in [a,b]$ is the result of the interaction valued on a scale from totally negative to totally positive.

This notion of trust can be represented by a bipartite graph or a hypergraph as shown on Fig. 1. In those representations the circle nodes represent the users. Every time a new group is assembled, an hyperedge (see Fig. 1a) or a group node (grey square, see Fig. 1b) is created or updated if it already exists. The two figures are equivalent and represent a state with 6 users where the four following groups $\{2,3,4\}, \{1,2,4,5\}, \{1,5\}, \{3,4,5,6\}$ have already been put together at least once. The hyperedges/group nodes store all the group related information: trust, cardinality, number of interactions, etc. This way it is possible to access the information regarding the previous experiences of a user directly from her node.

3.2 Group Formation

The group formation problem consists in selecting the "best" group of people for one task, meaning the "group" that has the highest chance of success or that will outperform all the other possible groups on this particular task. The group formation problem can be modeled as follows:

$\mathcal{U} = \bigcup_{i=1}^{p} \mathcal{U}_i$ a set of users where $\mathcal{U}_i$ represents a specific type of users. The subset of available users is $\mathcal{U}_a \subseteq \mathcal{U}$. This notation naturally transpose to the types of users and $\mathcal{U}_{i_a}$ will denote the set of available users of type i.

A task $T = (t_i)_{1 \leq i \leq p}$ where $\forall i, t_i \in \mathbb{N}$ is a $p-$tuple specifying how many users of each type are required to accomplish the task. All tasks belong to the set $\mathcal{T}$.

A function $eval : \mathcal{T} \times 2^{\mathcal{U}} \to [a,b]$ that evaluates the success of a group on a specific task. Classic values for $[a,b]$ are $[0,1]$ if the worst a group can do is being inefficient or $[-1,1]$ if a group can worsen the situation by doing something.

The objective of the group formation problem is to find the group of available users $g \in 2^{\mathcal{U}_a}$ that fits the requirement of the task and that will maximize the *eval* function over all the possible groups of available users. This can be written more formally as follows:

$$\begin{array}{rcl} \mathcal{T} \times 2^{\mathcal{U}} & \longrightarrow & 2^{\mathcal{U}_a} \\ group : (T, \mathcal{U}_a) & \longrightarrow & group(T, \mathcal{U}_a) \end{array} \tag{2}$$

such as $\forall T \in \mathcal{T}, group(T,\mathcal{U}_a) = \emptyset \vee \forall i, |group(T,\mathcal{U}_a) \cap \mathcal{U}_i| = t_i$ that maximizes the value of $eval(T, group(T,\mathcal{U}_a)), \forall T$. This function either returns a group fit

for the task or no group at all, if there are not enough available users of each type to complete the task.

We define $\mathcal{U}_a^T = \{u \in 2^{\mathcal{U}_a} | \forall i, |u \cap \mathcal{U}_i| = t_i\}$, *i.e.* this is the set of all possible groups for the task T. Then, $\forall g \in \mathcal{U}_a^T, |g| = \sum_{i=1}^{p} t_i = n$.

3.3 Collective trust for group formation

Our method to find the "best" group for any given task is based on the collective trust metric introduced previously. The main idea is that collaborations that were efficient in the past should be efficient again, if put back together. The proposed *group* function is the following:

$$group(T, \mathcal{U}_a) = \text{argmax}_{g \in \mathcal{U}_a^T} \left(ETr(g)\right) \tag{3}$$

where $ETr(g)$ is the estimated trust of the group g. The estimated trust of a group g is its trust value $Tr(g)$ if it has one. Otherwise, in order to estimate the trust we can have in a group that has never been put together before, we will look at the sub-groups it contains that have already been tested and use a linear combination of their weighted trust values as an estimate. The actual computation of the estimated trust goes as follows:

$$ETr(g) = \begin{cases} Tr(g) & \text{if } Tr(g) \neq \perp \\ \sum_{k=1}^{n} \frac{k}{\sum_{h \in C_g} |h \cap g|} \cdot \sum_{h \in C_g^k} \frac{k}{|h|} \cdot Tr(h) & \text{otherwise} \end{cases} \tag{4}$$

where $C_g^k = \{h \in 2^{\mathcal{U}} | Tr(h) \neq \perp \wedge |h \cap g| = k\}$ and $C_g = \bigcup_{k=1}^{n} C_g^k$. The idea is to use the trust of all groups $h \in C_g^k$ that share k members with the group g. $Tr(h)$ is multiplied by $k/|h|$ since only k members are selected and they only account for some amount of the whole group trust.

To guarantee that $ETr(g) \in [a, b]$, the value contributed by each group $h \in C_g^k$ is weighted by $k/(\sum_{h \in C_g} |h \cap g|)$. This particular weighting gives more importance to bigger groups since they will represent a bigger part of the final group and will have a bigger influence of the efficiency of the group.

Selecting the best group.

Evaluating the estimated trust for a group can be done in linear time, regarding the number of users and groups, using either the bipartite graph or hypergraph representations depicted in Fig. 1. The computational problem comes from the number of possible groups g, for which trust needs to be estimated, for every task T:

$$|\mathcal{U}_a^T| = \prod_{i=1}^{p} \binom{t_i}{|\mathcal{U}_{i_a}|} \tag{5}$$

This number can grow really fast and become exponential which will be untractable in most cases. Therefore it is really important to consider approximate algorithms that will try to build the most trustworthy group for a task without actually computing all the estimated trusts.

This algorithm returns the group corresponding to the task T with the highest estimated trust or the empty set.

input: $\mathcal{U}_a$, available users
input: $\mathcal{G}$, preexisting groups
input: T, the given task
output: g, such that $g = \emptyset \vee |g| = n$.

1. $g = \emptyset$
2. **if** $\exists i, |\mathcal{U}_{i_a}| < t_i$ **then return** g
3. **while** $(|g| < n)$ **do**
4. $\quad g' = argmax_{h \in \mathcal{G}}(Tr(h) \cdot \frac{i}{|h|})$ where $i \leq n - |g|$
5. $\quad \mathcal{U}_a = \mathcal{U}_a \setminus g'$
6. $\quad g = g \cup g'$
7. **return** g

Fig. 2: Greedy algorithm

Heuristic. This heuristic is a very simple greedy algorithm presented in Fig. 2. This algorithm builds a group by successively adding people from the most trusted groups. $\mathcal{G}$ represents the set of all groups with a non void trust value, *i.e.* $\mathcal{G} = \{g \in 2^{\mathcal{U}} | Tr(g) \neq \perp\}$. On step 3 of the algorithm, we select $i = n - |g|$ members at most from the available users. If the group we are selecting users from contains more than i users, we just select i users randomly. Steps 4 and 5 simply remove those users from the available ones and add them to the "best" group that will be selected for the task.

It is important to note that in the case where all the groups that have already been tested and possess a trust value are independent, the problem can be reformulated as a continuous 0-1 Knapsack problem [7] that is solved exactly by the greedy algorithm presented in Fig. 2.

4 Future Work

The first thing to do is to evaluate the proposed method trough extensive simulations and a real life experiment. The objective of the simulations will be to assess the quality of the proposed heuristic and to test its efficiency against several other methods like random assignment, methods presented in section 2 and pairwise trust schemes. The real life experiment will demonstrate the feasibility of the method.

A really important problem that requires further investigation is the estimated trust that one should have in a user that has never been part of any group. This estimated trust should be high enough to favor the incorporation of new users over members of poor previous collaborations but should not replace members of previous successful collaborations. The right threshold will be estimated using the simulations. It is important to notice that this threshold will in

reality be task dependent. For example, it may be better to test new combinations on a common task while relying on known "good" teams for more critical tasks.

Another really important problem is the group partition problem. The objective here is not to find the "best" possible group to achieve a task but rather to separate the set of available users into groups of same sizes with homogeneous trust levels. This problem arises often in education where professors have to divide their classes for group work. If all the professors inside a university were to log the groups they made together with their performance, it will provide all the data required to compute the other classes' groups' estimated trust. Reusing the model presented in section 3, the group partition problem consists in finding a function:

$$\begin{array}{rl} \mathcal{T} \times 2^{\mathcal{U}} & \longrightarrow \text{Partition}(\mathcal{U}_a) \\ partition_\epsilon : (T, \mathcal{U}_a) & \longrightarrow \quad (P_i^T)_i \end{array} \tag{6}$$

such as $\forall T, \forall i, \forall j, |P_i^T \cap \mathcal{U}_j| = t_j \wedge \forall T, \forall i, \forall j > i, |eval(T, P_i^T) - eval(T, P_j^T)| < \epsilon$

Our idea is to use the collective trust also for partitioning and make sure that all members of the partition have a similar estimated trust value. In order to provide a partition of $\mathcal{U}_a$ that provides groups with homogeneous trust levels, we will look for the partition that verifies one of the following properties:

$$\min_{P^T} \left(\sum_i \sum_{j>i} |ETr(P_i^T) - ETr(P_j^T)| \right) \tag{7}$$

$$\forall i, \forall j > i, |ETr(P_i^T) - ETr(P_j^T)| < \epsilon \tag{8}$$

Eq. 8 is more accurate since we want to guarantee that the level is homogeneous between groups but it might be difficult to set ϵ to get the best possible partition of users. On the other side Eq. 7 is always satisfied by at least one partition but this partition may not be really homogeneous especially if the number of groups in the partition is important. It is then application dependent to decide if having some outliers is really troublesome.

This collective trust metric can be adapted to recommend group of products to users. A good example can be online learning materials. People with different learning styles will be sensible to different kinds of learning materials and combination of learning materials.

5 Conclusion

In this paper, we presented a new scheme to select people to build a group based on the notion of collective trust. We strongly believe that this notion of collective trust is much more accurate in capturing the complexity of interactions between users than any individual based method. We also provided a heuristic

to efficiently build the "most" trustworthy group. We will design simulations to prove the efficiency of the proposed heuristic as well as investigate other promising domains of application for the collective trust like recommendations of group of products.

References

1. Artz, D., Gil, Y.: A survey of trust in computer science and the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 58–71 (2007)
2. Deibel, K.: Team formation methods for increasing interaction during in-class group work. In: Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education. pp. 291–295. ITiCSE '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1067445.1067525
3. Felder, R., Silverman, L.: Learning and teaching styles in engineering education. Engineering education 78(7), 674–681 (1988)
4. Gummadi, A., Yoon, J.: Modeling group trust for peer-to-peer access control. In: Database and Expert Systems Applications, 2004. Proceedings. 15th International Workshop on. pp. 971–978. IEEE (2004)
5. Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R., Toyoda, J.: How can we form effective collaborative learning groups? In: Gauthier, G., Frasson, C., VanLehn, K. (eds.) Intelligent Tutoring Systems, Lecture Notes in Computer Science, vol. 1839, pp. 282–291. Springer Berlin / Heidelberg (2000), http://dx.doi.org/10.1007/3-540-45108-0_32, 10.1007/3-540-45108-0_32
6. Isotani, S., Inaba, A., Ikeda, M., Mizoguchi, R.: An ontology engineering approach to the realization of theory-driven group formation. International Journal of Computer-Supported Collaborative Learning 4(4), 445–478 (2009)
7. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack problems. Springer Verlag (2004)
8. Martin, E., Paredes, P.: Using learning styles for dynamic group formation in adaptive collaborative hypermedia systems. In: In Proceedings of the First International Workshop on Adaptive Hypermedia and Collaborative Web-based Systems (AHCW 2004) (2004) 188-198 available at http://www.ii.uam.es/rcarro/AHCW04/MartinParedes.pdf
9. Muehlenbrock, M.: Learning Group Formation Based on Learner Profile and Context. International Journal on E-Learning 5(1), 19–24 (2006)
10. Oakley, B., Felder, R., Brent, R., Elhajj, I.: Turning student groups into effective teams. Journal of Student Centered Learning 2(1), 9–34 (2004)
11. Wang, Y., Vassileva, J.: Trust and reputation model in peer-to-peer networks. In: Peer-to-Peer Computing, 2003.(P2P 2003). Proceedings. Third International Conference on. pp. 150–157. IEEE (2003)
12. Wang, Y., Vassileva, J.: Trust-based community formation in peer-to-peer file sharing networks. In: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence. pp. 341–348. IEEE Computer Society (2004)
13. Wessner, M., Pfister, H.R.: Group formation in computer-supported collaborative learning. In: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work. pp. 24–31. GROUP '01, ACM, New York, NY, USA (2001), http://doi.acm.org/10.1145/500286.500293