Investigation of Group Formation using Low Complexity Algorithms

Christos E. Christodoulopoulos[1] and Kyparissia A. Papanikolaou[2]

[1]Technology Education and Digital Systems
University of Piraeus, Greece
[2] General Department of Education
School of Pedagogical and Technological Education, Greece
{christos.c@ieee.org, spap@di.uoa.gr}

Abstract. Designing tools that support group formation is a challenging goal for both the areas of adaptive and collaborative e-learning environments. Group formation may be used for a variety of purposes such as for grouping students that could potentially benefit from cooperation based on their individual characteristics or needs, for mediating peer help by matching peer learners, for facilitating instructors proposing an initial grouping approach. In this paper, we discuss several factors that need to be considered when assigning learners to groups. We also investigate the use of the c-means family clustering algorithms and uniform distribution, for group formation. The fuzzy c-means is compared to (a) the k-means algorithm for homogenously grouping students, and (b) a random selection algorithm (based on the uniform distribution) for formulating heterogeneous groups. Preliminary results from grouping 36 students based on 2 and 3 criteria, indicate the potential of the fuzzy c-means algorithm for homogenously grouping students, and the random selection algorithm as a low complexity approach for achieving a significant level of heterogeneity.

1 Introduction

Research on peer influences on learning suggests that students who form a group create a setting that facilitates or impedes learning above and beyond what would be expected. This denotes that how to form an effective group may have an impact to the educational benefit of group interaction. Support for group formation may be based on learner profile information [11], [7], [10] such as ability, prior knowledge, learning style, browsing behaviour, or learner context [8] such as location, time, and availability. Group formation may be used for a variety of purposes in different contexts such as (i) in a Computer-Supported Collaborative Learning (CSCL) context for grouping students that could potentially benefit from cooperation based on their complementarity of knowledge/skills or competitiveness, or for forming groups around problems with specific requirements [5], (ii) in a web-based learning environment for mediating peer help by matching peer learners based on their individual characteristics and/or learning needs on a particular subject/task [4], (iii) in a classroom-based context to facilitate instructors in formulating effective learning groups proposing an initial grouping approach [6]. Critical open issues in the area remain (a) the criteria based on which learners that should maximally benefit from each other when working together are grouped, and (b) the computational issues arising in the implementation of group formation support.

In this study we discuss different factors that need to be considered when assigning learners to groups and focus on the computational problem of selecting appropriate to effectively operate on small data sets algorithms for formulating groups. In particular, in Section 2 we suggest specific factors influencing the group formation process, present a brief literature review on algorithms used in group formation and introduce the c-means family algorithms. In Section 3 we compare the k-means to the fuzzy c-means algorithm for group formation purposes. Moreover, a random algorithm based on the uniform distribution is proposed as an alternative approach for generating heterogeneous groups. Preliminary results provide evidence for the effectiveness of the fuzzy c-means algorithm in formulating homogenous groups and the appropriateness of the uniform distribution in heterogeneous groups; however in order to reach a safe conclusion a series of tests should be performed in a real context.

In: P.Brusilovsky, M. Grigoriadou, K. Papanikolaou (Eds.): *Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level*, 11th International Conference on User Modeling, 2007, pp. 57-60

2 Algorithms for Assigning Learners to Groups

Support for group formation aims to facilitate the process of assigning students to groups and increase the possibility that groups will satisfy specific criteria. In particular, specific factors that need to be considered when assigning learners to groups concern:

- the *criteria* used for effective grouping: the number and type of criteria that the group members should satisfy; in an educational context these criteria may reflect specific learning characteristics such as ability, prior knowledge, style, competence, or context such as problem requirements, learners' location or availability
- the *level of homogeneity/heterogeneity* of the groups may be considered as a group characteristic or reflect the status of the group based on specific characteristics of the individuals, resulting in: (a) homogenous/heterogeneous groups such as a group of students with complementary knowledge/skills, or (b) groups that are homogenous according to specific criteria and heterogeneous based on others such as groups consisting of learners with same ability and mixed styles,
- the *size of the groups* in terms of the number of members included in each group.

Different algorithms have been used for formulating homogenous and heterogeneous groups. In particular, recent studies propose the use of optimization algorithms as an effective solution for assigning groups. Cavanaugh et al. [2] propose the repeated use of the 'hill climbing' optimization algorithm with weighted criteria defined by the instructor, for assigning homogeneous and heterogeneous groups in a web-based environment. Bekele focuses on heterogeneity proposing a mathematical approach which uses the Ant Colony Optimization algorithm for maximizing group heterogeneity [3]. However, the crucial parameter of low complexity remains an open issue.

In this study we focus on less complex and more time-saving approaches yet sufficiently effective algorithms such as clustering algorithms for homogeneous grouping and a simple random algorithm for heterogeneous grouping. Clustering algorithms are a category of optimization algorithms designed to discover groups in data. They try to minimize an objective function which is derived from (dis-)similarity measures (usually distance) between the data. The c-means algorithms belong to the partitional clustering algorithms as they try to form clusters by dividing the data. They present a series of advantages compared to other clustering and even most optimization algorithms. First of all they take as input the desired number of clusters to be found, which is a drawback for real-life data mining, but essential to our application. Moreover, they are easy to implement in scripting languages (PHP, JavaScript). Finally one of the main reasons for their popularity (especially for k-means) is the fact that they converge extremely quickly. Their computational complexity is $\Omega(n)$ where n is the number of data points. However, the use of clustering algorithms for group formation presents also several disadvantages: (a) they form only homogeneous groups: clustering algorithms are used for grouping similar data, so it is not possible for them to create clusters that maximize the dissimilarity measure, (b) inability to evenly distribute the data points along the clusters: they take as input the desired number of clusters/groups to be found whilst they are not interested in the number of group members, and (c) limited advantages in comparison to simple sorting algorithms when used with one criterion.

In this research, we compare two clustering algorithms of the c-means family, the k-means and fuzzy c-means algorithms, in grouping students based on specific criteria. Both algorithms have been extendedly used in application areas such as image processing or data mining in large sets of data. In group formation where data sets are usually small, the performance of both algorithms needs to be re-examined. k-means was proposed by McQueen in 1967 and since then it has become one of the most commonly used clustering algorithms. It is also referred Hard C-Means (HCM) in comparison to the Fuzzy C-Means (FCM) algorithm. The simplicity and the speed of HCM are obvious since it is based on the Euclidian distance that can be estimated by a series of multiplications. However its main drawback is the inability to evenly distribute the data points along the clusters. Interchanging members between neighbor clusters can face this problem, but the complexity of the process is greater than that of the main algorithm. Fuzzy c-means algorithm also known as Fuzzy ISODATA was proposed by James Bezdek in 1973 and is basically an extension of the k-means algorithm to fuzzy sets [1]. Although FCM is more complex than k-means it is still reported to have linear complexity ($\Omega(n)$) making it as fast as k-means. Besides being fast, FCM seems to perform better than k-means when they were both evaluated with standard data mining quality measures [9]. The main advantage of FCM for group formation

derives from the membership function. In FCM a data point may belong to more than one cluster with a different probability. This feature, allows us to address the problem of inequality of the clusters in a more effective way, as we can exchange data points between clusters based on their membership probabilities. This information could be also provided to the expert-teacher as a useful aid to support final decisions on grouping students.

3 Formation of Homogenous & Heterogeneous Groups

Group homogeneity: Comparing FCM with HCM. FCM and HCM have been tested in forming homogenous groups with a set of 36 students based on 2 and 3 criteria that assess students' style in 2 or 3 different style categories respectively. In our case, homogenous groups consist of students with similar characteristics. The algorithms used were the standard MATLAB's implementations.

FCM and HCM were compared based on their *effectiveness,* which was evaluated according to specific cluster validity measures. We decided to use such general measures since there is no advanced validity measure that apply to both fuzzy and non-fuzzy algorithms. A commonly used measure is Squared Sum Error (SSE) (see equation 3). Since SSE describes the coherence of a given cluster, we expect that "better" clusters give lower SSE values.

$$SSE(C) = \sum_{j=1}^{c} \sum_{x \in C_j} d^2\left(x, v_j\right) \quad (\mathbf{1})$$

In Table 1 we can see that in most cases the FCM algorithm gives lower SSE values than the HCM producing more coherent or homogeneous clusters/groups. For example, in rows 1 and 4, the value of the SSQ for the FCM is 68.1037 and 58.4633 and for the HCM 98.2000 and 78.3333 accordingly. Only in the 2nd row the HCM appears slightly better. Lastly, in three different groupings (rows 3, 5 and 6), the HCM could not respond to the input conditions.

Table 1. SSE values of FCM and HCM in varying number of groups and criteria.

Number of groups	Number of criteria	SSE of FCM	SSE of HCM
6	2	68.1037	98.2000
6	3	93.9199	88.4179
9	2	75.2139	not responding
9	3	58.4633	78.3333
12	2	77.1776	not responding
12	3	52.3504	not responding

One major advantage of FCM, which emerged from our tests, is its ability to work in spaces that contain a limited amount of data (i.e. students in class ~20-100) and with small groups (the number of students per group decreases when the number of groups increases), whereas HCM seems to be unstable under these conditions. In our data space that contains 36 students, HCM performed well in 9 groups (4 students per group) when used with 3 criteria, but it was unable to produce clusters with 2 criteria (see in Table 1 – rows 4 and 3 respectively). Moreover if we downsize even more the number of students per group (in cases where the algorithms should generate 12 groups), the standard implementation of HCM stops responding. Based on these results, and taking into account that usually group formation apply to small data sets whilst groups consist of a few members, we conclude that the classic HCM seems not to be a viable solution.

Group Heterogeneity: the standard random algorithm. Heterogeneity in group formation is a relatively vague term. For example in [3] heterogeneity refers to mixed ability groups and as the authors suggest "a reasonably heterogeneous group refers to a group where student-scores reveal a combination of low, average and high student-scores". Thus, a heterogeneous group might be defined as a group in which all the different values of the data space can be found. However in cases where more than one criterion is used, with a range of values for each one, then it becomes even harder to define group heterogeneity. In general, when defining the dissimilarity measure then it is a typical optimization problem to maximize its value. Another interesting approach to investigate, due to its low complexity, is to form heterogeneous groups by applying a uniform distribution on the data space. To this end a random algorithm may be used in order to achieve some level of heterogeneity. Especially in cases where the level of group heterogeneity needs not to be the maximum possible, the random algorithm could be effectively used. Moreover, compared

to most optimization algorithms, the standard random algorithm is by far faster, making it even more appealing as a choice. For validation purposes, we used MATLAB's implementation of random selection which follows the uniform distribution without replacement. We also use the cluster dispersion as a measure of heterogeneity in order to validate this approach. Cluster dispersion is defined as the cluster's diameter which is the maximum distance of any two data points belonging to the same cluster. In particular, we compare the maximum and mean diameters of all the clusters created by the random algorithm and FCM. The results are presented in Table 3, where in every grouping (each one corresponds to a different row), the maximum diameter of the clusters generated by the uniform distribution appears significantly greater than that of clusters generated by the FCM, and close enough to the maximum ones (≈7 for the squared space - 2 criteria - and ≈8.6 for the cubed space - 3 criteria).

Table 2. Cluster dispersion generated by FCM and uniform distribution algorithms.

Number of groups	Number of criteria	FCM		Uniform distribution	
		Max. Diam.	Mean Diam.	Max. Diam.	Mean Diam.
6	2	2.2361	1.5107	5	4.0271
6	3	2.4495	2.1748	6	4.5008
9	2	1.4142	0.8047	5	3.5068
9	3	3.3166	1.9170	5.8310	3.9531

5 Conclusions and Further Research

In this study we investigated the potential of the k-means and fuzzy c-means algorithms for assigning homogenous groups of students and the random selection algorithm for heterogeneous groups. Preliminary experiments in a simulated environment indicate the appropriateness of the fuzzy c-means and uniform distribution algorithm for assigning groups. Especially, the output of the FCM algorithm may also be used to support instructors in group formation or students in identifying appropriate peers, by providing valuable information about the different groups that a student might better fit based on specific criteria.

References

1. Bezdek, J.C.: Pattern Recognition with Objective Function Algorithms, Plenum Press, New York (1981)
2. Cavanaugh, R., Ellis, M.G., Layton, R.A., Ardis, M.A. Automating the Process of Assigning Students to Cooperative-Learning Teams. In Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition (2004)
3. Graf, S., Bekele, R.: Forming Heterogeneous Groups for Intelligent Collaborative Systems with Ant Colony Optimization, In Proc. of 8th International Conference in ITS, Taiwan (2006)
4. Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A., and Vassileva , J.: The Intelligent Helpdesk: Supporting Peer - Help in a University Course,Proceedings of ITS 1998: 4th Int Conference on Intelligent Tutoring Systems, San Antonio, Texas, Springer - Verlag: Berlin (1998) 494-503
5. Hoppe, H.U.: The use of multiple student modeling to parameterize group learning, In J. Greer (Ed), Proceedings of AI-ED 95, Washington D.C., USA (1995)
6. Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R., & Toyoda, J.: How Can We Form Effective Collaborative Learning Groups?, Proceeding of ITS 2000, 282-291, Montreal, Canada (2000)
7. Martin, E., Paredes, P.: Using Learning Styles for Dynamic Group Formation in Adaptive Collaborative Hypermedia Systems. International Workshop on Adaptive Hypermedia and Collaborative Web-Based Systems, International Conference on Web Engineering, Munich (2004)
8. Muehlenbrock, M.: Learning group formation based on learner profile and context, Int Journal on E-learning, 5(1) (2006) 19-24
9. Serban, G., Moldovan, G.S.: A Comparison of Clustering Techniques in Aspect Mining, Informatica, vol LI, no 1, Studia University (2006)
10. Tang, T., Chan, K., Winoto, P. and Wu, A.: Forming Student Clusters Based on Their Browsing Behaviors. Proceedings of the 9th International Conference on Computers in Education (ICCE 2001), Seoul, Korea (2001) 1229-1235
11. Wilkinsona, I.A.G, Fung, I.Y.Y.: Small-group composition and peer effects, International Journal of Educational Research 37 (2002) 425–447