LEARNING GROUP FORMATION FOR MASSIVE OPEN ONLINE COURSES (MOOCs)

Sankalp Prabhakar and Osmar R. Zaiane
University of Alberta, Canada

ABSTRACT

Massive open online courses (MOOCs) describe platforms where users with completely different backgrounds subscribe to various courses on offer. MOOC forums and discussion boards offer learners a medium to communicate with each other and maximize their learning outcomes. However, oftentimes learners are hesitant to approach each other for different reasons (being shy, don't know the right match, etc.). In collaborative learning contexts, the problem of automatic formation of effective groups becomes increasingly difficult due to very large base of users with different backgrounds. To address this concern, we propose an approach for group formation of users registered on MOOCs using a modified Particle Swarm Optimization (PSO) technique which automatically generates dynamic learning groups. The algorithm uses the profile attributes of users in terms of their age, gender, location, qualification, interests and grade as the grouping criteria. To form effective groups, we consider two important aspects: a) intra-group heterogeneity and b) inter-group homogeneity. While the former advocates the idea of diversity inside a particular group of users, the latter emphasizes that each group should be similar to one another. We test our algorithm on synthesized data sampled using the publicly available MITx-Harvardx dataset. Evaluation of the system is based on the fitness measures of groups generated using our algorithm which is compared against groups obtained using some of the standard clustering techniques like k-means. We see that our system performs better in terms of forming effective learning groups in the context of MOOCs.

KEYWORDS

Group Formation, MOOCs, Online Learning

1. INTRODUCTION

In many collaborative learning contexts, students are organized into small groups to complete their tasks with a common group related goal (Lou, 2008). During the past decades, hundreds of studies have been made to investigate the effectiveness of collaborative learning. Most of these studies conclude that well-constructed learning groups can effectively drive teamwork among the members of the group and can have better performance than poor-constructed groups (Shimazoe, 2010), (Deibel, 2005). Moreover, there are studies that tell us that the conventional approaches of grouping students together based on self-selection or random-selection are not well suited in educational domain (Shimazoe, 2010). Group formation in education essentially requires a broader study of students' backgrounds, their traits and a know-how of the instructional environment.

The emergence of Massive Open Online Courses (MOOCs) as a major source of learning in the modern world has created several challenges in terms of forming effective groups of learners. More people signed up for MOOCs in the year 2015 than they did in the first three years of the 'modern' MOOC movement (which started in late 2011 - when the first Stanford MOOCs took off) (Shah, 2015). The students registered on MOOCs have varied demographics in terms of the countries they originate from, languages they speak and their personality traits. Moreover, studies show that the lack of effective student engagement is one of main reasons for a very high MOOC dropout rate (Onah, 2014). Although many thousands of participants enroll in various MOOC courses, the completion rate for most courses is below 13%. Further studies (Lou, 2008), (Shimazoe, 2010), (Zepke, 2010) have been made to show how collaboration or active learning promotes student engagement. Hence, we believe that forming effective learning groups of students would foster better collaboration and could also help mitigate the dropout rates to some extent.

Keeping the above in mind, our work focuses on exploring the possibilities of assisting MOOC learners in the process of self-organization (e.g. forming study groups, finding partners, encourage peer learning etc.) by developing a group formation strategy based on predefined set of user attributes like age, gender, location, qualification, interests, grade etc. We use a modified particle swarm optimization (Kennedy, 2011) technique that helps in effective group formation by looking at the different user attributes along with the grouping conditions of intra-heterogeneity and inter-homogeneity. The idea is to form learning groups that are diverse internally while being similar to each other on certain aspects, to have the best possible learning outcomes.

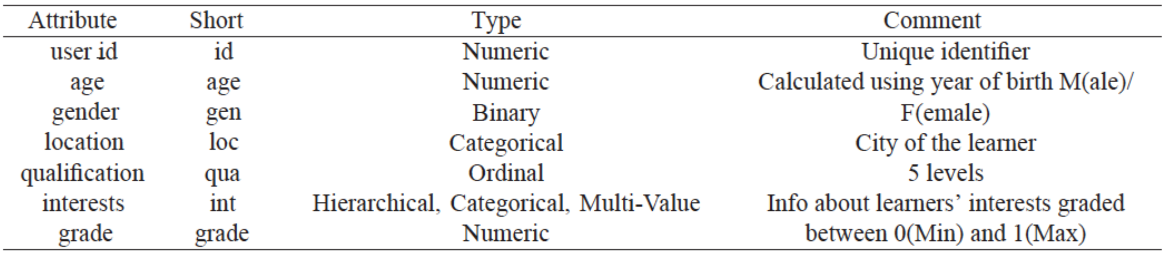
The remainder of the paper is organized as follows. Section 2 outlines the proposed model and data for generating effective groups using the modified particle swarm optimization technique. In section 3, experimental evaluation and results are presented. Finally, Section 4 ends with a conclusion and future work.

2. PROPOSED METHOD

We look at the data model along with the design and description of the group formation algorithm.

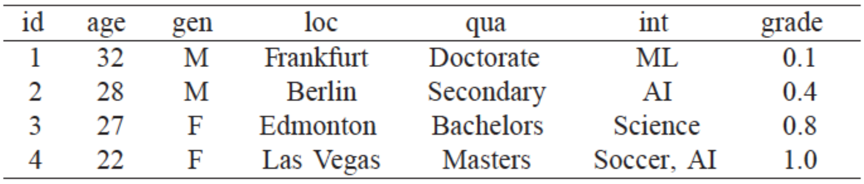
2.1 Data

The data used in our research comes from the de-identified release from the first year (Academic Year 2013: Fall 2012, spring 2013, and summer 2013) of MITx and HarvardX courses on the edX platform (HarvardX-MITx, 2014). These data are aggregate records, and each record represents an individuals' activity in one edX course and contains many diverse information about the profile of the learner (e.g. age, gender, location, qualification, grade etc.). For our analysis and without loss of generality, we selected records with attributes about age, gender, location, qualification and grade. Moreover, we enhance this information with synthesized data about learners' interests. This information is not available via the mentioned dataset but is potentially useful for creating effective groups. A brief overview of the dataset attributes can be found in Figure 1 along with a sample of our dataset in Figure 2.



Attribute	Short	Type	Comment
user id	id	Numeric	Unique identifier
age	age	Numeric	Calculated using year of birth M(ale)/
gender	gen	Binary	F(emale)
location	loc	Categorical	City of the learner
qualification	qua	Ordinal	5 levels
interests	int	Hierarchical, Categorical, Multi-Value	Info about learners' interests graded
grade	grade	Numeric	between 0(Min) and 1(Max)

Figure 1. Data Attribute Description



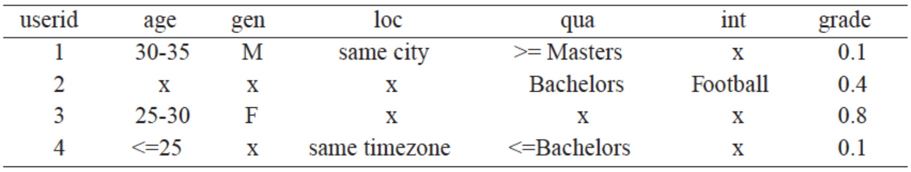
id	age	gen	loc	qua	int	grade
1	32	M	Frankfurt	Doctorate	ML	0.1
2	28	M	Berlin	Secondary	AI	0.4
3	27	F	Edmonton	Bachelors	Science	0.8
4	22	F	Las Vegas	Masters	Soccer, AI	1.0

Figure 2. Dataset Sample

2.2 Data Modelling

A description of how each attribute in Figure 1 is modeled, is as follows: 1) **age**: age range of the users' are segregated in these five bands: less than 20, 20-25, 25-30, 30-35, 35 and above. 2) **location**: location attribute is categorized into three options: same city, same country or same time zone. 3) **gender**: male or female gender options. 4) **qualification**: the qualification attribute has been divided into 5 levels: less than secondary, secondary, bachelors, masters and doctorate. 5) **interests**: the interest attribute contains one or more values about learners' interest. 6) **grade**: the grade attribute has averaged learners' grade from previous courses, between 0(min) and 1(max).

A sample of data vectors can be seen in Figure 3. The 'x's in the table represent null value. It must be noted that not all six attributes are required to be used for any kind of grouping. Our proposed algorithm is flexible enough to take one or more of these attributes for group formation. Moreover, we can tune the way each of these attributes contribute in group formation in terms of intra-group heterogeneity and inter-group homogeneity. For instance, a reasonably heterogeneous group would refer to a group where student-grades reveal a combination of low, average and high student-grades. This is justified by the recommendation of Slavin (Slavin, 1987) who proposed that students should work in small, mixed-ability groups. Hence, it is necessary that grade distribution is even across all groups i.e. the average grades of students across all groups should be same (inter-group homogeneity) while maintaining that within each groups the grades are diverse (intra-group heterogeneity).



userid	age	gen	loc	qua	int	grade
1	30-35	M	same city	>= Masters	x	0.1
2	x	x	x	Bachelors	Football	0.4
3	25-30	F	x	x	x	0.8
4	<=25	x	same timezone	<=Bachelors	x	0.1

Figure 3. Sample Data Vectors

Another important factor for group formation in collaborative learning is the interest of group members since it has the potential to change the involvement of individuals in learning (Freeman, 2014). A group with common interests will have more interactivity and discussions that is likely to make the learning process more engaging. The same can be said about the 'location' attribute. Students residing in the same city, country or time zone will be able to collaborate better due to minimal time differences.

2.3 Algorithm

In this section, we discuss our group formation algorithm in detail. In short, at first we use a modified K- means clustering algorithm (Hartigan, 1979) to fit our data attributes to seed initial swarm of particles. Then we use a hybrid particle swarm optimization technique to build the final group of learners.

2.3.1 Modified K-means

In modified K-means algorithm, at first all the cluster 'centroids' or 'mid-points' are randomly initialized using the data vectors. Then the distance for each data vector is calculated using a scoring system wherein the distance between each attribute of a data vector to that of its corresponding attribute of all centroids is calculated. The data vector is then assigned to that cluster where it has the least distance 'd' with its corresponding centroid as per the equation in the figure below:

$$d(z_p, m_j) = \sqrt{}(\sum_{k=1}^{N_d}(z_p k - m_j k)^2)$$

Figure 4. Distance Calculation

where k represents a particular dimension or an attribute, Nd denotes the input data dimension (number of attributes), Nc denotes the number of centroids of the clusters or the number of clusters to be generated, zp denotes the p-th data vector, mj denotes the centroid of cluster j.

The attributes are modelled in the following way for distance calculation: 1) age, qualification: age and qualification attributes are divided into levels in such a way that adjacent levels have a distance of one unit. The distance is then normalized in range [0 - 1] by dividing it by the maximum distance value possible. 2) gender, location: For any given categorical options of gender and location, if the values for any two users are same then the distance is 0 else 1. 3) interests: The hierarchy we used for interests of users is based on WordNet (Miller, 1995) and the similarity measure used is based on the Wu and Palmer method (Wu, 1994) score that considers the depths of the two synsets in the WordNet taxonomies, along with the depth of the LCS (Least Common Subsumer). Score for this similarity is between 0 and 1, since we are implementing our system in a distance measure (and not similarity) the final value of distance between the interests is [1 - score]. 4) grade: For grade attribute, distance measure between a data vector and centroid is simply the difference between their grade values.

In traditional k-means (Hartigan, 1979) algorithm, the centroids are typically recalculated by taking the average sum of all the data vectors present within a cluster until a stopping criteria is reached. However, in this case, centroids are recalculated in a different way based on each attribute value of every data vector within a particular cluster, as per the following rule:

1) *age, grade:* centroid value corresponding to these attributes is the mean of age and grade of every data vector present within the cluster.

2) *location, gender, qualification, interests*: centroid values for each of these attributes is the most common attribute value within the data vectors belonging to a particular cluster. Moreover, K-means clustering process ends when any one of the following stopping criteria is reached: when the maximum number of iterations has been exceeded or when there is little to no change in the centroid vectors over multiple iterations. We use k- means for two different purpose: 1) To formulate baseline clusters to compare against the clusters or groups generated using hybrid PSO algorithm and 2) To initialize one of the particles used in the hybrid PSO algorithm. We use two different baseline models for result comparison, as mentioned below:

1) Number of clusters/groups (k) is specified: In this case, the number of clusters to be formed using k-means is specified by the user. Each cluster obtained after running k-means will have data vectors that are very similar to each other. However, in order to have intra-cluster heterogeneity we need to have diverse data vectors within a cluster. To build an unbiased baseline model, we create equal number (k) of empty clusters. Then using the first cluster obtained via k-means, we evenly distribute the data vectors in them to each of these empty clusters. We repeat this process with all other data vectors from the clusters obtained using k- means. In the end, we have a new set of clusters with data vectors, which are diverse and can be used as a good baseline for result comparison.

2) Number of users (α) in a cluster/group is specified: In this case, the number of users in each cluster or group is pre-decided. In order to account for intra-cluster heterogeneity, we create empty clusters, each with size α. Every cluster is then filled with data vectors obtained from each of the clusters generated using k-means until a value is reached. In the end, we have new set of clusters (size α) with data vectors that are diverse and can be used as a good baseline for result comparison.

Next, we discuss the hybrid particle swarm optimization (PSO) algorithm. We modify the standard PSO algorithm for MOOCs and combine it with modified k-means to build a hybrid algorithm for group formation.

2.3.2 Hybrid Particle Swarm Optimization

Over several years, the particle swarm optimization (Kennedy, 2011) has been used to solve various problems of the level of complexity NP-Hard (Jarboui, 2008), (Yin, 2006). The results of these studies show that PSO has been very effective in solving problems of this level of complexity. Our problem involves optimization of different student attributes, hence we used hybrid PSO to form effective learning groups. The aim of hybrid PSO is to find an optimum solution based on a certain fitness function. Every particle is evaluated with respect to this fitness function; the fittest particle is accepted as solution. In hybrid PSO, we calculate the velocity and position of all particles after every iteration based on the equations below:

$$v_{i,k}(t+1) = wv_{i,k}(t) + c_1 r_{1,k}(t)(pBest_{i,k}(t) - x_{i,k}(t)) + c_2 r_{2,k}(t)(gBest_{i,k}(t) - x_{i,k}(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Figure 5. Velocity and Position Equation

where x_i is the current position of the particle, v_i is the current velocity of the particle, w is the inertia weight, $c1$ and $c2$ are the acceleration constants, $r1{,}k(t)$, $r2{,}k(t)$ are random numbers between (0,1), and $k = 1, \ldots, N_d$.

In the context of grouping, a single particle in PSO represents the N_C cluster centroid vectors, wherein each particle x_i is constructed as follows:

$$\mathbf{x}_i = (\mathbf{m}_{i1}, \ldots\ldots \mathbf{m}_{ij} \ldots\ldots, \mathbf{m}_{iNc})$$

where m_{ij} refers to the *j-th* cluster centroid vector of the *i-th* particle in the cluster C_{ij} .

Therefore, a swarm represents a number of candidate solutions, as each particle in itself is a solution. We use the modified k-means to initialize the N_C centroid vectors of one of the particles of the swarm. The centroid vectors of remaining particles are initialized randomly using the data vectors. Once the groups of all particles are initialized, we calculate the fitness of each particle, which is measured using the following fitness error functions:

$$fitness(P_{i_{grade}}) = \forall c_{ij} \epsilon N_c : |max_{(}c_{ij_{grade}}) - min_{(}c_{ij_{grade}})|$$

$$fitness(P_{i_{loc,int}}) = \frac{\sum_{j=1}^{N_c} [\sum \forall z_p \epsilon C_{ij}, d(z_p.m_j)/|C_{ij}|]}{N_c}$$

Figure 6. Fitness Equations

where ‘d’ is Euclidean distance defined in figure 1, |Cij| is the number of data vectors belonging to group C_{ij} .

Above-mentioned equations are fitness measures of a particle in terms of 'grade' and ['location', 'interest'] attributes respectively. The less the fitness error, the better the quality of groups formed. More specifically, if the grade difference between the max and min grade value for all groups within a particle is less than a threshold t (t=0.1), the particle is fit. We select the *gBest* (global best) and the *pBest* (personal best) particles based on the combination of fitness achieved using equations in Figure 6. The particle with least 'grade' difference and minimum 'location' and 'interest' distances, is selected as the global best. In addition, each particle stores its local best state, which has the least grade difference and minimum 'location' and 'interest' distances, in any given iteration.

Next, we update the group centroids of each particle using equations in Figure 5. However, the update for each attribute of the centroids is different from each other. In case of age and grade attributes, the updated values depend on the age, grade values of global best and personal best particle whereas for all other attributes [location, gender, qualification and interests], the updated values depend on the most common values of all data points in respective groups. This entire sequence completes one iteration of the algorithm. PSO is usually executed until a specified number of iterations has been exceeded or if a certain level of fitness has been achieved.

Below is the summary of group formation using hybrid particle swarm optimization (PSO):

```
1. Initialize each particle with $N_c$ randomly selected cluster centroids, except one centroid which is
   initialized using modified k-means.
2. for iteration $t$ = 1 to $t_{max}$
    (a) for each particle i do
    (b) for each data vector $z_p$
        i. Calculate the attribute distances $d(z_p, m_{ij})$, between the data vector $z_p$ with each
           cluster centroid $m_{ij}$, for all cluster centroids $C_{ij}$.
        ii. Assign $z_p$ to the Cluster $C_{ij}$ where the distance is minimum.
        iii. Calculate the fitness of particle using equations in Figure 6.
    (c) Update the global best particle in the swarm along with the personal best of each particle.
    (d) Update the group centroids of each particle using equations in Figure 5.
```

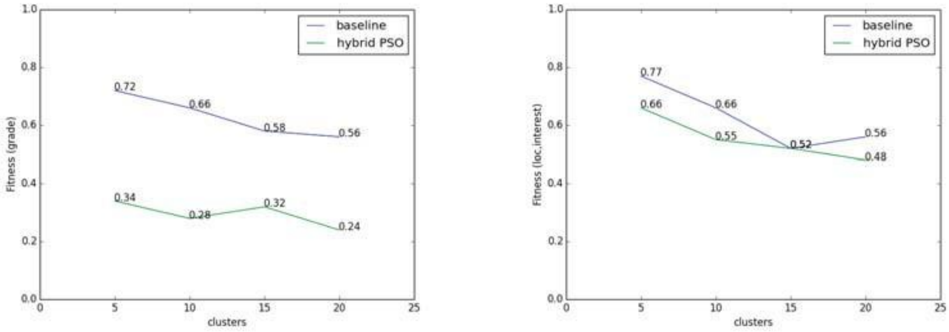
3. EXPERIMENTS AND RESULTS

Our experiments employed a series of testing to analyze the effectiveness of the PSO algorithm for group formation in MOOCs. We compare the quality of clusters generated using the modified k-means and hybrid PSO based on the calculated fitness error as defined in equations in Figure 6. The objective is to help us measure diversity inside each of the group while at the same time making sure that every group is similar to the other based on the grading levels.

The hybrid PSO algorithm was run on a computer with 2.7 GHz Intel Core i5 processor and 8 GB RAM. In order to examine the effectiveness of the PSO, five different sets of data for each [100, 1000, 5000] samples were generated randomly from the original dataset that has around 300k records. The parameters used for velocity update (refer equation 2) are, w = 0.72 and c1 = c2 = 1.49. These values were chosen to ensure good convergence (Bratton, 2007). In addition, the number of particles predefined is [10, 20, 50] respectively for data with volumes of [100, 1000, 5000] records. This was chosen based on the study (Chen, 2010) that any number of particles between 10 to 100 are capable of producing results that are clearly superior or inferior to any other value for a majority of the tested problems. The results reported is averaged over five different simulations, each simulation was run with different data samples. Our results will be analyzed on two different baseline models: 1) Number of groups (k) is specified and, 2) Number of users (α) in a group is specified.

3.1 Results

Figure 7 below shows the effect of varying the number of groups on the fitness values for 'grade difference', 'interest' and 'location' distances for 100 data records. As expected, the fitness error should go down as the number of groups increase. We calculate the grade fitness based on equation in Figure 6, wherein the difference between the maximum and minimum grade values is taken from all the groups. This difference is represented as the fitness score in figure 7 (a). It is seen that the fitness score decreases with increase in number of groups that means that the quality of groups formed increase as the number of groups increase. Next, we calculate the 'location' and 'interests' fitness based on equation in figure 6. The total distance for 'location' and 'interest' is normalized to produce a fitness score that is shown in Figure 7 (b). A similar pattern is seen wherein the fitness score decreases with increase in number of groups.



(a) Grade Fitness

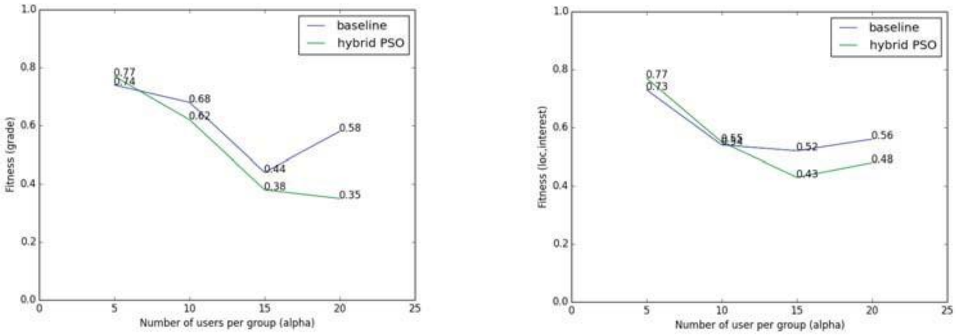
(b) Fitness for Location, Interests

Figure 7. Effect of Different Number of Groups on Fitness

We also compare the fitness results when the number of users in a group (α) is predetermined; the results are shown in Figure 8. Looking at the grade fitness graph (Figure 8 (a)), the fitness error decreases with increase in the number of users per group. This is expected because with more users the chances of 'grade' scores being skewed decreases, hence the grade fitness increases.

However, the grade, location and interests' fitness for the hybrid and baseline model is close for low values of (α). This can be attributed to the fact that with lower number of learners in a group, the chances of similar values for the mentioned attributes within a group decreases.

Similarly, for 'location' and 'interest' fitness (Figure 8(b)), the hybrid PSO models performs the same as the baseline model when the number of users per group are less. However, it outperforms the baseline when the number of users per group increase. However, the overall fitness error may increase even with the increase in number of users. It can be seen that the fitness score increases from 0.43 to 0.48 when the number of users per group increase from 20 to 25.



(a) Grade Fitness

(b) Fitness for Location, Interests

Figure 8. Effect of Number User Per Group (Α) on Fitness

Overall, the results show that the hybrid PSO model outperforms both the baseline models for generating better quality groups. Although, the algorithm could not be tested real-time on an actual MOOC platform, these results nevertheless provide promising insights when applying hybrid PSO technique in group formation using student attributes. Hence, it would be worthwhile to integrate it within an actual MOOC to get a realistic opinion on its performance.

4. CONCLUSION AND FUTURE WORK

In this paper, we presented a framework using a hybrid particle swarm optimization to form student groups based on attributes like [age, gender, location, qualification, interests and grade]. The evaluation of the proposed algorithm was done in the previous section to determine the overall quality of groups formed in terms of fitness. The results showed that the group quality was better when compared to the baseline model of groups formed using the modified k-means method. The proposed strategy can help the instructors to automatically generate suitable learning groups of students for online classes, which may foster better collaboration between the participating students by increasing their level of interaction with like-minded and diverse population.

As future work, we plan to conduct tests on an actual MOOC platform to get a real-time assessment of the quality of student groups formed based on the proposed algorithm. The algorithm can also be improved to add more attributes that could potentially increase the chances of forming better quality groups. These attributes could be derived based on the past courses that the students had registered for, or in some form of a feedback from students themselves based on a certain questionnaire. Case studies reveal that the number of participating users in MOOCs is increasing every year, hence it becomes quite challenging to establish the same kind of communication that exists within a classroom. However, by using hybrid PSO to generate dynamic learning groups, we believe we can bridge that gap to some extent.

REFERENCES

Bratton, D. (2007). Defining a standard for particle swarm optimization. *Swarm Intelligence Symposium*, (pp. 120-127).

Chen, W. N. (2010). A novel set-based particle swarm optimization method for discrete optimization problems. *IEEE Transactions on Evolutionary Computation*, (pp. 278-300).

Deibel, K. (2005). Team formation methods for increasing interaction during in-class group work. *ACM SIGCSE Bulletin*, (pp. 291-295).

Freeman, S. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, (pp. 8410-8415).

Hartigan, J. A. (1979). A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 100-108.

HarvardX-MITx. (2014). *HarvardX-MITx Person-Course Academic Year 2013 De-Identified dataset.* Retrieved from MITx and HarvardX Dataverse: https://dataverse.harvard.edu/dataset.xhtml?persistentId=10.7910/DVN/26147

Jarboui, B. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, (pp. 299-308).

Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760-766). Springer (US).

Lou, Y. (2008). Within-class grouping: A meta-analysis. *Review of educational research*, (pp. 423-458).

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, (pp. 39-41).

Onah, D. (2014). Dropout rates of massive open online courses: behavioural patterns. *EDULEARN14 Proceedings*, (pp. 5825-5834).

Shah, D. (2015). *By The Numbers: MOOCS in 2015?* Retrieved from Class Central: https://www.class-central.com/report/moocs-2015-stats/

Shimazoe, J. (2010). Group work can be gratifying: Understanding & overcoming resistance to cooperative learning., (pp. 52-57).

Slavin, R. E. (1987). Developmental and motivational perspectives on cooperative learning: A reconciliation. *Child development*, pp. 1161-1167.

Van Den Bergh, F. (2007). *An analysis of particle swarm optimizers (Doctoral dissertation, University of Pretoria).*

Wu, Z. (1994). Verbs semantics and lexical selection. *In Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, (pp. 133-138).

Yin, P. Y. (2006). A particle swarm optimization approach to the nonlinear resource allocation problem. *Applied mathematics and computation*, (pp. 232-242).

Zepke, N. (2010). Improving student engagement: Ten proposals for action. *Active learning in higher education*, (pp. 167-177).