LPR: An Adaptive Learning Path Recommendation System using ACO and Meaningful Learning Theory

by

Mehdi Niknam

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
May 2017

Thesis advisor | Author

Dr. Parimala Thulasiraman | **Mehdi Niknam**

LPR: An Adaptive Learning Path Recommendation System using ACO and Meaningful Learning Theory

Abstract

In recent years, the educational community has been interested in enabled learning systems. That is, having a personalized learning system that can adapt itself while providing learning support to different learners to overcome the weakness of 'one size fits all' approaches in technology-enabled learning systems. In this thesis, we address one known problem in adaptive learning systems called *curriculum sequencing*. We design and implement a learning path recommendation (LPR) system that selects an appropriate learning path for learners based on their characteristics and needs.

There are two components to the LPR system: *searching* for the learning paths and *clustering* the learners into groups based on their prior knowledge. Using bio-inspired ant colony optimization (ACO) algorithm and meaningful learning theory of Ausubel, the ACO path finder component searches for a suitable learning path for the learner. This component incorporates continuous learner's improvement in the process of a learning path selection. The LPR system, uses the pre-assessment/familiar degree calculator to gauge learner's prior knowledge and produces a learner's familiar degree of concepts. The clustering component uses Fuzzy C-Mean (FCM) algorithm. The LPR system can recommend more than one learning path to learners located on the cluster boundaries. We implement an interface to provide the recommendation to the learner.

We evaluate the effectiveness of the LPR system by designing and developing a database course and ask actual learners to complete the course. The results of our experiment show that the group that used the LPR system have higher performance and knowledge improvement in the course than the control group. The performance and knowledge improvement differences between the two groups are statistically significant. Based on the statistical tests, the LPR system has a moderate to large impact on the learners' performance and knowledge improvement. Although the course completion time for the LPR group was slightly less than the control group, no statistically significant difference is found between the time completion of both groups.

Contents

List of Figures

List of Tables

Acknowledgments

This research work would not have been possible without the support of many people. I wish to express my gratitude to my supervisor, Dr. Parimala Thulasiraman who was abundantly helpful and offered invaluable assistance, support and guidance. My gratitude also extends to the members of the advisory committee, Dr. Kathleen Matheos and Dr. Pourang Irani who helped me with their knowledge and assistance to complete this thesis successfully. Special thanks also goes to all my fellow graduate friends who helped and supported me through discussion and advice. I would also like to convey thanks to my colleagues at the Center for the Advancement of Teaching and Learning at the University of Manitoba for providing support and encouragement. My appreciation extends to the Natural Sciences and Engineering Research Council of Canada (NSERC) for providing some financial means to conduct my research. I would like to thank my wife Jennifer for her enormous support and assistance through this thesis. I wish to express my love and gratitude to my parents and brothers for their understanding and endless love through the duration of my studies.

This thesis is dedicated to my wife and my parents.

Chapter 1

Introduction

The educational community has always been interested in having a true personalized learning system. Personalized learning refers to adjusting the pedagogy, curriculum, and learning environment for learners to satisfy their learning needs and preferences. A personalized learning system can adapt itself when providing learning support to different learners to overcome the weakness of 'one size fits all' approaches in technology-enabled learning systems. The goal is to have a learning system that can dynamically adapt itself based on a learner's characteristics and needs and provide personalized learning support.

1.1 Research Problem

One known problem in adaptive learning systems is content planning. Content or curriculum planning (also known as content or curriculum sequencing) refers to the process of selecting appropriate learning objects for a learner or guiding the learner to an appropriate learning path (Wasson, 1990). According to Wiley (Wiley, 2003), "a learning object is any digital resource that can be reused to support learning". In other words, a learning object is a digital content, practice or assessment item designed based on a specific learning objective. Learning objectives are clear statements that describe the competences that students should possess upon completion of a learning activity (Simon and Taylor, 2009). The goal in content planning is to recommend a suitable learning path (a sequence of learning objects) to learners to achieve specific learning objectives.

Content planning problem is modeled as a graph problem where the learning objects are the nodes and edges are the possible paths between them. The possible paths are specified by the content expert and data induced from chosen paths by previous learners. Ideally, an algorithm would search in a set of content nodes (learning objects) and find the most suitable learning path (a sequence of learning objects) for a learner.

Content planning problem is a non-deterministic polynomial hard (NP-Hard) problem (Acampora et al., 2011); therefore, in the last four decades there has been interest in using soft computing methods including neural networks, fuzzy systems and genetic algorithms to analyze such large-scale data. According to Wong and Looi (Wong and Looi, 2012), swarm intelligent (or bio-inspired) techniques have been used in content/curriculum planning, computer adaptive testing, intelligent assessment paper generation and class/examination scheduling areas in the education field.

In this thesis, we design, develop and implement a learning path recommendation (LPR) system. This system is based on Ausubel's learning theory (meaningful theory) and ant colony optimization (Dorigo et al., 2006) algorithm. The LPR system incorporates four components: (i) pre-assessment/familiar degree calculator - this is done when a learner enters the LPR system. It calculates the familiar degree of concepts for the learner; (ii) clustering - based on the results of the pre-assessment test and familiar degrees, the cluster algorithm assigns the learner to a cluster. We use Fuzzy C-Mean clustering algorithm; (iii) learning path finder - using the familiar degrees of all ex-learners in the same cluster, the learning path is designed. We use ACO algorithm; (iv) recommend the learning path to the learner.

The remaining of this thesis is organized as follows. The chapter 2 contains the review of literature regarding content planning problem, educational theories, and clustering analysis. The chapter 3 includes detailed explanation of different components of the learning path recommendation (LPR) system. In chapter 4, the implementation details of each component of the LPR system and the details of course design and development is presented. The chapter 5 contains the experimental design and evaluation of the LPR system followed by a conclusion and future work in chapter 6.

1.2 Contributions

The contributions of this research are as follows:

- We design and develop a new ACO algorithm for content sequencing based on meaningful learning theory. The algorithm incorporates continuous learners improvement in the process of a learning path selection.

- To overcome the weakness of K-Means clustering, we have adapted Fuzzy C-Mean clustering algorithm to recommend more than one learning path for learners located on clusters boundaries.

- We evaluate the effectiveness of the learning path recommendation algorithm through an experiment with actual learners in a real course using University of Manitoba's UM Learn system.

Chapter 2

Background

As mentioned content planning problem is an NP-Hard problem. In the last four decades various methods have been used to solve this problem including rule-based intelligent methods, data mining and soft computing methods. In this section, we have provided a review of different methods utilized to solve the content planning problem. Also, we have provided a review of learning theories that could be utilized to improve the process of content planning. A brief section is included regarding clustering analysis as well.

2.1 Non-heuristic Methods

Non-heuristic methods such as prescriptive rule-based methods and data mining methods have been employed to solve the content planning problem are reviewed in this section.

2.1.1 Prescriptive rule-based Intelligent Tutoring Systems (ITS)

As mentioned, content or curriculum planning (also known as content or curriculum sequencing) refers to the process of selecting appropriate learning objects for a learner or guiding the learner to an appropriate learning path. The goal in content planning is to recommend a suitable learning path (a sequence of learning objects) to learners to achieve specific learning objectives.

Ideally, an algorithm would search in a set of content nodes (learning objects) and find the most suitable learning path (a sequence of learning objects) for a student. In early Intelligent Tutoring systems, content planning was performed by searching for the suitable learning paths on a network of content nodes based on 'prescriptive rules' defined by content experts (Wong and Looi, 2012). A content network includes a set of interconnected learning objects. The arc that connects two given learning objects represents the relationship between the two learning objects. For example in Figure 2.1, learning object A is the 'prerequisite knowledge' of B which means any student should be presented with learning object A before B. Examples of prescriptive rule-based ITS are discussed in (Barr and Alkinson, 1976), (Arshard, 1989) and (Peachey and McCalla, 1986). Since prescriptive rules are designed based on commonsense or experts' ideas, the rules are rigid and not valid for every learner.

Wong and Looi (Wong and Looi, 2012) state that prescriptive rules are not supporting the concept of self-directed learning (where learners have a degree of choice in their instructional situation) and the concept of indirect social navigation (which suggests navigation is a social process). Therefore, in order to perform more accurate content planning, there is a need to consider ex-learners' paths while planning content

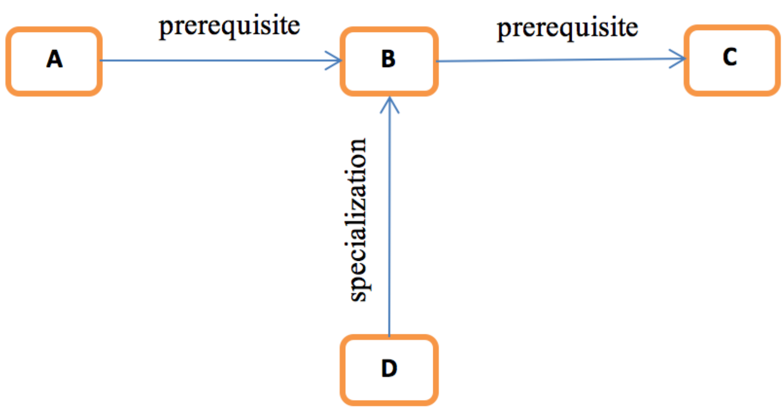

Figure 2.1: Example of content network

for new learners.

2.1.2 Data Mining Methods

With advancement in e-learning systems, it is convenient to collect navigation logs, activity logs and performance results for learners. Data mining techniques can help to select the learning path for a new learner based on the ex-learners. The most traveled path would be selected for the new learner.

Some examples of systems using data mining techniques are Hidden Markov Model-based Tutorlet Agents (Niemczyk, 2000), Time Spent Reading algorithm-based Knowledge Sea (Farzan and Brusilovsky, 2005) and FP-Tree-based cross-level frequent pattern mining for web-based instruction (Witten et al., 2016). These systems statistically induce the learning paths that most of the previous learners have taken and then recommend it to the present learner.

However, Tang and McCalla (Tang and McCalla, 2005) argue that the most traveled learning path induced from data mining may not be suitable for a new learner. Two other features should be considered while selecting the learning path: 1) ex-learners' characteristics; 2) ex-learners' performances after completing the path. Some of the data mining methods incorporated the ex-learners' characteristics using clustering. However, in order to incorporate ex-learners' performance there is a need to use soft computing and optimization algorithms (Wong and Looi, 2012).

2.2 Heuristic (Swarm Intelligent) Methods

Soft computing methods are found to be especially good in solving non-deterministic polynomial time-hard (NP-hard) problems (Zhang and Lu, 2007). Al-Muhaideb and Menai (Al-Muhaideb and Menai, 2011) categorize the content planning/sequencing methods into two distinct categories:

1. Methods that incorporate experiences of other similar learners in the process of recommendation (known as social sequencing).

2. Methods that only focus on individual learner experience in the process of recommendation (known as individual sequencing).

To solve individual sequencing, researchers have been using neural networks, genetic algorithms, etc. For social sequencing, researchers are only using swarm intelligence methods. Since this research is focused on social sequencing, only the swarm intelligence methods have been included under the heuristic methods in the literature review. We have reviewed the related work on the content planning problem using swarm intelligence techniques. We started with algorithms using Particle Swarm

Optimization (PSO) techniques and focused more on algorithms using Ant Colony Optimization (ACO).

2.2.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is inspired from social behavior models that exist in bird flocks and fish schools. In PSO, a particle refers to a bird or fish and represents a candidate solution. Randomly initialized particles fly in the solution space and cooperate to find a near optimal solution determined by a fitness function. A particle keeps track of the best solution it has achieved so far and the global best solution. A particle changes its velocity and position according to mentioned parameters to evolve a new generation that is closer to the optimal solution (Eberhart and Kennedy, 1995).

Algorithm 1 Procedure PSO

```
procedure PSO
    Initialize population in hyperspace
    repeat
        Evaluate fitness of individual particles
        Modify velocities based on previous best and global best
    until Termination condition
end procedure
```

Chang et al. (Chang et al., 2008) employed PSO in PCCPSO (Personalized Course Composition approach based on PSO) algorithm that composes suitable learning materials to generate a course. Huang et al. (Huang et al., 2008) employed PSO in

SBACPSO (Serial Blog Article Composition PSO) algorithm that selects topic-related blog articles written by instructors, course designers and the learners as learning references. The learning materials or blog articles retrieved and stored by both techniques must be tagged with their specific attributes such as difficulty levels and related learning points.

Both algorithms work in a similar fashion. At first, the system generates an initial set of particles; each represents a combination of learning materials or blog articles. Each particle makes use of a set of binary numbers to represent the selection status of individual materials or articles. For example, if there are five materials/articles called a, b, c, d, e stored in the database, then particle 11001 represents the combination of a, b, e being selected and c, d are not selected.

After initialization, the particle swarm starts to 'fly' (swarm) within the solution space. The algorithms utilize the fitness functions to evaluate the 'goodness' of each particle in order to adjust the 'swarming velocities' of the particles (i.e., to strike a balance between maintaining the same velocity with neighboring particles and flying towards the particle with the best solution so far). After each particle reaches its new location, it will adjust the solution that it carries according to the new coordinate and produces a new generation of particles. The particle swarm continues to evolve until the best particle reaches an expected fitness value or the computation process satisfies a stop condition.

Both algorithms ignore what the ex-learners have chosen in the process of material selection which ignores the concept of social navigation. Also, both algorithms compose materials or blog articles but do not provide any order for them.

Chu et al. (Chu et al., 2011) also employed PSO in PC2PSO algorithm to choose the material suitable for a review course based on the relevance degree, difficulty level and the number of available learning resources. A greedy-like algorithm is used to sequence this material and the reading order. The selection is made from an automatically created concept semantic map. Chu et al. claim that PC2PSO provides a truly personalized learning environment when used in conjunction with an Intelligent Tutoring System (ITS).

Govindarajan et al. (Govindarajan et al., 2016) recently employed the Parallel Particle Swarm Optimization (PPSO) method to predict a dynamic learning path for learners based on competence and meta-competence values observed in a learning environment. Their system is able to automatically configure itself to calculate an optimal learning pathway for learners. They have provided an evaluation of their system on an experiment within a Java Programming course.

2.2.2 Ant Colony Optimization (ACO)

The most popular swarm intelligent algorithm used in content planning is Ant Colony Optimization (ACO). ACO (Dorigo et al., 2006) algorithm is inspired from social creatures like ants and bees. Ants are able to construct networks of paths to link their nests to food sources. These networks form minimum spanning trees that minimize the energy ants spend to transfer food to the nest. Ants would find optimal paths from their colony to the food sources by releasing special chemicals called pheromones that guide them to either food or the nest (Dorigo and Gambardella, 1997).

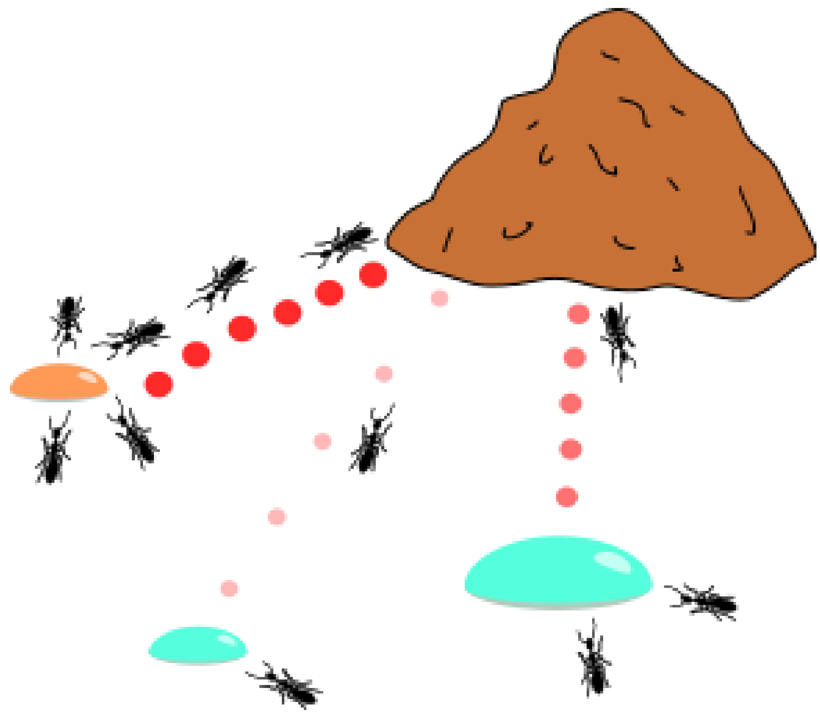

Figure 2.2: Demonstration of Ant Behavior by (Dake, 2006)

The initial path will not be straight, but the more ants traveling through a path, the more pheromone is laid in the path and the more this path attracts other ants to follow. As a result, paths would merge into a straighter path. The pheromone also evaporates as time goes therefore expired paths will be forgotten (Dorigo and Gambardella, 1997). Algorithm 2 indicates the general ACO pseudo algorithm and Figure 2.2 demonstrate ants behavior while moving from the nest to food source.

According to Wong and Looi (Wong and Looi, 2012), ACO-based content planning techniques treat ex-learners as ant-like agents. Each of the agents traverses the learning paths that its corresponding ex-learner has traveled in the graph (where nodes represent the learning objects and possible pedagogical sequence are modeled

Algorithm 2 Procedure ACO

```
procedure ACO
    Initialize parameters and pheromone trails
    repeat
        Construct Solution
        Apply local search
        Update pheromone trails
    until Termination condition
end procedure
```

as arcs among the nodes) and deposits pheromones (as a weight on arcs) on the way. The amount of the deposited pheromones varies based on each technique (e.g., either a constant or calculated based on learners' performances). Two major processes in ACO-based techniques are:

1. Computation of the pheromones in the entire content network;
2. Recommendation of a learning path to the new learner.

Wang et al. (Wang et al., 2008), proposed the style-based ant colony system (SACS) that simply categorizes ex-learners into four learning styles according to VARK model (visual, audio, reading/writing or kinesthetic). In this algorithm, when recommending the next node, the pheromones deposited by the ex-learner who falls into the same category with the current learner are favored in the computations. Also, the pheromone computation is only based on the number of times they have traveled not based on ex-learners' performances.

Yang and Wu (Yang and Wu, 2009) presented AACS (Attribute-based Ant Colony System) algorithm which is more complex than SACS. AACS requires course designers to tag each learning object with two attributes learning object type (graphs, videos, text or XML) and learning object level (initial, introductory, advance, professional). Moreover, the instructors need to tag each learner with two attributes learning style (corresponding to object type) and knowledge level (corresponding to object level). The amount of the pheromones deposited along the path by ants depend on the similarity level between the attributes of the learner that the ant represents and the attributes of the learning objects that the ant is visiting.

Wong and Looi (Wong and Looi, 2009) developed DYnamic Learning Path Advisor (DYLPA) algorithm which is another ACO-based content planning algorithm. DYLPA combines 'prescriptive rules' and ACO-based 'social navigation' methods to recommend personalized learning paths. In DYLPA the instructor is required to specify a 'DYLPA training size' of N. The learning path recommendation relies merely on the prescriptive rules at the early stage after a DYLPA-based e-learning system is launched and before the number of ex-learners hits N. This method prevents inaccurate recommendation by the ACO algorithm due to a low training size. After the number of ex-learners grows beyond N, the ACO-based algorithm is triggered. DYLPA selects up to N ex-learners with high similarity levels (based on seven learning preferences) with the target learner, and computes and deposits the pheromones according to their traveled paths and performances.

SACS, AACS, and DYLPA consider the pheromone evaporation process in their design in contrast to the earlier ACO-based algorithm. In other words, the pheromones are time decaying to avoid inaccurate recommendation (due to possibilities of getting stuck in local optima). However, only DYLPA uses the ex-learners' learning path and learners' performance to compute the pheromones.

More recently, Kurilovas et al. (Kurilovas et al., 2014) and (Kurilovas et al., 2015) presented yet another ACO-based algorithm which focuses on the issue of the static nature of previous algorithms. Kurilovas et al. argue that the course graph could be modified (i.e., removing a learning object, or adding new learning object to the course graph) and the algorithm should dynamically adapt to the modification and be able to reorganize itself for the path recommendation based on the new topology.

A new material pheromone was introduced to attract ants to select the new learning object added to the course graph. The authors indicated that the new material pheromone should evaporate faster than the regular pheromones and it should be aligned with learning styles (since they used learning styles to categorize similar learners).

Kardan et al. (Kardan et al., 2014) present a two-stage approach to construct an adaptive learning path called ACO-Map. In the first stage of this method, they utilize K-means clustering algorithm to divide learners into groups of learners who have a similar familiarity to the course concepts. Then, ant colony optimization is applied to construct a learning path for each group as their guidance tool for adaptive learning path in second stage.

In order to achieve learner's familiar degree with concepts, learners will be tested before entering the system. Based on the relevance of the questions in the test with each concept, the familiar degree of the learners are calculated and used to cluster them and to calculate the pheromones in the ACO algorithm.

Kardan et al. (Kardan et al., 2014) refer to Ausubel's learning theory (meaningful learning theory) that indicates the learning process occurs meaningfully when new concepts are linked to existent ones in the conceptual structure of the learner. Therefore, a personalized learning path would be recommended to the learner based on the learner's familiar degree to the concepts. Although this algorithm adheres to meaningful learning theory, it is static since it ignores learners' continuous improvement.

The familiar degree only calculated after the initial test and the learners' performance during the course is not included in the pheromone calculation and learning path selection process. Another issue with this algorithm is the method used for clustering. K-mean clustering method ignores complete personalization since each learner is aligned to the cluster centroid (i.e., the average learning familiarity of a group).

In summary, a good social sequencing algorithm must include the following features:

1. Use a feature or group of features to find similar ex-learners (prior knowledge, learning style, learning preference, etc.)

2. Incorporate learner's performance and continuous improvement in the process of learning path selection

3. Adapt to changes applied in the course graph dynamically

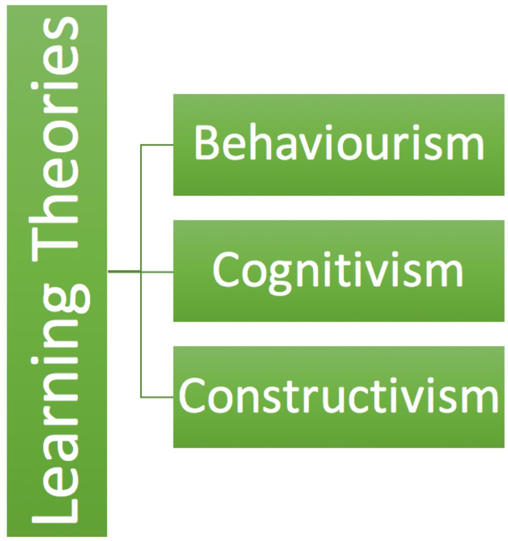

Figure 2.3: Learning theories behaviourism, cognitivism, and constructivism

2.3 Theories of Learning

This section provides a brief overview of three learning theories: behaviourism, cognitivism, and constructivism. Also, there is a discussion on how these theories could be utilized in the concepts of adaptive learning and content planning problem.

2.3.1 Behaviourism Theory

Behaviourism regards learning as a change in observable behaviour caused by external stimuli (Ally, 2008). As such, learning occurs when a proper response happens after a specific environmental stimulus is presented. Three basic assumptions of behavioural theory are objectivism, environmentalism and reinforcement. Objectivism refers to the ability to analyze behaviour as a result of external events. Environmentalism emphasizes the role the environment has in shaping behaviour. Reinforcement relates to the affect feedback and consequences have on subsequent behaviour. Key behavioural theorists with an impact on learning theories are Thorndike, Pavlov, and Guthrie (Schunk, 2012).

2.3.2 Cognitivism Theory

Cognitive theorists recognize learning as an internal process dependent upon "the processing capacity of the learner, the depth of the processing and the learner's existing knowledge structure" (Ally, 2008). The focus of cognitivism is on promoting complex cognitive processes which include thinking, problem solving, and concept formation. In addition, cognitive theories stress the importance of how information is received, organized, stored, and retrieved. Driscoll (Driscoll, 2005) outlines the transfer of information between three stages of the memory system: sensory, short-term memory, and long-term memory. The cognitive theorist Ausubel proposed the use of different instructional strategies such as advance organizers (i.e., concept maps) and the importance of meaningfulness to facilitate information processing (Ally, 2008).

2.3.3 Constructivism Theory

Constructivist learning theory regards learners as active participants rather than passive in the learning process. Learners are at the centre of the learning process and play an important role in determining their learning. The instructor acts as a facilitator to guide learners through the discovery of knowledge and encourage selfreflection. As described by Cunningham and Duffy (Cunningham and Duffy, 1996), "learners should be allowed to construct knowledge rather than being given knowledge through instruction". The intent is for learners to contextualize information through authentic and interactive learning activities. Social interaction and collaboration with peers and teachers facilitates learning. Theorists associated with constructivist theory include Piaget and Vygotsky (Ally, 2008).

2.3.4 Personalized Learning and Learning Theories

Reviewing the learning theories, the cognitivism theory is the most relevant theory to the concept of personalized learning since it recognizes the learner's processing capacity and prior knowledge as a characteristic of the learner. Also, the content planning according to this theory would improve learning performance since this theory values the importance of the way information is received and organized. Moreover, the level of cognitive complexity for a learning object could be taken into account in the process of content planning.

Meaningful learning theory is one subset of cognitivism proposed by Ausubel (Schunk, 2012). According to Ausubel (Schunk, 2012) learning is meaningful when new information connects to related concepts in a learner's long-term memory. In other words, the new information expands, modifies or elaborates on existing information stored the learner's memory. Meaningfulness can be influenced by factors such as prior experiences, age, socio-economic status, and educational background.

Concept maps were proposed by Joseph D. Novak (Novak and Cañas, 2008) based on the meaningful learning theory. A concept map is a tool to organize and represent knowledge. A concept map includes concepts in the forms of nodes and connecting lines between them as relationships. The concept map can provide the order and sequence of learning concepts for learners. In an effective and meaningful learning process, concepts should be learned in a proper order or sequence. Therefore, the constructed concept maps based on the learners' knowledge includes more organized and adapted relationships between concepts and presents the proper learning order of concepts. As a result, researchers focused on automatically constructing concept map to generate adaptive learning paths for learners. In this research, we also generated personalized and adaptive learning paths for learners by automatically constructing concept maps.

2.4 Cluster Analysis

In this section, we provide a definition and a categorization for clustering algorithms. We review the applications of the clustering algorithms used in the field of education. We also provide a comparison between K-Mean and Fuzzy C-Mean clustering algorithms.

2.4.1 Definition

Cluster analysis or clustering refers to a collection of unsupervised classification methods to group a set of objects into subsets of objects called clusters (Cebeci and Yildiz, 2015). The goal of clustering methods is to form clusters with objects that are more similar to each other than the objects in other clusters. Similarity among objects are based on the shared common characteristics of the objects. Data clustering plays an important role in a variety of disciplines ranging from sociology and psychology to commerce, biology, computer science, and education (Gao, 2016).

2.4.2 Categorization

Fahad et al. (Fahad et al., 2014) provide the following categorization for clustering algorithms. Partitioning-based clustering algorithms divide the data into clusters (partitions) where each cluster must contain at least one object. Hierarchal-based clustering algorithms organize the data hierarchically based on the medium proximity. It can start from one cluster and divide the cluster to more clusters gradually or start with a number of clusters and merge the clusters gradually. Density-based clustering algorithms separate the data based on their region of density, connectivity, and boundary. A cluster can grow in any direction that data density would lead to. Grid-based clustering algorithms divide the space of data objects into grids and then perform the clustering based on the grids' statistical values. Model-based clustering algorithms perform the clustering by optimizing the fit between the data and some predefined mathematical model.

As indicated in the literature, no clustering can be applied to all applications since each has some drawbacks. Researchers also looked at soft computing methods to overcome the drawbacks of traditional methods (Chandrasekhar and Naga, 2011). Specifically, Gao (Gao, 2016) presented an improved Ant Colony Clustering algorithm. Tan et al. (Tan et al., 2011) propose simplifications for the Ant-based clustering algorithm and argue that reducing the complexity of ant-based clustering will speed up the clustering process without reducing the clustering quality.

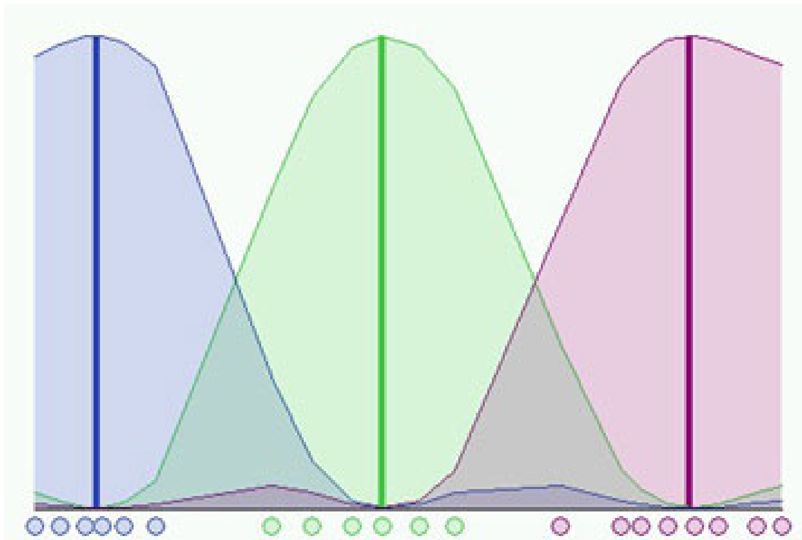

Figure 2.4: Demonstration of Fuzzy C-Mean (FCM) clustering by (fcm, 2017)

2.4.3 Applications of Clustering in Education

Dutt et al. (Dutt et al., 2015) and Priyadarshini (Priyadarshini, 2017) reviewed clustering algorithms used in the education sector. The applications of clustering algorithms range from classifying learners based on their learning styles, predicting learners' performance, identifying significant variables impacting learners' performance, developing a learner profile based on activities, and profiling instructors. As indicated in section 2.2, one major application of clustering is to identify similar ex-learners based on some features (e.g., prior knowledge, learning style, etc.). K-Mean clustering algorithm is used the most in educational applications. Specifically, Kardan et al. (Kardan et al., 2014) used K-Mean clustering in their learning path recommendation algorithm to group similar learners.

2.4.4 K-Mean and Fuzzy C-Mean

K-Mean clustering is robust and easy to implement with efficient time complexity. Despite these advantages, one major drawback of the K-Mean clustering algorithm is that it may not provide accurate clustering for the data objects located on boundaries of clusters since each data object only belongs to one cluster. Specifically, in the application of finding similar learners based on their prior knowledge (similar to the work in Kardan et al (Kardan et al., 2014)), learners located on the boundary of clusters are not clustered accurately since the distance to two clusters centroid are the same and K-Mean arbitrary would assign the learner to one of those clusters.

In order to resolve the issue of inaccurate clustering for the learners located on clusters boundaries and maintaining the aforementioned advantages of the K-Mean clustering, we decided to use the Fuzzy C-Mean (FCM) clustering algorithm. In FCM clustering algorithm, each data object can be a member of multiple clusters with a different degree of membership (a number between 0 and 1). With FCM, the data objects located on the clusters' boundaries will not be forced to belong to a specific cluster and can belong to multiple clusters with various degrees of membership. In addition, Fahad et al. (Fahad et al., 2014) indicated that FCM clustering algorithm shows excellent performance with respect to the quality of the clustering outputs.

Chapter 3

Learning Path Recommendation (LPR) System

3.1 Introduction

A Learning Path Recommendation (LPR) system is designed and implemented based on meaningful learning theory and ant colony optimization (ACO) algorithm as part of this study. Figure 3.1 depicts the diagram of the LPR system indicating its different components.

As can be seen from the recommendation system, the first step for the learner in entering the LPR system is to take a pre-assessment test. At this point, the system calculates the familiar degree of concepts for the learner based on the results of the pre-assessment test. Next, the system uses the FCM clustering algorithm to assign the learner to a cluster (or clusters). Then the system uses the familiar degrees of all ex-learners in the same cluster as the input to the ACO learning path finder. The

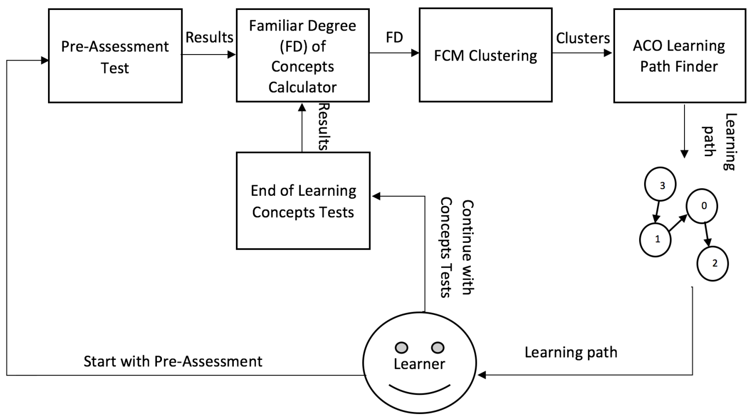

Figure 3.1: Diagram of Learning Path Recommendation (LPR) System

familiar degrees based on the pre-assessment and final exam of the ex-learners are used in the ACO learning path finder.

The ACO algorithm then finds the best learning path for the cluster that the learner is assigned to and recommends it to the learner. The learner will then complete the course activities of the first learning concept in the recommended learning path and will complete a short concept test. In the next step, the learner will request a learning path again. The learner's familiar degree of concepts might change based on the result of the concept test (since the questions in each concept test can be relevant to other learning concepts) that the learner completed at the end of the recent learning concept.

The same process is followed by the learner and the system to recommend a new learning path to the learner for the remaining learning concepts until all the learning concepts are covered. We will explain each component of the LPR system in detail after defining relevant terminologies.

3.2 Assumptions and Terminologies

Before defining the familiar degree of concepts used in our system to cluster learners and provide the learning path recommendation, we would need to define some terminologies used in this study.

3.2.1 Course

A course is a unit of instruction in one subject area. For this study, the course is "Introduction to Databases".

3.2.2 Course Goals

Course goals are broad, general statements of what we want our learners to learn by the end of a course. Course goals specify the big picture or general direction or purpose of the course. An example of a course goal in the "Introduction to Databases" course is "Use SQL to fetch data out of databases".

3.2.3 Course Unit

The topics to be covered in a course are organized into multiple units. Each course unit represents one major topic to be covered in the course. In addition, each course unit includes multiple learning concepts that are relevant to the unit. An example of a course unit in the "Introduction to Databases" course is "SQL select query".

3.2.4 Unit Learning Objectives

Learning objectives are derived from and need to align to course goals. They are specific, measurable statements of what we want our learners to learn by the end of a unit. Learning objectives are often written following Bloom's Taxonomy and are classified into three domains: 1) cognitive; 2) psycho-motor; 3) affective. In this study the unit learning objectives are written in a cognitive domain. As an example, for the course goal "Use SQL to fetch data out of databases" the unit learning objectives are:

- Identify elements in SQL select statement
- Identify SQL data types and comparison operators
- Identify SQL select clauses
- Write basic SQL select statement
- Write SQL select queries with aggregations and Joins
- Write SQL query with self joins
- Write a SQL query with sub-queries

3.2.5 Learning Concept

A learning concept in this study is defined as the learning materials and activities that a learner needs to complete in order to achieve a unit learning objective in the course. As an example, the learning concept "Foreign Keys" in the "Introduction to Databases" course includes a short video defining the concept of foreign keys in a database and an activity to identify foreign keys in database tables. The learning objective for this concept in the course is "Identify foreign keys in tables".

3.2.6 Assessment

Assessment is organized into two categories: 1) formative; 2) summative. Formative assessment is informal assessment methods provided throughout a course in order for learners to improve their progress in the course. In the "Introduction to Databases" course, the formative assessment is: 1) the pre-assessment test at the onset of the course to determine learners' prior knowledge; 2) the learning concept tests after completing each learning concept.

Summative assessment is formal assessment methods used to evaluate whether or not the learners achieved the unit learning objectives. In the "Introduction to Databases" course, learners completed the final test at the end of the course as a form of summative assessment to assess their knowledge once they have reviewed the course units and learning concepts.

3.3 Familiar Degrees of Concept Calculator

As mentioned earlier, concept map is a tool used to organize and represent knowledge. Concepts are shown as nodes and the relationship between them are shown as lines connecting the nodes. The main idea of a concept map is based on the meaningful learning theory of Ausubel indicating that the learning process occurs when new concepts are linked to existent ones in the conceptual structure of learners.

In a meaningful learning process, the learning of the concepts must be done in an appropriate sequence. Constructed concept maps based on the learners' prior knowledge include the relationships between concepts more organized and adapted to the learner. Therefore, researchers (Kardan et al., 2014), (Hung and Hung, 2009), (Lee et al., 2009) have focused on using automatically constructed concept maps as a useful guiding tool for learners as an adapted learning path.

As a result, in this thesis we use the automatically constructed concept map as an adapted learning path for learners. In the LPR system, learners' prior knowledge is gauged based on the result of a pre-assessment and a concept map is generated based on the learners' and similar ex-learners' (learners who completed the course previously) prior knowledge. The concept map is then recommended to the learner as a personalized learning path.

Assuming that each course includes C learning concepts and there are N learning concepts in each test item ($1 \leq N \leq C$), the relevance degree between learning concept C_j and test item Q_i is shown by $R_{Q_iC_j}$. The greater the value of $R_{Q_iC_j}$ indicates greater relevance between the item Q_i and learning concept C_j. When test item Q_i only assesses the learning concept C_j, the value of $R_{Q_iC_j}$ is represented by 1. In contrast, when test item Q_i does not assess the learning concept C_j, the value of $R_{Q_iC_j}$ is represented by 0. As a result, $0 \leq R_{Q_iC_j} \leq 1$. Also, assuming S learners take Q test items, $P_{S_iQ_j}$ denotes the grade of learner S_i in test item Q_i which grades are between 0 to P ($0 \leq P_{S_iQ_j} \leq P$). Kardan et al (Kardan et al., 2014) state that to estimate the learner's familiar degree of the learning concept, grade $P_{S_iQ_j}$ and relevance degree $R_{Q_iC_j}$ need to be considered together. $A_{S_iC_j}$ denotes the familiar

Table 3.1: Relevance degree (R) of Questions (Q) for Concept 2

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
R	0	0.2	1	0.1	0.1	0.1	0	0	0	0.2	0	0	0	0	0.2	0	0	0	0	0

Table 3.2: Learner 1's Grades (P) for Questions (Q)

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
P	0	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	1

degree of learner S_i of concept C_j where:

$$A_{S_iC_j} = \frac{\frac{1}{P}\sum_{k=0}^{Q-1} P_{S_i,Q_k} R_{Q_k,C_j}}{\sum_{k=0}^{Q-1} R_{Q_k,C_j}} \tag{3.1}$$

In the course developed for the prototype in this study we have 20 learning concepts, the pre-assessment test includes 20 questions as well. Each question is graded out of 1 ($P = 1$). The relevance degree of questions for concept 2 is indicated in table 3.1. As an example, for a learner with the grades in the pre-assessment indicated in table 3.2, the familiar degree of concept 2 is calculated as follows:

$$A_{S_1C_2} = \frac{\frac{1}{1}[(0\times0)+(1\times0.2)+(1\times1)+(1\times0.1)+(0\times0.1)+(1\times0.1)+(0\times0)+(1\times0)+(0\times0)+(0\times0.2)+]}{0+0.2+1+0.1+0.1+0.1+0+0+0+0.2}$$
$$\frac{\frac{1}{1}[(0\times0)+(0\times0)+(0\times0)+(0\times0)+(1\times0.2)+(0\times0)+(0\times0)+(0\times0)+(1\times0)+(1\times0)]}{0+0+0+0+0.2+0+0+0+0+0} = 0.84$$

Now that we have defined the learner familiar degree of a learning concept, we can move on to the clustering component of the LPR system that utilizes the familiar degree of concepts to assign the learner to a cluster or clusters.

3.4 Fuzzy C-Mean (FCM) Clustering

As indicated before, we adapted Fuzzy C-Mean (FCM) clustering algorithm to cluster similar learners. Lets look at the details of FCM and how learners have been clustered based on the familiar degree of concepts.

The FCM algorithm is a ”soft” computing clustering technique that assigns the data objects to the clusters with a degree of membership. That means, a data object can belong to more than one cluster with different degrees of membership. FCM would try to find each cluster's representative point called a prototype or centroid. Then, FCM calculates the membership degree of each data object for all clusters.

FCM algorithm iteratively searches the cluster prototypes and updates the memberships of data objects. The membership value is a value ranging from 0 to 1 showing the likelihood of a data object belonging to that cluster. The higher the membership value, the more likely the data object belongs to that cluster. The FCM clustering minimizes the objective function J in Equation 3.2:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} U_{ij}^{m} D_{ij}^{2} \tag{3.2}$$

where n is the number of data objects, c is the number of predefined clusters, U_{ij} is the membership value of data object i for cluster j, m is a fuzzifier factor ($m > 1$), D_{ij} is the Euclidean distance between i-th data object p_i and j-th cluster prototype v_j calculated based on the Equation 3.3:

$$D_{ij} = \sqrt{\sum_{i=1}^{n} (p_i - v_j)} \tag{3.3}$$

Cluster prototype v_j is updated based on the Equation 3.4:

$$v_j = \frac{\sum_{i=1}^{n} U_{ij}^{m} p_i}{\sum_{i=1}^{n} U_{ij}^{m}} \tag{3.4}$$

The fuzzy membership values are calculated based on the Equation 3.5:

$$U_{ij} = \frac{1}{\sum_{k=1}^{n} \left(\frac{|p_i - v_j|}{|p_i - v_k|} \right)} \tag{3.5}$$

As mentioned, FCM is an iterative process and will stop when either the number of iterations is reached to the maximum predefined number of iterations, or when the difference between two consecutive values of objective function is less than a predefined convergence value of ϵ. See the pseudo code for FCM algorithm 3.

Algorithm 3 Procedure FCM

```
procedure FCM
    Initialize parameters c, m, ε, and U matrix
    repeat
        Calculate prototype vectors v_j
        Calculate membership values U
        Compare U^{t+1} with U^t where t is iteration number
    until t reaches the maximum iteration number or | U^{t+1} − U^t |< ε
end procedure
```

As the FCM pseudo-code states, we would need to adapt the clustering algorithm to fit the problem of clustering learners based on their familiar degree of concepts.

The first input for the FCM algorithm is the data objects. Data objects are learners' familiar degree of concepts. The familiar degree of concepts are represented as an array of size 20 (the number of concepts in the course) with values between 0 and 1. That is, the data input of the FCM is a two-dimensional array with the number of rows equal to the number of learners and the number of columns equal to the number of concepts in the course.

The initial membership degrees U were randomly generated. We chose 4 for the number of clusters c. Based on the related literature by (Cebeci and Yildiz, 2015) and (Fahad et al., 2014), we chose 2 for the fuzzifier factor m and $1e-7$ for the convergence value ϵ.

The output from FCM is an array with the number of learners as the number of its rows and the number of columns as the number of clusters. Each column in a row includes the membership degree of the learner for the corresponding cluster.

As mentioned, one main reason to adapt the FCM clustering was to provide a more accurate clustering for the data points located in the clusters' boundaries. As a result, we modified the FCM component to find the two highest values of membership degrees for each learner. If the difference between the two values are less than 5% (0.05), these two clusters are considered for the learner. Thus, when recommending a learning path, we recommend the learning path for both clusters to the learner. This way the learner is not obligated to use only one learning path to complete the course.

For example, if the membership degrees for learner X are 0.0573 for cluster 1, 0.4134 for cluster 2, 0.1429 for cluster 3, and 0.3863 for cluster 4, both clusters 1 and 4 are considered for the learner since the difference between the two membership degrees is less than 0.05 $(0.4134 - 0.3863 = 0.0271 < 0.0500)$ and the learning paths corresponding to both clusters are recommended to the learner.

3.5 ACO Learning Path Finder

The most popular swarm intelligent algorithm used in content planning is ant colony optimization (ACO). ACO algorithm is inspired from social creatures like ants and bees. Ants construct networks of paths to link their nests to food sources. These networks form minimum spanning trees that minimize the energy ants spend to transfer food to the nest. Ants would find optimal paths from their colony to the food sources by releasing special chemicals called pheromones that guide them to either food or the nest.

As indicated, we have focused on the algorithm proposed by Kardan et al (Kardan et al., 2014). There are some weaknesses in their work. One weakness is their clustering algorithm. We have addressed this by adapting the FCM clustering algorithm and recommending more than one learning path for learners located in the clusters' boundaries. Another weakness is that the familiar degree of concepts are only calculated after the initial test and the learners' gradual improvement during the course is not included in the pheromone calculation and learning path selection process. We have addressed this issue by reassessing the learners after completing each concept and updating the familiar degrees of concepts and re-clustering the learner accordingly.

Also, regarding the authors ACO algorithm, it was not clear if they have only incorporated the data of those ex-learners who have been successful in the course to generate a learning path. In our algorithm only those ex-learners' data who have been successful in the course are incorporated in the ACO algorithm to search for the suitable learning path for a new learner. The following will explain our ACO algorithm in detail.

The ACO algorithm can be applied to optimization problems that can be modeled as a graph. As indicated previously, content planning problem fits this condition since the learning concepts are the nodes in the graph and edges would represent the relationship between the concepts. In our algorithm, we assume that all learning concepts are connected. That means a learner can go from any learning concept to any other learning concept. The goal in this problem is to find a sequence of learning concepts that can provide a better support for a group of learners in their learning. We have followed the Ant System algorithm presented by Dorigo et al. in (Dorigo and Gambardella, 1997).

Given a set of n learning concepts, the content planning problem can be defined as a problem to find a sequence of learning concepts that can provide a better support for a group of learners in their learning where the learner is offered each learning concept once. The course graph $G(N, E)$ where N is a set of learning concepts and E is the set of edges between the concepts. Let m be the total number of ants. Each ant is a simple agent that has the following characteristics:

- The ant chooses a learning concept with a probability that is a function of two parameters: 1) cost of choosing the learning concept (indicated as a weight on the connecting edge) and 2) amount of pheromone trails existing on the connecting edge;

- A list of selected learning concepts is created to ensure the ant will not select a learning concept twice in the same tour.

- When the ant selected all the learning concepts, it lays a pheromone trail on each edge traveled.

Let $\tau_{ij}(t)$ be the pheromone trail on $edge(i, j)$ at time t. At time t each ant selects the next learning concept to be visited at time $t + 1$. Therefore, in every n

iteration, each ant completes a tour including all the learning concepts. After every n iteration the pheromone trail is updated for the whole tour according to the following Equation 3.6:

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \triangle\tau_{ij} \tag{3.6}$$

where ρ is the evaporation factor of the pheromone trail between the time t and $t+n$,

$$\triangle\tau_{ij} = \sum_{k=1}^{m} \triangle\tau_{ij}^{k} \tag{3.7}$$

where $\triangle\tau_{ij}^{k}$ is amount of pheromone by ant k to be deposited for the $edge(i,j)$ between time t and $t+n$, this amount of pheromone is given by Equation 3.8:

$$\triangle\tau_{ij}^{k} = \begin{cases} FD_j^k & PFD_j^k \geq 0.75 \\ 0 & Otherwise \end{cases} \tag{3.8}$$

where FD_j^k is the familiar degree of ant k for concept j (that we referred to it as $A_{S_k C_j}$ in previous sections) on pre final assessment of the course and PFD_j^k is the familiar degree of ant k for concept j post final assessment of the course. That means only those ex-learners (ants) who have a familiar degree of concept higher than 0.75 will contribute to the search for the learning path and the amount of pheromone is equal to the ex-learners' familiar degree of the concept. This way the concepts that most ex-learners have a higher familiar degree would have more chance to be selected as the next concept.

As mentioned earlier, each ant chooses the next learning concept based on a probability function. The probability function of moving from learning concept i to learning concept j is indicated in the Equation 3.9:

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_i(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}} & if j \in allowed_k \\ 0 & Otherwise \end{cases} \tag{3.9}$$

where η_{ij} is the heuristic or visibility factor that does not change during the algorithm run as opposed to the pheromone. In our algorithm, the η_{ij} is the concept map (or the learning path) recommended by the course designer that is a two-dimensional array with the number of rows and columns equal to the number of concepts. The value of η_{ij} indicates the cost of moving from learning concept i to learning concept j. The smaller the value of η_{ij} the more visibility the node j would have. The visibility array used for our algorithm is indicated in Table 3.3. The $allowed_k$ is a set of learning concepts that have not been visited yet. The α and β are parameters that would control the importance of the visibility factor and pheromone trail. As can be seen, the value of pheromone trail, visibility factor, α, and β impact the probability of selecting the next learning concept for the ant. Dorigo (Dorigo and Gambardella, 1997) suggested values for α and β that produced a good solution with the same performance level. We were able to reproduce the same results in the algorithm. As a result, we choose $\alpha = 1$ and $\beta = 1$. We also set the number of iteration to 1000 and the value of ρ to 0.5.

Table 3.3: Visibility Array

Concept	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
2	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
3	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
4	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
6	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
7	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
8	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
9	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
10	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
11	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5
12	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5	0.5
13	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5	0.5
14	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5	0.5
15	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5	0.5
16	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5	0.5
17	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4	0.5
18	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	0.4
19	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1

The pseudo code of our ACO algorithm is shown in algorithm 4:

Algorithm 4 Procedure ACO

```
procedure ACO
    Input: FD and PFD of ex-learners in the same cluster as current learner
    Initialize parameters α, β, η, and τ
    repeat
        Generate ants based on ex-learners
        Assign a concept to each ant randomly
        repeat
            Compute the probability function for the current ant based on Equation 3.9
            Choose the next concept based on the probability function for each ant
        until allowed_k is empty
        Update the pheromone trails based on Equations 3.7 and 3.8
    until maximum iteration number is reached
    Return best learning path
end procedure
```

In summary, in the ACO learning path finder, one major contribution is the way we calculate the pheromone. We only use the familiar degree of concepts for the ex-learners who successfully completed the concept based on their post final test familiar degree of concept. In addition, we have used a learning path (concept map) recommended by course designer as the heuristic information (visibility factor). Consequently, the resulting learning path is optimized based on the learning path recommended by course designer and the similar ex-learners' familiar degree of concepts.

Chapter 4

Implementation

4.1 Implementation Details

In this section we will explain the implementation details of the LPR system. The LPR system includes the familiar degree of concepts calculator, FCM clustering algorithm, ACO learning path finder, and a learner interface component for the learner to request a learning path and then view the recommended learning path. All the components of the LPR system were developed in Java and deployed as a web application on Microsoft Azure. The LPR web application was accessible within the course as a widget.

The course materials reside in a Learning Management System (LMS) branded as UM Learn which is a Brightspace LMS from the desire2learn company. The test environment of UM Learn was used to host the course. Since there is no API (bri, 2017) available currently to access the results of questions of the quizzes in UM Learn, we decided to use a third-party open-source quiz tool called “Savsoft Quiz v3.0” (sav,

2017) to host our assessments. Savsoft Quiz tool is a web application implemented in PHP and MySQL. The Savsoft Quiz application was deployed on development servers at the University of Manitoba.

4.1.1 Familiar Degrees of Concepts Calculator

The familiar Degrees of Concepts Calculator would require the result of the pre-assessment test of the learner as an input. In order to get the learners' test (Quiz) results, a connection to the Savsoft Quiz tool MySQL database is established to retrieve the results for the learner on the pre-assessment quiz questions.

In the next step, the familiar degrees of concepts for the learner is calculated based on Equation 3.1. The familiar degrees of concepts are stored in a two-dimensional array as a return parameter and are added to a table in the MySQL database containing familiar degrees of concepts for ex-learners. The reason for this is because the current learner will be considered as an ex-learner for the future learners.

In addition, the same process is performed when a learner takes a learning concept test after completing a learning concept. The latest results are fetched from the MySQL database to be used for the familiar degrees of concepts calculation.

4.1.2 FCM Clustering

The FCM clustering component requires the ex-learners' familiar degrees of concepts for the pre-assessment that are retrieved from the MySQL database and the current learner's familiar degrees of concepts as an input. The inputs are passed to the FCM from the previous component. When clustering is run, the generated output is a two-dimensional array containing the membership degree values of each learner for each cluster.

However, as an output of the FCM component, we generate a two-dimensional array with the number of rows equal to the number of ex-learners plus the current learner and the number of columns equal to 2. The first column contains the learner id and we assign one of the numbers 1, 2, 3, or 4 (representing their cluster number) in the second column for the learners according to their highest membership degrees.

In the next step, the best two clusters are found for the current learner and are included as an output of the FCM component. As mentioned before, if the difference between the two highest membership degrees assigned to the learner is greater than 0.05, only one cluster is assigned to the learner and for the second cluster 0 is recorded. We also record the best two clusters that are assigned to the learner in each step into a table in the MySQL database.

4.1.3 ACO Learning Path Finder

The ACO component requires the familiar degrees of concepts for pre-assessment and the familiar degrees of concepts for the final test for all ex-learners that are in the same cluster as the current learner. Then, the ACO is executed. This algorithm generates an output array of size equal to the number of learning concepts in the course. This contains the learning concepts in an order shaping a learning path.

In case the current learner is assigned a secondary cluster, the ACO component is called again. This time the inputs for the ACO are the familiar degrees of concepts for pre-assessment and the familiar degrees of concepts for the final test for all ex-

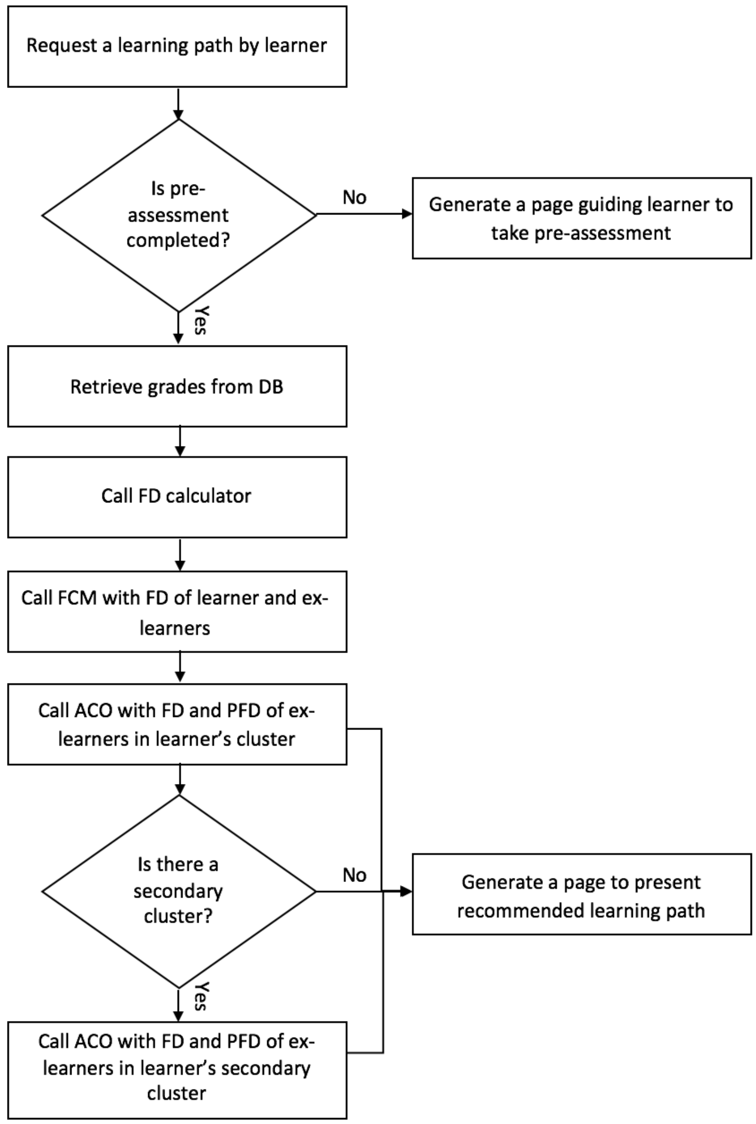

Figure 4.1: Diagram of calling order of different components within the Java servlet

learners that are in the learner's secondary cluster. In this case the output is another array containing the learning path based on the secondary cluster.

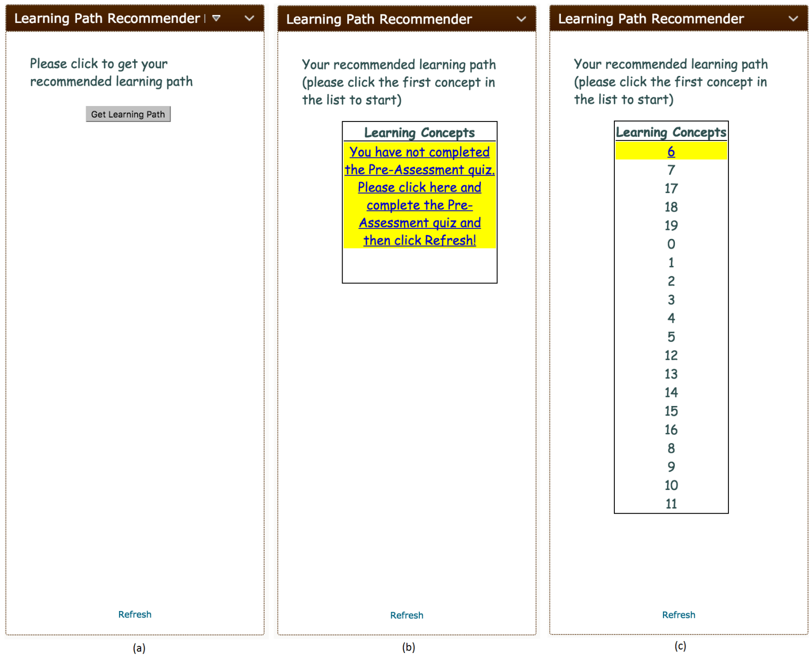

Figure 4.2: Screenshots of the user interface (a) Initial page (b) Requested learning path without completed Pre-assessment (c) Presented with learning path

4.1.4 User Interface

As mentioned, we used Java web application to implement the system. The user interface contains a web-page with a button for the learner to request a learning path from the system. When a learner clicks on the button, if the learner has not completed the pre-assessment, a message is generated indicating that the learner must complete the pre-assessment before requesting a learning path. If the pre-assessment has been completed by the learner, the application generates the learning path for the learner and indicates the learning path as a series of numbers (representing the corresponding learning concepts) in a list. There is also a refresh button in the page for the learners to reset the page to be able to request a new learning path for the next steps.

All the components of the LPR system are called within a Java servlet that produces the user interface pages for the learners based on their requests. The diagram in Figure 4.1 shows how the LPR system calls the different components within the Java servlet and Figure 4.2 shows the screen shots of the user interface within a widget in UM Learn.

4.2 Course Design and Development

We have designed and developed a short course to evaluate the LPR system. In this section, we explain the process of designing and developing the course as well as providing some details about the course.

Our goal in designing the course was that the course topic should be in the computer science field. However, the course should be designed in a way that learners with different backgrounds could take the course and achieve the objectives. We decided to design a course on introduction to relational databases. Since participants involvement in the study was voluntary and the course was a stand-alone course (not part of a program), we decided to design the course that can be completed in approximately 2 to 3 hours.

We followed the constructive alignment process proposed by Biggs and Tangs (Biggs, 1996) to design and develop the course. The constructive alignment process follows a sequential order to design a course which includes creating: 1) course goals 2) learning objectives 3) assessments 4) teaching strategies (materials and activities). These four elements of a course need to be aligned to produce a well developed course.

4.2.1 Course Goals

We started with generating the broad, general course goals while considering the above mentioned constraints. The learners would be able to achieve the following course goals by the end of the course:

- Organize and store data in relational databases
- Use SQL (Structured Query Language) to fetch data out of databases
- Use SQL to create databases and insert new data into databases

4.2.2 Unit Learning Objectives

In the next step, the course goals were broken down into specific, measurable unit learning objectives following the cognitive domain of Bloom's Taxonomy (Huitt, 2004). The unit learning objectives for the course goal "Organize and store data in relational databases" are:

- Define database and its purpose
- Structure data into tables
- Identify data types and meanings
- Extract data from tables
- Count rows in a table with aggregations
- Link table together using primary keys and joins

Here are the unit learning objectives for the course goal "Use SQL (Structured Query Language) to fetch data out of databases":

- Identify SQL data types and comparison operators
- Identify elements in SQL select statement
- Identify SQL select clauses
- Write a SQL select statement with aggregation
- Write SQL select queries with aggregations and joins
- Write SQL select queries with having clause
- Design normalized databases
- Identify and declare primary keys in tables
- Identify foreign keys in tables
- Write a SQL query with self joins
- Write a SQL query with sub-queries
- Define view and its purpose

The unit learning objectives for the course goal "Use SQL to create databases and insert new data into databases" are as follows:

- Write SQL insert statement
- Create and drop databases and tables

4.2.3 Assessments

As previously mentioned, the formative and summative assessments in the course to measure the course goals and unit learning objectives are: 1) pre-assessment quiz; 2) learning concept quizzes; 3) final exam quiz.

We created 20 multiple choice questions for the pre-assessment quiz. Each question measures a specific unit learning objective. For example, for the unit learning objective "Identify elements in SQL select statement", the aligned question from the pre-assessment quiz is:

Q7) Look at the query below. The elements *select*, *name*, and *or* in the query are:

Select name, gender from Students where birthdate < '2000-01-01' or birthdate > '1990-01-01';

(A) Keyword, table name, and Boolean operator

(B) Keyword, table name, and comparison operator

(C) Keyword, column name, and comparison operator

(D) Keyword, column name, and Boolean operator

The time allocated for the pre-Assessment quiz was set to 40 minutes (2 minutes per question) with a single attempt allowed.

Table 4.1: Relevance Degree of Concepts and Questions

Q / C	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0.2	1	0.2	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0.2	0	1	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0
3	0	0	0.1	1	0	0	0	0	0	0.2	0	0.1	0	0	0	0	0	0	0	0
4	0	0	0.1	0	1	0	0	0.4	0.2	0	0.1	0	0	0.2	0	0	0	0.1	0.1	0.1
5	0	0	0.1	0	0	1	0	0	0	0	0.2	0	0	0	0	0.2	0.1	0.2	0	0
6	0	0.5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0.1	0	0	0.4	0	0.1	1	0	0	0	0	0	0	0	0	0	0	0.2	0.2
8	0	0	0	0	0.2	0	0	0.2	1	0.6	0	0	0	0	0	0	0	0	0.2	0.1
9	0	0	0.2	0.2	0	0	0	0.1	0.6	1	0	0.3	0	0	0	0	0	0	0.1	0
10	0	0	0	0	0.1	0.2	0	0.1	0	0	1	0	0	0	0	0	0.2	0.7	0.1	0
11	0	0	0	0.2	0	0	0	0	0.1	0.2	0	1	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0.1	0.2	0	0	0
14	0	0.1	0.2	0	0	0	0.2	0	0	0	0	0	0	0	1	0	0	0	0	0
15	0	0.1	0	0	0	0.2	0.2	0	0	0	0	0	0	0.1	0.1	1	0.2	0	0	0
16	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
17	0	0	0	0	0.1	0.1	0	0.2	0.2	0	0.7	0	0	0	0	0	0	1	0.2	0
18	0	0	0	0.1	0	0	0	0.2	0.2	0.2	0	0	0	0	0	0	0	0	1	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

As part of the course design, we created a table that includes the relevance degree of learning concepts with questions in a test (test items). This relevance degree table is used later to calculate the familiar degrees of concepts for each learner as indicated in previous chapter. Table 4.1 shows the relevance degree of concepts and questions for both sets of questions (pre-assessment and final exam).

The learning concept quizzes that are completed after each learning concept, used the same questions from the question bank of the pre-assessment quiz. Each learning concept quiz includes one question that directly measures the unit learning objective of the learning concept. It also includes questions measuring other learning objectives that are relevant to the learning concept based on the information in the relevance degree of concepts and questions, mentioned in Table 4.1. For example, concept 3 quiz includes questions 3, 9, 11, and 18 based on the relevance degree of concepts and question.

The time allocated for each concept quiz depends on the number of questions it includes. For example, time allocated to the concept 3 quiz was set to 8 minutes (2 minutes per question) with a single attempt allowed.

Similar to the pre-assessment quiz, we created 20 multiple choice questions for the final exam quiz. Each question measures a specific unit learning objective. For example, for the unit learning objective "Identify elements in SQL select statement", the aligned question from the final exam quiz is:

Q7) Look at the query below. The elements *name*, *where* and $<$ in the query are:

Select name, gender from Students where birthdate $<$ *2000-01-01;*

(A) Keyword, table name, and Boolean operator

Table 4.2: Learning Concepts and Corresponding Unit Learning Objective

Learning Concept Title	Unit Learning Objective
Concept 0 What is a database	Define database and its purpose
Concept 1 Data Types and meanings	Identify data types and meanings
Concept 2 Anatomy of tables in databases	Structure data into tables
Concept 3 Aggregations	Count rows in a table with aggregations
Concept 4 Queries and results	Extract data from tables
Concept 5 Relationships in databases	Link table together using primary keys and joins
Concept 6 Types in SQL	Identify SQL data types and comparison operators
Concept 7 SQL select basics	Identify elements in SQL select statement
Concept 8 SQL select clauses	Identify SQL select clauses
Concept 9 Aggregations in SQL select	Write a SQL select statement with aggregation
Concept 10 Join in SQL select	Write SQL queries with aggregations and joins
Concept 11 SQL select having clause	Write SQL select queries with having clause
Concept 12 SQL insert statement	Write SQL insert statement
Concept 13 Database normalization	Design normalized databases
Concept 14 Create table and types	Create and drop databases and tables
Concept 15 Declare primary keys	Identify and declare primary keys in tables
Concept 16 Foreign keys	Identify foreign keys in tables
Concept 17 Self joins	Write a SQL query with self joins
Concept 18 Subqueries	Write a SQL query with subqueries
Concept 19 Database view	Define view and its purpose

(B) Keyword, table name, and comparison operator

(C) Column name, keyword, and comparison operator

(D) Column name, keyword, and Boolean operator

The time allocated for the final exam quiz was set to 40 minutes (2 minutes per question) with a single attempt allowed.

4.2.4 Teaching Strategies (materials and activities)

Once the unit learning objectives were established, the main themes of the learning objectives were determined and grouped to form the learning concepts. The learning concepts for the course as aligned to the unit learning objectives are shown in Table 4.2. The concepts in the course are named with a number between 0 to 19 and a short title (e.g., Concept 2 Anatomy of Tables in Relational Databases).

As a self paced on-line course, we determined the best teaching strategies to achieve the learning objectives were short videos and reading materials and practice query writing activities. The videos and reading materials were sourced selectively from an on-line course in Udacity (uda, 2017) and available under Creative Commons Attribution-NonCommercial- NoDerivs 3.0 License. The readings and videos introduced the content for each learning concept and posed practice questions for the learners to complete before they proceeded to the concept quiz.

The course materials and activities were organized and placed in a course shell in UM Learn for learners to access. However, the assessment quizzes were placed on a third party quiz tool Savsoft. The links to the quizzes were provided within the course in UM Learn. Learners could complete the course at their own pace.

Chapter 5

Experimental Setup and Evaluation

In this chapter, we explain details of the process of the experimental setup, the result of our experiment, and a discussion regarding the evaluation of our system.

5.1 Experimental Design

Based on similar recent research work on the content planning problem (Alshalabi, 2016) and (Kurilovas et al., 2015), we decided to evaluate our system with a real course and participants. In this section we explain the two stage process of our experiment: 1) to gather ex-learners' data for the system and 2) to evaluate the effectiveness of the LPR system.

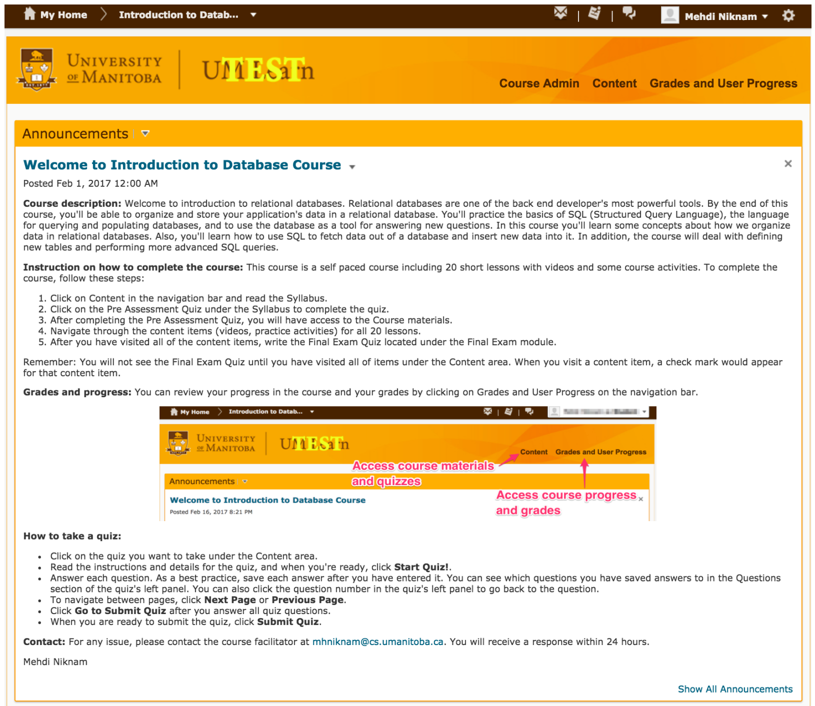

Figure 5.1: Screenshot of the course homepage for the first stage

5.1.1 First Stage

As mentioned, our LPR system requires data from ex-learners in order to find and recommend learning paths to the new learners. As a result, we created a course within UM Learn and recruited learners - undergraduate and graduate university level students from various majors - to complete the course and gather the data.

As indicated in the implementation details section, a short on-line self-paced "Introduction to Databases" course was designed for university level students. Figures 5.1 and 5.2 respectively show the course homepage and a sample learning concept for this course.

Learners were given an instruction in the course homepage on the steps needed

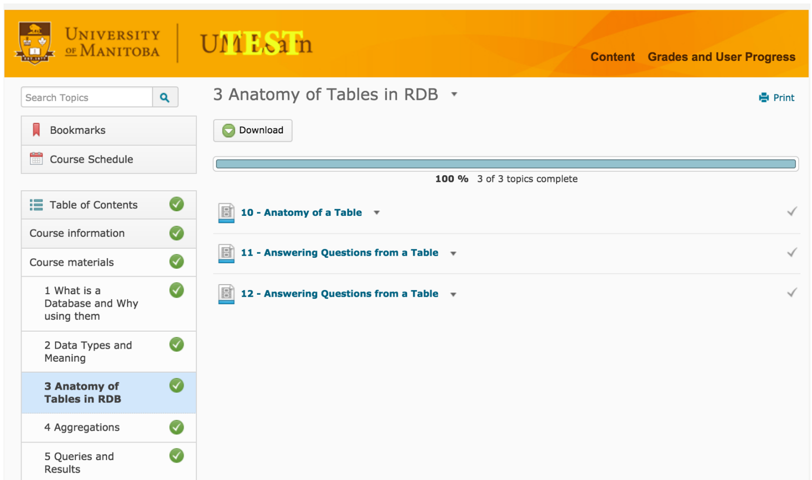

Figure 5.2: Screenshot of a sample learning concept's content materials of the course in the first stage

to complete the course. Learners would complete the pre-assessment quiz at the beginning of the course and then the course materials were released to them to review. The learners were required to review all course materials and complete all learning activities (within the 20 learning concepts) in order to be able to see the final exam quiz. Finally, learners would complete the final exam in order to complete the course.

For this course, we did not include the learning concept quizzes. The learning concept quizzes are necessary when the learning path recommendation is provided to learners. Since at this stage we did not use the learning path recommendation system, we did not include the learning concept quizzes to reduce the completion time of the course for learners.

The process of recruiting learners and completing the course by learners for this stage took 5 weeks where 40 learners completed the course. We used the results of assessments from these learners and calculated the ex-learners' familiar degree of concepts on the pre-assessment and final exam tests to be used in our LPR system. As mentioned in the LPR system chapter, the ACO algorithm requires a large number of ex-learners to converge during the search of the learning path. As a result, we equally replicated these familiar degrees of concepts for our experiment.

Based on the feedback received from this group of learners, we slightly modified the instruction at the beginning of the course to make it more clear for the learners in the main experiment. Besides the fact that we needed the ex-learners' data to be used in our system for the main experiment, the feedback received from the learners led us to reduce the confusions within the instructions provided to learners in the main experiment.

5.1.2 Second Stage

The main goal of the experiment is to investigate the effectiveness of our proposed system on the learning performance and completion time for learners. Therefore, we focused on the learners' results of pre-assessment and final exam and the time it took learners to complete the course.

As mentioned before, we modified the original course from the previous stage to be used in this stage. In this stage, we created two versions of the "Introduction to Databases" course. One course did not use our LPR system and the course designer's recommended learning path was provided to the learners. The other course included our LPR system and provided the personalized learning path recommendation to the learners.

In order to provide the similar experience to both groups of learners, we added a new widget called "Learning Path Recommender" to the course homepage for both

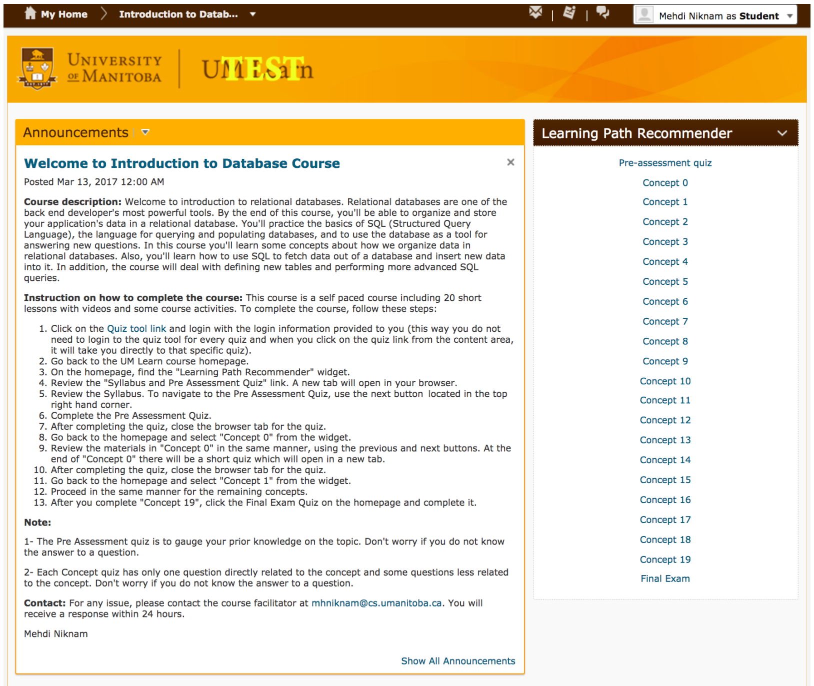

Figure 5.3: Screenshot of the course homepage for the course without recommendation in the second stage

courses. This widget in the course without learning path recommendation included a static learning path provided by the course designer that each learner can navigate through the learning concepts and assessments. Figure 5.3 shows the course homepage for the course without the learning path recommendation.

In the second version of the course, the widget includes our LPR system learner interface explained in the implementation details chapter. Figure 5.4 shows the course homepage for the course with the learning path recommendation.

In both courses in this stage, learners would start with taking the pre-assessment quiz. Next, learners navigate to the first learning concept's materials (either the first recommend concept by course designer for the course without recommendation or the first recommended concept by the LPR) to review and complete the related

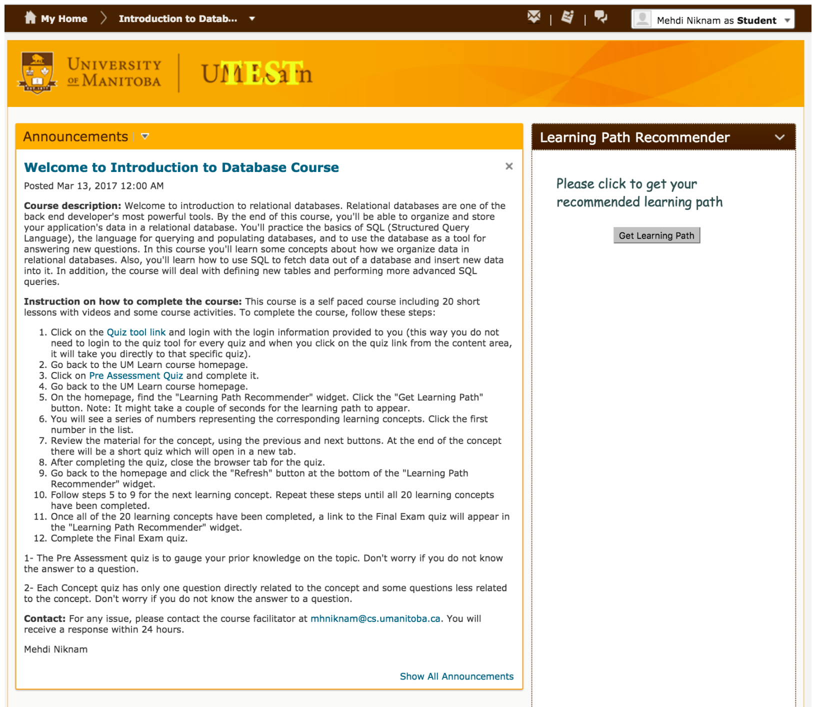

Figure 5.4: Screenshot of the course homepage for the course with recommendation in the second stage

activities. Then, learners complete the learning concept quiz. Learners continue the same process until all 20 learning concepts are completed. Finally, learners would take the final exam quiz to complete the course.

The difference between the courses in this stage and the first stage is the addition of the learning concept quizzes. The appropriate instruction for the learners have been provided in both courses. We have adjusted the instruction to reflect the fact that the quizzes are hosted on a third-party quiz tool (Savsoft). The scores for the pre-assessment and the final exam quizzes were captured in the third-party quiz tool's (Savsoft) database. Learners were given 40 minutes to complete each of the pre-assessment and the final exam quiz. The quizzing tool enforced the 40 minutes and the learners were given warning to complete the quiz before the time limit elapsed.

We recruited undergraduate and graduate level university students to complete the courses in our second stage experiment. We categorized the students based on their level and background into the following categories:

- Undergraduate level non-IT related major
- Undergraduate level IT related major
- Graduate level non-IT related major
- Graduate level IT related major

Each experimental group was assigned an equal number of learners based on their category to complete the courses. That means if the LPR group was assigned 6 learners with the category 1, 6 learners with the category 1 were assigned to the control group as well. With this categorization of learners' demographic and prior knowledge, we tried to have similar groups of learners in terms of familiarity to the course topic and prior knowledge to eliminate any bias in terms of the learners' prior knowledge in the experimental groups.

The process of recruiting learners and completing the course by learners for both courses in this stage took 3 weeks where 55 learners completed the courses. 28 learners in the control group completed the course without recommendation and 27 learners in the LPR group completed the course with recommendation.

As indicated, our experiments' objective is to investigate the effectiveness of the LPR system on learner's performance and course completion time. The learners' performance in the pre-assessment and final exam assessments were recorded in the quiz tool (Savsoft) as a number out of 100. UM Learn provides a detailed time report for the learners' interaction with course materials within the learning concepts. Also, the completion time for each quiz (pre-assessment, concept quizzes, and final exam) is provided in the quiz tool as well. We extracted and added the time spent for both mentioned items (content and assessments) to obtain the course completion time for each learner.

5.1.3 Data Analysis

As mentioned, 55 learners participated in our experiment. After reviewing the data, we noticed some learners from both groups (the group with recommendation and the group without recommendation) only completed the pre-assessment and the final exam quizzes and did not interact with the course content. The indicator was that these learners did not spend the required time on the learning concept materials in the course. These learners' completion times were significantly lower than the average completion time of both groups. Consequently, we removed the results related to those learners and other outliers from the data and ended up with two groups of 25 learners.

In our research we are trying to answer the following three questions:

1. Was there any significant difference in the pre-assessment score of the LPR group and the control group?
2. What impact has the usage of our LPR system on learners' learning performance?
3. What impact has the usage of our LPR system on learners' course completion

time?

5.2 Evaluation

In this section, we are reviewing and analyzing the results of our experiment to answer our three research questions mentioned in the previous section. In this experiment, the group of learners are our independent variable and the pre-assessment scores, final exam scores, and completion times are our dependent variables. We used SPSS statistical software to perform the statistical analysis on our data.

5.2.1 Research Question 1

Answering question 1 "Was there any significant difference in the pre-assessment score of the LPR group and the control group?" would indicate if the learners in both groups had similar prior knowledge of the course topic when entering the course. The answer to this question is very important to ensure no bias was included in selecting participants for our experiment.

Firstly, we have tested our data (the learners' pre-assessment scores for both groups) to make sure the data distributions were sufficiently normal for conducting a t-test (i.e., $skew <\mid 2.0 \mid$ and $Kurtosis <\mid 9.0 \mid$; (Schmider et al., 2010)). We also used the Shapiro-Wilk normality test in SPSS. Figure 5.5 and 5.6 respectively show the normal Quantile-Quantile plot of the pre-assessment scores for the LPR group and control group. Table 5.1 shows the statistical analysis of the pre-assessment data for both groups.

As can be seen in Table 5.1, the mean value for both groups are very close. This

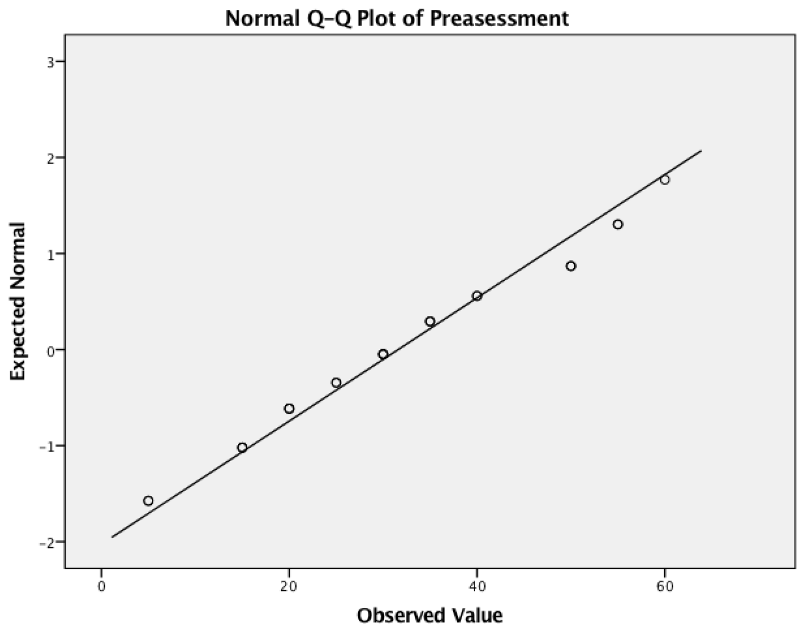

Figure 5.5: Normal Quantile-Quantile plot of the pre-assessment scores for LPR group

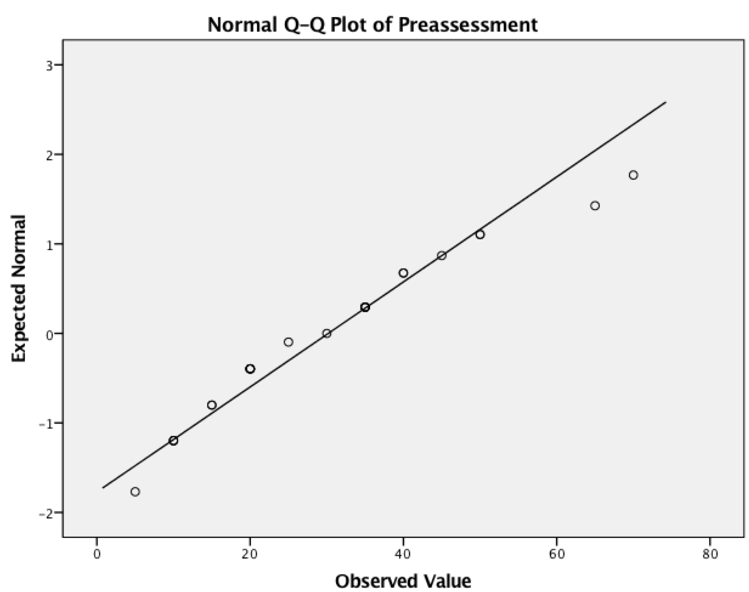

Figure 5.6: Normal Quantile-Quantile plot of the pre-assessment scores for control group

Table 5.1: Statistical analysis for pre-assessment score of LPR and control groups

Group	N (Sample Size)	Mean	Standard Deviation
LPR	25	31.6	15.5938
Control	25	30.2	17.04651

Table 5.2: Result of t-test on pre-assessment scores of both groups

t	df	p	Mean Difference	Std. Error Difference	CI
-0.303	47.624	0.763	-1.4	4.6206	(-10.6922, 7.8922)

indicates that both groups had a similar prior knowledge of the course topic when entering the course. We also used an independent sample t-test on the pre-assessment data of both groups to ensure that there was no significant difference between the pre-assessment scores of both groups.

Let M_x denote the mean for the pre-assessment score of LPR group and M_y denote the mean for the pre-assessment score of the control group. The statistical hypotheses for this research question are formulated as follows:

$$HO : M_x - M_y = 0$$

$$HA : M_x - M_y \neq 0$$

where the significance level α is equal to 0.05.

Table 5.2 shows the results obtained from the independent sample t-test run on the pre-assessment scores of both groups. The df stands for degree of freedom and CI stands for the confidence interval in the table.

As the result for the t-test indicates, the p value is 0.763 which is greater than 0.05 our α value. Also, zero is in the CI range ($-10.6922 < 0 < 7.8922$). Therefore, we fail to reject our null hypothesis and we can infer that there is no significant difference between the prior knowledge of learners in the LPR and control groups.

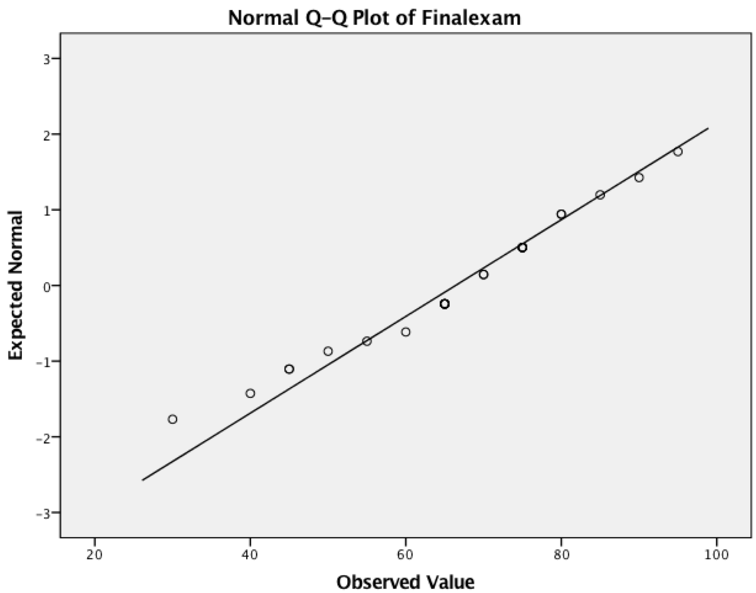

Figure 5.7: Normal Quantile-Quantile plot of the final exam scores for LPR group

5.2.2 Research Question 2

Our question 2 "What impact has the usage of our LPR system on learners' learning performance?" investigates the effectiveness of the LPR system on learners' learning performance. For this question, we have analyzed the final exam scores of both groups as well as the difference between learners' pre-assessments and final exam scores. The final exam would indicate the learner's ultimate performance and the difference of the pre-assessment and final exam would demonstrate the amount of improvement learners achieved after taking the course. We analyzed the result for the final exam score first and then the score difference.

Similar to the question 1, we tested our data (the learners' final exam scores for both groups) - to make sure the data distributions were sufficiently normal for conducting a t-test (i.e., $skew <| 2.0 |$ and $Kurtosis <| 9.0 |$). For this dependent

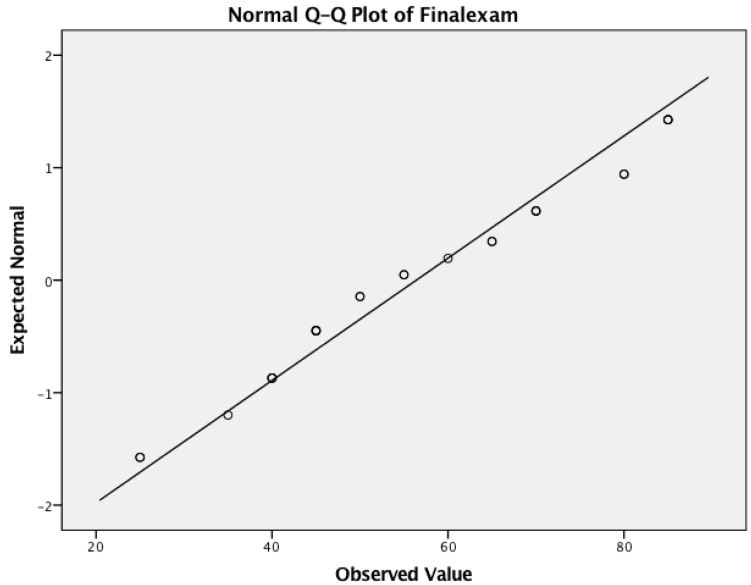

Figure 5.8: Normal Quantile-Quantile plot of the final exam scores for control group

variable, we also used the Shapiro-Wilk normality test in SPSS. Figure 5.7 and 5.8 respectively show the normal Quantile-Quantile plot of the final exam scores for the LPR group and control group.

Let M_x denote the mean for the final exam score of the LPR group and M_y denote the mean for the final exam score of the control group. The statistical hypotheses for this part of the research question are formulated as follows:

$$HO : M_x - M_y = 0$$

$$HA : M_x - M_y \neq 0$$

Where the significance level α is equal to 0.05.

Table 5.3 shows the statistical analysis of the final exam data for both groups. As can be seen, the mean value for the LPR group is 66.4 and for the control group is 56.4. Comparing the mean value indicates that the LPR group is higher than the

Table 5.3: Statistical analysis for final exam score of LPR and control groups

Group	N (Sample Size)	Mean	Standard Deviation
LPR	25	66.4000	15.64715
Control	25	56.4000	18.40063

Table 5.4: Result of t-test on final exam scores of both groups

t	df	p	Mean Difference	Std. Error Difference	CI
-2.070	46.792	0.044	-10.0000	4.83080	(-19.71947, -.28053)

control group by 10 points. In order to find out if this difference is significant, we also used an independent sample t-test on the final exam scores of both groups.

Table 5.4 shows the results obtained from the independent sample t-test run on the final exam scores of both groups.

As the result for the t-test indicates, the p value is 0.044 which is less than 0.05 our α value. Therefore, we reject our null hypothesis and we can infer that there is a significant difference between the final exam scores of learners in the LPR and control groups. In order to identify the magnitude of the positive impact of our algorithm on the final exam score, we calculated the effect size. We calculated the Cohen's d effect size using the following formula:

$$d = \frac{(M2 - M1)}{\sqrt{\frac{SD_1^2 + SD_2^2}{2}}} = \frac{10}{\sqrt{\frac{15.647^2 + 18.400^2}{2}}} = 0.59$$

The effect size $d = 0.59$ indicates a medium to large effect size ($d = 0.3$ small effect, $d = 0.5$ medium to large effect, and $d > 0.8$ large effect) of our LPR algorithm on the final exam score.

Similarly, we tested our data (the difference between the pre-assessment and final exam scores of learners) to make sure the data distributions were sufficiently normal

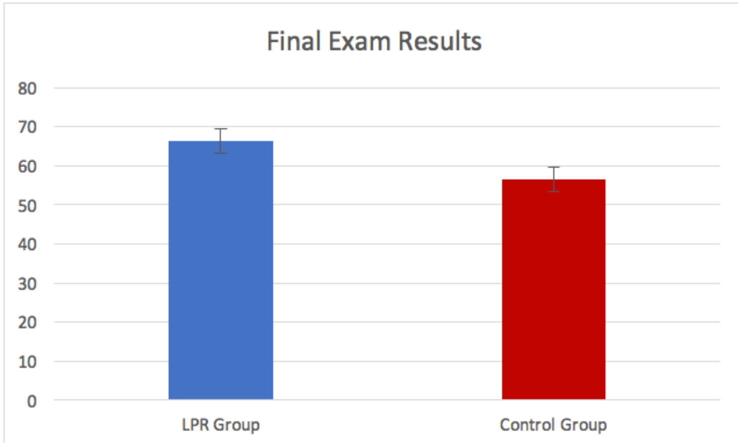

Figure 5.9: Average of the final exam scores for LPR and control group

for conducting a t-test (i.e., $skew <\mid 2.0 \mid$ and $Kurtosis <\mid 9.0 \mid$). We also used the Shapiro-Wilk normality test in SPSS. Figure 5.10 and 5.11 respectively show the normal Quantile-Quantile plot of the difference between the pre-assessment and final exam scores for the LPR group and control group.

Let M_x denote the mean for the difference between the pre-assessment and final exam scores of the LPR group and M_y denote the mean for the difference between the pre-assessment and final exam scores of the control group. The statistical hypotheses for this part of the research question are formulated as follows:

$$HO : M_x - M_y = 0$$

$$HA : M_x - M_y \neq 0$$

Where the significance level α is equal to 0.05.

Table 5.5 shows the statistical analysis of the difference between the pre-assessment and final exam for both groups. As can be seen, the mean value for the LPR group is

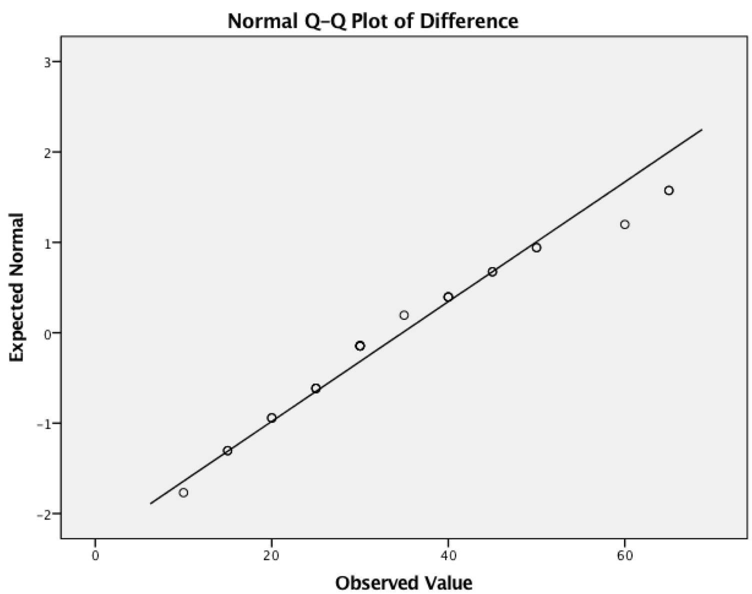

Figure 5.10: Normal Quantile-Quantile plot of the difference between pre-assessment and final exam scores for LPR group

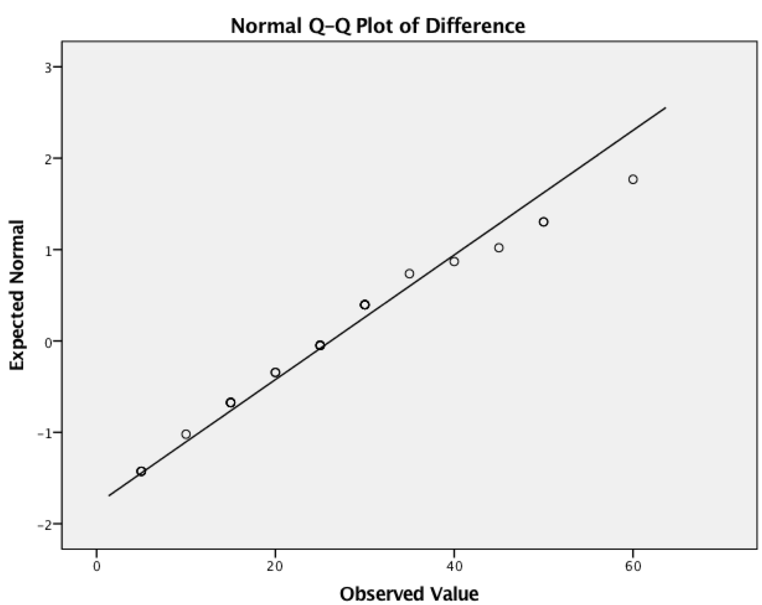

Figure 5.11: Normal Quantile-Quantile plot of the difference between pre-assessment and final exam scores for control group

Table 5.5: Statistical analysis for the difference between the pre-assessment and final exam score of LPR and control groups

Group	N (Sample Size)	Mean	Standard Deviation
LPR	25	34.8000	15.10243
Control	25	26.2000	14.66856

Table 5.6: Result of t-test on the difference between the pre-assessment and final exam scores of both groups

t	df	p	Mean Difference	Std. Error Difference	CI
-2.042	47.959	.047	-8.60000	4.21070	(-17.06637, -.13363)

34.8 and for the control group is 26.2. Comparing the mean values indicates that the LPR group is higher than the control group by 8.6 points. In order to find out if this difference is significant, we also used an independent sample t-test on the difference between the pre-assessment and final exam scores of both groups.

Table 5.6 shows the results obtained from the independent sample t-test run on the difference between the pre-assessment and final exam scores of both groups.

As the result for the t-test indicates, the p value is 0.047 which is less than 0.05 our α value. Therefore, we reject our null hypothesis and we can infer that there is a significant difference between the difference of the pre-assessment and final exam scores of learners in the LPR and control groups. In order to identify the magnitude of the positive impact of our algorithm on the difference of the scores, we calculated the Cohen's d effect size as follows:

$$d = \frac{(M2 - M1)}{\sqrt{\frac{SD_1^2 + SD_2^2}{2}}} = \frac{8.6}{\sqrt{\frac{15.102^2 + 14.668^2}{2}}} = 0.58$$

Similar to the final score, the effect size $d = 0.58$ indicates a medium to large effect size of our LPR algorithm on the amount of improvements the learners made during the course.

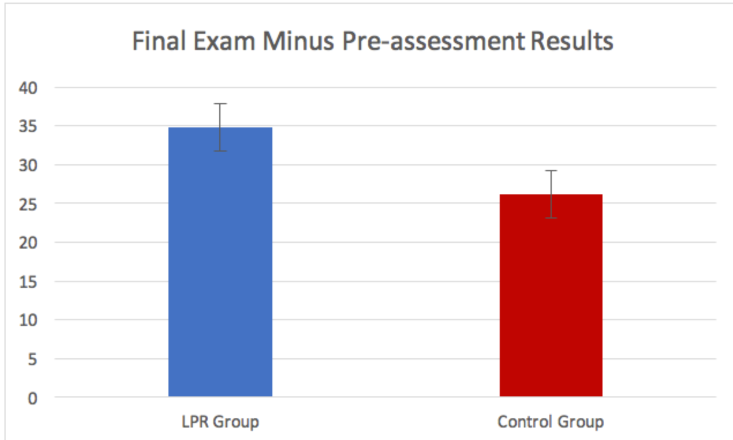

Figure 5.12: Average of the difference between pre-assessment and final exam scores for LPR and control groups

Consequently, the results of both t-tests on the final exam scores and the difference scores are significant between the LPR group and the control groups. Also, the effect size indicates that there is a moderate to large impact of our LPR algorithm contributing to this significant difference.

5.2.3 Research Question 3

With question 3 "What impact has the usage of our LPR system on learners' course completion time?" we are investigating if there was a significant difference between the course completion time of the learners in both groups and if the completion time of learners using the LPR system was significantly reduced.

We tested our data (the learners' completion time for both groups) to make sure the data distributions were sufficiently normal for conducting a t-test (i.e., $skew <| 2.0 |$ and $Kurtosis <| 9.0 |$). For this dependent variable, we also used the Shapiro-

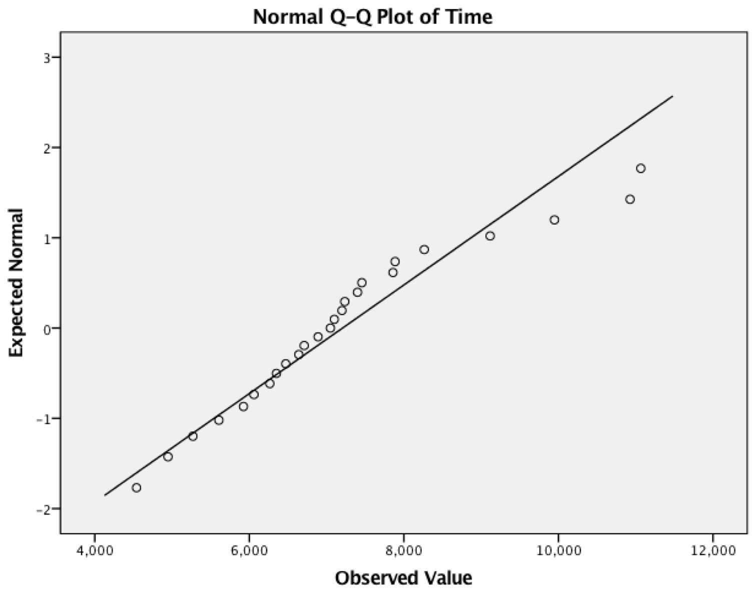

Figure 5.13: Normal Quantile-Quantile plot of the completion time for LPR group

Table 5.7: Statistical analysis for completion time of LPR and control groups

Group	N (Sample Size)	Mean	Standard Deviation
LPR	25	7207.1600	1662.84762
Control	25	7528.7600	1593.66392

Wilk normality test in SPSS. Figure 5.13 and 5.14 respectively show the normal Quantile-Quantile plot of the completion time for the LPR group and control group. Table 5.7 shows the statistical analysis of the completion time data for both groups.

As can be seen in Table 5.7, even though the mean value of the completion time for the LPR group is less than the mean value for the control group, they are close. We also used an independent sample t-test on the completion time of both groups to see if the difference between the completion times are significant.

Let M_x denote the mean for the completion time of the LPR group and M_y denote the mean for the completion time of the control group. The statistical hypotheses for

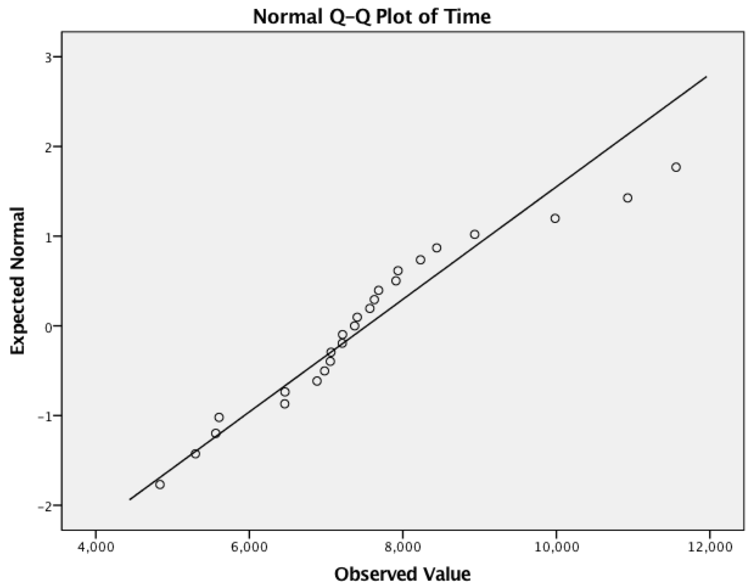

Figure 5.14: Normal Quantile-Quantile plot of the completion time for control group

Table 5.8: Result of t-test on completion time of both groups

t	df	p	Mean Difference	Std. Error Difference	CI
.698	47.914	.488	321.60000	460.64420	(-604.63039, 1247.83039)

this research question are formulated as follows:

$$HO : M_x - M_y = 0$$

$$HA : M_x - M_y \neq 0$$

where the significance level α is equal to 0.05.

Table 5.8 shows the results obtained from the independent sample t-test run on the completion time of both groups. The df stands for degree of freedom and CI stands for the confidence interval in the table.

As the result for the t-test indicates, the p value is 0.488 which is greater than 0.05 our α value. Also, zero is in the CI range ($-10.6922 < 0 < 7.8922$). Therefore, we fail to reject our null hypothesis and we can infer that there is no significant difference between the course completion time of learners in the LPR and control groups.

5.3 Summary of Evaluation

As part of the experimental setup, we categorized the participants and added an equal number of participants from categories to ensure both groups have similar prior knowledge of the course topic. As our evaluation results indicated, there was no significant differences between the pre-assessment scores of the learners in both the LPR and the control groups. This shows that no bias was included while selecting and assigning the participants in both groups.

In addition, the evaluation results show that there was a significant difference between the learners final exam scores between the LPR and control groups. The effective size of the impact was medium to large. That means our LPR system had a moderate to large positive impact on the learners' performance as the mean of the final exam scores of the learners in the LPR group was 10 points higher than the mean for the learners in the control group.

A similar result was achieved for the score difference between the pre-assessment and the final exam for learners in both groups. This would indicate that the amount of learning that learners of the LPR group achieved during the course was significantly higher than the learners in the control group. As the results show, the learning paths recommended by our LPR system provided more success to the learners' learning performance. Figure 5.15 shows the pre-assessment and the final exam scores of both groups side by side.

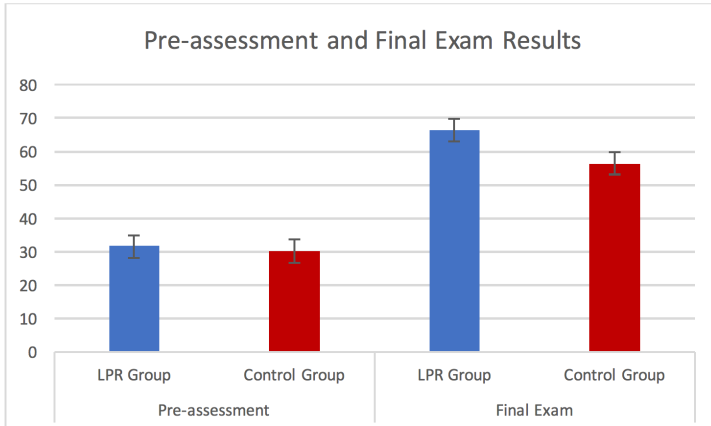

Figure 5.15: Pre-assessment and final exam scores of both groups

We also looked at the completion time for the course in both groups. Although the completion time for the LPR group was less than the completion time for the control group, there was no significant difference between them. One possible reason for no significant difference in completion time is that all learners would need to complete all the concepts in the course and that requires a fixed amount of time. The only part of the course that had variations in time would be the course activities and assessments.

Chapter 6

Conclusion

6.1 Summary of Thesis Achievements

With the advancement of technology, specifically in the field of education, there are more opportunities to provide personalized learning experiences for learners. An adaptive learning environment can provide personalized learning support for learners with different prior knowledge, learning preferences or styles. One way to personalize the learning experience is by recommending a personalized learning path to learners based on their characteristics.

In this thesis, we design and implement a learning path recommendation system called LPR. The learning path recommendation algorithm designed in the LPR system uses ant colony optimization and provides the learning path based on the meaningful learning theory.

The LPR system includes a pre-assessment/familiar degrees calculator, FCM learner clustering, ACO learning path finder, and learner interface components. The pre-assessment/familiar degree calculator is used to gauge learner's prior knowledge and to produce a learner's familiar degree of concepts. The clustering component assigns learners into groups based on their prior knowledge. The ACO learning path finder searches for the suitable learning path for the learner. The learner interface recommends the learning path to the learner.

To evaluate the LPR system, we designed and developed a course on the topic of databases. In order to achieve ex-learners' data to be used in our ACO learning path finder algorithm, we recruited learners to complete the course in the first stage of our experiment. In the second stage, we recruited learners and assigned them into two groups. The first group took the course with the LPR system and the second group took the course without the learning path recommendation.

The result of our experiment showed that the LPR group had a higher performance and improvement in the course than the control group. The performance difference between the two groups was significant based on the statistical test. Also, the amount of knowledge improvement was significantly higher in the LPR group compared to the control group. Based on the statistical tests, the LPR system had a moderate to large impact on the learners' performance and improvements. The course completion time for the LPR group was slightly less than the control group. However, we did not find a statistically significant difference between the completion time of both groups.

The positive results of our experiment indicate that the meaningful learning theory (one subset of cognitivism learning theory) is appropriate to generate personalized learning paths for learners.

In summary, the contributions of this research are as follows:

- We designed and developed a new ACO algorithm for content sequencing based on meaningful learning theory. The algorithm incorporates continuous learner's improvement in the process of a learning path selection.
- To overcome the weakness of K-Means clustering, we have adapted Fuzzy C-Mean clustering algorithm to recommend more than one learning path for learners located on the clusters boundaries.
- We evaluated the effectiveness of the learning path recommendation algorithm through an experiment with actual learners in a real course using the University of Manitoba's UM Learn system.

The LPR system is implemented as a prototype. The course materials and activities are residing in the LMS (UM Learn) and the course assessments are residing in a third-party quiz tool. The LPR learner interface is integrated into the LMS as a Java web application.

6.2 Future Work

This research can be extended in the following directions. First, developing a high enrollment course using the LPR system to extensively evaluate the system and provide improvements to the system. Running a course multiple times with more participants would increase the quality of the learning paths extracted for learners since there are more similar ex-learners' data to be used to find the suitable learning path and as a result it can improve the learners' performance.

Secondly, the process of course development including the course materials, activities and assessments are very time consuming. Coupling the LPR system with other systems that automate the course development process from learning object repositories and assessment repositories would be another direction of research which could be valuable to investigate.

Thirdly, other characteristics of learners, besides prior knowledge, could be incorporated in the process of clustering or the learning path selection. For example, learning style and/or learning preference of learners could be used to find similar ex-learners in the clustering component. The learning style can also be considered when selecting the learning concepts in the ACO component. The materials and activities in the learning concepts should match the learners learning style when selecting a new learning path for the learner.

Finally, both FCM and ACO algorithms are iterative and computationally intensive algorithms in nature. In our prototype, learners needed to wait for a couple of seconds for the LPR system to find and present the recommended learning path. With the recent advancement in multi-core architectures, these algorithms can be parallelized to speedup their performance. Multicore processors are shared-memory architectures where the processors are connected to a common memory. Communications among processors are facilitated by a single address space but synchronizations still have to be managed.

The ACO is an inherently parallelizable search technique (Islam et al., 2001). Therefore, as a future work the FCM and ACO components of the system can be parallelized using OpenMP. OpenMP (ope, 2017) is a portable parallel programming model for shared memory multi-processor or multi-core architecture. OpenMP is basically used to convert an existing sequential program into a parallel one by exploiting the parallelization of loops.

Appendix A

Supporting Data

Tables A.1 andA.2 includes our data for the LPR and control groups.

Table A.1: Supporting Data for LPR Group

Pre-assessment	Final Exam	Difference	Completion Time
30	55	25	4948
35	65	30	5606
25	65	40	6060
20	30	10	5270
30	95	65	4540
5	70	65	10927
30	75	45	6889
60	80	20	9949
40	90	50	5924
50	65	15	6639
55	70	15	7049
50	80	30	6471
30	60	30	7887
35	75	40	7859
5	45	40	7235
15	40	25	7456
40	75	35	11066
20	50	30	6265
15	45	30	6710
20	65	45	7100
15	65	50	7199
35	65	30	6350
25	85	60	9116
50	75	25	7400
55	75	20	8264

Table A.2: Supporting Data for Control Group

Pre-assessment	Final Exam	Difference	Completion Time
15	45	30	7570
10	40	30	7372
20	45	25	6980
35	85	50	9983
45	65	20	7406
20	45	25	7210
20	25	5	5562
70	85	15	4836
35	85	50	8936
65	80	15	11560
40	65	25	7214
50	80	30	6882
20	55	35	8230
5	35	30	8441
30	40	10	7055
50	70	20	7064
35	50	15	7937
20	50	30	7910
40	45	5	6464
35	60	25	10929
35	40	5	5298
10	70	60	7629
25	70	45	7684
10	25	15	5606
15	55	40	6461

Appendix B

Source Code

This chapter includes the source code for the LPR system containing Java and jsp.

```
package antsystemLPR;



import java.util.Random;
/*
 *  === Implementation of ant swarm LPR solver. ===
 *
 *
 * == References ==
 * [1] M. Dorigo and L. M. Gambardella. Ant colony system:
a cooperative learning approach to the traveling salesman
problem. IEEE Transactions on evolutionary
computation, 1(1):53{66, 1997.
 *
```

```
 *
 */

public class AntLPR {
    // Algorithm parameters:
    // original amount of trail
    private double c = 1.0;
    // trail preference
    private double alpha = 1;
    // greedy preference
    private double beta = 1;
    // trail evaporation coefficient
    private double evaporation = 0.5;
    // Success value for familiar degree
    private double successvalue = 7.5;
    // probability of pure random selection of the next concept
    private double pr = 0.01;

    // Reasonable number of iterations
    // - results typically settle down by 500
    private int maxIterations = 1000;

    public int n = 0; // # concepts
    public int m = 0; // # ants

    //Visibility information provided by the course designer

    private double graph[][] =
```

```
{{10,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5},
 {5,10,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5},
 {5,5,10,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5},
 {5,5,5,10,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5},
 {5,5,5,5,10,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5},
 {5,5,5,5,5,10,4,5,5,5,5,5,5,5,5,5,5,5,5,5},
 {5,5,5,5,5,5,10,4,5,5,5,5,5,5,5,5,5,5,5,5},
 {5,5,5,5,5,5,5,10,4,5,5,5,5,5,5,5,5,5,5,5},
 {5,5,5,5,5,5,5,5,10,4,5,5,5,5,5,5,5,5,5,5},
 {5,5,5,5,5,5,5,5,5,10,4,5,5,5,5,5,5,5,5,5},
 {5,5,5,5,5,5,5,5,5,5,10,4,5,5,5,5,5,5,5,5},
 {5,5,5,5,5,5,5,5,5,5,5,10,4,5,5,5,5,5,5,5},
 {5,5,5,5,5,5,5,5,5,5,5,5,10,4,5,5,5,5,5,5},
 {5,5,5,5,5,5,5,5,5,5,5,5,5,10,4,5,5,5,5,5},
 {5,5,5,5,5,5,5,5,5,5,5,5,5,5,10,4,5,5,5,5},
 {5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,10,4,5,5,5},
 {5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,10,4,5,5},
 {5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,10,4,5},
 {5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,10,4},
 {4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,10}};

private double trails[][] = null;
private Ant ants[] = null;
private Random rand = new Random();
private double probs[] = null;
//ex-learners familiar degree after pre-assessment
private double antsfd[][]=null;
private int startnode=0;
```

```
//ex-learners familiar degree after final exam
private double antsfdpost[][]=null;
private int currentIndex = 0;

public int[] bestTour;
public double bestTourLength;



public AntLPR(double f[][],double fp[][], int s){
    antsfd=f;
    antsfdpost=fp;
    //startnode is always set to 21 when AntLPR instantiated meaning
    //that the start node is assigned to each ant randomly
    startnode=s;
}

// Ant class. Maintains tour and tabu information.
private class Ant {
    public int tour[] = new int[graph.length];
    // Maintain visited list for concepts, much faster
    // than checking if in tour so far.
    public boolean visited[] = new boolean[graph.length];

    public double myfd[] = new double[graph.length];

    public double mypostfd[] = new double[graph.length];

    public double finalgarde=0;
```

```
public void visitConcept(int concept) {
    tour[currentIndex + 1] = concept;
    visited[concept] = true;
}

public boolean visited(int i) {
    return visited[i];
}

public double tourLength() {
    double length = graph[tour[n - 1]][tour[0]];
    for (int i = 0; i < n - 1; i++) {
        length += graph[tour[i]][tour[i + 1]];
    }

    return length;
}

public void clear() {
    for (int i = 0; i < n; i++)
        visited[i] = false;
}

public void setFamil(double fd[], double fpd[]) {
    myfd=fd;
    mypostfd=fpd;
}
```

```
}

// Read in couse graph .
// Allocates all memory.

public void readGraph() {



    n = graph.length;


    m= antsfd.length;


    // all memory allocations done here
    trails = new double[n][n];
    probs = new double[n];
    ants = new Ant[m];
    for (int j = 0; j < m; j++){
        ants[j] = new Ant();
        ants[j].setFamil(antsfd[j],antsfdpost[j]);

    }
}

// Approximate power function , Math.pow is quite slow.
//
// - 25 times faster
```

```
// - Does not harm results -- not surprising for a stochastic
   algorithm.
public static double pow(final double a, final double b) {
    final int x = (int) (Double.doubleToLongBits(a) >> 32);
    final int y = (int) (b * (x - 1072632447) + 1072632447);
    return Double.longBitsToDouble(((long) y) << 32);
}

// Store in probs array the probability of moving to each concept
// [1] describes how these are calculated.
// In short: ants like to follow stronger and shorter trails more.
private void probTo(Ant ant) {
    int i = ant.tour[currentIndex];

    double denom = 0.0;
    for (int l = 0; l < n; l++)
        if (!ant.visited(l))
            denom += pow(trails[i][l], alpha)
                    * pow(1.0 / graph[i][l], beta);



    for (int j = 0; j < n; j++) {
        if (ant.visited(j)) {
            probs[j] = 0.0;
        } else {
            double numerator = pow(trails[i][j], alpha)
                    * pow(1.0 / graph[i][j], beta);
```

```
                probs[j] = numerator / denom;
            }
        }

    }

    // Given an ant select the next concept based on the probabilities
    // we assign to each concept. With pr probability chooses
    // totally randomly (taking into account tabu list).
    private int selectNextTown(Ant ant) {
        // sometimes just randomly select
        if (rand.nextDouble() < pr) {
            int t = rand.nextInt(n - currentIndex); // random concept
            int j = -1;
            for (int i = 0; i < n; i++) {
                if (!ant.visited(i))
                    j++;
                if (j == t)
                    return i;
            }

        }
        // calculate probabilities for each concept (stored in probs)
        probTo(ant);
        // randomly select according to probs
        double r = rand.nextDouble();
        double tot = 0;
        for (int i = 0; i < n; i++) {
```

```
            tot += probs[i];
            if (tot >= r)
                return i;
        }

        throw new RuntimeException("Not supposed to get here.");
    }

    // Update trails based on ants tours
    private void updateTrails() {
        // evaporation
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                trails[i][j] *= evaporation;



        // each ants contribution
        for (Ant a : ants) {

            // cotribute if the ex-learner was successful in post famil
                degree
            for (int i = 0; i < n - 1; i++) {
                if(a.mypostfd[a.tour[i+1]] >= successvalue)
                    trails[a.tour[i]][a.tour[i + 1]] += a.myfd[a.tour[i
                        +1]];
                else
                    trails[a.tour[i]][a.tour[i + 1]] += 0;
            }
```

```
            if(a.mypostfd[a.tour[0]]>= successvalue)
                trails[a.tour[n - 1]][a.tour[0]] += a.myfd[a.tour[0]];
            else
                trails[a.tour[n - 1]][a.tour[0]] += 0;
        }
    }

    // Choose the next concept for all ants
    private void moveAnts() {
        // each ant follows trails...
        while (currentIndex < n - 1) {
            for (Ant a : ants)
                a.visitConcept(selectNextTown(a));
            currentIndex++;
        }
    }

    // m ants with random start concept
    private void setupAnts() {
        currentIndex = -1;
        for (int i = 0; i < m; i++) {
            ants[i].clear(); // faster than fresh allocations.
            //if startnode==21 means that the first concept is randomly
                assigned
            if(startnode==21)
                ants[i].visitConcept(rand.nextInt(n));
            else
                ants[i].visitConcept(startnode);
```

```
        }
        currentIndex++;

    }

    private void updateBest() {
        if (bestTour == null) {
            bestTour = ants[0].tour;
            bestTourLength = ants[0].tourLength();
        }
        for (Ant a : ants) {
            if (a.tourLength() < bestTourLength) {
                bestTourLength = a.tourLength();
                bestTour = a.tour.clone();
            }
        }
    }

    public static String tourToString(int tour[]) {
        String t = new String();
        for (int i : tour)
            t = t + " " + i;
        return t;
    }

    public int[] solve() {
        // clear trails
        for (int i = 0; i < n; i++)
```

```
            for (int j = 0; j < n; j++)
                trails[i][j] = c;

        int iteration = 0;
        // run for maxIterations
        // preserve best tour
        while (iteration < maxIterations) {
            setupAnts();
            moveAnts();
            updateTrails();
            updateBest();
            //System.out.println("Best tour length: " + (bestTourLength
                ));
            //System.out.println("Best tour:" + tourToString(bestTour));
            iteration++;
        }

        return bestTour.clone();
    }


}

/*
 * To change this license header, choose License Headers in Project
    Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
```

```
 */
package ServletPackage;

import antsystemLPR.AntLPR;
import cmean.CMean;
import java.io.IOException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author mehdiniknam
 */
@WebServlet(name = "LPServlet", urlPatterns = {"/LPServlet"})
```

```
public class LPServlet extends HttpServlet {



    private int exlearnerNum = 800;
    private int originalNum = 40;
    private int conceptNum = 20;
    private int questiontNum = 20;
    private int clusterNum = 4;
    // relevance matrix between concepts and questions
    private double rel [][]=
    {{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
     {0.2, 1, 0.2, 0, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
     {0.2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.1, 0, 0, 0, 0, 0},
     {0, 0, 0.1, 1, 0, 0, 0, 0, 0, 0.2, 0, 0.1, 0, 0, 0, 0, 0, 0, 0, 0},
     {0, 0, 0.1, 0, 1, 0, 0, 0.4, 0.2, 0, 0.1, 0, 0, 0.2, 0, 0, 0, 0.1,
        0.1, 0.1},
     {0, 0, 0.1, 0, 0, 1, 0, 0, 0, 0, 0.2, 0, 0, 0, 0, 0.2, 0.1, 0.2, 0,
         0},
     {0, 0.5, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
     {0, 0.1, 0, 0, 0.4, 0, 0.1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.2,
        0.2},
     {0, 0, 0, 0, 0.2, 0, 0, 0.2, 1, 0.6, 0, 0, 0, 0, 0, 0, 0, 0, 0.2,
        0.1},
     {0, 0, 0.2, 0.2, 0, 0, 0, 0.1, 0.6, 1, 0, 0.3, 0, 0, 0, 0, 0, 0,
        0.1, 0},
     {0, 0, 0, 0, 0.1, 0.2, 0, 0.1, 0, 0, 1, 0, 0, 0, 0, 0, 0.2, 0.7,
        0.1, 0},
     {0, 0, 0, 0.2, 0, 0, 0, 0, 0.1, 0.2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0},
```

```
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0.1, 0.2, 0, 0, 0},
  {0, 0.1, 0.2, 0, 0, 0, 0.2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0},
  {0, 0.1, 0, 0, 0, 0.2, 0.2, 0, 0, 0, 0, 0, 0, 0.1, 0.1, 1, 0.2, 0,
     0, 0},
  {0, 0, 0, 0, 0, 0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0},
  {0, 0, 0, 0, 0.1, 0.1, 0, 0.2, 0.2, 0, 0.7, 0, 0, 0, 0, 0, 0, 1,
     0.2, 0},
  {0, 0, 0, 0.1, 0, 0, 0, 0.2, 0.2, 0.2, 0, 0, 0, 0, 0, 0, 0, 0, 1,
     0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1}};
//links to the start of each concept in LMS
private String [] cUrl={
        "https://universityofmanitobatest.desire2learn.com/d2l/le/
            content/172516/viewContent/633179/View",
        "https://universityofmanitobatest.desire2learn.com/d2l/le/
            content/172516/viewContent/633181/View",
    "https://universityofmanitobatest.desire2learn.com/d2l/le/
        content/172516/viewContent/633187/View",
    "https://universityofmanitobatest.desire2learn.com/d2l/le/
        content/172516/viewContent/633190/View",
    "https://universityofmanitobatest.desire2learn.com/d2l/le/
        content/172516/viewContent/633191/View",
    "https://universityofmanitobatest.desire2learn.com/d2l/le/
        content/172516/viewContent/633194/View",
    "https://universityofmanitobatest.desire2learn.com/d2l/le/
        content/172516/viewContent/633201/View",
    "https://universityofmanitobatest.desire2learn.com/d2l/le/
```

```
    content/172516/viewContent/633202/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633207/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633213/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633231/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633238/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633277/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633245/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633251/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633255/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633258/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633261/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633279/View",
"https://universityofmanitobatest.desire2learn.com/d2l/le/
    content/172516/viewContent/633283/View"

};
```

```
/**
 * Processes requests for both HTTP <code>GET</code> and <code>POST
    </code>
 * methods.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
protected void processRequest(HttpServletRequest request,
   HttpServletResponse response)
       throws ServletException, IOException {

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
//Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
```

```
        response)
            throws ServletException, IOException {

        String userid = request.getParameter("uid");
        request.setAttribute("userid", userid);
        request.getRequestDispatcher("/index.jsp").forward(request,
            response);

    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
        //processRequest(request, response);

        String userID1=request.getParameter("userid");
        String userID=null;
        //clean the user id coming within the post request
        userID=userID1.replaceAll("(?i)<script.*?>.*?</script.*?>", "")
```

```
        // case 1
    .replaceAll("(?i)<.*?javascript:.*?>.*?</.*?>", "") // case 2
    .replaceAll("(?i)<.*?\\s+on.*?>.*?</.*?>", "");
    //to show if learner completed preassessment
    int preassessment=0;
    //to storescore of preassessment of learner
    int [] result= new int [questiontNum];



    //Get the score of the pre-assessment of the current learner
       from DB
    //
       ////////////////////////////////////////////////////////////////////////

    DbManager db=new DbManager();
    Connection conn=null;
    conn=db.getConnection();
    Statement myStmt=null;
    ResultSet myRs=null;
    try {
        conn=db.getConnection();
        myStmt =conn.createStatement();
        myRs=myStmt.executeQuery("select    max(aid), qid, score_u, 
            uid from savsoft_answers where (qid between 17 and 36) 
            and uid = " +userID+" Group BY qid");
        int i=0;
        while (myRs.next()){
            result[i]=myRs.getInt("score_u");
```

```
            i++;
        }
        if (i<1) preassessment=0;
        else preassessment=1;
    } catch (SQLException ex) {
        Logger.getLogger(LPServlet.class.getName()).log(Level.SEVERE
            , null, ex);
    } finally{
    // close everything
        try { myRs.close(); } catch (Exception e) { /* ignored */ }
        try { myStmt.close(); } catch (Exception e) { /* ignored */
            }
        db.closeConnection(conn);
    }
    //
        ///////////////////////////////////////////////////////////////////////


    if(preassessment==1) { //if preassessment was comoleted

        //calculate family degree for current learner
        double[] learnerfd = new double[conceptNum];
        double soorat=0;
        double makhraj=0;
        for(int i=0;i<conceptNum;i++){
            for(int q=0;q<questiontNum;q++){
                soorat+=result[q]* rel[q][i];
                makhraj+=rel[q][i];
```

```
        }
        learnerfd[i]=10*soorat/makhraj;
        soorat=0;
        makhraj=0;
    }
    //Store familiar degree of learner in DB that we did not
        implement
    //in this prototype



    //Get the familiar degree of ex-learners from DB
    //
        ////////////////////////////////////////////////////////////////////

    double afd[][]=new double [originalNum][conceptNum];
    double afdp[][]=new double [originalNum][conceptNum];
    try {
        conn=db.getConnection();
        myStmt =conn.createStatement();
        myRs=myStmt.executeQuery("select          exlid, cid, 
            prefamil, postfamil from famildegree");
        int ex=0;
        int c=0;
        int i=0;
        while (myRs.next()){
            c=myRs.getInt("cid");
            ex=myRs.getInt("exlid");
```

```
            afd[ex][c]=myRs.getDouble("prefamil");
            afdp[ex][c]=myRs.getDouble("postfamil");
            i++;
        }

    } catch (SQLException ex) {
        Logger.getLogger(LPServlet.class.getName()).log(Level.
            SEVERE, null, ex);
    } finally{
    // close everything
        try { myRs.close(); } catch (Exception e) { /* ignored
            */ }
        try { myStmt.close(); } catch (Exception e) { /* ignored
            */ }
        db.closeConnection(conn);
    }
    //
        ///////////////////////////////////////////////////////////////////////


    double[][] afdf = new double[exlearnerNum][conceptNum];
    double[][] afdfp = new double[exlearnerNum][conceptNum];

    //Replicte the pre familiar degree of original ex-learners
        for exlearnerNum
    for(int i=0;i<exlearnerNum;i++)
        System.arraycopy(afd[i % originalNum], 0, afdf[i], 0,
            conceptNum);
```

```
//Replicte the  post familiar degree of original ex-learners
     for exlearnerNum
for(int i=0;i<exlearnerNum;i++)
     System.arraycopy(afdp[i % originalNum], 0, afdfp[i], 0,
         conceptNum);

//Clustering
//
    //////////////////////////////////////////////////////////////////

//add the familar degree of the current learner to the data
    for
//clustering along with famil degree of ex-learners

double[][] clustdata = new double[exlearnerNum+1][conceptNum
    ];
//copy the ex-learners famil degree to cluster data
for(int i=0;i<exlearnerNum;i++)
     System.arraycopy(afdf[i ], 0, clustdata[i], 0,
         conceptNum);
//add the current learners famil degree to cluster data
clustdata[exlearnerNum]=learnerfd;

// run clustering
CMean clustering=new CMean(clustdata,clusterNum);
```

```
//get best two clusters for the data
int[][] bt=new int [exlearnerNum+1][2];

bt=clustering.getBestTwo();

//get the first cluster and second cluster of the current
    learner
int c1Numforl=bt[exlearnerNum][0];
int c2Numforl=bt[exlearnerNum][1];

//get the number of exlearners in the same cluster as the
    current learner
int c1Numforxl=clustering.getc1Count()[0][c1Numforl-1];
int c2Numforxl=clustering.getc2Count()[0][c1Numforl-1];
// assign the exlearners fd and post fd in the first choice
    cluster
double[][] fdc1 = new double[c1Numforxl][conceptNum];
double[][] fdpc1 = new double[c1Numforxl][conceptNum];
int j=0;
for(int i=0;i<exlearnerNum;i++)
    if(bt[i][0]==c1Numforl){
        fdc1[j]= afdf[i];
        fdpc1[j]= afdfp[i];
        j++;
    }
```

```
//
    ////////////////////////////////////////////////////////////////////

//Run ACO
//
    ////////////////////////////////////////////////////////////////////

int[][] lp=new int[2][conceptNum];
// run ACO to find the path for the first choice cluster
AntLPR antlpr = new AntLPR(fdc1,fdpc1,21);
antlpr.readGraph();
lp[0]=antlpr.solve();

if(c2Numforl!=0){
// assign the exlearners pre fd and post fd in the second
    choice cluster
    double[][] fdc2 = new double[c2Numforxl][conceptNum];
    double[][] fdpc2 = new double[c2Numforxl][conceptNum];
    int k=0;
    for(int i=0;i<exlearnerNum;i++)
        if(bt[i][0]==c2Numforl){
            fdc2[k]= afdf[i];
            fdpc2[k]= afdfp[i];
            k++;
        }
    // run ACO to find the path for the second choice
```

```
        cluster
    AntLPR antlpr2 = new AntLPR(fdc1,fdpc1,21);
    antlpr2.readGraph();
    lp[1]=antlpr2.solve();



}
//
    ///////////////////////////////////////////////////////////////////////


//Prepare the learning paths to show to learner
//
    ///////////////////////////////////////////////////////////////////////



//remove completed concepts
//first get the list of visited concepts for the learner
    from DB

List<String> visitedC=new ArrayList<String>();

try {
    conn=db.getConnection();
    myStmt =conn.createStatement();
    myRs=myStmt.executeQuery("select          d2luid,cid from
        userconcept where d2luid = " +userID);
```

```
        while (myRs.next()){
            visitedC.add("␣"+myRs.getInt("cid"));


        }
    } catch (SQLException ex) {
        Logger.getLogger(LPServlet.class.getName()).log(Level.
            SEVERE, null, ex);
    } finally{
    // close everything
        try { myRs.close(); } catch (Exception e) { /* ignored
            */ }
        try { myStmt.close(); } catch (Exception e) { /* ignored
            */ }
        db.closeConnection(conn);
    }



    //compare the list of visited concepts with lp to remove
        visted concept from lp

    String [] firstlp=new String[conceptNum];
    for(int i=0;i<conceptNum;i++)
        firstlp[i]="␣"+lp[0][i];

    List<String> finalLp=new ArrayList<String>();

    for(int vc=0;vc<conceptNum;vc++){
        if(!visitedC.contains(firstlp[vc]))
```

```
          finalLp.add(firstlp[vc]);


    }


    /// insert the first concept into db
    if(finalLp.size()!=0){
        String temp=finalLp.get(0);
        temp=temp.substring(1);
        int concept=Integer.parseInt(temp);
        String timeStamp = new SimpleDateFormat("yyyy/MM/dd HH:
            mm:ss").format(Calendar.getInstance().getTime());
        try {
             conn=db.getConnection();
            myStmt =conn.createStatement();
            String sql="insert into userconcept "
                    +"(d2luid, cid, gendate, firstcluster, 
                        secondcluster, lp) "
                    +"values ('"+userID+"',"+temp+",'"+timeStamp
                        +"', "+bt[exlearnerNum][0]+", "+bt[
                        exlearnerNum][1]+", '"+antlpr.
                        tourToString(lp[0])+"')";
            myStmt.executeUpdate(sql);
        } catch (SQLException ex) {
            Logger.getLogger(LPServlet.class.getName()).log(
                Level.SEVERE, null, ex);
        } finally{
        // close everything
            try { myStmt.close(); } catch (Exception e) { /*
```

```
                ignored */ }
            db.closeConnection(conn);
        }
    }
    //
        ////////////////////////////////////////////////////////////////////


    // Present the learning paths to current learner
    //
        ////////////////////////////////////////////////////////////////////

    int s=0;
    if(request.getParameter("subindex")!=null){
        //if there is still concept to show list the concept for
             the learner
        if(!finalLp.isEmpty()){
            int i=0;
            while(i<finalLp.size()){
                s=i+1;
                request.setAttribute("lp"+s, finalLp.get(i));
                i++;
        }
        String withoutFirstCharacter = finalLp.get(0).substring
            (1);
        request.setAttribute("lp1link", cUrl[Integer.valueOf(
            withoutFirstCharacter)]);
        for(;i<conceptNum;i++){
```

```
          s=i+1;
          request.setAttribute("lp"+s, "␣");
        }
      }else{//if the learner completed all concepts
        request.setAttribute("lp1", "Congradulations,␣you␣have␣
            completed␣the␣course!␣Please␣click␣here␣to␣complete␣
            the␣Final␣exam␣");
        for(int i=1;i<conceptNum;i++){
          s=i+1;
          request.setAttribute("lp"+s, "␣");
        }
        request.setAttribute("lp1link", "http://
            timemanagementmagazine.com/lpr/savsoftquiz_v3.0-
            master/index.php/quiz/quiz_detail/6");
      }
        request.getRequestDispatcher("/lprpage.jsp").forward(
            request, response);
        }
    } else{//if the learner has not yet completed the preassessment
      int s=0;
      request.setAttribute("lp1", "You␣have␣not␣completed␣the␣Pre-
          Assessment␣quiz.␣Please␣click␣here␣and␣complete␣the␣Pre-
          Assessment␣quiz␣and␣then␣click␣Refresh!");
      for(int i=1;i<conceptNum;i++){
          s=i+1;
          request.setAttribute("lp"+s, "␣");
        }
        request.setAttribute("lp1link", "http://
```

```
                    timemanagementmagazine.com/lpr/savsoftquiz_v3.0-
                    master/index.php/quiz/quiz_detail/5");
            request.getRequestDispatcher("/lprpage.jsp").forward(request
                , response);
        }
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}

/*
 * To change this license header, choose License Headers in Project
     Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package cmean;

import cern.colt.matrix.DoubleMatrix2D;
```

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Arrays;
import java.util.Random;
import org.apache.commons.math3.random.MersenneTwister;
/**
 *
 * @author mehdiniknam
 */
public class CMean {

    private double data[][];
    private double clusts[][];
    private int bestTwo[][];
    private double clustCoef;
    private double clustEnt;
    private int c1Count[][];
    private int c2Count[][];

    public CMean(double learnerFD[][], int cNum){
        data=learnerFD;

        int maxRow=data.length;
        int maxCol=data[0].length;

        clusts=new double[maxRow][cNum];
```

```
bestTwo=new int [maxRow][2];



c1Count=new int[1][cNum];

c2Count=new int[1][cNum];

DoubleMatrix2D cmeandata=cern.colt.matrix.DoubleFactory2D.dense.
    make(maxRow,maxCol);



for ( int row = 0; row < maxRow; row++) {
    for ( int column = 0; column < maxCol; column++) {
        cmeandata.setQuick(row, column, data[row][column]);
    }
}

// run fcm clustering
FuzzyCMeans fcm = new FuzzyCMeans();
fcm.setRandomGenerator(new MersenneTwister(123456789));
fcm.cluster(cmeandata, cNum);

DoubleMatrix2D partition = fcm.getPartition();

clustCoef=Partition.partitionCoefficient(partition);
clustEnt=Partition.partitionEntropy(partition);
```

```
for(int row=0;row<maxRow;row++){
        clusts[row][0]= partition.getQuick(row, 0);
        clusts[row][1]= partition.getQuick(row, 1);
        clusts[row][2]= partition.getQuick(row, 2);
        clusts[row][3]= partition.getQuick(row, 3);
}

// find the best two clusters for learners
int first=0;
int second=0;
for(int row=0;row<maxRow;row++){
     if(clusts[row][0]>= clusts[row][1]){
          first=0;
          second=1;
     }else {
          first=1;
          second=0;
     }
     for(int j=0;j<cNum;j++){

          if (clusts[row][j]>clusts[row][first]){
               second=first;
               first=j;

          }else if (clusts[row][j]>clusts[row][second] && j!=first
               )
               second=j;
```

```
            }

            bestTwo[row][0]=first+1;
            if(clusts[row][first]<=clusts[row][second]+(5*clusts[row][
                second]/100))
                  bestTwo[row][1]=second+1;
            else  bestTwo[row][1]=0;

        }

        for(int row=0;row<maxRow-1;row++)
            for (int j=0;j<cNum;j++)
                if(bestTwo[row][0]==j+1)
                    c1Count[0][j]++;

        for(int row=0;row<maxRow-1;row++)
            for (int j=0;j<cNum;j++)
                if(bestTwo[row][1]==j+1)
                    c2Count[0][j]++;


    }

    public double [][] getClusters(){
        return clusts;
    }

    public double getCcoef(){
```

```
        return clustCoef;
    }

    public double getCentropy(){
        return clustEnt;
    }

    public int[][] getBestTwo(){
        return bestTwo;

    }

    public int[][] getc1Count(){
        return c1Count;

    }
    public int[][] getc2Count(){
        return c2Count;

    }

}

/*
 * This file is part of the Trickl Open Source Libraries.
 *
 * Trickl Open Source Libraries - http://open.trickl.com/
 *
```

```
 * Copyright (C) 2011 Tim Gee.
 *
 * Trickl Open Source Libraries are free software: you can redistribute
     it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * Trickl Open Source Libraries are distributed in the hope that it will
      be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this project.  If not, see <http://www.gnu.org/licenses/>.
 */
package cmean;

import cern.colt.matrix.DoubleMatrix2D;

public interface ClusterAlgorithm {

   void cluster(DoubleMatrix2D data, int clusters);

   DoubleMatrix2D getPartition();
}
```

```
/*
 * This file is part of the Trickl Open Source Libraries.
 *
 * Trickl Open Source Libraries - http://open.trickl.com/
 *
 * Copyright (C) 2011 Tim Gee.
 *
 * Trickl Open Source Libraries are free software: you can redistribute
     it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * Trickl Open Source Libraries are distributed in the hope that it will
     be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this project.  If not, see <http://www.gnu.org/licenses/>.
 */
package cmean;

import cern.colt.matrix.DoubleMatrix2D;
import cern.colt.matrix.doublealgo.Statistic;
import cern.colt.matrix.doublealgo.Statistic.VectorVectorFunction;
import cern.colt.matrix.impl.DenseDoubleMatrix2D;
```

```java
import cern.colt.matrix.impl.SparseDoubleMatrix2D;
import org.apache.commons.math3.random.MersenneTwister;
import org.apache.commons.math3.random.RandomGenerator;


public class FuzzyCMeans implements ClusterAlgorithm {

  private DoubleMatrix2D means;
  private DoubleMatrix2D partition;
  private double fuzzification = 2.0;
  private double epsilon = 1e-7;
  private int maxIterations = 1000;
  private RandomGenerator randomGenerator = new MersenneTwister();
  private PartitionGenerator partitionGenerator = new
    FuzzyRandomPartitionGenerator();
  private VectorVectorFunction distanceMeasure = Statistic.EUCLID;

  public FuzzyCMeans() {
  }

  public void cluster(DoubleMatrix2D data, int clusters) {
    int n = data.rows(); // Number of features
    int p = data.columns(); // Dimensions of features

    partition = new SparseDoubleMatrix2D(n, clusters);
    partitionGenerator.setRandomGenerator(randomGenerator);
    partitionGenerator.generate(partition);
```

```
means = new DenseDoubleMatrix2D(p, clusters);

// Begin the main loop of alternating optimization
double stepSize = epsilon;
for (int itr = 0; itr < maxIterations && stepSize >= epsilon; ++
    itr) {
    // Get new prototypes (v) for each cluster using weighted
        median
    for (int k = 0; k < clusters; k++) {

        for (int j = 0; j < p; j++) {
            double sumWeight = 0;
            double sumValue = 0;

            for (int i = 0; i < n; i++) {
                double Um = Math.pow(partition.getQuick(i, k),
                    fuzzification);
                sumWeight += Um;
                sumValue += data.getQuick(i, j) * Um;
            }

            means.setQuick(j, k, sumValue / sumWeight);
        }
    }

    // Calculate distance measure d:
    DoubleMatrix2D distances = new DenseDoubleMatrix2D(n, clusters)
        ;
```

```
for (int k = 0; k < clusters; k++) {
    for (int i = 0; i < n; i++) {
        // Euclidean distance calculation
        double distance = distanceMeasure.apply(means.viewColumn(
            k), data.viewRow(i));
        distances.setQuick(i, k, distance);
    }
}

// Get new partition matrix U:
stepSize = 0;
for (int k = 0; k < clusters; k++) {
    for (int i = 0; i < n; i++) {
        double u = 0;

        if (distances.getQuick(i, k) == 0) {
            // Handle this awkward case
            u = 1;
        } else {
            double sum = 0;
            for (int j = 0; j < clusters; j++) {
                // Exact analytic solution given by Lagrange
                    multipliers
                sum += Math.pow(distances.getQuick(i, k) /
                    distances.getQuick(i, j),
                                1.0 / (fuzzification - 1.0));
            }
            u = 1 / sum;
```

```
                }

                double u0 = partition.getQuick(i, k);
                partition.setQuick(i, k, u);

                // Stepsize is max(delta(U))
                if (u - u0 > stepSize) {
                    stepSize = u - u0;
                }
            }
        }
    }
}

public DoubleMatrix2D getMeans() {
    return means;
}

public DoubleMatrix2D getPartition() {
    return partition;
}

public double getFuzzification() {
    return fuzzification;
}

public void setFuzzification(double fuzzification) {
    this.fuzzification = fuzzification;
```

```
    }

    public double getEpsilon() {
        return epsilon;
    }

    public void setEpsilon(double epsilon) {
        this.epsilon = epsilon;
    }

    public int getMaxIterations() {
        return maxIterations;
    }

    public void setMaxIterations(int maxIterations) {
        this.maxIterations = maxIterations;
    }

    public RandomGenerator getRandomGenerator() {
        return randomGenerator;
    }

    public void setRandomGenerator(RandomGenerator random) {
        this.randomGenerator = random;
    }
}

/*
```

```
 * This file is part of the Trickl Open Source Libraries.
 *
 * Trickl Open Source Libraries - http://open.trickl.com/
 *
 * Copyright (C) 2011 Tim Gee.
 *
 * Trickl Open Source Libraries are free software: you can redistribute
     it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * Trickl Open Source Libraries are distributed in the hope that it will
     be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this project.  If not, see <http://www.gnu.org/licenses/>.
 */
package cmean;

import cern.colt.matrix.DoubleMatrix2D;
import org.apache.commons.math3.random.MersenneTwister;
import org.apache.commons.math3.random.RandomGenerator;

public class FuzzyRandomPartitionGenerator implements PartitionGenerator
```

```
{

private RandomGenerator randomGenerator;

public FuzzyRandomPartitionGenerator() {
    randomGenerator = new MersenneTwister();
}

@Override
public void generate(DoubleMatrix2D partition) {
    for (int i = 0; i < partition.rows(); ++i) {
        // Randomise
        double sum = 0;
        for (int k = 0; k < partition.columns(); ++k) {
            double u = randomGenerator.nextDouble();
            partition.setQuick(i, k, u);
            sum += u;
        }

        // Normalise the weights
        for (int k = 0; k < partition.columns(); ++k) {
            partition.setQuick(i, k, partition.getQuick(i, k) / sum);
        }
    }
}

public RandomGenerator getRandomGenerator() {
    return randomGenerator;
```

```
    }

    @Override
    public void setRandomGenerator(RandomGenerator random) {
        this.randomGenerator = random;
    }
}

/*
 * This file is part of the Trickl Open Source Libraries.
 *
 * Trickl Open Source Libraries - http://open.trickl.com/
 *
 * Copyright (C) 2011 Tim Gee.
 *
 * Trickl Open Source Libraries are free software: you can redistribute
     it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * Trickl Open Source Libraries are distributed in the hope that it will
     be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
```

```
 * along with this project.  If not, see <http://www.gnu.org/licenses/>.
 */
package cmean;

import cern.colt.matrix.DoubleMatrix1D;
import cern.colt.matrix.DoubleMatrix2D;
import cern.colt.matrix.impl.DenseDoubleMatrix1D;
import cern.colt.matrix.impl.DenseDoubleMatrix2D;
import cern.jet.math.Functions;

public class Partition {

   public static double partitionCoefficient(DoubleMatrix2D partition) {
      double coefficient = 0;

      for (int i = 0; i < partition.rows(); i++) {
         for (int j = 0; j < partition.columns(); j++) {
            coefficient += Math.pow(partition.getQuick(i, j), 2.0);
         }
      }

      coefficient /= partition.rows();

      return coefficient;
   }

   public static double partitionEntropy(DoubleMatrix2D partition) {
      double entropy = 0;
```

```
    for (int i = 0; i < partition.rows(); i++) {
      for (int j = 0; j < partition.columns(); j++) {
        double membership = partition.getQuick(i, j);
        entropy -= membership > 0 ? membership * Math.log(membership
            ) : 0;
      }
    }

    entropy /= partition.rows();

    return entropy;
  }

  public static double xieBeniIndex(DoubleMatrix2D U, DoubleMatrix2D V,
      DoubleMatrix2D X, double m) {
    int n = X.rows(); // Number of features
    int p = X.columns(); // Dimensions of features
    int clusters = U.columns(); // Number of features

    double compactness = 0;
    for (int k = 0; k < clusters; k++) {
      // calculate the distance vector for this cluster
      for (int i = 0; i < n; i++) {
        double Um = Math.pow(U.getQuick(i, k), m);

        DoubleMatrix1D dev_i = new DenseDoubleMatrix1D(p);
```

```
        dev_i.assign(V.viewRow(k));
        dev_i.assign(X.viewRow(i), Functions.minus);

        // TODO prototype eigenvectors should be an input
        DoubleMatrix2D A = new DenseDoubleMatrix2D(p, p);
        for (int j = 0; j < p; ++j) {
            A.setQuick(j, j, 1);
        }

        // Euclidean distance using hyperellipsoid clusters
        DoubleMatrix1D Adev_i = new DenseDoubleMatrix1D(p);
        A.zMult(dev_i, Adev_i);

        compactness += dev_i.zDotProduct(Adev_i) * Um;
    }
}
compactness /= n;

double minimal_separation = 0;
for (int k = 0; k < clusters; k++) {
    for (int k2 = 0; k2 < clusters; k2++) {
        // Euclidean distance
        if (k != k2) {
            DoubleMatrix1D dev_k = new DenseDoubleMatrix1D(p);
            dev_k.assign(V.viewRow(k));
            dev_k.assign(X.viewRow(k2), Functions.minus);

            // Calculate magnitude of dev_k
```

```
                double separation = dev_k.aggregate(Functions.plus,
                    Functions.square);

                if (minimal_separation == 0) {
                    minimal_separation = separation;
                } else {
                    minimal_separation = Math.min(separation,
                        minimal_separation);
                }
            }
        }
    }

    return compactness / minimal_separation;
}

public static DoubleMatrix2D inclusion(DoubleMatrix2D U) {
    DoubleMatrix2D I = new DenseDoubleMatrix2D(U.columns(), U.columns
        ());

    for (int i = 0; i < I.columns(); ++i) {
        for (int j = 0; j < I.rows(); ++j) {
            if (i == j) {
                I.setQuick(i, j, 1);
            } else if (i < j) {
                DoubleMatrix1D umin = new DenseDoubleMatrix1D(U.columns()
                    );
                umin.assign(U.viewColumn(i));
```

```
                    umin.assign(U.viewColumn(j), Functions.min);

                    I.setQuick(i, j, umin.zSum()
                              / Math.min(U.viewColumn(i).zSum(),
                              U.viewColumn(j).zSum()));
                } else {
                    I.setQuick(i, j, I.getQuick(j, i));
                }
            }
        }

        return I;
    }
}
```

```
<%--
     Document    : index
     Author      : mehdiniknam
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
     <head>
          <meta http-equiv="Content-Type" content="text/html;␣charset=UTF
               -8">
          <title>JSP Page</title>
          <style>
```

```
            body{
                width:90%;
                font-family: "Comic Sans MS", cursive, sans-serif;
                color:#2F4F4F;
                margin-left: 20px;
            }
            td{text-align: center;}
            a{color:#0000CD;}

            a:hover{color:#DC143C;}
        </style>
    </head>
    <body>
        <p>Please click to get your recommended learning path</p>
        <form action="LPServlet" method="post" style="text-align: center
            ">
            <input type="hidden" name="userid" value=${userid} />

            <input name="subindex" type="submit" value="Get Learning
                Path"/>
        </form>
    </body>
</html>

<%--
    Document   : lprpage

    Author     : mehdiniknam
```

```
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF
            -8">
        <title>JSP Page</title>
        <style>
            body{
                width:90%;
                font-family: "Comic Sans MS", cursive, sans-serif;
                color:#2F4F4F;
                margin-left: 20px;
            }
            td{text-align: center;}
            a{color:#0000CD;}

            a:hover{color:#DC143C;}
        </style>
    </head>
    <body>

        <p> Your recommended learning path (please click the first
            concept in the list to start)</p>
        <table style="margin-left: 50px; border: 1px solid black;">
```

```
<tr><th style="border-bottom:1px␣solid␣black;␣">Learning
    Concepts</th></tr>
<tr><td style="background-color:␣yellow;" name="lp1" style="
    "><a name="lp1link" target="_blank" href= <%=request.
    getAttribute("lp1link") %>><%=request.getAttribute("lp1"
    ) %></a></td></tr>
<tr><td name="lp2"> <%=request.getAttribute("lp2") %></td></
    tr>
<tr><td name="lp3"> <%=request.getAttribute("lp3") %></td></
    tr>
<tr><td name="lp4"> <%=request.getAttribute("lp4") %></td></
    tr>
<tr><td name="lp5"> <%=request.getAttribute("lp5") %></td></
    tr>
<tr><td name="lp6"> <%=request.getAttribute("lp6") %></td></
    tr>
<tr><td name="lp7"> <%=request.getAttribute("lp7") %></td></
    tr>
<tr><td name="lp8"> <%=request.getAttribute("lp8") %></td></
    tr>
<tr><td name="lp9"> <%=request.getAttribute("lp9") %></td></
    tr>
<tr><td name="lp10"> <%=request.getAttribute("lp10") %></td>
    </tr>
<tr><td name="lp11"> <%=request.getAttribute("lp11") %></td>
    </tr>
<tr><td name="lp12"> <%=request.getAttribute("lp12") %></td>
    </tr>
```

```
            <tr><td name="lp13"> <%=request.getAttribute("lp13") %></td>
              </tr>
            <tr><td name="lp14"> <%=request.getAttribute("lp14") %></td>
              </tr>
            <tr><td name="lp15"> <%=request.getAttribute("lp15") %></td>
              </tr>
            <tr><td name="lp16"> <%=request.getAttribute("lp16") %></td>
              </tr>
            <tr><td name="lp17"> <%=request.getAttribute("lp17") %></td>
              </tr>
            <tr><td name="lp18"> <%=request.getAttribute("lp18") %></td>
              </tr>
            <tr><td name="lp19"> <%=request.getAttribute("lp19") %></td>
              </tr>
            <tr><td name="lp20"> <%=request.getAttribute("lp20") %></td>
              </tr>



        </table>



    </body>
</html>
```

Bibliography

Brightspace by D2L developer platform. `http://docs.valence.desire2learn.com/`, 2017. Accessed: 2017-04-25.

A Tutorial on Clustering Algorithms fuzzy c-means clustering. `https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html`, 2017. Accessed: 2017-04-25.

OpenMP the openmp api specification for parallel programming. `http://www.openmp.org/`, 2017. Accessed: 2017-04-25.

Savsoft quiz tool. `http://savsoftquiz.com/`, 2017. Accessed: 2017-04-25.

Udacity free online classes and nanodegrees. `https://www.udacity.com/`, 2017. Accessed: 2017-04-25.

G. Acampora, M. Gaeta, and V. Loia. Hierarchical optimization of personalized experiences for e-learning systems through evolutionary models. *Neural Computing and Applications*, 20(5):641–657, 2011.

S. Al-Muhaideb and M. E. B. Menai. Evolutionary computation approaches to the curriculum sequencing problem. *Natural Computing*, 10(2):891–920, 2011.

M. Ally. Foundations of educational theory for online learning. *Theory and practice of online learning*, 2:15–44, 2008.

I. A. Alshalabi. *An automated adaptive mobile learning system using optimal shortest path algorithms*. PhD thesis, UNIVERSITY OF BRIDGEPORT, 2016.

F. Arshard. Knowledge based learning advisors. In *Proceedings of International Conference on Technology and Education*, pages 213–217, Edinburgh, UK, 1989.

A. Barr and R. Alkinson. The computer as a tutorial laboratory: The stanford bip project. *Man-Machine Studies*, 8:567–596, 1976.

J. Biggs. Enhancing teaching through constructive alignment. *Higher education*, 32 (3):347–364, 1996.

Z. Cebeci and F. Yildiz. Comparison of k-means and fuzzy c-means algorithms on different cluster structures. *Agráinformatika/Journal of Agricultural Informatics*, 6(3):13–23, 2015.

U. Chandrasekhar and P. R. P. Naga. Recent trends in ant colony optimization and data clustering: A brief survey. In *Intelligent Agent and Multi-Agent Systems (IAMA), 2011 2nd International Conference on*, pages 32–36. IEEE, 2011.

Y.-J. Chang, C.-C. FTsai, and C.-P. Chu. A personalized course composition mechanism based on pso. In *Paper presented at TWELF08*, 2008.

C.-P. Chu, Y.-C. Chang, and C.-C. Tsai. Pc2pso: personalized e-course composition based on particle swarm optimization. *Applied Intelligence*, 34(1):141–154, 2011.

D. Cunningham and T. Duffy. Constructivism: Implications for the design and delivery of instruction. *Handbook of research for educational communications and technology*, 51:170–198, 1996.

Dake. Knapsack problem resolved using ants. `https://commons.wikimedia.org/wiki/File:Knapsack_ants.svg)`, note = Creative Commons Attribution-Share Alike 2.5 Generic Accessed: 2017-04-25, 2006.

M. Dorigo and L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66, 1997.

M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

M. P. Driscoll. Psychology of learning for instruction. 2005.

A. Dutt, S. Aghabozrgi, M. A. B. Ismail, and H. Mahroeian. Clustering algorithms applied in educational data mining. *International Journal of Information and Electronics Engineering*, 5(2):112, 2015.

R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995.

A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, 2(3):267–279, 2014.

R. Farzan and P. Brusilovsky. Social navigation support in e-learning: What are real footprints? 2005.

W. Gao. Improved ant colony clustering algorithm and its performance study. *Intell. Neuroscience*, 2016:19:19–19:19, Jan. 2016. ISSN 1687-5265. doi: 10.1155/2016/4835932. URL `https://doi.org/10.1155/2016/4835932`.

K. Govindarajan, V. S. Kumar, et al. Dynamic learning path predictiona learning analytics solution. In *Technology for Education (T4E), 2016 IEEE Eighth International Conference on*, pages 188–193. IEEE, 2016.

T.-C. Huang, Y.-M. Huang, and S.-C. Cheng. Automatic and interactive e-learning auxiliary material generation utilizing particle swarm optimization. *Expert Systems with Applications*, 35(4):2113–2122, 2008.

W. Huitt. Bloom et al.'s taxonomy of the cognitive domain. *Educational psychology interactive*, 22, 2004.

C.-L. Hung and Y.-W. Hung. A practical approach for constructing an adaptive tutoring model based on concept map. In *Virtual Environments, Human-Computer Interfaces and Measurements Systems, 2009. VECIMS'09. IEEE International Conference on*, pages 298–303. IEEE, 2009.

M. T. Islam, P. Thulasiraman, and R. K. Thulasiram. Implementation of ant colony optimization algorithm for mobile ad hoc network applications: openmp experiences. *Scalable Computing: Practice and Experience*, 5(2), 2001.

A. A. Kardan, M. A. Ebrahim, and M. B. Imani. A new personalized learning path generation method: Aco-map. *Indian Journal of Scientific Research*, 5(1):17, 2014.

E. Kurilovas, I. Zilinskiene, and V. Dagiene. Recommending suitable learning scenarios according to learners preferences: An improved swarm based approach. *Computers in Human Behavior*, 30:550–557, 2014.

E. Kurilovas, I. Zilinskiene, and V. Dagiene. Recommending suitable learning paths according to learners preferences: Experimental research results. *Computers in Human Behavior*, 51:945–951, 2015.

C.-H. Lee, G.-G. Lee, and Y. Leu. Application of automatically constructed concept map of learning to conceptual diagnosis of e-learning. *Expert Systems with Applications*, 36(2):1675–1684, 2009.

S. Niemczyk. On pedagogical-aware navigation of educational media through virtual tutors. Technical report, Cambridge, MA: Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, 2000.

J. D. Novak and A. J. Cañas. The theory underlying concept maps and how to construct and use them. 2008.

D. R. Peachey and G. I. McCalla. Using planning techniques in intelligent tutoring systems. *International Journal of Man-Machine Studies*, 24(1):77–98, 1986.

N. Priyadarshini. A review: Data mining techniques in education academia. 2017.

E. Schmider, M. Ziegler, E. Danay, L. Beyer, and M. Bühner. Is it really robust? *Methodology*, 2010.

D. Schunk. *Learning theories: An educational perspective.* Pearson Education Inc, Upper Saddle Hill, NJ, 6 edition, 2012. ISBN 0137071957.

B. Simon and J. Taylor. What is the value of course-specific learning goals. *Journal of College Science Teaching*, pages 53–155, 2009.

S. C. Tan, K. M. Ting, and S. W. Teng. Simplifying and improving ant-based clustering. *Procedia Computer Science*, 4:46–55, 2011.

T. Tang and G. McCalla. Smart recommendation for an evolving e-learning system: Architecture and experiment. *International Journal on e-learning*, 4(1):105–129, 2005.

T.-I. Wang, K.-T. Wang, and Y.-M. Huang. Using a style-based ant colony system for adaptive learning. *Expert Syst. Appl.*, 34(4):2449–2464, May 2008. ISSN 0957-4174. doi: 10.1016/j.eswa.2007.04.014. URL `http://dx.doi.org/10.1016/j.eswa.2007.04.014`.

B. Wasson. *Determining the Focus of Instruction: Content Planning for Intelligent Tutoring Systems.* PhD thesis, Department of Computational Science, University of Saskatchewans, Canada, 1990.

D. A. Wiley. *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.* 2003.

I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2016.

L.-H. Wong and C.-K. Looi. Adaptable learning pathway generation with ant colony optimization. *Educational Technology & Society*, 12(3):309–326, 2009.

L.-H. Wong and C.-K. Looi. Swarm intelligence: new techniques for adaptive systems to provide learning support. *Interactive Learning Environments*, 20(1):19–40, 2012.

Y. J. Yang and C. Wu. An attribute-based ant colony system for adaptive learning object recommendation. *Expert Systems with Applications*, 36(2):3034–3047, 2009.

T. Zhang and X. Lu. Research progress of swarm intelligence algorithms. *Shanxi Architecture*, 33(1):14–16, 2007.