DEMOGRAPHIC ATTRIBUTES BASED, COLD-START RECOMMENDATION OF MODULES IN ORGANIZATIONAL LEARNING

Peduru Hewage Suneth Dasantha Ekanayake

158213P

M.Sc. in Computer Science

Department of Computer Science & Engineering

University of Moratuwa

Sri Lanka

June 2019

DEMOGRAPHIC ATTRIBUTES BASED, COLD-START RECOMMENDATION OF MODULES IN ORGANIZATIONAL LEARNING

Peduru Hewage Suneth Dasantha Ekanayake

158213P

This dissertation submitted in partial fulfillment of the requirements for the

degree Master of Science in Computer Science

Department of Computer Science & Engineering

University of Moratuwa

Sri Lanka

June 2019

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works.

Signature: .... Date: ....

Name : P.H. Suneth Dasantha Ekanayake

The above candidate has carried out research for the Masters Dissertation under my supervision.

Signature of the Supervisor: .... Date: ....

Name : Dr. H.M.N. Dilum Bandara

ABSTRACT

Organizational learning is the process of creating, transferring, and retaining knowledge within an organization. It is of high importance due to the highly dynamic nature of the modern employee base. Moreover, new employees perform sub-optimally and get frustrated when such knowledge and expertise are not readily accessible to them. While many organizations use an organization-learning platform to bridge the knowledge gaps, both for new and existing employees, their effectiveness is being questioned due to lack of relevance, incoherent order of modules to be followed, and lack of fit with the learning style of an employee. While recommendation systems could overcome these challenges, it is difficult to provide a fitting set of recommendations for new employs who do not have any history with learning management system (aka., cold start problem).

We address the cold-start problem in recommender systems for organizational learning using the demographic information of employees. First, similar employees are grouped together based on their demographic attributes. Second, the modules that they follow are clustered according to their similarity. Then the orders of modules and the employee clusters are linked together in such a way that the number of module orders related to a user cluster is maximized. When a cold-start employee enters in to the system, his closest employee cluster is identified based on the demographic features and recommendations are generated considering the module sequences which have the least dissimilarities to the other module sequences in the linked module order cluster. We then tested the proposed technique using a synthetic dataset generated considering a medium scale organization. The dataset consists of age, gender, department, designation, and the order of learning modules followed by the employees. The proposed recommendation system has good accuracy, e.g., 71% of the module recommendations were more than 90% similar to the actual module orders.

Keywords: Collaborative Filtering, Order Clustering, Recommender System, Cold-Start Problem

ACKNOWLEDGEMENTS

My sincere thanks go to my project supervisor Dr. H.M.N. Dilum Bandara for guiding me throughout the project and giving a clear vision and a mission to accomplish during this period. It is essential to mention Mr. Sampath Tilakumara who is the Head of Technology at LSEG Technology, Sri Lanka for encouraging me on the research work.

The interest in machine learning was enriched by following the online learning materials in Coursera. Special mention has to be made regarding the Machine Learning module conducted by Prof. Andrew Ng of Stanford University in Coursera. The learning material and exercises followed in that course helped me a lot in this research work.

I take this moment to thank my parents for the support and the encouragement given to follow the MSc. in Computer Science amidst this busy schedule at working place. Last but not least, my sincere thanks go to the Head of Department of Computer Science and Engineering and the members of the academic staff for advising and guiding me during project evaluations and presentations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
CBF	Content-Based Filtering
CF	Collaborative Filtering
CRBM	Conditional Restricted Boltzmann Machine
MAE	Mean Absolute Error
RBM	Restricted Boltzmann Machine
RMS	Root Mean Square
RMSE	Root Mean Square Error

CHAPTER 1

INTRODUCTION

1.1 Background

Organizational learning is the process of creating, transferring, and retaining knowledge within an organization. Organizational learning occurs within an organization as a function of experience and enables the organization to remain competitive in this fast-changing environment. Individuals are primarily seen as the functional systems for organizational learning by generating understanding through experience. This knowledge of people only promotes learning within the organization only if they are transferred to others by formal means like knowledge transferring sessions. Individuals may withhold or leave the organization with their understanding. But the knowledge should be inteagrated within the organization irrespective of that. As one can see organizational learning is based on applying knowledge for a purpose and learning from the process and from the outcome. Brown and Duguid (1991) describe organizational learning as "the bridge between working and innovating". This once again links learning to action, but it also implies useful improvement. The following intentions should be met by a knowledge management platform in an organization [1]:

- One must understand how to create the ideal organizational learning environment
- One must be aware of how and why something has been learned
- One must try to ensure that the learning that takes place is useful to the organization

By considering all these, it is essential to have a proper learning mechanism for the new comers to transfer and retain the knowledge within an organization. These new comers should be guided so that they feel very comfortable in absorbing the knowledge related to the organization.

Recommender systems which have become one of the most widely used applications of machine learning can overcome these challenges by giving proper recommendations to the new employees. The role of the recommendation systems has become more important in many domains including organizational learning, and online purchasing. It can be imagined how important it is to have an accurate recommender system by conducting competitions like Netfilx Prize [2] for the best collaborative filtering algorithm which can predict the movie ratings based on users' previous behavior.

The job of the recommender system is to predict the unknown preferences of the users. There are two main techniques used in predicting the unknowns by the recommender systems. One is using the demographic information of the users like age, gender, occupation, place of living and the content information of the items like genre, release date, manufacturer in generating recommendations, which is called as Content-Based Filtering. The other is using the rating information by the users on the items without considering the content information in generating recommendations which is called as Collaborative Filtering [3]. For instance, when a user visits a purchasing web site, these recommender systems automatically recommends new items to the users based on the purchasing history, evaluation behavior and previous ratings given. This has a wide range of usage from online purchasing systems to digital libraries, knowledge management systems. There are also hybrid techniques [4] in which both the approaches are employed to generate more accurate recommendations. There, the advantages of both collaborative and content-based filtering are utilized in a fruitful manner.

No matter how accurate the recommender system is, there are several unresolved and unavoidable problems that these systems face. Two such most common problems are generating recommendations to a new user who has no purchasing or rating history and including an item which is freshly introduced to the system into generated recommendations [5]. This problem is called as the cold-start problem; specifically mentioning they are user cold-start and item cold-start respectively.

1.2 Problem Statement

In this research, our main goal is to research on how a recommender system can generate learning module sequences which are ordered according to the preference of learning, for the new comers (cold-start users) to an organization. In other words, it is to research on a recommender system which can support organizational learning process which is aimed at new comers.

As we all have understood, one of the main responsibilities of an organization is to give the new employees a proper guidance and an orientation to the organizational learning. Due to the inability of some organizations to fulfill the above responsibility, severe problems like employee attrition has happened. When a new employee joins an organization, due to the unavailability of the learning history of the employee, it is hard to recommend a series of learning modules that must be followed to be knowledgeable about the context of the organization. If we can present the new employee with a series of learning modules which is in line with the learning habits of the user and the requirements of the organization, that will benefit not only the organization but also the new employee.

The only information that is available when a new employee joins an organization is his/her demographic information. But we have information of the existing and past employees of the organization in terms of their demographic information and the order of learning modules they followed. If we can build a relationship between the demographic informtaiton of the new users and the mentioned historical information of the other users, it will lead the way to building a new recommender system which can help the new employees with a good set of learning module recommendations.

And also, the current recommender systems are mostly built aiming at the domain of online purchasing. The need for a recommendation system in the domain of organizational learning has become a pressing need. This comes to very high importance when it is related to the new comers (cold-start users). So, the main intention of the research to develop a recommender system for organizational learning which can address the user cold-start problem will be of high importance to both the research community and the organizations.

1.3 Objectives

Following objectives are to be achieved to solve the above problem statement:

- Identify potential demographic factors that reflect a user’s learning style and topics of interest

  When we are not aware of the exact learning history of a user, it is needed to get a close understanding before giving recommendations on learning modules. The only available information at hand with us of a new user is his or her demographic information. Most of the times, the demographic features have a close relationship to the attributes of users. In this research it is needed to identify the potential demographic features such as age, gender, department, and designation that can be used to related to the learning styles and habits of users.

- Develop a demographic-factor-based recommendation solution to address the cold-start problem

  The main problem that we are addressing in this research is generating recommendations for the cold-start users in organizational learning. For that it is needed to identify a relationship between the demographic attributes and the learning module preferences.

- Develop a synthetic dataset for model training and performance evaluation

  The dataset generation should be simple so that it can be configured easily, according to the requirement of the organization. Relevant demographic attributes such as age, gender, department, and designation should be chosen. Also, a sequence of learning modules must be generated based on the designation and the department of the employees.

- Evaluate the performance of the proposed demographic-factor-based recommendation solution

  The performance of the recommendation system will have to be tested to identify the effectiveness of the recommendation system. The testing will mainly focus on the percentage of the recommendations that are of a similarity value which is greater than a certain threshold.

1.4 Contribution

To achieve the main goal of building a recommender system for organizational learning we introduced a new approach to combine the demographic information of the employees in generating recommendations for the cold-start employees. The main contribution of this research can be mentioned as the development of a recommendation system that can generate learning module sequence recommendations for cold-start users in organizational learning.

New employees should be guided through a proper learning path to build necessary expertise to the organization. Recommendation systems can generate recommendations based on the historical behavior of the user. But for the employees who are freshly joining the organization, generating recommendations is a great challenge. In this research, demographic attributes were used to give learning module sequence recommendations to new employees who have no learning history.

1.5 Outline

The remainder of this thesis is organized as follows. Chapter 2 presents the literature survey of this research, which covers algorithms like Collaborative Filtering, Content-Based Filtering used in recommendation systems, the problem of user and item cold-start, clustering techniques and related research work. In Chapter 3, we present the implementation of the proposed recommendation system. Chapter 4 presents the synthetic dataset generation and the evaluation of the recommendation system and Chapter 5 summarizes our work and suggest future enhancements.

CHAPTER 2

LITERATURE REVIEW

This research seeks solutions to a recommender system for organizational learning using demographic filtering which suggests a sequence of learning modules which are arranged according to the preference of the user. We studied main algorithms used in recommender systems which are collaborative filtering and content-based filtering, data points clustering techniques like k-means and k-modes clustering, order clustering techniques and cold-start user and cold-start item problem that we encounter in recommender systems. Section 2.1 presents about the organizational learning and collaborative filtering is presented in Section 2.2 whereas Section 2.3 describes about content-based filtering. Section 2.4 explains about cold-start problem in collaborative filtering. Details about clustering algorithms, clustering of categorical values and orders are elaborated in the Sections 2.5, 2.6 and 2.7 respectively.

2.1 Organizational Learning

Organizational learning is the process of creating, retaining and transferring knowledge within an organization. The knowledge gathered inside an organization grows drastically when time passes. This knowledge should be disseminated among the employees effectively so that the organization has the required expertise built. Most of the organizational knowledge is scattered in many places like email conversations, wiki pages, blogs, notebooks, and forums. Online learning management systems play a big role in disseminating the knowledge among the employees and facilitating self-learning. This helps to build a structure that can assist to grasp, document and share knowledge using best practices, organizational level training material. Not only that these learning management systems are used to facilitate self-learning, transmit knowledge and measure the level learning of the employees. However, the effectiveness of these systems is being questioned with respect to their effectiveness, flexibility and personalization aspects. Each employee has his/her own learning style and pace.

Organizations gain knowledge in one of the four ways mentioned below:

1. Individual learning – Individuals master various skills at different levels and the organization can have this knowledge if that individual decided to share this knowledge with the organization.
2. Group learning – Group learning happens in many ways and one of the main methods is that sharing the knowledge of an individual with a set of individuals in an organization.
3. Organizational learning – This is the knowledge created related to the functions and the culture of the organization.
4. Inter-organizational learning – When different organizations interact with each other they learn about the strengths and weaknesses of each other and this helps to an organization to shape itself to yield better results in terms of profit and achievement of goals.

It can be assumed that the organizational risk is inversely proportional to the number of employees who have mastered a certain knowledge area. When a knowledge area is disseminated among number of employees the risk of losing that knowledge by an organization is very low. Hence, organizations should try to disseminate the knowledge among the employees effectively so that the organization has relevant experts in knowledge areas sufficiency. This is one of the main challenges of the contemporary learning management systems must address.

2.2 Collaborative Filtering

Collaborative Filtering (CF) uses the rankings given by the users in the past to predict the unknown preferences of the users [8]. The main concept in collaborative filtering is that the users will have preferences which are like the previous ones they had. It implements the real world “word-of-mouth” phenomenon [9]. Collaborative filtering techniques can be categorized into three main sections: memory-based approaches, model-based approaches and hybrid approaches.

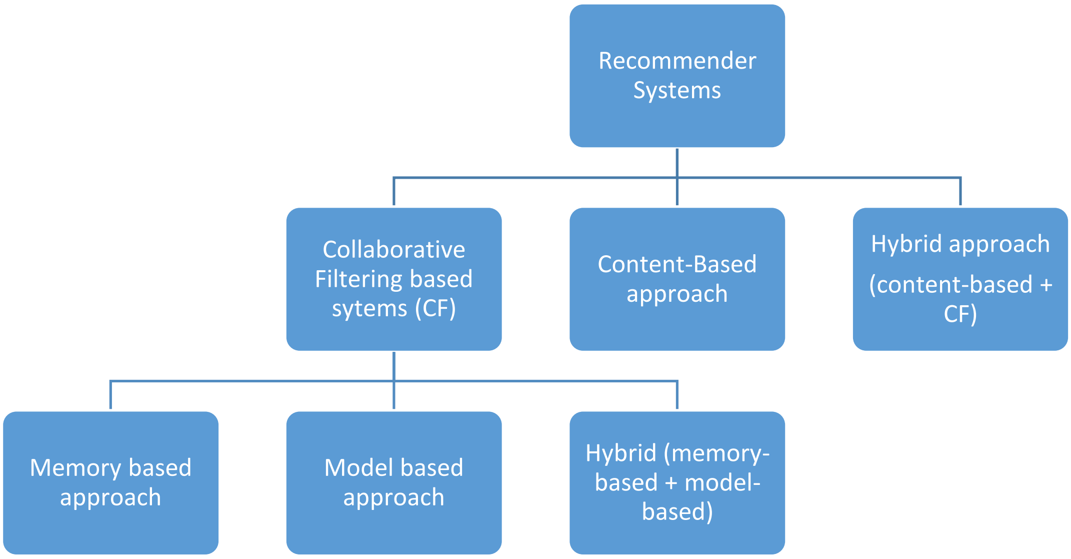

Recommender Systems
Collaborative Filtering based sytems (CF)
Content-Based approach
Hybrid approach (content-based + CF)
Memory based approach
Model based approach
Hybrid (memory-based + model-based)

Figure 2.1: Categorization of recommender systems [10]

Memory-Based Collaborative Filtering

2.2.1 In memory-based recommender systems, a complete record of user-item rating is maintained [9]. Item-based and user-based recommendations can be mentioned as examples to the neighborhood based collaborative filtering approaches. In user-based CF, a potential user-item based preference rating is calculated by taking the preference data of the neighbors of the user. Neighbors of the user are defined as the users with alike features to the target user. Let u be the user, i be the item and $r_{u,i}$ be the rating prediction of the user, then the user-based collaborative filtering can be expressed according to [9], as below:

$$\widehat{r_{u,\iota}} = \frac{\sum_{v\epsilon Nei(u)} sim(u,v) \times r_{v,i}}{\sum_{v\epsilon Nei(u)} sim(u,v)} \qquad (2.1)$$

where $\widehat{r_{u,\iota}}$ is the predicted rating, $Nei(u)$ denotes the u's neighbors who have rated item i, $sim(u,v)$ is the similarity between user u and v, and $r_{v,i}$ is the known preference expressed by u's neighbor v. In this research [9], k most users with alike features to an active user are identified using a similarity-based vector model. Having identified the most alike users, the recommendations are generated by aggregating their user-item rating matrices.

In memory-based collaborative filtering, the main sections of the algorithm include the similarity measure, the neighborhood selection and the normalization of ratings [11]. For the similarity measure several techniques are used. Cosine similarity, Pearson correlation, mean-square difference, Spearman rank correlation are the ones used mostly. It is a fact that is in argument that the selection between the user-based and item-based relies on the ratio of the user-item numbers in the system. The pros of this method include intuitive and expandable and fewer parameter are required to tune the algorithm [9]. But the memory-based CF suffers from sparseness of rating data and scalability issues. These short comings can easily be disappeared by the model-based collaborative filtering mechanisms.

2.2.2 Model-Based Collaborative Filtering

In model-based methods, ranking information is used to build a model to generate predictions rather than merely using that information. Several approaches are adopted to build the model for the recommendation generation like machine learning, clustering, data mining, Bayesian network and dimension reduction methods [12].

2.2.3 Hybrid Recommender Systems

Most of the applications use a hybrid approach to recommender systems to overcome the difficulties in collaborative filtering like sparseness and loss of information and improve the performance of the systems. But these systems have become more complex, expensive and hard to implement and troubleshoot. Massive systems like Google news recommender system use a hybrid approach [13].

2.3 ent-Based Filtering

In the content-based approach, the system learns to give recommendations based on the items rated by the user previously. For instance, if a user has bought books from an online purchasing site, the system suggests books related to the domain, author, genre of the previously bought books [11]. The pros of content-based recommender systems can be listed as follows [16]:

- Recommendations solely depends on the previous ratings of the user.
- No dependency on ratings of other users.

- Ability to recommend new items that are not previously rated by users, since the recommendations are made totally from the features of the item.

This method is useful when there are enough items rated by the user. Otherwise a recommender system with a hybrid mechanism of both collaborative and content-based filtering must be used [17].

2.4 Cold-Start Problem in Collaborative Filtering

In situations where there is almost no rating information available for a new user or item, collaborative filtering-based recommender systems are in difficulty in generating accurate recommendations. This is referred to as the cold-start problem and it is the worst thing that can happen to a collaborative filtering-based recommender system. This can happen in two ways; one is entering of a new user and the other is the entering of a new item to the system which are called as user cold-start and item cold-start respectively. This problem was addressed mainly using the demographic information of the users and content information of the items [9].

2.4.1 User Cold-Start Problem

Collaborative Filtering (CF) based recommender systems are unable to generate accurate recommendations, for the newly entered users to the system. Most of the times, there is an additional interview stage added into the system for the new users. There set of questions related to the recommendation generation are asked and based on that the initial recommendations are generated. For instance, for a movie recommender system, the questions for a new user might be your favorite actor, movie, genre and some demographic information like gender, age, place of living and occupation. Based on the answers, the system uses decision-tree-based methods to select the recommendations for the user. The questions are also asked conditionally in the interview phase to narrow down the recommendations [18].

Exploiting User Demographic Attributes for Solving Cold-Start Problem

Recommender systems are being intensively used commercially and academically to give proper recommendations to users based on their historical ratings. One of the main problems that these systems face is to give recommendations when a new user with no previous ratings comes to the play. This problem of the cold-start user is addressed in different ways by different authors. But the cold-start user problem user remains unresolved fully. To address the drawbacks of the recommender systems, researchers suggest having hybrid approaches like combining the two main recommender system types of collaborative filtering and content-based filtering.

In collaborative filtering, the system creates a neighborhood of users with similar ratings and recommends items based on user ratings with the same taste. In content-based filtering, the system assumes the user will like the items with the features like what he/she has liked before. In both of the cases, the user-cold-start problem remains and in the paper [19], the authors suggest a novel approach to use the demographic features of the users to give recommendations to the new users coming into the system. The main assumption in this paper is that the users of similar demographic features like age, gender, occupation, and place of living will have an alike taste of items.

This research [4] proposes a demographic-based recommender system which consists of three stages. Namely, they are data input, similarity calculation and recommendation calculation as illustrated in Figure 2.2.

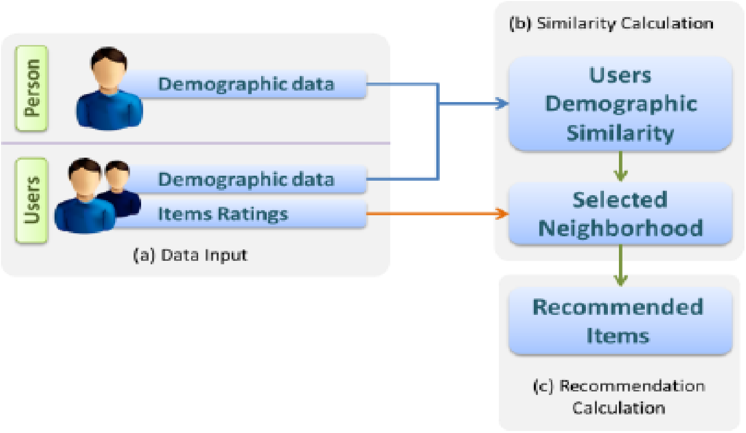


Figure 2.2: Demographic based approach for new users [4]

For the experiment, the authors have used the famous MovieLens 100k dataset which consists of 100,000 ratings evaluated by 943 users on 1,682 movies. In recommendation generation, the system uses the average ratings calculated according to the features selected in similarity calculation. For instance, if the cold-start user gender is female, then the system calculates the average rating given by females for movies and suggests the highest rated movies to the user. For the measurement of the accuracy of the system, MAE and RMSE metrics were used. In conclusion, almost all the demographic features available in the MovieLens dataset affects in a similar fashion to the recommendation result and the recommendation accuracy can be increased by considering the relationship with movie genres and demographic features.

Item Cold-Start Problem

When a system is fed with new items, they do not have previous user rating data.
2.4.2 Hence, the content information of the items is used in giving initial recommendation for the users. There are several approaches taken to user features of the items in addressing the item cold-start problem. One such approach is using Boltzmann machines to model the user ratings on the newly added items [20].

Using Boltzmann Machines for Item Cold-Start Recommendations

In different to collaborative filtering, content-based recommender systems do not have the cold-start problem. But the recommendations are generated only using the features of the items that are rated previously by the user. This causes the recommendations to have a lack of diversity. In [20], the authors have used condition restricted Boltzmann machines (RBMs) successfully by combining the collaborative information and content information successfully.

RBM is an undirected graphical model with two layers and it defines a joint distribution over visible variables v and a hidden variable h. The authors have considered both v and h vectors are to be binary.

RBM is an energy-based model, whose energy function is given by

$$E(v,h) = -\sum_{m=1}^{M} a_m v_m - \sum_{n=1}^{N} b_n h_n - \sum_{m=1}^{M}\sum_{n=1}^{N} v_m h_n W_{mn} \tag{2.2}$$

Where M is the number of visual units, N is the number of hidden units, v_m, h_n are the binary states of visible unit m and hidden unit n, a_m, b_n are their biases and W_{mn} is the weight between v and h.

In this flavor of Boltzmann machines, a node represents a movie. If two movies are rated by the same users in the same manner, then the two Boltzmann machines will be identical. When training the Boltzmann machine, it is done so that the energy of the item is lowered by adjusting the weights and biases in the system. There, approaches like gradient descent are employed.

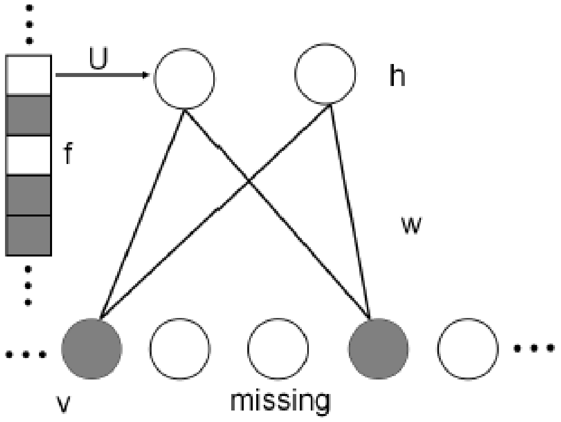


Figure 2.3: Conditional restricted Boltzmann machine with binary hide and binary visual units [20]

The authors have the tested the effectiveness of above mentioned three predictions methods using CRBMs in item cold-start problem using the MovieLens-100K dataset. The evaluation was done using the metrics, precision-recall curve and root mean square error (RMSE). The authors give the conclusion that the usage of CRBM in addressing item cold-start problem gives a considerable performance enhancement and this method can easily be adopted to the user cold-start problem.

2.5

Clustering Algorithms

Given a set of data points, clustering is one of the main techniques in machine learning which is used to group them with similar properties/features. Among the clustering algorithms the following are considered in selecting the clustering algorithm for the research:

- k-means clustering
- k-modes clustering

- Hierarchical clustering
- Self-Organization Map (SOM)
- Expectation Maximization (EM)

In [22], clustering algorithms were compared against the popularity, flexibility, applicability, and ability to handle high dimensionality. Based on these following conclusions were made:

- The processing time of k-means and EM algorithms is better than hierarchical clustering algorithm.
- SOM algorithm shows more accuracy in classifying most the objects into their suitable clusters than other algorithms.
- As the value of *k* becomes greater, the accuracy of hierarchical clustering becomes better until it reaches the accuracy of SOM algorithm.
  All the algorithms have some ambiguity in some (noisy) data when clustered.
- The quality of EM and k-means algorithms become very good when using huge datasets (usually a huge dataset contains 600 rows and 60 columns of data).
- Hierarchical clustering and SOM algorithms show good results when using small dataset.
- As a general conclusion, partitioning algorithms (like k-means and EM) are recommended for large dataset while hierarchical clustering algorithms are recommended for small datasets.
- Hierarchical clustering and SOM algorithms give better results compared to k-means and EM algorithms when acing random dataset and the vice versa.
- k-means and EM algorithms are way sensitive for noise in dataset. This noise makes it difficult for the algorithm to cluster an object into its suitable cluster. This will affect the results of the algorithm.
- Hierarchical clustering algorithm is more sensitive for noisy dataset than SOM algorithm.

By considering the size of the dataset, simplicity of implementation and the performance of the k-means algorithm the difference of the sensitivity to the noise and accuracy of the algorithm with other algorithms in consideration, make little impact to the results of the research.

2.6 Clustering of Categorical Values

k-means algorithm is used extensively due to its performance in numeric data clustering. The problem we see is that the data we encounter in real life is not always numeric; but categorical or combination of both numeric and categorical. When clustering categorical data, it is not possible to use the k-means algorithm directly. In [21], the authors suggest an algorithm named k-modes clustering which uses the frequency of data to identify the mode of the data cluster and a combination of said k-means and k-modes algorithm to group numeric and categorical data which is named as k-prototypes algorithm.

k-means algorithm is widely used in clustering numeric data and it is identified as a non-hierarchical clustering method. The k-means algorithm has key properties [21] such as the ability to process large data sets comparatively efficiently, local optimum solutions, dataset should consist of numeric values, and convex shaped cluster generation.

k-modes algorithm is like an extension to the k-means algorithm and the following modification have been added to the k-means clustering to perform on the categorical data [21]:

- Calculating dissimilarity of categorical objects using a simple equation
- Introducing modes instead of means
- Finding modes using a frequency-based method

Just like k-means algorithm, k-modes algorithms also generate locally optimal solutions. Hence, the selection of initial modes play an important role in proper clustering.

When performing clustering of a dataset which is a combination of numeric and categorical data, the authors of [21] suggest an algorithm named k-prototypes which is a combination of k-means and k-modes algorithms. This algorithm will be of wide usage since the datasets that we find in real life are mostly a combination of numeric and categorical data. The steps of the algorithm are almost the same with the k-means algorithm and the only differences are the cost function and the dissimilarity measure.

The authors have tested and proven the accuracy and the performance of the three algorithms using real world datasets.

2.7 Clustering of Orders

The outcome of our research is to give a learning module sequence which is ordered according to the preference of study to the newly joined employees. There, clustering of the orders is required to identify the best suited learning module sequence for the employee.

Several data like results of a search engine, items in a seller web site are ranked and sorted according to their relevance. In many cases the ranking method used in websites and other systems is to allow the user to give a number between 1 to 5 to an individual item. For instance, the rating given by the same user for two items might be the same. But the preference of the user over the two items cannot be identified only with that rating. Hence, the preference of the items over the others is needed in some scenarios [7], [22], [23].

Collaborative filtering is one of the main techniques employed in personalized recommender systems, where the similarity of the ratings given by the users is used in generating recommendations. One of the key disadvantages in this method is that almost all the CF-based recommender systems use an absolute rating given by the users which are not correct in most of the times. For instance, rating 5 given to a movie by two different users does not mean that both the users love the movie in a similar manner. Hence, the usage of a relative preference will make this shortcoming disappear. In [24], the authors suggest giving the collaborative filtering based recommendation based on an order of preference given for a set of items. When analyzing the preference order given to an item, most of the times it shows a skewed distribution as seen in Table 2.1: Rating distribution.

Table 2.1: Rating distribution [24]

Rating	**1**	**2**	**3**	**4**	**5**
	Dislike		Neither like nor dislike		Like
Ratio	0.082	0.095	0.226	0.224	0.372

There are several ways that we can generate the relative preference order of the users. As identified by the authors of the papers, they can be categorized into three.

Preference order generation methods

- Explicit preference order by the user – The user will directly give the order of preference to the set of items
- Transaction data based implicit order of preference – Preference order can be generated by evaluating the number of transactions associated with each item
- Web mining based implicit order of preference – The preference order is assumed to be in the order of time spent by the user in each web page

The preference order and the calculation of the preference order of the two sequences can be identified as below.

Take three items x_1, x_2, x_3 in preference order O_1 with the preference $x_3 > x_1 > x_2$ .So the relative rating given by a particular user can be elaborated as $r(O_1, x_1) = 2$, $r(O_1, x_2) = 3, r(O_1, x_3) = 1$. If the two order preferences have the same rank number as in Table 2.2, the equation **Error! Reference source not found.** can be used to calculate the similarity of the two orders.

Table 2.2: User Item rating matrix [24]

User/Item	**A**	**B**	**C**	**D**	**E**
U1	3	1	4	5	2
U2	4	2	5	1	3
U3	2	1	4	5	?

$$\rho = \frac{\Sigma_{x\epsilon X1}(r(O_1,x) - \overline{r_1})(r(O_2,x) - \overline{r_2}))}{\sqrt{}\Sigma_{x\epsilon X1}(r(O_1,x) - \overline{r_1})^2\sqrt{}\Sigma_{x\epsilon X2}(r(O_2,x) - \overline{r_2})^2} \qquad (2.3)$$

$where\ \overline{r_i} = \left(\frac{1}{|X_i|}\right)\ \Sigma_{x\epsilon X1}(r(O_{\mathrm{i}},x)$

Calculation of

$\overline{r_i} = \left(\frac{1}{|X_i|}\right)\ \Sigma_{x\epsilon X1}(r(O_{\mathrm{i}},x)\ = \frac{1}{1+2+3+4+5} = \frac{1}{15}$

If two orders have different items, the similarity of the two orders is calculated by dropping the dissimilar items. For instance,

$O_1 = x_1 > x_3 > x_4 > x_6\ O_2 = x_5 > x_4 > x_3 > x_2 > x_6$

Then these orders are transferred as below by taking only the common items

$O_1 = x_3 > x_4 > x_6 \quad O_2 = x_4 > x_3 > x_6$

$r(O_1,x_3) = 1, r(O_1,x_4) = 2, r(O_1,x_6) = 3$

$r(O_2,x_3) = 2, r(O_2,x_4) = 1, r(O_2,x_6) = 3$

$$\rho = 1 - \frac{6 \times\ \Sigma_{x\epsilon X1}\big(r(O_1,x) - r(O_2,x)\big)^2}{|X_1|^3 - |X_1|}$$

$$\rho = 1 - \frac{6 \times ((1-2)^2 +\ (2-1)^2 + (3-3)^2\ )}{|3|^3 - |3|} = 0.5$$

In conclusion, the researchers say that relative preference order can produce better results but when the item count gets increased it is not convenient for a human user to rank the items.

2.8 Summary

The study of the organizational learning and requirements made by the research and development related organizations to have a proper knowledge management system to overcome the challenges faced by them in retaining and disseminating knowledge within the organization led us to the solution of recommender systems. In the literature survey, we had an overview of the organizational learning process and the nature of the research and development-based organizations. In depth analysis of the types of the recommendation systems was a major part of the literature survey. The study of the clustering algorithms and their pros and cons helped us to choose the best clustering algorithms for the purpose. The study of the similar researches related to the field of cold-start user and item problem observed in collaborative filtering-based recommender systems paved the way to identify the solutions like using demographic features of the users and questionnaires to solve the problem.

CHAPTER 3

METHODOLOGY

In this research, we are using the demographic information of the users to combat with cold-start problem of users in collaborative-filtering-based recommender systems for organizational learning. In this chapter we are elaborating our solution to the above problem. Section 3.1 presents the overall architecture of the proposed recommendation system and the stages of the proposed recommendation system are explained under the coming sections respectively.

Overall Architecture of the Proposed Recommender System

3.1 The methodology of the research can be described in three main stages as elaborated in the Figure 3.1 and they can be listed as follows:

- Demographic clustering of users
- Clustering of orders of modules
- Order prediction for cold-start users

The dataset contains information of the demographic attributes of the users and the sequence of learning modules which are ordered according to the preference of the user. First, all the users are clustered according to the demographic features. According to the data types of the dataset, the clustering technique must be identified. We encounter datasets with numeric, categorical and combination of both numeric and categorical values in real life. Then the learning module sequences of the users must be clustered based on the order of preference. This is different from clustering individual items. Special algorithms like k-o' means clustering have to be used, in clustering order sequences. When a new user is entered in to the system, the demographic information like age, gender, designation of the user extracted and the user cluster which has the closest relationship to the user is identified by calculating the Euclidian distance to the mean/mode of the user cluster. Having identified the relevant user cluster, the learning module sequence cluster which consists of the maximum number of preferences of the users in the selected cluster is identified. Then the recommendations are generated by taking the learning modules sequences which have the minimum dissimilarity sum to the other learning modules sequences of the selected module cluster.

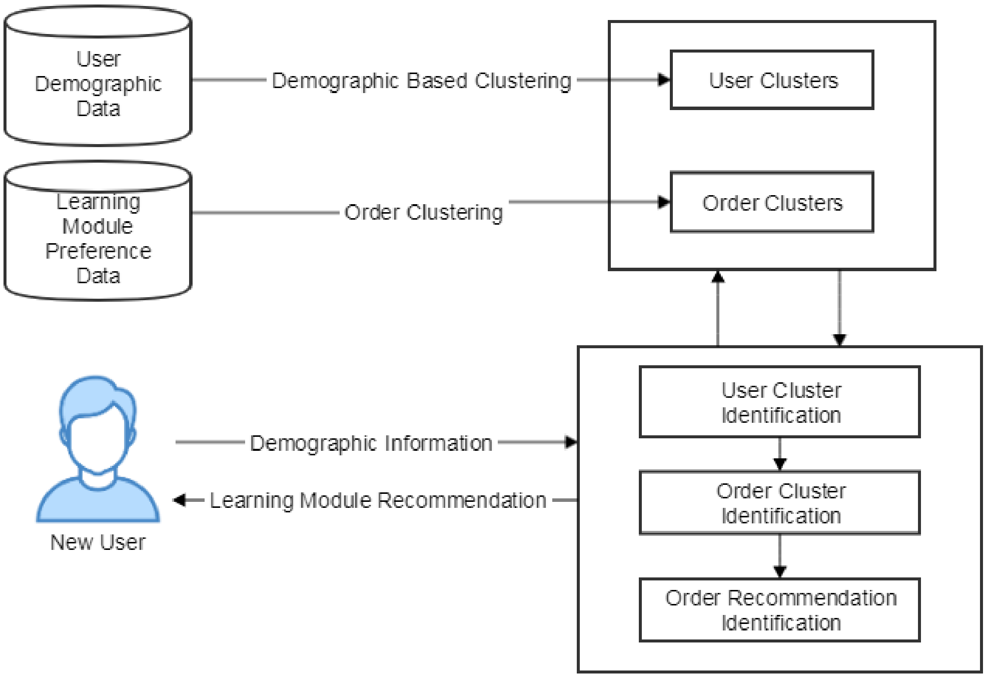

User Demographic Data
Demographic Based Clustering
User Clusters
Learning Module Preference Data
Order Clustering
Order Clusters
User Cluster Identification
Demographic Information
Order Cluster Identification
Learning Module Recommendation
New User
Order Recommendation Identification

Figure 3.1: Architecture of the proposed recommendation system

3.2 Demographic Clustering of Users

In demographic clustering of users, we must choose the algorithm based on the nature of the dataset, as explained above. In this research we give the user the ability to choose the algorithm, either k-means or k-modes in user clustering, depending on the dataset. At this stage we are not using datasets which have combination of both numeric and categorical values. If we encounter a dataset like that, it is advised to convert the data to one type by using a proper conversion logic. For instance, if the age is given as numeric values, it can be represented as categorical values by grouping the age into ranges.

3.2.1 Clustering of Numeric User Data

There are several algorithms like k-means clustering, expectation maximizing clustering, etc., available for clustering numerical data as mentioned in the Section 2.5. For the clustering of numerical data, most of the researches have used k-means based algorithms due to its various favorable aspects which are also mentioned under the Section 2.5 along with a comparison of other algorithms. For the clustering of user attributes when they are of numeric type, the proposed recommendation system uses k-means algorithm.

3.2.1.1 *k-means Algorithm*

k-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centers, one for each cluster. These centers should be placed far away from each other as much as possible [25].

Advantages:

- Fast, robust and easier to understand.
- Relatively efficient: O(tknd), where n is # objects, k is # clusters, d is # dimension of each object, and t is # iterations. Normally, k, t, d << n.
- Gives best results when dataset is distinct or well separated.

Disadvantages:

- The learning algorithm requires a priori specification of the number of cluster centers.
- The use of Exclusive Assignment - If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.
- The learning algorithm provides the local optima of the squared error function.
- Randomly choosing of the cluster center cannot lead us to the fruitful result.
- Applicable only when mean is defined i.e. fails for categorical data.
- Unable to handle noisy data and outliers.

3.2.2 Clustering of Categorical User Data

A lot of real-world data is categorical. For instance, gender, occupation, position, hobby of the employees is stored as categorical data. This statement is very relevant to the datasets that we encounter in the domain of organizational learning. k-modes algorithm is like an extension to the k-means algorithm and if we encounter a dataset of demographic information of users with categorical values, k-modes algorithm is selected to cluster the data based on the same facts that comes with the k-means algorithm.

3.2.2.1 *Dissimilarity Measure*

X and Y are to be two categorical objects with m number of attributes. The dissimilarity is measured as how many categorical values are different from each other in the two objects. Lower the dissimilarity higher the similarity of the objects [21].

$$d_1(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \tag{3.1}$$

where

$$\delta(x_j, y_j) = \begin{cases} 0 \ (x_j = y_j) \\ 1 \ (x_j \neq \ y_j) \end{cases}$$

3.2.2.2 *Mode of a Set*

The mode of a cluster is defined as; let $X = \{X_1, X_2, \dots, X_n\}$ be a collection of objects with categorical attributes with m $\{A_1, A_2, \dots, A_m\}$ number of categorical attributes, the mode is a vector with *m* attributes $Q = \{q_1, q_2, \dots, q_m\}$ which minimizes [21]

$$D(X,Q) = \sum_{i=1}^{n} d_1(X_i, Q) \tag{3.2}$$

Here Q is not necessarily an element of X.

3.2.2.3 *k-modes Algorithm*

The basis of selecting the k-modes algorithms for the clustering of user attributes is very similar to that of k-means algorithms since this is an extension to the k-means algorithm. Apart from them, the complexity of the similarity calculation of k-modes algorithm may increase when the dimension of the dataset increases.

3.2.2.4 *Selection of initial modes*

Just like k-means algorithm, k-modes algorithms also generate locally optimal solutions. Hence, the selection of initial modes plays an important role in proper clustering. There are two types of initial mode selection used. As the first method, distinct records from the initial dataset are chosen as modes. As the second method, the below steps are followed to set the initial modes [21].

1. Calculate the frequencies of all the categories of all the attributes and store them in the descending order of the frequency. $c_{i,j}$ denotes the frequency of the i^{th} category of the j^{th} attribute of the data set.
2. Equally assign the categories with the highest frequency to the modes (k – number of clusters)
3. Start with the first mode and find the most similar record to the selected mode and replace the mode with that record. Do this for the all the modes.

Step three was done to avoid having empty clusters at the end of the algorithm execution. In our research we have used the random selection of distinct records and the above-mentioned mode selection method will be implemented as an extension.

3.3 Clustering of Orders

The next step of our research is to cluster the learning module sequences according to the order of preference. This is different from clustering of individual items since we must consider a sequence of orders as a single item. The authors of [7], [22], and [23] have developed an algorithm named k-o' means clustering for clustering of orders ("an order is a group of objects sorted according to some property"). The authors have tested the algorithm against the traditional hierarchical clustering techniques: the minimum distance, maximum distance and group average methods. This gave them better results in terms of accuracy of the mean selection of clusters and RIL (ratio of information loss). Since the nature of the dataset of our research is very similar to the dataset used in this research and the algorithm is proven to be better over the traditional hierarchical clustering algorithms, we decided to use the same k-o' means clustering algorithm with some modifications in our recommender system to cluster the learning modules.

Defining Order Clusters

Order clusters are defined as follows by the authors of [7], [22] and [23].

3.3.1
- X^*- Universal object which consists of all possible objects
- $O = x^1 > x^2 > \cdots > x^3$ – Order representation
- If $x^1 > x^2$ and $x^2 > x^3$ then $x^1 > x^3$.
- $x^1 > x^2$ represents the order of two objects which says "x1 precedes x2".
- $Xi \subseteq X^*$ - Object set that has all the objects in the order O_i.
- $|A|$ - Size of the set A ($|Xi|$ is equal to the length of the order O_i)
- If $Xi = X^*$ then O_i is considered as a full order.
- If $Xi \subset X^*$. then O_i is considered as a sub order.

The clustering of orders will be done as follows:

Given a set of sample orders, $S = \{O_1, O_2, \ldots, O_{|S|}\}$, $X_i \neq X_j (i \neq j)$ and being $x^1 > x^2$ in the order O_i, does not guarantee that $x^2 > x^1$ in the order O_j . When clustering, S will be divided into a group such that $\pi = \{C_1, C_2, \ldots, C_{|\pi|}\}$. Clustering will produce mutually disjoint and exhaustive clusters, i.e., $Ci \cap Cj = \emptyset, \forall i, j, i \neq$
3.3.2 j and $S = C_1 \cup C_2 \cup \cdots \cup C_{|\pi|}$ and partitions with internal cohesion and external isolation.

Measuring the Similarity between Two Orders

The Spearman's Rank Correlation which is represented by ρ is used to measure the similarity between two orders. The similarity between two orders are based on the differences between the ranks of the objects and the number of discordant object pairs among all object pairs. Formally, an object pair, x_a and x_b, is discordant if

$r(O_1, x_a) < r(O_1, x_b)$ and $r(O_2, x_a) > r(O_2, x_b)$, or vice versa. The Kendall rank correlation coefficient is also used to measure the ordinal similarity between two measured quantities. But the computational complexity of Kendall rank correlation calculation is $O(|X|^2)$ whereas that of Spearman's Rank Correlation calculation is$O(|X|)$. Since, Spearman's Rank Correlation calculation can be done faster, it was used in the research [22], [26].

- ρ - correlation between ranks of objects
- $r(O\ , x)$ - indicates the location of the object x in the order O. In the order $O = x^1 > x^3 > x^2$, the $r(O\ , x_1) = 1$ and the $r(O\ , x_2) = 3$.

  The ρ between O_1 and O_2 which consist of the same objects $(i.e., X_1 = X_2)$ is calculated as [22]:

$$\rho = \frac{\Sigma_{x \epsilon X1}(r(O_1, x) - \overline{r_1})(r(O_2, x) - \overline{r_2}))}{\sqrt{}\Sigma_{x \epsilon X1}(r(O_1, x) - \overline{r_1})^2 \sqrt{}\Sigma_{x \epsilon X2}(r(O_2, x) - \overline{r_2})^2} \tag{3.3}$$

$$where\ \overline{r_i} = \left(\frac{1}{|X_i|}\right) \Sigma_{x \epsilon X1}(r(O_\mathrm{i}, x)$$

If no similar ranking is used inside an order, this can be calculated as follows:

$$\rho = 1 - \frac{6 \times \Sigma_{x \epsilon X1}\left(r(O_1, x) - r(O_2, x)\right)^2}{|X_1|^3 - |X_1|} \tag{3.4}$$

According to the equation 3.4,

- $\rho = 1$ denotes the orders are alike
- $\rho = -1$ denotes the other order is the exact reverse of the order

In situations where all the objects of one order do not appear in the other order, the rank correlation will be derived as explained in the below example.

In the example below, the rank is given according to the order of the preference given to a certain property of the objects. For instance, the rank can be given according to the order the subjects are chosen from a basket of subjects in an academic semester.

Consider the two orders,

$O_1 = x^1 > x^3 > x^4 > x^6$

$O_2 = x^5 > x^4 > x^3 > x^2 > x^6$

From these orders, only the common objects are considered.

$O_1' = x^3 > x^4 > x^6$

$O_2' = x^4 > x^3 > x^6$

The ranks of the objects can then be defined as:

$r(O_1', \mathrm{x}^3) = 1, r(O_1', x^4) = 2, r(O_1', x^6) = 3$

$r(O_2', \mathrm{x}^3) = 2, r(O_2', x^4) = 1, r(O_2', x^6) = 3$

Consequently, the ρ was

$$\rho = 1 - \frac{6 \times ((1-2)^2 + \ (2-1)^2 + (3-3)^2\ )}{|3|^3 - |3|} = 0.5$$

In clustering, it is useful to know how much the two orders are dissimilar to each other. Hence the dissimilarity between two orders are defined as:

$$d(O_1\,, O_2)\ =\ 1\ -\ \rho \tag{3.5}$$

The range of the dissimilarity will be [0,2], since the similarity varies between -1 and 1. $d = 0$ denotes that the two orders are similar.

3.3.3 Calculating the Order Mean

Before explaining the k-o' means algorithm, it is needed to explain how to obtain the mean of an order cluster [7]. In the k-means algorithm, the mean of the cluster C is calculated as follows:

$$\bar{x} = \arg\min_{x^i} \sum_{x^j \epsilon C} ||x^i - x^j|| \tag{3.6}$$

where x_i are the data points, C is the cluster, and $||.||$ is the norm of L_2. Similarly, the order mean $\bar{O}$ is defined as follows:

$$\bar{O} = \arg\min_{O_j} \sum_{O_j \epsilon C} d(O_i, O_j) \tag{3.7}$$

Here orders consist of the all objects related to the scenario. In such situations order mean is derived using the Borda rule in the 18th century, which can be expressed using the following algorithm [22]:

1. Calculate

$$\tilde{r}^*(x^a) = \frac{1}{|C|} \sum_{O_i \epsilon C} r(O_i, x^a) \tag{3.8}$$

   for each object x^a in the X^*.

2. Order mean of cluster C can be identified by arranging objects according to the $\tilde{r}^*(x^a)$ ascending order.

   Note: If $\tilde{r}^*(x^a) = \tilde{r}^*(x^b)$, $x^a \neq x^b$, either $x^a > x^b$ or $x^b > x^a$ is allowed.

   k-o’ means Algorithm

3.3.4 This algorithm has very similar steps like in the k-means algorithm and the result of the algorithm depends on the initial selection of the cluster means. The steps of the algorithm are elaborated in the research paper [22].

3.4 Order Prediction for Cold-Start Users

As the last step of the algorithm, the system should suggest a learning module sequence ordered by the preference, when the information of a new employee is given. We suggest the following approach mentioned in Figure 3.2 to generate the learning module recommendation for the new user.

3.5

Explanation Using a Sample Dataset

Consider the below sample dataset of 50 employees whose demographic and learning module preference data are as in Table 3.1 and Table 3.2. For the training of the recommender system 80% of the data is taken. In this scenario information of 40 employees are taken to train the system and the other 10 employees are taken to test the accuracy of the system. Number of the user and order cluster counts are considered to be 3 and 3 respectively.

```
Algorithm: Order Prediction ($u, n$)

Pre-condition: User and order clustering is completed.
$C_i$: user clusters, $O_i$: order clusers
$k$: user cluster count
$l$: order cluster count
$u$: new user
$n$: number of recommendations needed

1. get the user cluster id $C_i$ of $u$ so that the distance to the mean or mode of the cluster is minimum
2. get all the entities (users) $X_i$ that belong to cluster $C_i$
3. for each order cluster $O_j$, take all the relevant entities(users) $U = \{U_1, U_2, \ldots, U_l\}$, grouped
   according to the order clusters
4. get the intersection count $I = \{I_1, I_2, \ldots, I_l\}$ of the users who are common to both $X_i$ and user
   groups in $U$
5. select order cluster $O$ such that $I$ is maximum
6. take the first $n$ order sequences which have the minimum dissimilarity to the orders of the cluster
   $O$
7. recommend the $n$ number of items selected according to step 6
```

Figure 3.2: Order prediction algorithm.

This demographic information represents a dataset with categorical values. Hence, we have to select k-modes algorithm as the user clustering mechanism. As mentioned under Section Demographic Clustering of Users3.2, first the training dataset which consists of the demographic information of the users is clustered into the given number of clusters. In this example, first 40 employees clustered into 3 clusters as in the Figure 3.3. Here users with same demographic information are shown with overlapping ‘x’s.

As mentioned under Section 3.3, secondly the leaning module sequence of the employees which are shown in Table 3.2 are clustered. Learning module sequence information of the first 40 employees are taken and they are clustered into 3, using the k-o’ means algorithm.

A single row of Table 3.2 consists of the order of learning modules followed by the employee. -1 indicates that the employee has not yet taken that module.

cx – Common modules

rx – Research and Development modules

sx – Sales and Marketing modules

gx – General and Administration modules

ox – Operations modules

Table 3.1: Demographic information of users

ID	Age Category	Gender	Designation
1	5	1	16
2	1	0	17
3	1	0	9
4	1	0	18
5	2	0	2
6	1	0	9
7	1	0	4
8	1	0	3
9	3	1	11
10	2	0	4
11	1	1	1
12	1	0	9
13	2	0	10
14	1	0	1
15	0	0	1
16	1	0	2
17	3	0	14
18	3	0	11
19	1	1	4
20	6	0	6
21	2	1	10
22	4	1	11
23	1	0	3
24	3	0	10
25	1	0	10
26	1	0	3
27	3	0	11
28	1	0	9
29	2	1	9
30	3	0	19
31	5	0	20
32	1	0	4
33	1	1	3
34	3	0	1
35	3	0	18
36	6	1	12
37	1	0	3
38	2	1	9
39	3	1	9
40	5	1	11
41	4	0	8
42	2	0	10
43	5	0	11
44	1	1	13
45	1	1	13
46	4	0	4
47	2	1	9
48	4	1	12
49	1	1	9
50	1	1	13

After applying the k-o'means algorithm, these orders are clustered as in Figure 3.4. Learning module sequences are colored according to the cluster they are grouped into.

When order clustering happens, the system keeps the dissimilarity measures in the ascending order inside each cluster for each learning module sequence. In this scenario the system keeps the dissimilarity counts of the cluster 0 (which is colored in grey) as in **Error! Reference source not found.**. By analyzing the **Error! Reference source**

not found. and Table 3.4, it can be identified that the entry ids which are least dissimilar to the other entry ids.

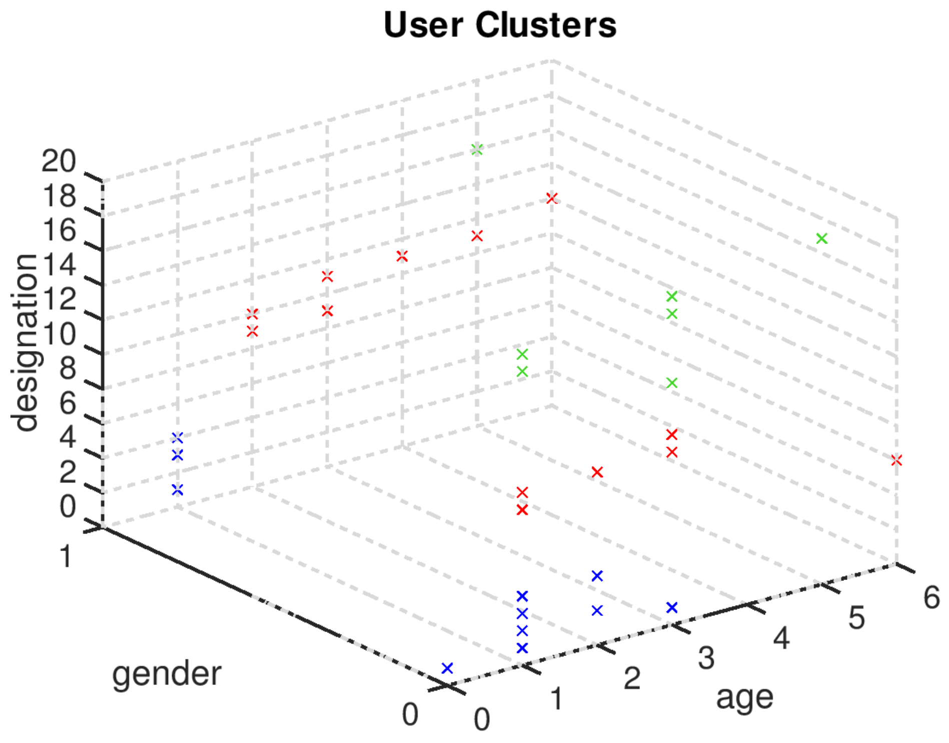
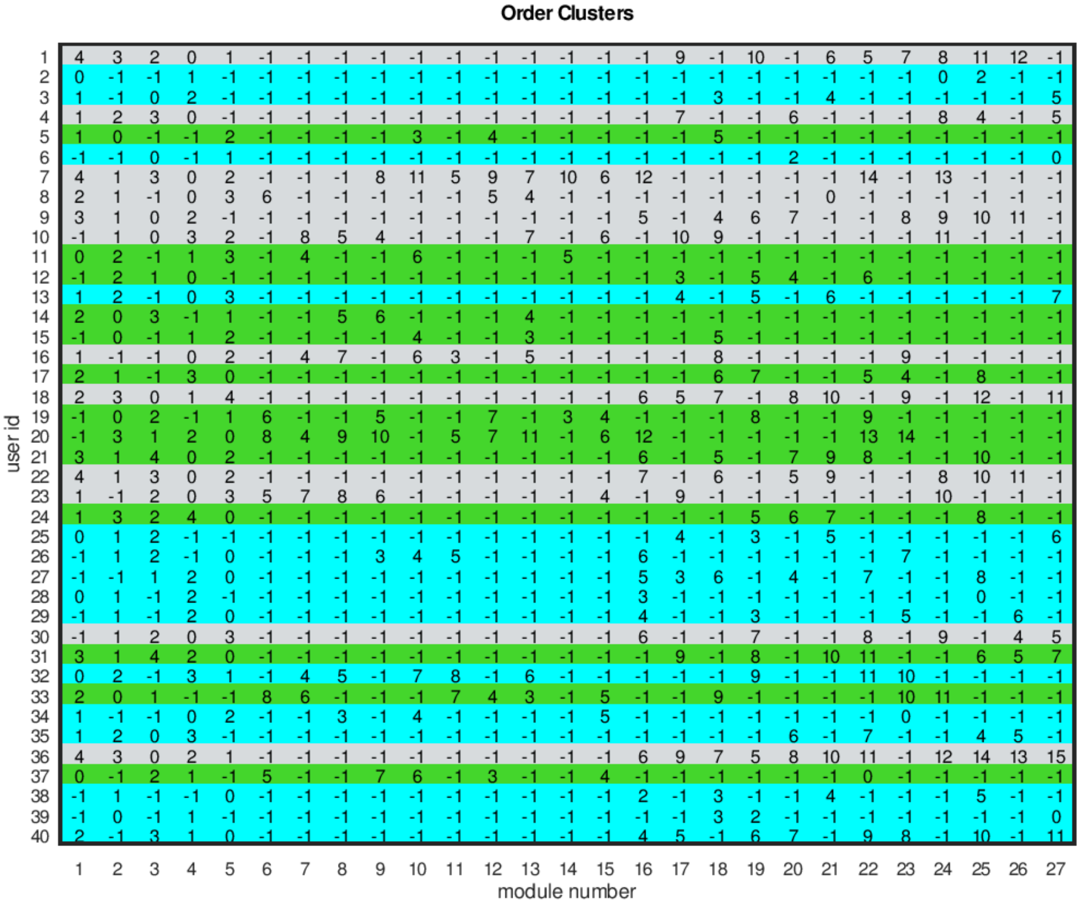


Figure 3.3: User Clusters, k = 3

Table 3.2: Learning module sequences ordered according to the preference

ID	**c1**	**c2**	**c3**	**c4**	**c5**	**r1**	**r2**	**r3**	**r4**	**s1**	**s2**	**s3**	**s4**	**g1**	**g2**	**o1**	**o2**	**o3**
1	4	3	2	0	1	-1	-1	-1	-1	-1	9	-1	10	6	5	11	12	-1
2	0	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	-1	-1
3	1	-1	0	2	-1	-1	-1	-1	-1	-1	-1	3	-1	4	-1	-1	-1	5
4	1	2	3	0	-1	-1	-1	-1	-1	-1	7	-1	-1	-1	-1	4	-1	5
5	1	0	-1	-1	2	-1	-1	-1	-1	-1	-1	5	-1	-1	-1	-1	-1	-1
6	-1	-1	0	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
7	4	1	3	0	2	-1	-1	-1	8	12	-1	-1	-1	-1	14	-1	-1	-1
8	2	1	-1	0	3	6	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1
9	3	1	0	2	-1	-1	-1	-1	-1	5	-1	4	6	-1	-1	10	11	-1
10	-1	1	0	3	2	-1	8	5	4	-1	10	9	-1	-1	-1	-1	-1	-1
11	0	2	-1	1	3	-1	4	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
12	-1	2	1	0	-1	-1	-1	-1	-1	-1	3	-1	5	-1	6	-1	-1	-1
13	1	2	-1	0	3	-1	-1	-1	-1	-1	4	-1	5	6	-1	-1	-1	7
14	2	0	3	-1	1	-1	-1	5	6	-1	-1	-1	-1	-1	-1	-1	-1	-1
15	-1	0	-1	1	2	-1	-1	-1	-1	-1	-1	5	-1	-1	-1	-1	-1	-1
16	1	-1	-1	0	2	-1	4	7	-1	-1	-1	8	-1	-1	-1	-1	-1	-1
17	2	1	-1	3	0	-1	-1	-1	-1	-1	-1	6	7	-1	5	8	-1	-1
18	2	3	0	1	4	-1	-1	-1	-1	6	5	7	-1	10	-1	12	-1	11
19	-1	0	2	-1	1	6	-1	-1	5	-1	-1	-1	8	-1	9	-1	-1	-1
20	-1	3	1	2	0	8	4	9	10	12	-1	-1	-1	-1	13	-1	-1	-1
21	3	1	4	0	2	-1	-1	-1	-1	6	-1	5	-1	9	8	10	-1	-1
22	4	1	3	0	2	-1	-1	-1	-1	7	-1	6	-1	9	-1	10	11	-1
23	1	-1	2	0	3	5	7	8	6	-1	9	-1	-1	-1	-1	-1	-1	-1
24	1	3	2	4	0	-1	-1	-1	-1	-1	-1	-1	5	7	-1	8	-1	-1
25	0	1	2	-1	-1	-1	-1	-1	-1	-1	4	-1	3	5	-1	-1	-1	6
26	-1	1	2	-1	0	-1	-1	-1	3	6	-1	-1	-1	-1	-1	-1	-1	-1
27	-1	-1	1	2	0	-1	-1	-1	-1	5	3	6	-1	-1	7	8	-1	-1
28	0	1	-1	2	-1	-1	-1	-1	-1	3	-1	-1	-1	-1	-1	0	-1	-1
29	-1	1	-1	2	0	-1	-1	-1	-1	4	-1	-1	3	-1	-1	-1	6	-1
30	-1	1	2	0	3	-1	-1	-1	-1	6	-1	-1	7	-1	8	-1	4	5
31	3	1	4	2	0	-1	-1	-1	-1	-1	9	-1	8	10	11	6	5	7
32	0	2	-1	3	1	-1	4	5	-1	-1	-1	-1	9	-1	11	-1	-1	-1
33	2	0	1	-1	-1	8	6	-1	-1	-1	-1	9	-1	-1	-1	-1	-1	-1
34	1	-1	-1	0	2	-1	-1	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
35	1	2	0	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	7	4	5	-1
36	4	3	0	2	1	-1	-1	-1	-1	6	9	7	5	10	11	14	13	15
37	0	-1	2	1	-1	5	-1	-1	7	-1	-1	-1	-1	-1	0	-1	-1	-1
38	-1	1	-1	-1	0	-1	-1	-1	-1	2	-1	3	-1	4	-1	5	-1	-1
39	-1	0	-1	1	-1	-1	-1	-1	-1	-1	-1	3	2	-1	-1	-1	-1	0

40	2	-1	3	1	0	-1	-1	-1	-1	4	5	-1	6	-1	9	10	-1	11
41	4	2	1	3	0	14	10	9	6	-1	15	-1	16	-1	20	-1	-1	-1
42	3	2	-1	0	1	-1	-1	-1	-1	4	5	-1	-1	-1	-1	-1	8	-1
43	-1	2	3	0	1	-1	-1	-1	-1	5	6	-1	4	8	-1	11	-1	10
44	-1	0	3	2	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
45	-1	-1	0	2	1	-1	-1	-1	-1	-1	-1	-1	-1	3	-1	6	-1	-1
46	-1	1	3	2	0	5	-1	-1	8	-1	-1	-1	-1	-1	11	-1	-1	-1
47	2	1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	3	-1	-1	-1	-1	5
48	0	2	3	-1	1	-1	-1	-1	-1	5	6	-1	4	9	11	12	13	-1
49	1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4
50	3	2	-1	0	1	-1	-1	-1	-1	-1	6	5	-1	-1	-1	-1	-1	7


Order Clusters
user id
module number

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	4	3	2	0	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	9	-1	10	-1	6	5	7	8	11	12	-1
2	0	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	2	-1	-1
3	1	-1	0	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	-1	-1	4	-1	-1	-1	-1	-1	5
4	1	2	3	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	7	-1	-1	6	-1	-1	-1	8	4	-1	5
5	1	0	-1	-1	2	-1	-1	-1	-1	3	-1	4	-1	-1	-1	-1	-1	5	-1	-1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	0	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	-1	-1	-1	-1	-1	-1	0
7	4	1	3	0	2	-1	-1	-1	8	11	5	9	7	10	6	12	-1	-1	-1	-1	-1	14	-1	13	-1	-1	-1
8	2	1	-1	0	3	6	-1	-1	-1	-1	-1	5	4	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1
9	3	1	0	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	5	-1	4	6	7	-1	-1	8	9	10	11	-1
10	-1	1	0	3	2	-1	8	5	4	-1	-1	-1	7	-1	6	-1	10	9	-1	-1	-1	-1	-1	11	-1	-1	-1
11	0	2	-1	1	3	-1	4	-1	-1	6	-1	-1	-1	5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
12	-1	2	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	-1	5	4	-1	6	-1	-1	-1	-1	-1
13	1	2	-1	0	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4	-1	5	-1	6	-1	-1	-1	-1	-1	7
14	2	0	3	-1	1	-1	-1	5	6	-1	-1	-1	4	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
15	-1	0	-1	1	2	-1	-1	-1	-1	4	-1	-1	3	-1	-1	-1	-1	5	-1	-1	-1	-1	-1	-1	-1	-1	-1
16	1	-1	-1	0	2	-1	4	7	-1	6	3	-1	5	-1	-1	-1	-1	8	-1	-1	-1	-1	9	-1	-1	-1	-1
17	2	1	-1	3	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	6	7	-1	-1	5	4	-1	8	-1	-1
18	2	3	0	1	4	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	6	5	7	-1	8	10	-1	9	-1	12	-1	11
19	-1	0	2	-1	1	6	-1	-1	5	-1	-1	7	-1	3	4	-1	-1	-1	8	-1	-1	9	-1	-1	-1	-1	-1
20	-1	3	1	2	0	8	4	9	10	-1	5	7	11	-1	6	12	-1	-1	-1	-1	-1	13	14	-1	-1	-1	-1
21	3	1	4	0	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	6	-1	5	-1	7	9	8	-1	-1	10	-1	-1
22	4	1	3	0	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	7	-1	6	-1	5	9	-1	-1	8	10	11	-1
23	1	-1	2	0	3	5	7	8	6	-1	-1	-1	-1	-1	4	-1	9	-1	-1	-1	-1	-1	-1	10	-1	-1	-1
24	1	3	2	4	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	5	6	7	-1	-1	-1	8	-1	-1
25	0	1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4	-1	3	-1	5	-1	-1	-1	-1	-1	6
26	-1	1	2	-1	0	-1	-1	-1	3	4	5	-1	-1	-1	-1	6	-1	-1	-1	-1	-1	-1	7	-1	-1	-1	-1
27	-1	-1	1	2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	5	3	6	-1	4	-1	7	-1	-1	8	-1	-1
28	0	1	-1	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1
29	-1	1	-1	2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4	-1	-1	3	-1	-1	-1	5	-1	-1	6	-1
30	-1	1	2	0	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	6	-1	-1	7	-1	-1	8	-1	9	-1	4	5
31	3	1	4	2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	9	-1	8	-1	10	11	-1	-1	6	5	7
32	0	2	-1	3	1	-1	4	5	-1	7	8	-1	6	-1	-1	-1	-1	-1	9	-1	-1	11	10	-1	-1	-1	-1
33	2	0	1	-1	-1	8	6	-1	-1	-1	7	4	3	-1	5	-1	-1	9	-1	-1	-1	-1	10	11	-1	-1	-1
34	1	-1	-1	0	2	-1	-1	3	-1	4	-1	-1	-1	-1	5	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1
35	1	2	0	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	6	-1	7	-1	-1	4	5	-1
36	4	3	0	2	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	6	9	7	5	8	10	11	-1	12	14	13	15
37	0	-1	2	1	-1	5	-1	-1	7	6	-1	3	-1	-1	4	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1
38	-1	1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	-1	3	-1	-1	4	-1	-1	-1	5	-1	-1
39	-1	0	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	2	-1	-1	-1	-1	-1	-1	-1	0
40	2	-1	3	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4	5	-1	6	7	-1	9	8	-1	10	-1	11

Figure 3.4: Clustering of learning module sequences, *k* = 3

Table 3.3: Dissimilarity values against each entry of cluster 0

Emp ID Emp ID	1	2	5	12	24	25	26	27	28	31	33	34	37	38	39
1	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.50	0.00	0.00	2.00	0.00	0.00	0.00	0.50
2	0.00	0.00	0.00	0.20	0.20	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	1.50	0.20
5	0.00	0.00	0.00	2.00	0.00	2.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.20
12	2.00	0.20	2.00	0.00	0.06	2.00	0.50	1.50	0.20	0.70	0.00	1.50	0.50	1.20	0.17
24	0.00	0.20	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.50	0.00	0.50	0.10
25	0.00	0.00	2.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	1.50	2.00	0.00	0.00	0.00
26	0.00	0.00	0.50	0.50	0.00	0.00	0.00	1.50	0.00	0.00	2.00	0.30	0.00	0.00	0.10
27	0.50	0.00	0.00	1.50	0.00	0.00	1.50	0.00	0.00	0.00	2.00	0.60	1.50	0.00	1.20
28	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	1.50	0.00	0.00	0.00	0.10
31	0.00	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.40	0.00	0.00	0.00	0.17
33	2.00	2.00	0.00	0.00	0.00	1.50	2.00	2.00	1.50	0.40	0.00	2.00	0.00	0.00	0.60
34	0.00	0.00	0.00	1.50	1.50	2.00	0.30	0.60	0.00	0.00	2.00	0.00	0.00	0.00	0.23
37	0.00	0.00	0.00	0.50	0.00	0.00	0.00	1.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	1.50	0.00	1.20	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.50
39	0.50	0.20	1.20	0.17	0.10	0.00	0.10	1.20	0.10	0.17	0.60	0.23	0.00	1.50	0.00

Table 3.4: Dissimilarity sum values against each entry of cluster 0

Employee ID	Dissimilarity Sum	Order of Dissimilarity Values
1	5	8
2	4.1	5
5	5.7	9
12	12.52857	14
24	2.357143	4
25	7.5	11
26	4.895238	7
27	8.8	13
28	1.8	2
31	1.271429	1
33	14	15
34	8.128571	12

37	2	3
38	4.7	6
39	6.066667	10

After that the system calculates the user counts who are common to both user and order clusters as in Table 3.5.

Table 3.5: Common User Counts in User and Order Clusters

Order Cluster ID **User Cluster ID**	**0**	**1**	**2**
0	10	4	4
1	5	8	4
2	6	4	5

Now the system has all the information required to generate the recommendations to the new users. The system is tested using the users who are left without taking to train the system. In this case, the last 10 users are used to test the system.

For instance, let's see the how the recommendation is generated for the 41st employee in the dataset according to the steps mentioned under Section 3.4. The order prediction algorithm is elaborated in Figure 3.2. The demographic information of the 41st employee is as in Table 3.6.

Table 3.6: Demographic information of the 41st employee

Employee ID	**Age category**	**Gender**	**Designation**
41	4	0	8

According to the k-modes algorithm, the user is assigned to the cluster to which it has minimum dissimilarity. Refer to the Table 3.7.

Table 3.7: Dissimilarity to user clusters

Employee ID	**Cluster 0**	**Cluster 1**	**Cluster 2**
41	0.2	0.35	0.8

Based on the dissimilarity values in Table 3.7, this employee belongs to the cluster 0.

Having identified the user cluster, we then see the order cluster, which has the maximum number of users who are in common to both selected user cluster (In this scenario, the selected user cluster is the 0th) and order cluster. According to the Table 3.5, the suitable order cluster is the cluster 0.

As the last step, we choose the number of recommendations which are the sequences of the learning modules based on the **Error! Reference source not found.**. The recommendation is selected such that it has the minimum dissimilarity to the orders in the cluster. If the system is configured to give multiple suggestions (say n number of suggestions), we then select the first n number of order sequences which have the least dissimilarity values to the other orders in the cluster. In this scenario, the selected first three recommendations to the 41st employee are as in Table 3.8.

Table 3.8: Learning module sequence recommendations to the 41st employee

ID	c1	c2	c3	c4	c5	r1	r2	r3	r4	r5	s1	s2	s3	s4	g1	g2	g3	o1	o2	o3
31	3	1	4	2	0	-1	-1	-1	-1	-1	-1	9	-1	8	10	11	-1	6	5	7
28	0	1	-1	2	-1	-1	-1	-1	-1	-1	3	-1	-1	-1	-1	-1	-1	0	-1	-1
37	0	-1	2	1	-1	5	-1	-1	7	6	-1	-1	-1	-1	-1	0	-1	-1	-1	-1

For the evaluation, the dissimilarity between the recommendations and the actual is calculated as elaborated in Chapter 4.

3.6 Summary

This chapter explained the overall architecture of the proposed methodology of the research. There are three main stages of the recommendation system. In the first stage, demographic clustering of users is done with the available demographic information of the users having the assumption that the users with the same demographic attributes share the same interests. In the second stage, clustering of learning module sequences is done which is different from clustering of individual items since it is needed to consider a sequence of orders as a single item. Now the recommender system is ready to generate recommendations. When a new user is entered in to the system, the demographic information like age, gender, designation of the user extracted and the user cluster which has the closest relationship to the user is identified by calculating the Euclidian distance to the mean/mode of the user cluster. Having identified the relevant user cluster, the learning module sequence cluster which consists of the maximum number of preferences of the users in the selected cluster is identified. Having identified the module sequence cluster, the recommendations are generated by taking the learning modules sequences which have the minimum dissimilarity sum to the other learning modules sequences of the selected cluster.

CHAPTER 4

PERFORMANCE EVALUATION

The evaluation process of the research outcome mainly based on the similarity measure of the predicted and actual sequence of learning modules. Unlike in most of the recommender systems, here the recommendation outcome is a series of learning modules ordered according to the preference. Then the similarity of the prediction and the original must be measured considering the whole sequence of learning modules. Here, Section 4.1 presents the steps taken in preparing the synthetic dataset, Section 4.2 presents the evaluation metrics, Section 4.3 presents the assumptions and decisions that we took in testing and evaluation and the Section 4.3 presents the results of the testing on the synthetic dataset.

4.1 Synthetic Dataset Generation

To generate a synthetic dataset that is suitable for a research and development organization, it was needed to research on the composition of the employees in different organizations. Prior to the dataset generation, sushi dataset [27] was used to perform the initial testing which gave us confidence in using a synthetic dataset. It was difficult to find a proper employee distribution that is available online. In Kaggle, there is an employee attrition dataset from which we can have a rough idea on the employee distribution in a general research and development organization [28]. Since we are attached to software engineering development organizations, employee composition of such organizations was taken as a sample in generating the dataset, which is also very similar to the said Kaggle dataset.

4.1.1 Employee Demographic Information Dataset

When preparing the user demographic dataset department, designation, age, and gender of the employees were decided as the demographic attributes of interest and it was decided to make it suitable for a medium scale organization, where there are around 300-500 employees. So, for the evaluation purpose, we have decided to use the number of employees as 400. Since the organization that we are considering is engineering oriented, the following departments were identified as appropriate.

- Research and Development
- Sales and Marketing
- General and Administration
- Operations

The percentage of employee distribution across the departments and the gender wise distribution of the employees within the department was identified as in the Table 4.1.

Table 4.1: Department wise employee distribution

Department	Employee Distribution % (from total no of employees)	Male/Female %
Research and Development	40	70/30
Sales and Marketing	30	40/60
General and Administrative	15	50/50
Operations	15	80/20

Designation wise employee distribution within the departments of Research and Development, Sales and Marketing, General and Administration and Operations can be identified as in Table 4.2, Table 4.3, Table 4.4 and Table 4.5, respectively.

Table 4.2: Designation wise distribution in Research and Development department

Designation	Distribution (%)
Engineer	25
Senior Engineer	20
Specialist Engineer	15
Associate Lead	12
Lead	10
Senior Lead	8
Associate Architect	5
Architect	3
Senior Architect	2

Table 4.3: Designation wise distribution in Sales and Marketing department

Designation	Distribution (%)
Entry	40
Level 1	30
Level 2	20
Level 3	10

Table 4.4: Designation wise distribution in General and Administration department

Designation	Distribution (%)
Entry	40
Level 1	30
Level 2	20
Level 3	10

Table 4.5: Designation wise distribution in Operations department

Designation	Distribution (%)
Entry	35
Level 1	25
Level 2	20
Level 3	20

In this dataset creation, six age categories were identified, and the ranges of age are as in Table 4.6.

Table 4.6: Age categories

Age Categories	Age Distribution
0	18-23
1	24-29
2	30-34
3	35-40
4	41-44
5	45-50
6	51 -

When deciding on the age distribution in each department, the designation of the employee was considered, and the dataset was prepared as in Table 4.7.

Table 4.7: Designation wise age distribution in departments

	Age Category							
Department	**Designation**	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Research and Development	Engineer	20	60	20	0	0	0	0
	Senior Engineer	20	80	10	8	0	0	0
	Specialist Engineer	10	70	20	9	0	0	0
	Associate Lead	0	60	30	10	0	0	0
	Lead	0	30	40	20	10	0	0
	Senior Lead	0	5	20	40	20	10	5
	Associate Architect	0	0	5	55	20	15	5
	Architect	0	0	0	20	50	25	5
	Senior Architect	0	0	0	10	40	30	20
Sales and Marketing	Entry	2	80	10	8	0	0	0
	Level 1	0	30	40	20	10	0	0
	Level 2	0	0	5	55	20	15	5
	Level 3	0	0	0	10	40	30	20
General and Administration	Entry	2	80	10	8	0	0	0
	Level 1	0	30	40	20	10	0	0
	Level 2	0	0	5	55	20	15	5
	Level 3	0	0	0	10	40	30	20
Operations	Entry	2	80	10	8	0	0	0
	Level 1	0	30	40	20	10	0	0
	Level 2	0	0	5	55	20	15	5
	Level 3	0	0	0	10	40	30	20

4.1.2

Learning Module Preference Sequence

When an employee joins an organization, there is a mandatory orientation program that must be followed. Some leaning is common to all the employees irrespective of the department that they are attached to. By considering these facts, the learning module distribution mentioned in

Table 4.8 was identified. This distribution includes common modules which must be followed by each employee of the organization and department specific modules. There may be some employees taking learning modules from other departments as well. For instance, we have considered the employees of the Research and Development department are taking modules related to Sales and Marketing and General and Administration when they are moving in the career ladder. This is same for the employees in other departments as well.

Table 4.8: Department wise module distribution

Module Nature	**Module Count**
Common Modules	5
Research and Development Modules	10
Sales and Marketing Modules	5
General and Administration Modules	4
Operations Modules	3
Total Module Count	27

When deciding on the module completion percentage, to avoid all the employees with the same designation having the same module completion percentage, it was decided to add a noise to the completion percentage which has a standard normal distribution. For instance, consider two employees of the Research and Development department with the designation, Architect.

Common module completion % of employee 1

$$= 0.80 + \left(\frac{random\ standard\ normal\ value}{10}\right) \times\ common\ module\ count$$

$$= 0.80 + (0.3/10)\ x\ 5 = 0.95$$

Common module completion % of employee 2

$$= 0.80 - (0.3/10)\ x\ 5 = 0.65$$

In the dataset, there are 27 learning modules in total as mentioned in Table 4.8. For an employee, a vector of 27 elements (as in Table 4.10) will be generated for the learning module preference order. This vector will indicate the order of preference of the employee in taking the module. -1 indicates that the module is not yet taken by the employee.

Table 4.9: Designation wise module completion percentage – mean value

Department	**Module Completion %** **Designation**	**Common Modules**	**Research and Development Modules**	**Sales and Marketing Modules**	**General and Admin Modules**	**Operations Modules**
Research and Development	Engineer	10	10	0	0	0
	Senior Engineer	20	20	0	0	0
	Specialist Engineer	30	30	0	5	0
	Associate Lead	40	40	0	10	0
	Lead	50	50	5	15	0
	Senior Lead	60	60	10	20	0
	Associate Architect	70	70	15	30	0
	Architect	80	80	25	50	0
	Senior Architect	90	90	50	70	0
Sales and Marketing	Entry	50	0	20	10	10
	Level 1	60	0	40	20	20
	Level 2	70	0	60	40	40
	Level 3	80	0	80	60	60
General and Administration	Entry	50	0	10	20	10
	Level 1	60	0	20	40	20
	Level 2	70	0	30	60	30
	Level 3	80	0	50	80	50
Operations	Entry	50	0	10	10	20
	Level 1	60	0	20	20	40
	Level 2	70	0	30	30	60
	Level 3	80	0	40	40	0

Table 4.10: Learning module preference vector

c1	c2	c3	c4	c5	r1	r2	r3	r4	r5	s1	s2	s3	s4	g1	g2	g3	g4	o1	o2	o3
2	0	-1	1	3	-1	-1	-1	-1	-1	4	6	-5	-1	-1	7	10	-1	8	-1	9

cx – Common modules

rx – Research and Development modules

sx – Sales and Marketing modules

gx – General and Administration modules

ox – Operations modules

When clustering the orders (vectors with 27 elements), k-o' means clustering algorithm was used.

4.2 Evaluation Metrics

The Spearman's Rank Correlation will be used to measure the similarity between two orders [7], [22], [29] as explained in the Section 3.3.3. Root mean square error of the predictions will be used to assess the quality of the results.

$$RMSE = \sqrt{\sum_{i \in TestSet} \frac{d_i^2}{|TestSet|}} \tag{4.1}$$

where d_i^2 is the dissimilarity calculated between the real order and the predicted order.

4.3 Experiment on Synthetic Dataset

The experiment was done varying the user cluster count from 2 to 5 and order cluster size from 2 to 5. Each experiment was run for 10 times and average values of dissimilarity were taken for the analysis. The following graphs show the dissimilarity of each input and its variation along with the cluster count changes.

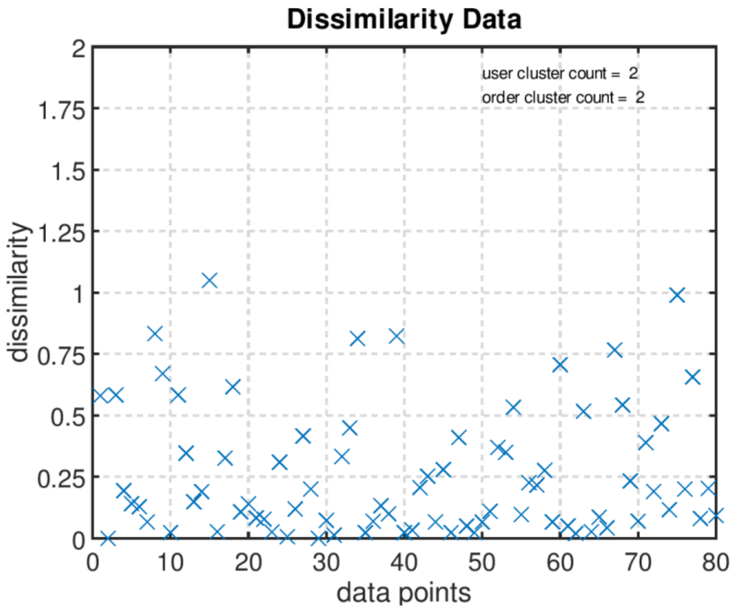


Figure 4.1: Dissimilarities of the prediction and the actual, user cluster count=2, order cluster count=2

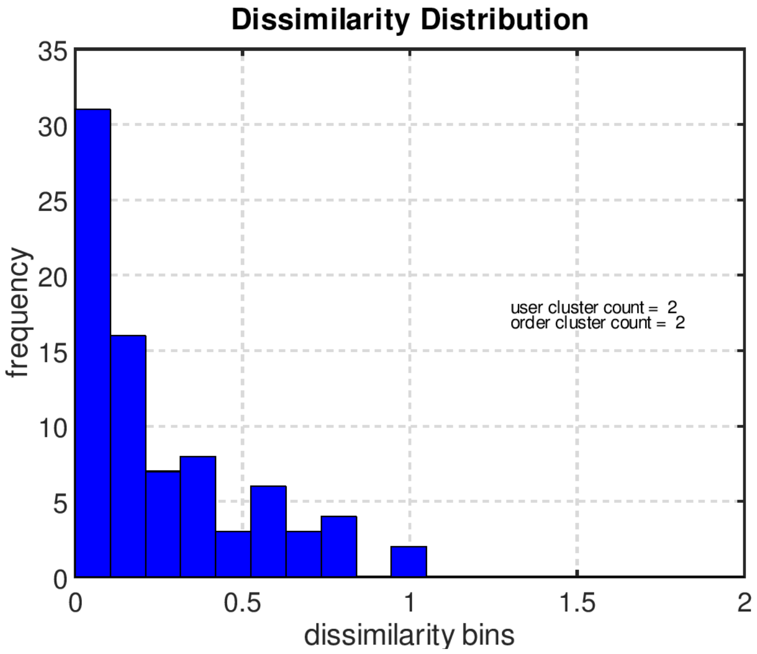


Figure 4.2: Dissimilarity distribution, user cluster count=2, order cluster count= 2

As seen in Figure 4.1 and 4.2 the dissimilarity variation is almost below 1 and it is concentrated more around 0 for user cluster count = 2 and order cluster count = 2.

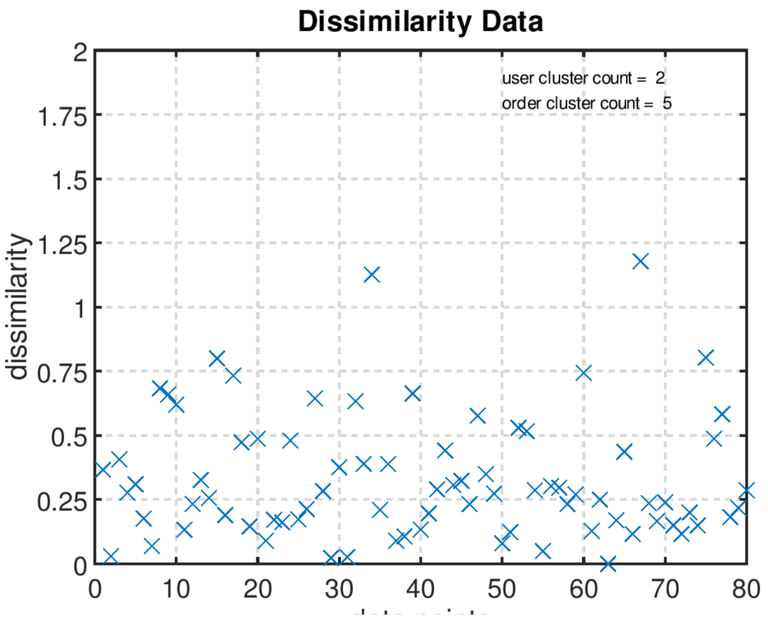


Figure 4.3: Dissimilarities of the prediction and the actual, user cluster count=2, order cluster count=5

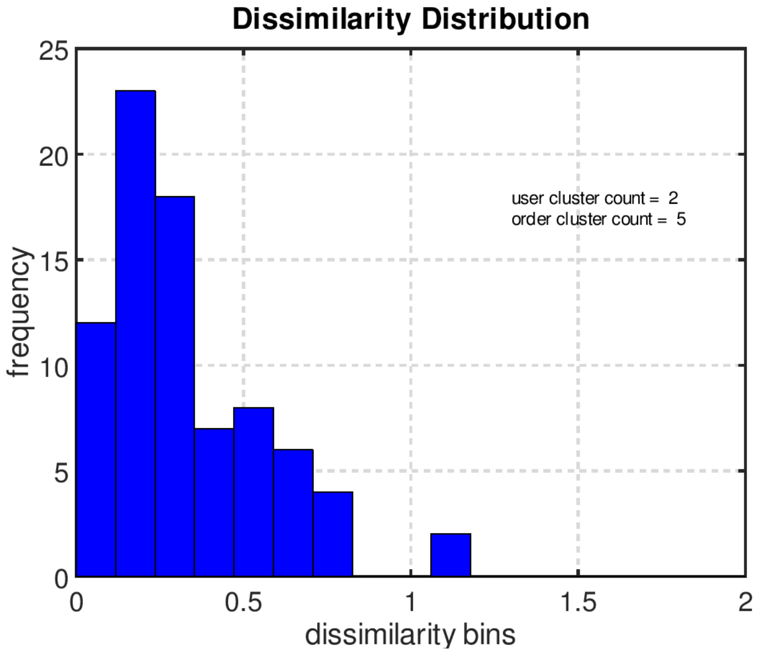


Figure 4.4: Dissimilarity distribution, user cluster count=2, order cluster count=5

As seen in Figure 4.3 and 4.4 the dissimilarity variation is almost below 1 and the concentration of dissimilarity has moved somewhat away from 0 for user cluster count = 2 and order cluster count = 2.

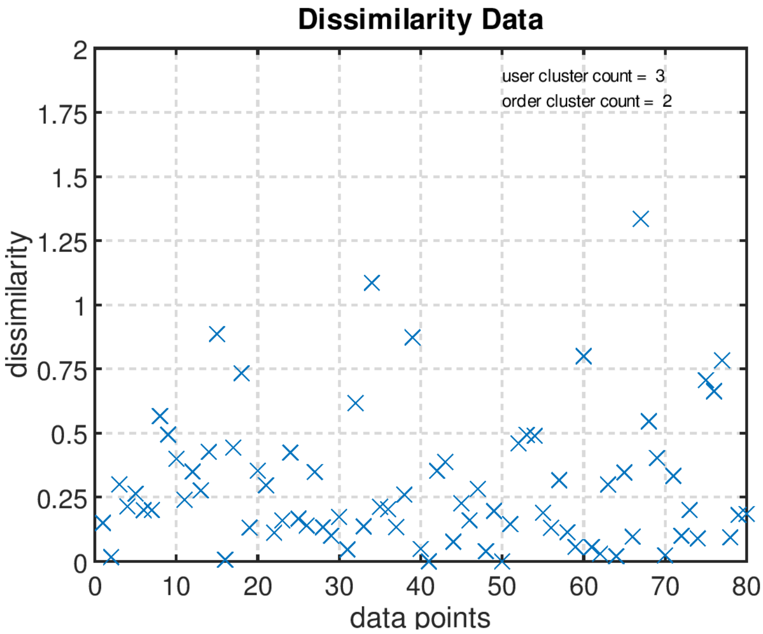


Figure 4.5: Dissimilarities of the prediction and the actual, user cluster count=3, order cluster count=2

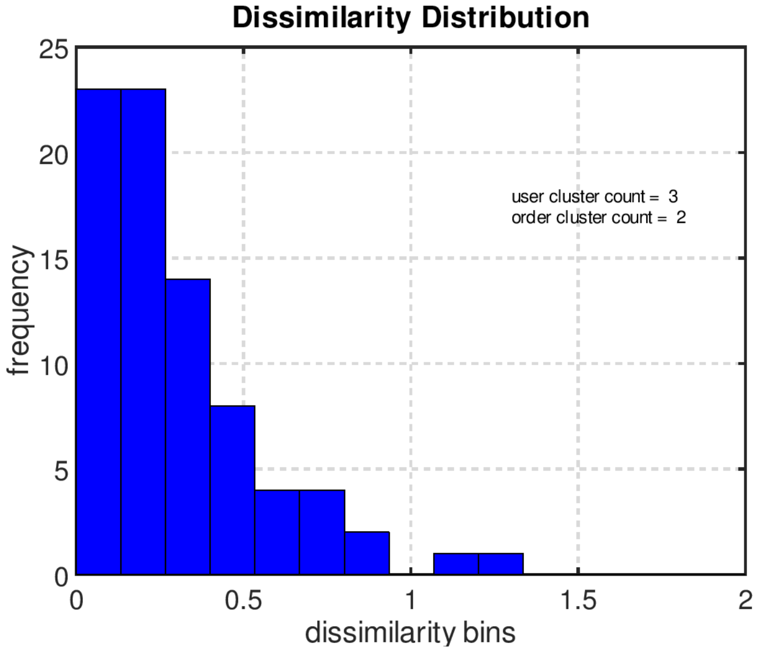


Figure 4.6: Dissimilarity distribution, user cluster count=3, order cluster count=2

As seen in Figure 4.5 and Figure 4.6, the dissimilarity variation is almost below 1 and the concentration of dissimilarity is more towards 0 for user cluster count = 3 and order cluster count = 2.

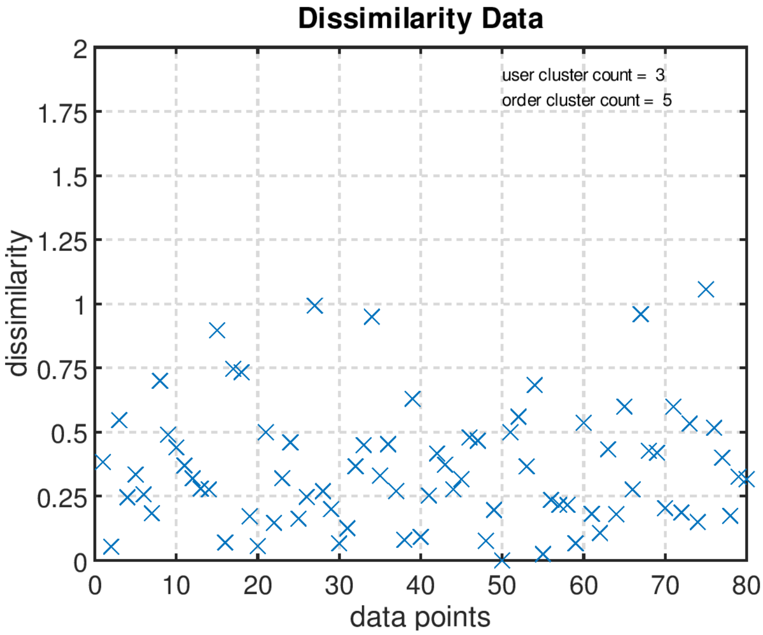


Figure 4.7: Dissimilarities of the prediction and the actual, user cluster count=3, order cluster count=5

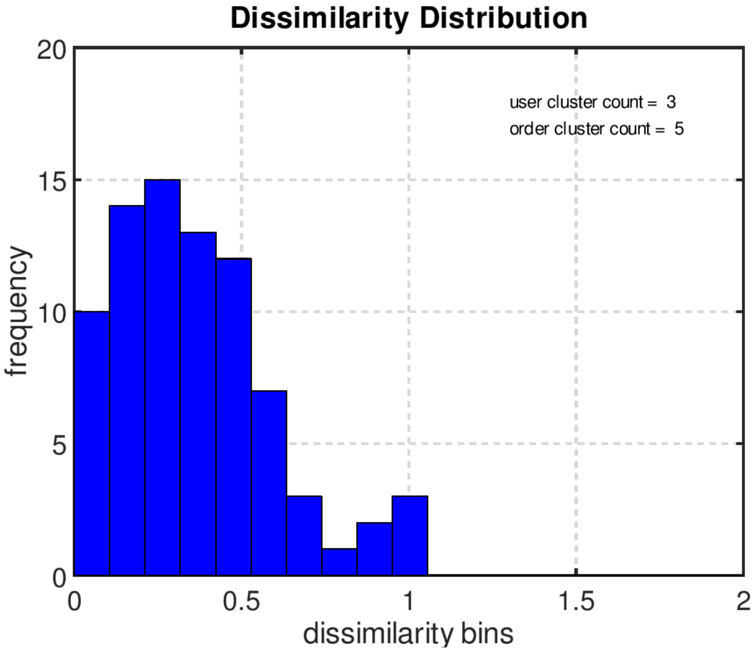


Figure 4.8: Dissimilarity distribution, user cluster count=3, order cluster count=5

As seen in Figure 4.7 and 4.8, the dissimilarity variation is almost below 1 and the concentration of dissimilarity has moved somewhat away from 0 for user cluster count = 3 and order cluster count = 5.

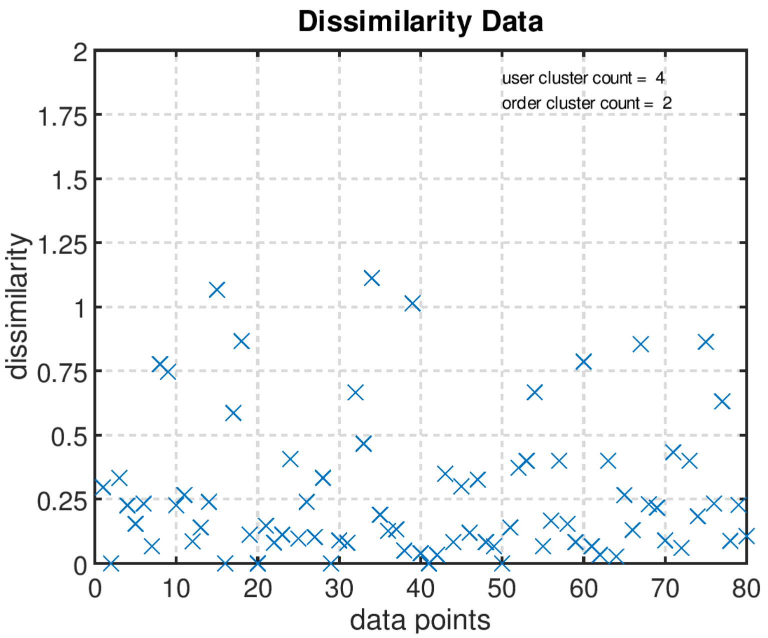


Figure 4.9: Dissimilarities of the prediction and the actual, user cluster count=4, order cluster count=2

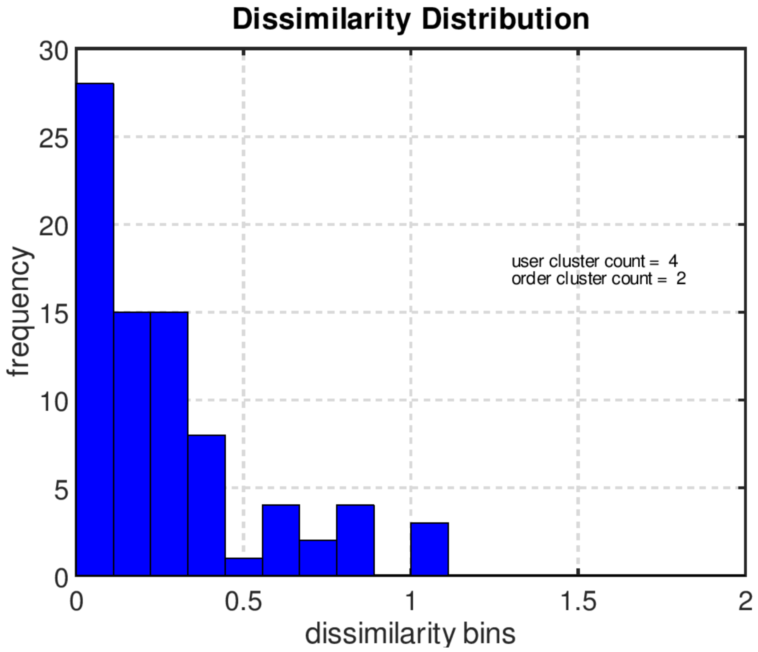


Figure 4.10: Dissimilarity distribution, user cluster count=4, order cluster count=2

As seen in Figure 4.9 and 4.10, the dissimilarity variation is almost below 1 and the concentration of dissimilarity is more towards 0 for user cluster count = 4 and order cluster count = 2.

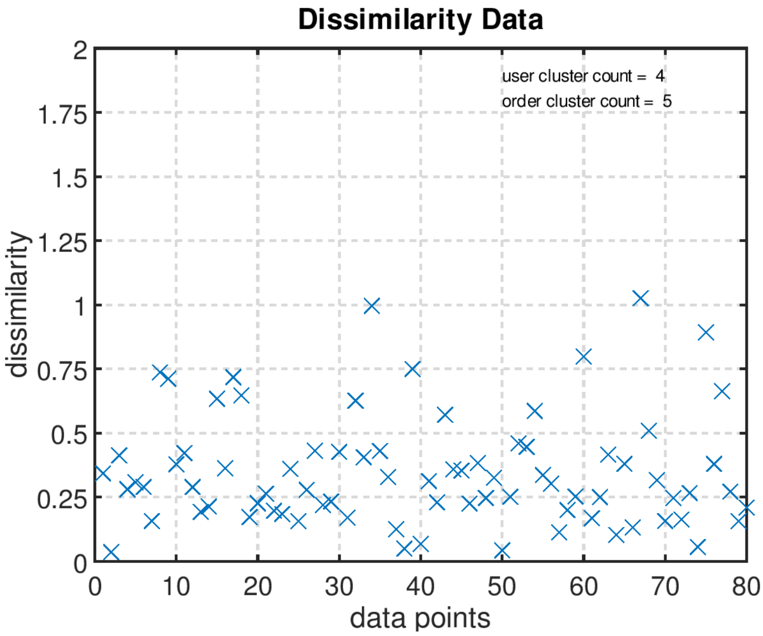


Figure 4.11: Dissimilarities of the prediction and the actual, user cluster count=4, order cluster count=5

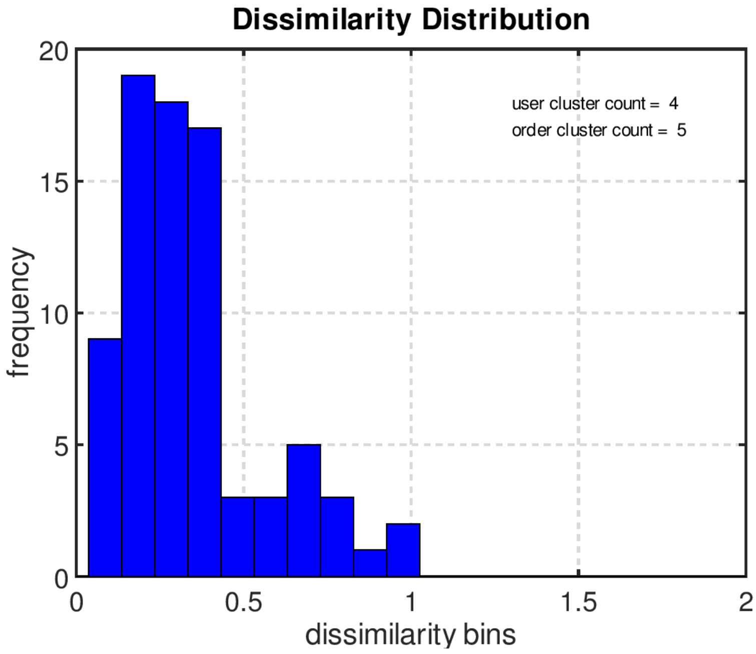


Figure 4.12: Dissimilarity distribution, user cluster count=4, order cluster count=5

As seen in Figure 4.11 and Figure 4.12, the dissimilarity variation is almost below 1 and the concentration of dissimilarity has moved somewhat away from 0 for user cluster count = 4 and order cluster count = 5.

After gathering the individual dissimilarity data as shown in above graphs, the RMSE value for each run was calculated for the same synthetic dataset. This process was done for several synthetic datasets and the test results are as follows.

RMSE Variation for the Synthetic Dataset 01

Table 4.11: RMSE variation with user and order cluster sizes

4.3.1

Order Cluster Count **User Cluster Count**	**2**	**3**	**4**	**5**
2	0.47240658	0.43201664	0.44146955	0.42023863
3	0.46864863	0.4173902	0.5056506	0.43227265
4	0.50124892	0.45715268	0.54307979	0.56392224
5	0.46055746	0.43914539	0.47550961	0.48972207

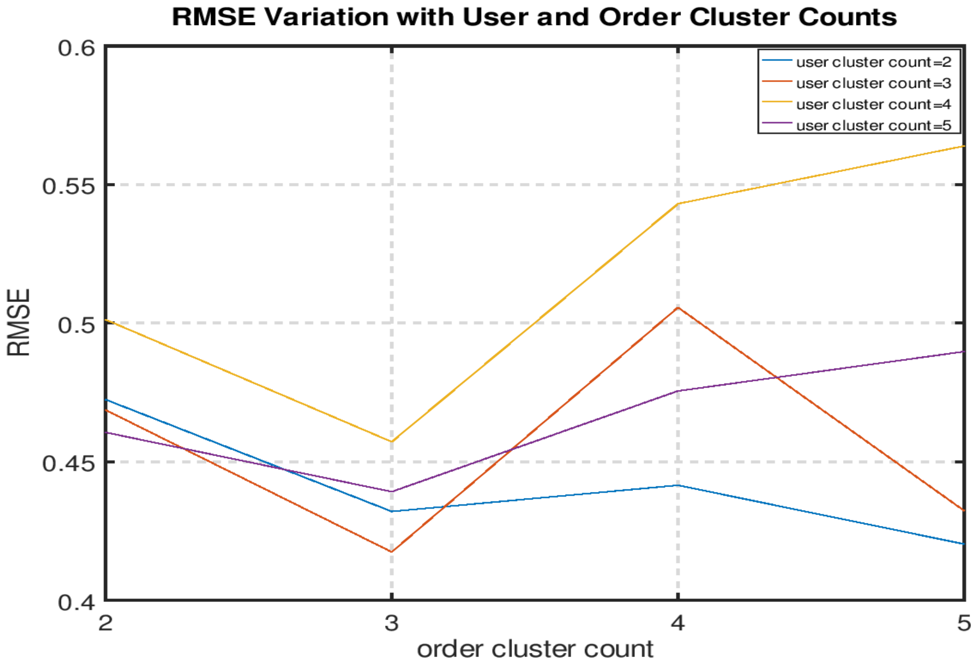


Figure 4.13: RMSE variation

By observing the graph in Figure 4.13, we can see that when user cluster count is set to five, the RMSE variation showed minimum values against the other order cluster counts except against when user cluster count is set to two. This showed a minimum variation of RMSE value when the order cluster count is varied. This is due to the predictions are coming very close to the actuals. When user cluster count is increased, more the similar users are grouped. The value of optimal k totally depends on the distribution of the data in the dataset which can be identified with by running the same algorithm on different synthetic datasets. Running of the system on the same dataset yielded almost similar results.

The minimum RMSE value is reported for the user and cluster count 3 and 3 respectively. The dissimilarity data generated for each individual prediction are as follows.

0.046666667, 0.066666667, 0.039047619, 0, 0.078571429, 0, 0.166666667, 0.354761905, 0.338095238, 0.486190476, 0.233333333, 0.173333333, 0.746666667, 0.033333333, 0.314444444, 0.113333333, 0.125079365, 0.026666667, 0.646666667, 0.026666667, 0.523809524, 0.046666667, 0.66, 0.200952381, 0.228571429, 0.213333333, 0.62, 0.435714286, 0.407936508, 0.966666667, 0.2, 0.012929293, 0.166666667, 0.08952381, 0.286666667, 0.233333333, 0.217142857, 0.146666667, 0.306666667, 0.232467532, 0.08, 0.141212121, 0.493333333, 0.266666667, 0.233333333, 0.87047619, 0.354285714, 0.54, 0.206666667, 0.5, 0.081269841, 0.046666667, 0.252121212, 0.36, 0.567619048, 0.03047619, 0.293333333, 0.213333333, 0.663809524, 1.3, 0.266666667, 0.74, 0.266666667, 0.722857143, 0.153333333, 0.134603175, 0.47, 0.251428571, 0.253809524, 0.553333333, 0.084761905, 0.66, 0.54, 0.7, 0.324761905, 0.533333333, 0.76, 0.540952381, 0.466666667, 0.112380952

By analyzing the mentioned dissimilarity values, we can say that

- Predictions of 98.75% of test data are more than 50% similar to the actual module order sequence
- Predictions of 73.75% of test data are more than 75% similar to the actual module order sequence
- Predictions of 66.25% of test data are more than 80% similar to the actual module order sequence

  Predictions of 33.75% of test data are more than 90% similar to the actual module order sequence

4.3.2 RMSE Variation for the Synthetic Dataset 02

Table 4.12: RMSE variation with user and order cluster counts

Order Cluster Count User Cluster Count	2	3	4	5
2	0.33285686	0.36694911	0.34971665	0.41978153
3	0.37440804	0.33410006	0.47385482	0.34847405
4	0.34453419	0.36617183	0.46224329	0.42756657
5	0.34147798	0.38821003	0.35941758	0.39269539

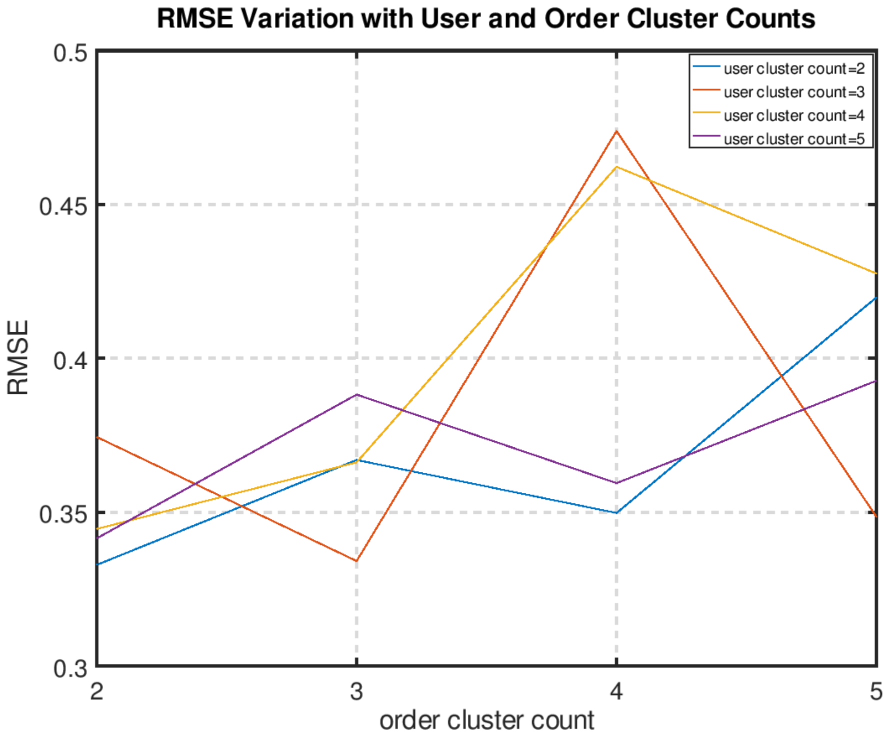


Figure 4.14: RMSE variation

By observing the graph in Figure 4.14, we can see that when user cluster count is set to 5, the RMSE variation showed minimum values against the other order cluster counts except against when user cluster count is set to 2 which is very similar to the behavior in Dataset 01 in Section 4.3.1. This is due to the predictions are coming very close to the actuals. When user cluster count is increased, more the similar users are grouped.

The minimum RMSE value is reported for the user and cluster count 3 and 3 respectively. The dissimilarity data generated for each individual prediction are as follows.

0.092207792, 0.3, 0.2204329, 0.103809524, 0.313333333, 0.114065934, 0.003809524, 0.214285714, 0.333333333, 0.109090909, 0.205238095, 0.939047619, 0.143015873, 0.06, 0.234285714, 0.128571429, 0.123333333, 0.013809524, 0.717619048, 0.014920635, 0.108388278, 0.7, 0.013333333, 0.10047619, 0.147619048, 0, 0, 0.243838384, 0.076507937, 0.073333333, 0.851024531, 0.013333333, 0.101111111, 0.532857143, 0.206666667, 0.397777778, 0.161587302, 0.126666667, 0.231428571, 0.126666667, 0.386666667, 0.040952381, 0.304761905, 0.285714286, 0.076507937, 0.173333333, 0.141855922, 0.116190476, 0.013333333, 0.299365079, 0.133333333, 0.196177156, 0.183809524, 0.145238095, 0.2, 0.303881674, 0.12, 0.22, 0.300952381, 0.086666667, 0.196883117, 0.440634921, 0.159206349, 0.126666667, 0.041904762, 0.776190476, 1.026666667, 0.162857143, 0.086103896, 0.106666667, 0.492141192, 0.693630814, 0.221125541, 0.145714286, 0.21047619, 0.093333333, 0.118888889, 0.085714286, 1.02, 0.48

By analyzing the mentioned dissimilarity values, we can say that

- Predictions of 97.50% of test data are more than 50% similar to the actual module order sequence
- Predictions of 88.75% of test data are more than 75% similar to the actual module order sequence
- Predictions of 85.25% of test data are more than 80% similar to the actual module order sequence

  Predictions of 58.75% of test data are more than 90% similar to the actual module order sequence

By considering 10 datasets below percentage of predictions of which are of 50%, 75%, 80% and 90% accuracy are found.

Similarity **Dataset**	**>50%**	**>75%**	**>80%**	**>90%**
1	98.75	73.75	66.25	33.75
2	97.5	88.75	85.25	58.25
3	98.75	85	77.5	51.25
4	96.25	85	73.75	40
5	97.5	77.5	62.5	30
6	97.5	76.25	62.5	26.25
7	95	75	70	32.5
8	96.25	85	73.75	40
9	100	68.75	61.25	23.75
10	97.5	90	78.74	38.75
Average %	**97.5**	**80.5**	**71.149**	**37.45**

4.3.3 Analysis of Results

We tested the recommender system for several synthetic datasets and the results displayed almost a very similar behavior to the above. When analyzing the above results;

When user cluster count is set to 5, the RMSE variation showed minimum values against the other order cluster counts except against user cluster count 2. This showed a minimum variation of RMSE value when the order cluster count is varied. This is due to the predictions are coming very close to the actuals. When user cluster count is increased, more the similar users are grouped.

- When user cluster count is set to 2, the RMSE variation showed the minimum values. By considering this variation we cannot conclude that the predictions are very close to the actuals. When analyzing the dissimilarity error distribution when user cluster count is equal to 2 scenarios, we see an impulse at error bin with the value 0 (Refer to Figure 4.2, Figure 4.6 and Figure 4.10). This type of behavior is not seen when the cluster count is equal to 5 (Refer to Figure 4.4, Figure 4.8, and

Figure 4.12). The reason for this behavior is that the prediction and the actual have less similar items rated. When calculating the dissimilarity, only the common items of the prediction and the actual is taken. This is explained in the below example.

Table 4.13: Learning module sequence prediction and actual

Modules	c1	c2	c3	c4	c5	r1	r2	r3	r4	s1	s2	s3	g1	g2	g3	g4	o1	o2	o3
Prediction$(\boldsymbol{O_1})$	2	0	-1	1	3	-1	-1	-1	-1	4	6	-5	-1	7	10	-1	8	-1	9
Actual$(\boldsymbol{O_2})$	1	-1	-1	0	-1	-1	-1	2	-1	-1	-1	-1	5	-1	-1	-1	-1	-1	-1

When considering O_1 and O_2 in Table 4.13, we have only c_1 and c_4 are in common for the comparison of the orders. The orders can then be narrowed as

$O_1 = c_1 > c_4$ and $O_2 = c_1 > c_4$

$r(\quad _1, c_1) = 1, r(O_1, c_4) = 2$

$r(O_2, c_1) = 1, r(O_2, c_4) = 2$

The similarity is defined as described in **Error! Reference source not found.**

$$\rho = 1 - \frac{6 \times \ \Sigma_{x \epsilon X1}\big(r(O_1, x) - r(O_2, x)\big)^2}{|X_1|^3 - |X_1|} = 1 - \frac{6 \times ((1-1)^2 + \ (1-2)^2}{|2|^3 - 2} = 1$$

Dissimilarity $d = 1 - \rho = 1 - 1 = 0$

- By analyzing Figure 4.13 and Figure 4.14, it can be seen that the RMSE values have followed almost a similar pattern, when order cluster count is equal to 3 and 4. When user cluster count and order cluster count are equal to 3, RMSE values have showed a minimum in the above 2 scenarios. Hence, for the above two datasets the optimum cluster counts would be 3 for both users and orders.
- In both the above cases, when user or order cluster counts are equal to 2, it has shown a behavior such that it can be considered as an outlier.

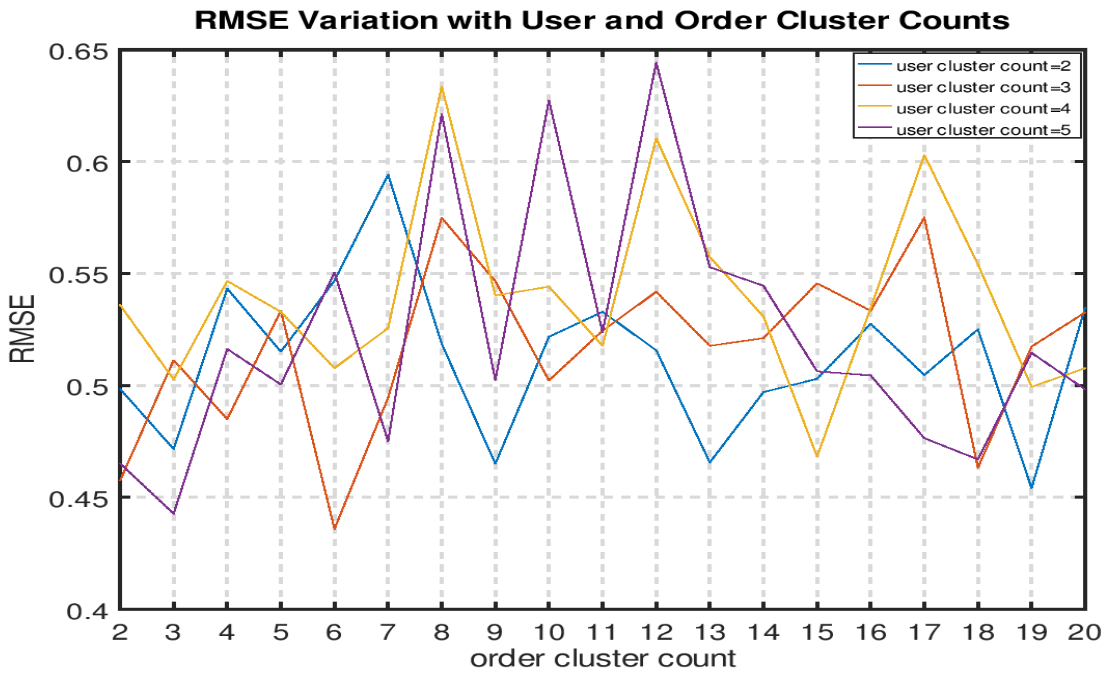


Figure 4.15: RMSE variation

We tested further the variation of RMSE values with the order and cluster counts. It can be observed that the RMSE variation is unique for the dataset and it cannot be generalized by giving effective user and order cluster counts in common. This must be identified by tuning the cluster count parameters according to the organization (i.e., Effective parameters differ from dataset to dataset.). When the dataset grows with time in an organization, the system should be tuned with correct cluster counts. This fact becomes obvious when considering the RMS variation of the Figure 4.15. For that dataset, the optimum cluster counts would be three and six for users and orders respectively.

4.4

Summary

When user cluster count is increased, the RMSE variation showed minimum values against the other order cluster counts because when user cluster count is increased, more the similar users are grouped. RMSE variation is unique for the dataset and optimal user and order cluster count for a data set can be identified when it gives the least RMSE value. When the dataset grows with time in an organization, the system should be tuned with optimal cluster counts.

CHAPTER 5

CONCLUSIONS

5.1 Summary

The focus on organizational learning is more than ever it was, due to the highly dynamic nature of the organizational environment. When a new employee joins an organization, he/she should be guided through a proper orientation process according to the up-to-date requirements of the organization. At the same time, the learning process should be in accordance with the interests of the employee. To address the above two requirements simultaneously, recommendation systems which is one of the widely used applications of machine learning can be used. Though machine learning is used extensively from book recommendation systems to complex expert systems in the medical field, there is less focus on recommender systems in the organizational learning process.

In this research, we proposed a solution to address the user cold-start problem in recommender systems for organizational learning. Demographic and learning module preference information of all the current employees are gathered and clustered accordingly. When a new employee joins the organization, the only information that we are exposed about the new employee is his or her demographic information. Having identified similar employees using the clustering of available demographic attributes of the new user, proper order cluster is selected so that maximum number of order preferences of the users in the identified order cluster are grouped together. Then, the recommendations are generated by considering the learning module sequences which have the least dissimilarities to the other sequences in the cluster.

The main challenge that we faced during the research was to find a proper dataset for the purpose. Therefore, we generated a synthetic dataset by analyzing the employee composition of a medium-scale research and development organization. The dataset was generated so that it can be configured easily to cater to the employee distribution changes of the organization.

We analyzed the recommendation effectiveness against the number of clusters that we use in user and order clustering. It was identified that each dataset gives better results for a specific user and order cluster count. For that optimal user and order cluster count which has the least RMSE value, the generated recommendations for 71% of the test data are more than 90% similar to the actual module orders, showing that the system can generate effective recommendations. The results and the analysis gave some insights to the researchers who wish to conduct more research in this area.

5.2 Research Limitations

Recommender systems for organizational leaning is a new area of research and therefore there is very little literature on that. In this research we encountered several limitations for which we were unable to implement solutions in this research scope.

Currently our recommender system supports either numeric or categorical data, but not a combination of both. Hence, the system is unable to cluster a user dataset which has both numeric and categorical values. To use that dataset, a conversion from numeric data to categorical data or vice versa must happen.

The selection of demographic attributes to be used in user clustering was done statically. The system does not analyze the effectiveness of the usage of the user attribute in generating recommendations.

In user clustering, we are using k-modes algorithm since the dataset comprises of categorical values. The dataset initially clustered using random and distinct modes selected from the dataset. In some scenarios, the initial selection of modes does not yield better results, since k-modes clustering gives a locally optimum result.

When testing the recommender system, a synthetic dataset was used, and it was generated after a research on the employee composition of a research and development-based organization. Dataset generation was done feeding a time-based random seed to the system so that the evaluation of the recommender system can be justified with the amount of randomness added into the dataset. Several rounds of execution of the recommendation system on the same synthetic dataset gave us the very similar results as elaborated in Performance Evaluation chapter. Within the limits of the logic implemented in the synthetic dataset generation, the results were satisfactory enough. The necessity to evaluate the performance of the system with a real-world dataset is identified as a requirement and in this research the evaluation is limited only to a synthetic dataset.

5.3 Future Work

Recommendation systems for organizational learning has become one of the pressing needs, nowadays. Though there are many recommendations systems which can recommend items, there is a very few systems available which can recommend a series of items which are sorted according to a preference. Hence, this research is of much importance and this can be enhanced further as suggested below.

When clustering the users according to their user information, we are using a fixed set of demographic features. In some scenarios, using the same features for clustering might not be effective as explained under Section 5.2. When suggesting learning modules for a sales and marketing employee, usage of gender as feature might be of less importance. Instead previous working place might be a good candidate for a feature. Based on the cluster similarity level, if a module can be designed so that it can dynamically choose relevant demographic features that will be beneficial in enhancing the final recommendation outcome.

In the current system, we are not taking a feedback from the given recommendation in such way whether it is useful or not. If we are taking the feedback of the current recommendation and uses it in next recommendations, when time passes the system will be more accurate.

In the current system, the system addresses only the user cold-start problem. There is another cold-start problem that arises when introducing new modules to the system which is called as item cold-start problem. When new modules are introduced to the system, it is difficult for the system to generate recommendations including that module since it doesn’t have historical ranking information. The only way to introduce the new module to the system and start including that in recommendation is to employ a content-based filtering mechanism. This feature will add immense value to the system.

REFERENCES

[1] P. D. J. S. Brown, Organizational Learning and Communities of Practice: Towards a Unified View of Working, Learning, and Innovation, The Institute of Management Sciences, 1991.

[2] Netflix, "Netflix Price," [Online]. Available: https://www.netflixprize.com/. [Accessed 20 12 2018].

[3] J. Christensen, D. Braziunas, L. Xie, S. Sedhain, and S. Sanner, "Social collaborative filtering for cold-start recommendations," 2014.

[4] H. Jindal and S. K. Singh, "A hybrid recommendation system for cold-start problem using online commercial dataset," *International Journal of Computer Engineering and Applications*, vol. Volume VII, no. Issue I, July, pp. 100-114, 2014.

[5] M. H. Nadimi-Shahraki and M. Bahadorpour, "Cold-start problem in collaborative recommender systems: Efficient methods based on ask-to-rate technique," *Journal of Computing and Information Technology,* 2014.

[6] N. M. Dixon, The Organizational Learning Cycle: How We Can Learn Collectively, London: Routledge, 2017.

[7] T. Kamishima and J. Fujiki, "Clustering Orders," 2010.

[8] "Collaborative filtering," [Online]. Available: https://en.wikipedia.org/wiki/Collaborative_filtering. [Accessed 23 Feb 2019].

[9] X. Zhao, "Cold-Start Collaborative Filtering," *ACM SIGIR Forum,* 2016.

[10] Maryam Nayebzadeh, Akbar Moazam, Amir Mohammad Saba, Hadi Abdolrahimpour, Elham Shahab, "An Investigation on Social Network Recommender Systems and Collaborative Filtering Techniques".

[11] M. J. Pazzani and D. Billsus, "Content-based Recommendation Systems".

[12] S. Biswas, L. V. Lakshmanan and S. Basu Ray, "Combating the Cold Start User Problem in Model Based Collaborative Fil-tering," 2016.

[13] M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon and H. Zha, Learning Multiple-Question Decision Trees for Cold-Start Recommendation.

[14] M. Vozalis and K. G. Margaritis, "COLLABORATIVE FILTERING ENHANCED BY DEMOGRAPHIC CORRELATION".

[15] "MovieLens," [Online]. Available: https://grouplens.org/datasets/movielens/. [Accessed 20 12 2018].

[16] M. Ferenc, "Content based Recommendation from Explicit Ratings," Charles University, Prague, 2016.

[17] J. C. Alva Liu, "Using Demographic Information to Reduce the New User Problem in Recommender Syste," STOCKHOLM, SWEDEN , 2017.

[18] A. Liu and J. Callvik, "Using Demographic Information to Reduce the New User Problem in Recommender Systems," 2017.

[19] L. Safoury and A. Salah, "Exploiting User Demographic Attributes for Solving Cold-Start Problem in Recommender System," *Lecture Notes on Software Engineering,* 2013.

[20] J. L. a. W. Zhang, *Conditional Restricted Boltzmann Machines for Cold Start Recommendations,* 2014.

[21] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values," *Data Mining and Knowledge Discovery,* pp. 283-304, 1998.

[22] T. K. a. J. Fujiki, "Clustering Orders," Springer-Verlag Berlin Heidelberg, 2003.

[23] T. Kamishima and S. Akaho, "Efficient clustering for orders," *Studies in Computational Intelligence,* 2009.

[24] L. Yu and X. Yang, "Collaborative filtering recommendation based on preference order," in *IFIP International Federation for Information Processing,* 2008.

[25] "Data Clustering Algorithms," [Online]. Available: https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm. [Accessed 01 02 2019].

[26] "Kendall rank correlation coefficient," [Online]. Available: https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient. [Accessed 01 Feb 2019].

[27] T. Kamishima, "SUSHI Preference Data Sets," [Online]. Available: http://www.kamishima.net/sushi/. [Accessed 20 12 2018].

[28] "Employee Attrition," 2019. [Online]. Available: https://www.kaggle.com/patelprashant/employee-attrition. [Accessed 20 12 2018].

[29] T. K. a. S. Akaho, "Efficient Clustering for Orders," in *Mining Complex Data, Vol.165 of Studies in Computational Intelligence*, Springer, 2009, p. 261–280.

[30] R. Rains and H. Carpenter, James Naismith : the man who invented basketball, Temple University Press, 2009, p. 198.

[31] C. S. D. Y. U. J. Osama Abu Abbas, "Comparisions Between Data Clustering Algorithms," *The International Arab Journal of Information Technology*, vol. 5, no. 3, pp. 320-325, 2008.