Active Learning and User Segmentation for the Cold-start Problem in Recommendation Systems

Rabaa Alabdulrahman[1], Herna Viktor[1] and Eric Paquet[1,2]
[1]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
[2]National Research Council of Canada, Ottawa, Canada

Keywords: Recommendation Systems, Collaborative Filtering, Cold-start, Active Learning.

Abstract: Recommendation systems, which are employed to mitigate the information overload faced by e-commerce users, have succeeded in aiding customers during their online shopping experience. However, to be able to make accurate recommendations, these systems require information about the items for sale and information about users' individual preferences. Making recommendations to new customers, with no prior data in the system, is therefore challenging. This scenario, called the "cold-start problem," hinders the accuracy of recommendations made to a new user. In this paper, we introduce the popular users personalized predictions (PUPP) framework to address cold-starts. In this framework, soft clustering and active learning is used to accurately recommend items to new users. Experimental evaluation shows that the PUPP framework results in high performance and accurate predictions. Further, focusing on frequent, or so-called "popular," users during our active-learning stage clearly benefits the learning process.

1 INTRODUCTION

Investors and businesses are turning increasingly toward online shopping to maximize their revenues. However, with the rapid development in technology and the increase in available online businesses, clients are increasingly overwhelmed by the amount of information to which they are submitted. Recommendation systems were introduced to aid customers in dealing with this vast amount of information and guide them in making the right purchasing decisions (Lu et al., 2015). Yet, a persistent drawback is that these systems cannot always provide a personalized or human touch (Kim et al., 2017). Intuitively, when a business owner does not directly, or verbally, interact with the customer, he or she has to rely on the data collected from previous purchases. In general, research has shown that vendors are better at recognizing and segmenting users (Kim et al., 2017) than existing recommendation systems are. This observation holds especially for new customers.

The primary purpose of recommendation systems is to address the information overload users experience and to aid the users in narrowing down their purchase options. These systems aim to achieve this by understanding their customers' preferences not only by recognizing the ratings they give for specific items, but also by considering their social and demographic information (Bhagat et al., 2014). Consequently, these systems create a database for both items and users where ratings and reviews of these items are collected (Minkov et al., 2010). Intuitively, the more information and ratings collected about the user, the more accurate the recommendations (Karimi et al., 2015).

Generally speaking, recommendation systems fall primarily into three categories. These are content-based filtering (CBF) (Tsai, 2016), collaborative filtering (CF) (Liao and Lee, 2016), and hybrid approaches (Ntoutsi et al., 2014). These systems rely on two basic inputs: the set of users in the system, U (also known as customers), and the set of items to be rated by the users, I (also known as the products) (Bakshi et al., 2014).

All these systems employ matrices based on past purchase patterns. With CBF, the system focuses on item matrices where it is assumed that if a user liked an item in the past, he or she is more inclined to like a similar item in the future (Minkov et al., 2010, Acosta et al., 2014).These systems therefore study the attributes of the items (Liao and Lee, 2016). On the other hand, CF systems focus on user-rating matrices, recommending items that have been rated by other

Alabdulrahman, R., Viktor, H. and Paquet, E.
Active Learning and User Segmentation for the Cold-start Problem in Recommendation Systems.
DOI: 10.5220/0008162901130123
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 113-123
ISBN: 978-989-758-382-7

users with preferences similar to those of the targeted user (Saha et al., 2015). These systems therefore rely on the historic data of user rating and similarities across the user network (Minkov et al., 2010). Lastly, hybrid systems employ both CBF and CF approaches. These systems concurrently consider items based on users' preferences and on the similarity between the items' content (Acosta et al., 2014). In recent years, research has trended toward hybrid systems (Liao and Lee, 2016). Another growing trend is the use of data mining and machine learning algorithms (Bajpai and Yadav, 2018) to identify patterns in user's interests and behaviours (Bajpai and Yadav, 2018).

In this paper, we present the popular users personalized predictions (PUPP) framework, designed to address the cold-start problem. In our framework, we combine cluster analysis and active learning, or so-called "user-in-the-loop," to assign new customers to the most appropriate groups. We create user segmentations via cluster analysis. Then, as new users enter the system, classification methods assign them to the right groups. Based on this assignment, we apply active learning. Cluster analysis is used to group similar user profiles, while active learning is employed to learn the labels associated with these groups.

The remainder of this paper is organized as follows: Section 3 presents our PUPP framework and its components; Section 4 discusses our experimental setup and data preparation, and Section 5 discusses the results. Section 6 concludes the paper.

2 RELATED WORK

In active learning, or "user in the loop," a machine learning algorithm selects the best data samples to present to a domain expert for labelling. These samples are then used to bootstrap the learning process, in that these examples are subsequently used in a supervised learning setting. In recommendation systems, active learning presents a utility-based approach to collect more information about the users (Karimi et al., 2011). Intuitively, showing the user a number of questions about their preferences, or asking for more personal information such as age or gender, may benefit the learning process (Wang et al., 2017).

The literature addressing the cold-start problem (Gope and Jain, 2017) is divided into implicit and explicit approaches. On the implicit side, the system utilizes existing information to create its recommendations by adopting traditional filtering strategies or by employing social network analysis. For instance, Wang et al. relies on an implicit approach based on questionnaires and active learning to engage the users in a conversation aimed at collecting additional preferences. Based on the previously collected data, the user's preferences and predictions, the active-learning method is used to determine the best questions to be asked (Wang et al., 2017). Similarly, explicit standard approaches may be extended by incorporating active-learning methods in the data-collection phase (Gope and Jain, 2017). For instance, Fernandez-Tobias et al. use an explicit framework to compare three methods based on the users' personal information (Fernandez-Tobias et al., 2016). First, they include the personal information to improve a collaborative filtering framework performance. Then they use active learning to further improve the performance by adding more personal information from existing domains. Finally, they supplement the lack of preference data in the main domain using users' personal information from supporting domains.

There are many examples in the literature of machine learning techniques being utilized in recommendation systems. Although hybrid filtering was proposed as a solution to the limitations of CBF and CF, hybrid filtering still does not adequately address issues such as data sparsity, where the number of items in the database is much larger than the items a customer typically selects, and grey sheep, which refers to atypical users. Further, a system may still be affected when recommending items to new users (cold starts). To this end, Pereira and Hruschka (2015) proposed a simultaneous co-clustering and learning (SCOAL) framework to deal with new users and items. According to their data-mining methodology, a cluster analysis approach is integrated in the hybrid recommendation system, which results in better recommendations (Pereira and Hruschka, 2015).

In addition, performances may be improved by implementing classification according to association rule techniques (Lucas et al., 2012). Such a system was built in order to deal with sparsity and scalability in both CF and CBF approaches. In (Soundarya et al., 2017), clustering and classifications are used to identify criminal behavior. Also, Davoudi and Chatterjee in (Davoudi and Chatterjee, 2017) use clustering to recognize profile injection attacks. Both methods utilize clustering techniques to create user segmentations prior to classification.

In our PUPP framework, we extend this approach when creating our user groups. Our PUPP framework is presented in the next section.

Algorithm 1: Popular user personalized prediction (PUPP).

```
Input
  R: a set of r class labelled
  training inputs;
  anonA_j: Clustering algorithm;
  k: Number of clusters;
  R_i: ratings per user;
  Y: class label of r;
  x: unkown sample;
User Segmentation
  1- A_j discover k objects from D as
       initial cluster centre
  2- Repeat:
      -(re)assign each object to
           cluster according to
           A_j distance measure
      -Update A_j
      -Calculate new value
 Until no change
  3- Output models (M_1,...M_n)
Initialization for classification and
prediction:
  1- Classify (R_i,Y,x);
  2- Output classification model
  3- Test model on R_i^test
  4- Output prediction list
Initialization for active user rating
stage:
  1- Select 2 highest prediction rate
  2- Return 2 highest r_n,r_k
  3- Remove r_n,r_k from R_i^test
Append r_n,r_k to R_i^train
```

3 FRAMEWORK

Our PUPP framework for prediction-based personalized active learning is tested for its ability to address the cold-start problem using a clustering and two classification algorithms. First, we use soft clustering, the EM method, to create our user segmentation. Then, we use k-NN (EM-k-NN framework) and subspace method (EM-subspace framework) with k-NN as a base classifier for the clustered data set. The results from these two frameworks are compared with the traditional CF (using k-NN) framework, which constitutes our baseline.

In active learning, the learner will query the instances' labels using different scenarios. In this framework, we use pool-based sampling, wherein instances are drawn from a pool of unlabelled data (Elahi et al., 2016). These instances are selected by focusing on the items with the highest prediction rates, using explicit information extraction (Elahi et al., 2014, Elahi et al., 2016). As mentioned above, active learning is an effective way to collect more information about the user. Hence, in this framework, if a new user rates a small number of highly relevant items, that may be sufficient for first analyzing the items features and then calculating the similarity to other items in the system.

3.1 Framework Components

Figure 1 shows the steps involved in the PUPP framework. Initially, we employ cluster analysis to assign customers to groups, using a soft clustering approach (Mishra et al., 2015). This results in overlapping clusters, where a user may belong to more than one cluster. Intuitively, this approach accurately reflects the human behavioural complexity. Once the groups are created, we apply two splitting methods to generate the training and test sets. We use a random split method—a common practice in machine learning. In addition, we designed an approach that focusses on so-called "popular" users, as detailed in section 4.3. The cold-start problem is addressed as follows. When a new user logs in to the system, the initial model is employed to find user groups with similar preferences. In our approach, we employ the k-nearest neighbour (k-NN) algorithm to assign a new user to a given group (Sridevi et al., 2016, Katarya and Verma, 2016). A machine learning algorithm is used to evaluate and potentially improve the group assignment. To this end, a human expert evaluates the predictive outcome and selects two records (for each user) with the highest prediction rate. These are appended to the training set (Flach, 2012, Elahi et al., 2014). Then, a new model is trained against the new, enlarged data set. This process is repeated until a stopping criterion is met. The following two subsections will discuss these steps in detail.

3.1.1 Cluster Analysis Component

Cluster analysis is an unsupervised learning technique used to group data when class labels are unknown (Flach, 2012). Cluster analysis allows for determining the data distribution while discovering patterns and natural groups (Pujari et al., 2001). In an e-commerce setting, the goal is to maximize the similarity of individuals within the group while minimizing the similarity of characteristics between groups (Cho et al., 2015). Therefore, similarities in opinion, likes and ratings of the users are evaluated for each group, (Isinkaye et al., 2015).

Numerous options for algorithm are available for cluster analysis. With soft clustering, the groups may overlap; as a result, a data point may belong to more than one group. Intuitively, in recommendation systems, users' group memberships are often fuzzy. In previous work done by (Alabdulrahman et al., 2018), the authors compare the performance of different clustering and classification techniques, and concluded that expectation maximization (EM) clustering outperforms the other algorithms in most cases. We therefore use EM clustering in our PUPP framework. The EM algorithm proceeds by re-estimating the assigned probabilities, adjusting the mean and variance values to improve the assignment points, iterating until convergence (Bifet and Kirkby, 2009).

3.1.2 Classification Component

In contrast to clustering, with classification, also called "supervised learning," the system learns from examples where the class labels are known, from which it develops classification models that it uses to predict unknown instances (Pujari et al., 2001). Since our framework is based on a CF recommendation system, the k-nearest neighbor (k-NN) classifier is employed in the PUPP framework. The latter also acts as a baseline in our experimental evaluations (Sridevi et al., 2016, Katarya and Verma, 2016).

In addition, we use the random subspace ensemble-based method, whose advantages have been demonstrated in our earlier research (Alabdulrahman et al., 2018). Specifically, ensemble improves the classification accuracy of a single classifier (Witten et al., 2016). Also, in the random subspace method, the learning process will focus on the features instead of the examples. Hence, this approach will evaluate all features in the subspace and select the most informative ones based on the selected features. That is, feature subsets will be created randomly with replacement from the training set. Then each individual classifier will learn from the created subsets while considering all training examples (Sun, 2007).

4 EXPERIMENTAL SETUP

The experimental evaluation was conducted on a desktop with an Intel i7 Core 2.7 GHz processor and 16 GB of RAM. Our framework was implemented using the WEKA data-mining environment (Frank et al., 2016).

4.1 Dataset Description

We used two data sets to evaluate our PUPP framework. We tested our framework on the Serendipity data set (Kotkov et al., 2018), which contains 2,150 movie ratings, as well as descriptions of the movies and users' responses to questionnaires about the movies they have rated.

The second data set used is the famous MovieLense data set (Harper and Konstan, 2016). This data set, which is well-known in recommendation system research, contains 100,836 ratings on 9,742 movies.

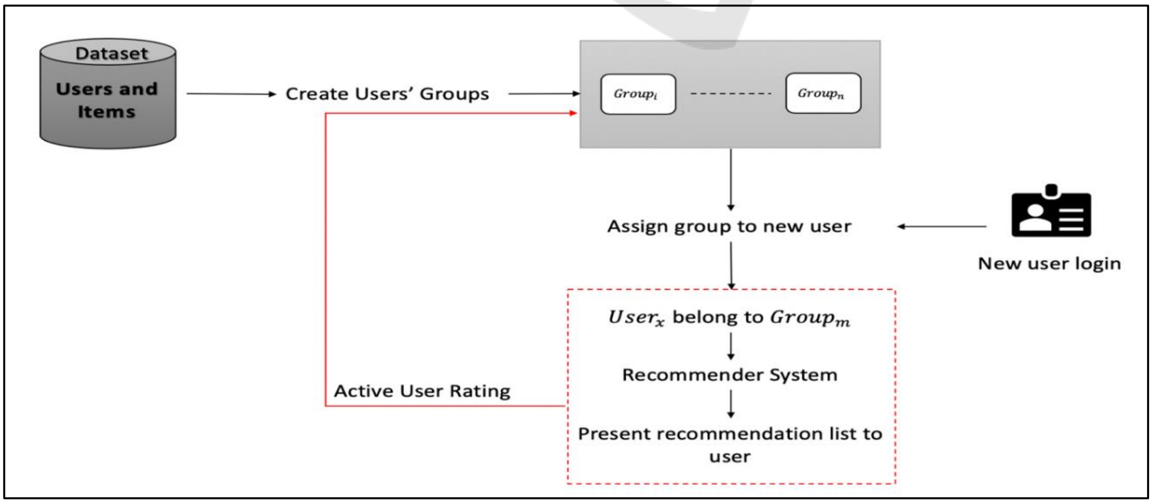

Figure 1: Outline of the PUPP framework.

4.2 Dataset Pre-processing

Initially, the movie genres were determined with the help of statista.com and imdb.com, as shown in Table 1.

Table 1: Genre Coding.

Genre	Code	Genre	Code
Adventure	1	Children	8
Action	2	Documentary	9
Drama	3	Sci-Fi	10
Comedy	4	Musical	11
Thriller (crime)	5	Animation	12
Horror	6	Others	13
Romantic Comedy - Romance			7

Additional preprocessing steps involved removing all ratings lower than 2.5 out of 5 to focus the recommendations on popular movies. Also, for the Serendipity data set, attributes S_1 to q provide information about survey answers. These answers relate to users' experience using the recommendation system and of the movie suggestions presented to them. If less than 5 questions were answered, the record was removed for lack of information. In total, we eliminated 18 records.

4.3 Experimental Setup

In this experimental evaluation, we employed the EM cluster analysis algorithm to segment users into potentially overlapping clusters. We utilized two classifiers, namely k-NN and the random subspace ensemble method with k-NN as the base learner. The value of k was set to 5, while the number of features to be included in a subspace was fixed at 0.50 (50%); both values were set by inspection.

As described above, our PUPP framework includes a prediction-based personalized active learning component. In our implementation, active learning consists of iterations, where in each iteration we select, for each user, the two (2) records with the highest prediction rate. After labelling, these two records are appended to the original training set and removed from the test set. In the present work, the number of iterations is limited to five to process the request in real time. Our model was evaluated using the 10-fold cross validation approach.

4.4 Cold-start Simulation

This section explains the approach for simulating the cold-start problem. We use two techniques to split our data sets, random split and popularity split. Each technique was evaluated against the traditional k-NN, EM-k-NN, and EM-subspace.

In the random split method, the data set is divided randomly between (70%) training and (30%) testing sets, where the training data set contains the known rating by the system, as already provided by the users. The test set, on the other hand, includes unknown ratings. Note that this approach is commonly taken in the literature (Flach, 2012).

Popularity split evaluates the popularity associated with the users and the items. In this scenario, we consider the users with the highest number of ratings and refer to them as "popular users," i.e. those who use the system frequently. These users are removed from the training set and used as test subjects for cold-start simulations. A "removed" user must have rated at least 5 popular movies to be considered for removal, 5 movies was determined by inspection. By removing members in this manner, we increase the chance for the system to find similarities among more users' segmentations in the system. This is, as far as we are aware, the first research to use the notion of popular, or frequent, users for guiding the determination of the recommendations made to cold starts. We do so based on the assumption of trends (such as in clothing recommendation systems) and top rating systems for movies or music (such as in Netflix and iTunes).

For a user to be considered as a test subject, the following criteria must be met:

The user must have a high number of ratings, as opposed to random split, where the number of items rated by the user is ignored, as shown in Table 8.

The rated movies must have a rating greater than 2.5 (out of 5).

User rated popular movies. The non-popular movie creates a grey sheep problem, which refers to users who are atypical. We do not address grey sheep in the present work.

We illustrate our results with 10 users. Table 8 shows some information about the selected users in the MovieLense data set. It is important to stress that we need to ensure that each selected user does not have any remaining records in the training set. This verification ensures a properly simulated cold-start problem.

4.5 Evaluation Criteria

As mentioned earlier, k-NN is widely employed in CF systems. Consequently, it is used as our baseline as well as the base learner in our feature subspace ensemble. The mean absolute error (MAE) measure, which indicates the deviation between predicted and

Table 2: Model accuracy for the MovieLense dataset.

		Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Popularity Split	kNN	38.50	38.43	38.28	38.37	38.44
	EM-kNN	98.45	98.47	98.44	98.50	98.47
	EM-Subspace	**98.81**	**99.18**	**98.83**	**98.86**	**98.68**
Random Split	kNN	39.08	38.94	38.91	39.11	39.22
	EM-kNN	81.88	81.76	81.81	81.87	81.83
	EM-Subspace	**86.55**	**88.51**	**87.23**	**87.14**	**86.38**

Table 3: Model accuracy for the Serendipity dataset.

		Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Popularity Split	kNN	42.18	42.35	43.29	43.07	44.83
	EM-kNN	81.84	81.63	81.87	82.58	**82.58**
	EM-Subspace	**83.14**	**83.18**	**84.04**	**84.17**	81.60
Random Split	kNN	43.87	44.18	45.08	45.24	46.51
	EM-kNN	64.43	65.07	65.23	65.33	66.47
	EM-Subspace	**67.78**	**68.40**	**69.27**	**69.21**	**69.77**

Table 4: MAE results for popularity split test method.

	kNN		EM-kNN		EM-Subspace	
	Serendipity	MovieLense	Serendipity	MovieLense	Serendipity	MovieLense
Iteration 1	0.214	0.237	0.120	0.039	0.167	0.106
Iteration 2	0.213	0.237	0.119	0.039	0.170	0.108
Iteration 3	0.211	0.237	0.119	0.039	0.167	0.110
Iteration 4	0.211	0.237	0.118	0.039	0.167	0.110
Iteration 5	0.210	0.237	0.118	0.039	0.167	0.095

Table 5: MAE results for random split rest method.

	kNN		EM-kNN		EM-Subspace	
	Serendipity	MovieLense	Serendipity	MovieLense	Serendipity	MovieLense
Iteration 1	0.210	0.237	0.175	0.116	0.204	0.164
Iteration 2	0.210	0.237	0.173	0.116	0.201	0.161
Iteration 3	0.209	0.237	0.171	0.116	0.199	0.168
Iteration 4	0.206	0.237	0.170	0.116	0.201	0.166
Iteration 5	0.205	0.236	0.168	0.116	0.198	0.163

Table 6: F-measure results for popularity split method.

	kNN		EM-kNN		EM-Subspace	
	Serendipity	MovieLense	Serendipity	MovieLense	Serendipity	MovieLense
Iteration 1	0.594	0.352	0.818	0.984	0.830	0.988
Iteration 2	0.595	0.351	0.816	0.985	0.831	0.992
Iteration 3	0.604	0.350	0.819	0.984	0.840	0.988
Iteration 4	0.602	0.351	0.826	0.985	0.841	0.989
Iteration 5	0.619	0.352	0.826	0.985	0.815	0.987

Table 7: F-measure for the random split test method.

	kNN		EM-kNN		EM-Subspace	
	Serendipity	MovieLense	Serendipity	MovieLense	Serendipity	MovieLense
Iteration 1	0.610	0.359	0.629	0.817	0.656	0.864
Iteration 2	0.613	0.358	0.636	0.816	0.660	0.884
Iteration 3	0.622	0.357	0.636	0.816	0.671	0.872
Iteration 4	0.623	0.360	0.637	0.817	0.668	0.871
Iteration 5	0.635	0.361	0.650	0.817	0.673	0.863

Table 8: Test subject from MovieLense dataset.

Popular users		Random split	
User ID	#Rating	User ID	#Rating
599	1096	1	226
474	1280	225	67
414	1491	282	190
182	805	304	194
477	772	34	56
603	773	374	32
448	698	412	90
288	724	450	48
274	780	510	74
68	677	602	118

actual rating, is employed as a predictive measure (Chaaya et al., 2017). In addition, the model accuracy and the F-measure (geometric mean of recall and precision) are employed to determine the usefulness of the recommendation list (Chaaya et al., 2017).

5 RESULTS AND DISCUSSIONS

In this section we discuss the performance of the model in terms of accuracy, MAE, and F-measure (Chaaya et al., 2017). Individual users are taken into account in our evaluation.

5.1 System Evaluation

Table 2 and Table 3 show the classification accuracy of the PUPP framework system for random and popularity split. In both cases, active learning improves the performance—by 39.66% for the Serendipity data set and 59.95% for the MovieLense data set. When considering the random split results, we notice increases of 20.56% for the Serendipity data set and 42.8% for the MovieLense data set. These results are obtained using the EM clustering technique.

In addition, we enhanced the performance of the traditional CF framework by introducing the subspace method. Recall that instead of using the k-NN algorithm as a single classifier, we apply an ensemble subspace method using k-NN as a base learner and a subspace of 50% features. Again, we notice improvement over the traditional CF system. Specifically, the random split method improves results by 23.91% for the Serendipity data set and 47.47% for the MovieLense data set compared to the traditional framework. Also, using the popularity split method, the accuracy increases by 40.96% and 60.31%, respectively. From Figure 2 and Figure 3, one may conclude that the popularity split method always results in a much higher accuracy.

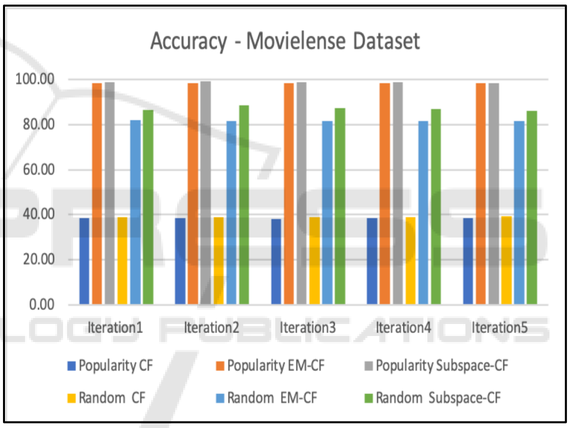

Figure 2: Accuracy for the MovieLense dataset.

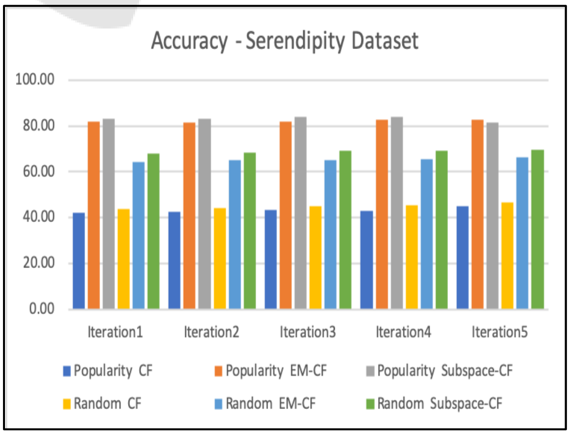

Figure 3: Accuracy for the Serendipity dataset.

Table 6 and Table 7 show the results for the F-measure, which again confirm the benefit of focusing on popular users while training. The same observation holds when the MAE metric is employed.

Table 9 shows a summary of the improvement in percentage over the traditional CF framework for both data sets. Notice that these improvements were calculated only for the first iteration, since we are interested in the immediate, cold-start problem. The outcome of the last four iterations confirms that the system can make appropriate recommendations to new users while performing adequately for existing users.

Table 9: Improvement in predictive accuracy measures for system-wide performance over traditional CF.

Framework	Accuracy Increase by %	F-measure Increase by %	MAE Decrease by %	Dataset
Popularity test method				
EM-CF	39.99	0.224	0.094	Serendipity
	59.95	0.632	0.198	MovieLense
EM-Subspace-CF	40.96	0.236	0.047	Serendipity
	60.31	0.636	0.131	MovieLense
Random split test method				
EM-CF	20.87	0.019	0.035	Serendipity
	42.80	0.581	0.243	MovieLense
EM-Subspace-CF	23.91	0.046	0.006	Serendipity
	47.47	0.628	0.195	MovieLense

5.2 Statistical Validation

This section discusses the results of our statistical significance testing, using the Friedman test: the confidence level was set to $a = 0.05$. That is, we wish to determine whether there is any statistical significance between the performance of the baseline CF method using k-NN and the two variants of our PUPP system (EM and EM-Subspace).

In this validation, the Friedman yields a p-value of 0.000171 for the Serendipity data set, and a p-value of 0.000139 for the MovieLense data set. Therefore, the null hypothesis is rejected for both data sets, which means there is a significant difference among the three frameworks. We report the results of the pairwise comparisons in Figure 4 and Figure 5.

Furthermore, to determine if there is a significant difference between each pair, we perform the Nemenyi post-hoc test. As shown in Table 10 there is a significant difference among three pairs: *EM-kNN versus kNN*, *EM-Subspace versus kNN*, and *kNN versus EM-kNN*. These results confirm that the system benefits from soft clustering and active learning. There is no statistical difference between the versions that use a baseline learning (k-NN) when compared to an ensemble, which indicates that a single classifier may be employed against these data sets. These results confirm our earlier discussion in which EM-k-NN and EM-subspace, when used with the popularity split method, have a significantly better performance when compared with random split. These two variants also outperform the traditional CF framework.

5.3 Prediction Rate

To further validate our approach, we considered the user prediction rate. In this section, the prediction rates for 10 users from the MovieLense data set are presented. From Table 11 one may see that EM-kNN has the best prediction rates of all. However, we noticed that after the third iteration, when random split is employed, the prediction rate begins to decrease, at least for some users. Also, by taking into consideration the overall performance of the system, it may be concluded that EM-subspace presents the best performance against these data sets when compared to the other two models.

6 CONCLUSION AND FUTURE WORK

In this paper, we presented the PUPP framework designed to address the cold-start problem in CF recommendation systems. Our results show the benefit of user segmentation based on soft clustering and the use of active learning to improve predictions for new users. The results also demonstrate the advantages of focusing on frequent or popular users to improve classification accuracy.

In our current approach, we included two classification algorithms in our experimentation, and we plan to extend this work to include other approaches. In our future work, we will also investigate the suitability of deep learning methods. Specifically, we are researching the use of deep composite models for optimal user segmentation and personalization (Zhang et al., 2019). Furthermore, this framework was tested in an offline setting; future plans include testing it in a real-world setting.

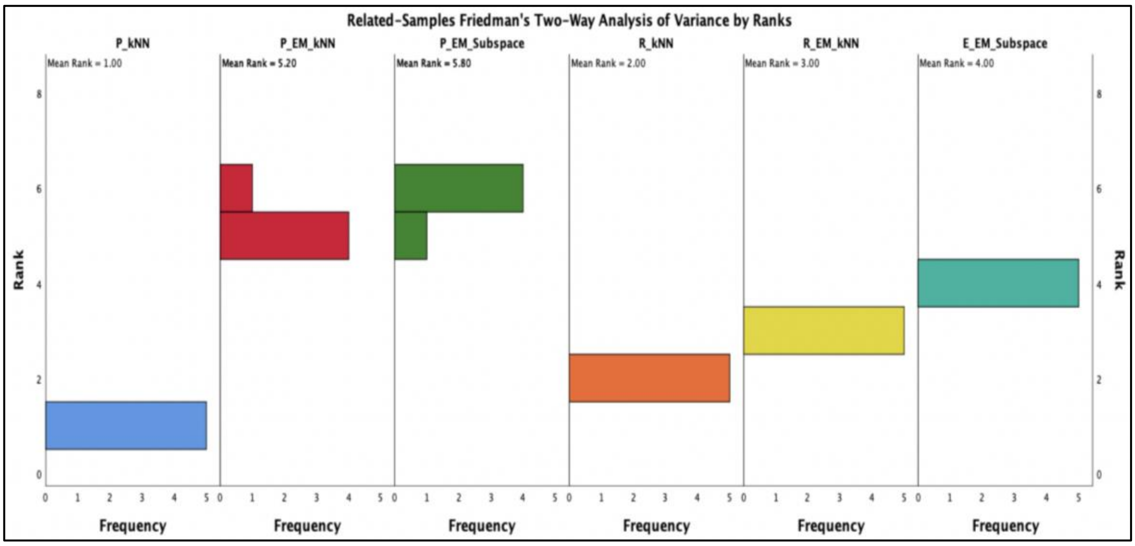

Figure 4: Friedman test mean ranks for the Serendipity dataset.

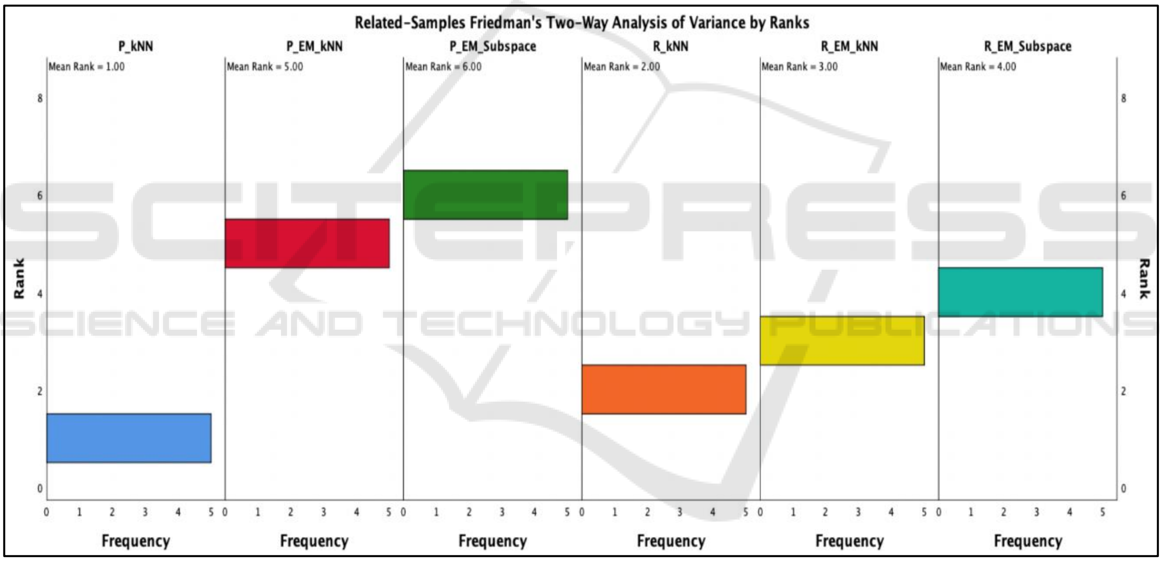

Figure 5: Friedman test mean ranks for the Serendipity dataset.

Table 10: Nemenyi $p - values$.

Serendipity Dataset					
	P-kNN	P-EM-kNN	P-EM-Subspace	R-kNN	R-EM-kNN
P-EM-kNN	**0.005178**				
P-EM-Subspace	**0.000708**	0.995925			
R-kNN	0.958997	0.074302	**0.016639**		
R-EM-kNN	0.538193	0.427525	0.168134	0.958997	
R-EM-Subspace	0.113891	0.91341	0.65049	0.538193	0.958997
	P-kNN	P-EM-kNN	P-EM-Subspace	R-kNN	R-EM-kNN
MovieLense Dataset					
	P-kNN	P-EM-kNN	P-EM-Subspace	R-kNN	R-EM-kNN
P-EM-kNN	**0.009435**				
P-EM-Subspace	**0.000343**	0.958997			
R-kNN	0.958997	0.113891	**0.009435**		
R-EM-kNN	0.538193	0.538193	0.113891	0.958997	
R-EM-Subspace	0.113891	0.958997	0.538193	0.538193	0.958997

Table 11: New user Prediction accuracy.

Popular user										
UserID	182	274	288	414	448	474	477	599	603	63
CF	80%	100%	100%	100%	100%	86%	100%	100%	85%	80%
EM-CF	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
EM-subspace-CF	91%	90%	91%	92%	91%	92%	91%	91%	91%	90%
Random split										
UserID	182	274	288	414	448	474	477	599	603	63
CF	71%	52%	48%	61%	41%	56%	71%	50%	55%	76%
EM-CF	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
EM-subspace-CF	63%	61%	61%	62%	58%	62%	59%	63%	61%	63%

REFERENCES

Acosta, O. C., Behar, P. A. & Reategui, E. B. 2014. Content recommendation in an inquiry-based learning environment. Frontiers in Education Conference (FIE), 2014. IEEE, 1-6.

Alabdulrahman, R., Viktor, H. & Paquet, E. 2018. Beyond k-NN: Combining Cluster Analysis and Classification for Recommender Systems. The 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018), 2018 Seville, Spain. KDIR 2018, 82-91.

Bajpai, V. & Yadav, Y. 2018. Survey Paper on Dynamic Recommendation System for E-Commerce. *International Journal of Advanced Research in Computer Science,* 9.

Bakshi, S., Jagadev, A. K., Dehuri, S. & Wang, G. N. 2014. Enhancing scalability and accuracy of recommendation systems using unsupervised learning and particle swarm optimization. *Applied Soft Computing,* 15**,** 21-29.

Bhagat, S., Weinsberg, U., Ioannidis, S. & Taft, N. 2014. Recommending with an agenda: Active learning of private attributes using matrix factorization. Proceedings of the 8th ACM Conference on Recommender systems, 2014. ACM, 65-72.

Bifet, A. & Kirkby, R. 2009. Data Stream Mining a Practical Approach. The University of Waikato: Citeseer.

Chaaya, G., Metais, E., Abdo, J. B., Chiky, R., Demerjian, J. & Barbar, K. 2017. *Evaluating Non-Personalized Single-Heuristic Active Learning Strategies for Collaborative Filtering Recommender Systems*.

Cho, Y. & Jeong, S. P. 2015. A Recommender System in u-Commerce based on a Segmentation Method. In Proceedings of the 2015 International Conference on Big Data Applications and Services, 2015. ACM, 148-150.

Davoudi, A. & Chatterjee, M. 2017. Detection of profile injection attacks in social recommender systems using outlier analysis. 2017 IEEE International Conference on Big Data (Big Data), 2017. IEEE, 2714-2719.

Elahi, M., Ricci, F. & Rubens, N. 2014. Active Learning in Collaborative Filtering Recommender Systems. *In:* HEPP, M. & HOFFNER, Y. (eds.) *E-Commerce and Webtechnologies.*

Elahi, M., Ricci, F. & Rubens, N. 2016. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review,* 20**,** 29-50.

Fernandez-Tobias, I., Braunhofer, M., Elahi, M., Ricci, F. & Cantador, I. 2016. Alleviating the new user problem in collaborative filtering by exploiting personality information. *User Modeling and User-Adapted Interaction,* 26**,** 221-255.

Flach, P. 2012. *Machine learning: the art and science of algorithms that make sense of data.* Cambridge University Press.

Frank, E., Hall, M. A. & Witten, I. H. 2016. The WEKA workbench. *Data mining: Practical machine learning tools and techniques,* 4.

Gope, J. & Jain, S. K. 2017. *A Survey on Solving Cold Start Problem in Recommender Systems*.

Harper, F. M. & Konstan, J. A. 2016. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis),* 5**,** 19.

Isinkaye, F. O., Folajimi, Y. O. & Ojokoh, B. A. 2015. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal,* 16**,** 261-273.

Karimi, R., Freudenthaler, C., Nanopoulos, A. & Schmidt-Thieme, L. 2011. Towards Optimal Active Learning for Matrix Factorization in Recommender Systems. *2011 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI).* IEEE.

Karimi, R., Freudenthaler, C., Nanopoulos, A. & Schmidt-Thieme, L. 2015. Comparing Prediction Models for Active Learning in Recommender Systems. Comparing Prediction Models for Active Learning in Recommender Systems, 2015. 171-180.

Katarya, R. & Verma, O. P. 2016. A collaborative recommender system enhanced with particle swarm optimization technique. *Multimedia Tools and Applications,* 75**,** 9225-9239.

Kim, H. M., Ghiasi, B., Spear, M., Laskowski, M. & Li, J. 2017. Online serendipity: The case for curated recommender systems. *Business Horizons,* 60**,** 613-620.

Kotkov, D., Konstan, J. A., Zhao, Q. & Veijalainen, J. 2018. Investigating serendipity in recommender systems based on real user feedback. Proceedings of the 33rd Annual ACM Symposium on Applied Computing, 2018. ACM, 1341-1350.

Liao, C.-L. & Lee, S.-J. 2016. A clustering based approach to improving the efficiency of collaborative filtering recommendation. *Electronic Commerce Research and Applications,* 18**,** 1-9.

Lu, J., Wu, D. S., Mao, M. S., Wang, W. & Zhang, G. Q. 2015. Recommender system application developments: A survey. *Decision Support Systems,* 74**,** 12-32.

Lucas, J. P., Segrera, S. & Moreno, M. N. 2012. Making use of associative classifiers in order to alleviate typical drawbacks in recommender systems. *Expert Systems with Applications,* 39**,** 1273-1283.

Minkov, E., Charrow, B., Ledlie, J., Teller, S. & Jaakkola, T. 2010. Collaborative future event recommendation. Proceedings of the 19th ACM international conference on Information and knowledge management, 2010. ACM, 819-828.

Mishra, R., Kumar, P. & Bhasker, B. 2015. A web recommendation system considering sequential information. *Decision Support Systems,* 75**,** 1-10.

Ntoutsi, E., Stefanidis, K., Rausch, K. & Kriegel, H. P. 2014. Strength lies in differences: Diversifying friends for recommendations through subspace clustering. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, 2014. ACM, 729-738.

Pereira, A. L. V. & Hruschka, E. R. 2015. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowledge-Based Systems,* 82**,** 11-19.

Pujari, A. K., Rajesh, K. & Reddy, D. S. 2001. Clustering techniques in data mining-A survey. *IETE Journal of Research,* 47**,** 19-28.

Saha, T., Rangwala, H. & Domeniconi, C. 2015. Predicting preference tags to improve item recommendation. Proceedings of the 2015 SIAM International Conference on Data Mining, 2015. SIAM, 864-872.

Soundarya, V., Kanimozhi, U. & Manjula, D. 2017. Recommendation System for Criminal Behavioral Analysis on Social Network using Genetic Weighted K-Means Clustering. *JCP,* 12**,** 212-220.

Sridevi, M., Rao, R. R. & Rao, M. V. 2016. A survey on recommender system. *International Journal of Computer Science and Information Security,* 14**,** 265.

Sun, S. 2007. An improved random subspace method and its application to EEG signal classification. International Workshop on Multiple Classifier Systems, 2007. Springer, 103-112.

Tsai, C.-H. 2016. A fuzzy-based personalized recommender system for local businesses. Proceedings of the 27th ACM Conference on Hypertext and Social Media, 2016. ACM, 297-302.

Wang, X., Hoi, S. C. H., Liu, C. & Ester, M. 2017. Interactive social recommendation. Information and Knowledge Management 2017. ACM, 357-366.

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

Zhang, S., Yao, L., Sun, A. & Tay, Y. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR),* 52**,** 5.