International Journal of Digital Information and Wireless Communications (IJDIWC) 6(4): 219-229
The Society of Digital Information and Wireless Communications, 2016 ISSN: 2225-658X (Online); ISSN 2412-6551 (Print)

Engineering Mining a Large Scale Data Based on Feature Engineering, Metadata, and Ontologies

Ahmed Adeeb Jalal [1] and Oğuz Altun [2]
[1 2] Computer Engineering Department, Yildiz Technical University, Istanbul, Turkey
[1] Al-Iraqia University, Baghdad, Iraq
Ahmedadeeb80@gmail.com, oguz@ce.yildiz.edu.tr

ABSTRACT

The growth of web especially in a social network in a continuously increasing. Multiplicity of offered items such as products or web pages, has made pick up relevant items for a user which searching for it a tedious. On the other hand, different tastes and behaviors of users is making likelihood to finding a neighbor user hard to get. Therefore, difficult for automated software systems to discover what is interesting to users. We have proposed a new approach to adapt to this widespread in e-commerce nowadays to reduce multiplicity impact of items and different views of users that can quickly produce the recommendations. We will exploit the domain knowledge of training data set to creating testing data set depending on an attribute of one feature that represents distinctive item genre. The testing data set will be the inputs to a hybrid recommender systems which is aspiring to achieve best recommendations through performing meta-level hybridization techniques that combine of content-based recommender systems and collaborative recommender systems. The proposed approach will reduce from effects of sparsity, cold start, and scalability very common problems with the collaborative recommender systems. Additionally to, improve the recommendations accuracy comparing with the pure collaborative filtering Pearson Correlation approach.

KEYWORDS

Recommender Systems, Feature Engineering, Hybrid Recommender Systems, Meta-level, Collaborative Filtering, Content-Based Filtering, Sparsity, Cold Start, Scalability, Metadata, Ontologies.

1 INTRODUCTION

The recommender systems are most popular intelligent software systems of the information filtering systems that applied in a various domains for example movies, music, books, jokes, restaurant, financial services [8], and Twitter followers [9]. It is recommending an interesting item to users who are didn't have any idea about it or which they are likely to prefer to users [4], [6], [7], [10], [11], [13]. These personalized suggestions are a useful alternative to searching algorithms to providing a way to help people picking the right items they might not have found by themselves. It became much easier to finding the necessary items easily from quantity of information available online.

Two main categories of the recommender systems: content-based recommender systems and collaborative recommender systems. Most recommendations systems use a hybrid recommender systems, which is a combination of these two approaches.

1.1 Collaborative Recommender Systems

The collaborative filtering approach is the most popular method of recommender systems [1], [10]. It generates the recommendations based only on the past users database ratings that represents full information about users' past rates. The collaborative filtering predicts preferable items to users by calculation the similarity score of user comparing with the other users. The collaborative filtering approach avoids semantics and systematically analyzes for items. Therefore, it characterized by quickly and accurately of recommendations for items without considering to the concept of item itself and what signifies.

The collaborative filtering is based on the assumption that consensuses people in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. The advantage of the collaborative filtering among other recommender systems, its recommended different items from what the user already knows. Also, the item unknown to the user yet this represents a surprise and the attraction of the user. Nevertheless, the collaborative filtering often suffers from three problems reduce its impact can be a challenge.

- Scalability: In many of the environments in order to find neighbors in collaborative filtering it requires a lot of time to doing the certain computations until finding similar users or items, because the data sets contain a million users and items. Furthermore, the number of users and items excessively increasing it becomes computationally difficult to find similar neighbors.
- Sparsity: Mostly, users do not rate the items, even the most popular items that they liked or purchased. Regarding the e-commerce companies, it strives to increase the amount of items which leads to increase sales and attract more consumers. Inasmuch to extremely increasing number of the users and items and very few ratings, most entries of the data sets matrix still remain zero. As a result, aggravation sparse problem.
- Cold start: Can be viewed as a special case of this sparsity problem [12], it happens because the user does not have a sufficient rating or any rating at all. Some companies are forcing consumers when login to the company's accounts to evaluate some of the most popular items in order to avoid this problem. Otherwise, it is difficult for recommender systems to provide an accurate recommendation to users.

1.2 Content-Based Recommender Systems

The content-based filtering approaches are based on a description of an attribute of the item features and the profile of the user's preference [15]. The recommended items at content-based filtering is matching predictions for the same kind of items that user already liked compared with various candidate items. So, it's considered a searched and compared process nearly, such as the processes used in the information retrieval systems but, without requiring user queries.

The content-based filtering retrieves information from two knowledge sources the features items and its rating that given by the user, simple approaches use the average values of the rated item. There are also more advanced techniques to infer to what is desirable by the user, such as decision trees, Bayesian classifiers and cluster analysis algorithms. For example, if the user has given a preferred rating toward action movies, so it will recommend more action movies to him. In many cases, getting common attributes is not easy and complimentary items are preferred, rather than similar items that enable a simple substitution [3], [13]. In addition, a content-based filtering depends on well-structured attributes and reasonable distribution of attributes across items [14].

1.3 Hybrid Recommender Systems

The hybrid recommender systems defined as a combinations of various knowledge sources as the inputs (such as user profile, community data, and item features) and multiple different recommender systems together to get the outputs.

The hybrid recommender systems could be luckier in some cases in different application domains to get right recommendations to user in a timely manner. As a result, there is one output for whatever the number of recommender systems contributed to the formation the hybrid recommender systems. The collaborative filtering uses a certain type of information, user profile (user's ratings) together with community data to derive recommendations, whereas the content-based filtering rely on textual descriptions of item features and user's ratings. Thus, the type of recommender systems chosen determines which kind of knowledge sources required. However, none of the basic approaches are able to use all of these knowledge sources. It divides into three different major categories of hybridization designs contain seven hybridization techniques. Each of these seven techniques operate under the context are different from each other, although it's participated in one hybridization design, that can be contributed to resolving some of problem as we mentioned.

- Monolithic hybridization design: Exploiting different knowledge sources of inputs for several recommender systems that implemented and combined in one algorithm to produce the final set of recommendations. Feature combination and feature augmentation techniques can be included into this category.
- Parallel hybridization design: Each recommender systems participating in this design operates independently of one another and each having its own outcomes (i.e. separate recommendation lists). The outcomes of several existing implementations are combining to generate the final set of recommendations. The mixed, weighted, and switching techniques classified among this design.

- Pipelined hybridization design: Sequentially outputs of previous recommender systems becomes inputs of subsequent one and final one produces recommendations for user. So, the outputs of the first recommender systems affects all chain of recommender systems that contributed to formation this algorithm. Optionally, subsequent recommender components may use parts of original input data, too [1]. The cascade and meta-level techniques are examples of such pipeline design.

1.4 Feature Engineering

The feature engineering exploits the domain knowledge of training data set to creating testing data set based on the features that managed machine learning algorithms to work function properly. The feature is a distinguishing characteristic that might help when analyze the problem in order to solve it [17]. The quality and quantity of the features will have great influence on whether the model is good or not [18].

The right features chosen require extensive testing to pick up a relevant feature that achieves better results, it's very important parts. The right features make a model simpler and more flexible, and they often yield better results [17]. However, the success of an algorithm is not entirely depending on the selected features, the model and the data set represented an important role in the success of the algorithm to achieving satisfactory results. The feature is a piece of information in the data set that might be containing many attributes, useful for prediction and will influence the recommendation that required to achieve. Any attribute could be a feature, as long as it is useful to the model [24].

1.5 Metadata

Metadata is data that provide information about other data [16]. Three types of metadata exist: structural, descriptive, and administrative metadata [22]. Structural metadata indicates to the containers of data that contain the compound objects, for example, how web pages are ordered to form the site. Descriptive metadata uses the item description, it can include features such as title, author, date, location, etc. Administrative metadata provides information about the management, such as creation, access, and file type information. Metadata could provide information about one or more aspects of the available data, it is used to summarize basic information about the data which can make tracking and working with specific data easier [29].

1.6 Ontologies

An ontology in computer science is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of discourse which variables needed for some set of computations and establishes the relationships between them [27], [28].

The ontology can be applied in many fields of software engineering, systems engineering, semantic web, and artificial intelligence in order to contribute the solving problems through limit complexity and to organize information.

Our work aims to overcome the very common problems with the recommender systems through create new feature from extracted attribute of movie genres. These features represent testing data set that will be feedback to the content-based approach to get average of distinctive genres ratings of the rated item for each feature depending on item description and user's rates. The testing data set will be the inputs to Pearson Correlation filtering.

2 RELATED WORK

We review some example of the hybrid recommender systems that applying in a various domains. Netflix Inc. [26] for the movie rent recommendation. It released a challenge in 2006 and offered grand prize of one million US dollars to person or team who could succeed in modeling a given data sets to within a certain specification [1], [2], [5]. It combines collaborative filtering and content-based filtering through similar habits of users as well as by higher rates of shared movies characteristics.

Lawrence et al. [20] describes a personalized recommender system to shoppers in supermarkets rely on their previous behavior towards the purchases to suggest new products for them. This system developed at IBM research has been implemented as a part of SmartPad, a personal digital assistant based remote shopping system. This system built based on combining content based filtering with collaborative filtering to improve the recommendations.

MovieLens [31] the online movie recommendation that used its data set in our approach, propose to new user login some watched movies which be most popular generally in order to evaluate it. Then, these ratings are exploited to recommend other movies not seen by the user. It also uses collaborative filtering based on similar users according to these ratings. These two approaches are combined to create personalized recommendations.

3 METHODOLOGY

3.1 Overview

The overall procedure of our proposed approach is as follows:

- Tests all the item features to choose the appropriate feature for the purpose of obtaining the better results.
- Extraction all the attributes of the selected item feature.
- Extracting the attributes without repetition.
- Creating testing data set with new features based on these attributes.
- Exploit the content based recommender systems to fill this testing data sets with average of distinctive genres ratings.
- Get the recommendation through the score of similarity between users depending on entire testing data sets based on collaborative recommender systems.
- Evaluate the results of proposed approach using two evaluation Metrics: predictive accuracy metrics and classification accuracy metrics to verify the accuracy of recommendation, at the next section.

3.2 Data Description

In this part introduces the data sets, we will describe the data sets collection process and the feature representations for each data set, as well as some basic statistics of the data set. The two data sets used in this study were downloaded from the GroupLens Research website [30].

- MovieLens 1M data set: GroupLens Research has collected and made available rating data sets from the MovieLens website [31]. The data sets were collected over various periods of time. The rating values are ranging between 0.5 to 5 of around 6,040 users and 3,883 items.
- HetRec 2011 data set: The 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems HetRec 2011 [32], has released data sets from Delicious, Last.fm Web 2.0, MovieLens, IMDb, and Rotten Tomatoes. These data sets contain social networking, tagging, and resource consuming (Web page bookmarking and music artist listening) information from sets of around 2,113 users. The rating values are ranging between 0.5 to 5 of around 2,113 users and 10,197 items.

Table 1 summarizes the statistics of training data sets, where the ratings matrix density is defined as the fraction of number of ratings over the total multiplies number of user and items in the rating matrix. The average number of the users who gave the rate of the items and the average number of the items that rated by user can be seen from Table 1.

Table 1. Statistics of training data sets

Statistics	HetRec	MovieLens
Number of users	2113	6040
Number of items	10197	3883
Number of ratings	855598	1000209
Average number of ratings by users	404.921	165.598
Average number of ratings for items	83.91	257.587
Density	3.97%	4.265%

The number of ratings given by one user to all items, HetRec 2011 data set, ranging from 20 to 3410 with percentage from 0.2% to 33.5% respectively. MovieLens 1M data set, ranging from 20 to 2314 with percentage from 0.52% to 59.6% respectively.
The number of ratings given by all users to one item, HetRec 2011 data set, ranging from 1 to 1670 with percentage from 0.05% to 79.05% respectively. MovieLens 1M data set, ranging from 1 to 3428 with percentage from 0.02% to 56.75% respectively.

3.3 Feature Learning

Machine learning, feature learning or representation learning is a set of techniques that learn a feature [19], [23]. The training data set (i.e. raw data) defined as a set of aggregated features, exploits to produce a sort of representation that can make the machine learning algorithms simpler and more flexible.
The training data set in our paper consists of two major categories: users and items (movies), each one

contains many of the features which include many attributes. For example, user's category contains gender, occupation, age and Zip-code, item's category contains title, genres, actors and year of release. Thus, a transformation of raw data into the sort of representation requires more than one feature testing in order to determine useful features. Feature learning is motivated by the fact that machine learning algorithms often require appropriate inputs mathematically and computationally. However, the success of the algorithm depending on the selected features besides the model and the data set to achieving satisfactory results, as mentioned earlier. Usually, the initial choice of feature based on our experience and a prior knowledge about the existing data set details.

Let $S_t(i)$ denote training sample *i*, then $S_t(i)$ can be represented as:

$$S_t(i) = \{V_t^i(i), V_t^o(i), D_t^o(i)\} \tag{1}$$

Where $V_t^i(i)$, $V_t^o(i)$ and $D_t^o(i)$ stand for the input vector and the two output vector for training sample *i*, respectively. $V_t^i(i)$ is represented all features of item *i*, whose structure can be shown as:

$$V_t^i(i) = \begin{Bmatrix} \text{Title } (i) \\ \text{Year } (i) \\ \text{Genre } (i) \\ \text{Location } (i) \\ \text{Director } (i) \\ \text{Actors } (i) \\ \text{Country } (i) \end{Bmatrix} \tag{2}$$

All the entries either textual or integers, Genre (*i*) is the genre of item *i* that will be extracted from other features of the item, it is textual.

Likewise, $V_t^o(i)$ is represented all extracted genres of items, whose structure can be shown as:

$$V_t^o(i) = \begin{Bmatrix} \text{1. Adventure, Children, Fantasy} \\ \text{2. Comedy, Romance} \\ \text{3. Comedy} \\ \text{4. Action, Crime, Thriller} \\ \text{5. Adventure, Children, Action} \\ \text{6. Comedy} \\ \text{7. Adventure, Children, Action} \\ \text{8. Animation} \\ \text{9. Musical, Romance} \end{Bmatrix} \tag{3}$$

Likewise, $D_t^o(i)$ is represented all distinctive genres of items (i.e. without repetition), whose structure can be shown as:

$$D_t^o(i) = \begin{Bmatrix} \text{1. Adventure} \\ \text{2. Comedy} \\ \text{3. Romance} \\ \text{4. Action} \\ \text{5. Crime} \\ \text{6. Musical} \\ \text{7. Animation} \end{Bmatrix} \tag{4}$$

Let $W_g(i)$ denotes the average of distinctive genres ratings based on Eq. (4), then $W_g(i)$ can be represented as:

$$W_g(i) = \frac{\sum_{V_t^o(i) \in D_t^o(i)} R}{TF} \tag{5}$$

Where R is represented the value of ratings of item *i*, TF is represented the term frequency of the distinctive genre in the user's profile for rated items.

Algorithm 1 explained the sequence operational for creating testing data set and feedback this data set with the average of distinctive genres ratings that rated by user.

Algorithm 1. Creating testing data set

```
1: input: read the item features file.
2: choose an appropriate step size
   (number of items).
3: for t=1,…..,T do
4: extracting the genres of items
   using Eq. (3).
5: end for
6: extracting the distinctive genres
   using Eq. (4).
7: get the average of distinctive
   genres ratings using Eq. (5).
8: output: creating testing data set
   based on Eq. (4) and Eq. (5).
```

Table 2 and Table 3 shows the structure and the statistics of testing data sets (HetRec 2011 and MovieLens 1M data set) that have been configured after implemented Algorithm 1, respectively.

Table 2. Structure of testing data set

	D1	D2	D3	D4	D5	D6	D7	Dn
U1	4.13	4.3	4.1	2.5	1.8	4.7	1.5	0
U2	3.34	0	2.7	1	5	3.2	1	3.9
U3	1.5	4.2	3.7	0	2.3	0	3	4.6
Um	2.17	3.5	3.26	4.7	0	3.26	0	0

Table 3. Statistics of testing data sets

Statistics	HetRec	MovieLens
Number of users	2113	6040
Number of items	19	18
Number of ratings	25029	62484
Average number of ratings by users	11.85	10.35
Average number of ratings for items	1317.32	3471.34
Density	62.35%	57.5%

In Table 3 the number of items in testing data sets has been reduced. Therefore, increasing the ratings matrix density that contributes to solving the problems of the recommender systems. The percentage decrease the items by compared the two data sets before and after implemented the Algorithm 1 up to 99.8%.
The number of ratings given by one user to all items in testing data sets. HetRec 2011 data set, ranging from 3 to 19 with percentage from 15.8% to 100% respectively. MovieLens 1M data set, ranging from 2 to 18 with percentage from 11.1% to 100% respectively.
The number of ratings given by all users to one item in testing data sets. HetRec 2011 data set, ranging from 2 to 2110 with percentage from 0.1% to 99.9% respectively. MovieLens 1M data set, ranging from 630 to 6012 with percentage from 10.5% to 99.6% respectively.
According to the description of testing data set above, the Pearson Correlation similarity of two users i, j is defined as:

$$S(i,j) = \frac{\sum_{y \in D}(r_{y,i}-\bar{r}_i)\times(r_{y,j}-\bar{r}_j)}{\sqrt{\sum_{y \in D}(r_{y,i}-\bar{r}_i)^2}\times\sqrt{\sum_{y \in D}(r_{y,j}-\bar{r}_j)^2}} \quad (6)$$

The formula used to predict the rating depending on the score of similarity, the user's rate of training data sets, and the distinctive genre rating which item belongs to it, can be represented as:

$$r_{i,n} = D_i + \frac{\sum_{j \in D} S(i,j)\times(r_{j,n}-D_j)}{\sum_{j \in D} S(i,j)} \quad (7)$$

3.4 Meta-Level Technique

Meta-level technique is one of seven hybridization recommendation techniques subordinates to the pipelined hybridization design, exploits to get a sort of model which will be the input of the next technique. As a result, the contributing recommender completely replaces the raw data with a learned model that the actual recommender uses in its computation.

Figure 1 shows the general schematic of proposed approach which applied in our paper.

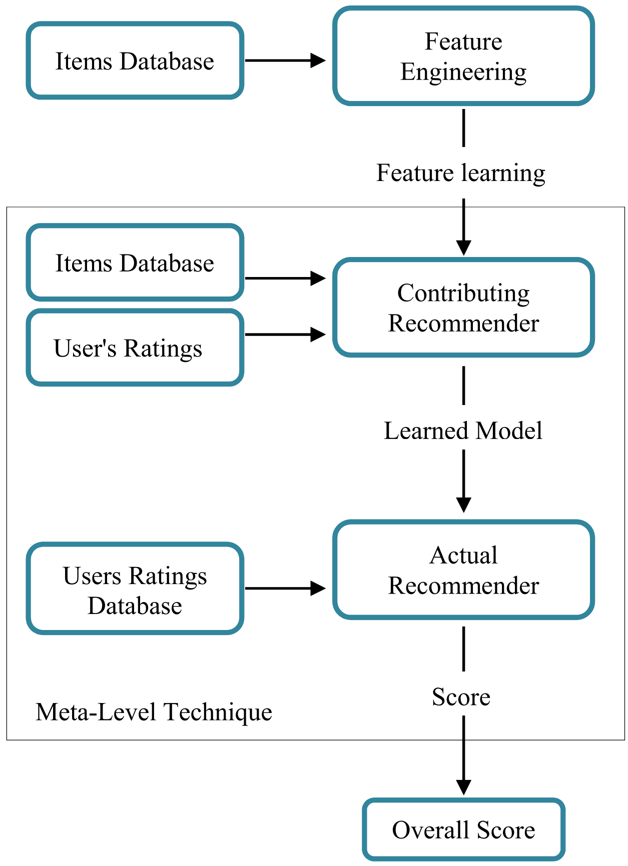


Figure 1. General schematic of proposed approach

In this section, the results obtained through creating testing data set can be summarized as follows:

- Decreasing the number of items.
- Increasing the ratings matrix density.
- Increasing the ratings of users.
- Increasing the ratings of items.

Now, we have two important questions will be provable in the next section:

- How useful are the reducing items?
- Can the proposed approach improve the accuracy of recommendation?

4 EXPERIMENTS

The recommender systems have been evaluated in different evaluation metrics. Evaluating recommender systems is difficult because the evaluation results mutable, it's based on algorithms, data sets and evaluation metrics together.
Many algorithms have been designed some of it applied effectively on some of the data sets, while not worked with others. Also, a variety of data sets are

available downloaded online, but some of it is not valid for performed with some algorithms. As a result, there should be a consensus between the algorithm and data sets selected, a potentially overwhelming set of choices. Finally, evaluation metrics can be divided into two major categories will be discussed later, the first category based on the numeric value (i.e. error ratio) that represents the difference value of the original rate and the predicted rate called predictive accuracy metrics, and the second category based on the related as if that the predicted rate is relevant or irrelevant compared with the original rate called classification accuracy metrics, this is the motivation of the both types of the evaluation metrics applied in this paper because every category follows a certain pattern for evaluation. It would be better to choose one or more evaluation metrics in order to compare the accuracy of different recommender systems [25].

4.1 Data Sets and Preprocessing

We used testing data set as the input data in our proposed approach, which got it after implemented the algorithm 1 on the two publicly available data sets HetRec 2011 and MovieLens 1M as we mentioned in Section 3. The purpose of this process to improve performance and get accurate recommendations.
As is well known, today the increasing growth in the web with thousands of users who interact with thousands of items if not millions. This growth requires reducing time-consuming and the number of similar users in order to make the software systems faster for getting the recommendations quickly. So, it was our approach focuses on reducing the items in testing data set to the extent that it can get satisfactory results. In addition, the testing data set increases the density of users' rates, it make to get the right similar user more flexible.
Figure 2 illustrated the percentage of increasing the ratings of users for training and testing data sets.
We proposed a method of the hybrid recommender systems according to testing data set that combine two approaches content-based filtering and collaborative filtering Pearson Correlation approach.
Let HRS denotes to combining two approaches content-based filtering and collaborative filtering Pearson Correlation approach, and CFP denotes to the pure collaborative filtering Pearson Correlation approach.
Then, we will compare our results that got it from HRS method based on testing data set with CFP method based on training data set, for the two data sets selected HetRec 2011 and MovieLens 1M.
Table 4 shows the advantage of reducing the items, through reducing the Time-consuming in order to predict the rate and reducing the average number of similar users for each predict operation with keeping an efficient result.

Table 4. Time-consuming and similar users

Data Sets	HetRec 2011		MovieLens 1M	
Methods	CFP	HRS	CFP	HRS
Time-consuming for one sample testing (s)	0.568	**0.061**	0.83	**0.133**
Average number of similar users for each sample	394	**308**	680	**528**

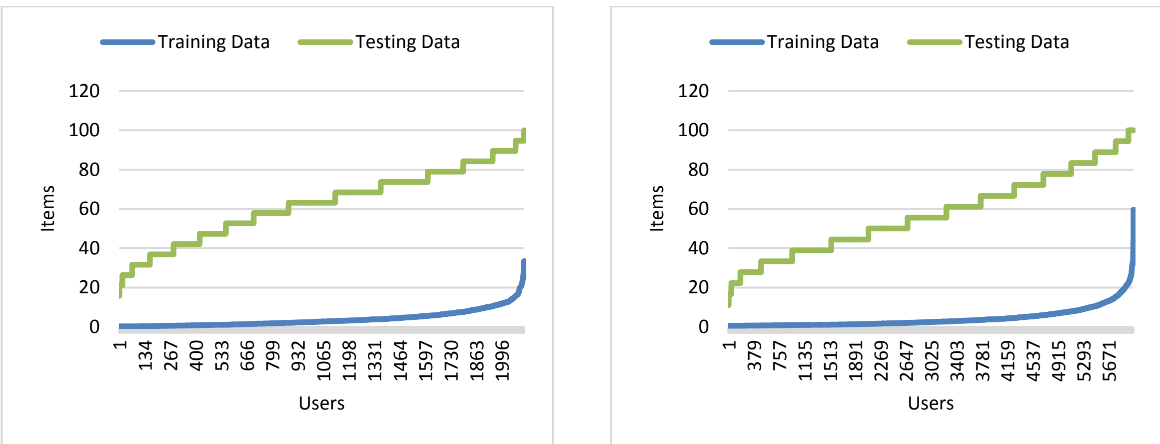


Figure 2. Percentage of the users' rates for training and testing data sets

4.2 Evaluation Metrics

There are many of the published evaluation metrics differ from each at its work and its results (such as predictive accuracy metrics, classification accuracy metrics, rank accuracy metrics and an empirical comparison of evaluation metrics etc.).

We will focus only on the most common evaluation metrics to evaluate the accuracy of recommender systems. Herlocker et al. [25] provide a comprehensive discussion of accuracy metrics together with alternate evaluation criteria, which is highly recommended for reading.

- Predictive Accuracy Metrics: Predictive accuracy metrics based on the numerical difference values between predicted ratings and true ratings that are given by the user to the movies which is an estimate of a five-star according to the data sets used HetRec 2011 and MovieLens 1M. The success of recommender systems evaluation relies on how close the predicted ratings and the true ratings (i.e. if the numerical difference values is small the recommender systems deemed successful vice versa).

  When evaluating the ability of a recommender systems to correctly predict for a specific item, mean absolute error (MAE) and Root Mean Squared Error (RMSE) one of the most important evaluation metrics of this class compared with other evaluation metrics.

$$MAE = \frac{\sum_{i=1}^{T} |p_i - r_i|}{T} \tag{8}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{T} |p_i - r_i|^2}{T}} \tag{9}$$

Where p_i and r_i represent the predicted ratings and the real ratings of users, respectively, and T denotes to the total number of predictions generated for all active users in the data set.

The performance evaluations of predictive accuracy metrics for HRS method compared to CFP method according to the two evaluation metrics: MAE and RMSE, are summarized in Table 5 and Figure 3.

- Classification Accuracy Metrics: Classification accuracy metrics based on the relevance between the predicted ratings and the true ratings in order to determine which items are relevant (i.e. good) and which are irrelevant (i.e. bad). It means the existence of different groups and the decision will be to any groups belongs the predicted ratings. For instance, the rating scale of the two data sets range (0.5,...,5), the separation threshold could be arbitrary to 4 according to fine estimate as in [33]. In our paper, we proposed 3 stars as a threshold to give more flexibility in the case of unavailability the items more than 4 stars also, the global average of the ratings in the HetRec 2011 and MovieLens 1M data set is less than 4 roughly 3.5. We can classify each recommendation such as [21]:
  1. True positive (TP, an acceptable item is recommended to the user).
  2. True negative (TN, an unacceptable item is not recommended to the user).
  3. False positive (FP, an unacceptable item is recommended to the user).
  4. False negative (FN, an acceptable item is not recommended to the user).

  Precision Eq. (8) and recall Eq. (9) are the most popular evaluation metrics in the information retrieval field depend on the separation of relevant "positive" and irrelevant "negative" items, it has been used in [34], [35]. F-measure Eq. (10) allows combines precision and recall into a single score.

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

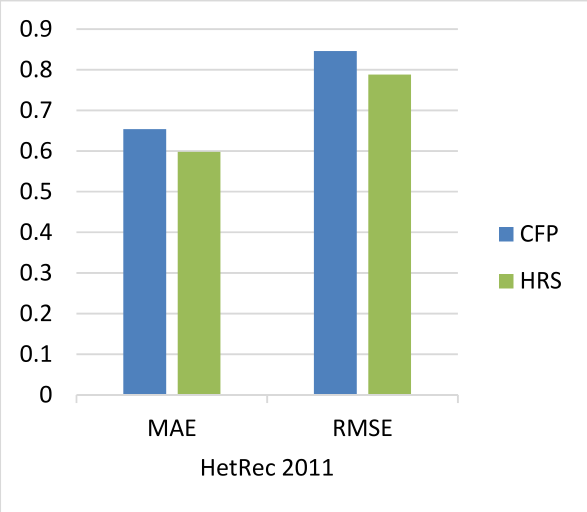
$$Recall = \frac{TP}{TP+FN} \tag{11}$$

$$F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

The performance evaluations of classification accuracy metrics for HRS method compared to CFP method according to the evaluation metrics: precision, recall and F-Measure, are summarized in Table 6 and Figure 4.

Table 5. MAE and RMSE evaluations

Data Sets	HetRec 2011		MovieLens 1M	
Methods	CFP	HRS	CFP	HRS
MAE	0.654	**0.598**	0.718	**0.679**
RMSE	0.846	**0.788**	0.9	**0.869**



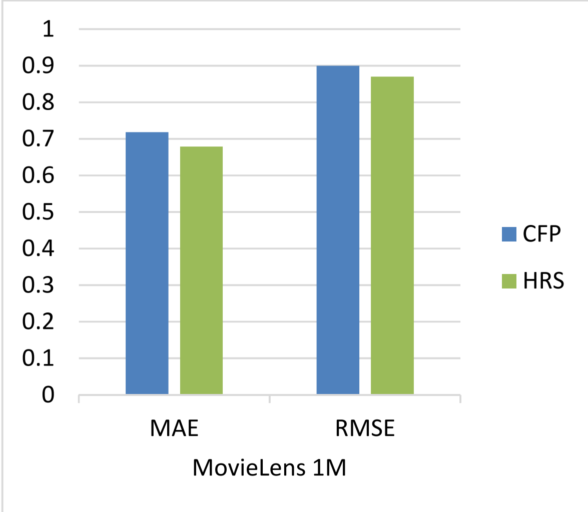
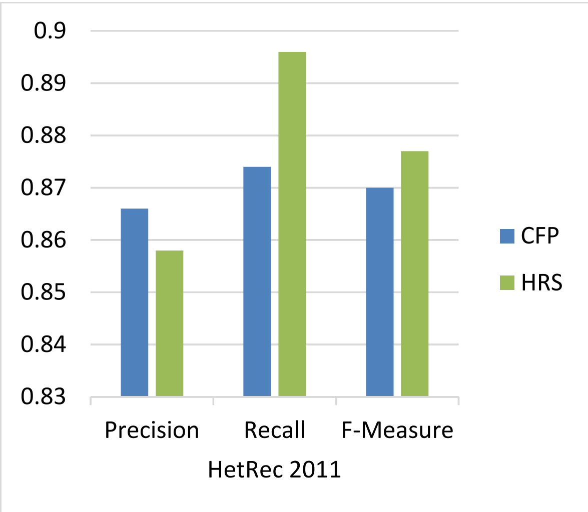


Figure 3. Comparison of evaluations of predictive accuracy metrics

Table 6. Precision, Recall and F-measure evaluations

Data Sets	HetRec 2011		MovieLens 1M	
Methods	CFP	HRS	CFP	HRS
Precision	0.866	0.858	0.891	0.878
Recall	0.874	**0.896**	0.904	**0.92**
F-Measure	0.87	**0.877**	0.897	**0.898**



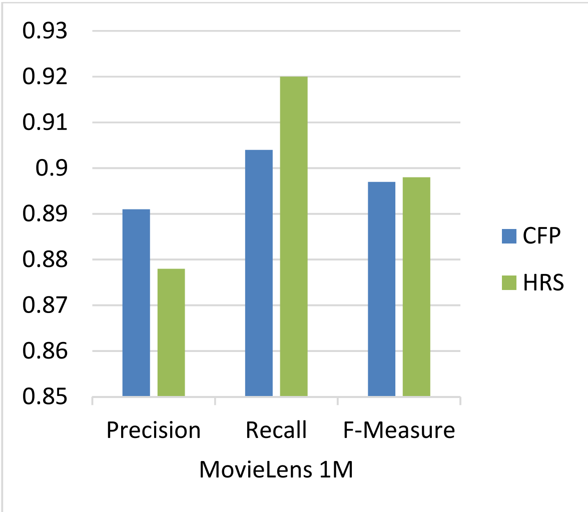


Figure 4. Comparison of evaluations of classification accuracy metrics

Inasmuch to the results obtained in this Section, we proved that:

- The reducing items was useful.
- The proposed approach improved the accuracy of recommendation.

In Table 7, all results obtained in this Section of HRS method based on testing data set compared with CFP method based on training data set, for the selected data sets: HetRec 2011 and MovieLens 1M are listed. The performance superiority of HRS method compared with CFP method represented by Yes or No. The HRS method excelled at the most results obtained, can be seen in Table 7.

Table 7. All results obtained

	Data Sets	HetRec 2011	MovieLens 1M
Performance	Time-consuming for one sample testing (s)	**Yes**	**Yes**
	Average number of similar users for each sample	**Yes**	**Yes**
predictive accuracy metrics	MAE	**Yes**	**Yes**
	RMSE	**Yes**	**Yes**
Classification accuracy metrics	Precision	No	No
	Recall	**Yes**	**Yes**
	F-Measure	**Yes**	**Yes**

5 CONCLUSIONS

In this paper, we proposed creating the testing data set that incorporates limited items in order to alleviating the impact of scalability, sparsity and cold start problem by increasing the ratings matrix density. As an additional benefit, we used the testing data set as the inputs for the hybrid recommender systems and evaluated the results according to two evaluation metrics to prove the accuracy of the recommendation. According to description above, we proved useful and effectiveness the proposed approach to all most aspects compared with the pure collaborative filtering Pearson Correlation approach based on the training data set for the selected data sets.

ACKNOWLEDGMENTS

I'm grateful to Al-Iraqia University and Computer Engineering Department, Yildiz Technical University for this opportunity to get the master's degree. Special thanks to Mona Mohamed Wafy employee at Al-Iraqia University. Finally, thanks everyone that helped and supported me.

REFERENCES

1. Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich, “Recommender Systems_ an Introduction,” US, 2011.
2. Andrey Feuerverger, Yu He, and Shashi Khatri, “Statistical Significance of the Netflix Challenge,” Institute of Mathematical Statistics, 2012.
3. Mohammad Amir Sharif and Vijay V. Raghavan, “A Large-Scale, Hybrid Approach for Recommending Pages Based on Previous User Click Pattern and Content,” pp. 8, USA, 2016.
4. Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Raghu Ramakrishnan, “Content recommendation on web portals,” In Communications of the ACM, 56(6): 92–101, 2013.
5. Robert M. Bell and Yehuda Koren, “Lessons from the Netflix prize challenge,” In ACM SIGKDD Explorations Newsletter, 2007.
6. Abhinandan Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram, “Google news personalization: scalable online collaborative filtering,” In WWW, pp. 271–280, May 2007.
7. Greg Linden, Brent Smith, and Jeremy York, “Amazon. Com recommendations: Item-to-item collaborative filtering,” In Internet Computing, IEEE, 7(1): 76–80, 2003.
8. Alexander Felfernig, Klaus Isak, Kalman Szabo, and Peter Zachar, “The VITA Financial Services Sales Support Environment,” pp. 1692-1699, Vancouver, Canada, 2007.
9. Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Bosagh Zadeh, “WTF:The who-to-follow system at Twitter,” Proceedings of the 22nd international conference on World Wide Web, ACM New York, NY, USA, 2013.
10. Francesco Ricci, Lior Rokach, and Bracha Shapira, “Introduction to Recommender Systems Handbook,” Recommender Systems Handbook, Springer, 2011.
11. “Facebook, Pandora Lead Rise of Recommendation Engines - TIME,” TIME.com. 27 May 2010. Retrieved 1 June 2015.
12. Zan Huang, Hsinchun Chen, and Daniel Zeng, “Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering,” ACM Transactions on Information Systems 22, no. 1, pp. 116–142, 2004.
13. Gediminas Adomavicius and Alexander Tuzhilin, “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions,” IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 6, pp. 734-749,2005.

14. Delgado Joaquin, Ishii Naohiro, and Ura Tomoki, "Content-based Collaborative Information Filtering: Actively Learning to Classify and Recommend Documents," in Proceedings of the Second International Workshop on Cooperative Information Agents II, Learning, Mobility and Electronic Commerce for Information Discovery on the Internet, pp. 206-215, 1998.
15. Peter Brusilovsky, "The Adaptive Web," pp. 325, 2007.
16. http://www.merriam-webster.com/dictionary/metadata.
17. "Discover Feature Engineering, How to Engineer Features and How to Get Good at It - Machine Learning Mastery," Machine Learning Mastery. Retrieved 11-11-2015.
18. "Feature Engineering: How to transform variables and create new ones?," Analytics Vidhya. 12-03-2015. Retrieved 12-11-2015.
19. Zdenek Zabokrtsky, "Feature engineering in Machine Learning," Retrieved 12 November 2015.
20. Richard D. Lawrence, George S. Almási, Vladimir Kotlyar, Marisa S. Viveros, and Sastry S. Duri, "Personalization of supermarket product recommendations," 2001.
21. Paolo Cremonesi, Roberto Turrin, Eugenio Lentini, and Matteo Matteucci, "An evaluation methodology for recommender systems," Proc. of the 4th International Conference on Automated Solutions for Cross Media Content and Multichannel Distribution (AXMEDIS), IEEE, 2008.
22. National Information Standards Organization (NISO), "Understanding Metadata," 2004.
23. Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation Learning: A Review and New Perspectives," IEEE Trans. PAMI, special issue Learning Deep Architectures 35: 1798–1828, 2013.
24. Navid Razmjooy, Bibi Somayeh Mousavi, and Fazlollah Soleymani, "A hybrid neural network Imperialist Competitive Algorithm for skin color segmentation," Mathematical and Computer Modelling, Volume 57, Issues 3–4, February 2013.
25. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Transactions on Information Systems (TOIS) 22, 2004.
26. http://www.netflix.com.
27. Fredrik Arvidsson and Annika Flycht-Eriksson, "A Ontologies I," Retrieved 26 November 2008.
28. Thomas R. Gruber, "A Translation Approach to Portable Ontology Specifications," June 1993.
29. www.theguardian.com, "A Guardian Guide to your Metadata," Guardian News and Media Limited, 12 June 2013.
30. http://www.grouplens.org.
31. http://www.movielens.org.
32. http://ir.ii.uam.es/hetrec2011/.
33. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John T. Riedl, "An algorithmic framework for performing collaborative filtering," In SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 230–237, New York, NY, USA, 1999.
34. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl, "Application of Dimensionality Reduction in Recommender System -- A Case Study," in ACM WebKDD 2000 Web Mining for E-Commerce Workshop, 2000.
35. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl, "Analysis of recommendation algorithms for E-commerce," In Proceedings of the 2nd ACM Conference on Electronic Commerce (EC'00). ACM, New York, pp. 285–295, 2000.