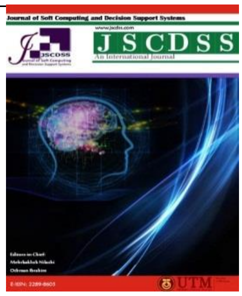
An International Journal
JSCDSS
http://www.jscdss.com
Vol.3 No.6 December 2016: 16-44
Article history:
Accepted 6 October 2016
Published online 6 October 2016

Journal of Soft Computing and Decision Support Systems



E-ISSN: 2289-8603

An Overview of Data Mining Techniques in Recommender Systems

Mehrbakhsh Nilashi [a,b,*]
[a] Faculty of Computing, Universiti Technologi Malaysia, Johor, Malaysia
[b] Department of Computer Engineering, Lahijan Branch, Islamic Azad University, Lahijan, Iran

*** Corresponding author email address**: nilashidotnet@hotmail.com

Abstract

Nowadays, recommender systems support the online customers in their decision making and buying process. Whereas, the information in the web is increasing through continuous growing of the number of websites, recommender systems have to recommend the items with maximal matching to the users' preference. Recommender systems are an active research topic in the data mining and machine learning fields. Data mining techniques have played an important role in the design and implementation of recommender systems. In this paper, an overview of the main data mining techniques used in the design and implementation of recommender systems is given. The relevant papers which have used the data mining techniques in the context of recommender systems are reviewed. We hope that this research helps researchers who are interested in developing recommender systems with an insight into its state-of-the-art methods.

Keywords: Recommender systems, Data mining techniques, Classification methods, Clustering methods, Prediction methods

1. Introduction

During the last decade the amount of information available online increased exponentially and information overload problem has become one of the major challenges faced by information retrieval and information filtering systems. The solutions to overload information problems can be found in the field of information retrieval and information filtering, where search engines like Google and various text-retrieval applications have been developed to deal with the problem. Recommender systems are one of the solutions to the information overload problem (Nilashi et al., 2014a, Nilashi et al., 2014b). They address the problem of filtering information that is likely of interest to individual users. In addition, providing recommendations to users by reflecting their personal taste and convincing the users to trust and explore the given recommendations are the main objectives of a recommender. Successful application have been used on the Internet by electronic commerce (e-commerce) Websites like Amazon.com, that offer millions of products to its customers, and by communities in the entertainment domain like MovieLens, a research project that runs a Website where people can become members and receive recommendations for movies.

Recommender systems have become an important and interesting research area since the coming out of the first research paper on Collaborative Filtering (CF) in the mid-1990s (Resnick et al., 1994; Shardanand and Maes, 1995). Several studies show that using a recommender system can lead to increased sales volumes in the short and long term or help to increase sales diversity by directing customers to other parts of the available product catalog (Senecal and Nantel, 2004; Zanker et al., 2006; Fleder and Hosanagar, 2007; Dias et al., 2008; Vahid et al., 2016). Although recently many different approaches and techniques to recommender systems have been developed, the interest in this area still remains high. This is because of growing demand for their practical applications, which are capable of dealing with information overload and to generate personalized recommendation to the users.

1.1. Classification of Recommender Systems

Recommendation systems predict items for users tailored to their preferences based on user-item interaction using either implicit or explicit information (Adomavicius and Tuzhilin, 2005). For an unseen item, reducing the time required to predict what rating a user would give is one of the main goals of the recommendation task. In addition, this can be to find a list of items that the user is most likely to enjoy. Explicit information is specific information provided by the user such as ratings or ranking. One of the most successful algorithms with this type of information in recommending items is CF (Nilashi et al., 2013; Bagherifard et al., 2013; Farokhi et al., 2016) which has been implemented in online platforms by corporations such as TiVo, Amazon and Netflix (Linden et al., 2003). The premise of CF is that users who agreed in the past tend to agree in the future. CF-based recommender systems rely solely on product ratings provided by a large user community to generate personalized recommendation lists for each individual online user. The ratings provided by users for items are the key input to CF recommender systems. They present information regarding the quality of the item along with the preference of the user who shared the rating. The key to successful collaborative recommendation lies in the ability to make meaningful associations between people and their product preferences, in order to assist the end-user in future transactions. Similarities between past experiences and preferences are exploited to form neighbourhoods of like-minded people from which to draw recommendations or predictions for a given individual user. For example users who have liked or disliked the same items could be grouped together in a neighbourhood, or similarly items that have been liked or disliked by the same people could also be grouped together.

CF algorithms can be divided into two categories, memory-based algorithms and model based algorithms (Adomavicius and Tuzhilin, 2005). Memory-based (heuristic-based) algorithms exploit the entire item-user database. A set of similar users are identified for the current user or active user, and rating predictions are generated based on ratings in the neighborhood of the current user. Memory-based CF is easy to implement and it is easy to add new data incrementally. In the memory-based category, the most popular non-probabilistic approach is the *k*-NN algorithm. Because heuristic-based approaches can make predictions based on the local neighbourhood of the active user, or can base their predictions on the similarities between items, these systems can also be classed into user-based and item-based approaches. User-based CF has been the most popular and commonly used (memory-based) CF strategy (Konstan et al., 1997). It is based on the premise that similar users will like similar items. A user profile is collected and maintained for each user which records the items that he has consumed over time, and usually a corresponding set of ratings that judge how much he liked or disliked each item. In this manner, a model of the user's preferences for different types of items is constructed. Sometimes other types of information pertaining to the user such as demographical information may also be collected in the user profile. One of the core challenges for user-based CF is the accurate identification of similarities between users based on their shared preferences. Sarwar et al. (2001) first proposed item-based CF as an alternative style of CF that avoids the scalability bottleneck associated with the traditional user-based algorithm. The bottleneck arises from the search for neighbours in a population of users that is continuously growing.

In contrast to the heuristics that are based mostly on information retrieval methods, model-based recommendation techniques provide item recommendation by first constructing a model of user ratings for offline phase. For model based approaches, algorithms take probabilistic methods and envision the recommendation process as computing the expected value of a user prediction, given his or other users' ratings on the rest of the items. The model building process is performed by different machine learning algorithms (Adomavicius and Tuzhilin, 2005; Breese et al., 1998). In addition, model-based methods adopt an eager learning strategy for predicting or recommending content, where a model of the data, i.e. the users, items and their ratings for those items, is pre-computed (Adomavicius and Tuzhilin, 2005).

Content-based approaches, also known as Content-Based Filtering (CBF), to recommendation build on the conjecture that a person likes items with features similar to those of other items he liked in the past. According to Pazzani and Billsus (2007), generally, CB recommender systems: (1) construct a user profile from rating information of each user on items; (2) identify like-minded users who rate items similar to a target user using a similarity function such as cosine similarity, Pearson correlation coefficient, or distance-based similarity; and (3) recommend Top-N items that like-minded users preferred after their ratings are predicted as an average weighted sum or adjusted weighted sum of ratings given on items identified by like-minded users.

Data mining techniques, mathematical modeling techniques and software tools are used to find patterns in data. They use the patterns to build models. In the context of recommender systems applications, these techniques are used to build recommendation models from large data sets. Recommender systems that incorporate data mining techniques make their recommendations using knowledge learned from the actions and attributes of users. In this paper, we give an overview of the main data mining techniques used in the context of recommender systems. The relevant papers which have used the data mining techniques are reviewed in the context of recommender systems

2. Data Mining Methods for Recommender Systems

Recommender systems generally implement techniques and strategies from other neighboring areas such as Information Recovery (IR). However, most of methods keep in their nucleus an algorithm that can be recognized as a particular case of a data mining technique that is the process of analyzing data from different perspectives and summarizing data into useful information. The procedure of data mining generally includes 3 steps, performed in sequence: Data Preprocessing, Data Analysis, and Result Interpretation. Fig. 1 demonstrates main steps and methods in a data mining problem successively (Ricci et al., 2011).

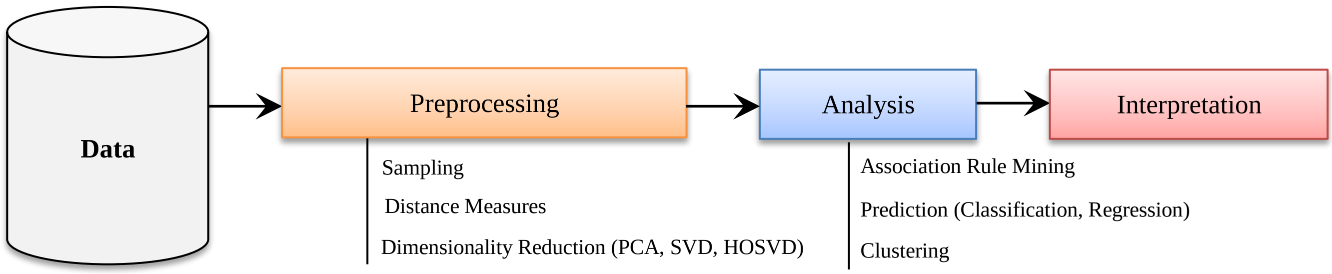


Fig. 1. Main steps and methods in a data mining problem

2.1. Data Preprocessing

The real-world data before can be preprocessed by machine learning techniques needs to be preprocessed in the analysis stage. Unprocessed data is not useful in the analysis stage as it may be incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data; Noisy: containing errors or outliers; and, Inconsistent: containing discrepancies in codes or names.

Therefore, a task in data preprocessing step basically includes (Pyle, 1999): Data cleaning; Data integration; Data transformation: normalization and aggregation; Data reduction; and, (5) Data discretization. For designing a recommender system, mainly three principal ways for preprocessing data are considered that are distance measures and similarity, sampling and dimensionality reduction.

Distance and similarity metrics. Distance and similarity metrics are used to solve many problems in retrieval and pattern recognition such as classification and clustering. Distance measures are extensively used in similarity estimation of two features. In recommender systems the classification method such as *k*-NN algorithms, as a well-known classifier, is highly reliant on defining a proper distance measure for similarity estimating.

The similarity function is analogous to the distance function because the larger values indicate the higher similarity (Cohen et al., 2003). Accordingly, the similarity function equals to:

$$Similarity = \frac{1}{Distance} \qquad (1)$$

Depends on the nature of data, many distance and similarities measures for data analysis have been formed which are based on two features identified by non-zero vectors $x = (x_l.\,.\,.\,.\,.\,x_n)$ and $y = (y_l.\,.\,.\,.\,.\,y_n)$ from R_n. Deza and Deza (2009) comprehensively introduced the distance and similarity metrics for different types of data such as binary and numeric. The methods for measuring similarity are continually growing and already numbered more than sixty types, such as inner product, Dice coefficient, cosine coefficient, Jaccard coefficient, and overlap coefficient. Some binary feature vector similarity measures have been enhanced and optimized by researchers (Cha and Tappert, 2006; Cha and Tappert, 2003) and some researchers (Baroni-Urbani, and Buser, 1976; Choi, 2008) have conducted studies for examining binary distance and similarity metrics and also comparative studies provided the wide variety of binary similarity measures (Hubalek, 1982; Jackson et al., 1989; Tubbs, 1989; Willett, 2003; Zhang and Srihari, 2003).

In the following, we introduce some of the most important similarity measures which are used in the recommender systems context, especially in the CF.

Cosine Similarity. Usually cosine similarity metric is used to estimate the similarity between two objects (e.g. object a and b) in information retrieval. The objects are in the shape of two vectors x_a and x_b and calculating the Cosine Vector (CV) (or Vector Space) similarity between these vectors indicate the distance of them to each other (Billsus and Pazzani, 1998; Billsus and Pazzani, 2000; Lang, 1995). The cosine similarity between two objects *T1* and *T2* is then calculated as (see Fig. 2):

$$\cos(T_1, T_2) = \frac{T_2.T_2}{\| T_1 \|^2 * \| T_2 \|^2} \qquad (2)$$

In the context of item recommendation, for computing user similarities, cosine similarity measure can be employed in which a user *u* indicates vector $x_u \in R^{|I|}$ where $x_{ui} = r_{ui}$ if user *u* has rated item *i* and for unrated item considers 0. The similarity between two users *u* and *v* would then be calculated as:

$$CV(u,v) = \cos(X_a, X_b) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r^2_{ui}} \sqrt{\sum_{j \in I_v} r^2_{vi}}} \qquad (3)$$

where I_{uv} once more indicates the items rated by both *u* and *v*. A shortcoming of this measure is that it does not examine the differences in the mean and variance of the ratings made by users *u* and *v*.

An International Journal JSCDS
E-ISSN: 2289-8603

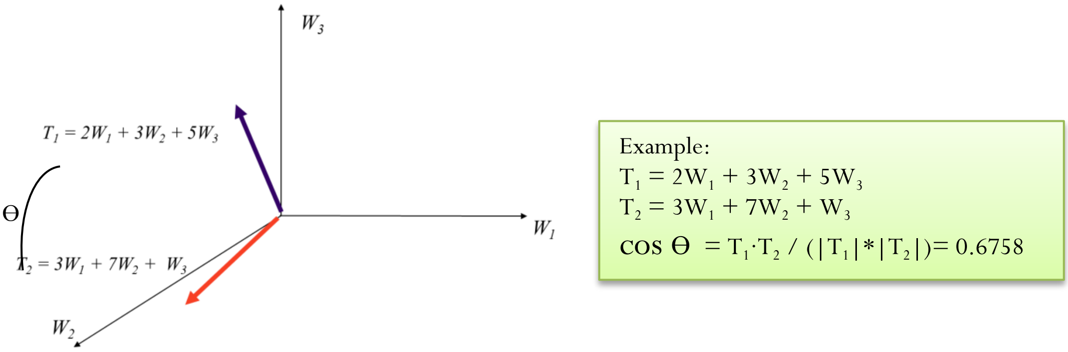


Fig. 2. Cosine similarity.

Cosine similarity is calculated on a scale between -1 and +1, where -1 implies the objects are completely dissimilar, +1 implies they are completely similar and 0 implies that the objects do not have any relationship to each other.

In prior researches, vector similarity has been proven to work well in information retrieval (Salton and Buckley, 1998) but it has not been found to carry out as well as Pearson's similarity approach for user-based CF (Breese et al., 1998).

Table 1 shows an example of rating matrix. The similarity between two users u and v would then be calculated as:

Table 2
An example of rating matrix.

	A	B	C	D	E
User 1	4	3	?	5	?
User 2	4	5	5	?	?
User 3	5	?	4	3	4
User 4	5	5	?	3	2

$$Sim(User\,3, User\,4) =$$

$$\frac{R_{a,1}R_{b,1}}{\sqrt{R_{a,1}^2+R_{a,4}^2+R_{a,5}^2}\sqrt{R_{a,1}^2+R_{a,4}^2+R_{a,5}^2}}+\frac{R_{a,4}R_{b,4}}{\sqrt{R_{a,1}^2+R_{a,4}^2+R_{a,5}^2}\sqrt{R_{a,1}^2+R_{a,4}^2+R_{a,5}^2}}=$$

$$\frac{R_{a,5}R_{b,5}}{\sqrt{R_{a,1}^2+R_{a,4}^2+R_{a,5}^2}\sqrt{R_{a,1}^2+R_{a,4}^2+R_{a,5}^2}}=\frac{5}{\sqrt{50}}\frac{5}{\sqrt{38}}+\frac{3}{\sqrt{50}}\frac{3}{\sqrt{38}}+\frac{4}{\sqrt{50}}\frac{2}{38}=0.96$$

Pearson Correlation. Pearson Correlation (PC) is a well-known metric that compares ratings where the effects of mean and variance have been eliminated is the Pearson Correlation (PC) similarity:

$$PC(u,v)=\frac{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)(r_{vi}-r_v)}{\sqrt{\sum_{i\in I_{uv}}(r_{ui}-\bar{r}_u)^2\sum_{i\in I_{uv}}(r_{vi}-\bar{r}_v)^2}} \tag{4}$$

In addition, for acquiring the similarity between two items *i* and j the ratings given by users that have rated both of these items is compared:

$$PC(i,j)=\frac{\sum_{u\in U_{ij}}(r_{ui}-\bar{r}_i)(r_{uj}-\bar{r}_j)}{\sqrt{\sum_{u\in U_{ij}}(r_{ui}-\bar{r}_i)^2\sum_{u\in U_{ij}}(r_{uj}-\bar{r}_j)^2}} \tag{5}$$

Spearman's Correlation Coefficient. Spearman's correlation coefficient is a rank coefficient that independent of the actual item rating values, estimates the difference in the ranking of the items in the profiles. First user's list of ratings is turned into a list of ranks, where the user's highest rating takes the rank of 1, and tied ratings take the average of the ranks for their spot (Herlocker et al., 2002). Herlocker et al. (1999) showed that Spearman's performs similarly to Pearson's for user-based CF.

$$SRC(i,j)=\frac{\sum_{i\in I}(r_{a,i}-\bar{r}_a)(r_{b,i}-\bar{r}_b)}{\sqrt{\sum_{i\in I}(r_{a,i}-\bar{r}_a)^2\sum_{i\in I}(r_{b,i}-\bar{r}_b)^2}} \tag{6}$$

The Spearman Correlation Coefficient for user-user similarity between two users *a* and *b* have been represented in Eq. (6). It is declared regarding the set of all co-rated items (*I*) that $r_{a,i}$ and $r_{b,i}$ indicate rank each user gave to each item *i* and $\bar{r}_a$ and $\bar{r}_b$ finally indicate each user's average rank. Once again, the correlation is measured on a scale between -1 to +1 where, -1 implies the objects are completely dissimilar, +1 implies they are completely similar and 0 implies that the objects do not have any relationship to each other.

Adjusted Cosine Similarity. To overcome the shortcoming of standard cosine similarities metric for item-based CF that does not take individual users' rating scales into account, adjusted Cosine similarity method was presented by Sarwar et al. (2001). After calculating the similarity between two items *i* and *j*, by subtracting the user's average rating from each co-rated pair, the adjusted metric compensates result. The formula seems similar to the Pearson coefficient for item similarities but it considers user average rather than the item average that is subtracted from each co-rated pair. Eq. (7) represents the similarity between items *i* and *j*.

$$Sim(i,j)=\frac{\sum_{u\in U}(r_{u,j}-\bar{r}_u)(r_{u,j}-\bar{r}_u)}{\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r}_u)^2\sum_{u\in U}(r_{u,j}-\bar{r}_u)^2}} \tag{7}$$

An International Journal
JSCDSS
E-ISSN: 2289-8603

Mean Squared Difference (MSD). For estimating the similarity between two users *u* and *v* MSD mature is applied as the reverse of the average squared difference between the ratings made by *u* and *v* on the same items (Shardanand and Maes , 1995):

$$MSD(u,v)=\frac{|I_{uv}|}{\sum_{i\in I_{uv}}(r_{ui}-r_{vi})^2} \tag{8}$$

While it could be modified to compute the differences on normalized ratings, the MSD similarity compared to PC similarity has one shortcoming that it does not capture negative correlations between user preferences (Gorii et al., 2007).

The Jaccard coefficient. The Jaccard coefficient is a measure for calculating the similarity between two users with binary profiles, i.e. ratings are not taken into account. Eq. (9) shows Jaccard for the similarity between two users *u* and *v*, determined by the profile intersection as a fraction of the profile union that values range between 0 and 1 are result of this measure, where 0 indicates there is no similarity and 1 indicates there is perfect similarity.

$$Sim(u,v)=\frac{|U\cap V|}{|U\cup V|} \tag{9}$$

Conditional Probability-based Similarity. Karypis (2001) proposed the conditional probability-based metric as a similarity metric for item-based collaborative filtering Top-N recommendations. The similarity between two items *i* and *j* is simply the probability of purchasing (rating) one given that the other has already been purchased. Thus the probability of purchasing *a* given that *b* has been purchased is determined as the number of users that purchased both items divided by the total number of users that purchased *b*. Note that this metric gives asymmetric similarities since ($P(i\,|\,j)\notin P(j\,|\,i)$). The similarity of *i* to *j* is given in Eq. (10) as:

$$Sim(i,j)=P(i\,|\,j)=\frac{freq(i,j)}{freq(j)} \tag{10}$$

According to the Deshpande and Karypis (2004), one of the shortcomings of an asymmetric metric is that each item tends to have high conditional probabilities with regard to the most favoured items. To solve this shortcoming, the following form of the conditional probability is presented in (Deshpande and Karypis, 2004):

$$Sim(a,b)=P(a\,|\,b)=\frac{\sum_{\forall u:R_{u,b>0}}R_{u,b}}{freq(a),(freq(b)^{\alpha})} \tag{11}$$

where $\alpha\in[0,\,1]$ and $freq(a)$ indicates the number of users that have a transaction on item in the training data, and R(*u*, *b*) is the (*u*, *b*) element in the normalized *i* user-item matrix. Clearly a number of different metrics have been tested as appropriate similarity metrics for CF. Sometimes the choice is defined by the associated profile data, for example if there are no numeric ratings and user preferences are binary then clearly a metric like Jaccard needs to be used. In general, the Pearson correlation coefficient is the measure of choice for user-based CF rating prediction, while the adjusted cosine measure is the measure of choice for item-based CF rating prediction. The Jaccard coefficient has performed best in our experimentation as the similarity metric for user-based collaborative Top-N recommendation, while Karypis (2001) propose using two different similarity metrics for item-based Top-N recommendation, the scaled cosine similarity and the conditional probability based similarity, both of which he found performed well with the conditional probability-based metric performing slightly better

Generally, the implementation of recommender systems by nearest neighbor algorithm has been very successful. But, there are some potential challenges such as Sparsity and Scalability. The problem of Sparsity influences the accuracy of recommendation systems. Although, commercial recommendation systems have large item sets, even active customers may have purchased well under 1% of the items. Accordingly, a recommendation system based on nearest neighbor selected from the customers may be unable to make effective item recommendations for a particular customer. As a result, the accuracy of recommendations may be poor. In addition, there exist many items and users in the dataset of recommender systems. Hence, finding the nearest neighbor requires computation for both the users and the items. This takes so much time with the millions of customers and items, a typical item and user-based recommender system will suffer serious Scalability problems.

As an example, in the cosine-based similarity method, suppose the rating item of the user is denoted by R_{ij} . Items that the users do not rated are zero ($R_{ij}=0$). Then,

$$R_{ij}=\begin{cases} r_{ij} & r_{ij}\neq 0\\ 0 & otherwise\end{cases} \tag{12}$$

where r_{ij} is the rating of user *i* on item *j*. R_{ij} equals to r_{ij} , if the user *i* rates the item *j*, else R_{ij} equals to zero.

This may alleviate the Scalability problem and enhance computational performance, however, in the case of the extreme sparsity of the items and the greatness of the quantity, the reliability of the assumption is poor. Because, in practice, the preference of the users is different for un-rating items and they cannot be the same rating, i.e., zero. Adjusted cosine similarity also has this problem.

In the correlation-based similarity methods, considering u_i as item set rated by the user *I* and u_j the item set rated by the user *j*, the intersection of items rated both by user *i* and user *j* is then $u_i \Lambda u_j$. In common sense, the similarity between two users u_i and u_j is high similarity when users have rated items very close for the two users. When the rating items are very sparse, the item set that both rated by the two users are very small, only one or two items. In this situation, even the two users have very high similarity; we cannot say they are similar actually. This method also has some deficiency. In addition, the traditional similarity measure cannot measure the similarity between the users effectively when the rating data are extremely sparse. This resulted in the inaccurate neighbors and the decrease of the recommender accuracy.

2.2. Sampling

Sampling is one of the necessary steps in data mining for selecting a part of appropriate information from a huge data set. Sampling is performed to create the abstraction of complicated problem as well as to obtain a sub set from a larger data set.

Two characteristics should be considered for the worthiness of the sample from the entire database which are the size and quality of the sample. The significance of sample size is rather easy to understand and the quality sample for one problem may not be a quality sample for another problem.

Computationally, the sampling process is too expensive and may this process be considered. Basically, several advantages have been considered for sampling (George et al., 1996):

(1) Greater economy: sampling can be applied as a tool for reducing the cost of maintaining data and also can be useful tool for reducing I/O costs as well as for data cleansing; (2) Reliability: usually it is not necessary to collect data about the entire environment for forming reliable generalizations under consideration and therefore samples may be more accurate and useful for a researcher in calculating the example of special problem; (3) Alleviate constraints in data collection: Data collection usually faces severe constraints in terms of, time, cost and effort and the sampling is a significant process to face these constraints; (4) Because the generalized samples is acquired by proper sampling methods from the entire database, therefore little information is lost; and (5) Greater scope: sampling has a greater opportunity regarding the wide range of information by keeping the quality and accuracy.

Generally, 7 groups of sampling have been studied as weighted, stratified, cluster, sequential, proportional, simple random, and multi-stage sampling (Saar-Tsechansky and Provost, 2004; Bryan et al., 2002; Chen et al., 2002; Scheffer and Wrobel, 2002; Palmer and Faloutsos, 2000; Srinvasan, 1999; Leung and Chen, 1999).

For example, simple random sampling chooses *n* samples tuple-by-tuple by forming random numbers between 1 and *N* from a population of size *N*. Systematic sampling is a random sampling technique which is more used by researchers for its simplicity and its periodic quality. Algorithms of simple random and systematic sampling are shown in the following. In these algorithms, F_{in} indicates the input from population of *N* tuples and F_{out} refers the selected samples as the output.

Furthermore, sampling as a usual task has been considered in many applications efforts to recommendation systems for assessing and choosing recommender systems algorithms. Some studies have been conducted on network/graph and sub-graph sampling that deal with sampling from a set of graphs and sub-graph as a population (Erdos et al., 1959; Erdos et al., 1960; Capobianco, 1982).

Algorithm RS (F_m, n)

$$\left[\begin{array}{l} begin \\ \quad F_{out} = \phi; \\ \quad while \ \ |F_{out}| \le n \ \ do \\ \qquad i := random(1, |F_{in}|); \\ \qquad F_{out} := F_{out} \cup \{t_i\}; \\ \quad end\ do \\ \quad return(F_{out}); \\ end; \end{array}\right.$$

Algorithm SS (F_{in}, n)

$$\left[\begin{array}{l} begin \\ \quad F_{out} = \phi; \\ \quad step := \lfloor |F_{in}| / n \rfloor; \quad i := start; \\ \quad while \ \ i \le |F_{in}| \ \ do \\ \qquad F_{out} := F_{out} \cup \{t_i\}; \ i := i + step; \\ \quad end\ do \\ \quad return(F_{out}); \\ end; \end{array}\right.$$

In the experimental analysis of the implemented recommender systems, often a subset of the entire user-item interaction data derived is used by including all transactions associated with a subset of random selected users. In addition, almost in all cases, datasets used in the literature are also samples of a population data. Although, applying the sample data has long been a standard practice, however no formal study has been considered on sampling of the recommendation data. Fig. 3 shows a sample of user-item interaction matrix, user-item graph and projected user/item graphs. As can be seen in Fig. 3, $G = (C, P, E)$ implies bipartite graph with the user node set *C*, item node set $P = \{p_1, \dots, p_N\}$, and edge set $E = \{< c_i, p_j >\}$. In this context, finding a sampling method for producing a sub-graph of *G* to provide best ratio among levels in sample size is a recommendation sampling problem.

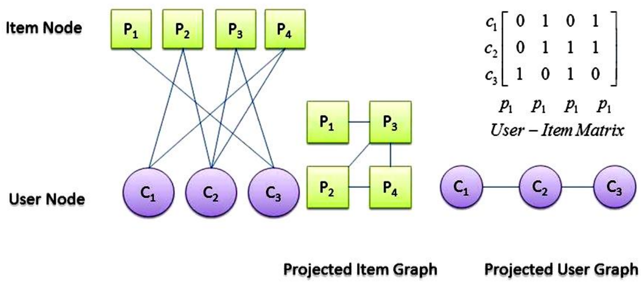


Fig. 3. A sample of user-item interaction matrix, user-item graph and projected user/item graphs

2.3. Dimensionality Reduction

Dimensionality reduction is an interesting alternative to feature selection. Similar to feature selection, it provides a low-dimensional representation of the data which can then be used as input for supervised or semi-supervised machine learning techniques. The low dimensional representation of the data also can be used in the unsupervised machine learning techniques such as clustering. Unlike feature selection, dimensionality reduction preserves information from all the original input variables. In fact, if the data indeed lies on a low-dimensional manifold, it may preserve almost all of the original information while representing it in a way that simplifies learning. Dimensionality reduction techniques are also used for visualizing the projected data (two or three dimensions at a time) so as to better understand it. They make new entities as combinations of the original entities in order to decrease the dimensionality

An International Journal JSCDSS
E-ISSN: 2289-8603

of a dataset. Rather than choosing a subset of the features, these techniques aim at reducing the dimension by keeping the originality with a lower dimension and redundancy is removed. In addition, selection of an appropriate structure for storage and analysis of complex datasets is vital in the design of data mining and machine learning experiments. In some situations, it is advantageous or even necessary to apply the dimensionality techniques for reducing the data to a reasonable size while retaining the appropriate nature of the original information. Furthermore, dimensionality reduction techniques help to eliminate noisy and irrelevant terms. As a result, savings in computational resources, storage, and memory requirements could be achieved. Fig. 4 demonstrates dimension reduction as a pre-processing stage.

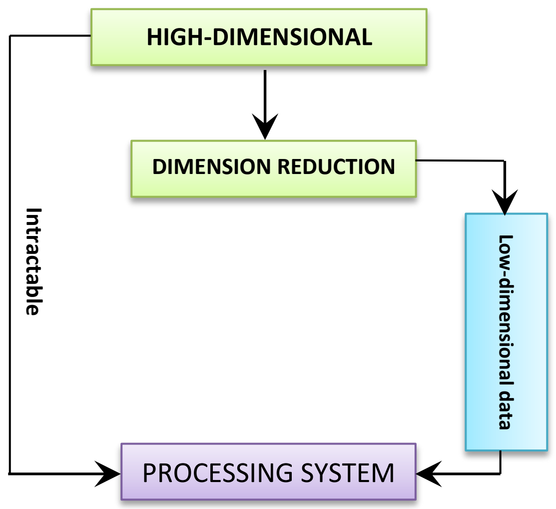


Fig. 4. Dimension reductions as a pre-processing stage

Many learning algorithms perform poorly in a high dimensional space given a small number of learning samples. Often some features in the data set are just "noise" and thus do not contribute to (and sometimes even degrade) the learning process. This difficulty in analyzing data sets with many features and a small number of samples is known as the curse of dimensionality. Dimensionality reduction can circumvent this problem by reducing the number of features in the data set before the training process. This can also reduce the computation time, and the resulting classifiers take less space to store. Models with small number of variables are often easier for domain experts to interpret. Dimensionality reduction is also invaluable as a visualization tool, where the high dimensional data set is transformed into two or three dimensions for display purposes. This can give the system designer additional insight into the problem at hand. The main drawback of dimensionality reduction is the possibility of information loss. When done poorly, dimensionality reduction can discard useful instead of irrelevant information. No matter what subsequent processing is to be performed, there is no way to recover this information loss.

For handling data which is inherently linear in nature, linear dimensionality reduction techniques such as Singular Value Decomposition (SVD) (Gao and Zhang, 2005; Castelli, 2003), Principal Component Analysis (PCA) (Hotelling, 1933; Pearson, 1901), Linear Discriminant Analysis (LDA) (Hastie, 2001; Duda, 2001; Fukunaga, 1990), Non-negative Matrix Factorization (NMF) (Lee and Seiing, 1999; Lee and Seiing, 2000; Lee and Lee, 2001; Guillamet and Vitria, 2002; Guillamet et al., 2003; Pauca et al., 2004), Independent Component Analysis (ICA) (Comon, 1994; Bell and Sejnowski, 1995; Hyvarinen, 1999) are used. Whereas, for handling nonlinear data with a certain type of topological manifold, nonlinear techniques such as Locally Linear Embedding (LLE) (Roweis and Saul, 2000) and Self-Organizing Map (SOM) (Kohonen, 1990; Villmann, 1997), Higher Order SVD (HOSVD), and Isometric Feature Mapping (ISOMAP) (Tenenbaum et al., 2000) are used.

Dimensionality reduction techniques have been extensively applied for solving many problems in the recommender systems field. They help to overcome the problems by transforming the original high-dimensional space into a lower-dimensionality. They have demonstrated to be effective in overcoming the major problems such as sparsity and scalability problems in recommender systems (Nilashi et al., 2014b; Nilashi et al., 2014c). In the following, we introduce some well-known dimensionality reduction techniques.

2.3.1. Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) of a real matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a very powerful computation tool for solving many problems in numerical linear algebra. From the theoretical point of view, it is also used in numerical analysis as a decomposition that reveals important information about the matrix and the problem it is involved with. The mathematical background of SVD is provided as follows (Gao and Zhang, 2005; Wall et al., 2003; Cichocki et al., 2009).

Given an $N \times M$ matrix $A \in R^{N\times M}$ of rank $r \le min(N,M)$, the SVD states that there exist orthogonal matrices $U \in R^{N\times N}$ and $V \in R^{M\times M}$ such that A is factored in the form:

$$A = U\Sigma V' \tag{13}$$

where $\Sigma \in R^{N\times M}$ is an $N\times M$ diagonal matrix, partitioned in the form:

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \tag{14}$$

with Σ_r a square diagonal matrix in $R_{r\times r}$:

$$\Sigma_r = diag(\sigma_1, \sigma_2, \ldots, \sigma_r) \tag{15}$$

with positive diagonal entries called the singular values of A and arranged in decreasing order:

$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r > 0 \tag{16}$$

SVD has been used as the key element of many CF techniques (Sarwar et al., 2000; Sarwar et al., 2002; Shlens, 2005; Canny, 2002; Koren, 2008; Sarwar et al., 2000;

An International Journal JSCDSS
E-ISSN: 2289-8603

Sarwar et al., 2002; Shlens, 2005; Canny, 2002; Koren, 2008; Adomavicius and Tuzhilin, 2005; Koren et al., 2009; Sarwar et al., 2000; Schafer et al., 2001) to improve the recommendation quality. In GroupLens, SVD has been applied in at least three distinct cases, which are: (i) in an approach which reduces the dimensionality of the user–item space and forms predictions in the reduced space, never building explicit neighborhoods during that procedure (Billsus et al., 1998), (ii) in an approach that generates a user neighborhood in the SVD reduced space and then applies normal User-based CF (Sarwar et al., 2000), and (iii) in an approach that aims at increasing the scalability by applying folding-in for the incremental computation of the user–item model (Sarwar et al., 2001). In (Sarwar et al., 2000), authors used SVD to reduce the dimensionality of recommender system databases in producing Top-N lists based on a real-life customer purchase database from an e-commerce site.

Billsus and Pazzani (1998) utilized SVD in order to formulate CF as a classification problem. In their work, they reduce the dimensions of a data matrix SVD before they feed it into an Artificial Neural Network (ANN).

In (Ariyoshi and Kamahara, 2010), authors proposed a hybrid information recommendation method using SVD to reduce the data size for time complexity improvement. The proposed method combined two reduction steps in reducing the number of documents used on the basis of the users' rating pattern and reducing the number of terms used on the basis of the term frequency pattern of these reduced documents. In (Kim and Cho, 2003), authors proposed a recommendation methodology, called Web usage mining driven collaborative filtering recommendation methodology using SVD (WebCF-SVD), to address the sparsity and scalability problems of CF-based recommender systems.

Fig. 5 shows the general procedure of SVD for dimensionality reduction in User-Item matrix in recommender systems that A implies the rating matrix of user to items, U refers to user concepts matrix, S indicates singular values and V' that is a reprehensive of item concepts. Therefore, using SVD algorithm, it is possible to convert a given matrix A into $A = USV^T$.

Accordingly, we can decompose the matrix A with rank *r* using SVD. By considering the matrix A with rank *k* we obtain the matrix B using Eq. (2) that gives the approximation of A based on arbitrary *k*.

Example 1: Using SVD, we can decompose the matrix A and get an appropriate approximation by considering two dimensions. Also in recommendation context, the user-item matrix for example in Table 3 with dimension 4×6 can be reduced in two dimensions (see Fig. 6) to get latent relationships between its objects.

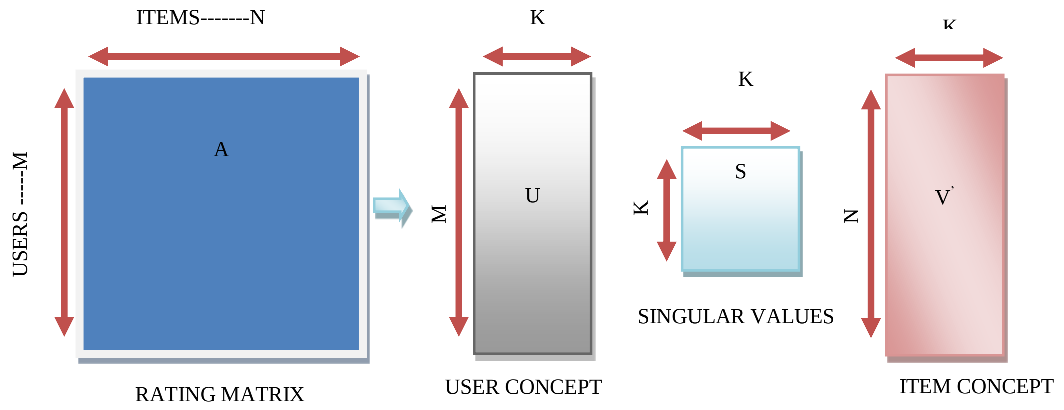


Fig. 5. Illustrating the basic SVD theorem

$$A = \begin{pmatrix} 11.08 & 6.82 & 1.76 & -6.82 \\ 2.5 & -1.01 & -2.6 & 1.19 \\ -4.88 & -5.07 & -3.21 & 5.2 \\ -0.49 & 1.52 & 2.07 & -1.66 \\ -14.04 & -12.4 & -6.66 & 12.65 \\ 0.27 & -8.51 & -10.19 & 9.15 \\ 9.53 & -9.84 & -17 & 11 \\ -12.01 & 3.64 & 11.1 & -4.48 \end{pmatrix} = U_{8\times8}\Sigma_{8\times4}V_{4\times4}{}^{T} \Rightarrow A_{Approx} = \begin{pmatrix} 11.08 & 6.82 & 1.76 & -6.82 \\ 2.5 & -1.01 & -2.6 & 1.19 \\ -4.88 & -5.07 & -3.21 & 5.2 \\ -0.49 & 1.52 & 2.07 & -1.66 \\ -14.04 & -12.4 & -6.66 & 12.65 \\ 0.27 & -8.51 & -10.19 & 9.15 \\ 9.53 & -9.84 & -17 & 11 \\ -12.01 & 3.64 & 11.1 & -4.48 \end{pmatrix} = \begin{pmatrix} -0.25 & -0.45 \\ 0.07 & -0.11 \\ 0.21 & 0.19 \\ -0.08 & 0.02 \\ 0.50 & 0.55 \\ 0.44 & -0.03 \\ 0.59 & -0.43 \\ -0.30 & 0.51 \end{pmatrix} \times \begin{pmatrix} 36.83 & 0.00 \\ 0.00 & 26.24 \end{pmatrix} \times \begin{pmatrix} -0.04 & -0.54 & -0.61 & 0.58 \\ -0.92 & -0.17 & 0.33 & 0.14 \end{pmatrix} \approx A$$

Table 3
Ratings of user-item

	User 1	User 2	User 3	User 4
Item 1	5	5	0	5
Item 2	5	0	3	4
Item 3	3	4	0	3
Item 4	0	0	5	3
Item 5	5	4	4	5
Item 6	5	4	5	5

An International Journal
JSCDSS
E-ISSN: 2289-8603

$$U=\begin{pmatrix}0.41 & 0.33\\ 0.40 & 0.53\\ 0.41 & 0.13\\ 0.33 & -0.58\\ 0.41 & -0.51\\ 0.48 & 0.00\end{pmatrix} \quad S=\begin{pmatrix}48.98 & 0\\ 0 & 10.15\end{pmatrix} \quad V=\begin{pmatrix}0.556 & -0.044\\ 0.493 & -0.012\\ 0.467 & -0.683\\ 0.479 & 0.729\end{pmatrix}$$

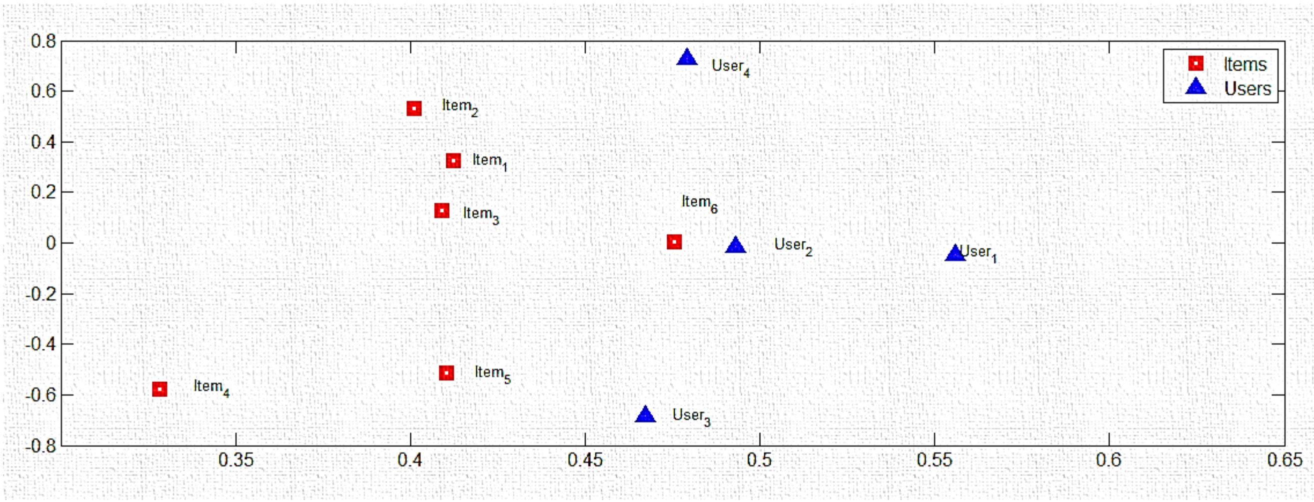


Fig. 6. 2-Dimensional space of applying SVD for users and items

The SVD approach can approximate the missing rating values based on the matrix factorization $(\hat{r}=(U_k S_k^{1/2})_u.(S_k^{1/2}V_k^{'})_i))$. To do so, the following steps need to be performed by an example to estimate an unknown rating to the active user.

Let $Y=\{a_{ij}\}\in R^{m,n}$ be a user-item matrix that contains the users' ratings $U=\{u_1, u_2, \dots, u_m\}$ to the items $U=\{i_1, i_2, \dots, i_n\}$.The goal is to predict unknown ratings in this matrix.

Step 1: The user-item matrix $R^{m,n}$ with raw data is converted to the dense matrix $D^{m,n}$.

Step 2: Matrix $D^{m,n}$ is normalized using Z-score to the matrix $Z^{m,n}$ by

$$Z_{ij}=\frac{D_{ij}-\bar{D}_{ij}}{\sigma_j},$$

where B_j and σ_j indicate average value and standard deviation (SD) for the ratings in the B_j, respectively. The B_j is calculated by:

$$B_j=\frac{1}{m}\sum_{i=1}^{m}B_{ij},\quad \sigma_j^2=\frac{1}{m-1}\sum_{i=1}^{m}(B_{ij}-\bar{B}_j)^2$$

Step 3: The SVD method is applied on Z.

Step 4: An approximation of Z is calculated as Z_d.

Step 5: P_{ij} is calculated based on $B^j+\sigma_j(Z_d)_{ij}$.

Example: Considering the user-item matrix R, we have:

$$R=\begin{array}{c}\\ U_1\\ U_2\\ U_3\\ U_4\end{array}\begin{array}{c}\begin{array}{cccccc}I_1 & I_2 & I_3 & I_4 & I_5 & I_6\end{array}\\ \begin{pmatrix}2 & 3 & 5 & ? & 1 & 4\\ ? & 3 & ? & 4 & ? & 3\\ 3 & 5 & ? & 2 & 4 & ?\\ 3 & 3 & 4 & ? & 3 & ?\end{pmatrix}\end{array},\quad p_{23}=?$$

$$(\text{Calculating } \bar{B}^i)\xrightarrow{Step\,2}\begin{pmatrix}\bar{B}^1\\ \bar{B}^2\\ \bar{B}^3\\ \bar{B}^4\\ \bar{B}^5\\ \bar{B}^6\end{pmatrix}=\begin{pmatrix}2\\ 3.5\\ 2.25\\ 1.5\\ 2\\ 1.75\end{pmatrix},\quad (\text{Calculating } \sigma_i)$$

$$\xrightarrow{Step\,2}\begin{pmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \sigma_6\end{pmatrix}=\begin{pmatrix}1.41\\ 1.00\\ 2.66\\ 1.91\\ 1.83\\ 2.06\end{pmatrix}$$

$$\xrightarrow{Step3(Z-scores)} Z_{4\times 6}=\begin{pmatrix}0.00 & -0.50 & 1.04 & -0.79 & -0.55 & 1.09\\ -1.42 & -0.50 & -0.85 & 1.31 & -1.10 & 0.61\\ 0.71 & 1.50 & -0.85 & 0.26 & 1.10 & -0.85\\ 0.71 & -2.00 & 0.66 & -0.79 & 0.55 & -0.85\end{pmatrix}$$

$$\xrightarrow{Step\;4(Z_d,d=2)}$$

$$\begin{pmatrix} 0.05 & 0.45 \\ -0.05 & -0.71 \\ -0.89 & -0.18 \\ -0.45 & 0.50 \end{pmatrix} \times \begin{pmatrix} 3.17 & 0.00 \\ 0.00 & 2.86 \end{pmatrix} \times \begin{pmatrix} -0.32 & 0.31 \\ -0.67 & -0.05 \\ 0.32 & 0.72 \\ 0.09 & -0.58 \\ -0.38 & 0.20 \\ 0.44 & -0.08 \end{pmatrix} =$$

$$\begin{pmatrix} -0.17 & -1.12 & 0.63 & -0.37 & -0.45 & 0.39 \\ -1.35 & 0.01 & -0.47 & 0.99 & -1.24 & 1.02 \\ 0.68 & 1.60 & -0.75 & 0.21 & 1.04 & -0.87 \\ 0.81 & -1.37 & 1.12 & -1.20 & 0.39 & -0.30 \end{pmatrix}$$

$$\xrightarrow{Step\ 5} p_{23} = \bar{B}_3 + \sigma_3 (Z_d)_{23} \approx 1$$

It should be noted that the precision of the estimated ratings is strongly dependent on the dimension of the decomposed matrices.

2.3.2. Higher Order SVD (HOSVD)

To represent and recognize high-dimensional data effectively, the dimensionality reduction is conducted on the original dataset for low-dimensional representation. Visualizing, comparing and decreasing processing time of data are the main advantages of dimensionality reduction techniques. SVD and HOSVD are two powerful techniques of the dimensionality reduction for matrix and tensor decomposition, respectively.
The SVD of a real matrix $\mathbf{A} \in R^{n \times n}$ is a very powerful computation tool for solving many problems in numerical linear algebra. From the theoretical point of view, it is also used in numerical analysis as a decomposition that reveals important information about the matrix and the problem it is involved with. HOSVD proposed by De Lathauwer et al. (2000) is a generalization of the SVD that can be applied on tensors. In many applications, involving tensor data the objective is to compute low-rank approximations of the data for modeling, information retrieval and explanatory purposes. These approximations are usually expressed in terms of tensor decompositions. In the following, we explain the tensor decomposition for HOSVD in 3rd and 4th order tensors.

Definition 1 (Unfolding). The mode-n unfolding of tensor $\underline{A} \in R^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers into columns of a matrix. More specifically, a tensor element ($i_1, i_2, \ldots, i_N$) maps onto a matrix element (i_n, j), where

$$j = 1 + \sum_{p \neq n} (i_p - 1) j_p, \quad with \qquad J_p = \begin{cases} 1, & if \ p = 1 \ or \ if \ \ p = 2 \ and \ n = 1, \\ \prod_{m \neq n}^{p-1} I_m & otherwise \end{cases} \tag{17}$$

For a tensor of order N = 3 (see Fig. 7), we have 3 modes n = 1, 2, 3, and for a tensor of order N = 4 we have 4 modes and so on. The n-mode fibers are the columns of the n-mode unfolded matrix. The flattening of a tensor $\underline{A}$ in its n-mode is symbolized by the matrix $\mathbf{X}(n)$. For example, for a tensor of order N = 3, by fixing the n-th index to some value j, there exist three matrix unfolding as (De Lathauwer, 2004):

$$\begin{aligned} &\textbf{mode-1}: \ j = i_2 + (i_3 - 1) I_3, \\ &\textbf{mode-2}: \ j = i_3 + (i_1 - 1) I_1, \\ &\textbf{mode-3}: \ j = i_1 + (i_2 - 1) I_2. \end{aligned} \tag{18}$$

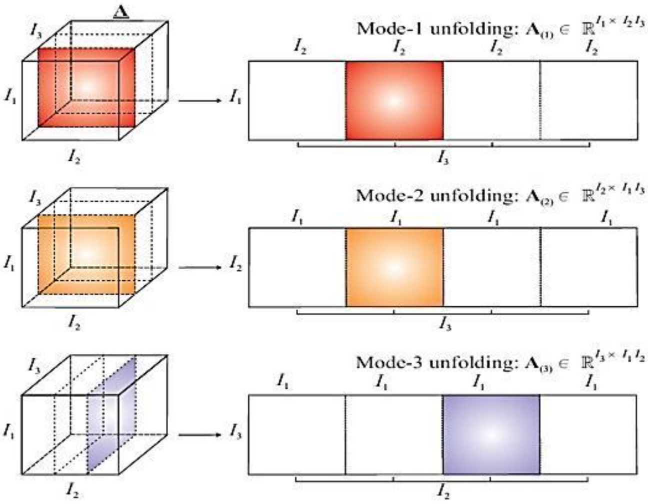


Fig. 7. Unfolding of a third-order tensor.

Definition 2 (Frobenius-norm). The Frobenius-norm of a tensor $\underline{A}$ of size $I \times J \times K$ is defined by Eq. (19).

$$\|\underline{A}\|_F = \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} |t_{ijk}|^2 \right)^{1/2} \tag{19}$$

The Frobenius-norm can be interpreted as a measure for the "size" of the tensor. The square of this norm can be seen as the "energy" in the tensor.
Definition 3 (Rank). The rank of a tensor can be defined as the minimal number of terms when expressing a tensor as a sum of rank-one tensors. The second way to define a rank of a tensor is given by the dimension of the subspaces spanned by the different n-mode vectors. Given an order n tensor $\underline{A}$ we write

$$\text{Rank}(\underline{A}) = (r_1, \ldots, r_n), \quad r_i = \dim(\text{span}(\mathrm{A}^{(i)})) \tag{20}$$

where $\mathbf{A}(i)$ is the matricization of $\underline{A}$ along mode i.
This is called the multilinear rank of a tensor. For tensors in general the ranks r_i are different. The approximation problem of an order three tensor $\underline{A} \in R^{I_1 \times I_2 \times I_3}$ is stated

$$\min_{\underline{B}} \|\underline{A} - \underline{B}\|, \qquad \text{Rank } \underline{B} = (r_1, r_2, r_3) \tag{21}$$

Assuming the rank constraint on $\underline{B}$, we can decompose $\underline{A} = (\mathbf{U}, \mathbf{V}, \mathbf{W}).\underline{S}$ where $\mathbf{U} \in R^{I \times r_1}$, $\mathbf{V} \in R^{J \times r_2}$, $\mathbf{W} \in R^{K \times r_3}$ have full column rank and $\underline{S} \in R^{r_1 \times r_2 \times r_3}$.
Definition 4. Every complex $(I \times J \times K)$-tensor $\underline{X}$ can be written as the product $\underline{X} = (\mathbf{U}, \mathbf{V}, \mathbf{W}).\underline{S}$, where $\mathbf{U} \in R^{I \times I}$, $\mathbf{V} \in R^{J \times J}$, $\mathbf{W} \in R^{K \times K}$, are orthogonal matrices, and the tensor $\underline{S} \in R^{I \times J \times K}$ is all-orthogonal and we have

$\|\underline{S}(1,:,:)\| \geq \|\underline{S}(2,:,:)\| \geq ... \geq 0,$
$\|\underline{S}(:,1,:)\| \geq \|\underline{S}(:,2,:)\| \geq ... \geq 0,$
$\|\underline{S}(:,:,1)\| \geq \|\underline{S}(:,:,2)\| \geq ... \geq 0,$

which are the 1-mode, 2-mode, and 3-mode singular values, also denoted $\sigma_i^{(1)}, \sigma_i^{(2)}, \sigma_i^{(3)}$. The tensor $\underline{S}$ in the HOSVD is in general full and not sparse or diagonal as $\mathbf{\Sigma}$ (a matrix contains of singular values) in the SVD of a matrix. But the all-orthogonality concept is still valid in matrix SVD. The HOSVD is an important result both for analysis and applications with tensors since it gives an ordering of the basis vectors in **U**, **V** and **W**. In fact truncating the HOSVD, i.e. taking the first r_1, r_2 and r_3 columns from **U**, **V** and **W**, respectively and correspondingly truncating $\underline{S}$, will give a good approximation of $\underline{A}$.

Similarly, the HOSVD represents a 4th-order tensor $\underline{A} \in R^{I_1 \times I_2 \times I_3 \times I_4}$ as a product of another fourth-order tensor with four unitary matrices of sizes $I_j \times I_j$, respectively. In this case, the decomposition of the fourth-order tensor is given by

$$\underline{A} = \underline{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)} \Rightarrow$$

$$A_{ijkl} = \sum_{m=1}^{I_1} \sum_{n=1}^{I_2} \sum_{p=1}^{I_3} \sum_{q=1}^{I_4} S_{mnpq} U_{im}^{(1)} U_{jn}^{(1)} U_{kp}^{(1)} U_{lq}^{(1)} \quad (22)$$

The matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$ and $\mathbf{U}^{(4)}$ are the matrices containing the left singular vectors of the four matrices that one can obtain by attending the tensor $\underline{A}$ (Kolda and Bader, 2009). In other words, $\mathbf{U}^{(n)}$ is obtained via the SVD of $\mathbf{X}^{(n)}$, the n-mode matrix unfolding of the tensor $\underline{A}$ is defined as :

$$X^{(n)} = \mathbf{U}^{(n)} \times \mathbf{\Sigma}^{(n)} \times \mathbf{V}^{(n)T} \quad (23)$$

where $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ are the left and right side matrices of singular vectors, respectively. The matrix $\mathbf{\Sigma}^{(n)}$ represents the diagonal matrix containing the singular values of $\mathbf{X}^{(n)}$. Since $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$ and $\mathbf{U}^{(4)}$ are orthogonal, the core-tensor $\underline{S}$ can be easily estimated via the following expression:

$$\underline{S} = \underline{A} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)^T} \times_3 \mathbf{U}^{(3)^T} \times_4 \mathbf{U}^{(4)^T}, \quad (24)$$

where $\mathbf{U}^{(i)T}$ denotes the complex conjugate transpose of $\mathbf{U}^{(i)}$. The core-tensor $\underline{S}$ plays a role similar to that of the matrix of singular values $\mathbf{\Sigma}$ in the SVD. In fact, one can reduce the rank of the tensor by truncating the core tensor. However, the definition of rank for tensors is not as straightforward as for matrices. There are several definitions of “rank” (Kolda and Bader, 2009). The truncated HOSVD is defined as a multi-rank approximation. Or symbolically, if rank ($\mathbf{D}^{(j)}$) = I_j , j = 1, 2, 3, 4, then $\underline{A}$ has rank-(I_1, I_2, I_3, I_4). The truncated HOSVD consists of the representation of the 4th-order tensor $\underline{A} \in R^{I_1 \times I_2 \times I_3 \times I_4}$ by the product of four unitary matrices $\tilde{\mathbf{U}}^{(1)}$, $\tilde{\mathbf{U}}^{(2)}$, $\tilde{\mathbf{U}}^{(3)}$ and $\tilde{\mathbf{U}}^{(4)}$ of sizes $I_j \times R_j, R_j < I_j$, respectively, and a small core-tensor $\underline{S} \in R^{R_1 \times R_2 \times R_3 \times R_4}$.

$$\underline{A} \approx \tilde{\underline{A}} = \underline{S} \times_1 \tilde{\mathbf{U}}^{(1)} \times_2 \tilde{\mathbf{U}}^{(2)} \times_3 \tilde{\mathbf{U}}^{(3)} \times_4 \tilde{\mathbf{U}}^{(4)} \quad (25)$$

We denote $\tilde{\underline{A}}$ the rank-reduced approximation of tensor $\underline{A}$. Clearly, the matrices $\tilde{\mathbf{U}}^{(n)}$, n = 1, 2, 3, 4 contain the first R_n left singular vectors of the unfolded matrix $\mathbf{D}^{(n)}$.
With above explanations, we can define HOSVD in Algorithm 1 (Kolda and Bader, 2009).

Algorithm 1: Procedure for decomposing tensors via HOSVD	
Input	dth-order tensor $\underline{X} \in \mathbb{R}^{I_1 \times ... \times I_d}$, $pruning\ tuple(m_1,...,m_d) \in [1, I_1] \times ... [1, I_d]$
Output	dth-order tensor $\tilde{\underline{X}} \in \mathbb{R}^{I_1 \times ... \times I_d}$, $as\ the\ best\ rank-(m_1,...,m_d)\ approximation\ to\ \underline{X}$
Unfolding	**For** i=1, …, d **Compute** the unfolding $\mathbf{A}_{(i)}$ of $\underline{X}$ **End**
Matrix SVD	**For** i=1, …, d **Compute** the SVD $\mathbf{A}_{(i)} = \mathbf{U}_{(i)} \mathbf{\Sigma}_{(i)} \mathbf{V}_{(i)}^T$ **End**
Pruning	**For** i=1, …, d $\mathbf{W}_{(i)} := [u_{i,1}, ..., u_{i,m_i}]$ where $u_{i,1}$ column vectors of $\mathbf{U}_i$ **End**
Core tensor	**Compute** $\underline{S} := \underline{X} \times_1 \mathbf{W}^{(1)^T} \times_2 \mathbf{W}^{(2)^T} \times ... \times_d \mathbf{W}^{(d)^T}$
Approximation	**Compute** $\tilde{\underline{X}} := \underline{S} \times_1 \mathbf{W}^{(1)} \times_2 \mathbf{W}^{(2)} \times ... \times_d \mathbf{W}^{(d)}$

HOSVD is a robust method for dimensionality reduction. It is flexible to choose different for column in different mode of a tensor. The size of the data goes down to $r_1r_2r_3 + I_1r_1 + I_2r_2 + I_3r_3$ from $I_1I_2I_3$, and if $r_1 = r_2 = r_3$ the size of the data goes down to $r^3 + r\left(I_1 + I_2 + I_3\right)$. If we flat the tensor into a $I_1 \times I_2I_3$ matrix, the size of the data only goes down to $R^2 + R\ \left(I_1 + I_2I_3\right)$.

In the field of recommender systems, some recommendation models, which use three dimensional tensors for recommending music, tags and objects, have been proposed. Recommender models, using HOSVD for dimension reduction, have been proposed for recommending personalized music (Ruxanda and Manolopoulos, 2008), Tags (Symeonidis et al., 2008; Rashidi et al., 2015) and multi-criteria CF (Nilashi et al., 2014b). Symenonidis et al. (2008) introduced a recommender based on HOSVD where each tagging activity for a given item from a particular user is represented by value 1 in the initial tensor, all other cases are represented with 0.

In the area of Personalized Web Search, Sun et al. (2005) proposed CubeSVD to improve Web Search. Xu et al. (2006) used HOSVD to provide item recommendations. Thus, they compared their work with a standard CF algorithm, without focusing in tag recommendations. In (Leginus et al., 2012), authors utilized clustering techniques for reducing tag space that improves the quality of recommendations and also the execution time of the factorization and decreases the memory demands. Their proposed method is adaptable with 3rd order tensor decomposition methods such as HOSVD. They also introduced a heuristic method to speed-up parameters tuning process for HOSVD recommenders. Symenonidis et al. (2009) developed a recommender based on HOSVD where each tagging activity for a given item from a particular user is represented by value 1 in the initial tensor, all other cases are represented with 0. The HOSVD factorization of a tensor results into an approximated tensor which reveals the latent relationships and patterns of the users.

2.4. Classification

Classification methods play an important role in data mining tasks by classifying the available information based on some characteristics of the elements.

Since many decision-making tasks are reprehensive of classification problem or can be easily formulated into a classification problem, e.g., prediction and forecasting tasks, therefore choosing proper classification method suited to the type of problem is very important.

Many of classification techniques have been used to decision-making scenarios such as business failure prediction (Tam and Kiang, 1992), portfolio management (Trippi and Turban, 1993), and debt risk assessment (Kiang et al., 1993) and wide variance in the performance of classification algorithms under different scenarios have been provided by some researches (Dietterich et al., 1995; Meila and Heckerman, 2001; Tam and Kiang, 1992).

2.4.1. Supervised classification

Supervised classification is one of the most important tasks in data mining (Han et al., 2002; Wu et al., 2008; Hastie et al., 2001), which seeks procedures for classifying objects in a set (i.e. Ω) into a set of classes (i.e. C) with labels or known categories in advance as a trained set. It has been used in many application field such as cancer diagnosis (Guyon et al., 2002; Mangasarian et al., 1995), machine vision (Papageorgiou et al., 1998), text categorization (Sebastian, 2002) and classification of gene expression data (Furey et al., 2000; Zhang et al., 2006). Mathematical optimization has played a critical role in supervised classification (Bennett and Mangasarian, 1992; Herna´ndez, 2006; Bradley et al., 2002; Doumpos et al., 2006; Glady et al., 2009). Nearest-neighbor methods (Cover and Hart, 1967; Dasarathy, 1991; Kim et al., 2009), Linear Discriminant Analysis (LDA) (Fisher ,1936), classical Logistic Regression (LR) (Hastie et al.,2001), Support Vector Machines (SVMs), Classification decision trees (Kim et al., 2002) such as Classification and Regression Trees (CARTs) (Breiman, 1984), C4.5 (Quinlan, 1993), Rule-Based (Cohen, 1995; Basu et al., 1998), Artificial Neural Network (ANN) (Zurada,1992; Ibnkahla, 2000; Christakou and Stafylopatis, 2005; Hsu et al., 2007), Bayesian Belief Networks (BBN) (Pronk et al., 2007; Yu et al., 2004; Zhang and Koren, 2007) and Association rule (Cho et al., 2002) are the well-known supervised classification methods.

Support Vector Machines (SVMs). Support Vector Machines (SVMs) is one of machine learning methods for supervised learning that has proved to be one of the successes of mathematical optimization (Vapnik, 1995; Vapnik, 1998; Joachims, 1998). SVMs are a family of machine learning techniques for tasks such as classification and regression (Vapnik, 1995; Cristianini and Shawe-Taylor, 2000; Farahmand et al., 2014a; Farahmand et al., 2015).

They are inherently two-class classifiers however, can be extended for multi-class. In SVM, the classification is done by intersecting a hyperplane through the feature space that separates one cluster of similarly labeled training data from another. The SVM learns the parameters for this hyperplane by maximizing the margin from the hyperplane to the two training data clusters. More advanced SVMs will use soft margins that react gracefully to abnormally labeled data points and semi-overlapping data clusters that cannot be separated by a simple hyperplane. Though, the more overlap between the two clusters of data points, the worse a SVM will do. This can be mitigated by transforming the initial feature space into a higher dimensional feature space by using the kernel trick.

The SVM incorporates the maximal margin strategy and the kernel method. The decision function of the SVM is an expansion of the kernel function. The decision function is

An International Journal
JSCDSS
E-ISSN: 2289-8603

used to predict the output for a given input. The maximal margin method is applied to improve the accuracy of the prediction. The main goal of this method is to find a hyperplane separating the data with the largest possible margin.

Fig. 8 shows two types of SVM namely: linear and non-linear SVM. The general idea in non-linear SVM is that original input space can always be mapped to some higher-dimensional feature space where the training set is separable.

In its simplest form, when examples are linearly separable and belong to one of two classes, the algorithm finds the hyperplane that correctly classifies the examples, but also has the maximum distance to the examples. This hyperplane is called a maximum margin hyperplane, since the margin between the classes is maximized.

Given a set of categories, which contain an arbitrary number of items, SVMs predict which category a new item belongs to. Figs. 8a and b illustrate this: Given 2 categories (red items and blue items), SVM creates a hyperplane which separates the two categories with the highest possible margin (H_1 and H_2 in Fig. 8b). The items which determine this margin are called the Support Vectors (2 blue and 1 red item in this example).

Grčar et al. (2006) confronted the *k*-NN algorithm with SVM in the CF framework. They found that *k*-NN is dominant on datasets with relatively low sparsity. They showed that on dataset with high to extremely high level of sparsity, *k*-NN is unable to form reliable neighborhoods. In such case it is best to use a model-based approach, such as SVM classifier or SVM regression. Another strong argument for using the SVM approaches on highly sparse data is the ability to predict more ratings than with the variants of the memory-based approach. Joachims (1998) demonstrated that on a Reuters-21578 data set, SVMs performed better than *k*-NN (86.4% accuracy vs. 82.6%).

Xia et al. (2006) proposed a heuristic method based on Smoothing SVM (SSVM) method from Lee and Mangasarian (1999) to overcome the problem caused by the sparsity of user-item matrix. They compared the heuristic method with item-based Zhang and Iyengar (2002) and user-based Breese (1998) CF algorithms.

The SVM's algorithm first learns from data that has already been classified, which is represented in numerical labels (e.g. 1, 2, 3, etc.) with each number representing a category. SVM then groups the data with the same label in each convex hull. Accordingly, it determines where the hyperplane(s) is by calculating the closest points between the convex hulls (Bennett, 2000). Once SVM determines where the hyperplane(s) is, it creates a model file that is used to classify new data. For example, any new data that lies on the side of the positive plane is classified with a positive label and any new data that lies on the side of the negative plane is classified with a negative label. In this consideration, SVM is excellent for classification, but requires labeled objects to use for training. Therefore, before training data, the classes of object must be determined using some machine learning techniques such as SVD for two dimensional data and HOSVD and SOM for high dimensional data. These approaches also can decompose the data to obtain approximation of the data with data to a lower-dimensional space. Basically, the cosine similarity method is used for determining the classes after decomposing the high dimensional data to the low dimension. The highest values that are similar (close) to the selected object are selected to be in one class. After defining the class numbers, the result will be as training data for SVM. The library LIBSVM has been provided for this purpose to be used for different type of SVM applications (Chang and Lin, 2006).

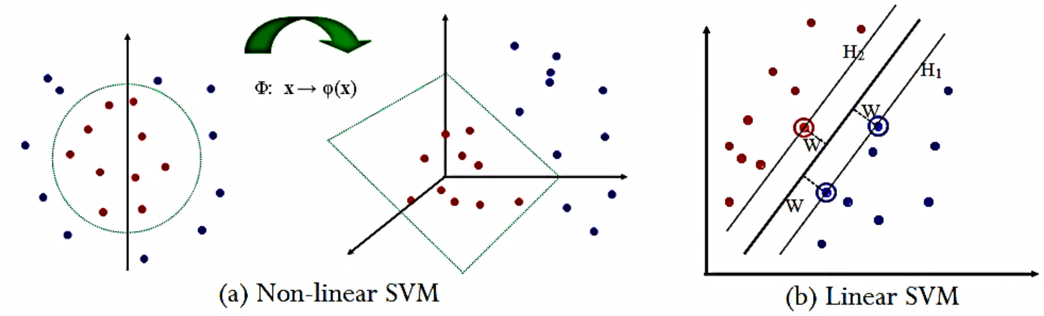


Fig. 8. Illustration of linear and non-linear SVM

***k*-Nearest Neighbor (*k*-NN).** *k*-Nearest Neighbor (*k*-NN) classifier is a well-known and powerful instance-based machine learning technique for classification data. By learning from all sorted training instances, *k*-NN simply can be applied to get results from training instances. The *k*-NN algorithm consists of two phases: training phase and classification phase. In training phase, the training examples are vectors (each with a class label) in a multidimensional feature space. In this phase, the feature vectors and class labels of training samples are stored. In the classification phase, *k* is a user-defined constant (see Fig. 9), a query or test point (unlabelled vector) is classified by assigning a label, which is the most recurrent among the *k* training samples nearest to that query point. In other words, the *k*-NN method compares the query point or an input feature vector with a library of reference vectors, and the query point is labelled with the nearest class of library feature vector. This way of categorizing query points based on their distance to points in a training dataset is a simple, yet an effective way of classifying new points. One of the main advantages of the *k*-NN method in classifying the objects is that it requires only few parameters to tune: *k* and the distance metric, for achieving sufficiently high classification accuracy. Thus, in *k*-NN based implementations, the best choice of *k* and distance metric for computing the nearest distance is an important task.

In *k*-NN classifier, the distance function usually is considered Euclidean distance when the input vectors and outputs are real numbers and discrete classes, respectively. Assume $x_1, x_2, \ldots, x_{mx}$ indicates the first row vectors and $y_1, y_2, \ldots, y_{my}$ indicates the second row vectors, the various distance metrics for measuring distance between x_s and y_t are defined as follows:

$$d_{st} = \sqrt{\sum_{j=1}^{n}(x_{sj} - y_{tj})^2} \tag{26}$$

$$d_{st} = \sum_{j=1}^{n} | x_{sj} - y_{tj}) \tag{27}$$

$$d_{st} = 1 - \frac{(x_s - \bar{x}_s)(y_t - \bar{x}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}\sqrt{(y_t - \bar{y}_t)(y_t - \bar{y}_t)'}}$$
$$\bar{x}_s = \frac{1}{n}\sum_j x_{sj} \quad \text{and} \quad \bar{y}_t = \frac{1}{n}\sum_j y_{sj} \tag{28}$$

where Eq. (26), Eq. (27) and Eq. (28) stand for Euclidean, City-Block and correlation distance metrics, respectively.

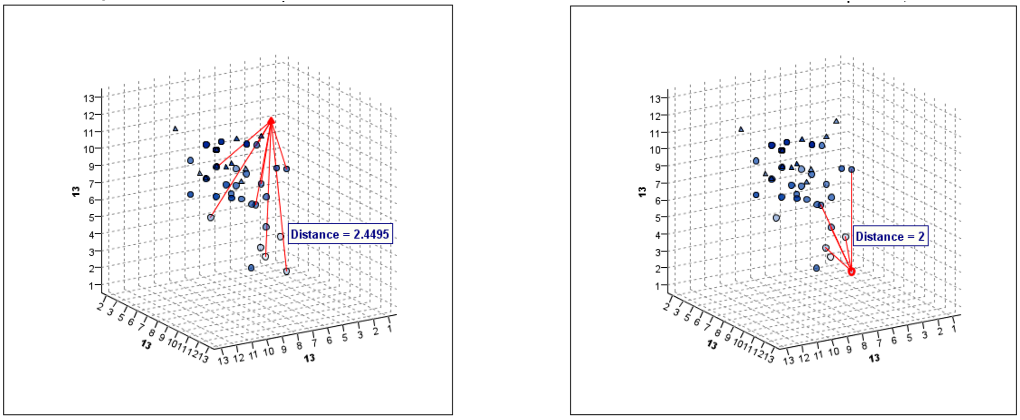


Fig. 9. *k*-Nearest Neighbor for *k*=7 and *k*=4

Many specialized distance and similarity metrics have been proposed for *k*-NN. The survey by Herlocker et al. (1999) mentions Pearson correlation, Spearman rank correlation, vector similarity, entropy, and means squared difference. Despite its simplicity, the *k*-NN method shows good accuracy with very short running time. In *k*-NN, generating a neighbourhood involved calculating the similarity between the given users within the user-item matrix. Similarity will be used to generate a recommendation for a specific user.

The algorithm follows these steps:

Step 1. Compare the similarity between all users with the active user.

Step 2. Select *n* users that have the highest similarity to build a neighbourhood.

Step 3. Compute the prediction based on this similarity matrix.

Within the user-based recommendation system, similarity between two users is calculated using the Pearson's correlation coefficient. Pearson's correlation has become a standard way of calculating correlation. Similarity between users u_i and u_k are calculated below:

$$sim_{ik} = corr_{ik} = \frac{\sum_{j=1}^{l}(r_{ij} - \bar{r}_i)(r_{kj} - \bar{r})}{\sqrt{\sum_{j=1}^{l}(r_{ij} - \bar{r}_i)\sum_{j=1}^{l}(r_{kj} - \bar{r})^2}} \tag{29}$$

where *l* is the total number of items, r_{ij} is the rating to a given item *j* by user *i*, and is the mean average rating given by user *i*.

Using rating data in Table 4, we can calculate and derive a similarity matrix between the users.

Table 4
Sample user rating to items

User\Laptop	Dell	Acer	Compaq	HP
Robby	4	3	3	4
Larry	2	2	?	2
Dan	?	4	3	2
Cory	5	3	2	?
Nick	?	5	3	1

An International Journal JSCDSS
E-ISSN: 2289-8603

$Avg(Cory) = {}^{10}\!/_{3} = 3.3333$, Avg(Dan)= ${}^{9}\!/_{3} = 3$, $Avg(Larry) = {}^{6}\!/_{3} = 2$, $Avg(Robby) = {}^{14}\!/_{4} = 3.5$

$Avg(Nick) = {}^{9}\!/_{3} = 3$

$Overlap(Cory, Nick) = (Acer, Compaq)$

$$Sim(Cory, Nick) = \frac{(3-3.3333)(5-3)+(2-3.3333)(3-3)}{\sqrt{(3-3.3333)^2+(2-3.3333)^2 \times (5-3)^2+(3-3)^2}} = -0.248$$

We can use this calculation and the given data in Table 4 to formulate the similarity matrix.

The prediction is a numerical value that represents a predicted opinion of the active user about a specific item. The prediction for a user-based CF algorithm needs both the user-item matrix and the similarity matrix.

$$pr_{aj} = \overline{r}_a + \frac{\sum_{j=1}^{3}(r_{ij}-\overline{r}_i)\times sim_{ia}}{\sum_{j=1}^{3}|sim_{ai}|} \qquad (30)$$

Eq. (30) shows how a prediction pr_{aj} represents the predicted opinion for the active user u_a bout item i_j. r_{ij} represents the rating that user u_i gives item i_j and $\overline{r}_i$ denotes the average rating for user u_i. Eq. (30) retrieves similarity from Eq. (29), which represents the similarity between the active user u_a and user u_i.

Also, we can have hybrid algorithms of *k*-NN and SVD for predicting unknown ratings for target user. To do so, the hybrid method uses the algorithms from both, as described as following:

Step 1. A dense matrix is calculated form raw rating matrix by one of the appropriate matrix filling method.
Step 2. The dense matrix is normalized.
Step 3. The SVD method is applied on normalized matrix to obtain U, S, V.
Step 4. $(U_d \sum_d^{1/2})$ is calculated.
Step 5. Similarity between users is calculated using

$$sim(a,b) = \frac{(U_d \sum_d^{1/2})^{(a)} . (U_d \sum_d^{1/2})^{(b)}}{\left\| U_d \sum_d^{1/2})^{(a)} \right\|_2 * \left\| U_d \sum_d^{1/2})^{(b)} \right\|_2} .$$

Step 5. Compute the prediction for unknown rating p_{ij} using Eq. (30).

Feedforward Backpropagation Neural Network (FBNN). A FBNN is a multilayer network that consists of an input layer, one or more hidden layers, and an output layer. All neurons in each layer are fully connected to all neurons in the successive layer. An input pattern is propagated through a hierarchy of layers in a forward direction; i.e. input, hidden, and output layers. In this research, a three-layer network consisting of an input layer, a hidden layer, and an output layer is applied for classification task. The backpropagation is the classical algorithm used for learning. It is an iterative gradient descent algorithm which is designed to minimize the mean squared error between the desired output and the generated output for each input pattern. After the input is propagated from the input layer through the output layer, an error is computed from the difference between the desired output and the generated output obtained from the output layer. If the error is not satisfied then the weights are modified while the error is propagated backward from the output layer to the input layer.

Multiclass NN classification involves building NNs that map the input feature vector to the network output containing more than two classes (Murphey and Luo, 2002).

Multi-Layered Perceptron (MLP) is most widely utilized in ANN paradigm that approximates nonlinear relationships existing between an input set of data (and the corresponding output data set (Nilashi et al., 2015c, Farahmand et al., 2014b). A three-layer MLP with a single intermediate layer housing a sufficiently large number of nodes can approximate any nonlinear computable function to an arbitrary degree of accuracy.

The number of hidden neurons in the hidden layer is one of the major issues to be considered in the establishment of a FBNN for classification. There are many algorithms used to determine the number of hidden neurons. Igelnik and Pao (1995) were found that at least 2D (D denotes the dimension of the input vectors) hidden neurons can be sufficient for approximating the posteriori probability in classification problem with arbitrary accuracy. In this research, the numbers of hidden neurons are freezed by applying 2D hidden neurons to all the experiments. With considering 2D hidden neurons the FBNN classifier obtained the best classification accuracy.

NNs also have been successful in recommendation system implementation (Postorino and Sarne 2011; Gong and Ye 2009; Gao and Wu 2009). Lee et al. (2002) proposed a recommender system which combines CF with Self-Organized Map (SOM) NN. Christakou et al. (2007) proposed a recommendation system based on content and CF for recommendations concerning movies. The content filtering part of the system was based on trained NNs representing individual user preferences. They evaluated the hybrid system on the MovieLens data. Postorino and Sarne (2011) proposed a NN hybrid recommender system which was able to provide customers, associated with XML-based personal agents within a multi-agent system called MARF, with suggestions about flights purchases. For solving data sparsity for CF, a personalized recommendation approach based on Back-Propagation NNs (BPNNs) and item based CF was presented by Gong and Ye (2009). In the method, they used the BPNNs to fill the null values in user-item matrix of ratings and item based CF to form nearest neighborhood. Kogel (2002) implemented a Java based platform for neural networks in the Weka system. Using this platform, they performed a series of experiments using data from the EachMovie database and additional information from the MovieLens and Internet Movie Databases. In order to make recommendations to the target users, they trained an Artificial Neural Network (ANN) for each target user using a portion of the dataset, allowing thereby the ANN to learn

An International Journal
JSCDSS
E-ISSN: 2289-8603

the taste of the target user. Next they tested trained neural network, by passing a set of test data through the ANN which produced predictions about which movies the target users would like and dislike. The movies predicted to be liked by the target users would then be recommended.

Nilashi et al. (2014c) proposed a new recommendation model to improve the recommendation quality and predictive accuracy of multi-criteria CF and solve the scalability and alleviate the sparsity problems in the multi-criteria CF. The experimental results of applying their approaches on Yahoo!Movies and TripAdvisor datasets showed the enhancement of multi-criteria CF recommendation quality and predictive accuracy. The experimental results also demonstrated that SVM dominates the *k*-NN and FBNN in improving the multi-criteria CF predictive accuracy evaluated by most broadly popular measurement metrics, F1 and mean absolute error.

Naïve Bayes classifier. The Naïve Bayes classifier is a classification method that is used for categorical data based on applying Bayes' theorem. The Naive Bayes classifier can be compactly represented as a Bayesian network as shown in Fig. 10.

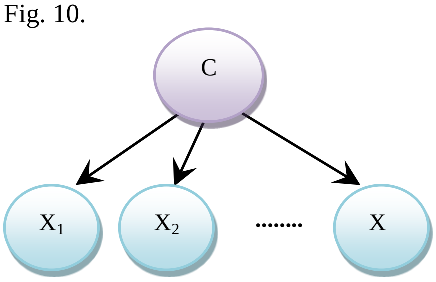


Fig. 10. Naive Bayes Classifier

The nodes represent random variables corresponding to the class label C, and the components of the input vector $X_1;\ldots;X_M$. The Bayesian network in Fig. 10 reveals the primary modeling assumption present in the naive Bayes classifier: the input attributes X_j are independent given the value of the class label C. This is referred to as the naive Bayes assumption from which the name of the classifier is derived.

In Bayesian networks, the value of a variable depends only on the value of its parent variables. For example, the probability of a variable x is described by the conditional probability $P(x \mid p_{ax})$ where p_{ax} is the parent variable of x. The joint probability of several independent variables can be factorized as $P(x_1, \ldots, x_N) = \prod_{i=1}^{N} P(x_i \mid pa_{x_i})$ which represents the dependency structure of the variables. Also, the following supposition is considered in Bayesian networks as

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)$$

In the contest of recommender systems, several studies have been conducted by Naïve Bayes classifiers. Condliff et al. (1999) proposed a Bayesian methodology for recommender systems that incorporates user ratings, user features, and item features in a single unified framework. In principle their approach addressed the cold-start, scalability and sparsity issues.

Ghazanfar and Prugel-Bennett (2010) proposed a unique switching hybrid recommendation approach by combining a Naive Bayes classification approach with the CF. Miyahara and Pazzani (2000) reported on SF with the Simple Bayesian Classifier. They proposed two representations for the Simple Bayesian Classifier. They found that the Sparse Data Model performs better than the Transformed Data Model and the typical correlation-based approach. The Transformed Data Model also outperformed the correlation-based approach although it shows similar accuracy to the correlation approach in some parts of the experiment with EachMovie dataset.

Yoshii et al. (2008) developed a hybrid recommender system that utilizes both the user's rating and musical content. The goal was to get more accurate recommendations referring to a large variety of artists. Here, a fundamental problem is that the observed rating scores and acoustic features incompletely represent user preferences. To overcome problem of the incompletely representation of user preferences, they used a Bayesian network model called a three-way aspect model.

Decision Tree. Decision trees are well recognized and useful prediction and classification tools. Using these tools, we are able to categorize and label objects into set of distinct classes. Each internal and terminal node in the decision tree represents a test on an attribute and a class prediction. Also, each branch in decision tree indicates a result of the test (Park et al., 2012; Hastie et al., 2009; Golbandi et al., 2011).

In general, in the recommender systems context, each node of the decision tree evaluates user preference toward a certain aspect. As with all CF methods, a decision tree is created and optimized by analyzing past user preferences. In the contest of recommender systems, several studies have been conducted by decision trees. Cho et al. (2002) suggested a personalized recommendation methodology for improving the effectiveness and quality of recommendations when applied to an Internet shopping mall. The suggested methodology was based on a variety of data mining techniques such as web usage mining, decision tree induction, association rule mining and the product taxonomy. Bouza et al. (2008) proposed an ontology-based decision tree algorithm that uses a domain ontology and a reasoner to split instances with more generalized features (superclasses of features) then the features in cases where generalized features in form of superclasses perform better. Nilashi et al. (2016) proposed a recommendation method using Classification and Regression Tree (CART) and Expectation Maximization (EM) for accuracy improvement of multi-criteria recommender systems. They also applied Principal Component Analysis (PCA) for dimensionality reduction and to address multi-collinearity induced from the interdependencies among criteria in multi-criteria CF datasets. They tested the method results on Yahoo! Movies and TripAdvisor datasets.

The list of research papers conducted in the context of recommender systems using supervised machine learning techniques is represented in Tables 5-7.

An International Journal JSCDSS
E-ISSN: 2289-8603

Table 5
Neural Network Technique in Recommender System

Data mining Technique	Reference
Neural Network	Han and Chen (2009); Martin-Guerrero, Lisboa, Soria-Olivas, Palomares, and Balaguer (2007); Postorino and Sarne (2011); Yuan and Tsao (2003); Gong and Ye (2009); Liu, Hsieh, and Tsai (2010); Christakou et al. (2007); Gao and Wu (2009); Kim et al. (2004); Lee, Hui, and Fong (2002); Lee et al. (2006); Christakou and Stafylopatis (2005); Sevarac et al. (2012); Nilashi et al. (2014c)

Table 6
KNN Technique in Recommender Systems

Data mining Technique	Reference
***k*-NN**	Lee, Hui, and Fong (2002); Tang and McCalla (2009); Hsu (2008);Munoz-Organero, Ramiez-Gonzalez, Munoz-Merino, and Kloos (2010); Lee, Park, and Park (2008); Kim, Kim, and Cho (2008); Lee, Park, and Park (2009); Blanco-Fernandez, Lopez-Nores, Pazos-Arias, Gil-Solla, and Ramos-Cabrer (2010); Liu and Shih (2005a); Zanker, Jannach, Gordea, and Jessenitschnig (2007); Chen, Cheng, and Chuang (2008); Zheng, Li, Liao, and Zhang (2010); Ganesan, Garcia-Molina, and Widom (2003); Naren, Benjamin, Batul, Ananth, and George (2001); Liu and Shih (2005b); Roh, Oh, and Han (2003); Han, Xie, Yang, and Shen (2004); Cho and Kim (2004); Zeng, Xing, Zhou, and Zheng (2004); Herlocker, Konstan, Terveen, and Riedl (2004); Li, Lu, and Xuefeng (2005); Hurley, O'Mahony and Silvestre (2007); Symeonidis, Nanopoulos, and Manolopoulos (2008); Symeonidis, Nanopoulos, Papadopoulos, and Manolopoulos (2008); Lee and Olafsson (2009); Jeong, Lee, and Cho (2009a); Jeong, Lee, and Cho (2009b); Chen, Wang, and Zhang (2009); Bobadilla, Serradilla, and Hernando (2009); Bobadilla, Serradilla, and Bernal (2010); Lee, Ahn, and Han (2007); Vezina and Militaru (2004); Kim, Yum, Song, and Kim (2005); Albadvi and Shahbazi (2009); Martinez et al. (2010); Martin-Vicente et al. (2010); Gemmell et al.(2009); Cohen and Fan (2000); Rohini and Ambati (2005); Nilashi et al. (2015a); Nilashi et al. (2014a); Nilashi et al. (2014b); Nilashi et al. (2014c)

Table 7
Decision Tree Technique in Recommender Systems

Data mining Technique	Reference
Decision Tree	Nilashi et al. (2016); Hernandez del Olmo, Gaudioso, and Martin (2009); Wang, Chiang, Hsu, Lin, and Lin (2009); Kim et al. (2002); Cho, Kim & Kim (2002); Yu, Ou, Zhang, and Zhang (2005); Lee and Yang (2003); Sun et al. (2011); Hijikata et al. (2006); Lee (2010); Golbandi et al., (2011); Nikovski and Kulev (2006); Bouza et al (2008); Cheng et al. (2009); Sergio (1999)

2.4.2. Unsupervised classification

Other popular type of classification technique is unsupervised classification (clustering) which doesn't need predefined classes of data (Lillesand and Kiefer, 2000). Clustering techniques aim to minimize the sum of squared error by minimizing the distance between the data object and the cluster representative. There are number of clustering techniques in the literature, e.g. the well-known k-means algorithm (Tou and Gonzales, 1974; Berkhin, 2002; Wu et al., 2008; Berry and Linoff, 2004), fuzzy k-mean (Bezdek, 1973) and Self-Organizing Map (SOM) (Kohonen, 1990). However, two generic categories (Farley and Raftery, 1998) of the clustering methods can be defined as: hierarchical clustering (El-Hamdouchi and Willett, 1989; Steinbach et al., 2000) and partitional clustering (Forgy, 1965; Lloyd, 1982; MacQueen, 1967).

In recommender systems, the clustering techniques mainly group of users that have similar preferences. Once the clusters are recognized, predictions for an active user can be made by averaging the opinions of the other users in that cluster. Fig. 11 shows Neighborhood formation from clustered partitions. In the following, some clustering methods are introduced.

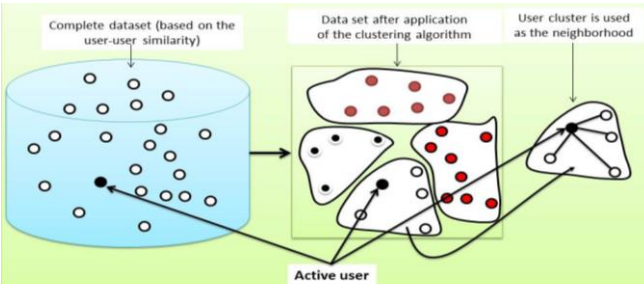


Fig. 11. Clustered partitions by neighborhood formation

***k*-means.** *k*-means clustering is one of the simplest unsupervised machine learning algorithms. It was first proposed by MacQueen (1967). Given a set of elements, *k*-means clustering algorithm is used to group or classify the elements based on some features into *k* number of clusters (Nilashi et al., 2011). According to Tou and Gonzalez (1974), the *k*-means clustering minimizes the objective:

$$\sum_{k=1}^{k} \sum_{x_i \in C_k} || x_i - c_k ||^2 \tag{31}$$

where c_k is the centroid of the k^{th} cluster. The *k*-means clustering algorithm is shown as follows:

An International Journal JSCDSS
E-ISSN: 2289-8603

Algorithm 2: K-means Clustering.

Step 0 **Initialization**: Given data set D, integer K, $2 < K < N$, **select** K initial centers $\{c_k\}$

Step 1 **Compute** the distances $d(x_i,c_k)$, $i = 1,\ldots, N$, $k = 1, \ldots, K$.

Step 2 **Partition** the dataset $D = C_1 \cup C_2 \cup \ldots \cup C_k$ by assigning each data point to the cluster whose center is the nearest

Step 3 **Re-compute** the cluster centers.

Step 4 **If** the centers have not changed, **stop**.
else go to Step 1.

Notes:

i. The initial "centers" in Step 0 are just points, and not yet associated with clusters. They can be selected randomly as any K points of D.
ii. In Step 3 the center of each cluster is computed using the points assigned to that cluster.
iii. The stopping rule in Step 4 implies that there are no further re-assignments.
iv. The center updates in the iterations are computed by

$$c_k = \frac{\sum_{i=1}^{N} u_{ik} x_i}{\sum_{i=1}^{N} u_{ik}}, \quad k = 1,\ldots,k \tag{32}$$

where $u_{ik} = 1$ if $x_i \in C_k$, and $u_{ik} = 0$ otherwise. Eq. (32) gives the centers as the geometrical centroids of the data points of the cluster.

v. Using Euclidean distances, iterating Steps 2 and 3 leads to the minimization of the objective in Eq. (31).

Self-Organized Map (SOM). The SOM is a neural network algorithm learns to classify data without supervision that depends on the clustering of the measured data from different stations (Kohonen, 1990). A SOM is trained with input data. SOM is a very valuable tool for pattern recognition and clustering process, which has been used in many varied applications (Sharpe and Caleb, 1998; Zhang and Mlynski, 1997; Witkowski et al.m1997; Lihua et al., 2005). It provides a non-linear mapping of the data to a 2D map grid that can be applied after preprocessing in explaining similarities and differences within measured data and cluster structures. Fig. 12 demonstrates an illustration of the SOM model.

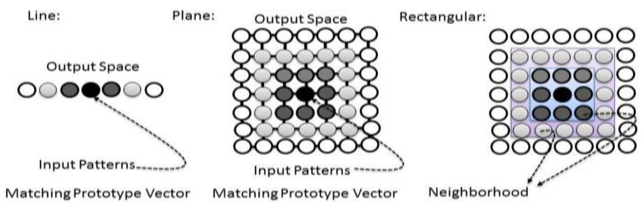


Fig. 12. Illustration of the SOM model with one, two dimensional and rectangular map

Expectation Maximization (EM) clustering. It is well known that the k-means algorithm is an instance of Expectation Maximization (EM) algorithm which is a general algorithm of density estimation. This algorithm is based on distance. Gaussian mixture model with EM algorithm is a powerful approach for clustering. EM algorithm is model based iterative algorithm for solving the clustering problem where the data is incomplete or considered incomplete. EM algorithm is an optimization algorithm for constructing statistical models of the data (Mitra et al., 2003). In this algorithm each and every data instance belongs to each and every cluster with a certain probability. EM algorithm starts with initial estimates and iterates to find the maximum likelihood estimates for the parameters. The quality of EM algorithm become very good when using huge dataset. It has been also demonstrated that EM is a good clustering method in terms of computation time and accuracy (Jung et al., 2014; Nathiya et al., 2010). In addition, in this study EM is chosen to cluster data for the following reasons among others (Ordonez and Omiecinski, 2002). (1) It has a strong statistical basis, (2) It is linear in database size, (3) It is robust to noisy data, (4) It can accept the desired number of clusters as input, (5) It can handle high dimensionality, and (6) It converges fast given a good initialization.

The mathematical background of EM algorithm is shown here in this section (Mitra et al., 2003).

Given a dataset $\{x_i\}_{i=1}^{N}$ the task of assigning a cluster for each instance in the dataset, is the goal that we aspire for. Let there be N data points in the dataset and let us assume that the number of clusters is k. Let the index of the cluster be modeled as a random variable $z = j$ and let its probability be given by a multinomial distribution satisfying $\sum \pi_j = 1$, Such that

$$\pi_j = p(z = j), \forall j, j = 1,\ldots k \tag{33}$$

It is assumed that $p(x \mid z = j) \sim N(\mu_j, \sigma_j I_j)$ is a Gaussian distribution. I_j denotes the identity matrix of order j. The unknown parameters of the model namely the mean μ_j. variance $\sum_j$ = diag $(\sigma_1, \sigma_2, \ldots, \sigma_j)$ and the distribution function π_j are estimated.

$$\theta = \{\mu_j, \sum_j, \pi_j\}_{j=1}^{k} \quad p(x \mid \theta) = \sum_{z=1}^{k} (p(x \mid z, \theta) p(z \mid \theta) \pi_j, \tag{34}$$

where z is an unknown hidden variable. The total log likelihood of all data is given by

$$l(\theta, D) = \log \prod_{i=1}^{N} \sum_{j=1}^{k} \pi_j \exp\left[-\frac{\| x_i - \mu_j \|^2}{2\sigma_i^2} \right] \tag{35}$$

The parameter values that maximize the likelihood function $l(\theta, D)$ are the ones that are chosen. Here D denotes the data. This optimization is complicated and to

An International Journal JSCDSS
E-ISSN: 2289-8603

solve this some of the unknowns are assumed to be known, while estimating the others and vice versa. For each class, the conditional expectation of $z = j$ given the data and the parameters

$$w_j = p(z = j \mid x, \theta) = \frac{p(x \mid z = j, \theta)\, p(z = j \mid \pi_j)}{p(x \mid \theta)} = \frac{\pi_j N(x_i \mid \mu_j, \sum_j)}{\sum_{i=1}^{k} \pi_j N(x_i \mid \mu_j, \sum_j)} \quad (36)$$

Since each point x contributes to w_j in some proportion, for particular x_i we have

$$w_{ij} = \frac{\pi_j N(x_i \mid \mu_j, \sum_j)}{\sum_{i=1}^{k} \pi_j N(x_i \mid \mu_j, \sum_j)}. \quad (37)$$

The optimization algorithm is called EM and has the following steps: Assume we have some random initial estimates of the means and variances of the model $\mu_j^{(0)}, \sum_j^{(0)}, \pi_j^{(0)}$. Algorithm 3 describes the EM algorithm.

Algorithm 3 EM Algorithm.

Initialize: means and variances of the model $\mu_j^{(0)}, \sum_j^{(0)}, \pi_j^{(0)}$.

Step 1. Expectation: Using the estimates of $\theta^{(t)} = \{\mu_j^{(t)}, \sum_j^{(t)}, \pi_j^{(t)}\}$, parameters compute the estimate of w_{ij}

$$w_{ij}^{(i)} = p(z = j \mid x_i, \theta^{(t)}) = \frac{\pi_j^i\, p(x_i \mid z_i = j, \theta)}{\sum_{m=1}^{k} \pi_m^t\, p(x_m \mid z_m = m, \theta^{(t)})}$$

Step 2. Maximization: Using estimates of $w_{ij}^{(i)}$, update the estimates of the model parameters

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} w_{ij}^{(t)} x_i}{\sum_{i=1}^{N} w_{ij}^{(t)}}$$

$$\sigma_j^{(t+1)} = \frac{\sum_{i=1}^{N} w_{ij}^{(t)} \| x_i - \mu_i \|^2}{\sum_{i=1}^{N} w_{ij}^{(t)}}$$

$$\pi_i^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} w_{ij}^{(t)}$$

Step 3. Repeat steps expectation and maximization until the parameter change gets small enough.

Fuzzy K-means Clustering. A clustering is hard (or crisp) if each data point x is assigned to one, and only one, cluster C, so that the statement $x \in C$ is unambiguous. A point x is labeled if its cluster C is known, in which case C is the label of x. In soft (or fuzzy) clustering the rigid assignment $x \in C$ is replaced by a cluster Membership Function (MF) $u(x,C)$ representing the belief that x belongs to C (Hammouda and Karray, 2000; Panda et al., 2012). The numbers $u(x,C_k)$ are often taken as probabilities that x belongs to C_k, so that

$$\sum_{k=1}^{k} u(x,C_k) = 1, \text{ and } u(x,C_k) \geq 0 \quad \text{for all} \quad k = 1,\ldots,K. \quad (38)$$

The k-means algorithm can be adapted to soft clustering. A well-known center-based algorithm for soft clustering is the fuzzy k-means algorithm. The objective function minimized in this algorithm is:

$$f = \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ik}^m d_{ik}^2 = \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ik}^m \|x_i - v_k\|^2 \quad (39)$$

where u_{ik} are the MFs of $x_i \in C_k$, and typically satisfy Eq. (39), and m is a real number, $m > 1$, known as fuzzifier. The equation for finding the centers is similar to equation of k-means algorithm, but u_{ik} takes values between 0 and 1 (see Eq. (40)).

$$c_k = \frac{\sum_{i=1}^{N} u_{ik}^m x_i}{\sum_{i=1}^{N} u_{ik}^m}, \quad k = 1,\ldots,k. \quad (40)$$

When m tends to 1, the algorithm converges to the k-means method.

Clustering methods also have been successful in recommendation system implementation (Ghazanfar and Prügel-Bennett, 2014; Shepitsen et al., 2008; Xue et al., 2005; Lee et al., 2002; Roh et al., 2003). For example, Nilashi et al. (2014a) proposed recommendation methods using Adaptive Neuro-Fuzzy Inference Systems (ANFIS) and SOM proposed to improve predictive accuracy of criteria CF. Furthermore, new fuzzy-based algorithms, Weighted Fuzzy MC-CF (WFuMC-CF), Fuzzy Euclidean MC-CF (FuEucMC-CF) and Fuzzy Average MC-CF (FuAvgMC-CF), were presented for prediction task in multi-criteria CF. Experimental results on real-world dataset demonstrated that the proposed hybrid methods remarkably improve the accuracy of multi-criteria CF in relation to the previous methods based on multi-criteria ratings. Nilashi et al. (2015b) have also developed a multi-criteria CF recommender system for hotel recommendation to enhance the predictive accuracy using Gaussian mixture model with Expectation Maximization (EM) algorithm and Adaptive Neuro-Fuzzy Inference System (ANFIS). They used the Principal Component Analysis (PCA) for dimensionality reduction and to address multi-collinearity induced from the interdependencies among criteria in multi-criteria CF dataset. The list of research papers conducted in the context of recommender systems using unsupervised machine learning techniques is represented in Table 8.

An International Journal JSCDSS
E-ISSN: 2289-8603

Table 8
Clustering Technique in Recommender Systems

Data mining Technique	Reference
Clustering	Cantador and Castells (2010); Kim and Ahn (2008); Choi, Kang, and Jeon (2006); Linden (2008); Lee and Park (2007); Ha (2006); Liu, Hsieh, and Tsai (2010); Li, Myaeng, and Kim (2007); Zhu, Shi, Kim, and Eom (2006); Kwon, Cho, and Park (2009); Merve and Arslan (2009); Symeonidis, Nanopoulos, and Manolopoulos (2008); Min and Han (2005); Weng and Liu (2004); Roh, Oh, and Han (2003); Jalali, Mustapha, Sulaiman, and Mamat (2010); Lai and Liu (2009); Wei, Yang, and Hsiao (2008); Lihua et al. (2005); Rosaci, Sarne, and Garruzzo (2009); Linden, Smith, and York (2003); Subhash and Uday (2012); Kim and Ahn (2005); Agarwal et al.(2005); Martin-Guerrero, Lisboa, Soria-Olivas, Palomares, and Balaguer (2007); Cheung, Tsui, and Liu (2004); Kim and Yang(2005); Georgiou and Tsapatsoulis (2010); Honda et al. (2001); Treerattanapitak and Jaruskulchai (2010); Chau, Zeng, Chen, Huang, and Hendriawan (2003); Uchyigit and Clark (2004); Nilashi et al. (2015a); Nilashi et al. (2014a); Nilashi et al. (2014b); Nilashi et al. (2015b); Nilashi et al. (2014c)

2.5. Supervised Prediction Methods

2.5.1. Neuro-Fuzzy

Fuzzy Logic (FL) and Fuzzy Inference Systems (FIS), first proposed by Zadeh (1965), provide a solution for making decisions based on vague, ambiguous, imprecise or missing data (Nilashi et al., 2016). FL represents models or knowledge using IF–THEN rules (Nilashi et al., 2015d; Ahmadi et al., 2015; Salahshour et al., 2015; Ahmadi et al., 2014a; Ahmadi et al., 2014b; Nilashi and Ibrahim, 2014; Bagherifard et al., 2014; Nilashi and Janahmadi, 2012; Nilashi et al., 2011a; Nilashi et al., 2011b; Nilashi et al., 2011c; Nilashi et al., 2011d; Akbari et al., 2015)

An appropriate combination of neural network and fuzzy logic technologies (Neuro-Fuzzy) can effectively solve the problems of fuzzy logic and neural networks and, thus, can more effectively address the real problems. A Neuro-Fuzzy approach was used to take advantage of the neural network’s ability to learn, and membership degrees and functions of fuzzy logic. The weights of the neural networks are mapped to fuzzy logic rules and member functions. Expressing the weights of the neural network by fuzzy rules also provides a better understanding of the “Black Box” and thus helps the better design of the neural network itself. Thus, while the learning of neural network is parameterized by the variation in input data, the learning of ANFIS is fixed by the rules and membership function values that we define. A Neuro-Fuzzy system is functionally equivalent to a FIS. A FIS mimics a human reasoning process by implementing fuzzy sets and approximate reasoning mechanism that uses numerical values instead of logical values. A FIS requires a domain expert to define the MFs and to determine the associated parameters in both the MFs, and the reasoning section. However, there is no standard for the knowledge acquisition process and thus the results may be different if a different knowledge engineer is at work in acquiring the knowledge from experts.

A Neuro-Fuzzy system can replace the knowledge acquisition process by humans using a training process with a set of input-output training dataset. Thus instead of dependent on human experts the Neuro-Fuzzy system will determine the parameters associated with the Neuro-Fuzzy system through a training process, by minimizing an error criterion. A popular Neuro-Fuzzy system is called an ANFIS. ANFIS is a fuzzy system that uses artificial neural network theory to determine its properties (fuzzy sets and fuzzy rules).

The fuzzy logic field has grown considerably in a number of applications across a wide variety of domains like in the semantic music recommendation system (Lesaffre and Leman 2007), movie recommendation (Nilashi et al., 2014a) and product recommendations (Cao and Li 2007; Henrik Stormer et al. 2006). Castellano et al. (2007) developed a Neuro-Fuzzy strategy combined with soft computing approaches for recommending URLs to the active users. They used fuzzy clustering for creating a user profile considering the similar browsing behavior. de Campos et al. (2008) proposed a model by combining Bayesian network for governing the relationships between the users and fuzzy set theory for presenting the vagueness in the description of users‘ ratings. A conceptual framework based on fuzzy logic-based was proposed by Yager (2003) to represent and then justify the recommendation rules. In the proposed framework, an internal description of the items was used that relied solely on the preferences of the active user. Carbo and Molina (2004) developed an algorithm based on CF that ratings and recommendations were considered as linguistic labels by using fuzzy sets. A model proposed by Pinto et al. (2012) that combined fuzzy numbers, product positioning (from marketing theory) and item-based CF. Nilashi et al. (2014a) proposed recommendation methods using Adaptive ANFIS and SOM proposed to improve predictive accuracy of criteria CF. Furthermore, new fuzzy-based algorithms, Weighted Fuzzy MC-CF (WFuMC-CF), Fuzzy Euclidean MC-CF (FuEucMC-CF) and Fuzzy Average MC-CF (FuAvgMC-CF), were presented for prediction task in multi-criteria CF. Experimental results on real-world dataset demonstrated that the proposed hybrid methods remarkably improve the accuracy of multi-criteria CF in relation to the previous methods based on multi-criteria ratings.

Nilashi et al. (2015b) developed a multi-criteria CF recommender system for hotel recommendation to enhance the predictive accuracy using Gaussian mixture model with EM algorithm and Adaptive ANFIS. They used the PCA for dimensionality reduction and to address multi-collinearity induced from the interdependencies among criteria in multi-criteria CF dataset.

An International Journal
JSCDSS
E-ISSN: 2289-8603

2.5.1 Support Vector Regression (SVR)

As a powerful machine learning technique, SVM is becoming increasingly popular. SVR is able to model complex non-linear relationships by using an appropriate kernel function that maps the input matrix X onto a higher-dimensional feature space and transforms the non-linear relationships into linear forms. The feature space is then used as a new input to deal with the regression problem. By introducing an ε-insensitive loss function, Vapnik extended SVM for classification to regression (Vapnik et al. 1996). In this sense, SVR transforms the regression problem into a special classification problem. Moreover, like the support vector classifier, the SVR uses soft margins to tolerate misclassification. Finally, SVR uses a tactic named ε-insensitive loss function to balance the approximate accuracy and computational complexity.

SVR also have been successful in recommendation system implementation. Jannach et al. (2012a) improved the accuracy of multi-criteria CF by proposing a method using SVR for automatically detecting the existing relationships between detailed item ratings and the overall ratings. In addition, the learning process of SV regression models was per item and user and lastly combined the individual predictions in a weighted approach. Jannach et al. (2012b) showed through an empirical evaluation based on a real-world data set from the tourism domain that the predictive accuracy of recommender systems can be significantly improved when the multi-dimensional rating information is taken into account. They used SVR to construct the prediction models. They evaluated the method on a real-world datasets provided by a major European tourism platform and Yahoo!Movies and compare it with state-of-the-art baseline algorithms based on matrix factorization. Nilashi et al. (2015a) proposed a novel CF recommendation approach in which customer segments are automatically detected through clustering and preference models are learned for each customer segment. Their proposed method also supports incremental updates of the preference models.

3. Conclusion

Recommender systems have become an important and interesting research area in e-commerce. The results represented in this research have several significant implications:

- This paper introduced the most popular data mining methods and techniques that can be implemented in the design of recommender systems.
- Data preprocessing tools such as sampling, dimensional reduction and distance measures methods were investigated.
- We reviewed the main classification (supervised learning) methods that can be used in the in the design of recommender systems namely: *k*-Nearest Neighbors, Bayesian Networks, Neural Networks, Support Vector Machines and Decision Trees.
- We reviewed the clustering (unsupervised learning) methods that can be implemented in the design of recommender systems namely: Expectation Maximization, *k*-means, Fuzzy *k*-means and Self-Organized Map.
- We reviewed the prediction methods that can be implemented in the design of recommender systems namely: Support Vector Regression and Neuro-Fuzzy.
- Some important dimensional reduction techniques such as Singular Value Decomposition (SVD) and Higher-order Singular Value Decomposition (HOSVD) were described and their applications in the context of recommender systems were investigated.

In this study, we introduced well-known data mining techniques and presented their applications in the context of recommender systems, but in-depth investigation is need for their applications in overcoming the shortcomings of recommender systems. Future works could therefore focus on shortcomings of recommender systems such as Sparisity and Scalability and the applications of the data mining techniques in overcoming these shortcomings.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6), 734-749.

Agarwal, N., Haque, E., Liu, H., & Parsons, L. (2005). Research paper recommender systems: A subspace clustering approach. In Advances in Web-Age Information Management (pp. 475-491). Springer Berlin Heidelberg.

Ahmadi, H., Darvishi, M., Nilashi, M., Almaee, A., Ibrahim, O., Zolghadri, A. H., & Farahmand, M. (2014a). Evaluating the Critical Factors for Electronic Medical Record Adoption Using Fuzzy Approaches. International Journal of Innovation and Scientific Research, 9(2), 268-284.

Ahmadi, H., Nilashi, M., & Ibrahim, O. (2015). Prioritizing Critical Factors to Successful Adoption of Total Hospital Information System. Journal of Soft Computing and Decision Support Systems, 2(4), 6-16.

Ahmadi, H., Rad, M. S., Almaee, A., Nilashi, M., Ibrahim, O., Dahlan, H. M., & Zakaria, R. (2014b). Ranking the macro-level critical success factors of electronic medical record adoption using fuzzy AHP method. International Journal of Innovation and Scientific Research, 8(1), 35-42.

Ahn, H. J., Kang, H., & Lee, J. (2010). Selecting a small number of products for effective user profiling in collaborative filtering. Expert Systems with Applications, 37(4), 3055–3062

Akbari, E., Buntat, Z., Shahraki, E., Zeinalinezhad, A., & Nilashi, M. (2015). ANFIS modeling for bacteria detection based on GNR biosensor. Journal of Chemical Technology and Biotechnology.

Albadvi, A., & Shahbazi, M. (2009). A hybrid recommendation technique based on product category attributes. Expert Systems with Applications, 36, 11480–11488.

Alpaydin, E. Introduction to Machine Learning. Cambridge, Massachusetts: MIT Press. 2004.

Application of dimensionality reduction in recommender system

An International Journal JSCDSS
E-ISSN: 2289-8603

Ariyoshi, Y. and J. Kamahara. A hybrid recommendation method with double SVD reduction. in Database Systems for Advanced Applications. 2010. Springer.

Bagherifard, K. B., Tafreshi, F. S., Nilashi, M., & Jalalyazdi, M. (2014). Assessing the critical factors for e-learning systems using fuzzy TOPSIS and fuzzy logic. International Journal Of Computers & Technology, 12(6), 3546-3561.

Bagherifard, K., Nilashi, M, Ibrahim, O, Janahmadi, N and Ebrahimi, Leila (2012). Comparative Study of Artificial Neural Network and ARIMA Models in Predicting Exchange Rate. Research Journal of Applied Sciences, Engineering and Technology, 4(21): 4397-4403.

Bagherifard, K., Nilashi, M., Ibrahim, O., Ithnin, N., & Nojeem, L. A. (2013). Measuring semantic similarity in grids using ontology., Volume 2, Issue 3, March 2013, Pages 230–237.

Baroni-Urbani, C., Buser, M.W., (1976), “Similarity of Binary Data”, Systematic Zoology, Vol. 25, No. 3, pp. 251-259.

Basu, C., Hirsh, H., and Cohen, W., Recommendation as classification: Using social and content-based information in recommendation. In In Proceedings of the Fifteenth National Conference on Artificial Intelligence, pages 714–720. AAAI Press, 1998.

Bell, A and Sejnowski ,TJ. An information-maximization approach to blind separation and blind deconvolution. Neural Comput 1995;7: 1129–59.

Bennett, K. P., & Mangasarian, O. L. (1992). Robust linear programming discrimination of two linearly inseparable sets. Optimization methods and software, 1(1), 23-34.

Bennett, K.P., Campbell, C.: Support Vector Machines: Hype or Hellelujah?, ACM SIGKDD Explorations 2(2) (2000) 1-13.

Berkhin, P. (2006). A survey of clustering data mining techniques. In Grouping multidimensional data (pp. 25-71). Springer Berlin Heidelberg.

Berry, M. J. A., & Linoff, J. S. (2004). Data Mining Techniques for Marketing. Sales and Customer Relationship Management (2nd ed.,). Wiley.

Bezdek, J. C. (1973). Fuzzy mathematics in pattern classification. PhD-thesis,Applied Math Center, Cornell University, Ithaca.

Billsus, D., & Pazzani, M. J. (1998, July). Learning Collaborative Information Filters. In ICML (Vol. 98, pp. 46-54).

Billsus, D., and Pazzani, M. Learning Collaborative Information Filters. In Proceedings of the 15th International Conference on Machine Learning (ICML 98) (Madison, Wisconsin, USA, July 1998), Morgan Kaufmann Publishers, pp. 46 – 54.

Blanco-Fernandez, Y., Lopez-Nores, M., Pazos-Arias, J. J., Gil-Solla, A., & Ramos-Cabrer, M. (2010). Exploiting digital TV users’ preferences in a tourism recommender system based on semantic reasoning. IEEE Transactions on Consumer Electronics, 56, 904–912.

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems, 23, 520–528.

Bobadilla, J., Serradilla, F., & Hernando, A. (2009). Collaborative filtering adapted to recommender systems of e-learning. Knowledge-Based Systems, 22, 261–265.

Bouza, A., Reif, G., Bernstein, A., and Gall, H., Semtree: ontology-based decision tree algorithm for recommender systems. In International Semantic Web Conference, 2008.

Bradley, PS. Mangasarian, OL., Musicant,D. Optimization methods in massive datasets. In: Abello J, Pardalos PM, Resende MGC, editors. Handbook of massive datasets. Boston: Kluwer Academic Publishers; 2002. p. 439–71.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43-52.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.

Canny, J.F. Collaborative filtering with privacy via factor analysis. In SIGIR, pages 238–245. ACM, 2002.

Cantador, I., & Castells, P. (2010). Extracting multilayered communities of interest from semantic user profiles: Application to group modeling and hybrid recommendations. Computers in Human Behavior, 27, 1321–1336.

Cao, Y., & Li, Y. (2007). An intelligent fuzzy-based recommendation system for consumer electronic products. Expert Systems with Applications, 33(1), 230-240.

Capobianco, M., & Frank, O. (1982). Comparison of statistical graph-size estimators. Journal of Statistical Planning and Inference, 6(1), 87-97.

Carbo, J., & Molina, J. M. (2004). Agent-based collaborative filtering based on fuzzy recommendations. International journal of Web engineering and technology, 1(4), 414-426.

Castellano, G., Fanelli, A. M., & Torsello, M. A. (2007). A neuro-fuzzy collaborative filtering approach for web recommendation. International Journal of Computational Science, 1(1), 27-29.

Castelli, V., Thomasian, A., and Li, C.-S. CSVD: Clustering and singular value decomposition for approximate similarity search in high-dimensional spaces. IEEE Trans. Knowledge and Data Engin., 15(3):671–685, 2003.

Cha, S.-H. and Tappert, C.C., Enhancing Binary Feature Vector Similarity Measures, in Journal of Pattern Recognition Research (JPRR), Vol 1 No 1, 2006, pp 63-77.

Cha, S.-H., Tappert, C.C., (2003), “Optimizing Binary Feature Vector Similarity Measure using Genetic Algorithm”, ICDAR, Edinburgh, Scotland.

Chang, C., Lin, C.: LIBSVM: a library for support vector machines , 2006.

Chau, M., Zeng, D., Chen, H., Huang, M., & Hendriawan, D. (2003). Design and evaluation of a multi-agent collaborative Web mining system. Decision Support Systems, 25, 167–183.

Chen, B., Haas, P., & Scheuermann, P. (2002, July). A new two-phase sampling based algorithm for discovering association rules. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 462-468). ACM.

Chen, G., Wang, F., & Zhang, C. (2009). Collaborative filtering using orthogonal nonnegative matrix tri-factorization. Information Processing & Management, 45,368–379.

Chen, L., & Pu, P. (2010). Experiments on the preference-based organization interface in recommender systems. ACM Transaction on Computer-Human Interaction, 17, 1–33.

Chen, Y. L., Cheng, L. C., & Chuang, C. N. (2008). A group recommendation system with consideration of interactions among group members. Expert Systems with Applications, 34, 2082–2090.

Cheng, W., J. H¨uhn, and E. H¨ullermeier. Decision tree and instance-based learning for label ranking. In ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning, pages 161–168, New York, NY, USA, 2009. ACM.

Cheung, K. W., Tsui, K. C., & Liu, J. (2004). Extended latent class models for collaborative recommendation. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 34, 143–148.

An International Journal JSCDSS
E-ISSN: 2289-8603

Cho, Y. H., & Kim, J. K. (2004). Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. Expert Systems with Applications, 26, 233–246.

Cho, Y. H., Kim, J. K., & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. Expert Systems with Applications, 23, 329–342.

Choi, S. H., Kang, S. M., & Jeon, Y. J. (2006). Personalized recommendation system based on product specification values. Expert Systems with Applications, 31, 607–616.

Choi, S.-S, (2008), "Correlation Analysis of Binary Similarity Measures and Dissimilarity Measures", Doctorate dissertation, Pace University.

Christakou C, Vrettos S, Stafylopatis A (2007) A hybrid movie recommender system based on neural networks. International Journal on Artificial Intelligence Tools 16 (05):771-792

Claypool ,M., Gokhale,A., Mir,T., Pavel Murnikov, Netes,D., and Sartin,M., Combining content-based and collaborative filters in an online newspaper, In Proceedings of ACM SIGIR Workshopon Recommender Systems, 1999.

Cohen, W., Fast effective rule induction. In Machine Learning: Proceedings of the 12th International Conference, 1995.

Cohen, W., Ravikumar, P., Fienberg, S. (2003). A comparison of string distance metrics for name-matching tasks. In Proc. IJCAI-03 Workshop on Information Integration on the Web.

Comon P. Independent component analysis — a new concept? Signal Processing 1994;36:287–314.

Condliff, M. K., Lewis, D. D., Madigan, D., & Posse, C. (1999, August). Bayesian mixed-effects models for recommender systems. In ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation (Vol. 15, No. 5).

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. Information Theory, IEEE Transactions on, 13(1), 21-27.

Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines. Cambridge, UK: Cambridge University Press.

Dasarathy, B. V. (1991). Nearest neighbor ({NN}) norms:{NN} pattern classification techniques. Los Alamitos, CA: IEEE Computer Society Press

de Campos, L. M., Fernández-Luna, J. M., & Huete, J. F. (2008). A collaborative recommender system based on probabilistic inference from fuzzy observations. Fuzzy Sets and Systems, 159(12), 1554-1576.

De Lathauwer, L., & Vandewalle, J. (2004). Dimensionality reduction in higher-order signal processing and rank-(R 1, R 2,…, R N) reduction in multilinear algebra. Linear Algebra and its Applications, 391, 31-55.

De Lathauwer, L., De Moor, B., & Vandewalle, J. (2000). A multilinear singular value decomposition. SIAM journal on Matrix Analysis and Applications, 21(4), 1253-1278.

Deza, M. M., & Deza, E. (2009). Encyclopedia of distances: Springer.

Dietterich, T.G., Hild ,H., Bakiri ,G. A comparison of ID3 and backpropagation for english text-to-speech mapping, Machine Learning 18 (1995) 51– 80.

Doumpos, M., Chatzi, E., & Zopounidis, C. (2006). An experimental evaluation of some classification methods. Journal of Global Optimization, 36(1), 33-50.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification. John Wiley & Sons.

El-Hamdouchi, A., & Willett, P. (1989). Comparison of hierarchic agglomerative clustering methods for document retrieval. The Computer Journal, 32(3), 220-227.

Erd6s, P., & Rényi, A. (1960). On the evolution of random graphs. Publ. Math. Inst. Hungar. Acad. Sci, 5, 17-61.

ERDdS, P., & R&WI, A. (1959). On random graphs I. Publ. Math. Debrecen, 6, 290-297.

Farahmand, M., Desa, M. I., & Nilashi, M. (2014a). A Combined Data Envelopment Analysis and Support Vector Regression for Efficiency Evaluation of Large Decision Making Units. International Journal of Engineering and Technology (IJET), 2310-2321.

Farahmand, M., Desa, M. I., & Nilashi, M. (2015). A Comparative Study of CCR-(ε-SVR) and CCR-(ν-SVR) Models for Efficiency Prediction of Large Decision Making Units. Journal of Soft Computing and Decision Support Systems, 2(1), 8-17.

Farahmand, M., Desa, M. I., Nilashi, M., & Wibowo, A. (2014b). An Improved Method for Predicting and Ranking Suppliers Efficiency Using Data Envelopment Analysis. Jurnal Teknologi, 73(2).

Farokhi, N., Vahid, M., Nilashi, M., & Ibrahim, O. (2016). A Multi-Criteria Recommender System for Tourism Using Fuzzy Approach. Journal of Soft Computing and Decision Support Systems, 3(4), 19-29.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. Annals of eugenics, 7(2), 179-188.

Fukunaga, K. (1990). Introduction to statistical pattern recognition. Academic press. San Diego, CA, USA

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 16(10), 906-914.

Ganesan, P., Garcia-Molina, H., & Widom, J. (2003). Exploiting hierarchical domain structure to compute similarity. ACM Transactions on Information Systems, 21,64–93.

Gao, J., & Zhang, J. (2005). Clustered SVD strategies in latent semantic indexing. Information processing & management, 41(5), 1051-1063.

Gao, M., & Wu, Z. (2009). Personalized context-aware collaborative filtering based on neural network and slope one. In Cooperative Design, Visualization, and Engineering (pp. 109-116). Springer Berlin Heidelberg.

Gemmell, J., Schimoler, T., Ramezani, M., & Mobasher, B. (2009). Adapting K-Nearest Neighbor for Tag Recommendation in Folksonomies. ITWP, 528.

Georgiou,O. and Tsapatsoulis, N. Improving the Scalability of Recommender Systems by Clustering Using Genetic Algorithms ,Lecture Notes in Computer Science, 2010, Volume 6352, Artificial Neural Networks – ICANN 2010, Springer, Pages 442-449

Ghazanfar, M. A., & Prügel-Bennett, A. (2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. Expert Systems with Applications, 41(7), 3261-3275.

Ghazanfar, M., & Prugel-Bennett, A. (2010). An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering.

Glady, N., Baesens, B., & Croux, C. (2009). Modeling churn using customer lifetime value. European Journal of Operational Research, 197(1), 402-411.

Golbandi, N., Koren, Y., & Lempel, R. (2011, February). Adaptive bootstrapping of recommender systems using decision trees. In Proceedings of the fourth ACM international conference on Web search and data mining (pp. 595-604). ACM.

Gong, S., & Ye, H. (2009, April). An item based collaborative filtering using bp neural networks prediction. In Industrial and

Information Systems, 2009. IIS'09. International Conference on (pp. 146-148). IEEE.
Grčar, M., Fortuna, B., Mladenič, D., & Grobelnik, M. (2006). kNN versus SVM in the collaborative filtering framework. Data Science and Classification, 251-260.
Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine learning, 46(1-3), 389-422.
Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (2006). Multivariate data analysis (Vol. 6). Upper Saddle River, NJ: Pearson Prentice Hall.
Han, J., Altman, R. B., Kumar, V., Mannila, H., & Pregibon, D. (2002). Emerging scientific applications in data mining. Communications of the ACM, 45(8), 54-58.
Han, L., & Chen, G. (2009). HQE: A hybrid method for query expansion. Expert Systems with Applications, 36, 7985–7991.
Han, P., Xie, B., Yang, F., & Shen, R. (2004). A scalable P2P recommender system based on distributed collaborative filtering. Expert Systems with Applications, 27, 203–210.
Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer, 27(2), 83-85.
Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 230-237). ACM.
Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53.
Hernandez del Olmo, F., Gaudioso, E., & Martin, E. H. (2009). The task of guiding in adaptive recommender systems. Expert Systems with Applications, 36,1972–1977.
Hijikata, Y., Iwahama, K., & Nishida, S. (2006, April). Content-based music filtering system with editable user profile. In Proceedings of the 2006 ACM symposium on Applied computing (pp. 1050-1057). ACM.
Honda, K., Sugiura, N., Ichihashi, H., & Araki, S. (2001). Collaborative filtering using principal component analysis and fuzzy clustering. In Web Intelligence: Research and Development (pp. 394-402). Springer Berlin Heidelberg.
Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. Journal of educational psychology, 24(6), 417.
Hsu, M. H. (2008). A personalized English learning recommender system for ESL students. Expert Systems with Applications, 34, 683–688.
Hsu, S. H., Wen, M. H., Lin, H. C., Lee, C. C., & Lee, C. H. (2007). AIMED-A personalized TV recommendation system. In Interactive TV: a Shared Experience (pp. 166-174). Springer Berlin Heidelberg.
Hubalek, Z., (1982), "Coefficients of Association and Similarity, Based on Binary (Presence-Absence) Data: An Evaluation", Biological Reviews, Vol.57-4,669-689.
Hurley, N. J., O'Mahony, M. P., & Silvestre, G. C. M. (2007). Attacking recommender systems: a cost-benefit analysis. IEEE Intelligent Systems, 22, 64–68.
Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. Neural Networks, IEEE Transactions on, 10(3), 626-634.
Ibnkahla, M. (2000). Applications of neural networks to digital communications–a survey. Signal processing, 80(7), 1185-1215.
Igelnik, B., & Pao, Y. H. (1995, November). Estimation of size of hidden layer on basis of bound of generalization error. In Neural Networks, 1995. Proceedings., IEEE International Conference on (Vol. 4, pp. 1923-1927). IEEE.
Jackson, D. A., Somers, K. M., & Harvey, H. H. (1989). Similarity coefficients: measures of co-occurrence and association or simply measures of occurrence?. American Naturalist, 436-453.
Jalali, M., Mustapha, N., Sulaiman, M. N., & Mamat, A. (2010). WebPUM: A Web-based recommendation system to predict user future movements. Expert Systems with Applications, 37(9), 6201-6212.
Jannach, D., Gedikli, F., Karakaya, Z., & Juwig, O. (2012b). Recommending hotels based on multi-dimensional customer ratings. In International Conference on Information and Communication Technologies in Tourism (ENTER) (pp. 320-331).
Jannach, D., Karakaya, Z., & Gedikli, F. (2012a). Accuracy improvements for multi-criteria recommender systems. In Proceedings of the 13th ACM Conference on Electronic Commerce (pp. 674-689). ACM.
Jeong, B., Lee, J. W., & Cho, H. B. (2009a). User credit-based collaborative filtering. Expert Systems with Applications, 36, 7309–7312.
Jeong, B., Lee, J. W., & Cho, H. B. (2009b). An iterative semi-explicit rating method for building collaborative recommender systems. Expert Systems with Applications, 36, 6181–6186
Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features (pp. 137-142). Springer Berlin Heidelberg.
John, G. H., & Langley, P. (1996, August). Static Versus Dynamic Sampling for Data Mining. In KDD (Vol. 96, pp. 367-370).
Karypis, G. Experimental Evaluation of Item-based Top-N Recommendation Algorithms. In Proceedings of 10th International Conference on Information and Knowledge Management (ACM CIKM 2001) (Atlanta, Georgia, USA, November 2001), ACM, pp. 247–254.
Kiang, M. Y., Chi, R. T., & Tam, K. Y. (1993). DKAS: a distributed knowledge acquisition system in a DSS. Journal of Management Information Systems, 59-82.
Kim, H. K., Kim, J. K., & Ryu, Y. U. (2009). Personalized recommendation over a customer network for ubiquitous shopping. IEEE Transactions on Services Computing, 2, 140–151.
Kim, J. K., Cho, Y. H., Kim, W. J., Kim, J. R., & Suh, J. H. (2002). A personalized recommendation procedure for internet shopping support. Electronic Commerce Research and Applications, 1, 301–313.
Kim, J. K., Kim, H. K., & Cho, Y. H. (2008). A user-oriented contents recommendation system in peer-to-peer architecture. Expert Systems with Applications, 34,300–312.
Kim, K. J., & Ahn, H. (2005). Using a clustering genetic algorithm to support customer segmentation for personalized recommender systems. In Artificial Intelligence and Simulation (pp. 409-415). Springer Berlin Heidelberg.
Kim, K. J., & Ahn, H. C. (2008). A recommender system using GA K-means clustering in an online shopping market. Expert Systems with Applications, 34, 1200–1209.
Kim, M.-W., Kim, E. J., and Ryu ,J. W. A collaborative recommendation based on neural networks, in Database Systems for Advanced Applications, Lecture Notes in Computer Science, Y.-J. Lee, J. Li, K.-Y. Whang, and D. Lee, Eds., Vol. 2973, Springer, pp. 425–430, 2004.
Kim, T. H., & Yang, S. B. (2005). An effective recommendation algorithm for clustering-based recommender systems. In AI 2005: Advances in Artificial Intelligence (pp. 1150-1153). Springer Berlin Heidelberg.

Kim, Y. S., Yum, B. J., Song, J. H., & Kim, S. M. (2005). Development of a recommender system based on navigational and behavioral patterns of customers in ecommerce sites. Expert Systems with Applications, 28, 381–393.

Kogel, W., Faster Training of Neural Networks for Recommender Systems. Boston College. , Master Thesis, 2002.

Koh J, Suk M, Bhandarkar SM (1995) A multilayer selforganizing feature map for range image segmentation. Neural Netw 8(1):67–86

Kohonen T (1990) The Self-Organizing Map. Proceedings of the IEEE, 78, pp. 1464-1480.

Kohonen, T. The self-organizing map. Neurocomputing 21, 1–6 (1998) .

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. SIAM review, 51(3), 455-500.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. SIAM review, 51(3), 455-500.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. GroupLens: Applying Collaborative Filtering to Usenet News. Communications of ACM 40, 3 (March 1997), 77 – 87.

Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434,New York, NY, USA, 2008. ACM.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42, 30–37.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, (8), 30-37.

Kwon, K. S., Cho, J. H., & Park, Y. T. (2009). Multidimensional credibility model for neighbor selection in collaborative recommendation. Expert Systems with Applications, 36, 7114–7122.

Lai, C. H., & Liu, D. R. (2009). Integrating knowledge flow mining and collaborative filtering to support document recommendation. Journal of Systems and Software, 82(12), 2023-2037.

Lee PY, Hui SC, Fong ACM (2002) Neural networks for web content filtering. Intelligent Systems, IEEE 17 (5):48-57

Lee, D. and Seiing, H. Algorithms for non-negative matrix factorization. NIPS 2000, 2000.

Lee, H. J., & Park, S. J. (2007). MONERS: A news recommender for the mobile web. Expert Systems with Applications, 32, 143–150.

Lee, H. Y., Ahn, H. C., & Han, I. G. (2007). VCR: Virtual community recommender using the technology acceptance model and the user's needs type. Expert Systems with Applications, 33, 984–995.

Lee, J. S., & Olafsson, S. (2009). Two-way cooperative prediction for collaborative filtering recommendations. Expert Systems with Applications, 36, 5353–5361.

Lee, M., Choi, P., and Woo, Y. (2002). A Hybrid Recommender System Combining Collaborative Filtering with Neural Network. In AH '02: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pages 531–534, London, UK. Springer-Verlag.

Lee, P. Y., Hui, S. C., & Fong, A. C. M. (2002). Neural networks for web content filtering. IEEE Intelligent Systems, 17, 48–57.

Lee, T. Q., Park, Y., & Park, Y. T. (2009). An empirical study on effectiveness of temporal information as implicit ratings. Expert Systems with Applications, 36, 1315–1321.

Lee, W. P., & Yang, T. H. (2003). Personalizing information appliances: a multi-agent framework for TV program recommendations. Expert Systems with Applications, 25, 331–341.

Lee, Y. J., & Mangasarian, O. L. (2001). SSVM: A smooth support vector machine for classification. Computational optimization and Applications, 20(1), 5-22.

Leginus, M., Dolog, P., Lage, R., & Durao, F. (2012). Methodologies for improved tag cloud generation with clustering. In Web Engineering (pp. 61-75). Springer Berlin Heidelberg.

Lesaffre M, Leman M (2007) Using Fuzzy Logic to Handle the Users' Semantic Descriptions in a Music Retrieval System. In: Theoretical Advances and Applications of Fuzzy Logic and Soft Computing. Springer, pp 89-98

Leung, H., & Chen, T. Y. (1999). A new perspective of the proportional sampling strategy. The Computer Journal, 42(8), 693-698.

Li, Q., Myaeng, S. H., & Kim, B. M. (2007). A probabilistic music recommender considering user opinions and audio features. Information Processing & Management, 43, 473–487.

Li, Y., Lu, L., & Xuefeng, L. (2005). A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce. Expert Systems with Applications, 28, 67–77.

Lihua, W., Lu, L., Jing, L., & Zongyong, L. (2005). Modeling user multiple interests by an improved GCS approach. Expert Systems with Applications, 29, 757–767.

Lillesand, T.M. and Kiefer, R.W. (2000) Remote Sensing and Digital Image Interpretation, Wiley, New York, 724 p.

Lin, W. Association rule mining for collaborative recommender systems. Master's thesis, Department of Computer Science, Worcester Polytechnic Institute, May 2000.

Lin, W.-Y., Alvarez,S.A., and Ruiz,C. Efficient adaptive{support association rule mining for recommender systems. Data Mining and Knowledge Discovery Journal, 6(1):83-105, January 2002.

Linden, G. (2008). People who read this article also read. . .. IEEE Spectrum, 5.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-toitem collaborative filtering. IEEE Internet Computing, 7, 76–80.

Liu, D. R., & Shih, Y. Y. (2005a). Integrating AHP and data mining for product recommendation based on customer lifetime value. Information & Management, 42, 387–400.

Liu, D. R., & Shih, Y. Y. (2005b). Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences. Journal of Systems & Software, 77, 181–191.

Liu, N. H., Hsieh, S. J., & Tsai, C. F. (2010). An intelligent music playlist generator based on the time parameter with artificial neural networks. Expert Systems with Applications, 37, 2815–2825.

Lloyd, S.P., 1982. Lease square quantization in pcm. IEEE Trans. Inform. Theory 28 (2), 129–136.

MacQueen, J.B., 1967. Some method for classification and analysis of multivariate observations. Proc. Berkeley Symp. on Mathematical Statistics and Prob., Berkeley, vol. 1. U. of California Press, pp. 281–297.

Mangasarian, O. L., Street, W. N., & Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), 570-577.

Manly, B. F., Akroyd, J. A. M., & Walshe, K. A. (2002). Two-phase stratified random surveys on multiple populations at multiple locations. New Zealand Journal of Marine and Freshwater Research, 36(3), 581-591.

Martinez, A. B. B., Lopez, M. R., Montenegro, E. C., Fonte, F. A. M., Burguillo, J. C., & Peleterio, A. (2010). Exploiting social tagging in a Web 2.0 recommender system. IEEE Internet Computing, 14, 23–30.

Martin-Guerrero, J. D., Lisboa, P. J. G., Soria-Olivas, E., Palomares, A., & Balaguer, E. (2007). An approach based on the Adaptive Resonance Theory for analyzing the viability of recommender systems in a citizen Web portal. Expert Systems with Applications, 33, 743–753.

Martin-Vicente, M. I., Gil-Solla, A., Ramos-Cabrer, M., Blanco-Fernandez, Y., & Lopez- Nores, M. (2010). A semantic approach to avoiding fake neighborhoods in collaborative recommendation of coupons through digital TV. IEEE Transactions on Consumer Electronics, 56, 54–62.

Meilă, M., & Heckerman, D. (2001). An experimental comparison of model-based clustering methods. Machine learning, 42(1-2), 9-29.

Min, S. H., & Han, I. G. (2005). Detection of the customer time-variant pattern for improving recommender systems. Expert Systems with Applications, 28, 189–199.

Min, S.H and Han,I. Dynamic Fuzzy Clustering for Recommender Systems Lecture Notes in Computer Science, 2005, Volume 3518, Advances in Knowledge Discovery and Data Mining , Springer, Pages 67-126.

Miyahara, K., & Pazzani, M. J. (2000). Collaborative filtering with the simple Bayesian classifier. In PRICAI 2000 Topics in Artificial Intelligence (pp. 679-689). Springer Berlin Heidelberg.

Mobasher, B., Dai, H., Luo, T., and Nakagawa, M., Effective personalization based on association rule discovery from web usage data. In Workshop On Web Information And Data Management, WIDM '01, 2001.

Munoz-Organero, M., Ramiez-Gonzalez, G. A., Munoz-Merino, P. J., & Kloos, C. D. (2010). A collaborative recommender system based on space-time similarities. IEEE Pervasive Computing, 9, 81–87.

Murphey, Y. L., & Luo, Y. (2002). Feature extraction for a multiple pattern classification neural network system. In Pattern Recognition, 2002. Proceedings. 16th International Conference on (Vol. 2, pp. 220-223). IEEE.

Naren, R., Benjamin, J. K., Batul, J. M., Ananth, Y. G., & George, K. (2001). Privacy risks in recommender systems. IEEE Internet Computing, 5, 54–62.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems, 2, 849-856.

Nikovski, D.,and Kulev, V., Induction of compact decision trees for personalized recommendation In SAC '06: Proceedings of the 2006 ACM symposium on Applied computing, pages

Nilashi, M., & Ibrahim, O. B. (2014). A model for detecting customer level intentions to purchase in B2C websites using TOPSIS and fuzzy logic rule-based system. Arabian Journal for Science and Engineering, 39(3), 1907-1922.

Nilashi, M., & Janahmadi, N. (2012, January). Assessing and prioritizing affecting factors in e-learning websites using AHP method and fuzzy approach. In Information and Knowledge Management (Vol. 2, No. 1, pp. 46-61).

Nilashi, M., Ahmadi, H., Ahani, A., Ravangard, R., & bin Ibrahim, O. (2016b). Determining the importance of Hospital Information System adoption factors using Fuzzy Analytic Network Process (ANP). Technological Forecasting and Social Change.

Nilashi, M., Bagherifard, K., Ibrahim, O., Alizadeh, H., Ayodele, N. L., & Nazanin, R. (2013). Collaborative filtering recommender systems. Research Journal of Applied Sciences, Engineering and Technology, 5, 4168–4182.

Nilashi, M., Bagherifard, K., Ibrahim, O., Janahmadi, N., & Barisami, M. (2011a). An application expert system for evaluating effective factors on trust in B2C. Engineering, 3(11), 1063.

Nilashi, M., bin Ibrahim, O., & Ithnin, N. (2014a). Hybrid recommendation approaches for multi-criteria collaborative filtering. Expert Systems with Applications, 41(8), 3879-3900.

Nilashi, M., bin Ibrahim, O., & Ithnin, N. (2014b). Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system. Knowledge-Based Systems, 60, 82-101.

Nilashi, M., bin Ibrahim, O., Ithnin, N., & Sarmin, N. H. (2015b). A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS. Electronic Commerce Research and Applications.

Nilashi, M., Esfahani, M. D., Roudbaraki, M. Z., Ramayah, T., & Ibrahim, O. (2016a). A Multi-Criteria Collaborative Filtering Recommender System Using Clustering and Regression Techniques. Journal of Soft Computing and Decision Support Systems, 3(5), 24-30.

Nilashi, M., Fathian, M., Gholamian, M. R., & Ibrahim, O. B. (2011c). Propose a model for customer purchase decision in B2C websites using adaptive neuro-fuzzy inference system. International Journal of Business Research and Management (IJBRM), 2(1), 1-18.

Nilashi, M., Fathian, M., Gholamian, M. R., bin Ibrahim, O., & Khoshraftar, A. (2011b). The Identification Level of Security, Usability and Transparency Effects on Trust in B2C Commercial Websites Using Adaptive Neuro Fuzzy Inference System (ANFIS). INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS (IJAE), 126.

Nilashi, M., Fathian, M., Gholamian, M. R., bin Ibrahim, O., Talebi, A., & Ithnin, N. (2011d). A comparative study of adaptive neuro fuzzy inferences system (ANFIS) and fuzzy inference system (FIS) approach for trust in B2C electronic commerce websites. JCIT, 6(9), 25-43.

Nilashi, M., Ibrahim, O. B., Ithnin, N., & Zakaria, R. (2014c). A multi-criteria recommendation system using dimensionality reduction and Neuro-Fuzzy techniques. Soft Computing, 1-35.

Nilashi, M., Ibrahim, O., Mirabi, V. R., Ebrahimi, L., & Zare, M. (2015d). The role of Security, Design and Content factors on customer trust in mobile commerce. Journal of Retailing and Consumer Services, 26, 57-69.

Nilashi, M., Jannach, D., bin Ibrahim, O., & Ithnin, N. (2015a). Clustering-and regression-based multi-criteria collaborative filtering with incremental updates. Information Sciences, 293, 235-250.

Nilashi, M., Zakaria, R., Ibrahim, O., Majid, M. Z. A., Zin, R. M., Chugtai, M. W., ... & Yakubu, D. A. (2015c). A knowledge-based expert system for assessing the performance level of green buildings. Knowledge-Based Systems, 86, 194-209.

O'Connor, M., & Herlocker, J. (1999, August). Clustering items for collaborative filtering. In Proceedings of the ACM SIGIR workshop on recommender systems (Vol. 128). UC Berkeley.

Palmer, C. R., & Faloutsos, C. (2000). Density biased sampling: an improved method for data mining and clustering (Vol. 29, No. 2, pp. 82-92). ACM.

Papageorgiou,C., Oren,M., Poggio,T. A general framework for object detection. In: Proceedings of international conference on computer vision; 1998. p. 555–62.

Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. Expert Systems with Applications.

Pazzani, M. A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review 13, 5 - 6 (1999), 393 – 408.
Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The adaptive web (pp. 325-341). Springer Berlin Heidelberg.
Pearson, K., On lines and planes of closest fit to systems of points in space, Phil. Mag., 2 (1991), pp. 559–572.
Pinto MA, Tanscheit R, Vellasco M Hybrid recommendation system based on collaborative filtering and fuzzy numbers. In: Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, 2012. IEEE, pp 1-6
Postorino MN, Sarne GM A Neural Network Hybrid Recommender System. In: Neural Nets WIRN10: Proceedings of the 20th Italian Workshop on Neural Nets, 2011. IOS Press, p 180
Pronk, V., Verhaegh, W., Proidl, A., and Tiemann, M., Incorporating user control into recommender systems based on naive bayesian classification. In RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems, pages 73–80, 2007.
Pyle, D., Data Preparation for Data Mining. Morgan Kaufmann, second edition edition, 1999.
Quinlan,JR. C4.5: programs for machine learning. San Mateo, CA: Morgan Kaufmann; 1993.
Rashidi, M., Hussin, A. R. C., & Nilashi, M. (2015). Entropy-based Ranking Approach for Enhancing Diversity in Tag-based Community Recommendation. Journal of Soft Computing and Decision Support Systems, 3(1), 1-7.
Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994, October). GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work (pp. 175-186). ACM.
Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: an Open Architecture for Collaborative Filtering of Netnews. In CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work, pages 175–186, New York, NY, USA. ACM Press.
Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. Recommender Systems Handbook.
Roh, T. H., Oh, K. J., & Han, I. (2003). The collaborative filtering recommendation based on SOM cluster-indexing CBR. Expert Systems with Applications, 25(3), 413-423.
Rohini, U. and Ambati ,V. A Collaborative Filtering Based Re-ranking Strategy for Search in Digital Libraries Lecture Notes in Computer Science, 2005, Volume 3815, Digital Libraries: Implementing Strategies and Sharing Experiences, Springer,Pages 194-203.
Rosaci, D., Sarne, G. M. L., & Garruzzo, S. (2009). MUADDIB: A distributed recommender system supporting device adaptivity. ACM Transactions on Information Systems, 27, 1–41.
Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500), 2323–2326.
Ruxanda, P. S. M., & Manolopoulos, A. N. Y. (2008). Ternary Semantic Analysis of Social Tags for Personalized Mucis Recommendation. In Proceedings of the 9th International Conference on Music Information Retrieval (p. 219).
Saar-Tsechansky, M., & Provost, F. (2004). Active sampling for class probability estimation and ranking. Machine learning, 54(2), 153-178.
Salahshour, M., Dahlan, H. M., Iahad, N. A., Nilashi, M., & Ibrahim, O. (2015). Using a Multi-Criteria Decision Making Approach for Assessing the Factors Affecting Social Network Sites Intention to Use. Journal of Soft Computing and Decision Support Systems, 2(3), 20-28.
Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In ACM conference on electronic commerce (pp. 158–167).
Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.
Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.
Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., Incremental svd-based algorithms for highly scalable recommender systems. In 5th International Conference on Computer and Information Technology (ICCIT), 2002.
Sarwar, B.M., Karypis, G., Konstan, J.A., and Riedl, J.T., Application of dimensionality reduction in recommender systemsa case study. In ACM WebKDD Workshop, 2000.
Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. In Applications of Data Mining to Electronic Commerce (pp. 115-153). Springer US.
Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. Data Mining and Knowledge Discovery.
Scheffer, T., & Wrobel, S. (2003). Finding the most interesting patterns in a database quickly by using sequential sampling. The Journal of Machine Learning Research, 3, 833-862.
Sebastian,F. Machine learning in automated text categorization. ACM Computing Surveys 2002;34(1):1–47.
Sergio ,A. Alvarez and Carolina Ruiz. Combined collaborative and content-based recommendation using arti_cial neural networks. Pre print, 1999.
Sevarac, Z., Devedzic, V., & Jovanovic, J. (2012). Adaptive neuro-fuzzy pedagogical recommender. Expert Systems with Applications.
Shardanand, U., and Maes, P. Social Information Filtering: Algorithms for Automating "Word of Mouth". In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 95) (Denver, Colorado, USA, May 1995), ACM Press/Addison-Wesley Publishing Co., pp. 210 – 217.
Sharpe,PK, Caleb P (1998) Self organising maps for the investigation of clinical data: a case study. Neural Comput Appl 7:65–70
Shepitsen, A., Gemmell, J., Mobasher, B., & Burke, R. (2008, October). Personalized recommendation in social tagging systems using hierarchical clustering. In Proceedings of the 2008 ACM conference on Recommender systems (pp. 259-266). ACM.
Shlens, J. A. Tutorial on Principal Component Analysis, December 2005.
Srinvasan,A., "A study of two sampling methods for analyzing large datasets with ILP", pp. 95-123 Journal Special Issues on ILP, published in 1999 KLuwer Academic Publishers, Boston
Steinbach,M., Karypis,G., Kumar,V. A comparison of document clustering techniques, Department of Computer Science and Engineering, University of Minnesota, 2000.
Stormer, H., Werro, N., & Risch, D. (2006). Recommending products with a fuzzy classification. CollECTeR Europe 2006, 29.
Subhash, S.K. and Uday, K. Hybrid personalized recommender system using centering-bunching based clustering

algorithm ,Expert Systems with Applications vol. 39 issue 1 January, 2012. p. 1381-1387.

Sun, D., Luo,Z, Zhang Fuhai A Novel Approach for Collaborative Filtering to Alleviate the New Item Cold-Start Problem, ISCIT, 2011.

Sun, J. T., Zeng, H. J., Liu, H., Lu, Y., & Chen, Z. (2005, May). Cubesvd: a novel approach to personalized web search. In Proceedings of the 14th international conference on World Wide Web (pp. 382-390). ACM.

Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2008). Providing justifications in recommender systems. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 38, 1262–1272.

Symeonidis, P., Nanopoulos, A., and Manolopoulos, Y., "Tag recommendations based on tensor dimensionality reduction," in RecSys '08', Lausanne, Switzerland, 2008, pp. 43-50.

Symeonidis, P., Nanopoulos, A., Papadopoulos, A. N., & Manolopoulos, Y. (2008). Collaborative recommender systems: Combining effectiveness and efficiency. Expert Systems with Applications, 34, 2995–3013.

Tam, K.Y., Kiang, M.Y. Managerial applications of neural networks: the case of bank failure predictions, Management Science 38 (7) (July 1992) 926–947.

Tang, T. Y., & McCalla, G. (2009). A multidimensional paper recommender: Experiments and evaluations. IEEE Internet Computing, 13, 34–41.

Tenenbaum, J., de Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500), 2319–2323.

Tou, J.T., and R.C. Gonzales. 1974. Pattern recognition principles. Addison-Wesley Publishing Co.

Treerattanapitak, K., & Jaruskulchai, C. (2010). Membership enhancement with exponential fuzzy clustering for collaborative filtering. In Neural Information Processing. Theory and Algorithms (pp. 559-566). Springer Berlin Heidelberg.

Trippi, R. R., & Turban, E. (1992). Neural networks in finance and investing: Using artificial intelligence to improve real world performance. McGraw-Hill, Inc..

Tubbs, J. D. (1989). A note on binary template matching. Pattern Recognition, 22(4), 359-365.

Uchyigit, G., & Clark, K. (2004). Hierarchical agglomerative clustering for agent-based dynamic collaborative filtering. In Intelligent Data Engineering and Automated Learning–IDEAL 2004 (pp. 827-832). Springer Berlin Heidelberg.

Upendra Shardanand and Patti Maes. Social information filtering: Algorithms for automating \word of mouth". In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, volume 1, pages 210{217, 1995.

Vahid, M., Farokhi, M., Ibrahim, O., & Nilashi, M. (2016). A User Satisfaction Model for E-Commerce Recommender Systems. Journal of Soft Computing and Decision Support Systems, 3(3), 42-54.

Vapnik V Golowich S, Smola A (1996) Support vector method for function approximation, regression estimation, and signal processing. Adv Neural Inf Process Syst 9:281–287

Vapnik V. Statistical learning theory.Wiley; 1998.

Vapnik V. The nature of statistical learning theory.Springer-Verlag; 1995.

Vezina, R., & Militaru, D. (2004). Collaborative filtering: theoretical positions and a research agenda in marketing. International Journal of Technology Management, 28, 31–45.

Villmann ,T., R. Der, M. Herrmann, and T. Martinetz. Topology preservation in self-organizingfeature maps: Exact definition and measurement. IEEE Transactions on Neural Networks,8(2):256–266, 1997.

Wang, H. F., & Wu, C. T. (2009). A mathematical model for product selection strategies in a recommender system. Expert Systems with Applications, 36, 7299–7308.

Wang, L. X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. Systems, Man and Cybernetics, IEEE Transactions on, 22(6), 1414-1427.

Wei, C. P., Yang, C. S., & Hsiao, H. W. (2008). A collaborative filtering-based approach to personalized document clustering. Decision Support Systems, 45, 413–428.

Weng, S. S., & Liu, M. J. (2004). Feature-based recommendations for one-to-one marketing. Expert Systems with Applications, 26, 493–508.

Willett, P. (2003). Similarity-based approaches to virtual screening. Biochemical Society Transactions, 31(Pt 3), 603-606.

Witkosski, U., Rüping, S., Rückert, U., Schütte, F., Beineke, S., & Grotstollen, H. (1997, July). System identification using selforganizing feature maps. In Artificial Neural Networks, Fifth International Conference on (Conf. Publ. No. 440) (pp. 100-105). IET.

Wu, X., Kuman, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., Mclachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z., Steinbach, M., Hand, D.J., Steinberg, D., 2008. Top 10 algorithms in data mining. Knowl. Inform. Systems J. 14 (1), 1–37.

Xia, Z., Dong, Y., & Xing, G. (2006). Support vector machines for collaborative filtering. Paper presented at the Proceedings of the 44th annual Southeast regional conference.

Xu, Y., Zhang, L., & Liu, W. (2006). Cubic analysis of social bookmarking for personalized recommendation. In Frontiers of WWW Research and Development-APWeb 2006 (pp. 733-738). Springer Berlin Heidelberg.

Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., & Chen, Z. (2005, August). Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 114-121). ACM.

Yager RR (2003) Fuzzy logic methods in recommender systems. Fuzzy Sets and Systems 136 (2):133-149

Yoshii, K., Goto, M., Komatani, K., Ogata, T., & Okuno, H. G. (2008). An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. IEEE Transactions on Audio, Speech, and Language Processing, 16, 435–447.

Yu, J. X., Ou, Y., Zhang, C., & Zhang, S. (2005). Identifying interesting visitors through Web log classification. IEEE Intelligent Systems, 20, 55–59.

Yu, K., Schwaighofer, A., Tresp, V., Xu, X., & Kriegel, H. P. (2004). Probabilistic memory-based collaborative filtering. Knowledge and Data Engineering, IEEE Transactions on, 16(1), 56-69.

Yu, K., Tresp, V., and Yu, S., A nonparametric hierarchical bayesian framework for information filtering. In SIGIR '04, 2004.

Yuan, S. T., & Tsao, Y. W. (2003). A recommendation mechanism for contextualized mobile advertising. Expert Systems with Applications, 29, 399–414.

Zadeh LA (1965) Fuzzy sets. Information and control 8 (3):338-353

Zeng, C., Xing, C. X., Zhou, L. Z., & Zheng, X. H. (2004). Similarity measure and instance selection for collaborative filtering. International Journal of Electronic Commerce, 8, 115–129.

Zhang, B., & Srihari, S. N. (2003, January). Binary vector dissimilarity measures for handwriting identification. In Electronic Imaging 2003 (pp. 28-38). International S

Zhang, C. X., & Mlynski, D. (1997). Mapping and hierarchical self-organizing neural networks for VLSI placement. Neural Networks, IEEE Transactions on, 8(2), 299-314.

Zhang, H. H., Ahn, J., Lin, X., & Park, C. (2006). Gene selection using support vector machines with non-convex penalty. Bioinformatics, 22(1), 88-95.

Zhang, T., & Iyengar, V. S. (2002). Recommender systems using linear classifiers. The Journal of Machine Learning Research, 2, 313-334.

Zhang, Y.,and Koren, J., Efficient bayesian hierarchical user modeling for recommendation system. In SIGIR 07, 2007.

Zheng, N., Li, Q., Liao, S., & Zhang, L. (2010). Which photo groups should I choose? A comparative study of recommendation algorithms in Flickr. Journal of Information Science, 36, 733–750.

Zhu, X., Shi, Y. Y., Kim, H. G., & Eom, K. W. (2006). An integrated music recommendation system. IEEE Transactions on Consumer Electronics, 52, 917–925.

Zurada, J., Introduction to artificial neural systems. West Publishing Co., St. Paul, MN, USA, 1992.

An International Journal JSCDSS
E-ISSN: 2289-8603