Applied Soft Computing 12 (2012) 1417–1425

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Cluster ensembles in collaborative filtering recommendation

Chih-Fong Tsai[a,*], Chihli Hung[b]

[a] Department of Information Management National Central University, Taiwan
[b] Department of Information Management Chung Yuan Christian University, Taiwan

ARTICLE INFO

Article history:
Received 20 April 2011
Received in revised form 10 October 2011
Accepted 7 November 2011
Available online 20 November 2011

Keywords:
Recommender systems
Collaborative filtering
k-Means
Self-organizing maps
Cluster ensembles

ABSTRACT

Recommender systems, which recommend items of information that are likely to be of interest to the users, and filter out less favored data items, have been developed. Collaborative filtering is a widely used recommendation technique. It is based on the assumption that people who share the same preferences on some items tend to share the same preferences on other items. Clustering techniques are commonly used for collaborative filtering recommendation. While cluster ensembles have been shown to outperform many single clustering techniques in the literature, the performance of cluster ensembles for recommendation has not been fully examined. Thus, the aim of this paper is to assess the applicability of cluster ensembles to collaborative filtering recommendation. In particular, two well-known clustering techniques (self-organizing maps (SOM) and *k*-means), and three ensemble methods (the cluster-based similarity partitioning algorithm (CSPA), hypergraph partitioning algorithm (HGPA), and majority voting) are used. The experimental results based on the Movielens dataset show that cluster ensembles can provide better recommendation performance than single clustering techniques in terms of recommendation accuracy and precision. In addition, there are no statistically significant differences between either the three SOM ensembles or the three *k*-means ensembles. Either the SOM or *k*-means ensembles could be considered in the future as the baseline collaborative filtering technique.

1. Introduction

Advances in Internet technology have resulted in much information being available online. Recommenders systems can solve the information overload problem. Their aim is to identify users' preferences and to filter out data which is unfavorable to them. As a result, recommender systems can save users' time in searching for information [1].

For purposes of e-commerce, more and more commercial websites, including Amazon.com, CDnow.com, eBay, Levis, and Moviefinder.com, etc. have started to provide recommendation services [2]. These recommender systems can suggest related information or products which are of interest to users, and can increase the relationships between the commercial websites and their customers. Consequently, it is believed that providing users with a certain level of customer satisfaction and high quality recommendations can improve customer loyalty [3].

There are two commonly used approaches in recommender systems, to recommend relevant items to users, which are content-based filtering and collaborative filtering [4]. In content-based filtering, the user information is collected to create his/her profile, and then any items that have been bought are recommended. Therefore, content-based filtering recommends items to those who have similar profile contents to the recommended items. In order to collect users' opinions, rating is used as the universal method. The advantage of content-based filtering is that it can recommend previously un-rated items to users with unique interests and to provide explanations for the recommendations [5].

On the other hand, collaborative filtering is widely applied in recommender systems, and it is the most successful recommendation technique to date [3]. Unlike content-based filtering, collaborative filtering is based on the assumption that people who share the same preferences on some items tend to share the same preferences on other items [6]. Typically, for each user a set of "nearest neighbor" users is found who's past ratings have a high level of correlation. Scores for unseen items are predicted based on the combination of the scores known from the nearest neighbors. Therefore, the main task of this approach is to find users with similar affinities, and it relies on the preferences of these "similar neighbors" to provide recommendations [3].

In general, recommendation can be approached by clustering techniques, such as *k*-means and self-organizing maps (SOM). That is, similar purchased items or products from different customers can be grouped based on the similarity measure of some specific clustering technique. The purchased items in one specific group (or cluster) of customers are then recommended to the new users whose characteristics are similar to this group.

* Corresponding author. Tel.: +886 3 4227151; fax: +886 3 4254604.
E-mail address: cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.11.016

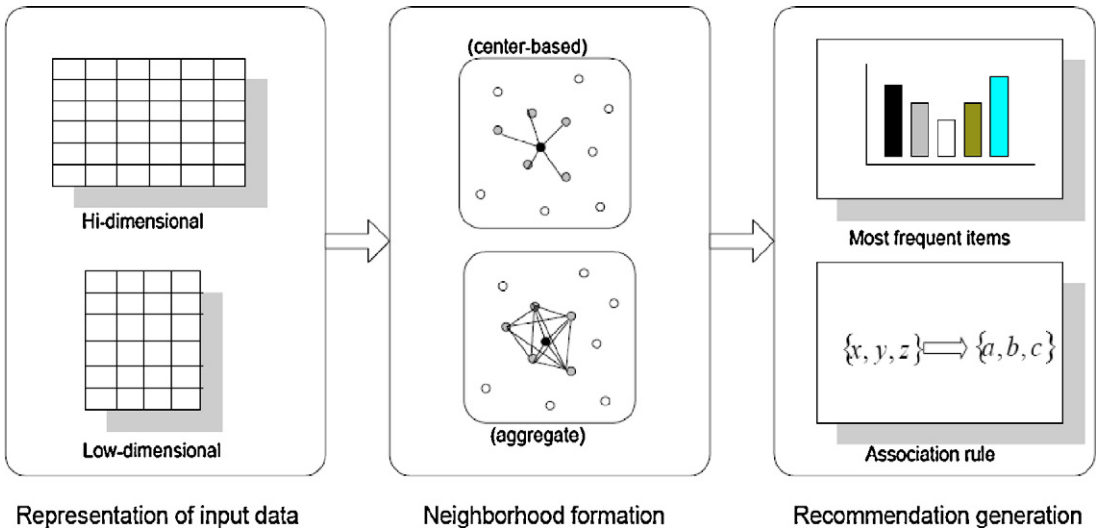

Fig. 1. The collaborative filtering process.

In the literature, cluster ensembles [7] have been proposed by combining multiple partitions of the given data into a single clustering solution in order to improve both the robustness and the stability of unsupervised classification solutions. Although Zhou and Tang [8] show that cluster ensembles outperform single clustering techniques in several domain problems, the performances of cluster ensembles have not been fully examined in the domain of recommender systems.

Therefore, the aim of this paper is to see whether cluster ensembles can improve recommendation performance compared with single clustering techniques. Furthermore, cluster ensembles could be considered in the future as alternative baseline models in addition to single clustering models. In particular, two well-known clustering techniques, which are *k*-means and SOM, and three ensemble methods, which are the cluster-based similarity partitioning algorithm (CSPA), hypergraph partitioning algorithm (HGPA), and majority voting are applied. In addition, the MovieLens dataset [9] is used for the experiments, and the evaluation measures are based on the rate of recommendation accuracy and precision.

The rest of this paper is organized as follows. Section 2 briefly reviews related literature, including the collaborative filtering recommendation, the single clustering techniques, and cluster ensembles. In addition, recent related work is also discussed. Section 3 presents the research methodology of this paper, including the dataset used, clustering techniques developed, the methods for creating cluster ensembles, and the evaluation measures. Section 4 shows the experimental results. Finally, the conclusion is provided in Section 5.

2. Literature review

2.1. Collaborative filtering recommendation

The most commonly used collaborative filtering algorithms for recommendation are based on neighborhood-based techniques. The process of a typical neighborhood-based collaborative filtering system is composed of three phases shown in Fig. 1 [11].

- *Representation*: This step transforms the original user preference matrix into a lower dimensional space. The original input data is a collection of historical purchasing transactions of n customers on m products. This original $m \times n$ representation can be transformed into the reduced dimensional representation in order to solve the sparsity, scalability, and synonymy problems.
- *Neighborhood formation*: For an active user, the similarities between all the other users and the active user are computed to form a proximity-based neighborhood with a number of like-minded users for the active user. There are a number of different ways to compute the similarity between items, such as Pearson's correlation coefficient, constrained Pearson's correlation coefficient, Spearman's rank correlation coefficient, cosine similarity, and mean-squared difference [10,11].
- *Recommendation generation*: The final step is to generate recommendations based on the preferences of the set of nearest neighbors of the active user. For example, Sarwar et al. [11] present two different techniques for performing this task. They are most-frequent item recommendation and association rule-based recommendation.

2.2. Clustering techniques

Clustering can be defined as the process of organizing objects in a database into clusters or groups such that objects within the same cluster have a high degree of similarity, while objects belonging to different clusters have a high degree of dissimilarity [12]. In collaborative filtering recommender systems, users can be regarded as the objects to be clustered. Therefore, where a new user is identified as similar to one specific cluster (or user group) by a clustering algorithm, items which that user group likes are then recommended to the new user.

In the literature, two well-known and widely used clustering techniques for recommender systems are SOM and *k*-means. They are usually used as the baseline recommendation algorithms.

2.2.1. Self-organizing maps

The learning process of SOM is based on a competitive and unsupervised artificial neural network. It is a clustering algorithm that is used to map high-dimensional data into a two-dimensional representation space. SOM can be used to explore the groupings and relations within such data by projecting the data onto a two-dimensional image that clearly indicates regions of similarity [13].

In other words, it consists of an input layer and the Kohonen layer which is designed as a two-dimensional arrangement of neurons that maps the d-dimensional input to the two-dimensional output. Kohonen's SOM associates each of the input vectors to a representative output. The network finds the node closest to each

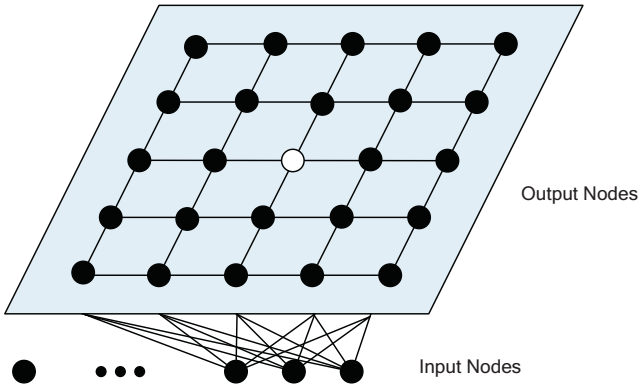

Fig. 2. A 5 × 5 SOM.

training case and moves the winning node, which is the neuron closest (i.e., with the minimum distance) to the training case. Fig. 2 shows an example of a 5 × 5 self-organizing map.

The training stage to construct a SOM is based on competitive learning. That is, when a training example is fed into the SOM, its Euclidean distance to all weight vectors is computed. The neuron with the weight vector, which is the most similar to the input is called the best matching unit (BMU). The weights of the BMU and neurons close to it in the SOM are adjusted for the input vector. The formula for a neuron with the weight vector $W_v(t)$ is

$$W_v(t+1) = W_v(t) + \Theta(v,t)\alpha(t)(D(t) - W_v(t)) \quad (1)$$

where $\alpha(t)$ is a monotonically decreasing learning coefficient and $D(t)$ is the input vector. The neighborhood function $\Theta(v,t)$ depends on the SOM distance between the BMU and neuron v.

2.2.2. k-Means

The k-means algorithm is one of the most frequently used and simplest clustering algorithms because of its ease of implementation, simplicity, and superior capability and efficiency in dealing with large amounts of data. The k-means algorithm is a nonparametric approach that aims to partition objects into k different clusters by minimizing the distances between objects and cluster centers.

Given a set of observations or objects $(x_1, x_2, ...x_n)$ where each object is represented by a d-dimensional vector, the k-means clustering algorithm partitions these n objects into k sets $(k \le n)$ $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2 \quad (2)$$

where μ_i is the mean of data points in S_i.

The basic algorithm is composed of the following steps [14,15].

- Randomly select k data items as the centers of cluster.
- Assign each data item to the group that has the closest centroid.
- When all data items have been assigned, recalculate the positions of the k centroids.
- If there is no further change, end the clustering task; otherwise return to step 2.

2.2.3. Related work using SOM and k-means

Li and Kim [16] propose an adjusted k-means clustering approach to the item-based collaborative filtering framework to solve the cold start problem. The result shows that their proposed approach outperforms k-means in terms of prediction quality.

Roh et al. [17] introduce a three-step collaborative filtering recommendation model, which is composed of the profiling, inferring, and predicting steps. This model combines two machine learning techniques for recommendation, which are SOM and case based reasoning. The combined model performs better than the SOM based collaborative filtering model.

In Liu and Shih [18], two hybrid methods are proposed that exploit the merits of the weighted RFM-based method (WRFM-based method) and the preference-based collaborative filtering method to improve the quality of recommendations. In particular, the k-means clustering algorithm is used to group customers with similar lifetime value or loyalty based on weighted RFM.

Christakou et al. [19] propose a semi-supervised algorithm called MK-means (mass k-means). The method is used to construct a hybrid recommender system for movies that combines content-based and collaborative information, producing high accuracy recommendations.

In Kim and Ahn [20], a novel clustering algorithm, namely the GA k-means, is introduced. In this approach, the initial seeds of the k-means algorithm are optimized by genetic algorithms (GA). It is evaluated by a real world case of online shopping market segmentation. The GA k-means algorithm is compared with k-means and SOM in terms of the segmentation performance. The result shows its usefulness as a pre-processing tool for recommendation systems.

2.3. Cluster ensembles

Cluster ensembles are composed of a number of (different) clustering algorithms to produce a common partition of the original dataset, and focus on reinforcement of results from a combination of individual clustering results [7,21].

In particular, the results of several independent runs of the same clustering algorithm are appropriately combined to obtain a partition of the data which is not affected by initialization, and which overcomes the instabilities of clustering methods. Finally, the fusion procedure starts with the clusters produced by the combining process and finds the optimal number of clusters in the dataset according to some predefined criteria [22]. The cluster ensemble approaches (or consensus functions) are briefly described as follows.

2.3.1. Co-association based functions

According to Strehl and Ghosh [7], in co-association based functions (also known as the pair wise approach), the consensus function operates on a co-association matrix.

Let the dataset D contain N data points in the d-dimensional feature space. The input data can be represented as an $N \times d$ pattern matrix or $N \times N$ dissimilarity matrix. Suppose that $X = \{X1,\ldots,XB\}$ is a set of bootstrap samples or sub-samples of the samples in X that results in B partitions $P = \{P1,\ldots,PB\}$. Each component partition in P is a set of clusters $P_i = \left\{ C_1^i, C_2^i, \ldots, C_{k(i)}^i \right\}$, $X_i = C_1^i \cup C_2^i, \ldots, \cup C_{k(i)}^i$ and $k(i)$ is the number of clusters in the ith partition.

$$\text{co-association}(x,y) = \frac{1}{B}\sum_{i=1}^{B} \varphi(P_1(x), P_i(y)),$$

$$\varphi(a,b) = 1 \text{ if } a = b;\ 0 \text{ if } a \ne b \quad (3)$$

Note that the similarity between a pair of objects simply counts the number of clusters shared by these objects in the partitions $\{P1,\ldots,PB\}$.

Numerous hierarchical agglomerative algorithms (criteria) can be applied to the co-association matrix to obtain the final partition, including single link (SL), average link (AL), complete link (CL) and evidence accumulation clustering (EAC). However, the computational complexity of co-association based functions is very high [23,24].

2.3.2. Hypergraph partitioning

For hypergraph partitioning, the given cluster label vectors are transformed into a hypergraph representation. In particular, a hypergraph consists of vertices and hyperedges. An edge in a regular graph connects two vertices, and a hyperedge is a generalization of an edge which connects any set of vertices. Therefore, the clusters are represented as hyperedges on a graph whose vertices correspond to the clustered objects. Each hyperedge describes a set of objects belonging to the same cluster.

For each label vector $\lambda^{(q)} \in N^n$, the binary membership indicator matrix $H^{(q)}$ is constructed with a column for each cluster represented as a hyperedge. All entries in a row in the binary membership indicator matrix $H^{(q)}$ add up to 1, if the row corresponds to an object with a *known* label. Rows for objects with an unknown label are all zero. Fig. 3 shows an example of transforming the original label vectors into the hypergraph representation.

The concatenated block matrix $H = H^{(1,\ldots,r)} = (H^{(1)} \ldots H^{(r)})$ defines the adjacency matrix of a hypergraph with n vertices and $\sum_{q=1}^{r} k^q$ hyperedges. Each column vector h_a specifies a hyperedge h_a, where 1 indicates that the vertex corresponding to the row is part of that hyperedge and 0 indicates that it is not. Therefore, each cluster is mapped to a hyperedge and the set of clusterings is mapped to a hypergraph.

There are three consensus functions used to transform the set of clusterings into the hypergraph representation. They are the CSPA, HGPA, and hypergraph-based meta-clustering algorithm (MCLA) [7].

CSPA induces a graph from a co-association matrix and clusters it using the METIS algorithm [25]. In CSPA, a clustering means a relationship between objects in the same cluster, and it can be used to establish a measure of pairwise similarity. The similarity measure is then used to re-cluster the objects to produce a combined clustering.

For the HGPA algorithm, the cluster ensemble problem is regarded as a partitioning problem of a suitably defined hypergraph where hyperedges represent clusters. Good hypergraph partitions are found using the minimal cut algorithms, such as HMETIS [26] coupled with the proper objective functions, which also control the partition size.

In MCLA, it is based on the clustering of clusters, and provides object-wise confidence estimates of cluster membership [27]. It is viewed as a cluster correspondence problem whereby groups of clusters have to be identified and consolidated.

2.3.3. Majority voting

The majority voting approach is the simplest method for combining multiple clustering results. The main idea is to permute the cluster labels such that best agreement between the labels of two partitions is obtained [28]. That is, to combine k individual clusters the binary outputs from each of the k individual clusters are pooled together. The output label which receives the largest number of votes is then selected as the final clustering decision.

3. Research methodology

3.1. Dataset

In this paper, the Movielens dataset[1] is used for the experiments because it is recognized as the major dataset for evaluating recommendation algorithms in related studies (cf. Section 2.2.3). The dataset contains 100,000 evaluations (on a scale from 1 to 5) of 1682 films by 943 users. In addition, each user has evaluated at least 20 films. Furthermore, the types to which films belong are found to be in accordance with the types in the Internet Movies Data Base. Finally, simple demographic information for the users (sex, profession, age) is available.

Table 1
The dataset for clustering.

	Movie 1	Movie 2	Movie 3	–	Movie *X*
User 1	5	5	0	–	5
User 2	0	5	5	–	2
User 3	2	0	2	–	2
User 4	3	0	1	–	1
–	–	–	–	–	–
–	–	–	–	–	–
–	–	–	–	–	–
User *Y*	4	1	2	–	3

For the rating information, the user evaluates films that he/she has seen on a scale of one to five, i.e., from 5 (masterpiece) to 1 (bad film). Grades 4 and 5 are good suggestions but grades 1–3 are rejected. This is essential in order to learn the preferences of the user and construct the user's profile.

Therefore, in this paper we investigate two different recommendation types described below.

- *Type I: 45/321*: Grades 4 and 5 belong to the recommendation class (1), and grades 1–3 to the non-recommendation class (0). That is, the output of 1 from the single clustering methods and cluster ensembles means that the movies are recommended where the recommended movies' ratings are either grade 4 or 5.
- *Type II: 5/4321*: Grade 5 is set to belong only to the recommendation class (1), and grades 1–4 to the non-recommendation class (0). As the quality of recommendations is very important [3], the purpose of setting this recommendation type is to decrease the amounts of recommended movies, which is very likely to enhance the recommendation quality.

Note that the original dataset is transformed into user-based feature training data. That is, the user file and the movie file are combined. Table 1 shows the data that are used during the clustering process. It contains all movies that each user has rated in the new file for the experiments, in which each rating ranges from 0 to 5.

3.2. Experimental process

The experimental process is shown in Fig. 4. Initially, five-fold cross validation is used to divide the original data into the training and testing data [29]. Thus, the chosen dataset is divided into 5 non-overlapping subsets. The system is trained by 4 subsets as the training data and tested by the remaining subset as the testing data. As a result, there are 5 different results of the system based on the 5 different testing subsets. Average clustering accuracy can then be computed.

Next, two clustering techniques, which are k-means and SOM, are trained individually. Then, the clustering results from each of the clustering techniques can be obtained after the testing dataset is tested. For cluster ensembles, three cluster ensemble methods, which are CSPA, HGPA, and majority voting, are used to combine multiple k-means and SOM models respectively.

Specifically, for each cluster ensemble, we choose ensemble sizes from 2 to 7 in order to obtain the 'optimal' number of combining multiple clusters which will provide the best recommendation result.

[1] http://www.grouplens.org/node/73.

	$\lambda^{(1)}$	$\lambda^{(2)}$	$\lambda^{(3)}$	$\lambda^{(4)}$
x_1	1	2	1	1
x_2	1	2	1	2
x_3	1	2	2	?
x_4	2	3	2	1
x_5	2	3	3	2
x_6	3	1	3	?
x_7	3	1	3	?

(a) original label vectors

	$H^{(1)}$			$H^{(2)}$			$H^{(3)}$			$H^{(4)}$	
	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}
v_1	1	0	0	0	1	0	1	0	0	1	0
v_2	1	0	0	0	1	0	1	0	0	0	1
v_3	1	0	0	0	1	0	0	1	0	0	0
v_4	0	1	0	0	0	1	0	1	0	1	0
v_5	0	1	0	0	0	1	0	0	1	0	1
v_6	0	0	1	1	0	0	0	0	1	0	0
v_7	0	0	1	1	0	0	0	0	1	0	0

(b) hypergraph representation with 11 hyperedges

Fig. 3. A cluster ensemble problem with $r = 4$, $k^{(1,\ldots,3)} = 3$, and $k^{(4)} = 2$ [7] (a) original label vectors. (b) Hypergraph representation with 11 hyperedges.

3.3. Clustering methods

3.3.1. k-Means and SOM

In this paper, SOM and *k*-means are used as the baseline recommendation systems since they are well-known clustering algorithms in neural networks and statistical fields respectively. Their common features are efficiency and ease of understanding. As our purpose in this paper is the comparison between different ensemble techniques, two straightforward clustering algorithms, i.e., SOM and *k*-means, could make this aim more focused.

Several different parameters are set in order to find the *k*-means and SOM clusters (or models), which can provide the highest rate of recommendation accuracy. For the *k*-means clustering algorithm, the cluster numbers (i.e., *k*) used are from 3 to 15. However, 2×3, 2×4, 2×5, 2×6, 3×3, 3×4, and 4×4 SOMs are developed for comparison.

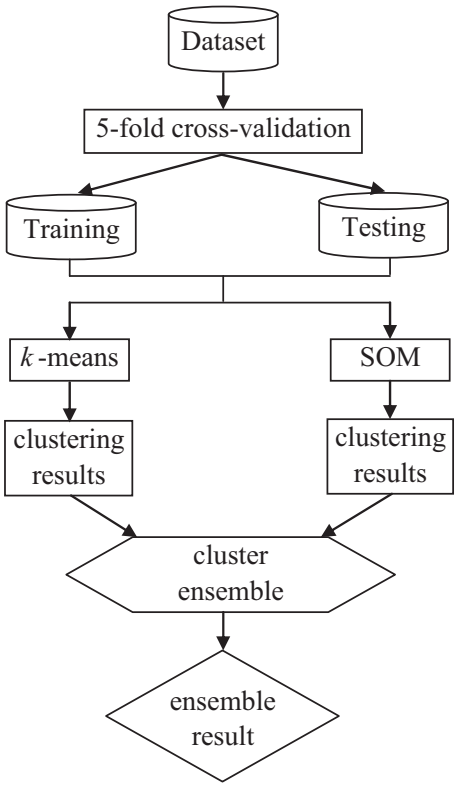

Fig. 4. The experimental process.

Table 2
Confusion matrix.

		Correct result	
		Recommend	Not recommend
Obtained result	Recommend	True positives (TP)	False positives (FP)
	Not recommend	False negatives (FN)	True negatives (TN)

3.3.2. Clustering ensembles

To create cluster ensembles, three ensemble methods are used. These are the CSPA, HGPA, and majority voting.[2]

- *CSPA and HGPA*: After a number of different SOM and *k*-means clusters are developed, we can decide which clustering results are to be combined. For example, to combine the *k*-means clusters with $k = 2$ and 3, the output labels from the two clusters are used for CSPA and HGPA respectively. Each ensemble method will then generate new output labels as the final recommendation results.
- Majority voting. To perform the majority voting approach, after obtaining the clustering results of different *k*-means and SOMs, we consider three ensemble sizes, which are 3, 5, and 7, to develop *k*-means and SOMs cluster ensembles respectively.

3.4. Evaluation metrics

To evaluate the recommendation performance of cluster ensembles as well as the baselines (i.e., *k*-means and SOM), recommendation accuracy and precision/recall are examined. They can be measured by a confusion matrix shown in Table 2.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{4}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

The accuracy rate (or decision support accuracy metric) is based on selecting high-quality items from the set of all items. The prediction process is a binary operation whereby items are predicted either as “good” or “bad” [3].

[2] The cluster ensemble toolbox is downloadable at http://www.lans.ece.utexas.edu/~strehl/.

Table 3
SOM-type I and SOM-type II.

SOM		Accuracy	Precision	Recall
2 × 3	Type I	62.01%	63.25%	61.88%
	Type II	75.56%	77.86%	20.77%
2 × 4	Type I	61.25%	62.11%	61.92%
	Type II	75.49%	77.84%	24.39%
2 × 5	Type I	63.21%	63.56%	62.35%
	Type II	75.56%	77.57%	22.66%
2 × 6	Type I	60.24%	64.20%	63.23%
	Type II	75.71%	77.22%	21.80%
3 × 3	Type I	63.56%	65.22%	64.21%
	Type II	76.23%	77.85%	28.20%
3 × 4	Type I	61.85%	60.22%	64.26%
	Type II	74.76%	76.44%	22.14%
4 × 4	Type I	63.03%	64.53%	63.12%
	Type II	75.08%	73.91%	20.15%
AVG	Type I	62.16%	63.30%	63.00%
	Type II	75.49%	76.95%	21.87%

Precision is the fraction of the recommended items that are interesting to users, and recall is the fraction of the items having higher ratings that are recommended.

4. Experimental results

4.1. The baselines

For the baselines, i.e., SOM and *k*-means, the two different types of recommendation classes are examined (cf. Section 3.1) in terms of their recommendation performances including accuracy, precision, and recall.

4.1.1. SOM

Table 3 shows the performances of SOM by Types I and II recommendation classes. We can see that a SOM using the Type II strategy performs better than Type I in terms of accuracy and precision.

4.1.2. k-Means

For comparison with the SOMs, Table 4 shows the performances of *k*-means by Types I and II recommendation classes respectively. The result also shows that *k*-means using the Type II recommendation class performs better than Type I in terms of accuracy and precision.

Table 4
k-Means-type I and *k*-means-type II.

k-Means		Accuracy	Precision	Recall
3 clusters	Type I	65.73%	64.13%	62.54%
	Type II	75.14%	77.90%	28.31%
5 clusters	Type I	65.61%	68.22%	64.53%
	Type II	75.15%	77.82%	26.52%
7 clusters	Type I	67.59%	68.07%	65.25%
	Type II	75.09%	77.94%	24.94%
9 clusters	Type I	65.60%	64.70%	67.99%
	Type II	75.43%	78.38%	27.06%
11 clusters	Type I	64.34%	67.08%	67.92%
	Type II	74.89%	77.05%	26.30%
13 clusters	Type I	64.02%	64.22%	66.63%
	Type II	74.54%	76.77%	23.34%
15 clusters	Type I	63.41%	62.25%	64.25%
	Type II	74.34%	76.38%	23.13%
AVG	Type I	66.33%	65.38%	65.59%
	Type II	74.94%	77.46%	25.66%

Table 5
The best cluster by types I and II.

Clustering algorithm	Best cluster	Accuracy	Precision	Recall	Amounts of recall
SOM-type I	3 × 3	63.56%	65.22%	64.21%	8400
k-means-type I	7	67.59%	68.07%	65.25%	8651
SOM-type II	3 × 3	76.23%	77.85%	28.20%	3166
k-means-type II	9	75.43%	78.38%	27.06%	3038

4.1.3. Comparisons of SOM and k-means

For the baseline SOM and *k*-means clusters, the best clustering results by Types I and II are compared and shown in Table 5. Both SOM and *k*-means based on Type II can provide higher rates of accuracy and precision and lower recall rates.

For the rates of recall, the best SOM and *k*-means by Type II are much lower than Type I. This indicates that Type I utilizes many recommendations for prediction, which causes lower accuracy and precision rates. However, more accurate recommendations to users are more important than a greater number of recommendations. That is, it is better to recommend movies that users will actually favor rather than to recommend many movies that may possibly contain many less favored movies. Consequently, the accuracy and precision rates should be as high as possible. On the other hand, the recall rate should be as low as possible. Therefore, the cluster ensembles are developed by the clusters using the Type II strategy.

4.2. Cluster ensembles

4.2.1. SOM ensembles

Table 6 shows the performances of SOM ensembles by CSPA, HGPA, and majority voting. From the results, we can see that on average SOM ensembles by HGPA perform best, providing the highest rates of accuracy and precision and the lowest rates of recall.

4.2.2. k-Means ensembles

The performances of *k*-means ensembles by CSPA, HGPA, and majority voting are shown in Table 7. These results show that *k*-means ensembles by CSPA and HGAP perform better, and provide very similar accuracy, precision, and recall rates.

4.3. Further comparisons

Figs. 5–7 further compare the best cluster for each category in terms of their average accuracy, precision, and recall rates respectively.

On average, SOM–HGPA provides the highest rate of average accuracy (76.41%) and precision (78.03%). On the other hand, *k*-means–HGPA provides the lowest rate of recall (19.6%). In short, the SOM and *k*-means ensembles by HGPA and CSPA perform better than other clusters in terms of recommendation

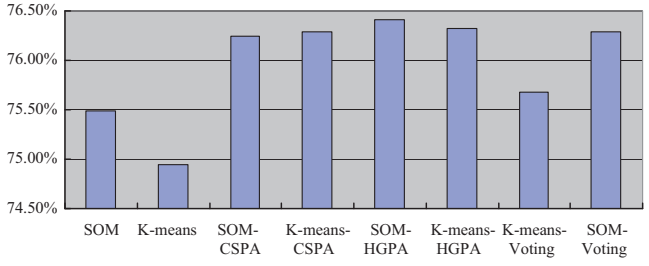

Fig. 5. Average accuracy of the clusters.

Table 6
SOM ensembles by CSPA, HGPA, and majority voting.

Ensemble size	Ensemble Technique	Accuracy	Precision	Recall
$2(3\times3+2\times3)$	CSPA	76.01%	76.80%	21.31%
	HGPA	76.17%	78.02%	18.87%
$3(3\times3+2\times3+2\times5)$	CSPA	75.85%	77.65%	21.19%
	HGPA	76.08%	77.44%	19.45%
	Majority voting	76.35%	76.93%	20.16%
$4(3\times3+2\times3+2\times5+2\times6)$	CSPA	75.78%	77.39%	21.26%
	HGPA	76.01%	77.59%	20.59%
$5(3\times3+2\times3+2\times5+2\times6+2\times4)$	CSPA	76.24%	77.48%	21.73%
	HGPA	76.33%	77.96%	20.10%
	Majority voting	76.23%	77.23%	21.43%
$6(3\times3+2\times3+2\times5+2\times6+2\times4+4\times4)$	CSPA	76.97%	77.63%	24.37%
	HGPA	76.97%	79.06%	24.34%
$7(3\times3+2\times3+2\times5+2\times6+2\times4+3\times4+4\times4)$	CSPA	76.61%	77.32%	19.65%
	HGPA	76.91%	78.12%	19.88%
	Majority voting	76.14%	77.12%	20.44%
Average	CSPA	76.24%	77.38%	21.59%
	HGPA	76.41%	78.03%	20.54%
	Majority voting	76.24%	77.09%	20.68%

Table 7
k-Means ensembles by CSPA, HGPA, and majority voting.

Ensemble size	Ensemble Technique	Accuracy	Precision	Recall
$2(k=5, 9)$	CSPA	76.45%	78.01%	20.51%
	HGPA	75.88%	77.73%	20.72%
$3(k=5, 7, 9)$	CSPA	76.18%	78.63%	19.28%
	HGPA	76.36%	77.73%	18.05%
	Majority voting	75.90%	77.81%	22.92%
$4(k=5, 7, 9, 11)$	CSPA	76.26%	78.26%	19.14%
	HGPA	76.58%	77.71%	17.77%
$5(k=3, 5, 7, 9, 11)$	CSPA	76.57%	77.93%	22.68%
	HGPA	76.82%	77.84%	22.03%
	Majority voting	75.46%	77.92%	23.66%
$6(k=3, 5, 7, 9, 11, 13)$	CSPA	76.23%	77.44%	20.05%
	HGPA	76.28%	77.08%	19.90%
$7(k=3, 5, 7, 9, 11, 13, 15)$	CSPA	76.05%	77.05%	19.51%
	HGPA	76.01%	77.14%	19.10%
	Majority voting	75.55%	77.84%	22.14%
Average	CSPA	76.29%	77.89%	20.19%
	HGPA	76.32%	77.54%	19.60%
	Majority voting	75.64%	77.86%	22.91%

accuracy. In addition, although the SOM and *k*-means ensembles by the voting approach are the worst ensemble methods, they still provide greater accuracy than the baseline SOM and *k*-means.

Fig. 8 shows the relationship between the ensemble size and its accuracy. The result indicates that SOM–HGPA performs better than the other cluster ensembles, i.e., SOM–CSPA, *k*-means–CSPA, *k*-means–HGPA, *k*-means–voting, and SOM–voting. In addition, the best ensemble size for these cluster ensembles is either 5 or 6, except in the voting approach.

Finally, we compare the SOM and *k*-means–CSPA/HGPA ensembles with their associated traditional voting ensemble models. The Friedman test is used to test whether or not an overall difference between our proposed CSPA/HGPA ensembles and their associated traditional voting ensemble models achieves statistical significance [30]. According to the results of the Friedman test, there is not a

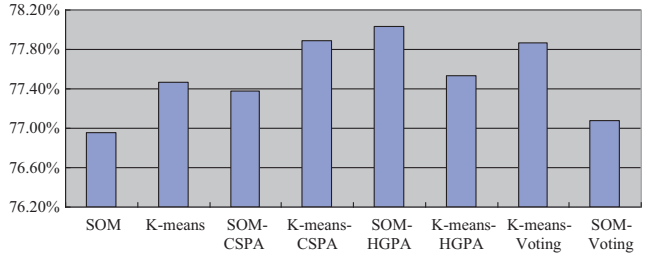

Fig. 6. Average precision of the clusters.

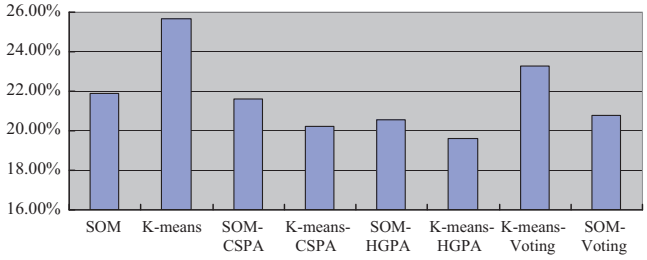

Fig. 7. Average recall of the clusters.

Table 8
Wilcoxon signed rank test for *k*-means–CSPA and *k*-means–HGPA.

	k-means	*k*-means–voting	*k*-means–CSPA	*k*-means–HGPA
k-Means–CSPA	$Z=-2.201$ ($p=0.028$)	$Z=-2.201$ ($p=0.028$)	–	$Z=-0.734$ ($p=0.463$)
k-Means–HGPA	$Z=-2.201$ ($p=0.028$)	$Z=-1.997$ ($p=0.046$)	$Z=-0.734$ ($p=0.463$)	–

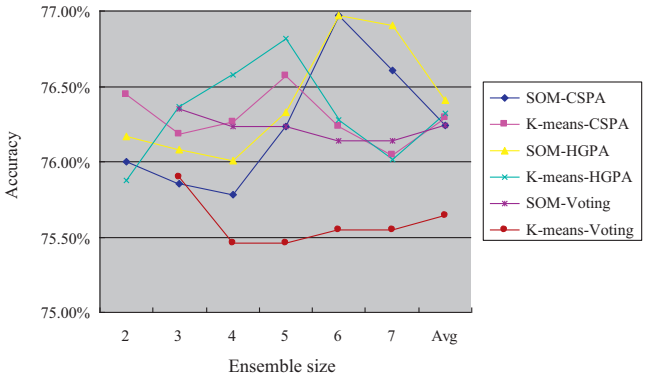

Fig. 8. Ensemble size vs. accuracy.

statistically significant difference between SOM–CSPA, SOM–HGPA and SOM–voting ensembles due to $\chi^2(2)=2.174$, $p=0.337$. Thus, there is no need to run the post-hoc test. For the *k*-means–CSPA, *k*-means–HGPA and *k*-means–voting ensembles, the results of the Friedman test is $\chi^2(2)=6.333$, $p=0.042$, thus there is an overall significant difference between them.

We then run separate Wilcoxon signed rank tests on the different combinations of *k*-means–CSPA, *k*-means–HGPA and *k*-means–voting ensembles to find exactly where those differences lie. As this is a multiple comparison, we consider a Bonferroni adjustment on the results from the Wilcoxon tests. Our original significance level is 0.05 and thus we have a new significance level of $(0.017 = 0.05/3)$ due to the use of three models. Therefore, if the *p* value is larger than 0.017 then we do not have a statistically significant result. As shown in Table 8, the pair wise differences between *k*-means–CSPA, *k*-means–HGPA and *k*-means–voting ensembles do not reach a statistically significant level of difference. Thus, we do not have enough evidence to declare that SOM and *k*-means ensembles by CSPA, HGPA, and majority voting have a significant level of difference.

5. Conclusion

As the development of recommender systems cannot only save users' time but also provide users with specific information that they actually want, how higher recommendation accuracy can be provided is the major research problem in this field. In the literature, clustering techniques have been widely used in recommender systems. In addition, cluster ensembles made by combining multiple clustering techniques have been shown to outperform single clustering techniques in many pattern recognition problem domains. However, very few studies examine the recommendation performance of cluster ensembles.

This paper applies three ensemble approaches to combine multiple clustering techniques, which are the CSPA, HGPA, and majority voting. In particular, the baseline clustering techniques are based on SOM and *k*-means which have been widely considered in recommender systems.

Based on the Movielens dataset, the experimental results show that cluster ensembles perform better than the baseline SOM and *k*-means clustering algorithms. Specifically, the cluster ensembles by SOM–HGPA provide the highest rate of accuracy, i.e., 76.97% and the highest precision rate, i.e., 79.06% to recommend movies for users. However, based on the results of the Friedman test and Wilcoxon signed rank test with the Bonferroni adjustment, there are no statistically significant differences between these ensembles. Therefore, using either cluster ensemble can select more favorable movies and filter out non-favorable movies for users, which can save users' time in examining many movies that they are not interested in. Moreover, these results allow future studies proposing novel clustering ensemble techniques to not only consider single clustering techniques as the baselines, but also compare either SOM or *k*-means ensembles for further validation in order to reach a more reliable conclusion.

For future work, the following issues could be considered:

- *Clustering techniques*: From the practical standpoint, it is difficult to conduct a comprehensive study on all existing clustering methods. Although this paper considers two traditional and well-known clustering algorithms, i.e., SOM and *k*-means, other clustering techniques, such as fuzzy c-means, expectation maximization, etc., can be investigated for their recommendation performances.
- *Evaluation methods*: Some evaluation metrics, such as the receiver operating characteristics (ROC) curve can be considered for an information retrieval system [31] integrated with a recommender system. They can be used to further assess the performances of existing recommendation algorithms.
- *Datasets*: Besides movie recommendations, other domains of recommendations can also be involved for comparison.
- *Input features*: An item based feature or the combination of user and item based features can be further used for comparison with the user based feature.
- *Baselines*: More sophisticated algorithms can be compared with cluster ensembles as the 'advanced' baselines in addition to the single clustering algorithms.

References

[1] P. Resnick, H.R. Varian, Recommender systems, Communications of the ACM 40 (March (3)) (1997) 56–58.
[2] J.B. Schafer, J. Konstan, J. Riedi, Recommender systems in e-commerce, in: Proceedings of the ACM Conference on Electronic Commerce, Denver, Colorado, 1999, pp. 158–166.
[3] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, Hong Kong, 2001, pp. 285–295.
[4] S. Perugini, M.A. Gonçalves, E.A. Fox, Recommender systems research: a connection-centric survey, Journal of Intelligent Information Systems 23 (2) (2004) 107–143.
[5] R.J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, in: Proceedings of the 5th ACM Conference on Digital Libraries, San Antonio, Texas, 2000, pp. 195–204.
[6] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J. Riedl, GroupLens: applying collaborative filtering to usenet news, Communications of the ACM 40 (3) (1997) 77–87.
[7] A. Strehl, J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, Journal on Machine Learning Research 3 (2002) 583–618.
[8] Z.-H. Zhou, W. Tang, Cluster ensemble, Knowledge-Based Systems 19 (1) (2006) 77–83.

[9] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, J. Riedl, MovieLens unplugged: experiences with a recommender system on four mobile devices, in: Proceedings of the 8th International Conference on Intelligent User Interfaces, Miami, Florida, 2003, pp. 263–266.
[10] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Ridel, An algorithmic framework for performing collaborative filtering, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, California, 1999, pp. 230–237.
[11] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: Proceedings of the 2nd ACM Conference on Electronic Commerce, Minneapolis, Minnesota, 2000, pp. 158–167.
[12] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Survey 31 (3) (1999) 264–323.
[13] T. Kohonen, Self-Organizing Maps, 3rd ed., Springer, 2000.
[14] J.A. Hartigan, M.A. Wong, A *k*-means clustering algorithm, Applied Statistics 28 (1) (1979) 100–108.
[15] M. David, An example inference task: clustering Information theory, inference and learning algorithms, Cambridge University Press, 2003 (Chapter 20).
[16] Q. Li, B.M. Kim, Clustering approach for hybrid recommender system, in: Proceedings of the International Conference on Web Intelligence, Halifax, Canada, 2003, pp. 33–38.
[17] T.-H. Roh, K.-J. Oh, I. H, The collaborative filtering recommendation based on SOM cluster-indexing CBR, Expert Systems with Applications 25 (3) (2003) 413–423.
[18] D.-R. Liu, Y.-Y. Shih, Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences, Journal of Systems and Software 77 (2) (2005) 181–191.
[19] C. Christakou, L. Lefakis, S. Vrettos, A. Stafylopatis, A movie recommender system based on semi-supervised clustering, in: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, Vienna, Austria, 2005, pp. 897–903.
[20] K.-j. Kim, H. Ahn, A recommender system using GA *k*-means clustering in an online shopping market, Expert Systems with Applications 34 (2007) 1200–1209.
[21] A.P. Topchy, M.H.C. Law, A.K. Jain, A.L. Fred, Analysis of consensus partition in cluster ensemble, in: Proceedings of the 4th IEEE International Conference on Data Mining, Brighton, UK, 2004, pp. 225–232.
[22] M. Al-Razgan, C. Domeniconi, D. Barbara, Random subspace ensembles for clustering categorical data, in: O. Okun, G. Valentini (Eds.), In Supervised and Unsupervised Ensemble Methods and their Applications, Springer, 2008.
[23] B. Minaei-Bidgoli, A. Topchy, W.F. Punch, Ensembles of partitions via data resampling, in: Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, Nevada, 2004, pp. 188–192.
[24] A.L.N. Fred, A.K. Jain, Data clustering using evidence accumulation, in: Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, 2002, pp. 276–280.
[25] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM Journal on Scientific Computing 20 (1) (1998) 359–392.
[26] G. Karypis, R. Aggarwal, V. Kumar, S. Shekhar, Multilevel hypergraph partitioning: applications in VLSI domain, IEEE Transactions on Very Large Scale Integration Systems 7 (1) (1999) 69–79.
[27] H. Luo, F. Jing, X. Xie, Combining multiple clusterings using information theory based genetic algorithm, in: Proceedings of the International Conference on Computational Intelligence and Security, Guangzhou, China, 2006, pp. 84–89.
[28] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 226–239.
[29] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the International Joint Conference on Artificial Intelligence, Montréal, Québec, Canada, 1995, pp. 1137–1143.
[30] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[31] B.Y. Richard, R.N. Berthier, Modern Information Retrieval, Addison Wesley, 1999.