Clustering- and regression-based multi-criteria collaborative filtering with incremental updates

Mehrbakhsh Nilashi[1a], Dietmar Jannach[b], Othman bin Ibrahim[a], Norafida Ithnin[a]

[a] *Faculty of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia*
[b] *TU Dortmund, Germany*[2]

Abstract

Recommender systems are a valuable means for online users to find items of interest in situations when there exists a large set of alternatives. Collaborative Filtering (CF) is a popular technique to build such systems which is based on explicit rating feedback on the items by a larger user community. Recent research has demonstrated that the predictive accuracy of CF based recommender systems can be measurably improved when multi-criteria ratings are available, i.e., when users provide ratings for different aspects of the recommendable items. Technically, in particular regression-based techniques have been shown to be a promising means to predict the user's overall assessment of an item based on the multi-criteria ratings.

Since in many domains customer subgroups (segments) exist that share similar preferences regarding the item features, we propose a novel CF recommendation approach in which such customer segments are automatically detected through clustering and preference models are learned for each customer segment. In addition, since in practical application constantly new rating information is available, the proposed method supports incremental updates of the preference models. An empirical evaluation of our method shows that the predictions of the resulting models are more accurate than previous multi-criteria recommendation methods.

Keywords: Recommender Systems, Collaborative Filtering, Multi-criteria Ratings, Evaluation

[1] Corresponding author, email: `nilashidotnet@hotmail.com`
[2] Email: `dietmar.jannach@tu-dortmund.de`

1. Introduction

During the last decade, the amount of information available on the Web increased exponentially leading to the problem of information overload for users. Information retrieval and information filtering systems are designed to help users deal with this information overload (Bordogna and Pasi, 2010) and Recommender Systems (RS) represent one particular type of such systems.

Some early recommender systems were designed as personal tools, e.g., for news filtering, but soon became popular as valuable means to support the online customer in the product discovery and the decision making and buying process in e-commerce settings (Linden et al., 2003; Jannach et al., 2011; Jannach and Kreutler, 2007). Today, personalized information filtering and recommendation is omnipresent on the web and includes a diverse set of services such as friend, group or resource recommendation on Web 2.0 platforms (Cai et al., 2011; Serrano-Guerrero et al., 2011; Porcel et al., 2012), personalized music recommendation on online music services (Hariri et al., 2012; Bonnin and Jannach, 2014), or tag recommendation on image and general resource sharing platforms (Jäschke et al., 2007).

The literature differentiates between the following basic strategies to build recommender systems (Adomavicius and Tuzhilin, 2005; Jannach et al., 2011): (1) *Collaborative filtering* approaches rely on the existence of explicit or implicit rating feedback on items provided by a larger user community to predict the relevance of an unseen item for a given user. (2) *Content-based* methods try to estimate the user's interest in certain item features (the *user profile*) based on the past behavior of the user and assess the relevance of an item by matching the features of an unseen item with the profile. (3) *Knowledge-based* approaches encode explicit domain knowledge to match user preferences with item features and rank the items, e.g., based on utility or similarity functions. (4) *Hybrid approaches* finally combine two or more techniques to better exploit the available knowledge sources and to avoid the drawbacks of individual approaches.

The work presented in this paper falls into the category of collaborative filtering approaches, which represents the most popular family of techniques in the literature (Jannach et al., 2012b). In contrast to the typical assumption that the rating database consist of a matrix containing the users' *overall assessments* of the items, our work targets scenarios where *multi-criteria* ratings are available. Today, allowing online visitors to provide fine-grained multi-criteria rating feedback is for instance common in the domain of travel and tourism. On the popular TripAdvisor[3] tourism platform, users can for example rate hotels in various dimensions such as cleanliness, service or value for money. Figure 1 shows a sketch of a typical review and rating interface which can be found in similar form on large tourism platforms.

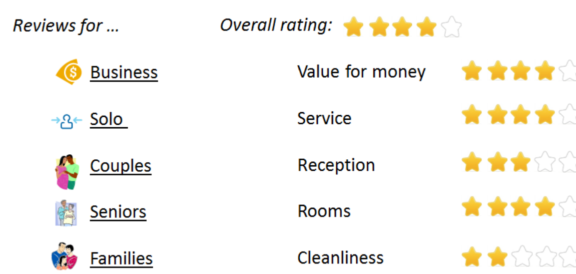

Figure 1: Example for multi-criteria ratings and customer segments

In (Adomavicius and Kwon, 2007), a number of basic strategies were proposed to exploit multi-criteria ratings to improve the predictive accuracy of a recommender in terms of typical information retrieval measures. Later on, a number of additional techniques to leverage the detailed ratings in the recommendation process were proposed, e.g., in (Liu et al., 2011), (Zhang et al., 2009), (Sahoo et al., 2012), (Manouselis et al., 2014), (Bilge and Kaleli, 2014), (Mikeli et al., 2013a), (Shambour and Lu, 2011), (Jannach et al., 2012a) and (Jannach et al., 2014).

The work presented in this paper continues these lines of research. Specifically, we build on the approach by Jannach et al. (2012a) who proposed to learn per-user and per-item regression models from the data, combined them in a weighted approach and showed that their methods outperform the approaches of Adomavicius and Kwon (2007).

A recent analysis by Jannach et al. (2014) of a multi-criteria rating dataset has shown that the importance of different quality factors is not consistent across different customer segments. In the analyzed tourism domain, for example, senior couples – not so surprisingly – apply other decision criteria than young solo travelers.

Therefore, in our approach, as a first contribution we propose to automat-

[3]http://www.tripadvisor.com

ically detect such customer segments based on a clustering process and then learn importance weights for the different quality factors through regression. Using such a clustering approach also helps us to deal with "cold-start" users for which only a limited amount of ratings is available and for which no user-specific regression models can be learned as was proposed in (Jannach et al., 2012a). In order to reduce the dimensionality of the models, to deal with noise in the data, and to address possible multi-collinearity effects induced by interdependencies between the criteria ratings, we additionally propose to apply Principal Component Analysis (PCA).

From a practical perspective, recommendation models are typically trained in an offline process and retrained after a certain time frame. Since, however, new rating information arrives at high rates on real-world platforms, it is desirable to incrementally update the once trained models to ensure a constantly high prediction accuracy. As a second contribution of this work we therefore propose an approach in which newly arriving data can be directly incorporated into the models without retraining, which we achieve by applying incremental versions of the used regression and dimensionality reduction techniques.

The paper is organized as follows. In Section 2, we characterize the multi-criteria recommendation problem and review existing works. Next, in Section 3, we first give an overview of the different steps of our approach and then discuss the technical aspects in more detail. The results of an experimental evaluation are presented in Section 4. The paper ends with a discussion of the limitations of the work and an outlook on possible directions for future work.

2. Multi-criteria Recommendation Techniques

2.1. General Schemes

Figure 2 shows an excerpt of a typical multi-criteria rating database from the tourism domain, where users u1 and u2 not only assigned overall ratings to the hotels i1, i2, and i3 but in addition provided detailed ratings for the quality dimensions Service, Value and Rooms.

One question in this example could be if we should recommend hotel i3 to user u2, which means that we have to predict u3's overall rating for it. In cases in which we only consider only the overall ratings, we could follow a neighborhood-based approach and analyze how users that are similar to u2 with respect to the past ratings have rated hotel i3. In our example, the

User	Item	Overall	Service	Value	Rooms
u1	**i1**	4	2	4	1
u1	**i2**	2	3	2	4
u1	**i3**	4	2	4	2
u2	**i1**	4	4	2	3
u2	**i2**	2	2	4	3
u2	**i3**	?	?	?	?

Figure 2: Excerpt of a multi-criteria rating dataset

overall rating behavior u1 and u2 was identical for i1 and i2 and we would therefore predict that u2 will agree with u1 also on i3 – u1 liked hotel i3 – and recommend the hotel.

If we however look closer at the detailed ratings, we can observe that the overall ratings of user u1 are probably strongly determined by the assessment of the Value criterion. The overall rating of user u2, in contrast, seems to follow the Service rating. This means that the users ended up with the same overall rating but had different reasons. If we assume that the Service aspect is truly important for user u2 and that the service of hotel i3 is in fact not good – as indicated by the low rating for the Service dimension by user u1 – we would therefore rather predict a low rating for this hotel by user u2 and not recommend it.

In (Adomavicius and Kwon, 2007), two general strategies were proposed to exploit the multi-criteria rating information in recommender systems to make more accurate predictions.

Similarity-based approaches. The idea of this family of approaches is basically to rely on a traditional nearest-neighbor recommendation technique as described in (Resnick et al., 1994) to predict the overall rating. However, instead of using, for example, the Pearson correlation coefficient of the overall ratings as a similarity measure[4], the idea is to estimate the similarities of the users either by aggregating the similarity values obtained for the individual rating dimensions or by applying multidimensional distance metrics[5].

[4]Alternative measures could be used; see Ekstrand et al. (2011) for a recent performance comparison of different similarity measures.

[5]See also (Bilge and Kaleli, 2014) for a recent comparative evaluation of such approaches or (Shambour and Lu, 2011) for an approach that uses semantic information to compute

$$
\begin{aligned}
&RO_{i1} = w1_{i1} * Service + w2_{i1} * Value + w3_{i1} * Rooms + c_{i1}\\
&RO_{i2} = w1_{i2} * Service + w2_{i2} * Value + w3_{i2} * Rooms + c_{i2}\\
&\ldots\\
&RO_{in} = w1_{in} * Service + w2_{in} * Value + w3_{in} * Rooms + c_{in}
\end{aligned}
$$

Figure 3: Learning regression functions for each item

Aggregation function based approaches. When adopting this approach, the idea is to learn regression models to predict the overall rating for an item from the detailed criteria ratings. Adomavicius and Kwon (2007) propose to learn a corresponding regression function for each item as shown in Figure 3.

In their approach, the predicted overall rating RO for an item is a weighted combination of the individual criteria ratings, where the individual weights w are fitted to the training data. In order to predict the overall rating for a user u1, one has to first make an estimate about the criteria ratings that user u1 will assign for the item. To determine these criteria ratings, each rating dimension can be considered as an individual collaborative filtering problem. In the example, one could for instance use a classical nearest neighbor approach to predict the user's rating for the Service criterion based on his past ratings in this dimension for other hotels and the ratings of other users regarding the service.

2.2. Empirical Evaluation and Algorithmic Improvements

In (Adomavicius and Kwon, 2007), the authors evaluate several variations of the similarity-based approach and the regression-based method on a set of multi-criteria ratings from the Yahoo!Movies platform[6]. Their experiments show that relying on multi-criteria rating feedback can in fact lead to higher prediction accuracy in terms of precision and recall when compared to a traditional neighborhood-based approach. In particular, the approach based on aggregation functions performed quite well.

Jannach et al. (2012a) later on proposed a number of improvements for the aggregation function based approach. In particular, the following three enhancements to the basic scheme were introduced.

---

item similarities based on multiple criteria.

[6]`https://www.yahoo.com`

1. In addition to the regression functions that are learned for each item, an additional function is learned *for each user*. The two predictions are then combined in a weighted approach, where the combination weights are optimized on the training data.
2. Support Vector Regression (SVR) (Drucker et al., 1997) is used instead of Ordinary Least Squares (OLS) regression, in particular because SVR can be applied even when only a limited amount of data points is available; furthermore SVR shows to have a limited tendency of overfitting.
3. *Feature selection* was used when there were many quality factors in a domain in order to factor out rating dimensions that carry little information or even introduce noise.

Jannach et al. (2012a) evaluated the improvements on two different data sets, one again based on data from Yahoo!Movies and one from the tourism domain. Their experiments showed that the suggested improvements not only lead to better results than those achieved with the techniques presented in (Adomavicius and Kwon, 2007), but also that the predictions are more accurate than more recent single-rating approaches based on matrix factorization. At the same time, it was observed that the similarity-based approaches – even though reasonably accurate – have a limited coverage and can generate recommendations only for a smaller fraction of the users.

The work presented in this paper builds upon the work described in (Jannach et al., 2012a). Details about the enhancements proposed in this paper will be given in Section 3.

2.3. Other Existing Approaches

Compared to the huge amount of approaches proposed in the literature over the last years that rely on single-rating feedback, research on building recommender systems using multi-criteria rating information is still limited. Adomavicius et al. (2011) propose to categorize existing approaches as follows.

- Classical content-based recommenders in which the user's preference weights for each item feature are estimated based on the user's overall ratings for the items.

- Knowledge-based approaches in which users explicitly state their *general* preferences in an interactive elicitation process.

- Systems in which the users can specify their evaluation for individual items in different dimensions as it is typically done, e.g., in the tourism domain (see Figure 1).

The work presented in this paper falls into the third category of this classification scheme. Note that there exists a number of works in the field of Multi-Criteria Decision Making (MCDM) and optimization, see (Manouselis and Costopoulou, 2007) for an overview of such approaches or (Hdioud et al., 2013) for a recent MCDM-based approach. The relation of these approaches to our work is however limited and we will therefore focus the following discussion of previous works on learning-based collaborative filtering approaches.

In (Sahoo et al., 2006) and later in (Sahoo et al., 2012) the authors extend the Flexible Mixture Model (FMM) approach for collaborative filtering proposed in (Si and Jin, 2003) for the multi-criteria rating case. In their works, the aim is to automatically detect dependencies between the overall ratings and what they call "multi-component ratings" with a probabilistic approach and the Chow-Liu tree structure discovery algorithm (Chow and Liu, 1968). Empirical evaluations on the Yahoo!Movies dataset showed that higher predictive accuracy can be achieved than when using the original FMM formulation that only relies on the overall ratings.

Another probabilistic approach was proposed in (Zhang et al., 2009) which is based on the Probabilistic Latent Semantic Analysis (PLSA) model (Hofmann, 2004). An evaluation on the Yahoo!Movies dataset showed that the approach is favorable when compared with a typical item-to-item collaborative filtering method.

Li et al. (2008) presented a multi-criteria rating approach to improve personalized services in mobile commerce using Multi-linear Singular Value Decomposition (MSVD). The aim of their work was to exploit context information about the user as well as multi-criteria ratings in the recommendation process. They evaluated their approach based on data from a restaurant recommendation scenario. The results showed improvements in terms of precision and recall when compared with a relatively weak baseline method.

Later on, Liu et al. (2011) proposed a multi-criteria recommendation approach which is based on the clustering of users. Their idea is that for each user, one of the criteria is "dominant" and users are grouped according to their criteria preferences. They applied linear least squares regression, assigned each user to one cluster, and evaluated different schemes for the generation of predictions. An evaluation based on a dataset from TripAdvisor showed that the multi-criteria approach leads to better accuracy than traditional single-rating schemes.

In their recent work, Mikeli et al. propose to learn personalized preference weights for the different rating dimensions based on the Analytic Hierarchy Process (AHP) and an optimization procedure which is based on pairwise comparisons of the detailed item ratings of each user (Mikeli et al., 2013a,b). The resulting user-specific weight vectors are then clustered and predictions for unknown items are made as in usual nearest-neighbor methods through a weighted combinations of the ratings of similar users. Experiments with the Yahoo!Movies dataset showed that their method leads to more accurate results than a recent matrix factorization method SVD++ (Koren, 2008) and the multi-criteria recommendation method of Lakiotaki et al. (2011), where the latter method seemed to performed worse than the single-rating method SVD++.

In general, in most of the previous works the baseline algorithms used for the comparison consist of traditional single-rating collaborative filtering techniques, whereas in our work, we aim to compare our approach with more recent techniques based on matrix factorization like the one described in Funk (2006). Furthermore, in some of the approaches, data pre-processing was done and the evaluation was limited to users that have rated at least 20 items, e.g., (Liu et al., 2011) or (Li et al., 2008). In our view, this can represent a strong assumption, for example in the tourism domain.

Overall, our work is similar to the works of Adomavicius and Kwon (2007), Liu et al. (2011) and Jannach et al. (2012a) in that we use regression techniques to predict the overall ratings from the given multi-criteria ratings. Furthermore, our work is in some sense similar to that of Liu et al. (2011) as we apply clustering. In our work, however, clustering is based on a different criterion. In addition, the predictions from two regression models are combined in a weighted approach and we furthermore use automated feature selection to remove noise in the data. Finally, we support incremental updates. We will describe our research methodology and the technical approach in more detail in the next section.

3. A Clustering-based Approach Supporting Incremental Updates

Our approach consists of two phases. In the *offline* training phase, multiple processing steps are performed to cluster the data and then learn regression functions. These functions should at the end help us obtain more accurate predictions than when using the approaches of Adomavicius and Kwon (2007) or Jannach et al. (2012a).

In the *online* phase, later on, rating predictions are made based on the existing models. At the same time, our approach is designed in a way that newly arriving user ratings can be incrementally incorporated into the models without requiring a full retraining phase over all the data.

3.1. Offline Phase

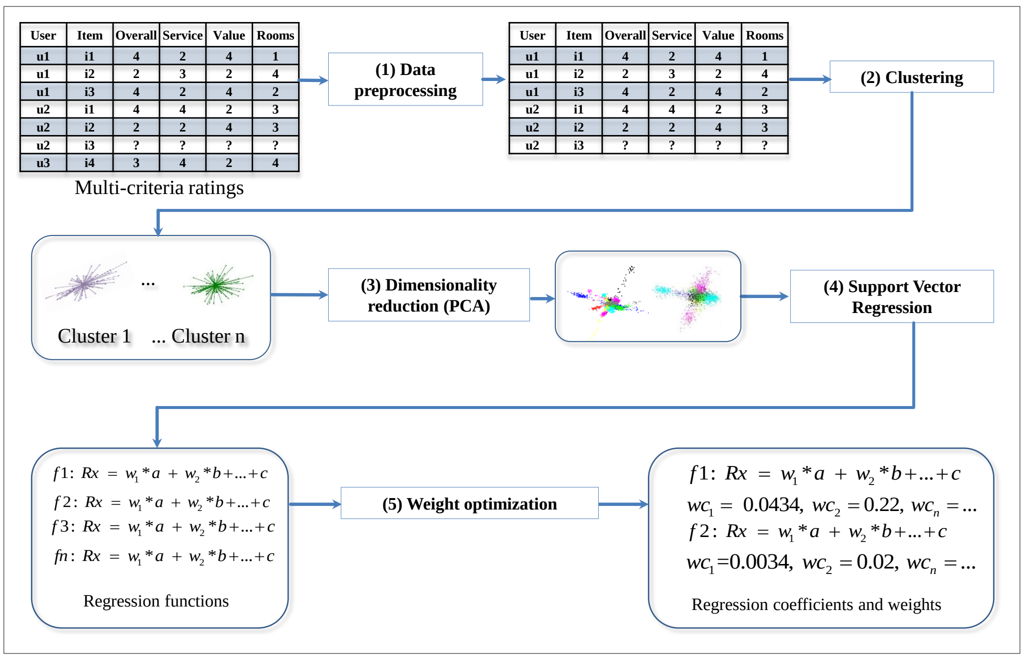

Figure 4: Overview of the offline learning phase

Figure 4 shows an overview of the different steps in the offline learning process. First, we pre-process the data (1) in order to filter out users who have only rated very few items and for which no reasonable prediction can be made using collaborative filtering techniques[7]. The second step (2) consists of clustering the users and the items and then (3) apply Principal Component Analysis (PCA) to reduce the dimensionality of the data and reduce potential

[7]In our empirical evaluation we will apply a ten-fold cross-validation procedure. All steps after (1) would therefore only be applied to the corresponding training data set.

noise. Next (4), regression functions are learned for each user cluster and each item cluster. This is different to the work of Jannach et al. (2012a), who learned such models for each user and item, which can be problematic when there are only few data points available for a user or an item. Finally, we learn optimal combination weights for the cluster-based functions by minimizing the prediction error on the training data as done in (Jannach et al., 2012a). In the following, we we present more details about the individual steps.

Data pre-processing. Data sparsity represents a major issue in many real-world recommender applications. In the tourism domain, for example, where multi-criteria ratings are quite common, the average user only rates a small number of items, see e.g. (Jannach et al., 2012a). Also the multi-criteria dataset from Yahoo!Movies that we will use in our experiments has a considerably higher data sparsity the popular datasets from MovieLens and Netflix, where each user has rated several dozen items. In the pre-processing phase, we have therefore created several subsamples of the original datasets in order to assess the effects of varying levels of sparsity on the prediction accuracy. More details about the dataset statistics, data analysis results, and the applied transformation and pre-processing steps will be given later in Section 4.

Clustering (ASCA + AK). Clustering helps us to identify customer segments with similar tastes and to thereby overcome cold-start situations, since learning regression functions for users and items for that only very few data points exist can be problematic.

In the literature, a large variety of clustering techniques have been proposed. Some of these approaches – like the popular k-means method – require some initial information like the number of clusters as an input. Some clustering methods can work very efficiently based on heuristics; however, the danger exists that they get stuck in local minima. In our work, we propose to use a Genetic Algorithm (GA) approach for clustering based on ant colonies as proposed e.g., by Maulik and Bandyopadhyay (2000), which overcomes some of the mentioned challenges. When using such a method, for example, no information about the future classification of the data is required. In addition, the stochastic nature of this nature-inspired family of algorithms helps to escape local minima.

Specifically, we apply the clustering approach proposed in (Kuo et al., 2005) and (Kuo et al., 2007) called Ant System-based Clustering Algorithm

(ASCA) and Ant K-means algorithm (AK), which were applied by the authors in the context of an association rule mining problem setting. In this two-stage approach, the ASCA algorithm is used to determine the number of clusters. AK is then applied to optimize the resulting set of non-overlapping clusters based on the Total Within Cluster Variance (TWCV).

In our work, we rely on the Silhouette coefficient (Kaufman and Rousseeuw, 1990) to determine the quality of the resulting clusters and to assess the best cluster formation. The Silhouette coefficient is an internal index that measures how good the clustering fits the original data based on statistical properties of the clustered data. External indices, by contrast, measure the quality of a clustering by comparing it with externally provided labels.

The Silhouette coefficient of an element i of a cluster k is defined by the average distance $a(i)$ between i and the other elements of k (the intra-cluster distance), and the distance $b(i)$ between i and the nearest element in the nearest cluster (i's minimal inter-cluster distance):

$$sc_i = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{1}$$

An overall score for a set of n_k elements (one cluster or the entire clustering) is calculated by taking the average of the Silhouette coefficients sc_i of all elements i in the set:

$$SC_k = \frac{1}{n_k} \Sigma_{i=1}^{n_k} sc_i \tag{2}$$

The Silhouette coefficient can take values between -1 and 1. Higher values indicate a better fit of the clusters for the given data. Table 1 lists a general rule of thumb on how to interpret the Silhouette coefficient.

Range	**Interpretation**
>0.7	Strong structure was found
0.5-0.7	Reasonable structure was found
0.25-0.5	The structure is weak and could be artificial
< 0.25	No substantial structure was found

Table 1: Interpretation of Silhouette coefficient values

In Section 4, we will report the detailed observations of applying the ASCA + AK clustering technique on the data. Note that in our methodology also other clustering approaches can in principle be applied. In our experiments, we however picked the ASCA + AK technique due to its favorable characteristics as mentioned above.

Dimensionality reduction (PCA). Principal Component Analysis (PCA) is a powerful and wide-spread multivariate statistical method to structure, compress, interpret and visualize larger datasets which was introduced in (Pearson, 1901). Given a dataset consisting of potentially noisy observations for a larger set of possibly correlated and possibly redundant variables, the main idea of PCA is to construct a typically much smaller set of variables, the principal components, which are not linearly correlated and which approximate the original data.

In the field of recommender systems, PCA has for example been applied by Goldberg et al. (2001) in the Jester joke recommender. In their approach, PCA was performed in an offline phase and they applied clustering on the resulting projection of the data in a two-dimensional space. Our approach is different from their work in two ways. First, we apply PCA *after* the initial clustering process individually on each cluster and determine a suitable number of principal components to retain for each cluster.

Secondly, our goal is to support incremental updates when new data arrives. Applying a classical PCA procedure on large real-world rating datasets can be computationally challenging. Even if a recent randomized procedure is applied, re-computing the eigenvectors can take a significant amount of computing time (Halko et al., 2011). In our work, we therefore rely on the *incremental* PCA procedure proposed by Hall et al. (1998), which allows us to update the eigenvectors and eigenvalues whenever new observations arrive, which typically happens very frequently on large e-commerce sites. Following this approach should finally allow us to achieve highly accurate and up-to-date recommendations without requiring a costly retraining of the model.

Learning regression functions (SVR). This step is similar to what has been done in (Jannach et al., 2012a). First, we use Support Vector Regression to learn the rating prediction functions. We use SV-regression due to their advantages over other regression methods in the given application scenario. Differently from previous works, we learn these functions not for each individual user and item but for each of the clusters of ratings that were computed and optimized in the previous steps.

Again, to be able to update our models when new observations arrive, we use an incremental version of the algorithm. In (Cauwenberghs and Poggio, 2001), an approach for "online" training of a classifier based on Support Vector Machines (SVMs) was proposed. Similar mechanisms can be applied for Support Vector Regression as shown, e.g., in (Martin, 2002). Ma et al. (2003) experimentally showed that their incremental version is more efficient than a batch training procedure, which is also mentioned to be comparably time-consuming when the number of data points is large.

Details about specific parameterizations of the regression process will be given later in Section 4,

Weight optimization. In their multi-criteria recommendation scheme, Jannach et al. (2012a) used a weighted hybrid approach, in which the overall prediction for user u and item i is computed as a weighted combination of the predictions of a user-specific function $\hat{r}_{u,i}^{user}$ and an item-specific regression function $\hat{r}_{u,i}^{item}$ as shown in Equation 3.

$$\hat{r}_{u,i} = w_u * \hat{r}_{u,i}^{user} + w_i * \hat{r}_{u,i}^{item} \tag{3}$$

The weights w_u and w_i for each user and each item are determined by minimizing the prediction error on the training data using a gradient descent optimization procedure. In our cluster-based approach, we adopt a similar strategy. However, since we have no user- and item specific regression functions to combine, we instead combine the cluster-based regression that were learned in the previous step. The selection of the regression functions to combine therefore depends on the cluster assignment of the individual user-item tuples.

3.2. Online Phase

During the online phase, i.e., after the system has been initially trained and deployed, two sorts of computations are required. First, on arrival of a new multi-criteria rating sample, the models have to be incrementally updated; second, requests for recommendations for a given user have to be answered by the system based on the models. Figure 5 illustrates the required steps for the two computations.

Integrating new ratings. When a new rating arrives, we first add it to the set of all ratings. Next, we compute the distance of the new data point to the center of each cluster learned in the offline phase. As a similarity measure, we use the Euclidian distance (Adomavicius et al., 2011). Once the nearest cluster is determined, we apply the incremental PCA and incremental SVR methods to finally update the regression models of the corresponding cluster.

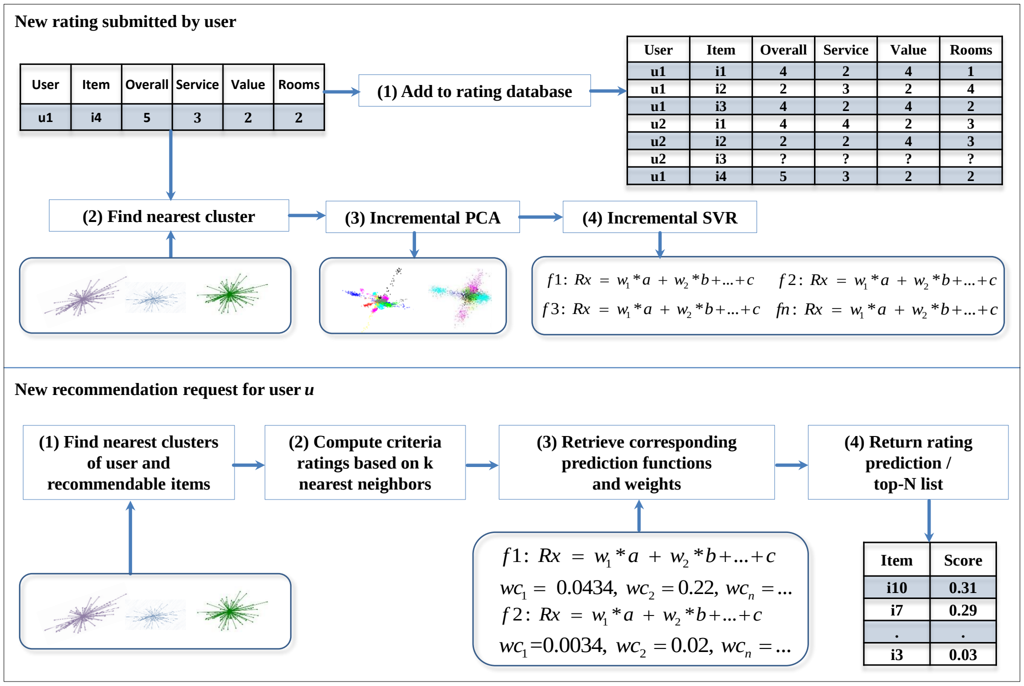

Figure 5: Overview of the online phase

Recommending items. In the literature, two typical tasks of a recommender algorithm are mentioned, *rating prediction* and *item recommendation*. The latter task, i.e., generating a ranked list of items to be recommended to the user, can be accomplished by predicting ratings for those items that can be recommended to the user and then return a list that is sorted by these predictions. In our method, we follow this approach and the main task is therefore to make rating predictions[8].

Computing rating predictions in our method is done in two steps as typically done in aggregation-function based approaches. First, given a target user u and a target item i for which we search for a prediction, we try to estimate the criteria ratings that u would assign to item i. In the hotel recommendation application, we would for example first make a prediction about the user's rating for the *Service* or the *Cleanliness* aspect of the hotel

[8]Note that computing rating predictions is only one form of ranking the items, think, e.g., of learning-to-rank approaches.

in question. Similar to (Jannach et al., 2012a), these estimates are then used as parameters for the regression functions for the target user and target item which we have learned in the offline phase.

For the first step, i.e., determining the criteria ratings, we however propose a new method which is different from existing approaches proposed in the literature. The procedure consists of three phases.

1. We first assign the target user u to one of the clusters determined in the offline phase. We therefore calculate a "mean rating vector" for the target user by calculating the user's average rating value for each dimension given the user's past ratings. The resulting rating vector is then compared to the center of each of the clusters and the user is finally assigned to the *target cluster* whose center has the highest similarity with the user's mean rating vector in terms of the Pearson correlation coefficient.
2. Next, we look for the k most similar users (neighbors of the target user) in the cluster, considering only the ratings that are contained in the target cluster. Again, the comparison is made by comparing the mean vectors of all users and the target user based on the Pearson coefficient[9].
3. Finally, for each dimension we use the average of the criteria ratings of the k most similar users as a prediction for the target user's criteria ratings.

4. Empirical Evaluation

In the following section, we will describe how we applied our methodology to the multi-criteria movie rating dataset which was also used in the work of Jannach et al. (2012a).

4.1. Dataset Characteristics and Pre-processing

Dataset characteristics. The dataset used in the following analysis was obtained through a crawling process from the Yahoo!Movies platform[10]. At the

[9] Alternative neighbor selection techniques as proposed, e.g., in (Ortega et al., 2013), are possible.

[10] `http://movies.yahoo.com`. The multi-criteria rating feature is not available for users on the current version of the platform.

time of data acquisition, platform users could rate movies in the four dimensions Acting, Directing, Story and Visuals. In addition, they can provide an overall rating. The 13-point rating scale used on the platform ranges from A+ (the best rating) to F (the worst rating). We transformed the ratings into numerical values, where 13 corresponds to the best and 1 to the worst possible rating.

The "raw" dataset contained 251,317 multi-criteria ratings. Quite interestingly, the mean ratings provided for the individual quality dimensions are much lower than the mean of the overall ratings. While the overall mean rating was at 10.06 (B+), the mean rating, e.g., for the Acting dimension was only at 7.84 (between B- and C+); the lowest average was observed for the Directing aspect (7.57).

Sparsity and subsampling. The ratings were provided by 127,829 users and were concerning 8,272 different movies. The general data sparsity is therefore extremely high as more than 99.99% of the entries of the user-item rating matrix are empty. On average, every user has provided about 2 ratings in the original dataset. Preference estimates for users who provided only one or two ratings are however not very reliable. Therefore, we created different subsamples of the original dataset for which we varied density constraints on the minimum number of ratings per user and ratings per item.

As done in a similar way in (Jannach et al., 2012a), we created three datasets called YM-5-5, YM-10-10, and YM-20-20, where YM-10-10 for example means that we only retained users who had at least rated 10 movies. At the same time, we only considered movies in that dataset which were rated by at least 10 different users. The statistics for the resulting datasets are shown in Table 2.

Name	#Users	#Items	#Overall ratings
`YM-20-20`	429	491	18,504
`YM-10-10`	1,827	1,471	48,026
`YM-5-5`	5,978	3,079	82,599

Table 2: Overview of the dataset characteristics

In order to check possible collinearities between the criteria ratings, we computed the linear correlations between them. Table 3 shows the results and we can actually observe statistical significant correlations between the variables ($p < 0.01$).

X	**Y**	**r**	r^2	**t**
Acting	Directing	0.858	0.736	426.928
Acting	Story	0.832	0.693	384.148
Acting	Visuals	0.791	0.625	330.887
Directing	Story	0.861	0.740	432.358
Directing	Visuals	0.840	0.705	395.382
Story	Visuals	0.788	0.621	327.448

Table 3: Correlation between criteria ratings

4.2. Clustering the Data

We applied the two-stage ASCA + AK clustering method by Kuo et al. (2007) described in Section 3. When applying the ant system based clustering algorithm (ASCA) in the first stage of the algorithm, suitable parameter values for α (relative trail importance), β (relative importance of visibility), ρ (trail persistence), and Q (a constant) have to be determined. Similar to (Dorigo et al., 1996), we tested a number of value combinations for the parameters. In particular, we tested the following parameter values: $\alpha = \{0, 0.5, 1, 2, 5\}$, $\beta = \{0, 1, 2, 5\}$, $\rho = \{0.3, 0.5, 0.7, 0.99, 0.999\}$, and $Q = \{1, 100, 1000\}$. A systematic evaluation of the 300 possible combinations on the data revealed that the parameters setting $\alpha = 0.5$, $\beta = 0.5$, $\rho = 0.9$ and $Q = 1$ led to the smallest Total Within-Cluster Variance (TWCV).

In the agglomeration phase of the ASCA algorithm, we used the Silhouette coefficient (SC) to determine the cluster quality and used it as a basis to decide whether two sets of objects T should be merged. We merged two cluster candidates when the SC value was lower than the threshold value α. After applying the ASCA algorithm we ended up with 8 clusters, which were then modified (optimized) using the Ant k-Means (AK) algorithm as described in (Kuo et al., 2005). In that stage, we used TWCV to quantify the quality of the partitioning.

To assess the quality of the resulting clusters, we finally computed the Silhouette values for them. The Silhouette plots are shown in Figure 6. The average value of the Silhouette coefficient is at about 0.89 which indicates a relatively high clustering quality according to the interpretation rules given in Table 1.

In Table 4, we show the values for the cluster centers. As mentioned above, these cluster centers are used to assign newly arriving data points to a cluster based on their Euclidian distance.

4.3. Applying Principal Component Analysis

The next step in our approach is the application of Principal Component Analysis (PCA) on the clusters obtained in the previous steps. In this analysis, we

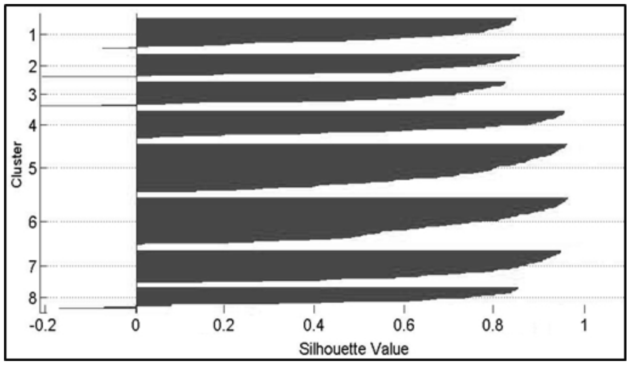

Figure 6: Silhouette plot for the clusters

	1	2	3	4	5	6	7	8
Acting	12.71	8.00	9.38	8.85	2.20	6.40	10.33	10.44
Directory	12.70	2.00	10.85	5.08	1.60	5.80	9.75	11.44
Story	12.57	5.00	6.54	8.77	1.87	5.50	9.47	12.31
Visual	12.60	1.00	4.54	8.85	1.80	8.35	10.22	8.44
Overall	12.30	3.00	6.54	6.62	1.80	5.90	9.73	9.94

Table 4: Cluster centers

considered the four criteria ratings but not the overall rating. In Table 5, we exemplarily show the results of the analysis for the first cluster and the eigenvalues of the four principal components (axes).

Axis	Eigenvalue	Difference	Proportion (%)	Cumulative (%)
1	3.46	3.23	86.48%	86.48%
2	0.23	0.06	5.85%	92.33%
3	0.18	0.05	4.44%	96.77%
4	0.13	-	3.23%	100.00%
Total	4.00	-	-	-

Table 5: Eigenvalues, individual and cumulative variance for the first cluster

In Table 6 we furthermore show the detailed factor loadings (weights) for the data, where the loadings correspond to the correlation of the variables (criteria) with the principal components. The squared correlation values indicate the percentage of the variance of a variable that is explained by the factor. In the last row we see the percentage of the total variance that was accounted for by the first

n factors. For this cluster, for example, the two first factors explain 92.33% of the total variance, see also Table 5.

Attribute	**Axis 1** % (Tot. %)	**Axis 2** % (Tot. %)	**Axis 3** % (Tot. %)	**Axis 4** % (Tot. %)
Directing	90 % (90 %)	0 % (90 %)	0 % (90 %)	10 % (100 %)
Acting	86 % (86 %)	3 % (90 %)	9 % (99 %)	1 % (100 %)
Story	86 % (86 %)	4 % (90 %)	9 % (99 %)	1 % (100 %)
Visuals	83 % (83 %)	16 % (99 %)	0 % (99 %)	1 % (100 %)
Var. Expl.	8**6 % (86 %)**	**6 % (92 %)**	4 % (97 %)	3 % (100 %)

Table 6: Factor loadings with respect to the different components

Table 7 finally shows the corresponding component score coefficients used to map individual data points.

Attribute	**Axis 1**	**Axis 2**	**Axis 3**	**Axis 4**
Directing	**0.51**	-0.06	0.01	**0.86**
Acting	**0.50**	-0.37	-0.72	-0.32
Story	**0.50**	-0.40	**0.68**	-0.33
Visuals	**0.49**	**0.84**	0.01	-0.24

Table 7: Component score coefficients for the example cluster

Since the goal of applying PCA is to deal with collinearities in the data and to avoid noise, the question has now to be answered how many of the principal factors we should retain. Retaining too many factors will probably not help us reducing potential noise in the data. On the other hand, if we retain too few components, the danger exists that valuable information is lost.

The eigenvalues that are associated with the factors are indicators for their importance. In our work, we decided to use the rule proposed by Cattell (1966) and create "scree" plots as shown in Figure 7 where we plot the eigenvalues of the factors to detect "elbows" that indicate possible changes in the structure of the data. The figure shows that there is an elbow at factor 2 and we should therefore retain the first two factors. To validate this choice, we can again look at Table 5 and see that more than 92% of the variance in the data for the particular cluster is explained by these two factors.

The same procedure was applied to all eight clusters, leading to the final selection of principle components (factors) as show in Table 8.

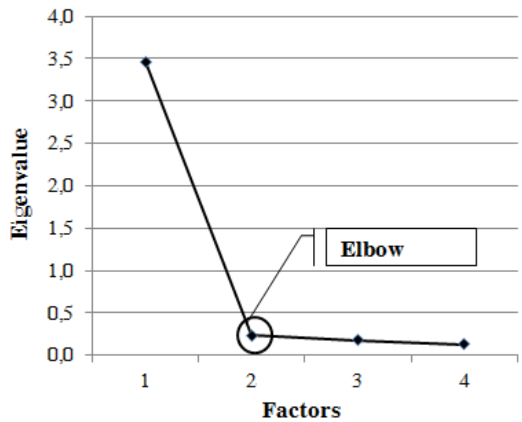

Figure 7: Scree plot with an "elbow" at the second factor

	PC 1	PC 2	PC 3	PC 4
Cluster 1	x	x	-	-
Cluster 2	x	x	x	-
Cluster 3	x	x	-	-
Cluster 4	x	x	-	-
Cluster 5	x	x	x	-
Cluster 6	x	x	-	-
Cluster 7	x	x	-	-
Cluster 8	x	x	-	-

Table 8: Results of applying PCA on the eight clusters

4.4. Choosing a Suitable SVR Configuration

In (Jannach et al., 2012a), no details about the specific parameters of the SVR learning process are provided except for the penalty factor c. In order to maximize the accuracy of our regression models, we evaluated different configurations of the Support Vector Regression learner. In particular, we made experiments for parameter tuning in which we tested four different configurations. One point of variation lies in the choice of the kernel function and we tested both a linear kernel and a Radial Basis Function (RBF) kernel. The second variation lies in the penalty configuration and the corresponding choice of the nu-SVR or ϵ-SVR method.

Testing the different configurations on our data revealed that the RBF kernel function leads to a better prediction accuracy than when a linear kernel was used. The difference between nu-SVR and ϵ-SVR were however very low in the first experiments. Therefore, we decided to test both variants in the final evaluation.

4.5. Evaluation Method

We rely on the typical evaluation protocol and accuracy measures described, e.g., in (Shani and Gunawardana, 2011), to determine the quality of the recommendations and to compare our method with previous works. Specifically, we split the data into training and test splits and try to predict the rating or ranking of the hidden items in the test dataset. To factor out random effects, we apply a ten-fold cross-validation procedure. As quality measures we use the Root Mean Squared Error (RMSE) for the rating prediction task and precision, recall and the F1 measure to assess the quality of the item rankings. To be able to compare our results with those reported in the literature, we furthermore measure the Mean Absolute Error (MAE) and use different list lengths when determining precision and recall.

Both the RMSE and the MAE measure the deviation of the rating values predicted by the recommender $\hat{x_i}$ from the true ratings x_i. The RMSE formula is shown in Equation 4. The difference to the MAE measure is that the RMSE applies a higher penalty when the deviation is higher.

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{n}(\hat{x_i} - x_i)^2}{n}} \tag{4}$$

To determine the measures precision, recall and the F1 measure, we classify the recommendable items as *relevant* and *irrelevant* for each user. Given a set of actually *recommended* items for a user, e.g., of size 10, then precision (precision@10) is defined as shown in Equation 5.

$$precision = \frac{|relevant| \cap |recommended|}{|recommended|} \tag{5}$$

Recall, in contrast, is the ratio of the relevant items that were actually retrieved, i.e., we have $|relevant|$ in the denominator of Equation 5. The F1 measure is usually defined as the harmonic mean of precision and recall.

Beside these standard measurements, we will furthermore use a special evaluation protocol to assess the value of the incremental update procedure. More details will be given later in Section 4.7.

4.6. Accuracy Results

We use the `WeightedSVR` method of Jannach et al. (2012a) as a baseline in our evaluation, because it has shown to outperform the methods proposed in (Adomavicius and Kwon, 2007) and a recent matrix factorization method.

To assess the individual contribution of each of our enhancements, we compare the baseline to the following configurations.

- `WeightedSVR` + Clustering (ASCA + AK)
- `WeightedSVR` (nu-SVR) + Clustering + PCA
- `WeightedSVR` (ϵ-SVR) + Clustering + PCA

The accuracy results are shown in Tables 9 and 10. In Table 9, we report the RMSE values that were obtained when using a 1-to-5 scale as done in (Jannach et al., 2012a)[11]. The results show that the combination of the clustering and the PCA technique helps us to measurably decrease the prediction error. The best configuration in this experiment is the one that uses *epsilon*-SVR. The coverage of all approaches is 100%, i.e., predictions for all user-item pairs in the test set could be made.

Method	**YM-20-20**	**YM-10-10**	**YM-5-5**
`WeightedSVR` (Jannach et al., 2012a)	0.57	0.60	0.63
`WeightedSVR` + Clust.	0.54	0.56	0.59
`WeightedSVR` (nu) + Clust. + PCA	0.52	0.54	0.58
`WeightedSVR` (ϵ) + Clust. + PCA	**0.50**	**0.52**	**0.55**

Table 9: RMSE results for the Yahoo!Movies datasets using a 1-to-5 scale

In Table 10, we report the numbers for precision and the MAE when using the original 1-to-13 scale in order to be able to compare the numbers to those reported in (Jannach et al., 2012a) and (Adomavicius and Kwon, 2007). Again, we can see that the clustering and PCA techniques help to increase precision and recall and at the same time reduce the Mean Absolute Error (MAE)[12].

Method	**Prec@5**	**Prec@7**	**MAE**
`WeightedSVR` (Jannach et al., 2012a)	75.62	73.26	1.05
`WeightedSVR` + Clust.	78.39	75.45	0.98
`WeightedSVR` (nu-SVR) + Clust. + PCA	80.24	79.58	0.93
`WeightedSVR` (ϵ-SVR) + Clust. + PCA	**82.14**	**81.45**	**0.87**

Table 10: Precision and MAE results for the YM-10-10 dataset on the original 1-13 scale

[11]The first row repeats the RMSE numbers of Jannach et al. (2012a). In their work, they used 95% of the data for training. Here, we use 10-fold cross-validation and only use 90% of the data in the training phases.

[12]Jannach et al. (2012a) used 5-fold cross-validation to determine precision and recall. Therefore, they only used 80% for learning, while we always had 90% in the training set.

4.7. Evaluation with Incremental Updates

In order to assess the value of being able to incrementally update the models when new ratings arrive, we made the following experiment. First, we split the rating data of each of our 8 clusters into three parts.

1. The first 20% of the ratings were chosen as a test.
2. Another 20% of the ratings formed the initial training set.
3. The remaining 60% were incrementally added to the training set.

Based on these data splits, we first trained the algorithms on the initial 20% of the training set and measured precision and the MAE using the test set. Then, we incrementally added the ratings of the remaining 60% of the data to the training set and made repeated measurements of precision and the MAE. Specifically, we made a measurement after adding the next 5% of the "incremental data", therefore ending up with 12 measurement points. The additional data points were integrated into the models based on the incremental SVR and incremental PCA procedures.

Figure 8 and Figure 9 show the results of this experiment in terms of the development of the MAE and precision when new ratings are incorporated.

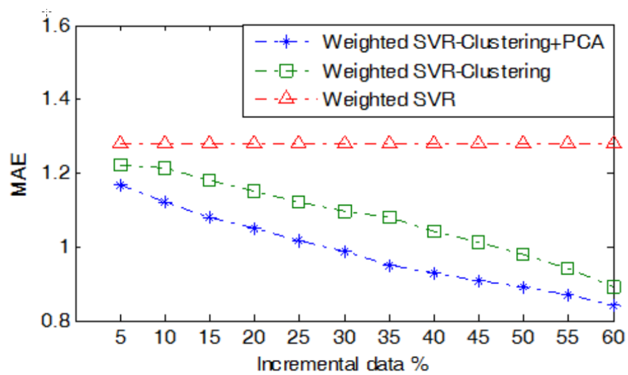

Figure 8: Effect of incremental updates on the MAE

Since the original `WeightedSVR` method does not support incremental updates, the obtained values for this method remain constant. When incremental updates are supported, the accuracy of the new methods proposed in this paper relatively quickly increases both in terms of the MAE and precision. Again, the method that combines both clustering and Principal Component Analysis leads to the most accurate results.

Overall, in application domains in which new ratings arrive at a high rate, incremental updates allow us to constantly improve our recommendation models without computationally costly retraining based on the whole dataset.

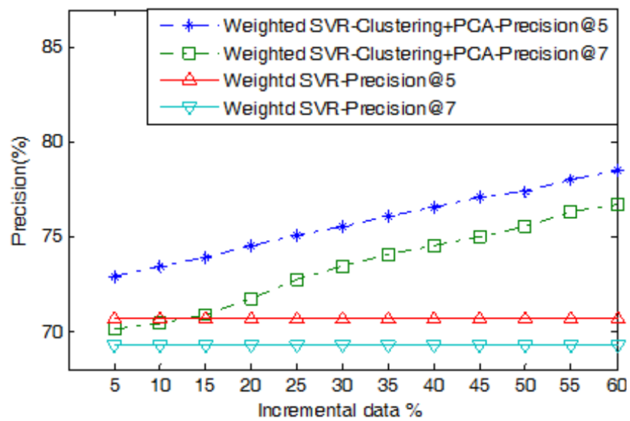

Figure 9: Effect of incremental updates on precision

4.8. Research Limitations

In the area of single-rating recommendation scenarios, a number of publicly available datasets – e.g., the various MovieLens datasets – exist. Unfortunately, such public datasets do not exist so far for the multi-criteria rating case. Research in the field is therefore often based on one single and proprietary collection of user ratings, including the early work of Adomavicius and Kwon (2007) as well as the more recent works reported in (Fuchs and Zanker, 2012), (Liu et al., 2011), (Sahoo et al., 2012), or (Zhang et al., 2009).

Our work shares the limitation of these previous works since we only made experiments in one single domain. However, by making experiments with varying density levels (see Table 2), we could validate that accuracy improvements can consistently be achieved even when the characteristics of the datasets are quite different. Nonetheless, more experiments for different domains are required to confirm the generalizability of our observations across domains as was done for the regression-based method proposed in (Jannach et al., 2012a).

5. Summary and Outlook

Previous works have shown that multi-criteria ratings can contain valuable information that helps us improve the prediction accuracy of recommender systems. Multi-criteria rating information can however be both sparse and noisy leading to a situation where the usually well-performing techniques based on aggregation functions cannot unleash their full potential.

In this work, we propose a novel way and different algorithmic approaches to deal with such situations which are actually quite common in realistic settings. We address the data sparsity problem by clustering the data. Specifically, we propose to use an Ant System-based Clustering Algorithm (ASCA) and an Ant K-means algorithm (AK) to identify customer segments with similar tastes. These clusters allow us to learn more reliable regression functions. To deal with the problem of noise in the data, we then apply Principal Component Analysis (PCA) and thereby identify the most important quality dimensions for the different customer segments. Finally, to be able to incorporate the continuous stream of new rating data that can arrive at online platforms, our data processing chain supports incremental updates using incremental SVR and PCA techniques. These methods allow us to immediately exploit newly available information and constantly update our recommendation models.

In our empirical evaluation, we tested several configurations of our new set of techniques and benchmarked them with the recent WeightedSVR method of Jannach et al. (2012a). The results show that our clustering and noise removal techniques help to improve the prediction accuracy by more than 10% in all tested scenarios in terms of the RMSE and the MAE.

Furthermore, for the incremental situation, we could show through a simulation experiment that incorporating newly arriving ratings can be crucial for the success of a practical recommendation system as they have an immediate impact on the accuracy of the predictions. Previous approaches from the literature often require a full retraining of the recommendation model. Therefore, they are not able to integrate new customer feedback, which would however be particularly valuable for cold-start users who have not provided many ratings yet.

In our future work, we plan to evaluate our techniques on additional datasets and in particular on data from the tourism domain, where multi-criteria ratings are common. Since rating data in tourism application is even sparser and many users often rate a very small number of hotels, we assume that the clustering and noise reduction techniques will be particularly helpful also in this domain.

Acknowledgments

The authors gratefully acknowledge the support of this research from MOE, Universiti Teknologi Malaysia grant (Grant No. FRGS/1/2014/ICT07/UTM/02/1), Research Management Centre, K-Economy, Information and Service Systems Innovation Research Group (ISSI), Faculty of Computing, UTM. Appreciation also goes to the anonymous reviewers whose comments helped us to improve the manuscript.

References

Adomavicius, G., Kwon, Y., May 2007. New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems 22, 48–55.

Adomavicius, G., Manouselis, N., Kwon, Y., 2011. Multi-criteria recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. B. (Eds.), Recommender Systems Handbook. Springer US, pp. 769–803.

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

Bilge, A., Kaleli, C., 2014. A multi-criteria item-based collaborative filtering framework. In: 11th International Joint Conference on Computer Science and Software Engineering (JCSSE '14). pp. 18–22.

Bonnin, G., Jannach, D., 2014. Automated generation of music playlists: Survey and experiments. ACM Computing Surveys (to appear).

Bordogna, G., Pasi, G., 2010. A flexible multi criteria information filtering model. Soft Computing 14 (8), 799–809.

Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y., Compton, P., Mahidadia, A., 2011. Collaborative filtering for people to people recommendation in social networks. In: AI 2010: Advances in Artificial Intelligence. Vol. 6464 of Lecture Notes in Computer Science. pp. 476–485.

Cattell, R. B., 1966. The scree test for the number of factors. Multivariate Behavioral Research 1 (2), 245–276.

Cauwenberghs, G., Poggio, T., 2001. Incremental and decremental support vector machine learning. In: Advances in Neural Information Processing Systems (NIPS*2000). Vol. 13.

Chow, C., Liu, C., 1968. Approximating discrete probability distributions with dependence trees. Information Theory, IEEE Transactions on 14 (3), 462–467.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26 (1), 29–41.

Drucker, H., Chris, Kaufman, B. L., Smola, A., Vapnik, V., 1997. Support vector regression machines. In: Advances in Neural Information Processing Systems 9. Vol. 9. pp. 155–161.

Ekstrand, M. D., Ludwig, M., Konstan, J. A., Riedl, J. T., 2011. Rethinking the recommender research ecosystem: Reproducibility, openness, and lenskit. In:

Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11). pp. 133–140.

Fuchs, M., Zanker, M., 2012. Multi-criteria ratings for recommender systems: An empirical analysis in the tourism domain. In: Proc. E-Commerce and Web Technologies - 13th International Conference (EC-Web 2012). pp. 100–111.

Funk, S., 2006. (pen name), netflix update: Try this at home, http://sifter.org/ simon/journal/20061211.html.

Goldberg, K., Roeder, T., Gupta, D., Perkins, C., 2001. Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval 4 (2), 133–151.

Halko, N., Martinsson, P.-G., Shkolnisky, Y., Tygert, M., 2011. An algorithm for the principal component analysis of large data sets. SIAM Journal on Scientific Computing 33 (5), 2580–2594.

Hall, P., Marshall, D., Martin, R., 1998. Incremental Eigenanalysis for Classification. In: British Machine Vision Conference. pp. 286–295.

Hariri, N., Mobasher, B., Burke, R., 2012. Context-aware music recommendation based on latenttopic sequential patterns. In: Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12). pp. 131–138.

Hdioud, F., Frikh, B., Ouhbi, B., 2013. Multi-criteria recommender systems based on multi-attribute decision making. In: Proceedings of International Conference on Information Integration and Web-based Applications & Services (IIWAS '13). pp. 203:203–203:210.

Hofmann, T., 2004. Latent semantic models for collaborative filtering. ACM Transactions on Information Systems 22, 89–115.

Jannach, D., Karakaya, Z., Gedikli, F., 2012a. Accuracy improvements for multi-criteria recommender systems. In: Proceedings of the 13th ACM Conference on Electronic Commerce (EC 2012). pp. 674–689.

Jannach, D., Kreutler, G., 2007. Rapid development of knowledge-based conversational recommender applications with advisor suite. Journal of Web Engineering 6 (3), 165–192.

Jannach, D., Zanker, M., Felfernig, A., Friedrich, G., 2011. Recommender Systems - An Introduction. Cambridge University Press.

Jannach, D., Zanker, M., Fuchs, M., 2014. Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations. Journal of Information Technology and Tourism 14 (2), 119–149.

Jannach, D., Zanker, M., Ge, M., Gröning, M., 2012b. Recommender systems in computer science and information systems - a landscape of research. In: Proceedings 13th International Conference on E-Commerce and Web Technologies (EC-WEB 2012). Vienna, pp. 76–87.

Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G., 2007. Tag recommendations in folksonomies. In: Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007). pp. 506–514.

Kaufman, L., Rousseeuw, P. J., 1990. Finding groups in data: an introduction to cluster analysis. John Wiley and Sons, New York.

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08). Las Vegas, USA, pp. 426–434.

Kuo, R., Lin, S., Shih, C., 2007. Mining association rules through integration of clustering analysis and ant colony system for health insurance database in taiwan. Expert Systems with Applications 33 (3), 794 – 808.

Kuo, R., Wang, H., Hu, T.-L., Chou, S., 2005. Application of ant k-means on clustering analysis. Computers and Mathematics with Applications 50 (10–12), 1709 – 1724.

Lakiotaki, K., Matsatsinis, N. F., Tsoukias, A., 2011. Multicriteria user modeling in recommender systems. IEEE Intelligent Systems 26 (2), 64–76.

Li, Q., Wang, C., Geng, G., 2008. Improving personalized services in mobile commerce by a novel multicriteria rating approach. In: Proceedings of the 17th International Conference on World Wide Web (WWW 2008). Beijing, China, pp. 1235–1236.

Linden, G., Smith, B., York, J., 2003. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 7 (1), 76–80.

Liu, L., Mehandjiev, N., Xu, D.-L., 2011. Multi-criteria service recommendation based on user criteria preferences. In: Proceedings of the fifth ACM Conference on Recommender Systems (RecSys 2011). Chicago, IL, USA, pp. 77–84.

Ma, J., Theiler, J., Perkins, S., 2003. Accurate on-line support vector regression. Neural Computing 15 (11), 2683–2703.

Manouselis, N., Costopoulou, C., December 2007. Analysis and classification of multi-criteria recommender systems. World Wide Web 10, 415–441.

Manouselis, N., Kyrgiazos, G., Stoitsis, G., 2014. Exploratory study of multi-criteria recommendation algorithms over technology enhanced learning datasets. Journal of e-Learning and Knowledge Society 10 (1), 33–49.

Martin, M., 2002. On-line support vector machine regression. In: Proceedings of the 13th European Conference on Machine Learning. ECML '02. pp. 282–294.

Maulik, U., Bandyopadhyay, S., 2000. Genetic algorithm-based clustering technique. Pattern Recognition 33 (9), 1455 – 1465.

Mikeli, A., Apostolou, D., Despotis, D., 2013a. A multi-criteria recommendation method for interval scaled ratings. In: Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT '13). pp. 9–12.

Mikeli, A., Sotiros, D., Apostolou, D., Despotis, D., July 2013b. A multi-criteria recommender system incorporating intensity of preferences. In: Fourth International Conference on Information, Intelligence, Systems and Applications (IISA '13). pp. 1–6.

Ortega, F., Sánchez, J.-L., Bobadilla, J., Gutiérrez, A., 2013. Improving collaborative filtering-based recommender systems results using pareto dominance. Information Sciences 239, 50 – 61.

Pearson, K., 1901. On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2 (6), 559–572.

Porcel, C., Tejeda-Lorente, A., Martínez, M., Herrera-Viedma, E., 2012. A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. Information Sciences 184 (1), 1 – 19.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., 1994. Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer Supported Cooperative Work. Chapel Hill, NC, United States, pp. 175–186.

Sahoo, N., Krishnan, R., Duncan, G., Callan, J. P., 2006. Collaborative filtering with multi-component rating for recommender systems. In: Proceedings of the Sixteenth Annual Workshop on Information Technologies and Systems (WITS'06). Milwaukee, USA.

Sahoo, N., Krishnan, R., Duncan, G., Callan, J. P., 2012. The Halo Effect in multi-component ratings and its implications for recommender systems: The case of Yahoo! Movies. Information Systems Research 23 (1), 231–246.

Serrano-Guerrero, J., Herrera-Viedma, E., Olivas, J. A., Cerezo, A., Romero, F. P., 2011. A google wave-based fuzzy recommender system to disseminate information in university digital libraries 2.0. Information Sciences 181 (9), 1503 – 1516.

Shambour, Q., Lu, J., 2011. A hybrid multi-criteria semantic-enhanced collaborative filtering approach for personalized recommendations. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT '11). pp. 71–78.

Shani, G., Gunawardana, A., 2011. Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. B. (Eds.), Recommender Systems Handbook. Springer US, pp. 257–297.

Si, L., Jin, R., 2003. Flexible mixture model for collaborative filtering. In: Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003). Washington, DC, pp. 704–711.

Zhang, Y., Zhuang, Y., Wu, J., Zhang, L., 2009. Applying probabilistic latent semantic analysis to multi-criteria recommender system. AI Communications 22 (2), 97–107.