Multi-Criteria Recommender Systems based on Multi-Attribute Decision Making

Conference Paper · December 2013

DOI: 10.1145/2539150.2539176

3 authors:

Ferdaous Hdioud
École Supérieure de Technologie, Fès

Bouchra bouchra Frikh
École Supérieure de Technologie, Universit…

Brahim Ouhbi
Université Moulay Ismail

Multi-Criteria Recommender Systems based on Multi-Attribute Decision Making.

Ferdaous Hdioud
LTTI Lab
Ecole Supérieure de Technologie
B.P. 2427, Route d'imouzer, Fès, Morocco
00212650654540
ferdaous.hdioud@usmba.ac.ma

Bouchra Frikh
LTTI Lab
Ecole Supérieure de Technologie
B.P. 2427, Route d'imouzer, Fès, Morocco
00212661248985
bfrikh@yahoo.com

Brahim Ouhbi
LM2I Lab
ENSAM, Moulay Ismaïl University, Marjane II, B.P. 4024, Meknès, Morocco
00212661116766
ouhbib@yahoo.co.uk

ABSTRACT

The Multi-Criteria Recommender systems continue to be interesting and challenging problem. In this paper we will propose an approach for selection of relevant items in a RS based on multi-criteria ratings and a method of computing weights of criteria taken from Multi-criteria Decision Making (MCDM). This method proposes a correlation coefficient and standard deviation integrated approach for determining weight of criteria in multi-criteria recommender systems. We evaluated the proposed method on an example of movies recommendation. Our approach was compared to some other metrics used in Information Theoretic approach to illustrate its potential applications.

Categories and Subject Descriptors

H.1.1 [**Information Systems**]: Models and Principles Systems and Information Theory [Value of Information]

G.3 [**Mathematic of Computing**]: Probability and Statistics [Statistical Computing]

J.1 [**Computer Application**]: Administrative Data Processing [Business]

I.6.4 [**Computing Methodologies**]: Simulation and modeling Model Validation and Analysis [Value of Information]

General Terms

Algorithms, Measurement, Performance, Design, Reliability, Experimentation.

Keywords

Recommender systems, Collaborative Filtering, Multi-criteria, single-criterion, Multi-criteria Decision Making, weight's attribute.

iiWAS2013, 2-4 December, 2013, Vienna, Austria.

1. INTRODUCTION

Recommender systems (RSs) [3][15] capture interest more and more and became an important area of research, thanks to their help for users to discover potential interesting items easily and quickly. Despite of the overload of information in the internet, RS allows users to accede to a limited amount of items well filtered according to user's preferences, instead of crawling thousands or hundreds of items until finding the most adequate.

RSs are important tools to mine content items and/or collective users' opinions in order to suggest the users the items they might prefer and to provide the users with the information to help them to make a decision. They are more and more commonly used in community based online places such as shopping sites, subscription service sites and online meeting places for various interest groups to recommend items to the community members that they are likely to appreciate.

Despite much of the work that has been done on this topic, several interesting research directions remain unexplored. Readers are referred to the review work [1] for an overview of the recommender system literature and several interesting research ideas.

RSs match a user's attributes to the attributes of the items in the system and identify items that might be of interest to the user. The user for which recommendation is being generated is known as the active user.

RSs work through two phases: prediction and recommendation [28]. In the prediction phase, the rating of an item for a specific user is estimated through a utility function based on this user's past historical ratings, or the content of a particular item, or the user profile, etc. The recommendation phase takes place after predicting the ratings of all the candidate items for the user, where different strategies are used to choose the most suitable items to support the user's decision. However, a broad recommendation process includes both phases.

The most popular classification of recommender systems in the literature is based on how the recommendations are made, i.e., the recommendation approaches. Therefore, recommender systems are usually classified into three categories:

a) Collaborative filtering approaches – recommendation is based on the items liked before by the other people with similar tastes and preferences [5] [19] [20] [30] [31];

b) content-based approaches – recommendation is based on the items that have similar content and characteristics to those the user liked before [16] [17] [24];
c) Hybrid approaches – the different combinations of CF and content-based approaches [8][11][29].

These techniques are considered as classical techniques since they reset only on single-criterion evaluation of items where the user is invited to evaluate items by giving one rating for the item in gross, but in the other side the evaluation process is subject of several separated criteria, as is the case of restaurants, a rating can be shared between for example: food, service, decoration, etc…. Hence, the usefulness of Multi-Criteria RSs.

The RSs use several data analysis techniques to deliver good recommendations for users. In our paper we will take an interest in another extension of RSs –MultiCriteria ones- that permit us studying our data and extract the most significant items while taking inspiration of a Multi-Attribute Decision Making discipline.

This paper is organized as follows: in section 2, we present related works done in the context of recommendation techniques which include the first three parts devoted for the different techniques used on RSs. Section 3 concerns the different Multi-criteria aspects starting by the role of Multi-Criteria in the decision making process, followed by the Multi-criteria RSs. We end this section by establishing analogies between the two fields of decision making and RSs in the context of Multi-Criteria. In the section 4, we focused on our approach. We propose an objective weight determination method called CCSD method for determining the weights of attributes, to provide decision supports to MCRS problems. In section 5, we will present some methods used for selection of relevant items from a large dataset, our application and a discussion of the results. Finally we conclude by the section 6.

2. RELATED WORKS

There are many types of Recommender systems, each one has its different approaches and can be used in different context according to needs. This section will introduce briefly the different approaches used in recommendation.

2.1 Content-Based Recommenders

The recommendation's process using the content-based approach is carried out in the following steps [16] [17] [24]:

1) Item representation: is performed by the module of content analyzer, which use information source coming from items' description, treat them to extract features and finally produce structured items' representations. These last are stored on "Represented items".
2) Learning a user profile : is handled by Profile learner , this module requires two types of inputs:
   - Feedback: this repository serves to construct as update the profile of an active user, his reactions against items are called annotations / feedback and can be used later to improve and up-to-date user's preferences.
     The feedback can be implicit or explicit, the former can be inferred automatically by monitoring user's behaviors and analyze it later, the latter consists in asking the user to express his appreciation towards an item, on various ways:
       - Like / dislike: binary feedback, the user evaluates the items as being relevant or not for him.
       - Ratings: the user assigns a score to an item, based on rating scale offered by the system.
       - Comments: the user expresses his appreciation with a text without imposing on him, the way to do it. This method seems to be the most complicated, as it requires an intelligent system, which can interpret the text and decide if the comment seems to be a positive / negative feedback.
   - Training set $\mathrm{TR_K}$: it's a set of pairs $\langle \mathrm{I_k}, R_k \rangle$ which are extracted from the "represented items repository", where R_k is rating of the item $\mathrm{I_k}$ provided by the active user $\mathrm{u_a}$.

   Feedback and the set of pairs are gathered to be processed by the Profile learner to generate a user profile, which is stored in "Profile repository".
3) Recommendations' generation: when a new item comes, the filtering component module compare item's features with representation of user's profile, to suggest the most likely interesting items $\mathrm{L_a}$ (List of recommended items to the active user).
   The filtering component module can include also the top-ranked items, to $\mathrm{L_a}$, taking into account that user's preferences can change in the time. In this way, the user's profile will be updated automatically, once the $\mathrm{L_a}$ is shown to the user, letting him express his satisfaction against items of $\mathrm{L_a}$. Therefore, the feedback is returned to the profile learner module, in order to execute the learning process again and update the profile for user $\mathrm{u_a}$.

2.2 Collaborative Filtering Approaches

The collaborative recommendation (abbreviated CF) are considered ([19], [5]) as a successful recommendation and widely used for their ease of implementation. The main idea of these approaches is that if certain users shared the same tastes and preferences in the past, they can have the same tendencies to choose similar items in the future. The rationale is to filter items that are likely appreciated by the user from a large set of items, relying to preferences of similar users.

CF approaches [20] [21], don't exploit or require any knowledge about the items themselves as content-based approaches do (e.g. genre, author of the book), but rather rely only on *ratings* of the actual user as well as those of others that are present in the system to infers similarity between either users or items. The advantage of this strategy is that these data do not have to be entered into the system or maintained, also recommended items are various, novel and not with same content as recommendations offered by content-based recommender .

As shown in Figure.1, the sum of ratings gathered from users can be represented as a user × Item matrix, with an entry r_{ui} representing either the rating user *u* gave to item *I*, if he rated it, or *null* otherwise.

$$
\begin{array}{cc}
 & \textbf{\textit{Item}} \\
\textbf{\textit{User}} & \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{bmatrix}
\end{array}
$$

Figure 1. User × Item Matrix (m×n)

The classical CF technique is based mainly on mono-criteria ratings, i.e. the user is invited to mention his opinion on an item by giving *one and only one* rating's value that corresponds to an overall rating given generally to this item without really knowing the reasons behind. The Table.1 below shows what looks like the dataset constructed based on single-criterion CF technique.

Table 1. Example of mono-criteria ratings

	Item 1	Item2	Item 3	Item4
User1	5	7	5	7
User2	5	7	5	7
User3	5	7	5	7
User4	6	6	6	6
User5	6	6	6	6

2.3 Hybrid Recommender Systems

Each approach of the existing recommenders differs from another in terms of inputs required, and has its pros and cons especially in terms of data Sparsity, cold-start problem and ramp-up problem. So, Hybrid systems [22] was appeared to cope with these flaws, by mixing different types of inputs as the aforementioned and combining two or more algorithm's implementations or recommendation components within a *signal* recommender system. In order to benefit from strengths of the various approaches and alleviate their shortcomings, which improves the recommendation's accuracy.

When it consists to hybridize two algorithms or more within a single recommender system, we must take into account two dimensions. The first dimension is the basic recommendation paradigms that define on which the hybrid recommender will be based, and what inputs are required. Concerning the second is Systems' hybridization design [23] [24]*:* that describes the strategy used to combine several recommendation's algorithms or components within the hybrid recommender. Burk's taxonomy [22] distinguishes between seven strategies to combine several recommendation algorithms.

These techniques are considered as classical ones since they reset only on mono-criteria evaluation of items where the user is invited to evaluate items by giving one rating for the item in gross, but in the other side the evaluation process is subject of several separated criteria. Hence, the usefulness of Multi-Criteria RSs. A recent overview on research in multi-criteria RS can be found in [28]. A systematic classification of such multi-criteria recommender systems and, more generally, systems for multi-criteria decision making and optimization is provided by [27]. In the context of more traditional recommender systems, [28] identify the following categories of existing systems which have some form of a multi-criteria nature:

(1) Systems such as classical content-based recommenders, which try to learn content based preferences based, for example, on the given overall ratings for the items. In classical information retrieval (IR) scenarios, the learned user profile for instance consists of a vector containing a relevance-weighted list of the terms appearing in the documents.

(2) Systems for content or item retrieval that allow users to state their *general* preferences using a set of (predefined) categories.

(3) Multi-criteria rating recommenders, in which users are allowed to specify their preferences (ratings) *for individual products* along different dimensions.

Recently, Liu et al. presented a multi-criteria recommendation approach which is based on the clustering of users [26]. Their idea is that for each user one of the criteria is "dominant" and users are grouped according to their criteria preferences [10]. A prediction for a user is then based on the ratings of other users belonging to the same cluster. To determine the importance of the different criteria, they apply linear least squares regression, assign each user to one cluster, and evaluate different schemes for the generation of predictions.

Our aim is to propose a method in which we tried to identify the reasons behind which an item may be deemed relevant to the rest of the items based on the notion of *criteria*. Since the problem is selecting from a large dataset of items, just those the more significant based on their attributes, it returns to resolving a Multi-Attributes Decision making problem [9] which consists on itself to deal with decision problems under the presence of a number of decision criteria.

By applying our method we assume that we can get what is most significant item and why .i.e. of course the results displayed for the user will be easily explained, since their selection is controlled by criteria.

The next section is devoted at first to introduce briefly the Multi-Attributes Decision Making (MADM) on which we rested to formalize our approach; secondly we introduce the Multi-Attributes Recommender Systems as a new type of RSs that begin to attract attention these last times thanks to their utility, compared to the classical RSs, then we end by presenting analogies between the two fields previously introduced.

3. MULTI-CRITERIA ASPECTS

Traditional CF recommenders (based on single-criterion ratings) are characterized in [14] as systems that try to estimate a rating function R: Users × Items → R_0 for predicting a rating for a given user-item pair.

In multi-criteria RS, in contrast, the rating function R has the form Users × Items → $R_0 \times R_1 \times .... \times R_k$. Therefore we have to predict an overall rating R_0 as well as k additional criteria ratings.

The current section presents in details the different multi-criteria aspects, starting by the Multi-Attribute Decision Making field, followed by Multi-criteria RSs, and then analogies between the two fields are established in the last sub-section.

3.1. Multi-Attributes Decision Making

Called also Multi-Criteria Decision Analysis (MCDA) [9] [13] is a discipline, which consists on making choice of the best alternative among a finite set of decision alternatives in terms of multiple usually conflicting criteria (called also attributes). The objective of MCDM is to assist a decision maker in choosing the best alternatives when multiple criteria conflict and compete with each other.

The most commonly used decision aiding methods, such as outranking methods and the analytical hierarchy process, are based on multi-criteria aggregation procedures. Outranking methods determine which alternatives are preferred to others by systematically comparing possible alternatives on each criterion. In fact, we have taken inspiration from that to rank the different items based on their attributes and then select the best ones to display for the new user, as they represent the most relevant items.

In the process of decision making the weights of criteria (attributes) play a very significant role [6]. So, answering the question "How to determine the weights of attributes?" is very crucial in this field of research.

There are different methods have been suggested in the MCDA literature for determination of attributes' weights. The method, on which we rested on, is an ***objective*** weight determination method, which is referred to as ***Correlation Coefficient*** (CC) and ***Standard Deviation*** (SD) integrated approach for determining the weights of attributes, to provide decision supports to MADM problems (see [6]). The detailed approach will be explained in the section dedicated to our approach.

The multi-criteria problems are not intended for personalization and recommendation settings. But these problems find the solutions or *items that are optimal in general* (i.e. optimal with respect to all users), and differences in individual user preferences are not explicitly considered. For these reasons we opted for the choice of working with multi-criteria recommender systems. The next sub-section is dedicated for introducing generally the operating principle of such systems.

3.2 Multi-Criteria Recommender Systems

The vast majority of current recommender systems typically use a single criteria rating to represent the utility of an item to a user in the two-dimensional *Users× Items space*. The recommendation process starts with the specification of the initial set of ratings that is either explicitly provided by the users or is implicitly inferred by the system. Once these initial ratings are specified, a recommender system tries to estimate the rating function *R:*

$$R: Users \times Items \rightarrow R_0 \quad \textit{where } R_0 \textit{ is overall rating}$$

The goal of multi-criteria recommender systems is to find items that maximize each user's utility, just as in the single-rating recommender systems. The difference between single-rating and multi-criteria rating systems is that the latter have more information about the users and items, which can be effectively used in the recommendation process. More formally, the general form of a rating function in a multi-criteria recommender system [4] [5] is: $R: Users \times Items \rightarrow R_0 \times R_1 \times \ldots \times R_k$, where R_0 *is the set of possible overall rating values and* R_k *represents the possible rating values* for each individual criterion *i* $(i = 1, \ldots, k)$. For example in the Movies' recommendation, instead of voting a movie by giving a single global rating, in Yahoo! Movies[1] the user is invited to indicate four additional ratings corresponding respectively to Story, Acting, Visual and Direction as they are the four criteria of evaluation. In the single-rating case the rating function R is estimated based on a sparse user-item rating matrix. In the multi-criteria case, the rating database contains a sparse matrix of the overall ratings, and the detailed criteria ratings of the user community.

Table 2. Example of multi-criteria ratings (four criteria)

	Item 1	Item2	Item 3	Item4
User1	$5_{2,2,8,8}$	$7_{5,5,9,9}$	$5_{2,2,8,8}$	$7_{5,5,9,9}$
User2	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$5_{8,8,2,2}$	$7_{9,9,5,5}$
User3	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$5_{8,8,2,2}$	$7_{9,9,5,5}$
User4	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$6_{3,3,9,9}$	$6_{4,4,8,8}$
User5	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$6_{3,3,9,9}$	$6_{4,4,8,8}$

The Table 2 above shows the same example presented on Table 1, but at this time items are evaluated according four criteria, these multi-criteria rating are aggregated as an average to get the overall rating [5]. However, in single-criteria RSs, this information would be *"hidden"* within the aggregate overall rating, which may lead to *inaccurate* insights about the true similarity between user preferences. In the contrary of multi-criteria RSs, these information is better identified which permits to detect connections between either users or items.

In [5] two basic schemes called similarity based and aggregation function based are proposed. In the similarity based approach, the similarity between users is determined based on their past rating behavior and a measure such as the Pearson correlation coefficient. Different potential ways were proposed to measure the ways were proposed to measure the similarity between users based on their detailed ratings. The user's ratings for an item can for example be viewed as *k*-dimensional vectors so that standard multidimensional distance metrics such as the Euclidian or the Chebyshev distance can be calculated. Alternatively, one could calculate the Pearson correlation coefficient for each rating dimension individually and take the average or smallest value as an overall similarity metric. For computing the final rating prediction, a standard nearest-neighbor approach can be used. The only difference between a single-rating approach and similarity based multi-criteria recommendation approaches therefore lies in the usage of a different similarity metric. In the aggregation function based approach the general and intuitive assumption is that there exists a relationship between the overall item ratings and the individual criteria ratings.

Technically, the overall rating r_0 can therefore be seen as being determined by a function f of individual criteria ratings: $r_0 = f(r_1, \ldots, r_k)$. The prediction of r_0 for a given user u and a target item I can be accomplished in a multi-step process. First, in an offline phase, the function f *has* to be determined. One option could be to define f based on domain expertise or by averaging the criteria ratings. A more promising approach, however, is to apply statistical or machine learning techniques to automatically detect the hidden relationship between the overall rating and the criteria ratings. For example, in [5] it is proposed to approximate a function f for each item in the catalog using multiple linear regression techniques. In the online phase, first the criteria ratings r_1 to r_k for I have to be estimated. Afterwards, the overall rating

[1] http://movies.yahoo.com/

can be calculated using *f*. For estimating r_1 to r_k, any standard CF algorithm can be used separately for each criterion.

3.3 MADM vs. MCRS

In the previous sub-sections devoted for Multi-criteria aspects, a Multi-criteria Recommender Systems (MCRS) was presented as based mainly on multi-criteria ratings assigned to items according their attributes, So each item receive a rating's value per attribute. As our problem is summarized in the fact of finding the most interesting items, thus the problem is none other than a simple Multi-Criteria Decision Making (MCDM) problem, on which decision alternatives are simply items that are selected based on criteria (attributes on MCDM).

In this case the multi-criteria ratings matrix will be considered as a Decision matrix X cited on the previous sub-section, which will help us to compute weights of both attributes and items.

Table 3. Analogies between MCDM and MCRS

MCDM	MCRS
Finding best decision alternatives	Finding the relevant items
Decision alternatives	Items
Criteria	items attributes
Decision matrix X	multi-criteria rating matrix
Xij the performance value	the rating value according a criteria

The Table 3 above shows analogies between MCDM and MCRS that we had establish, in order to facilitate the passage between the two fields.

In the next section our approach will be detailed setting on what was explained on the current sub-section and its previous.

4. OUR APPROACH

The motivation of this paper is twofold. First, the proposed method as an objective weight determination method provides a methodological choice for the RSs and second it allows us to take into account the dependencies between attributes.

The two basic schemes proposed in [14] called *similarity based* and *aggregation function based* don't take into account intrinsic properties of items based on the attributes correlations. In this section we propose an objective weight determination method called CCSD method, which is referred to as correlation coefficient (CC) and standard deviation (SD) integrated approach for determining the weights of attributes, to provide decision supports to MCRS problems. The CCSD method determines the weights of attributes by integrating SD of each attribute with their correlation coefficients (CCs) with the overall assessment of items. The CCs are determined by removing one attribute at a time from the set of attributes and considering its correlation with the overall assessment of items without the inclusion of this removed attribute. If the CC for this removed attribute turns out to be very high, then the removal of this attribute has little effect on recommender system; otherwise, this removed attribute should be given an important weight.

4.1. Extracting the Decision Matrix

An MCRS problem can be easily expressed in matrix format. A decision matrix A is an (m × k) matrix in which element xij indicates the performance of item I_i when it is evaluated in terms of decision criterion Cj, (for i =1,2,3,..., m, and j = 1,2,3,..., k). Let $u_1, \ldots, u_N$ be N users who have evaluated m items based on k criteria. As each item is evaluated by different users according the k attributes, the performance value x_{ij} is the average of all rating's values given to the item i according to attribute j by user who voted for.

$$x_{ij} = \frac{\sum_l (r_{li})_j}{N}$$

$(r_{li})_j$: The rating given by user l to item i according attribute j.

4.2. Computing Attributes' Weights

Lets m decision alternatives $I_1, \ldots, I_m$ to be evaluated in terms of k attributes $C_1, \ldots, C_k$ which forms a decision matrix denoted by $X=(x_{ij})_{m,k}$, where x_{ij} is the performance value of I_i with respect to C_j. Let W= $(w_1,\ldots,w_k)$ be a normalized weights' matrix in such way that $\sum w_j=1$ where w_j is the weight of the attribute C_j.

The overall assessment value of each item is computed as follows:

$$d_i = \sum_{j=1}^{k} x_{ij} w_j \,.\, i = 1, \ldots m$$

The bigger the overall assessment value, the better the decision alternative. The best item is the one with the biggest overall assessment value.

By removing criteria C_j from the set of criteria, we define the overall assessment value of each item as:

$$d_{ij} = \sum_{\substack{l=1 \\ l \neq j}}^{k} x_{il} w_l \,.\, i = 1, \ldots m$$

The coefficient correlation (CC) between the values of C_j and the above overall assessment values can be expressed by:

$$R_j = \frac{\sum_{i=1}^{m}(x_{ij} - \overline{x_j})(d_{ij} - \overline{d_j})}{\sqrt{\sum_{i=1}^{m}(x_{ij} - \overline{x_j})^2 \sum_{i=1}^{m}(d_{ij} - \overline{d_j})^2}},$$

Where $\overline{x_j}$ and $\overline{d_j}$ are respectively the mean values of x_{ij} andd_{ij}, for i=1,…,m.

If R_j is high enough and close to one then the criteria C_j has little effect on decision recommendation. If R_j is very low then C_j has a significant impact on decision recommendation and items ranking. So, the criteria C_j should be given a very important weight. Based on the above steps, the weight of an attribute is computed as:

$$w_j = \frac{\sigma_j \sqrt{1-R_j}}{\sum_{l=1}^{k} \sigma_l \sqrt{1-R_l}} \quad \text{, j=1,….k.}$$

Where the SD is calculated by:

$$\sigma_j = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (z_{ij} - \overline{z_j})^2} \text{ , j=1,….k.}$$

5. RESULTS AND DISCUSSION

The current sub-sections will introduce our proposed approach step by step, accompanied by an example of movies recommendation (see Table 4). Our aim is to propose a method in which we tried to identify the reasons behind which an item may be deemed relevant to the rest of the items based on the notion of *criteria*. Since the problem is selecting from a large dataset of items, just those the more significant based on their attributes, it returns to resolving a Multi-Attributes Decision making problem [9] which consists on itself to deal with decision problems under the presence of a number of decision criteria.

5.1. Selection Measures of Relevant Items

The high availability and variety of information instead of being benefic may drives users to make poor decisions when it comes to choose the best items from a large dataset. The role of RSs is essentially to assist the user during his discovery of products, while showing him the best of them. This assistance begins as soon the user arrived and continued until the user is in contact with the system.

As the first occupation of the RS is to satisfy its users and reducing their efforts to find relevant items from the first moment they came, there are many strategies (see [7]) which had for aim to select a more significant items according to a number of item selection measures based on information theoretic, to learn user's profile effectively. Among these measures we mention:

Popularity: this measure indicates how frequently users rated the item. Popular items are those familiar ones that receive the most ratings. The advantage of the popularity is that it is very easy and inexpensive to compute, but use it to elicit preferences worsening the prefix bias that is, popular items garnering even more evaluations. Unpopular items, lacking enough user opinions, may be hard to recommend. The popularity of the item *i* is equal for all the users, and it is the number of not null ratings for *i* from the whole number of users.

Contention: The rational is that items on which we ask the user to rate should not only be familiar to the users, but also indicative of their tendencies. The contention is negatively correlated with popularity. i.e. while popular items are less controversial. The Two common measures to quantify the contention of an item are the *variance* and *entropy* of its ratings.

$$entropy\,(item) = -\sum_i p_i \lg(p_i)\,.$$

Where p_i denotes the fraction of item's ratings that equals to i.

Coverage: Items are useful when they possess predictive power on other items.

$$coverage(i) = \sum_j n_{ij}$$

Where n_{ij} is the number of users that rated both items i and j.

Note that it was proved that is unwise to maximize the Contention as it's insufficient to repose only on Popularity, for this it's better to combine them both, as is the case for *enropy0* and *HELF (*for more details see [16]).

$$entropy0\,(item) = -\frac{1}{\sum_i w_i}\sum_{i=0}^{5} p_i\ w_i \lg(p_i)$$

We treat the missing evaluations as a separate category of evaluation, for example, a rating value of 0 in the datasets we use, since 1-5 is the usual scale; and fill all the missing evaluations with this new rating category. w_i Denotes the weights of the rating's value i in the 1-5 scale.

The strategy of items' selection that is based on these measures suffers from two essential issues, which are Arbitrariness of selection which means that the criteria mentioned are detached from any reasonable end goal of optimizing user experience and the independence of selected items all above criteria score the utility of each single item independently, and consequently select those of highest utility. No consideration is given to how all these items play together, they are treated as if they are independent and no relationship is binding upon each other.

These classical methods presented above will be served as baseline methods with which we will compare our results, in the following text we present the steps taken in our approach.

5.2.Movies' Recommendation Example

Our proposed approach will be applied on an example of movies recommendation depicted in Table 4 above. Then, obtained results will be compared to baseline methods shown in previous sub-section.

Table 4. Example of multi-criteria movie recommender system (ratings for each item: overall, story, acting, direction, visual)

	Wanted	Wall-E	Star Wars	Seven	Fargo
Alice	$5_{2,2,8,8}$	$7_{5,5,9,9}$	$5_{2,2,8,8}$	$7_{5,5,9,9}$	*Null*
John	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$5_{8,8,2,2}$	$7_{8,8,2,2}$	$9_{8,8,10,10}$
Mason	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$5_{2,2,8,8}$

Step 1. The multi-criteria ratings shown in the table above must be converted to construct the decision matrix X. As each item is evaluated by different user according the four attributes, the performance value X_{ij} is the average of all rating's values given to the item i according to attribute j by user who voted for.

$$X_{ij} = \frac{\sum_k (r_{ki})_j}{N}, k = 1, \dots, m$$

$(r_{ki})_j$: The rating given by user k to item i according attribute j.

Step 2. Computing the weights of the four attributes as explained in the sub-section 4.1 based on CCSD integrated. There are resulting weights in our example:

Acting: 0.2676545	Story: 0.23234549
Visual: 0.23234549	Direction: 0.2676545

We can see the two criteria acting and direction and most interesting (have a big impact in relevance of a movie) compared to the two others which make a sense, because as a user, our choice at first to movies are always influenced by the actors and theater director (that we already know), since we have not yet seen the movie.

Step 3. Computing the weights of each Movie as an overall assessment based on attributes' weights founded in the previous step. Obviously the bigger a weight is, the more significant movie is. The resulting weights of movies are as follows:

Table 5. Weights of movies based on CCSD integrated method

Wanted	Wall-E	Star Wars	Seven	Fargo
5.4039483	6.713744	5.4039483	6.0235395	7.1412363

The table shows that Fargo Movie is the most relevant and both Wanted and Star Wars are less interesting. These results will be compared to the baseline methods (Popularity, Contention, Coverage), to see if there is a change in the order of relevance of movies. The Table 6 shows the score of movies for the baseline methods:

Table 6. Comparison of results with baseline methods

	Wanted	Wall-E	Star Wars	Seven	Fargo
Popularity	3	3	3	3	2
Entropy	0.27031004	0.0	0.27031004	0.3662041	0.3465736
Entropy0	0.24923107	0.16126716	0.24923107	0.23852018	0.212111
Coverage	11.0	11.0	11.0	11.0	8.0

Based on the Table 6 above we established The Table 7 that shows the movies in descending order according to the values resulted:

Table 7. Movies' order according to different methods

CCSD method	Popularity	Entropy	Entropy0	Coverage
Fargo	Wanted	seven	Wanted	Wanted
Wall-E	Wall-E	Fargo	Star Wars	Wall-E
Seven	Star Wars	Wanted	Seven	Star Wars
Wanted	Seven	Star Wars	Fargo	Seven
Star Wars	Fargo	Wall-E	Wall-E	Fargo

We can notice that the order of Movies' set presented by our approach is very varied, as in the first position we find Fargo movie which is less popular, more controversial (2nd position for entropy) and have a low coverage's value (non-correlated with other movie and haven't a predictive power on them), we can deduce in this case that popular items aren't usually liked by everyone since there are users whose preferences are specific and differ from the community. Also we see that our method is in alliance with both Popularity and Coverage when it comes to the second position, i.e. Wall-E movie and with Entropy0 –which combine the notion of popularity and contention- concerning the Third position, i.e. Seven Movie.

6. CONCLUSION

As recommendation systems attract increasing research interests in last times, many practical applications start to adopt data analysis techniques to derive recommendations for their users. In this paper, we have extended the concept of mono-criterion ratings to multi-criteria ones to meet the requirements of more practical recommendation systems. We proposed an approach for selection of relevant items in a RS based on multi-criteria ratings and a method of computing weights of criteria. This method proposes a correlation coefficient and standard deviation integrated approach for determining weight of criteria in multi-criteria recommender systems. We evaluated the proposed method on an example of movies recommendation. Our approach was compared to some other metrics used in Information Theoretic approach to illustrate its potential applications. It was seen that our approach has shown a variety on selected items' set that is considered interesting from our point of view.

7. REFERENCES

[1] Gediminas, A. Adomavicius and Tuzhilin, "Towards the Next Generation of Recommender Systems:A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, p. 43, 2005.

[2] Nava, J. Tintarev and Masthoff, "A Survey of Explanations in Recommender Systems," *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, p. 10, 2007.

[3] Joseph, J. A. Konstan and Riedl, "Recommender systems: from algorithms to user experience," *User Modeling and User-Adapted Interaction*, p. 23, 2010.

[4] A. Gediminas, M. Nikos and K. YoungOk, "Multi-Criteria Recommender Systems," p. 34, 2010.

[5] T. Wei-Guang and L. Hsin-Hsien, "Collaborative Recommendation with Multi-Criteria Ratings," 2007.

[6] W. Ying-Ming and L. Ying, "Integration of correlations with standard deviations for determining attribute weights in multiple attribute decision making," *Mathematical and Computer Modelling*, p. 12, 2010.

[7] R. Al Mamunur, K. George and R. John, "Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach," *SIGKDD Workshop on Web Mining and Web Usage Analysis (WEBKDD)*, p. 21, 2008.

[8] M. Nikos and C. Constantina, "Analysis and Classification of Multi-Criteria Recommender Systems," p. 27, 2007.

[9] B. S. S. N. S. a. T. R. E. Triantaphyllou, "Multi-Criteria Decision Making: An Operations Research Approach," *Encyclopedia of Electrical and Electronics Engineering*, p. 22, 1998.

[10] DIETMAR, Z. JANNACH and F. KARAKAYA, "Accuracy Improvements for Multi-criteria Recommender Systems," *EC '12 Proceedings of the 13th ACM Conference on Electronic Commerce*, p. 16, 2012.

[11] N. Sahoo, "Experiments with Multi-component Rating Collaborative Filtering for Improved Recommendation," p. 34.

[12] Kleanthi, N. Lakiotaki1, A. F. Matsatsinis and Tsoukiàs,

"Multi-Criteria User Modeling in Recommender Systems," *Intelligent Systems, IEEE* , p. 26, 2011.

[13] Gilberto, A. Montibeller and Franco, "Multi-Criteria Decision Analysis for Strategic Decision Making," *Springer: Handbook of Multi-Criteria Analysis,* 2010.

[14] Gediminas, Y. Adomavicius and Kwon, "New Recommendation Techniques for Multi-Criteria Rating Systems," *Intelligent Systems, IEEE,* p. 28, 2007.

[15] P. R. a. H. R. Varian, "Recommender Systems," *Communications of the ACM,* vol. Vol.40, p. 2, march 1997.

[16] Y. P.-a. J. Blanco-Fernandez, A. Gil-Solla, M. Ramos-Cabrer and M. Lopez-Nores, "Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender systems," *IEEE Transactions on Consumer Electronics 54,* Vols. 727-735, 2008.

[17] Iván, M. Cantador, H. Szomszor, M. Alani, P. Fernández and Castells, "Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations," *In Proceedings of the 1st International Workshop on Collective Semantics: Collective Intelligence and the Semantic Web (CISWeb 2008),*, 2008.

[18] C. man, N. Au Yeung, N. Gibbins and Shadbolt., "A Study of User Profile Generation from Folksonomies," *Workshop Proceedings, CEUR-WS.org,* 2008.

[19] Robert, Y. M. Bell and Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights," *ICDM '07 Proceedings of the 2007 Seventh IEEE International Conference on Data Mining,* 2007.

[20] Jun, A. Wang, M. P. de Vries and J. T. Reinders, "Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion," *SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval,* 2006.

[21] Andreas, M. Töscher, R. Jahrer and Legenstein, "Improved Neighborhood-Based Algorithms for Large-Scale Recommender Systems," *NETFLIX '08 Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition,* 2008.

[22] R. Burke, "Hybrid web recommender systems," *The Adaptive Web: Methods and Strategies of Web Personalization,springer,* 2007.

[23] M. Zanker and M., "Collaborative feature-combination recommender Exploiting Explicit and Implicit User Feedback," *Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing (CEC '09) (Vienna), IEEE Computer society,*, 2009.

[24] Prem, R. Melville, R. J. Mooney and Nagarajan, "Content-Boosted Collaborative Filtering for improved recommendation," *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI) (Edmonton, Alberta, Canada),* 2002.

[25] M. Nikos, Constantina and Costopoulou, "Analysis and Classification of Multi-Criteria Recommender Systems," *World Wide Web 10(4,* pp. 415-441, 2007.

[26] L. Liu, M. Nikolay and X. Dong-Ling, "Multi-criteria service recommendation based on user criteria," *In Proceedings of the fifth ACM Conference on Recommender Systems (RecSys '11),* pp. 77-84, 2011.

[27] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction,* pp. 331 - 370, November 2002.

[28] Gediminas, R. Adomavicius, S. Sankaranarayanan, A. Sen and Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems (TOIS),* pp. 103 - 145, 2005.

[29] Gediminas and N. M. a. Y. K. Adomavicius, "Multi-Criteria Recommender Systems," *RECOMMENDER SYSTEMS HANDBOOK, Springer,* pp. 769-803, 2011.

[30] Kenneth and Karta, "An Investigation on Personalized Collaborative Filtering for Web Service Selection," 2005.

[31] Uwe, H. Leimstoll and Stormer, "Collaborative Recommender Systems for Online Shops.," *In proceeding of: Reaching New Heights. 13th Americas Conference on Information Systems, AMCIS 2007, Keystone, Colorado, USA,* 2007.