Multi-Criteria Based Recommender System Scalability Optimization: The Approach Based on Clustering of Users

Sheetal R. Thakare¹
¹Department of Computer Engineering
¹University of Mumbai, India

Abstract— The era of Internet & the ever blooming E-Commerce applications have exposed people to the mines of information and variety of products to choose from while shopping on-line. In such situation it is quite obvious to get lost in exploring abundance of products and related information. This causes the waste of large amount of precious time and may at the same time, create confusion in the minds of prospective on-line shoppers regarding purchasing decision to be made, leading to in-vein search and unfruitful buying attempt. This is categorized as the information overload problem. Recommender systems play an important role in such scenario by presenting the buyer with the ranked list of products preferred by existing buyers for the same/similar products. The number of features or attributes of the product used to determine the order of the product in the ranked list will define the approach used by Recommender system. Traditional approach is called as single criterion based approach for recommender system while the other one is multi-criteria based approach for recommender system. As the number of users of the e-commerce site increases, the time required to form ranked list will also increase with the great amount in case of multi-criteria based recommender systems. This requires optimization criteria to be applied to minimize the time required to form ranked list while maintaining the accuracy of the result.

Key words: Recommender System, Criterion, Multi-Criteria Based, Content-Based Filtering, Collaborative Filtering, Personalization, Prediction

I. INTRODUCTION

The tremendous growth of customers and products in recent years poses some key challenges for recommender systems. These are: producing high quality recommendations and performing many recommendations per second for millions of customers and products. Traditional recommendation techniques in recommender systems mainly focus on improving recommendation accuracy. However, personalized recommendation, which considers the multiple needs of users and can make both accurate and diverse recommendations, is more suitable for modern recommender systems[1]. Almost all decisions people take are based on multiple factors or criteria. Decision makers generally pursue multiple, and often conflicting, objectives. A feasible solution is the one that is optimum with respect to all such objectives[6]. Multi-Criteria Recommender Systems (MCRS) can be defined as Recommender Systems that incorporate preference information upon multiple criteria. Instead of developing recommendation techniques based on a single criterion values, using the overall preference of user u for the item i, these systems try to predict a rating for unexplored items of u by exploiting preference information on multiple criteria that affect this overall preference value. Several researchers approach MCRS as a Multi-criteria Decision Making (MCDM) problem, and apply MCDM methods and techniques to implement MCRS systems. As almost all decisions people make are based on multiple factors or criteria[2] there is a need for multi criteria recommender system. Most of the existing recommender systems, based on collaborative filtering, content-based filtering, hybrid techniques and demographic recommendation consider single criterion for recommendation. In recent work, this use of single criterion has been considered as limited [3], because the suitability of the recommended item for a particular user may depend on more than one utility-related aspect that the user takes into consideration when making the choice. Particularly in systems where recommendations are based on the opinions of others, the incorporation of multiple criteria that can affect the users’ opinions may lead to more accurate recommendations. For example, in a movie recommender system, two users A and B both assign a single-criterion rating of 12 (out of 13) for Avatar. The recommender systems will conclude they have the same tastes even if A likes its story and B likes its visuals. This is called a “without distinction of interest” problem. Furthermore, even if both users like the same movie features (e.g. actors, visuals, etc.), they might select different movies. This is because people usually select a movie based on different movie features. This situation is referred to as an “unsuitable weight feature” problem[26]. To solve such type of problems there is a need for multi-criteria based recommender systems. The Collaborative Filtering technique has been successfully applied to solve the recommendation problem with single criterion. When the problem is extended to be multiple conflicting criteria, new techniques are needed in order to effectively incorporate the multi-criteria rating information into the recommendation process.

II. BACKGROUND

Personalization technologies and recommender systems help on-line consumers avoid information overload by making suggestions regarding which information is most relevant to them. Most on-line shopping sites and many other applications now use recommender systems. The most popular examples include Netflix, which recommends movies, and Amazon.com which recommends books, CDs, and various other products. Users offer feedback on purchased or consumed items, and the recommender system uses the information to predict their preferences for yet unseen items and subsequently recommends items with the highest predicted relevance[1]. Traditional recommender systems based on a single criterion, usually presentsa numerical rating that represents user’s preference of the whole item. Two types of entities, users and items are used for the recommendation, this gives it its two classical dimensions. This is presented as[2]
R : Users × Items → Ratings.

The system is initialized by user’s ratings that are either explicitly or implicitly collected. Then it tries to estimate the utility function \( R \) of the item based on these two dimensions, user and items:

\[ \forall u \in \text{Users}, i_u' = \arg \max_{i \in \text{Items}} R(u, i) \]

Only the items which can maximize the utility will be chosen [2].

**Fig. 2.1: Single Criterion Based Filtering [1]**

Recommender systems can be based on content-based filtering, collaborative filtering and hybrid filtering [2][3]. Content-based recommendation suggests items to users that are similar to those items that they were interested previously. Collaborative filtering (CF) recommends items based on the information about items bought by similar users[4]. In summary, both basic recommendation approaches rely on similarity measure. The content-based approach relies on the similarity of items, whereas the collaborative filtering approach relies on the similarity of users. Most of the traditional recommender systems are developed by using either one of two basic approaches. However, there were several good research papers in the literature that proposed the techniques to combine both basic approaches together in order to gain advantages of both approaches and improve the performance of the recommender system. They are known as a “Hybrid approaches”. Hybrid approaches combine content-based and collaborative-filtering methods in several different ways to find overall item ratings[1]. But in case of multi-criteria ratings, in addition to the overall rating, information about user preferences for different aspects or components of an item is also provided. Thus difference between single-rating and multi-criteria rating systems is that the latter have more information about the users and items to use in the recommendation process. More formally, the general form of a rating function in a multi-criteria recommender system is

\[ R : \text{Users} \times \text{Items} \rightarrow R_0 \times R_1 \times \ldots \times R_k \]

Where \( R_0 \) is the set of possible overall rating values, and \( R_i \) represents the possible rating values for each individual criterion \( i (i = 1, \ldots, k) \).

Typically on some numeric scale (for example, from 1 to 13), Recommender systems should benefit from leveraging this additional information because it can potentially increase the recommendation accuracy[1].

**Fig. 2.2: Multi-Criteria Based Filtering [1], Where The Overall Rating For Each Movie Is A Simple Average Of Four Rating Criteria: Story, Acting, Direction And Visuals.**

### III. Motivation

Existing recommender systems use information from users’ profiles (demographic filtering), similar neighbors (collaborative filtering), and textual description (content-based model) to make recommendations, which easily generate irrelevant suggestions to users due to the ignorance of users’ intentions and the visual similarity among products[20]. The exponential rise in the number of customers and products in recent years has made it difficult to perform many recommendations per second for millions of customers and products. So new recommender system technologies are needed that can quickly produce high quality recommendations, even for very large-scale problems. This issue can be addressed by scaling up the neighborhood formation process through the use of clustering techniques[18]. First challenge for large scale recommender system, used in E-commerce applications to suggest users the products to buy from, is to improve the scalability of the collaborative filtering algorithms. These algorithms are able to search tens of thousands of potential neighbors in real-time, but the demands of modern E-commerce systems are to search tens of millions of potential neighbors. Further, existing algorithms have performance problems with individual consumers for whom the site has large amounts of information. For instance, if a site is using browsing patterns as indications of product preference, it may have thousands of data points for its most valuable customers. These “long customer rows” slow down the number of neighbors that can be searched per second, further reducing scalability. The second challenge is to improve the quality of the recommendations for the consumers. Consumers need recommendations they can trust to help them find products they will like. If a consumer trusts a recommender system, purchases a product, and finds out he does not like the product, the consumer will be unlikely to use the recommender system again. In some ways these two challenges are in conflict, since the less time an algorithm spends searching for neighbors, the more scalable it will be, and the worse its quality. For this reason, it is important to treat the two challenges simultaneously so the solutions discovered are both useful and practical.
IV. CLUSTERING METHODS

The clustering problem is the ordering of a set of data into groups, based on one or more features of the data. Cluster analysis is an unsupervised learning method that constitutes a main role of an intelligent data analysis process. The relationship among members of the clusters is often expressed as similarity or dissimilarity measurement and is calculated through distance function.

A. Distance Functions:

It is useful to denote the distance between two instances $x_i$ and $x_j$ as $d(x_i, x_j)$. A valid distance measure should be symmetric and obtains its minimum value (usually zero) in case of identical vectors. The distance measure is called a metric distance measure if it also satisfies the following properties:

1) Triangle inequality $d(x_i, x_k) \leq d(x_i, x_j) + d(x_j, x_k)$  
   $\forall x_i, x_j, x_k \in S$.

2) $d(x_i, x_j) = 0 \iff x_i = x_j \forall x_i, x_j \in S$.


B. Similarity Functions:

An alternative concept to that of the distance is the similarity function $s(x_i, x_j)$ that compares the two vectors $x_i$ and $x_j$. This function should be symmetrical (namely $s(x_i, x_j) = s(x_j, x_i)$) and have a large value when $x_i$ and $x_j$ are somehow “similar” and constitute the largest value for identical vectors. A similarity function where the target range is $[0, 1]$ is called a dichotomous similarity function. Examples of this functions are:

1) Cosine Measure:

$$s(x_i, x_j) = \frac{x_i^T \cdot x_j}{\|x_i\| \cdot \|x_j\|}$$

2) Pearson Correlation Measure:

$$s(x_i, x_j) = \frac{(x_i - \bar{x}_i) \cdot (x_j - \bar{x}_j)}{\|x_i - \bar{x}_i\| \cdot \|x_j - \bar{x}_j\|}$$

3) Extended Jaccard Measure

$$s(x_i, x_j) = \frac{x_i^T \cdot x_j}{\|x_i\|^2 + \|x_j\|^2 - x_i^T \cdot x_j}$$

4) Dice Coefficient Measure:

$$s(x_i, x_j) = \frac{2x_i^T \cdot x_j}{\|x_i\|^2 + \|x_j\|^2}$$

C. Clustering using Neighborhood Formation:

Clustering is useful technique for the discovery of data distribution and patterns in the underlying data[19]. Most collaborative filtering based recommender systems build a neighborhood of like minded customers. The Neighborhood formation scheme usually uses Pearson correlation or cosine similarity as a measure of proximity [18]. The neighborhood formation process is in fact the model-building or learning process for a recommender system algorithm. The main goal of neighborhood formation is to find, for each customer $C$, an ordered list of $k$ customers $N = \{N1, N2, \ldots, Nk\}$ such that $C \in N$ and $\text{sim}(C, N1)$ is maximum, $\text{sim}(C, N2)$ is the next maximum and so on. Where $\text{sim}(C, Ni)$ indicates similarity between two customers, which is most often computed by finding the Pearson correlation between the customers $C$ and $Ni$. The normalized Pearson correlation is defined as:

$$\text{where } x_i \text{ denotes the average feature value of } x \text{ over all dimensions}[21].$$

As Nearest neighbor algorithms rely upon exact matches that cause the algorithms to sacrifice recommender system coverage and accuracy. In particular, since the correlation coefficient is only defined between customers who have rated at least two products in common, many pairs of customers have no correlation at all [21]. Accordingly, Pearson nearest neighbor algorithms may be unable to make many product recommendations for a particular user. This problem is known as reduced coverage, and is due to sparse ratings of neighbors. Nearest neighbor algorithms require computation that grows with both the number of customers and the number of products. With millions of customers and products, a typical web based recommender system running existing algorithms will suffer serious scalability problems. The application of clustering techniques reduces the sparsity and improves scalability of recommender systems. Clustering of users can effectively partition the ratings database and thereby improve the scalability and sparsity[18].

D. Clustering-Approaches:

Han and Kamber (2001) suggest categorizing the methods into additional three main categories: density-based methods, model-based clustering and grid-based methods. An alternative categorization based on the induction principle of the various clustering methods is presented in (Estivill-Castro, 2000) stating two more categories as hierarchy-based and partition-based clustering. Soft-computing methods are also useful in clustering tasks and include i.Fuzzy Clustering ii.Evolutionary Approaches for Clustering iii.Simulated Annealing for Clustering.
A majority of the approaches and algorithms proposed in the literature cannot handle large data sets. Approaches based on genetic algorithms, tabu search and simulated annealing are optimization techniques and are restricted to reasonably small data sets. Implementations of conceptual clustering optimize some criterion instead of storing them.

The convergent K-means algorithm and its ANN equivalent, the Kohonen net, have been used to cluster large data sets. The reasons behind the popularity of the K-means algorithm are: 1. Its time complexity is $O(mkl)$, where $m$ is the number of instances; $k$ is the number of clusters; and $l$ is the number of iterations taken by the algorithm to converge. Typically, $k$ and $l$ are fixed in advance and so the algorithm has linear time complexity in the size of the data set. 2. Its space complexity is $O(k+m)$. It requires additional space to store the data matrix. It is possible to store the data matrix in a secondary memory and access each pattern based on need. However, this scheme requires a huge access time because of the iterative nature of the algorithm. As a consequence, processing time increases enormously. 3. It is order-independent. For a given initial seed set of cluster centers, it generates the same partition of the data irrespective of the order in which the patterns are presented to the algorithm. However, the K-means algorithm is sensitive to initial seed selection and even in the best case, it can produce only hyper-spherical clusters. Genetic algorithms (GAs) are a class of adaptive stochastic optimization algorithms. GAs have been successfully applied to partitioning problems and in particular to clustering problems. For instance, Zhang et al. [25] used a genetic clustering method to find a globally optimal partition and embed it in the CF process. Experimental results showed that the genetic clustering CF approach outperformed the traditional CF approach in scalability and outperformed the traditional k-means and the traditional CF approach in prediction quality[24]. Hierarchical algorithms are more versatile. But they have the following disadvantages: 1. The time complexity of hierarchical agglomerative algorithms is $O(m^2 \log m)$. 2. The space complexity of agglomerative algorithms is $O(m^2)$. This is because a similarity matrix of size $m^2$ has to be stored. It is possible to compute the entries of this matrix based on need functions and are typically computationally expensive.

Incremental clustering is based on the assumption that it is possible to con-sider instances one at a time and assign them to existing clusters. Here, a new instance is assigned to a cluster without significantly affecting the existing clusters. Only the cluster representations are stored in the main memory to alleviate the space limitations.

V. CONCLUSION

In single criterion based Recommender Systems, single ratings provided by users provide no information regarding the reason behind the user’s preference. So evaluating Recommender Systems on multiple criteria provides accuracy in results of recommendation. Moreover with increasing number of existing users and items, traditional CF algorithms will suffer serious scalability problems. To remove the scalability problems, cluster-based approach is used in multi-criteria recommender systems, which integrates the clustering algorithm into the multi-criteria CF process. The clustering algorithm finds the optimal partition of user clusters. These clusters provide a reliable neighborhood for the CF process to limit the neighborhood search to a single group, rather than search the entire user base. This reduces response time of Recommender Systems to produce a product recommendation list for users.

REFERENCES


Multi-Criteria Based Recommender System Scalability Optimization: The Approach Based on Clustering of Users

[10] Rafael S. Parpinelli, Heitor S. Lopes, Member, IEEE, and Alex A. Freitas, Data Mining With an Ant Colony Optimization Algorithm, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTING, VOL. 6, NO. 4, AUGUST 2002


[16] Pakapon Tangphoklang, Chaiyasit Tanchotsrinon, Saranya Maneeroj, and Peraphon Sophatsathit, A Design of Multi-Criteria Recommender System Architecture for Mobile Banking Business in Thailand, Department of Mathematic, Faculty of Science, Chulalongkorn University Bangkok, 10330, Thailand


[18] Badrul M. Sarwar, George Karypis, Joseph Konstan, John Riedl, Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering, Currently with the Computer Science Department, SanJose State University, San Jose, CA 95112, USA. Email:sarwar@cs.sjsu.edu, Phone: +1 408-245-8202


[21] Lior Rokach, Oded Maimon, Clustering Methods, Data Mining And Knowledge Discovery Handbook, Chapter 15


