Research Article | **Impact Factor: 0.621** | **ISSN: 2319-507X**
Kalyani Sable, IJPRET, 2014; Volume 2 (9): 1002-1010 | **IJPRET**

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

RECOMMENDATION ALGORITHM: ITEM-BASED COLLABORATIVE FILTERING

MS. KALYANI P. SABLE[1], DR. H. R. DESHMUKH[2], PROF. N. S. BAND[3], PROF. R. N. GADBAIL[3]

1. M. E. First Year, Department of Computer Science & Engineering, IBSS College of Engineering, Amravati.
2. Prof and HOD, Department of Computer Science & Engineering, IBSS College of Engineering, Amravati.
3. Asst. Prof, Department of Computer Science & Engineering, IBSS College of Engineering, Amravati.

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: Recommendation algorithms are best known for their use on e-commerce Web sites, where they use input about a customer's interests to generate a list of recommended items. Recommender systems apply knowledge discovery techniques to the problem of making personalized recommendations for information, products or services during a live interaction. The tremendous growth in the amount of available information and the number of visitors to Web sites in recent years poses some key challenges for recommender systems. These are: producing high quality recommendations, performing many recommendations per second for millions of users and items and achieving high coverage in the face of data sparsity. In traditional collaborative filtering systems the amount of work increases with the number of participants in the system. New recommender system technologies are needed that can quickly produce high quality recommendations, even for very large-scale problems. To address these issues we have explored item-based collaborative filtering techniques. Item-based techniques first analyze the user-item matrix to identify relationships between different items, and then use these relationships to indirectly computer recommendations for users. In this paper we analyze different item-based recommendation generation algorithms. We look into different techniques for computing item-item similarities (e.g., item-item correlation vs. cosine similarities between item vectors) and different techniques for obtaining recommendations from them (e.g., weighted sum vs. regression model). Finally, we experimentally evaluate our results and compare them to the basic k-nearest neighbor approach. Our experiments suggest that item-based algorithms provide dramatically better performance than user-based algorithms.

Keywords: Recommendation algorithm, Collaborative Filtering, Model-based Collaborative Filtering Algorithms.

PAPER-QR CODE

Corresponding Author: MS. KALYANI P. SABLE

Access Online On:

www.ijpret.com

How to Cite This Article:

Kalyani Sable, IJPRET, 2014; Volume 2 (9): 1002-1010

INTRODUCTION

Most recommendation algorithms start by finding a set of customers whose purchased and rated items overlap the user's purchased and rated items.2 The algorithm aggregates items from these similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user. Two popular versions of these algorithms are *collaborative filtering* and *cluster models*. Other algorithms –including search-based methods and our own item-to-item collaborative filtering focus on finding similar items, not similar customers. For each of the user's purchased and rated items, the algorithm attempts to find similar items. It then aggregates the similar items and recommends them. The amount of information in the world is increasing far more quickly than our ability to process it. All of us have known the feeling of being overwhelmed by the number of new books, journal articles, and conference proceedings coming out each year. Technology has dramatically reduced the barriers to publishing and distributing information. Now it is time to create the technologies that can help us sift through all the available information to find that which is most valuable to us. One of the most promising such technologies is collaborative filtering. Collaborative filtering works by building a database of preferences for items by users. A new user, Neo, is matched against the database to discover neighbors, which are other users who have historically had similar taste to Neo. Items that the neighbors like are then recommended to Neo, as he will probably also like them. Collaborative filtering has been very successful in both research and practice, and in both information filtering applications and E-commerce applications.

Recommendation Algorithms

Most recommendation algorithms start by finding a set of customers whose purchased and rated items overlap the user's purchased and rated items. The algorithm aggregates items from these similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user. Two popular versions of these algorithms are *collaborative filtering* and *cluster models*. Other algorithms — including search-based methods and our own item-to-item collaborative filtering — focus on finding similar items, not similar customers. For each of the user's purchased and rated items, the algorithm attempts to find similar items. It then aggregates the similar items and recommends them.

COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEMS

Recommender systems apply data analysis techniques to the problem of helping users find the items they would like to purchase at E-Commerce sites by producing a predicted likeliness score or a list of top{N recommended items for a given user. Item recommendations can be made using different methods. Recommendations can be based on demographics of the users, overall top selling items, or past buying habit of users as a predictor of future items.

Collaborative Filtering (CF) is the most successful recommendation technique to date. The basic idea of CF-based algorithms is to provide item recommendations or predictions based on the opinions of other like-minded. users. The opinions of users can be obtained explicitly from the users or by using some implicit measures.

Overview of the Collaborative Filtering Process

The goal of a collaborative filtering algorithm is to suggest new items or to predict the utility of a certain item for a particular user based on the user's previous likings and the opinions of other like-minded users. In a typical CF scenario, there is a list of m users U = {u1,u2,u3,... u_m} and a list of n items I = {i1, i2,..i_n}. Each user u_i has a list of items I_{ui} , which the user has expressed his/her opinions about. Opinions can be explicitly given by the user as a rating score, generally within a certain numerical scale, or can be implicitly derived from purchase records, by analyzing timing logs, by mining web hyperlinks and so on. Note that $I_{ui} \subseteq I$ and it is possible for I_{ui} to be a null-set. There exists a distinguished user $u_a \in U$ called the active user for whom the task of a collaborative filtering algorithm is to find an item likeliness that can be of two forms.

- Prediction is a numerical value, P_{aj}, expressing the predicted likeliness of item $i_j \notin Iua$ for the active user u_a. This predicted value is within the same scale (e.g.,from 1 to 5) as the opinion values provided by u_a.
- Recommendation is a list of N items, $I_r \subset I$, that the active user will like the most. Note that the recom-mended list must be on items not already purchased by the active user, i.e., $I_r \cap I_{ua} = Ø$. This interface of CF algorithms is also known as Top-N recommendation.

Figure 1 shows the schematic diagram of the collaborative filtering process. CF algorithms represent the entire m × n user-item data as a ratings matrix, A. Each entry $a_{i,j}$ in A represents the preference score (ratings) of the ith user on the jth item. Each individual ratings is within a numerical scale and it can as well be 0 indicating that the user has not yet rated that item. Researchers have devised a number of collaborative filtering algorithms that can be divided into two main categories

Memory-based (user-based) and Model-based (item-based) algorithms:

In this section we provide a detailed analysis of CF-based recommender system algorithms. Memory-based Collaborative Filtering Algorithms. Memory-based algorithms utilize the entire user-item database to generate a prediction. These systems employ statistical techniques to find a set of users, known as neighbors that have a history of agreeing with the target user

(i.e., they either rate different items similarly or they tend to buy similar set of items). Once a neighborhood of users is formed, these systems use different algorithms to combine the preferences of neighbors to produce a prediction or top-N recommendation for the active user. The techniques, also known as nearest-neighbor or user-based collaborative filtering, are more popular and widely used in practice.

Model-based Collaborative Filtering Algorithms.

Model-based collaborative filtering algorithms provide item recommendation by first developing a model of user ratings. Algorithms in this category take a probabilistic approach and envision the collaborative filtering process as computing the expected value of a user prediction, given his/her ratings on other items. The model building process is performed by different machine learning algorithms such as Bayesian network, clustering, and rule-based approaches. The Bayesian network model formulates a probabilistic model for collaborative filtering problem. Clustering model treats collaborative filtering as a classification problem and works by clustering similar users in same class and estimating the probability that a particular user is in a particular class C, and from there computes the conditional probability of ratings. The rule-based approach applies association rule discovery algorithms to find association between co-purchased items and then generates item recommendation based on the strength of the association between items

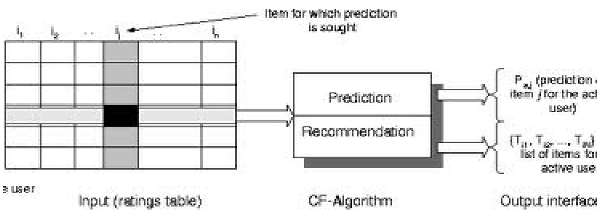

Figure1: The collaborative filtering process

IV. ITEM-BASED COLLABORATIVE FILTERING ALGORITHM

In this section we study a class of item-based recommendation algorithms for producing predictions to users. Unlike the user-based collaborative filtering algorithm discussed in

Section 2, the item-based approach looks into the set of items the target user has rated and computes how similar they are to the target item i and then selects k most similar items {i1, i2, i_k}. At the same time their corresponding similarities {s_{i1}, s_{i2}, ... s_{ik}} are also computed. Once the most similar items are found, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items. We describe these two aspects, namely, the similarity computation and the prediction generation in details here.

a) Item Similarity Computation

One critical step in the item-based collaborative filtering algorithm is to compute the similarity between items and then to select the most similar items. The basic idea in similarity computation between two items i and j is to first isolate the users who have rated both of these items and then to apply a similarity computation technique to determine the similarity s_{ij} Figure 2 illustrates this process; here the matrix rows represent users and the columns represent items. There are a number of different ways to compute the similarity between items. Here we present three such methods.

These are cosine-based similarity, correlation-based similarity and adjusted-cosine similarity.

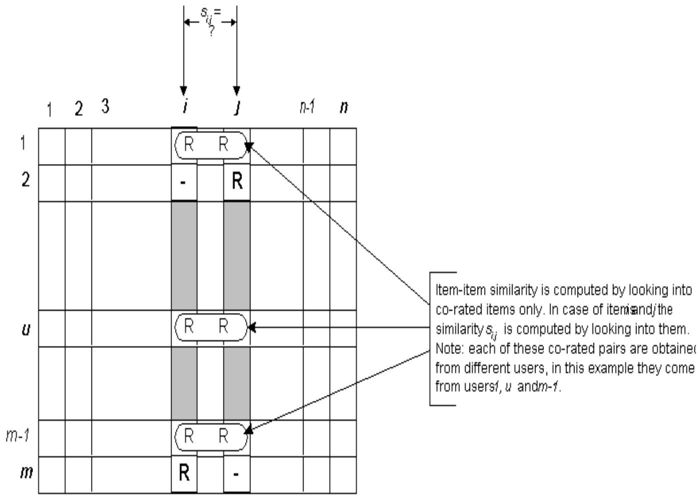

Figure 2: Isolation of the co-rated items and similarity computation

b) Prediction Computation

The most important step in a collaborative filtering system is to generate the output interface in terms of prediction. Once we isolate the set of most similar items based on the similarity measures, the next step is to look into the target users ratings and use a technique to obtain predictions. Here we consider two such techniques.

i)Weighted Sum

As the name implies, this method computes the prediction on an item i for a user u by computing the sum of the ratings given by the user on the items similar to i. Each ratings is weighted by the corresponding similarity $s_{i,j}$ between items i and j. Formally, using the notion shown in Figure 3 we can denote the prediction $P_{u,i}$ as

$P_{u,i} = \sum_{\text{all similar items, N}} (s_{i,N} * R_{u,N}) / P_{\text{all similar items, N}} (|s_{i,N}|)$

Basically, this approach tries to capture how the active user rates the similar items. The weighted sum is scaled by the sum of the similarity terms to make sure the prediction is within the predefined range.

ii) Regression

This approach is similar to the weighted sum method but instead of directly using the ratings of similar items it uses an approximation of the ratings based on regression model. In practice, the similarities computed using cosine or correlation measures may be misleading in the sense that two rating vectors may be distant (in Euclidean sense) yet may have very high similarity. In that case using the raw ratings of the \so-called" similar item may result in poor prediction. The basic idea is to use the same formula as the weighted sum technique, but instead of using the similar item N's "raw" ratings values $R_{u,N}$'s, this model uses their approximated values $R'_{u,N}$ based on a linear regression model. If

we denote the respective vectors of the target item i and the similar item N by Ri and R_N the linear regression model can be expressed as

$R'_N = \propto R'_I + \beta + \epsilon$

The regression model parameters $\propto$, β are determined by going over both of the rating vectors. ϵ is the error of the regression model.

ITEM-TO-ITEM

COLLABORATIVE FILTERING

Amazon.com uses recommendations as a targeted marketing tool in many email campaigns and on most of its Web sites' pages, including the high traffic Amazon.com homepage. Clicking on the "Your Recommendations" link leads customers to an area where they can filter their recommendations by product line and subject area, rate the recommended products, rate their previous purchases, and see why items are recommended (see Figure 3). As Figure 4 shows, our shopping cart recommendations, which offer customers product suggestions based on the items in their shopping cart.

The feature is similar to the impulse items in a supermarket checkout line, but our impulse items are targeted to each customer. Amazon.com extensively uses recommendation algorithms to personalize its Web site to each customer's interests. Because existing recommendation algorithms cannot scale to Amazon.com's tens of millions of customers and products, we developed our own. Our algorithm, item-to-item collaborative filtering, scales to massive data sets and produces high-quality recommendations in real time.

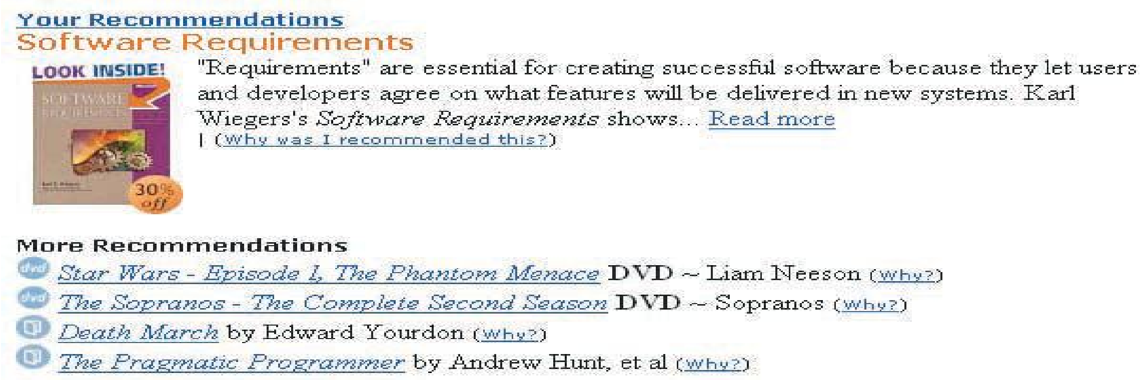

Figure3. The "Your Recommendations" feature on the Amazon.com homepage. Using this feature, customers can sort recommendations and add their own product ratings.

Figure 4. Amazon.com shopping cart recommendations. The recommendations are based on the items in the customer's cart: The Pragmatic Programmer and Physics for Game Developers

How It Works

Rather than matching the user to similar customers, item-to-item collaborative filtering matches each of the user's purchased and rated items to similar items, then combines those similar items into a recommendation list. To determine the most-similar match for a given item, the algorithm builds a similar-items table by finding items that customers tend to purchase together. We could build a product-to-product matrix by iterating through all item pairs and computing a similarity metric for each pair. However, many product pairs have no common customers, and thus the approach is inefficient in terms of processing time and memory usage. The following iterative algorithm provides a better approach by

calculating the similarity between a single product and all related products:

For each item in product catalog, I1

For each customer C who purchased I1

*For each item I2 purchased by*customer C Record that a customer purchased I1 and I2 For each item I2

Compute the similarity between I1 and I2

It's possible to compute the similarity between two items in various ways, but a common method is to use the cosine measure we described earlier, in which each vector corresponds to an item rather than a customer, and the vector's *M*dimensions correspond to customers who have purchased that item. This offline computation of the similar-items table is extremely time intensive, with $O(N2M)$ as worst case. In practice, however, it's closer to $O(NM)$, as most customers have very few purchases. Sampling customers who purchase best-selling titles reduces runtime even further, with little reduction in quality.

Given a similar-items table, the algorithm finds items similar to each of the user's purchases and ratings, aggregates those items, and then recommends the most popular or correlated items. This computation is very quick, depending only on the number of items the user purchased or rated.

CONCLUSION

Recommendation algorithms provide an effective form of targeted marketing by creating a personalized shopping experience for each customer. For large retailers like Amazon.com, a good recommendation algorithm is scalable over very large customer bases and product catalogs, requires only sub second processing time to generate online recommendations, is able to react immediately to changes in a user's data, and makes compelling recommendations for all users regardless of the number of purchases and ratings. Unlike other algorithms, item-to-item collaborative filtering is able to meet this challenge. In the future, we expect the retail industry to more broadly apply recommendation algorithms for targeted marketing, both online and offline. While e-commerce businesses have the easiest vehicles for personalization, the technology's increased conversion rates as compared with traditional broad-scale approaches will also make it compelling to offline retailers for use in postal mailings, coupons, and other forms of customer communication.

Recommender systems are a powerful new technology for extracting additional value for a business from its user data- bases. These systems help users find items they want to buy from a business. Recommender systems benefit users by enabling them to find items they like. Conversely, they help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web. New technologies are needed that can dramatically improve the scalability of recommender systems.

REFERENCES

1. J.B. Schafer, J.A. Konstan, and J. Reidl, "E-Commerce Recommendation V Applications," *Data Mining and Knowledge Discovery*, Kluwer Academic, 2001, pp. 115-153.

2. P. Resnick et al., "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proc. ACM 1994 Conf. Computer Supported Cooperative Work*, ACM Press, 1994, pp. 175-186.

3. J. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proc. 14th Conf. Uncertainty in Artificial Intelligence*, Morgan Kaufmann, 1998, pp.43-52

4. Agarwal, C. C., Wolf, J. L., Wu, K., and Yu, P. S. (1999). Horting Hatches an Egg: A New Graph-theoretic Approach to Collaborative Filtering. In Proceedings of the ACM KDD'99 Conference. San Diego, CA. pp. 201-212.

5. Basu, C., Hirsh, H., and Cohen, W. (1998). Recommendation as Classification: Using Social and Content-based Information in Recommendation. In Recommender System Workshop'98.