Building Item–based Recommender System for Ladies' wear Personalization Service

Nwe Nwe Win, Nang Tom Hawm Saw
Computer University (Mandalay)
violetskyblue@gmail.com

Abstract

In recent years, the need for personalized service has been increased. However, personalization services must be improved to lighten user's burden in the process of personalization and produce results that are more adaptable. As one of the most promising approaches to improve the current personalized services, recommender systems have emerged in domains such as E-commerce, digital libraries. In our work, we firstly attempted to apply a recommender system in the field of E-commerce applications. Then, we decided to build a recommender system for Ladies' wear personalization services to make it more user-friendly and user-adaptive. One of the most successful technologies for recommender system is collaborative filtering. The bottleneck in conventional collaborative filtering algorithm (such as traditional user-based algorithm) is the search for neighbors among a large user population of potential neighbors. Our system uses item-based algorithm to avoid this bottleneck. The algorithm explores the relationships between items first rather than the relationship between users. Because the relationships between items are relatively static, item-based algorithms may be able to provide the same quality as the user-based algorithms with less online computation.

Keywords: personalization service, recommender system, collaborative filtering algorithm, item-based collaborative filtering algorithm.

1. Introduction

Personalization is the ability to provide content and services that are tailored to individuals based on knowledge about user preference and behavior. The need for personalization services has been increased with the amount of information on the internet that is increasing far more quickly than our ability to process it. But, personalization must be improved to lighten user's burden in the process of personalization and produce results that are more adaptable.

As one of the most promising approaches to improve the current personalization services, recommender system have emerged in domains such as E-commerce, digital libraries and knowledge management. Recommender system uses rating from user on items for the purpose of predicting the user preferences on items that have not been rated.

In our work, we focused on building recommender system in the field of e-commerce application. To be more detail, we build a recommender system for Ladies' wear personalization services. In our system, we apply collaborative filtering technology that has been developed as the most successful technology for recommender technique for date.

In traditional collaborative filtering systems, the amount of work increases with the number of participants in the system. Our system uses item-based collaborative filtering algorithm to avoid this bottleneck. The algorithm avoids this bottleneck by exploring the relationships between items first rather than between users.

Recommendations for users are computed by finding items that are similar to other items the user has liked. Because the relationships between items are relatively static, item-based algorithms are able to provide the same quality as the user-based algorithms with less online computations.

In collaborative filtering algorithms, one critical step is to compute the similarity between items and then to select the most similarity items. Three common methods that compute the similarity between items are cosine-based similarity, correlation-based similarity and adjusted-cosine similarity.

Computing similarity using basic cosine measure in item-based has a clear advantage, as the MEA is significantly lower in Badrul Sarwar Research [7], two items are thought of as two vectors in the m dimensional user-space. Because of this advantage, we used cosine-based similarity method to compute the similarity between items (dresses) of our system.

1.1 Related Works

In this section we briefly present some of the research literature related to collaborative filtering, recommender systems, and personalization.

Tapestry [2] is one of the earliest implementations of collaborative filtering-based recommender system.

Schafer et al., [6] present a detailed taxonomy and examples of recommender systems used in E-commerce and how they can provide one to one personalization. Sparsity problem in recommender has been addressed in [4, 3].The problems associated with high dimensionality in recommender systems have been discussed in [1] and application of dimensionality reduction techniques to address issues has been investigated in [5].

My work explores the extent to which item-based recommenders, a new class of recommender algorithms, are able to solve this problem.

2. Theoretical Background

In this section, we will introduce the background theories that are related to our system.

2.1. 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 $\mu = \{\mu_1, \mu_2, \ldots \mu_m\}$ and a list of n items i= $\{i_1, i_2, \ldots, i_n\}$. Each user μ_1 has a list of items I_{ui}, which the user has expressed his/her opinions.

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 son on. Note that $I\ \mu i \subseteq I$ and it is possible for I_{ui} to be *a mull-set*. There exists a distinguished user $u_a \in \boldsymbol{u}$ 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 $i_i \notin I$, 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 recommended list must be on items not already purchased by the active user, i.e., $I_r \cap I_{\mu a} = \emptyset$. This interface of CF algorithms is also known as Top-N *recommendation.*

CF algorithms represent the entire *m X n* user-item data as a ratings matrix. *A*. Each entry $a_{i,j}$ in *A* represent the preference score (ratings) of the i^{th} user on the j^{th} item. Each individual rating is within a numerical scale and it can as well be 0 indicating that the user has not yet rated that item.

2.2 User-Based Collaborative Filtering

The user-based method makes recommendations with the following simple step-by-step procedure:

1. Users explicitly assign numeric rating to items.
2. A recommender system correlates the ratings in order to determine which user's ratings are the most similar to other ones.
3. The system predicts ratings of new items for the target user, based on the ratings of similar users.
4. If these new items seem to be preferred, the system recommends to the user.

The basic idea of this method is to provide item recommendations or predictions based on the opinions of other like-minded users. The opinion of users can be obtained explicitly from the users or sometimes by using some implicit measures.

2.2.1 Challenges of User-based Collaborative Filtering Algorithms.

User-based collaborative filtering systems have been very successful in past, but their widespread use has revealed some real challenges such as:

- **Sparsity**. In most real-world cases, users rate only a very small percentage of items. This causes data to become sparse. In such cases, the recommendation engine cannot provide precise proposals, due to lack of information.
- **Scalability.** Nearest neighbor algorithms require computation that grows with both the number of users and the number of items. One of the major drawbacks of the user-based CF systems in general is that they do not scale well.

In other words, the weakness of the user-based CF algorithm led to explore alternative recommender system algorithms. In user-based collaborative system, the amount of work increases with the number of participants in the system.

In other words, the bottleneck in item-based collaborative filtering is the relationships between items first, rather than the relationship between users. Consequently, our work uses item-based collaborative filtering algorithm to avoid this issue. We will continue to present the overview of item-based collaborative filtering algorithm in the following section.

2.2.2 Overview of the Item to Item Collaborative Filtering Algorithm

Items are rated and used as parameters instead of users. This type of filtering uses the ratings to group various items together in groups so consumers can compare them as well as rating scale that is available to manufactures so they can locate where their products stand in the market in a consumer based rating scale.

3. Overview of the System

In this section, we will explain how we implement the collaborative filtering algorithm in our system. And, we will explain the item-based collaborative filtering by using cosine-based similarity.

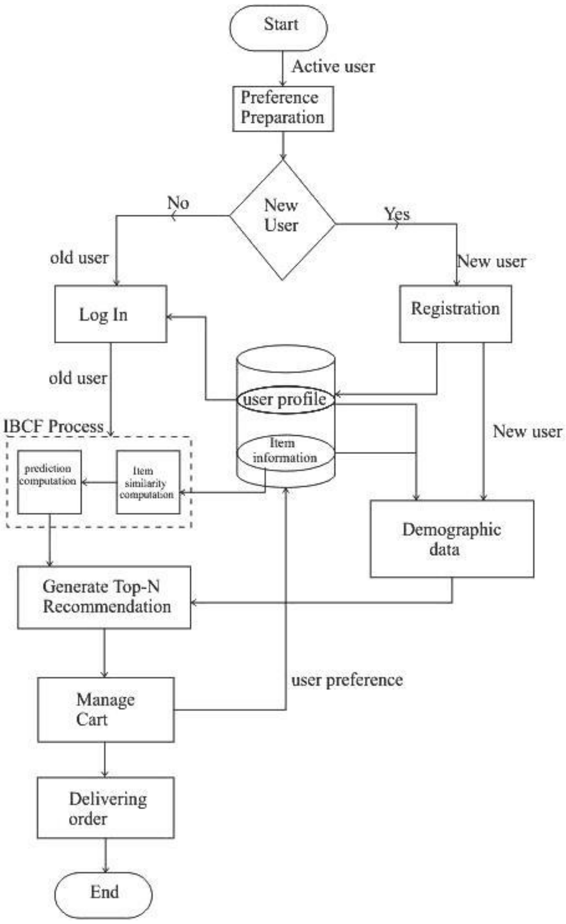

Figure 1. System Flow Design

3.1 How Item-based Collaborative Filtering Algorithm works in Our System

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.

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.I_1
For each customer C who rated I_1
For each item I_2 rated by customer C
Record that a customer rated I_1 and I_2
For each item I_2
Compute the similarity between I_1 and I_2

Given a similar items table, the algorithm finds items similar to each of the user's purchase and ratings, aggregates those items, and then recommends the most popular or highly correlated similar items. This computation is very quick, depending only on the number of items the user purchased or rated; independently of the user catalog size or the total number of customers.

The main idea here is to analyze the user- item representation matrix to identify relations between different items and then to use these relations to compute the prediction score for a given user-item pair. The intuition behind this approach is that a user would be interested in purchasing items that are similar to the items the user liked earlier and would tend to avoid items that are similar to the items the user didn't like earlier.

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 $i_1, i_2, \ldots, i_k$, . At the same time their corresponding similarities $S_{i1}, S_{i2}, \ldots, S_{ik}$, $\{S_{i1}, S_{i2}, \ldots S_{ik}\}$ 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.

3.2. Item Similarity Computation

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 $s_{i,j}$.

There are a number of different ways to compute the similarity between items. The most commonly used similarity computation methods are;

- Cosine-based similarity
- Correlation-based similarity
- Adjusted-cosine similarity

Computing similarity using basic cosine measure in item-based has a clear advantage, as the MEA is significantly lower in Badrul Sarwar Research [7], two items are thought of as two vectors in the m dimensional user-space. Because of this advantage, we used cosine-based similarity method to compute the similarity between items (dresses) of our system.

3.2.1 Cosine-based Similarity

The similarity between them is measured by computing the cosine of the angle between these vectors. Formally, the similarity between items *i* and *j* using this scheme is given by

$$\mathrm{Sim}(i, j) = \cos = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \times \|\vec{j}\|} \qquad (1)$$

where "." denotes the dot-product of the two vectors.

Table1. User-Item matrix table

	I1	I2	I3	I4	Ave
u1(out of 5)	3	5	3		3.67
u2(out of 5)	1		2		1.5
u3(out of 5)	4	2	4	2	3
u4(out of 5)		5	4	2	3.67
Average	2.67	4	3.25	2	

$$\text{Sim}(\vec{i}, \vec{j}) = \frac{(3,1,4,0).(5,0,2,5)}{\sqrt{3^2+1^2+4^2} \times \sqrt{5^2+2^2+5^2}} \qquad (2)$$

3.2.2 Prediction Computation

Once we isolate the set of most similar items based on the similarity measures, the next step is to look into the target user's ratings and use a technique to obtain predictions.

The following equation (Weighted Sum) is used to compute 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 we can denote the prediction Pu,i as

$$Pu,i = \frac{\Sigma \text{ all similar items},N(s_{i,N} * R_{u,N})}{\Sigma \text{ all similar ites},N(|s_{i,N}|)} \qquad (3)$$

Table2. Item's similarities table

	I1	I2	I3	I4
I1	1	0.61	0.79	0.55
I2	0.61	1	0.87	0.67
I3	0.79	0.87	1	0.84
I4	0.55	0.67	0.84	1

I4 Similar = I2, I3, I4

$$P_{(U4,I1)} = \left(\frac{(0.61*5)(0.79*4)(0.55*2)}{0.61+0.79+0.55}\right) = 3.75 \qquad (4)$$

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

3.3 Process of the System

Our work is the implementation of the item-based recommender system for Ladies' wear personalization service that generates a list of Ladies' wear recommendation which will probably liked by the active user.

There are three major phases in developing in this system.

3.3.1 Preference preparation phase

In this phase, preference means how much user is interested in ladies' wear items. In this ladies' wear recommender system, users can express their preferences as rating scores to ladies' wear items. User Preferences are collected using only explicit measure.

To get recommendation service, users actually need to rate ladies' wear items. The more rating the user offers, the more accurate recommendation he can get. Ratings are on a scale of 1 to 5 from the lowest to highest liking. Here, 1 represents "strongly dislike", 2 means " dislike", 3 means "just ok", 4 is for " like" and 5 is " the best like". Rating values from users are then stored into rating table in System Database.

3.3.2 Item-Item Similarity Computation Phase

The static nature of items leads to the ideas of pre-computing the item similarities. One possible way of pre-computing the item similarities is to compute all-to-all similarity and then performing a quick table look-up to retrieve the required similarity values. This system uses the Cosine Similarity to compute item similarities. Here, *sim(i,j)* must generate a similarity score which may vary from 0 to 1, where 0 means "no similarity" and 1 is identical.

3.3.3 Prediction and Recommendation phase

The system provides two kinds of output:

- Prediction
- Recommendation

Prediction is essential to compute recommendations. The system uses weighted sun method. The result value from $P_{u,i}$ must be numerical value which must be system defined rating scale, here, from 1 to 5. Prediction represents a guess at how much a user would like an item.

Recommendation is a top-*N* list of items. They system sorts all predicted items for the active user in descending order and select *N* items with the highest prediction values which may consists of the user's probable most favorite items. It is the final output of recommender systems.

4. Conclusion

Current personalization services must be improved by making it more user-friendly and more user-adaptive. Recommender systems have been emerged as a crucial technology to improve it. Recommender system can help users find items they want to buy from a business. But, recommender systems are being stressed by the huge volume of user data available on the Web. New technologies are needed that can dramatically improve the scalability of recommender system. Item-based techniques hold the promise of allowing collaborative filtering based algorithms to scale to large data sets and at the same time produce high-quality recommendation item-based collaborative filtering recommendation algorithms. In our work, we built an item-based recommender system for Ladies' wear personalization services that can benefits users by enabling them find the items (dresses) they like. Conversely, the system can help the business by generating more sales.

5. References

[1] Billsus, D., and Pazzani, M.J.(1998). Learning Collaborative Information Filters. In *proceedings of ICML '98.* Pp.

[2] Goldberg, d., Nichols, D., Oki, B.M., and Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communi- cations of the ACM.* December.

[3] Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J. (1999). Combining Collaborative Filtering With Personal Agents for Better Recommendations. In *Procedings of the AAAI-'99 conference,* pp 439-446.

[4] Sarwar, B., M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J. (1998). Using Filtering Agents to Improve Prediction Quality in the Grouplens Research Collaborative Filtering System. In *Proceedings of CSCW '98, Seattle, WA.*

[5] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2000). Application of Dimensionality Reduction in Recommender System-A Case Study. In *ACM WebKDD 2000 Workshop.*

[6] Schafer, J.B., Konstan, J., and Riedl, J. (1999). Recommender Systems in E-Commerce. In *Proceedings of ACM E-Commerce 1999 conferece.*

[7] Badrul Sarwar, George Karypis, Joseph Ko-nstan, and John Riedl: *Item-based Collaborative Filtering Recommendation Algorithms.* (2001).