Cost-oriented Recommendation Model for E-Commerce

Grzegorz Chodak[1] and Grażyna Suchacka[2]

[1]Institute of Organisation and Management, Wroclaw University of Technology, Wroclaw, Poland
[2]Institute of Mathematics and Informatics, Opole University, Opole, Poland
grzegorz.chodak@pwr.wroc.pl,gsuchacka@uni.opole.pl

Abstract. Contemporary Web stores offer a wide range of products to e-customers. However, online sales are strongly dominated by a limited number of bestsellers whereas other, less popular or niche products are stored in inventory for a long time. Thus, they contribute to the problem of frozen capital and high inventory costs. To cope with this problem, we propose using information on product cost in a recommender system for a Web store. We discuss the proposed recommendation model, in which two criteria have been included: a predicted degree of meeting customer's needs by a product and the product cost.

Keywords: recommendation method, recommender system, e-commerce, B2C, Business-to-Consumer, Web site, Web store, cost.

1 Introduction

Over the last years many businesses have been moved to the Internet, taking advantage of offering products to a worldwide e-customer population. As a result of the ubiquitous information overload on the Web, as well as a multitude of products offered by online stores, recommendation methods came into play.

Online stores are implemented through Web sites, which allow Internet users (also called customers in this context) to browse and purchase products online. Recommender systems are essential components of such sites as they help customers to cope with a huge amount of items available in the store. Various recommendation methods have proved to be successful in enhancing e-commerce sales in the following ways [11]:

- converting page views into transactions through helping customers find products they wish to buy,
- increasing cross-sell by suggesting additional products and increasing the average order size,
- building customer loyalty by creating a value-added relationship between the site of the store and the customer.

The idea of product recommendation is simple: a user interacting currently with a Web store site (who is a target user) is presented additional information on products in which they may be interested. To predict the most relevant recommendations different techniques have been proposed based on user demographic data, user transactional behavior, product and user characteristics, and so on. We propose introducing a novel criterion into a recommender system, namely the information on product cost. To the best of our knowledge, such a criterion has not been considered in the context of recommendation systems so far.

The idea of cost-oriented recommendation for a web store is based on the main purpose of each online retailer, which is profit maximization. The revenue level is one of profit determinants but the second one, also very important, is cost. Up-to-date recommender systems for e-commerce sites take into consideration only the aspect of increasing the revenue through additional sales whereas the cost aspect has been forgotten. As we analyze the information environment of a Web store, we can build very detailed behavioral profiles of customers based on their shopping carts, products' views, and data obtained from tracking systems. The Web store's databases combined with server logs and data mining system can give us a lot of information about customer needs and tastes. To maximize revenues it is important to propose customers products in which they can be interested. For example, complementary products offered to a customer can increase the value of the shopping cart and the resultant revenue from e-business. However, from the cost-side point of view, which is crucial to online retailers, it is important to sell products which generate the highest cost in inventory.

Motivated by the aforementioned observations, the authors propose applying a multicriteria analysis in a Web store system in order to recommend products in which an e-customer may be interested, on the one hand, and which generate the highest cost, on the other hand. The general idea of the proposed recommendation model is the following: for the online retailer it is more profitable to recommend to a customer products which generate much higher cost than other products, even if the probability of buying them is a little bit smaller than for other products generating much lower cost.

The rest of the paper is organized as follows. Section 2 outlines the state of the art in recommendation methods for e-commerce Web sites, with a focus on the kind of information used and criteria applied. Section 3 discusses the proposed recommendation model, which aims both at meeting user's needs and limiting online retailer's cost while recommending products to a target user. Finally, section 4 concludes the paper.

2 The state of the art in recommender systems for e-commerce

Recommender systems for online stores may use different kind of knowledge and apply various techniques to generate product recommendations. In this section, we overview these approaches with special respect to data sources used by them.

2.1 General and impersonalized recommendations

Basic recommendations in most of contemporary Web stores include novelties, bestsellers, and products selected at random. Such recommendations are easy to implement and do not involve complex computations.

2.2 Approaches using context-free information on associations between products

Some products may be related to each other, e.g. they may be complements or substitutes. Such information may be very useful in predicting customer needs and offering to them products with high probability of being purchased together. In practice, this kind of knowledge is usually provided by experts and is hand-coded. However, knowledge on relations between products may be automatically discovered by using association rules or collaborative filtering. An association rule describes the relationship that one product is often purchased with other products. Collaborative filtering methods are capable of computing similarities between items, e.g. based on product text description or keywords [1, 6, 11].

2.3 Approaches using information on products in the context of a single customer

Customer behavioral data connected with products viewed, purchased, or rated by the customer may be recorded during multiple customer visits to the site. Then, at the next customer visit, the information may be used by some of content-based recommendation methods. For example, the utility of a new item for a target user may be estimated based on ratings assigned by this user to other items which are similar to the estimated item. In such a case the recommender system tries to discover similarities among the items which have been rated highly by the user in the past and then, new items matching user preferences the most are recommended.

Some methods use a customer profile based on user characteristics, such as age, gender, interests, etc. This information may be given explicitly by the user, e.g. through a questionnaire, may be assessed based on ratings assigned by the customer to specific products, or may be learned from user transactional behavior over time.

Techniques for content-based recommendation may use traditional heuristics based mostly on information retrieval methods and calculate predictions based on heuristic formulas, such as a cosine similarity measure. Other approaches for content-based recommendation are based on a user model. Such model is often built using statistical learning techniques, e.g. Bayesian classifiers or machine learning techniques: clustering, decision trees, and artificial neural networks [1].

2.4 Approaches using information on products in the context of multiple customers

Some data mining techniques, such as association rules between co-purchased products, are applied to discover associations between products purchased in the past not only by a target customer but also by other customers [10]. Such recommendations are offered by popular Web stores. An example may be a leading online bookstore, amazom.com [2], in which a user entering the page describing a selected book faces the following recommendation sections:

– "*Frequently Bought Together*" - a group of a few books that have been often purchased together, along with a common price for all of them;
– "*Customers Who Bought This Item Also Bought*" - a more numerous group of books often purchased together, with each item being described with a degree of association with the selected book, graphically presented by the appropriate number of colored stars;
– "*Customers Also Bought Items By*" - a group of other authors' books, often bought with the selected book;
– "*What Other Items Do Customers Buy After Viewing This Item?*" with links to the corresponding items.

More advanced, personalized techniques try to assess a degree of similarity between users based on their behavioral data over multiple visits and make recommendations connected with a new item for a target user based on the items viewed, purchased or rated in the past by users which are similar to the target user. Thus, these methods are particularly useful when the information on a particular user is limited.

To build models of similar user groups collaborative filtering methods are used. They may be either memory-based (heuristic-based) or model-based methods.

a) *Memory-based collaborative recommendation* approaches are based on the collection of previously viewed, purchased or rated items by similar users. Utility of a new product for a target user is usually computed as an aggregate of utilities of the same product according to the most similar users. The aggregation may be the simple average, the weighted sum, or the adjusted weighted sum [1, 10].

Example approaches include nearest neighbor algorithms based on computing the distance between customers based on their preference history. A prediction concerning a given product is based on the weighted average of opinions for the product given by nearest neighbors of the target customer. In practice, nearest neighbor algorithms use heuristics to search for neighbors, as well as use opportunistic sampling in the case of very large customer population [11].

Various approaches are used to compute similarity measures between two users in collaborative recommender systems, including a correlation-based approach, a cosine-based approach, and other approaches.

b) *Model-based collaborative recommendation* approaches use different kind of data (e.g. product ratings provided by a user) to learn a user model, which is then used in a recommender system. A probabilistic approach to collaborative filtering may use Bayesian networks or cluster models to estimate a probability that a target user will give a particular rating to an item given ratings assigned by that user to other items [4].

A Bayesian networks model is based on a graph, in which nodes and edges represent some customer information described in terms of probabilities. Cluster models are created by using different clustering techniques, which make it possible to identify groups of customers with similar preferences. A recommendation decision for a target customer is made by averaging opinions of other customers in the corresponding cluster. Some clustering techniques may assign a customer to several clusters with different degrees of participation and then, recommendation is made as a weighted average across the clusters [11].

There is also possible to combine machine learning techniques (e.g. artificial neural networks) and feature extraction techniques (e.g. singular value decomposition) in a collaborative filtering [3]. Furthermore, there have been proposed hybrid approaches to product recommendation, e.g. by combining content-based methods and collaborative filtering [5, 7, 8, 12].

3 Description of the cost-oriented recommendation model

We consider a set of n products available in a Web store, $\{X_1, X_2, ..., X_n\}$. Each ith product ($i = 1, 2, ..., n$) can be described with a pair of attributes $(\mathbf{Q}_i, C_i)$, where:

- $\mathbf{Q}_i$ is a vector of recommendation quality ranks of the ith product being recommended with other products, $\mathbf{Q}_i = [Q_{i1}, Q_{i2}, ..., Q_{i\ n-1}]^T$,
- C_i s a cost of the ith product.

We now discuss the proposed way of including these attributes in a recommendation model for an e-commerce site.

3.1 Recommendation quality ranks

Let us consider the first attribute of the ith product, i.e. vector $\mathbf{Q}_i$. The jth element of the vector, Q_{ij} ($j = 1, 2, ..., n-1$) is a recommendation quality rank of the ith product being recommended with the jth product. In general, Q_{ij} may be computed using various techniques, some of which have been discussed in section 2. We assume that Q_{ij} is a function of three variables:

$$Q_{ij} = F(NV_{ij}, NO_{ij}, CM_{ij}), \tag{1}$$

where NV_{ij} is the number of ith product's views generated by other customers who bought the jth product; NO_{ij} is the number of ith product's orders made by customers who bought the jth product; CM_{ij} is a complement measure from the range [0, 1] representing the objective level of complements, which is an attribute of the pair (ith product, jth product).

Values of NV and NO for a given Web store can be computed by applying statistical techniques to analyze historical data on customer viewing and purchasing behavior.

Values of CM for product pairs are given by an expert (e.g. a manger) and stored in database. For example, lens which suits the camera are complements with a high value of CM near 1. In theory, the number of stored complement measures can be huge, even up to $n^2/2$. However, in practice usually only a small percentage of all products are complements and thus, a complements matrix has mainly 0 values because only a few pairs of inventory items are complementary products.

There is also a possibility to simplify the model by using boolean values of CM; in such a case two products i and j can be complements (CM_{ij} equal to 1) or not (CM_{ij} equal to 0).

CM is a symmetric relationship whereas NV and NO are not symmetric.

There is also a question why not to show customers not only complementary products but also substitutes. It does not complicate the model, because CM could be treated as a general value indicating whether two products need to be shown together to a customer, or not.

To build the multicriteria recommendation function we should transform non-zero values of NV_{ij} and NO_{ij} to a comparable base. We propose the standard normalization of each of these values according to the formula:

$$y_{ij\_norm} = \frac{y_{ij} - min_{k=1,2,\ldots,n}\{y_{ik}\}}{max_{k=1,2,\ldots,n}\{y_{ik}\} - min_{k=1,2,\ldots,n}\{y_{ik}\}}, \tag{2}$$

where y_{ij_norm} is a normalized value of the measure under consideration (i.e. NV_{ij} or NO_{ij}) for the ith item relative to the jth item, $y_{ij\_norm} \in [0,1]$; y_{ij} is the measurement of NV_{ij} or NO_{ij} for the ith item; $min_{k=1,2,\ldots,n}\{y_{ik}\}$ is the minimum value of NV or NO measures for the ith item; $max_{k=1,2,\ldots,n}\{y_{ik}\}$ is the maximum value of NV or NO measures for the ith item.

The shape of the recommendation quality rank function is another problem that requires deeper research. We propose computing the recommendation quality rank of the ith product recommended with the jth product as a simple weighted sum:

$$Q_{ij\_norm} = w_1 \cdot NV_{ij\_norm} + w_2 \cdot NO_{ij\_norm} + w_3 \cdot CM_{ij}, \tag{3}$$

where w_1, w_2, w_3 are weights assigned to the corresponding elements, $\sum_{k=1}^{3} w_k = 1$; NV_{ij_norm} and NO_{ij_norm} are normalized values of NV_{ij} and NO_{ij}, respectively, according to (2); CM_{ij} is a complement measure representing the objective level of complements.

The weights in the recommendation quality rank function reflect the importance of each criterion. Establishing proper values of these weights is one of the most important problem in this model. We propose an exogenous approach, in which a decision maker has to rank the criteria according to their importance.

3.2 Product cost

Let us consider the second attribute of the ith product, i.e. the cost of the product, C_i. From the economic point of view one can distinguish many kinds of cost connected with products for sale. We propose choosing only two kinds of cost which suit the best the purpose of the model, namely the cost of purchasing the product (*purchase cost*) and the *inventory cost*.

On the one hand, the purchase cost of the product indicates the frozen capital which decreases the cash flow and that is why it is very important to take this kind of cost into consideration in our recommendation model.

On the other hand, the inventory carrying cost is significant. It can be calculated as the operational cost of carrying the product on stock for one month (or year). This cost is the function of the product size and additional product features connected with requirements of special inventory environment (temperature, humidity, etc.).

Therefore, the cost of the ith product can be computed according to the following formula:

$$C_i = PC_i + IC_i, \tag{4}$$

where PC_i is the purchase cost of the ith product, calculated as the net purchase price; IC_i is the inventory cost of the ith product, calculated as the inventory carrying cost.

After calculating the cost of the ith product we need to normalize its value taking the cost of other products into account:

$$C_{i\_norm} = \frac{C_i - min_{k=1,2,\ldots,n}\{C_k\}}{max_{k=1,2,\ldots,n}\{C_k\} - min_{k=1,2,\ldots,n}\{C_k\}}, \tag{5}$$

where C_{i_norm} is the normalized value of C_i, i.e. the cost of the ith product calculated according to (4); $min_{k=1,2,\ldots,n}\{C_k\}$ is the minimum value of all n products' costs; $max_{k=1,2,\ldots,n}\{C_k\}$ is the maximum value of all n products' costs.

Furthermore, we propose taking into account a decline in the inventory cost factor for the ith product, denoted by LV_i. This factor indicates changes in time of the ith product's value: the higher LV_i, the faster the ith product loses its value. This factor is included in the recommendation function (6) so that it affects the value of weight w; thus, the faster a product loses its value, the higher pressure we put on selling it. We propose LV_i being a percentage annual loss of value of the ith item.

3.3 Recommendation function

The idea of our recommendation model is to offer products which are likely to be purchased by a target customer (i.e. products with high recommendation quality ranks) and which generate high cost in inventory at the same time. Depending on a specific Web store, these two criteria (the recommendation quality ranks and the product costs) can be combined together in many possible ways. For example, one could use any recommendation technique to obtain a set of N top recommendations as a first step and then use product costs to obtain the final set of recommendations.

We decided to use both criteria in one step. That is why the normalized values of Q_{ij} and C_i are included in a multicrieria recommendation function.

Taking into consideration the weights of the criteria two different kinds of models are possible. In some models the weights are endogenous whereas in others they are automatically generated when the model is optimized (e.g. in [9]). In our model the weights are partially exogenous: firstly, a decision maker (e.g. a manager) has to rank the criteria according to their importance and determine the value of weight w; then, w is corrected by LV_i, the factor reflecting a decline in the inventory cost of the ith product. The weights obtained are common to all items in the product population of a given Web store.

The value of recommendation function for the ith product in relation to the jth product is computed according to the following formula:

$$RF_{ij} = LV_i \cdot w \cdot Q_{ij\_norm} + (1 - LV_i \cdot w) \cdot C_{i\_norm}, \tag{6}$$

where RF_{ij} is the recommendation function value for the ith product, estimated to take decision on showing the jth product together with the ith product or not; LV_i is the inventory cost factor indicating a percentage annual decline in the value of the ith product; w is the weight established by a manager, $w \in [0, 1]$; Q_{ij_norm} is the normalized value of recommendation quality rank Q_{ij}, $Q_{ij\_norm} \in [0, 1]$; C_{i_norm} is the normalized value of the ith product's cost, $C_{i\_norm} \in [0, 1]$.

Values of the recommendation function are computed for all n products periodically and stored in database. The proposed recommender system should point out a given (parametrized) number of products with the highest values of RF. Products determined for the ith product will be presented on the Web page describing the ith product in a recommendation rectangle, which is usually placed under the product description.

4 Concluding remarks

Motivated by problems of inventory management and high inventory costs encountered by online retailers, we propose a novel recommendation method for a Web store. We discuss the proposed model, which includes a predicted degree of meeting customer's needs and product cost while considering each product to be recommended with a product selected by a customer at a given moment. The innovativeness of the proposed recommendation model is including product cost, which has not yet been taken into consideration in recommender systems for e-commerce. We proposed including the following product costs: purchase cost, inventory cost, and the inventory cost factor indicating an annual decline in the product value.

Our future work will concern a detailed design and implementation of the proposed model. We also plan to introduce some enhancements to the model. Firstly, the problem of establishing the most proper weight value should be explored; one of the possibility is optimizing the weight using the total profit value as the criterion. It is also important to compare efficacy of the proposed method with efficacy of other recommendation methods based on association rules, collaborative filtering, etc.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng. 17(6), 734–749 (2005)
2. amazon.com, http://www.amazon.com
3. Billsus, D., Pazzani, M.J.: Learning Collaborative Information Filters. In: International Conference on Machine Learning, pp. 46–54. Morgan Kaufmann, San Francisco (1998)
4. Breese, J.S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: 14th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco (1998)
5. Burke, R.: Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web 2007. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007)
6. Deshpande, M., Karypis, G.: Item-Based Top-N Recommendation Algorithms. ACM TOIS 22(1), 143–177 (2004)
7. Jung, K.-Y., Jung, J.J., Lee, J.-H.: Discovery of User Preference in Personalized Design Recommender System through Combining Collaborative Filtering and Content Based Filtering. In: Grieser, G., Tanaka, Y., Yamomoto, A. (eds.) DS 2003. LNCS, vol. 2843, pp. 320–327. Springer, Heidelberg (2003)
8. Kazienko, P., Kolodziejski, P.: Personalized Integration of Recommendation Methods for E-Commerce. International Journal of Computer Science and Applications 3(3), 12–26 (2006)
9. Ramanathan, R.: ABC Inventory Classification with Multiple-Criteria Using Weighted Linear Optimization. Computers and Operations Research 33(3), 695–700 (2006)
10. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of Recommendation Algorithms for E-Commerce. In: 2nd ACM Conference on Electronic Commerce (2000)
11. Schafer, J.B., Konstan, J.A., Riedl, J.: E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery 5(1-2), 115–153 (2001)
12. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and Metrics for Cold-Start Recommendations. In: 25th Annual International Conference on Research and Development in Information Retrieval, pp. 253–260. ACM, New York (2002)