

# Maximizing Cross Sells By Optimizing Similar Item Recommendation

Senthilkumaran V  
{vsk.amcs@psgtech.ac.in}

Department of Applied Mathematics and Computational Sciences, PSG College of Technology,  
Coimbatore – 641 004<sup>1</sup>

**Abstract.** Recommendation systems in e-commerce have been researched extensively in the past decades and it has advanced in parallel with the general progress of technology, in particular with the growth of the web. Recommendations account for the majority of the site's earnings because they keep visitors connected with the site content for a longer period. The key obstacle here is anticipating the user's intent based on their online behaviour patterns history. For instance, an entrepreneur can entice a user by offering a discount on a specific element depending on the user's online profile data and behaviour history; or an online travel agency can promote a specific flight destination based on the user's most recent query or information demand. Machine learning is generally used to identify these intentions. Customization systems benefit both businesses and consumers: users receive more pertinent orders and reach their purchasing targeted goals, while businesses can boost revenue and average basket capacity. Using strategies spanning from collaborative filtering to learning sequence by producing embedding for products, this research emphasises various issues in providing excellent product suggestions to users in real-time that are peculiar to the scope and variety of the e-Commerce domain data.

**Keywords:** Item recommendation, cross sells maximization, Recommender system.

## 1 Introduction

Recommender systems (RSs) play a vital role in a customer's everyday life ranging from e-commerce websites, social media, news outlets, OTT and so on [5]. With the rapid growth of the amount of information on the Internet, RSs became fundamental for helping users diminish the problem of information overload and in curating unique choices [10]. RSs are specific tools and algorithms that are designed to stitch and present suggestions for items that are of great use to the customer [1].

In an e-commerce company, it is crucial to understand customer behaviour and therefore spending time and effort in designing RSs is a win-win strategy for both the customer as well as the company [10]. A significant example of this would be to personalize a customer's shopping experience using their co-purchased, co-viewed and co-search history, blending with the popular products thus making their overall journey on the website quick and easy. Suggestions provided by RSs include various decision-making ignitions such as what products to buy, which music to listen to, which hotel to stay in and many more [8]. The main focus point of RSs are users who lack the ability to single out and choose from a wide range of alternative products. Since these recommendations are personalized, different

users receive diverse suggestions which tend to their interests [3]. In addition, there are also generic recommendations such as the current top 10 bestselling books, top 10 cosmetics sold, etc. So how do RSs collect information about its customers? Well, it collects users' preferences which are either explicitly expressed or by interpreting user actions. An example of explicit method of collecting preference is their ratings for a product bought or used earlier. The basic idea of RSs evolved from a single observation: individuals often rely on suggestions and improvisations provided by others in making daily decisions. For instance, it is common to rely on an expert in car manufacturing to know the type of cars that are worth a buy in the current times.

As the e-commerce domain started to expand and many websites got developed over the years, the need for RSs overwhelmed [11]. A lot of customers were finding it very difficult to choose from a huge variety of options leading them to make poor decisions. Thus, the availability of choices not leading to producing a benefit became a drawback in disguise. As time passed on, the RS gained an increasing amount of data by capturing transaction details, ratings, etc., of a customer. Upon suggested with a product, the event of a user accepting or declining the suggestion which in turn becomes an input for the RS.

To summarize, the main purpose of RS in e-commerce is: [11]

To satisfy the users by selling a wider range of products by understanding what they need.

Two basic types of RSs are: [5]

- A Content-Based system that focuses on the attributes of the products.
- Collaborative-Filtering system that focuses on the relationship between the users and the products. [3]

Developing RSs is a combined multi-disciplinary effort that has benefitted from various fields of computer science and technology especially Machine Learning, Information Retrieval, Data Mining and Human-Computer interaction [2]. Machine learning allows the computer to learn to optimally perform a task using past experiences. Using the given data as a training set, the recommendation system builds a classifier for each customer that predicts the rating of all items making use of the concept of utility matrices.[4]

## **2. Proposed Model**

While recommendations can be given using the user's history from a different application, this work discusses the setting of a frequently viewed placement of a cart page. This page displays the seed product (product added to the cart), with five recommendations shown below, any of which could be bought together with the seed. The factors for giving recommendations include similarity amongst the product added to the cart and the recommended product and the probability that the user will purchase the bundle. The system has two stages. As a first step, a subset of relevant products to the product added to the cart is retrieved. Following this, the subset of possible recommendations enhancing the chance of conversion is ranked.

### **2.1 Objective**

In this work, the goal is to maximize conversion, that is, to convert a customer visiting customer (visitor) to buying a product from the site. This conversion can be a result of an upsell - encouraging customers to purchase a comparable higher-end product than the one in cart (for example, when a customer adds a previous generation Mobile to the cart, a current release model can be recommended as a possible upgrade) or a cross sell - encouraging customers to buy related or complementary items to the one in cart (for example, when a

customer adds a Smartphone to the cart, a smartphone-compatible accessory can be recommended for a co-purchase) [6]. Although both are significantly beneficial in improving sales, recommending an upselling product, which might be more expensive than the one in cart, has a shortfall in losing the customer journey so far and thereby losing a potential sale. Thus, the techniques for generating a recommendation focus majorly on making a cross sell.

To achieve an improvement of this metric, a recommendation system to suggest similar products to the ones that are added to the cart is created. Further sections explain the nature and type of data that was dealt with, the techniques and system architecture of the proposed model, evaluation of the proposed model based on different metrics.

## 2.2. Architecture of the Proposed Model

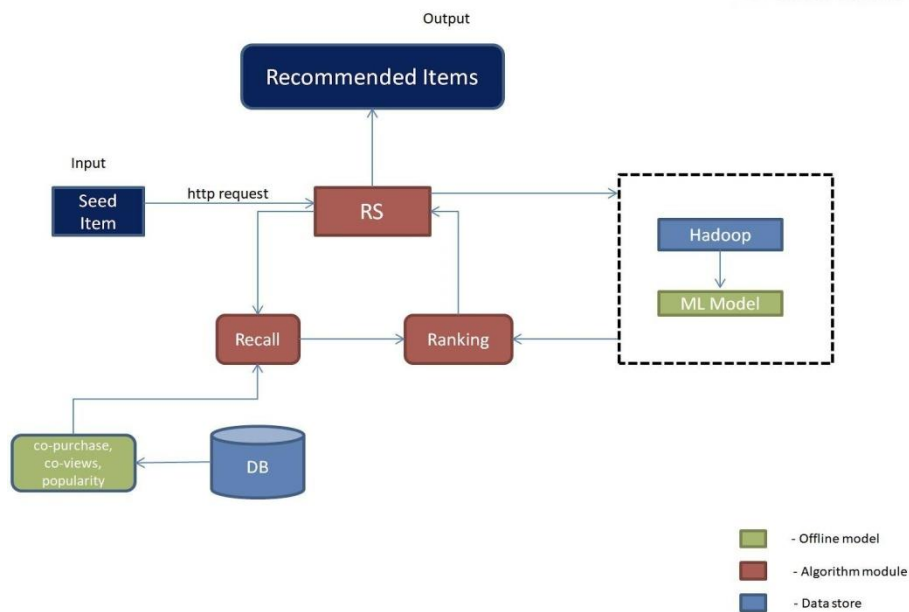


Figure 1: Architecture of the Recommendation system

The recommendation engine's architecture is split into two sections: printed and digital systems (the online components contains click history logs, behavioural data, and consists of a classifier on previous historical data to determine probability of the purchase to happen). The various aspects of a recommendation process are similar to those of a search problem, which has two phases: **recollection**, which is the process of retrieving candidates that match the seed item in focus, and **ranking**, which is the process of ordering the pool of options in order to estimate the probability of being decided to buy together. Figure 1 depicts a broad opinion of the backend architecture. The algorithm is fed data in the form of an HTTP request to the recommendation system and a seed item. This sends multiple concurrent requests to various services, which yield candidate recommendations that are similar to the seed item in a certain way. During the ranking process, the set of candidate recommendations is then ranked. The system's output is the top 5 items, which are then presented to the user.

## 2.3 Recollection

Using implicit user feedback (item purchases) with traditional collaborative filtering methods alone does not work due to the extreme sparsity of the user-item matrix. However,

performing item-based collaborative filtering on aggregations of item-level implicit user data makes sense.

A natural choice for choosing the appropriate level of aggregation would be to aggregate items at the category level. The current context has a category taxonomy and all items belong to a specific leaf category in this category tree. We can aggregate user purchases to form the user-category matrix, where the columns represent leaf categories and the entries in the matrix are either 1 if the user has purchased from that category or 0 otherwise. We then use cosine similarity, with appropriate thresholds, to find the top-K nearest categories to the input seed category.

Finding the nearest categories (related categories) constrains our search space of possible recommendation candidate items significantly and also reduces the possibility of irrelevant recommendations. All of the items that we recommend, in all the recall sets described below, will come from these related categories.

A way to decide whether to include the original seed category in the list of related categories is also discussed. Most of the time when performing a K-nearest neighbour (KNN) search, the input entity is not included in the search results. However, in the case of categories, it is possible to want to recommend items from the same category as the category of the seed item. For example, in the case where a customer buys an accessory, it is more relevant to suggest a different product from the same accessory category than suggesting a smartphone or a TV which the customer may not buy given the current setting of the cart. We capture this logic with the following heuristic: we calculate  $d$ , the mean number of purchases/users for each category. If the value of  $d$  for a specific seed category is above a threshold, we include the seed category as a related category and exclude it otherwise.

After obtaining a first-level relevance filter (related categories), actual candidate recommendation items are generated. A set of such candidate items is referred to as a Recall Set. The input to generating the recall sets is the information about the seed item. This is a very strong piece of context, so it is imperative that the recommendations shown to the user have some relevance to the seed item. As seen in the previous section, we use the seed category to generate a set of related categories.

Some of the ways to generate candidate items for recommendations using a variety of signals are:

**Related Products:** This recall set uses the collaborative filtering approach seen in the previous section, but is aggregated at the product level. We generate a recall set of related products by taking the cosine similarity of vectors of implicit feedback in the form of product-level purchase data. The relevance quality of the recommended products depends highly on the minimum thresholds set for the Otsuka-Ochiai coefficient (Cosine Similarity for binary data).

**Co-views:** While the last recall set utilized the purchase behavioural signal, this recall set utilizes the view behavioural signal. An item purchase is an ultimate sign of user intention. While a view signal carries less intention, for instance, a user might simply be browsing, the benefit to using this signal is the sheer increase in volume/coverage of recommendations. We use this signal to generate recall sets at the product level directly since co-view data is dense enough. Recall sets that use the co-view signal are high quality in terms of conversion.

**Compatibility:** Issues with compatibility between the seed and recommended items can be a serious concern for quality in hard goods categories such as electronics. Suggesting an incompatible product is a bad user experience that will make the user lose trust in recommendations. In general, there is an implicit user assumption that recommended items

will fit well with the seed item. Therefore it is important to take compatibility into consideration when generating complementary recommendations.

The above sections discussed sources of recall that use behavioural signals in some way. Often, behavioural signals are not available (cold start problem) so content-based signals are used to generate recommendations.

**Complementary of Similar:** Often we encounter the situation when there will be complementary recommendations for a product (Ex: “Silver Color Mobile”), but a nearly identical product (Ex: “Black Color mobile”) will not have results, perhaps due to lack of behavioural data for instance. This problem is addressed by developing a “complementary of similar product” type algorithm. Product embedding is generated using textual information from the product title and aspects to find similar products with a KNN search of the product embedding. Therefore, when there are no direct complementary results from a seed product, this recall set will return complementary items from similar products (from product embedding).

**Popular:** Popular items in a related category recall set can be used when there are no other behaviour or content-based signals that can be used. Due to its low relevance quality, this recall set is not used in all versions of the algorithm and it is used as a baseline when developing new recall sets.

#### **2.4 Ranking Models**

In the context of serving item recommendations, the problem is reduced to a binary classification problem. Here, the ranking is based on the probability of a recommended product being purchased(positive class). To generate training and testing data set, traits are derived from the initial seed product, recommended product pairs and the binary class labels(0 for non-clicked and 1 for purchased). During production, the Recollect stage will provide all the possible recommended product pairs. After that at the Ranking stage, the model will assign a likelihood score to all recommended product pairs from the previous stage. At last, the top 5 product pairs with the highest likelihood will be shown to the users as recommendations.

#### **2.5 Strategy**

The initial considerations to rank the list of recommendations by the probability of user clicks assumed the class labels of the model to be “non-clicked” and “clicked”. Since users search for various needs and not just for buying a specific product, there will be noises in the user click patterns and so the classes are not well distinguished. But from this, the user’s intention can be derived, the non-clicked product recommendations are the ones the users have zero interest whereas the purchased recommended products indicate the user’s interest in that product(conversion). Clicked recommendations but not purchased recommended products cannot be used as effectively for classification. The users might have clicked the product by mistake or they might have not found what they intended or they might have saved the product for a future buy.

The purchased recommendations, a subset of product recommendations are recommended more(occurs more in the product recommendations) than the clicked recommendations. Since there is an extreme class imbalance, the subsampling process is there to balance the classes, which will further improve the classifier performance.

#### **2.6 Feature Engineering**

Comparison features and product quality features are the two elements to be considered in this model. The seed item’s traits are analogized to the recommendation candidate item’s characteristics through comparison attributes. Assessing pricing, formats, and headlines (using TF/IDF-based algorithms) is indeed an example [9]. The quality affirmation features are designed to make sure that the user is provided with things of superior quality. These factors

include, for example, the item's popularity in the category and how frequently it is purchased, among several others.

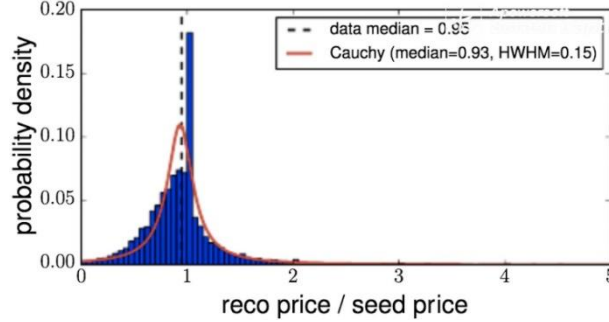


Figure 2: Price ratio and Cauchy fit histogram

This section will also examine and describe the specifics of one of the most important comparative features: the price function [7]. We analyze the ratio  $p_{\text{reco}} / p_{\text{seed}}$  rather than a straight comparison of the seed item price,  $p_{\text{seed}}$ , to the recommended price of the item,  $p_{\text{reco}}$ .

The ratio is centred at unity, according to the analysis. A normalised histogram of the price ratio (blue bars) from prior purchase occurrences is shown in Figure 2. The user first looked at the seed item before purchasing the suggested item. Since it provides the best fit for prior purchase data, the Cauchy distribution is employed to represent the model [12]. The price feature score is then generated from a normalized Cauchy distribution  $s_{\text{price}} = f_{\text{Cauchy}}(x, x_0, \gamma) \cdot \pi\gamma$ , which is 1 when  $p_{\text{reco}} = p_{\text{seed}}$  and smoothly transitioning to 0 otherwise.

### 3. Experiments And Evaluation

#### 3.1 Data Description

The website stores both the traffic (click-based data) and transactional (order-related) data in their persistent storage. The data for this work requires order details as well as user click history. From the transactional data, features such as order id, cart id, names of products added to the cart, the taxonomy of the products, features such as price, offers applied etc are obtained. The user click history provides data on the sequence of product pages visited by the user, products that were bookmarked etc. For simplicity, we use only order-related data for maximizing the impact currently.

#### 3.2 Evaluation

A linear model with manually adjusted weights receives inputs from subject matter experts and human interpretation of test output is implemented as the baseline ranking model for discovering similarities. A Naïve Bayes model and a model that returns results ranked by candidate item popularity.

On a balanced dataset containing non-clicked / purchased recommendation class labels, we analyzed the performance of classification techniques. The dataset was randomly divided into two sets, with the training set and validation set each receiving 80% and 20% of the total. The ROC-AUC values from the validation data set are given in Table 1 and show an improvement over the baseline model. The logistic regression's accuracy (0.70), positive class precision (0.70), and recall (0.70) all suggested satisfactory classifier performance. We used

the raw unsampled recommendations from impressions that featured a purchase, omitting impressions used for training, to evaluate ranking performance. We applied the normalised discounted cumulative gain truncated at rank k (NDCG@k) metric, which is typically used to evaluate ranking performance, to assess the performance of the ranking.

Table 1: Classification and Ranking Metrics

Classifier	AUC	NDCG, k=1	NDCG, k=5
Baseline	0.56	0.366	0.634
Naive Bayes	0.67	0.421	0.722
Logistic Regression	0.79	0.393	0.729
Decision Trees	0.82	0.397	0.778
Random Forest	0.86	0.335	0.735
Gradient Boosting	0.87	0.356	0.724

#### 4. Conclusion

In this research work, a highly scalable recommendation engine architecture is presented. It produces high-quality similar item suggestions in a diverse digital e-commerce space. An extensively applicable and interpretable pointwise machine-learned ranking model trained on implicit user shopping behaviour is developed. The model optimizes the ranking of recommendations based on the probability of purchase. While a majority of the e-commerce space characteristics are unique, the ranking model and sampling strategy are general enough for most domains that require ranking recommendations against a seed item. The future scope of the work consists of including new features, implementing better-performing online classifiers for the runtime ranking model, and incorporating user personalization and segmentation as well.

#### References

- [1] Elahi, M., Beheshti, A., & Goluguri, S. R. (2021). Recommender Systems: Challenges and Opportunities in the Age of Big Data and Artificial Intelligence. In *Data Science and Its Applications* (pp. 15-39). Chapman and Hall/CRC.
- [2] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). *Mining of massive data sets*. Cambridge university press.
- [3] Jiang, M., Zhang, Z., Jiang, J., Wang, Q., & Pei, Z. (2019). A collaborative filtering recommendation algorithm based on information theory and bi-clustering. *Neural Computing and Applications*, 31(12), 8279-8287.
- [4] Sheu, H. S., & Li, S. (2020, September). Context-aware graph embedding for session-based news recommendation. In *Fourteenth ACM conference on recommender systems* (pp. 657-662).
- [5] Kunaver, M., & Požrl, T. (2017). Diversity in recommender systems—A survey. *Knowledge-based systems*, 123, 154-162.
- [6] Ghoshal, A., Mookerjee, V. S., & Sarkar, S. (2021). Recommendations and Cross-selling: Pricing Strategies when Personalizing Firms Cross-sell. *Journal of Management Information Systems*, 38(2), 430-456.
- [7] Jannach, D., & Adomavicius, G. (2017). Price and profit awareness in recommender systems. arXiv preprint arXiv:1707.08029.

- [8] Yerasani, S., Appam, D., Sarma, M., & Tiwari, M. K. (2019). Estimation and maximization of user influence in social networks. *International Journal of Information Management*, 47, 44-51.
- [9] Kocas, C., Pauwels, K., & Bohlmann, J. D. (2018). Pricing best sellers and traffic generators: the role of asymmetric cross-selling. *Journal of Interactive Marketing*, 41, 28-43.
- [10] Pham, T. (2018). THE IMPACTS OF CULTURAL VALUES ON ELECTRONIC WORD OF MOUTH: OPINION LEADERSHIP AND OPINION SEEKING.
- [11] D. S. Vijayan, A. Mohan, J. J. Daniel, V. Gokulnath, B. Saravanan, and P. D. Kumar, "Experimental Investigation on the Ecofriendly External Wrapping of Glass Fiber Reinforced Polymer in Concrete Columns," vol. 2021, 2021.
- [12] Wang, E. (2021). Efficient Cauchy distribution based quantum state preparation by using the comparison algorithm. *AIP Advances*, 11(10), 105307.