Comparative Evaluation of Recommendation Algorithms for Implicit Feedback Data

Madhura Vishwanath Jadhav
MSIS, Pace University
NY, New York
madhura.jadhav@pace.edu

***Abstract*—A growing number of customers are switching to online shopping instead of traditional brick and mortar. To gain a competitive edge, recommender systems that can accurately provide personalized recommendations to their customers are becoming increasingly indispensable for ecommerce retailers. This paper presents an experimental evaluation of some of the popular Collaborative Filtering recommendation algorithms using implicit customer purchase data by one of the largest fashion retailer H&M. The primary task of the implemented recommender system was to recommend top-N products for each customer. This study examines three algorithms, namely, Item based Collaborative Filtering (IBCF), Bayesian Personalized Ranking (BPR) and Alternating Least Squares (ALS). These algorithms are evaluated based on the popular performance metric – Mean Average Precision i.e., MAP.**

Keywords—Recommendation Systems, Collaborative Filtering, Implicit Feedback, Item based Collaborative Filtering, Bayesian Personalized Ranking, Alternating Least Squares Matrix Factorization

I. INTRODUCTION

Recommendation systems became an important research area since the mid-1990s and over the last decade, lots of work has been done on developing new approaches for recommender systems. The interest in this area remains high because of its large number of practical applications. With rise in e-commerce industry, an important challenge is helping customers sort through a large variety of offered products to easily find the ones they will enjoy the most. Recommendation systems are addressing this challenge by providing users with personalized recommendations for products or services, which hopefully suit their unique taste and needs [7]. The technology behind these systems is based on predicting items that are likely to be the most appealing to users based on their preferences and historical purchases There are two major approaches to solving the recommendation problem: Content-Based Filtering (CBF) and Collaborative Filtering (CF) [1].

Content Based Filtering approach creates a profile for each user or product to characterize its nature. For example, user profiles might include demographic information like age, location, interests, etc. while movie profiles could include attributes like genre, actors, directors, box office popularity, etc. The resulting profiles allow recommendation systems to associate users with products they might be interested in [2].

Collaborative filtering approach, focus of this work, is the process of filtering or evaluating items using the opinions of other users. It analyzes relationships between users and interdependencies among products, in order to identify new user-item interactions. For example, some CF systems identify pairs of items that tend to be rated or purchased similarly or identify like-minded users with similar history of rating or purchasing to deduce unknown interaction between users and items [16].

Recommender systems depend on two different types of user feedback- Explicit and Implicit. Explicit feedback is clear input given by users about their preference for products. For example, movie or book ratings given by users. On the other hand, Implicit feedback indirectly reflects user opinion through observing their behavior. For example, purchase history or actions like save for later, add to cart on ecommerce websites. Implicit feedback comes with some important challenges:

1. Implicit feedback captures only positive user-item interactions. The missing data is ambiguous and is a mixture of actual negative feedback and unknown values [2]. For example, if a customer has never purchased a product, it can mean any of following: he/she might not like it, or not be aware of it, or not purchase it because of availability/price issues. This direction has not received much attention compared to rating prediction.

2. Implicit feedback does not necessarily indicate customer preference for the product. The product may have been purchased as a gift, or perhaps customer did not like the product and returned it after purchase.

A great deal of the literature in Collaborative Filtering is focused on processing explicit feedback. However, in many practical situations, recommender systems need to be centered on implicit feedback as explicit ratings are not always available [7]. For example, in retail industry customers do not usually provide an explicit rating on the products. In this case, their purchase transactions are considered as binary implicit feedback for implementing recommendation systems.

This paper conducts an experimental exploration of algorithms specifically suitable for processing implicit feedback. Various recommendation algorithms are systematically implemented and evaluated using purchase transactions dataset provided by fashion retailer H&M.

The rest of paper is organized as follows: Section II presents a brief theoretical foundation of the recommendation algorithms evaluated in this paper. Section III contains experiment details, including dataset and its characteristics, as well as the performance evaluation metric used for comparison. Section IV describes the analysis of experiment results. Finally, Section V concludes this research paper.

II. IMPLEMENTED ALGORITHMS

Collaborative Filtering (CF) models are based on the assumption that people like things similar to other things they like, and things that are liked by other people with similar tastes. Most CF recommendation algorithms start by finding a set of customers whose purchased products overlap with other customer's purchased products. The algorithm aggregates products from these similar customers, eliminates products the customer has already purchased, and recommends the remaining products to the customer.

To implement CF, general approach is - purchase matrix is created with users and items. Let U = {*u1, u2, ..., um*} be the set of all users and I = {*i1, i2, ..., in*} be the set of all items. For a given user u, Ru ⊆ I denotes the interaction i.e. purchase transaction. Purchase Matrix is developed in such a way that columns correspond to all the items from item inventory and rows represent all the users in dataset. An entry R*u,i* in the matrix contains a boolean value (0 or 1), where R*u,i* = 1 means that item *i* is bought by user u at least once and R*u,i* = 0 means that item *i* is never bought by user u. Based on different techniques, different computations are performed on this matrix.

Based on the implementation approach, CF recommendation can be divided into two, memory-based and model-based. The memory-based approach is centered on computing similarity between two users or items to generate recommendations. Typical examples of this approach are item-item and user-item recommendation. Alternatively, the model-based approach uses machine learning techniques to predict users' future purchases/ ratings. Matrix Factorization, Bayesian Personalized Ranking are some of the most popular model-based CF techniques.

A. Item based Collaborative Filtering

Neighborhood approaches can be divided into two main categories: user-item filtering and item-item filtering. A user-item filtering considers a particular user, find users that are similar to target user based on similarity of purchase history, and recommend items that those similar users purchased. In contrast, Item based filtering considers a target item, find users who purchased the same target item, and find other items that those users also purchased. The closest items to target item are calculated using similarity coefficients and are recommended to user.

According to literature around this topic, common similarity measurement methods are Cosine, Pearson, and Jacquard [4]. This experiment is conducted using Cosine similarity coefficient. Cosine similarity is calculated as the dot product of two vectors divided by the magnitude of each vector. For two items a and b, the cosine similarity between these items is calculated by using the formula given below.

$$\text{Cosine similarity } (a, b) = \frac{a.b}{||a||.||b||}$$

Higher the cosine similarity value, the items are considered to be more similar. Once the most similar items are found, the prediction is then computed by taking a weighted average of the target user's purchases on these similar items and top N items are selected for recommendation

B. Bayesian Personalized Ranking

Bayesian Personalized Ranking [5] is an optimization principle for CF methods, designed explicitly to deal with implicit feedback datasets. This method falls into the category of "learning-to-rank" methods as a general framework for pairwise learning. Different from matrix-factorization methods, it uses item pairs as training data and optimizes for correctly ranking item pairs instead of estimating scores for single items. This algorithm assumes that if a user u has expressed an implicit preference such as a purchase on an item *i*, then u prefers this item over all other non-observed items. Hence, a training instance in this case is a triple *(u, i, j)*, where it is assumed that customer u prefers item *i* over *j*. The set of all inferred preferences $\mathcal{D}_S$, i.e., the training data used for optimization, is defined as follows:

$$\mathcal{D}_S = \{(u, i, j) \mid i \in H_u \wedge j \in \mathcal{I} \setminus H_u\}$$

The generic optimization criterion is given as:

$$OPT(\mathcal{D}_S) = \text{argmax} \sum_{(u,i,j)\in\mathcal{D}_S} \ln \sigma(\hat{x}_{u,i,j}) - \lambda_\Theta \parallel \Theta \parallel^2$$

Where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid function, $\hat{x}$ *u,i,j* is the pairwise prediction for user u and items *i, j*. Θ is a parameter vector of an arbitrary model and λ_Θ is a model specific regularization parameter to prevent over-fitting the model.

$\hat{x}u,i,j$ is a real-valued function of . Θ which captures the relationship between customer u, product i and product j. The estimation of x̂u,i,j is performed through matrix-factorization but since it can only predict single scores, the estimator is decomposed into single prediction tasks: $\hat{x}u,i,j = \hat{x}u,i - \hat{x}u,j$. The optimization is performed using Stochastic Gradient Descent with bootstrap sampling of training triples using the following update rule:

$$\Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{x}_{u,i,j}}}{1 + e^{-\hat{x}_{u,i,j}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{u,i,j} + \lambda_\Theta \cdot \Theta \right)$$

where α is the learning rate.

C. Alternating Least Squares

Alternating Least Squares (ALS) is a Matrix Factorization based algorithm. Basically, Matrix Factorization approach mathematically reduce the dimensionality of the large matrix and factor it into its smaller representations. ALS uses iterative optimization process, where for every iteration it attempts to reach closer to a factorized representation of the original purchase matrix.

Consider original user-item matrix *R* of size *u * i*, containing purchase interactions. This matrix is turned into two matrices, one matrix with users and hidden features of size *u * f* and other with items and hidden features of size *f * i*. In *U* and *V* there are weights for how each user/item relates to each feature. Now, calculate *U* and *V* so that their product approximates *R* as closely as possible: $R \approx U * V$. By randomly assigning the values in *U* and *V* and using least squares iteratively, weights yielding the best approximation of *R* are found out. The least squares approach means fitting some line to the data, measuring the sum of squared distances from all points to the line and trying to get an optimal fit by minimizing this value. With the alternating least squares approach, the idea remains the same but iteratively alternate between

optimizing *U* and fixing *V* and vice versa. Koren et al extended their approach for the case of implicit feedback datasets [7].

In this experiment, the number of times a customer u purchased a product i is considered as the implicit rating, denoted r_{ui}. A set of binary variables p_{ui}. indicating the preference of product i with respect to customer u are introduced:

$$p_{ui} = \begin{cases} 1, & r_{ui} < 0 \\ 0, & r_{ui} = 0 \end{cases}$$

These binary preferences are associated with varying confidence levels. The confidence c_{ui} is defined as follows:

$$c_{ui} = 1 + \alpha \times r_{ui}$$

The confidence is calculated using the magnitude of the implicit feedback r_{ui}, giving us a larger confidence when a product is purchased many times by the same customer. The rate at which confidence increases is set through a linear scaling factor α, which is data-dependent and thus determined by a grid search over a set of values. Having addition of 1 is kept, so that a minimal confidence exists even if α * r equals zero.

The goal now, is to find a vector $x_u \in R^r$ and each customer u, and a vector $y_i \in R^r$ for each product *i* that will factor customer preferences.

$$min \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda\left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2\right)$$

The Alternating Least Squares method is used for the optimization [7] of the loss function and once the user and item vectors are computed, preferences are estimated as inner products: $p_{ui} = x_u^T y_i$.

To generate the top-N recommendations for a user *u*, all items *i* are sorted by decreasing scores of p_{ui} and the top-N products are recommended to user *u*.

III. Experiment

A. Dataset

H&M is a family of brands and businesses with 53 online markets and approximately 4,850 stores. They have extensive selection of fashion products including clothing and fashion through online website and offline retail stores. Original dataset consisted of approximately 10mn transactions for 1.4mn customers and 1mn products for timeline of 3 years. The data sets contain previous transactions, as well as customer and product meta data. The available meta data consists of various data points, such as garment type, customer age, text data of product descriptions, etc. This dataset was filtered considering system processing constraints. For this research purpose, most recent ~3 months i.e., July – September, 2020 purchase data is considered. Also, only womenswear segment is considered which comprises of different product types like dress, trousers, tops, blouses, jackets, shirts, nightwear etc. Certain niche product categories like maternity wear are excluded. This led to a dataset whose basic characteristic are summarized in Table I. It consists of 77,142 customers and 915 products, with 123,481 corresponding customers × product transactions.

TABLE I. SUMMARY OF THE FILTERED DATASET

Product Domain	Fashion - Womenswear
Time Span	July 2020 -Sept 2020
Total Customers	77,142
Total Products	915
Total Transactions	123,481

B. Evaluation Metric

The widely used strategy for evaluating recommendation algorithms in data science experiments is to randomly split the data into training and test sets. However, this setting does not reflect well the reality in the retail context as it is time agnostic. The availability of interaction (transaction) date in the transaction's dataset allows us to split and train the algorithms on past purchases and test the recommendations on future purchases. The dataset is split to use transaction records from July 1st, 2020 to Sept 15th, 2020 for training the algorithms and records from Sept 16th, 2020 to Sept 22nd, 2020 for testing them.

For all test customers each algorithm outputs a list of top-12 products. Recommendation lists are then evaluated for all customers, regardless of whether these customers made purchases in the training data. Customer that did not make any purchase during test period are excluded from the scoring.

Evaluation is performed according to the Mean Average Precision @ 12 (MAP@12):

$$MAP@12 = \frac{1}{U}\sum_{u=1}^{U} \frac{1}{min(m, 12)} \sum_{k=1}^{min(n,12)} P(k) \times rel(k)$$

Where U is the number of customers, P(k) is the precision at cutoff k, n is the number predictions per customer, m is the number of ground truth values per customer, and rel(k) is an indicator function equaling 1 if the item at rank k is a relevant (correct) label, zero otherwise [11].

IV. Comparison Results

The values of MAP@12 and Precision for various algorithms are shown in Table II. It can be seen that ALSMF algorithm perform better than other algorithms, followed by IBCF. BPRMF has lowest performance compared to other two algorithms. The rather low performance of BPRMF is surprising, especially that this approach is specifically designed for implicit feedback datasets. A possible explanation lies in the choice of the model parameters. Some recent works [6] improve over BPRMF to account for the different types of implicit user feedbacks such as click and add-to-cart. However, in H&M dataset, the only available feedback is the purchase decision which renders the latest findings inapplicable.

TABLE II. EXPERIMENT RESULTS

Algorithm	**MAP@12**
Item based Collaborative Filtering (IBCF)	0.002834
Bayesian Personalized Ranking (BPR)	0.000325
Alternating Least Squares (ALS)	0.004774

V. Conclusion and future work

Recommender system is an important tool to enhance customer experience and increase conversion rate in e-commerce portals. Fashion retail is a niche area where explicit feedback is inexistent and implicit feedback is sparse as well as of binary nature because repeat purchases for same clothing item are unusual. Although great deal of research has been done for recommendation systems based on explicit input like ratings and non – binary implicit input like number of views or repeated purchases, this typical e-commerce setting use case has received limited exposure. In this applied research paper, I have studied the effectiveness of some of the popular collaborative filtering recommendation models using H&M purchase dataset. Experiments confirm the superior performance of Alternating Least Squares and Item based Collaborative Filtering to more advanced Bayesian Personalized Ranking when applied on binary purchase data.

In my future work, I plan to further evaluate implemented models by considering larger purchase history data. I also plan to segment customers based on their purchase frequency to see the relative performances of algorithms.

References

[1] Isinkaye, Folasade & Folajimi, Yetunde & Ojokoh, Bolanle. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal. 16. 10.1016/j.eij.2015.06.005

[2] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," in *Computer*, vol. 42, no. 8, pp. 30-37, Aug. 2009, doi: 10.1109/MC.2009.263.

[3] Hwangbo, Yujeong, et al. "Recommendation system with minimized transaction data." Data Science and Management 4 (2021): 40-45

[4] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," in IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan.-Feb. 2003, doi: 10.1109/MIC.2003.1167344

[5] Rendle, Steffen, et al. "BPR: Bayesian personalized ranking from implicit feedback." arXiv preprint arXiv:1205.2618 (2012)

[6] L. Lerche and D. Jannach, "Using graded implicit feedback for bayesian personalized ranking," in Proceedings of the 8th ACM Conference on Recommender systems. ACM, 2014, pp. 353–356

[7] Y. Hu, Y. Koren and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets," 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 263-272, doi: 10.1109/ICDM.2008.22

[8] Haein Kim, Geunho Yang, Hosang Jung, Sang Ho Lee, Jae Joon Ahn (2019) An Intelligent Product Recommendation Model to Reflect the Recent Purchasing Patterns of Customers. Mobile Networks and Applications

[9] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep Item-based Collaborative Filtering for Top-N Recommendation. ACM Trans. Inf. Syst. 37, 3, Article 33 (July 2019), 25 pages. DOI:https://doi.org/10.1145/3314578

[10] Huang, Zan & Zeng, Daniel Dajun & Chen, Hsiu-chin. (2007). A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce. IEEE Intelligent Systems. 22. 68-78. 10.1109/MIS.2007.4338497

[11] Valcarce, D., Bellogín, A., Parapar, J., & Castells, P. (2020). Assessing ranking metrics in top-N recommendation. Information Retrieval Journal, 23, 411-448

[12] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," in IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749, June 2005, doi: 10.1109/TKDE.2005.99

[13] Chakraborty, S.; Hoque, M.S.; Rahman Jeem, N.; Biswas, M.C.; Bardhan, D.; Lobaton, E. Fashion Recommendation Systems, Models and Methods: A Review. Informatics 2021, 8, 49.

[14] R Pan et al., "One-Class Collaborative Filtering," 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 502-511, doi: 10.1109/ICDM.2008.16

[15] Sarwar, Badrul & Karypis, George & Konstan, Joseph & Riedl, John. (2001). Item-based Collaborative Filtering Recommendation Algorithms. Proceedings of ACM World Wide Web Conference. 1. 10.1145/371920.372071.

[16] Yang, Chong & Yu, Xiaohui & Liu, Yang & Nie, Yanping & Wang, Yuanhong. (2016). Collaborative Filtering with Weighted Opinion Aspects. Neurocomputing. 210. 10.1016/j.neucom.2015.12.136.