Collaborative Filtering Based Recommender System Design For E-Commerce: A Case Study

Merve Artukarslan
Galatasaray University
Department of Industrial Engineering
Istanbul, Turkey
email: merveartukarslan@gmail.com

S. Emre Alptekin
Galatasaray University
Department of Industrial Engineering
Istanbul, Turkey
email: ealptekin@gsu.edu.tr

Abstract— **Recommender Systems (RS) are one of the core engagement functions for e-commerce industry. In a typical recommender system, customer and product data is analyzed and a prediction model is generated, which evaluates products for prospective customers. In terms of business value, it helps individuals identify their interest among an overwhelming variety of products. In this paper, a collaborative filtering based recommender system framework is proposed for Turkey's leading e-commerce platform hepsiburada. First of all, implicit feedback and customer-product prediction pairs are prepared from collected data. Second, a regularized Singular Value Decomposition (SVD) based matrix factorization model is established for Collaborative Filtering (CF). Customers and products are represented with latent factor vectors. This model is trained with implicit feedback, as the SVD problem is solved with Alternating Least Squares (ALS). Third, predictions are gathered from the CF model. Then, predictions are limited to ten-product recommendation sets. Finally, recommendations are evaluated by behavioral data generated by prospective customers. The initial results show that 19% of recommendations match customers' interests.**

Keywords- Collaborative Filtering; Singular Value Decomposition; Alternating Least Squares; Recommender Systems; Matrix Factorization.

I. INTRODUCTION

Nowadays, e-commerce platforms accommodate a high diversity of choices for a vast number of visitors. Under these circumstances, there are two anticipated challenges. One of them is to expedite the decision making process for the prospective customers and the other one is scaling the technological solutions for high demand.

People usually consider the recommendations and mentions of their peers for purchase decisions. Recommender systems in this context are intelligent pieces of software, which interpret the digital footprint of users and products and then predict users' future behavior or requirements. They count as one of the core engagement functions of modern online retail businesses. These tools serve users with personalized recommendations suiting their unique desires and tastes via different feedback mechanisms.

Explicit feedback is defined as the categorical assessment of the customer for a product regarding their interest such as star ratings. Implicit feedback is the customer behavior inferring the user's preferences. Purchase history, browsing history, search patterns or page view period are examples of implicit feedback. Explicit feedback is preferred as it leads to a pure classified information. However, implicit feedback is less limited in terms of data collection effort. The numerical value of explicit feedback represents the customer preference, while implicit feedback supports its confidence [1].

This study aims to build a comprehensive recommender system for an e-commerce platform by targeting the prospective customers based on behavior. Recommender systems are introduced in Section 2. Value proposition of implicit feedback and collaborative filtering are analyzed to introduce the essence of customer preference notion. Section 3 consists of the proposed methodology. The purpose behind using matrix factorization and ALS is explained. The utilization of ALS with weighted λ regularization is detailed. Then, implicit feedback data with confidence level approach is introduced. Section 4 presents a business case implementation. We fine-tune the model parameters and hyper parameters of the SVD problem. We retrieve predictions from the model for the predefined customer-product pairs. Finally, prediction and recall metrics are calculated and analyzed. Future work and improvements are also discussed in Section 5.

II. RECOMMENDATION SYSTEMS

CF is a recommendation algorithm based on selecting and aggregating other users' behavior and ratings. It was first articulated by Goldberg in 1992 as a collaboration of people to aid one another to execute document filtering by interpreting readers' reaction to documents they read [2]. A user's preference is predicted, in a way, by interpreting others opinions. If users agree about the relevance of certain items, they will likely agree about others. CF highlights the serendipity of recommendations [3].

Besides CF, there are also other recommendation methodologies. Content based recommendations are based on user and product profiles which require external information and strategy. User content and product content are associated for recommendations. However, CF is based on users' behavioral history. Users' analytical judgments for the products they use are shared for better decisions. A personalized recommendation set is the deliverable of a CF based model.

Comparing to content based filtering, collaborative filtering has several major advantages. First of all, in CF, the only information needed is a user showing an intention to a product. Second, CF engages as a product satisfies a user's wishes. Hence, it aims to be more than a mere content analysis. Quality or taste as woven within human decisions is incorporated into the recommendation. Third, people may make desirable decisions by accident. The CF technique can generate serendipitous recommendations, which are valuable to the user but not expected according to the content of the product or user [4]. Despite the advantages, there are also disadvantages of collaborative filtering. Regarding the sparsity of user preference, it is not easy to find users with similar intentions. Recommendations may include many similar products and outliers may bias the model.

There are two frequently used approaches for CF. The first one is based on neighborhood methods discovering the relationships within the users or products. The second approach is based on latent factors models. A simple recommender system models the similarities between people or products. A latent factor model tackles the problem with a more sophisticated approach by converting data into a theme space. Then, the similarities in this theme space are explored. Latent factor models are preferred as latent space explains ratings by characterizing both users and products as factors inferred from implicit feedback [5].

In an user to user CF, one's predicted preference is based on the similarity with other users in terms of common ratings [6]. There are several methods for calculating user similarities. Pearson correlation [4] finds the statistical correlation between two users' common ratings to determine similarity. One pitfall of this method is that it may result in a high similarity between users that have few ratings in common. Constrained Pearson correlation scales the ratings in a like-dislike range [7]. Spearman rank correlation coefficient [4] is another method. It is derived from Pearson, except the ratings are replaced by ranks. Cosine similarity is a vector-space approach where users are represented by item rating vectors and similarity is measured by cosine distance between item rating vectors. The cosine distance is calculated by dividing the dot product of two vectors by the product of their Euclidean norms [8]. Item to item CF, on the other hand, uses the similarities between the rating patterns of items. The similarity of items can be calculated by the methods mentioned for user similarity.

Schafer et al. [15] explained how recommender systems bring value to e-commerce systems and analyzed different e-commerce platforms. RS enhance e-commerce services by converting browsers into buyers by utilizing sorting tasks for users. As mentioned in [15], Amazon book recommendations are based on 'customers who bought' strategy which refers to user similarity. They also recommend authors other than items. EBay builds a feedback profile feature allowing buyers and merchants to provide satisfaction rating. Feedback is used for merchant recommendation for users. They also have a personal shopper feature, which allows users to flag items they are interested in.

User's preference for an item in a recommender system is represented by the combination of the user's interest in the topic and an item's relevance to the topic. Hence, user-item ratings are established using a vector-space, which is likely to be high dimensional. This high dimensional model representation constructs a purchase history vector for each customer by producing one output for a set of inputs at a time. Prospective customers are targeted and similar customers are identified with cosine similarity based on the purchase history vectors of customers [9]. To increase robustness of the model and simplify model training, dimensionality reduction of rating space by dropping the singular values is recommended. This conversion helps to reduce noise in data and results in higher quality recommendations, as strong trends in the model are kept [10].

Similarity evaluations in CF approaches have to deal with the sparsity of data and dependency to common rated items. Solutions have been proposed to deal with these issues [11]. Wang et al. [11] developed an extend Proximity–Significance–Singularity model combined with item similarity. Their approach was tested in various sparse data sets and their results promise flexibility and break the constraint of common rated items. Furthermore, CF literature makes use of matrix factorization methodology to increase the level of accuracy and scalability. Single value decomposition technique as part of matrix factorization is applied to identify latent semantic factors. The latent space characterizes products and users on factors which are a form of user feedback and demonstrate ratings. The user and item latent factors are calculated using the alternating least squares technique which solves the optimization problem by fixing in each iteration either user latent factors or item latent factors and solving for the other and iterating until conversion [13].

In order to deal with limited data availability in recommender systems, implicit feedback based recommender systems have been developed [12]. The model is optimized by minimizing a ranking objective problem instead of the conventional mean square error. The key components of this model are a matrix factorization model, a ranking based objective function and an optimizer [12].

In real life scenarios, a vast number of user-item pairs complicates the optimization process. Methodologies demonstrated as in [1] make use of SVD for implicit feedback dataset-based collaborative filtering applications. Since the cost function of an SVD contains a vast number of user-item pairs, this minimization problem cannot be solved by a conventional technique, such as stochastic gradient descent. Hence, the quadratic nature of the cost function of ALS methodology proves useful as its complexity linearly increases in data size [1]. Moreover, as proposed in [1], the data sparsity and dense cost function could be dealt with confidence levels implementations. New factor models are also proposed by dividing ratings into confidence level and prediction [1].

III. Proposed Methodology

User and item similarity based recommender systems are implemented for e-commerce systems. However, they

 ISBN: 978-1-61208-728-3

require manual effort for content profiling and are based on unchanged customer behavior, contrary to real life. User similarity based CF is preferred, as it is close to a persistent and automated process, yet it assumes the sessions are long enough to learn about customer patterns [15]. CF with implicit feedback data has major advantages such as behavioral data dependency, focusing on customer preference, serendipity of recommendations and efficient responsiveness for cold starters. As mentioned in [14], CF techniques are commonly utilized for suggesting products to users. Considering the penetration and traffic of digital retail services, scalability and user profile sparseness problems eventually arises. [14] proposed an ALS algorithm with weighted λ regularization, which tackles these problems.

A. *Alternating Least Squares for Optimizing Singular Value Decompostion Problem*

Matrix factorization is a method used for latent factor models. It characterizes users and items as vectors of factors inferred from item rating patterns. A valuable recommendation carries high correspondence within user and item factors. This method is preferred for two reasons, it is scalable and accurate in predictions. Matrix factorization models fit users and items into a latent factor space of dimensionality. User-item interactions are considered as inner products in this space.

Singular value decomposition is a technique for identifying latent semantic factors in information retrieval. In collaborative filtering, it is applied to the user-item rating matrix. To learn the factor vectors, the model minimizes the regularized squared error on the set of known ratings. The learning model is generated by fitting the previous quantitative implicit feedback data in terms of ratings. The overall goal of a model is to reuse the model for unknown rating predictions [13].

Alternating Least Squares factorizes a rating matrix into two factors, user and item matrices, having the number of latent factors as row dimension. By fixing one of the matrices, the problem becomes quadratic, which can be solved directly. Alternately, this step is applied to user and item matrices, and the matrix factorization problem is iteratively improved.

B. *Tackling Overfitting with Weighted λ Regularization*

Overfitting is considered as overtraining a model by feeding it noisy and inaccurate data. Therefore, we may end up with an unrealistic model. Regularization is implemented to reduce the variance of the model without increasing the bias. Bias is considered as the error of the model. Variance is the change in predictions observed with different training models. Therefore, a tuning parameter λ is added to the model to deal with variance. As the tuning parameter increases, it reduces the value of coefficients and variance.

An ALS with weighted λ regularization model is proposed in [14] for large scale collaborative filtering. The main purpose of weighted regularization is to ensure the model does not overfit with increased number of features (latent factors) or iterations. Besides that, the authors mentioned that only about 1% of the user-movie matrix has been observed, with the majority of ratings missing, which is a challenge for the training data. Our study applies the method of implicit data feedback, unlike explicit movie ratings, which is used for the Netflix Prize competition referred in [14]. Equation (1) below is used to calculate the objective function of the model.

$$F(u,i) = \sum_{(u,i)\in\mathcal{K}} (r_{ui} - q_i^T p_u)^2 + \lambda\left(\sum_i n_{q_i}\|q_i\|^2 + \sum_u n_{p_u}\|p_u\|^2\right) \tag{1}$$

where

$u: user$
$i: item$
$\mathcal{K}: user - item\ pairs\ in\ training\ data$
$r_{ui}: rating\ of\ user\ for\ an\ item$
$\hat{r}_{ui}: estimated\ rating\ of\ user\ for\ an\ item, q_i^T p_u$
$q_i: latent\ factor\ vector\ of\ item\ i, q_i \in R^f$
$p_u: latent\ factor\ vector\ of\ user\ u, p_u \in R^f$
$\lambda: regularization\ factor$
$n_i: number\ of\ items$
$n_u: number\ of\ users$
$n_{q_i}: number\ of\ ratings\ of\ item\ i$
$n_{p_u}: number\ of\ ratings\ of\ user\ u$
$I_u: Set\ of\ items\ that\ user\ u\ rated$
$I_i: Set\ of\ users\ that\ rated\ item\ i$
$Q: item\ feature\ matrix$
$P: user\ feature\ matrix$
$R: user - item\ matrix, \{r_{ui}\}_{n_u \times n_i}$
$Q_{I_u}: Submatrix\ of\ Q,\ i \in I_u$
$R(u, I_u): ratings\ row\ vector\ of\ items\ that\ user\ u\ rated$
$P_{I_U}: Submatrix\ of\ P,\ u \in I_i$
$R(I_i, i): ratings\ column\ vector\ of\ users\ that\ rated\ item\ i$
$n_f: feature\ dimension\ space$
$E: n_f \times n_f\ identity\ matrix$

$R = \{r_{ui}\}_{n_u \times n_i}$ represents the user-item matrix. Each element $\{r_{ui}\}$ represents the implicit feedback from customer u for item i. There is a user and item feature vector corresponding to each and every user and item, denoted by q_i and p_u, respectively. Each given and estimated rating, or implicit feedback, is the inner product of the corresponding latent factor vectors. The authors of [14] suggested to minimize the summation of loss of user and item feature matrices of known ratings, P and Q. The loss function is regularized for handling the overfitting of sparse data set.

Since the SVD algorithm is not able to find P and Q with a large number of missing ratings, ALS is applied. The minimization problem has two sets of decision variables as part of the optimization goal. Therefore, as one of the decision variables set is fixed to solve the problem for the remaining set, the problem is solved. As mentioned in the previous section, ALS rotates the problem by fixing item latent factors and user latent factors sequentially. The least squares computation problem is solved and the regularized squared error is decreased until convergence. ALS is preferred over gradient descent as it can use parallelization. ALS with weighted λ regularization will also address the scalability limitations related to the number of latent factors and the number of ALS epochs.

Matrix $Q = [q_i]$ is initialized by assigning the average rating for an item as the first row, and small random numbers for the remaining entries. Then, Q is fixed and $P = [p_u]$ is solved by minimizing the sum of squared error in the objective function. Then, P is fixed and Q is solved similarly. This rotation is repeated until the mean squared error converges. A given column of P, which latent factor vector of user u denoted as p_u, is determined by solving a regularized linear least squares problem involving the known ratings of user u and feature vectors q_i of the items that user u rated. p_u becomes an expression of (3) and (4), which is given in (2).

$$\frac{1}{2}\frac{\partial f}{\partial i_{kj}} = 0,\ \forall u,k$$
$$\Rightarrow \sum_{i\in I_u}(p_u^T q_i - r_{ui})q_{ki} + \lambda n_{p_u} p_{ku} = 0,\ \forall u,k$$
$$\Rightarrow \sum_{i\in I_u} q_{ki} p_u^T q_i + \lambda n_{p_u} p_{ku} = \sum_{i\in I_u} r_{ui} q_{ki},\ \forall u,k$$
$$\Rightarrow \left(Q_{I_u} Q_{I_u}^T + \lambda n_{p_u} E\right) p_u = Q_{I_u} R^T(u, I_u),\ \forall u$$

$$p_u = A_u^{-1} V_u,\ \forall u \qquad (2)$$

where

$$A_u = Q_{I_u} Q_{I_u}^T + \lambda n_{p_u} E \qquad (3)$$
$$V_u = Q_{I_u} R^T(u, I_u) \qquad (4)$$

Q_{I_u} denotes the sub-matrix of Q (item feature matrix) consisting of columns $i \in I_u$ (set of items rated by user u). $R(u, I_u)$ denotes the row vector retrieved from the u-th row of R (user-item matrix) for $i \in I_u$ (set of items rated by user u).

Similarly, when Q is updated, individual q_i can be computed via regularized linear least squares solution including the feature vectors of users who rated item i. q_i becomes an expression of (6) and (7), which is given in (5).

$$q_i = A_i^{-1} V_i,\ \forall i \qquad (5)$$
$$A_i = P_{I_i} P_{I_i}^T + \lambda n_{q_i} E \qquad (6)$$
$$V_i = P_{I_i} R^T(I_i, i) \qquad (7)$$

P_{I_U} denotes the sub-matrix of P (user feature matrix) consisting of columns $u \in I_i$ (set of users rated item i). $R(I_i, i)$ denotes the column vector retrieved from the i-th column of R (user-item matrix) for $u \in I_i$ (set of users rated item i) [14].

C. Confidence of Implicit Feedback

As suggested by Hu el al. [1], at this stage, we tried to identify the unique properties of implicit feedback data. The objected function stated in (1), which is based on ALS with weighted λ regularization, is extended in this step. t_{ui} is a binary set which indicates the preference of user u for item i. In other words, if user u has interacted to item i, t_{ui} is equal to 1. On the other hand, if user u never encountered item i, then, that preference is set equal to 0. Preference values are poor in confidence, as having no preference may have a variety of reasons other than not liking an item. Thus, a confidence level model representing the user's preference is required.

Consequently, as r_{ui} grows, the strength of preference should be increased. c_{ui} is measurement for the confidence in t_{ui} equals $(1+\propto r_{ui})$. The squared error part of the goal function $(r_{ui} - p_u^T q_i)^2$ is extended as $c_{ui}(t_{ui} - p_u^T q_i)^2$. Replacing the ratings with confidence values, A_u, V_u, A_i and V_i are updated, as shown in (8), (9), (10) and (11).

$$A_u = Q_{I_u} C^u Q_{I_u}^T + \lambda n_{p_u} E \qquad (8)$$
$$V_u = Q_{I_u} C^u R^T(u, I_u) \qquad (9)$$
$$A_i = P_{I_i} C^i P_{I_i}^T + \lambda n_{q_i} E \qquad (10)$$
$$V_i = P_{I_i} C^i R^T(I_i, i) \qquad (11)$$

C^u is a diagonal $n_i \times n_i$ matrix where $C_{ii}^u = c_{ui}$. C^i is a diagonal $n_u \times n_u$ matrix where $C_{uu}^i = c_{ui}$ [1].

IV. Case Study

A. Business Model & Proposed Framework

In this work, a collaborative filtering based recommendation engine is proposed for an e-retailer. Based on the number of visitors and products, the recommendation engine is utilized for Small Domestic Appliances (SDA) and Fast Moving Consumer Goods (FMCG) categories.

The behavioral events used in this study are listing page visit, product page visit, product added to cart, product saved for later, saved product added to cart and purchase. These events are fitted into a three phase sales funnel, displayed in Figure 1, for customer-product interaction association.

There are three major stages: discovery, intention and purchase. The more a customer carries a product within the funnel, the more involved the customer is. Every event is associated with a stage. Implicit feedback is determined by the last event for each user-item pair. The general framework proposed for this business case is visualized in Figure 2.

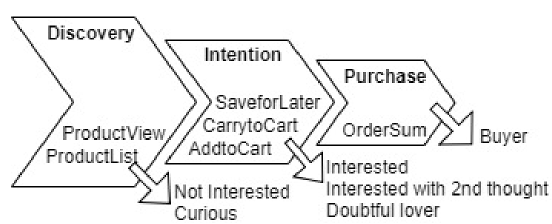

Figure 1. Sales Funnel Design

B. Training and Prediction Data Preparation

Implicit feedback is built on event correlations related to the pre-defined products. Collected events are filtered within a time interval of 4 weeks. The start date is selected as January 1st, 2019 and the end date is January 31st, 2019. Interactions are converted to numeric values from 1 to 6. Not interested is 1, curious is 2, interested is 3, interested with second thoughts is 4, doubtful lover is 5 and buyer is 6. The numeric rating values are directly proportional to the incline degree of the interaction portrait. The form of implicit feedback data is a (Customer, Product, Rating) tuple.

Prediction pairs are prepared for users who showed intention or purchased a kitchen appliance product on January 31st, 2019. In short, predictions are generated within one day of interaction by a CF model trained with 4 weeks of feedback data. The users who intended to buy an SDA product but did not purchase one are characterized as inclined users and the ones who purchased one as purchased users. For purchased users, the purchased category is excluded from predictions. For the inclined users, products belonging to the inclined category are predicted.

C. Recommendation Model Generation

The proposed model is prepared for evaluating user-item pairs by generating a prediction score. Weighted λ regularization is implemented in the model. Users and items are represented as latent factor vectors for SVD. The rank parameter defines the number of latent factors. The alpha parameter is used as multiplier for rebalancing rating data. The number of iterations represents the number of rotations for ALS. Apache Spark [16] is preferred as it is a large scale data processing platform, which enables parallelized operations.

Mean Squared Error (MSE) is used as optimization metric. When MSE converges, the problem is assumed to be optimized with given parameters. λ is set as 0.01 and model parameters are fine-tuned by MSE convergence. Rank is set from 2 to 128, alpha is sequentially set as 0.01, 0.1, 0.5 and 1. Consequently, how these parameters leverage the MSE is analyzed. Regarding the memory limitations of our sources, the number of iterations is selected as 10 epochs. The model is trained with different rank values when alpha is 1.0 and the number of epochs is 10. The convergence rate is analyzed and the rank is determined to be 60. The MSE is observed to decrease from 6.0488 to 4.3722 with an increased rank and fixed alpha and λ.

D. Tailored Recommendations

Despite the fact that predictions are generated for hundreds of products for each user, recommendations are limited to 10 as it is a realistic value for user experience. Regarding the business objectives such as showing the assortment of products and converting more of the cross-sale opportunities, these 10 items are divided into two subgroups. One subgroup represents the products with the highest predicted ranking. The other subgroup is tailored in accordance with business objectives. A total number of 6 items are the ones with highest predictions. 4 items are tailored to ensure that there are both FMCG and SDA products in a 10-product recommendation set.

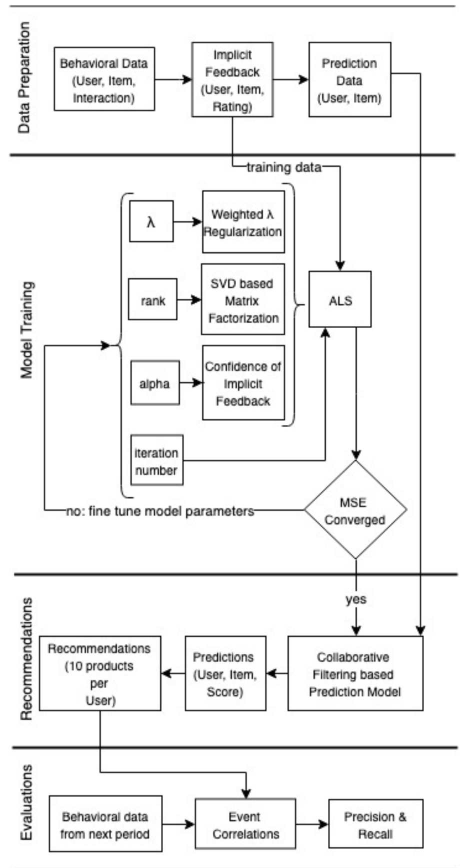

Figure 2. General Framework

E. Evaluation and Results

A subtle recommender system should be empowered by customer behavior, up to date, relevant yet unforeseen and personalized. These goals are addressed with the following steps:

- A total of 174626 implicit feedback data points were generated with 39926 customers for 1403 different products.
- With respect to the question of customer taste, ratings are decomposed with latent factors.
- The CF model is optimized for predicting prospective customer-product association strength with respect to proximity and serendipity. As regularization is applied, predictions are considered as adaptive and confident.
- Predictions are generated for 1209 customers and 565 products via the CF model and tailored recommendations are prepared.

In this study, we prepared the recommendations, however, they are not displayed to the customers. In order to evaluate the recommendations, ProductView and AddtoCart events of prospective customers were collected in February, 2019 and interpreted. The Customers who showed an intention to purchase SDA products were targeted as prospective customers. The expected retention of SDA purchasers is 4-5 weeks, given the nature of the purchased product. Therefore,

ISBN: 978-1-61208-728-3

customers were tracked for 4 weeks. 1209 prospective customers were tracked during February, 2019. Comprehensively, 19% of our customers ended up discovering what we predicted for them and 7% of our customers showed a purchase intention to what we predicted for them. Current recommendations perform between 2% and 8% depending on the strategy (e.g. customers who bought this, category based selections, and complementary products) and position (e.g. listing pages, product detail pages, and basket). Precision represents the engagement of the tailored recommendations. This metric is the ratio of true positives over predicted positives. Predicted positive is the customers we made recommendations. True positive is the customers who interacted with the products we recommend. In this study, precision is 0.19. These recommendations are not more than predictions without the proper positioning and marketing communication. Also, prospective customers are not evaluated with a retention perspective. It is inevitable to observe that most of them did not make a secondary purchase. Recall represents the coverage of tailored recommendations over all product interactions. This metric is the ratio of true positives over actual positives. Actual positives represents the total number of customers who interacted with a product from our product spectrum. In this study, recall is 0.76. 76% of prospective customers who visited the website within 4 weeks interacted with a product from our recommendations.

V. CONCLUSION

A comprehensive recommendation engine for Turkey's leading e-commerce platform hepsiburada is proposed in this study. Considering the accessibility of behavioral data and sophistication of customer taste, latent factors based collaborative filtering is applied. Implicit product feedback from customers is retrieved from data. Customers and products are represented by latent factors. A prediction model is generated by solving a dynamically regularized SVD problem with ALS. The model's training parameters are fine-tuned and predefined predictions are delivered.

This framework can be enhanced with further implementations. One of them is to update the model to display to the user an explanation of the strategy behind the recommendations. The examples are 'you are seeing this because people like you purchased this product' or 'you are seeing this because you purchased that product'. To inform the customer about the reason behind the recommendations is more trustworthy and the customer can know the coherence. The other improvement opportunity is to enrich the implicit feedback model with after sales data, such as review context, return status, replenishment status. Considering the visit numbers and high assortments, millions of events are generated every day. Our case study is limited in data. However, solving the problem with ALS and weighted λ regularization is suitable for big data. This framework can be extended for larger datasets.

Recommendations are generated for the customers who at least added an item to their basket within a given day. Thus, the cold start problem is excluded and the model can be trained on larger datasets and cold starters can be tested.

ACKNOWLEDGMENT

This research has been financially supported by Galatasaray University Research Fund, with the project number 19.402.007.

The data used in this work is provided by hepsiburada, one of Turkey's leading e-commerce platforms. Regarding the EU general data protection regulation and Turkish personal data protection law, customer data and behavioral data was completely anonymized and processed with internal resources.

REFERENCES

[1] Y. Hu, Y. Koren and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets," Eighth IEEE International Conference on Data Mining, pp. 263-272, 2008.

[2] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," Commun. ACM, vol. 35, no. 12, pp. 61-70, 1992.

[3] Z. Batmaz and H. Polat, "Randomization-Based Privacy-Preserving Frameworks for Collaborative Filtering," Procedia Computer Science, vol. 96, pp. 33-42, 2016.

[4] J. L. Herlocker, J. A. Konstan, L.G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," ACM Transactions on Information Systems (TOIS), vol. 22, no.1, pp. 5-53, 2004.

[5] F. Ricci, B. Shapira, L. Rokach, and P. Kantor, "Recommender Systems Handbook,", pp.1-35, 2011.

[6] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative Filtering Recommender Systems," Foundations and Trends® in Human–Computer Interaction, vol. 4, no. 2, pp. 81-173, 2011.

[7] N. Polatidis and C. K. Georgiadis, "A Dynamic Multi-Level Collaborative Filtering Method for Improved Recommendations," Computer Standards & Interfaces, vol. 51, pp. 14-21, 2017.

[8] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, "A New Similarity Measure Using Bhattacharyya Coefficient for Collaborative Filtering in Sparse Data", Knowledge-Based Systems, vol. 81, pp. 163-177, 2015.

[9] O.Y. Kasap and M. A. Tunga, "A Polynomial Modeling Based Algorithm in Top-N Recommendation," Expert Systems with Applications, vol. 79, pp. 313-321, 2017.

[10] D. Billsus and M. Pazzani, "Learning Collaborative Information Filters," ICML, vol. 98, pp. 46-54, 1998.

[11] Y. Wang, J. Deng, J. Gao, and P. Zhang, "A Hybrid User Similarity Model for Collaborative Filtering," Information Sciences, vol. 418, pp.102-118, 2017.

[12] G. Takács and D. Tikk, "Alternating Least Squares for Personalized Ranking," Sixth ACM Conference on Recommender Systems, pp. 83-90, 2012.

[13] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems", Computer, vol. 8, pp. 30-37, 2009.

[14] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-Scale Parallel Collaborative Filtering for the Netflix Prize," International Conference on Algorithmic Applications in Management, pp. 337-348, 2008.

[15] J. B. Schafer, J.A. Konstan, and J. Riedl, "E-Commerce Recommendation Applications", Data Mining and Knowledge Discovery, vol.5, no. 1-2, pp. 115-153, 2001.

[16] Meng et al., "MLlib: Machine Learning in Apache Spark", Journal of Machine Learning Research, vol.17, pp. 1-7, 2016.