An Adjusted Recommendation List Size Approach for Users' Multiple Item Preferences

Serhat Peker(✉) and Altan Kocyigit

Department of Information Systems, Middle East Technical University, 06800 Ankara, Turkey
{speker,kocyigit}@metu.edu.tr

Abstract. This paper describes the design and implementation of a novel approach to dynamically adjust the recommendation list size for multiple preferences of a user. By considering users' earlier preferences, machine learning techniques are employed to estimate the optimal recommendation list size according to current conditions of users. The proposed approach has been evaluated on real-life data from grocery shopping domain by conducting a series of experiments. The results show that the proposed approach achieves better overall recommendation quality than the standard approach and it outperforms the benchmark method in efficiency by shortening the recommendation list while maintaining the effectiveness.

Keywords: Top-N recommender systems · Recommendation list size · Recommendation length · Recommendation quality · Recommendation efficiency

1 Introduction

Nowadays, recommender systems are commonly used by commercial companies for the purpose of helping their customers on decision making process. These systems usually use two main recommendation strategies: "find all good items" and "recommend top-N items" [1]. The recommender systems using "find all good items" approach offer all the recommendable items that can suit the user's tastes, whereas in the "recommend top-N items" approach, only the top ranked N items are recommended to the user. The latter one is the most common solution for product recommendations and many commercial companies take the take advantage of this technique in their recommender systems [2, 3].

In the top-N recommender systems that employ "showing top k matching items" as the recommendation strategy, of the length recommendation list usually ranges from 5 to 20 [4]. It is possible to increase the proportion of items that are correctly identified (recall) by providing more items in the recommendation list. likely to However, users are not be overwhelmed by recommendation lists containing a large number of items, and it is also important to fit recommendation lists in small display devices such as mobile phones [4]. Moreover, increasing the number of recommended items, N, improves recall, but it is likely to deteriorate precision [1, 5, 6] which is the proportion of recommended items that result in matches. Together with recall, precision shows the

C. Dichev and G. Agre (Eds.): AIMSA 2016, LNAI 9883, pp. 310–319, 2016.
DOI: 10.1007/978-3-319-44748-3_30

quality of the recommendation, and high precision is more preferable in recommender systems using "recommend top-N items" approach [6].

Although, top-N recommender systems typically return a fixed number of items in each recommended list, individuals may have multiple preferences at a time or in a specific time interval, and the number of such preferences may vary depending on cases and context in which the individual is. For example, consider a man being time pressed during weekday daytime in the grocery shopping. He prefers to buy a couple of items (e.g., 3 unique items) on weekday evenings for daily needs such as bread, eggs, milk, etc., whereas he has too many items (e.g., 15 unique items) in his shopping basket on a weekend afternoon.

To evaluate the quality of existing recommender systems producing a fixed number of items for users with multiple preferences at a time, let us consider the above example again. Suppose that this man uses a recommendation agent in his smart phone for the grocery shopping and this application generates a fixed length of recommendation list containing 10 items for his next visit. For the first case, agent may probably perform with high recall and low precision, since it produces a long recommendation list for a number of purchased items, but agent also overwhelms the user with many irrelevant items. For the second case, on the other hand, recall is most probably lower than the one in first case and precision may higher, because the number of recommended items is closer to the number of preferred items. However, recommendation agent also misses some relevant items in this case, since it recommends fewer items than the user prefers.

As explained in the examples, for multiple preferences at a time, recommender systems returning a fixed number of items may cause some issues which are undesirable from the users' point of view, and it is obvious that the length of recommendation list has a significant impact on the recommendation quality. In this respect, this paper aims to dynamically adjust the recommendation list size of a user with multiple preferences by employing machine learning techniques. The proposed approach dynamically determines the optimal recommendation list size based on the previous preferences of the user. The applicability of the approach is experimentally evaluated by using real-life data obtained from the grocery shopping domain and the results show the effectiveness of our approach.

The remainder of this paper is organized as follows. Section 2 describes proposed approach. In Sect. 3, evaluation methodology is presented together with the experimental results. Finally, Sect. 4 concludes the study and points out directions for future work.

2 Proposed Approach

The major steps of our approach for adjusted recommendation list size are depicted in Fig. 1. First, a predictive model is constructed based on previous preferences of users. Then, identified features' values pertaining to the user and the current recommendation are used as input for the model to estimate an adjusted recommendation list size that will be finally used as a reference in the construction of recommendation list.

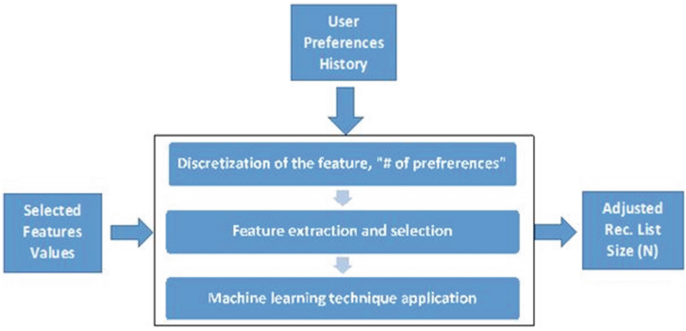

Fig. 1. Schematic overview of our approach

In our predictive model, "number of preferences", the numeric target value, is firstly discretized into a set of intervals in order to use machine learning classifiers. The reason behind this step is to construct a categorical variable in order to transform the problem into a classification one, and to employ powerful classification techniques. In case of a continuous target attribute, unsupervised discretization techniques are used to divide the variable into discrete intervals. Famous representatives of unsupervised techniques are equal-width and equal-frequency discretization [7, 8]. In this study, we use an equal-frequency discretization method, because the equal-width method does not perform well when the variable observations are not distributed evenly, and thereby may cause information loss after the discretization process [9]. Equal-frequency as an unsupervised method requires the user to specify the number of discrete intervals and fewer number of intervals is preferable in order to avoid the fragmentation problem occurring in classifiers [10]. Because of that, we set the number of intervals to three (as low, medium and high) in this study.

Not surprisingly, the same number of preferences may mean different things to different people. For instance, in grocery shopping, buying 10 items could be taken as too many to a user with an average number of purchased items of 5, whereas too few to a consumer with an average number of purchased items of 20. Hence, it is important to apply the discretization technique for the number of preferences at the individual level. Therefore, the proposed approach employs equal-frequency technique at the individual level to produce personalized cut points for each user.

After the "number of preferences" is transformed to categorical variable via equal-frequency discretization, features related to users' preferences are identified to build a model that produces better performance. At the final stage of the model construction, a machine learning technique is trained with users' previous preferences with selected features. In this study, we use two classifiers and a regression technique in order to compare their performance for recommendation quality. The classifiers used are decision tree (J48) and KNN algorithms. These classifiers are chosen because they are in the top 10 list of classification techniques [11]. For KNN method, we searched the optimal k value and identified it as 75. We also use multiple regression by coding the values of discretized "number of preferences" variable (low, medium and high) as 1, 2, and 3.

In this way, the feature is ordinal and the continuous nature of it is preserved which allow the researchers to employ regression analysis [12].

After training the model, identified features' values of the user and the current recommendation are used as input for the model. Then, it outputs the estimated number of preferences of the user for the recommendation list size. Note that, machine learning classifiers produce the output as a categorical value such as low, medium or high. On the other hand, regression model generates the output as numerical nominal value as 1, 2, or 3. Since, our approach employs the equal-frequency discretization method for "number of preferences", these produced values correspond to bins. Because of that, our approach replaces each bin value by its mean. Note that, if this value is decimal, it is rounded to nearest integer. Therefore, the proposed approach returns this final result as the possible recommendation list size.

3 Evaluation

In this study, we performed an offline evaluation technique to measure the effectiveness of our approach. We chose this evaluation method, since it does not require interaction with real users, and the performance of different recommender algorithms and approaches can be easily evaluated by using existing datasets at low cost [13]. Grocery shopping was chosen as the application domain, since customers in the grocery industry prefer multiple products in a single transaction. A comprehensive set of experiments were conducted on a dataset obtained from a Turkish retail grocery store. In the following subsections, we first describe the dataset and data pre-processing task. Next, we explain the experimental settings and evaluation measures. Then, the recommendation method that used in the experiments and a benchmark method for the comparison are introduced. Finally, we present the results we obtained.

3.1 Dataset

The dataset we used in the experiments contains purchase transactions gathered over 104 weeks (January 1, 2013 – December 31, 2014). In the dataset, many of the customers have visited the store irregularly (only a few times during the period) and transactions of these customers most probably mislead the results of the study. Thus, we pre-processed the dataset by excluding customers who have visited the store less than 25 times. After this elimination, the dataset is left with 46 customers with 534616 purchase records. In the dataset, products are categorized according to a four-level hierarchy. The lowest level includes information about the product brand, amount and etc. For ketchup, for example the third level category is "ketchup", and one of the examples of fourth level category for this product is "Tat Ketchup, 600 Grams". "Tat" is the brand of product, whereas "600 Grams" is the amount of it. Since the fourth level is highly specific, we considered third-level categories in the experiments. There are unique 335 third-level categories in the dataset. Customers within our sample bought 47.63 distinct third-level categories on average with the standard deviation of 26.07 and average basket size for our sample is 11.43 with the standard deviation of 6.61.

3.2 Data Pre-processing

This process mainly includes outlier elimination and feature extraction steps. One way to define an outlier is using interquartile range (IQR). If a value is more than (Q3 + 3*IQR) or less than (Q1 - 3*IQR), then it is considered as an extreme outlier. Note that, Q1 and Q3 represent the lower and upper quartiles, respectively. Outlier detection is applied to the "number of preferences" (number of unique products purchased). However, this variable may have different distribution for different customers. Because of that, we applied the outlier detection for each customer separately. Therefore, for each customer, we remove the instances having an outlier value for the corresponding variable.

Among existing contextual dimensions, temporal information is precious and easy-to-collect feature for increasing the performance of recommendations [14]. As a result, we considered timestamp of transactions as one of the features. Since timestamp is a composite variable in our dataset, we extracted two distinct features from this variable, which are day of the week and time of the day. We also categorized the values of these variables and Table 1 shows both categorical and actual values of each feature.

Table 1. Time features

Feature	Categorical values	Range of actual values
Day	Weekend	Monday to Friday
	Weekday	Saturday, Sunday
Time	Morning	08:00 to 11:59
	Afternoon	12:00 to 17:59
	Evening	18:00 to 20:59
	Night	21:00 to 22:59

In the experiments, categorical values were used for the features of time and day. Moreover, simple n-visit moving average feature which is the average number of purchased unique products in last n visit is calculated. We selected n as 5. The number of purchased unique products in last visit is also computed and formed as another feature.

3.3 Experimental Settings and Design

To generate the training and test sets, we sorted each customer's shopping trips according to their timestamps. For each customer's purchase history, we use the first 80 % of the visits as training and the latter 20 % as test data for all set of experiments. The training data is utilized to train the predictive model of our approach. In the testing set, for each visit of the customer a ranked list of recommended products is generated and the recommended products are compared with the actual ones purchased by the customer in the test set to compute the corresponding evaluation metric.

3.4 Evaluation Metrics

Recall and precision, which are first originated in the field of information retrieval, have been widely used in the performance evaluation of top-N recommender systems [1, 15]. These metrics are defined as follows:

$$Recall@N = \frac{|rec@N \cap purc|}{|purc|} \tag{1}$$

$$Precision@N = \frac{|rec@N \cap purc|}{|rec@N|} \tag{2}$$

where rec@N donates the top-N recommended products for the test instance and purc is the actual product set that the customer has purchased in the same test instance. However, as known well, there is a tradeoff between these two measures. For instance, increasing the number N tends to increase recall but is likely to reduce precision. Since both are critical measures in the quality assessment, F-measure [16] which is the harmonic mean of precision and recall was used in the performance evaluation. It is computed as:

$$F\text{-}measure = 2 \times \frac{recall \times precision}{recall + precision} \tag{3}$$

We first computed F-measure for each customer separately by averaging all computed values in the customer's test set, and then average these customers' personal values to get overall F-measure value for a given recommendation list size, N.

3.5 Recommendation Method

In the experimental evaluation, most-frequent item recommendation approach [5] was used to generate top-N recommendation lists for the customers' visits in the test set. The reasons behind this choice are that it is simple and well-known method and it was also employed in similar studies on predicting grocery shopping lists [5, 17, 18]. This approach sorts the products in customer's purchase history according to their frequency count and simply returns the N most frequently products as the current shopping list. This approach may not be so efficient in the recommendation performance, but it is not our aim in this study. Most-frequent item recommendation method is only considered as a reasonable baseline to predict top-N recommendation lists and to evaluate the effectiveness of our proposed approach.

3.6 Benchmark Method

To compare the performance of our proposed approach on the recommendation efficiency, we selected last visit's N as a benchmark. This method identifies the number of products purchased in the previous visit of the customer as the recommendation list size for the next visit of that customer. Additionally, we also conducted experiments by

varying the recommendation list size, N from 5 to 15 (in increments of 1) for comparing our approach on adjusted recommendation list size with the standard one using different fixed length of recommendation list sizes.

3.7 Results

Figure 2 shows the performance of typical recommendation technique with fixed size recommendation list for consecutive experiments which were conducted by using different recommendation list sizes (N) varying from 5 to 15.

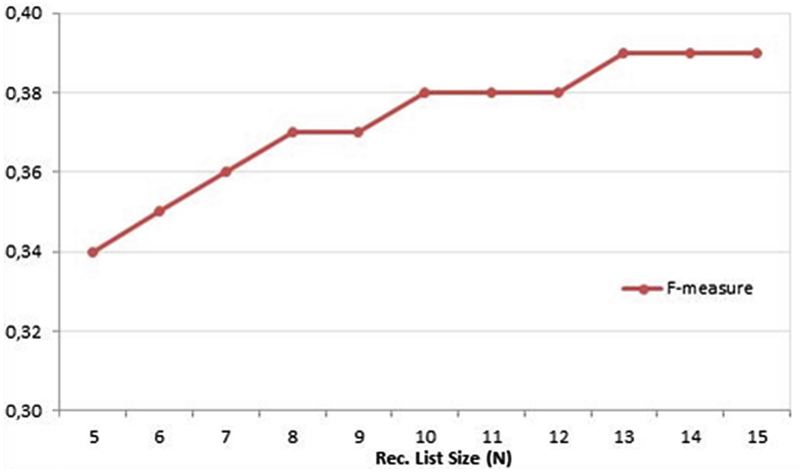

Fig. 2. F measure w.r.t recommendation list size

As shown in Fig. 2, F-measure increases slightly at the beginning, but stay constant as the recommendation list size increases. The experimental results of our proposed approach with different ML methods and benchmark methods are listed in Table 2. In the table, there are three different methods that adjust recommendation list size using different machine learning techniques and a benchmark method that adjusts the recommendation list size by using last visit's item count. In addition to these, only two relevant methods using fixed recommendation list size were added to the table for the purpose of comparison.

Table 2. Comparison of methods for recommendation list size

Method	Avg Rec. List Size	F-Measure
Adjusted with J48	8.96	0.4
Adjusted with KNN	8.94	0.39
Adjusted with Multi. Reg.	11.98	0.44
Last Visit N	11.18	0.4
Fixed N	9	0.37
Fixed N	12	0.38

In Table 2, "Avg Rec. List Size" column indicates the average number of recommended items to per customer. The results listed in Table 2 reveals the following findings:

- Our approach with J48 classifier achieves 8 % improvement in terms of F-measure, compared to the standard method using fixed recommendation list size (N = 9). Similarly, our approach with KNN classifier also improves F-measure by 5 %, compared to related standard method with the same size of recommendation list.
- Our approach with multiple regression achieves 16 % improvement in terms of F-measure, compared to the standard method with the same size of recommendation list (N = 12).
- Our approach with J48 classifier achieves same F-measure as the method of Last Visit N by shortening the average recommendation list size by nearly 20 % (i.e., with average recommendation list size of 8.96, down from 11.18 of Last Visit N approach).
- When we compare our approach based on KNN classifier with the method of Last Visit N, our approach provides 20 % reduction in the recommendation list size with only a small amount of accuracy loss (2.5 %)
- By increasing recommendation list size by 7 % (from 11.18 of Last Visit N approach to 11.98), our approach based on multiple regression achieves 10 % improvement in the recommendation performance, compared to the method of Last Visit N.

The above findings indicate that for the same size of recommendation list, our approach provides a better quality of recommendation than the standard one does. It also outperforms the benchmark, Last Visit N method in efficiency by providing reduction in the recommendation list size while preserving the accuracy.

4 Conclusion

This research proposes an approach to dynamically adjust the recommendation list size of a user with multiple preferences in order to improve the recommendation quality. By taking advantage of users' previous preferences and machine learning techniques, the proposed approach adjusts the number of items that will be preferred by the user according to his changing conditions. We evaluated our approach by conducting extensive experiments on a real-life dataset in the grocery shopping domain. According to the experimental results, our approach with all three selected machine learning techniques outperforms the traditional and widely used standard approach in effectiveness and it also provides better performance than the benchmark, Last Visit N method in efficiency by shortening the recommendation list while maintaining the effectiveness.

The proposed approach has the advantages of recommending a reasonable number of items without overwhelming the users and increasing recommendation quality. In this manner, recommender systems can utilize our approach to determine the optimal number of items to be recommended for especially the cases where the user has multiple preferences at a time. Therefore, such an efficient recommender system creates a win-win opportunity for both companies and customers by raising both customer satisfaction and company's revenue.

In this study, we presented the effectiveness of our approach for grocery shopping and the features related to this domain were identified as input for the proposed approach. However, our approach can potentially be applied to other domains as well, and the selected feature set may be different depending on the application domain and the chosen data set. Further, we used a set of machine learning methods in this study and other different machine learning techniques can also be employed in the proposed approach. In this context, a possible direction in the future work might be to apply the proposed approach in other application domains and using other machine learning techniques. In addition to this, in order to measure the user satisfaction, evaluating the proposed approach by conducting a live user study would be another future research issue.

Acknowledgments. This work is partially supported by the Scientific and Technological Research Council of Turkey (TUBITAK).

References

1. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. (TOIS) **22**, 5–53 (2004)
2. Ricci, F., Rokach, L., Shapira, B.: Introduction to Recommender Systems Handbook. Springer, New York (2011)
3. Schafer, J.B., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 158–166. ACM (1999)
4. Pu, P., Faltings, B., Chen, L., Zhang, J., Viappiani, P.: Usability guidelines for product recommenders based on example critiquing research. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 511–545. Springer, New York (2011)
5. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce, pp. 158–167. ACM (2000)
6. Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. J. Mach. Learn. Res. **10**, 2935–2962 (2009)
7. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Machine Learning: Proceedings of the Twelfth International Conference, pp. 194–202 (1995)
8. Catlett, J.: On changing continuous attributes into ordered discrete attributes. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS(LNAI), vol. 482, pp. 164–178. Springer, Heidelberg (1991)
9. Kotsiantis, S., Kanellopoulos, D.: Discretization techniques: a recent survey. GESTS Int. Trans. Comput. Sci. Eng. **32**, 47–58 (2006)
10. Quinlan, J.R.: C4.5: Programs for Machine Learning. Elsevier, Amsterdam (2014)
11. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**, 1–37 (2008)
12. Rucker, D.D., McShane, B.B., Preacher, K.J.: A researcher's guide to regression, discretization, and median splits of continuous variables. J. Consum. Psychol. **25**, 666–678 (2015)
13. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer, New York (2011)

14. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model. User-Adapt. Interact. **24**, 67–119 (2014)
15. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 39–46. ACM (2010)
16. Yang, Y.M., Liu, X.: A re-examination of text categorization methods. In: Proceedings of 22nd International Conference on Research and Development in Information Retrieval, SIGIR 1999, pp. 42–49 (1999)
17. Cumby, C., Fano, A., Ghani, R., Krema, M.: Predicting customer shopping lists from point-of-sale purchase data. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 402–409. ACM (2004)
18. Cumby, C., Fano, A., Ghani, R., Krema, M.: Building intelligent shopping assistants using individual consumer models. In: Proceedings of the 10th International Conference on Intelligent User Interfaces, pp. 323–325. ACM (2005)