Structured Learning Applied to Consumer Goods Recommended

Yu-Ching Chen   Chia-Ching Yang   Yu-Han Chen
Sébastien Montella   Chia-Hui Chang
Department of Computer Science and Information Engineering
National Central University Taoyuan, Taiwan
julie25599@gmail.com   uiuityty2003@gmail.com
Jmail840619@gmail.com   sebastien.montella@utbm.fr
chia@csie.ncu.edu.tw

Pin-Liang Chen   Ping-Che Yang   Tsun Ku
Institute for Information Industry
Taipei, Taiwan
mileschen@iii.org.tw   maciaclark@iii.org.tw
cujing@iii.org.tw

Abstract—Nowadays, E-commerce websites and recommendation systems are so common in our lives. From collected data in E-commerce websites, we found there are some repeated products in customers' buying history in different period. These consuming product will be purchased once again after customers run out of these products. Therefore, our recommendation system will focus on consuming product.

The approach we used differs from previous recommendation systems by taking into consideration time series and repeated purchases. We think there is a strong relationship between time and consuming products according to users' behavior of purchasing repeatedly. On the other hand, we also try to use structured learning, which can add constraint on different users to get products which customers much more prefer. That is, it exists ranking concept in recommendation system, and we try to use structured learning to solve it. In this paper, we also design user features, item features, user-item features, and time-related features. Data collected is used with state-of-the-art Machine Learning algorithms. We establish a prediction model to test data split by time. The experiments shows that the top 5 F-measures are 66.38% higher than the previous study [3] done last year, and we thus providing an effective recommendation system.

Keywords—Machine Learning, Structured Learning, Consuming Product Recommendation

1. INTRODUCTION

In recent years, E-commerce and social networking websites have recently been in vogue. More and more users have changed their shopping patterns from browsing traditional catalogs and purchasing physical stores to going on-line shopping. E-commerce shortens transaction time and cost. There are many kinds of products online, but for users, what they are really concerned is to choose rapidly and purchase their favorite products. For company, the primary goal of business operators is to be able to deliver products that customers need.

On the other hand, with the rapid development of mobiles, community websites and social media has become a platform for users, such as Facebook and Line. These means offer free services, so that the users can rely on their platforms which record information about the users, and thus enhance the opportunities for customized marketing. Take Netflix as an example. About 80% of the purchases are done thanks to sophisticated film recommendation. For Google and Facebook, more than 85% of revenues are from advertisement. That is the reason why so many researchers are into recommendation systems as a topic of research.

The recommendation system relies on large data, based on the user's past purchasing behaviors to predict the user's preferences. This paper mainly analyzes consumer products, such as facial cleanser, pads, hand warmer, and so on. At regular intervals, the customers flow back or generate demands during specific season. This kind of consumption pattern is different from a film recommendation system. The same movie will not recommended twice if a user already watched it. The main differences are time series and repeated purchases. Therefore these parameters need to be taken into consideration for consumer product recommendation.

We use structured learning which is a hot topic in machine learning that can be adopted for different forms of task such as image recognition, sequence marking and other issues. We use structured learning algorithm to calculate the scores of each item for each user. Scores of purchased products should be higher than scores of non-purchased ones. If this constraint cannot be satisfied, the weight *w* is updated. However for some users, the number of non-purchased products scores (*N*) may be greater than the purchased product scores (*P*), so we defined a cost value as N_{max} - P_{min} that by summing all cost for all users make our objective function. We minimize this objective function, and find *w* to predict the score of the user to buy a product.

Due to the consideration of time factors, we divide the training data and testing data by time, and design time-related features. We use the past three months of consumption information as features, and then take the purchasing history in the next two months as labels. The best performance is with a F-measure up to 0.4413.

2. RELATED WORK

The techniques used for recommendation system can be roughly divided into three categories, content-based filtering [6], collaborative filtering [11] and hybrid recommendation system which combined content-based filtering and collaborative filtering. Content-based filtering focuses more on the merchandise information, and the system recommends the similar products of previous products bought by customers. The leverage of such approach is that it can be used to solve cold start problem, but the results might be too similar and not customized. The collaborative recommendation system is based on the products purchased by other similar users. It can be divided into Memory-based CF [4] and Model-based CF [2]. In technology of Memory-based CF, using k-Nearest Neighbor [1] as the mechanism of Filtering doesn't need to prepare the training model in advance.

On the contrary, Model-based CF is the use of data mining algorithm to establish a training model. It needs to build a training model for historical data in advance, and predict the results based on the model. Common methods include classification [2], factorization [10] and so on. In the paper, we discuss the classification problem by SVM [9], SVMRank [12] and structured learning to predict whether user will buy an item or not.

3. DATA, FEATURE DESIGN AND METHOD

The data in this study is from a task-oriented App. The main purpose of the App is to encourage users to make consumption at the physical store. There are three types of tasks provided: check-in, scan and consumption. User can get points by completing the tasks, and get free products by using the points in physical stores. Because of the binding between App and Facebook login, we can get personal user information, including age, gender, residence and the fan pages they like. In addition, the information about the shop, the content of tasks, consumption period and the amount of consumption are also recorded in the App system.

The main challenge in our paper is the prediction of products bought periodically. Stores we analyze are department stores which sell make-up, accessories, food, hardware, and other products. Most of them are consuming product. Consuming product will be purchased repeatedly after the user uses it. The consuming recommendation is very different from the previous recommendation system. For instance, most readers will not purchase the same book, but they will repeatedly buy toilet paper. Because of this, consuming recommendation is a new challenge for us.

The other key factor is time. When the seasons go on, the sales of goods will follow the change, such as warm bags are mainly sold in winter rather than summer. On the contrary, sanitary napkins will be purchased by customers after a fixed period of time.

Unlike general commodity recommendations, this question is set to recommend the product category to the user due to higher autonomy of the company. Company can choose high-profit products or the product with closer terms in consideration of its inventory status to do recommendation. It is easier for company to control their commodities. This approach of recommending the product category gives company more flexibilities.

3.1. User features

The app acquires user information, including users' Facebook ID, gender, age, residence, and the category of Facebook Pages the user likes. We designed the features about users' personal information. Age is divided into three categories: under 20 years old, 20 to 40 years old and over 40 years old. Gender: male or female. Residence: north, south, east, center, off-island area and traveler.

There are 18 groups which are categorized by Facebook fan pages the user likes, including female or male students, professional men, and so on. Each user belongs to a group. In addition, we also divide fan pages into five categories, those are food, clothing, transportation, education and entertainment. According to those fan page clicked by users, we would find the each user's preference for different categories.

We also designed the features of users' purchasing habits, which are about tasks users carried out before (check-in, scanning and consumption), spending amount is more or less than mean, commodity categories (type of food or drinks, type of clothing or use and type of entertainment or education), transaction time (morning, noon, night, no business hour, weekday, holiday, festival, non-festival). Each feature is represented in binary, and there are 50 user features in total.

3.2. Item features

The store is especially popular in rural areas that lack department stores, as it sells makeup, accessories, food, hardware, and other products. There are 2,177 products in total, which are classified into 28 primary categories, 144 secondary categories, and 375 minor categories. We use a web crawler to acquire product information and use keyword extraction and word2vec [8] to represent item features.

In keyword extraction, we use Google to search the names of the secondary and minor categories, fetching the top-10 titles on the page. After using the word segmentation with the package *jieba*, we choose nouns (N), verbs (V), place names (NS), and gerunds (VN) to extract unneeded data by the two methods of keyword extraction: term frequency-inverse document frequency (TFIDF[5]) and TextRank[7]. Then we extract each top five keywords for item features.

In addition, we take top-10 Google snippets and Yahoo snippets as our corpus. We preprocess the corpus as follow steps. First, the dates, quantifiers and prices are replaced. For example, we "*10 g*" or "*2 pairs*" with the token "*_QUANTITY_*". The segmentation is done by *jieba*. Next, we make use of the data we already prepared to train a word embedding model which is set to 300 dimensions and a window size 7. Thus, our item feature will be this 300-length vector.

3.3. User- item pair features

We are concerned not only about the user and item features but also the interaction between customers and commodities. We take the probability of each commodity bought by users as user-item pair feature. However, there are some users who have less purchasing history and leading almost probability to be zero. Therefore we cluster all users by user features into different groups, and calculate the probability of each commodity bought by different user groups. The probability will be the user-item pair features.

Because the periodic products will be bought repeatedly, the features should have relation with time, which contributes to the generation of time-related features. We prepare 8 time-related features. Five of them are binary, including whether the user purchased this item before or not, whether the user bought the item more than once, whether the user bought the item more than the average period (user-item average period, user average period and item average period). Others are numerical, including the total number of times that a user bought the product, the number of times that a user bought the product in a period. Since there is a gap between the numbers of purchasing time for different users, we normalize data by using the log function. The last feature is the density of product bought by user. In another hand, that is the ratio of the average number of purchased product in a certain period and a long-term period.

3.4. Structured learning

Structured learning provides us with a framework to solve machine learning problems. It is no longer limited to only output a numerical number or label, but can accept any structured data as input or output, e.g. sequence, image or bounding box. Overall, we can use structured learning more directly to deal with the problems. That is, we can regard any kinds of output as a certain structure. In structured learning, it involves two parts, training and inference. In training part, there is a function F: X×Y→R. We want to find F(x, y) depending on that function so that we can use F(x, y) to know how close the object X and Y are. In inference part, we exhaust all possible answer y for one object x to find a best answer through this function $\tilde{y} = \underset{y \in \hat{y}}{\operatorname{argmax}} F(x, y)$.

In our structured learning design, we hope to find the function F(*x*, *y*) = *w* · Ø(*x*, *y*). *x* represents users, *y* represents items, Ø(*x*, *y*) represents user and item's feature vector, and *w* is the weight trained from structured learning. For each user, we will count out the value P and value N in terms of function F(x,y). The value P is count by putting all purchased items ($\boldsymbol{y} \in \hat{\boldsymbol{y}}$) in function F(x,y). The value N is count by putting all non-purchased items ($\boldsymbol{y} \notin \hat{\boldsymbol{y}}$) in function F(x,y). And we hope value P can be higher than N for each user. Because a user may buy more than one product, we want P_{min} to fulfill $P_{min} \geq N_{max}$. However, N_{max} still has a chance to be greater than P_{min}, so we define a cost $C_i = N_{max} - P_{min}$, which represents how wrong our model is. For this condition, even if we have this error, we still want the gap between N_{max} and P_{min} to be reduced. Finally, we sum up the C_i's from all users, and minimize the objective function $\sum_{i=1}^{N} C_i$. Since the distribution of training data and testing data might be different, if weight is close to 0, we can reduce the influence of mismatch. Therefore we will add a regulating factor to the objective function, and optimize it by the gradient descent. The objective function is then as follows.

$$\mathrm{Cost} = \frac{1}{2}||W||^2 + \sum_{i=1}^{N} C_i = \frac{1}{2}||W||^2 + \sum_{x \in X} w \cdot \emptyset(\mathbf{x}, \mathbf{y_n}) - w \cdot \emptyset(\mathbf{x}, \mathbf{y_p})$$

$$\nabla \mathrm{Cost} = W + \sum_{x \in X} w \cdot \emptyset(\mathbf{x}, \mathbf{y_n}) - w \cdot \emptyset(\mathbf{x}, \mathbf{y_p})$$

In our algorithm, we count the minimal value of $P_{min} = \min_{y \in \hat{y}} F(x,y)$ in each purchased product ($\boldsymbol{y} \in \hat{\boldsymbol{y}}$) for each user and the corresponding product $y_p = \operatorname{argmin}_{y \in \hat{y}} F(x,y)$. In the same way, we also count the maximum value of $P_{max} = \max_{y \notin \hat{y}} F(x,y)$ in each not purchased product ($\boldsymbol{y} \notin \hat{\boldsymbol{y}}$) for each user and the corresponding product $y_p = \operatorname{argmin}_{y \notin \hat{y}} F(x,y)$. If $P_{min} < N_{max}$, we will update the weight. If the data is separable, that is all values for purchased items are more than non-purchased items after training. One epoch is equivalent of going through all the users at a time.

Algorithm 1. Structure Learning Algorithm

Initialize w vector is random number from -1 to 1
For each user (x, ŷ)
$N_{max} = \max_{y \notin \hat{y}} F(x, y)$
$P_{min} = \min_{y \in \hat{y}} F(x, y)$
if $P_{min} < N_{max}$
$y_n = \operatorname{argmax}_{y \notin \hat{y}} F(x, y)$
$y_p = \operatorname{argmin}_{y \in \hat{y}} F(x, y)$
$w = (1 - \eta) \cdot w - \eta[\emptyset(x, y_n) - \emptyset(x, y_p)]$
End While;

4. EXPERIMENT

Data used is from January 2015 to June 2016 in a point-oriented App. There are 259,550 transactions in this time period. We chose 1,715 people who consumed over 20 times to reduce the influence of sparse data. There are 144 secondary product categories. We chose to recommend product category to users for the company to have autonomy and to decide how the recommendation will be done, i.e. the company can choose to recommend high profit product or some products which are drugs on the market to user.

4.1. Data Preparation

Figure 1. Data Preparation about time series

We consider 5 months at a time. The data on first 3 months are features and used for training, and the purchased situation on the 2 last months are labels, as shown in Figure 1. By splitting data in such configuration, it can solve the problem of insufficient information. In the experiment, we selected information from January 2015 to April 2016 as a training data. According to Figure 1, our training datasets will become 13 times as many than original datasets. The data on May 2016 and June 2016 are used as test data.

4.2. Evaluation

We use F-measure to test our performance. In recommendation system, not only precision but recall is crucial. The formulas are as follows:

$$\text{Precision(hit rate)} = \frac{c(r \cap b)}{c(r)} \qquad \text{Recall} = \frac{c(r \cap b)}{c(b)} \qquad F = \frac{2RP}{(R+P)}$$

Where *r* represents the products we recommend to the user, and *b* represents the products that user bought. In general, the recommendation systems will not deliver too much products to the user, because once it does, the customer might not afford all of it. Therefore, we will observe the effectiveness of top1, top3, top5, top10, and top15.

4.3. Experimental Result

Because all features are over 2,000 dimensions, we want to know which features have the influence in our training part.

After comparing performance of each feature in structured learning, we show the result in Table 1. If the value is greater than the performance of all features (2,453 dims), then it shows in red. We found out that the performance is better when one type of user features is not included, which means that this type of features have few influence for the prediction. On the contrary, for remaining features, their performance is almost the same with all features' performance.

Table 1. F-measure of each feature

Top-k	F-measure	Increase Rate					
	all feature (2,453 dims)	without Personal Information (-11 dims)	without Facebook (-23 dims)	without Task Execution (-16 dims)	without Keyword (-2,045 dims)	without Word2vector (-300 dims)	without user-item pair feature (-58 dims)
1	0.0062	1.53	0.34	0.92	0.03	-0.10	-0.10
3	0.0250	0.10	0.34	0.00	-0.30	-0.10	0.14
5	0.0379	0.05	-0.05	0.14	0.09	0.03	0.02
10	0.0463	0.21	0.07	0.04	-0.05	0.02	0.02
15	0.0526	0.12	0.16	0.00	-0.06	-0.05	-0.07

Table 1 shows that the F-measures are close to each other. All the execution times are high due to numerous features. An epoch cost more than 3 hours. Therefore, we decided to use 58 dimensions for our study. The reason is we can find there is nearly no difference on those features performance from Table 1, and 58 dimensions features can represent the interaction between users and items. We can avoid time-consuming operation time by using these low dimension features.

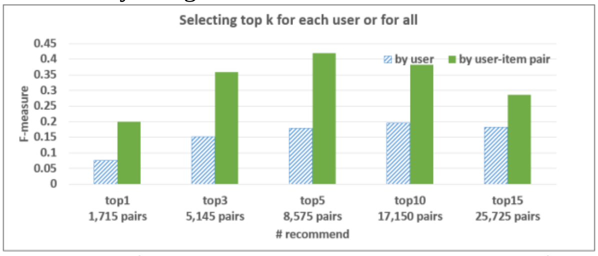

Figure 2. Performance comparison on the same numbers of recommending products for ranking on user or user-item pair

In addition, we compare the performance of recommending the same numbers of product or the top k to each user. In another words, we chose every user's top 1 to count its F-measure. Because there are 1,715 users, after the sorting by user-item pair, we count the F-measure of top 1,715 pairs. According to figure 2, we found that the F-measure of user-item pair ranking (green bar) is much more beyond then the user ranking (gray bar). The reason is that some users have few consumption, and we fail when he rejects to buy the product. On another hand, for the user that often consumes, we can recommend more products to him and make him consumes more, so that our recommendation system could be more successful.

Finally, we compare the effectiveness of different methods, as shown in Fig. 3. We compare five methods, including SVMRank, structured learning, structured learning (QP), CCAM, and heuristic scoring. We found that SVMRank comes better in top-1, and F-measure of structured learning is close to it. When it comes to top-3 and top-5, structured learning is much better than all other methods. In top5, F-measure achieves an accuracy of 0.4201. Structured learning (QP) and heuristic scoring are better in top-10 and top-15.

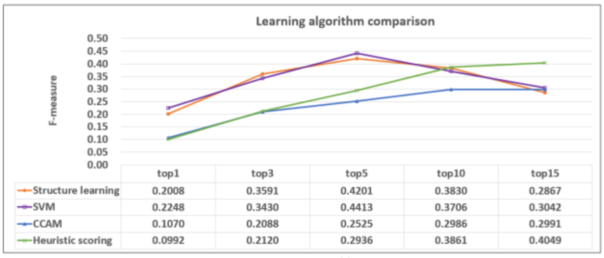

Figure 3. Comparison on different methods

5. CONCLUSION

Generally speaking, consuming products will be bought once again after customers use them. Therefore, we need to consider the repeatability and the time series of the product. In general, the movie recommendation system has been used to predict which movie the customer will like. However, most people will not watch the same movie once again. Based on our special case, we redesign our feature structure and change our way to collect data.

In this paper, we designed time-related features and time series data preparation method, which is that we use the first three months of the purchase status as features and the information in next two months as label. Then we used top-K of the recommendation data to do the evaluation. Structured learning and SVM had the best performance but structured learning had a shorter programming time and can achieve the timeliness.

ACKNOWLEDGMENT

This study is conducted under the "System-of-systems driven emerging service business development project(2/4)" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

REFERENCES

[1] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185.

[2] Billsus, D., & Pazzani, M. J. (1998, July). Learning Collaborative Information Filters. In Icml (Vol. 98, pp. 46-54).

[3] Chen, Yu-Ching, et al. "User behavior analysis and commodity recommendation for point-earning apps." Technologies and Applications of Artificial Intelligence (TAAI), 2016 Conference on. IEEE, 2016.

[4] Delgado, J., & Ishii, N. (1999). Memory-based weighted majority prediction. In SIGIR Workshop Recomm. Syst. Citeseer.

[5] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). Mining of massive datasets. Cambridge university press.

[6] Melville, P., Mooney, R. J., & Nagarajan, R. (2002, July). Content-boosted collaborative filtering for improved recommendations. In Aaai/iaai (pp. 187-192).

[7] Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing Order into Text. In EMNLP (Vol. 4, pp. 404-411).

[8] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[9] Murty, M. N., & Raghava, R. (2016). Support Vector Machines and Perceptrons: Learning, Optimization, Classification, and Application to Social Networks. Springer.

[10] Rendle, S. (2012). Factorization machines with libfm. ACM Transactions on Intelligent Systems and Technology (TIST), 3(3), 57

[11] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web (pp. 291-324). Springer Berlin Heidelberg.

[12] Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. Journal of machine learning research, 6(Sep), 1453-1484.