A Brick-and-mortar Store Recommendation System based on Online Shopping Behavior

Shihao Wang
College of Computer Science and Information
Northeastern University
`UID: 001850916`
Instructor: Prof. Rose Yu

Abstract

With mobile devices becoming more and more popular, location based data is shared widely and lots of researches focus on recommendation systems based on such data. As one of typical recommendation techniques, cross-domain recommendation stands out by taking the advantage of the knowledge from other domains to improve the recommendation performance. In this work, we found that the online shopping behavior data can effectively help the brick-and-mortar stores' recommendation. We build up such correlation using user cluster results as the intermediary because user clustering can reflect user's purchase preference on both online and on-site platforms. Finally, we combine multiple strategies to do this recommendation task and this system achieves a promising F1 score.

1 Introduction

As mobile devices become ubiquitous in our daily life, people are getting more comfortable sharing their real-time locations with various location-based services, such as navigation, car ride hailing, restaurant/hotel booking, etc. On the other hand, huge amount of user data has been accumulated in online services like online shopping. We may explore the value of users' behavior to help the public find their demand in location based services using supervised and unsupervised integration techniques.

Concretely, we want to recommend the potential brick-and-mortar merchants that users will visit in the future, based on the user behavior records on both online website and on-site merchants. For this recommendation task, the input consists of two parts given a certain user: 1. the behavior records(click, buy, etc.) of a user from online websites. 2. the user's location. And the recommendation system outputs an ordered merchant list including the nearby stores the user may like.

Generally speaking, a large online retail platform serves far more people than the on-site merchants. The online platform will accumulate lots of valuable user data and utilize it to improve user's experience, such as personalized recommendation and search. However, it is difficult for brick-and-mortar stores. First, the number of people visiting a store is limited compared to an online platform. Second, it's relatively hard for them to collect user behavior data. Therefore, it is very meaningful to explore users' online shopping behavior for providing them more accurate location-based store recommendation, especially when users enter new areas they rarely visited in the past.

The difficulty of this work comes from mainly two aspects. First, mining latent correlations between merchants and online sellers is essential but difficult. We cannot directly measure the distance between merchants and online sellers because they don't lie in the same feature space. Such correlations are hard to define because they are under control by lots of latent variables. Second, Extracting effective features for measuring the distance or similarity of different users is also important but tricky. Because the meta data doesn't give us many useful features directly. We need to construct them by analyzing multiple tables jointly. Considering the huge amount of entities (user, sellers and merchants) in our system, insufficient features quantity will cause those entities indivisible in the low-dimensional space. So we need to promise both the quantity and quality of our feature engineering.

The followings are the contributions of this work: 1. For extracting promising features, we creatively use latent factorization model to generate latent features and combine them with the explicit features. Latent factorization model has been used in recommendation task heavily. The most famous one should be the solution of [1] at the Netflix Competition. Here, we use it as a feature engineering method. 2. To uncover the correlation between on-site merchants and online sellers, we conduct clustering first for users, merchants, and sellers separately. Then we quantify the correlation based on the clustering result. For human's intuition, learning two sellers' correlation from their belonging category is a better way to explain.

2 Related Work

Point-of-interest (POI) recommendation system based on Location Based Data (LBD) starts to be a popular research topic as mobile devices are used ubiquitously. Many of them are mainly based on Location Based Social Network (LBSN), where geographical data [2] and collaborative filtering play important roles when mining the user's potential interest. However, only using geographical data cannot fully draw the user profile. To incorporate user preference into the recommendation, Li et al [3] build a review-based recommendation system, which explores the utility of ratings/reviews in LBSN. In our work, we use user's purchase habits of online and brick-and-mortar shopping to reveal the correlations between online sellers and on-site merchants.

Specially, using online shopping behavior data to help brick-and-mortar store recommendation can be regraded as a cross-domain recommendation application. Cross-domain recommendation is an emerging research topic, which incorporates relevant data sources from different domains and combines with the original target data to improve the recommendation [4]. Kuflik et al. [5] first propose cross domain mediation problem and introduce several techniques for importing relevant data. Pan et al. [6] propose the model to transform knowledge from domains which have heterogeneous forms of user feedback. Cremonesi et al. [7] considere to model the classical similarity relationships as a direct graph and explore all possible paths connecting users or items in order to find new cross-domain relationships. Tang et al. [8] propose the cross-domain topic learning model to predict the cross-domain collaborations through topic layers instead of at author layers, which alleviated the sparseness issue. Hu et al. [9] also apply topic models (LDA) to denote user features. They regard user as document and item as word in LDA to analyze user's shopping preference by topic models. Winoto et al. [10] propose to uncover the association between user preferences on related items across domains to provide the cross-domain recommendation. However, the majority of the existing work assumes that the source and target domains are related but do not suggest methods to quantify the correlations among domains, which has been addressed in our work. There is another problem for transferring knowledge across domains. That is, it cannot guarantee that the knowledge of other domains is useful for the target domain. And it is pretty hard to directly evaluate the quality of such across-domain correlations. To tackle with this problem, we apply multiple recommendation strategies to promise a low-bias results.

3 Method

This section will first introduce the pipeline of our system and then provide detailed review of two key points in feature engineering and correlation computing.

3.1 Framework

According to the Fig.1, our modeling process has five basic steps.

Feature engineering After we obtained the meta data, which are three tables from both Taobao and Koubei dataset, we extract both handcrafted and latent features for users/merchants/sellers (for convenience, we denote online stores as sellers and brick-and-mortar stores as merchants). Table 1, 2, and 3 show the handcrafted features and their meanings. For generating latent features, it will be introduced in the following.

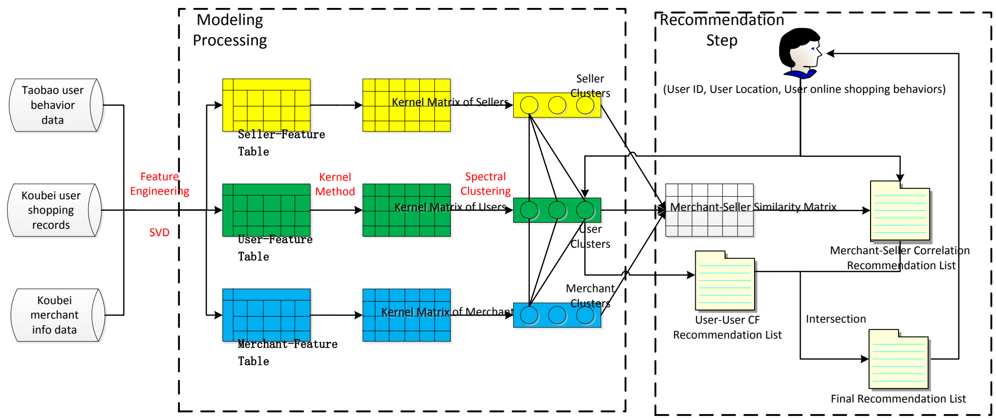


Figure 1: The framework of our system: modeling processing and reco mmending pipeline

Feature	Description
popularity	#(merchant m, location l)/ #(location l)
merchant_loc_records	the number of purchase records of this merchant
openning_date	the day's number from the first record to 20150701
last_purchase_date	the day's number from the last record to 20150701
average_purchase_interval	the average time period of being purchased
merchant_budget	the coupon constraint of this merchant
merchant_rank	the rank of this merchant based on purchase amount
purchase_user_number	the number of distinct consumers this merchant has
buying_days	the day's number with any purchase records
merchant_svd_feature	the latent vector for merchant obtained from matrix factorization

Table 1: Merchant-related features

Similarity matrix transformation We will apply the kernel method to transfer feature matrices into affinity matrices, where $k_{user}(i, j)$ can be regarded as the similarity between user i and j. We have attempted Gaussian(rbf), Polynomial and cosine similarity kernel for that and the performance of different kernels can be seen in the evaluation section.

Spectral Clustering Spectral Clustering [11] gives us the cluster results of users, merchants and sellers separately. We use Spectral Clustering mainly because it can conduct clustering on the non-convex regions and have a better performance on sparse data than k-means. One thing should be noticed is the selection of the clustering number k. Here we use both index evaluation and elbow method to select best k number.

Correlation computing After we obtained the clustering results, we use the user's clusters as the intermediary to build up the correlation between merchant and seller. We use the purchase frequency tables of user-merchant as well as user-seller and adopt soft alignment method to the do this work. Details will be elaborated in the section Correlation Computation.

Recommendation We combine two strategies to generate the final recommend list. First, we use the correlation table generated before to recommend the most 'similar' or 'closed' merchant with those sellers that the given user has been scanned. Second, we also apply the User-User Collaborative Filtering to provide another candidate recommend list (Here we use cosine similarity to measure the affinity between two users). Finally, we pick up the intersection of both recommend lists and deliver the final results to the user.

Feature	Description
seller_id	unique seller identity from 1 to 10000
seller_records	the number of records of this seller
buy_count	the number of buy records
click_count	the number of click records
successful_trade_ratio	(buy_count)/(buy_count+click_count)
item_number	total items' number that this seller has
category_number	total item' category that this seller has
average_purchase_interval	the average time period of being clicked or bought
maximum_purchase_per_day	the maximum number of being purchased in one day
maximum_click_per_day	the maximum number of being clicked in one day
seller_svd_feature	the latent vector for seller obtained from matrix factorization

Table 2: Seller-related features

Feature	Description
taobao_records_count	the number of the user's records on Taobao.
taobao_buy_ratio	the ratio between the number of the user bought and clicked on Taobao.
taobao_buy_count	the number of records that the user bought on Taobao.
items_buy_count	the number of different items the user bought on Taobao.
items_click_count	the number of different items the user viewed on Taobao.
items_buy_ratio	the ratio between the different items of the user bought and clicked on Taobao.
categories_buy_count	the number of different categories the user bought on Taobao.
koubei_records_count	the number of the user's Koubei records.
shopped_locations_count	the number of locations the user shopped on Koubei.
shopped_merchants_count	the number of merchants the user shopped on Koubei.

Table 3: User-related features

3.2 Feature Engineering

The generated features play an important role in modeling. The essential task in the recommendation problem is to design effective features. We extract features for users, sellers and merchants to construct the vectors representing them. Our features come from two parts, the handcrafted explicit features and latent features generated by SVD factorization.

The handcrafted/explicit features have clear meanings and can be easily interpreted. We designed a set of features for users, sellers and merchants shown in the Table 1, 2 and 3.

The latent features are generated by SVD factorization of the purchase history records on Taobao and Koubei. We construct a User-by-Seller co-occurrence matrix and a User-by-Merchant co-occurrence matrix, then decompose them and take the first 100 dimensions as the latent feature vectors of users, sellers and merchants.

3.3 Correlation Computation

One of the key step we use online shopping data to recommend brick-and-mortars is computing the correlation between online sellers and offline merchants. We take 3 steps for the computation:

1. Measure the probability of a seller/user/merchant assigned to each seller/user/merchant cluster. Take user as an example, we measure it by the normalized inverse of the distance between a user and a user cluster centroid:

$$P(u \in CU_i) = \frac{\frac{1}{d(u,CU_i)}}{\sum_k \frac{1}{d(u,CU_k)}} \tag{1}$$

Here $P(u \in CU_i)$ is the probability of user u assigned to the user cluster CU_i, and $d(u, CU_i)$ is the distance between u and the centroid of user cluster CU_i. The probability of sellers and merchants could be computed similarly.

2. Compute the correlation between user clusters and seller/merchant clusters. We integrate all users and merchants for computing the correlation between user clusters and seller clusters. Note that here

we take the use of soft-alignment:

$$R(CU_i, CM_j) = \sum_{u,m} P(u \in CU_i)R(u,m)P(m \in CM_j) \tag{2}$$

Here $R(CU_i, CM_j)$ is the correlation between user cluster CU_i and merchant cluster CM_j, and $R(u,m)$ is the number of user u shopping in merchant m. We can compute the correlation between user clusters and seller clusters similarly.

3. Compute the correlation between seller clusters and merchant clusters. We can easily derive this if we adopt user clusters as the intermediary. It is nothing more than a matrix multiplication with a normalizing factor:

$$R(CS_i, CM_j) = \frac{\sum_k R(CU_k, CS_i)R(CU_k, CM_j)}{Z} \tag{3}$$

Here $R(CS_i, CM_j)$ is the correlation between seller cluster CS_i and merchant cluster CM_j, and Z is a normalizing factor. The objective function of our latent factor model is the following:

$$min_\Sigma(A_{s,m} - [U\Sigma V^T])^2 \tag{4}$$

where Σ is the cluster-by-cluster matrix between seller and merchant, A is the objective seller-by-merchant correlation matrix we want to learn. U an V represent two stochastic matrices for seller-by-cluster and merchant-by-cluater.

4 Evaluation

4.1 Dataset

Our datasets come from Taobao and Koubei as Table 4, 5, and 6. Taobao.com is the largest online retail platforms in China, serves for more than 10 million merchants and over 300 million customers. Meanwhile, Koubei offers restaurant and retail store recommendation and payment services for a huge number of customers. A user enjoying services provided by these two groups often has a unified online account. So these datasets are adequate and representative. The involved datasets are mainly two parts: 1. Taobao Online user behavior dataset, which provides 963923 users and 44528127 purchase records. 2. Koubei Brick-and-mortar users shopping records dataset, which provides 230496 users and 1081724 purchase records, and 5910 on-site merchants.

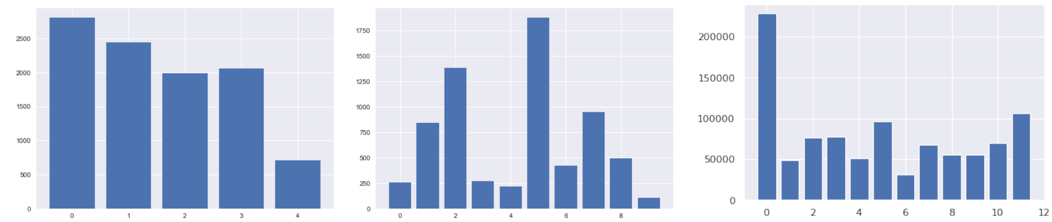
We held out about 20% of users in Koubei dataset as our test set. So finally we have a test set with 199129 entries. We use the left 922595 entries in Koubei dataset and the whole Taobao dataset for training our model.

Field	Description
User_id	unique user id
Seller_id	unique online seller id
Item_id	unique item id
Category_id	unique category id
Online_Action_id	"0" denotes "click" while "1" for buy
Time_Stamp	date of the format "yyyymmdd"

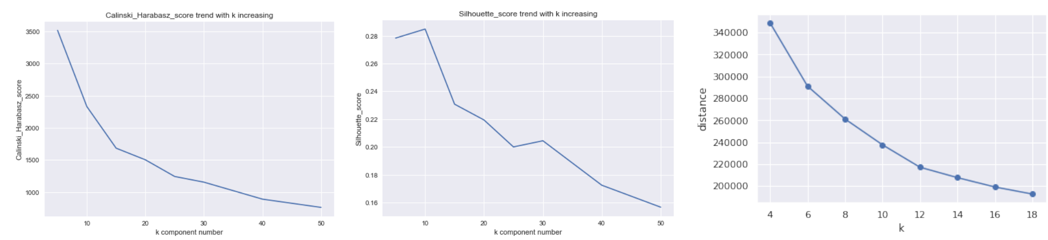
Table 4: Online user behavior data on Taobao.com

Field	Description
Merchant_id	unique merchant id
Location_id	unique location id
Time_Stamp	date of the format "yyyymmdd"

Table 5: Brick-and-mortar stores' shopping records on Koubei.com



(a) Data clustering distribution under best k number for seller (b) Data clustering distribution under best k number for merchant (c) Data clustering distribution under best k number for user



(d) Silhouette coefficient score's trend with k for seller (e) Silhouette coefficient score's trend with k for merchant (f) Total distance trend with k for user

Figure 2: k selection for seller, merchant and user

4.2 Experimental Setup

We use a Google Cloud Compute Engine with 24 vCPU and 260GB memory. It is mainly because there are some computations consuming huge amount of memory, such as the affinity matrix. Our evaluation mainly consists of two parts: clustering analysis and recommendation performance evaluation. There are total four experiments to be discussed in the following sections.

4.3 Kernel Method

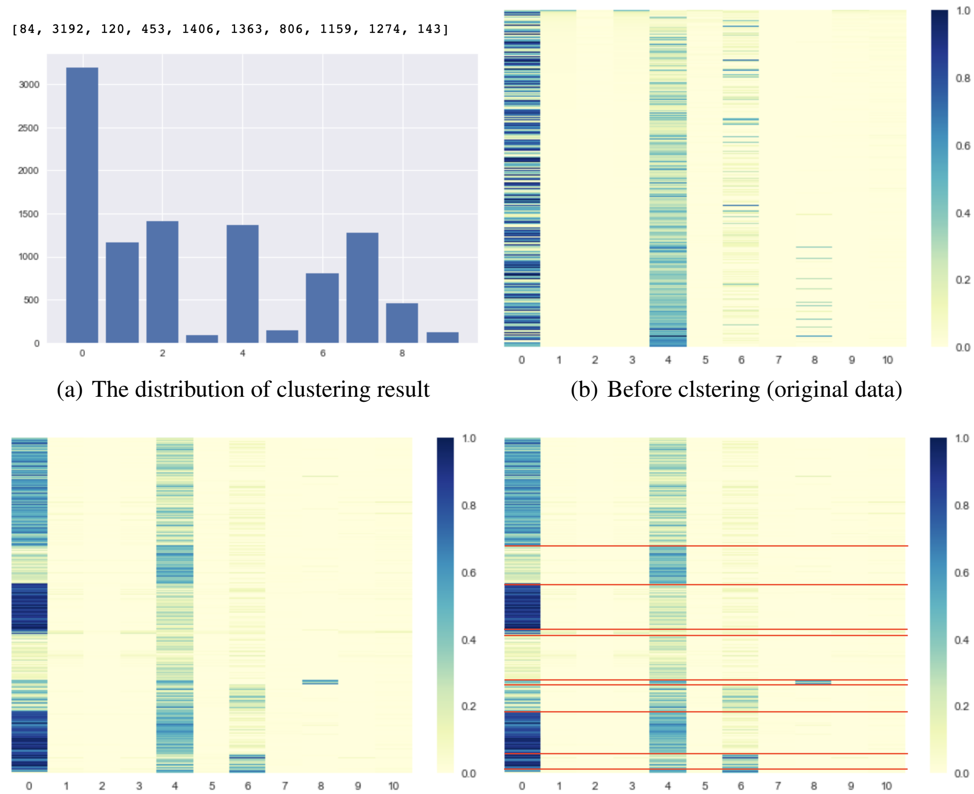
Metric	k	Calinski-Harabasz index	Silhouette coefficient
polynomial without k_nearest_neighbors	5	2986.682	**0.312**
cosine similarity without k_nearest_neighbors	4	1667.811	0.258
rbf without k_nearest_neighbors	5	3515.953	0.259
rbf with k_nearest_neighbors	5	**3768.589**	0.259

Table 6: Comparison of different kernels' best performance on seller clustering.

Metric	k	Calinski-Harabasz index	Silhouette coefficient
polynomial without k_nearest_neighbors	8	1921.955	0.285
cosine similarity without k_nearest_neighbors	5	1748.363	**0.330**
rbf without k_nearest_neighbors	10	**2106.844**	0.284
rbf with k_nearest_neighbors	10	1515.760	0.204

Table 7: Comparison of different kernels' best performance on merchant clustering.

We suppose that good clustering results promise a good recommendation behavior. So we attempt different kernels for spectral clustering and try to find the best setting for each clustering. Table 7 and Table 8 show the kernels comparison by Calinski-Harabasz index and Silhouette coefficient. We can see that rbf kernel performs well for both seller and merchant clustering. So we choose rbf without knn and rbf with knn as the final kernels for both clusterings.



(a) The distribution of clustering result

(b) Before clstering (original data)

(c) After clustering

(d) Heat map with clustering division lines

Figure 3: Visualization of seller's spectral clustering with k=10 and explicit features only

4.4 Clustering Number Selection

For the clustering number k selection of merchant and seller, we use Silhouette coefficient and observe the tendency. For the seller, the Silhouette coefficient curve is monotonically decreasing. This might be caused by the variances of sellers' features are not very big and concentrate on only few feature dimensions. However, for the merchant, we can see a maximum at k=10. For the user, we observe the total distance sum change and use the elbow method to select k=12 as the best clustering number.

4.5 Clustering Quality Analysis

To proof the clustering quality, we conduct a study case to show the power of spectral clustering in the Fig. 3. Fig.3(b) shows the original distribution of seller's feature vectors, which are pretty erratic. After the clustering, Fig.3(d) shows that spectral clustering successfully group together the data with similar feature vectors, that is, with the same color distribution. Meanwhile, it also reflects some key features that affect the clustering results. We can see that seller_id, successful_trade_ratio, category_number and last_purchase_date help build up the clusters most.

4.6 Recommendation

Table 9 shows the F1 score of all methods which are conducted on these datasets. The first, second and fifth appear in [12] while the third, forth and sixth are from [9]. The baseline would be recommend stores based on user's purchase history, whose F1 score is 0.25. We conduct an ablation experiment and draw that the result of user CF is very similar with the result of combining user and merchant statistics. We can see our best unsupervised result is acceptable. It beats the single supervised GBDT model. However, we notice that the results vary a little among all methods except the first and naivest approach. We have to be cautiously optimistic about the result. We doubt that all those fancy ideas means less than what we thought before.

Metric	F1 score
user's history [12]	0.25
user's + merchant's history [12]	0.42
User-User CF	0.424
Merchant-Seller correlation	0.433
Single GBDT [9]	0.447
Merchant-Seller correlation+User-User CF (our best)	**0.449**
GBDT+LDA&K-Means [9]	0.4513
Xgboost [12]	0.452
Ensemble with Isotonic Regression [9]	0.4638

Table 8: Comparison of final recommendation performance by F1 score.

5 Conclusion

In this work, we build up a brick-and-mortar recommendation system taking the advantage of huge amount of online shopping data. We use matrix factorization to extract latent features and adopt user as the intermediary to build up the correlations between merchant and sellers. User can connect merchant and seller because the same user shares similar shopping habits on both online and brick-and-mortar stores. Combining User-User Collaborative Filtering and merchant-seller correlations improves the recommendation results. Finally, we obtained a F1 score with 0.449 to beat some supervised model.

References

[1] Bell, Robert M., Yehuda Koren, and Chris Volinsky. "The bellkor 2008 solution to the netflix prize." Statistics Research Department at AT&T Research 1 (2008).

[2] Cheng C, Yang H, King I, et al. Fused matrix factorization with geographical and social influence in location-based social networks[C]// AAAI Conference on Artificial Intelligence. 2012.

[3] Li X, Xu G, Chen E, et al. Learning User Preferences across Multiple Aspects for Merchant Recommendation[C]// IEEE International Conference on Data Mining. IEEE, 2016:865-870.

[4] Ignacio Fernadez-Tobis, Iva Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems: A survey of the state of the art. In In Proceedings of the 2nd Spanish Conference on Information Retrieval, 2012.

[5] S. Berkovsky, T. Kuik, and F. Ricci. Cross-domain mediation in collaborative filtering. User Modeling 2007, pages 355-359, 2007.

[6] W. Pan, E. Xiang, N. Liu, and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. In Proceedings of the 24rd AAAI Conference on Artificial Intelligence, pages 425-434, 2010.

[7] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. Cross-domain recommender systems. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops, ICDMW'11, pages 496-503, 2011.

[8] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1285-1293, 2012.

[9] Hu, Ke, Xiangyang Li, and Chaotian Wu. "Cold Start Purchase Prediction with Budgets Constraints." SocInf@ IJCAI. 2016.

[10] Pinata Winoto and Tiany Ya Tang. If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. New Generation Comput., pages 209-225, 2008.

[11] Ng, Andrew Y., Michael I. Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." Advances in neural information processing systems. 2002.

[12]Wei, Hao, Bei Shi, and Junwen Chen. "Location Based Services Recommendation with Budget Constraints." (2016)