Group 7, Big Data Techniques and Technologies

Get to know your right customers

A development of customer acquisition prioritization system

Liu Yue, Zhu Yilin

4-20-2017

1 Introduction

1.1 Motivation

The economy of United States has been stagnant since the global financial crisis in 2008. In 2016, the GDP growth of United States reached the lowest since 2011 (Bureau of Economic Analysis, 2017). As individuals tend to cut down their personal expenditures, companies in various merchandising industries have experienced fierce competitions under the current market situation (Sheffer, 2016). The battle for market share drives companies to increase their customer base, and customer acquisition has become increasingly important. However, the intensified competition also restricted companies' budget in marketing campaigns. Companies need not only to acquire new customers, but to acquire them efficiently. Therefore, recognizing the "right" customers is of paramount importance.

1.2 Problem statement

Our study aims to propose a customer acquisition system to identify the more profitable potential customers for a company, based on multi-company customer transaction records. The system should allow the company to access the improve marketing efficiency.

1.3 Overview of solution

We use machine learning techniques to suggest potential customers for a company, by observing the purchase patterns of over 2 million customers and the revenue structure of the company. The algorithms employed include various collaborative filtering (CF) methods and popularity-based rules.

2 Literature review

2.1 Collaborative filtering recommender systems

Collaborative filtering (CF), as a popular recommendation algorithm commonly used in E-commerce, makes predictions or recommendations based on the ratings and opinions of other users in the system that share similar preferences. The implication behind CF is that users are assumed to agree on some item are likely to agree on other items (Ekstrand, et al., 2011). In the collaborating process, a predicted likeliness score is calculated for between an item and a given user. Traditionally, two commonly mentioned CFs in recommender designs are item-based CF and user-based CF. For a given active user for whom the CF algorithm is performed, user-based CF utilize the ratings from other like-minded users to produce a prediction score, while item-based CF generates a list of recommendations with the most similar items to a user-rated item. In both user-based CF and item-based CF, a similarity matrix is computed first to determine relationships between all pairs of users (for user-based CF; items, for item-based CF). Similarities between item- or user-pairs could be computed through different metrics. In this study, however, building the item-item matrix is extremely computationally expensive and ineffective in terms of accuracy due to large sparse item database (Sarwar, et al., 2001). Hence, we conduct user-based CF with kNN algorithms for customer recommendation.

In addition to using item- and user-similarity matrices, matrix-factorization based CF algorithms are also of researchers' and practitioners' interests. An algorithm among these is the famous SVD (singular value decomposition) algorithm, as popularized by Simon Funk during the Netflix Prize (Bennett & Lanning, 2007). In this method, it is assumed that a user's rating of an item is composed of a sum of preferences about the various aspects of that item (Funk, 2016). More recently, co-clustering methods have also been developed to allow for real-time collaborative filtering that does not incur the high costs of model parameter re-estimating with the entrance of new users, items and ratings (Banerjee, et al., 2007).

2.2 Popularity-based recommender systems

Popularity-based recommendation takes “popularity” score as criterion and is commonly seen in recommendation systems for social media, news sites and entertainment sites (Xia, et al., 2008). One application is the “most-viewed” recommendation features of online news. The underlying assumption is that the popularity of an item could, to some extent, reflect its worthiness to be recommended. In our study, we build recommender based on overall popularity of customers, as well as on weighted popularity in top product areas and top major categories.

3 Solution

3.1 Data description

Dataset used in this study comes from Data Set 11 from the Marketing EDGE (formerly Direct Marketing Educational Foundation) database. This dataset is from a cooperative database company and compiles household purchasing data from hundreds of companies (Courtheoux, n.d.). It contains household purchasing records from approximately 2.5 million households, with time span from January 2005 to December 2007. Total 13,382,011 order records were extracted, with 35,536,676 line item records for 207 participating companies associated with these orders. A main file ‘*DMEFLines3Dataset2.csv*’ provides order-related information such as line item records include the company ID, household ID, order date, major category and area the ordered product belongs to, purchased quantity, and total dollars spent. Besides, a reference table (‘*DEMEFYrBase.csv*’) that maps each household ID to its ZIP code is also available. Reference tables with 14 product area, 73 major category descriptions are also included.

We split the 36-month transactional records into a 30-month training period and a 6-month testing period. To be specific, to train our models, we use the household purchase records from January 2005 to June 2007, which contain all customer $i \in I$ and all company $u \in U$, credit a customer value score (rating by company for customer, or rating in short in the following sections) for each potential customer of the selected company $i \in I'_u$ (i.e. $i \in I \; and \; i \notin I_u$), and test the score (rating) with data from July 2007 to December 2007. The design of such customer acquisition system should vary with the context of chosen firm, and thus we need to select a focal company for our study. Most acquisition systems require the company’s existing customer base as input to analyze and identify potential customers. A large enough and heterogeneous customer base will result in a more general and robust model in terms of market profitability. Regarding this principle, we select Company 36, the company with the largest number of unique customers in the training period, to be the focus of our study.

3.2 Model design

We aim to design a model that recommend customers from potential customers to our focal company. Potential customers refer to the households who are yet to become customers of the targeting company but have made purchases with other companies in the training period. Two general types of recommenders are adopted to construct the proposed model:

1) Collaborative filtering recommenders based on customer purchase history matrix

2) Popularity-based recommenders based on customer expenditure and company’s revenue source

Each recommender model we build would return a ranked list of recommended potential customers with a rating. Based on the rating predicted, we first plot an acquisition profitability curve, which is similar to customer acquisition curve, with number of customers recommended as the horizontal axis and cumulative revenue generated as the vertical axis. With such graph, the focal company will be able to find the number of potential customer to target for any revenue generated by new customers. In addition, for each decile of top-rated customers, we obtain the mean revenue generated by each customer in the decile. This higher-level summary will allow the focal company to observe the change in mean revenue in each decile for different deciles of customers recommended by different recommenders.

4 Experiment

4.1 Collaborative filtering recommenders

As we have discussed in the variable construction section, for company u, we give each of its existing customer $i \in I_u$ some rating r_{ui}, based on the expenditures of the customer. Thus, by employing CF algorithms, for company u, we can predict the rating of the potential customers $i \in I'_u$. In this section, we will discuss how we construct different ratings first, following by the algorithms and specifications of models.

4.1.1 Rating construction

We define a variable r_{ui}, as the rating of company u for customer i. This variable represents the predicted value of customer i to company u, and is calculated in the following method.
For all orders that are placed with company u in the training period $i \in I_u$, we aggregate the household-company level expenditure/revenue and number of orders. In such way, we obtain the amount of money and number of purchases each household, as an existing customer, spend with different companies. r_{ui} can be defined as a transformation of such expenditure and number of orders. Four possible transformations are available: 1) original value of expenditures without transformation (dol, num); 2) scaling according to minimum and maximum of the value (dol_scale, num_scale); 3) logarithm transformation (dol_log, num_log); 4) scaling after logarithm transformation (dol_log_scale, num_log_scale).

p_{ui} is a variable that represents the performance of customer i for company u and is constructed as follows. For all potential customers $i \in I'_u$ for company u, we obtain the aggregated expenditure and number of orders at customer-company level in the testing period. For potential customers without any purchases in the testing period, the expenditure and number of orders are both 0. p_{ui} is thus defined as a transformation of such expenditure or number of orders. The transformations are generally similar to those of r_{ui}. However, since such expenditure can be zero for certain u and i, i.e., for company u, its potential customer i remain a non-customer in the testing period, we will hence add 1 to the value of expenditure before logarithm transformation, in order to prevent mathematical error. With these two variables constructed, we can have additional measures to evaluate the accuracy of different algorithms under the same type of rating. We consider Root Mean Square Error (RMSE). That is, we compare how the predicted rating $\hat{r}_{ui}$ for $i \in I'_u$, is different from p_{ui}. RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i \in I'_u} (\hat{r}_{ui} - \mathrm{p}_{ui})^2}{|I'_u|}}$$

In the following subsections, we will discuss the algorithm designs and parameters settings, for which the different ratings designed will be used in.

4.1.2 User-based k nearest neighbors

Firstly, we conduct collaborative filtering based on k nearest neighbours (kNN), which is a common method. We first to use the conduct kNN collaborative filtering with user similarity as the Mean Squared Difference similarity between all pairs of users. Mean Squared Difference (MSD) is defined as:

$$\mathrm{msd}(\mathrm{u},\mathrm{v}) = \frac{1}{|I_{uv}|} \cdot \sum_{i \in I_{uv}} (\mathrm{r}_{ui} - \mathrm{r}_{vi})^2$$

The MSD-similarity is then defined as:

$$\mathrm{sim}(\mathrm{u},\mathrm{v}) = \frac{1}{msd(u,v)+1}$$

We then predict $\hat{r}_{ui}$ for $i \in I'_u$ as the weighted mean of r_{vi} for v being in the nearest 40 neighbours of company u:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^{40}(u)} sim(u,v) \cdot \mathrm{r}_{vi}}{\sum_{v \in N_i^{40}(u)} sim(u,v)}$$

To possibly improve the results, we also explored cosine similarity, Pearson Correlation coefficients, and Pearson Correlation coefficients with baselines, whose definition will not be discussed in details here.

4.1.3 Singular Value Decomposition (SVD)

We conduct collaborative filtering based on SVD, a common matrix factorization-based method (Salakhutdinov & Mnih, 2007). With SVD, we give both companies and customers coordinates in a low dimensional feature space. Each known and unknown rating (r_{ui} for $i \in I, u \in U$) is modelled by the mean score in training set μ, the inner product of estimated corresponding customer feature and company feature vectors ($q_i^T p_u$), and estimated user- and item-bias (b_u and b_i). We predict the unknown rating, r_{ui} for $i \in I'_u$ as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

To estimate the unknown parameters (user- and item-related score biases, b_u and b_i; estimated customer feature and company feature vectors, q_i^T and p_u), we use 20 iterations of stochastic gradient descent and regularization level λ being 0.02 to minimize the following regularized squared error:

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2 \right)$$

4.1.4 Co-clustering

We conduct collaborative filtering based on co-clustering, whereby users and items are assigned some clusters C_u, C_i and some co-clusters C_{ui}, with straight forward optimization method much like k-means (George & Merugu, 2005). With average scores of clusters and co-clusters computed ($\overline{C_u}$, $\overline{C_i}$, $\overline{C_{ui}}$) and mean scores for customer i and company u (μ_i, μ_u), we predict the unknown rating, r_{ui} for $i \in I'_u$:

$$\hat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i})$$

4.2 Recommender system based on popularity rules

With the hypothesis that customers who spend more in the past are likely to spend more in the future, we recommend potential customers with high existing expenditures to the targeted company. In the following subsections, we will discuss how we construct different ratings based on customer expenditures and company's revenue sources.

4.2.1 Rating customers by their respective total expenditure

r_{ui} can be defined as the total expenditure of customer *i* in all purchase orders in the training period:

$$\hat{r}_i = \sum\nolimits_{o \in O_{Train}} \mathrm{v_{i,o}}$$

4.2.2 Rating customers by weighted expenditure in the company top product areas

The proportion of the revenue of company *u* from product area *a* is defined as:

$$\mathrm{w_{u,a}} = \frac{\sum_{o \in Train} \mathrm{v_{u,a}}}{\sum_{o \in O_{Train}, j \in A} \mathrm{v_{u,j}}}\text{, where } A \text{ is a set of all 10 product areas}$$

We then identify the top 3 product area for the company as a_1, a_2, a_3, which forms a set A'. The rating defined as the weighted expenditure of customer *i* in purchase order *o* in top product area $a_0 \in A'$in the training period:

$$\hat{r}_{ui} = \sum\nolimits_{o \in Train,} \mathrm{v_{i,o}} * w_{u,a_o}\text{, where } a_o \text{ is the area of order and } a_0 \in A'$$

4.2.3 Rating customers by weighted expenditure in the company top product categories

Similarly, the proportion of the revenue of company *u* from product category *c* is defined as:

$$\mathrm{w_{u,c}} = \frac{\sum_{o \in Train} \mathrm{v_{u,c}}}{\sum_{o \in O_{Train}, j \in C} \mathrm{v_{u,j}}}\text{, where } C \text{ is a set of all 69 product categories}$$

We then identify the top 10 product categories for the company, which forms set C'. The value rating of customer *i* is defined as the weighted expenditure of customer *i* in purchase orders in top product categories in the training period:

$$\hat{r}_{ui} = \sum\nolimits_{o \in Train,} \mathrm{v_{i,o}} * w_{u,c_o}\text{, where } c_o \text{ is the area of order and } c_0 \in C'$$

5 Results

In this section, we will first discuss the results obtained by conducting basic CF models on expenditure related ratings, for which kNN will be only based on MSD-similarity. Then, we would like to improve upon the models by also considering number of orders-related ratings, and altering the similarity measure used among companies. Thereafter, we will show the results obtained from popularity-based rules.

5.1 Performance based on basic collaborative filtering

For each rating r_{ui} constructed and performance p_{ui} constructed, we have run the models with Python Surprise implementation for kNN, SVD and co-clustering algorithms. First, we compare the algorithm performances by RMSE and running time in Table 1. Across the 3 algorithms, running time required is generally the fastest for kNN, followed by Co-clustering and SVD. However, RMSE is generally the highest for kNN. This means the estimated rating $\hat{r}_{ui}$ is further apart from the true performance p_{ui}, and can represent p_{ui} by a smaller degree. Thus, kNN saves computational resources at the expense of some degree of accuracy. In addition, RMSE for co-clustering is the lowest, possibly indicating higher accuracies.

However, the aim of our models is to prioritize the potential customers. Thus, we need to investigate further in the customers with high predicted rating $\hat{r}_{ui}$. To do so, we observe the acquisition profitability curve in Figure 1. One main finding is that for using the expenditure directly as ratings (dol), the actual revenue per potential customer based on quantiles is almost constant. Drilling down to the details of $\hat{r}_{ui}$ across all these predictions, we observe that $\hat{r}_{ui}$is always constantly at 5. The limitation is from the Python Surprise library used, which only predicts ratings from 1 to 5. This is also the reason for us to design the ratings with logarithm transformations or min-max scaling to the range of 1 to 5.

	dol		dol_scale		dol_log		dol_log_scale	
	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time
kNN	9.9981	31:58	0.0166	06:49	4.3425	07:19	1.4845	04:33
SVD	9.9993	27:26	0.0155	39:31	3.7073	43:19	1.2045	30:50
Co-clustering	9.5782	21:18	0.0156	33:16	3.5792	34:53	0.9245	21:57

Table 1. Performance for algorithms under each expenditure/revenue-related rating design

To illustrate the revenue generated by potential customers more clearly, for each decile of potential customer $i \in I'_u$ ranked by predicted rating $\hat{r}_{ui}$, we compute the mean of p_{ui}.We then plot the trend of mean revenue generated for each decile in Figure 2. The result indicates that the mean revenue in top deciles are usually higher for SVD across all across rating types. This indicates collaborative filtering with SVD is the best algorithm to use for prioritization.

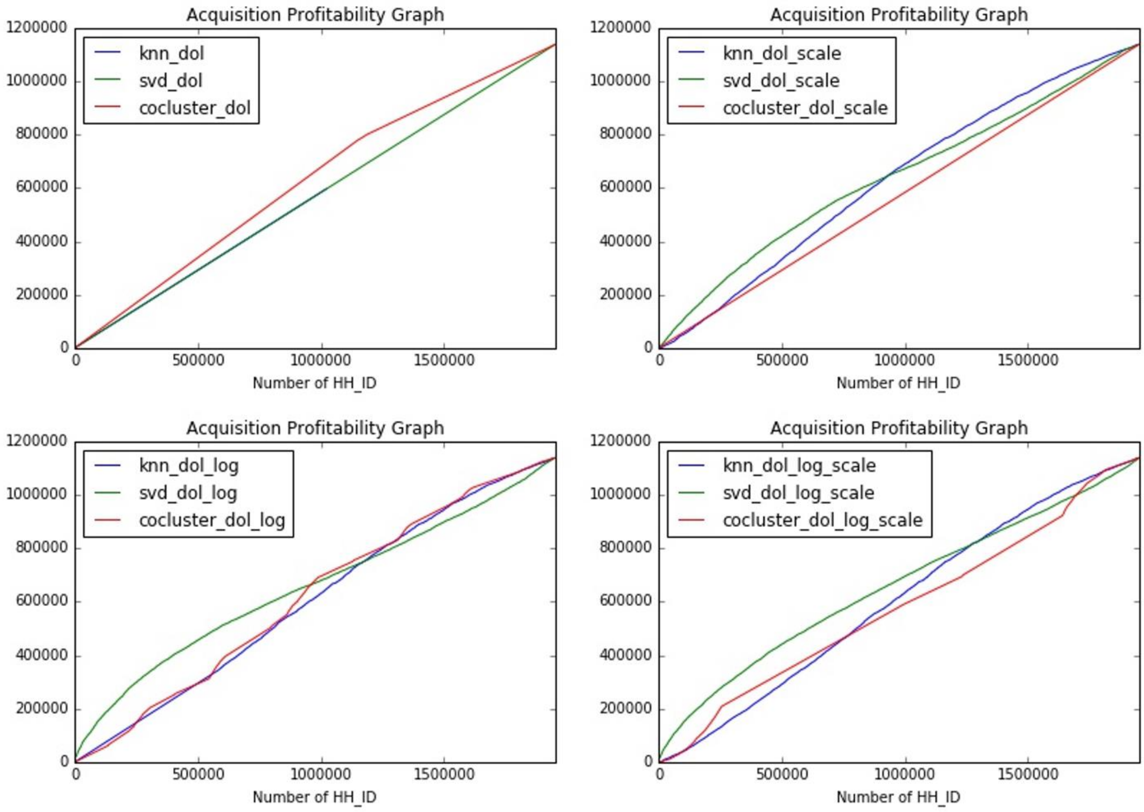


Figure 1. Acquisition profitability curves for algorithms with dol-related ratings

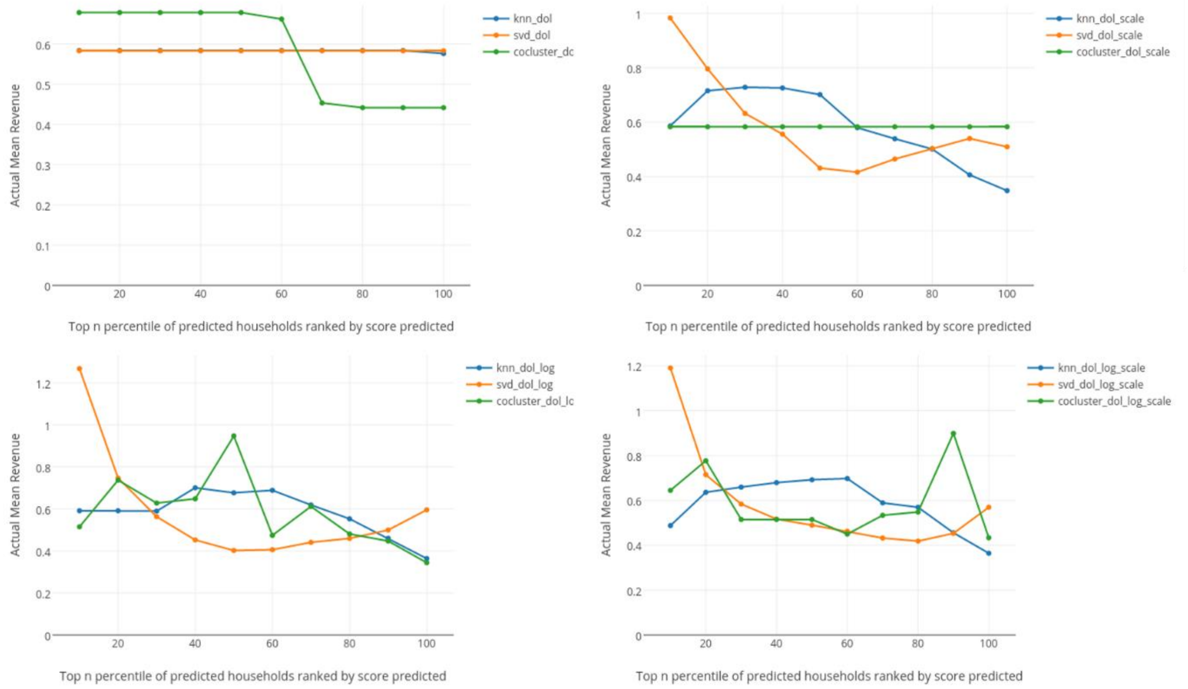


Figure 2. Comparison of mean revenue in each decile for algorithms under dol-related ratings

We look into SVD further in Figure 3, of dol_log and dol_log_scale outperform dol and dol_scale. The best score design is dol_log, with top decile potential customers providing the company with revenue over $1.2 per customer in the testing period.

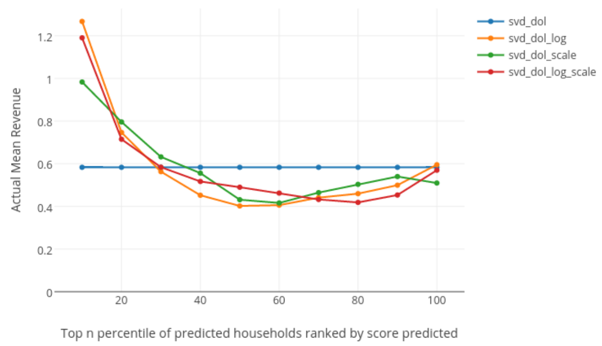
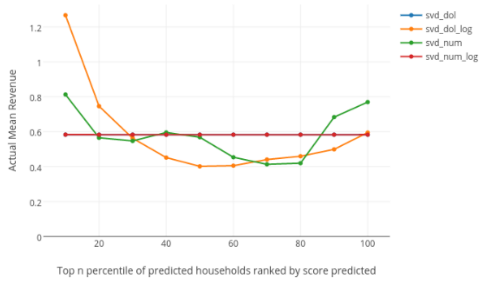


Figure 3. Comparison of mean revenue in each decile for different dol-related ratings (SVD)

5.2 Improving the performance of collaborative filtering models

The best results obtained previously are SVD models and scaling seems unnecessary with small difference between svd_dol_log and svd_dol_log_scale (Figure 3). Thus, in this section, we will try to improve the top-decile per-customer revenues by considering:1) using unscaled ratings related to number of orders for the more accurate SVD models; 2) using unscaled ratings related to number of orders and different similarity measures among users for the fastest kNN models.
Figure 4 indicates the ineffectiveness of the first method, with using number based ratings giving a lower mean revenue per potential customer. This is probably because we are using the actual revenue contributed to the focal company in testing period by per potential customer as the vertical axis, which

is by dollars but not by numbers or probabilities. Thus, using number-based ratings may be less suitable for prioritizing non-customers with potentially high values for the SVD algorithm.
Figure 5 include 13 lines of revenue mean and can be viewed more clearly with the demo, which offers the function of group selection by labels (rating types and similarity measures). Among all the 13 models, the best model is kNN algorithm with ‘num’ as rating scheme and cosine similarity among companies. By its prediction with top decile potential customers providing the company with revenue about $0.988 per customer in the testing period.



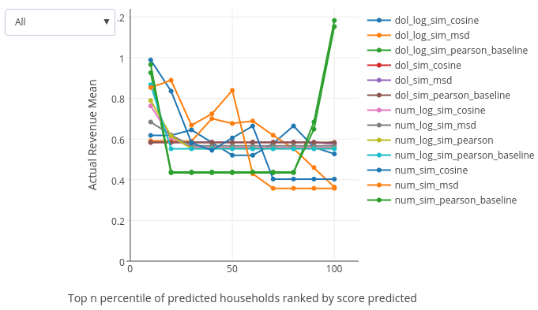
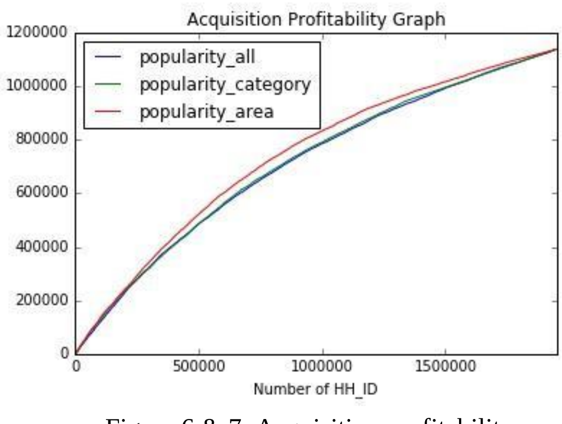


Figure 4 & 5. Comparison of mean revenue in each decile for different dol- and num-based ratings (SVD) & Comparison of mean revenue in each decile for different ratings and similarity measures (kNN)

5.3 Performance based on popularity-based rules

In both Figure 6 and Figure 7, we can observe the an almost strictly decreasing customer value in the training period by the predicted ratings. This means the popularity-based rules are very robust in defining customer segments for the focal company. In addition, the model with popularity_area rating slightly outperforms the other two. This indicates that the general product areas, rather than specific product sub-categories, are more informative about a customer’s purchasing behaviors.



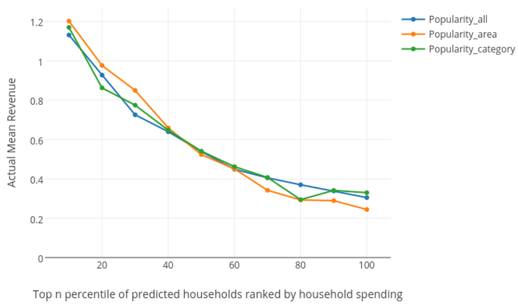
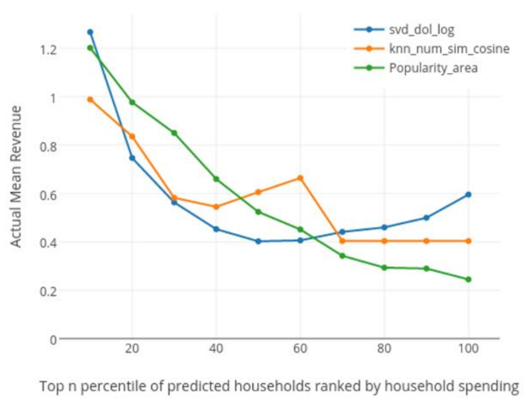


Figure 6 & 7. Acquisition profitability curves & Comparison of mean revenue in each decile

6 Evaluation

Multiple plots in Figure 8 display the similar trend that the actual profitability of prospecting customers with higher ranking is much greater than that of prospects with lower ranking for our selected models. This suggests that our models are generally effective in terms of acquiring the “right” customers, and customer past purchase behaviors and popularity are solid indicators in identifying profitable potential customers from non-customers. New customer acquisition efficiency is increased with our solution compared to untargeted mass marketing.
The acquisition-profitability curve also provides insight for marketers’ decision making. Marketers with a clearly defined revenue goal would know the minimum of customers to be acquired to achieve

the goal, thus reducing the acquisition costs. Companies with limited marketing budget would learn which are the more promising customers to target to maximize future revenues. In all, the company's profitability would increase.



Decile by $\hat{r}_{ui}$	Actual Revenue Mean (6 months)		
	SVD (dol_log)	Popularity (Top3 areas)	kNN (num, sim: cosine)
1	**1.267**	**1.202**	**0.988**
2	**0.747**	**0.977**	**0.835**
3	0.563	0.850	0.582
4	0.452	0.659	0.545
5	0.402	0.524	0.605
6	0.406	0.451	0.664
7	0.441	0.342	0.403
8	0.460	0.293	0.403
9	0.499	0.289	0.403
10	0.595	0.245	0.403

Figure 8. The most effective models in Section 5 for customer acquisition prioritization

7 Limitations

In this study, we developed and examined multiple models for a specific merchandising company, which are proved to be efficient in customer acquisition. However, there still exist several limitations. Firstly, in model design we assume that customer's profitability can be represented by the revenue generated by the customer, which, considering that various other variable costs are not included in this study, does not always hold in the real world. Other indicators of profitability should be developed and studied in future work. Secondly, due to the constraint of computational power, we are unable to perform algorithms that are computationally costly. For our recommender model, we are unable to probe item-based collaborative filtering, as it is almost impossible for us to compute the 2.5 million by 2.5 million household similarity matrix. Similarly, due to computational costs, we are unable to optimize our models with more iterations in SGD or more clusters for co-clustering. All these can be improved with future studies.

8 References

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. S. (2007). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, *8*(Aug), 1919-1986.

Bennett, J., & Lanning, S. (2007). The netflix prize. In *Proceedings of KDD cup and workshop* (Vol. 2007, p. 35).

Bureau of Economic Analysis. (2017). National Income and Product Accounts. Retrieved from https://www.bea.gov/newsreleases/national/gdp/gdpnewsrelease.htm

Courtheoux, R.J. (n.d.). Description and Documentation for the Cooperative Database Company Dataset Version 1.0. Retrieved from http://www.marketingedge.org/sites/default/files/content/multiuse/DMEFDataSet11.pdf

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, *4*(2), 81-173

George, T., & Merugu, S. (2005, November). A scalable collaborative filtering framework based on co-clustering. In *Data Mining, Fifth IEEE international conference on* (pp. 4-pp). IEEE.

Sheffer, R. (2015). Why there's high competition in the US telecom industry. Retrieved from http://marketrealist.com/2015/01/theres-high-competition-us-telecom-industry/

Salakhutdinov, R., & Mnih, A. (2007). Probabilistic Matrix Factorization. *Nips, Vol. 1, No. 1*, pp. 2-1

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms.* Published in WWW '01 Proceedings of the 10th international conference on World Wide Web, pp. 285-295. Retrieved from http://dl.acm.org/citation.cfm?id=372071

Xia, W., He, L., Ren, L., Chen, M., & Gu, J. (2008). *A New Collaborative Filtering Approach Utilizing Item's Popularity.* Published in Industrial Engineering and Engineering Management International Conference on IEEE. Retrieved from http://ieeexplore.ieee.org/document/4738117/

Funk, S. (2016). Netflix Update: Try This at Home http://sifter.org/~simon/journal/20061211.html