Prediction of Co-purchasing Products

Yilun Liu
Computer Science and Engineering
University of California, San Diego
La Jolla, California, 92093

Chunhao Wu
Computer Science and Engineering
University of California, San Diego
La Jolla, California, 92093

Xiaohui Tong
Computer Science and Engineering
University of California, San Diego
La Jolla, California, 92093

***Abstract*—Predicting co-purchasing products based on previous order history of users can help online shopping websites to recommend proper products to users. Successful prediction of co-purchasing products can help the websites increase revenue. In our project, we analyze data from Amazon product co-purchasing network metadata collected by crawling Amazon website and contains product metadata and review information about 548,552 different products (Books, music CDs, DVDs and VHS video tapes) to find out some key reasons behind co-purchasing and create a model to predict the co-purchasing products of a product based on features that relate to co-purchasing.**

***Keywords*—IEEEtran, journal, LaTeX, paper, template.**

I. Dataset Characteristics

The dataset we use is Amazon product co-purchasing network metadata from Stanford Network Analysis Project. The data was collected by crawling Amazon website and contains product metadata and review information about 548,552 different products (Books, music CDs, DVDs and VHS video tapes).For each product the following information is available:
Title
Salesrank
List of similar products (that get co-purchased with the current product)
Detailed product categorization
Product reviews: time, customer, rating, number of votes, number of people that found the review helpful
The data was collected in summer 2006[1].

A. Basic Stats

The total number of products in the dataset is 548,552. Within the dataset 1,788,725 pairs of products are listed as co-purchasing products. The dataset also includes 7,781,990 different reviews for the products. On average, product in the dataset has 3.29607101002 co-purchasing items.

B. Product Groups

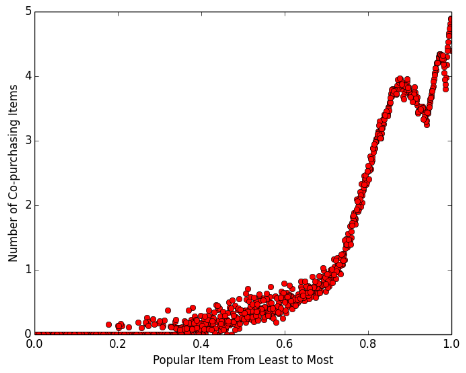
As mentioned ealier, the dataset contains products in the following 4 groups: Books, music CDs, DVDs and VHS video tapes. Below are the number of products in each group:

Number of Books: 393,561
Number of DVDs : 19,882
Number of Music CDs: 103,144
Number of Videos: 26,132

C. Dataset Analysis: Single Item Analysis

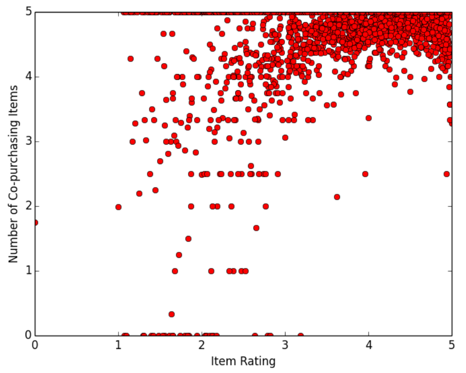
In order to find the features that matter when deciding if two products are co-purchasing pairs, we analyze the data by plotting the distribution of number of co-purchasing pairs based on different feature.

(1) Sales Rank (correlation:0.8507009966)
Sales rank analysis tries to see if there is a correlation between the popularity of a product and the number of co-purchasing products the product has.



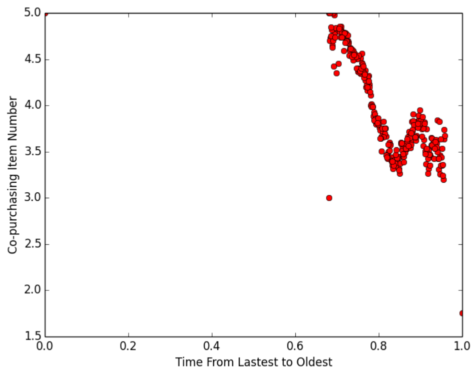
Based on this graph, popular items(items with more purchasing times) tend to have a higher number of co-purchasing items.

(2) Item Rating (correlation:0.2965513)
Item rating analysis tries to see if there is a correlation between the average rating of a product and the number of co-purchasing products the product has.



Item rating and number of co-purchasing items have a weak correlation, but high rating items tend to have more co-purchasing items.

(3) Time of First Review
Time of first review analysis tries to see if there is a correlation between the time the product is first reviewed and the number of co-purchasing products the product has.



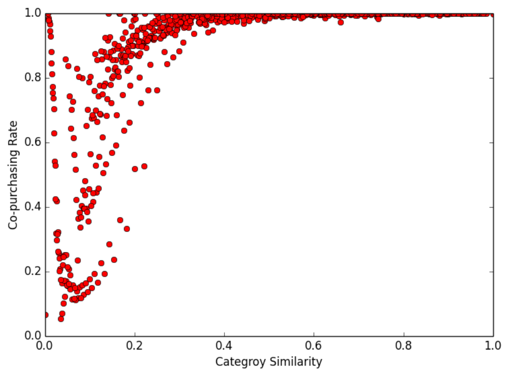
Items whose first reviews were made earlier tend to have less number of co-purchasing items.

D. Dataset Analysis: Pair Analysis

Next we try to find out shared features between each pair of co-purchasing products.

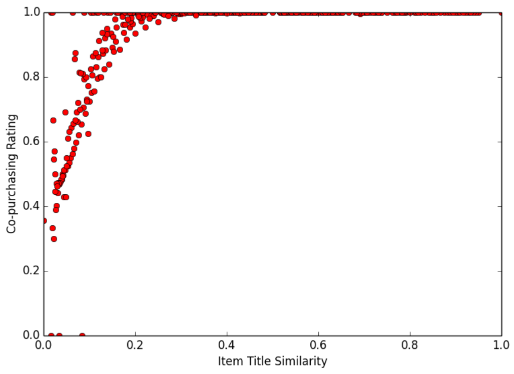
(1) Person Similarity based on reviews (correlation : NA)
While testing on this similarity, we found that no two items share reviews written by the same user. Therefore, this similarity cannot be measured.

(2) Category Similarity (correlation:0.607774)
Category similarity tries to find out the relation between the categories of two products and see how the similarity of categories influence the results of co-purchasing. The way to calculate the category similarity for product A and B is:
$\frac{|category(A) \cap category(B)|}{|category(A) \cup category(B)|}$



From the graph we can tell that two items that belong to similar categories tend to be purchased together more frequently.

(3) Title Similarity (correlation:0.56355022)
Title similarity focuses on whether two items that are of similar titles will be purchased together more often or not. The way to calculate the title similarity for product A and B is:
$\frac{|wordsintitle(A) \cap wordsintitle(B)|}{|wordsintitle(A) \cup wordsintitle(B)|}$



According to our analysis, two items that share similar titles tend to be co-purchased more frequently.

(4) Pairs Group Attributes
Since the metadata contains products that belong to 4 different group: Book, Music, DVD, and Video, we are also curious about that, for all the co-purchasing pairs, what are the groups of products in each pair.

	Book	Music	DVD	Video
Book	1260354	274851	55666	70965
Music	274851	182945	16575	19063
DVD	55666	16575	41192	35215
Video	70965	19063	35215	18980

Book group tends to have many co-purchases items than other groups, probably because the dataset includes a large number of books.

II. Predictive Task

A. Task: Co-purchasing Pairs Prediction

In this project, we are interested in the problem that, given two products and their basic information(title, sales rank, detailed product categorization, and product reviews), decide if they will be co-purchased, or if they are co-purchasing products. With the results of that, we can further choose to recommend certain products when the users are browsing certain products, or after the users purchase some products.

To actually perform the predictive task, we randomly choose co-purchasing pairs and then randomly choose the same number of non co-purchasing pairs to build a test set, so that among all the pairs in the test set, exactly 50% of the pairs correspond to co-purchasing items and the other 50% do not.

B. Relevant Features

(1) Category Similarity
Based on our data analysis, two items that belong to similar categories tend to be co-purchased more often. The way to calculate the category similarity for product A and B is:
$\frac{|category(A) \cap category(B)|}{|category(A) \cup category(B)|}$

(2) Sales Rank of Two Items
Based on our data analysis, items that have a higher sales rank(or items that have been purchased for more

times) tend to have a higher number of co-purchasing products.

(3) Title Similarity
Based on our data analysis, two items that share similar titles tend to be co-purchased more often. The way to calculate the title similarity for product A and B is:
$\frac{|wordsintitle(A) \cap wordsintitle(B)|}{|wordsintitle(A) \cup wordsintitle(B)|}$

(4) Average Rating
Based on our data analysis, we find that there is a weak correlation between average rating of an item and the number of its co-purchasing items. Items that are of higher average ratings tend to have a larger number of co-purchasing items.

(5) Time of First Review Of Two Items
Based on our data analysis, items whose first reviews were make earlier tend to have a smaller number of co-purchasing items.

(6) Group
According to the chart in Section 1.D, item 4, each group has its own distribution of number of co-purchasing pairs among four groups.

C. Data Preprocessing

The metadata we get is of text format, containing products and their information as described in section 1. In order to have a data file that is easy to process and only keeps the relevant features, we parse the data so that we have json, in which each line records the information of a product. For each product, we save its title, key, ASIN(unique identification number of the product that Amazon uses), Id(the identification number of the product in the metadata text file), sales rank, number of similar items and the ASIN of similar items, category, and reviews, each review have its date, customer(who made the review), rating, number of votes, and number of helpful votes.

III. Relevant Literature and Model

A. Relevant Literature

Based on the relevant features and their relations to our predictive task, we start our research on recommender systems based on co-purchasing items. We find the following models that may be useful.

(1) Collaborative Filtering
Collaborative filtering is a technique usually used by recommender system. It is the process of filtering for information or patterns based on different viewpoints[2]. It is often used to match people with similar interest or items with similar features. In our predictive task, we are going to use various features related to co-purchasing pairs to collaboratively filtering similar items of an item.

(2) Jaccard Similarity The Jaccard coefficient measures similarity between two sets, and is defined as the size of two sets intersection divived by the size of their union[3]:
$\frac{|A \cap B|}{|A \cup B|}$
Coefficient zero means no similarity and one represents two sets are identical. We use Jaccard Similarity to figure out the similarities between titles of two items, and their categories.

(3) K-Nearest Neighbors Algorithm
K-Nearest Neighbors is a non-parametic method used for unsupervised learning and supervised learning[4]. In supervised learning scenario, the method first records all training data with their labels and label test data according to its k-nearest neighbor. We use k-Nearest Neighbors algorithm to classify the co-purchasing items based on the co-purchasing items characteristics of k-Nearest Neighbors.

(4) Logistic Regression
Logistic Regression is a linear classification method that trains weights to minimize square error between ground truth labels and predicted labels. In our predictive task we can use logistic regression to classify if a given pair of products are considered co-purchasing items.

(5) Decision Tree Learning
Decision Tree is a learning method that splits input feature space according to the information gain[5]. The feature with the highest information gain is used to split the data first. It can automatically detect the most useful feature. In our predictive tasks, we use it to train our model and test if a pair of item has co-purchasing relationship. However, decision trees are prone to overfit training data.

(6) Adaptive Boosting
AdaBoost is an ensemble method that trains multiply classifiers. The method first trains a classifier on the original data. In subsequent steps, it trains new classifier by adding weights on previously mislabeled data. This method allows classifiers to predict difficult instances correctly[6]. In our predictive task, we use decision tree as AdaBoosts base classifier.

(7) Random Forest
A random forest[7] is an ensemble classifier that trains multiple decision trees. Each decision tree is trained on a subset of original data. In predicting stage, each decision tree outputs a label and the final result is decided through a voting strategy. Random Forest largely improves test accuracy by reduce overfitting caused by a single decision tree.

(8) Cross-validation
Cross-validation is a statistical method to test a models accuracy in real situation[8]. It randomly splits data into k parts and performs k validation rounds. In each round, one part of data is used as test data and the rest as training data. We use this method to test the accuracy of our model.

B. Model

Based on all the related models, the data we have, the relevant features, and their relations with the our predictive task, we finally decide to use the following model.

We first start with logistic regression. Based on the results of data analysis, we get a list of relevant features. We train our model with different combinations of features to see which combination gives us the best training result. After we find out the combination of features that produces the best training result using logistic regression, we apply these features to other models, including decision tree, Ada boost, k-neighbors classifier, and random forest.

The way we value the quality of our training result is through accuracy:
$\frac{|CorrectPrediction|}{|TotalPrediction|}$
Accuracy number that is closer to 1 is considered a better result.
The baseline of our predictive task is calculated using logistic regression model. We choose the feature sales rank to train our baseline. That it, we decide if a pairs of products are co-purchasing items only based on it they are sold a lot or not.

We randomly separate our data to 5 sets, each containing same number of products to do a 5 fold cross validation. At the training stage, we choose 4 out of these 5 sets as training data. The set left will be used to generate testing data, we scan through the set to find the similar products of each product in this set, and id the similar products are also in this set, we add the pairs to the testing set as positive pairs, and randomly choose the same number of negative pairs from this set to compose our testing set. We do this for 5 rounds, each time one of these 5 sets will be chosen to generate the test data, while the other 4 sets will be used as training data for that round. This can help us prevent overfitting the test data in one specific set.

IV. Results

A. Choosing Features

As mentioned in the previous sections, we first train to choose the best combination of features. We start from our baseline, sales rank.

(1) Sales Rank
We tune the sales rank such that 50% of the pairs in our test data will be categorized as co-purchasing items, while others will not.
The accuracy rate we get from this baseline model is 0.672029100271.

(2) Category Similarity
We calculate the category similarity between two products in each pair using Jaccard Similarity, and then try this feature on the training data.
The accuracy rate we get from this baseline model is 0.874189777097. We see a great improvement in accuracy this time, suggesting that category similarity can be an important feature to include.

(3) Sales Rank and Category Similarity
We include both sales rank and category similarity this time to build the feature matrix.
The accuracy rate we get from this baseline model is 0.881858477787, we see a minor improvement in accuracy.

(4) Sales Rank, Category Similarity, and Group This time we add extra group features based on group distribution of co-purchasing pairs, then tune the features again.
The accuracy rate we get from this baseline model is 0.896765552514, we see a minor improvement in accuracy.

(5) Sales Rank, Category Similarity, Group, and Average Rating
We then try to incorporate average rating of the product to train our model as products of higher average rating tend to have more similar products.
The accuracy rate we get from this baseline model is 0.896427814035, which is not an improvement.

(6) Sales Rank, Category Similarity, Group, and Title Similarity
This time we calculate the Jaccard Similarity of titles of two products in the pair. Then we re-train our model with additional title similarity feature.
The accuracy rate we get from this baseline model is 0.907579080482, we see a minor improvement in accuracy.

(7) Sales Rank, Category Similarity, Group, Title Similarity, and Time of First Review
Since products with earlier time of the first review tend to have a larger number of co-purchasing products, we add this feature to our model and train again.
The accuracy rate we get from this baseline model is 0.910581337478, we see a minor improvement in accuracy.

Now we finished logistic regression training with different combinations of relevant features, based on the following result chart, we choose the combination of sales rank, category similarity, group, title similarity, and time of first review features.

Feature	Accuracy
SALES only	0.672029100271
CATE only	0.874189777097
CATE + SALES	0.881858477787
CATE + SALES + G	0.896765552514
CATE + SALES + G + AR	0.896427814035
CATE + SALES + G + TS	0.907579080482
CATE + SALES + G + TS + TIME	0.910581337478

CATE = Category Similarity
SALES = Sales Rank of two items
TS = Title Similarity
AR = Average Rating
TIME = Time of First review of two items
G = Group

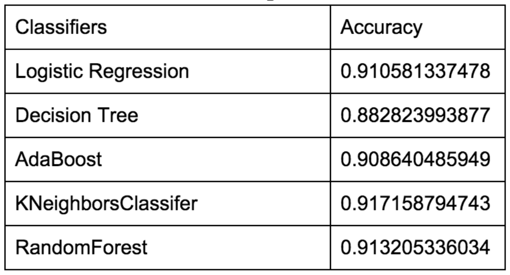
B. Choosing Training Functions

Based on our research on relevant literature, we decide to try the following training functions besides linear regression:

Decision Tree, AdaBoost, K-Neighbors Classifier, Random Forest.

For each classifier, we build a M by N matrix to represent the features of all the pairs in the training set. M is the number of training pairs, and N is the dimension of features. We build another M by 1 matrix(vertor) to track if each pair is considered a co-purchasing pair or not. Then we use these two matrix as parameters of each training functions to generate the result.

After the training, we get similar results for all these functions, but still some perform better than others. The results are listed in the following chart:



Classifiers	Accuracy
Logistic Regression	0.910581337478
Decision Tree	0.882823993877
AdaBoost	0.908640485949
KNeighborsClassifer	0.917158794743
RandomForest	0.913205336034

We can see from the above chart that K-Neighbors Classifier gives us the best result.

V. Conclusion

Based on the results of data analysis and our predictions of different combinations of features, we find that category similarity is really useful when predicting the co-purchasing pairs, which means that items within the similar categories tend to be purchased together more often. Besides, sales rank is also pretty helpful for prediction, suggesting that popular items will be purchased along with other items more often. Other features including title similarity, time, and group help improve the accuracy by a little bit, while average ratings are not really playing a role when predicting co-purchasing pairs.

Training function wise, for this particular prediction task, according to our test results, k-neighbors algorithm is the most successful one, followed by random forest, logistic regression, and AdaBoost, while decision tree is not of very well performances.

References

[1] J. Leskovec, L. Adamic and B. Adamic. ”The Dynamics of Viral Marketing” in *ACM Transactions on the Web (ACM TWEB)*, 1(1), 2007.

[2] P. Melville and V. Sindhwani, ”Recommender Systems,” in *Encyclopedia of Machine Learning* Claude Sammut and Geoffrey Webb (Eds): Springer, 2010.

[3] P. Tan, et al, *Introduction to Data Mining* 3rd ed. New York, U.S.: Pearson, 2014

[4] N. S. Altman, *An introduction to kernel and nearest-neighbor nonparametric regression* The American Statistician, Vol. 46, No. 3. August 1992, pp. 175-185.

[5] L. Rokach, O. Z. Maimon, *Data mining with decision trees: theory and applications* World Scientific Pub Co Inc., 2008

[6] Y. Freund, R. Schapire, *A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting*, 1995.

[7] L. Breiman, Random Forests in *Machine Learning*, 45(1), 5-32, 2001.

[8] P. A. Devijver, J. Kittler, *Pattern Recognition: A Statistical Approach* London, GB: Prentice-Hall, 1982