Team 07 - Update 2

CS6220 - Data Mining Techniques - Fall 2017
Northeastern University

Nakul Camasamudram, Rosy Parmar, Rahul Verma, Guiheng Zhou

November 21, 2017

1 Introduction

We are building a recommender system that provides personalized recommendations based on prior implicit feedback using The Instacart Online Grocery Shopping Dataset 2017. Instacart is an American company that operates as a same-day grocery delivery service. This anonymized dataset contains a sample of over 3 million grocery orders from more than 200,000 Instacart users.

1.1 Data Analysis

Orders from Instacart are available in four .csv files: ”orders.csv”, ”order_products__train.csv”, ”order_products__prior.csv” and ”sample_submission.csv”. The key to understanding the dataset and the train/test split is the orders table (”orders.csv”).

Take for example User 1[Fig 1] , who happens to be a train user. User 1 has 10 prior orders, and 1 train order whose details are provided in ”order_products__prior.csv” and in ”order_products__train.csv” respectively.

Similarly, User 4 is a test user. He has 5 prior orders, and his 6th is a test order. Their details are available in ”order_products__prior.csv” and ”sample_submission.csv” respectively.

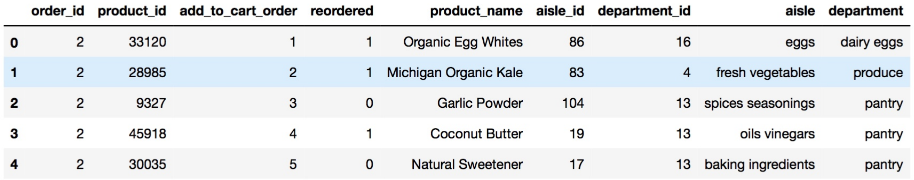
order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
2539329	1	prior	1	2	08	
2398795	1	prior	2	3	07	15.0
473747	1	prior	3	3	12	21.0
2254736	1	prior	4	4	07	29.0
431534	1	prior	5	4	15	28.0
3367565	1	prior	6	2	07	19.0
550135	1	prior	7	1	09	20.0
3108588	1	prior	8	1	14	14.0
2295261	1	prior	9	1	16	0.0
2550362	1	prior	10	4	08	30.0
1187899	1	train	11	4	08	14.0

(a) User 1

3343014	4	prior	1	6	11	
2030307	4	prior	2	4	11	19.0
691089	4	prior	3	4	15	21.0
94891	4	prior	4	5	13	15.0
2557754	4	prior	5	5	13	0.0
329954	4	test	6	3	12	30.0

(b) User 4

Figure 1: Train/Test Split

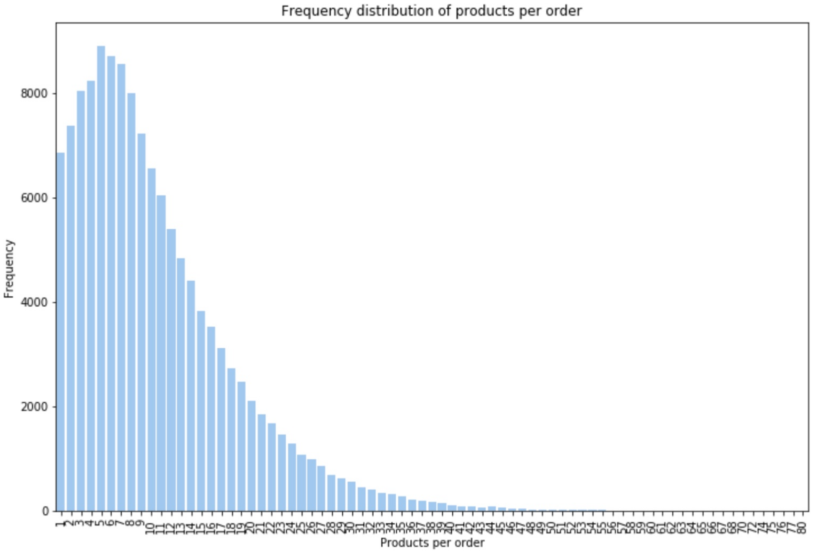


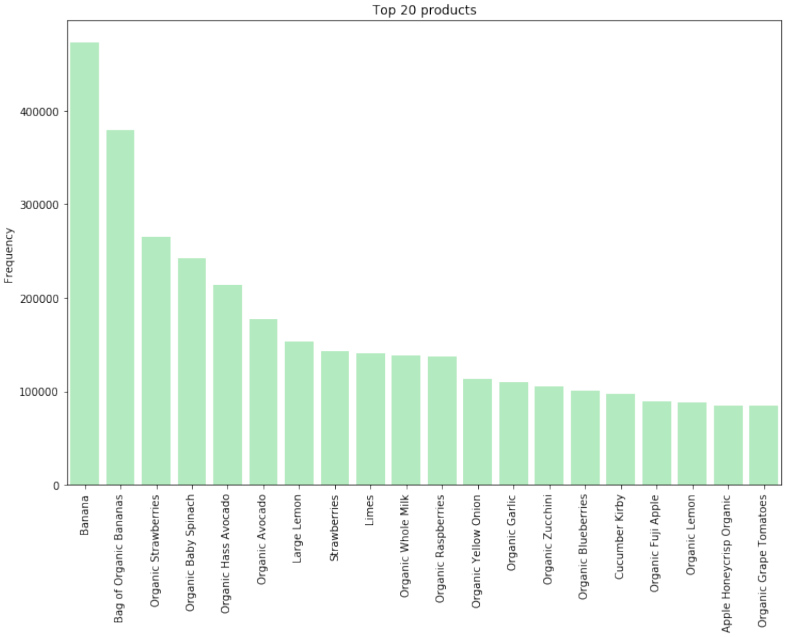
	order_id	product_id	add_to_cart_order	reordered	product_name	aisle_id	department_id	aisle	department
0	2	33120	1	1	Organic Egg Whites	86	16	eggs	dairy eggs
1	2	28985	2	1	Michigan Organic Kale	83	4	fresh vegetables	produce
2	2	9327	3	0	Garlic Powder	104	13	spices seasonings	pantry
3	2	45918	4	1	Coconut Butter	19	13	oils vinegars	pantry
4	2	30035	5	0	Natural Sweetener	17	13	baking ingredients	pantry

Figure 2: Merged Prior Orders

Figure 2 is a glimpse at ”order_products__prior.csv” when merged with three other .csv files that represent products, aisles and departments. The format of ”sample_submission.csv” and ”order_products__train.csv” is exactly the same.

The below figures depicts the product frequency distribution across orders as well as the most popular products. The ”long tail” phenomenon is clearly visible in the former.





1.2 Challenges

We are using algorithms specifically suited for processing implicit feedback. It is important to highlight unique characteristics of implicit feedback, which prevent the application of algorithms that were designed for explicit feedback data.

1. The dataset has information about users prior purchases from which we can infer what they like. However, there is no concrete metric to deduce what a user dislikes.

2. Just because a user purchased a product, does not necessarily mean he/she likes the product. The purchase could've been made as a gift or perhaps after receiving the product, the user might've been disappointed with it. Also, the frequency of product purchases doesn't necessarily indicate a user's preference, it's more of an indicator for a user's confidence in the product. Hence, a user's true preferences can only be guessed.

3. Systems dealing with explicit feedback are generally evaluated using metrics such as mean average error. With implicit systems, the recommender's output is compared with a user's current purchase on the test set using set based measures.

2 Approaches

2.1 Reviewed Approaches

2.2 Neighborhood-based Methods

2.2.1 Term Frequency-Inverse Document Frequency (tf-idf)

2.2.1.1 Basic Idea

In this method, we made an analogy between the documents and user purchase histories, and between the terms and the products users purchased. Term frequency is represented by the number of occurrences of an

2.2.2 Word2Vec Similar Product/Similar User Based Model

With the help of Word2Vec we have done these 2 tasks:

1. Finding Similar Users to a User. a. We do this by first making a list of procuts purchased in an order as a numpy se

2. Finding Products similar to a product.

2.3 Latest Factor Methods

2.3.1 Implicit Alternating Least Squares [1]

The key idea of the implicit ALS is to transform a user item matrix of product purchase frequencies to a confidence matrix C_{ui} for users u and items i.

$$C_{ui} = 1 + \alpha R_{ui}$$

α represents a linear scaling of the rating preferences (in our case number of purchases) and R_{ui} is the original matrix of purchases. The paper suggests $\alpha = 40$ to be a good starting point.

Now, similar to other matrix factorization methods, the goal is to find user-factor vectors $x_u \in \mathbb{R}^f$ and item-factor vectors $y_i \in \mathbb{R}^f$ for each user and item. These factors are computed by minimizing the cost function:

$$\min_{x,y} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$

This then leads to an alternating-least-square optimization process, where the algorithm alternates between re-computing user-factors and item-factors with each step guaranteed to lower the value of the cost.

In our case, we are using the library "implicit"[1] to perform ALS optimization.

3 Evaluation

Recommendations for each user are ordered list of products, from the most preferred to the least preferred. The dataset has no feedback about product that are disliked and hence, precisiion based metrics are not appropriate. Instead, recall-oriented measures are applicable since a prior purchase of a product is an indication of liking it.

After training the respective models, we are generating recommendations for each user in the test dataset. We will then compute the mean recall over all the users in test dataset by comparing the actual current purchase of the user and the products recommended

$$\text{Mean Recall} = \sum_{\text{test users}} \frac{|\{\text{recommended}\}| \cap |\{\text{actual}\}|}{|\{\text{actual}\}|}$$

Furthermore, we will be evaluating all the models by comparing them to a baseline model that suggests the most popular items to every user.

4 Next Steps

References

[1] Y.F. Hu, Y. Koren, and C. Volinsky, Collaborative Filtering for Implicit Feedback Datasets, Proc. IEEE Intl Conf. Data Mining (ICDM 08), IEEE CS Press, 2008, pp. 263-272.

[2] A. C. Melissinos and J. Napolitano, *Experiments in Modern Physics*, (Academic Press, New York, 2003).

[1] Fast Python Collaborative Filtering: https://github.com/benfred/implicit

[3] N. Cyr, M. Têtu, and M. Breton, IEEE Trans. Instrum. Meas. **42**, 640 (1993).

[4] *Expected value*, available at `http://en.wikipedia.org/wiki/Expected_value`.