Generating Effective Item Recommendations for Etsy Users

Shirmung Bielefeld, Howard Jing, Karen Li

1. Abstract

Creating an effective item recommendation system for commercial website use is becoming increasingly important. With an effective item recommendation system, commercial websites are better able to incite users to purchase items by presenting them with items that are more relevant to their interests. In our project, we focus on one particular website, Etsy, and the unique difficulties that come with recommending items to users on Etsy. In particular, we will utilize two different methods on our data sets: item-based collaborative filtering using the k-Nearest Neighbor algorithm and the “revealed preference through weighted tags” algorithm, and we will analyze our results accordingly.

2. Introduction

Etsy is a marketplace where users can buy and sell handmade goods. Etsy users are also able to mark items that are for sale as favorites. Each item is represented by a set of tags that have been selected by the seller. This is illustrated in figure 1.

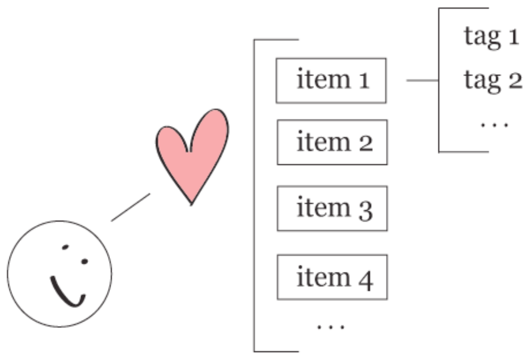

Figure 1

We believe that an item’s set of tags is a relatively accurate representation of that particular item as it is to the seller’s advantage to provide faithful representations of items. We also believe that users will favorite items according to their interests and that a particular user's interests can be discerned by their favorite items' set of tags.

There are many challenges inherent in our project. For example, unlike other websites like Netflix or Yelp, Etsy does not have an item rating system, which means that we are unable to determine what exactly a particular user does *not* like (much work that has been done thus far in creating effective item recommendation systems utilizes this helpful rating system). Furthermore, we are not able to access private user information so we do not have the ability to include additional helpful information such as user clicks or user purchases in our project. Instead, we will attempt to generate effective item recommendations by using item tags alone.

3. Methodology

3.1 Data

We have obtained our data by using Etsy's API. We have focused on generating effective item recommendations for one user and have collected the tags of items that that particular user (which we will now refer to as "corduroy") has marked as a favorite item. The tag counts of corduroy's favorite items are graphically displayed as a word cloud (created using Wordle) in figure 2.

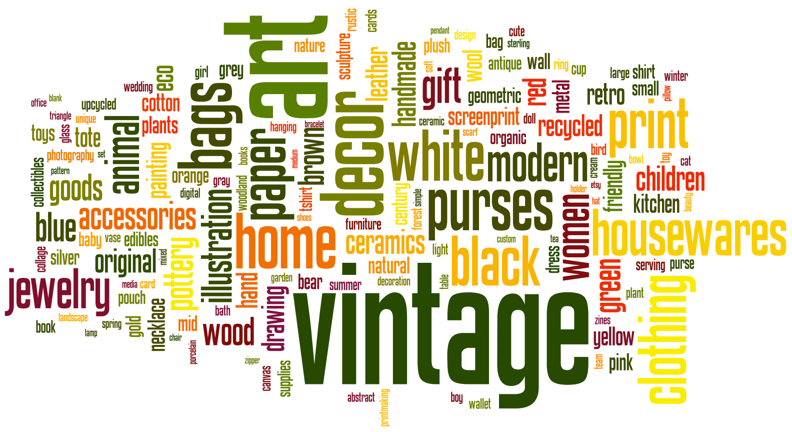

Figure 2

We have also collected the tags of “randomly” selected items for use in section 4.1 and 4.2.

3.2 Experimental Setup

We have “randomized” and partitioned each data set using 4-fold cross-validation. We will use the majority of corduroy’s favorite items as the training set, setting aside a small portion for the testing set. The testing set is additionally made up of our “randomly” selected items. When our recommendations are generated, in an order from the most preferred to the least preferred, it is our hope that the subset of corduroy’s favorite items that was included in our testing set will be clustered towards the top of the recommendation list—that is, corduroy’s favorite items should be recommended to corduroy, since we already know that these items are of interest to her.

4. Implementation and Results

4.1 Item-Based Collaborative Filtering Using the k-Nearest Neighbor

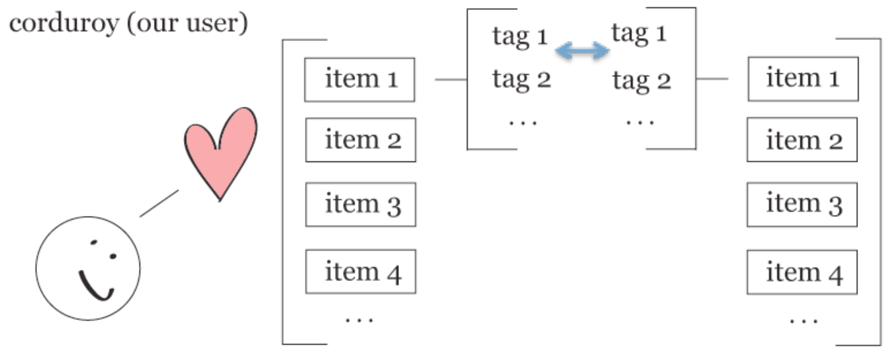

Figure 3

Our setup for item-based collaborative filtering using the k-Nearest Neighbor algorithm in which we will calculate the similarity between corduroy’s item tags and the random items’ tags is illustrated in figure 3. At the heart of our calculations, we will be using cosine similarity to determine the similarity between two vectors. Cosine similarity is defined as follows:

For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$,

$$similarity_{cosine}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||||\mathbf{b}||}$$

We will attempt item-based collaborative filtering using the k-Nearest Neighbor algorithm as follows:

Let the item preference score of our user u (corduroy) for a random item i_r be defined as the following expression where i_u represents one of corduroy's items:

$$iPreference(i_r) = \sum_{i_u \in u\text{'s Items}} similarity(i_r, i_u)$$

where the similarity score is defined as:

$$similarity(i_r, i_u) = \begin{cases} similarity_{cosine}(i_r, i_u), & \text{if } i_r \text{ is a nearest neighbor of } i_u \\ 0, & \text{otherwise} \end{cases}$$

Our results are shown in the following figures. In figure 4, we determine the amount of error by presenting the mean number of corduroy's favorite items within the top n% of the 5972 total items (1079 of corduroy's favorite items + 4893 random items). In figure 5, our results are plotted from left to right, top to bottom (where left, top is the top of the recommendation list). In this plot, the green lines represent corduroy's favorite items and the red lines represent random items. In figure 6, we plot the cumulative distribution of our results.

Top %	**Mean**	**Standard Deviation**
25%	545.4	12.8
50%	853.67	22.67
75%	1002.08	35.97

Figure 4

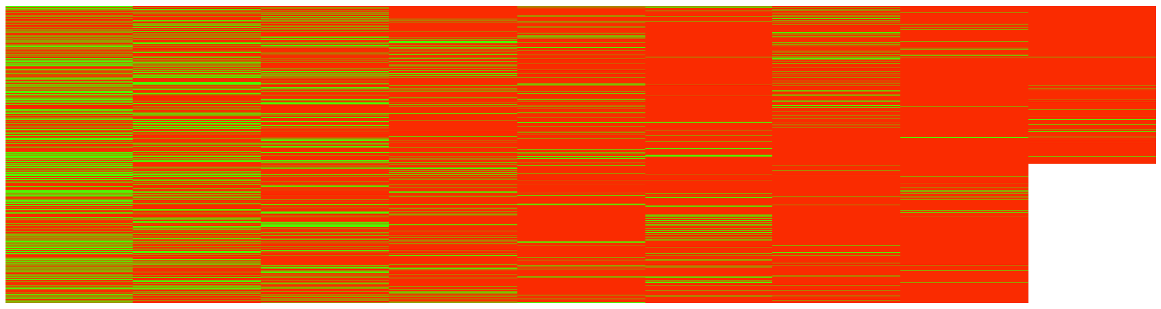

Figure 5

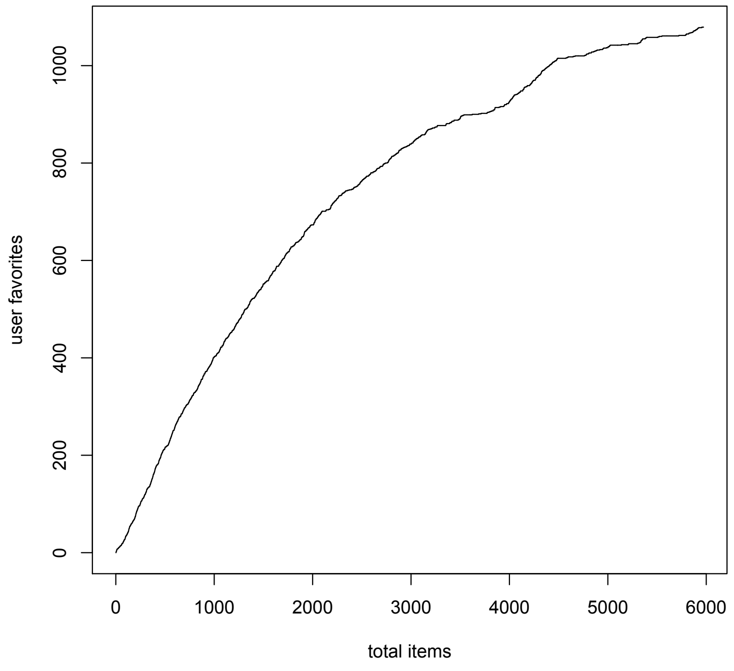

Figure 6

As the above figures show, corduroy's favorite items tended to cluster towards the top of the recommendations list. Additionally, items that were not in corduroy's favorites were more heavily featured near the bottom of the recommendation list.

4.2 Revealed Preference Through Weighted Tags

As an alternate approach to item-based collaborative filtering using the k-Nearest Neighbor algorithm, we developed an algorithm that weighs tags based on their occurrence count within corduroy's favorite items—that is, after constructing a dictionary containing unique tags as keys and their occurrence counts as values, we find the "weight" (or estimated probability) of a tag by dividing its occurrence count by the total number of unique tags. Corduroy's favorite items now form our weight look-up table for evaluating the preference of a particular item.

We first determined preferences on the items within corduroy's favorite items. The algorithm run on one example item (which we will now refer to as "bag") is as follows:

Assume our weight look-up table has a total of 1000 unique tags. Given an item, bag, with tags [green, art, bubbles], we refer to our weight look-up table and find:

Tag	**Tag Count**	**Weight**
"green"	100	100/1000 = 0.10
"bubbles"	25	25/1000 = 0.025
...	...	...
"art"	75	75/1000 = 0.075

We then sum the weights of the tags that represent bag to determine the item's preference score: 0.10 + 0.025 + 0.075 = 0.2. Thus, bag has a preference score of 0.2 and we will compare this value to other items' preference scores and rank items accordingly.

Our results are shown in the following figures as described in section 4.1.

Top %	**Mean**	**Standard Deviation**
25%	516	18.62
50%	853.5	12.72
75%	1005.67	7.83

Figure 7

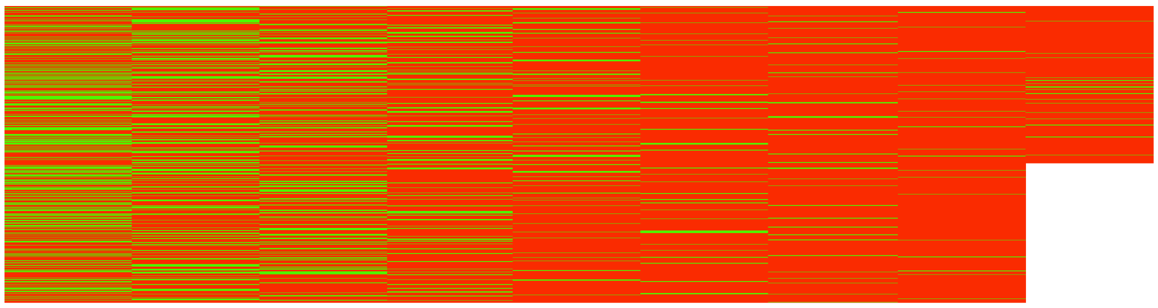

Figure 8

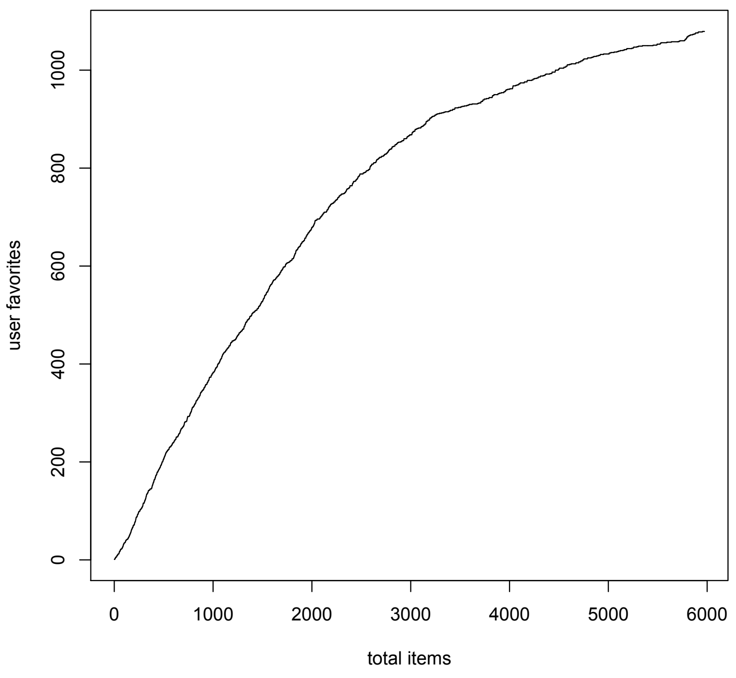

Figure 9

Our results using the “revealed preference through weighted tags” algorithm are similar to the results of the item-based collaborative filtering using the k-Nearest Neighbor algorithm in section 4.1. However, the item-based collaborative filtering using the k-Nearest Neighbor algorithm was the slower algorithm of the two. This is because, rather than comparing every random item to every item in corduroy’s favorite items, we only had to compare each random item to the weight look-up table.

4.3 Effects of Ignoring Top Categories

We noticed many of the random items’ tags were also the top categories provided by Etsy. For example, over 1/5 of the random items had the tag “supplies”. Because of the prevalence of the “supplies” tag, we questioned how effective it was in determining whether or not corduroy would like a random item. As an experiment, we ran our two algorithms again on the same data sets—but excluding tags that were also these top categories—to see how ignoring top categories would affect our results. We hoped that by ignoring the top categories, rarer and more specific tags would be weighted more heavily.

Our results for item-based collaborative filtering using the k-Nearest Neighbor algorithm are shown in figures 10—12 as described in section 4.1. Our results for the “revealed preference through weighted tags” algorithm are shown in figures 13—15 as described in section 4.1.

Top %	**Mean**	**Standard Deviation**
25%	351.5	108.27
50%	493.25	157.18
75%	637.3	207.67

Figure 10

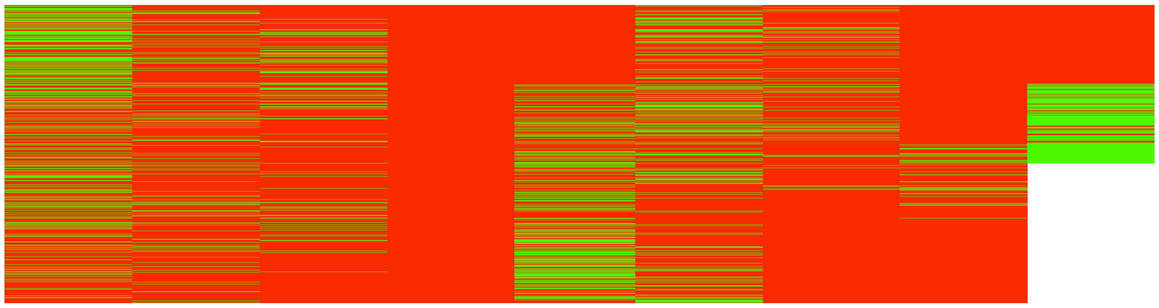

Figure 11

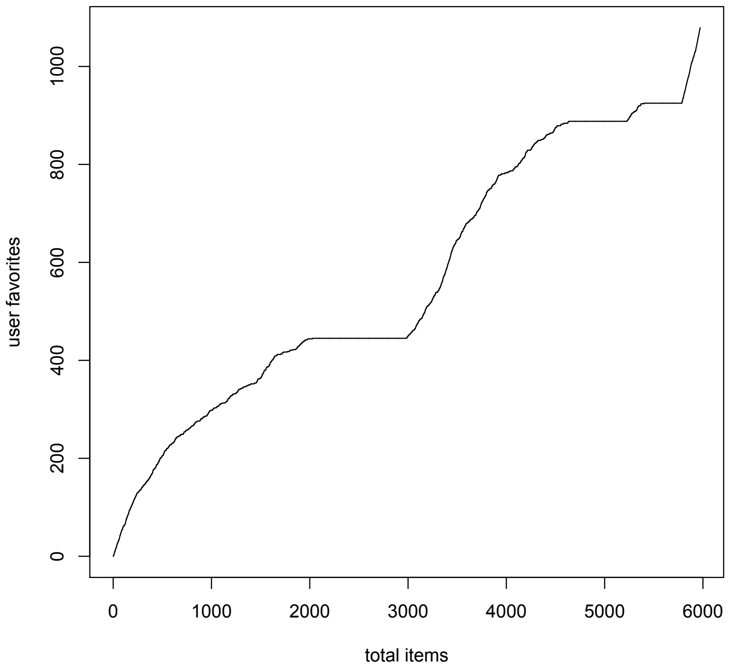

Figure 12

Top %	**Mean**	**Standard Deviation**
25%	565	9.45
50%	828.8	9.85
75%	976.6	10.3

Figure 13

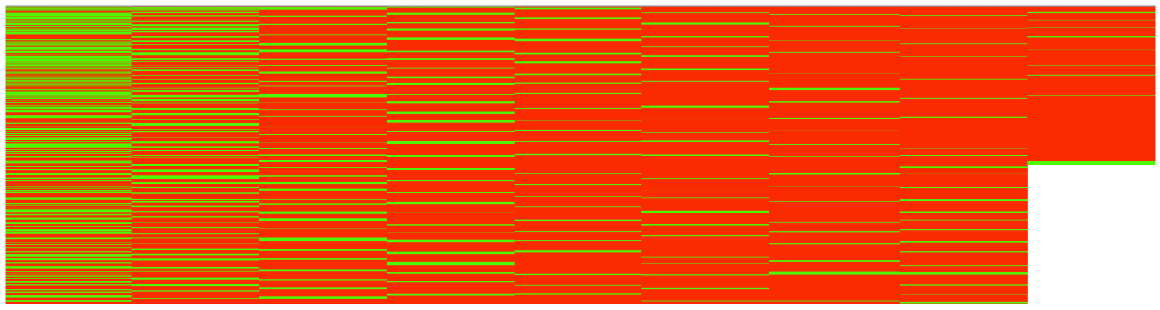

Figure 14

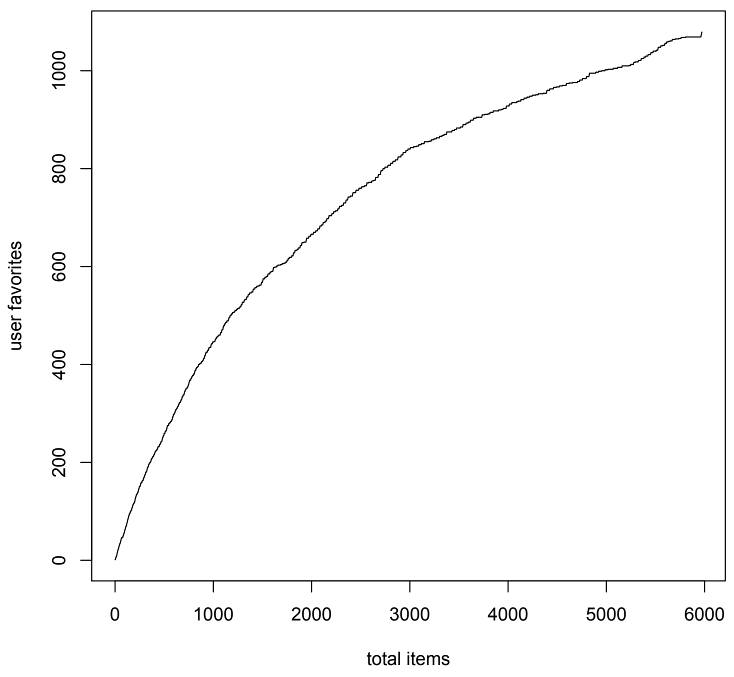

Figure 15

In the case of the “revealed preference through weighted tags” algorithm, our results were slightly better if we ignored top categories. However, with the item-based collaborative filtering using the k-Nearest Neighbor algorithm the results were baffling: each of the results was wildly different as illustrated by the large standard deviation in figure 13. One possible explanation for this is that by excluding tags that were also top categories many of the items had no tags left, therefore there was no way to measure the similarity between two items.

5. Conclusions

Overall we were fairly happy with the results of our project. Corduroy's favorite items that were included in our testing set were, for the most part, clustered towards the top of the recommendation list. Furthermore, the injection of random items towards the top of the recommendation list could be attributed to the fact that some of the random items could be very similar to corduroy's favorite items as a whole. From this recommendation list we could then go on to recommend items above a certain selected threshold that we believe our user will like.

Another way that we could approach generating item recommendations for Etsy users in the future is by finding users that are similar to our particular user and recommending items that these similar users have marked as favorites. By doing this, we could introduce new items to our user that our user had not considered before. An effective item recommendation system should be able to generate items that the user might like outside of the user's schema. For example, if a user *really* likes "blue vintage crocheted dresses" an effective item recommendation system should not *always* show the user blue things or vintage things.

We could also consider grouping tags that are similar to or correlate with each other. For example, we could group tags such as "knitting", "knitted", and "knits", or "hats" and "beanies", or combine certain tags that occur frequently together such as "red" and "dress".

Generating effective item recommendations to Etsy users is a unique problem with several inherent challenges but, in the end, our results seem to produce a relatively accurate recommendation system that could be put to use in the real world.

6. References

- http://public.research.att.com/~volinsky/netflix/cfworkshop.pdf

- http://ilk.uvt.nl/~toine/publications/bogers.2009.recsys2009-workshop.pdf
- http://www.springerlink.com/content/m812ng6155r2v7l6/fulltext.pdf

7. Source Code

- https://github.com/howardjing/MLEtsy