Group A – Dabas, Daly, Madhavaram, Tse
CIS 9650
Project Report

Amazon Recommendation System

Problem to be solved

For this project we wanted to work on a problem that is still being researched and as a team we came to a conclusion to work on a recommendation system. Given the time limit, we decided to work to improve certain aspect of Amazon recommendation system. In our research we came across a feature that Amazon needs to improve. Many times, we have seen that while we are looking at some product on Amazon, it recommends an item which is not related at all. An example can be seen in Figure 1 at the end of the report. Hence, we decided to build a recommendation system to recommend similar items based on a product that a customer is viewing.

Solution

Our solution is a 7 step process explained below.

Step 1: The first step was to decide how to store the data. We considered three options, pandas dataframe, dictionary of dictionaries, list of class objects. After weighing the pros and cons of each option we decided to move ahead with list of class objects due to ease of accessing a property and traversing the items in list. Finally we defined a class in python code file with some required and optional fields.

Step 2: The next step was to extract data from metadata file for further processing. Since our metadata file was in loose json, a simple json.loads(file) did not work so we had to first convert it into a strict json file. This step took a long time as we required to understand the methods used which were not easily understood. **Libraries used: gzip and json**

Step 3: The next step was to extract data from the ratings file. This was a csv file and was easy to work on since we have practiced on csv files in class. The data from this file was saved in a dictionary with product ID (asin) as the key and computed average rating as value. We choosed dictionary as a data structure for this as we needed to access the rating while creating an object for each transaction row. **Library used: csv**

Step 4: In this step, we had to create list of objects by using the data extracted in previous steps. We had to solve quite a few errors as each record didn't have all the fields required to create a class. Hence, we used try and except for error handling.

Step 5: Next, we needed to decide on the algorithm so that we have a better recommendation system. For this we researched about what Amazon and other companies use and found that item-to-item based collaborative filtering algorithm is most popular. This algorithm is also used by Amazon with their customizations since they have customer data too. But our data was

limited as we could not make use of customer preference data. Hence, our system is solely based on transaction data. Our algorithm recommends items that have categories and title similar to the item being viewed.

Step 6: We needed to determine how to implement our algorithm and which statistical models will be suitable for the given problem. Since categories and title are both text, we researched about the text mining techniques that can be used to determine similarity between categories and title of two products. We found that cosine similarity measure has proven to work best on text as it normalizes the length of text and can also be interpreted easily without the need of statistical background. It ranges between 0 and 1 with 0 showing no similarity. However, cosine similarity function works on vectors so we needed to convert the categories and titles into vectors. For that we implemented tf-idf vectorization. In this step we wrote two functions. First function returns a list of items with categories similar to the categories of the item being viewed and in this we do not consider the items already bought. This list is passed to second function which returns a list of items that have similar title compared to the title of product being viewed. Another major challenge was to decide the threshold of cosine similarity value for each function. For this we increased the value from 0.5 to 0.95 with an increase of 0.2 in each iteration and analysed the results. For each category the threshold was required to be changed. Since, our final submission is for Electronics data, threshold values are 0.85 and 0.1 for categories and title respectively. The threshold for categories is higher as the products should belong to the same category while their title can be very different. **Libraries used: collections, random, sklearn**

Step 7: To show our results we created a UI. The biggest problem we faced here was that QT Creator does not support jpg format and most of the product's image URL had jpg image. We tried a lot of methods which makes QT creator compatible for many formats but jpg is still not supported. We solved this by converting images to png. **Libraries used: PyQt4, PIL, urllib, requests, io, sys**

One limitation of our system is that we do not have any dataset to validate our results. Hence, the accuracy of our system can be based on manual analysis of products being recommended at this stage.

Enhancements:

With more time, our system can be improved in many areas.

- Add features to UI to let customer see recommended products according to different criterias like price, relevance, cost, rating.
- Enhance UI to study customer behaviour about a product based on the items clicked and viewed. With this we can include customer preferences in our system for customized recommendation. This will also help get feedback on our system which can be used to test the accuracy of our system in future.

- We can make use of list of items in bought_together field in metadata file to recommend related items.
- With better computing performance of the machine we can avoid latency

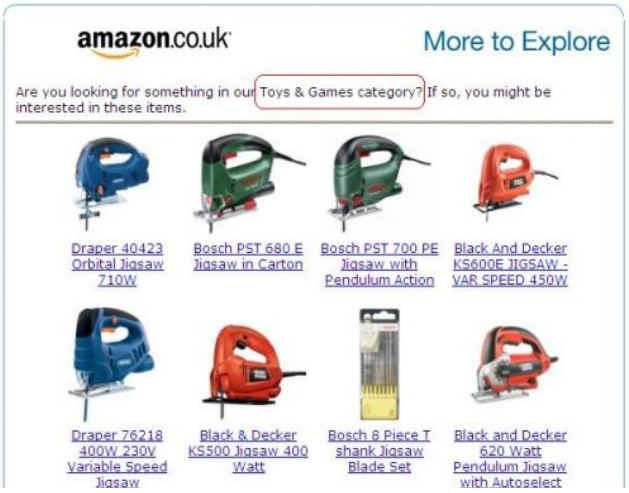

Figure 1: Jigsaw recommended in Toys and Games Category

Citation:

1. Image-based recommendations on styles and substitutes

   J. McAuley, C. Targett, J. Shi, A. van den Hengel

   SIGIR, 2015

   pdf

2. Inferring networks of substitutable and complementary products

   J. McAuley, R. Pandey, J. Leskovec

   Knowledge Discovery and Data Mining, 2015

   pdf