Project Disposition: A recommender system based on customer interaction with an e-commerce website

Big Data Analytics: brownian boys

Richard Hruby (17-619-172) Johan Faxner (21-603-204)
Tim Matheis (21-603-907) Giovanni Magagnin (17-300-914)

Research Question

How well can we predict the purchasing behaviour of an online shopping customer based on their previous shopping behaviour as well as the purchasing behaviour of other customers.

E-commerce is a booming industry leading to tremendous numbers of purchased products each day. Although the customers' online behaviour is often collected and stored, this data is often not leveraged due to its big size or complexity. However, figuring out desired products for online shoppers may add economic value. We aim to determine shoppers' preferences to predict similar products. These products can then be recommended to customers.

Methods/Empirical Strategy

We are planning on implementing a collaborative filtering strategy to answer our research question. Collaborative filtering filters out products that an individual customer may be interested in, based on how similar customers have interacted with the same products.

As a first step, we will create an SQLite database to analyse the data by running queries (such as filtering for the number of unique users). As a second step, we will also build a MySQL database for running queries on AWS. This enables us to run queries on devices with a not too much memory.

The basic data requirement for this method is a set of products and a set of customers who, in some way, have reacted to these products. The reaction may, e.g., be an explicit rating on a scale or an implicit reaction, where the user's preferences are inferred from, e.g., their purchase or viewing history. In our case, we are dealing with data that provides the latter type of information. Explicit preferences would likely be better than implicit ones. However, a collaborative filtering model on a binary variable (purchased/not purchased) is totally fine, and it can be coded as 1 or 0.

When building the recommender engine, we need to identify similar products or similar customers. The former approach would lead to a so-called item-based approach and the latter to a user-based approach to collaborative filtering. I believe that we are planning to go down the user-based route, which means that for every user, we'll have a set of similar users, and we'll be able to predict the rating (purchase decision) of that user by sampling those similar users and checking what their rating of the product tended to be.

We will arrive at a customer-product rating matrix, where each row is a customer, each column is a product, and each specific entry is a rating (in our case, 0 or 1). Since less than 1% of observations involve an actual purchase, we will likely want to engage in dimensionality reduction in order to improve efficiency. For this, we are planning to use matrix factorization. Using some ML algorithm, we extract product features from user actions. These features are not labelled, however, so they'll be termed "latent features". Two smaller matrices will be derived from the one large matrix, and these two matrices can be multiplied together in order to get back to the large matrix.

As for the similarity metric, there are several to choose between, e.g.,

- Pearson
- Euclidean
- Manhattan
- Cosine

A lot of websites recommend using the cosine similarity metric. It is said to be especially applicable when data is sparse, i.e., when we don't have a rating for many of the products. However, if we're interested in a "horse race", one way to do that might be to use different similarity measures. It might be worth noting that cosine similarity is not a proper distance metric, as it doesn't satisfy the triangle inequality property. The Jaccard index is also a strong contender for the similarity measure.

Once we have calculated the similarity of users, we will obtain a matrix that looks like a correlation matrix, but which shows the Jaccard index or cosine similarity values. Both rows and columns are the users/customers.

We need pick the n most similar users to any given user and then calculate the purchase probability of that user based on a average purchase probability of similarly users for the same product. Then we will try to forecast the purchasing behaviour for certain customers and test if they actually buy the products our system recommended.

Issues we might face (how to deal with big data)

- It is likely very computationally expensive to calculate large similarity matrices among millions of customers
  - Potential solutions:
    - Filter users that have not enough meaningful interactions with the website
    - Sample users based on criteria and calculated matrices within groups
    - Use cloud computing or distributed systems to parallelize operations
- For every prediction, we should theoretically recalculate the similarities (after every interaction technically)
  - Potential solutions:
    - We could assume that preferences and therefore similarities are somewhat stable in the short term, therefore we could recalculate e.g. every month only.
- RAM issues will likely be very significant for our large matrices
  - Potential solution:
    - Leverage cloud computing
    - Use dimensionality reduction to compress the matrices

Data Source(s)

E-commerce store dataset: We have 2 data sets, 2019-Nov (around 9gb) and 2019-Oct (around 6gb). The files contain behaviour data for 7 months (October to April 2020) from a multi-category online store. The data sets contain the following columns:

- Event time: complete data, it shows the date and time of the event.
- Event type: complete data, it could be only one among these options (view, cart, remove, purchase).
- Product id: complete data.
- Category id: complete data.
- Category code: some missing data points.
- Brand: some missing data points.
- Price: complete data.
- User id: complete data.
- User session: complete data.