Recommender Systems - Assignment

Sihem Amer-Yahia, Idir Benouaret

April 20, 2019

Recommendation systems are important filtering methods that are designed to predict whether a user will like an item or not. An item could be a movie, a product, a song ,etc.

1 Total Dataset

In this assignment, you will use a dataset provided by TOTAL. The dataset represents customers purchasing products at gas stations that are geographically distributed in France. Examples of such products are: car washing, gas, drinks, food, etc. Each time a customer visits a gas station a transaction is recorded with the information about which products the customer bought, at what time of the day and at which station. Thus, each line of the dataset corresponds to the following attributes:

- Transaction identifier : TRANSACTION_ID
- Customer Identifier: CUST_ID
- Product Identifier: ARTICLE_ID
- Station identifier: STATION_ID
- Time of the day: PERIOD (0 for morning, 1 for afternoon, 2 for evening)

You will use collaborative filtering algorithms to provide our customers with relevant recommendations.

2 Matrix Factorization

Matrix factorization is among widely used techniques for collaborative filtering. The common approach in matrix factorization models is to factorize the user-item rating matrix into the product of two lower dimensionality matrices. One matrix can be seen as the user matrix where rows represent users and columns are latent factors. The other matrix is the item matrix where rows are latent factors and columns represent items.

In many real-world applications, it is common to only have access to *implicit feedback* (such as: view, click, purchase). The approach used in **spark.mllib** to deal with such implicit data is taken from **Collaborative Filtering for Implicit Feedback Datasets**, by Yehuda Koren *et. al.* Essentially, instead of trying to model the matrix of ratings directly, this approach treats the data as numbers representing the strength in observations of user actions. In particular, in our dataset, you will use **the number of times a customer purchased a product as the implicit user preference signal.**

2.1 Dataset pre-processed for you

We provide you with a transformation of the dataset to get it ready to work with Spark Mllib ALS Implicit. Each line of the dataset is a triplet (user, product, Number of purchases). Please download this dataset and save it in your working directory.

2.2 Training the implicit ALS model

For this exercise, you will use a method of the Matrix Factorization family called **ALS**, which is available in **Spark** through the MLlib library.

You will first need to load the dataset into the RDD ratings format. The tuples in this RDD are instances of: Spark.mllib.recommendation.Rating(user: Int, product: Int, rating: Double).

As you noticed, the RDD ratings format takes as input Integer for users and products identifiers. Thus, **you will need to transform the dataset. Both customer and product identifiers have to be mapped to integers**.

After this step, to train a model you will you ALS.trainImplicit, you will set the following parameters:

- *rank*: is the number of latent factors in the model
- *iterations*: is the number of iterations of ALS to run
- *lambda*: specifies the regularization parameter in ALS
- *alpha*: specifies the confidence in implicit preference observations

An example for running such a model:
model = ALS.trainImplicit(dataset, rank=20, numIterations =30, lambda=0.01, alpha =40)

2.3 Recommendations

Once the model is trained, you can now recommend to a customer a list of products with the function provided in Spark Mllib:
recommendProducts(user, N): recommends the top "*N*" products to a given user and returns a list of Rating objects sorted by the predicted rating in descending order.
Try to produce recommendations lists for different customers, and for different values of N.
For this exercise, we provide you with a code example, that you find in the following link.
Save your work (all steps and results) in a folder as name.surname.als

3 Item-based Collaborative Filtering

The item-based collaborative filtering approach relies on estimating the preference of a user for a target product according to the similarity between the target product and the products previously purchased by the user.

1. First you will need to transform the dataset into binary values, 1 if a customer purchased a product and 0 if not.
2. Then, represent the data as a purchase matrix P, where each row corresponds to a customer, and each column to a product. Each entry in the purchase matrix is a Boolean value (1 for purchased, 0 for the opposite)

We provided you the code that will help you constructing the purchase matrix. You will find the code in this link.

The item-based approach computes a similarity matrix between products. The similarity between two products k and l is estimated using the cosine between their corresponding columns in the purchase matrix:

$$Sim(k,l) = \frac{P_{.k}.P_{.l}}{\sqrt{P_{.k}^T P_{.k}}.\sqrt{P_{.l}^T P_{.l}}}$$

3. Compute the Similarity Matrix S between products

4. The recommendation list for a customer is obtained as follows:

   - For each product k purchased by the user, select the list of N products that are the most similar to k according to the similarity matrix.
   - Merge the lists, and sort the products by decreasing similarity score. If a product is in several lists, assign it its maximum score.
   - Recommend N products from the top of the list.

Save your work (all steps and results) in a folder as name.surname.ItemCF

4 Contextual recommendations

So far, we did not use the information about the time period, and the stations where the purchases occurred.

An important aspect in recommendations is the incorporation of contextual factors into the recommendation process. A customer may have different purchase preferences according to the period she visits a station: she might buy a coffee if it is morning, but maybe not if it is evening.

4.1 Contextual pre-filtering

In this assignment, you will revisit the two approaches you tested before by taking into account the contextual parameters. First you will download the full dataset. Then, you will filter your dataset according to contextual information, after that you will run your algorithms on the filtered datasets (this approach is referred to as pre-filtering).

1. Period of the day:
    - Construct a projection of three different datasets according to purchases that occurred during morning, afternoon and evening.
    - Apply the algorithms in these three different contexts.
    - Are the results different within each context?
2. Station:
    - Pick 10 stations at random
    - For each station, construct a projection of the dataset according only to purchases that occurred in that station.
    - Apply your algorithms at each station.
    - Are the results different?

Save your work (all steps and results) in a folder as name.surname.Context

5 Send you assignment

Create a Folder name.surname containing the three folder corresponding to each exercise. Zip this file and send it as a google drive link.