IITM-CS5691 : Pattern Recognition and Machine Learning
New Method Writeup

Release Date: 29th Apr, 2021
Due Date: 28th May, 2021

Roll No: CS18B005

Name: Soumith Basina

- Dear Student, You may have tried or thought of trying different methods for your data contest. Choose one of the methods that was not taught in class, and submit a writeup of this "new method" in the template provided below. This is an individual submission - i.e., while you would've done your kaggle submission as a team of two members or while you may've discussed this method with your teammate, **you will have to write about the new method in your own words independently and submit it individually**.
- **Template:** Fill in whatever fields are applicable for your algorithm (overall 1-2 page writeup; since some fields may not be applicable for certain methods, we haven't shown points below).

1. (points) [Name of Method, and its ML Problem and Paradigm: problem could be regression/classification/clustering/etc., and paradigm could be supervised/unsupervised/..., generative/discriminative/direct, linear/non-linear models, etc.)]:

Solution:

Name of Method: Item-Item Collaborative Filtering

ML Problem: Regression, Clustering

Paradigm: Unsupervised

2. (points) [Brief introduction/motivation: One paragraph to describe briefly the new method (its name, what it does, its main application, etc.)]

Solution:

Collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc. In the context of the data contest, it is the method of making automatic predictions about the interests of user by collecting preferences from many users. The assumption of this method is that a person with a similar opinion as the current user has more significance in predictions than a randomly picked user or similarly a song that is similar to the current song has more significance in predictions.

Collaborative filtering is often used in recommender systems, to recommend them movies/-songs similar to the ones they have enjoyed/gave favourable ratings to.

3. (points) [Closely related method seen in class, and relation of your selected new method to method you eventually used for the data contest]:

Solution:

Item-Item Collaborative Filtering has some similarities to K Nearest Neighbours method to determine the most similar songs. We considered k songs with most similarities to make predictions, similar to how we take k nearest neighbours to predict the class of the data point.

It also has similarities to linear regression, we learn similarities (basis) from the data and we use a dot product of similarities (w_i) and ratings (x_i) to predict the scores.

4. (points) [Training Input and Output: (e.g., $\{x_i, y_i\}_{\{i=1\ldots N\}}, x_i \in R^d, y_i \in \{-1, +1\}$, etc.]:

Solution:

The input for the training step is the pivot table of the training dataset of the shape (# of customer_id) $\times$ (# of song_id).

The output is a (# of song_id) $\times$ (# of song_id) matrix of item-item similarities obtained by calculating Pearson Correlations.

5. (points) [Training Objective function (e.g., loss function that is to be optimized) or probabilistic model (over which MLE or Bayesian inference done):]

Solution:

The loss function that is optimised is the RMSE value over the test dataset. It is optimised by tweaking the similarity value.

6. (points) [Training Algorithm: Brief description of key aspects of the algorithm]

Solution:

We first calculate the similarity matrix between the songs. For the similarity metric, we calculate Pearson correlations between all possible pairs of score vectors of songs. Pearson correlation coefficient is a measure of linear correlation between 2 sets of data.

Pearson Correlations can be calculated by calculating the cosine similarities in the normalised pivot table.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}$$

7. (points) [Testing Input and Output: (e.g., $x \in R^d$, $y \in \{-1, +1\}$)]

Solution:

The input for the testing step is the matrix of item-item similarities and the cut-off threshold of the similarity value.

The output is a (# of customer_id) $\times$ (# of song_id) matrix with the predictions of all users and songs.

8. (points) [Testing Algorithm: Brief description of key aspects of the algorithm]

Solution:

To predict the score of a user-song pair, we calculate the weighted mean of the ratings of the songs whose similarities are above the threshold.

$$r_{xi} = \frac{\sum_{j \in N} r_{xj} s_{ji}}{\sum_{j \in N} s_{ji}}$$

where, x, i are the user, song that are being predicted
N is the set of all the songs whose similarity is above the threshold

We then test it on the test set and tweak the threshold accordingly to minimise the RMSE.

An efficient way to predict the scores is to vectorise this formula. The numerator can be calculated by the cross-product of the pivot table (with missing values as 0) and the similarity matrix. The denominator is the numerator but the scores in the pivot table are 1 wherever it

is non-zero. Dividing both matrices element-wise gives the complete set of predictions from which the required predictions can be extracted.

9. (points) [Critique of the method: (1-2 paragraphs discussing its strengths and weaknesses in your own words)]

Solution:

The advantages are

- The results of this method are very explainable, as it is the mean of the similar songs weighted according to their similarity, which is important in recommender systems.
- Domain knowledge is unnecessary because the latent factors are automatically learned.
- Creation and use of the model is relatively simple.
- Content-independence from the determined similar items as the data used to train the model is only the scores.

The disadvantages are

- As many recommender systems are based on large datasets, the user-item matrix will be very large and sparse. This can have many problems such as the cold start problem, a user need to rate sufficient number of items before the system can provide accurate recommendations.
- Collaborative filtering is not scalable as number of users and items grow, even if it is $O(N)$. The response for recommendations also should be very quick, which demands further scalability.