IITM-CS5691 : Pattern Recognition and Machine Learning | Release Date: 29th Apr, 2021
New Method Writeup | Due Date: 28th May, 2021

Roll No: EE18B067 | Name: Abhishek Sekar

- 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: K-NN based Item-Item Collaborative Filtering with baseline correction.
   ML Problem: Collaborative Filtering is an *information filtering* problem.
   ML Paradigm: It is an Unsupervised Learning problem that directly produces the prediction.

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

   Solution:
   " A man is known by the company he keeps" - Aesop

   As the name suggests and in accordance with the above quote, Collaborative Filtering (C.F) filters (predicts) items to cater to a user based on the information present about the items liked by other users with similar taste (collaboration). Item by Item C.F places the onus on items and recommends an item to a user based on the information on how similar items were received by him/her. Its main application is unsurprisingly for recommendation systems given that it was proposed by researchers at Amazon.

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:
Closely related method seen in class:
A method that is somewhat similar to the one implemented by us is the K-NN Density Estimation discussed in class. The rating of a song to be given by a user is estimated by taking into the account the ratings given by the user to 'K' other songs most similar(with respect to the similarity metric) to the required one.
Method used for data contest:
We used an ensemble algorithm combining this new algorithm with K=20 and an lgbm regressor.

$$\text{predicted score} = (\lambda)\text{lgbm predictions} + (1-\lambda)\text{CF predictions} \qquad (\lambda \in [0,1]) \qquad (1)$$

We used $\lambda = 0.40$ for the contest and it resulted in an MSE of 0.70153

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:
Training Input:
$\{x^{(i)}, y^{(i)}, \text{score}\left(x^{(i)}, y^{(i)}\right)\}_{\{i=1\ldots N\}}$
Where,

- $x^{(i)}$ is the customer id of the i^{th} row in the training data.
- $y^{(i)}$ is the song id of the i^{th} row in the training data.
- score $\left(x^{(i)}, y^{(i)}\right)$ is the rating given by customer with customer id $x^{(i)}$ to the song with song id $y^{(i)}$.

Training Output:

- *Similarity* matrix S, where S_{ij} denotes the similarity between songs with song ids 'i+1' and 'j+1'.
- *Scoring* matrix A, which is basically a matrix visualization of the training data with its rows being the songs and the columns being the customers and its element A_{ij} being the rating given to song inhabiting i^{th} row by the customer inhabiting the j^{th} column.
- *Baseline score* matrix B, where element B_{ij} incorporates the individuality of the song inhabiting the i^{th} row and the customer inhabiting the j^{th} column.

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:
Not applicable for the algorithm.

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

Solution:
Rough pseudocode:

for k in N:
 i = $y^{(k)}$
 j = $x^{(k)}$
 A[i][j] = score $\left(x^{(i)}, y^{(i)}\right)$

for i in N_{songs}:
 for j in N_{songs}:
  S[i][j] = pearson similarity(A[i,:],A[j,:])

for i in N_{songs}:
 for j in N_{user}:
  B[i][j] = - A.mean( ) + A[i,:].mean( ) + A[:, j].mean( )

Description:

- The time complexity of the training algorithm is $O\left(N_{songs}^2 N_{user} + N\right)$ where N_{songs} and N_{users} are the number of distinct songs and distinct customers.
- The training algorithm is heavy on computation and took 3 hours to run.
- The computation of S matrix has been optimized by taking into account its symmetric nature, i.e, S[i][j] = S[j][i]. Also S[i][i] = 1. This means, we can just compute less than half the values.
- Pearson similarity(**x**,**y**) is defined as $\frac{\left\langle (\mathbf{x}-\bar{x}),(\mathbf{y}-\bar{y}) \right\rangle}{\|\mathbf{x}-\bar{x}\|\|\mathbf{y}-\bar{y}\|}$ where $\bar{x}$ represents the mean across elements of x. This similarity metric is chosen since it is agnostic to the stinginess of the customer when it comes to ratings,i.e, some customers may rate generously while others may not. Subtracting the mean before taking the inner product removes this issue.

- The element of the baseline matrix, B_{ij} incorporates the deviation from a customer's average rating and songs average rating. This sheds light on whether a customer is a stingy or generous rater and how he/she perceives a song that is above or below average.

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

Solution:
Testing Input:

- $\{x^{(i)}, y^{(i)}\}_{\{i=1...N\}}$
  Where,
  - $x^{(i)}$ is the customer id of the i^{th} row in the test dataset.
  - $y^{(i)}$ is the song id of the i^{th} row in the test dataset.
- Baseline score matrix B after training, Similarity matrix S and Scoring matrix A.

Testing output:
$\{\text{score}\left(x^{(i)}, y^{(i)}\right)\}_{\{i=1...N\}} \in (0, 5]$ which is the array of predicted ratings such that the i^{th} rating is what the customer with customer id $x^{(i)}$ gives to the song with song id $y^{(i)}$

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

Solution:
Rough pseudo code:
for n in N:
Find K most similar songs to $y^{(n)}$ listened by $x^{(n)}$

$$\text{score}\left(x^{(i)}, y^{(i)}\right) = B\left[y^{(n)}\right]\left[x^{(n)}\right] + \frac{\sum_{k=1}^{K}\left(S\left[y^{(n)}\right]\left[y_k^{(n)}\right]\right)\left(A\left[y_k^{(n)}\right]\left[x^{(n)}\right] - B\left[y_k^{(n)}\right]\left[x^{(n)}\right]\right)}{\sum_{k=1}^{K}\left(S\left[y^{(n)}\right]\left[y_k^{(n)}\right]\right)}$$

Description:

- score $\left(x^{(i)}, y^{(i)}\right)$ is predicted as the weighted average of ratings given by user $x^{(i)}$ to K other similar songs $\{y_k^{(i)}\}_{\{k=1,,K\}}$. The baseline correction introduced takes into account the deviation between the actual rating of the song and how the user would've perceived the song as per the baseline matrix. This is then further added to how the user would perceive the current song as per the baseline matrix.

- To optimize the algorithm, the rows of a matrix, $S_{indices}$ stores the songs which are most similar to the song represented by the row. This is done beforehand to speed up the algorithm so that finding the K closest songs rated by the customer doesn't involve parsing through the entire row.
- Lastly, a score clipping is performed wherein,if the predicted score is within 0.05 of an integer, it is approximated to the integer, i.e, if 3.95 is the predicted score, it is approximated to 4. This approximation decreases the MSE by about 0.7 percent and the baseline correction decreases the MSE by about 4 percent when compared to a conventional C.F algorithm.
- The time complexity of the testing algorithm is $O\left(N_{songs}^2 \log\left(N_{songs}\right) + KN\right)$ and it takes less than an hour to run even for high K values.

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

Solution:
Strengths:
Being unsupervised, collaborative filtering is a very convenient algorithm to use when we don't know much about the underlying features present in the dataset which was the case in the Data Contest.Item-Item C.F is appropriate for the problem statement since items are less confusing than users. They're confined to genres while users may have peculiar tastes. This renders item-item similarity to be more meaningful than user-user.The baseline correction incorporates the generosity of a rater and the quality of the song and this makes the prediction more refined.
Weaknesses:
A vast amount of customers are required to have rated for the algorithm to make predictions.This is called the cold start problem. Even when there are enough customers, not all customers would rate a lot of songs. This is the data sparsity problem which we overcame by using an ensemble model.C.F can't make predictions towards a new user and also tends to recommend popular items.