Project No 4: Item-based Filtering Enhanced by SVD

Serdar Biçici
Artificial Intelligence and Data Engineering Department
Faculty of Computer and Informatics Engineering
İstanbul Technical University
150210331
bicici21@itu.edu.tr

***Abstract*—The paper implementation done here is an excellent example of Item-based filtering and the use of linear algebra principles for recommender systems. Singular Value Decomposition (SVD) is a matrix factorization technique that decomposes a matrix into three matrices: a left orthogonal matrix containing the basis vectors for the rows, a diagonal matrix containing singular values representing the importance of each basis vector, and a right orthogonal matrix containing the basis vectors for the columns. It is widely used for ity reduction and data compression across various fields such as image compression and machine learning. SVD allows us to reveal the underlying structure of the original matrix and extract essential features, making it a powerful tool for efficient data analysis and modelling. In our case, we use SVD for dimensionality reduction and for keeping the essential features of the original data matrix for identifying similar features between rows and columns to generate predictions and recommendations for users.**

***Index Terms*—SVD, recommendation, movies, users, rating**

I. SVD Application

In the paper, SVD combined with neighbourhood formation (adjusted cosine similarity) is being used to identify similarities in the user-item matrix and generate recommendations. However, data cleaning and pre-processing are important for the true implementation of the paper. Not using built-in libraries for linear algebra calculations increases the processing cost and writing more optimized code is becoming more important. Data cleaning and pre-processing are really important for this particular implementation. Only taking account of reviewers that have entered more than 20 reviews is an important decision to minimize the dataset. Also dropping columns with no entries and filling out the user-item matrix with row and column average operations makes the data matrix normalized.

II. Dataset

The dataset is an open-source Imdb dataset that includes basic information about the user, date-time and review. Here is an example:

```
#{'review_id': 'rw1133942',
#'reviewer': 'OriginalMovieBuff21',
#'movie': 'Kill Bill: Vol. 2 (2004)',
#'rating': '8',
#'review_summary': 'Good follow up that answers all
    the questions',
#'review_date': '24 July 2005',
#'spoiler_tag': 0,
#'review_detail': "After seeing Tarantino's Kill
    Bill Vol: 1... ... *** B",
#'helpful': ['0', '1']}
```

However, while inserting rating data into the matrix, the rating must be divided by 2 in order to be at the same level as the original paper data. In our dataset, the ratings can be a maximum of 10, unlike the paper's dataset whose maximum rating can be 5.

III. Methods

A. Computing SVD

For computing SVD without built-in features, the Power Method is used for this particular implementation. The Power Method is a method that can find the dominant eigenvalue and eigenvector pair. However, for calculating the SVD, just one pair is not enough. For this purpose, the deflating technique is used. Deflating a matrix involves subtracting the influence of its dominant eigenvectors and eigenvalues. These eigenvectors represent the most significant tendencies in the data and their corresponding eigenvalues indicate their importance. By removing these dominant components, we can reveal other eigenvectors and eigenvalues that capture additional patterns and relationships within the data. In addition, in order to reduce processing costs, GPU-based matrix calculations are used in the implementation. In running the tests for this particular SVD, the GPU calculations' runtime was 16 minutes while the CPU calculations' runtime was estimated as six hours.

Power method:

$$x_0 = \begin{bmatrix} z \\ y \end{bmatrix}$$

$$x_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} z \\ y \end{bmatrix}$$

$$x_{n+1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} x_n$$

Therefore,

$$\lambda = \frac{Ax_n \cdot x_n}{x_n \cdot x_n},\ v = x_n$$

Deflating Technique:

$$\mathbf{A}_{n+1} = \mathbf{A_n} - \lambda_1 \mathbf{v}_{n+1} \mathbf{v}_{n+1}^T$$

B. Neighborhood Formation

After taking the k number of pseudo-users and having done the dimensionality reduction step, we can calculate the adjusted cosine similarity for each pair of the items (movies). The matrix will be a symmetric matrix with diagonals being 1 means an item is most similar to itself. For calculating this matrix, we can avoid recalculation of the same pairs and only calculate iteratively from the first index. In the end, the matrix may look like this:

$$\begin{bmatrix} 1 & 0.423 & 2.310 & \\ 0.423 & 1 & 0.987 & \dots \\ 2.310 & 0.987 & 1 & \\ & \vdots & & \ddots \end{bmatrix}$$

C. Prediction Generation

For prediction generation, we use a weighted sum formula to determine the best option for the user based on an item. This method can be applied to every item that the user rated originally and the top n returns may be presented in applications.

$$pr_{aj} = \frac{\sum_{k=1}^{l} \text{sim}_{jk} * (rr_{ak} + \overline{r_a})}{\sum_{k=1}^{l} |\text{sim}_{jk}|}$$

IV. Experiments

In the paper, it is stated that $k = 6$ and $n = 10$ are the optimal parameters for the most accurate predictions. For this particular implementation, several experiments were conducted to replicate the result.

A. Optimal k Value

In order to deduce the optimal k value, predictions were generated for each combination of k and n pairs and corresponding recommended-item points' averages were summed. When visualized the result can be shown as this: As can be deducted, $k = 6$ gives the least of the average recommended-item points and can be admissible as the most selective k value.

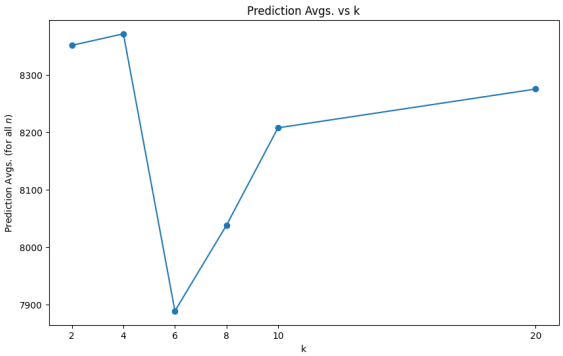

Fig. 1. k value avgs.(k=6)

B. Optimal n Value

For deducing the optimal n value, the newly accepted k value will be taken as a parameter. For $k = 6$, again we get averages of different n values and the least greater one will be more selective.

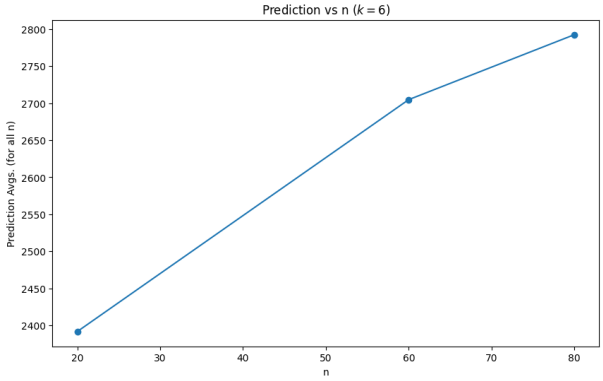

Fig. 2. n value avgs.(k=6)

References

[1] Vozalis, Manolis Margaritis, Konstantinos G.. (2005). Applying SVD on item-based filtering. 464- 469. 10.1109/ISDA.2005.25.