**A Report on Movie Recommendation System**

**Web Search and Information Retrieval**

**Bhavik Patel**

1. **User Based Collaborative Filtering Algorithms**
   1. **Vector Similarity and Pearson Correlation**
      1. **Vector Similarity Method.**

Vector Similarity between two documents is the similarity measured by treating each document as a vector of word frequencies and computing the cosine of the angle formed by the two frequency vectors [Salton and McGill 1983].My experience with the vector similarity was straightforward I didn’t need to implement any tricks to get below mentioned MAE. So, I can say that vector similarity is the simplest technique we can apply for recommendation system. But, it doesn’t perform well because it doesn’t account the user’s rating attitude in the calculation of the cosine angle.

**MAE: 0.79559021507142**

* + 1. **Pearson Correlation**

Pearson correlation takes user’s rating attitude into calculation of similarity between two users. It is not as straightforward as vector similarity because we need to deal with some special cases as we are subtracting user average from his rating. There may be the situations where this subtraction can result in zero especially when every element of that user are same so it makes average and elements same. So, the similarity in this case becomes zero. To handle this case I have put a check to eliminate this case and find the similarity based on basic similarity in this type of cases.

**MAE: 0.786570349696273**

* 1. **Extension to Basic Pearson Correlation with IUF and Case Amplification.**

Basic Pearson correlation weighs all movies equally whether it is universally liked or disliked. It seems obvious that we should give more weight to the universally disliked movies because it tells more about user attitude than universally liked movie. So, I have applied this concept to get more accuracy in ratings by calculating IUF during ratings matrix initialization and using IUF of particular movie to multiply ratings while calculating similarity between two users.

Case amplification widens the gaps between the lower similarity and higher similarity so the prediction can move to its accurate digits.

**MAE: 0.75804465604991**

1. **Item Based Collaborative Filtering Algorithm**

In Item based approach we are finding the similarity between the items unlike with users in user based approach. I found this algorithm tricky than any other algorithm because I was getting negative similarity and that makes the prediction negative. So, It was hard for me to make the algorithm work in proper way. Moreover, it didn’t perform better that user based approach. In my opinion, the reason behind its poor performance is data sparseness in the training data and there were no enough rating data for the active user to select the k nearest neighbor.

**MAE: 0.894885897225415**

1. **Discussions**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **MAE** | **Accuracy** |
| Vector Similarity | 0.79559021507142 | Good |
| Pearson Correlation | 0.786570349696273 | Good |
| Pearson Correlation with IUF and Case Amplification | 0.764119192250862 | Best |
| Item Based | 0.894885897225415 | OK.(Due to data Sparseness) |
| Modified Pearson(My Algorithm) | 0.75804465604991 | Best |

* 1. **Efficiency of Algorithms**

**User based** algorithms (Vector Similarity, Pearson Correlation with IUF and Case Amplification) take too much time to compute prediction compared to item based collaborative filtering algorithm because in user based algorithm K nearest depends upon the current user rating. If user rates some new items then nearest neighbor might get changed. So, every time we predict for active user we need to find its current nearest neighbors. For E-commerce website this approach is time consuming since it does not provide real time recommendations.

**Item based** algorithms can recommend in real time as the K similar items always remain same. So, we can compute the similarity between every possible pair of items and can store them into the matrix in O (n2) time and space complexity, where n is the total number of items in the catalogue. Later during prediction phase we can access them in O (1) constant time. For commercial use, item based approach outperforms the user based collaborative filtering algorithm [Sarwar et al. 2001].

**3.2 Advantages and Disadvantages**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Advantages** | **Disadvantages** |
| **Basic Cosine Similarity** | * Simple to implement. | * Doesn’t account the user rating attitude * Does not fit for real time recommendation. * Takes too much time. |
| **Pearson Correlation** | * Accounts user behavior in similarity calculation | * If average and elements are same then it gives similarity zero. * Does not fit for real time recommendation. * Takes too much time. |
| **Pearson Correlation +IUF+CA** | * Accounts user behavior in similarity calculation. * Weighs rating of globally disliked movies for similarity calculations * Widens the gap between similarities | * Does not fit for real time recommendation. * Takes too much time. |
| **Item based** | * Fit for the commercial recommendations. | * It is hard to deal with negative similarity and prediction * Data Sparseness can be the issue. |