CSEN 272 Web Search and Information Retrieval

Project II

Movie Recommendation System

This project aims to develop collaborative filtering (CF) algorithms to predict movie ratings for users. The goal is to compare the performance of user-based, item-based, neural, and ensemble methods using Mean Absolute Error (MAE) as the evaluation metric. The training data includes 200 users and 1000 movies, while the test data contains 100 users with varying numbers of known ratings (5, 10, 20).

Algorithms Implemented:

1. User-Based Collaborative Filtering (UBCF) with Cosine Similarity
2. User-Based Collaborative Filtering (UBCF) with Pearson Correlation
3. User-Based Collaborative Filtering (UBCF) with Pearson Correlation (Inverse User Frequency)
4. User-Based Collaborative Filtering (UBCF) with Pearson Correlation (Case Amplification)
5. User-Based Collaborative Filtering (UBCF) with Pearson Correlation (Inverse User Frequency + Case Amplification)
6. Item-Based CF with Adjusted Cosine Similarity
7. Neural Collaborative Filtering (NCF)
8. Ensemble Method Weighted Average
9. Ensemble Method Hybrid (NCF + Pearson Amplified for test5/test10; NCF + IBCF for test20)

Results (MAE Scores)

Algorithm	**MAE (Given 5)**	**MAE (Given 10)**	**MAE (Given 20)**	**Overall MAE**
UBCF Cosine	0.854	0.765	0.779	0.800
UBCF Pearson	0.906	0.833	0.828	0.855
UBCF Pearson IUF	0.907	0.833	0.829	0.855
UBCF Pearson Amplified	0.847	0.780	0.774	0.799
UBCF Pearson IUF Amplified	0.854	0.794	0.784	0.809
IBCF Adjusted Cosine	0.921	0.827	0.769	0.833
Ensemble of All Methods Above	0.837	0.761	0.762	0.786

Ensemble (Pearson Amplified and Pearson IUF Amplified)	0.837	0.761	0.753	0.7831
NCF	**0.816**	**0.782**	**0.761**	**0.784**
Ensemble (NCF, Pearson Amplified and Pearson IUF Amplified)	0.825	0.765	0.756	0.781
Ensemble (NCF and Pearson Amplified)	0.811	0.775	0.758	0.780
Ensemble Hybrid	0.812	0.775	**0.747**	**0.775**

NCF achieved the lowest MAE due to its ability to capture non-linear user-item interactions. The normal ensemble method combines predictions from multiple models using weights inversely proportional to their MAE. The hybrid ensemble (combining NCF, Pearson Amplified, and IBCF) outperformed individual models by leveraging diverse strengths. Case amplification improved Pearson's performance by emphasizing strong correlations. Item Based CF performed better with more data (Given 20), likely due to stable item similarities.

The best results came from the hybrid ensemble and NCF, demonstrating the value of combining model strengths and leveraging deep learning. User-based methods suffered from sparsity, while item-based and NCF handled it better. Future work could explore hybrid models with matrix factorization.

Code Instructions:

- **UserBasedCFAllMethods.py** : This file has the first five implementations. To run it change the test file name and prediction file name as needed. Update the similarity_metric to the desired metric. (Read Comments to see what to enter). I used Spyder IDE so didn't need to run any command, just run the program file and output will be stored in same directory as the code. Update train and test file paths accordingly.
- **ItemBasedCF.py** : This file implements the item based CF with adjusted cosine. Just update output file name and test file name to get prediction files accordingly, also update train and test paths. I used Spyder IDE so didn't need to run any command, just run the program file and output will be stored in same directory as the code.

- **NCFTensorFlow.py** : This file implements neural CF. Update train and test file paths and run the code directly, it will save all files in the same directory. No commands needed on Spyder IDE.
- **EnsembleMethod.py** : This file implements the ensemble methods. Replace test_case with the test file to run. Update model_files2 with the different metrics you wish to test ensemble methods with. Update prediction file name. No commands needed on Spyder IDE.
- **EnsembleHybridMethod.py** : This file implements the hybrid ensemble method. Run the file directly, no commands needed on Spyder IDE.