Empirical Analysis of Predictive Algorithms for Collaborative Filtering

Arun Rajagopalan
(arajago6@hawk.iit.edu | A20360689)
Dhayalini Nagaraj
(dnagaraj@hawk.iit.edu | A20359686)
Sarath Kumar Prabhakaran
(sprabha7@hawk.iit.edu | A20351832)
Srinivasan Ramaraju
(sramaraj@hawk.iit.edu | A20352385)
Vijay Bharrathi
(vvijayas@hawk.iit.edu | A20356386)

Problem Statement:

- Collaborative Filtering is a technique used by recommender systems to build personalized recommendations for users, Algorithms predict a user’s preferences based on preferences from similar users.
- The inherent assumption is that if preferences of two users match on a certain thing, then their preferences might match for other things too.
- In this project we implement, analyze and compare several different algorithms designed to predict user preferences.
- Our project is thus aimed to find the best algorithm that provides relevant entities rather than a random subset from a large pool of entities and use our documentation on strengths and weaknesses of several algorithms to improve upon them.

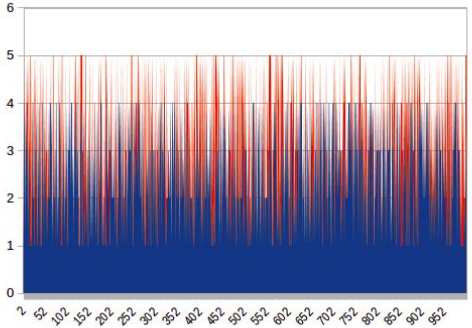
Data and approaches

- We used the ratings of books from the Amazon Product data as our dataset.
- The algorithms we evaluated are divided as 3 categories based on approach.
    a. Memory based approaches (Predict the rating of a user considering an entire d
    user votes)
        - Correlation - Uses Pearson correlation coefficient to find similarity between two users.
        - Vector Similarity - Treats users as documents, book titles as words and ratings as word frequencies.
    b. Extensions to Memory based approaches (To remedy deficiencies of (a) such as cold start)
        - Default voting - Adds neutral or somewhat negative default ratings for unrated books.
        - Inv User Frequency - Reduces weight for universally liked books to find unique interests.
        - Case amplification - Favours only strong correlations between users & punishes others.
    c. Model based approaches (Builds a model from data and uses model to make predictions)
        - Cluster - Uses Bayesian classifier where probability of ratings are independent of others
        - Matrix Factorization- The process of breaking down a matrix into a product of multiple matrices(S,U and V) to find most similar users.

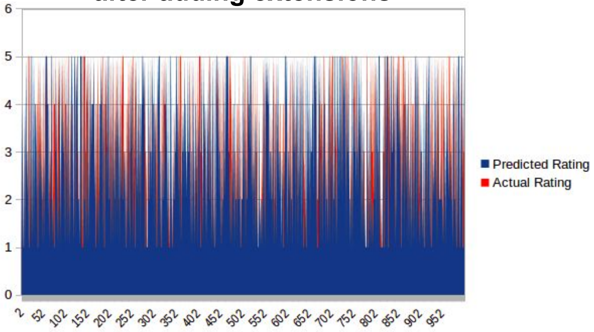
Experiments and results with MAE

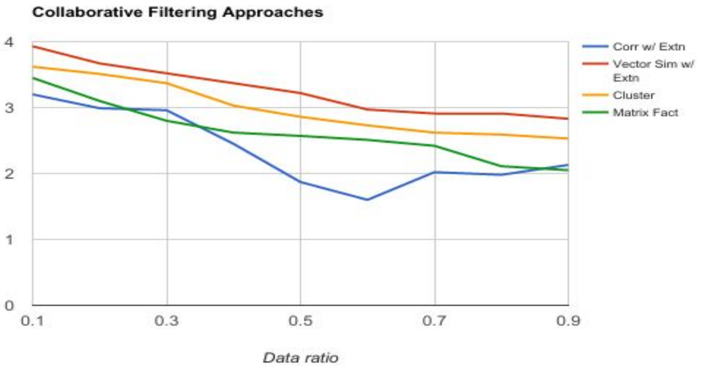
Sample Ratio	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Correlation with Extensions	3.20	2.99	2.96	2.45	1.87	1.6	2.02	1.98	2.13
Vector Similarity with Extensions	3.93	3.67	3.52	3.37	3.22	2.97	2.91	2.83	2.75
Cluster Model	3.62	3.51	3.37	3.03	2.86	2.73	2.62	2.59	2.43
Matrix factorization	3.45	3.10	2.80	2.62	2.57	2.51	2.42	2.11	2.05

Rating comparison from memory based algorithms before adding extensions



Rating comparison from memory based algorithms after adding extensions





Conclusions

- Matrix Factorization and Correlation methods with extensions outperform vector similarity methods with extensions and cluster model.
- The Model based algorithms requires small memory, predicts faster than memory based algorithms.
- Memory based correlation algorithm did not perform well, when there are relatively few votes in the training data.
- We used Mean Absolute Error to calculate the effectiveness of each algorithm and compared the results.
- We learnt about collaborative systems and what are the various ways to implement them.