Project Progress Report-3

Empirical Analysis of Predictive Algorithms for Collaborative Filtering

Dhayalini Nagaraj

A20359686

Abstract:

Collaborative filtering is a method of making automatic predictions about the interests of a user by collecting preferences or taste information from many users. In this project, we use different algorithms for this task based on correlation coefficients, similarity calculations to predict the users preferences. We analyse and compare different algorithms based on accuracy in a set of problem domains.

Experiments:

In Model-based collaborative filtering, it uses the user database to estimate or learn a model, which is then used for predictions. Model based algorithm is building a model from the dataset of book ratings and use that model to build the recommendation system without using the complete dataset. We use Probability of predicting method, Bayesian networks and clustering is used for this algorithm.

I have used K-means clustering algorithm to build the model. The idea I have used in this is there are certain groups or types of users who have common set of preferences. So I have used K-means clustering algorithm to cluster similar users who has similar rating for books.

K is an input to the algorithm that specifies the desired number of clusters. I have used k =5 for the clustering algorithm. In the first pass the algorithm takes the first k users as the centroid of k unique clusters. The remaining users are then compared to the closest centroid. The cluster centroids are recomputed based on cluster centroids formed in the previous pass. The N number of users has to be clustered. The ratings dataset has 22,507,155 ratings. In order to cluster the users I have used 500,000 ratings. Clustering results of the users U={u1, u2, u3, u4, u5} are represented as {C^1, C^2, C^3, ..., C^k}.

In order to cluster similar users, I calculated the similarities between pairs of users and identify their neighbourhood. I used Pearson correlation coefficient function for measuring the similarity between the users.

$$sim_{u,u'} = \frac{\sum_{t \in T(u) \wedge T(u')} (R_u(t) - \overline{R_u}) \cdot (R_{u'}(t) - \overline{R_{u'}})}{\sqrt{\sum_{t \in T(u) \wedge T(u')} (R_u(t) - \overline{R_u})^2} \sqrt{\sum_{t \in T(u) \wedge T(u')} (R_{u'}(t) - \overline{R_{u'}})^2}}$$

In this equation t is the items and $R_u(t)$ is the ratings given by the user for the item t.

Data Smoothing.

Based on the cluster results, the data smoothening methods to items with no ratings.

$$R_u(t) = \begin{cases} R_u(t) & \text{if user } u \text{ rate the item } t \\ \hat{R}_u(t) & \text{else} \end{cases}$$

$$\hat{R}_u(t) = \overline{R_u} + \Delta R_{C_u}(t)$$

The $R_{Cu}(t)$ is average deviation rating for all the users in the cluster.

Based on the similarities, the neighbour of the user is found. For this we can search the whole database. But when the number of users are more, it is difficult. So I used the cluster concept. The centroid of a cluster is represented by average rating of all users in the cluster. The similarity between each group and the active users are again calculated.

Now we have two ratings, the user ratings and the group ratings. The confidential weight for the user u to the item t is

$$w_{ut} = \begin{cases} 1-\lambda & \text{if user } u \text{ rate the item } t \\ \lambda & \text{else} \end{cases}$$

Here λ is the parameter for tuning the weight between original rating and group rating. The value of λ is set to 1 for the cluster based filtering algorithm.

The probability expression is the probability that the active user will have a particular vote value for item j given the previously observed votes.

$$\Pr(C = c, v_1, \ldots, v_n) = \Pr(C = c) \prod_{i=1}^{n} \Pr(v_i | C = c)$$

Based on this Cluster Probability Model, we then build a recommendation system and trained the test data.

Evaluation Crietria.

After building the cluster probability model, the test data set is then passed to the model and the results of the book dataset is analysed. I varied the number of rated items provided by the active user and analysed the results. When the parameter λ was varied the the performance of the prediction increased. When it was 0.1, 0.4,0.5 and 1 the prediction performance increased. This shows when the cluster model is formed, the neighbours has to be selected with high similarities which would in turn affect the predictions.

When the k value or number of clusters vary, the predictions varied. When the cluster is more it provides the information more specific while the small clusters represent general difference among dissimilar users.

When number of rated items was varied for active users from 5,10 and 20 the Mean absolute Error decreased.

5	1.857
10	1.768
20	1.3667

Similarly when I increased the sparsity of the data from 20, 40 and 60%, the Mean Absolute Error decreased gradually. When the pre-selected neighbours are increased the predictions were more similar, which showed the effect of clustering.

The recommendations is dependent on the cluster model that is build and the cluster model has to be built by studying the similarities of the users and items. By using probability cluster model, the accuracy of the prediction is increased. It also solves the scalability problem. The prediction speed is also faster because the time required to query the model is faster than to query the whole dataset. It is difficult to add a new data to the model based system which makes it inflexible. Quality of predictions depends on the way the model is built. If the cluster model does not select its similar neighbours then the prediction quality would be poor.

References:

http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/modelbased.html

https://ashokharnal.wordpress.com/2014/12/18/worked-out-example-item-based-collaborative-filtering-for-recommenmder-engine/

http://aimotion.blogspot.com/2009/11/collaborative-filtering-implementation_13.html

https://www.codementor.io/jadianes/build-data-products-django-machine-learning-clustering-user-preferences-du107s5mk

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.227.5662&rep=rep1&type=pdf
http://www.cis.upenn.edu/~ungar/Datamining/Publications/clust.pdf