CS229 Final Report
Statistical Analysis and Application of Ensemble Method on the Netflix Challenge

Jack Cheng, Virginia Chu, Yang Wang

December 15th, 2006

1. Introduction

The Netflix Prize project is proposed by the Neflix Inc., in order to seek accurate predictions on movie ratings. As one group in the Stanford Netflix Prize team, our responsibility is to explore useful statistics and data curation in the training data set, and to explore ensemble methods for improving prediction accuracies. We imported the Netflix data into a MySQL database for data aggregation, and then the aggregated results can be analyzed using Matlab or C++ scripts. So far, we have finished multiple clustering analyses to the movies and the customers by the K-means clustering techniques learnt from class [1]. We clustered the movies by multiple interesting criteria, such as the number of ratings to a movie, the average ratings to a movie, time progression on monthly numbers of ratings and rating averages, and the probability of different ratings for a movie. The customers are clustered with similar criteria except the time progression because the monthly numbers of ratings and rating averages change from time to time, depending on the movies the customers watch in those months. After the training data have been properly clustered through various criteria, we used ensemble methods to effectively combine the advantages of various classifiers and obtain improved results.

2. Statistics and Clustering Results

2.1 Long tail phenomenon

Online DVD sellers can beat the local sellers because there is virtually unlimited "shelf-space" for the online sellers to satisfy the needs for a large amount of customers. While the local sellers can capture and sell the top thousands of movies, the amount of movies not captured can be enormous and the revenue generated from selling these unavailable movies can be comparable to that generated from selling only the top thousands. This is an example of the famous phenomenon in the online business world, called *Long Tail Phenomenon.*

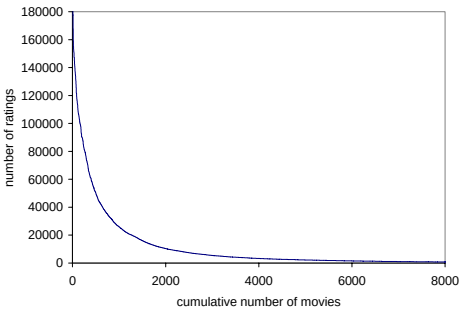

Fig. 1. Number of Ratings to Different Movies

The items' popularity and users' participation in online rating systems often demonstrate Long Tail Phenomenon as well. From Figure 1, it can be observed that a large amount of the movies have small amount of ratings whereas a small amount of movies have an extremely large amount of ratings. Similarly, a large amount of users contribute small amount of ratings whereas a small amount of users contribute an extremely large amount of ratings. This shows the typical extremity of the movie popularity and user contribution in the online technology world.

2.2 Rating averages and standard deviations

The movie and user rating averages and standard deviations are distributed close to the normal distribution. Therefore, performing k-means clustering would just give most of the cluster centroids close to the average rating of 3.5 and standard deviation of 1. One exception to the normal distribution was found in the standard deviation of the user ratings. There is a group of users that had a rating standard deviation of zero. This group comprises of users that gave only 1 rating or gave the same rating every time.

2.3 Time progression on monthly number of ratings and rating averages for movies

Monthly numbers of movie ratings in the first 12 months after the first rating are considered for k-means clustering. K-means clustering with k = 10 is performed (Fig. 2). Numbers of movies in these 10 clusters from top to bottom are 13, 154, 44, 15, 63, 548, 40, 18, 19, and 16856, respectively. According to the cluster centroids, almost all movies started out having very little rating during the first 2 months. However, some movies gained a lot more ratings in the 3 months to follow, peaked at around 7 months, and then the numbers started to decline. Yet, there are movies that slowly picked up speed and only started to observe significant increase in the number of ratings in the 7th or 8th month, and continued to increase throughout the first year. Despite these trends observed, most of movies (16856 out of 17770) in the probe set remained very low in the number of ratings throughout the year. The average number of ratings these movies got per month remained at 40 per month.

We also performed clustering on the time progression for average rating. The average rating time progression is clustered using k-means (k=7). Looking at the cluster centroids, most clusters seemed to have ratings that stay fairly constant throughout the months, with only small increasing trend or slight declines within the first four months.

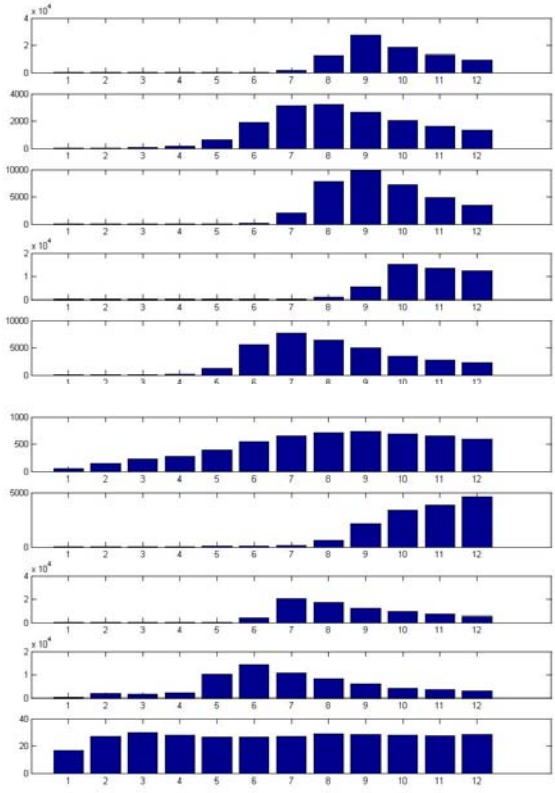

Fig. 2. Time Progression on Monthly Number of Ratings

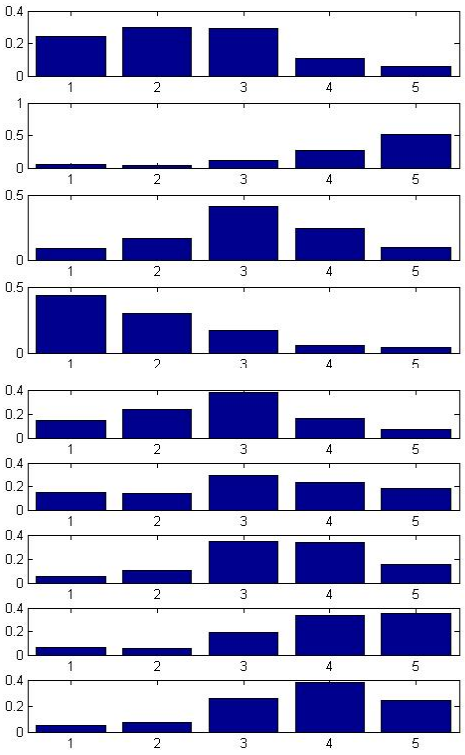

Fig. 3. Movie Clusters by Probability of Different Ratings.

2.4 Movie rating profile

Rating profile is the number of each rating (1 to 5) divided by the total amount of ratings. Therefore, this is the probability distribution of the movie receiving each rating, $P(rating = k)$ where $k \in \{1, 2, 3, 4, 5\}$, calculated empirically based on the sample data.

We used a clustering of k = 9 cluster centroids (Fig. 3). Number of movies in these 9 clusters from top to bottom are 1605, 657, 2847, 630, 2407, 2149, 2908, 2005, 2562, respectively. The clustering showed that most movies have a peak rating that they receive the most counts, and it tapers off the two ends. Variations come from the peak rating value and how fast the ends taper off. None of the clusters centroids represented movies that received uniform distribution of ratings. This information is useful, as it tells us

the importance of the predictive power mean and standard deviation. All the clusters have relatively the same number of movies in them.

2.5 User rating profile

Similarly, the rating profile of each user is empirically determined using the same way as the movie rating profile. To cluster the user rating profiles, we used a clustering of k = 9. The cluster centroids are plotting in Fig 4. The number of users in each cluster are as follows: 29017, 53123, 90821, 74504, 20615, 30999, 73535, 48173, 59402, from the top to the bottom respectively. The cluster centroids are representative of the users in that cluster. The centroids revealed that there is a group of users that give a very evenly distributed number of ratings from rating 1 to 5. However, most of the other users have a peak rating that they tend to give a lot of. There are variations on how fast they taper off. Some clusters showed a very distinct peak, where they almost always give the same rating; where other groups distribution resembles a normal distribution.

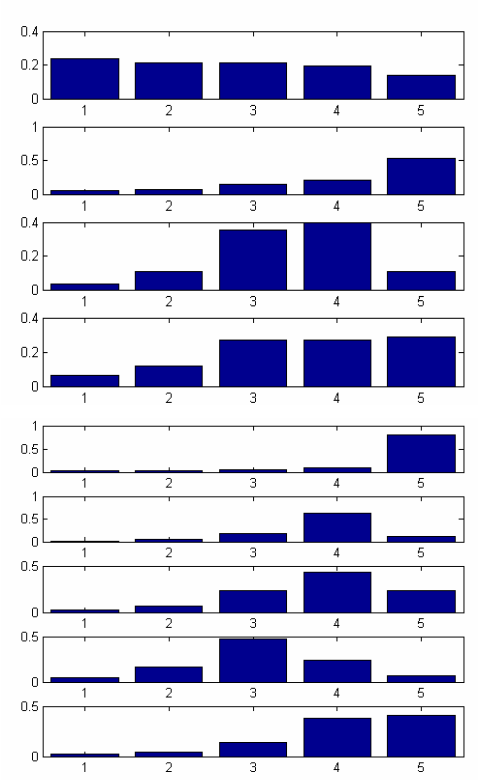

Fig. 4. User Clusters by Probability of Different Ratings.

3. Ensemble Method

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or un-weighted voting) to classify new examples [2]. If the classifiers are accurate and diverse, i.e. if the individual classifiers have error rates below 0.5 and their errors are at least somewhat uncorrelated, an ensemble of classifiers are more accurate than the individual classifiers. An ensemble method by weighting predictions from multiple approaches is employed in our project. We try to find an optimal way of combing the predictions from algorithms such as logistic regression, mixture of multinomial, matrix factorization, K-nearest neighbor, and K-means.

3.1 Concept of the ensemble method

First, we cluster the whole training data into *n* clusters ($C_i, i = 1, \cdots, n$) by certain criterion. Assume that we have predictions for the test data set from *m* classifiers: $H_j, j = 1, \cdots, m$. The weighting factor of classifier H_j on cluster C_i is denoted as w_{ij} . If the entry *x* to be predicted belongs to cluster C_k according to the clustering criterion, the ensemble prediction is:

$$\hat{r}(x) = \sum_{j=1}^{m} w_{kj} H_j(x)$$

3.2 Two approaches of training the weighting factors

We train our weighting factors w_{ij} using the probe dataset specified by Netflix. A deterministic approach and a gradient descent approach are employed to compute the factors. In the deterministic approach, the root of mean square error of classifier H_j on cluster C_i is fist computed:

$$RMSE_{ij} = \sqrt{\sum_{k=1}^{n_i} \frac{1}{n_i} \left[w_{i,j} H_j\left(C_i(k)\right) - r\left(C_i(k)\right) \right]^2}$$

Here n_i is the number of probe entries that belong to cluster C_i, and $C_i(k)$ is the *k*-th probe entry that belong to cluster C_i. A performance index is then computed using following equations:

$$P_{ij} = \begin{cases} \dfrac{1}{RMSE_{ij}}, RMSE_{ij} \neq 0 \\ 5 \max\limits_{k, RMSE_{ik} \neq 0} P_{ik}, RMSE_{ij} = 0 \end{cases}$$

Finally, the deterministic weighting factor is calculated:

$$w_{ij} = \frac{P_{ij}}{\sum_{l=1}^{m} P_{il}}, (\sum_{j=1}^{m} w_{ij} = 1)$$

In the gradient descent approach, an error index to be minimized is first defined:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{n_i} \left[w_{i,j} H_j\left(C_i(k)\right) - r\left(C_i(k)\right) \right]^2$$

The derivative of this index over weighting factor $w_{p,q}$ is:

$$\frac{\partial J}{\partial w_{p,q}} = \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{n_i} 2\left[w_{i,j} H_j\left(C_i(k)\right) - r\left(C_i(k)\right) \right] \right\} \sum_{k=1}^{n_p} H_q\left(C_p(k)\right)$$

Then the gradient descent searching is defined by (a learning rate α of 0.001 is found to be appropriate):

$$w_{p,q} := w_{p,q} - \alpha \frac{\partial J}{\partial w_{p,q}}$$

3.3 K-fold cross validation while computing the weighting factors

For each of the above two approaches computing the weighting factors, the concept of cross validation is employed:

1. The probe data is randomly divided into totally *k* = 10 subsets: $S_1, S_2, \cdots S_k$.
2. For $j = 1, \cdots, k$

   Compute weighting matrix W_j on $S_1 \cup \cdots \cup S_{j-1} \cup S_{j+1} \cup \cdots \cup S_k$ (i.e. leave out S_j).

   Test W_j on S_j to get generalization error $\hat{\varepsilon}_j$.
3. Pick W_j that has the lowest generalization error $\hat{\varepsilon}_j$. Then use W_j to apply the ensemble method over the whole probe data set to compute the overall ensemble RMSE $\hat{\varepsilon}$.

3.4 Ensemble analysis results

The RMSE over the whole probe data set predicted by the five available algorithms is listed as following:

Mixture of Multinomial	KNN	Matrix Factorization	Logistic Regression	K-means
0.9614	1.0097	0.9330	0.9387	1.3399

From our experience, including KNN and K-means does not help in improving ensemble prediction. The reason is probably that the RMSEs from these two algorithms are much higher than the other algorithms. Therefore, for the results presented in this paper, the ensemble analysis only combines predictions from mixture of multinomial, matrix factorization, and logistic regression.

Five different clustering criteria (*CC*) are tested. Three of them are by movies: time progression on monthly number of ratings to the movie (CC_1), time progression on monthly rating averages to the movie (CC_2), and movie rating profile (CC_3). The other two criteria are by customers: time progression on monthly number of ratings by the customer (CC_4), and customer rating profile (CC_5). Using the weighting matrix computed by the deterministic or gradient searching approach, ensemble predictions for the whole probe data set are made. Finally, the overall ensemble RMSE $\hat{\varepsilon}$ is found for each of the above clustering criteria.

Deterministic		Gradient Descent	
Clustering Criterion	Ensemble RMSE $\hat{\varepsilon}$	Clustering Criterion	Ensemble RMSE $\hat{\varepsilon}$
CC_1	0.914119	CC_1	0.914283
CC_2	0.914133	CC_2	0.914301
CC_3	0.914128	CC_3	0.914316
CC_4	0.914122	CC_4	0.914322
CC_5	0.914067	CC_5	0.914285

The minimum RMSE out of the original five algorithms is 0.9330 by Matrix Factorization, while the ensemble method with deterministic approach generally results into an RMSE around 0.9141. Therefore, the ensemble analysis improves the prediction performance by about 2%. The RMSE from the ensemble method with gradient descent approach is about 0.9143, which is slightly higher than the deterministic approach, but still obviously lower than this of the Matrix Factorization algorithm.

4. Conclusions and Future Plans

We have performed K-means clustering on the movies and users by different clustering criteria. With the movie and user clusters, the ensemble method shows a 2% improvement on the rating prediction RMSE compared to the best classifier from 0.9330 to 0.9143, which is approximately 4% improvement to *Cinematch*, the existing prediction system Netflix is using.

The performance of ensemble method depends on both the performance of classifiers and the ability of clustering to group the data into portions each classifier perform well and portions the classifier does not. Therefore, it would be helpful to investigate the clusters in which each classifier has the least RMSE so as to evaluate the effectiveness of the clustering criteria and suggest a finer grouping in those clusters to better fit the classifier to the data in those clusters.

More clustering criteria, for instance content-based grouping by genre and actors, would be used. Implementation of new classifying algorithms or modification on classifiers such as higher K-value for K-means would also be done. By employing a rich and diverse set of clusters and classifiers, we look forward to higher benefits from the strengths of the classifiers and hence further improvements on the result RMSE.

5. Acknowledgements

We hope to express our sincere thankfulness to Professor Andrew Ng for leading the Netflix Prize group and providing helpful comments and recommendations. Thuc Vu, Chuong Tom Do and Vasco Chatalbashev are also greatly appreciated for their mentoring and technical assistance. We would like to extend our gratitude to other teams in the group for their prediction results and cooperation.

References

[1] A. Ng. Lecture Notes. *CS 229 Machine Learning*, Department of Computer Science, Stanford University, USA, 2006

[2] T.G. Dietterich. Ensemble methods in machine learning. *In Multiple Classier Systems*, Cagliari, Italy, 2000.