Keeping it Simple: Revisiting the Netflix Challenge

Evan Huang (evhuang@stanford.edu)[1], Victor Cheung (hoche@stanford.edu)[2]

Stanford University
Electrical Engineering Department[1]
Computer Science Department[2]

Abstract

The Netflix Prize rewarded the top team with an incremental 10% improvement on their existing Cinematch algorithm with a million dollars. It turns out that improving the RMSE of predictions is difficult. Furthermore, the winning solution was cumbersome and infeasible for commercial use.

We will study the Netflix problem in a simplified setting under discretization, and combine prior information calculated on the dataset to make predictions on each individual movie. This method achieves nearly the performance of KNN in the discretized setting but can be up to 420x faster.

Introduction

Dataset We have a matrix of size 480189×17770 where rows are users and columns are integral ratings (1-5) assigned to each movie. The matrix can contain more than nine billion observations, but actually holds only a hundred million. We treat the missing data as having been uniformly random selected and masked.

Discretization In the simplified setting, we select a threshold T such that for ratings $R > T$, $DR = 1$ and for $R \leq T$, $R = 0$.

Figure 1: Movies with the highest prior p

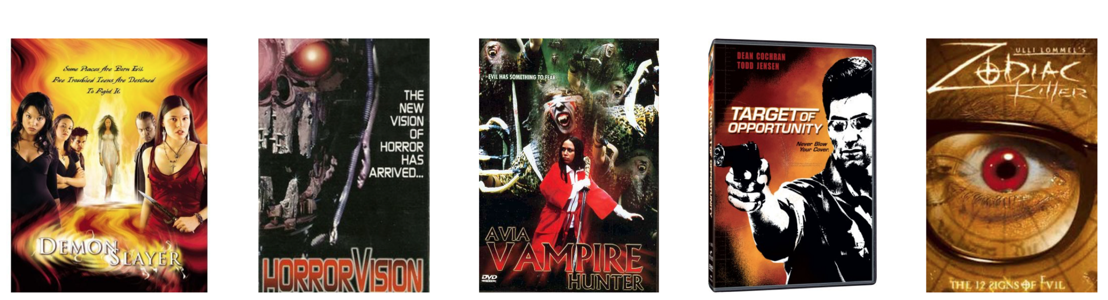

Figure 2: Movies with the lowest prior p

K-Nearest Neighbors

Due to the prevalence of user data, we used K-Nearest Neighbors as our baseline model. For each of our test users, we identified the 100 nearest neighbors using euclidean distance. With test user i, neighbor j, and respective ratings R_i and R_j for N movies watched, the distance is:

$$d_i j = \frac{||R_i - R_j||^2}{N} \quad (1)$$

K-Nearest Neighbors is not without limitations. The equal weighting of neighbor ratings underweights the closest neighbors. Thus we also consider the a K-Nearest Neighbors approach with constant scaling. Here we multiply the k^{th} nearest neighbor with $(1-k)$ before averaging, putting larger weight on closer neighbors than farther ones.

$$\hat{R} = \frac{\sum_{j=1}^{k} (1-k) * R_j}{N} \quad (2)$$

Estimating Population Parameters

This problem resembles an estimation problem of the following form: there are n users, each with an unknown parameter p_i, and we observe n independent random variables $X_1, \ldots, X_n$ with $X_i \sim Binomial(t, p_i)$.

Consider the discretized setting of ratings into "like" and "dislike"; then we can make a statement about the average individual's latent parameter by using information from the entire set of observations across all n individuals.

Our estimate for the set of the population parameters is an ϵ-net of size m, where we solve a linear (or quadratic) program that minimizes the distance between the empirical moments and the population moments (from the ϵ-net). See [1] for details.

We use this technique to infer the latent parameters of users/movies with few surviving observations.

For movies/users with a sufficiently large number of observations, we use the standard MLE estimate.

Recovered CDFs for Movies

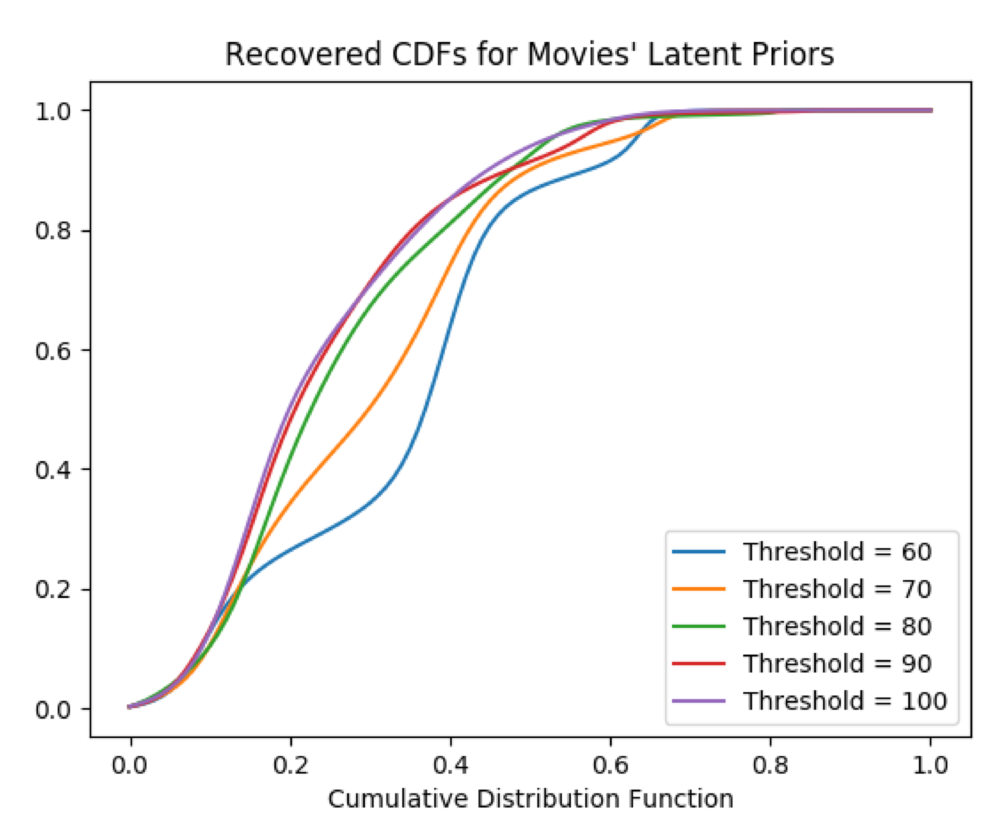

Figure 3: CDF for latent parameters for movies; characterizes how often any user rates a given movie highly

Recovered CDFs for Users

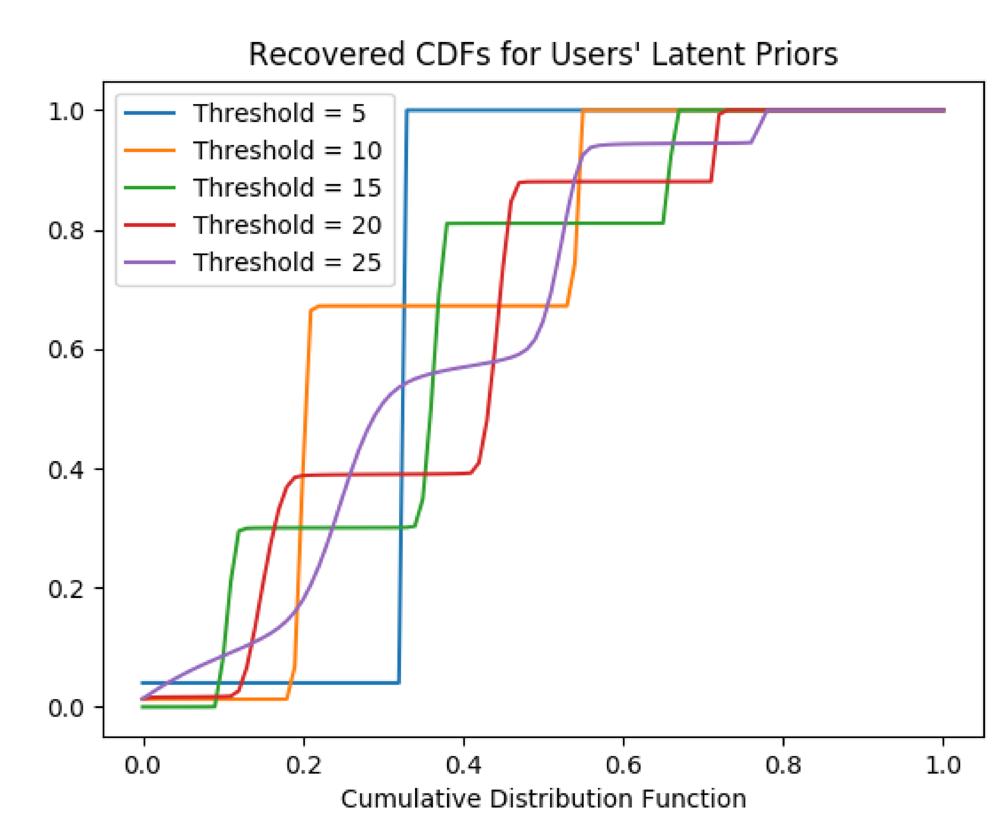

Figure 4: CDF for latent parameters for users; characterizes how often users rates any movie highly

Results

Classifiers	Accuracy	Precision	Recall	F1-Score
KNN - Bernoulli	0.751	0.735	0.694	0.712
KNN - L2	(0.947)			
KNN - CS	(0.930)			
M60 U5	0.748	0.683	0.748	0.714
M70 U10	0.710	0.667	0.685	0.676
M80 U15	0.743	0.702	0.735	0.718
M90 U20	0.703	0.629	0.697	0.661
M100 U25	0.730	0.683	0.680	0.682
M80 U15	**0.743**	**0.702**	**0.735**	**0.718**

Table 1: Classification metrics under discretized setting with KNN as comparison; CS refers to constant scaling; accuracy for KNN-L2 and KNN-CS refers to RMSE

Discussion and Future Directions

We find that precomputing priors scales well with number of users ($O(1)$ inference time) when compared to KNN ($O(n^2)$ inference time). We met our expectation that inference from prior should perform close to KNN since it incorporates all relevant and available knowledge about a specific user and a specific movie.

For future directions, we would want to extend to the multinomial setting for full use of data; enrich the dataset with covariates about movies and users (browsing behavior, completion rate, completion time, etc.); and use a many-to-many recurrent neural network on the enriched dataset.

References

[1] Kevin Tian, Weihao Kong, and Gregory Valiant.
Optimally learning populations of parameters.
09 2017.

[2] N N S S Altman and N S Altman.
An introduction to kernel and nearest-neighbor nonparametric regression.
The American Statistician, 46(3):175–185, 1992.