

Movie Rating Prediction System using Content-Boosted Collaborative Filtering

CS-771 course project

by Group 10

Divyanshu Bhartiya (10250)
Nitish Gupta (10461)
Satyajit Bhadange (13111056)
Umair Z Ahmed (13111166)

under the guidance of

Prof. Harish Karnick

Department of CSE
Indian Institute of Technology, Kanpur

July - Nov 2013

Abstract

Recommender Systems are becoming a quinessential part of our lives with a plethora of information available and wide variety of choices to choose from in various domains. Recommender sytems have a wide domain of application from movies, books, music to restaurant, financial services etc. Recommender systems apply knowledge discovery techniques to the problem of making product recommendations. In this project we combine the two most common methods of ‘Content-Based’ & ‘Collaborative Filtering’ which works by matching consumer preferences to other costumers in making recommendations. We use the famous dimensionality reduction method of SVD to capture the relationship in users demographics and his/her prefrerences. We apply our method on the Movielens data to predict movie ratings for users.

1 Introduction

Recommender systems are a class of information filtering systems which try to tackle the problem of making product recommendations and predicting the rating which a user would assign to it. Recommender systems are used widely to provide suggestions to a multitude of products such as movies, books, news articles, etc. They can help to save money (by discovering the potential for efficiencies) as well as make more money (by discovering ways to sell more products to customers) [2].

2 Related Work

The two prevalent approaches for building recommender systems are Collaborative Filtering and Content-based recommending. Collaborative Filtering systems determine how to recommend an item by exploiting similarities and differences among profiles of several users. On the other hand, content-based methods are based on characteristics of the product and those items which interests the concerned user.

Melville et al. [1] proposed a hybrid *content boosted collaborative filtering* system which creates augmented user profiles by adding *pseudo-ratings*, generated using content based techniques on unrated items, to original user profiles. And then, the rating prediction is computed using user-based collaborative approach where user similarities are computed as the Pearson correlation coefficient between the original user profiles.

3 Problem Statement

Our problem at hand is to use Movielens data set ¹ to create a movie rating prediction system. The movie lens dataset was collected through the MovieLens web site and consists of 100,000 ratings given by 943 users on 1682 movies. It also contains the genres of each of the movie and simple demographic information about the users which include a users ‘*Sex, Age, Profession & Zipcode*’. We have divided the problem into two parts :

¹<http://www.grouplens.org/datasets/movielens/>

1. Predict the rating for an existing user for movies the user has not seen.
2. Predict the ratings for a user who has not seen any movie from the dataset.

4 Proposed Approach

4.1 Predicting ratings for user in the database

4.1.1 Data Preprocessing

After extracting the data, we first create a ratings matrix R whose each row corresponds to a user and each column represents a movie and each cell holds the rating given by a user for a movie. $R[user][movie] = \langle rating \rangle$. Hence this is a 943×1682 sized matrix.

4.1.2 Generating Pseudo Ratings

The matrix R is very sparse since the users have seen very few movies from the dataset and rated them. In order to capture meaningful latent relationship, this sparsity in the matrix R needs to be removed. In the work done by Sarwar et al.[2], they had used the average rating of a movie or average rating given by a customer to fill the sparsity. We on the other hand used a Naive Bayesian classifier² for each user using the genres of the movies seen by the user as features to predict ratings for movies not seen by the user. We complete this matrix by predicting ratings for movies not seen by the users and call this matrix R_{pseudo} . This way we use the content of the movies seen by the users to predict his ratings for other movies. This is also called *Content - Based Filtering*.

4.1.3 Feature Space Reduction & Collaborative Filtering

After obtaining the complete matrix R_{pseudo} we apply the well-known matrix factorization method, Singular Valued Decomposition (SVD) on the $[R_{pseudo}]$ matrix that factors this 943×1682 matrix into three matrices as the the following :

$$R_{pseudo} = U.S.V' \quad (1)$$

Where U and V are two orthogonal matrices of size $943 \times r$ and $1682 \times r$ respectively: r is the rank of the matrix R_{pseudo} . S is a diagonal matrix of size $r \times r$ having all singular values of Matrix R as its diagonal entries. All entries of S are positive and arranged in decreasing order of their magnitude. It is possible to reduce S to a $k \times k$ matrix to have only k largest values to obtain S_k . The matrices U & V are reduced accordingly to obtain :

$$R_{pseudo,k} = U_k.S_k.V_k' \quad (2)$$

We use SVD in our approach for twofold reasons. First of all, we expect SVD to capture the latent relationships between users and movies that allow us to compute the predicted likeliness of a certain movie by a user. Secondly, we use SVD to produce *low-dimensional* representation of the original data and then use this space to perform our

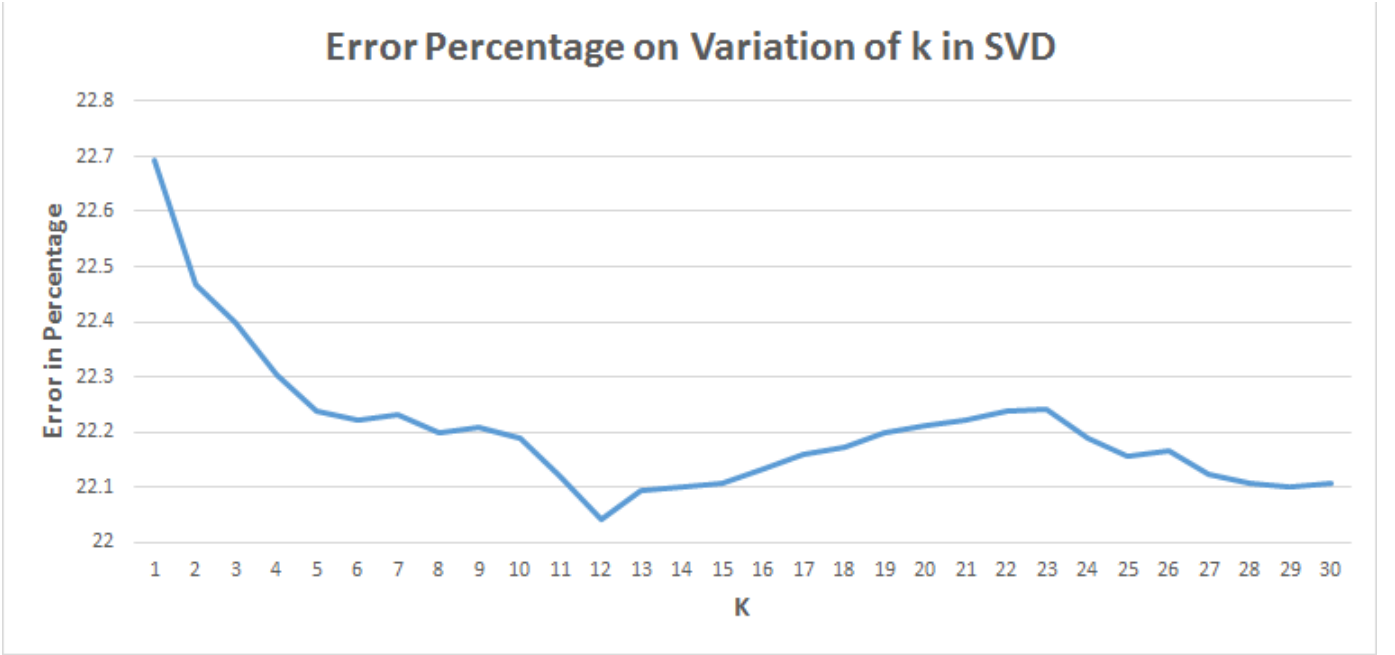
²We make use of weka toolkit <http://www.cs.waikato.ac.nz/~ml/weka/> for our classifiers and clustering algorithms.

predictions. We also test by doing a 5-cross validation test that applying SVD improves the rating predictions by 3% over the Naive-Bayes Classifier.

The optimum value of ' k ' is predicted by doing a 5-cross validation test by first filling R with only 20% of known ratings (80k ratings) chosen at random and then finding average error in prediction on the remaining 20% of ratings(20k ratings) using the following formula :

$$error = \frac{|PredictedRating - KnownRating|}{5} \quad (3)$$

The graph below shows variation in error for different values of k .



From the graph we see that the minimum error occurs at $k = 12$, hence we take the top 12 values in S while reducing it.

Hence we now get the final complete rating matrix we call R_f . This contains predicted ratings for all users in the database for all the movies in the database.

4.2 Predicting Ratings for a User who has not seen any movie from the dataset.

On obtaining the Matrix R_f containing all the ratings for all users in the database we have to build a method to predict the ratings for a completely new user about which we only know his/her demographics.

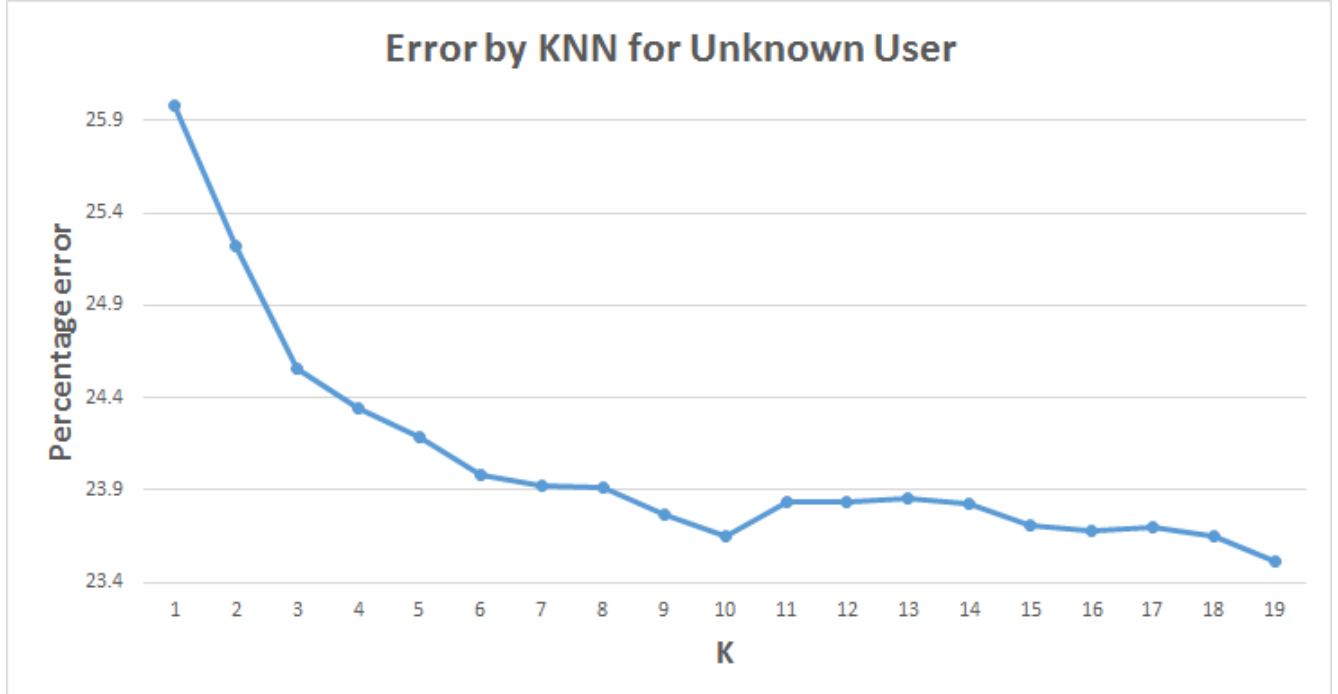
4.2.1 Nearest Neighbour Computation

We assume that the users ratings and likings are affected by his neighbours and peers. For eg. A 21 year old student will mostly have the same likings as a fellow student of the same age. Also likings in movies is affected by your location. We use the k- Nearest Neighbour

method to find k nearest neighbours of a new user by using his/her demographic features. After finding the k -nearest neighbours, we take an overage of the ratings by these k users for a particular movie and predict that the new user will give this rating to that movie. This method takes into account the collaborative set of features.

- The distance metric used in the k-nn method is Euclidian Distance.
- We use normalized values for all the demographic features over its values since the weight of each feature is the same although the range of values that each feature can take is different. For example, two users differing by a binary feature (0 or 1) such as Gender would be marked less similar compared to two users with a large difference in zip-codes due to equal weightage.
- We can use the euclidian distance metric on the feature of Zip-Code because we verified on the map of USA that Zip-Codes that are closer numerically are also closer geographically.

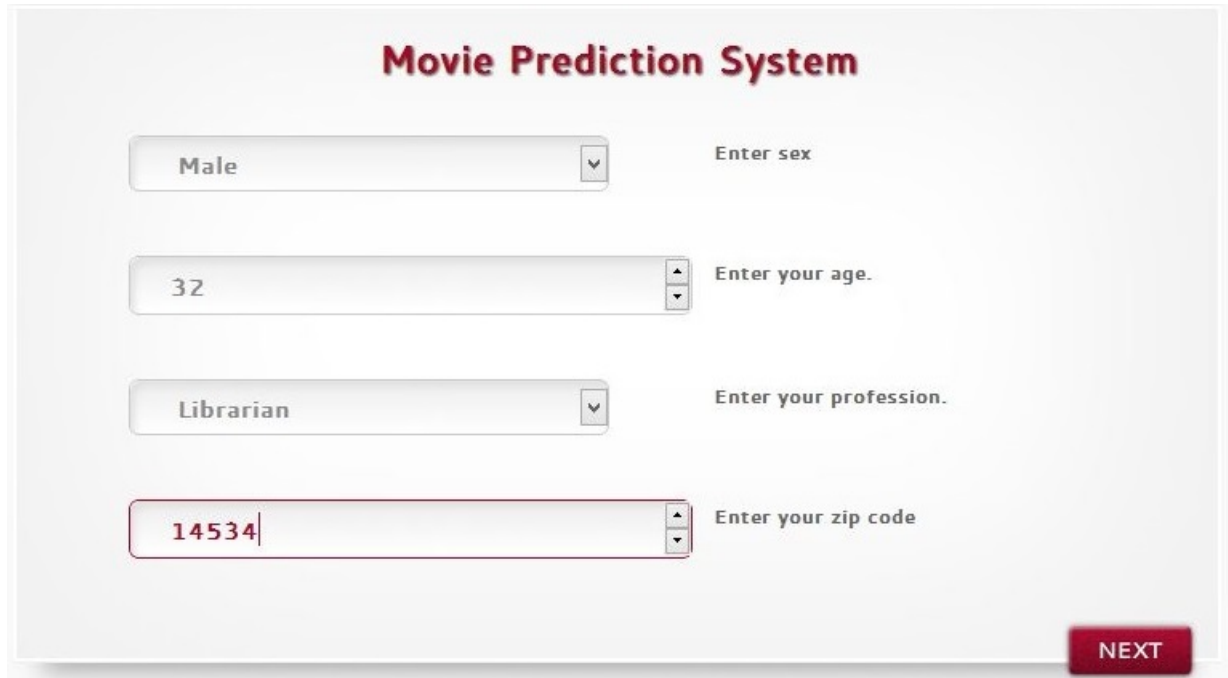
For validating the optimum number of nearest neighbours, we perform a 5-cross validation test by removing 10% (95 users) at random from our dataset, compute the new matrix R , then R_{pseudo} and finally the matrix R_f by SVD. We find the k -nearest neighbours for all the 95 users and take an average of their ratings to predict the rating for these users. As we already know their predicted rating when they were a part of the dataset we can calculate the average error using Eq. 3. The graph for the average error as a function of k is shown below.



From the graph we find that the optimum number of nearest neighbours to be used is $k = 10$. From the approach explained above we can also decide whether a given unknown user will watch a given movie. If the predicted rating for a new user is less than 2 we can safely say that the given user will not watch the movie.

5 Graphical User Interface

We built a GUI to ease the process of form filling and rating prediction. The screenshots for the same are given below.



Movie Prediction System

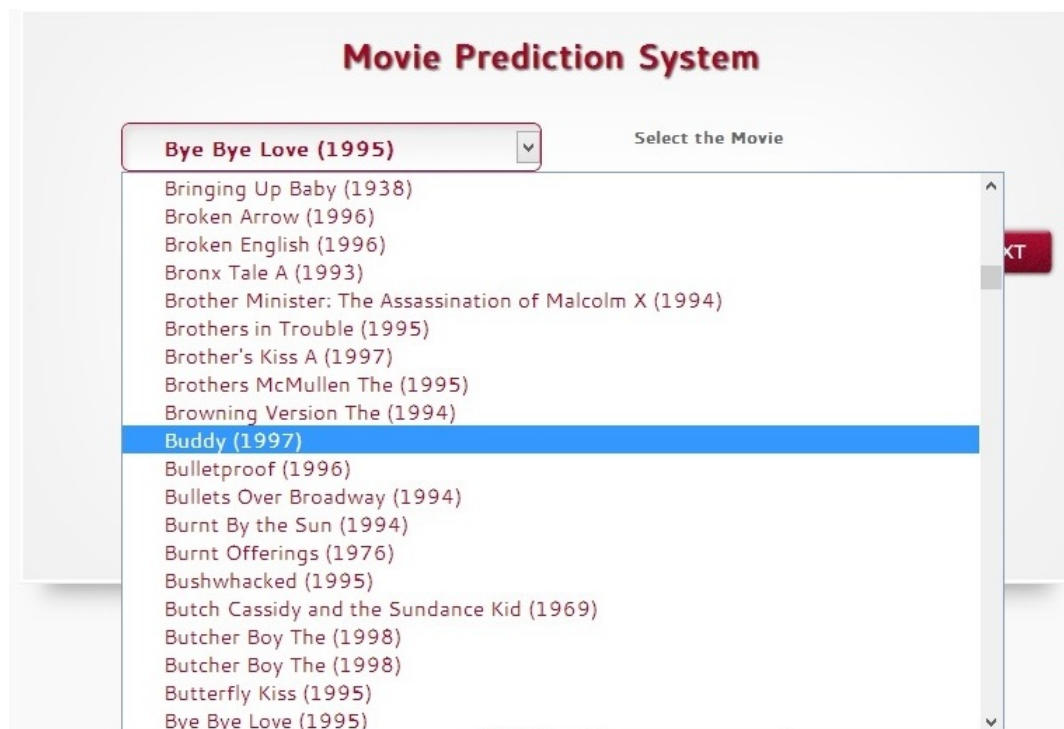
Male Enter sex

32 Enter your age.

Librarian Enter your profession.

14534 Enter your zip code

NEXT

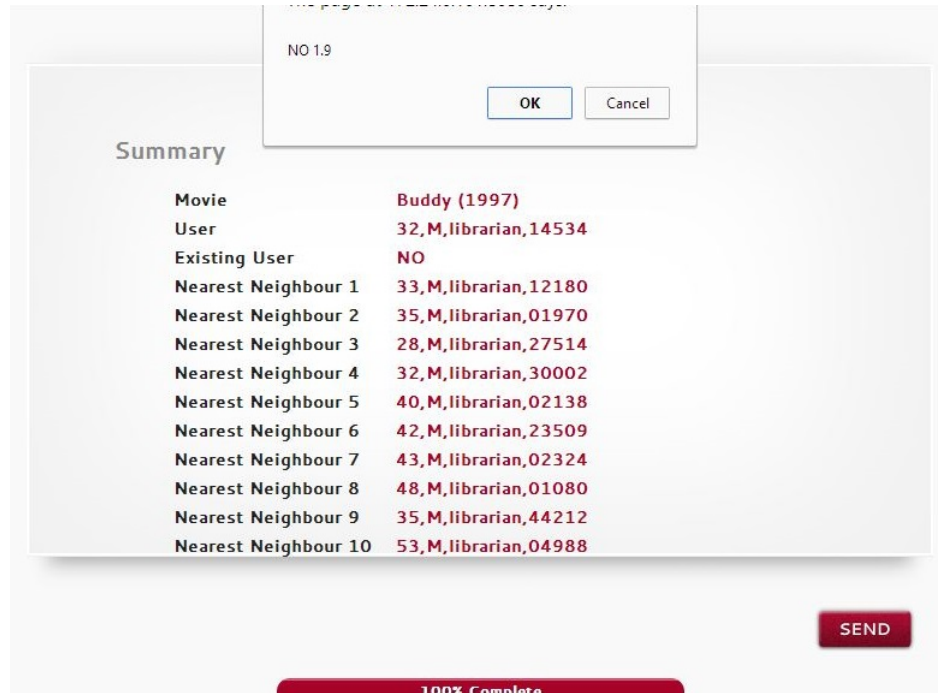


Movie Prediction System

Bye Bye Love (1995) Select the Movie

- Bringing Up Baby (1938)
- Broken Arrow (1996)
- Broken English (1996)
- Bronx Tale A (1993)
- Brother Minister: The Assassination of Malcolm X (1994)
- Brothers in Trouble (1995)
- Brother's Kiss A (1997)
- Brothers McMullen The (1995)
- Browning Version The (1994)
- Buddy (1997)**
- Bulletproof (1996)
- Bullets Over Broadway (1994)
- Burnt By the Sun (1994)
- Burnt Offerings (1976)
- Bushwhacked (1995)
- Butch Cassidy and the Sundance Kid (1969)
- Butcher Boy The (1998)
- Butcher Boy The (1998)
- Butterfly Kiss (1995)
- Bye Bye Love (1995)

NEXT



6 Future work and Improvement

In future, we would like to make the following additions to improve upon our classification

- Users tend to have a positive or negative bias while assigning a rating. We could overcome this in future by normalizing the ratings with respect to each user before training the classifier.
- Exploring the effect of using classifiers other than Naive Bayesian for learning individual user's preferences based on movie genres as features.
- Experimenting with a better distance metric to formulate the distance between users in a much more realistic way from the demographics given.

7 Acknowledgement

We thank Prof. Harish Karnick for his valuable support throughout the project and guiding us when required.

References

- [1] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. *AAAI*, 2002.
- [2] B. M. Sarwar, G. Karypis, J. A. K., and J. T. R. Application of dimensionality reduction in recommender system – a case study. 2000.