Assignment 2 - KNN Recommendation System

Team name: M2 Robo

Chan Kwan Yin	Lee Chun Yin	Chiu Yu Ying
3035466978	3035469140	3035477630
Team leader		

1. Background

In the KNN (k-nearest neighbours) model, we assume that similar users will give close ratings on similar movies. To predict the rating $\hat{r}_{ij}$ of user i on movie j ($1 \leq i \leq m, 2 \leq j \leq n$), the k most similar users $(n_1, \ldots, n_k)$ to user i are computed based on similar choices of ratings, and $\hat{r}_{ij}$ is estimated based on the known values $r_{n_1 j}, \ldots, r_{n_k j}$.

1.1. Similarity (Nearness) metrics

The similarity metric $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ used in the KNN model satisfies

$$\begin{cases} d(x,y) = 0 & \iff x = y \\ d(x,y) > 0 & \iff x \neq y \\ d(x,y) = d(y,x) & \forall x, y \end{cases} \tag{1}$$

While $d(i,j) < d(i,k)$ implies that j is more similar to i than k is, it is not necessary that the metric follows the triangular inequality. This enables the use of correlation coefficients in addition to common norm metrics.

1.1.1 Pearson correlation

Considering a user u_i rates movies with a distribution $R_i \sim (\mu_i, \sigma_i)$, the correlation between user u_i and the other user u_j would be the coefficient between the two distributions R_i and R_j:

$$\rho_{ij} = \frac{E[(R_i - \mu_i)(R_j - \mu_j)]}{\sigma_i \sigma_j}$$

To get the k similar data by considering the n movies that both user u_i and u_j have, we estimate the co-variance and variances:

$$E[(R_i - \mu_i)(R_j - \mu_j)] \approx \frac{1}{n} \sum_k (r_{ik} - \mu_i)(r_{jk} - \mu_j)$$

$$\sigma_i \approx \sqrt{\frac{1}{n} \sum_k (r_{ik} - \mu_i)}, \sigma_j \approx \sqrt{\frac{1}{n} \sum_k (r_{jk} - \mu_i)},$$

Note that the range of the Pearson Correlation Coefficient is $[-1, 1]$. To make the coefficient satisfy the conditions in 1 and not have negative values [1], we define

$$d(i,j) = 1 - \rho_{ij}$$

such that d has the range $[0, 2]$. A smaller value implies higher similarity between users.

1.1.2 Spearman Correlation

Contrary to Pearson's choice of mean and variance, Spearman Correlation uses the ranking of variables in the vector as the metric to eliminate effects of non-uniform distributions.

$$\rho_{ij} = \frac{\mathrm{Cov}(\mathrm{rank}_i, \mathrm{rank}_j)}{\sigma_{\mathrm{rank}_i} \sigma_{\mathrm{rank}_j}}$$

Similar to Pearson Correlation, the metric based on Spearman correlation is defined as

$$d(i,j) = 1 - \rho_{ij}$$

1.1.3 Euclidean Distance

We take the 2-norm squared:

$$d(i,j) = \sum_{l=1}^{n} (r_{il} - r_{jl})^2$$

1.1.4 Taxicab as a similarity metric

We take the 1-norm:

$$d(i,j) = \sum_{l=1}^{n} |r_{il} - r_{jl}|$$

1.2. Aggregation of ratings

After the k nearest neighbour users have been selected, $\hat{r}_{ij}$ can be estimated by aggregating $\{r_{n_1 j}, r_{n_2 j}, \ldots, r_{n_k j}\}$.

A naive aggregation method is to just take the arithmetic mean of these rating values without considering other factors such as the actual correlation.

2. Technical Details

The training data set provided by Netflix consists of more than 100 million ratings with 17770 movies and 480189 users. Such a huge data set would consume a significant amount of training time and memory ($O(m^2 n)$, since a correlation matrix between users is to be constructed), which is not possible for our hardware available. Therefore, only a subset of data is used for evaluation. To be specific, only first 1000 movies and first 1000 users that appear in the data set are considered.

2.1. Data preprocessing

The first 1000 movies are loaded into a numpy array with columns of Movie ID, User ID and Rating. The movie IDs and user IDs are reordered from 0 for the ease of indexing. Approximately 80% data are then reformatted into a rating matrix $R \in \mathbb{R}^{m \times n}$ for training, where r_{ij} is the rating of user i on movie j; the rest are retained for performance evaluation. The retained and missing data are imputed with the mean rating for the corresponding movie.

2.2. Hyperparameter selection

In this KNN model, there are three hyperparameter to be selected, namely

- Value of k
- Similarity metric
- Aggregation function

2.2.1 Choice of k

A large value of k would reduce accuracy as users with lower similarity are selected. On the other hand, a small value of k would be biased over the choice of the most similar user.

As a baseline model, we also set $k = n$, i.e. to evaluate the RMSE by taking all functions regardless the metric and using only naive arithmeitc mean for aggregation.

2.2.2 Choice of metric

Since the ratings are discrete in nature, it is expected that little difference is observed between the different metrics.

2.2.3 Choice of Aggregation function

Aggregation can be tuned by the similarity metric.

2.3. Predictive test set score (RMSE)

The model is evaluated by computing the RMSE between predicted and actual rating values:

$$\text{RMSE} = \sqrt{\sum_{(i,j)\in E} \frac{(\hat{r}_{ij} - r_{ij})^2}{|E|}}$$

where E is the set of retained evaluation data.

3. Model performance

The following table exhaustively lists our test results.

k	Similarity metric	Aggregation function	RMSE
10	Euclidean	Naive arithmetic mean	1.20094589890527
10	Taxicab	Naive arithmetic mean	1.1948902737623308
10	Pearson	Naive arithmetic mean	1.2001428015368176
10	Spearman	Naive arithmetic mean	1.1996683300387065

References

[1] Ted Hong and Dimitris Tsamis. Use of knn for the netflix prize. *CS229 Projects*, 2006. 2