Recommendation System Using Yelp Data

CS229 Stanford University

Yingran Xu, Jiale Xu

Background & Motivation

The reviews on Yelp implies people's tastes, personalities and lifestyles. The first motivation is to recommend friends for Yelp users based on the similarities they show; secondly, we can also predict how a user may like certain business based on his/her past experience and the experience from people who share similar interests with him/her.

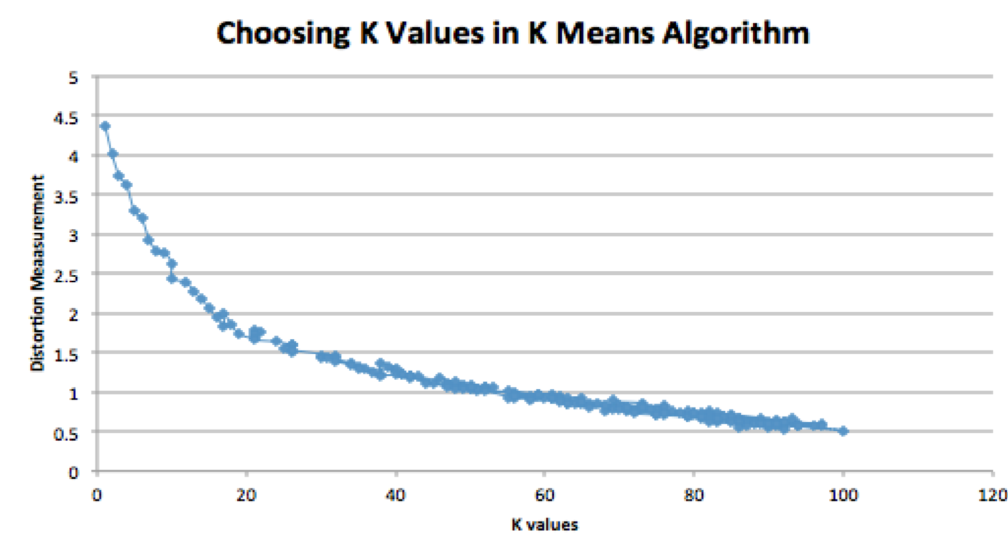
Data Preprocessing & Feature Selection

We preprocess the data to consider only one business type (restaurants) and only the users who has more than a threshold number (20, 50, 100, 150 or 200) of reviews. For K-means, we group the data into business categories, and use features of ***Star rating*** and ***Times of Visit*** to each category of business . For matrix factorization, we fetch the business and user pairs for each state. For evaluation, we use cross validation and split the data into 70% training data, and 30% test data.

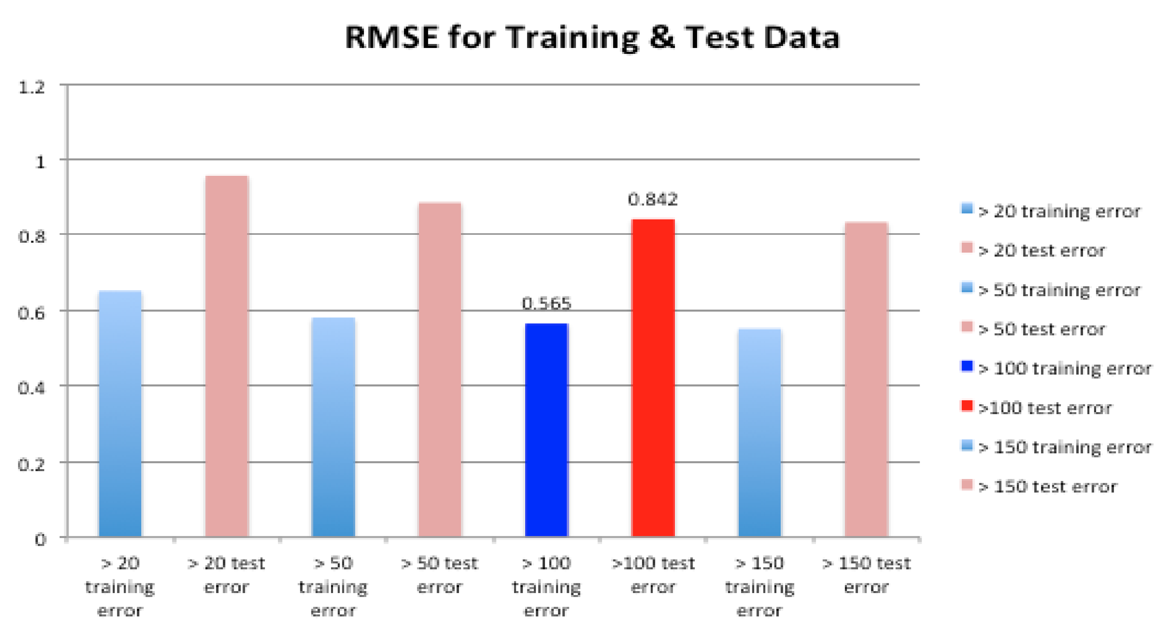
K-Means

We group the users based on features selected, train the weights of the features, and then calculate the RMSE. K value in the K-means algorithm is chosen by plotting the error versus each K value and finding the "knee" of the graph. Then, we run K-means for data in each state. However, due to smaller number of data we have, the algorithm gives some high variance result.

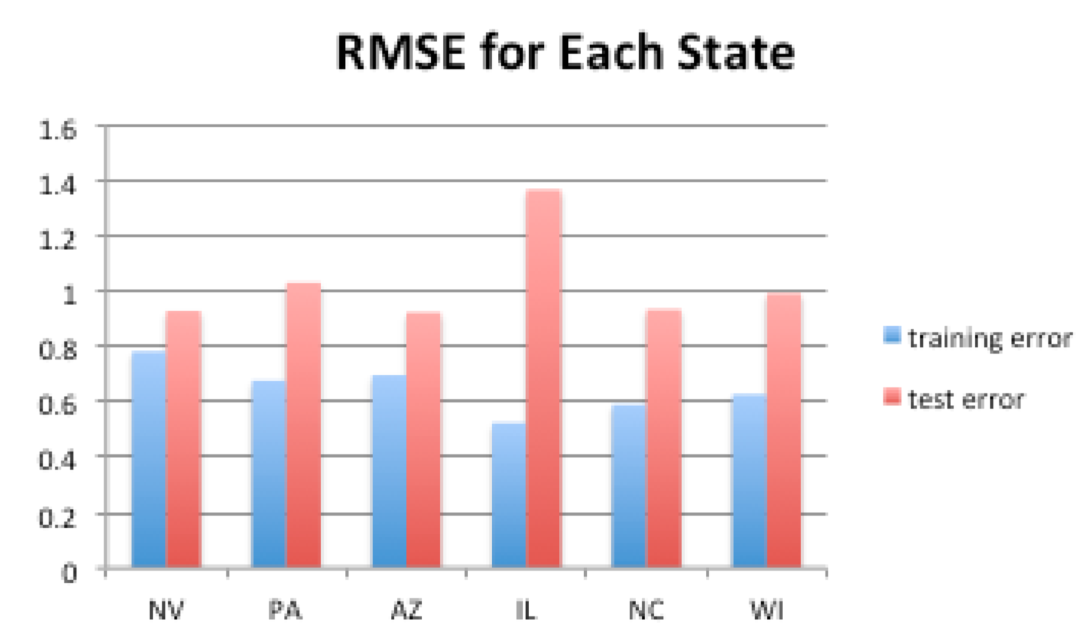
K-Means Result



Finding K value for number of review > 50



Number of reviews > 100 has 380 training data, 182 test data (13 clusters), and gives the smallest RMS test error of 0.842.



High variance result is shown, especially for state IL, which has only 34 training data.

Matrix Factorization

Regularized SVD: factorize the user-business rating matrix $R \in \mathbb{R}^{(N\times M)}$ into 2 matrices:

$$R \approx PQ \Rightarrow \hat{r}_{ub} = \sum_{k=1}^{K} p_{uk} q_{kb}$$

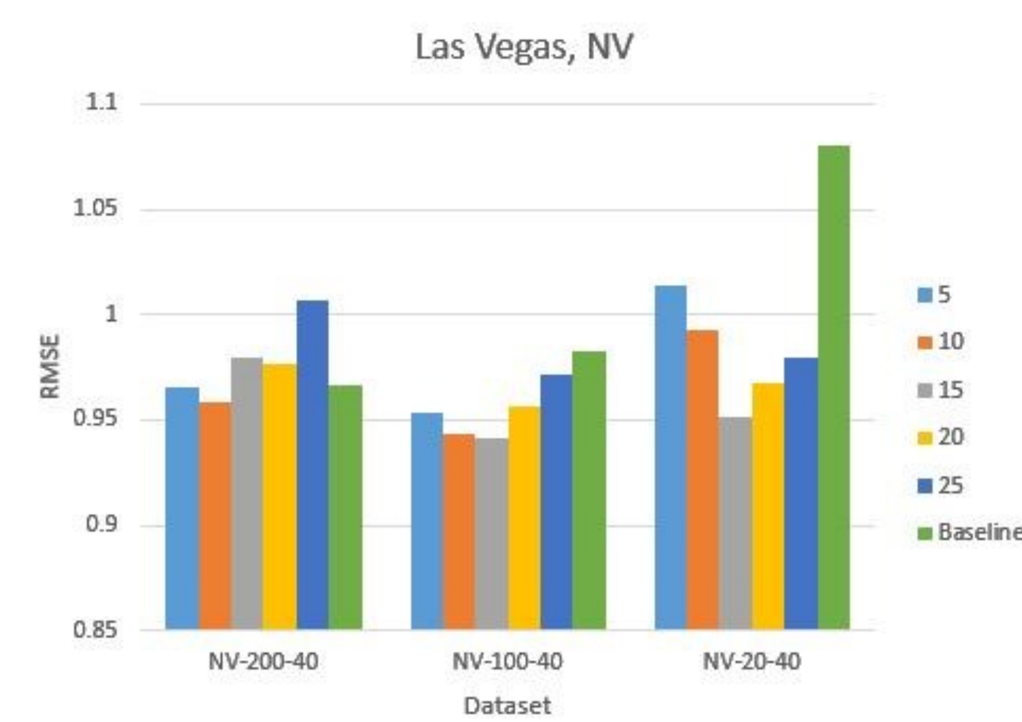
$P \in \mathbb{R}^{(N\times K)}$, $Q \in \mathbb{R}^{(K\times M)}$, N: # of users, M: # of business, K: # of features.
Cost funciton:

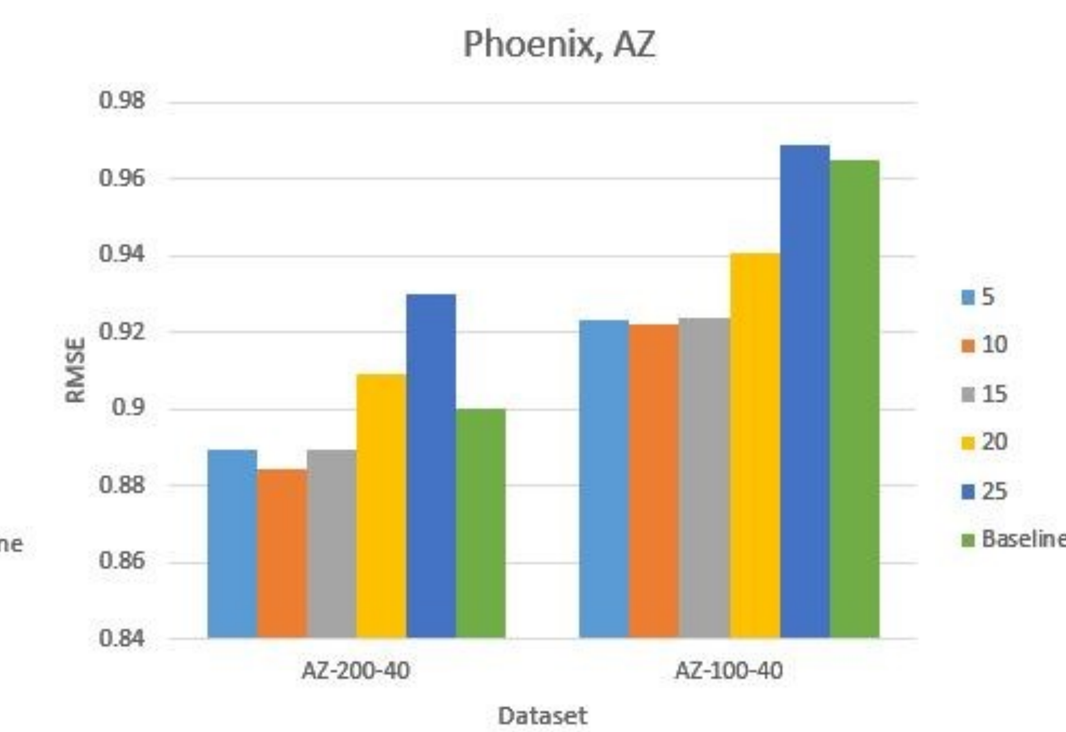
$$e_{ub} = \frac{1}{2}((r_{ub} - \hat{r}_{ub})^2 + \lambda(p_u^T p_u + q_b^T q_b))$$

Use gradient descent to find: $(P^*, Q^*) = \arg\min_{(P,Q)} \text{RMSE}$:

$$p_{uk} := p_{uk} + \alpha * (2 e_{ub} q_{kb} - \lambda(p_{uk}))$$

Improved Regularized SVD (add bias for user and businesses): $\hat{r}_{ub} = c_u + d_b + \sum_{k=1}^{K} p_{uk} q_{kb}$ trained as $c_u := c_u + \alpha * (\hat{r}_{ub} - \beta * (c_u + d_b - globalmean))$





NV-200-40: 91 users, 509 businesses, 5313 reviews.

NV-100-40: 357 users, 789 businesses, 16411 reviews.

NV-20-40: 3415 users, 1362 businesses, 86685 reviews.

AZ-200-40: 64 users, 238 businesses, 3044 reviews.

AZ-100-40: 309 users, 580 businesses, 15961 reviews.