Divergent Recommendations

Christopher Heung, Sigberto Alarcon, Shifan Mao

Problem Definition

Given a Yelp user, we want to give recommendations that are very different from what a user has previous tried. We call these 'divergent recommendations'. For example, given that I like Domino's Pizza, a current recommendation system might recommend Pizza my Heart. But perhaps I have tried every single topping combination and want to try something entirely different, like Ethiopian food. Our recommendation system will find and suggest the best divergent recommendations.

Preprocessing Data

Yelp Dataset

- Consists of 1.6 million reviews, 366k users, and 61k businesses

"Another favorite of mine is the barbecue chicken pizza, it has a great kick to it and it's big enough for two people"

NLP Techniques

- Bag of words of review texts minus stop words
- Learned weights of each word in vocab
- Created feature vectors based on tf-idf vectorization of the bag of words and optionally other metadata

$$\phi(u) = \{\text{tf-idf of u}\}$$

$$\phi(i) = \{\text{tf-idf of i, business metadata}\}$$

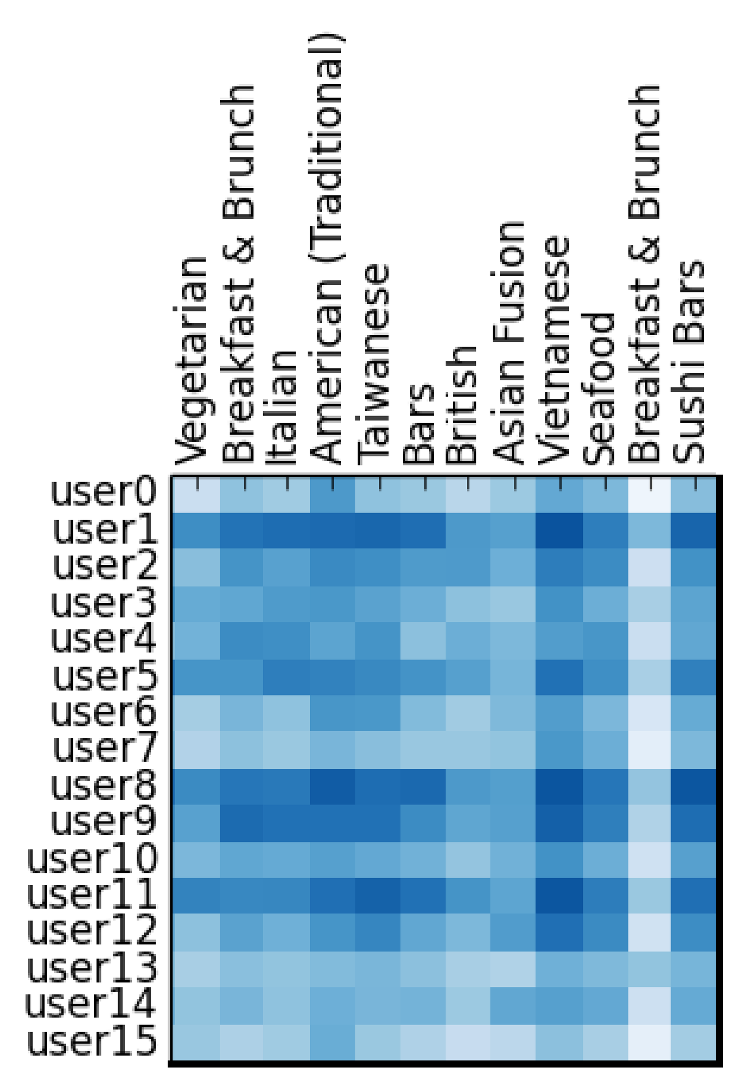
Similarity Metric

- Cosine Similarity

Given tf-idf_u and tf-idf_i,

$$\cos(\text{tf-idf}_u, \text{tf-idf}_i) = \frac{\text{tf-idf}_u \cdot \text{tf-idf}_i}{||\text{tf-idf}_u|| ||\text{tf-idf}_i||}$$

- The lower the similarity, the more divergent the recommendation
- We want to minimize the cosine similarity while maximizing how much we predict the user will enjoy a recommendation (measured through stars)

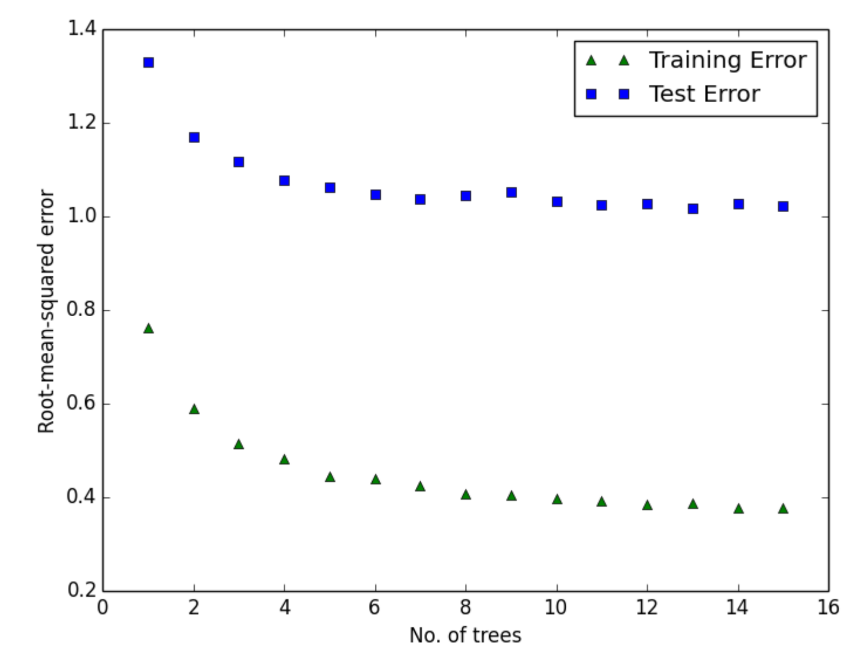


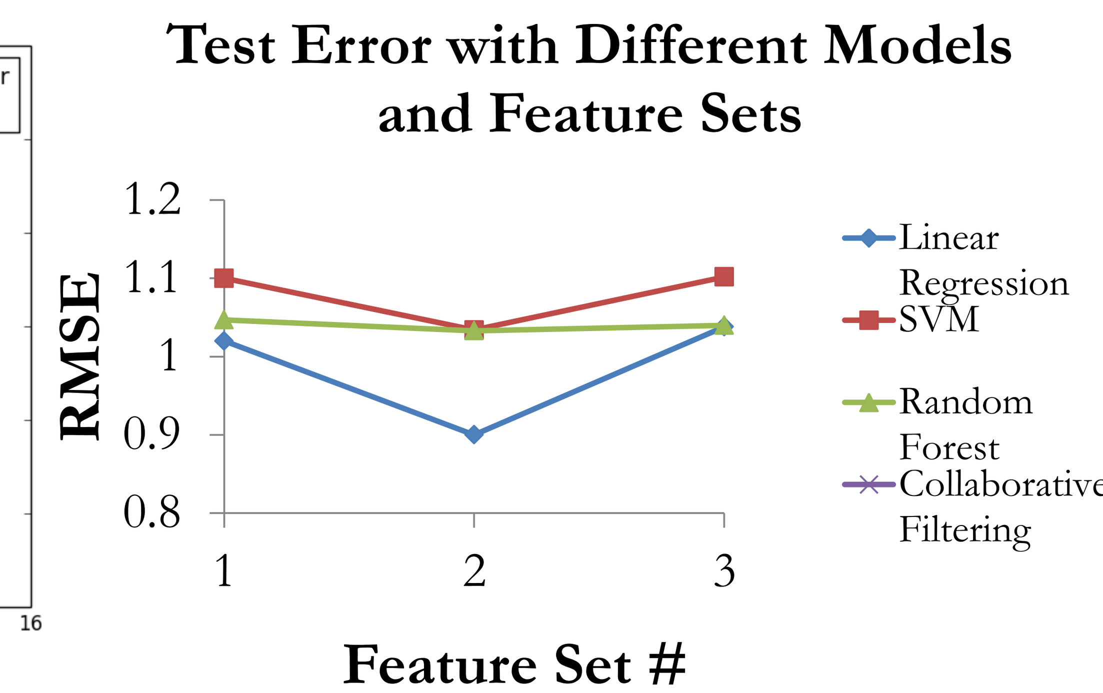
Methodology

To predict a user's rating, we learn the weights of each feature. We tried 4 different models: Linear Regression, SVM, Random Forests, and CF. We also had 2 different feature sets of varying complexity

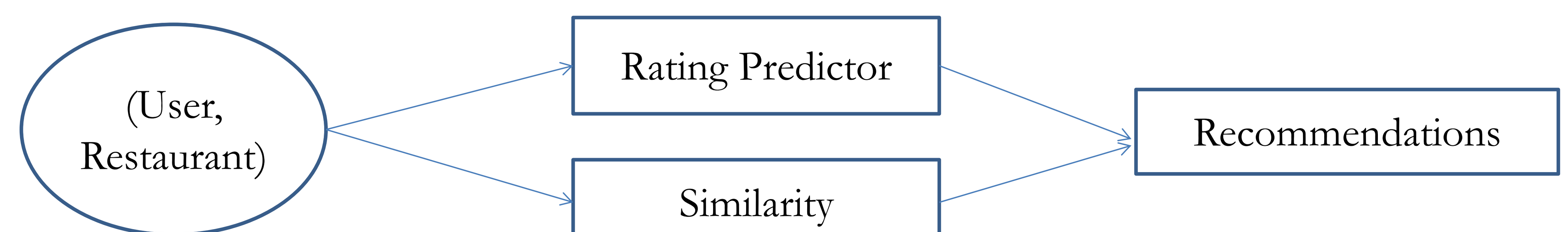
1) Linear Regression
2) Support Vector Machines
3) Random Forests
4) Collaborative Filtering

$$p_{u,i} = \bar{r}_u + \frac{\sum_{u' \in N} s(u,u')(r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} |s(u,u')|}$$





Making Divergent Recommendations



Incorporating similarity

- Compare user-to-restaurant similarity
- Can think of the 'best' recommendation in terms of whether the user likes exploring vs exploitation

$$\text{score}_{final} = \alpha p_{u,i} + (1-\alpha) sim(u,i) \text{ where } \alpha \text{ represents an exploration constant}$$

- If α is low, then we want to explore more than exploit. If α is high then we want to exploit more than explore

Future Work

1) Other Similarity Metrics – Pearson Correlation, KNN
2) Other Models – Neural Networks
3) Predicting Ratings as Classification – Naïve Bayes, Clustering, etc.
4) Content-based Filtering