BIDM Yelp Clustering Write-up

Peter Sadosky

One of our business topics for the Yelp dataset was to predict a user’s rating for a restaurant based on his or her previous restaurant ratings. The goal of the project was to make an informed recommendation about what restaurant the user would like to go to next. This approach requires a recommendation engine. A recommendation engine is a function that inputs manifest characteristics about a user and outputs a unique business record. The manifest characteristics might include fields like “user A gives high ratings to Italian restaurants” or “user B gives everything low ratings.” The unique business record is the recommended restaurant, which is the target of this study.

We implemented our recommendation engine using clustering. Each business record in the Yelp dataset has a number of variable fields associated with it, which are the fields used for clustering. These variables include the ordinal price of the restaurant, whether the restaurant takes reservations or not, and what category of food it falls under. Based on these characteristics, restaurants are clustered into smaller groups based on k-means clustering. The implementation was done in SAS EM, which automatically decides on an appropriate value for k based on the results of the cubic clustering criterion.

Once restaurants are assigned to the clustering space, an individual user’s past ratings are taken into account. This procedure is best described through an example:

Suppose Samantha has rated three restaurants on Yelp. She gave Everyday Noodles five stars, Ali Baba three stars, and Steel Cactus three stars. Now imagine that Everyday Noodles and Ali Baba are in cluster A and Steel Cactus is in cluster B. The first step in making a recommendation is deciding which cluster to base our decision on. We make this decision using the heuristic of choosing the cluster that has the combination of a high average rating and a high frequency for its constituent restaurants. Because the average rating for cluster A is four stars and the number of restaurants in it is two, and in comparison cluster B has a three star average and one restaurant in it, we choose cluster A. In the case of ties or edge cases where one cluster has a high rating but low frequency and another has a low rating but a high frequency, we make what we consider to be appropriate heuristics for deciding between two similarly-good clusters.

Now that we are in cluster A, we can throw out all of the restaurants that do not fall in the cluster. For the ones that are in the cluster, we have to decide which is the best for the recommendation. Because Everyday Noodles has five stars and Ali Baba only has three stars, we want to look around the area of Everyday Noodles. We now take into account the distance measure provided by SAS for how far apart restaurants are from one another within a cluster. We select the restaurant that is both close in distance to Everyday Noodles and also has a higher-than-average Yelp rating. This ensures that the restaurant is similar in some way to the user’s highest rated restaurant, and also has a consensus opinion on being a good restaurant.

This approach has applications in a number of interesting fields, and can certainly be studied further. There are a few approaches that we would like to implement in the future. We would like to take into account a user’s friends’ ratings, perhaps using a lower weighting but incorporating these ratings as it’s likely that if a user’s friend gave a restaurant a high rating, he or she too would like that restaurant. We would also like to look into uncovering more latent fields in the data, to discover what kinds of users fit what kinds of restaurants.