Application of Unsupervised Learning Techniques to Business Meta-Data, using Yelp Data

Eric Wang
Department of Statistics
Stanford University
eriwan@stanford.edu

Charles Zhang
Department of Statistics
Stanford University
cyzhang@stanford.edu

1 Introduction

When designing effective marketing strategies for products, marketers often consider the "marketing mix" – the value of the product, the price it might be offered at, how the product is promoted and the places where the product can be accessed or purchased. In competitive marketplaces, a better understanding of the elements of the marketing mix might lead to better marketing strategies and be incredibly valuable.

Our project attempts to better understand several aspects of the marketing mix in the context of firms that utilize brick-and-mortar stores. We think that the use of unsupervised techniques to learn, without strong assumptions, directly from the data may be able to provide valuable insights that help refine existing models or uncover questions previously overlooked in theoretical frameworks.

In this project, we apply unsupervised learning techniques to a dataset of brick-and-mortar businesses cataloged by Yelp in order to better understand the types of businesses that offer face-to-face services. Furthermore, we exploit geographical data and apply self-organizing maps to create "business neighborhoods" – clusters of businesses that are physically close to each other and far from other businesses – and examine the mixture and levels of different types of businesses within each of these "business neighborhoods. Finally, we examine whether the unsupervised clusters we obtain improve prediction of average Yelp stars for businesses.

In Section 2, we describe the raw data and our clustering dataset. In Section 3, we describe our clustering procedure and highlight the specific techniques that we used. In Section 4, we discuss our results and present visualizations.

2 Data

The dataset we used for our experimentation was provided by the Yelp Dataset Challenge (Sixth Round). This dataset includes data on $\sim 61K$ businesses from 10 different metropolitan areas from around the world. The dataset included rich and detailed categorizations of the various businesses categorizations ('restaurant', 'bar', 'shopping'), various attributes ('good for kids', 'good for groups', 'price range'), etc. In addition, it also provides Yelp data on Yelp user engagement (1.6M reviews, 500K tips, 61K aggregated check-ins). We limited the scope of our project to examining the two largest metropolitan areas in our dataset: Las Vegas, with $\sim 15k$ businesses and Phoenix, with $\sim 12k$ businesses. Furthermore, we restricted the scope of our inquiry to non-professional services – restaurants, bars, casinos and other entertainment – using the filters provided by Yelp.

2.1 Data Processing

To create our clustering dataset, we began by extracting and reshaping all of the categorical business data into a matrix of "attributes" where 1 indicated the appearance of the attribute. For example, there are attribute indicators for 'nightlife', 'bars', and 'Asian-fusion'. From the "check-in" data, we created day-of-the-week and hour-of-the-day features to capture time-of-day trends. In addition, for each business, we identified all users who had left a "tip" (a short review) about the business and then created aggregate features (median, average) from the user-level features of the users who had visited the business.

After creating these features, we then applied principal component analysis to subsets of the attribute matrix. We selected the principal components that captured 80% of the variance (we also looked for an 'elbow' in the proportion of variance explained) and used these as inputs to our clustering algorithm instead of the original attribute matrix features.[1]

Finally, for continuous variables (e.g., $review\_count$), we scaled these variables to have zero mean and unit variance in order to approximately balance the influence of different feature vectors.

3 Methodology

We used several unsupervised learning techniques in this project and we provide a brief outline of our procedure below.

3.1 Procedure

1. **Create Business Neighborhoods:** Given the geographical location of all of the businesses within a city, we wanted to be able to identify "business neighborhoods" where a group of businesses were geographically close to each other and, relatively, far from other clumps of businesses.

   To create these neighborhoods, we trained Self-Organizing Maps on the longitude and latitude of the businesses in each city (we explain the learning algorithm in more detail in section 3.3).

2. **Cluster Businesses:** Using our compiled business level data, we then performed K-Medoids clustering, choosing the number of clusters K by using the Gap Statistic.

3. **Aggregate Data on Business Clusters:** Next, we compiled summary statistics for the business clusters, such as average Yelp star rating, average number of reviews, and percentage of businesses falling into various attributes. We created a heat map to examine emergent patterns and arranged the clusters by generating a dendrogram (i.e., hierarchical clustering) of the clusters against these summary statistics.

4. **Cluster Business Neighborhoods:** Next, we examined the cross-tabulation of business cluster and neighborhood cluster membership for businesses in the Las Vegas area. We then performed an additional K-Medoids clustering (with each neighborhood cluster as a row, and features as the number of businesses belonging to each of the business clusters). We also assigned clusters to businesses in the Phoenix area and compared the geographical distribution to Las Vegas.

5. **Supervised Learning:** We then examined the usability of our business clusters in a supervised learning exercise using ϵ-support vector regression to predict Arizona business' average Yelp star ratings.

3.2 K-Means and K-Medoids

The K-Medoids algorithm[1] is a variant of the K-Means algorithm that we have discussed in class. Specifically, K-Medoids replaces the cluster centroids of the K-Means algorithm with cluster medoids. Whereas cluster centroids are calculated as the expected value of all points within a cluster, the cluster medoid is constrained to be an actual training example. The medoid can alternatively be described as the data point in the cluster that minimizes the reconstruction error. The use of the medoid is more suitable for our problem because it is more robust to extreme outliers and can more sensibly handle categorical variables translated into binary features.

3.3 Self-Organizing Map

Self-organizing maps[2] is a technique for unsupervised learning that can be used for online learning. In our application of self-organizing maps, we begin with a connected lattice of neurons that were initialized evenly over the city's geographical boundaries (indicated by the minimum and maximum longitude/latitude coordinates). We examined each business' location one at a time, found it's closest neuron and "pulled" the closest neuron and its neighbors in the direction of the training example. After presenting the dataset to the learning algorithm many times, we would eventually have stable neighborhood centers which would implicitly define a clustering (all of the businesses that are closest to a certain center are in one cluster).

[1] More specifically, we applied Gaussian kernel PCA after observing that a higher proportion of variance was explained with fewer principal components using this dimension reduction technique, relative to regular PCA.

Algorithm 1 Self-Organizing Maps (Online)

repeat ▷ Repeatedly present the training data to the algorithm
  for $i = 1 : m$ **do**
    $w_i \leftarrow arg\min_{n_j} ||x_i - n_j||_2^2$ ▷ Find the closest neuron
    for $n_j \in w_i \cup \Gamma(w_j)$ **do**
      $n_j \leftarrow n_j + \alpha\Theta(n_j, w_i)(x_i - n_j)$ ▷ Update the closest neuron, and its neighbors
    end for
  end for

In this algorithm, α is the learning rate, $\Theta(n_j, w_i)$ is a decay function that depends on the distance between n_j and w_i on the grid and $\Gamma(x)$ denotes the set of neighbors of x. In our implementation, we chose an alpha of 0.05, used a rectangular neighborhood and allowed the decay function to simply be a function that indicated membership within a radius r of w_i. We specifically chose to present our training data to the algorithm 500 times as we found that additional repetitions did not significantly affect the choice of final neuron positions.

3.4 Gap Statistic

When using an unsupervised learning technique for clustering, the correct number of clusters is often unknown. In the case of K-Means, the metric used to evaluate convergence – the within cluster sum of squares – generally decrease as you increase the size of k (e.g., consider that when $k = m$, you can achieve 0 error). The gap statistic[3] is a method to choose k in a principled way. Specifically, it compares a cluster's within-cluster sum of squares against the expected within-cluster sum of squares in a similarly sized space (i.e., if you generated m data points from a uniform distribution within the bounds of your data). The suggested choice of k is the first k that satisfies $Gap(k) \geq Gap(k+1) - s_{k+1}$ where:

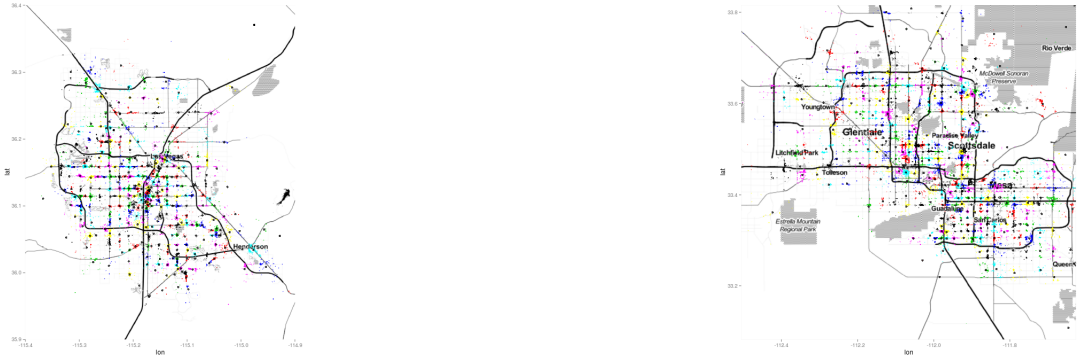
$$Gap(k) = \frac{1}{B}\sum_b log(W^*_{kb}) - log(W_k)$$

where W_k is the within-cluster sum of squares with k clusters, $\frac{1}{B}\sum_b log(W^*_{kb})$ is the estimated within-cluster sum of squares and s_k is the standard deviation of the estimated within-cluster sum of squares.[2]

4 Results

4.1 Business Neighborhoods

The generated neighborhoods for the Las Vegas and Phoenix metro areas are shown below.



(a) Las Vegas Metro Area

(b) Phoenix Metro Area

Figure 1: Business Neighborhoods Trained By Self-Organizing Maps

[2]We used the Gap Statistic to choose our number of clusters whenever we used k-medoids (we also checked for the robustness of these choices of k by examining silhouette plots).

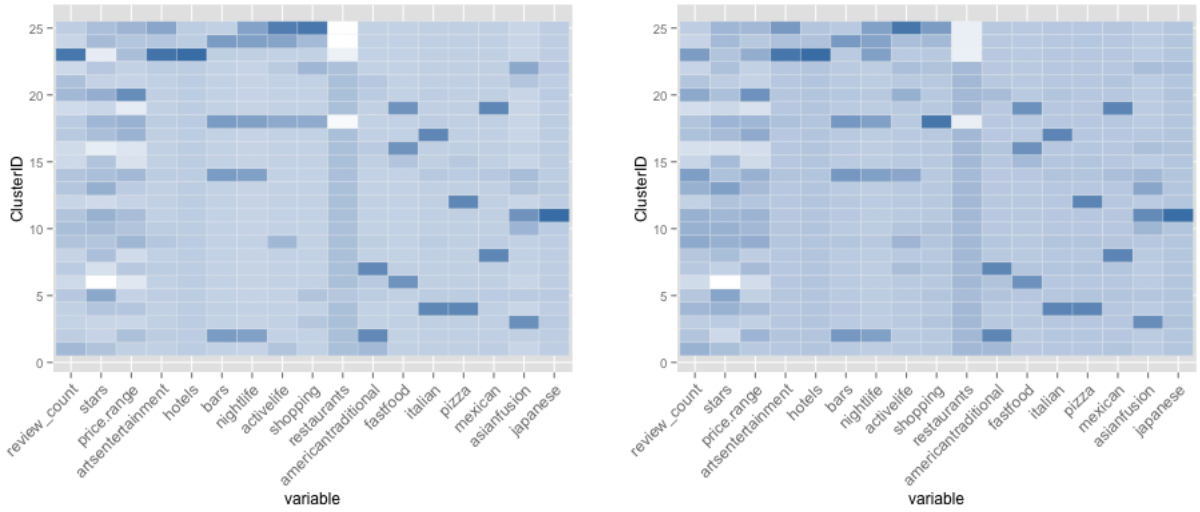
4.2 Clustering Businesses and Visualizing Clusters

After evaluating the gap statistic for the K-Medoids algorithm, we selected 25 clusters. We then generated a heat map for summary characteristics within each of the clusters, as shown below. The clusters are sorted vertically based on a dendrogram generated from hierarchical clustering the 25 clusters using the summary characteristics as features.

The heat map shows that our clustering has picked up relatively coherent clusters.[3] For example, clusters 23-25 represent a non-restaurant region of businesses that are hotels (cluster 23) and bars, nightlife, and shopping related (clusters 24-25). The row arrangements suggest that these non-restaurant clusters represent a super-cluster apart from other restaurant clusters. Moreover, the clusterings are not driven by single attributes. There are several fast food clusters as well as several Asian-fusion food clusters, for example, so that our algorithm has provided a differentiation of business types that are based on patterns across features.

Interestingly, there are also clusters that combine restaurants across different cuisines, suggesting that the unsupervised learning is leveraging other features such as the check-in and tips data.

We then assigned clusters based on this analysis to the Phoenix Metro Area as well to assess whether businesses in other regions adhere to a similar concentration around the medoids we generated. As shown in Figure 2, the average characteristics business in each cluster are consistent across regions.



(a) Las Vegas Metro Area

(b) Phoenix Metro Area

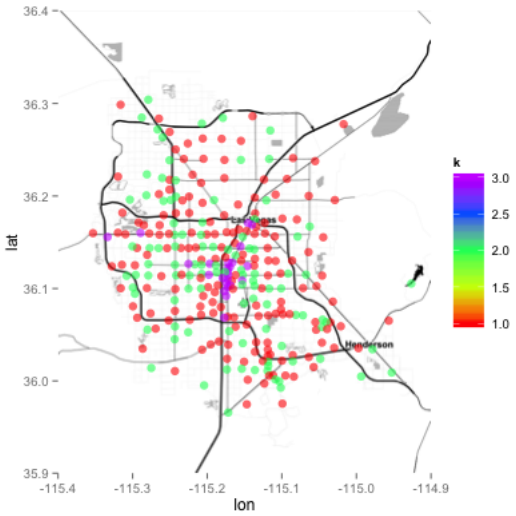
Figure 2: Summary Characteristics for Business Clusters

4.3 Cluster Business Neighborhoods

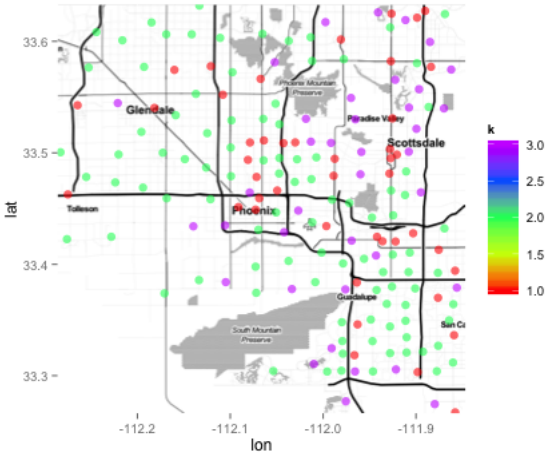
Next, we examined whether business neighborhoods had systematic patterns in the distribution of businesses across our 25 business clusters. For example, if certain neighborhoods were strongly affiliated with ethnic cuisines, or bars, we might be able to detect this through cross tabulating business neighborhoods with the number of businesses belonging in each of the business clusters. We used this matrix (500 neighborhoods x 25 business clusters) to cluster the neighborhoods, arriving at 3 clusters through K-Medoids and Gap Statistic analysis.

As shown below, we observe that the neighborhood appear to be clustered by the level of business density, with a downtown area clearly identified (magenta clusters). We also repeated this process for the Phoenix Metro Area, by assigning businesses to the 25 clusters generated from the Las Vegas data. The clustering here seems to be less interpretable, although once again high business density areas appear to be captured.

[3]We note that businesses are relatively evenly distributed across clusters, with the largest cluster including roughly a tenth of the observations.



(a) Las Vegas Metro Area - Clustering SOM-Clusters



(b) Phoenix Metro Area - Clustering SOM-Clusters

Figure 3: Grouping SOM-Neighborhoods by Business Cluster

4.4 Supervised Learning

Finally, we explored the usage of our business clusters as additional features in a supervised learning problem. We predicted average Yelp star rating for businesses in Arizona, utilizing our compilation of business attributes, check-in and tip data. We supplemented these features with home price data collected from Zillow as well as median household income data from the US Census Bureau. Finally, we leveraged the business clusters generated from the Las Vegas data by assigning the Arizona businesses to these clusters based on Euclidean distance. We then generated a set of indicators for cluster memberships to include in the predictive model.

We performed ϵ-support vector regression with a linear kernel, after tuning both ϵ and the cost of constraints violation parameter using 5-fold cross validation. The resulting RMSE for a test set of 850 businesses were very close, with and without the indicators for cluster membership (0.538 vs. 0.534, respectively), suggesting that the clusters do not improve this prediction problem beyond the constituent features. We also explored calculating distances of businesses to the cluster medoids, which did not result in predictive gains for this particular supervised learning problem.

5 Further Work

We note that our supervised learning analysis requires further exploration, since we only utilized one modeling technique (SVR). Other regularized learning methods such as LASSO would be worth exploring for example, or kernelizing the SVR. We also would want to directly cluster the Arizona businesses for feature creation (as opposed to applying clusters generated from Las Vegas). Furthermore, our business and neighborhood clustering may be better suited for other supervised learning problems, such as in improving recommendations to users.

Another potential avenue for further work is in enhancing our unsupervised learning with textual data from tips and reviews. Thus far, we have used attribute, check-in, and aggregated tips data; utilizing the tips and review language data would likely yield subtler, more refined clustering results for businesses. Finally, our usage of K-Medoids assumed that businesses fall into one category. An alternative clustering scheme would be a mixture model where businesses fall into overlapping groups.

References

[1] Leonard Kaufman and Peter J Rousseeuw. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis*, pages 68–125, 1990.

[2] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1):1–6, 1998.

[3] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.