Clustering of Cities by Craigslist Posts

Charles Johnson and Michael Kim, *Stanford University - CS 229*

14 December 2012

We aimed to evaluate the similarity between US cities by clustering them based on postings from the classifieds website Craigslist. Craigslist is a data source that could provide particular insight into the character of a city because the content is community-driven and contains unfiltered natural language. Our clustering was performed agnostic to geographic location so as to determine a more nuanced indicator of similarity. Here, we report our methods for creating feature vectors from the raw posts and our results from subsequently clustering the cities. We experimented with features that considered the metadata, categorical distribution, as well as the natural language of posts. After generating feature vectors for each city, we applied two clustering algorithms, k-means and mean shift, and then compared their outputs. Clustering with k-means produced fairly consistent, promising clusters, whereas clustering with mean shift did not produce any meaningful clusters. Our results from clustering could prove useful as an input to supervised learning problems that involve cities, or the clustering may be interesting as a qualitative metric in and of itself. feature vector.

Data Collection

To obtain data from Craigslist, we elected to use the 3taps API (`3taps.com`), an API for querying up-to-date and historical posts from all of Craigslist in a JSON format. The JSON response includes source data from the posts such as post titles, post bodies, and post categories, as well as additional annotations such as location codes at varying granularity.

For our exploratory work, we collected a dataset of the 11,000 most recent posts at the time of collection. We initially created feature vectors and clustered based on city codes (e.g. at the level of Palo Alto); however, as we expanded our dataset, we found that there were far too many city codes to work with and most of the cities were small, unrecognizable ones. Instead, we chose a list of 55 major metro areas based on "America's Best Cites" (Business Week 2012). With these 55 metro areas, we collected the 1000 most recent posts for each city, which served as the dataset for the work that follows.

Initial Clustering

As a first pass, we created feature vectors from the batches of 1000 posts by calculating three meta data: the average post length, the average title length, and the average number of images used per post. Instead of using these raw features, we normalized them in order to weight the features equally against one another. Otherwise, a feature which is naturally larger (like post body length compared to post title length) would artificially create larger distances between vectors. As a first pass, we simply normalized all values of a each feature dimension by dividing by its maximum observed value, moving the values of all features onto a $[0, 1]$ scale.

We used these three-dimensional feature vectors as input for the k-means clustering algorithm. Prior to actually running k-means, we plotted the average within cluster distance versus the number of clusters in order to determine a reasonable value for k. Based on figure 1, we chose the smallest reasonable value of k that minimizes the average within cluster distance. The curve is promising because it does have a natural "knee" where we can choose an optimal value for k[1].

[1] Stanford NLP website: "Cluster cardinality in k-means"

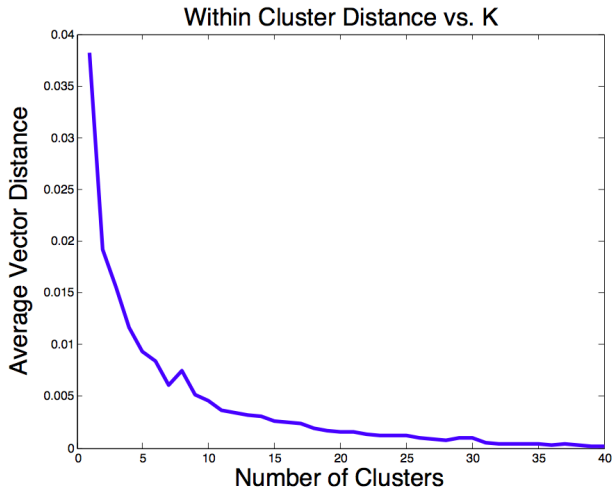

Figure 1: *Average Euclidean Distance from Nearest Centroids.*

Figure 2 shows an example run of k-means clustering on these initial feature vectors. In order to qualitatively analyze the clustering of the feature data, we performed principal component analysis on the feature vectors so that we could plot them in two dimensions. Qualitatively, the initial clustering seems reasonable but not fantastic. There are distinct clusters which appear to be reasonably grouped, but there certainly is not clear separation between the centroids.

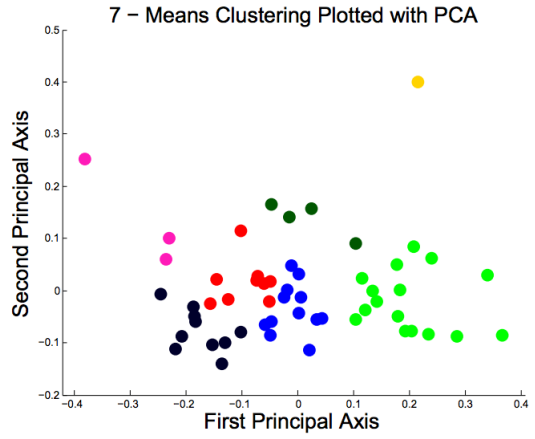

Figure 2: *7 - means clustering plotted with PCA*

Feature Expansion

Though the initial clustering results were favorable, looking exclusively at meta data ignores the potentially informative and valuable natural langauge and categorical information of the posts. Conceptually, the natural language and categorical data seem to be the likely sources of any intangible characteristics of the cities that might be similar and thus influence clustering.

Categorical Features

On Craigslist, posts are categorized under seven major buckets: 'For Sale', 'Vehicles', 'Housing', 'Discussion', 'Community', 'Jobs', and 'Services'. For each city, we calculated the percentage of its posts that were in each category. Furthermore, within each category for each city, we also calculated the percent of posts that were unpriced (i.e. free or not related to any transaction) and the average price of those posts that were priced. This created a total of 21 new feature dimensions.

LDA Features

In addition to the categorical features described above, we also computed features of the natural language used in the posts using an LDA topic model. We used the Stanford Topic Modeling Toolbox to run LDA on the post titles and bodies from each of the Craigslist categories, creating a set of topics discussed within each category. We experimented with various techniques to preprocess the post titles and bodies before running LDA, such as keeping or removing stop words and setting a minimum word length. The topics that seemed most promising were produced from posts that ignored case and punctuation and had a minimum word length of 5. We also filtered out stop words and the 20 most common words from each category and required that a word occur in at least 10 posts.

Upon inspection of the 'topics' generated by LDA, we observed that the model using these rules seemed to find vectors of words that seemed fairly inuitive. Once we had generated topics, we created feature vectors for each city by concatenating the average per-document topic distribution for each category. Because many cities had very few posts in the 'Discussion' and 'Community' categories, we did not include these in the expanded feature vector.

system	iphone	season	dryer	mattress
laptop	phone	level	washer	furniture
computer	samsung	stadium	kitchen	queen
power	galaxy	parking	silver	brand
audio	sprint	lower	jewelry	mattresses
video	verizon	upper	brand	spring
drive	screen	group	bedroom	patio
speakers	mobile	yards	sectional	frame
windows	phones	various	diamond	delivery
memory	repair	state	delivery	outdoor
digital	trade	thursday	leather	pillowtop
screen	unlocked	field	microfiber	plastic
sound	tmobile	football	electric	springs
remote	service	sunday	stainless	pillow
player	white	party	rings	quality
wireless	android	cheap	warranty	warranty
display	broken	monday	appliances	plush
processor	store	together	white	support
monitor	factory	weekend	steel	foundation
stereo	green	playoff	works	bedding

Figure 3: *Examples of topics from 'For Sale' Posts*

Clustering Results

In this section, we lay out the results of clustering the cities according to their expanded feature vectors. We used the same normalization technique as described above. We also experimented with other normalization methods throughout, but the results did not change significantly.

K-Means

Once again, we plotted the average within cluster distance versus the number of clusters.

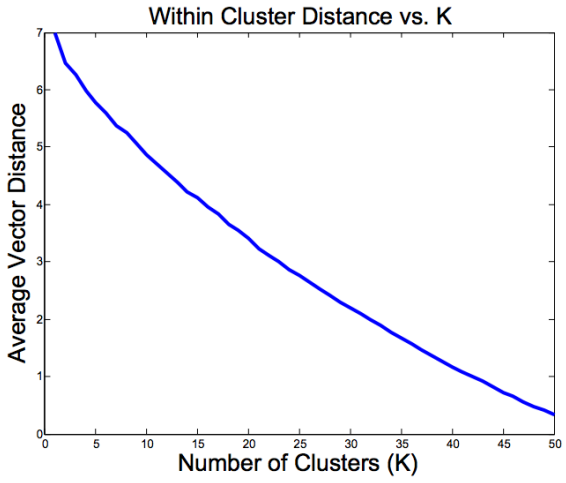

Figure 4: *Average Euclidean Distance from Nearest Centroids.*

As can be seen in Figure 4, this distance curve does not have a natural 'knee' as the initial distance curve had. Unfortunately, this makes a choice of k slightly more arbitrary. Nonetheless, after running the clustering with various values, 15 clusters seemed to work fairly well. Because we were working with such a high dimensional feature space (roughly $\mathbb{R}^{170}$), PCA was somewhat ineffective in visualizing the clusters. Instead, we used the tSNE algorithm for visualizing the clusters in a plane[2].

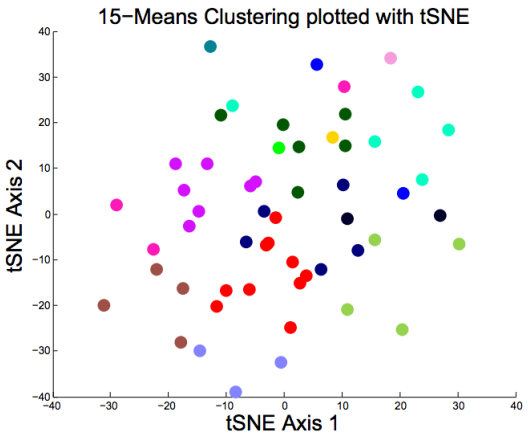

Figure 5: *k-means on expanded feature vector*

An example run of k-means is shown in Figure 4. The clusters seem to be reasonably distinct and well grouped. However, even with the tSNE algorithm, the high-dimensionality of the feature vectors makes the visual evaluation of the clustering quite difficult.

Mean Shift

Because the within cluster distance curve for k-means was not particularly compelling, we decided to experiment with the mean shift clustering agorithm[3]. Mean shift does not presuppose the number of clusters as an input to the algorithm but rather takes a bandwidth parameter which affects the convergence rate and number of clusters. Before implementing any sophisticated choice of bandwidth, we experimented with a range of values. We in turn found that there was no bandwidth parameter that produced any reasonable clustering. If the bandwidth parameter was too large, then there would be only one cluster, and if too small, there would be only singletons. Furthermore, in all of the intermediate values there was one dominant cluster and the remaining clusters would be all singeltons or memberless centroids. Figure 5

[2]Using a matlab implementation written by Dr. Laurens J.P. van der Maaten of Delft University of Technology

[3]Using Matlab File Exchange submission by Bart Finkston

shows an example plot of such. For this application, the mean shift algorithm proved a worse method than k-means.

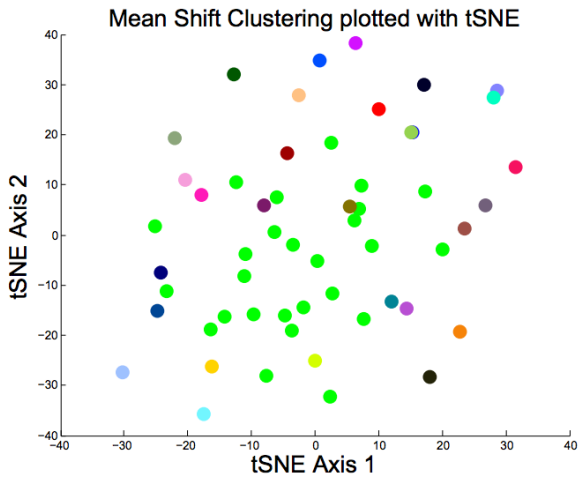

Figure 6: *Mean Shift on expanded feature vector*

Correlation through Iteration of K-Means

While the results from our trials of k-means seemed promising, the clusterings were unsatisfying for two main reasons. Firstly, the feature vectors that were used as input to k-means were in such high dimension that plotting the clusters in two or even three dimensions in a meaningful way proved nearly impossible. Secondly, the individual runs of k-means frequently produced slightly different clusters, possibly due to the random initialization of the algorithm and the lack of obvious separation of clusters.

To circumvent both these issues, we aimed to produce clusters with a greater confidence by grouping cities that were clustered across multiple runs. Instead of relying on a single run of k-means, ran the algorithm 100 times and looked at the frequency with which cities were clustered together. We generated 'pseudo-clusters' by grouping together cities that were clustered together more than 50% of the time. We found that these psuedo-clusters provide a more stable and reliable measure of whether cities are similar or not by executing this process multiple times and comparing the results. Figure 5 presents an example of a pseudo-clustering. While not graphed in a real space, this type of representation allows us to understand more about which cities are related to one another and also about how they are related. For instance, we note that while many of the

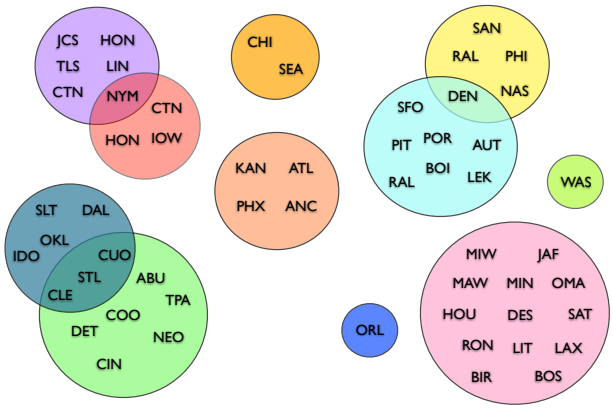

Figure 7: *Abstract representation of Pseudo-Clusters*

pseduo-clusters are self-contained, some contain overlaps with other pseudo-clusters. These overlaping cities represent cities that were strongly correlated to both groups. Even though they were not as highly correlated, we can still interpret this to mean that the two pseudo-clusters which share some cities, are more related than any two pseudo-clusters chosen at random.

Discussion

Using the small set of three meta data features, k-means clustering performed reasonably well. However, we were more interested in the rich categorical and natural language features of the posts. We expanded the feature vector to include categorical data as well as average LDA topic weights within each category for each city. The clustering on this expanded feature space was reasonably promising but certainly presented some drawbacks. Herein we discuss the results in more depth.

Though this new feature space is certainly more conceptually compelling for the purpose of evaluating the similarity of cities, it is generally known that the easiest clustering algorithms, including k-means, do not perform well for high-dimensional data[4]. The reason, briefly, is that with random initialization, it is possible to get stuck in a local optimum prior to converging to a global optimum. This is one plausible explanation for the less-than-ideal within cluster distance curve for the expanded feature vector. Ideally, the

[4]Ding et al.: "Adaptive dimension reduction for clustering high dimensional data"

distance curve would sharply descend, forming a bend in the curve near lower values of k, but in our case the curve was close to being linear, which suggests a non-optimal convergence for most values of k. More sophisticated algorithms for high-dimensional clustering have been developed, which we discuss in our section on future work.

The mean shift algorithm performed noticably worse than k-means. One possible hypothesis is that mean shift, using a Gaussian Kernel, is attempting to converge on distinct normal-like cluster distributions, and so with only 55 data points it is unlikely to have separate clusters that appear to have any normally distributed qualities about them simply because the separate clusters would be too sparse. Admittedly, a more thorough understanding of the mean shift algorithm is necessary to evaluate any hypotheses of its poor performance.

Lastly, we can ask ourselves what it means for two cities to be clustered together given our model of the feature vector. On first pass, the clusterering does not map to any obvious features of a city such as its geography, population size, wealth and economic activity. While it seems tempting to interpret this as a red flag–that something must be wrong with the clusters–in fact this might suggest that our clustering discovers some intangible similarity that can't be found with standard metrics. Testing the true quality of the clusters as a new metric of similarity would require further tests which are discussed below.

Future Work

The results of k-means clustering on the expanded feature vector are a promising start, but a number of steps need to be taken to improve the clustering method further as well as to numerically evaluate the results as we iterate on the method.

Before moving forward and applying more complicated clustering algorithms, an exciting and valuable metric would be to apply the clustering results as an input to an already explored supervised learning problem. An example problem could be predicting whether someone will enjoy visiting a particular city to which they have never been (via couch-surfing or a hotel stay). For this problem, it could be very relevant whether they have enjoyed or disliked visiting a similar city in the past, where the measure of similarity in this case could be based on the clusters we produced. With a supervised problem like this, we could quantitatively evaluate whether using the clustering as an input improved or degraded the accuracy of predictions.

Furthermore, with such a supervised framework we could also perform a more sophisticated analysis of our feature space using feature-wise cross-validation. In other words, we could remove a subset of the features (such as vehicle topics) and observe the effect on the performance in the supervised predictions. This could be a fantastic tool towards better understanding the feature space before embarking on any attempts to employ complicated high-dimensional clustering methods. Nonetheless, there is a decent amount of literature on k-means based high-dimensional clustering methods. After setting up the framework to numerically evaluate the clustering in a supervised setting and exploring the feature space with cross-validation, if the results continue to be promising it would certainly be worthwhile to attempt to implement k-means based algorithms specifically tailored for high-dimensional feature vectors.

Even further down the road, one exciting application of the methods we use here would be to track the clusters of cities throughout time. Because all Craigslist posts have a timestamp, extending the methods described here, we could track the relationships of cities throughout time, effectively viewing not only how cities are related but also how they trend and interact on a personal level.