Clustering The Zip Codes of California

Dongyu Lang
University of California, Berkeley
SID: 24174288
yaleldy@berkeley.edu

Billy Jiang
University of California, Berkeley
SID: 25341123
jiangxiaoyu@berkeley.edu

ABSTRACT

In this paper, We will classify the zip codes of California by comparing various clustering methods, including K-Means, Spectral Clustering, Agglomerative Clustering, DBSCAN and HDBSCAN. The application of clustering zip codes can be used in setting up warehouses, mail offices or even Starbucks.

KEYWORDS

Clustering

1 INTRODUCTION

Clustering has always been one important part of machine learning in recent years. The applications of clustering cover a large amount of areas, including Biology, Business, Computer Science, Social Science and Supply Chain, etc. The idea of this paper is to come up with an effective clustering method that captures population centers using zip codes of California. Especially, suppose that a company wants to set up six warehouses, and where should the locations be to best serve the customers around California, assuming that the demand is the same for each zip code. We can always make the number of clusters different, such as 1000; thus, the scenario can be constructing Starbucks. However, due to the complexity of computation, in this paper, We only assume that the number of clusters is 6; that is setting up 6 warehouses.

2 DATA DESCRIPTION

The data that I use is the 2010 US censuses dataset, which was collected by United States Census Bureau. The data can be found in its official website. The data consists of zip codes in each state, but in this paper, IWe only extract the zip codes of California. Besides, there are also latitude and longitude of each zip code.

Below, there is a graph (Figure 1) showing the locations of zip codes in California. We can notice that the zip codes are dense around areas like San Francisco,

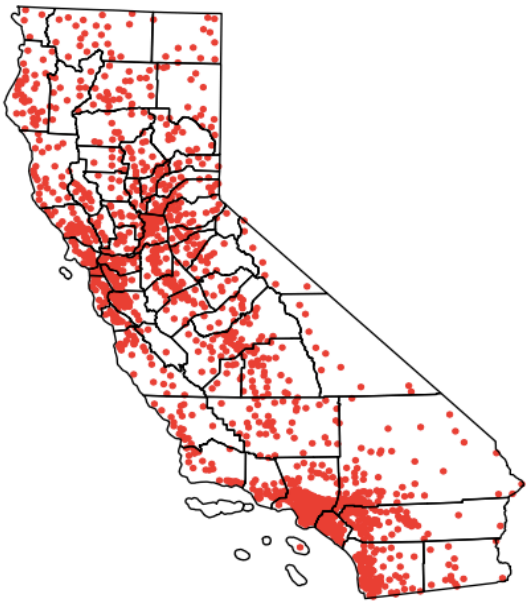

Figure 1. Plot of California with county boundary, red points represent the zip codes

Los Angeles and San Diego. Thus, we want to use clustering to successfully capture this feature.

3 METHODS AND RESULTS

3.1 K-Means

K-means clustering is one of the most popular methods people use in classification, because it is simple and requires less computation. The idea is that K-means aims to partition the n data points into k sets, which we denote as S = $\{ s_1, s_2, \ldots, s_k \}$. The objective is to minimize

$$\arg min \sum_{i=1}^{k} \sum_{x \in s_i} \|x - \mu_i\|^2$$

where μ_i is the mean of points in s_i. [1]

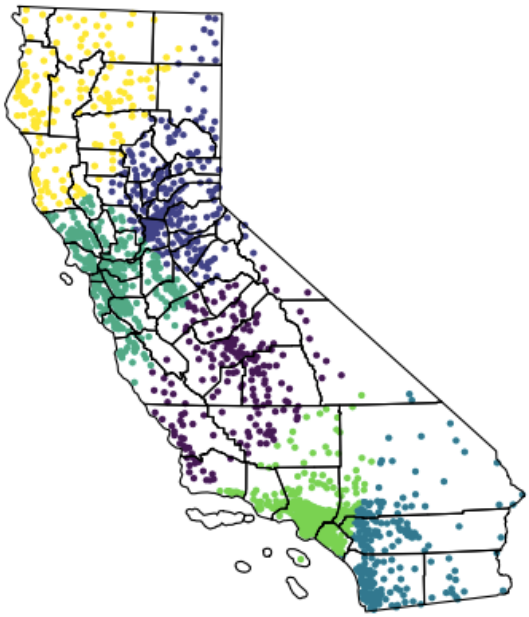

Figure 2. Clustering the zip codes of California using K-means.

From the result (Figure 2), the method performs reasonable. We can notice that there are 6 clusters representing the north part of California, Bay areas, areas around Sacramento, middle part of California, areas around Los Angeles and areas around San Diego.

Overall, K-means method clusters around either some important cities of California or geographically of California (North and middle).

3.2 Spectral Clustering

The intuition of this method is that, spectral clustering first performs dimensionality reduction on the data using the eigenvalues, then it performs clustering. Consequently, the main difference of spectral clustering and K-means is the space for partitioning. However, the method is computational expensive than K-means.

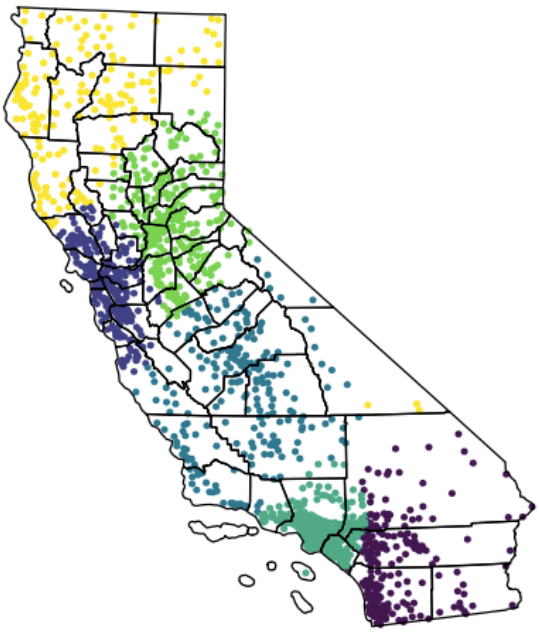

Figure 3. Clustering the zip codes of California using spectral clustering

Comparing to the result of K-means, there is not too much difference. One difference is that part of upper north of the zip codes, which belonged to Sacramento in K-means belongs to the North part of California now.

Also, there are some zip codes which are north of San Diego misclassified clearly to the north of California. The mistake might be happened in the process of dimensionality reduction.

Overall, the result still looks promising despite some minor mistakes.

3.3 Agglomerative Clustering

The idea of agglomerative clustering is that at first, each data point is a cluster of its own. Then, in each run, the algorithm merges one cluster with another cluster using some criteria. Finally, all the data points are merged into one huge cluster and become a binary tree. The we can trace back the tree to find the desired number of clusters.

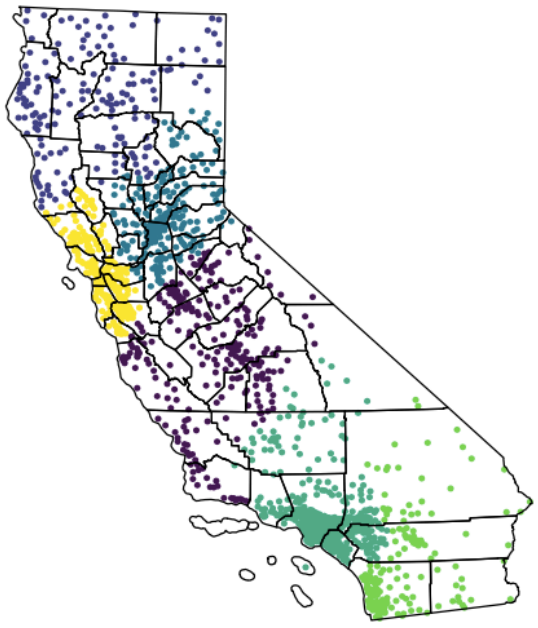

Figure 4. Clustering the zip codes of California using agglomerative clustering

Still, the result (Figure 4) is similar to that of K-means and spectral clustering. However, there is one part that is worth discussing about. We can notice that the part between Los Angeles and San Diego is different from the previous two methods. That part which was originally belongs to San Diego, now belongs to Los Angeles.

The reason can be that the cluster of that part is much closer to the cluster of Los Angeles than that of San Diego; thus, when using agglomerative clustering, that part belongs to Los Angeles.

3.4 DBSCAN

DBSCAN is a density based algorithm. The idea is that the algorithm looks for the dense region and extract the information and leave the other part of data as noise; thus, the algorithm do not require all data points to be assigned to a cluster.

This method has several benefits, we get the manifold following behavior of agglomerative clustering, and we get actual clustering as opposed to partitioning. Better yet, since we can frame the algorithm in terms of local region queries we can use various tricks such as kdtrees to get exceptionally good performance and scale to dataset sizes that are otherwise unapproachable with algorithms other than K-Means. [2]

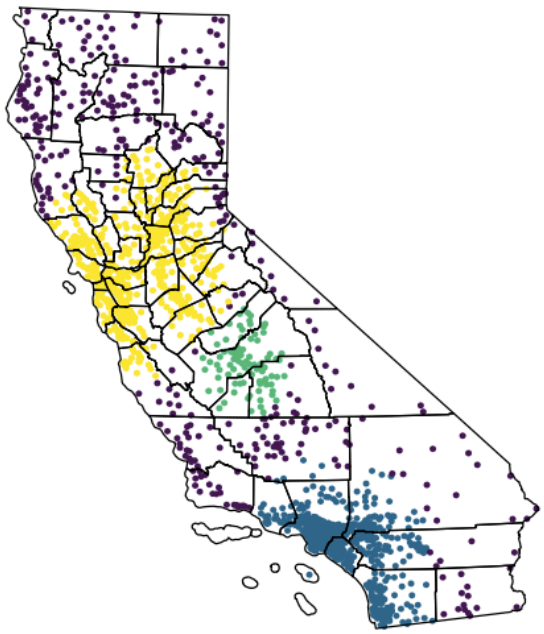

Figure 5. Clustering the zip codes of California using DBSCAN

The result (Figure 5) is interesting for DBSCAN. The purple points are treated as noises. Unlike previous three methods, not all zip codes are in a cluster. Only the zip codes in dense region are classified in a cluster. We can notice that there are only three clusters. The yellow one is the Bay areas and areas around Sacramento. The green part is centered around Fresno, and finally, the dark blue part consists of areas around Los Angeles and San Diego.

DBSCAN performs good on extracting the dense area; however, it cannot further separate the clusters. For example, it

leaves Los Angeles and San Diego together and also bay areas and Sacramento together.

3.5 HDBSCAN

The main difference between HDBSCAN and DBSCAN is that HDBSCAN allows varying density clustering. It performs DBSCAN over varying epsilon values and integrates the result to find a clustering that gives the best stability over epsilon. This allows HDBSCAN to find clusters of varying densities (unlike DBSCAN), and be more robust to parameter selection. [3]

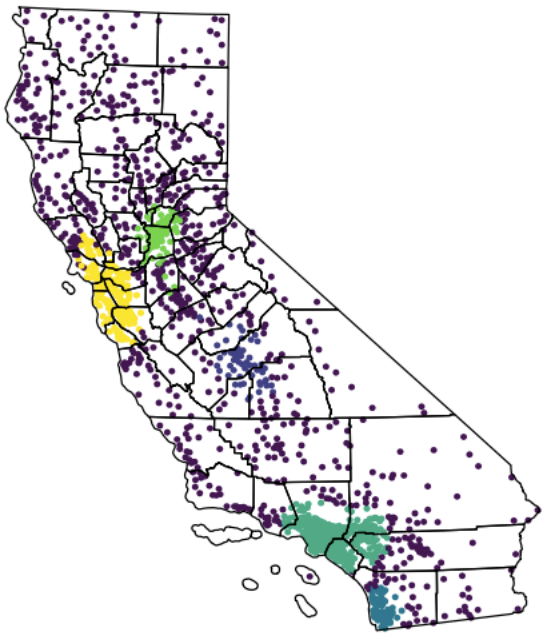

Figure 6. Clustering the zip codes of California using HDBSCAN

The purple points are considered as noises. In HDBSCAN, we can notice that there are more clusters than that in DBSCAN, since the model assumes varying density clustering.

The algorithm clearly shows 5 clusters, which are bay area, area around Sacramento, Fresno, Los Angeles and San Diego. Thus, HDBSCAN further divides the cluster in DBSCAN and performs better.

4 Conclusion

We have compared 5 different methods. K-means, spectral clustering and agglomerative clustering generate about the same results. For DBSCAN and HDBSCAN, we can extract the dense region and leave the other parts as noises. The performance of HDBSCAN is better than DBSCAN, since it can separate the clusters clearly.

5 Future works

There are two more objectives that we can achieve. First, we can perform clustering method on a larger data set, for example, the zip codes of Unite States, and compare different methods. Second, for each zip code, the population is different from others, we can perform a weighted K-means, and compare the results.

6 REFERENCE

[1]. “K-Means Clustering.” Wikipedia, Wikimedia Foundation, 30 Apr. 2018, en.wikipedia.org/wiki/K-means_clustering.

[2]. “Comparing Python Clustering Algorithms¶.” How HDBSCAN Works - Hdbscan 0.8.1 Documentation,hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html#affinity-propagation.

[3]. McInnes et al, (2017), hdbscan: Hierarchical density based clustering, Journal of Open Source Software, 2(11), 205, doi:10.21105/joss.00205