

## Dimension Reduction

Lecturer: Xiyuan Cheng

Scribes: Dev Dabke and Andrew Cho

# 1 Introduction

The lecture covered

- A discussion KK10 paper on consistency of Lloyd's k-means algorithm
- An introduction to spectral clustering

## 2 k-means clustering

### 2.1 Lloyd's algorithm

Recall Lloyd's algorithm from last lecture that given some data points as input  $\{x_i\}_{i=1}^n$ , we want to search for  $k$  clusters  $C = \{C_1, \dots, C_k\}$  with "centers"  $\mu = \{\mu_1, \dots, \mu_k\}$ .

#### Definition 2.1: Objective Functions

Depending on our constraints, also recall we can choose different objective functions and have different interpretations of what these clusters represent.

$$\begin{array}{ll} \min_{C, \mu} \sum_{l=1}^k \sum_{i \in C_l} \|x_i - \mu_l\|_2^2 & \text{k means} \\ \min_{C, \mu} \sum_{l=1}^k \sum_{i \in C_l} \|x_i - \mu_l\|_1 & \text{k medians} \\ \min_{C, \mu} \sum_{l=1}^k \sum_{i \in C_l} \|x_i - \mu_l\|_2 & L_2 - L_1 \text{ norm} \end{array}$$

We can either choose initial seeds by randomly selecting  $k$  points or by singular value decomposition (SVD). Ultimately, we are interested in knowing if these results are *consistent*. Can these seeding methods lead to the true centroids?

### 2.2 Strong Law of Large Numbers

#### Definition 2.2: Laplacian

$$\mathcal{L}_n(\mu) = \int \|x - \mu\|^2 dP_n(x)$$

where

$$dP_n(x) = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}(x) dx$$

which is the empirical measure of the data.

### Definition 2.3: Element-Set Norm

Where  $A = \{a_1, \dots, a_n\}$  for all  $x \in \mathbb{R}$ , we define  $\|x - A\|$  to be

$$\|x - A\| = \min_{1 \leq i \leq n} \|x - a_i\|$$

### Theorem 2.1: Strong Law of Large Numbers

As  $n \rightarrow \infty$

$$\begin{aligned} \mathcal{L}(\mu) &= \int \|x - \mu\|^2 dP(x) \\ \min_{\mu} \mathcal{L}_n(\mu) &\rightarrow \min_{\mu} \mathcal{L}(\mu) \end{aligned}$$

## 2.3 Consistency of Lloyd

If we satisfy some requirements of “separation,” then we can provide some bound for misclassification, assuming some true centroid partition. [KK10]  $\|\mu_k^* - \mu_l^*\|$  if  $k \neq l$  needs to be bigger than a gap  $\Delta_{kl}$ .

$$\Delta_{kl} > c \frac{\sigma_l}{\sqrt{w_{min}}} \quad w_{min} = \min_l \frac{|C_l^*|}{n}$$

$w_{min}$  denotes proportions of points in cluster  $l$ . Letting  $\sigma_k^2$  be the variance of the data in the  $k^{th}$  cluster and if we can assume that  $\sigma_k \approx \sigma_l$ , then we know that misclassification error from SVD initialization occurs less than some  $\epsilon * n$  with high probability.

## 2.4 In Practice

There are some practical considerations for k-means.

1. What is  $k$ ?
2. How to choose an initial seed
3. Some cases of k-means fails
  - (a)  $\sigma_k$  might be too large compared to separation
  - (b) Clusters might be too small (i.e. cluster sizes may not be balanced)
  - (c) Cluster might be convex and piecewise can't do concave (Voronoi)

## 3 Spectral clustering

### 3.1 Graph Laplacian

Given some data  $\{x_i\}_{i=1}^n$  and  $k$ , we want to

1. Build positive-definite, symmetric affinity matrix  $W_{n \times n}$  by  $k$  nearest neighbors,  $\epsilon$  - neighbor, or the Gaussian kernel  $W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\epsilon}}$ .
2. Consider the eigenvalue decomposition of the graph Laplacian  $\mathcal{L}$ . Note that

$$L_{un} = D - W \quad \text{Unnormalized}$$

$$L_{rw} = D^{-1}(D - W) \quad \text{Shi-Malik '00}$$

$$L_{sym} = I - D^{-1/2} W D^{-1/2} \quad \text{Ng-Jordan-Weiss '02}$$

3. Apply  $k$  means to  $\Psi = [\varphi_1, \dots, \varphi_k]_{n \times k}$  and denote  $y_i$  as the  $i^{th}$  row of  $\Psi$ .

#### Remark 3.1: Truncation

If  $k = 2$ , you can use truncation because the first eigenvalue is constant. Thus, you can use  $\text{sign}(\Psi_2)$  to indicate clustering.

#### Definition 3.1: Connected Components

Let  $\mathcal{G} = (V, E)$  such that on each edge  $(i, j) \in \mathcal{G}$  that  $w_{ij} > 0$ . If node  $i$  is connected to node  $j$ , there is a path from  $i$  to  $j$ . So then, set  $A$  is a connected component if every pair of  $i$  and  $j$  is connected and  $A$  is the maximum set that satisfies this condition to preserve connectivity.

#### Theorem 3.1: Eigenspace of $\lambda = 0$ of $\mathcal{L}$

Suppose the graph has  $k$  connected components  $A_1, \dots, A_k$ . Then the eigenspace of  $\lambda = 0$  of dim  $k$  is spanned by

$$\{\mathbf{1}_{A_1}, \dots, \mathbf{1}_{A_k}\}$$

where

$$\mathbf{1}_{A_i}(i) = \begin{cases} 1 & i \in A \\ 0 & \text{otherwise} \end{cases}$$

#### Proof 3.1: Eigenspace of $\lambda = 0$ of $\mathcal{L}$ , Theorem 3.1

Suppose  $\mathbf{f}$  is an eigenvector with  $\lambda = 0$ . So  $\mathcal{L}\mathbf{f} = \mathbf{0}$  and  $\mathbf{f}^T \mathcal{L}\mathbf{f} = 0$ . We also know  $\mathbf{f}^T \mathcal{L}\mathbf{f} = \mathbf{0} = \frac{1}{2} \sum_{(i,j)} w_{i,j} (f_i - f_j)^2$  and  $\mathbf{f}^T \mathcal{L}\mathbf{f} = \mathbf{0} \Leftrightarrow f_i = f_j$  whenever  $w_{i,j} > 0$ . Thus,  $\mathbf{f}$  is a piecewise constant in each of the connected components. Meanwhile, for each  $v \in \text{span}\{\mathbf{1}_{A_1}, \dots, \mathbf{1}_{A_k}\}$ ,  $\mathbf{v}^T \mathcal{L}\mathbf{v} = 0$ .

#### Exercise

- Does this generalize to  $\mathcal{L}_{rw}$  and  $\mathcal{L}_{sym}$ ?
- What about consistency of spectral clustering?

## 4 Applications

### 4.1 Spectral vs. k-means Clustering

When should we use spectral clustering and when should we use  $k$ -means clustering? One prototypical example is with spirals.  $k$ -means excels when the clusters are assumed to be dense point clouds separated from other dense point clouds. Spectral clustering works well where clusters may have many points with small local distances (with some metric), i.e. concentric circles.

To analyze this situation, we can use R to study this behavior.

```
1 ## Spiral example
2 library(kernlab)
3 library(mlbench)
4
5 obj = mlbench.spirals(100,1,0.025)
6 my.data = 4 * obj$x
7
8 # spectral clustering
9 plot(my.data)
10 sc = specc(my.data, centers=2)
11 plot(my.data, col=sc, pch=4)
12
13 # kmeans
14 clusters = kmeans(my.data, 2, iter.max = 10, nstart = 1, algorithm = "Lloyd", trace=FALSE)
15 plot(my.data, col = clusters$cluster)
16
17 ## Two clumps example
18 x = rbind(matrix(rnorm(100, sd = 0.2), ncol = 2),
19           matrix(rnorm(100, mean = 1, sd = 0.2), ncol = 2))
20 colnames(x) = c("x", "y")
21
22 # spectral clustering
23 sc = specc(x, centers = 2)
24 plot(x, col = sc)
25
26 # kmeans
27 clusters = kmeans(x, 2)
28 plot(x, col = clusters$cluster)
```

sc\_kmeans\_example.R

### 4.2 Using k-means on Images

We can also use  $k$ -means to compress an image.

```
1 library(png)
2 library(dplyr)
3
4 photo = readPNG("raccoon.png")
5
6 get_kmeans_centers = function(locs, k){
7   # choose two random centers to start
8   centers = sample(locs, k)
9
10  counter = 1 # just curious how many iterations it takes to converge
11
12  avgdiff = 1
13  while(avgdiff > 0.01){
14    # find distance
15    dist = list()
16
17    for (i in 1:k){
18      center = centers[i]
19      dist[paste0("group",i)] = list(sapply(locs, function(pixel) abs(pixel-center)))
20    }
21
22    dist_m = bind_rows(dist) # distance matrix
23    clusters = apply(dist_m, 1, which.min) # cluster assignments
24  }
```

```

25     new_locs = as.data.frame(cbind(locs, clusters))
26     new_centers = tapply(new_locs$locs, new_locs$clusters, mean)
27
28     avgdiff = sum(abs(centers-new_centers))/k
29
30     locs = new_locs$locs
31     centers = new_centers
32
33     counter = counter + 1
34 }
35
36 res = list("cluster" = clusters, "centers" = centers)
37
38 return (res) # returns a list of group assignments and mean for each group
39 }
40
41 compress = function(k, data){
42     v = as.vector(data)
43     km = get_kmeans_centers(v, k)
44
45     # quantization
46     compressed_m = matrix(km$cluster, ncol = dim(data)[2])
47     book = km$centers
48
49     print (book*256) # print the centers in terms of 256 scale for reference
50
51     # plotting
52     compressed_v = as.vector(compressed_m)
53     new_v = rep(0,length(v))
54     for (i in 1:length(v)){
55         new_v[i] = book[compressed_v[i]]
56     }
57     new_m = matrix(new_v, ncol = dim(photo)[2])
58
59     # rotating clockwise (R plots this weird)
60     new_m = apply(new_m, 2, rev)
61     image(1:1024, 1:768, t(new_m), col = gray((0:32)/32), useRaster = TRUE)
62 }
63
64 original = readPNG("raccoon.png")
65 original = apply(original, 2, rev)
66 image(1:1024, 1:768, t(original), col = gray((0:32)/32), useRaster = TRUE)
67 title(main = "Original")
68
69 # testing k = 2
70 compress(2, photo); title(main = "k = 2")
71
72 # testing k = 4
73 compress(4, photo); title(main = "k = 4")

```

kmeans\_compression\_example.R

We can, as an example, compress the Raccoon image in Figure 1.



Figure 1: A black and white image of a raccoon.

### 4.3 Basketball Position Data Analysis

Spectral clustering can also be used with basketball position data.

## References

- [KK10] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS '10*, pages 299–308, Washington, DC, USA, 2010. IEEE Computer Society.