STAT 3690 Lecture 34

zhiyanggeezhou.github.io

Zhiyang Zhou (zhiyang.zhou@umanitoba.ca)

Apr 22, 2022

Clustering

- Problem: given observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ group the observations into K populations
  - Unknown K
  - Unsupervised: no label/training data
- Why
  - Summarize a representation of the full data set
  - Exploration for structure of the data
  - Checking the validity of pre-existing group assignments
  - Assistance for prediction: sometimes clustering prior to prediction
- Clustering $C : \mathbb{Z}^+ \to \mathbb{Z}^+$
  - $C(i) = k$: assign $\boldsymbol{x}_i$ to group k

K-means

- Within-cluster scatter

$$W(K) = \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i:C(i)=k} \sum_{j:C(j)=k} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 = \sum_{k=1}^{K} \sum_{i:C(i)=k} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k\|_2^2$$

  - $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$: the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
  - $\bar{\boldsymbol{x}}_k = n_k^{-1} \sum_{i:C(i)=k} \boldsymbol{x}_i$
  - Smaller $W(K)$ is better

- (Approximately) minimizing the within-cluster scatter

$$\min_{C} W(K) = \min_{C, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_K} \sum_{k=1}^{K} \sum_{i:C(i)=k} \|\boldsymbol{x}_i - \boldsymbol{c}_k\|_2^2$$

- Implementation:

  1. Specify K and start with an initial guess for $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$, then repeat
     a. Labeling each point based the closest center: for each i, put $\boldsymbol{x}_i$ to the kth cluster such that $\boldsymbol{c}_k$ is closest to $\boldsymbol{x}_i$
     b. Replacing each center by the average of points in its cluster: for each k, take $\boldsymbol{c}_k = \bar{\boldsymbol{x}}_k$
  2. Terminate when $W(K)$ doesn't change

- Comments

  - Always converge
  - No guarantee to lead to the smallest W
  - Depend on K and initial cluster centers
    * Typically run K-means multiple times and pick up the result with the smallest W
- Determination of K
  - Between-cluster variation

$$B(K) = \sum_{k=1}^{K} n_k \|\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}\|_2^2$$

    * $\bar{\boldsymbol{x}} = n^{-1} \sum_{i=1}^{n} \boldsymbol{x}_i$
  - CH index (Caliński & Harabasz (1974), *Communications in Statistics*, 3:1–27)

$$\text{CH}(K) = \frac{B(K)/(K-1)}{W(K)/(n-K)}$$

  - To choose K as the maximizer of CH(K), i.e.,

$$\widehat{K} = \arg \max_{K \in \{2, \ldots, K_{\max}\}} \text{CH}(K)$$

- Example (`iris`)

```
set.seed(3690)
options(digits = 4)
x = iris[, !(names(iris) %in% c('Species'))]
y = (iris$Species == unique(iris$Species)[1]) +
  2*(iris$Species == unique(iris$Species)[2]) +
  3*(iris$Species == unique(iris$Species)[3])
decomp = prcomp(x)
s = 2
PCscores = decomp$x[,1:s]
K = 3; cols = c("red","darkgreen","blue","pink","purple")
km = kmeans(PCscores, centers=K, nstart=100, algorithm="Lloyd", iter.max = 100)
# cluster plot with centers
plot(PCscores,col=cols[km$cluster]); points(km$centers,pch=19,cex=2,col=cols)
# comparison with true groups
par(mfrow=c(1,2)); plot(PCscores,col=cols[km$cluster], main="K-means"); plot(PCscores,col=cols[y], main

# determine K
Ks = 2:20
Ws = numeric(length(Ks))
Bs = numeric(length(Ks))
CHs = numeric(length(Ks))

for(l in 1:length(Ks)){
  km = kmeans(PCscores, centers=Ks[l], nstart=100, algorithm="Lloyd", iter.max = 100)
  Ws[l] = km$tot.withinss
  Bs[l] = sum(km$size * rowSums(sweep(km$centers, 2, colMeans(PCscores))^2))
  CHs[l] = (Bs[l]/(Ks[l]-1))/(Ws[l]/(nrow(PCscores)-Ks[l]))
}
plot(Ks, CHs,
  type="b", pch = 19,
  xlab="Number of clusters K",
  ylab="CH index")
```

- An application to image compression/color quantization
  - Basic idea: compress images by reducing the color palette of an image to K colors

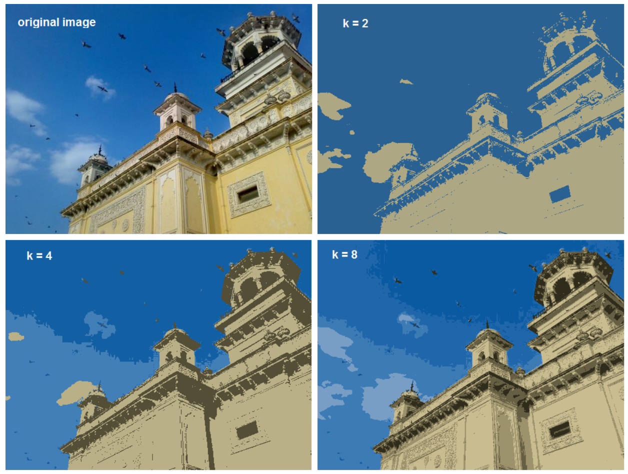

Figure 1: Image compression with K-means clustering (http://opencvpython.blogspot.com/2012/12/k-means-clustering-2-working-with-scipy.html)