K-means, K-Medoid, Mean-shift Clustering

Pattern Recognition, Jul-Nov 2019

Indian Institute of Technology Madras

September 6, 2019

K Means Clustering

1. Randomly initialize centroids of K Gaussian
2. Assign data-points to to C_1, C_2 or C_3 according to each one's distance
3. Reestimate means.
4. Repeat 2 & 3 until convergence. (after that there is no point in changing the clusters)

$$\begin{aligned}\vec{\mu_k} &= [\mu_{k1}, \mu_{k2} \dots \mu_{kd}]^t \\ &= \quad \text{mean of } k^{th} \text{ cluster}\end{aligned}$$

Our goal is :

- To find an assignment of data-points to clusters
- To find μ_k such that sum of distances of each data point closest to the $\vec{\mu_k}$ is minimum

Let us define γ_{nk} for each $\vec{x_n}$ such that $\gamma_{nk} \in \{0, 1\}$

$$\gamma_{nk} = 1, \text{if } \vec{x_n} \text{ belongs to } k^{th} \text{ cluster}$$
$$\gamma_{nj} = 0, \text{if j} \neq \text{k}$$

Distortion

$$D = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \|\vec{x_n} - \vec{\mu_k}\|^2 \quad \text{objective is to minimise D.}$$

But we need to address few issues before moving forward.

- γ_{nk} is not known
- More variables than equations

$$
\begin{aligned}
D &= \mathbf{E}\left[\mathrm{d}(\vec{x_n}, \vec{\mu_k})\right] \\
&= \sum_{k=1}^{K} \int_{\vec{x_n} \in C_k} \mathrm{d}(\vec{x_n}, \vec{\mu_{k(\mathrm{old})}})\ \mathrm{p}(\vec{x_n}, \vec{\mu_{k(\mathrm{old})}})\ \mathrm{d}\vec{x} \\
&= \sum_{k=1}^{K} \mathrm{P}(\mu_k) \int_{\vec{x_n} \in C_k} \mathrm{d}(\vec{x_n}, \vec{\mu_k})\ \mathrm{p}(\vec{x_n}/\vec{\mu_k})\ \mathrm{d}\vec{x} \\
&= \sum_{k=1}^{K} D_k
\end{aligned}
$$

$q(\vec{x_n}) = \vec{\mu_k}$ if $\vec{x_n} \in C_k$ also called as vector quantization

happens if $= \mathrm{d}(\vec{x_n}, \vec{\mu_k}) \leq \mathrm{d}(\vec{x_n}, \vec{\mu_j})$ $k \neq j$ $1 \leq k, j \leq K$

Derivating w.r.t μ_k

$$\nabla_{\mu_k} D = \frac{1}{N} \sum_{\vec{x_n} \in C_k} \nabla_k (\vec{x_n} - \vec{\mu_k})^t (C_k^{-1})(x - \vec{\mu_k})$$

$$\sum_{\vec{x_n} \in C_k} \vec{x_n} = N_k \vec{\mu_k}$$

$$\vec{\mu_k} = \frac{1}{N_k} \sum_{\vec{x_n} \in C_k} \vec{x_n}$$

$$C_k = \frac{1}{N_k} \sum_{\vec{x_n} \in C_k} (\vec{x_n} - \vec{\mu_k})(\vec{x_n} - \vec{\mu_k})^t$$

K-Means Clustering

We need to prove that $D_{i+1} \leq D_i$ where D_{i+1} is the Distortion in $(i+1)^{th}$ iteration
Let $\vec{Z} \in \{\vec{z_1}, \vec{z_2} \ldots \vec{z_k}\}$ and $\mathrm{L}(\vec{x_i}, \vec{z_k})$ be an assignment that minimises distortion.

$$D_i = E\left[L(\vec{x_n}, \vec{z_k})\right], n = 1, 2 \ldots N$$

$\vec{Z_k}$ set of prototypes, k = 1,2 ... K
The $(i+1)^{th}$ distortion is getting computed using z_k of the i^{th} iteration
Now, Using Euclidean Mean

$$E[\vec{z}] = \frac{1}{2}\sum_{n=1}^{N}((\vec{x_n} - \vec{z_{nk}}_{S_{ni}(\vec{z})}))^2$$

$S_{ni}(\vec{z})$ is the closest centroid to example $\vec{x_n}$ in i^{th} iteration.

But we don't have $S_i(\vec{z})$ so we define an auxilary function:

$$Q(\vec{z}, \vec{z'}) = \frac{1}{2} \sum_{n=1}^{N} (\vec{x_n} - \vec{z'}_{S_i(\vec{z})})^2$$

Now derivating $Q(\vec{z}, \vec{z'})$ w.r.t $\vec{z'_k}$ and equating it to 0.

$$\vec{z'_k} = \frac{1}{N_k} \sum_{\gamma_{nk}=S_i(\vec{z})}^{N} \vec{x_n} \gamma_{nk} \quad (N_k = \text{no. of pts. assigned to } k^{th} \text{ cluster})$$

$$\text{To prove:} E[\vec{z'}] - E[\vec{z}] \leq 0$$

$$\text{LHS} = E[\vec{z'}] - Q(\vec{z'}, z) + Q(\vec{z'}, \vec{z}) - Q(\vec{z}, \vec{z}) \quad (\text{Writing } E[\vec{z}] = Q(\vec{z}, \vec{z}))$$

We can see that in LHS, $Q(\vec{z'}, \vec{z}) - Q(\vec{z}, \vec{z}) \leq 0$ because using $\vec{z}$ we are creating $\vec{z'}$ i.e $\vec{z'}$ is the best assignment obtained by minimizing the auxilary function. Hence LHS ≤ 0 . Hence proved.

K-Means Vs K-medoid

- Centroid is an actual vector coming from training data.
- We first compute k-means centroid, which might not be an actual datapoint. So, the nearest data point to this centroid is called k-medoid centroid.
- k-Medoid is computationally expensive
- k-Medoid is more robust to outliers – in that outliers are not taken into account during the computation of the medoid.
- In K-Medoid it is possible to have a cluster with one point, while in K-Means it is possible to have a cluster with no points – why?

K-Means for representing the distribution of variables for a class

- The training data of a class is used to find K prototypes for every class.
- During testing, given a set of $\mathcal{D}_{test} = \{\bar{x}_{t1}, ...\bar{x}_{tN_t}\}$ test vectors corresponding to a given class, the Distortion is computed with the prototypes of every class, and the class which yields smallest distortion is identified as the class corresponding to the set of test vectors.
- Voting can also be used – although leads to quantization.
- K-Means clustering is the pre-cursor for GMMs to be discussed later.

Mean shift clustering

The mean update equation for mean shift clustering is given by

$$m(\bar{x}) = \frac{\sum_{\bar{x_i} \in N(x_i)} K(\bar{x} - \bar{x_i}) \cdot x_i}{K(x - \bar{x_i})}$$

where k is a kernel given by

$$K(x_i - \bar{x}) = \frac{e^{-||x_i - \bar{x}||^2}}{h^2}$$

here h is band width parameter

Drawbacks of K Means clustering algorithm

- The clustering is hard, i.e., a point can belong to only one cluster
- Solution: Soft clustering of points
- Each cluster is replaced by a Gaussian
- Every point belong to all the clusters to some extent
- The belongingness of a point $\bar{x}$ to a cluster k is given by Gaussian PDF.

$$p(\bar{x}|z_k = 1) = \sum_{k=1}^{K} \pi_k N(\bar{x}|\bar{\mu_k}, \Sigma_k)$$

where $\sum_{k=1}^{K} \pi_k = 1$