Introduction to Machine Learning

Lecture 2: Revisiting Clustering by Notion of Risk

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

Winter 2026

Recap: K-Means Clustering Algorithm

`K-Means():`

1: Initiate $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
2: **while** $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ changing **do**
3: Set $\mathcal{C}_1, \ldots, \mathcal{C}_K \leftarrow$ `Cluster_Assignment`$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$
4: Update $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \leftarrow$ `Centroid_Update`$(\mathcal{C}_1, \ldots, \mathcal{C}_K)$
5: **end while**
6: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

`Cluster_Assignment`$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$:

$$\textit{index of cluster for } \boldsymbol{x}_n \leftarrow \underset{k \in \{1, \ldots, K\}}{\operatorname{argmin}} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|$$

`Centroid_Update`$(\mathcal{C}_1, \ldots, \mathcal{C}_K)$:

$$\boldsymbol{\mu}_k \leftarrow \texttt{average} \{\boldsymbol{x}_n \in \mathcal{C}_k\} = \frac{1}{|\mathcal{C}_k|} \sum_{\boldsymbol{x}_n \in \mathcal{C}_k} \boldsymbol{x}_n$$

K-Means Clustering is *Good!*

When we derived the K-means clustering algorithm, we considered

K-centroid model for clustering, i.e.,

$$f(\boldsymbol{x}) = \operatorname*{argmin}_{k \in \{1,\ldots,K\}} \|\boldsymbol{x} - \boldsymbol{\mu}_k\|$$

for K centroids $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$

K-means clustering is a learning algorithm

$$\mathcal{A} : \mathbb{D} \mapsto \boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_K^*$$

- ! *We claim it finds a good model*
- ? *How can you define "good" set of centroids?*
- ! *Let's see*

Clustering: *Alternative Formulation*

We have dataset $\mathbb{D}$: *we want to*

group samples into K clusters described by K centroids

For each $\boldsymbol{x}_n \in \mathbb{D}$, we define K weights $r_{n,1}, \ldots, r_{n,K}$

$$r_{n,k} = \begin{cases} 1 & \text{if } \boldsymbol{x}_n \in \mathcal{C}_k \\ 0 & \text{otherwise} \end{cases}$$

Properties of $r_{n,k}$

$$\sum_{k=1}^{K} r_{n,k} = 1 \qquad \text{and} \qquad \sum_{n=1}^{N} r_{n,k} = |\mathcal{C}_k|$$

K-Means Clustering: *Alternative Formulation*

`Cluster_Assignment`$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$:

for $n = 1 : N$ **do**

  Assign K weights $r_{n,1}, \ldots, r_{n,K}$ to sample $\boldsymbol{x}_n$ as

$$r_{n,k} = \begin{cases} 1 & \text{if } \boldsymbol{\mu}_k = \text{closest centroid to } \boldsymbol{x}_n \\ 0 & \text{otherwise} \end{cases}$$

end for

Return $r_{n,k}$ for $k = 1 : K$ and $n = 1 : N$

K-Means Clustering: *Alternative Representation*

`Centroid_Update`$(\{r_{n,k}\})$:

1: **for** $k = 1 : K$ **do**
2:     **if** $\sum_n r_{n,k} > 0$ **then**
3:         Move $\boldsymbol{\mu}_k$ to the center of cluster k, i.e.,

$$\boldsymbol{\mu}_k = \frac{\displaystyle\sum_{n=1}^{N} r_{n,k}\boldsymbol{x}_n}{\displaystyle\sum_{n=1}^{N} r_{n,k}}$$

4:     **else**
5:         Leave $\boldsymbol{\mu}_k$ unchanged
6:     **end if**
7: **end for**
8: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

K-Means Clustering: *Alternative Representation*

We could iterate till we converge

K-`Means()`:

1: Initiate $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
2: **while** $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ changing **do**
3: Set $\{r_{n,k}\} \leftarrow \texttt{Cluster\_Assignment}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$
4: Update $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \leftarrow \texttt{Centroid\_Update}(\{r_{n,k}\})$
5: **end while**
6: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

Defining Objective: *Risk*

- ? *Where should be good centroids?*
- ! *Probably, close to the samples of that cluster!*

Let's aggregate how far samples are from their centroid

$$\mathcal{J}\left(\{r_{n,k}\},\{\boldsymbol{\mu}_k\}\right) = \frac{1}{N}\sum_{k=1}^{K}\sum_{n=1}^{N} r_{n,k}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$$

- $r_{n,k}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$ *is non-zero only if* $\boldsymbol{x}_n \in \mathcal{C}_k$
- *Sum over* n *aggregates distances of samples in cluster* k *from* $\boldsymbol{\mu}_k$
- *Sum over* k *aggregates distances of all samples from their clusters*

We compute how far each point is in average from its centroid

Notion of Optimality

? *Where should be good centroids?*

! *Probably, close to the samples of that cluster!*

Optimal Clustering with K-Centroid Models

Optimal assignments $\{r^{\star}_{n,k}\}$ and centroids $\{\boldsymbol{\mu}^{\star}_k\}$ minimize the risk

$$\left\{r^{\star}_{n,k}\right\}, \{\boldsymbol{\mu}^{\star}_k\} = \underset{\left\{r_{n,k}\right\}, \{\boldsymbol{\mu}_k\}}{\operatorname{argmin}} \ \mathcal{J}\left(\left\{r_{n,k}\right\}, \{\boldsymbol{\mu}_k\}\right)$$

Clustering by *Risk Minimization*

Risk minimization for clustering is hard, so we can use alternating optimization

```
Risk_Minimzation():
```

1: Initiate $\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_K^*$
2: **while** $\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_K^*$ changing **do**
3: Minimize the risk for fixed centroids $\{\boldsymbol{\mu}_k^*\}$

$$\{r_{n,k}^*\} \leftarrow \operatorname*{argmin}_{\{r_{n,k}\}} \mathcal{J}\left(\{r_{n,k}\}, \{\boldsymbol{\mu}_k^*\}\right)$$

4: Minimize the risk for fixed assignments $\{r_{n,k}^*\}$

$$\{\boldsymbol{\mu}_k^*\} \leftarrow \operatorname*{argmin}_{\{\boldsymbol{\mu}_k\}} \mathcal{J}\left(\{r_{n,k}^*\}, \{\boldsymbol{\mu}_k\}\right)$$

5: **end while**
6: Return $\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_K^* \approx \boldsymbol{\mu}_1^\star, \dots, \boldsymbol{\mu}_K^\star$

Clustering via *Risk Minimization*

Minimize the risk for fixed centroids $\{\boldsymbol{\mu}_k^*\}$

$$
\begin{aligned}
\{r_{n,k}^*\} &= \underset{\{r_{n,k}\}}{\operatorname{argmin}} \mathcal{J}\left(\{r_{n,k}\}, \{\boldsymbol{\mu}_k^*\}\right) \\
&= \underset{\{r_{n,k}\}}{\operatorname{argmin}} \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} r_{n,k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k^*\|^2 \\
&= \underset{\{r_{n,k}\}}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} \left(r_{n,1} \|\boldsymbol{x}_n - \boldsymbol{\mu}_1^*\|^2 + \ldots + r_{n,K} \|\boldsymbol{x}_n - \boldsymbol{\mu}_K^*\|^2 \right)
\end{aligned}
$$

So, the solution is given by setting for each n

$$
r_{n,k}^* = \begin{cases} 1 & \text{if } k = \operatorname{argmin}_{j \in \{1,\ldots,K\}} \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\| \\ 0 & \text{otherwise} \end{cases}
$$

Clustering via *Risk Minimization*

Minimizing the risk for fixed centroids $\{\boldsymbol{\mu}_k^*\}$ *is accomplished as*

`Cluster_Assignment`$(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K)$:

1: **for** $n = 1 : N$ **do**

2: Assign K weights $r_{n,1}^*, \dots, r_{n,K}^*$ to sample $\boldsymbol{x}_n$ as

$$r_{n,k}^* = \begin{cases} 1 & \text{if } k = \operatorname{argmin}_{j \in \{1,\dots,K\}} \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\| \\ 0 & \text{otherwise} \end{cases}$$

3: **end for**

4: Return $r_{n,k}^*$ for $k = 1 : K$ and $n = 1 : N$

Clustering by *Risk Minimization*

Minimize the risk for fixed assignments $\left\{ r^*_{n,k} \right\}$

$$
\begin{aligned}
\{\boldsymbol{\mu}_k^*\} &= \operatorname*{argmin}_{\{\boldsymbol{\mu}_k\}} \mathcal{J}\left(\left\{r^*_{n,k}\right\}, \{\boldsymbol{\mu}_k\}\right) \\
&= \operatorname*{argmin}_{\{\boldsymbol{\mu}_k\}} \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} r^*_{n,k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \\
&= \operatorname*{argmin}_{\{\boldsymbol{\mu}_k\}} \frac{1}{N} \sum_{k=1}^{K} \left(\sum_{\boldsymbol{x}_n \in \mathcal{C}_k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \right)
\end{aligned}
$$

The solution for each k *is given by*

$$
\boldsymbol{\mu}_k^* = \frac{1}{|\mathcal{C}_k|} \sum_{\boldsymbol{x}_n \in \mathcal{C}_k} \boldsymbol{x}_n
$$

Clustering by *Risk Minimization*

Minimizing the risk for fixed assignments $\left\{ r_{n,k}^* \right\}$ *is done by*

`Centroid_Update`$\left(\left\{ r_{n,k}^* \right\} \right)$:

1: **for** $k = 1 : K$ **do**
2:     **if** $\sum_n r_{n,k}^* > 0$ **then**
3:         Move $\boldsymbol{\mu}_k$ to the center of cluster k specified by $r_{1,k}^*, \ldots, r_{N,k}^*$
4:     **else**
5:         Leave $\boldsymbol{\mu}_k$ unchanged
6:     **end if**
7: **end for**
8: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

Clustering by *Risk Minimization*

Risk_Minimzation():

Initiate $\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_K^*$

while $\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_K^*$ changing **do**

Minimize the risk for fixed centroids $\{\boldsymbol{\mu}_k^*\}$

$$\{r_{n,k}^*\} \leftarrow \texttt{Cluster\_Assignment}\left(\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_K^*\right)$$

Minimize the risk for fixed assignments $\{r_{n,k}^*\}$

$$\{\boldsymbol{\mu}_k^*\} \leftarrow \texttt{Centroid\_Update}\left(\{r_{n,k}^*\}\right)$$

end while

Return $\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_K^* \approx \boldsymbol{\mu}_1^\star, \dots, \boldsymbol{\mu}_K^\star$

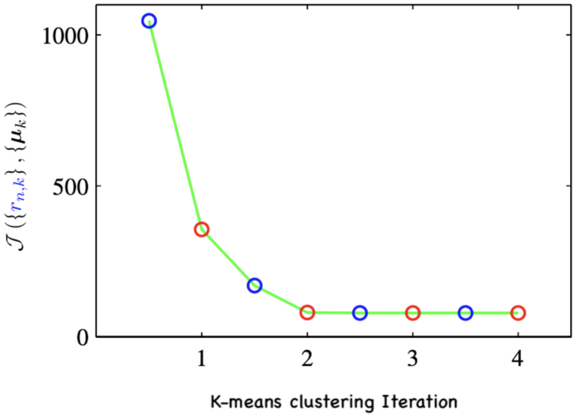
So we conclude that

$$\texttt{Risk\_Minimzation()} \equiv K\texttt{-Means()}$$

K-Means Clustering $\equiv$ *Risk Minimization*

Risk also gives us means to evaluate the learned pattern

Back to our binary example

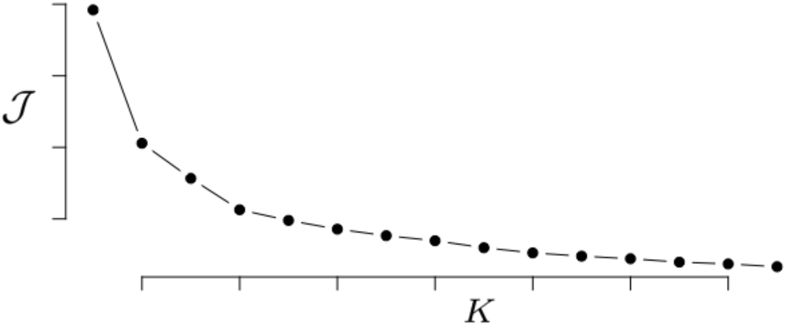


More Sophisticated Example: *Segmentation*[1]

Each RGB pixel is a sample $x_n \in \mathbb{R}^3$*: we cluster with* $K = 10$ $K = 3$ $K = 2$

Choice of *Hyperparameter*

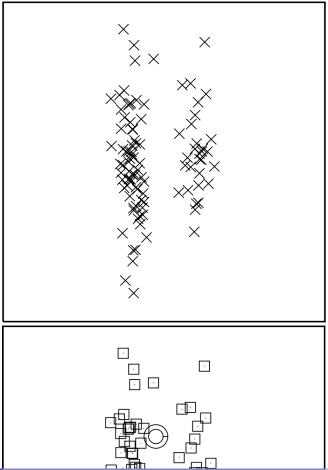
- ? *How do we know K?*
- ! *This is a hyperparameter*



K-Means Clustering Always Converge

- ? Does K-means clustering always converge to a stable state?
- ! Yes! You can show it!

However, it does not necessary end with what we want![2]



Soft Clustering

Recall that

$$\sum_{k=1}^{K} r_{n,k} = 1$$

? What if we think of any $r_{n,k} \in [0,1]$?!

Probabilistic Assignment

In probabilistic assignment we assume $r_{n,k} \in [0,1]$ such that

$$\sum_{k=1}^{K} r_{n,k} = 1$$

$r_{n,k}$ *is hence a probability, i.e., probability of $\boldsymbol{x}_n \in \mathbb{C}_k$*

Risk as Expected Error

By this viewpoint, we can interpret the risk as an expected error

Risk $\equiv$ *Expected Error*

Let $r_{n,k}$ *be probability of* $\boldsymbol{x}_n \in \mathbb{C}_k$

$$\begin{aligned}\mathcal{J}\left(\left\{r_{n,k}\right\}, \left\{\boldsymbol{\mu}_k\right\}\right) &= \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} r_{n,k} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \\ &= \mathbb{E}\left\{\mathcal{E}\left(\boldsymbol{x}\right)\right\}\end{aligned}$$

with $\mathcal{E}\left(\boldsymbol{x}\right)$ *quantifying how bad we have classified, i.e.,*

$$\mathcal{E}\left(\boldsymbol{x}\right) = \|\boldsymbol{x} - \mathtt{cluster}\left(\boldsymbol{x}\right)\|^2$$

Soft Clustering

We could revise our algorithm for *soft clustering*

Soft Clustering with K-Centroid Model

Optimal assignments $\{r^\star_{n,k}\}$ *and centroids* $\{\boldsymbol{\mu}^\star_k\}$ *minimize the risk*

$$\left\{r^\star_{n,k}\right\}, \left\{\boldsymbol{\mu}^\star_k\right\} = \underset{\left\{r_{n,k}\right\},\left\{\boldsymbol{\mu}_k\right\}}{\operatorname{argmin}} \mathcal{J}\left(\left\{r_{n,k}\right\}, \left\{\boldsymbol{\mu}_k\right\}\right)$$

for $\boldsymbol{\mu}_k \in \mathbb{R}^d$ *and* $r_{n,k} \in [0,1]$ *such that*

$$\sum_{k=1}^{K} r_{n,k} = 1.$$

Soft K-Means Clustering Algorithm

`Soft_Cluster_Assignment`$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$:

1: **for** $n = 1 : N$ **do**

2: Assign K weights $r_{n,1}, \ldots, r_{n,K}$ to sample $\boldsymbol{x}_n$ for some β as

$$r_{n,k} = \frac{e^{-\beta \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2}}{\displaystyle\sum_{k=1}^{K} e^{-\beta \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2}}$$

3: **end for**

4: Return $r_{n,k}$ for $k = 1 : K$ and $n = 1 : N$

? *Does it remind you of any distribution?!*

Soft K-Means Clustering Algorithm

`Centroid_Update`$(\{r_{n,k}\})$:

1: **for** $k = 1 : K$ **do**

2: Move $\boldsymbol{\mu}_k$ to the center of cluster k, i.e.,

$$\boldsymbol{\mu}_k = \frac{\displaystyle\sum_{n=1}^{N} r_{n,k}\boldsymbol{x}_n}{\displaystyle\sum_{n=1}^{N} r_{n,k}}$$

3: **end for**

4: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

Soft K-Means Clustering Algorithm

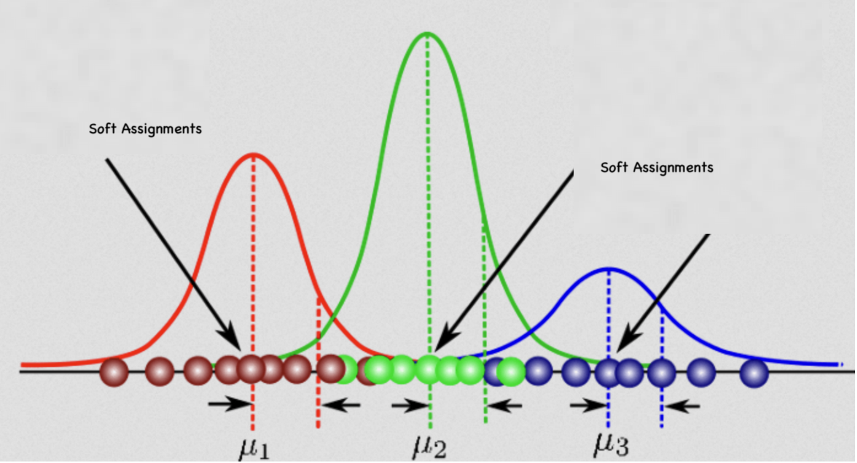
We could iterate till we converge

`Soft_`K`-Means():`

1: Initiate $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
2: **while** $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ changing **do**
3: Set $\{r_{n,k}\} \leftarrow$ `Soft_Cluster_Assignment`$(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$
4: Update $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \leftarrow$ `Centroid_Update`$(\{r_{n,k}\})$
5: **end while**
6: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

Gaussian Prior

K-means assumes Gaussian distributed data!



Gaussian Prior

We understand it better if we discuss

Density Learning

This is what we do next!

Further Read

- MacKay
  - ↳ Chapter 20 ***Soft K-means***
- Bishop
  - ↳ Chapter 9 ***K-means***