ECE 1513: Introduction to Machine Learning

Lecture 1: Preliminaries and Clustering

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

Winter 2025

What is Machine Learning?

It's a hard question to answer accurately

Mitchel defines ML as *"... the study of computer algorithms that improve automatically through experience..."*

and

Goodfellow et al. *informally* define ML as *"... a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions..."*

and ...

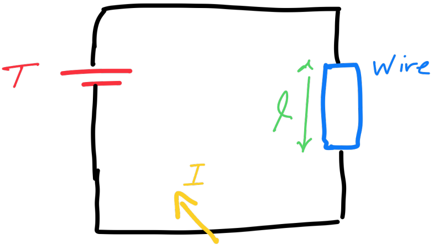
What is Machine Learning?

But not too hard to answer practically

> We define ML as *the set of data-driven approaches that help us understand the environment and its behavior, and generalize it!*

Data-driven approaches have long been with us in science and engineering!

Early Example from 1827: *Ohm's Law*



What Did Ohm Do?

Georg Ohm did three major steps

- He saw a pattern and hypothesized some mathematical model
  - ↳ *Electric current increases with voltage*
  - ↳ *The constant changes with the length and material*
  - ↳ ...
- He collected data
  - ↳ *Electric currents and voltages*
- He used mathematical tools to extract the modeled pattern
  - ↳ *Some curve-fitting technique*

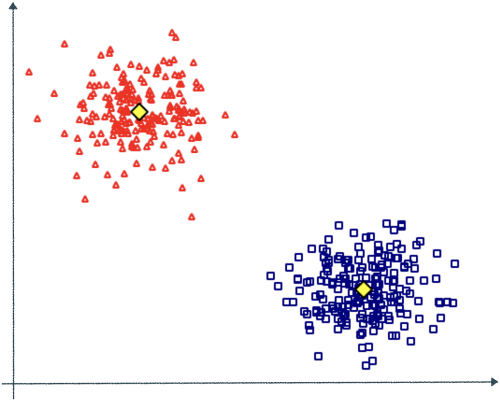
Learning Task

Any learning task has three components

- Model *that captures the Pattern*
- Data
- Learning Algorithm

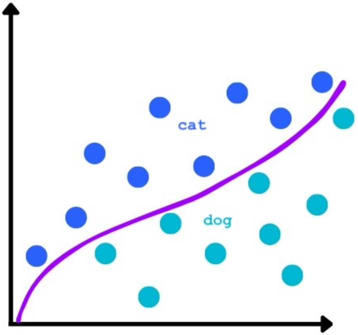
Example: *Clustering*

Monthly amount of transactions versus *# of transactions per month*



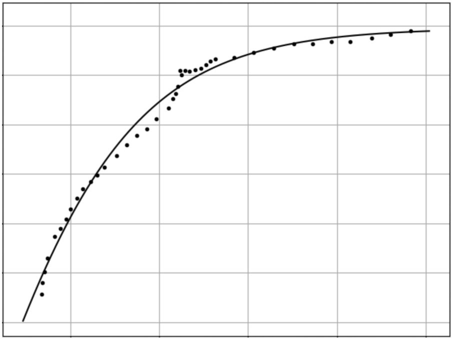
Example: *Classification*

Sleep time versus *# of times the pet makes noise*

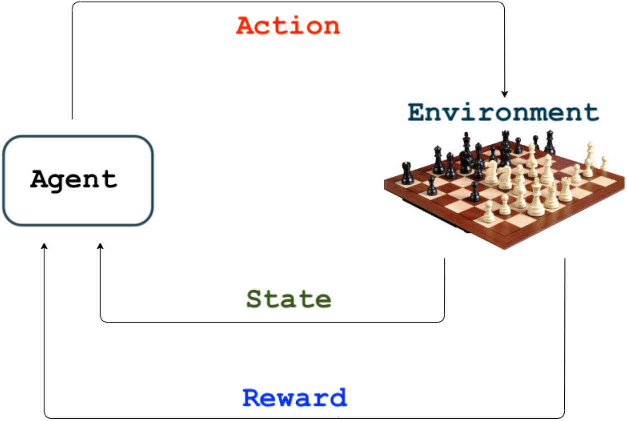


Example: *Regression*

Salary versus *years of experience*



Example: *Playing Chess*



Dataset

A set of data samples

$$\mathbb{D} = \{\boldsymbol{x}_n : n = 1, \ldots, N\}$$

with $\boldsymbol{x}_n \in \mathbb{R}^d$

Let's formulate data in our examples

- Clustering
- Classification
- Regression
- Playing Chess

Model

A pre-assumed function

$$f : \boldsymbol{x} \mapsto y$$

for a data sample $\boldsymbol{x}$ and output y that *fits the learning task*

Let's formulate model in our examples

- Clustering
- Classification
- Regression
- Playing Chess

Learning Algorithm

Algorithm that gets dataset and returns the exact model

$$\mathcal{A} : \mathbb{D} \mapsto f^\star$$

$f^\star$ does the mapping such that we get to the desired output

---

Let's formulate learning algorithm in our examples

- Clustering
- Classification
- Regression
- Playing Chess

? *How can we define a "good" learning algorithm?*

Unsupervised Learning

Data samples are not labeled

$$\mathbb{D} = \{\boldsymbol{x}_n : n = 1, \ldots, N\}$$

Here, we are looking for a pattern in the data

Other components of an unsupervised task

- Model *captures the pattern hidden in data*
- Learning Algorithm

Examples of unsupervised learning

- ✔ Clustering
- Dimensionality Reduction
- Distribution Learning

Supervised Learning

Data samples are labeled

$$\mathbb{D} = \{(\boldsymbol{x}_n, v_n) : n = 1, \ldots, N\}$$

Here, we are looking for a model that describes the relation

Other components of a supervised task

- Model *describes the relation between data samples and their labels*
- Learning Algorithm

Examples of supervised learning

- Classification
- Regression

Reinforcement Learning

Data samples are series of *actions*, *states* and *rewards*

$$\mathbb{D} = \{\{(a_n^t, s_n^t, r_n^t) : t = 1, \ldots\} : n = 1, \ldots, N\}$$

Here, we are looking for optimal policy, i.e., policy that maximizes future returns

$$G_t = r^t + r^{t+1} + \ldots$$

Other components of a reinforcement task

- Model *describes a policy*
- Learning Algorithm

Examples of reinforcement learning

- Playing Game, Control Robots, ...

Reinforcement learning is not discussed in this course, but you may consider taking Reinforcement Learning in next Fall

Further Read

- Bishop
  - ↳ Chapter 1: *Sections 1.1 and 1.3* — ***Introductory***
- ESL
  - ↳ Chapter 1 — ***Introductory***
  - ↳ Chapter 2: *Sections 2.1 and 2.2* — ***Supervised***
  - ↳ Chapter 14: *Sections 14.1 and 14.2* — ***Unsupervised***
- Mitchell
  - ↳ Chapter 13: *Sections 13.1 and 13.2* — ***Reinforcement***
- Goodfellow, et al.
  - ↳ Chapter 5: *Sections 5.1 and 5.2* — ***Introductory***

Unsupervised Learning

Why do we start with *unsupervised learning*?

- Many basic problems are unsupervised
  - ↳ *We naturally cluster everything around us*
  - ↳ *We get sense about quantities by understanding its statistical behavior*
- It helps us to recap some basics we need later
  - ↳ *Linear Algebra*
  - ↳ *Probability Theory*

Problem of Clustering

This is a basic sign of intelligence

- *We cluster everything around us*
  - ↳ *Trees, flowers, animals, . . .*
- *We often start with simple clustering and extend hierarchically*
  - ↳ *Plants and animals*
  - ↳ *Plants could be trees, flowers, . . .*
  - ↳ *Animals could be mammals, birds, . . .*
- *The further we go, the more intelligent we get!*

Basic Clustering Task: *Data*

Data samples are points in d-dimensional space

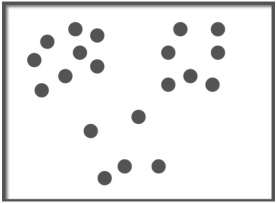
$$\mathbb{D} = \{\boldsymbol{x}_n : n = 1, \dots, N\}$$

with $\boldsymbol{x}_n \in \mathbb{R}^d$

In Examples

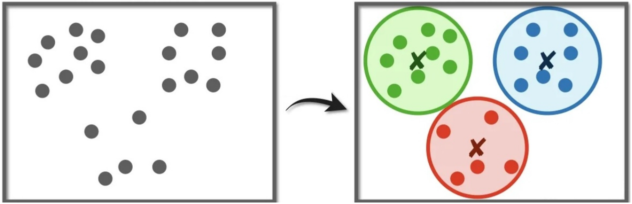
In examples, we always think of two dimensions for sake of simplicity

Recall our bank record example



Basic Clustering Task: *Pattern*

We assume that the samples can be grouped into clusters



Recall our bank record example

Basic Clustering Task: *Model*

We use a model to capture the clustering pattern

$$f\left(\boldsymbol{x}_n\right) = k \in \{1, \ldots, K\}$$

for some integer K

Some definitions and assumptions

- *Cluster subspace* k

$$\mathbb{C}_k = \left\{\boldsymbol{x}_n : f\left(\boldsymbol{x}_n\right) = k\right\}$$

- *Cluster subspaces partition the data space*

$$\mathbb{C}_1 \cup \ldots \cup \mathbb{C}_K = \mathbb{X} \rightsquigarrow \textit{all possible samples}$$
$$\mathbb{C}_j \cap \mathbb{C}_k = \varnothing \rightsquigarrow \forall j \neq k$$

Basic Clustering Task: *Learning Algorithm*

The learning algorithm gets data and find a good f

$$\mathcal{A} : \mathbb{D} \mapsto f^\star$$

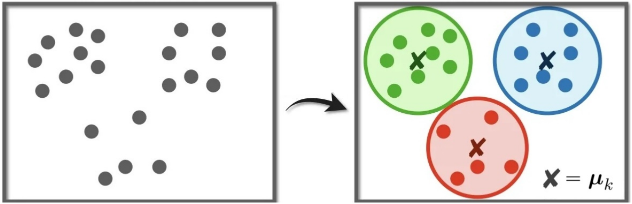
- ? *What is a "good" model?*
- ! *We'll answer it!*

An Intuitive Model: K *Centroids*

Let's use a simple and intuitive model

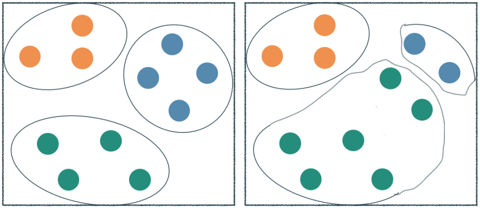
$$f(\boldsymbol{x}) = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmin}} \left\| \boldsymbol{x} - \boldsymbol{\mu}_k \right\|$$

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



K Centroids: *Learning Algorithm*

The model is valid for any set of centroids!



The learning algorithm is to start from $\mathbb{D}$ and learn good centroids

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

? *What is a "good" set of centroids?*

! *We'll answer it!*

K-Means Clustering Algorithm: *Intuitive Derivation*

Given the centroids, we can easily assign each $\boldsymbol{x}_n \in \mathbb{D}$ to a cluster-set

```
Cluster_Assignment(μ_1, ..., μ_K):
    # we want to find C_1 ∪ ... ∪ C_K = D
```

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

2: Assign $\boldsymbol{x}_n$ to cluster-set $\mathcal{C}_{f(\boldsymbol{x}_n)}$ with

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

3: **end for**

4: Return $\mathcal{C}_1, \ldots, \mathcal{C}_K$

K-Means Clustering Algorithm: *Intuitive Derivation*

Given the cluster sets, we can move centroids to the center of cluster-sets

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

  `# we want to find` $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K$

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

  if $\mathcal{C}_k \neq \emptyset$ **then**

    Move $\boldsymbol{\mu}_k$ to the center of cluster $\mathcal{C}_k$, i.e.,

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

  else

    Leave $\boldsymbol{\mu}_k$ unchanged

  end if

end for

Return $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K$

K-Means Clustering Algorithm

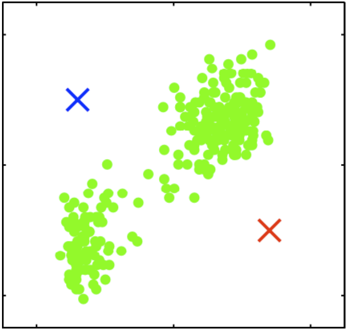
We could iterate till we converge

K`-Means()`:

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

Example: *2-Means Clustering*[1]

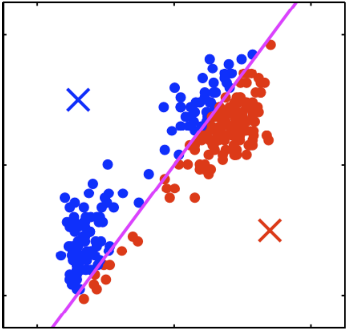
Initial centroids



[1] *This example is taken from Bishop's book, Chapter 9*

Example: *2-Means Clustering*[1]

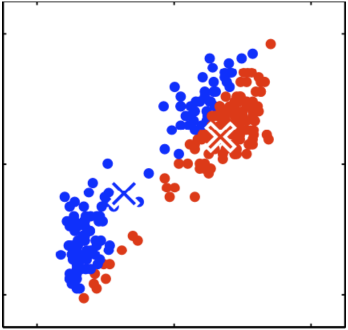
Iteration 1



[1] *This example is taken from Bishop's book, Chapter 9*

Example: *2-Means Clustering*[1]

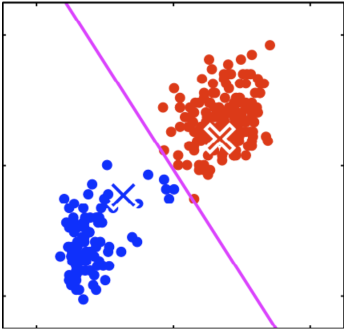
Iteration 1



[1] *This example is taken from Bishop's book, Chapter 9*

Example: *2-Means Clustering*[1]

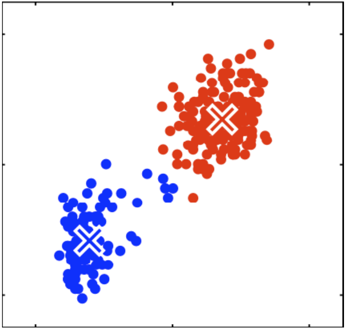
Iteration 2



[1] *This example is taken from Bishop's book, Chapter 9*

Example: *2-Means Clustering*[1]

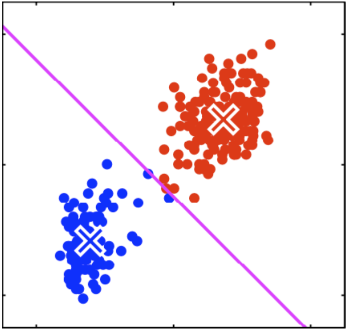
Iteration 2



[1] *This example is taken from Bishop's book, Chapter 9*

Example: *2-Means Clustering*[1]

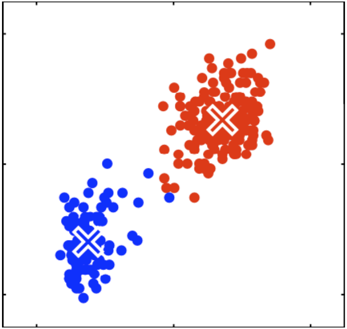
Iteration 3



[1] *This example is taken from Bishop's book, Chapter 9*

Example: *2-Means Clustering*[1]

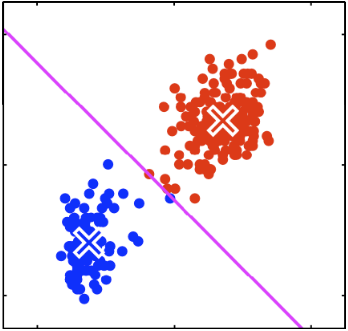
Iteration 3



[1] *This example is taken from Bishop's book, Chapter 9*

Example: *2-Means Clustering*[1]

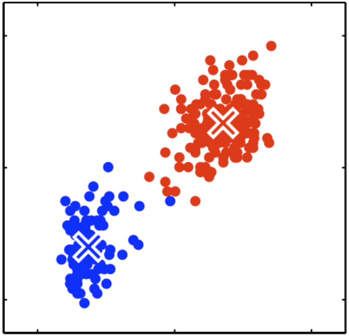
Iteration 4 – Converged



[1] *This example is taken from Bishop's book, Chapter 9*

Example: *2-Means Clustering*[1]

Iteration 4 – Converged



[1] *This example is taken from Bishop's book, Chapter 9*

K-Means Clustering Algorithm: *Alternative Formulation*

```
Cluster_Assignment(μ_1, ..., μ_K):
```

1: **for** $n = 1 : N$ **do**
2: Assign K weights $r_{n,1}, \ldots, r_{n,K}$ to sample $\boldsymbol{x}_n$ as

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

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

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 Algorithm: *Alternative Formulation*

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

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

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

  else
    Leave $\boldsymbol{\mu}_k$ unchanged
  end if
end for
Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

K-Means Clustering Algorithm: *Alternative Formulation*

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 $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

This is a better form to extend K-means clustering to a soft format

Defining Objective: *Risk*

- ? *What is a "good" set of centroids?*
- ! *We'll answer it!*

We may define a metric to evaluate how our model performs

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

This specifies the risk we take with this model

Notion of Optimality

? *What is a "good" set of centroids?*

! *We'll answer it!*

Optimal Clustering

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

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

K-Means Clustering: *Risk Minimization*

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

`Risk_Minimzation():`

1: Initiate $\boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_K^*$

2: **while** $\boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_K^*$ changing **do**

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

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

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

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

5: **end while**

6: Return $\boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_K^* \approx \boldsymbol{\mu}_1^\star, \ldots, \boldsymbol{\mu}_K^\star$

K-Means Clustering: *Risk Minimization*

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

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

which is done by

$\texttt{Cluster\_Assignment}(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K)$:

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

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

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

end for

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

K-Means Clustering: *Risk Minimization*

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

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

which is done by

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

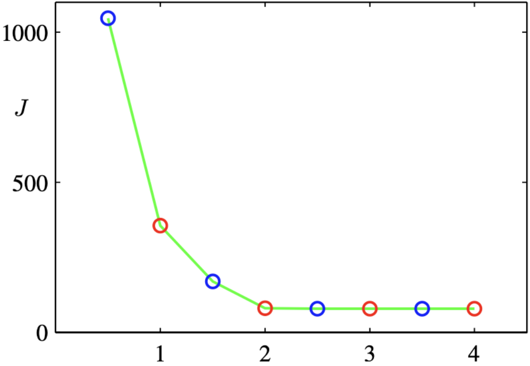
1: **for** $k = 1 : K$ **do**
2:     **if** $\sum_n r_{n,k} > 0$ **then**
3:         Move $\boldsymbol{\mu}_k$ to the center of cluster k
4:     **else**
5:         Leave $\boldsymbol{\mu}_k$ unchanged
6:     **end if**
7: **end for**
8: Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

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

So we conclude

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

Back to our binary example



More Sophisticated Example: *Segmentation*[2]

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

[2] *This example is taken from Bishop's book, Chapter 9*

More Sophisticated Example: *Segmentation*[2]

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

[2] *This example is taken from Bishop's book, Chapter 9*

More Sophisticated Example: *Segmentation*[2]

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

[2] *This example is taken from Bishop's book, Chapter 9*

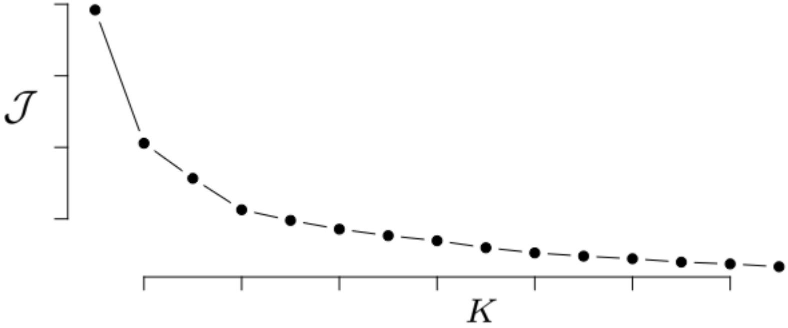
More Sophisticated Example: *Segmentation*[2]

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

[2] *This example is taken from Bishop's book, Chapter 9*

Choice of *Hyperparameter*

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

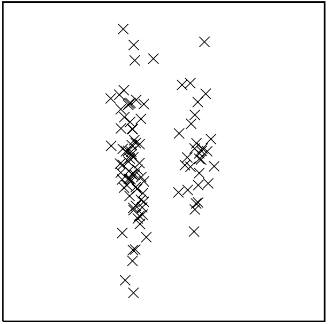


K-Means Clustering Always Converge

? *Does K-means clustering always converge to a stable state?*

! *Yes! You can show it!*

However, it doesnot necessary end with what we want![3]

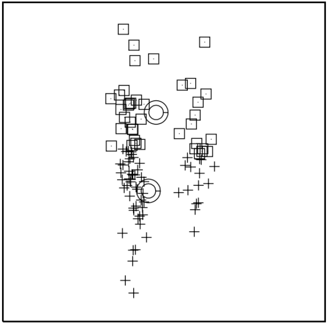


[3] *This example is taken from MacKay's book, Chapter 20*

K-Means Clustering Always Converge

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

However, it doesnot necessary end with what we want![3]



[3] *This example is taken from MacKay's book, Chapter 20*

Risk as Expected Error

Recall that

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

We can look at $r_{n,k}$ *for* $k = 1 : K$ *as a probabilities, i.e.,* $r_{n,k} \in [0, 1]$

Risk $\equiv$ Expected Error

We can then interpret the risk as an expected error for clustering

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

with $\mathcal{E}(\boldsymbol{x})$ *quantifying how* *bad* *we have classified*

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

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

3: **end for**

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

Soft K-Means Clustering Algorithm

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

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

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

  else
    Leave $\boldsymbol{\mu}_k$ unchanged
  end if
end for
Return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

Soft K-Means Clustering Algorithm

We could iterate till we converge

```
Soft_K-Means():
1: Initiate μ_1, ..., μ_K
2: while μ_1, ..., μ_K changing do
3:     Set {r_{n,k}} ← Soft_Cluster_Assignment(μ_1, ..., μ_K)
4:     Update μ_1, ..., μ_K ← Centroid_Update({r_{n,k}})
5: end while
6: Return μ_1, ..., μ_K
```

Further Read

- MacKay
  - ↳ Chapter 20 — ***K-means and Soft K-means***
- Bishop
  - ↳ Chapter 9: *Section 9.1* — ***K-means***
- ESL
  - ↳ Chapter 14: *Section 14.3* — ***Clustering***