

CX4240 Spring 2026

Clustering:

Gaussian Mixture Models vs. k-means

Bo Dai
School of CSE, Georgia Tech
bodai@cc.gatech.edu

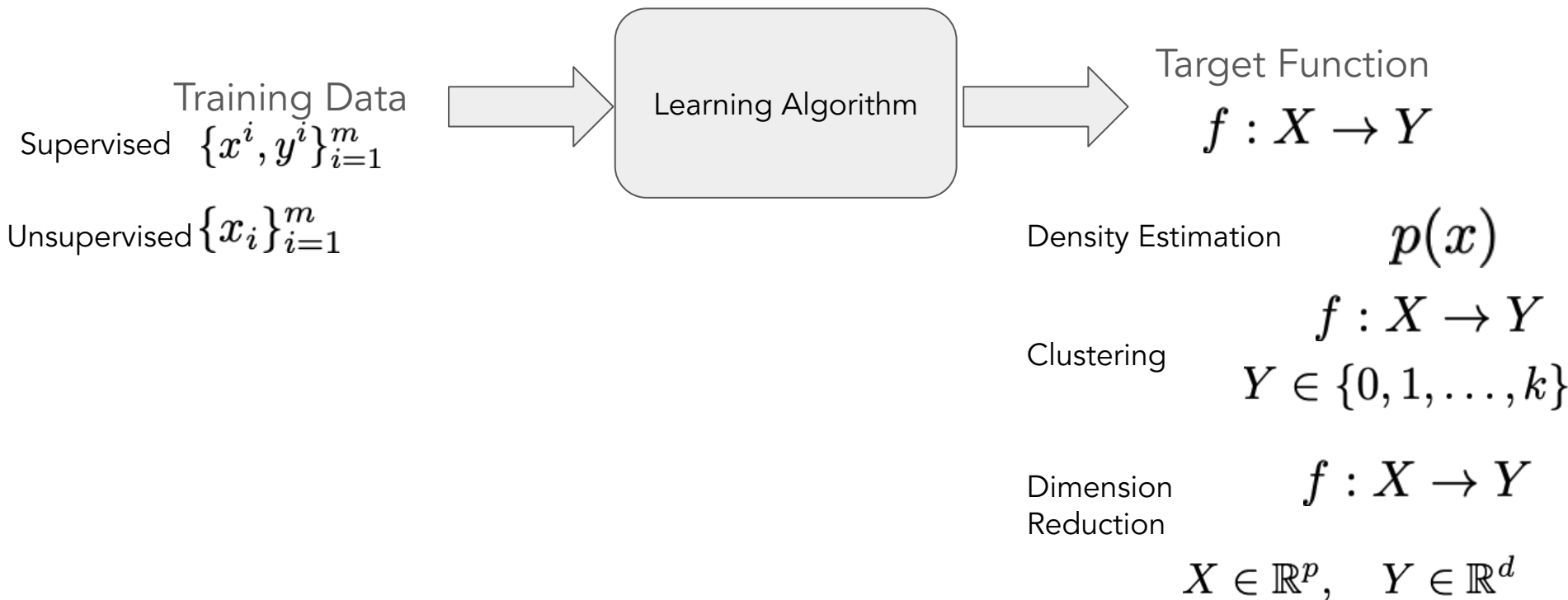
Grading Policy Update

- Based on the poll on Ed, we will have an additional proposal with **10% bonus** credit.
- The deadline will be on **March 13th**.
- The practice exam will be released on **March 6th**.
- One exam will be held on **March 18th** in lieu of the regular class.
- We will have a makeup exam on **March 16th or 17th** (TBD)
 - You need an approval of absence for this makeup exam

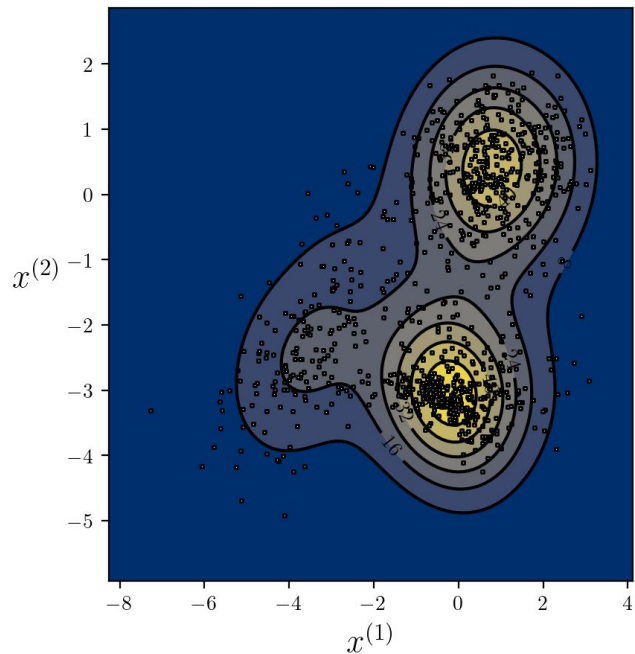
Proposal Requirement

- All write-ups should use the [NeurIPS style](#).
- Your final report is expected to be up to **3 pages** excluding references. It should have roughly the following format:
 - Introduction: Problem definition and motivation
 - Background & Related Work: Background info and literature survey

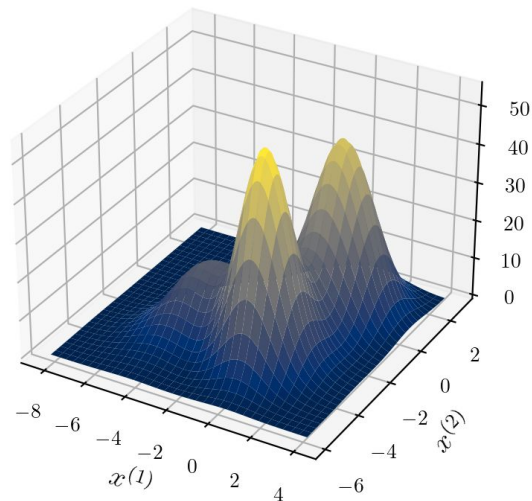
Supervised Learning vs. Unsupervised Learning



Density Estimation



$$\{x_i\}_{i=1}^m$$

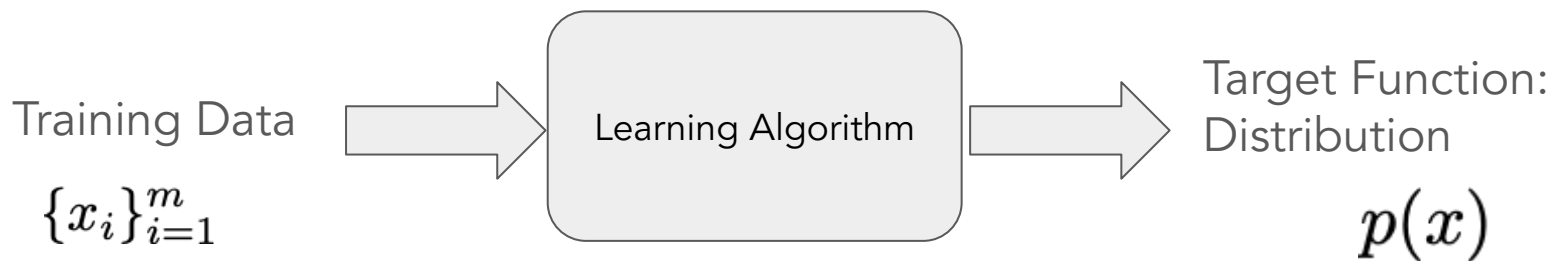


$$p(x)$$

Generative Models

$$x \sim p(x)$$

Density Estimation: Gaussian Mixture Model



Density Estimation Pipeline

1. Build probabilistic models
Gaussian Mixture Model
2. Derive loss function (by MLE or MAP....)
MLE
3. Select optimizer
EM

Gaussian Mixture Model

Class mixture prior: $P(y)$ $\pi = (\pi_1, \pi_2, \dots, \pi_k), \sum_{i=1}^k \pi_i = 1, \pi_i \geq 0$

Class conditional distribution: $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$

Marginal distribution: $P(x) = \sum_y P(x|y)P(y) = \sum_{i=1}^k \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$

Expectation-Maximization

For $t = 1, \dots$

- **E-Step**: Guess sample labels based on current model

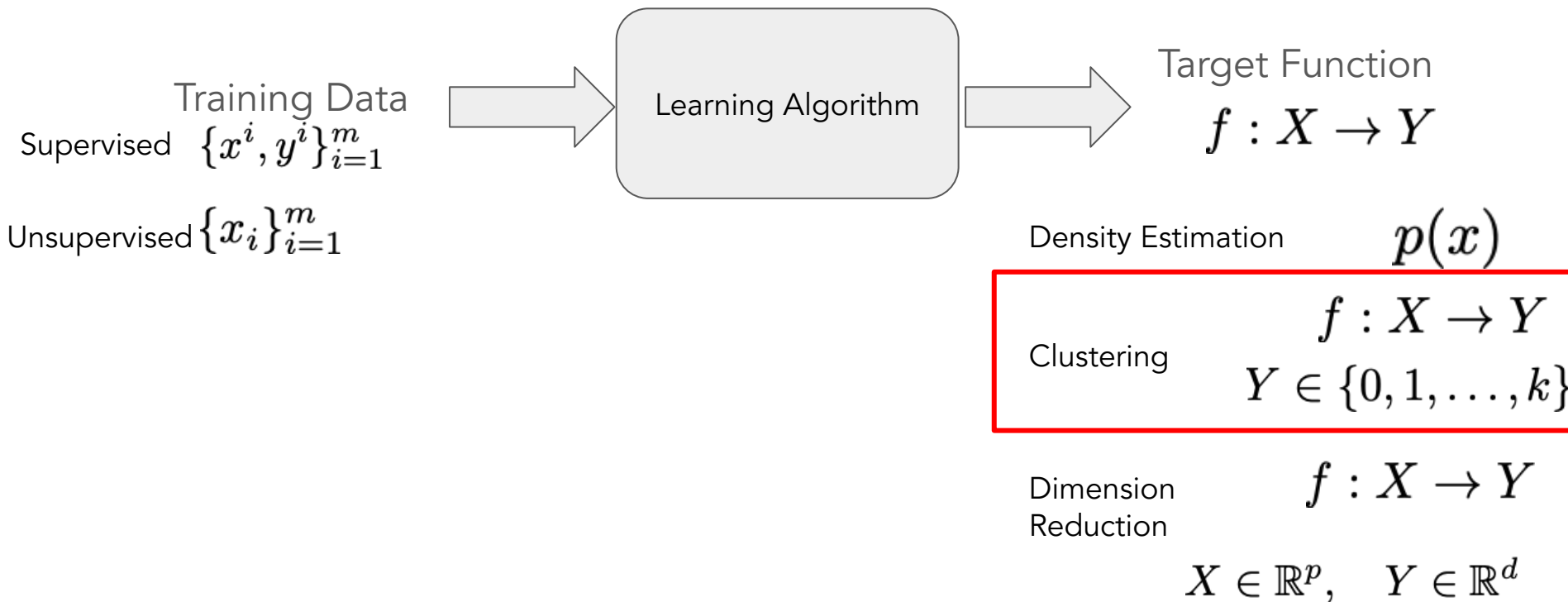
$$y_j^l = \frac{\pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}$$

- **M-Step**: Update the parameters with current labels (**Gaussian-Naive Bayes**)

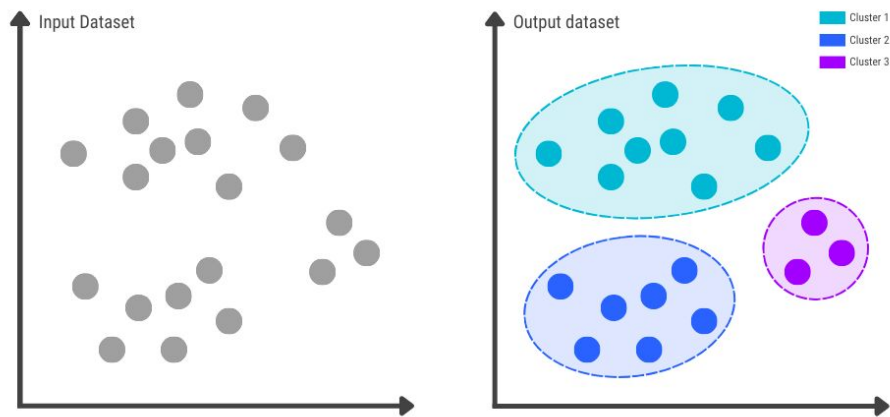
$$\mu_k = \frac{\sum_{i=1}^m y_k^i x^i}{\sum_{i=1}^m y_k^i} \quad \pi_k = \frac{\sum_{i=1}^m y_k^i}{m} \quad \Sigma_k = \frac{\sum_{i=1}^m y_k^i (x^i - \mu_k) (x^i - \mu_k)^\top}{\sum_{i=1}^m y_k^i}$$

This procedure is actually optimizing an upper bound of MLE, therefore, it converges

Supervised Learning vs. Unsupervised Learning

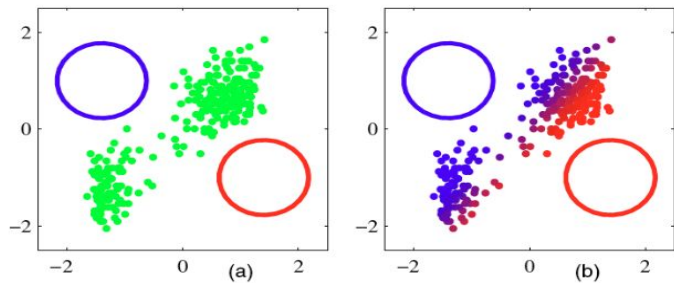


Clustering



- Assume the data $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ lives in a vector space, $\mathbf{x}^{(n)} \in \mathbb{R}^d$.
- Assume the data belongs to K classes (patterns).
- How can we identify those classes (data points that belongs to each class)?

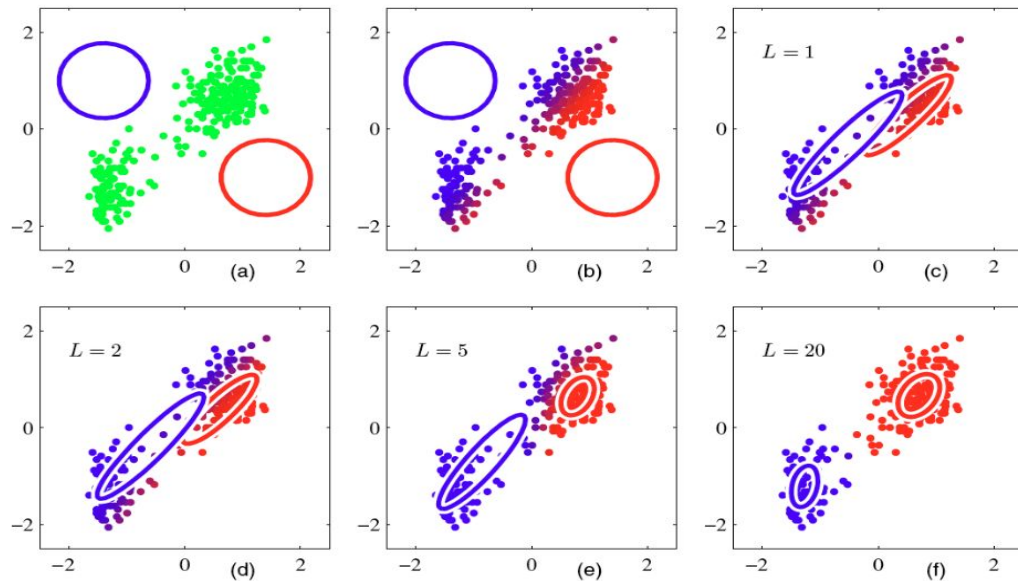
GMM for Clustering



- **E-Step:** Guess sample labels based on current model

$$y_j^l = \frac{\pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}$$

GMM for Clustering



- **E-Step:** Guess sample labels based on current model

$$y_j^l = \frac{\pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}$$

K-means algorithm (Lloyd, 1957)

- Initialize k cluster centers, $\{\mu_1, \mu_2, \dots, \mu_k\}$, randomly
- Do
 - (Assignment) Decide the cluster memberships of each data point, x^i , by assigning it to the nearest cluster center

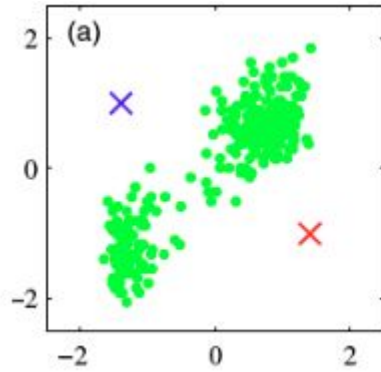
$$y^i = \arg \min_{j=1, \dots, k} \|x^i - \mu_j\|^2.$$

- (Center Update) Adjust the cluster centers

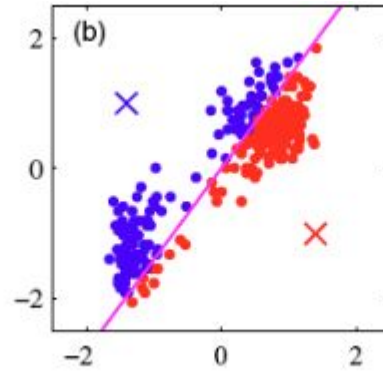
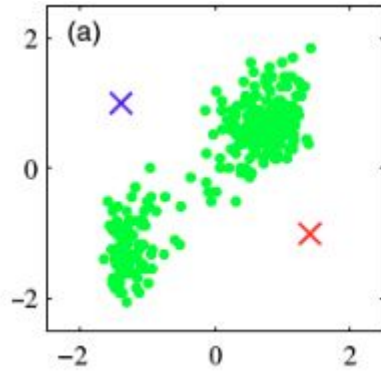
$$\mu_j = \frac{1}{|\{i : y^i = j\}|} \sum_{i: y^i = j} x^i.$$

- While any cluster center has been changed

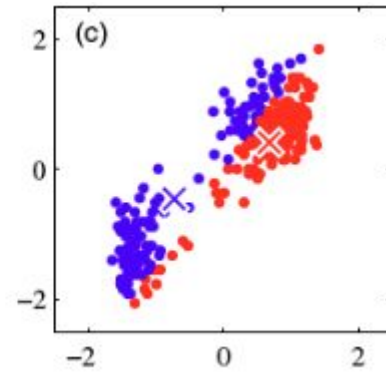
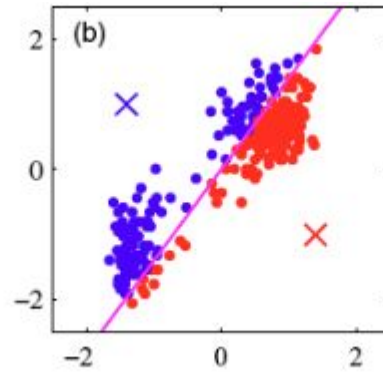
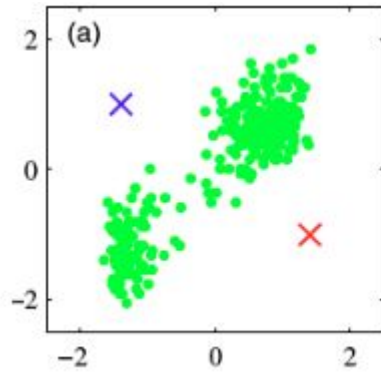
K-means for Clustering (Lloyd, 1957)



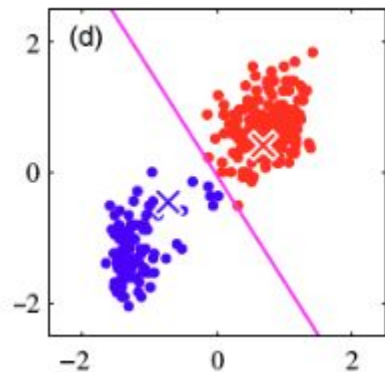
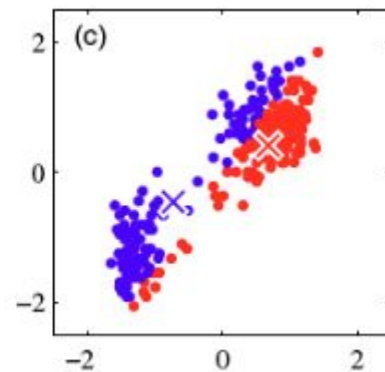
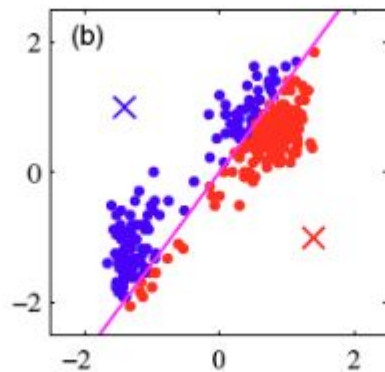
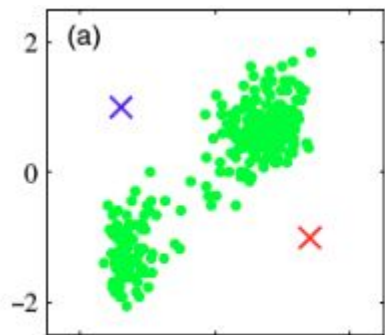
K-means for Clustering (Lloyd, 1957)



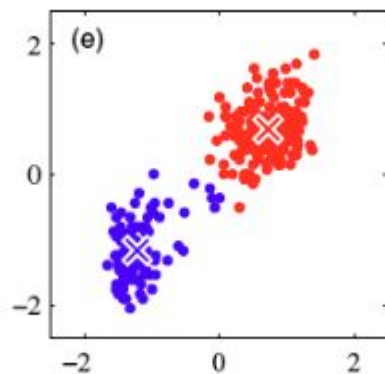
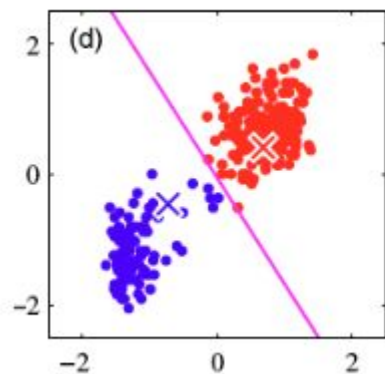
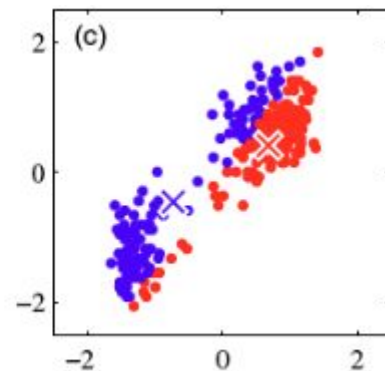
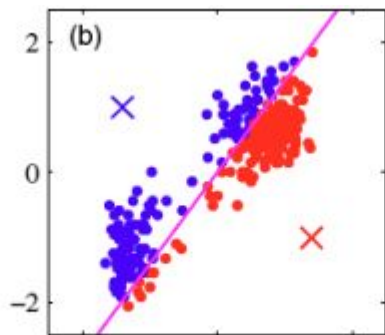
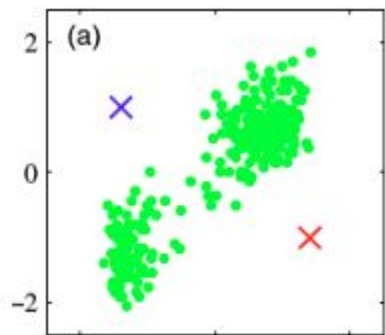
K-means for Clustering (Lloyd, 1957)



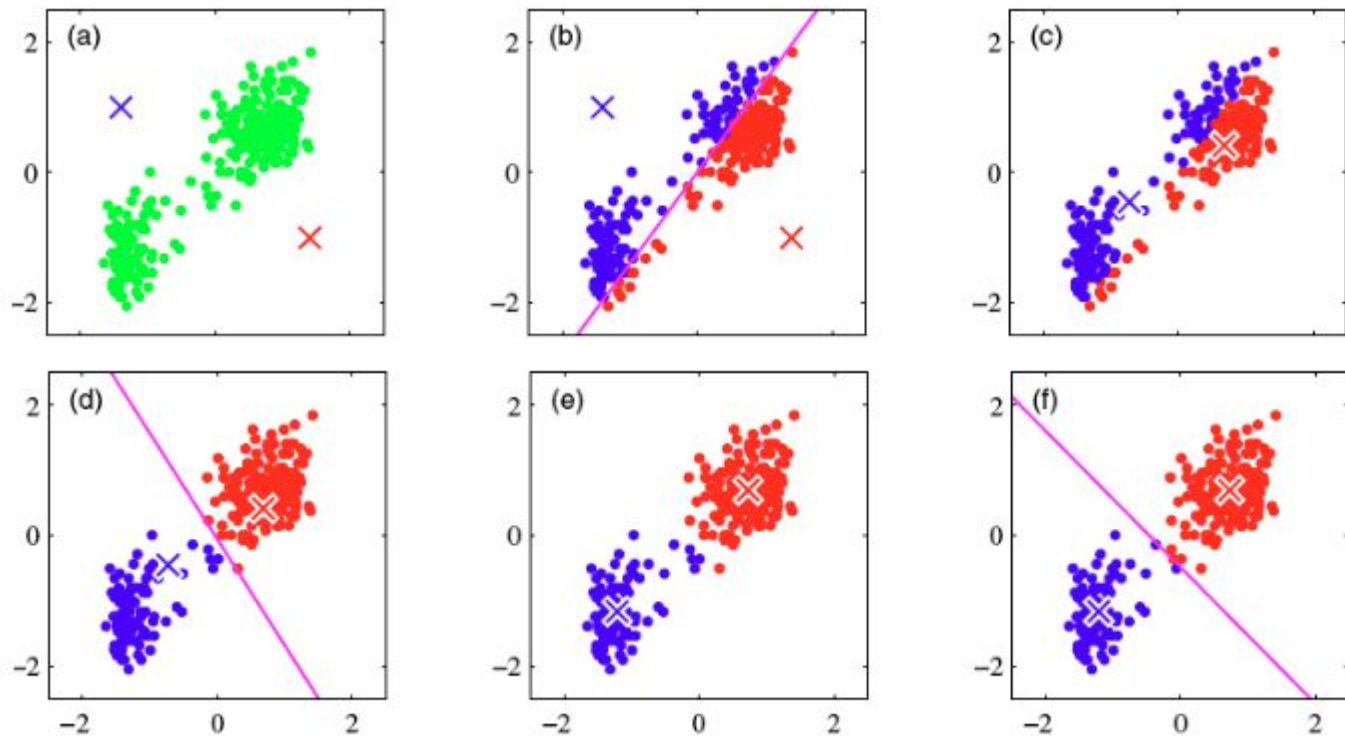
K-means for Clustering (Lloyd, 1957)



K-means for Clustering (Lloyd, 1957)



K-means for Clustering (Lloyd, 1957)



K-means vs. GMM

For $t = 1, \dots$

- **E-Step:** Guess sample labels based on current model

$$y_j^t = \frac{\pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}$$

- **M-Step:** Update the parameters with current labels (**Gaussian-Naive Bayes**)

$$\mu_k = \frac{\sum_{i=1}^m y_k^i x^i}{\sum_{i=1}^m y_k^i} \quad \pi_k = \frac{\sum_{i=1}^m y_k^i}{m} \quad \Sigma_k = \frac{\sum_{i=1}^m y_k^i (x^i - \mu_k)(x^i - \mu_k)^\top}{\sum_{i=1}^m y_k^i}$$

- Initialize k cluster centers, $\{c^1, c^2, \dots, c^k\}$, randomly
- Do
 - (**Assignment**) Decide the cluster memberships of each data point, x^i , by assigning it to the nearest cluster center

$$y^i = \arg \min_{j=1, \dots, k} \|x^i - \mu_j\|^2.$$

- (**Center Update**) Adjust the cluster centers

$$\mu_j = \frac{1}{|\{i : y^i = j\}|} \sum_{i: y^i=j} x^i.$$

- While any cluster center has been changed

K-means vs. GMM

For $t = 1, \dots$

- **E-Step:** Guess sample labels based on current model

$$y_j^l = \frac{\pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)} \quad \xrightarrow{\pi_l = \frac{1}{k}} \quad \sum_k y_k^{(n)} = 1, \forall n, \text{ where } y_k^{(n)} \in \{0, 1\}, \forall k, n$$

- **M-Step:** Update the parameters with current labels (**Gaussian-Naive Bayes**)

$$\mu_k = \frac{\sum_{i=1}^m y_k^i x^i}{\sum_{i=1}^m y_k^i} \quad \pi_k = \frac{\sum_{i=1}^m y_k^i}{m} \quad \Sigma_k = \frac{\sum_{i=1}^m y_k^i (x^i - \mu_k) (x^i - \mu_k)^\top}{\sum_{i=1}^m y_k^i}$$

- Initialize k cluster centers, $\{c^1, c^2, \dots, c^k\}$, randomly
- Do
 - (**Assignment**) Decide the cluster memberships of each data point, x^i , by assigning it to the nearest cluster center

$$y^i = \arg \min_{j=1, \dots, k} \|x^i - \mu_j\|^2.$$

- (**Center Update**) Adjust the cluster centers

$$\mu_j = \frac{1}{|\{i : y^i = j\}|} \sum_{i: y^i = j} x^i.$$

- While any cluster center has been changed

K-means vs. GMM

For $t = 1, \dots$

- **E-Step:** Guess sample labels based on current model

$$y_j^i = \frac{\pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}$$

$$\xrightarrow{\pi_l = \frac{1}{k}} \sum_k y_k^{(n)} = 1, \forall n, \text{ where } y_k^{(n)} \in \{0, 1\}, \forall k, n$$

- **M-Step:** Update the parameters with current labels (**Gaussian-Naive Bayes**)

$$\mu_k = \frac{\sum_{i=1}^m y_k^i x^i}{\sum_{i=1}^m y_k^i} \quad \pi_k = \frac{\sum_{i=1}^m y_k^i}{m} \quad \Sigma_k = \frac{\sum_{i=1}^m y_k^i (x^i - \mu_k)(x^i - \mu_k)^\top}{\sum_{i=1}^m y_k^i}$$

- Initialize k cluster centers, $\{c^1, c^2, \dots, c^k\}$, randomly
- Do
 - (**Assignment**) Decide the cluster memberships of each data point, x^i , by assigning it to the nearest cluster center

$$y^i = \arg \min_{j=1, \dots, k} \|x^i - \mu_j\|^2.$$

- (**Center Update**) Adjust the cluster centers

$$\mu_j = \frac{1}{|\{i : y^i = j\}|} \sum_{i: y^i = j} x^i.$$

- While any cluster center has been changed

K-means vs. GMM

For $t = 1, \dots$

- **E-Step:** Guess sample labels based on current model

$$y_j^t = \frac{\pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}{\sum_{l=1}^k \pi_l \mathcal{N}(x_j | \mu_l, \Sigma_l)}$$

- **M-Step:** Update the parameters with current labels (**Gaussian-Naive Bayes**)

$$\mu_k = \frac{\sum_{i=1}^m y_k^i x^i}{\sum_{i=1}^m y_k^i} \quad \pi_k = \frac{\sum_{i=1}^m y_k^i}{m} \quad \Sigma_k = \frac{\sum_{i=1}^m y_k^i (x^i - \mu_k) (x^i - \mu_k)^\top}{\sum_{i=1}^m y_k^i}$$

- Initialize k cluster centers, $\{c^1, c^2, \dots, c^k\}$, randomly
- Do
 - (**Assignment**) Decide the cluster memberships of each data point, x^i , by assigning it to the nearest cluster center

$$y^i = \arg \min_{j=1, \dots, k} \|x^i - \mu_j\|^2.$$

- (**Center Update**) Adjust the cluster centers

$$\mu_j = \frac{1}{|\{i : y^i = j\}|} \sum_{i: y^i=j} x^i.$$

- While any cluster center has been changed

K-means can be understood as hard-GMM
GMM can be understood as soft k-means

K-means is **Approximating** Gaussian Mixture Model



Density Estimation Pipeline

1. Build probabilistic models
Gaussian Mixture Model with fixed isotropic covariance
2. Derive loss function (by MLE or MAP...)
Approximated MLE
3. Select optimizer
EM

K-means from MLE Perspective

- K-means Objective:

Find cluster centers $\boldsymbol{\mu}$ and assignments \mathbf{y} to minimize the sum of squared distance of the data points $\{\mathbf{x}^{(n)}\}$ to their assigned cluster centers

$$\begin{aligned} \min_{\{\boldsymbol{\mu}\}, \{\mathbf{y}\}} J(\{\boldsymbol{\mu}\}, \{\mathbf{y}\}) &= \min_{\{\boldsymbol{\mu}\}, \{\mathbf{y}\}} \sum_{n=1}^N \sum_{k=1}^K y_k^{(n)} \|\boldsymbol{\mu}_k - \mathbf{x}^{(n)}\|^2 \\ \text{s.t. } \sum_k y_k^{(n)} &= 1, \forall n, \text{ where } y_k^{(n)} \in \{0, 1\}, \forall k, n \end{aligned}$$

where $y_k^{(n)} = 1$ means that $\mathbf{x}^{(n)}$ is assigned to cluster k (with center $\boldsymbol{\mu}_k$).

K-means from MLE Perspective

- K-means Objective:

Find cluster centers μ and assignments y to minimize the sum of squared distance of the data points $\{\mathbf{x}^{(n)}\}$ to their assigned cluster centers

$$\begin{aligned} \min_{\{\mu\}, \{y\}} J(\{\mu\}, \{y\}) &= \min_{\{\mu\}, \{y\}} \sum_{n=1}^N \sum_{k=1}^K y_k^{(n)} \|\mu_k - \mathbf{x}^{(n)}\|^2 \\ \text{s.t. } \sum_k y_k^{(n)} &= 1, \forall n, \text{ where } y_k^{(n)} \in \{0, 1\}, \forall k, n \end{aligned}$$

where $y_k^{(n)} = 1$ means that $\mathbf{x}^{(n)}$ is assigned to cluster k (with center μ_k).

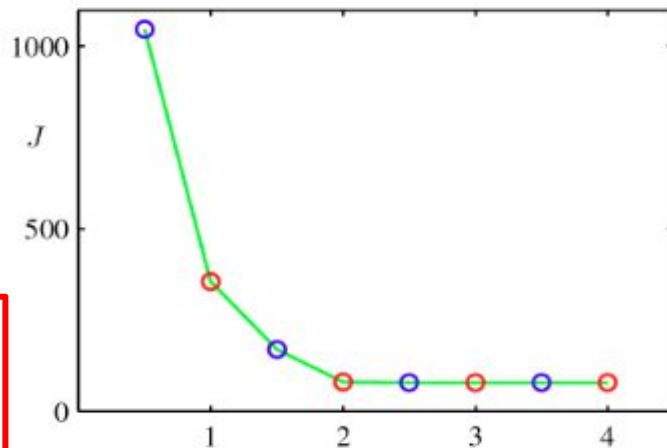
$$\begin{aligned} \max_{y_j^i} \max_{\pi, \mu, \Sigma} & \sum_{i=1}^m \sum_{j=1}^k y_j^i \log \pi_j - \sum_{i=1}^m \log Z - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^k y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j) \\ & \text{subject to } \sum_{j=1}^k \pi_j = 1 \end{aligned}$$

Convergence of k-means

$$\begin{aligned} \min_{\{\boldsymbol{\mu}\}, \{\mathbf{y}\}} J(\{\boldsymbol{\mu}\}, \{\mathbf{y}\}) &= \min_{\{\boldsymbol{\mu}\}, \{\mathbf{y}\}} \sum_{n=1}^N \sum_{k=1}^K y_k^{(n)} \|\boldsymbol{\mu}_k - \mathbf{x}^{(n)}\|^2 \\ \text{s.t. } \sum_k y_k^{(n)} &= 1, \forall n, \text{ where } y_k^{(n)} \in \{0, 1\}, \forall k, n \end{aligned}$$

where $y_k^{(n)} = 1$ means that $\mathbf{x}^{(n)}$ is assigned to cluster k (with center $\boldsymbol{\mu}_k$).

K-means can be understood as hard-GMM
GMM can be understood as soft k-means

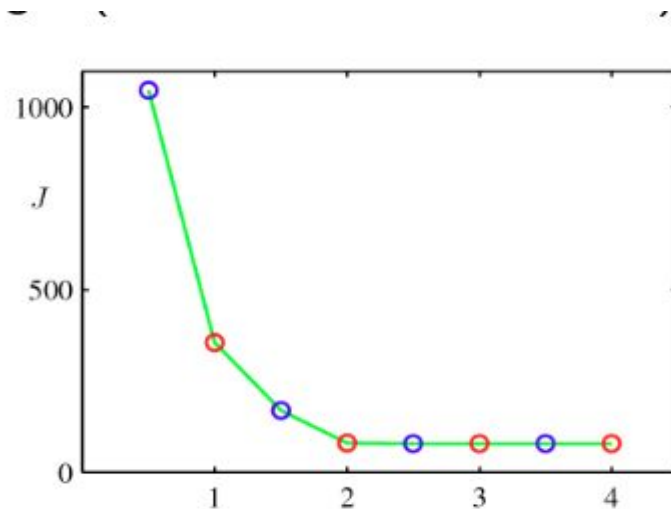


Optimization View of k-means

$$\begin{aligned} \min_{\{\boldsymbol{\mu}\}, \{\mathbf{y}\}} J(\{\boldsymbol{\mu}\}, \{\mathbf{y}\}) &= \min_{\{\boldsymbol{\mu}\}, \{\mathbf{y}\}} \sum_{n=1}^N \sum_{k=1}^K y_k^{(n)} \|\boldsymbol{\mu}_k - \mathbf{x}^{(n)}\|^2 \\ \text{s.t. } \sum_k y_k^{(n)} &= 1, \forall n, \text{ where } y_k^{(n)} \in \{0, 1\}, \forall k, n \end{aligned}$$

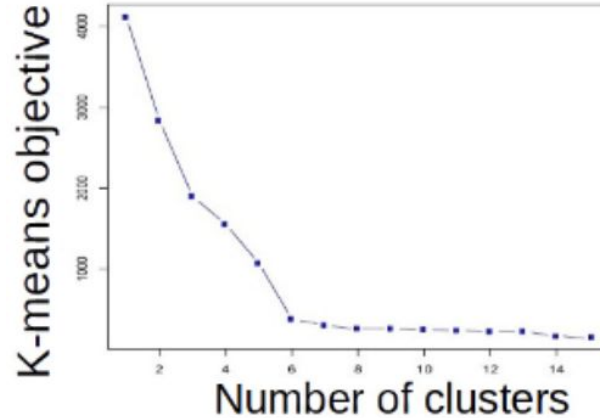
where $y_k^{(n)} = 1$ means that $\mathbf{x}^{(n)}$ is assigned to cluster k (with center $\boldsymbol{\mu}_k$).

- Optimization method is a form of **coordinate descent** ("block coordinate descent")
 - Fix centers, optimize assignments (choose cluster whose mean is closest)
 - Fix assignments, optimize means (average of assigned datapoints)
- Each iteration of K-means algorithm decrease the objective
- Note: The algorithm usually converges to a **local minima** (though may not always, and it may just convergence "somewhere"). Multiple runs with different initializations can be tried to find a good solution.



Hyperparameters: Choosing K

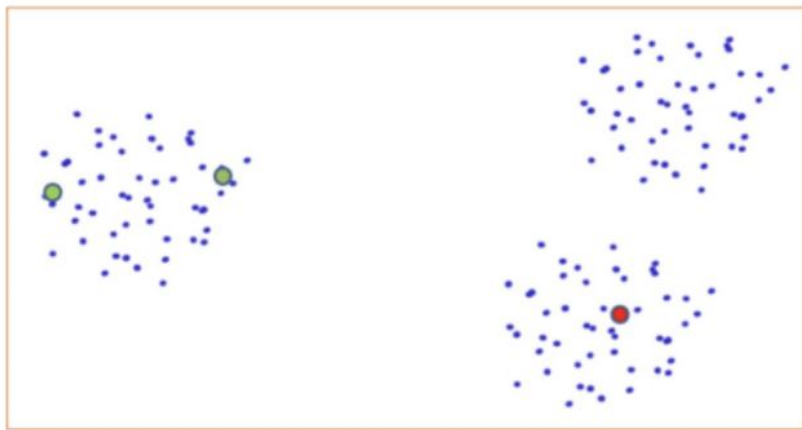
- One way to select K for the K-means algorithm is to try different values of K , plot the K-means objective versus K , and look at the “elbow-point”.



- For the above plot, $K= 6$ is the elbow-point.

Hyperparameter: Initialization

- The results of the K-means algorithm can vary based on initial placement of centers.
 - Some placements can in poor convergence rate, or convergence to sub-optimal clustering
→ K-means can easily get stuck in **local minima** (of optimization landscape)



Convergence (to the wrong clustering) in one iteration

K-means Applications: Data Compression

$K = 2$



$K = 3$



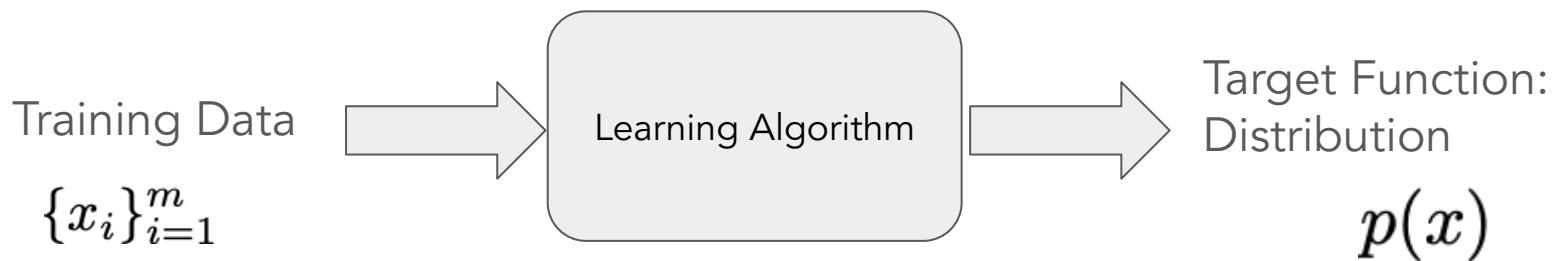
$K = 10$



Original image



K-means is **Approximating** Gaussian Mixture Model



Density Estimation Pipeline

1. Build probabilistic models
Gaussian Mixture Model with fixed isotropic covariance
2. Derive loss function (by MLE or MAP....)
Approximated MLE
3. Select optimizer
EM

Ambiguity in Clustering



Clustering is subjective



Simpson's Family



School Employees



Females



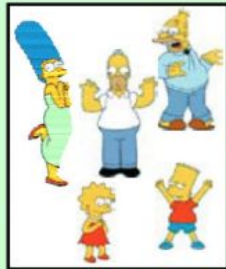
Males

Ambiguity in Clustering



What is consider similar/dissimilar?

Clustering is subjective



Simpson's Family



School Employees



Females



Males

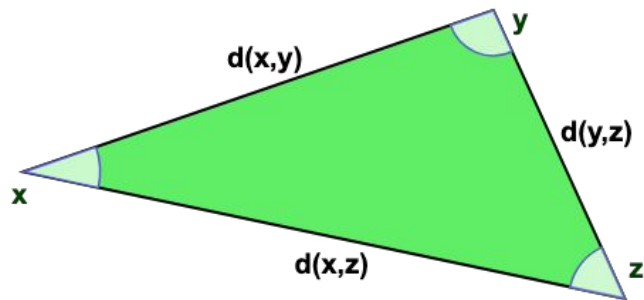
Generalization of K-means

- Given m data points, $\{x^1, \dots, x^n\}$
- Find k cluster centers, $\{\mu_1, \dots, \mu_k\}$
- And assign each data point i to one cluster, $y^i \in \{1, \dots, k\}$
- Such that the sum of the squared distances from each data point to its respective cluster center is minimized

$$\min_{\{\mu\}, \{y\}} \sum_{n=1}^N \sum_{k=1}^K y_k^{(n)} \|\mu_k - \mathbf{x}^{(n)}\|^2 \longrightarrow \min_{\mu, y} \sum_{i=1}^n d(x^i, \mu_{y^i})$$
$$\text{s.t. } \sum_k y_k^{(n)} = 1, \forall n, \text{ where } y_k^{(n)} \in \{0, 1\}, \forall k, n$$

What similarity/dissimilarity function or distance function

- Desired properties of dissimilarity function
 - Symmetry: $d(x, y) = d(y, x)$
 - *Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex."*
 - Positive separability: $d(x, y) = 0$, if and only if $x = y$
 - *Otherwise there are objects that are different, but you cannot tell apart.*
 - Triangular inequality: $d(x, y) \leq d(x, z) + d(z, y)$



Distance functions for vectors

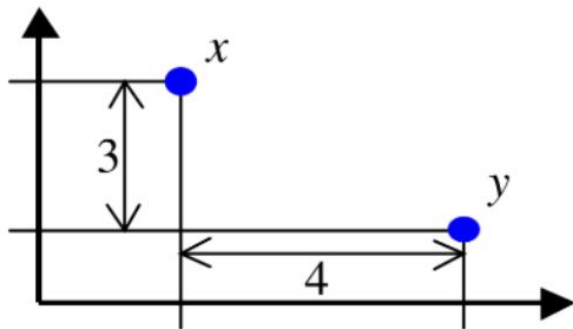
- Suppose two data points, both in \mathbb{R}^n

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T$$

$$\mathbf{y} = (y_1, y_2, \dots, y_n)^T$$

- Euclidean distance: $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
- Minkowski distance: $d(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^n |x_i - y_i|^p)^{\frac{1}{p}}$
 - Euclidean distance: $p = 2$
 - Manhattan distance: $p = 1, d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i|$
 - "inf"-distance: $p = \infty, d(\mathbf{x}, \mathbf{y}) = \max_{i=1}^n |x_i - y_i|$

Distance example



Euclidian distance: $\sqrt{4^2 + 3^2} = 5$

Manhattan distance: $4 + 3 = 7$

“inf”-distance: $\max\{4,3\} = 4$

Hamming distance

- Manhattan distance is also called *Hamming* distance when all features are binary
 - Count the number of difference between two binary vectors
 - Example, $x, y \in \{0, 1\}^{17}$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
x	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
y	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1

$$d(x, y) = 5$$

Edit distance

- Transform one of the objects into the other, and measure how much effort it takes

<i>x</i>	I	N	T	E	*	N	T	I	O	N
<i>y</i>	*	E	X	E	C	U	T	I	O	N
	d	s	s		i	s				

d: deletion (cost 5)

s: substitution (cost 1)

i: insertion (cost 2)

$$d(x, y) = 5 \times 1 + 3 \times 1 + 1 \times 2 = 10$$

Generalized K-means algorithm

- Initialize k cluster centers, $\{c^1, c^2, \dots, c^k\}$, randomly
- Do
 - (Assignment) Decide the cluster memberships of each data point, x^i , by assigning it to the nearest cluster center

$$y^i = \arg \min_{j=1, \dots, k} d(x^i, \mu_j).$$

- (Center Update) Adjust the cluster centers

$$\mu_j = \arg \min_{v \in \mathbb{R}^n} \sum_{i: y^i=j} d(x^i, v)$$

Squared Euclidean distance:

$$\mu_j = \frac{1}{\#\{y^i = j\}} \sum_{i: y^i=j} x^i$$

- While any cluster center has been changed

Generalized K-means algorithm



Generalized
Mixture Models

- Initialize k cluster centers, $\{c^1, c^2, \dots, c^k\}$, randomly
- Do
 - (Assignment) Decide the cluster memberships of each data point, x^i , by assigning it to the nearest cluster center

$$y^i = \arg \min_{j=1, \dots, k} d(x^i, \mu_j).$$

- (Center Update) Adjust the cluster centers

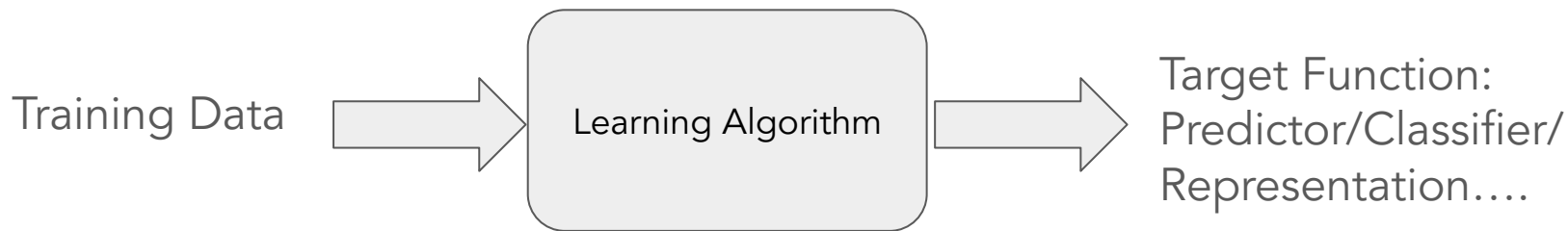
$$\mu_j = \arg \min_{v \in \mathbb{R}^n} \sum_{i: y^i = j} d(x^i, v)$$

Squared Euclidean distance:

$$\mu_j = \frac{1}{\#\{y^i = j\}} \sum_{i: y^i = j} x^i$$

- While any cluster center has been changed

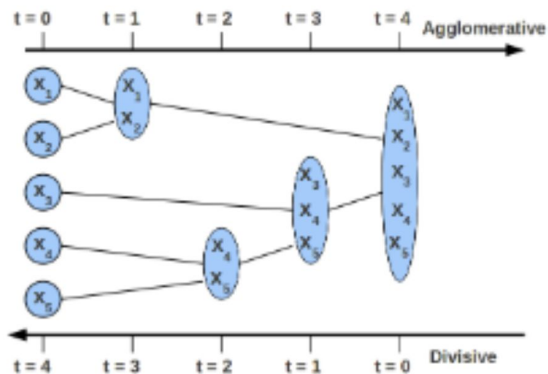
ML Algorithm Pipeline



General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP....)
3. Select optimizer

Hierarchical Clustering



- Agglomerative (bottom-up) Clustering
 1. Start with each example in its own **singleton cluster**
 2. At each time-step, greedily **merge** 2 most similar clusters
 3. Stop when there is a single cluster of all examples, else go to 2.
- Divisive (top-down) Clustering
 1. Start with all examples in the same cluster
 2. At each time-step, remove the "outsiders" from the **least cohesive cluster**
 3. Stop when each example is in its own singleton cluster, else go to 2

Bottom up hierarchical clustering

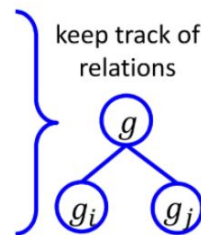
- Assign each data point to its own cluster:

$$g_1 = \{x_1\}, g_2 = \{x_2\}, \dots, g_m = \{x_m\}, \text{ and let } G = \{g_1, g_2, \dots, g_m\}$$

- Do

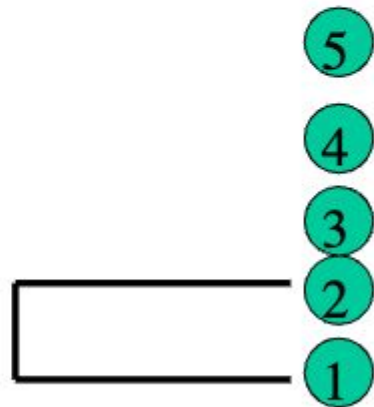
- Find two clusters to merge: $i, j = \arg \min_{1 \leq i, j \leq |G|} D(g_i, g_j)$
- Merge the two clusters to a new cluster: $g \leftarrow g_i \cup g_j$
- Remove the merged clusters: $G \leftarrow G \setminus g_i, \quad G \leftarrow G \setminus g_j$
- Add the new cluster: $G \leftarrow G \cup \{g\}$

$$D(g_i, g_j) = \min_{x \in g_i, y \in g_j} d(x, y)$$



- While $|G| > 1$

$$\begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \\ 1 \left[\begin{array}{ccccc} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{array} \right] \end{array}$$

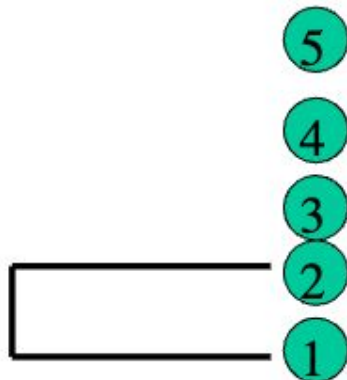


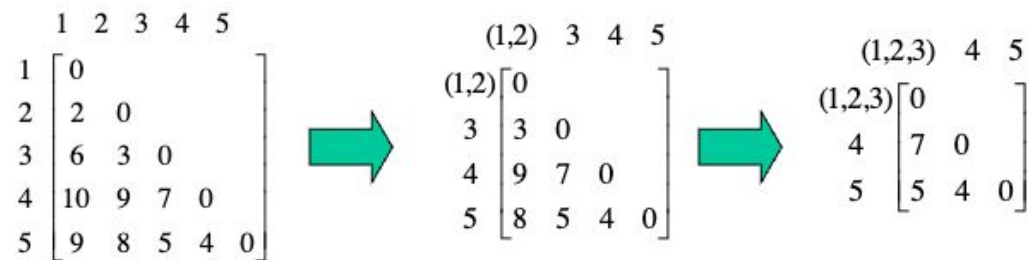
$$\begin{array}{c}
 \begin{array}{ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
 1 & [& 0 & & & \\
 2 & [& 2 & 0 & & \\
 3 & [& 6 & 3 & 0 & \\
 4 & [& 10 & 9 & 7 & 0 \\
 5 & [& 9 & 8 & 5 & 4 & 0]
 \end{array}
 & \rightarrow &
 \begin{array}{c}
 \begin{array}{ccccc}
 & (1,2) & 3 & 4 & 5 \\
 (1,2) & [& 0 & & & \\
 3 & [& 3 & 0 & & \\
 4 & [& 9 & 7 & 0 & \\
 5 & [& 8 & 5 & 4 & 0]
 \end{array}
 \end{array}
 \end{array}$$

$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6, 3\} = 3$$

$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10, 9\} = 9$$

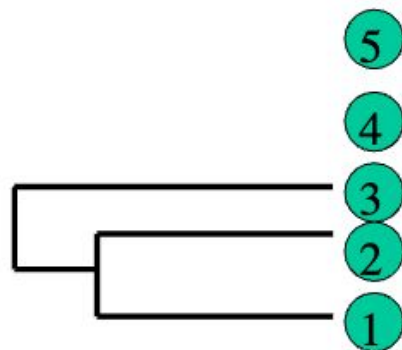
$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9, 8\} = 8$$

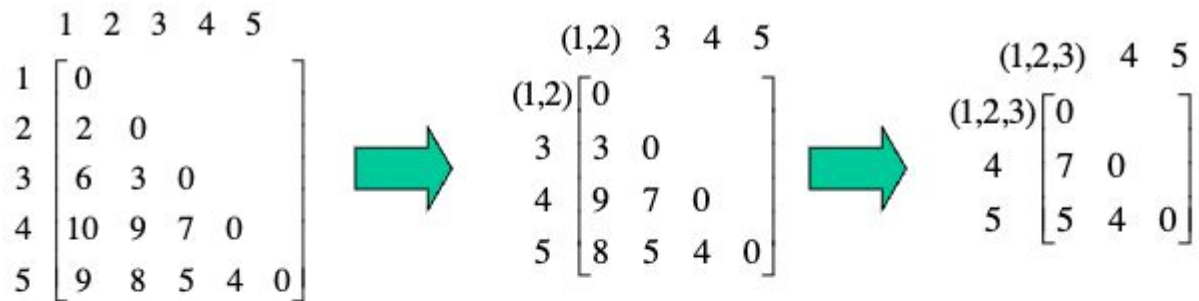




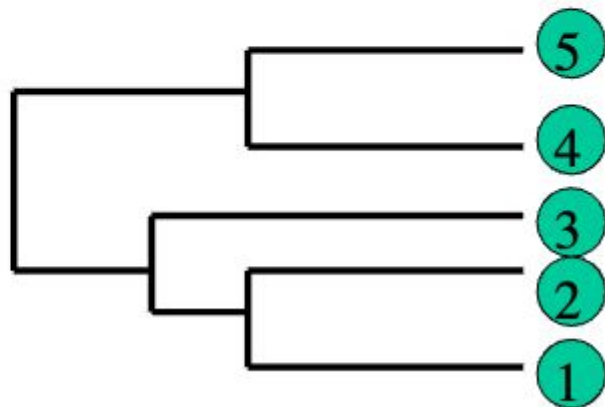
$$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9, 7\} = 7$$

$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8, 5\} = 5$$





$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$



Q&A