Lecture 17: Clustering

Fall 2022

Kai-Wei Chang

CS @ UCLA

kw+cm146@kwchang.net

The instructor gratefully acknowledges Dan Roth, Vivek Srikuar, Sriram Sankararaman, Fei Sha, Ameet Talwalkar, Eric Eaton, and Jessica Wu whose slides are heavily used, and the many others who made their course material freely available online.

Announcement

- Quiz due today
- The practice final will be released
- Hw1/Midterm regrading

Clustering

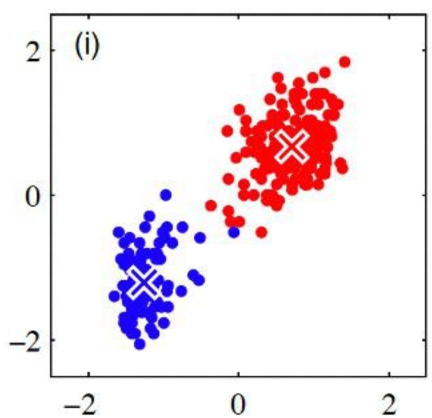
Goal of Clustering

- Given a collection of data points, the goal is to find structure in the data: organize that data into sensible groups.

- Applications
  - Topics in news articles
  - Identify communities within social networks

How to define clusters?

- A set of entities that are “alike”
- May be described as connected regions of a multi-dimensional space



How to define clusters?

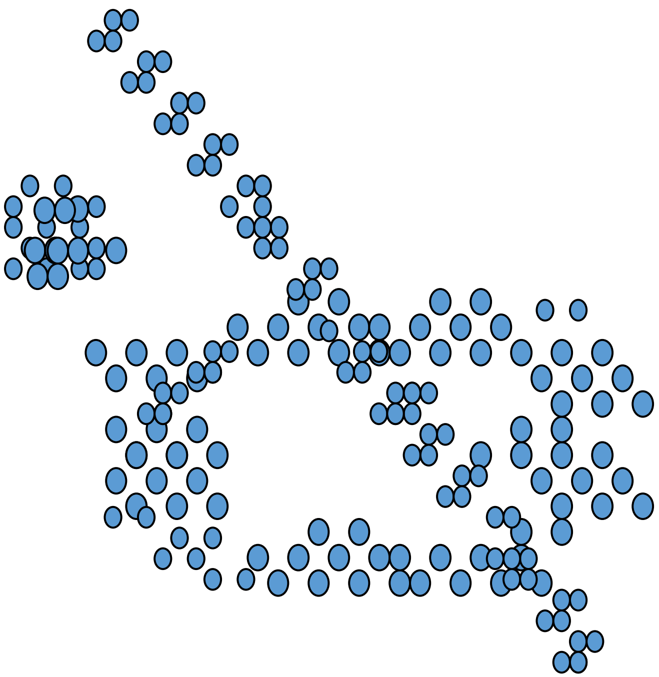







How to define a cluster?

How many clusters do we have?



Today’s lecture

- K-Means

- K-Medoids

- GMM (probabilistic version)

K-Means

Hogwarts (Harry Potter)

- Sorting Hat – cluster kids into four groups based on four underlying prototypes

Godric Gryffindor

Helga Hufflepuff

Rowena Ravenclaw

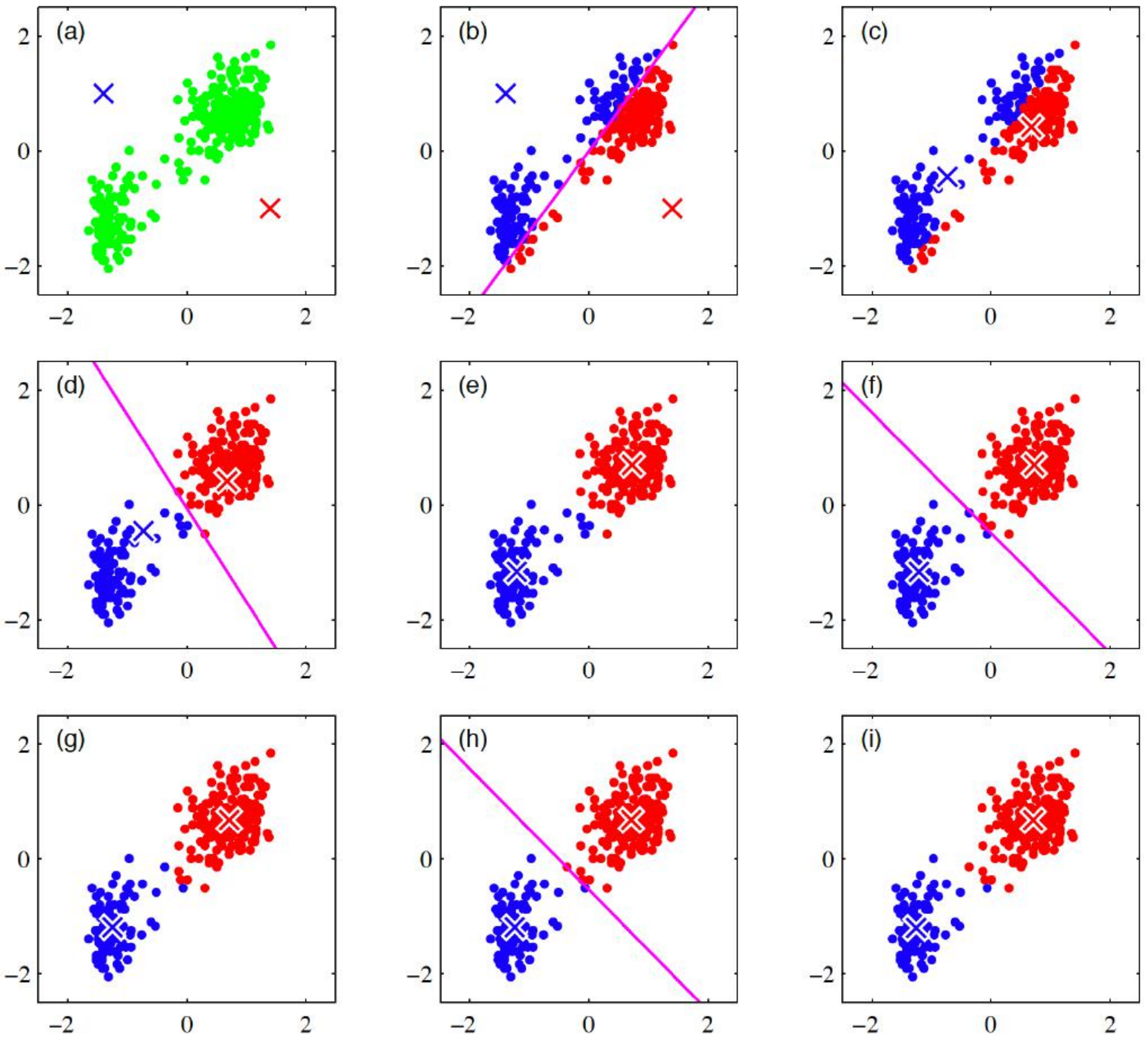
Salazar Slytherin

K-Means Intuition

- Sorting Hat – cluster kids into four groups based on four underlying prototypes

- The prototype of each house is *the average of all kids of the house*
- Algorithm*:*
  Alternatively, updating the prototype & the cluster assignment

Intuition of K-Means



http://shabal.in/visuals/kmeans/6.html

Problem setting

- An optimization problem:
  - Given $D = \{x_n\}_{n=1}^{N}$ and a number K, we want to group data point to K clusters
  - $A(x) \in \{1,2,\dots K\}$: the cluster membership
  - $r_{nk} \in \{0,1\}$ indicates whether $A(x_n) = k$

Problem setting

- An optimization problem:
  - Given $D = \{x_n\}_{n=1}^{N}$ and a number K, we want to group data point to K clusters
  - $A(x) \in \{1,2, \dots K\}$: the cluster membership
  - $r_{nk} \in \{0,1\}$ indicates whether $A(x_n) = k$

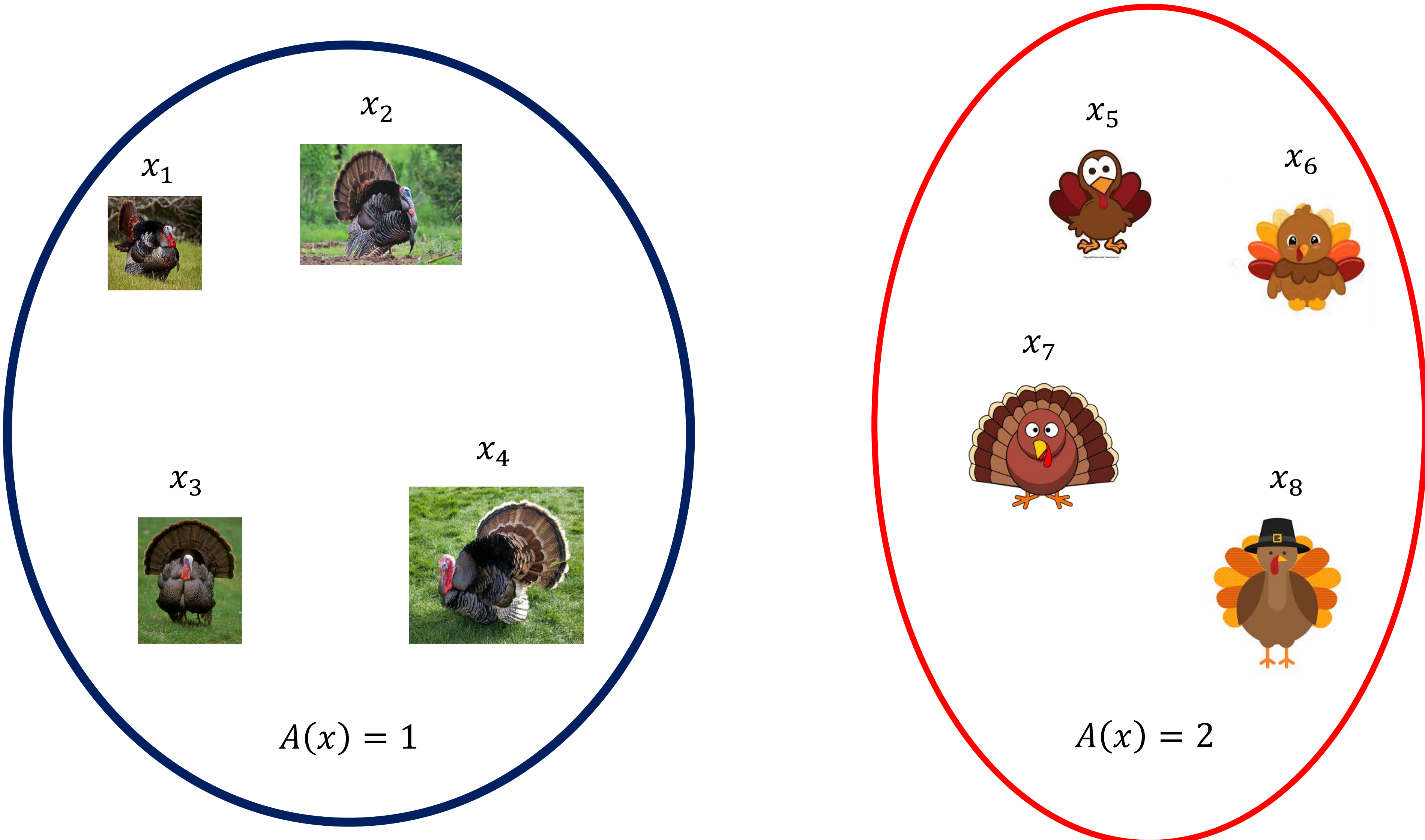
x_1
$A(x_1) = 1$
$r_{11} = 1$
$r_{12} = 0$

x_2
$A(x_2) = 2$
$r_{21} = 0$
$r_{22} = 1$

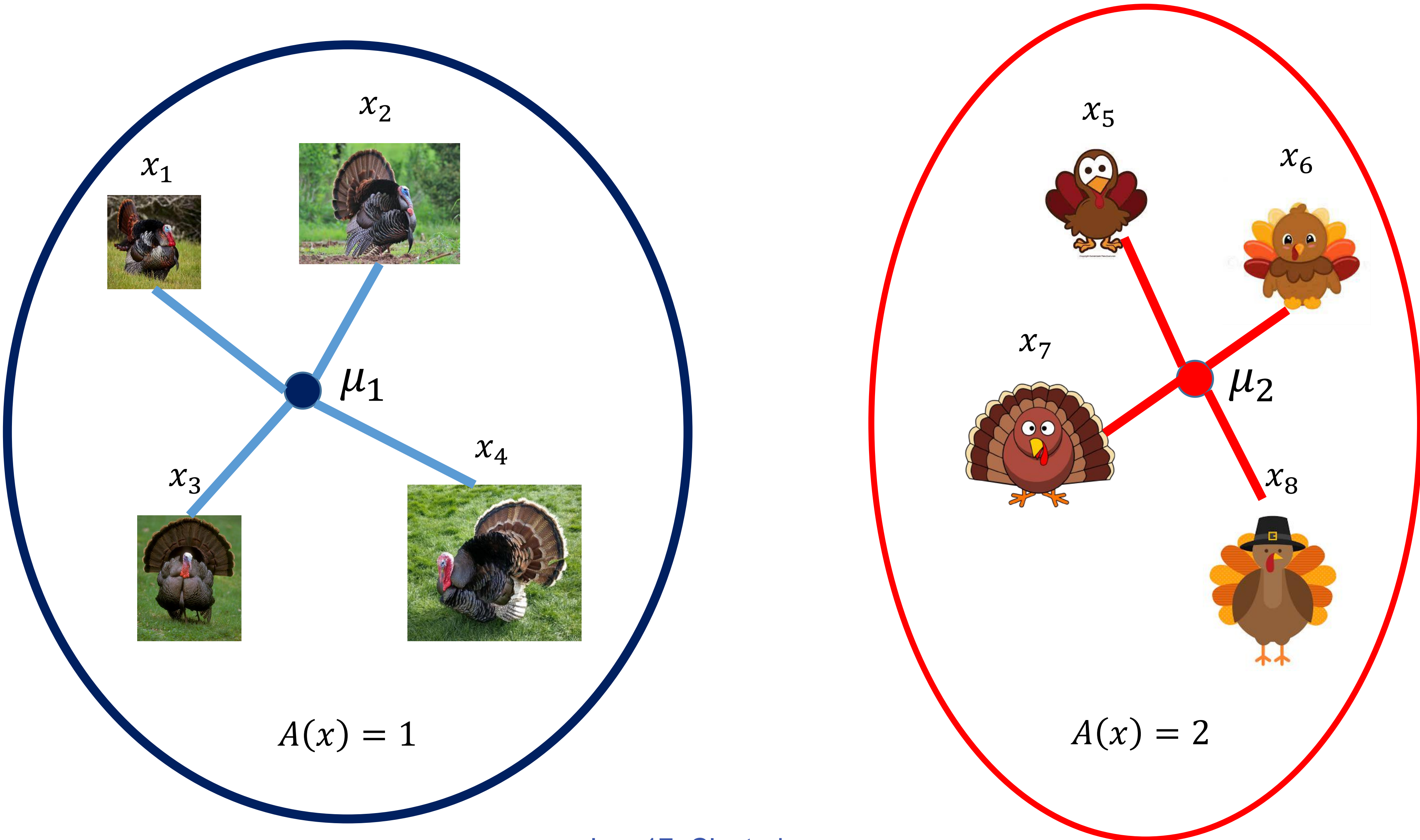
x_3
$A(x_3) = 1$
$r_{31} = 1$
$r_{32} = 0$

K-Means clustering

K-Means clustering

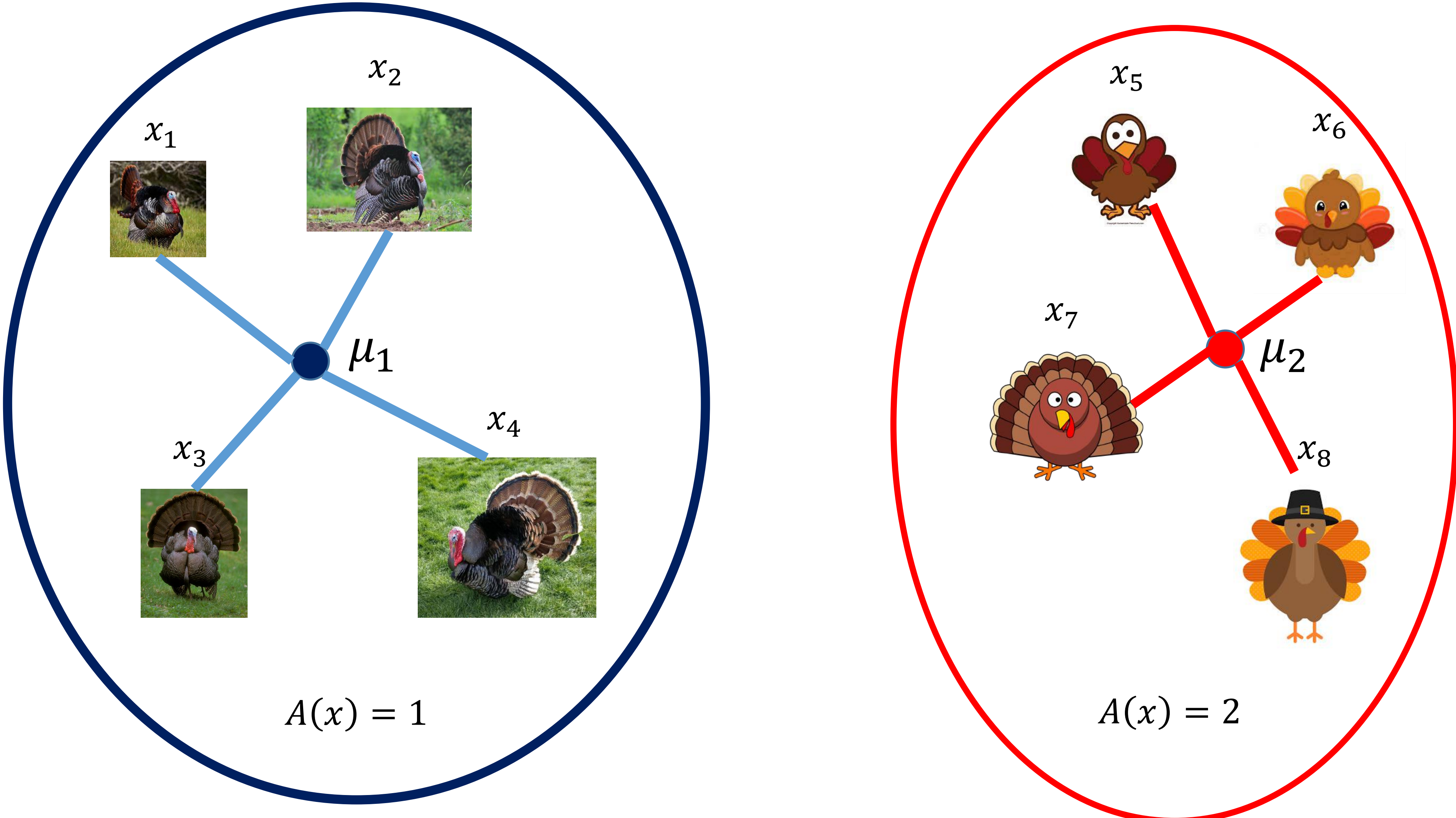


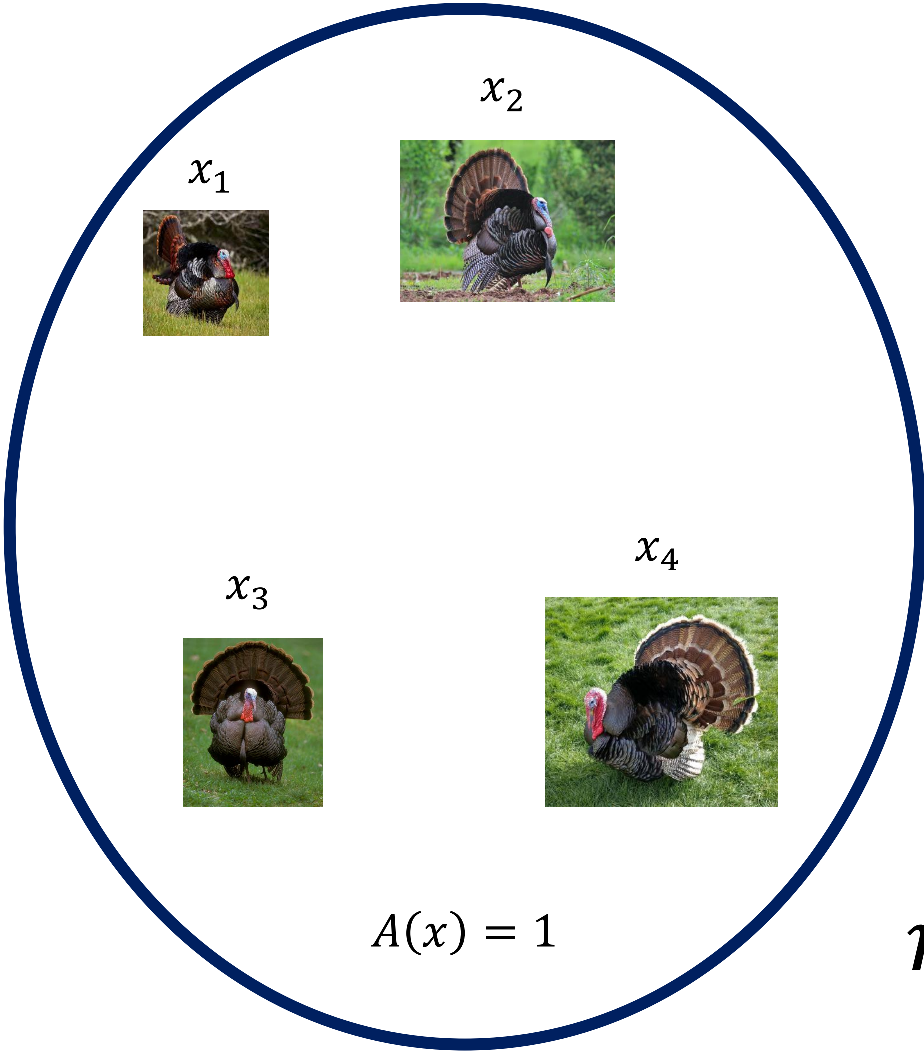
K-Means clustering



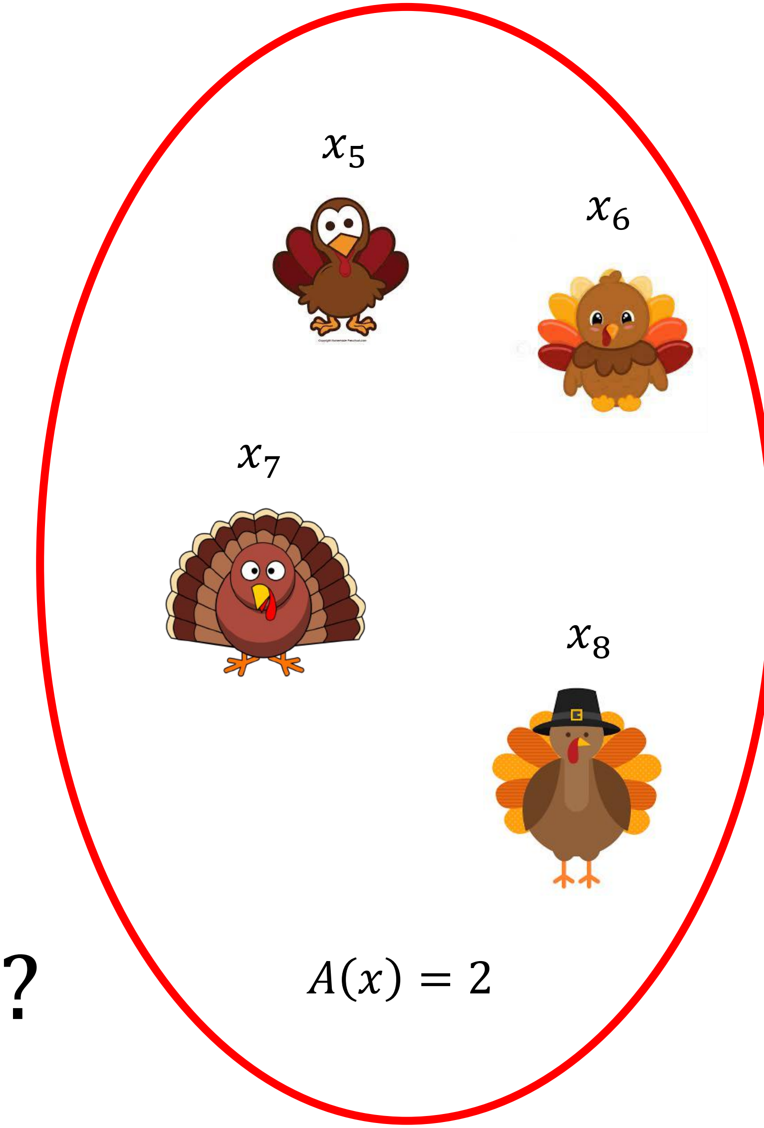
Quantify the clustering quality by $\sum_{n=1}^{4} ||x_n - \mu_1||^2 + \sum_{n=5}^{8} ||x_n - \mu_2||^2$

Mean square distance

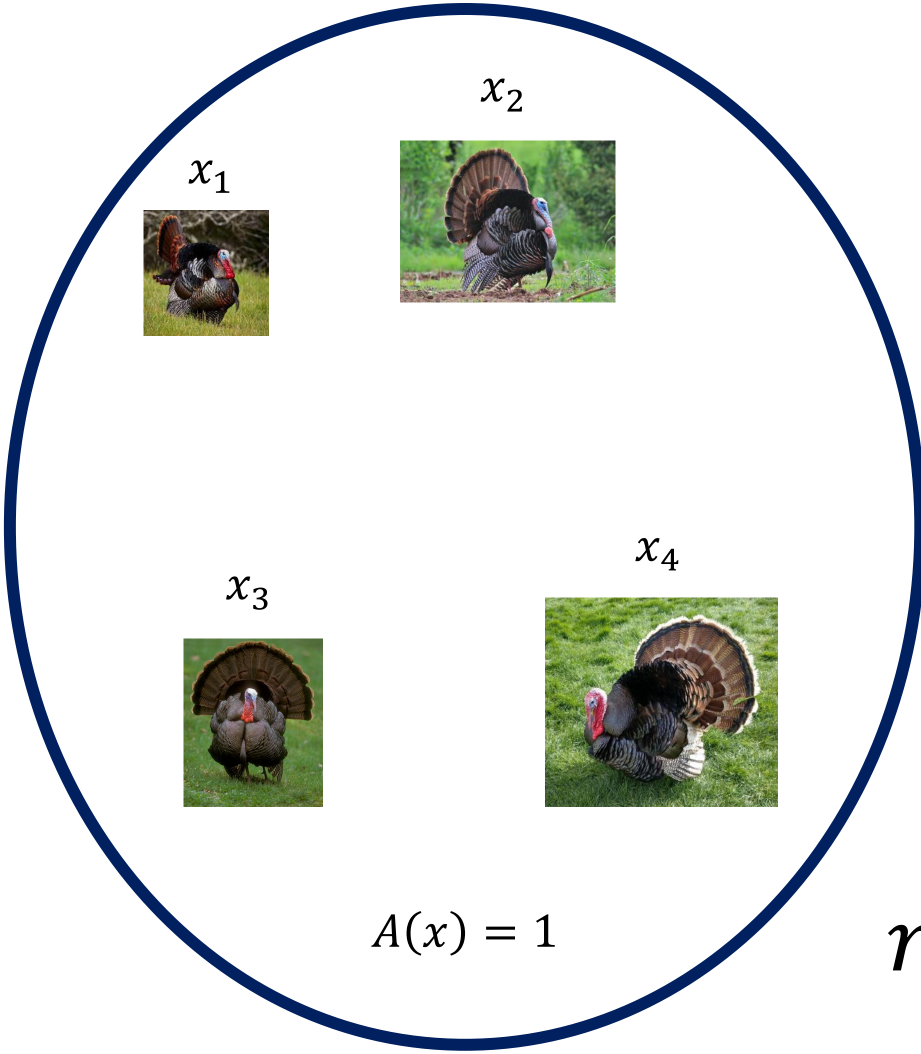




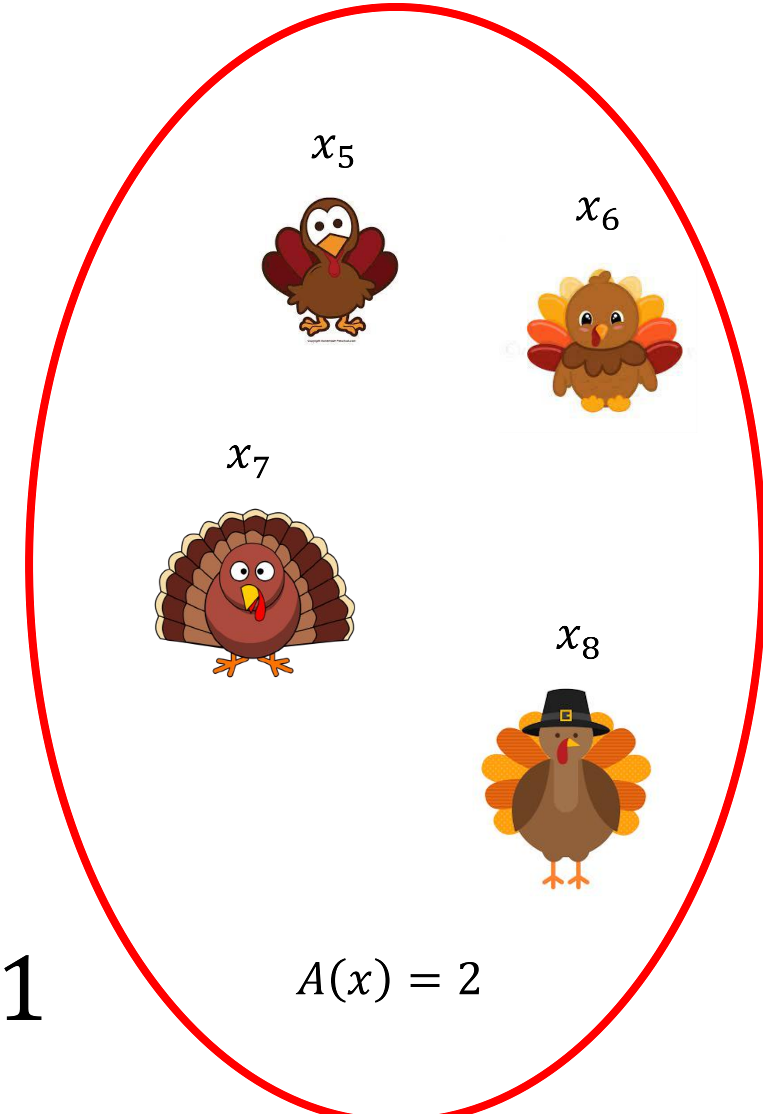
$$r_{31} = ?$$



Quantify the clustering quality by $\sum_{n=1}^{8} \sum_{k=1}^{2} r_{nk} \, ||x_n - \mu_k||^2$



$$r_{31} = 1$$



K-means clustering

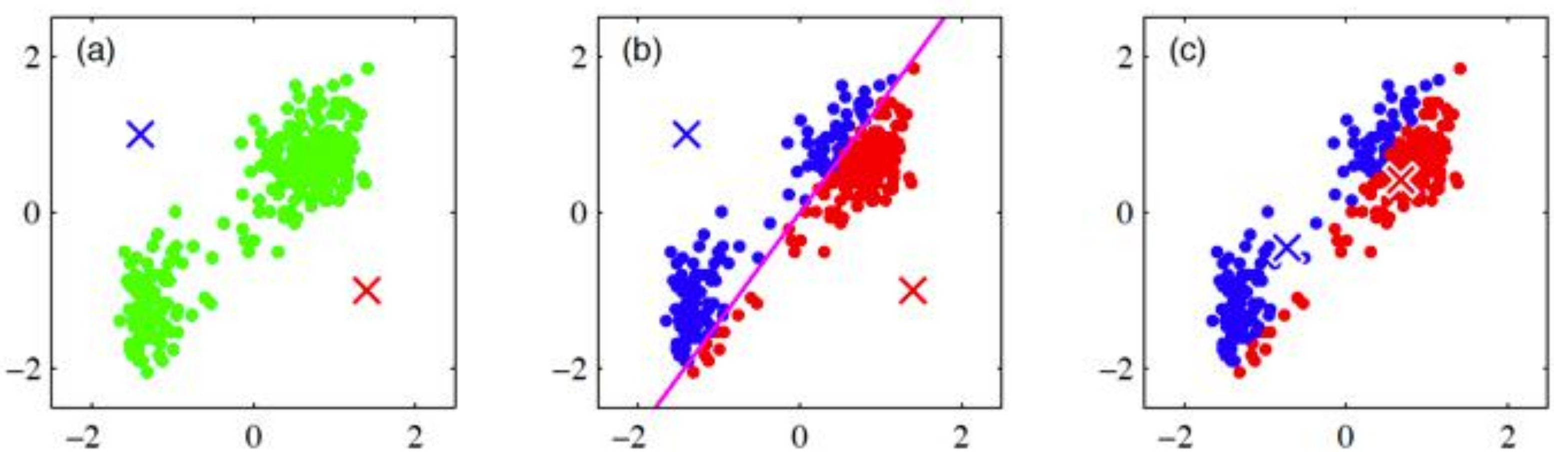
Sum of distances of all the points to their cluster center

- Distortion measure (loss function for clustering)

$$J(\{r_{nk}\}, \{\boldsymbol{\mu}_k\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|_2^2$$

where $r_{nk} \in \{0, 1\}$ is an indicator variable

$$r_{nk} = 1 \quad \text{if and only if} \quad A(\boldsymbol{x}_n) = k$$



K-means objective

$$argmin_{\{r_{nk}\},\{\boldsymbol{\mu}_k\}} J(\{r_{nk}\},\{\boldsymbol{\mu}_k\}) = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|_2^2$$

where $r_{nk} \in \{0,1\}$ is an indicator variable

$$r_{nk} = 1 \quad \text{if and only if} \quad A(\boldsymbol{x}_n) = k$$

- It is a non-convex objective function
- Minimizing the above objective is NP-hard.

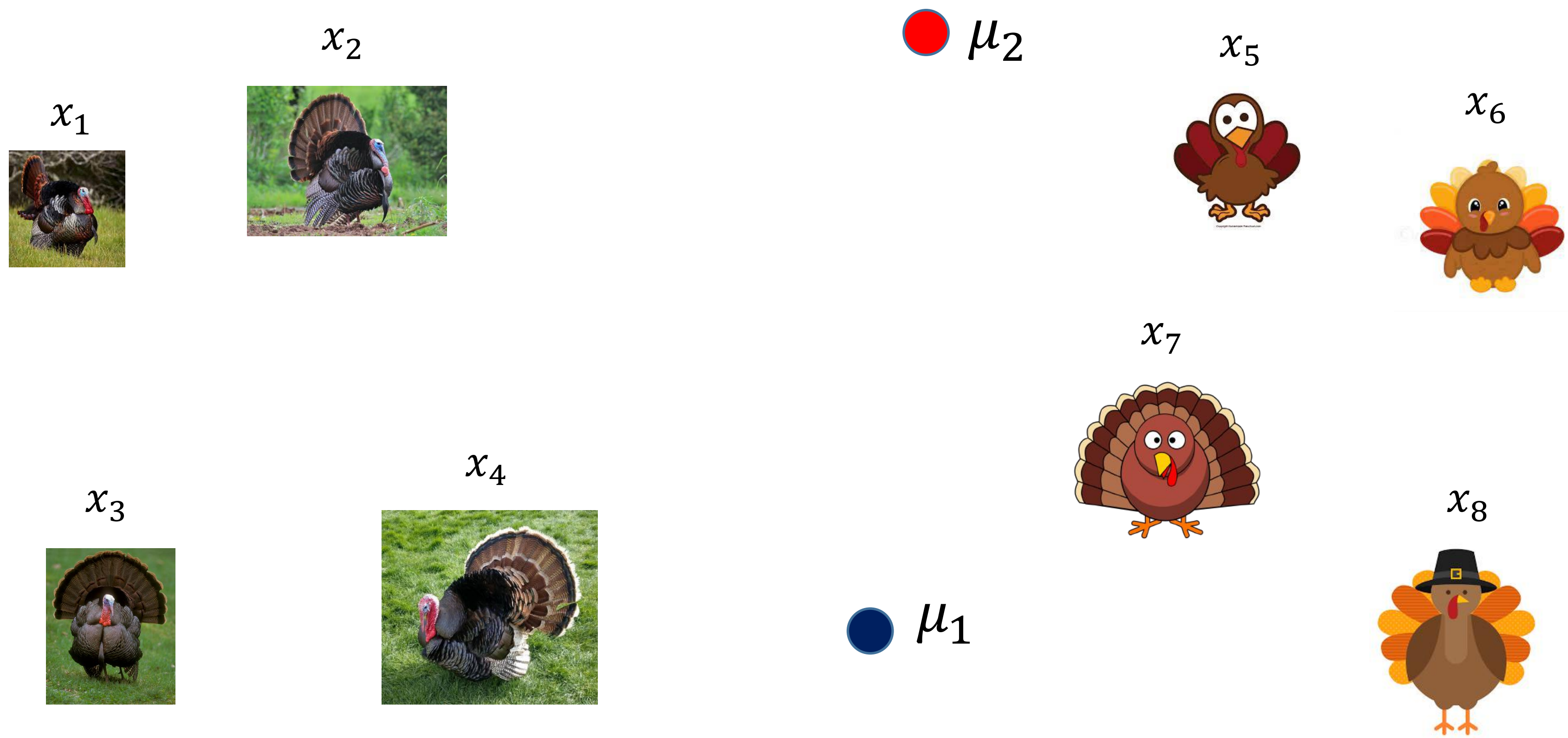
K-means algorithm a.k.a Llyod's algorithm

- A greedy algorithm for minimizing K-means objective – alternative update $\{r_{nk}\}, \{\boldsymbol{\mu}_k\}$
- Step 0: randomly assign the cluster centers $\{\mu_k\}$
- Step 1: Minimize J over $\{r_{nk}\}$ -- reassign cluster member
- Step 2: Minimize J over $\{\mu_k\}$ -- update the cluster centers
- Loop until it converges

$$\boxed{J(\{r_{nk}\}, \{\boldsymbol{\mu}_k\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|_2^2}$$

Step 1: Minimize J over $\{r_{nk}\}$

$$J(\{r_{nk}\}, \{\boldsymbol{\mu}_k\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|_2^2$$

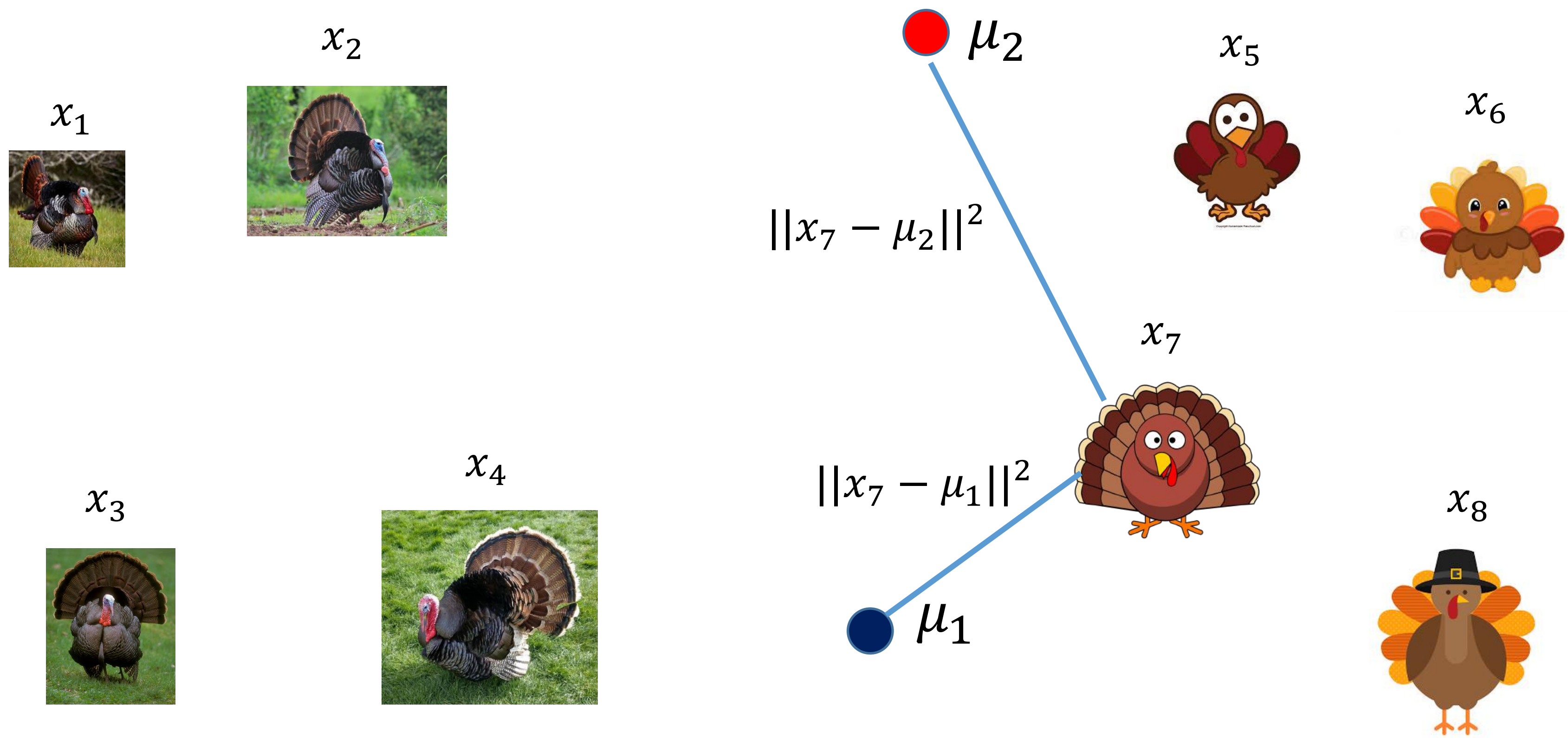


$A(x) = 1$

$A(x) = 2$

Step 1: Minimize J over $\{r_{nk}\}$

$$J(\{r_{nk}\}, \{\boldsymbol{\mu}_k\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|_2^2$$



$A(x) = 1$ $\qquad\qquad$ $A(x) = 2$

K-means algorithm
a.k.a Llyod's algorithm

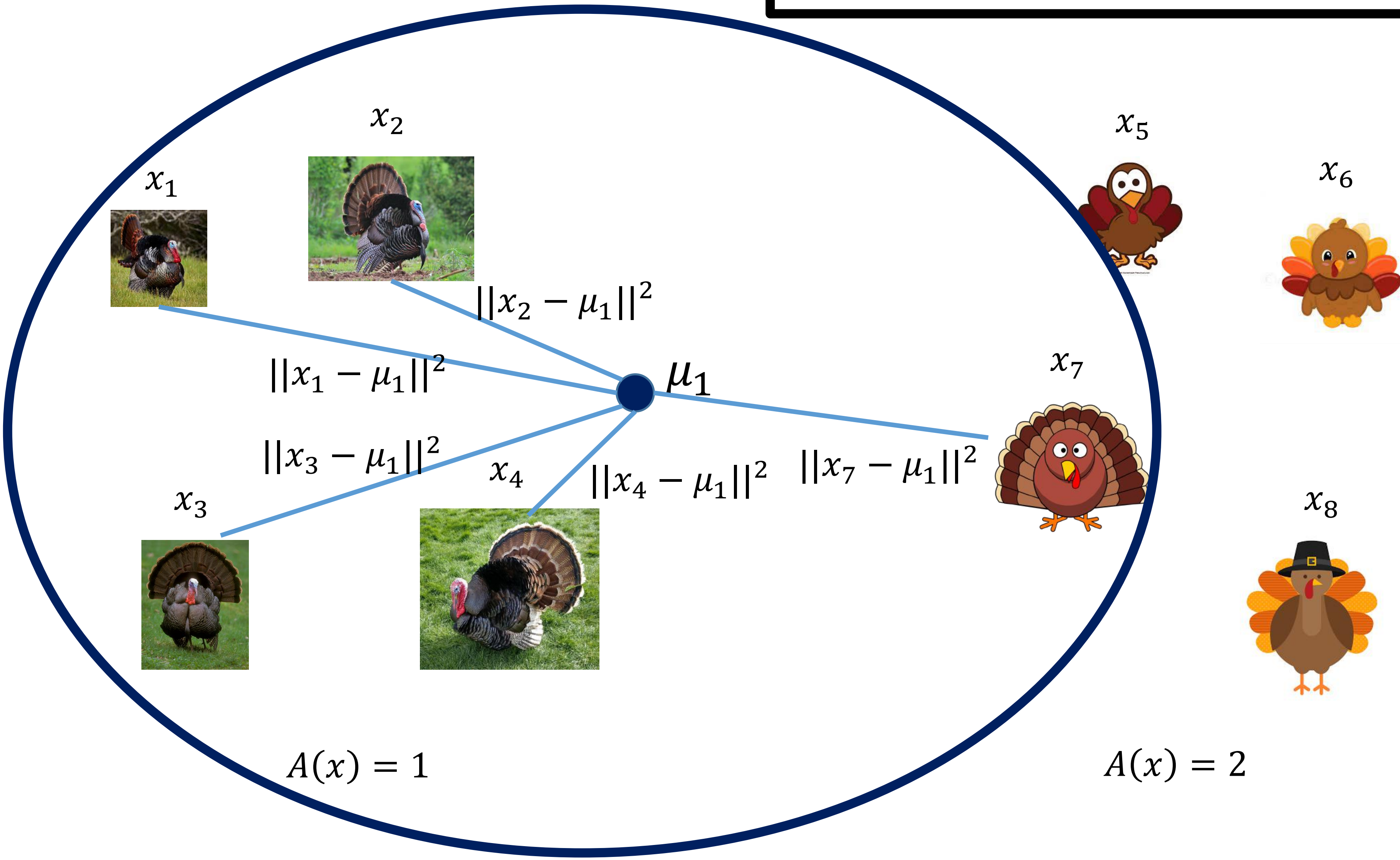
- A greedy algorithm for minimizing K-means objective – alternative update $\{r_{nk}\}, \{\boldsymbol{\mu}_k\}$
- Step 0: randomly assign the cluster centers $\{\mu_k\}$
- Step 1: Minimize J over $\{r_{nk}\}$ -- reassign cluster member

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Step 2: Minimize J over $\{\mu_k\}$ -- update the cluster centers
- Loop until it converges

Step 2: Minimize J over $\{\mu_k\}$

$$J(\{r_{nk}\}, \{\boldsymbol{\mu}_k\}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|_2^2$$



K-means algorithm
a.k.a Llyod's algorithm

- Step 0: randomly assign the cluster centers $\{\mu_k\}$
- Step 1: Minimize J over $\{r_{nk}\}$ -- Assign every point to the closest cluster center

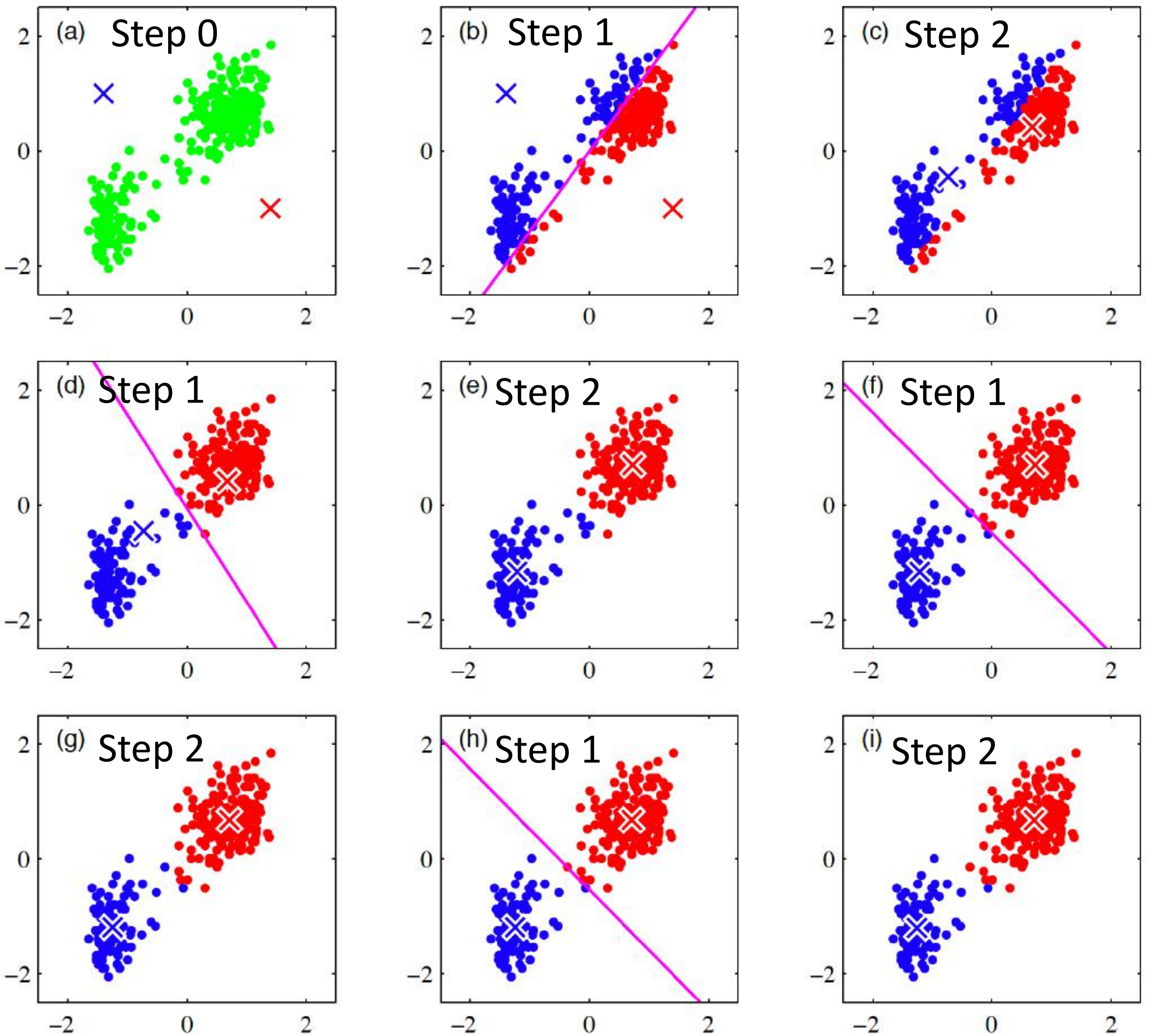
$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Step 2: Minimize J over $\{\mu_k\}$ -- update the cluster centers

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\boldsymbol{x}_n}{\sum_n r_{nk}}$$

- Loop until it converges

Example



(a) Step 0
(b) Step 1
(c) Step 2
(d) Step 1
(e) Step 2
(f) Step 1
(g) Step 2
(h) Step 1
(i) Step 2

Remarks

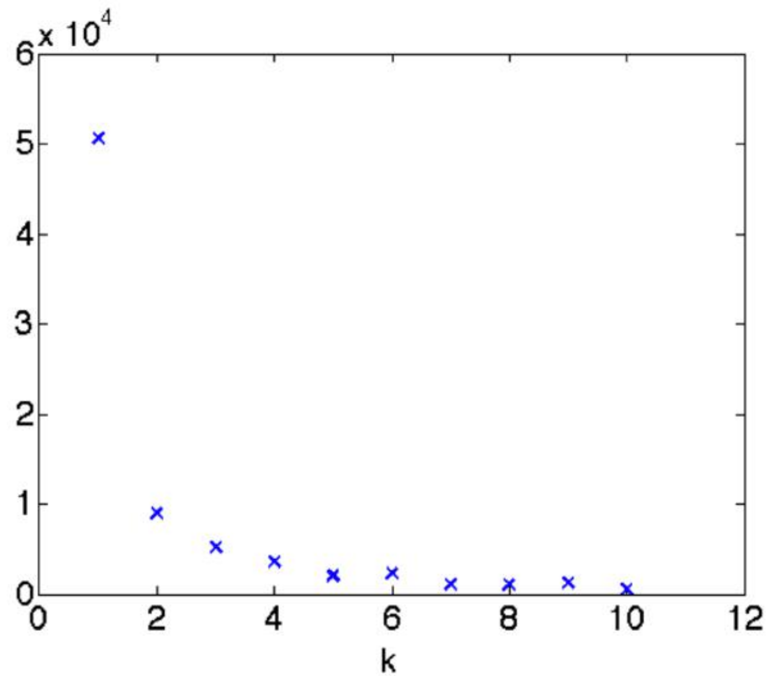
- Prototype μ_k is the mean of data points assigned to the cluster k, hence `K-means'
- μ_k may not in the training set
- Need to pre-define K
  - There are some other approaches for the case k is unknown – not cover in class
- The procedure reduces J in both Step 1 and Step 2 and thus makes improvements or stay the same on each iteration

Properties of the K-means algorithm

- Does the K-means algorithm converge
  - Yes
- How long does it take to converge?
  - In the worst case, exponential in the number of data points
  - In practice, usually quick
- How good is its solution?
  - Local minimum (depends on the initialization)

Choosing K

- Increasing K will always decrease the optimal value of the K-means objective
  - It doesn't mean a better clustering
  - Analogous to overfitting in supervised learning.



K-means can be sensitive to the outlier

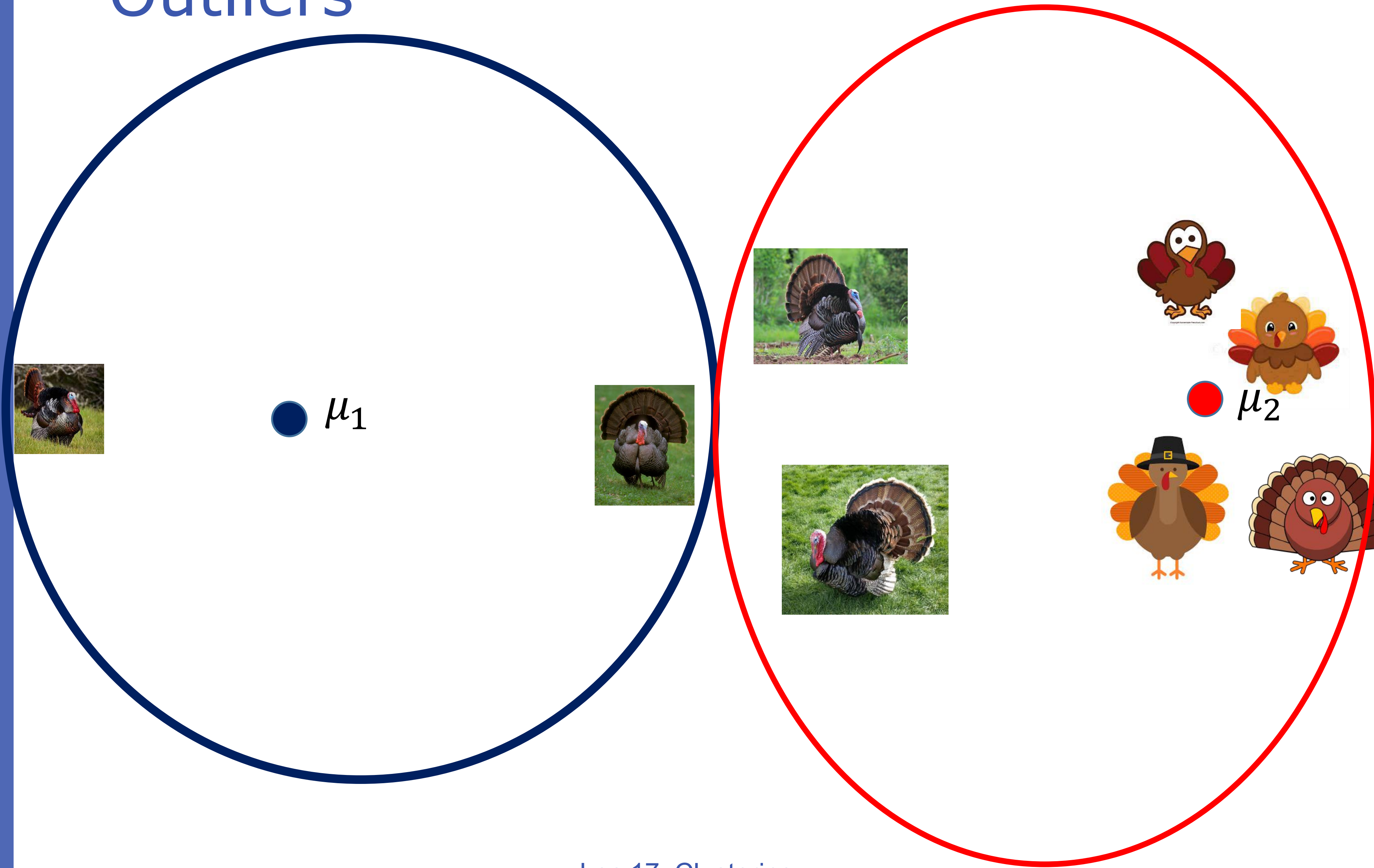
- One data point can make the center shift

K-Medoids

K-medoids

- K-means is sensitive to outliers.
- In some applications we want the prototypes to be one of the points.
- Leads to K-medoids.

Outliers

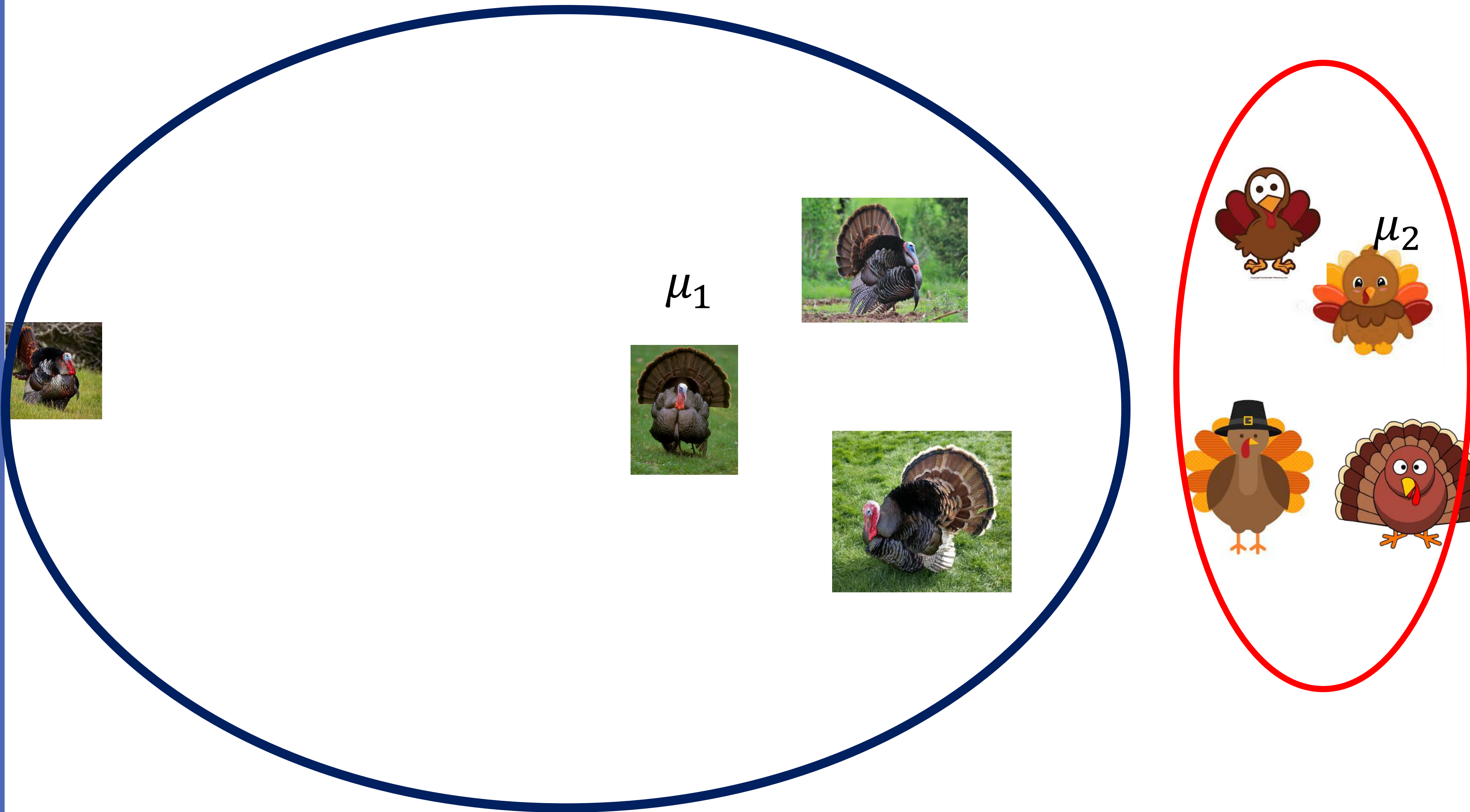


Intuition@ Hogwarts

- Sorting Hat – cluster students into four groups based on four underlying prototypes

- The prototype of each house is the most represented student of the house
  - Alternatively, updating the prototype & the student assignment

K-medoids



K-medoids algorithm

- Step 0: randomly selecting K points as the cluster centers $\{\mu_k\}$
- Step 1: Minimize J over $\{r_{nk}\}$ -- Assign every point to the closest cluster center

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Step 2: Update the cluster centers– the porotype for a cluster is the data that is closest to all other data points in the cluster

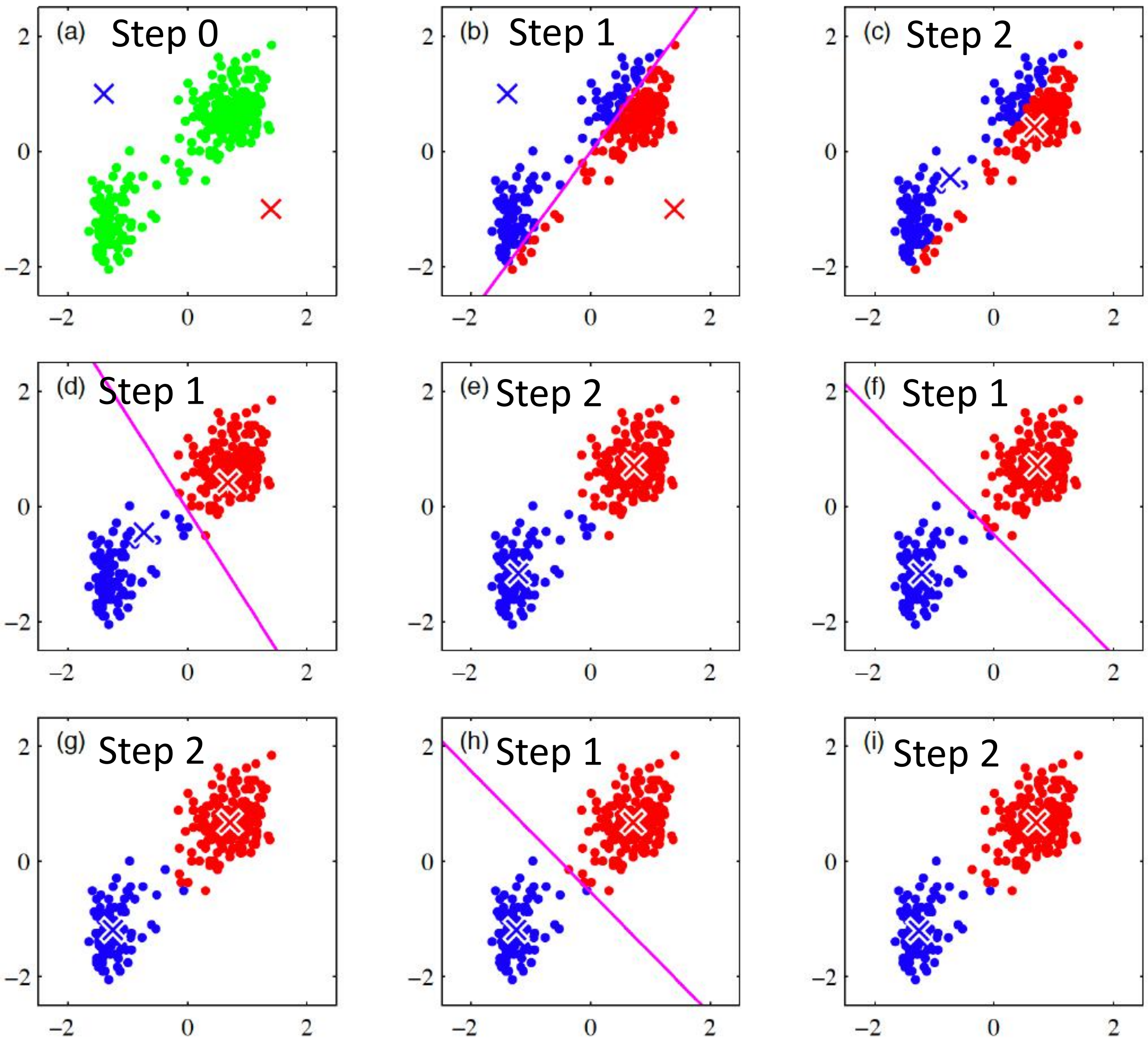
$$k* = \arg\min_{m: r_{mk}=1} \sum_n r_{nk} \|\boldsymbol{x}_n - \boldsymbol{x}_m\|_2^2$$

$$\boldsymbol{\mu}_k = \boldsymbol{x}_{k*}$$

- Loop until it converges

Gaussian Mixture Models

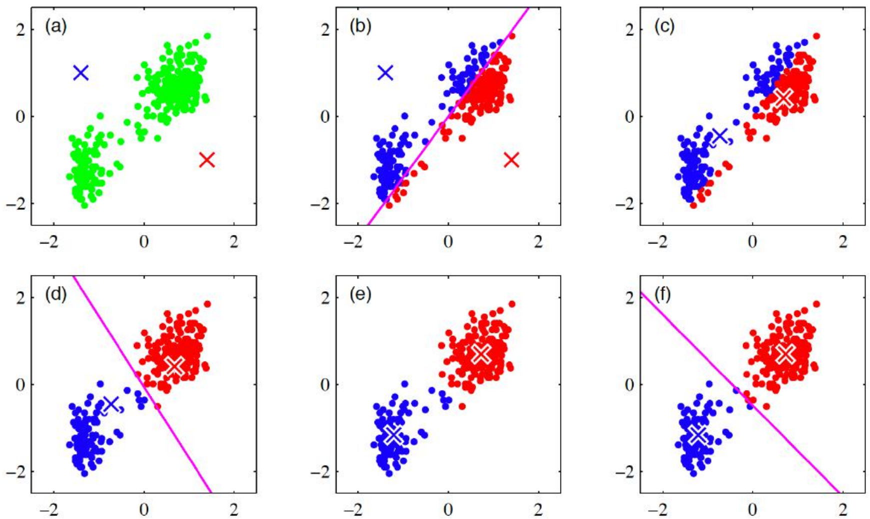
K-Means



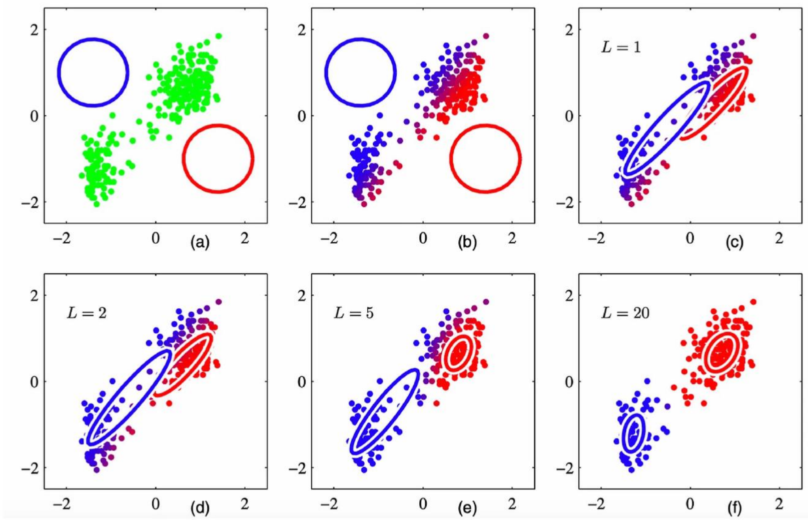
(a) Step 0
(b) Step 1
(c) Step 2
(d) Step 1
(e) Step 2
(f) Step 1
(g) Step 2
(h) Step 1
(i) Step 2

Similar to K-means

- Alternatively:
  1) assign points to clusters
  2) update cluster centers/variances…

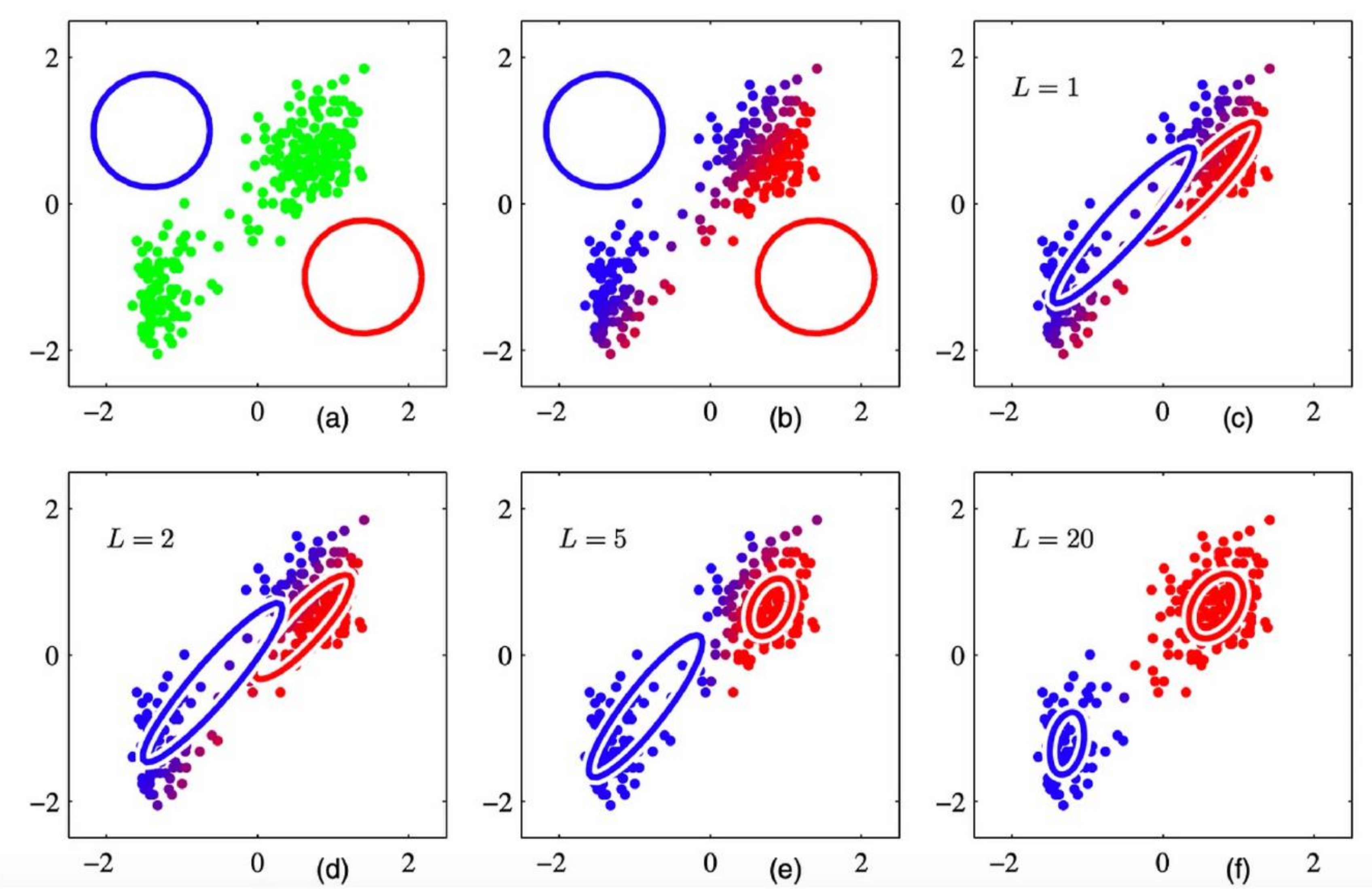


K-Means



GMM

GMM

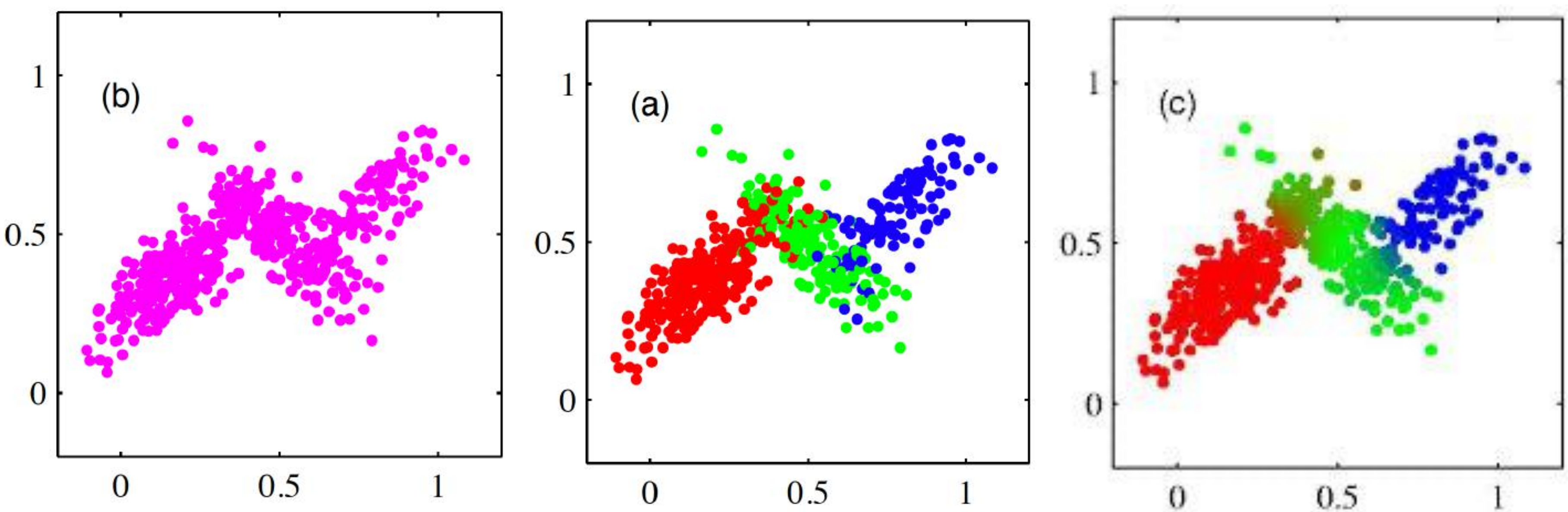


Soft cluster

- Assignment based on the conditional probability $P(A(x_n)|x_n)$

Hard cluster assignment

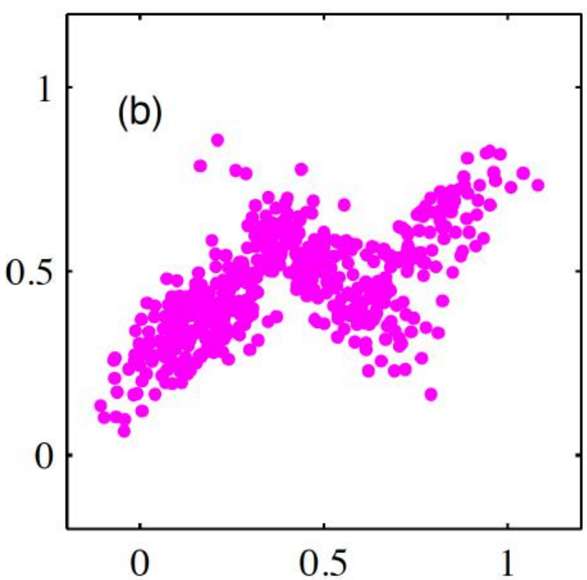
Soft cluster assignment



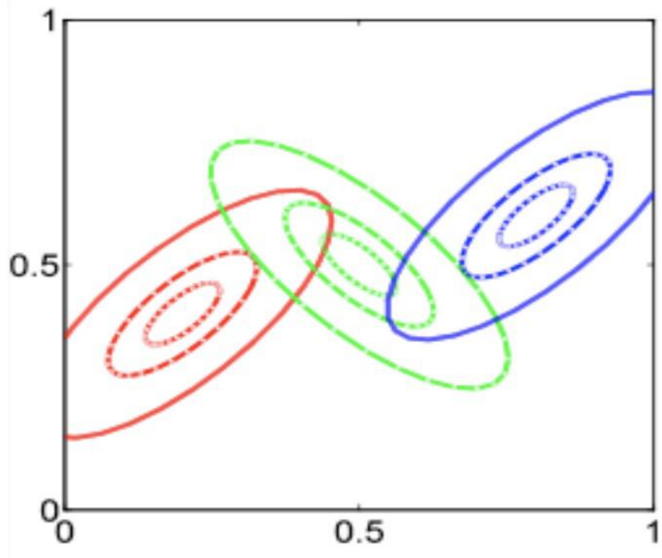
Gaussian mixture models

- Assume the probability density function for x as

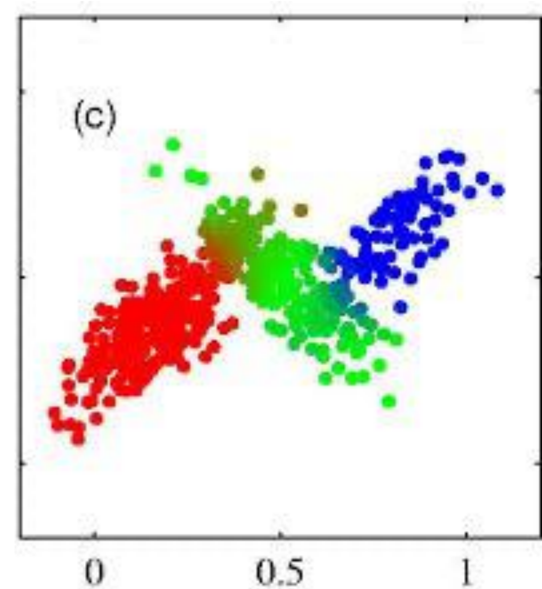
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



Model by 3 Guassians



Soft cluster assignment

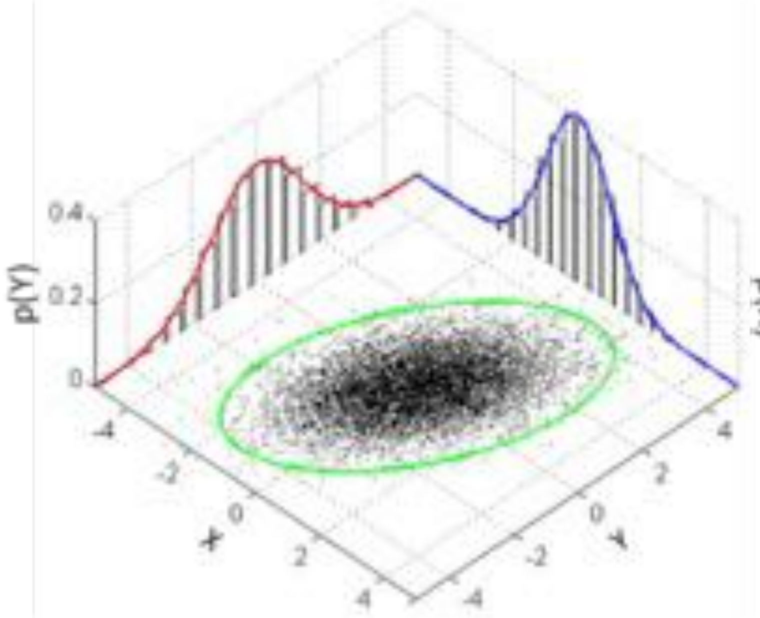


Multivariate Gaussian

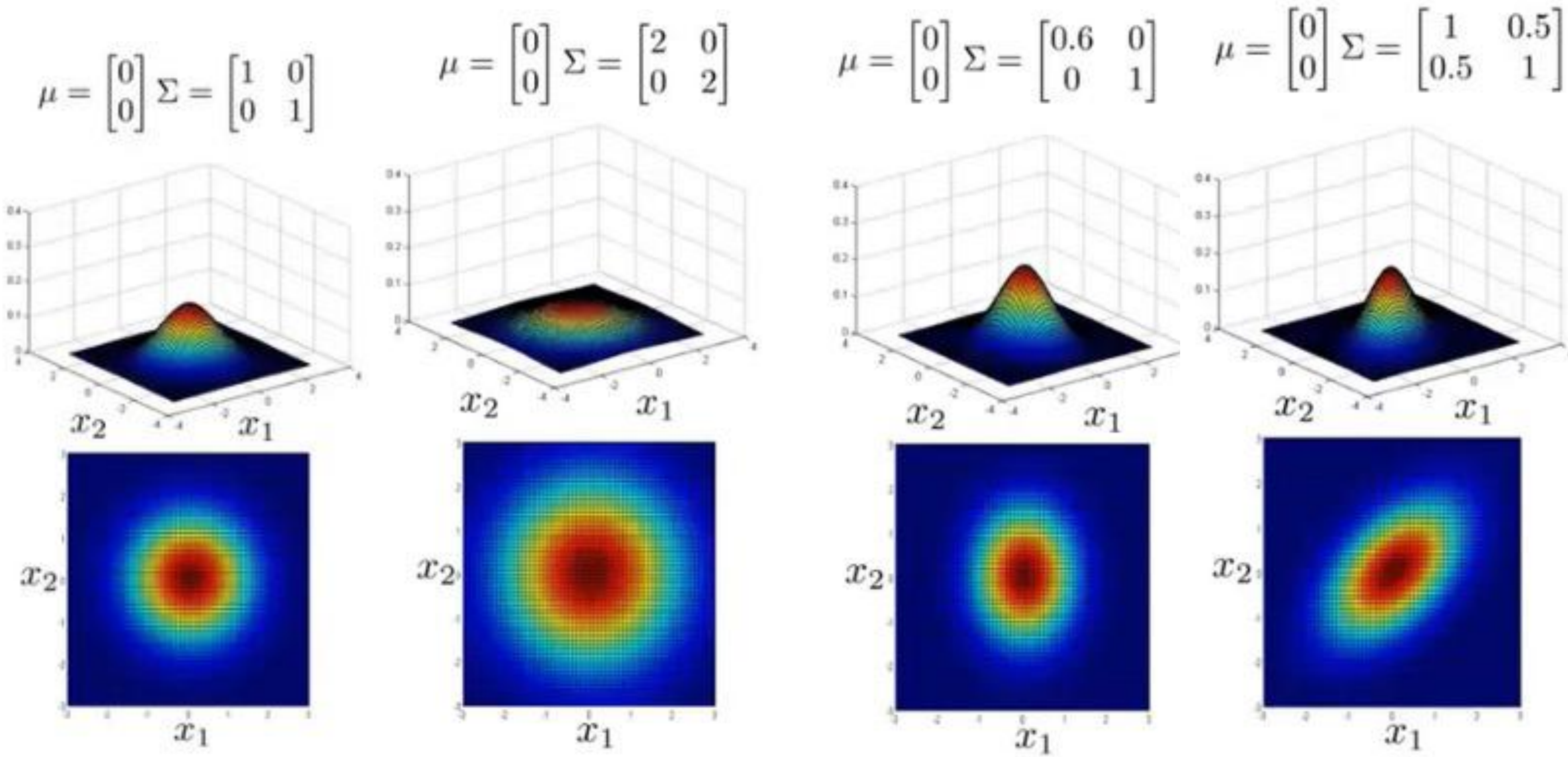
- Mean $\boldsymbol{\mu} \in \mathbf{R}^k$

- Variance $\boldsymbol{\Sigma} \in \mathbf{R}^{k \times k}$

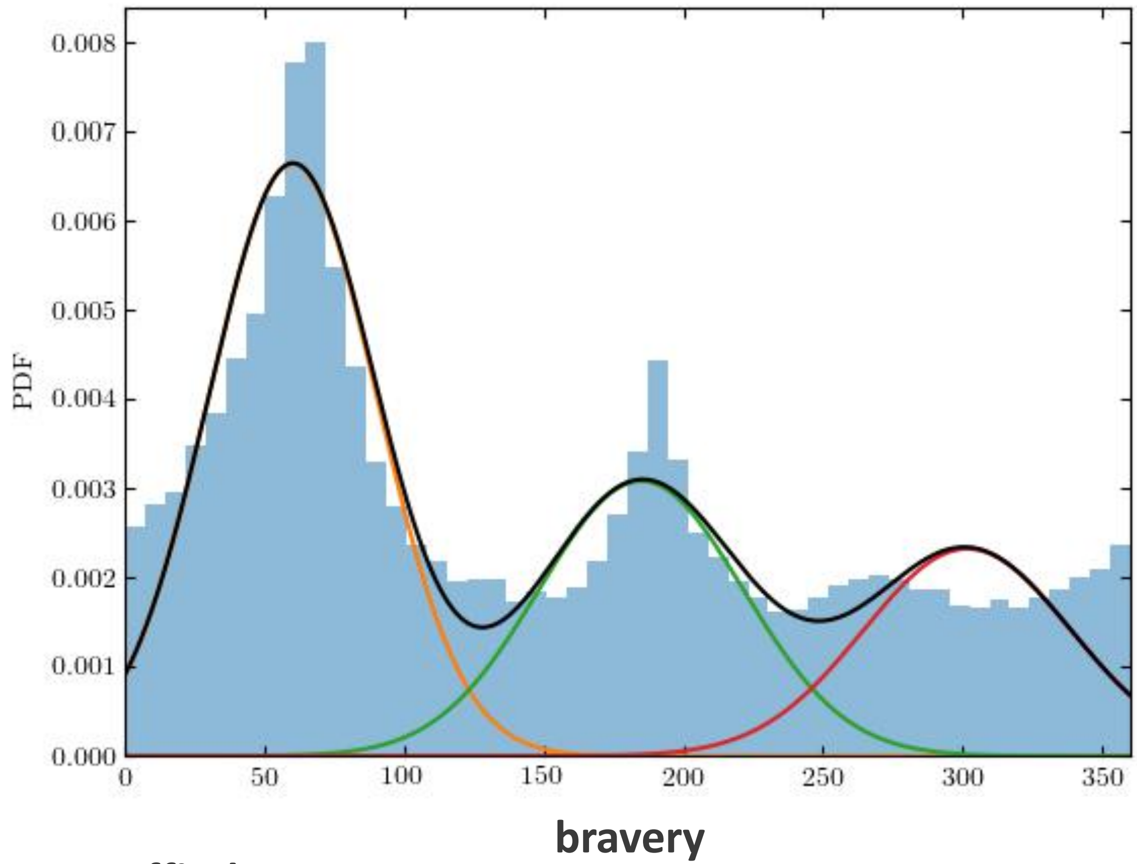
- PDF:



$$(2\pi)^{-\frac{k}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))$$


$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$
x_2
x_1

Model Four Houses of Hogwarts

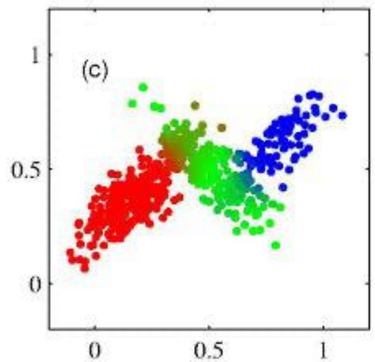


Distribution for students in **Gryffindor**

Distribution for students in **Slytherin**

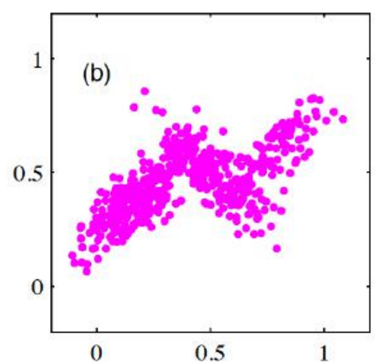
Distribution for students in **Ravenclaw**

Why Mixed Gaussian?



The conditional distribution between $\boldsymbol{x}$ and z (representing color) are

$$p(\boldsymbol{x}|z = red) = N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$p(\boldsymbol{x}|z = blue) = N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$
$$p(\boldsymbol{x}|z = green) = N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

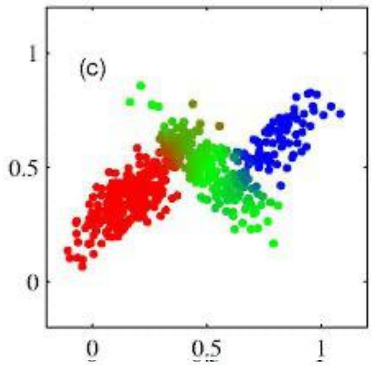


The marginal distribution is thus

$$p(\boldsymbol{x}) = p(red)N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p(blue)N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + p(green)N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

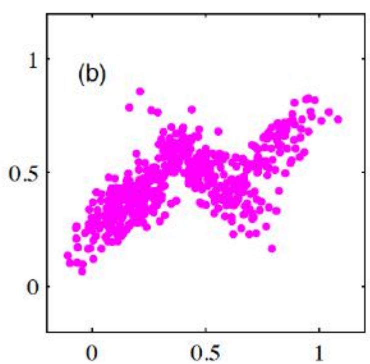
ω_k

Bayes Rule – Posterior distrbution



The conditional distribution between $\boldsymbol{x}$ and z (representing color) are

$$p(\boldsymbol{x}|z = red) = N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$p(\boldsymbol{x}|z = blue) = N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$
$$p(\boldsymbol{x}|z = green) = N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$



The marginal distribution is thus

$$p(\boldsymbol{x}) = p(red)N(\boldsymbol{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p(blue)N(\boldsymbol{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + p(green)N(\boldsymbol{x}|\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

$$\boxed{p(z_n = k|\boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{\sum_{k'=1}^{K} p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}}$$

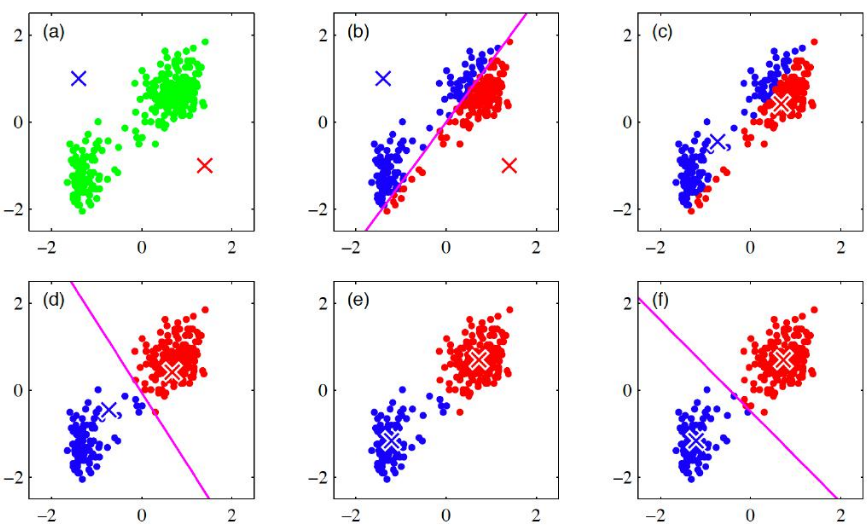
Similar to K-means

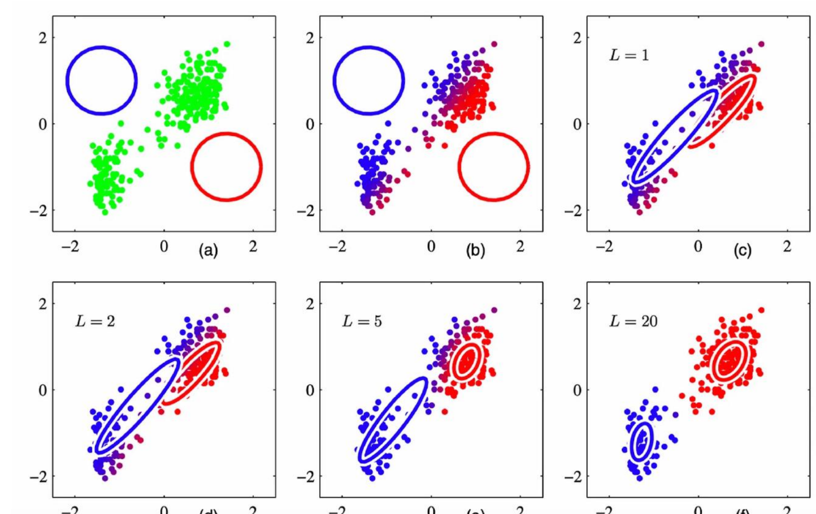
❖ Alternatively:

1) soft assign points to clusters based on

$$p(z_n = k|\boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{\sum_{k'=1}^{K} p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}$$

2) update cluster centers/variances…





Similar to K-means

- Alternatively:
  1) assign points to clusters
  2) update cluster centers/variances based on the weighted mean/variance