Announcements

- PSet-2 announced today
- No laptops during the class.

Last time

- Dimensionality reduction
  - PCA
  - PCoA
  - CCA
- Bag of Words (language)
- Bag of Visual Words

Bag of words

Word	Appearance count	Index
the	2	0
brown	1	1
fox	1	2
jumps	1	3
over	1	4
lazy	1	5
dog	1	6
oov	0	7

the	br	fox	jum	ove	laz	dog	oov	whit	cat
2	1	1	1	1	1	1	0	0	0

Image credit:mlarchive.com

Bag of visual words

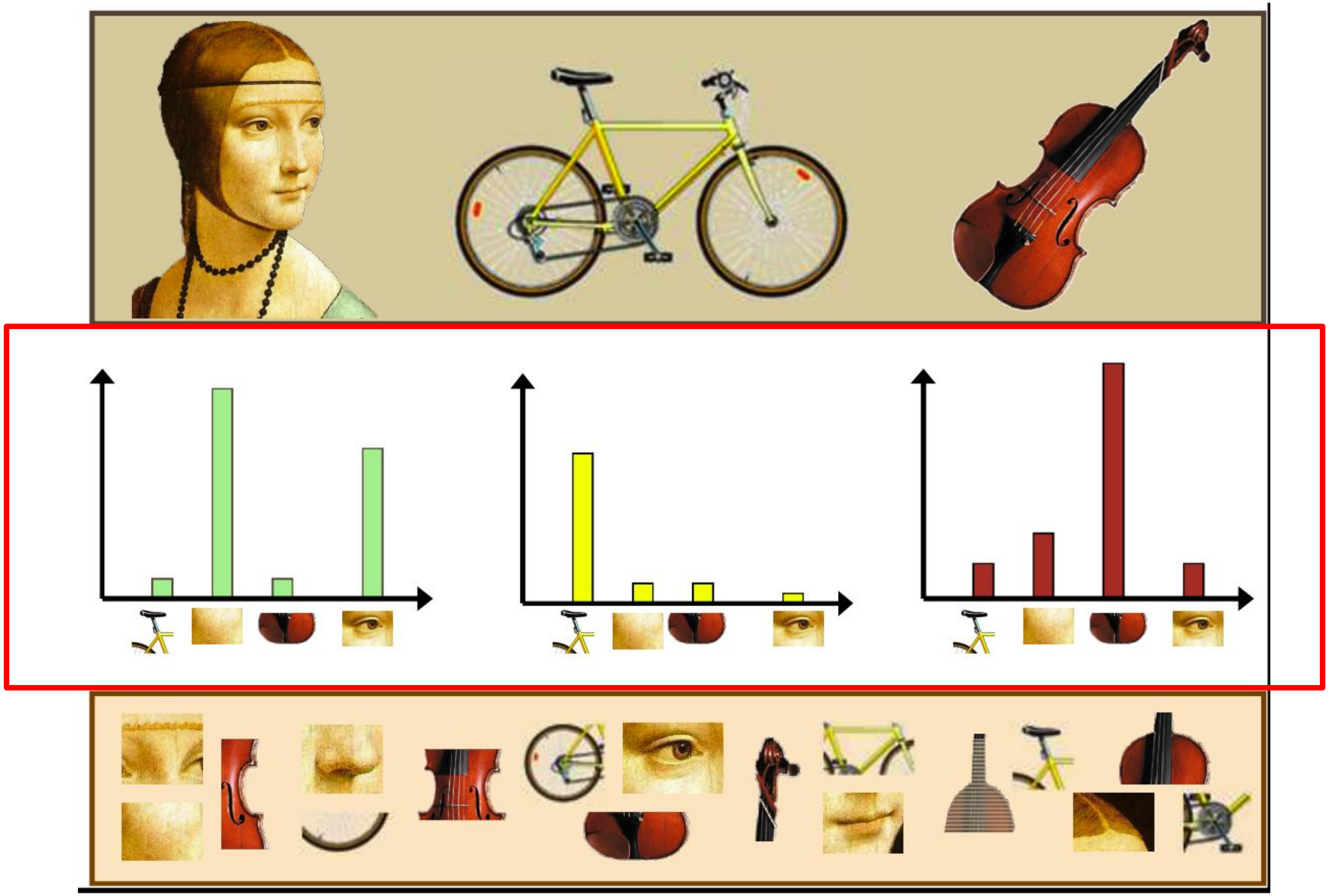


Image credit:Kristen Grauman

Learning between multiple modalities

Teddy bears shopping for groceries in ancient Egypt

Input

Generative Model

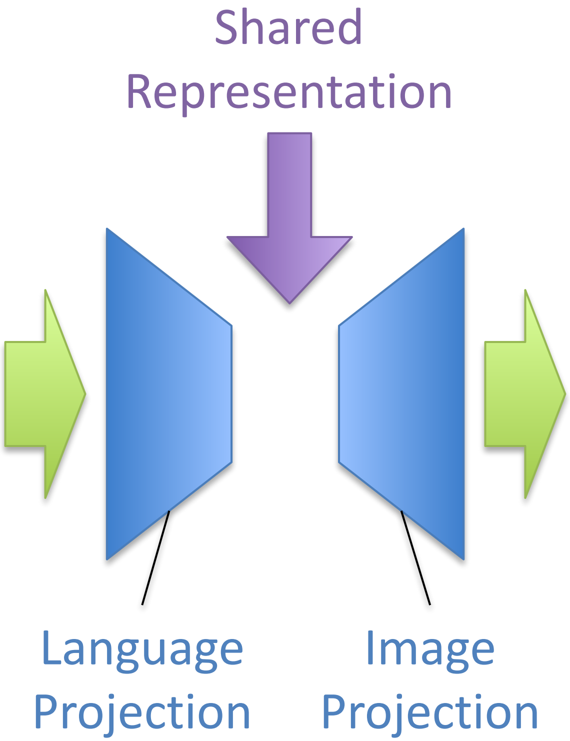
DALL·E 2

Output

Learning between multiple modalities

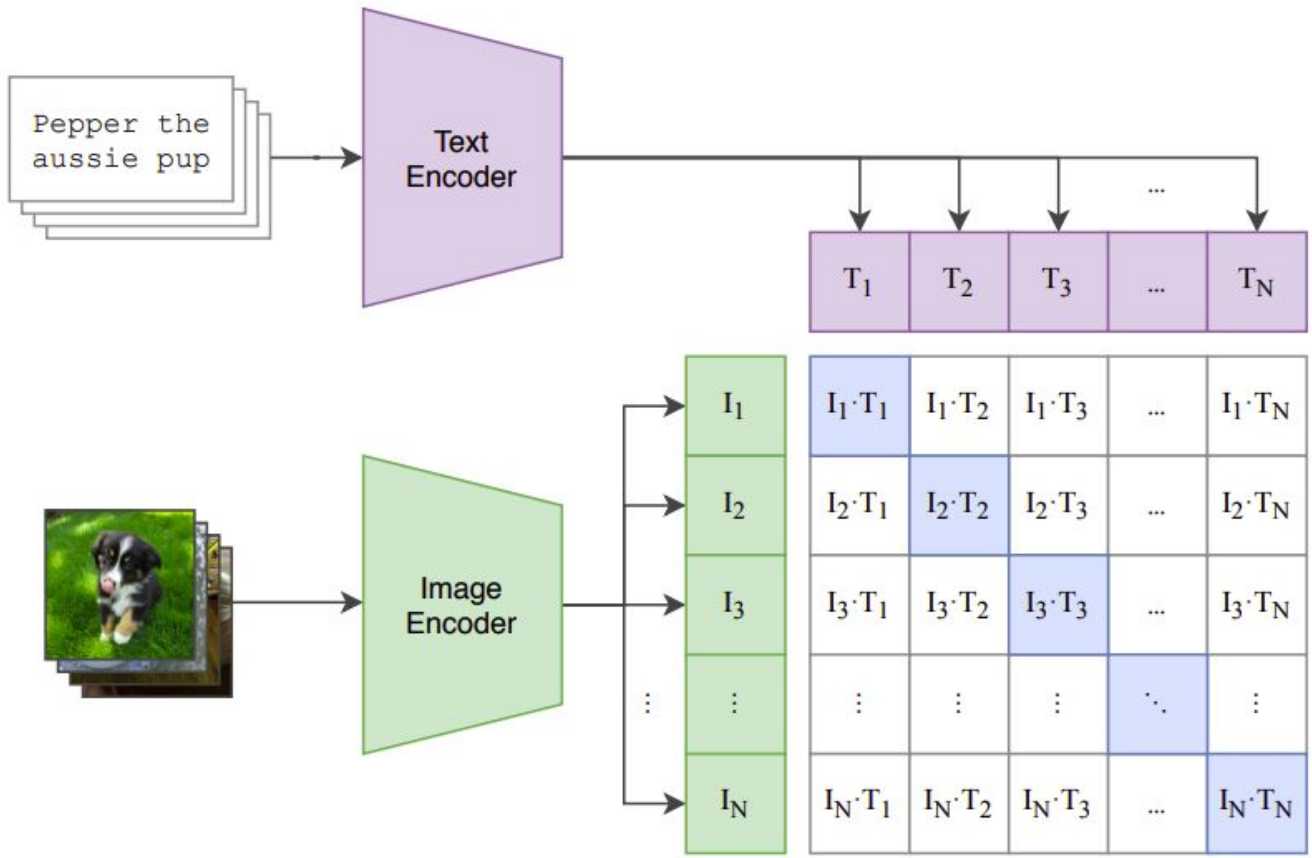
Teddy bears shopping for groceries in ancient Egypt

Input



Output

CLIP (Contrastive Language Image Pre-training)



Today: Clustering

- Agglomerative Clustering
- Divisive Clustering
- K-means
- Vector Quantization with K-Means
- Mixtures of Gaussians
- Expectation Maximization

Recall: Types of learning

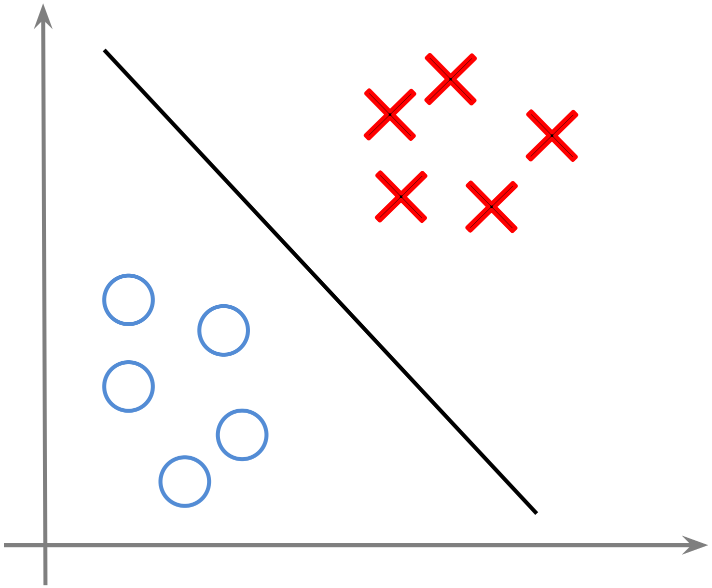
Supervised

Unsupervised

Reinforcement

Supervised Learning

Supervised



Training set: $\{(x_1, y_1), (x_2, y_2) \ldots, (x_N, y_N)\}$

Decision Trees, SVMs, etc.

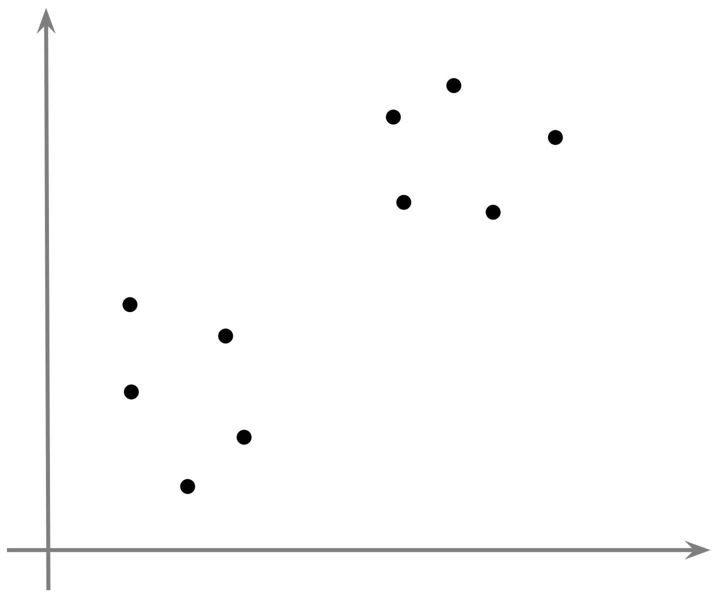
Recall: Types of learning

Supervised

Unsupervised

Reinforcement

Unsupervised Learning

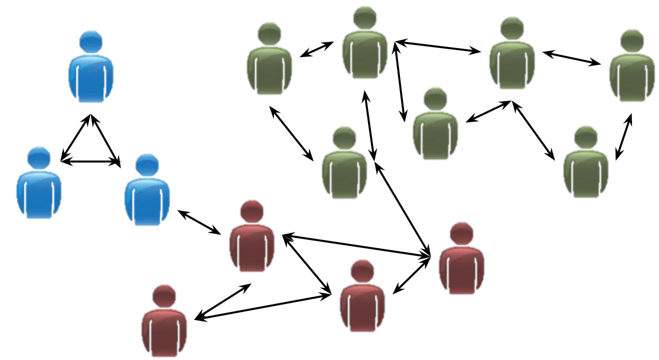


Training set: $\{(x_1, \cancel{y_1}), (x_2, \cancel{y_2}) \ldots, (x_N, \cancel{y_N})\}$

Training set: $\{x_1, x_2, \ldots, x_N\}$

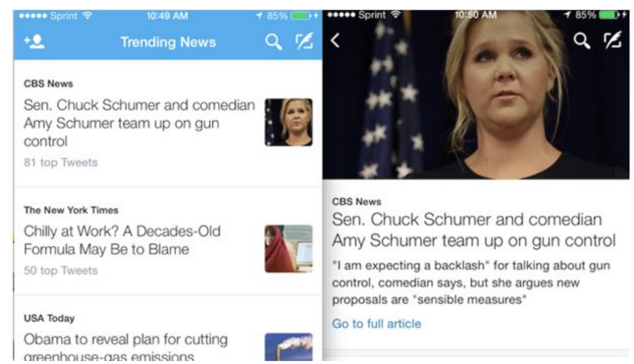
Clustering

Gene analysis



Social network analysis

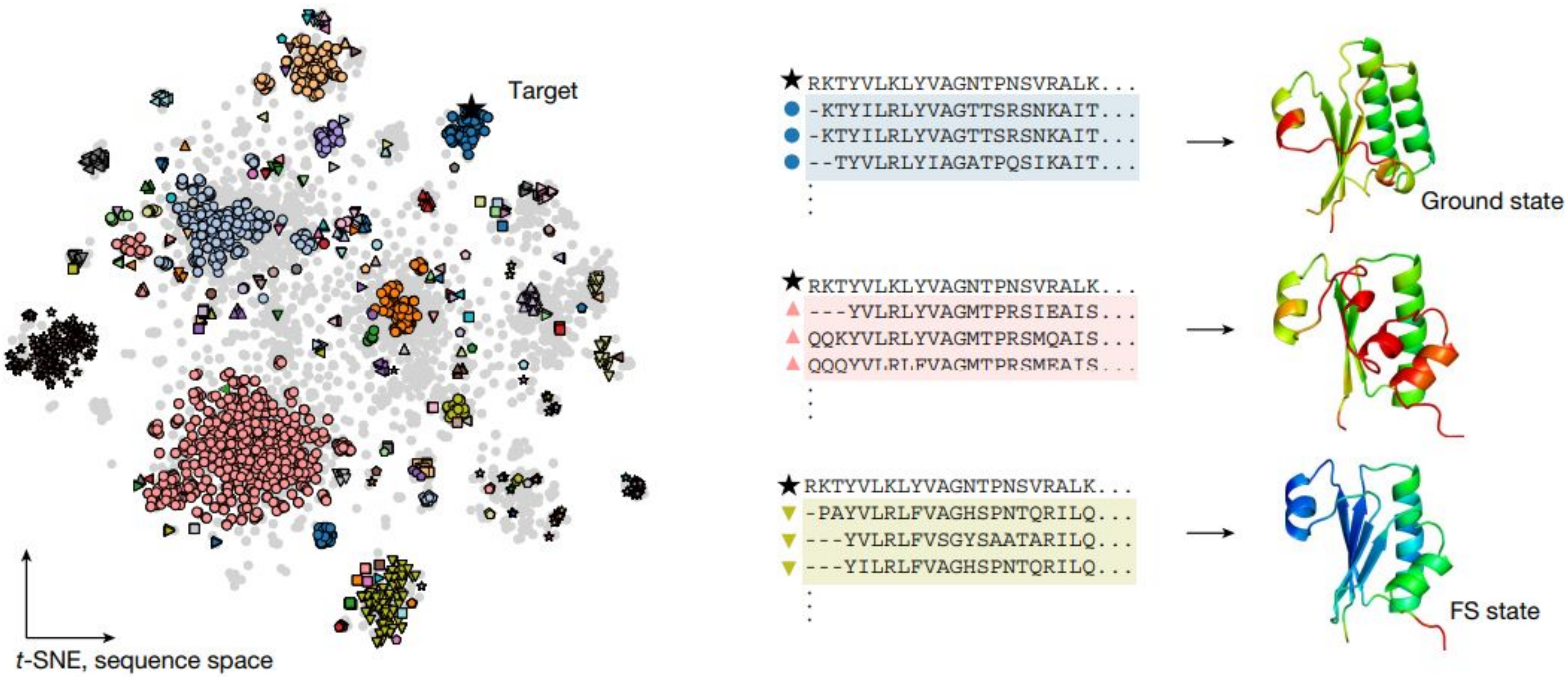
Types of voters



Trending news

Example Clusters

Protein folding: AlphaFold2



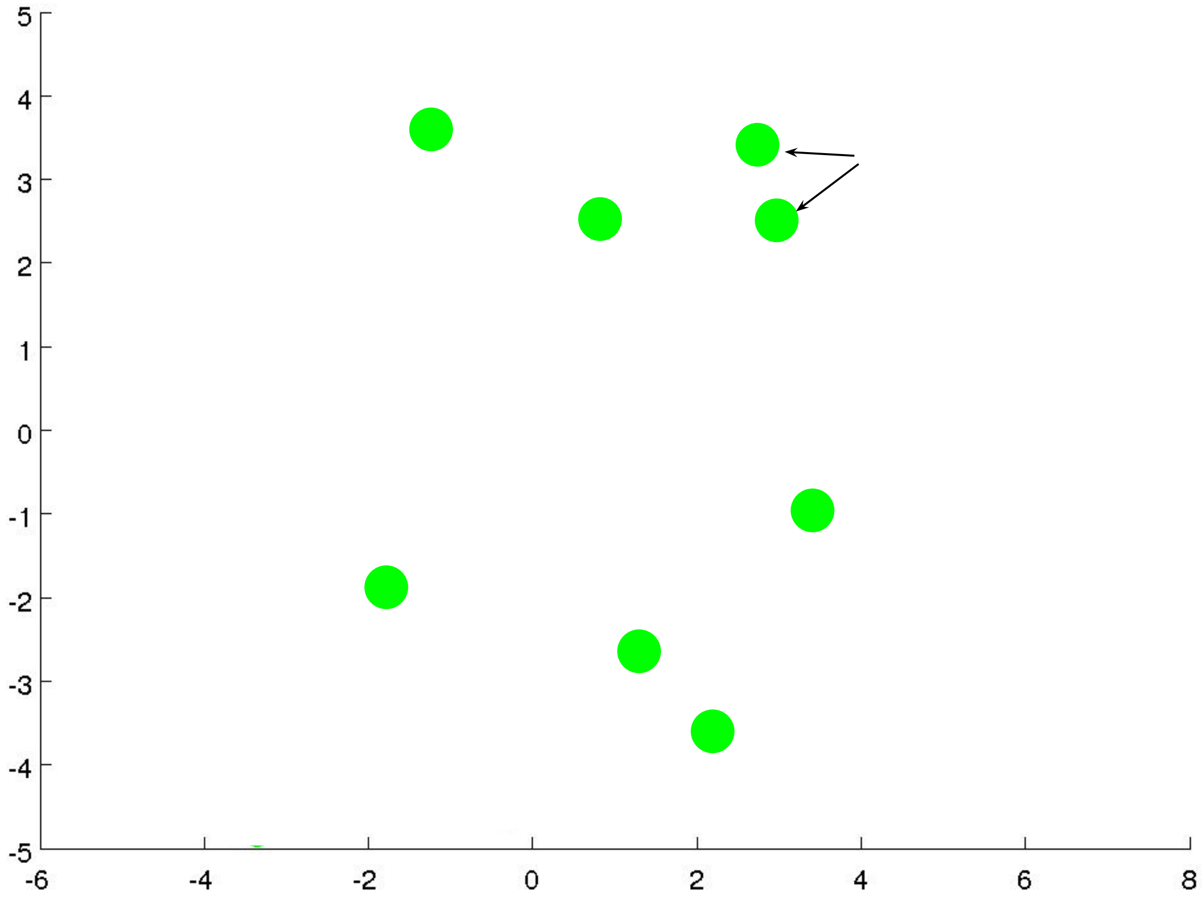
Today: Clustering

- Agglomerative Clustering
- Divisive Clustering
- K-means
- Vector Quantization with K-Means
- Mixtures of Gaussians
- Expectation Maximization

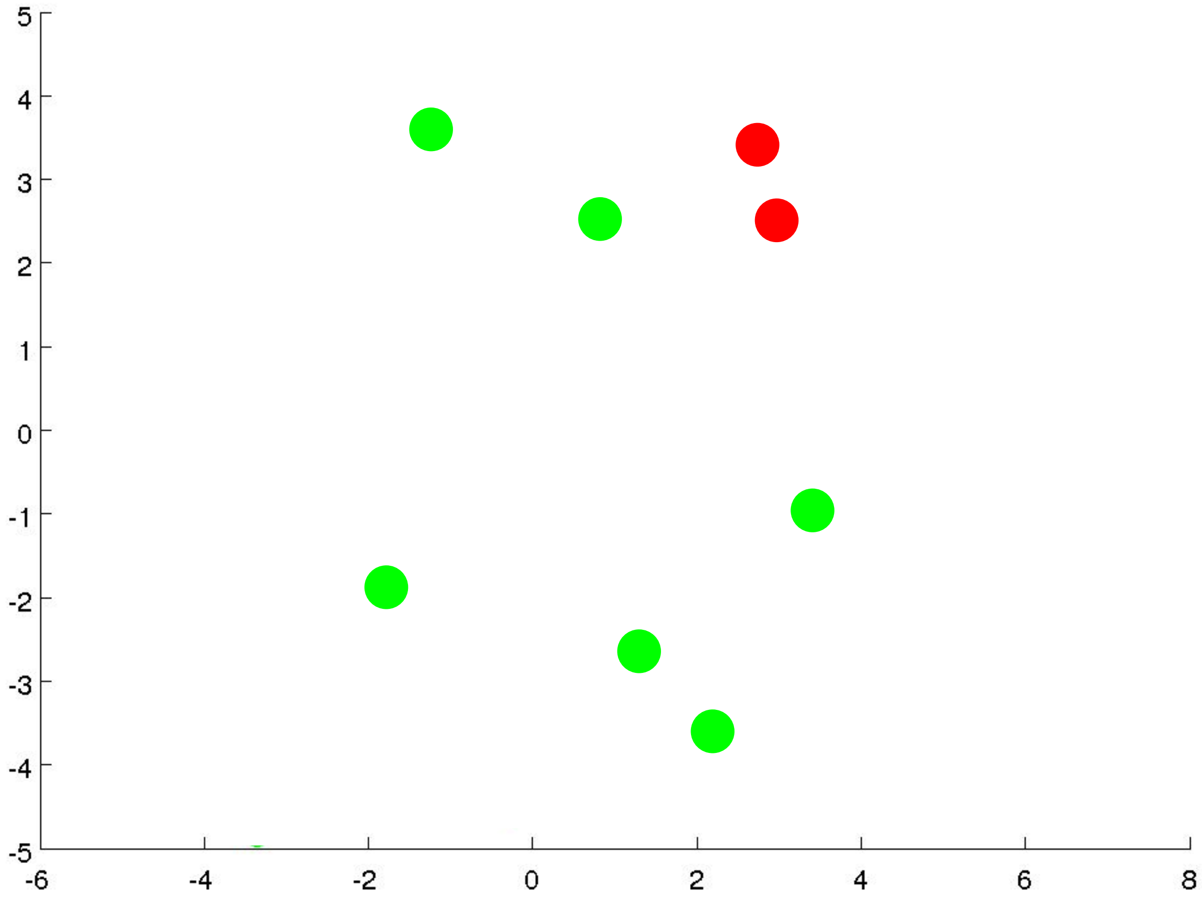
Today: Clustering

- **Agglomerative Clustering**
- Divisive Clustering
- K-means
- Vector Quantization with K-Means
- Mixtures of Gaussians
- Expectation Maximization

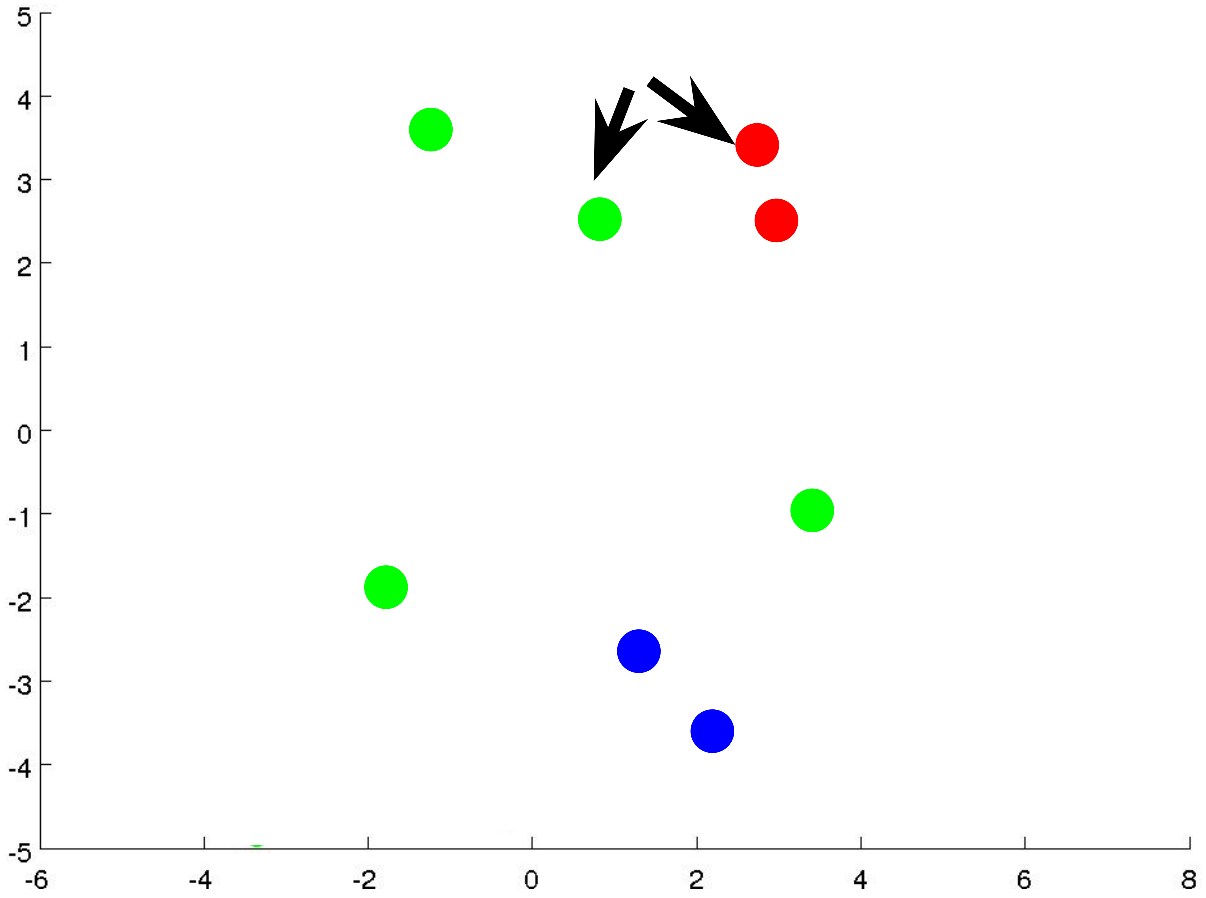
Single-link clustering: *Iteratively combine the two closest points*



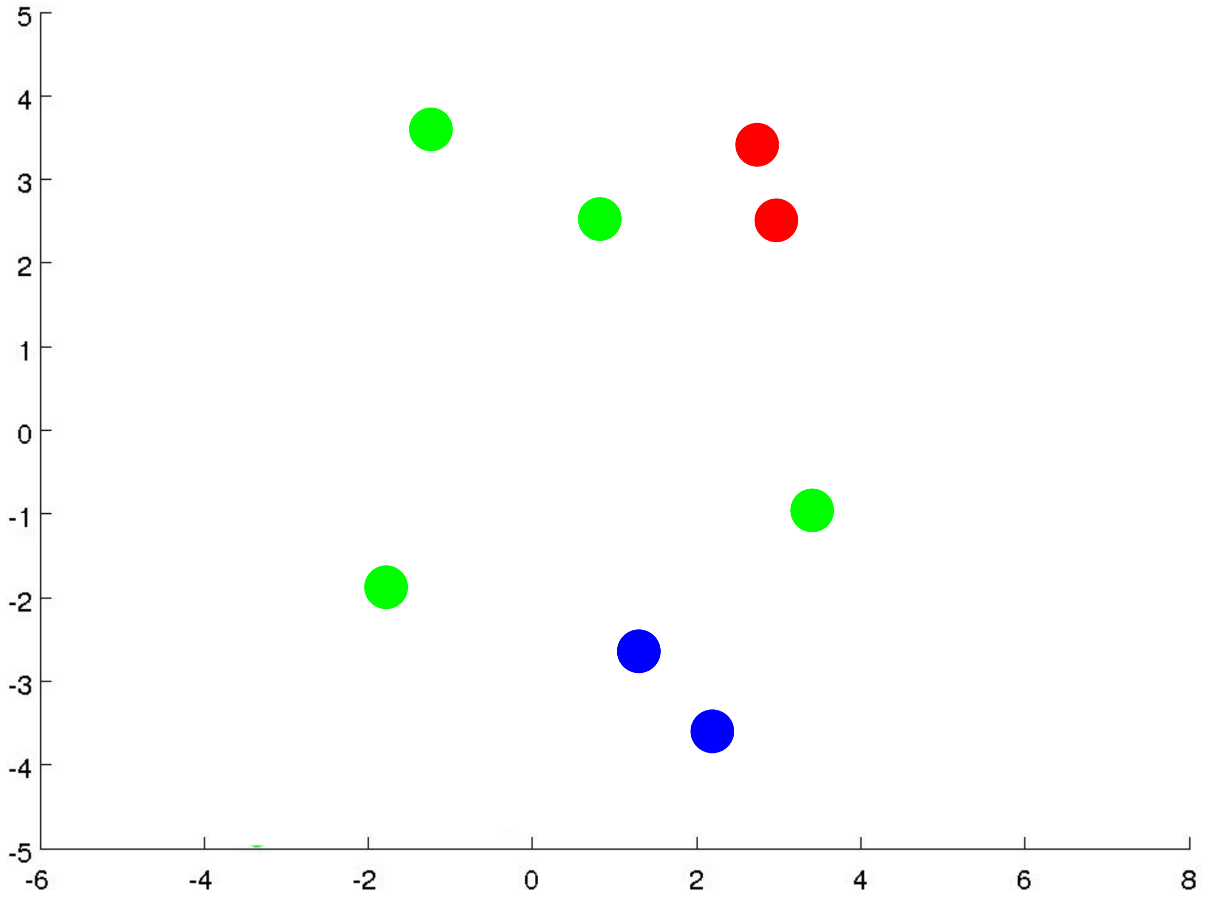
Single-link clustering: *Iteratively combine the two closest points*



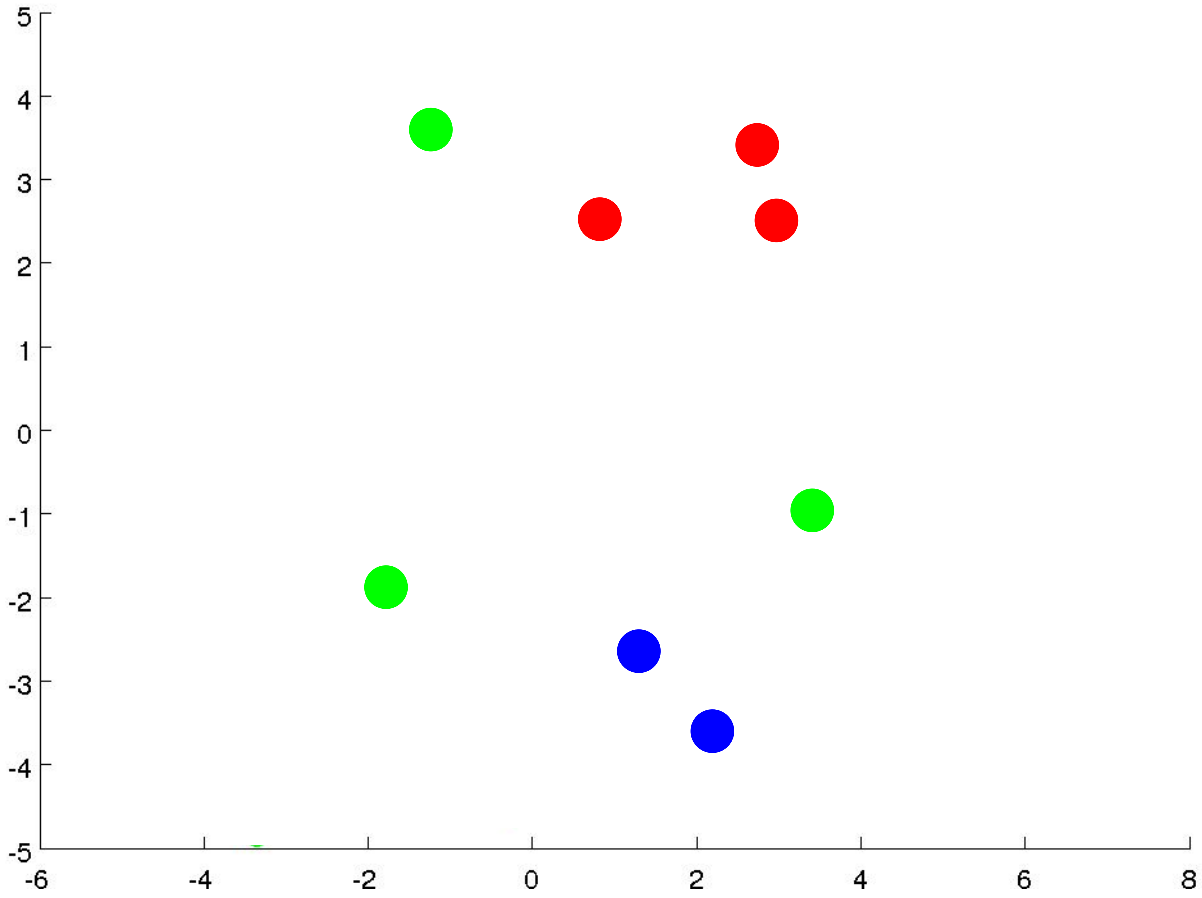
Single-link clustering: *Iteratively combine the two closest points*



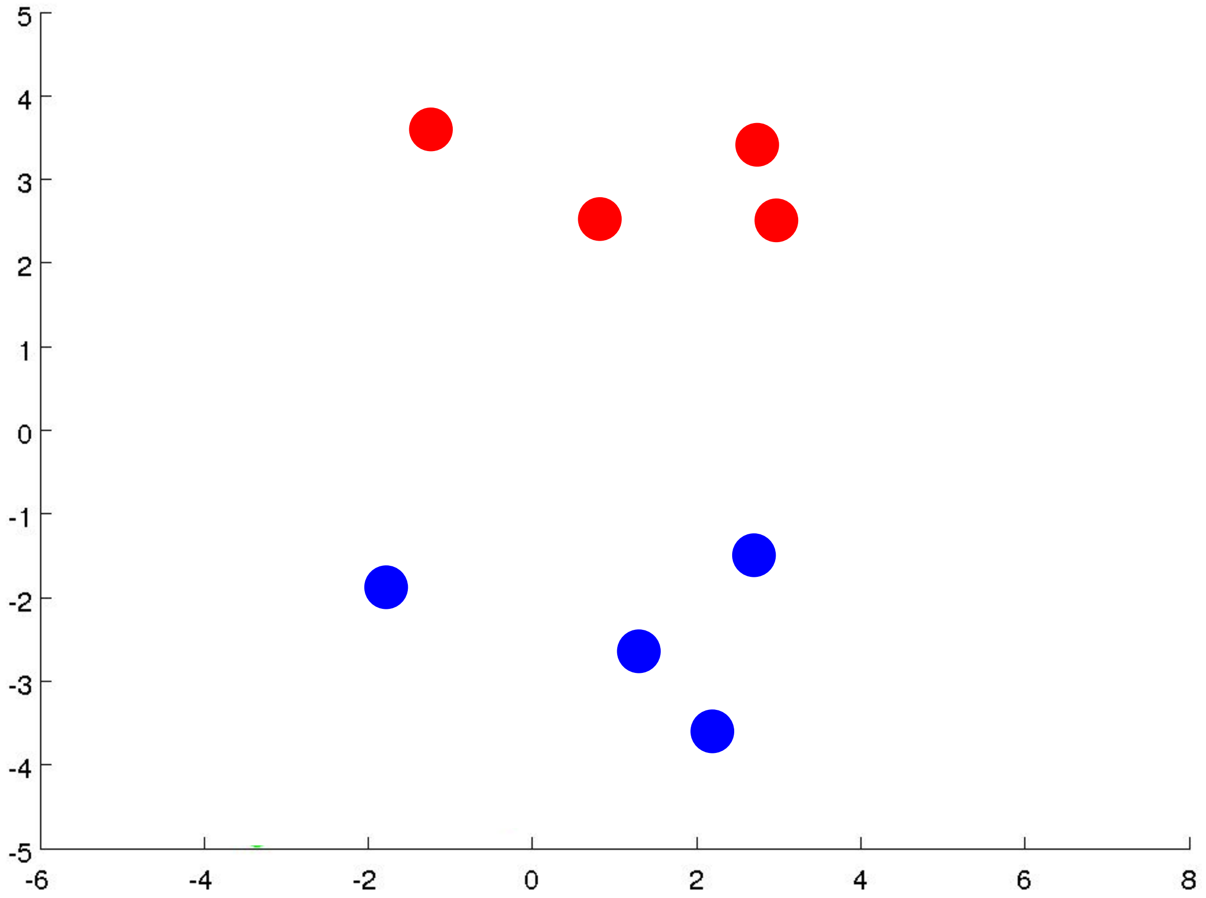
Single-link clustering: *Iteratively combine the two closest points*



Single-link clustering: *Iteratively combine the two closest points*

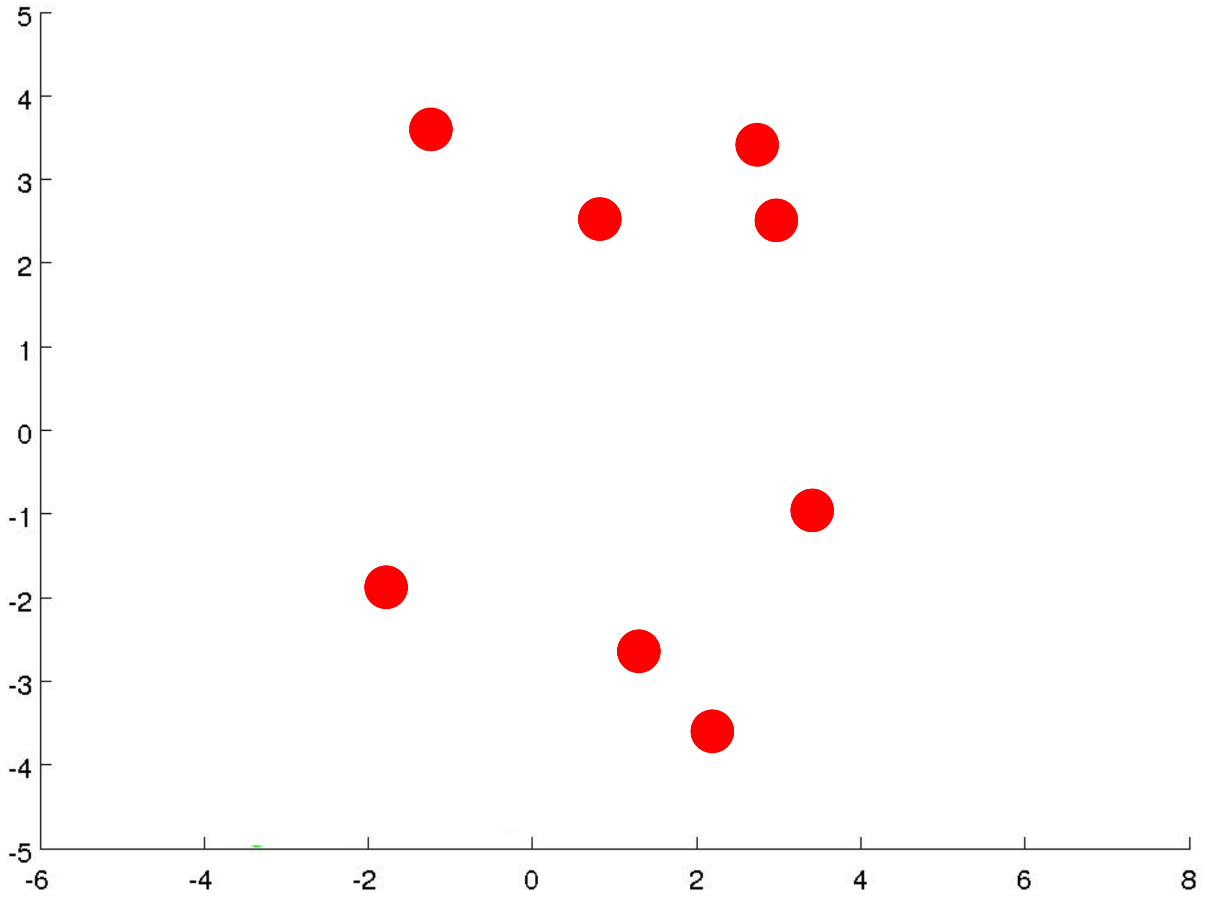


Single-link clustering: *Iteratively combine the two closest points*



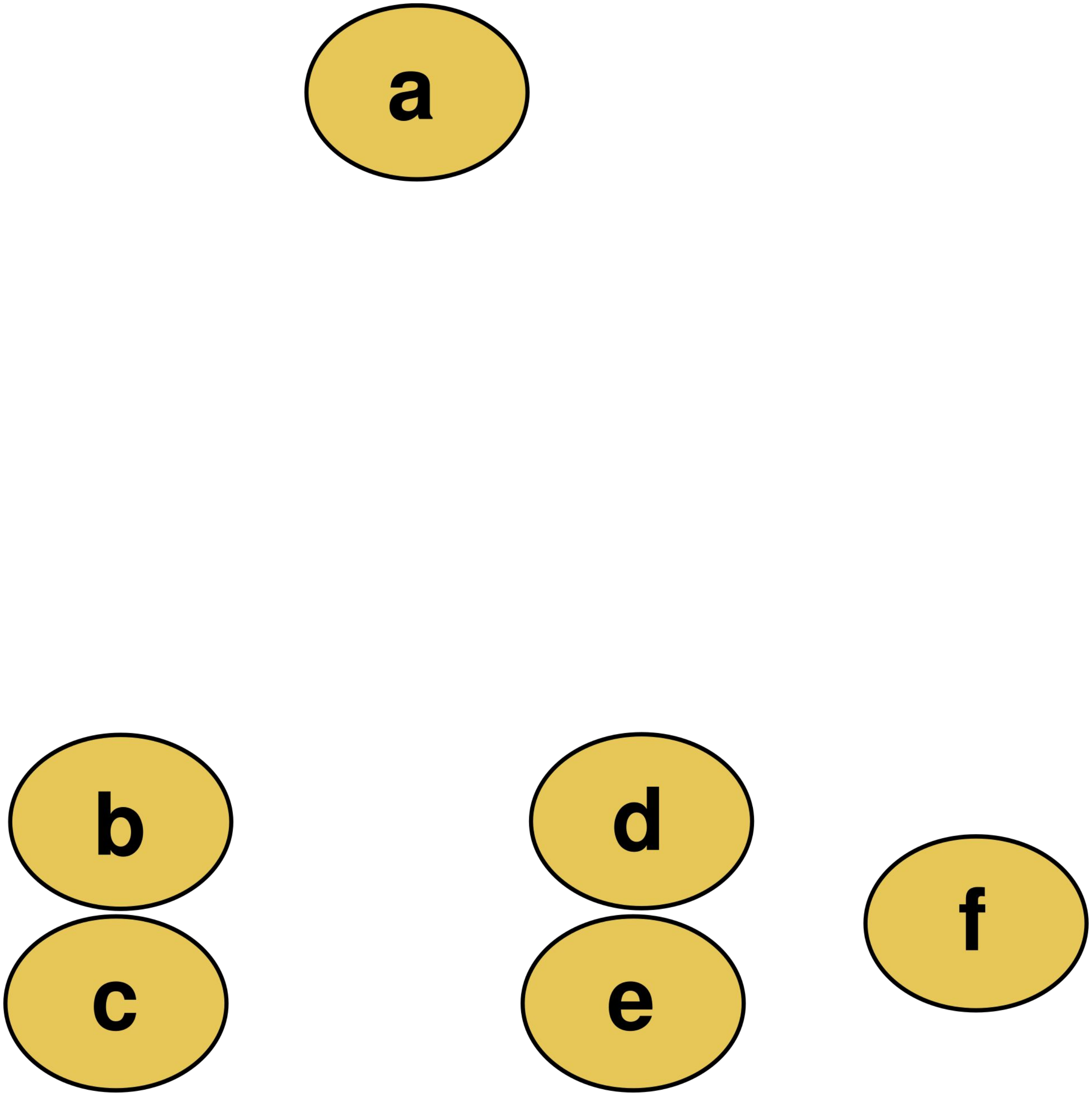
What are some good ways to compute distances between green points and the red points (select all that apply)

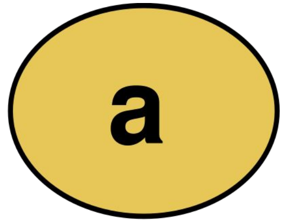
Single-link clustering: *Iteratively combine the two closest points*



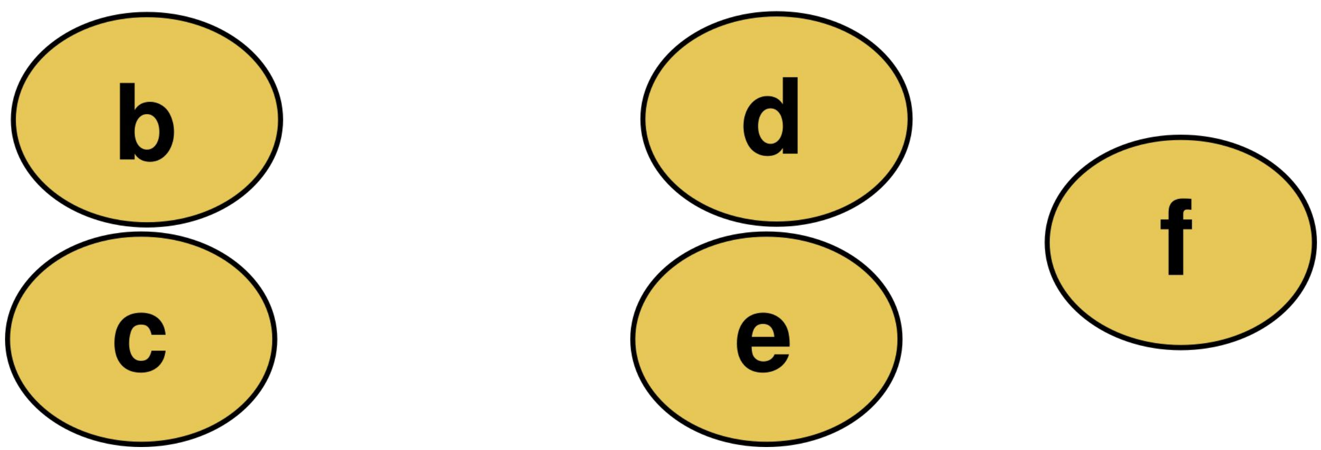
When to stop combining?

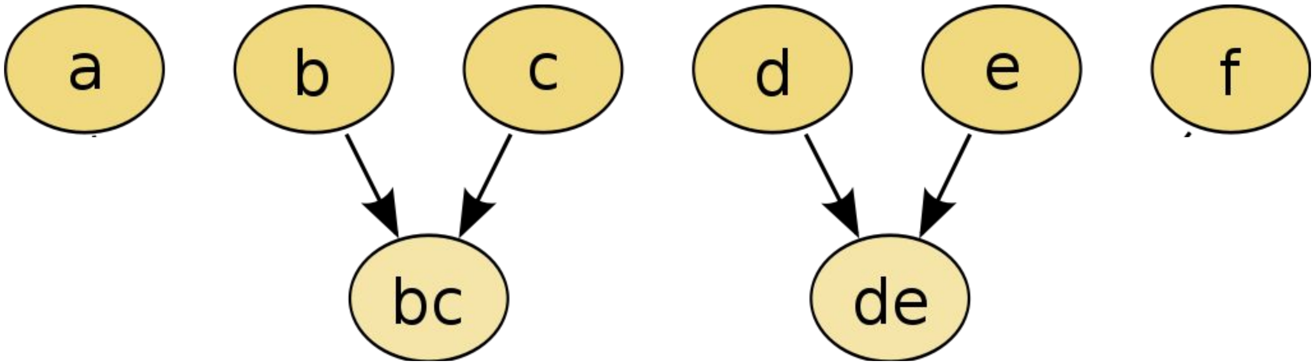
How to decide when to stop clustering? (select all that apply)

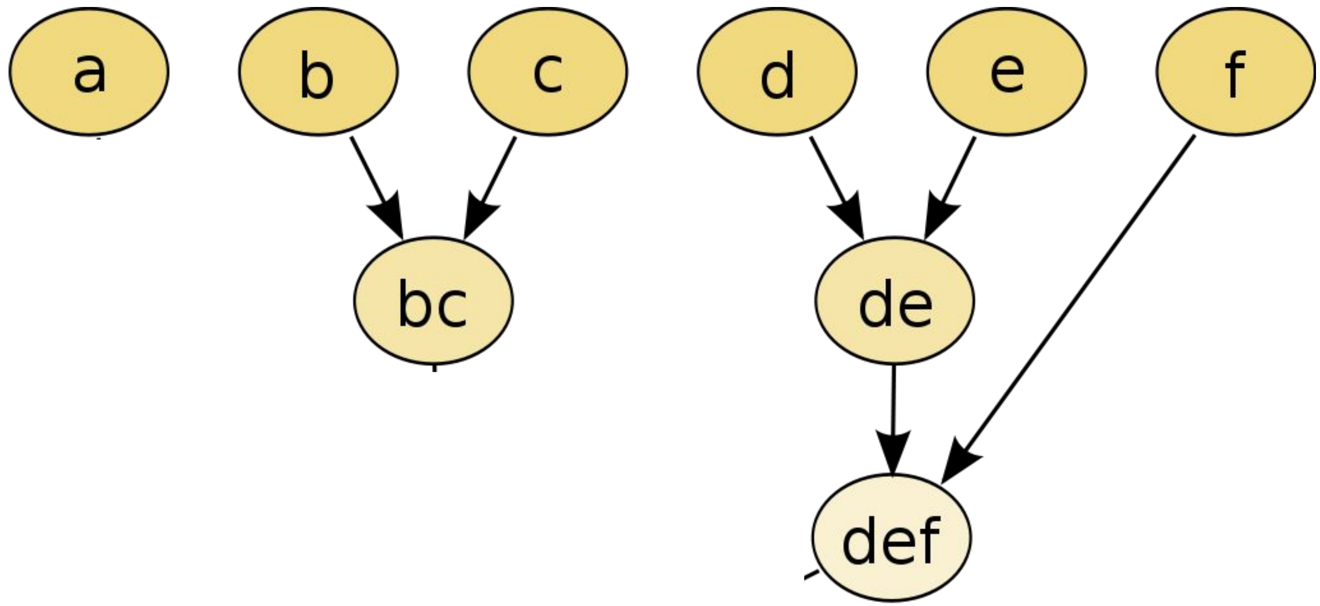

a
b
c
d
e
f

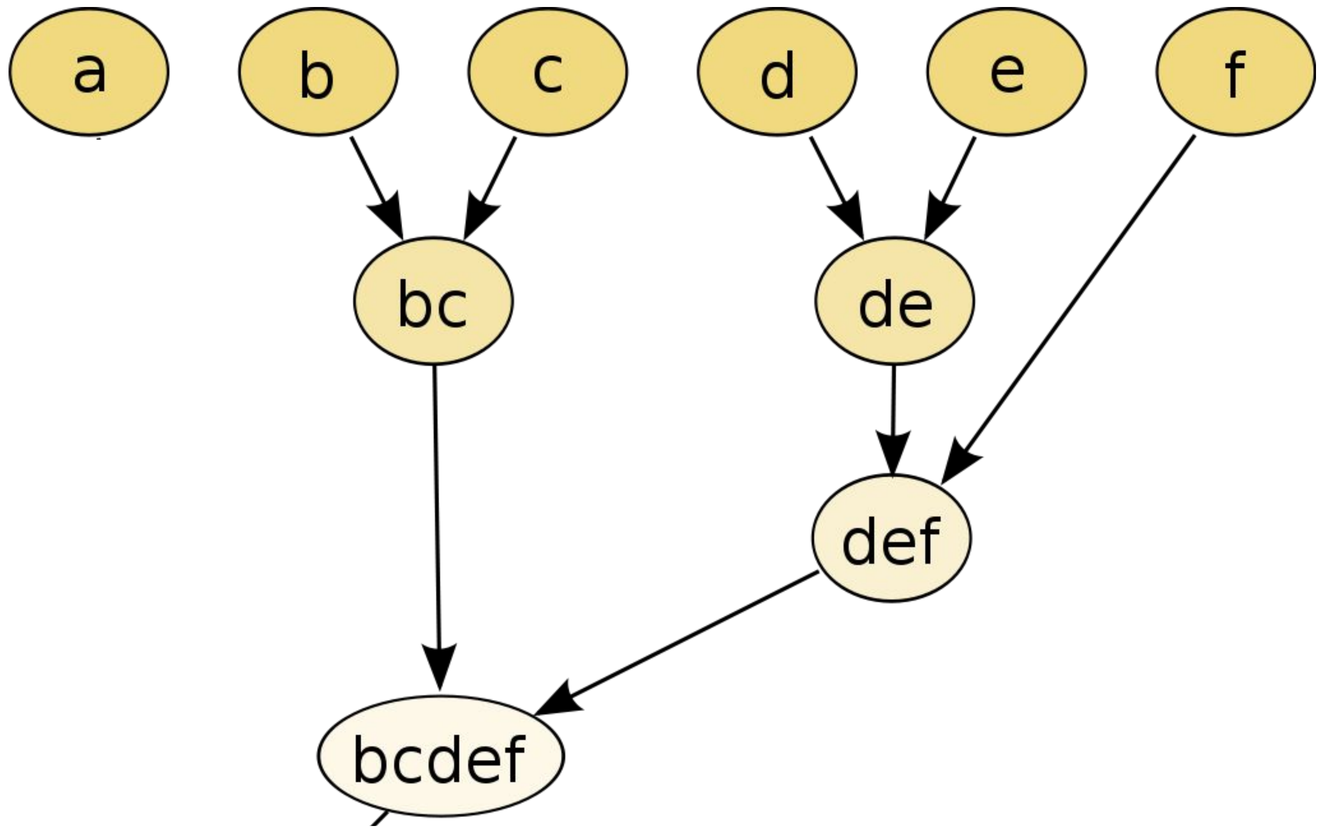


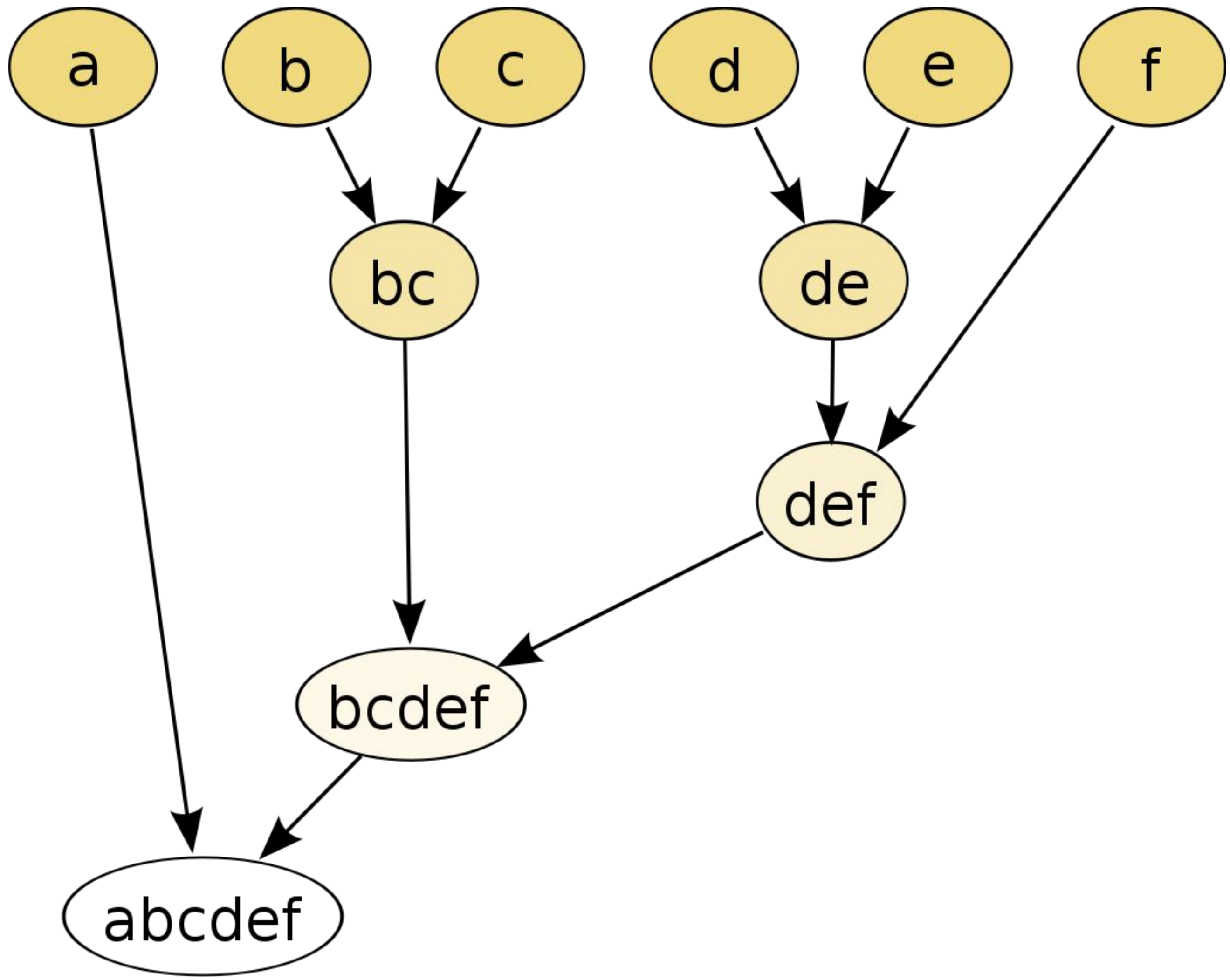
Compute pairwise distances between nodes




a
b
c
d
e
f
bc
de


a
b
c
d
e
f
bc
de
def


a
b
c
d
e
f
bc
de
def
bcdef


a
b
c
d
e
f
bc
de
def
bcdef
abcdef

What is the average space complexity of agglomerative clustering for N datapoints? Space complexity quantifies the amount of memory taken by an algorithm to run

What is the space complexity of agglomerative clustering for N datapoints? Space complexity quantifies the amount of memory taken by an algorithm to run

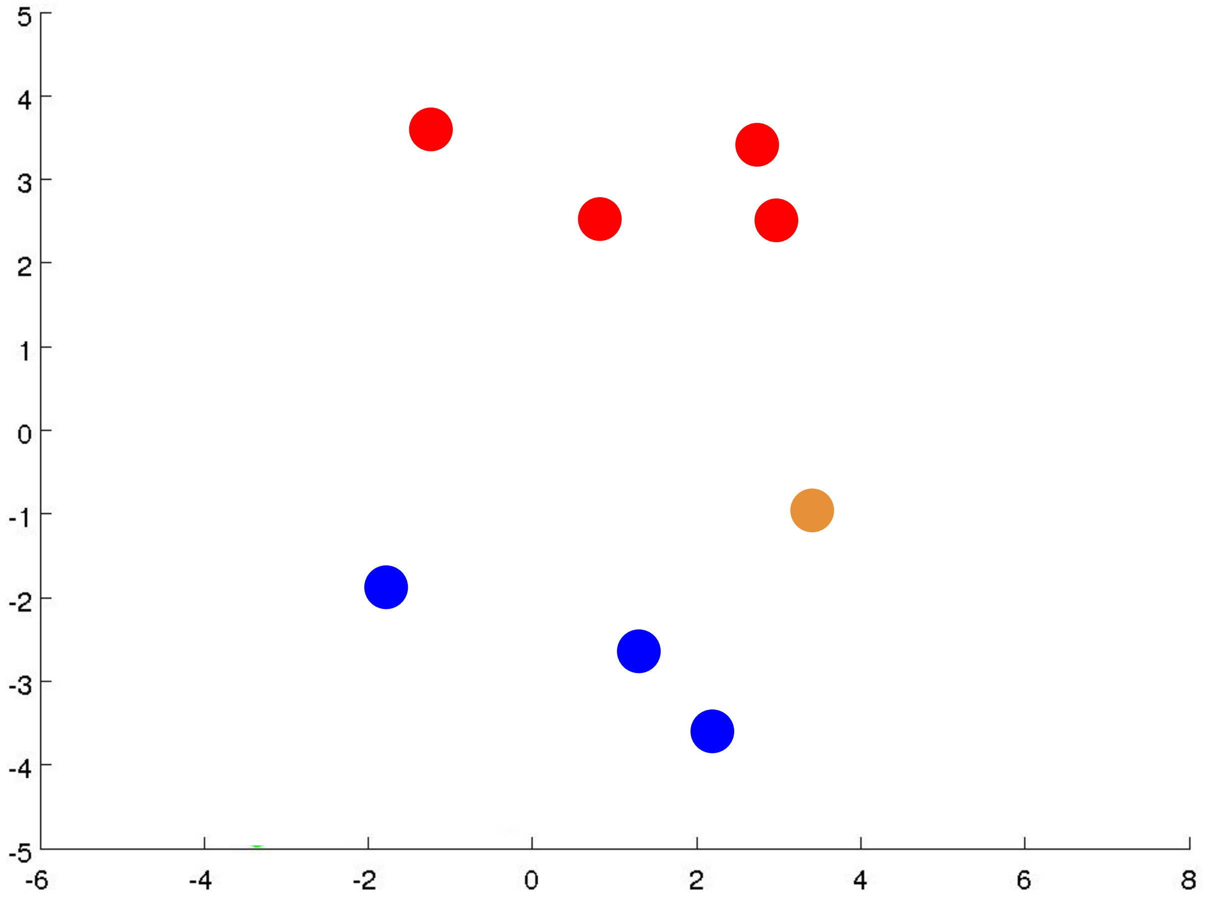
O(N)
18%

O(N^2)
57%

O(N^3)
6%

O(2^N)
19%

Single-link clustering: *Iteratively combine the two closest points*



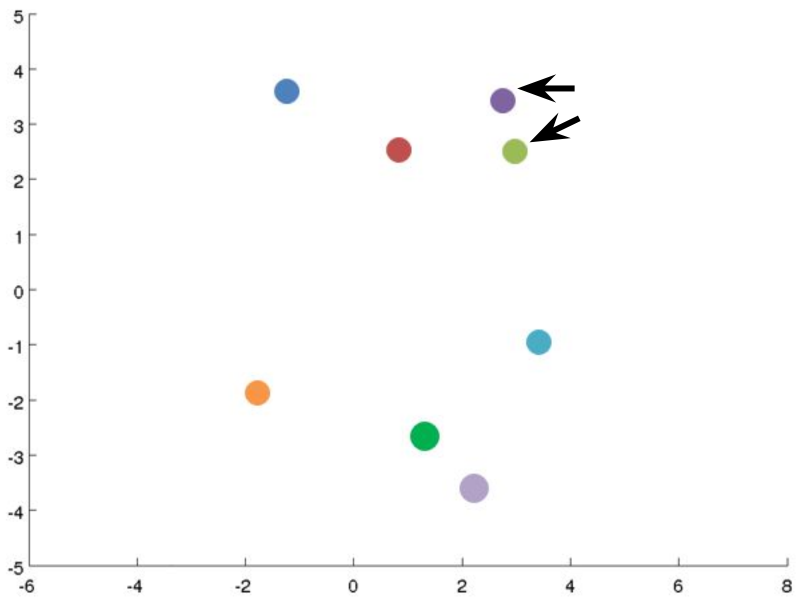
Which cluster should the orange dot belong to?

Today: Clustering

- Agglomerative Clustering
- **Divisive Clustering**
- K-means
- Vector Quantization with K-Means
- Mixtures of Gaussians
- Expectation Maximization

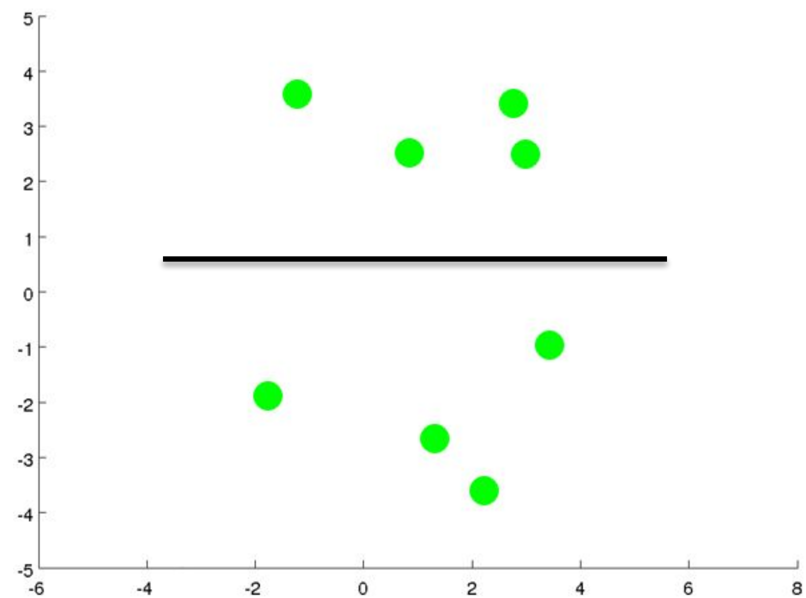
Clustering Method Comparison

Agglomerative Clustering

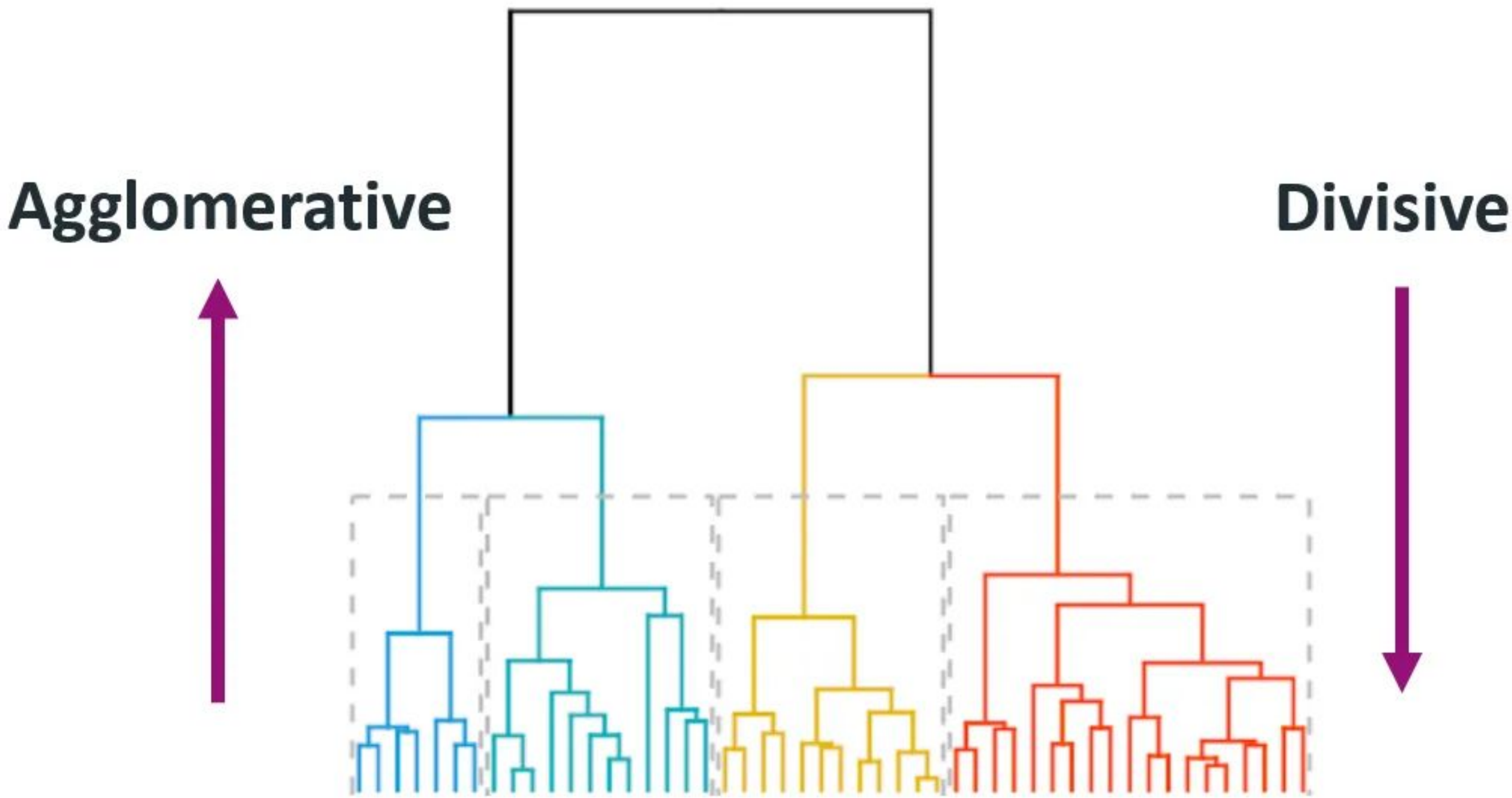


- Initializes each data point as its own cluster
- Merges cluster on each step

Divisive Clustering



- Initializes all data points as a single cluster
- Splits a cluster on each step


Agglomerative
Divisive

Do not edit

How to change the design

What are the scenarios where divisive clustering is more beneficial than agglomerative clustering? Select all that apply

What are the scenarios where divisive clustering is more beneficial than agglomerative clustering? Select all that apply

Divisive clustering is more suitable for large-scale datasets. ✓
83%

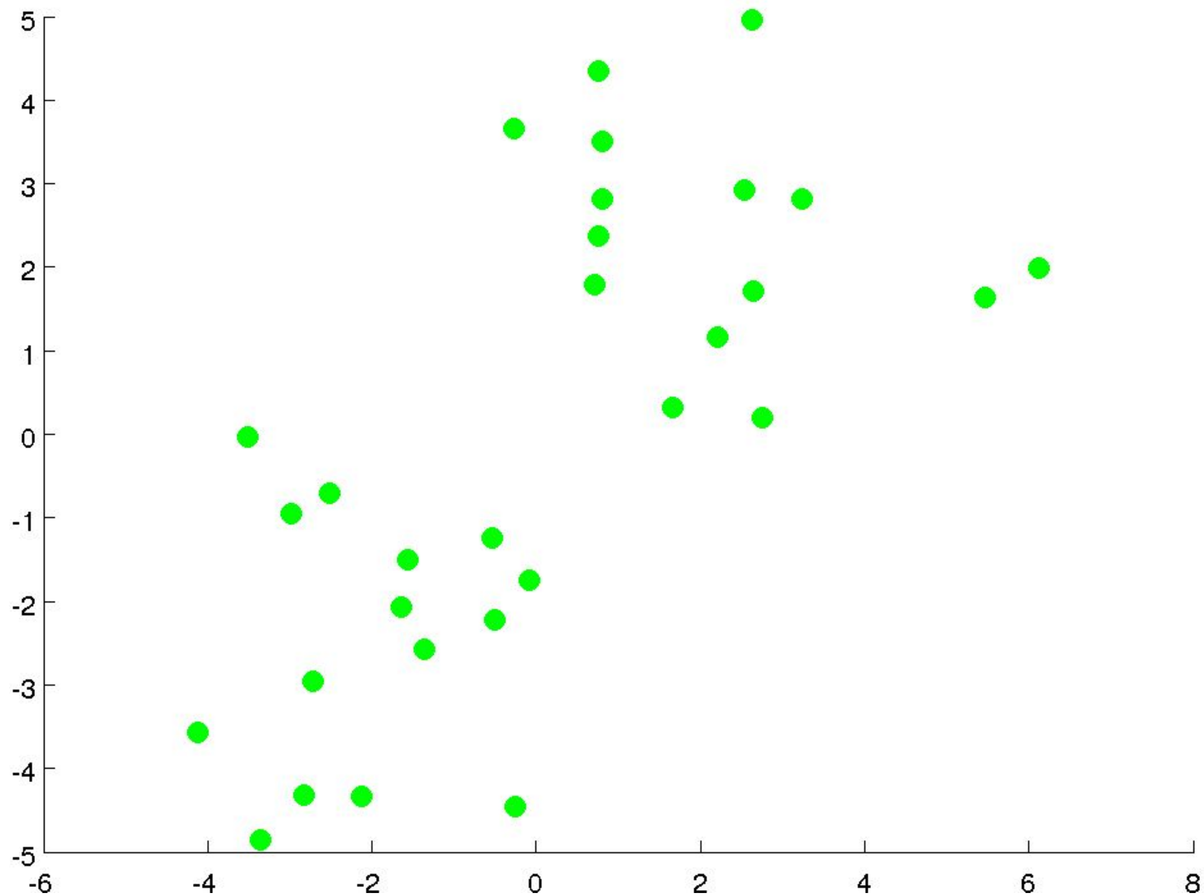
Divisive clustering is better at identifying larger, well-separated clusters ✓
90%

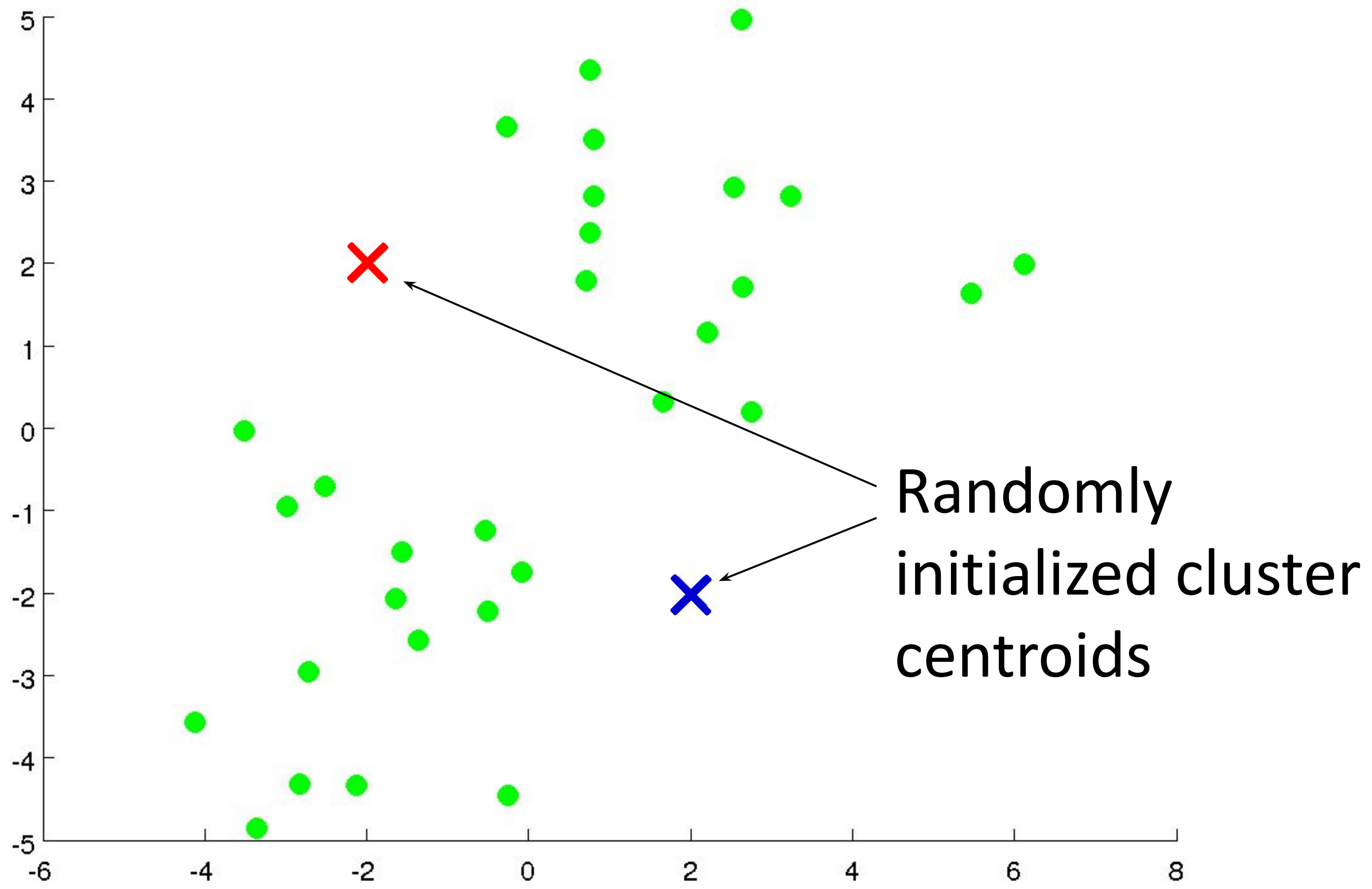
Divisive clustering is more intuitive than agglomerative clustering
43%

Both algorithms will converge to the same solution
24%

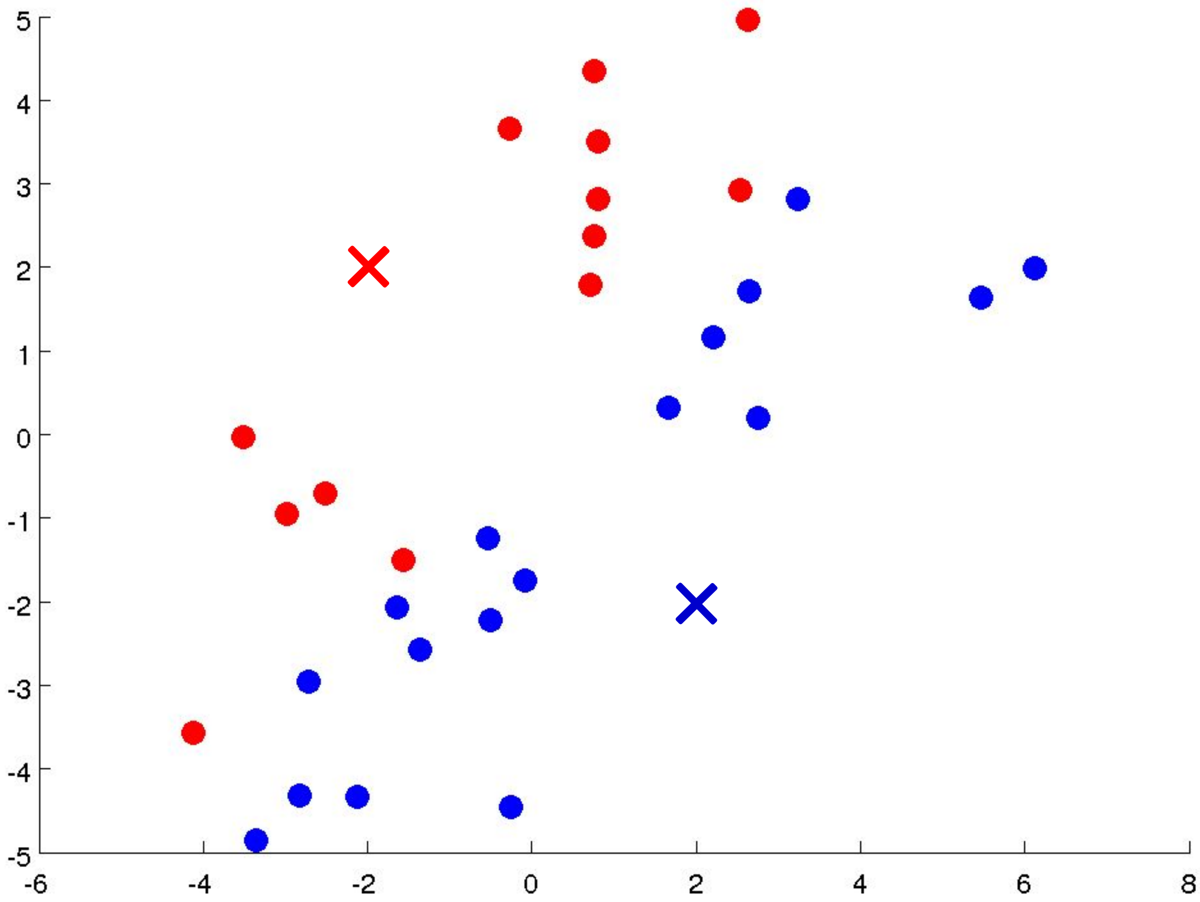
Today: Clustering

- Agglomerative Clustering
- Divisive Clustering
- **K-means**
- Vector Quantization with K-Means
- Mixtures of Gaussians
- Expectation Maximization

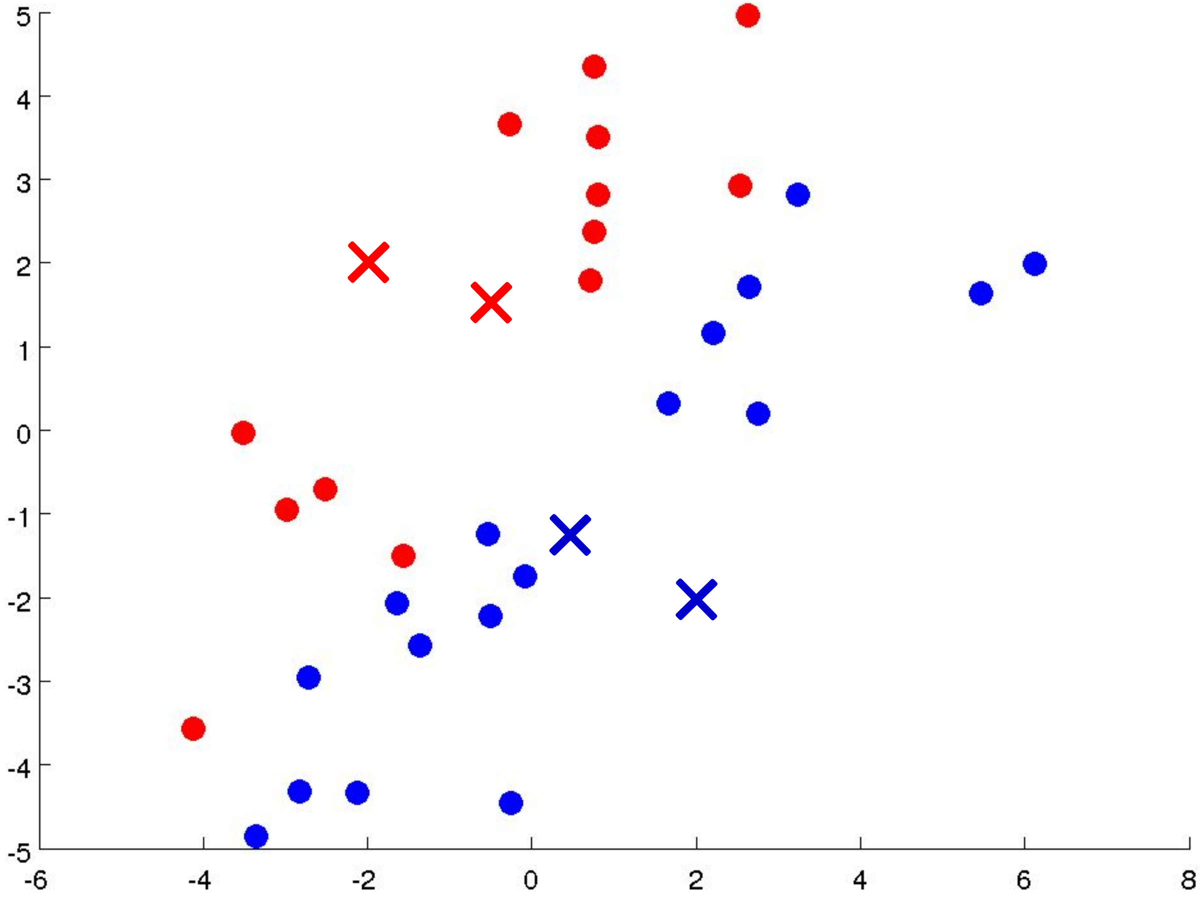

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8
Randomly initialized cluster centroids

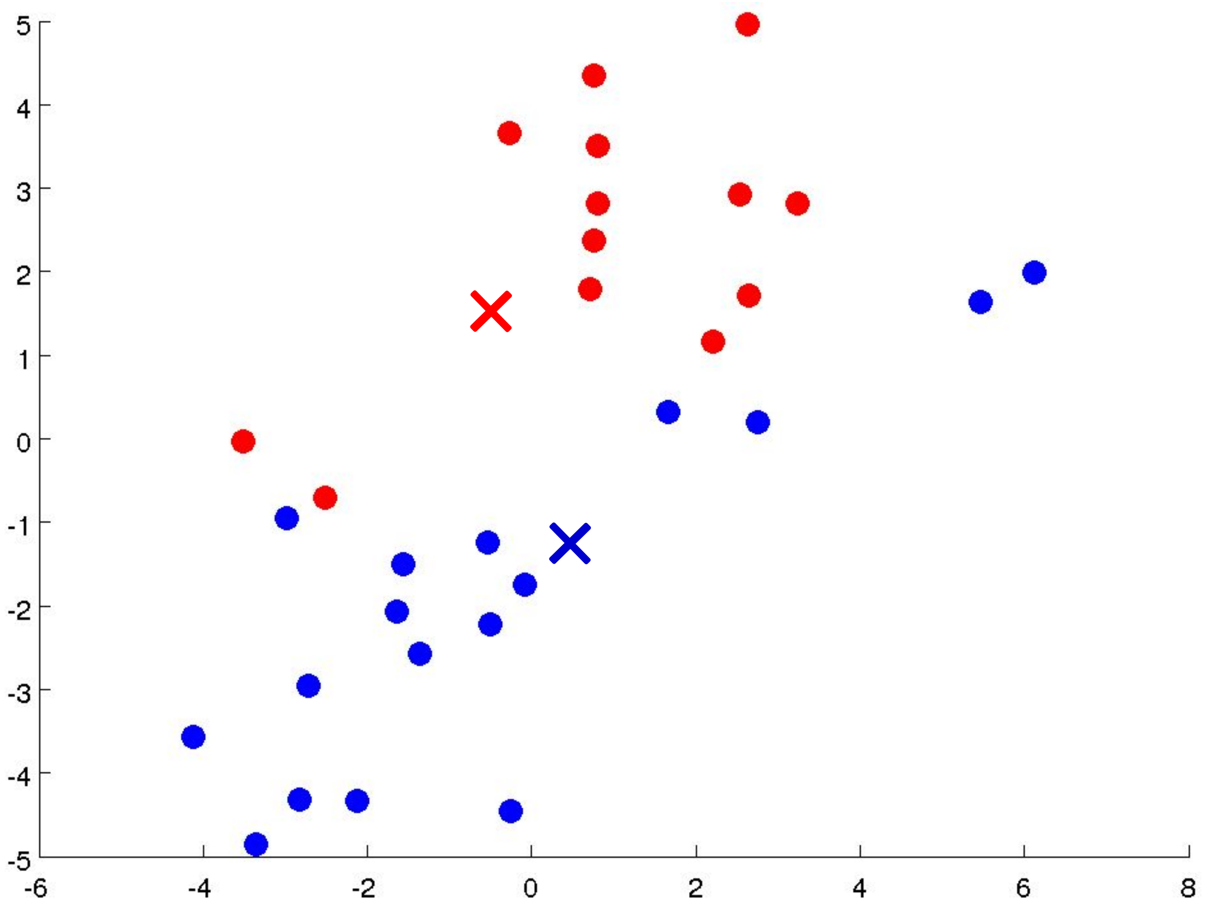
Assign rest of the points to closest cluster centroids



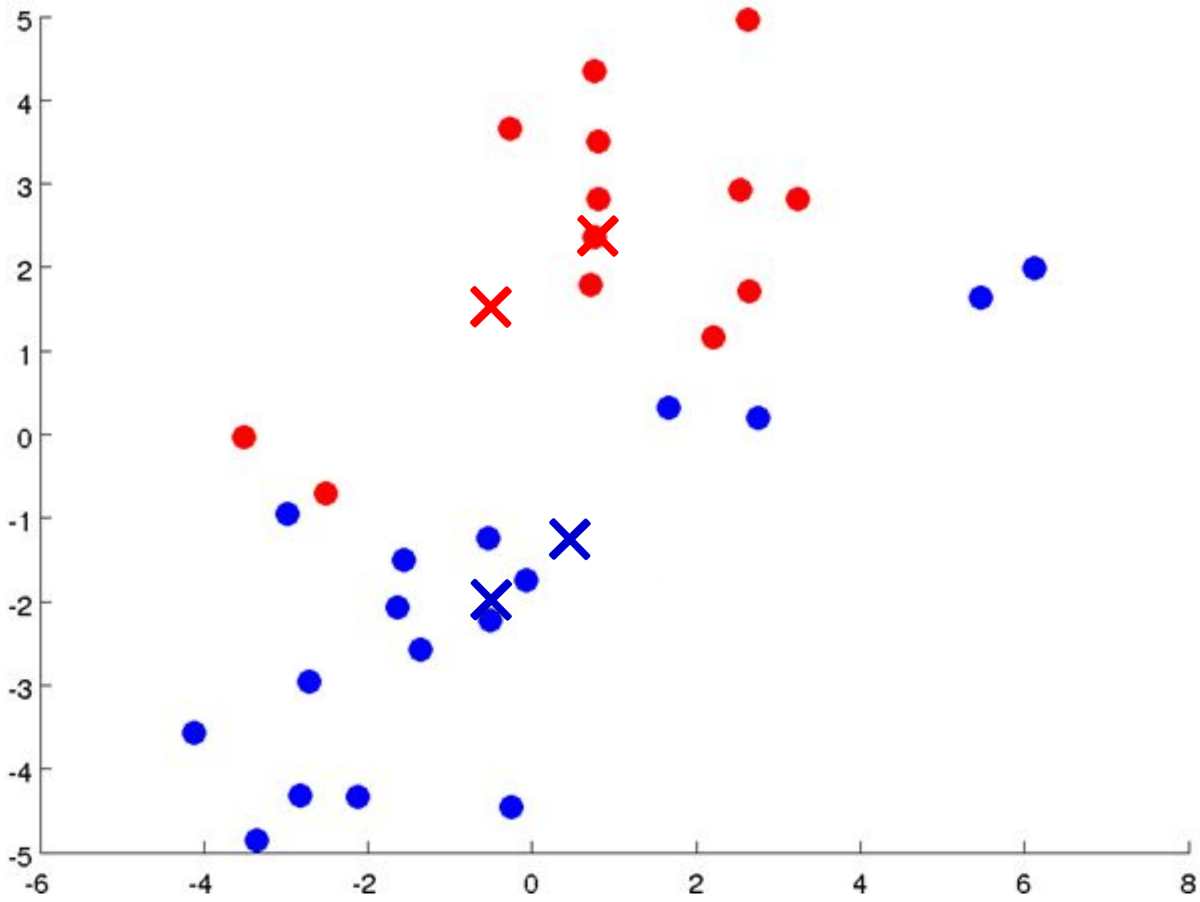
Recompute the cluster centroids



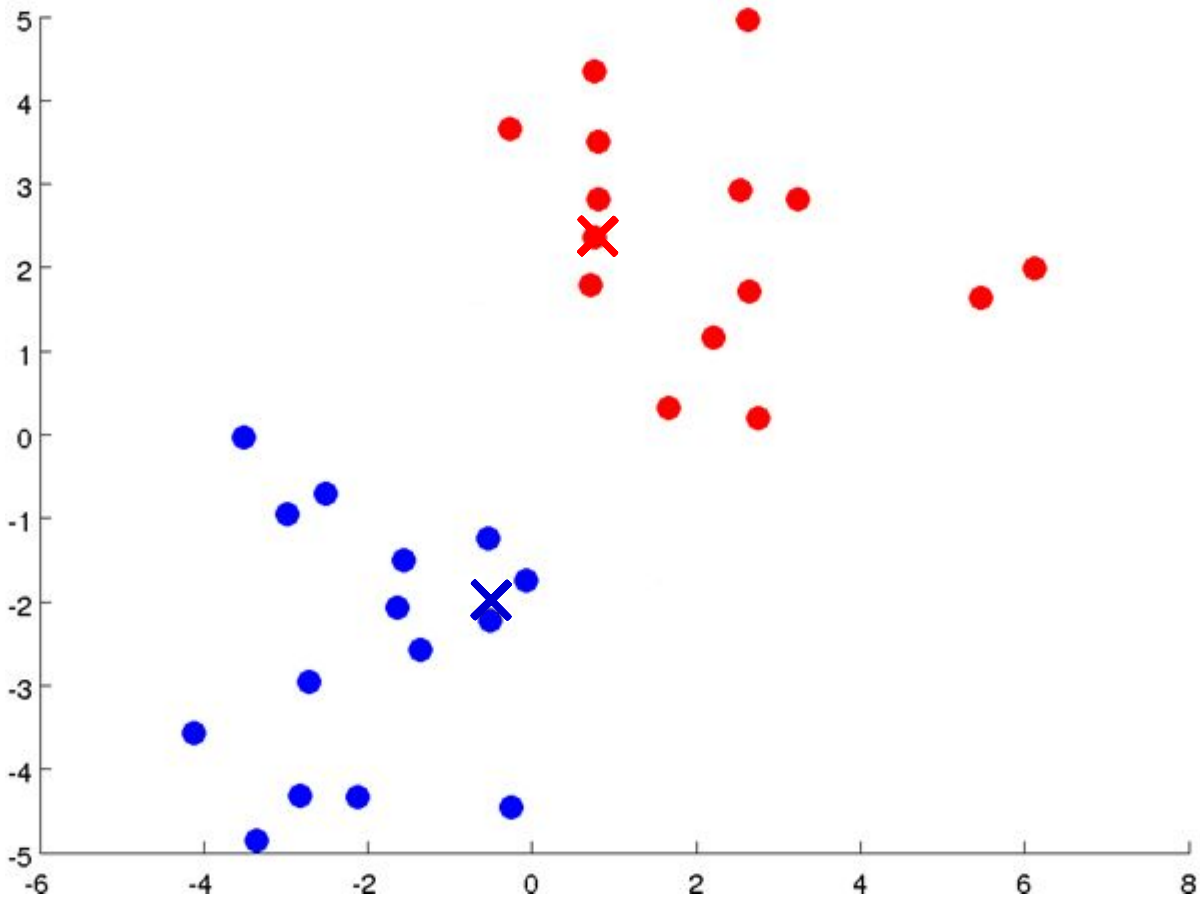
Reassign the points

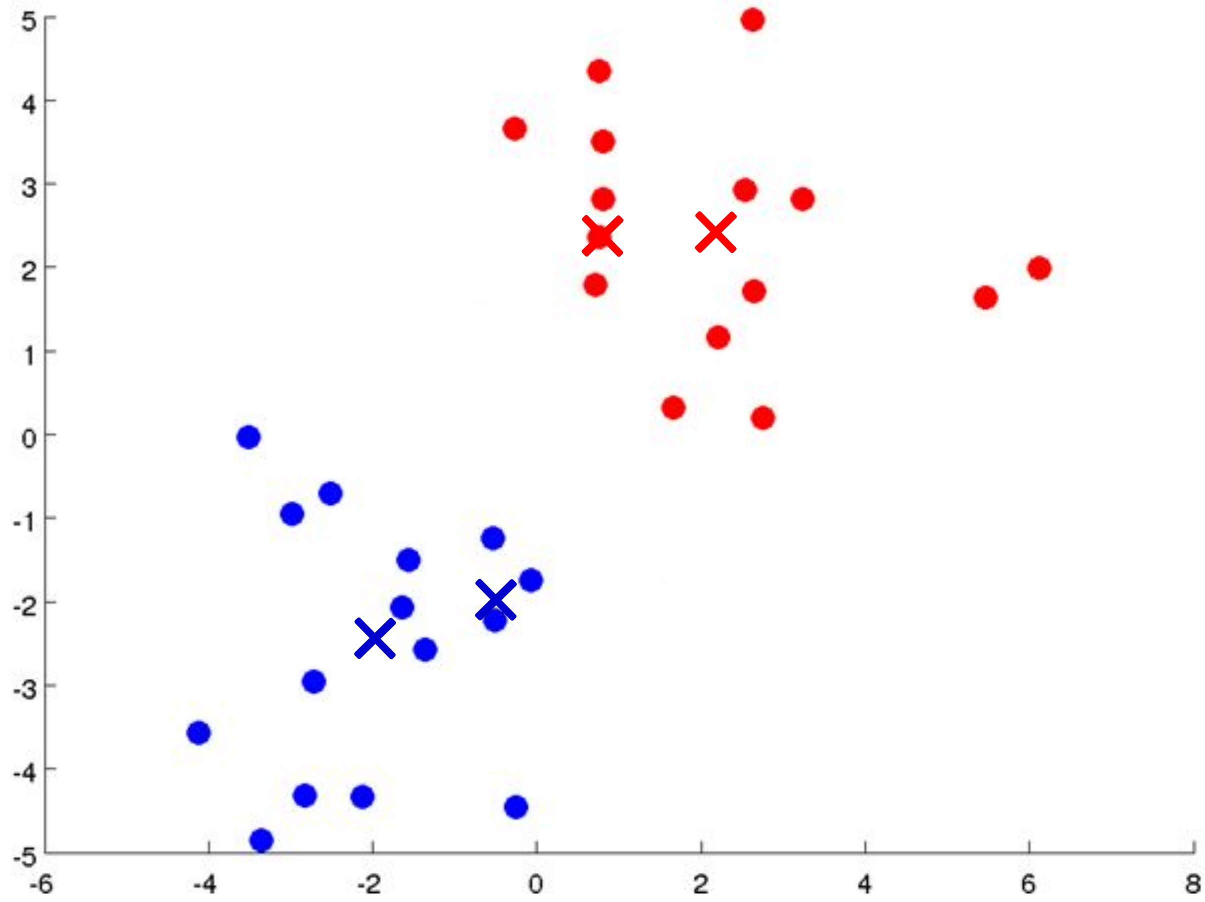


Recompute the cluster centroids

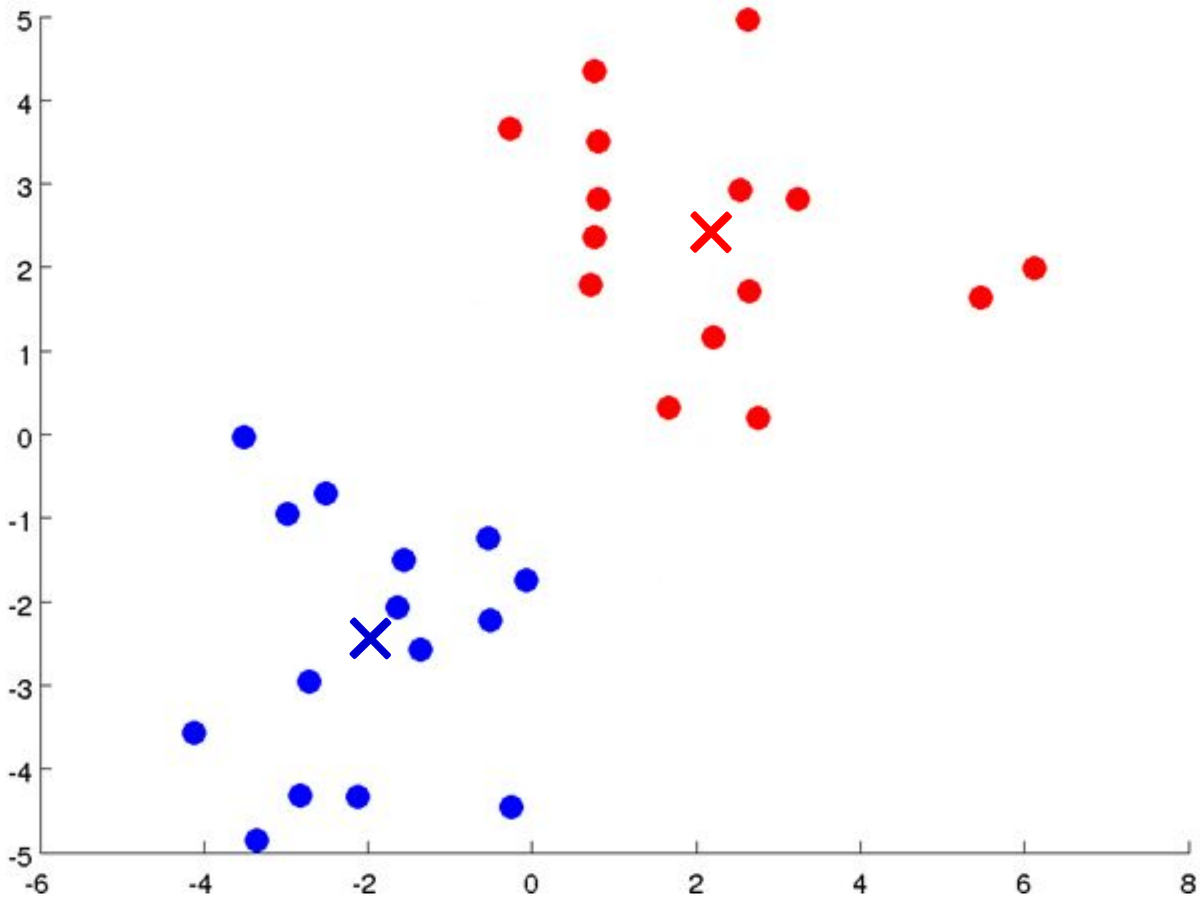


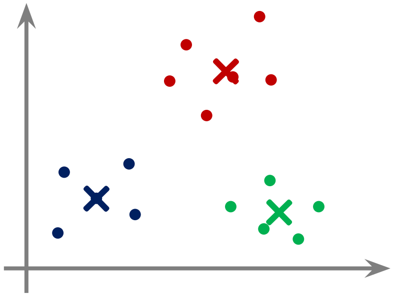
Reassign the points




5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

Recompute the cluster centroids

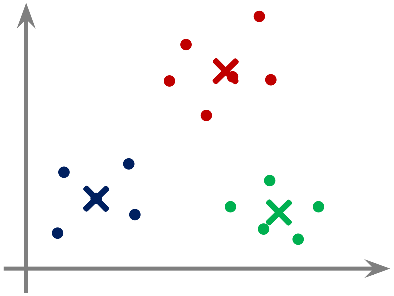




K-means algorithm

Input:

- K (number of clusters $\{c\}$)
- Training set $\{x_1, x_2, \ldots, x_N\}$



K-means algorithm

Randomly initialize K cluster centroids $c_1, c_2, \ldots, c_K$

Repeat {

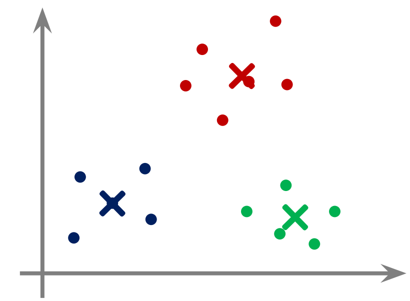
    for i = 1 to N

        $\delta_{i,j}$:= one–hot vector (of length K) where the cluster centroid j closest to x_i has value 1

    for $k = 1$ to K

        c_k:= average (mean) of points assigned to cluster k

}



K-means Cost Function

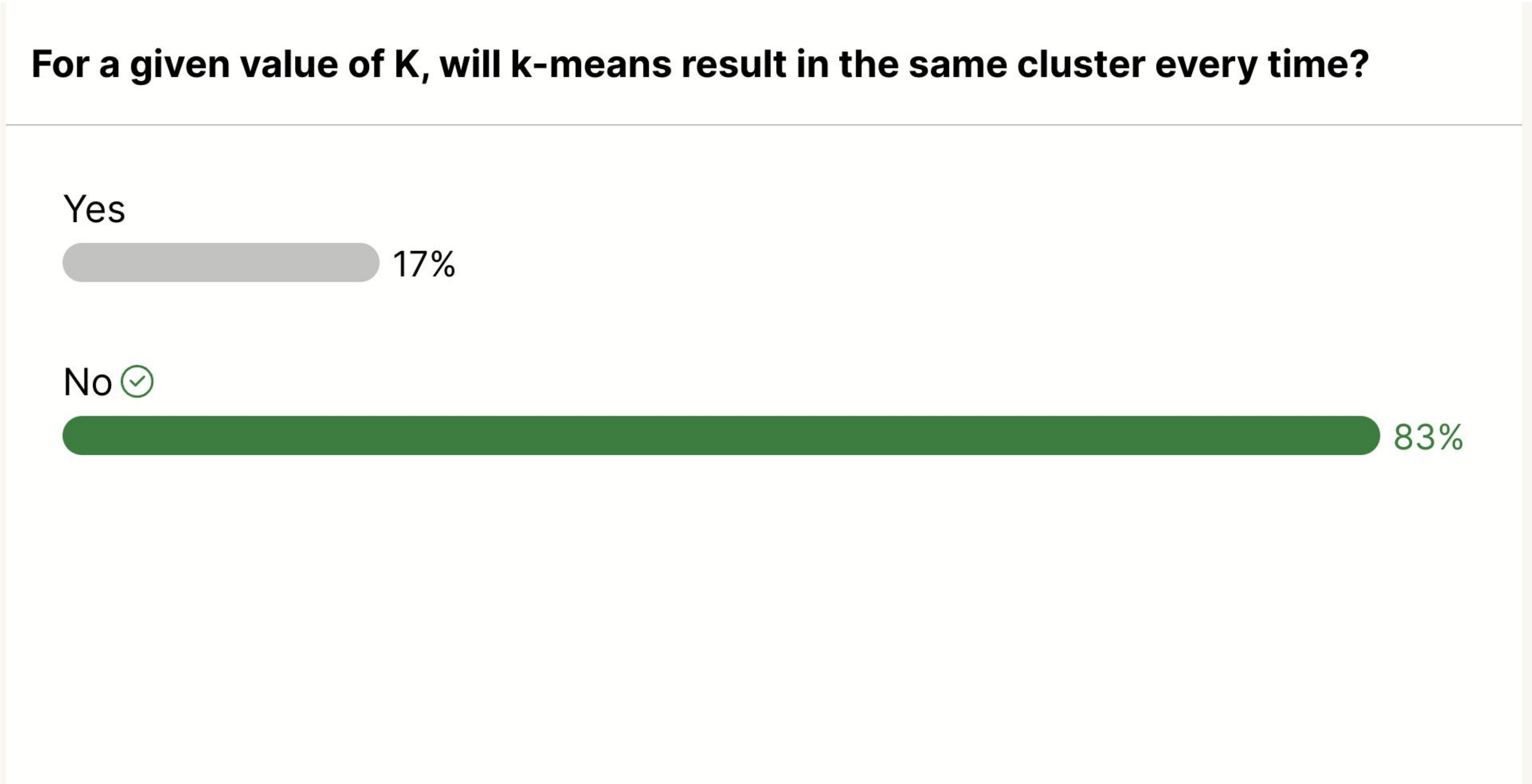
$\delta_{i,j}$ = one-hot vector vector (of length K) where the cluster centroid j closest to x_i has value 1

c_j = cluster centroid j

Optimization cost: “distortion”

$$\Phi(\delta, c) = \sum_{i,j} \delta_{i,j} \left[\underbrace{(x_i - c_j)^T (x_i - c_j)}_{\text{Intra-cluster compactness}} \right]$$

For a given value of K, will k-means result in the same cluster every time?


For a given value of K, will k-means result in the same cluster every time?
Yes
17%
No
83%

Factors that lead to different clusters for the same dataset

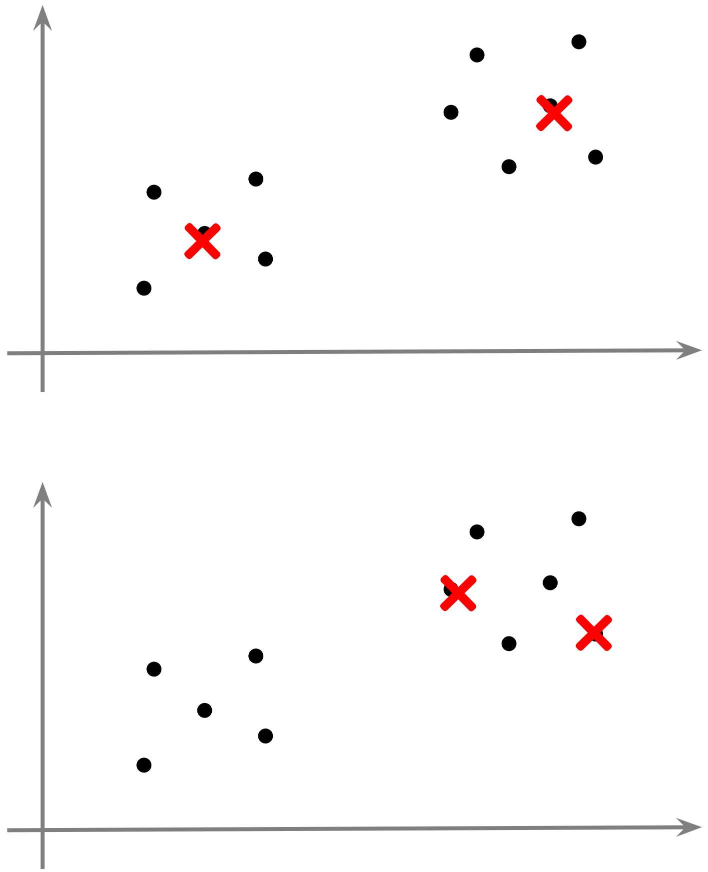
- Random initialization of cluster centers
- Distance metric
- Cluster assignment criteria.

Random initialization

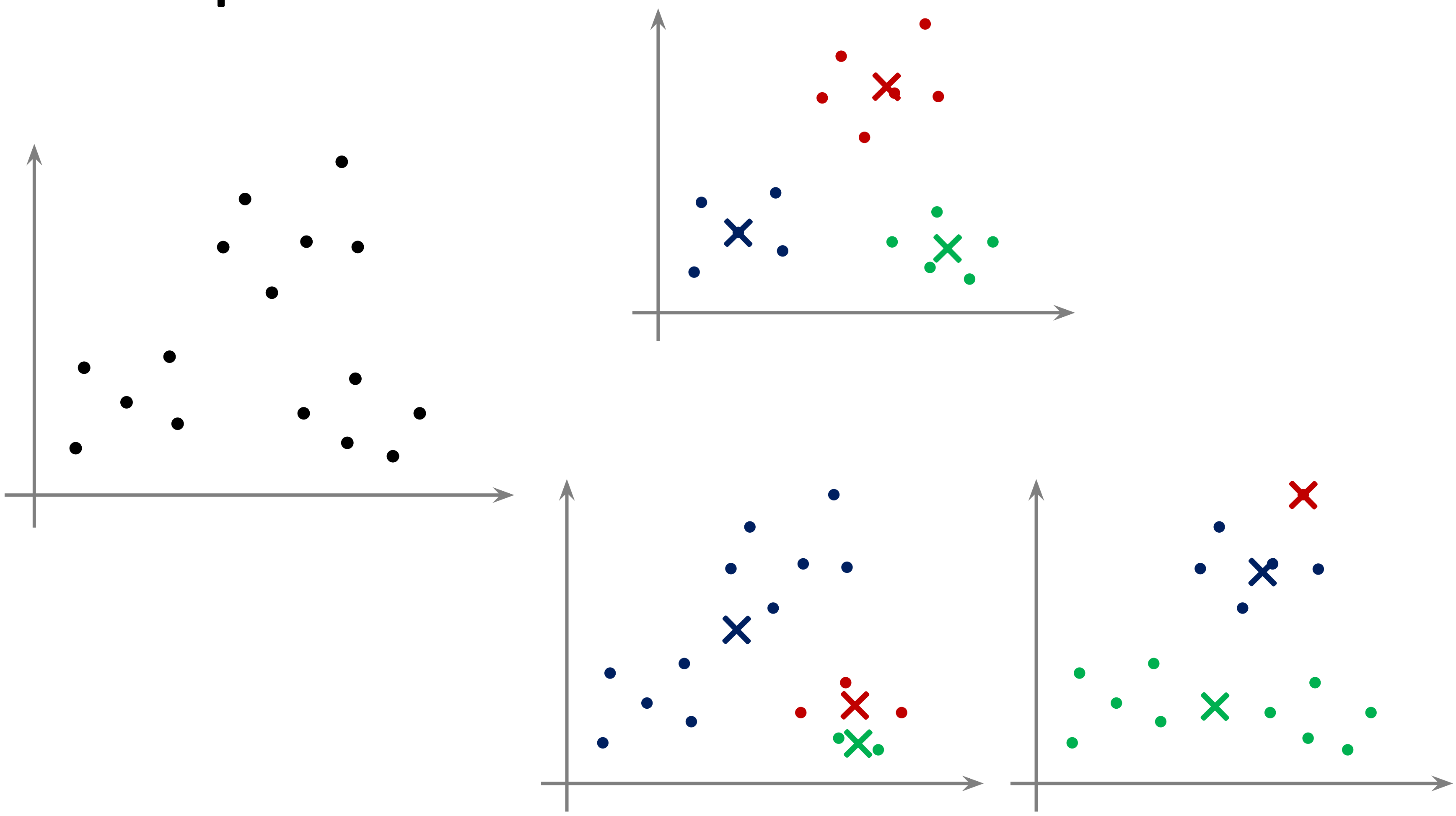
Should have $K < N$

Randomly pick K training examples.

Set $c_1, c_2, \ldots, c_K$ equal to these K examples.



Local Optima



Avoiding Local Optima with Random Initialization

For i = 1 to 100 {

Randomly initialize K-means.
Run K-means. Get $\delta_1, \delta_2, \ldots, \delta_N$ and $c_1, c_2, \ldots, c_K$.
Compute cost function (distortion)

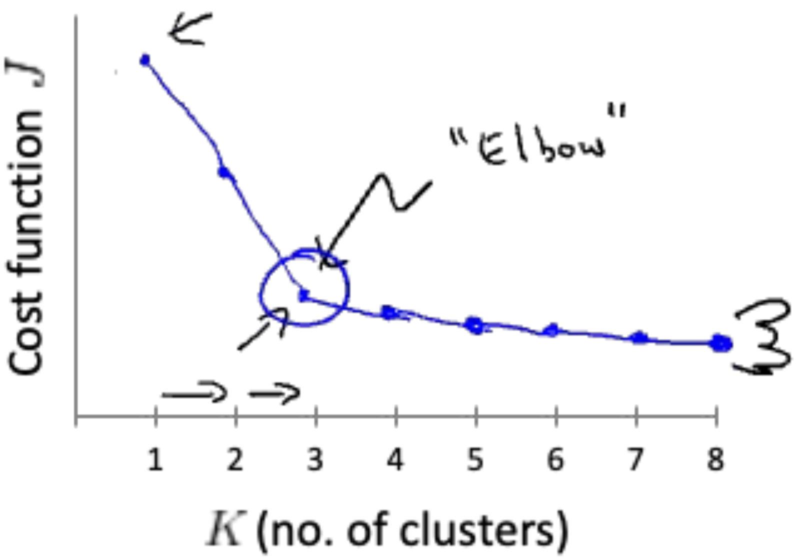
$$\Phi(\delta, c)$$

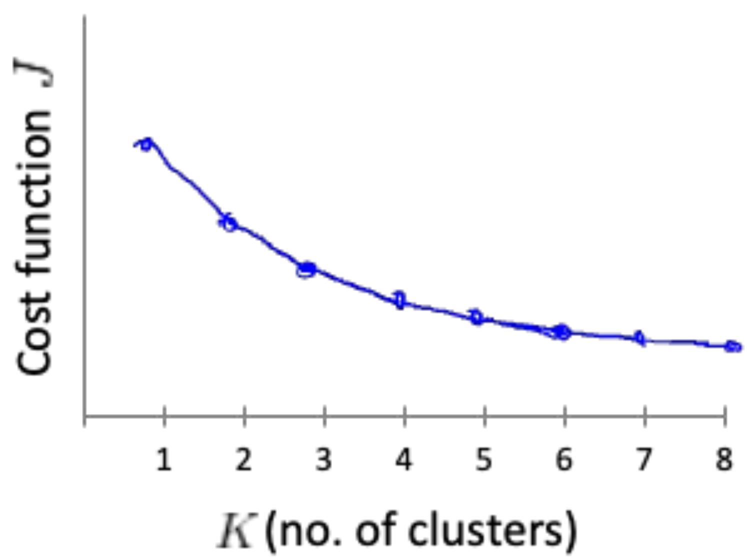
}

Pick clustering that gave lowest cost $\Phi(\delta, c)$

How to choose K?

Called the elbow method

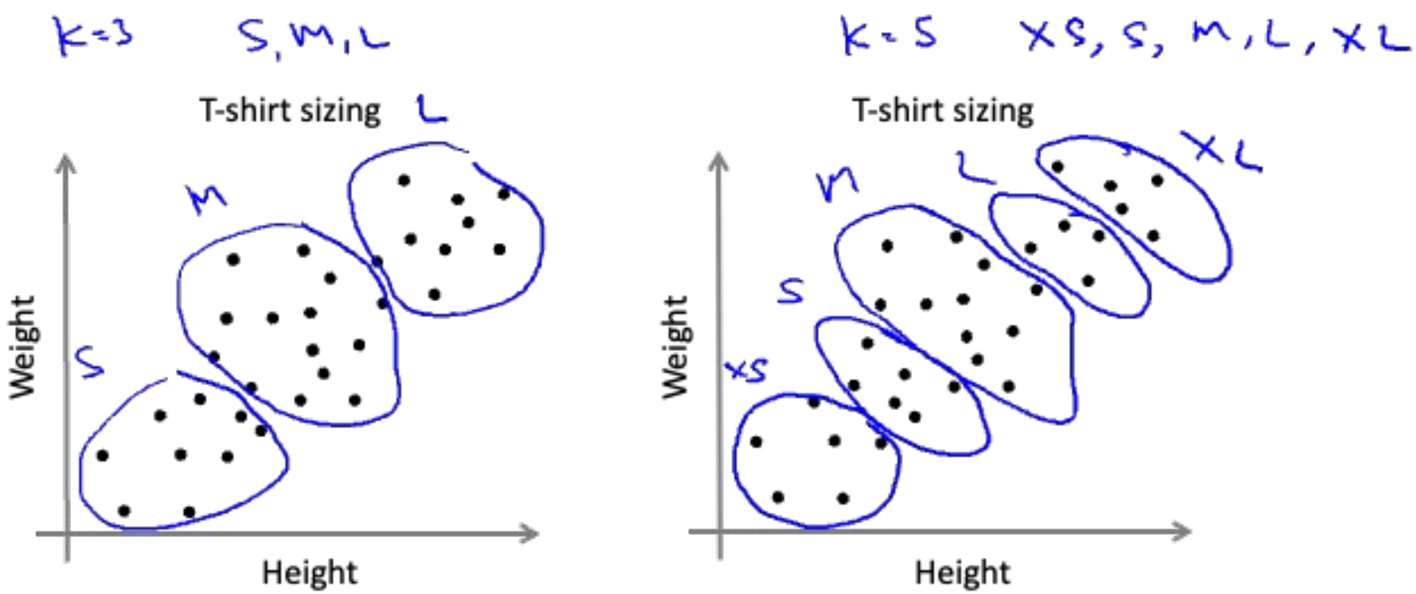


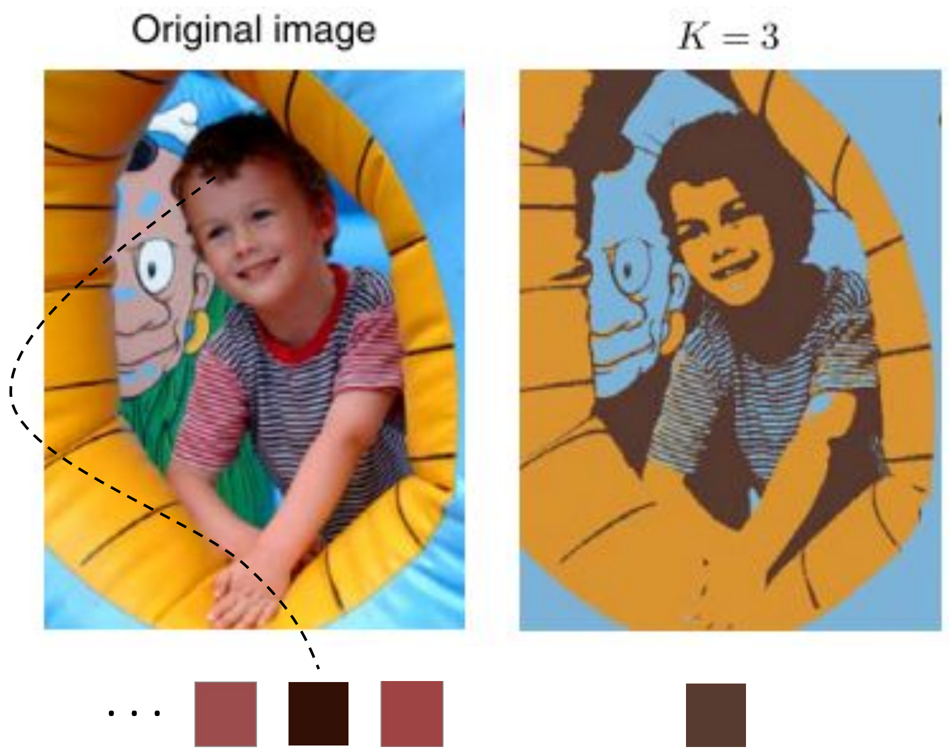


How to choose K?

Sometimes, you're running K-means to get clusters to use for some later/downstream purpose. Evaluate K-means based on a metric for how well it performs for that later purpose.

E.g.





$$x_i = \begin{bmatrix} 138 \\ 80 \\ 79 \end{bmatrix} \rightarrow \delta_{i,j}$$

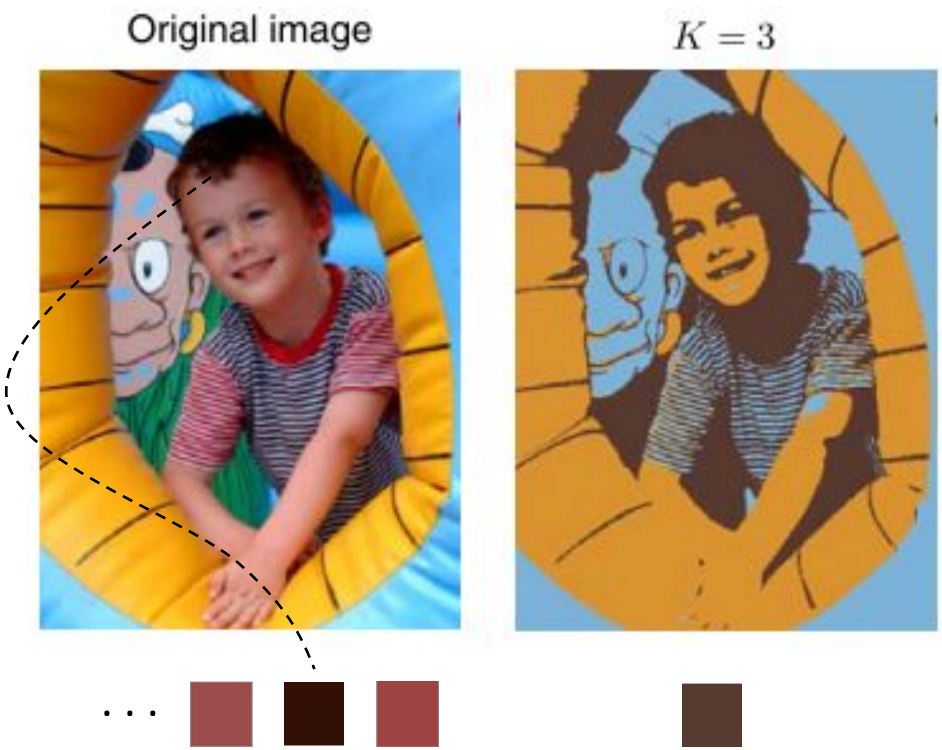
- Each {R, G, B} pixel value is an input vector x_i (255 x 255 x 255 possible values)
- **Problem:** Memory scales exponentially with image resolution.

- **One solution:** Compress an image using K-means

Today: Clustering

- Agglomerative Clustering
- Divisive Clustering
- K-means
- **Vector Quantization with K-Means**
- Mixtures of Gaussians
- Expectation Maximization

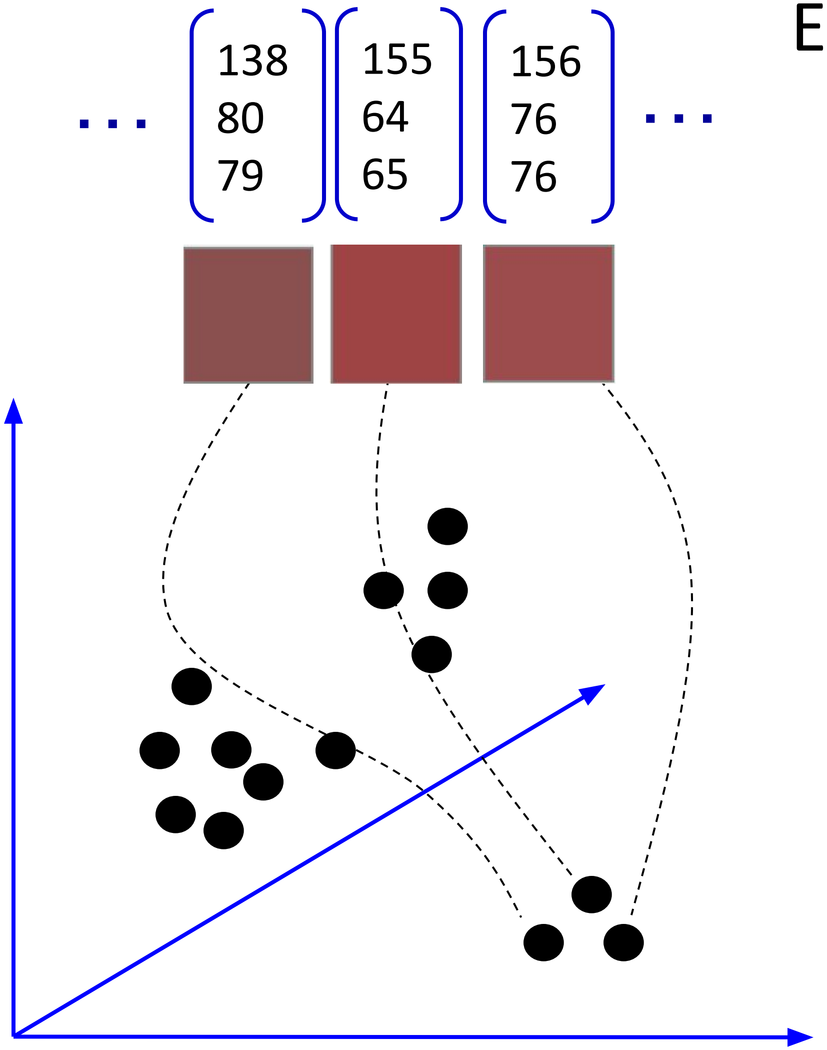
Application of Clustering: Vector Quantization



$$x_i = \begin{bmatrix}13\\8\\80\\79\end{bmatrix} \rightarrow \delta_{i,j}$$

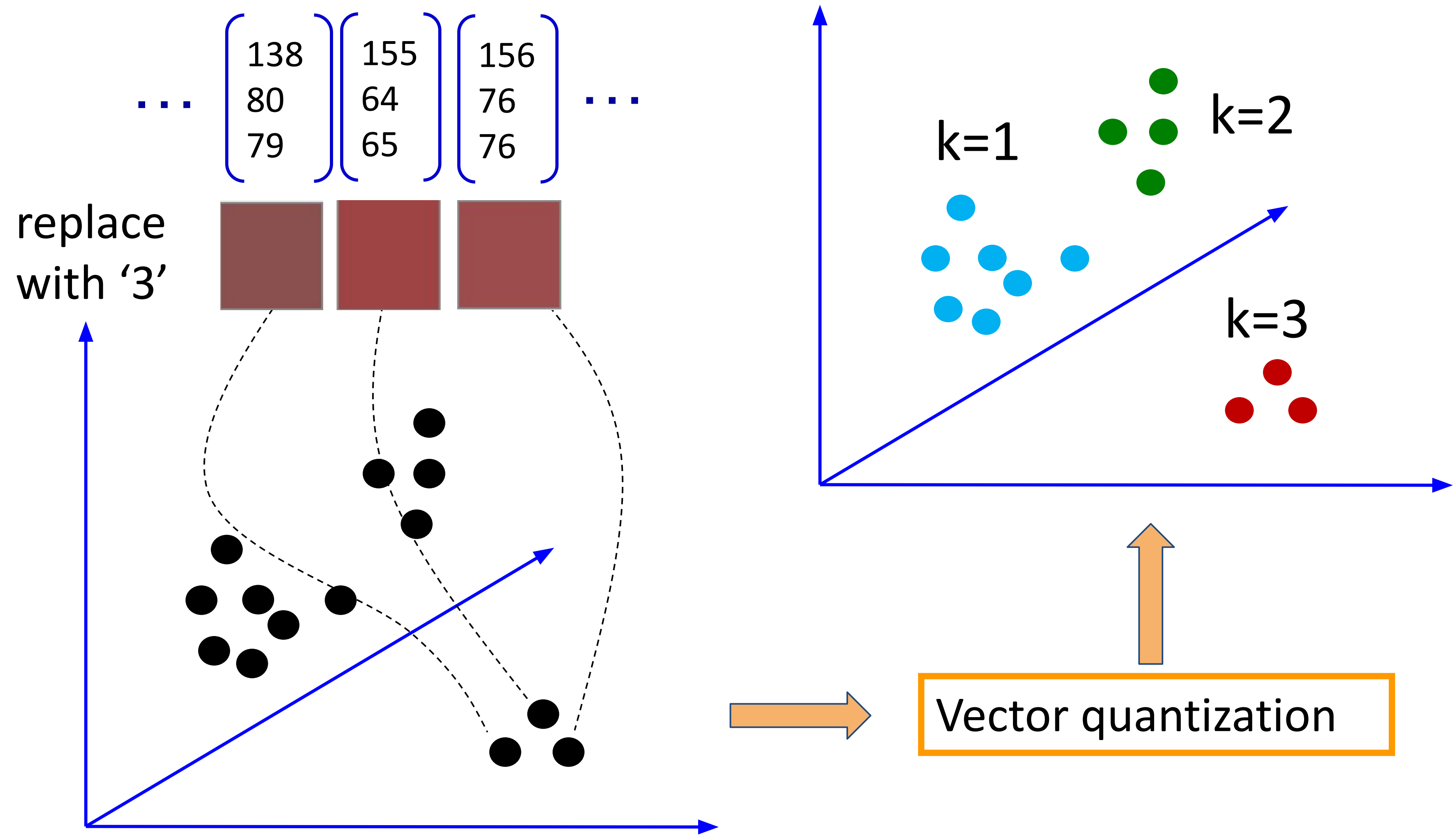
- Each {R, G, B} pixel value is an input vector x_i (255 x 255 x 255 possible values)
- **Problem:** Memory scales exponentially with image resolution.
- **One solution:** Compress an image using K-means
- Replace each vector by its cluster assignment $\delta_{i,j}$ (K possible values)

Vector quantization: color values

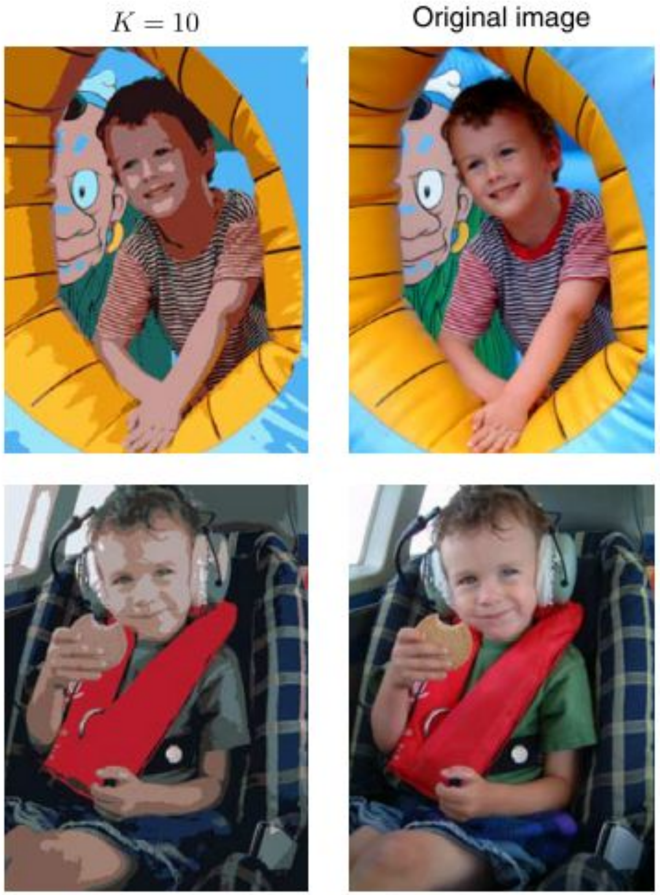


Example: R, G, B vectors

Vector quantization: color values

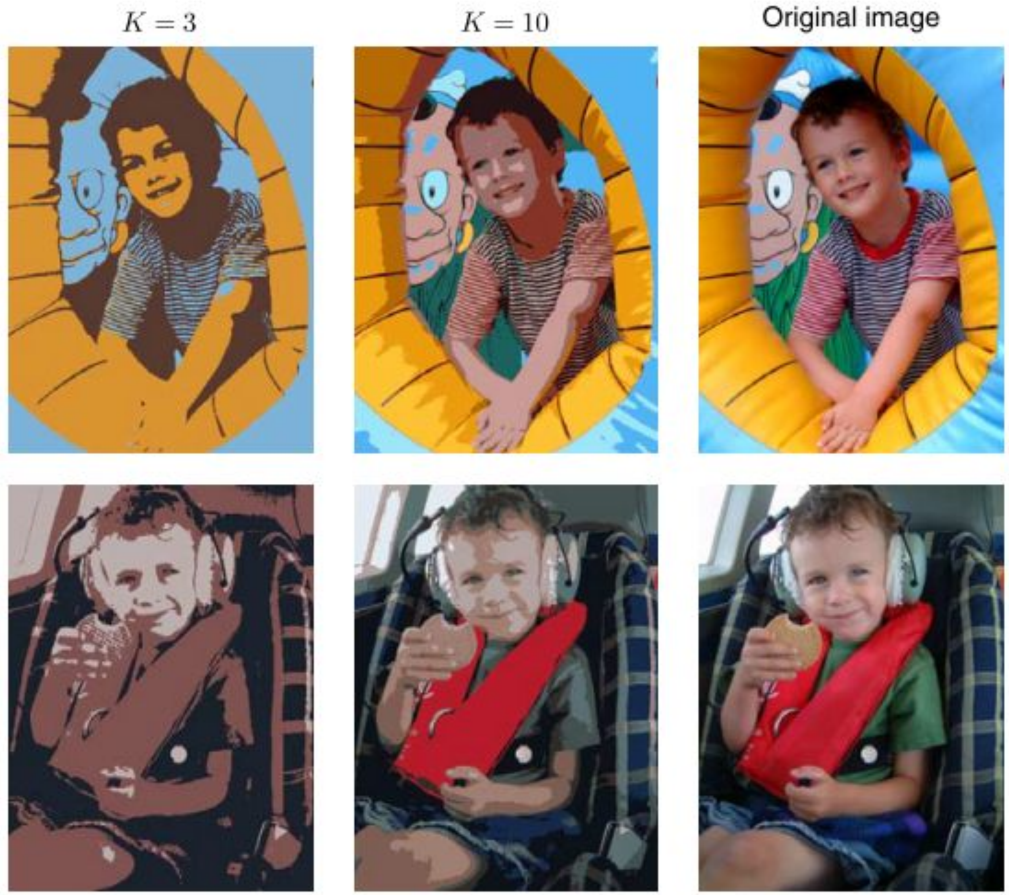


K-Means for Image Compression



Bishop **Figure 9.3** Two examples of the application of the K-means clustering algorithm to image segmentation showing the initial images together with their K-means segmentations obtained using various values of K. This also illustrates of the use of vector quantization for data compression, in which smaller values of K give higher compression at the expense of poorer image quality.

K-Means for Image Compression



Bishop **Figure 9.3** Two examples of the application of the K-means clustering algorithm to image segmentation showing the initial images together with their K-means segmentations obtained using various values of K. This also illustrates of the use of vector quantization for data compression, in which smaller values of K give higher compression at the expense of poorer image quality.

K-Means for Image Compression

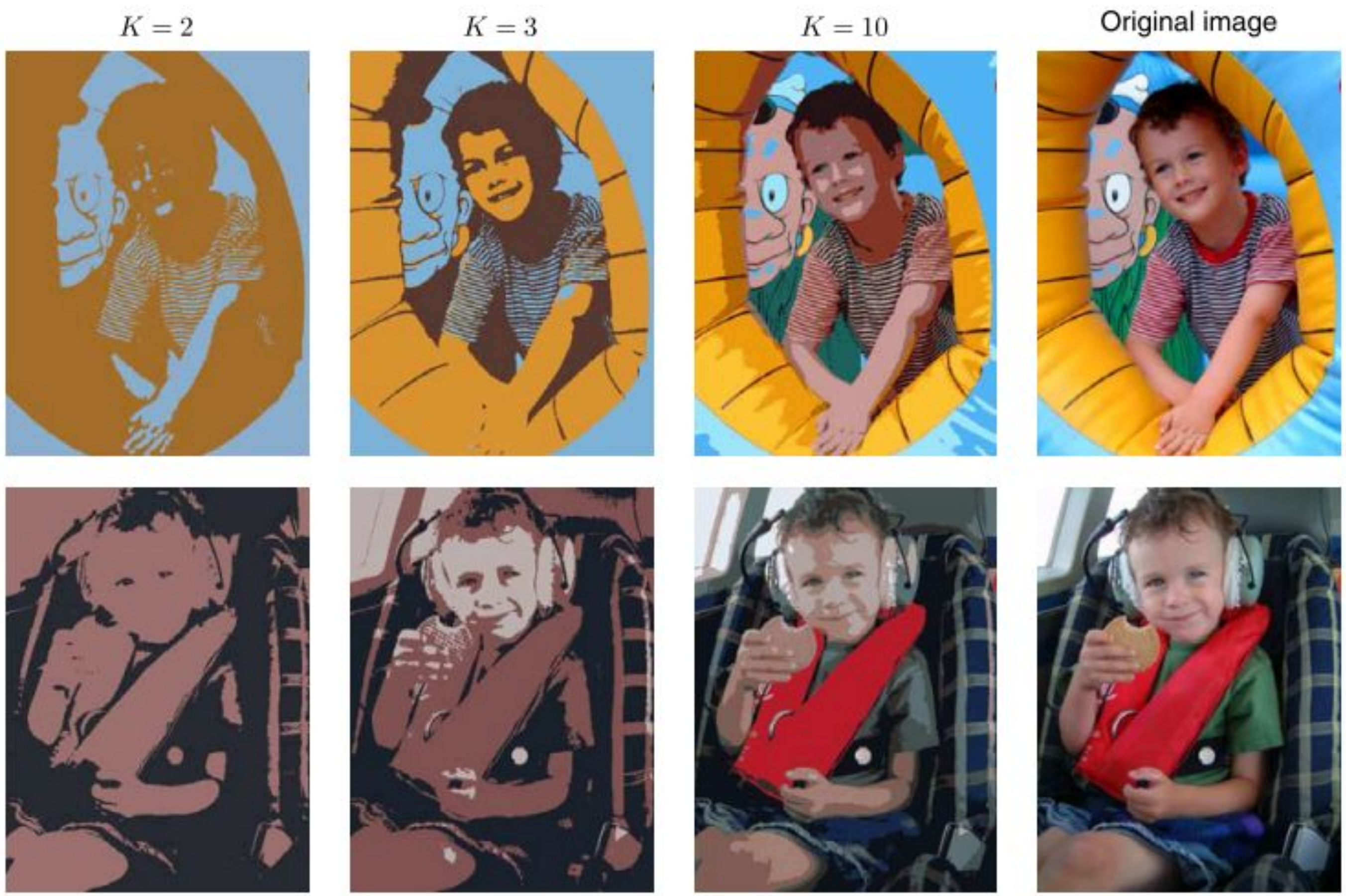
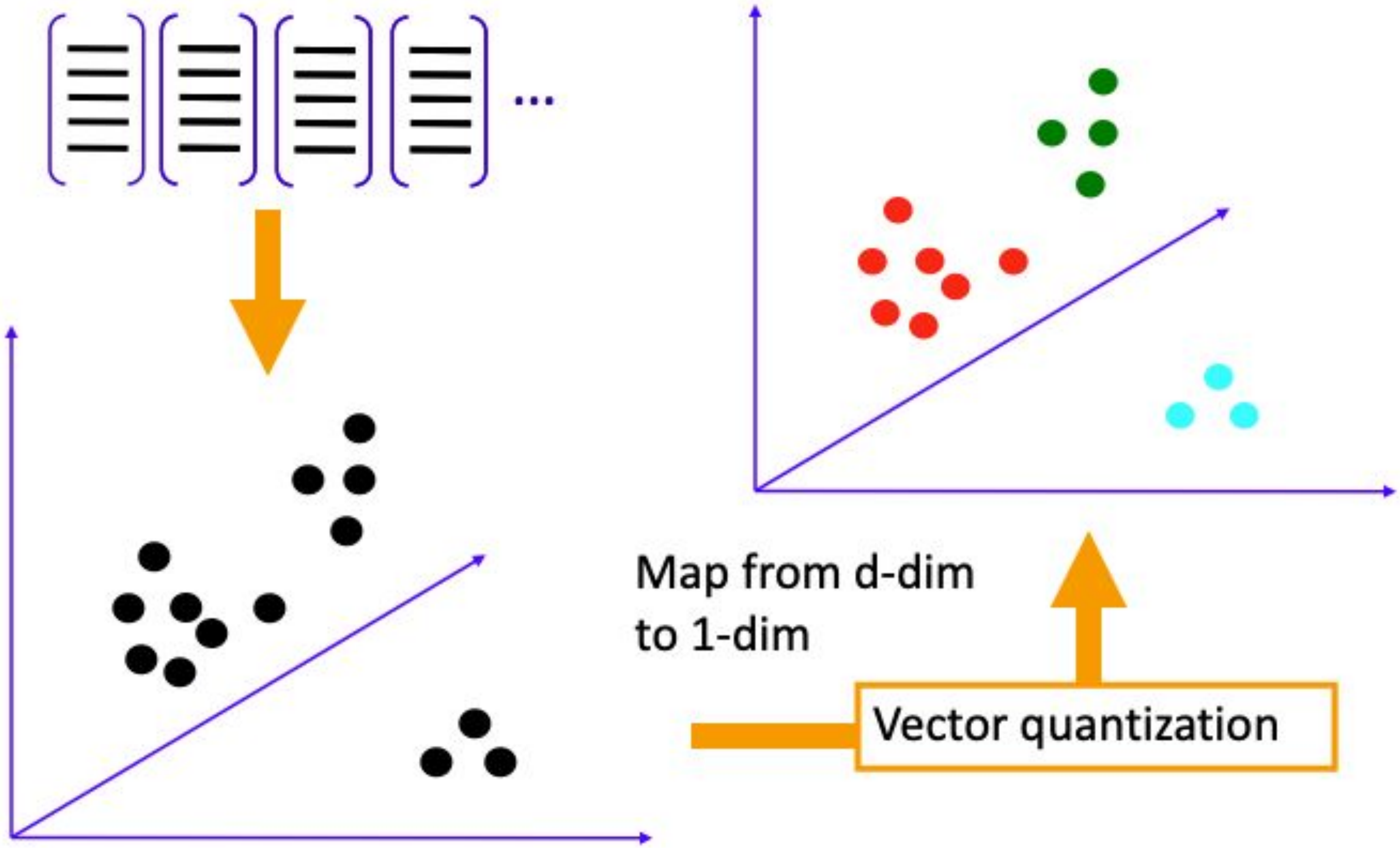


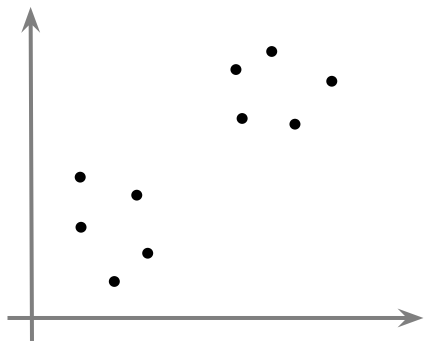
Figure 9.3 Two examples of the application of the K-means clustering algorithm to image segmentation showing the initial images together with their K-means segmentations obtained using various values of K. This also illustrates of the use of vector quantization for data compression, in which smaller values of K give higher compression at the expense of poorer image quality.

Vector quantization: general case



Where else can vector quantization come handy?

Unsupervised learning

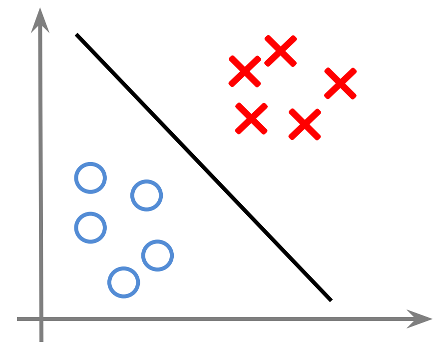


Training set: $\{x_1, x_2, x_3, \dots\}$

Vector quantization

v/s

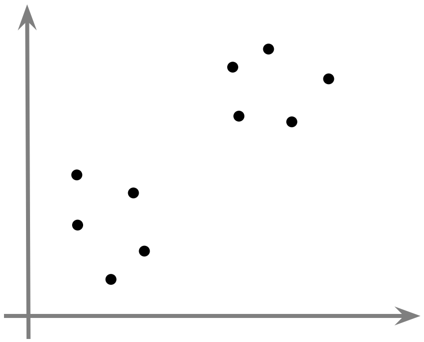
Supervised learning



Training set: $\{(x_1, y_1), (x_2, y_2) \dots, (x_N, y_N)\}$

Where else can vector quantization come handy?

Unsupervised learning

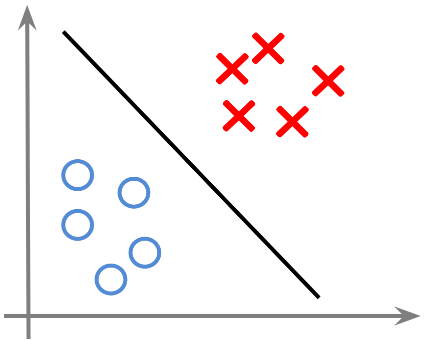


Training set: $\{x_1, x_2, x_3, \ldots\}$

Vector quantization

v/s

Supervised learning



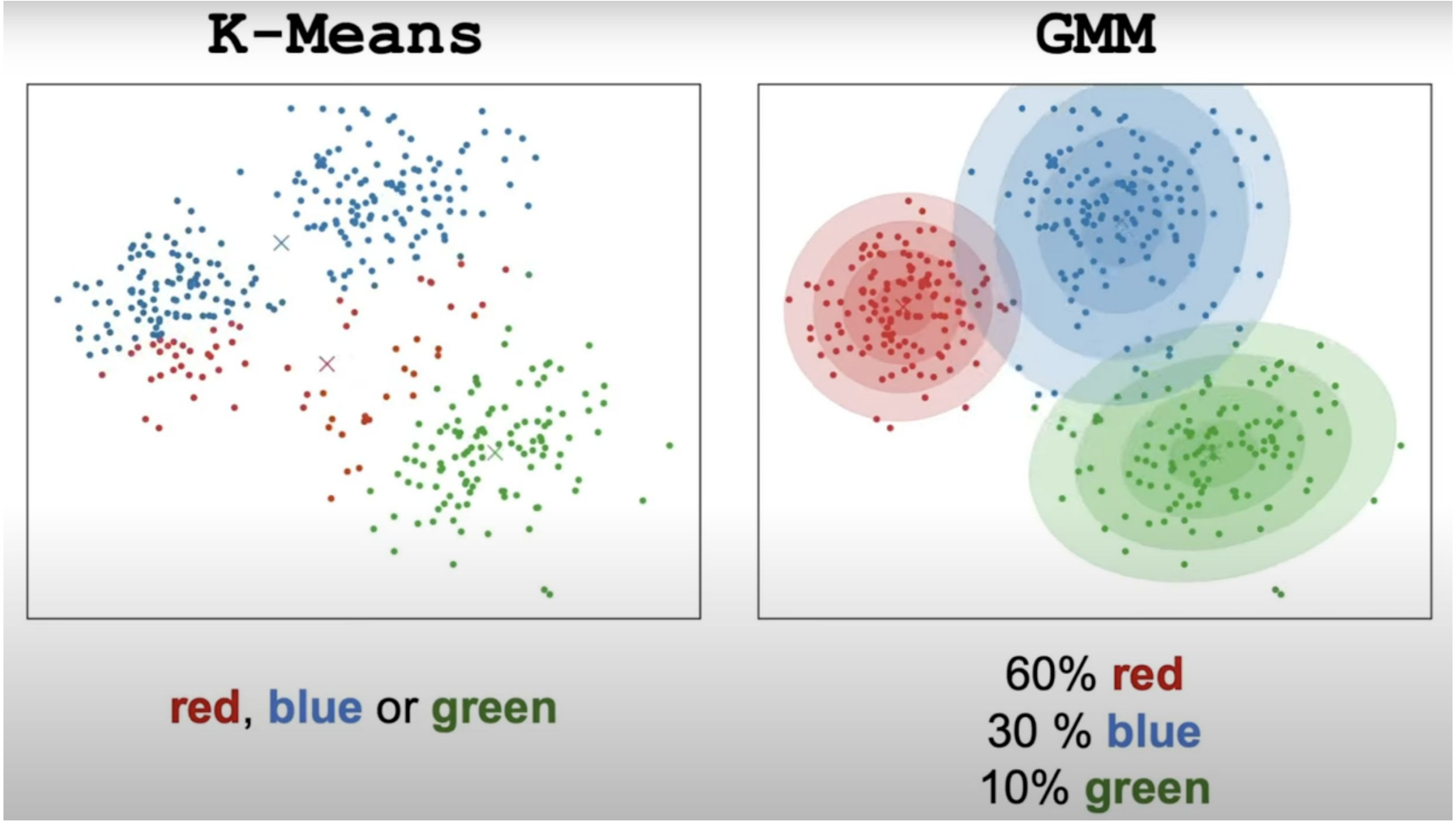
Training set: $\{(x_1, y_1), (x_2, y_2) \ldots, (x_N, y_N)\}$

Label learning

Today: Clustering

- Agglomerative Clustering
- Divisive Clustering
- K-means
- Vector Quantization with K-Means
- **Mixtures of Gaussians**
- Expectation Maximization

K-means v/s Gaussian Mixture models



What does Gaussian Mixture Models offer over k-means?

What does Gaussian Mixture Models offer over k-means?

GMMs are more complex given the lack of hard assignments to a given cluster

72%

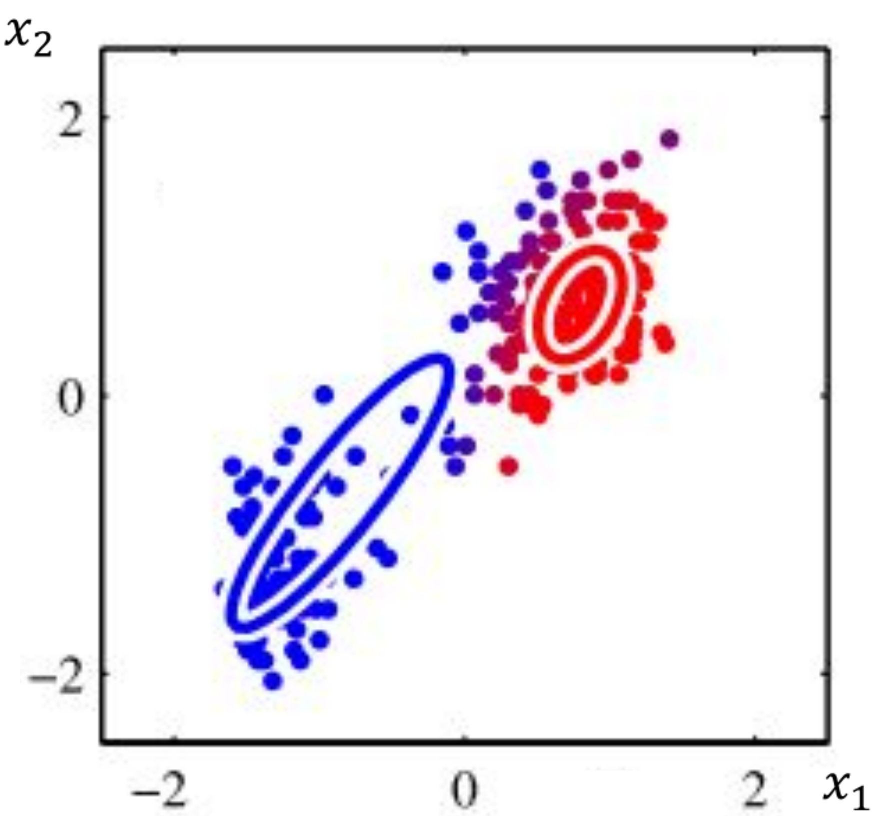
Robustness to noise and outliers

87%

Flexibility in terms of data labeling due to soft assignments

91%

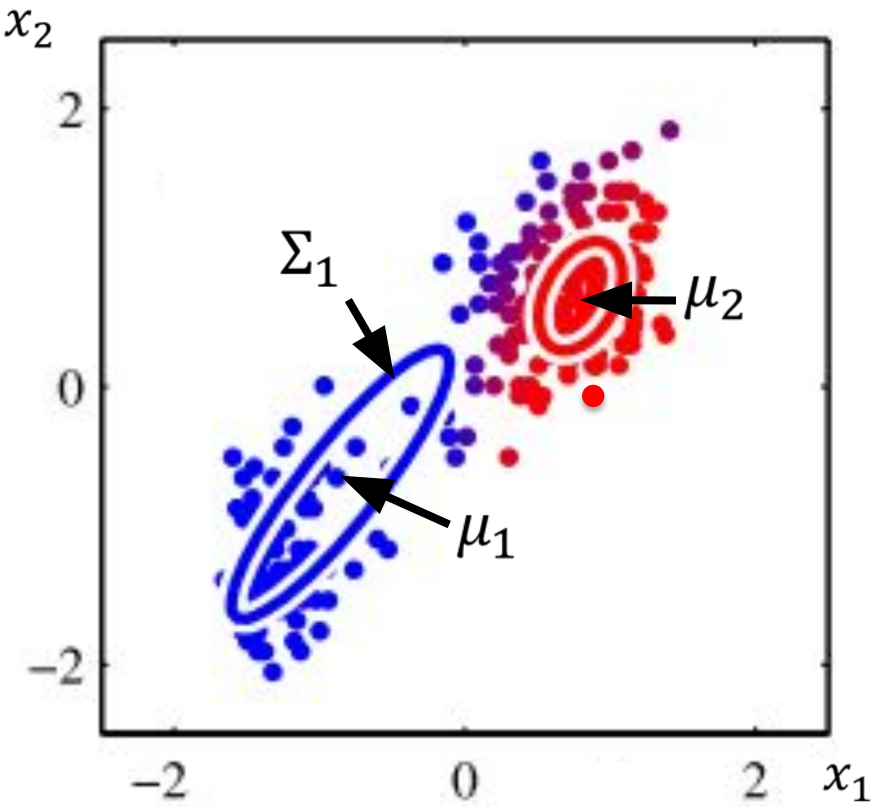
Mixtures of Gaussians: Intuition



“Soft” cluster membership

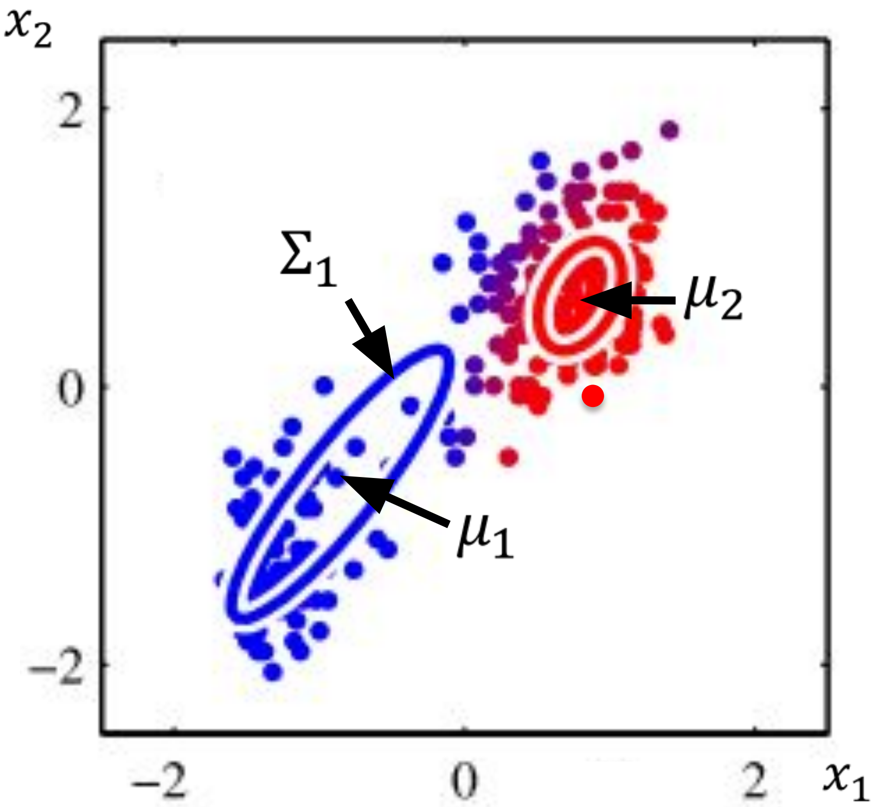
To generate each point in $\boldsymbol{x}$,
- Choose its cluster component δ
- Sample $\boldsymbol{x}$ from the Gaussian distribution for that component
- What do we need to define a Gaussian distribution?

Mixtures of Gaussians



- Two parameters:
  mean (μ), variance(Σ)
- Assume K components, k-th component is a Gaussian with parameters μ_k, Σ_k

Mixtures of Gaussians



- Introduce discrete r.v. $\delta \in R^K$ that denotes the component that generates the point
- one element of δ is equal to 1 and others are 0, i.e. "one-hot": $\delta_k \in \{0,1\}$

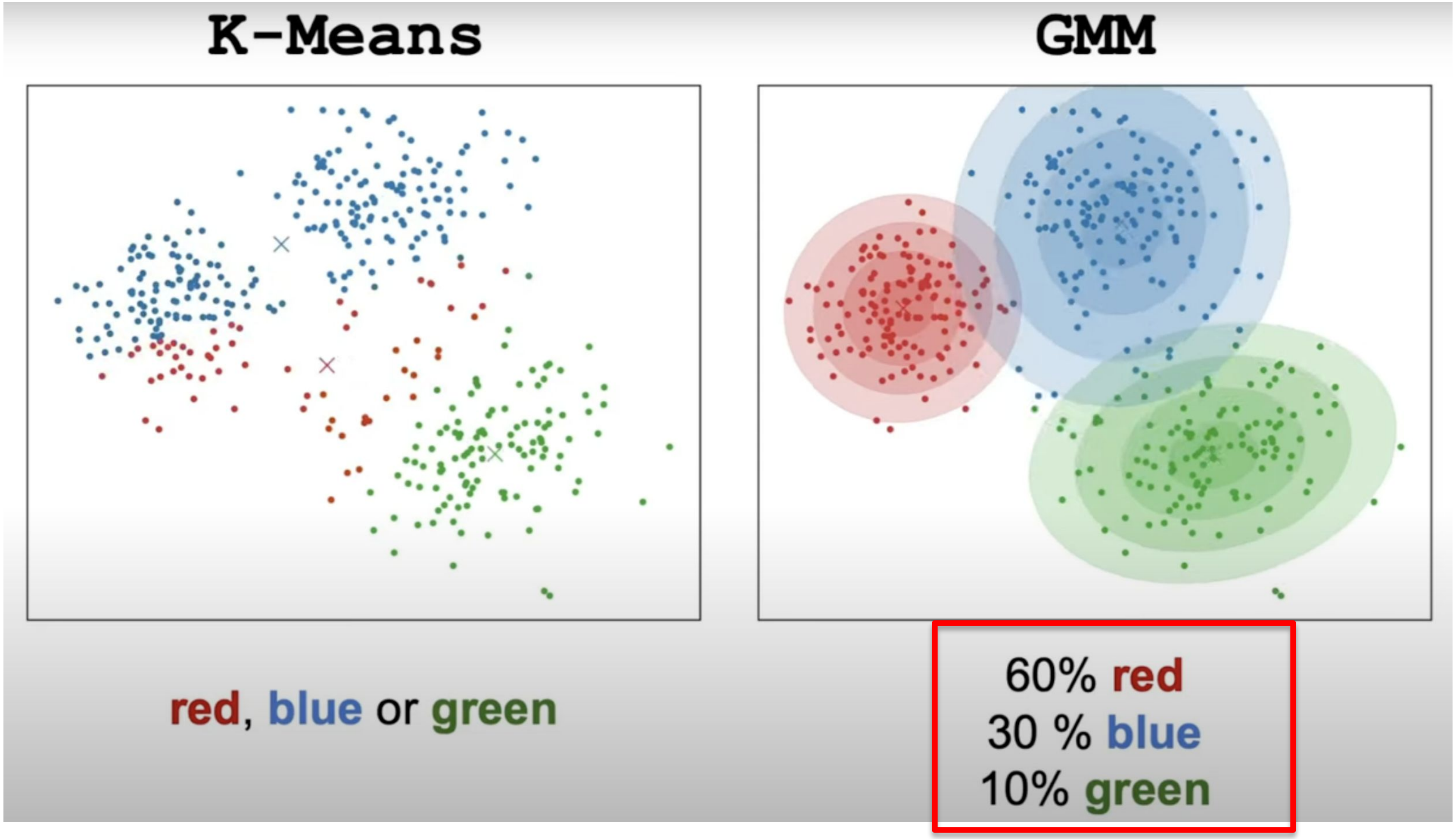
Variables we have so far

Variable	Role
K	Number of clusters / mixture models
μ_k	Mean of Gaussian distribution (k)
Σ_{k}	Variance of Gaussian distribution (k)
δ_k	

Variables we have so far

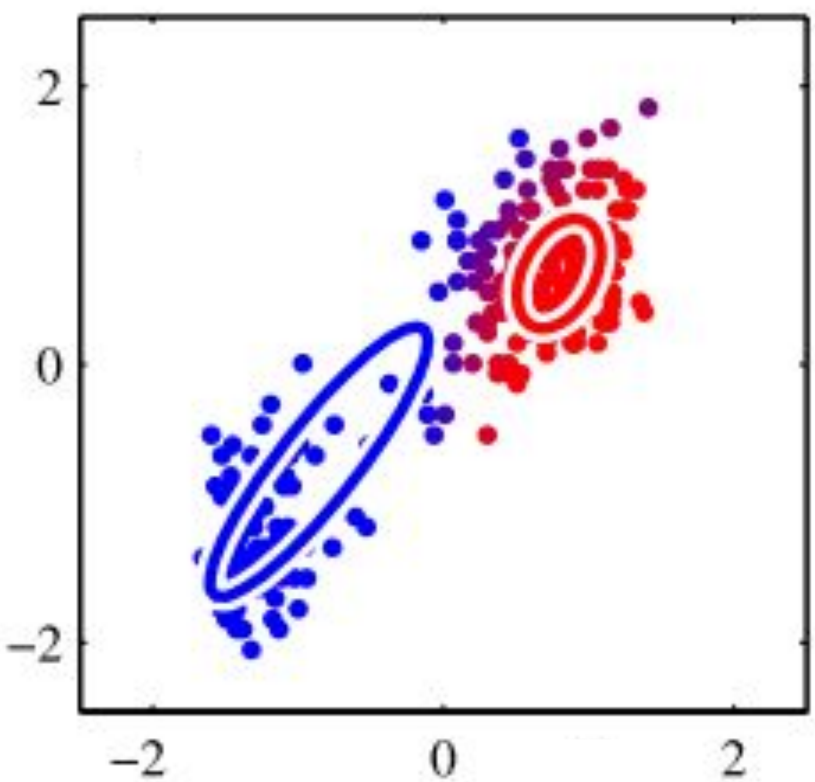
Variable	Role
K	Number of clusters / mixture models
μ_k	Mean of Gaussian distribution (k)
Σ_k	Variance of Gaussian distribution (k)
δ_k	Cluster membership indicator

Mixtures of Gaussian models



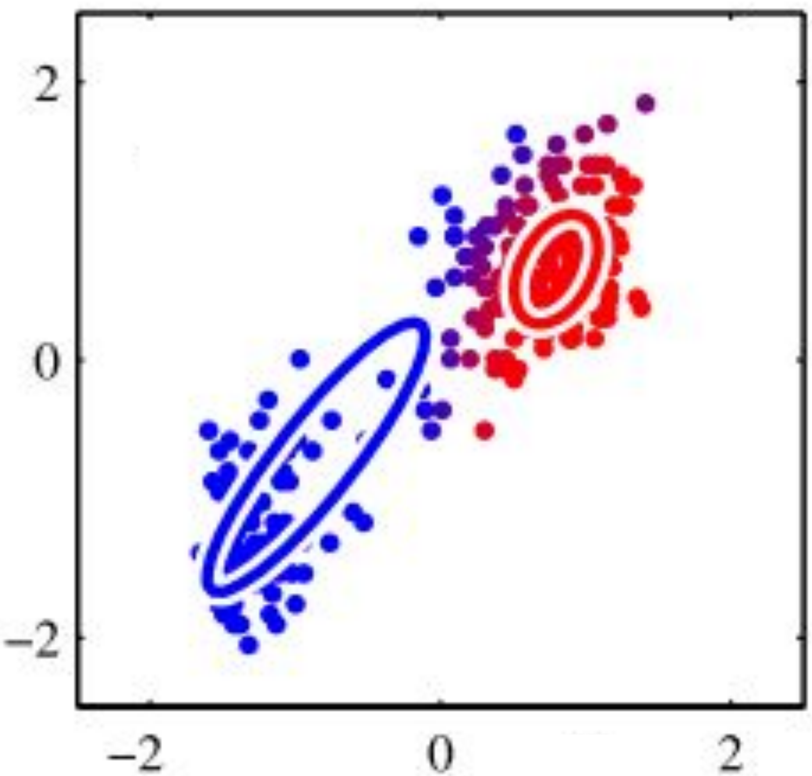
Membership probability π_k

Mixtures of Gaussians:
Data generation example



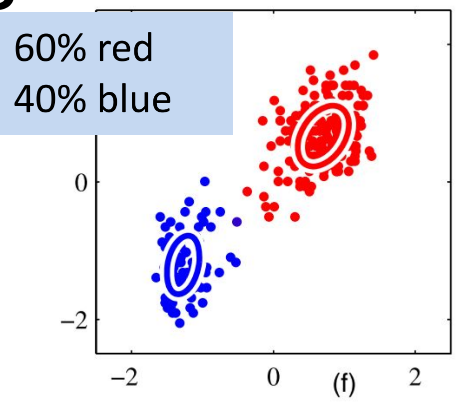
- Suppose $K = 2$ components, k-th component is a Gaussian with parameters μ_k, Σ_k

Mixtures of Gaussians:
Data generation example



- Suppose $K = 2$ components, k-th component is a Gaussian with parameters μ_k, Σ_k
- To sample i-th data point:
  - Pick component δ^i with $\mathrm{p}(\delta_k = 1) = \pi_k$ (parameter)
  - for example, $\pi_k = 0.5$, and we picked $\delta^1 = [0, 1]^T$
  - Pick data point x^i with probability $\mathrm{N}(\mathrm{x}; \mu_k, \Sigma_k)$

Mixtures of Gaussians



sum of

- $\delta_k \in \{0,1\}$ and $\sum_k \delta_k = 1$
- K components, k-th component is a Gaussian with parameters μ_k, Σ_k

- define the joint distribution $p(\mathbf{x}, \delta)$ in terms of a marginal distribution $p(\delta)$ and a conditional distribution $p(\mathbf{x}|\delta)$

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

- where

$$p(\delta_k = 1) = \pi_k \qquad 0 \le \pi_k \le 1 \qquad \sum_{k=1}^{K} \pi_k = 1$$

$$p(x|\delta) = \sum_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)^{\delta_k}$$

Substitute and simplify

Variables we have so far

Variable	Role
K	Number of clusters / mixture models
μ_k	Mean of Gaussian distribution (k)
Σ_k	Variance of Gaussian distribution (k)
δ_k	Cluster membership indicator
$p(\delta)$	Marginal distribution of mixture of Gaussian membership
$p(x)$	Distribution of the Mixture of Gaussians

Maximum Likelihood Solution for Mixture of Gaussians

- This distribution is known as a Mixture of Gaussians

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

- What are the unknowns here?

- We can estimate these parameters via Expectation Maximization (EM)

Today: Clustering

- Agglomerative Clustering
- Divisive Clustering
- K-means
- Vector Quantization with K-Means
- Mixtures of Gaussians
- **Expectation Maximization**

Maximum Likelihood Solution for Mixture of Gaussians

- This distribution is known as a Mixture of Gaussians

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

- We can estimate these parameters via Expectation Maximization (EM)

- **Solution:** Use coordinate descent

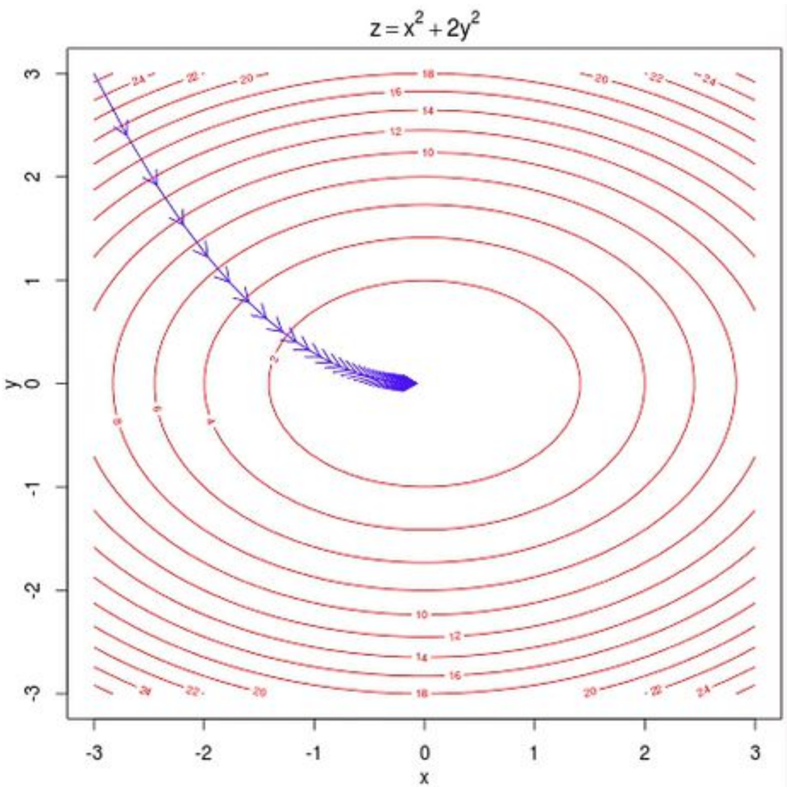
Coordinate Descent

gradient descent:

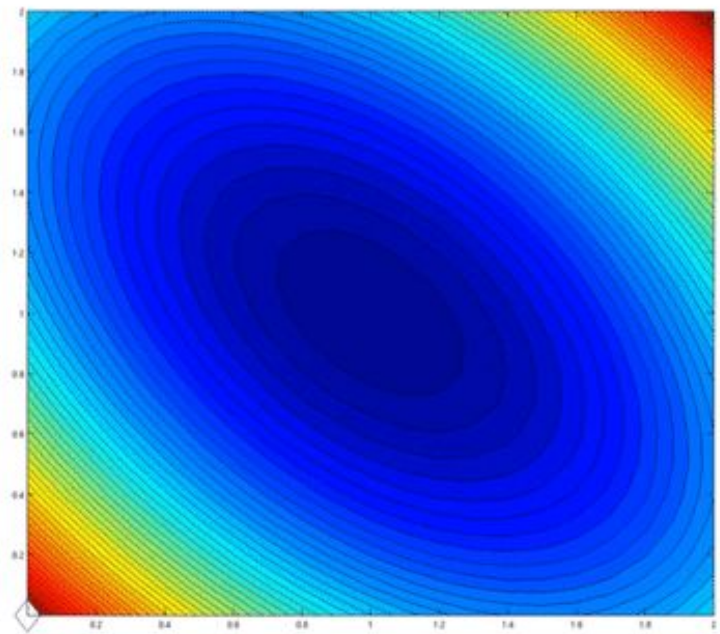
- Minimize w.r.t all parameters at each step

coordinate descent:

- fix some coordinates, minimize w.r.t. the rest
- alternate



Credit: http://vis.supstat.com/2013/03/gradient-descent-algorithm-with-r/



Credit: Martin Takac

Is K-means a type of coordinate descent algorithm?

Is K-means a type of coordinate descent algorithm?