Generative Models

Swati Mishra

Applications of Machine Learning (4AL3)

Fall 2024

McMaster University

ENGINEERING

Review

- Discriminative vs Generative Models
- Naïve Bayes Theorem

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

- Text to Vector Conversion: Bag Of Words
- Naïve Bayes Classification applied to test

Training Naïve Bayes: Review

function TRAIN NAIVE BAYES(D, C) **returns** V, $\log P(c)$, $\log P(w|c)$

for each class $c \in C$ # Calculate $P(c)$ terms
  N_{doc} = number of documents in D
  N_c = number of documents from D in class c
  $\boxed{logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}}$
  $V \leftarrow$ vocabulary of D
  $bigdoc[c] \leftarrow$ **append**(d) **for** d $\in$ D **with** class c
  for each word w in V # Calculate $P(w|c)$ terms
    $count(w,c) \leftarrow$ # of occurrences of w in $bigdoc[c]$
    $loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$
return *logprior*, *loglikelihood*, V

function TEST NAIVE BAYES(*testdoc*, *logprior*, *loglikelihood*, C, V) **returns** best c

for each class $c \in C$
  $sum[c] \leftarrow logprior[c]$
  for each position i in *testdoc*
    $word \leftarrow testdoc[i]$
    if $word \in V$
      $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c\ sum[c]$

Training Naïve Bayes: Review

- Goal is to learn probabilities

	Label	documents
Training	-	just plain boring
Training	-	entirely predictable and lacks energy
Training	-	no surprises and very few laughs
Training	+	very powerful
Training	+	the most fun film of the summer
Test	?	predictable with no fun

$$P(-) = \frac{3}{5}$$

$$P(+) = \frac{2}{5}$$

$$\frac{\text{Number of d in class c}}{\text{Number of documents (d)}}$$

Training Naïve Bayes: Review

function TRAIN NAIVE BAYES(D, C) **returns** V, $\log P(c)$, $\log P(w|c)$

for each class $c \in C$ # Calculate $P(c)$ terms
  N_{doc} = number of documents in D
  N_c = number of documents from D in class c
  $\boxed{logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}}$
  $V \leftarrow$ vocabulary of D
  $bigdoc[c] \leftarrow$ **append**(d) **for** d $\in$ D **with** class c
  for each word w in V # Calculate $P(w|c)$ terms
    $count(w,c) \leftarrow$ # of occurrences of w in $bigdoc[c]$
    $\boxed{loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}}$
return $logprior$, $loglikelihood$, V

function TEST NAIVE BAYES($testdoc$, $logprior$, $loglikelihood$, C, V) **returns** best c

for each class $c \in C$
  $sum[c] \leftarrow logprior[c]$
  for each position i in $testdoc$
    $word \leftarrow testdoc[i]$
    if $word \in V$
      $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c\ sum[c]$

McMaster University

Training Naïve Bayes : Review

- Goal is to learn probabilities:

$P(\text{predictable}|-)$ $\frac{1+1}{14+20}$

$P(\text{predictable}|+)$ $\frac{0+1}{9+20}$

$P(\text{no}|-)$ $\frac{1+1}{14+20}$

$P(\text{no}|+)$ $\frac{0+1}{9+20}$

$P(\text{fun}|-)$ $\frac{0+1}{14+20}$

$P(\text{fun}|+)$ $\frac{1+1}{9+20}$

	Label	documents
Training	-	just plain boring
Training	-	entirely predictable and lacks energy
Training	-	no surprises and very few laughs
Training	+	very powerful
Training	+	the most fun film of the summer
Test	?	predictable with no fun

Using add -1 smoothening $\frac{\text{Count}(f_i, c)+1}{\sum_{f \in V}(\text{Count}(f, c)) + |V|}$

McMaster University

Training Naïve Bayes: Review

```
function TRAIN NAIVE BAYES(D, C) returns V, log P(c), log P(w|c)

for each class c ∈ C              # Calculate P(c) terms
   N_doc = number of documents in D
   N_c = number of documents from D in class c
   logprior[c] ← log N_c / N_doc
   V ← vocabulary of D
   bigdoc[c] ← append(d) for d ∈ D with class c
   for each word w in V                  #  Calculate P(w|c) terms
      count(w,c) ← # of occurrences of w in bigdoc[c]
      loglikelihood[w,c] ← log (count(w,c) + 1) / (Σ_{w' in V} (count(w',c) + 1))
return logprior, loglikelihood, V

function TEST NAIVE BAYES(testdoc, logprior, loglikelihood, C, V) returns best c

for each class c ∈ C
   sum[c] ← logprior[c]
   for each position i in testdoc
      word ← testdoc[i]
      if word ∈ V
         sum[c] ← sum[c]+ loglikelihood[word,c]
return argmax_c sum[c]
```

$P(-)P(S|-)$

$$\frac{3}{5} \times \frac{2 \times 2 \times 1}{34 \times 34 \times 34}$$

$P(+)P(S|+)$

$$\frac{2}{5} \times \frac{1 \times 1 \times 2}{29 \times 29 \times 29}$$

Maximum of the two ?

Converting Text to Vectors: Review

- Techniques used:
  - Bag of Words

word	frequency
It	6
I	5
the	4
satirical	1
whimsical	1
would	1
adventure	1
and	3

"I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!"

Converting Text to Vectors : Review

with respect to document (**not** position within a sentence) : More accurate description is “**relative** frequency of occurrence in a document”

- Techniques used:
  - TF- IDF

word	position
It	6
I	1
the	4
satirical	9
whimsical	1
would	1
adventure	1
and	3

“I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I’ve seen it several times, and I'm always happy to see it again whenever I have a friend who hasn’t seen it yet!”

Converting Text to Vectors : Review

- Techniques used:
  - TF- IDF

with respect to document (**not** position within a sentence) : More accurate description is “**relative** frequency of occurrence in a document”

word	frequency	df
It	6	
I	1	
the	4	
satirical	9	
whimsical	1	
would	1	
adventure	1	
and	3	

the number of documents in which each term can be found

“It manages to be **whimsical** and romantic while laughing at the conventions of the fairy tale genre.”

“I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be **whimsical** .

Generative vs Discriminative Models

Generative models model the problem

$$\mathrm{P}(x, y) = X, Y \rightarrow [0,1]$$

Discriminative models model the problem

$$\mathrm{P}(y|x) = X, Y \rightarrow [0,1]$$

Generative vs Discriminative Models

Generative models model the problem

$$P(x, y) = X, Y \rightarrow [0,1]$$

- They can generate new data instances

Discriminative models model the problem

$$P(y|x) = X, Y \rightarrow [0,1]$$

- Discriminate between different kinds of data instances.

Generative vs Discriminative Models

Generative models model the problem

$$\mathrm{P}(x, y) = X, Y \ \rightarrow [0,1]$$

- They can generate new data instances
- Capture the joint probability $\mathrm{P}(x, y)$.

Discriminative models model the problem

$$\mathrm{P}(y|x) = X, Y \ \rightarrow [0,1]$$

- Discriminate between different kinds of data instances.
- Capture the conditional probability $\mathrm{P}(y|x)$.

Generative vs Discriminative Models

Generative models model the problem

$$\mathrm{P}(x, y) = X, Y \ \rightarrow [0,1]$$

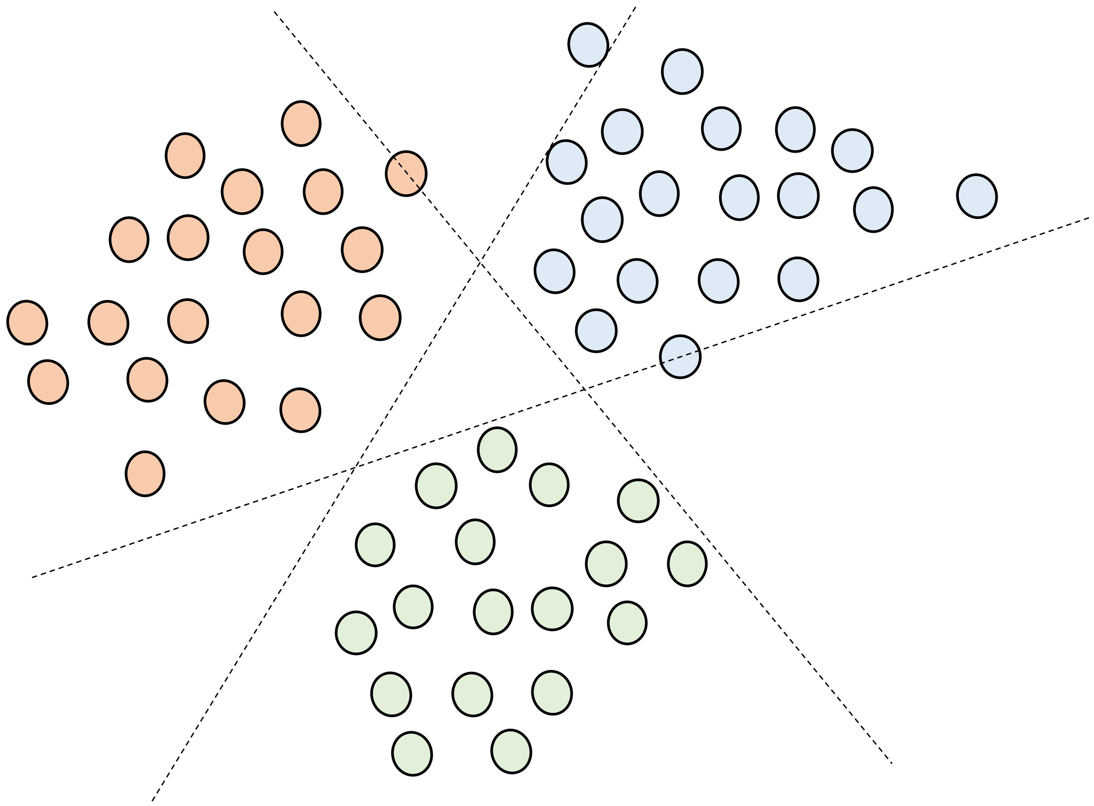
- They can generate new data instances
- Capture the joint probability $\mathrm{P}(x, y)$.
- Can work without labels, so can compute $\mathrm{P}(x)$

Discriminative models model the problem

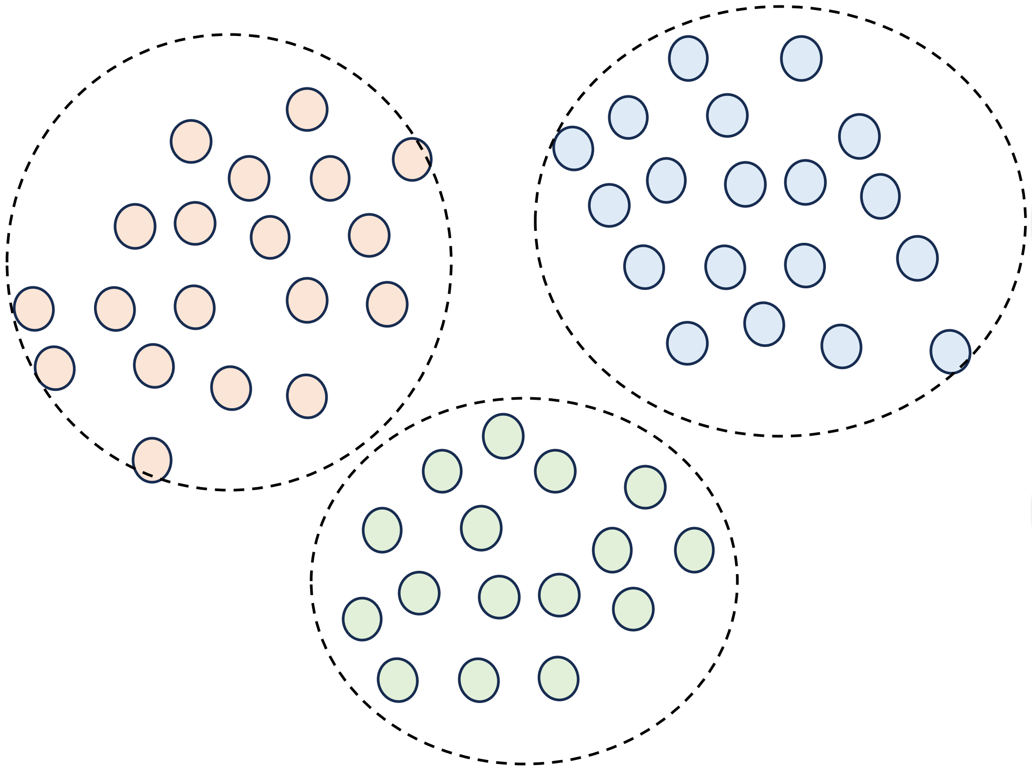
$$\mathrm{P}(y|x) = X, Y \ \rightarrow [0,1]$$

- Discriminate between different kinds of data instances.
- Capture the conditional probability $\mathrm{P}(y|x)$.
- Cannot work without labels.

Clustering



Discriminative Model



Generative Model

Generative vs Discriminative Models

Generative models model the problem

$$P(x, y) = X, Y \rightarrow [0,1]$$

- They can generate new data instances
- Capture the joint probability $P(x, y)$.
- Can work without labels, so can compute $P(x)$

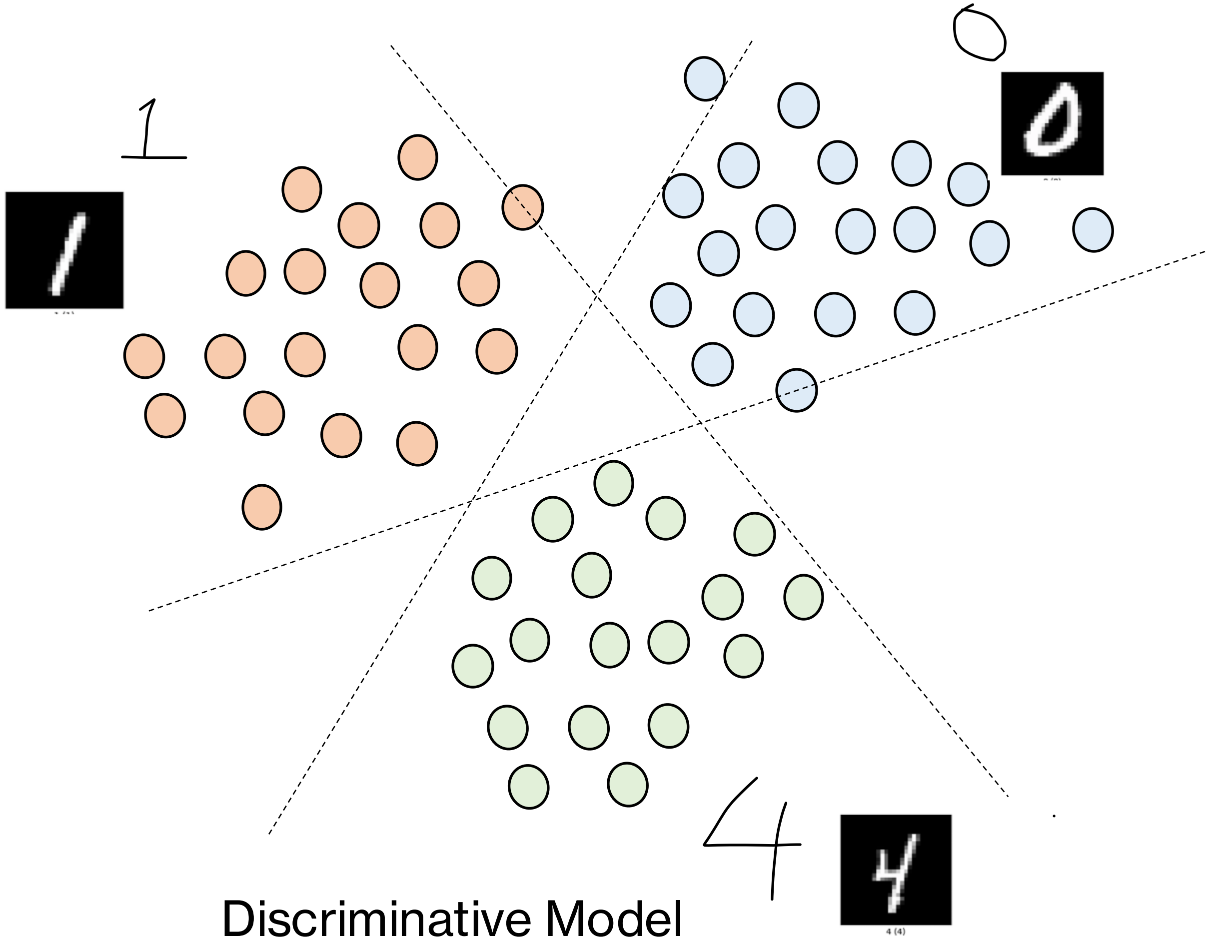
Discriminative models model the problem

$$P(y|x) = X, Y \rightarrow [0,1]$$

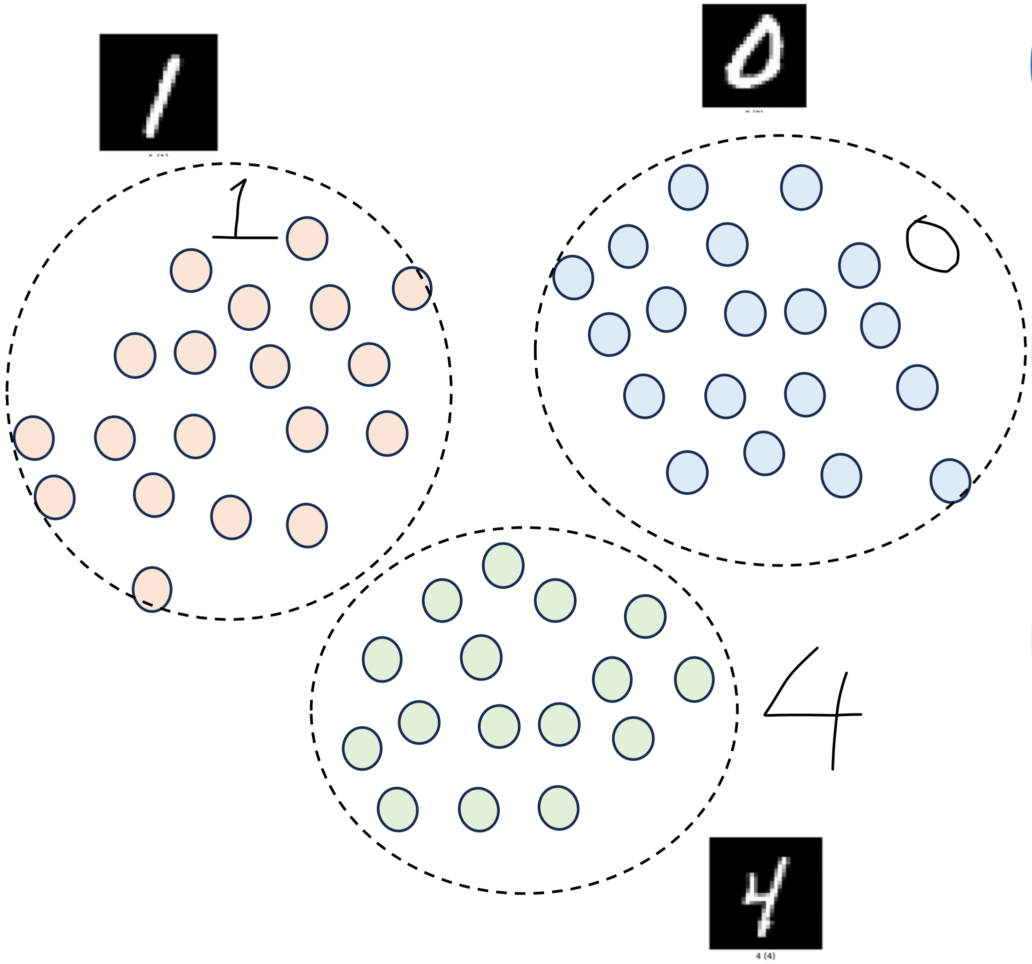
- Discriminate between different kinds of data instances.
- Capture the conditional probability $P(y|x)$.
- Cannot work without labels.

What other generative approaches exist?

Clustering

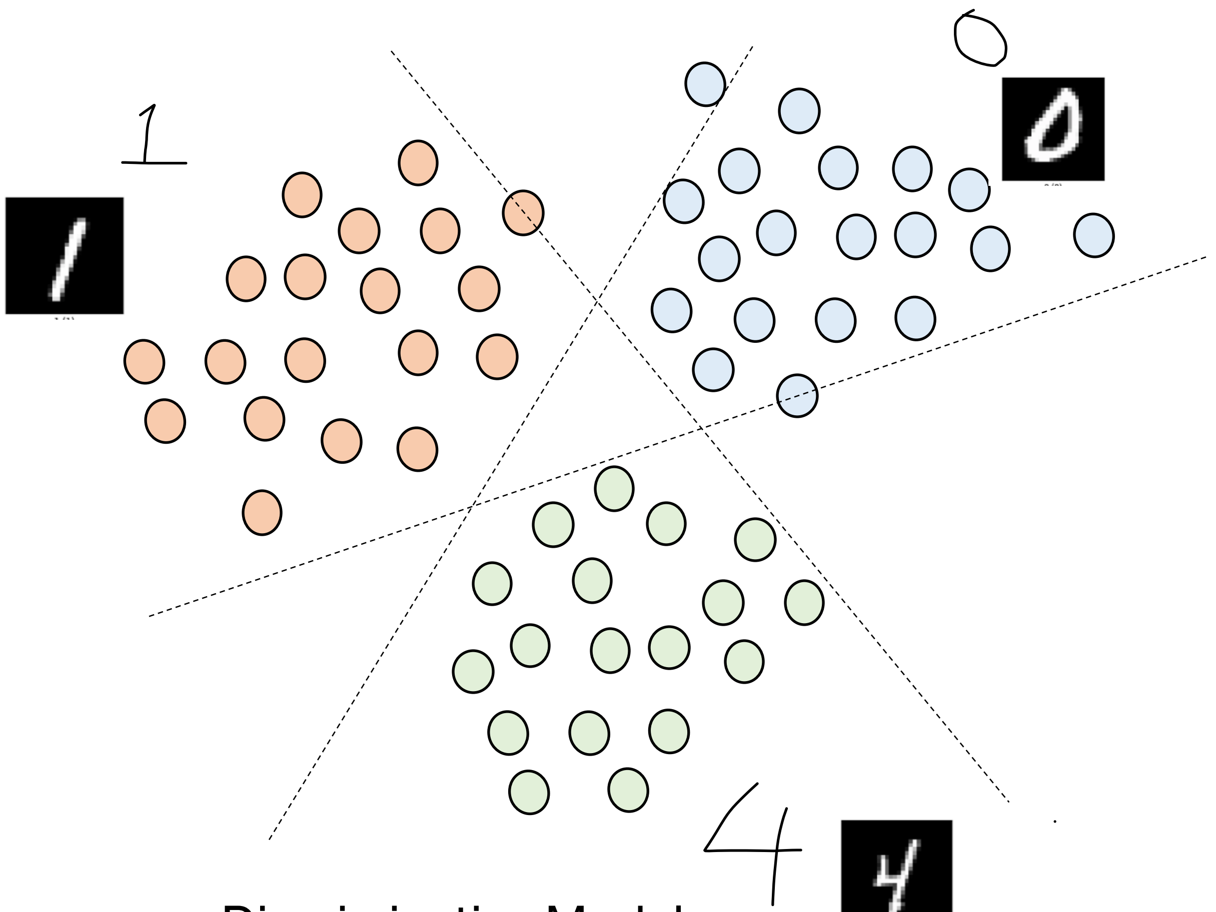


Discriminative Model

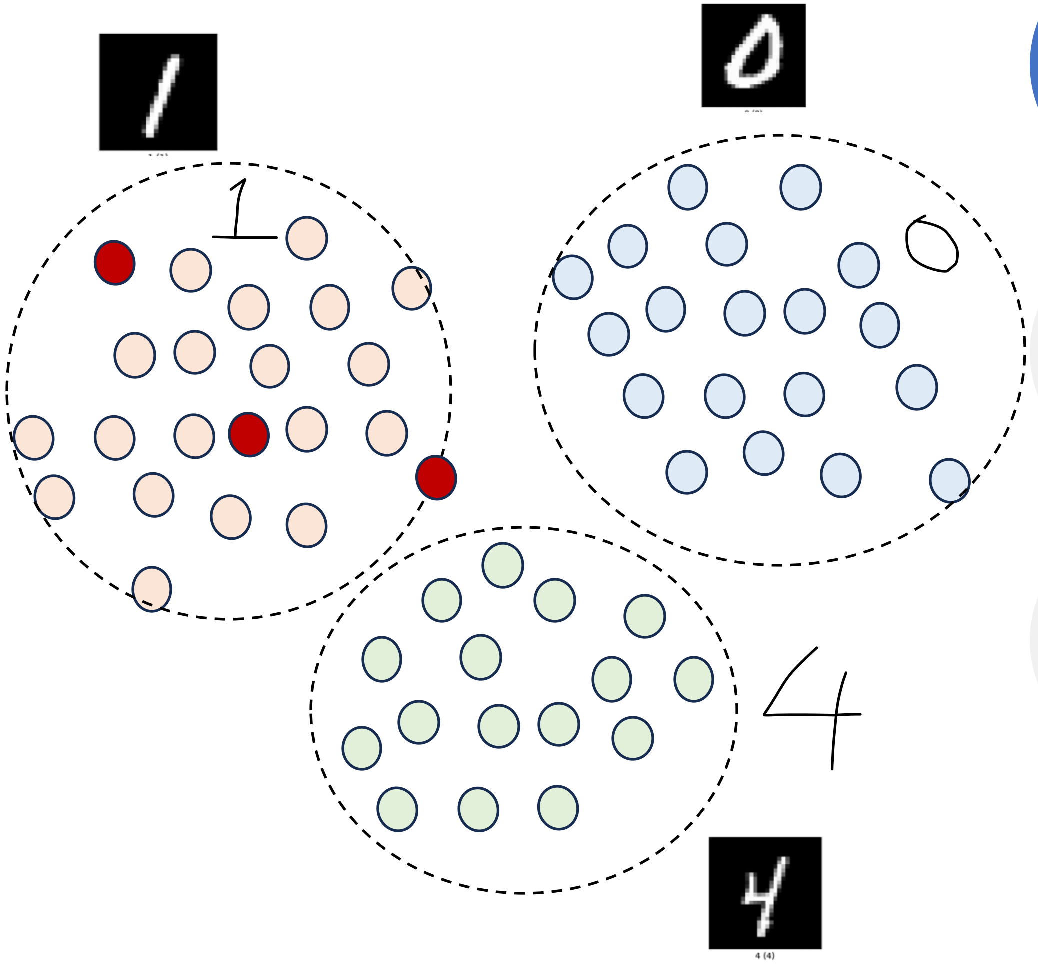


Generative Model

Clustering



Discriminative Model



Generative Model

Clustering

- We may cluster the data into subgroups and partition the entire data distribution based on similarity.
- The goal of clustering algorithms is to find homogenous subgroups among the observations.
- When a new data instance arrives, then we find the subgroup it might belong to, or is closest to.
- Relies on data distribution to build a classification model.
- They are most challenging, and tackle more difficult tasks

Clustering

- We may cluster the data into subgroups and partition the entire data distribution based on similarity.
- The goal of clustering algorithms is to find homogenous subgroups among the observations.
- When a new data instance arrives, then we find the subgroup it might belong to, or is closest to.
- Relies on data distribution to build a classification model.
- They are most challenging, and tackle more difficult tasks

Clustering

- We may cluster the data into subgroups and partition the entire data distribution based on similarity.
- The goal of clustering algorithms is to find homogenous subgroups among the observations.
- When a new data instance arrives, then we find the subgroup it might belong to, or is closest to.
- Relies on data distribution to build a classification model.
- They are most challenging, and tackle more difficult tasks

Clustering

- We may cluster the data into subgroups and partition the entire data distribution based on similarity.
- The goal of clustering algorithms is to find homogenous subgroups among the observations.
- When a new data instance arrives, then we find the subgroup it might belong to, or is closest to.
- Relies on data distribution to build a classification model.
- They are most challenging, and tackle more difficult tasks

Clustering

- We may cluster the data into subgroups and partition the entire data distribution based on similarity.
- The goal of clustering algorithms is to find homogenous subgroups among the observations.
- When a new data instance arrives, then we find the subgroup it might belong to, or is closest to.
- Relies on data distribution to build a classification model.
- They are most challenging, and tackle more difficult tasks.

Clustering

- Types of Clustering:
  - K-Means

Clustering

- Types of Clustering:
  - K-Means
    - Let $C_1, C_2, \ldots, C_k$ be sets containing observations in each cluster. Then these sets must satisfy two properties.

Clustering

- Types of Clustering:
  - K-Means
    - Let $C_1, C_2, \ldots, C_k$ be sets containing observations in each cluster. Then these sets must satisfy two properties.
      - Each observation must belong to at least one of the K clusters. If there are n observations

$$C_1 \cup C_2 \cup \cdots \cup C_k = \{1,2,\ldots,n\}$$

Clustering

- Types of Clustering:
  - K-Means
    - Let $C_1, C_2, \dots, C_k$ be sets containing observations in each cluster. Then these sets must satisfy two properties.
      - Each observation must belong to at least one of the K clusters. If there are n observations
        $$C_1 \cup C_2 \cup \cdots \cup C_k = \{1,2,\dots,n\}$$
      - The clusters are non overlapping, no observation belongs to more than one cluster
        $$C_k \cap C_{k'} = \emptyset$$

Clustering

- Types of Clustering:
  - K-Means
    - Let $C_1, C_2, \dots, C_k$ be sets containing observations in each cluster. Then these sets must satisfy two properties.
      - Each observation must belong to at least one of the K clusters. If there are n observations
        $$C_1 \cup C_2 \cup \cdots \cup C_k = \{1,2,\dots,n\}$$
      - The clusters are non overlapping, no observation belongs to more than one cluster
        $$C_k \cap C_{k'} = \emptyset$$
    - Let us assume that if i^{th} observation is in the k^{th} cluster, then $i \in C_k$

Clustering

- Types of Clustering:
  - K-Means
    - Let $C_1, C_2, \ldots, C_k$ be sets containing observations in each cluster. Then these sets must satisfy two properties.
      - Each observation must belong to at least one of the K clusters. If there are n observations
        $$C_1 \cup C_2 \cup \cdots \cup C_k = \{1,2,\ldots,n\}$$
      - The clusters are non overlapping, no observation belongs to more than one cluster
        $$C_k \cap C_{k'} = \emptyset$$
    - Let us assume that if i^{th} observation is in the k^{th} cluster, then $i \in C_k$
    - Let us assume that within a cluster the variation (how much different each of the samples are from each other) is defined by $W(C_k)$

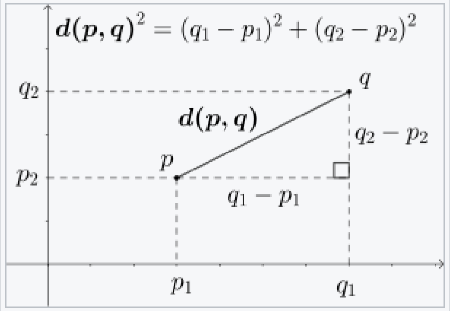
Clustering

- Types of Clustering:
  - K-Means
    - Good clustering is one for which the within-cluster variation is as small as possible.
      - Optimization Problem: $\underset{C_1, C_2, \ldots, C_k}{\textit{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Clustering

- Types of Clustering:
  - K-Means
    - Good clustering is one for which the within-cluster variation is as small as possible.
      - Optimization Problem: $\underset{C_1, C_2, \ldots, C_k}{\textit{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$
    - Defining similarity:
      - Feature similarity using Euclidean distance



Using the Pythagorean theorem to compute two-dimensional Euclidean distance

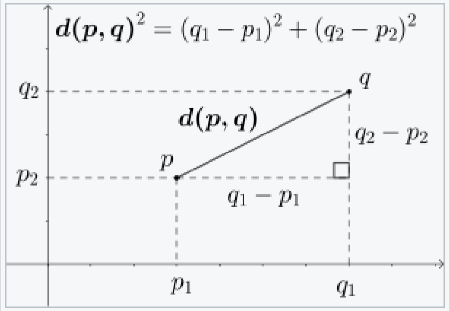
Distance between two points $d(p, q) = |p, q|$

Picture Source: Wikipedia

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Clustering

- Types of Clustering:
  - K-Means
    - Good clustering is one for which the within-cluster variation is as small as possible.
      - Optimization Problem: $\underset{C_1, C_2, \ldots, C_k}{\textit{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$
    - Defining similarity:
      - Feature similarity using Euclidean distance



Using the Pythagorean theorem to compute two-dimensional Euclidean distance

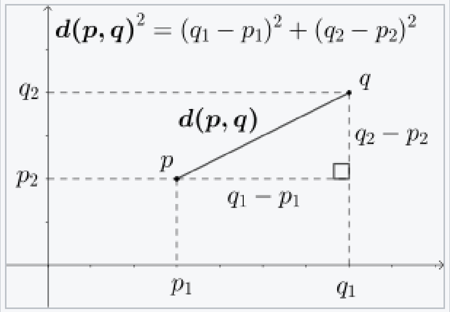
Distance between two points $d(p,q) = |p,q|$

In high dimensionality : $d(p,q) = \sqrt{(p-q)^2}$

Picture Source: Wikipedia

Clustering

- Types of Clustering:
  - K-Means
    - Good clustering is one for which the within-cluster variation is as small as possible.
      - Optimization Problem: $\underset{C_1, C_2, \ldots, C_k}{\textit{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$
    - Defining similarity:
      - Feature similarity using Euclidean distance



Using the Pythagorean theorem to compute two-dimensional Euclidean distance

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Where $|C_k|$ = denotes the number of observations in k^{th} cluster

Picture Source: Wikipedia

Clustering

- Types of Clustering:
  - K-Means
    - Good clustering is one for which the within-cluster variation is as small as possible.
      - Optimization Problem:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Where $|C_k|$ = denotes the number of observations in k^{th} cluster

Picture Source: Wikipedia

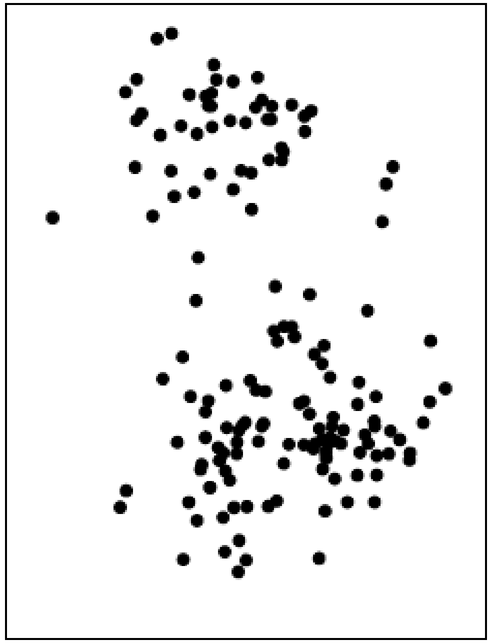
Clustering : Training

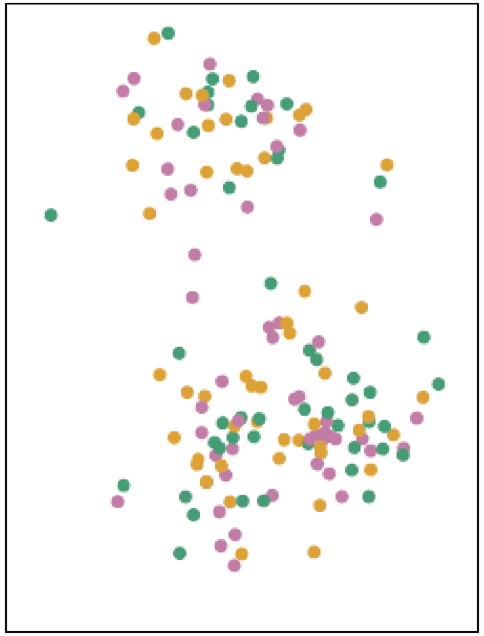
- Types of Clustering:
  - K-Means algorithm:

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

Clustering : Training

- Types of Clustering:
  - K-Means algorithm:



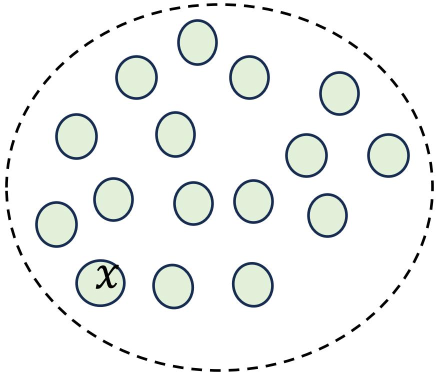


Clustering : Training

- Types of Clustering:
  - K-Means algorithm:

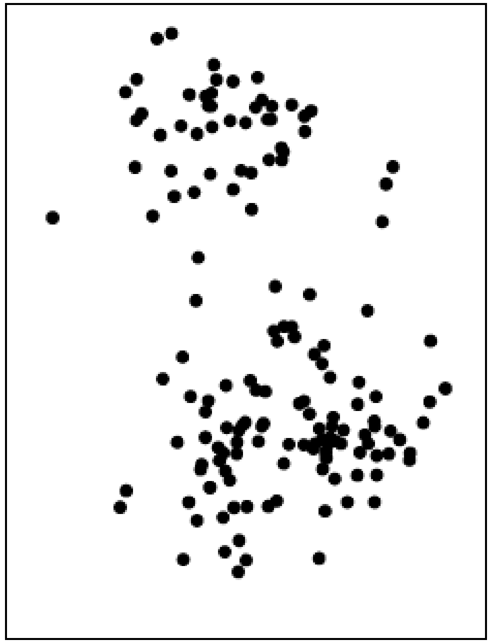
1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

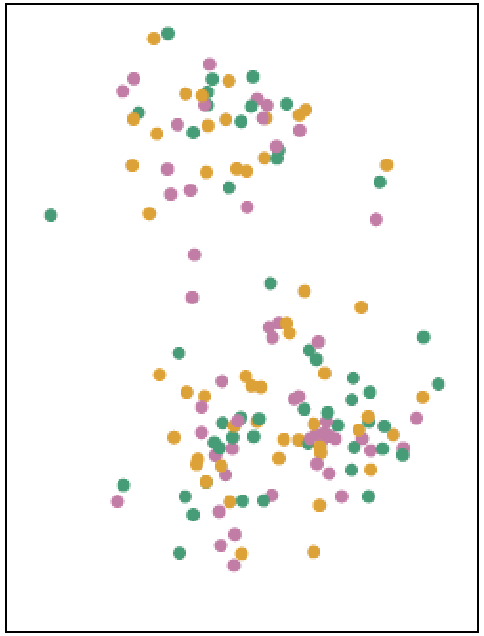
$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$$

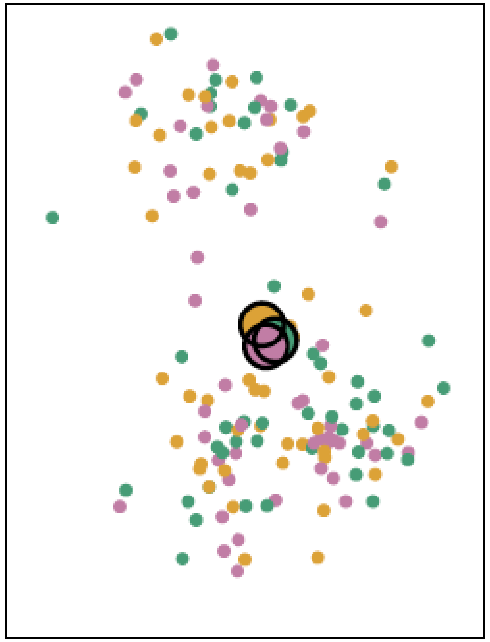


Clustering : Training

- Types of Clustering:
  - K-Means algorithm:







Clustering : Training

- Types of Clustering:
  - K-Means algorithm calculating centroid

Observation	X_1	X_2
A	7	9
B	3	3
C	4	1
D	3	8

Centroid

Cluster	X_1	X_2
A,B	$(7+3)/2 = 5$	$(9+3)/2 = 6$
C,D	$(4+3)/2 = 3.5$	$(1+8)/2 = 4.5$

McMaster University

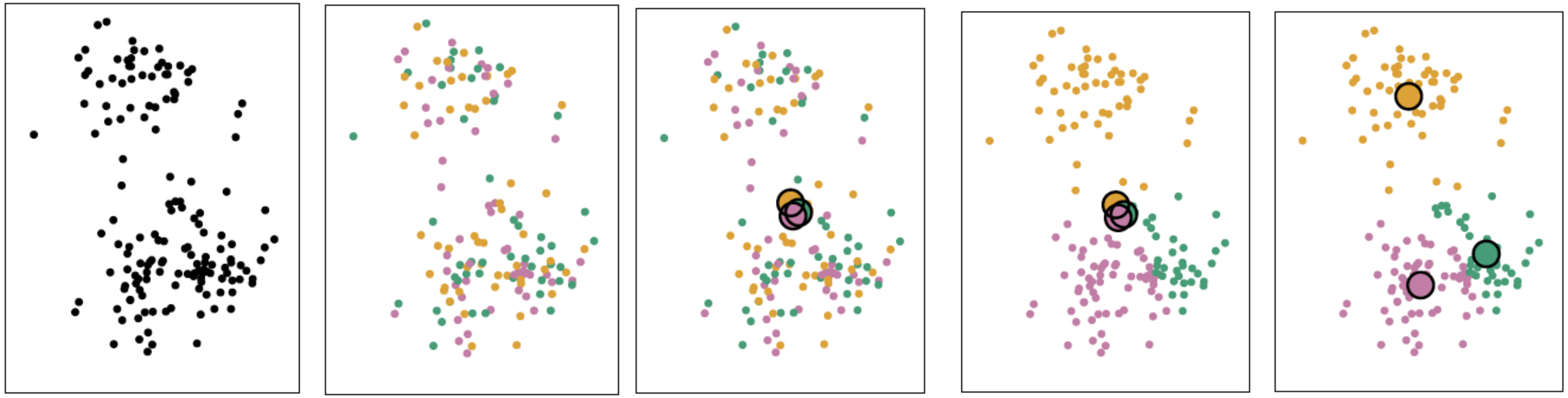
Clustering : Training

- Types of Clustering:
  - K-Means algorithm:

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Clustering : Training

- Types of Clustering:
  - K-Means algorithm:



Clustering : Training

- Types of Clustering:
  - K-Means algorithm:

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

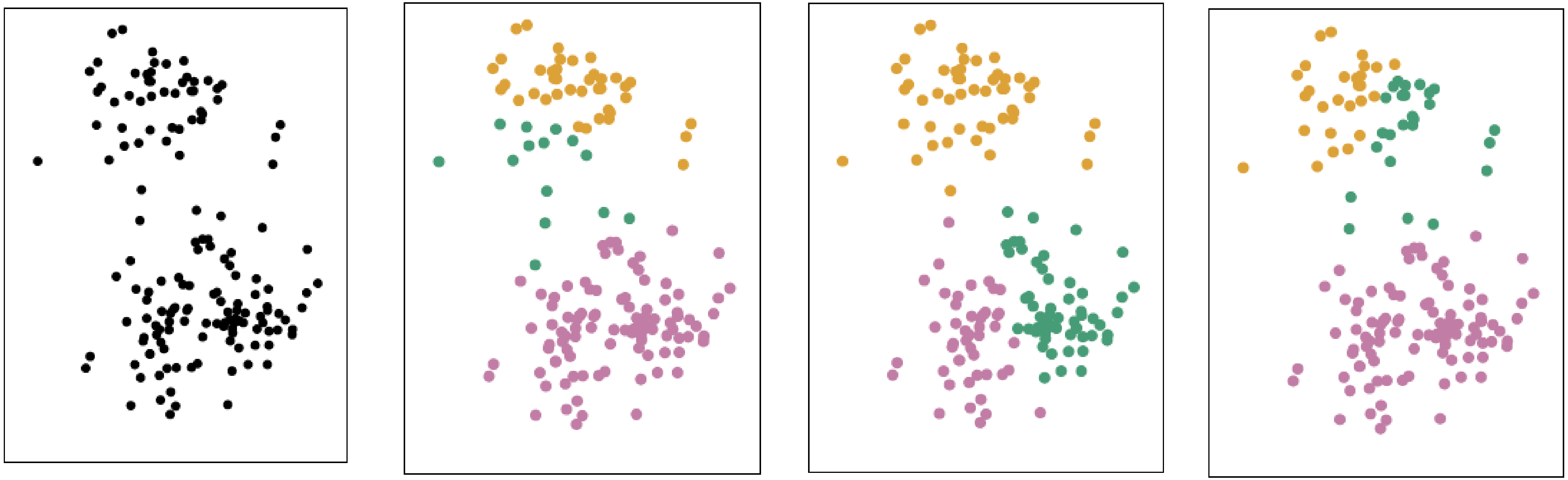
2. Iterate until the cluster assignments stop changing:

   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Clustering : Evaluation

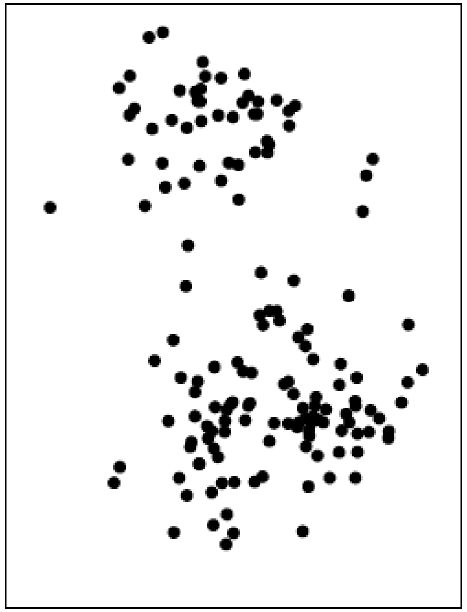
- Types of Clustering:
  - K-Means algorithm: Initial assignment matters



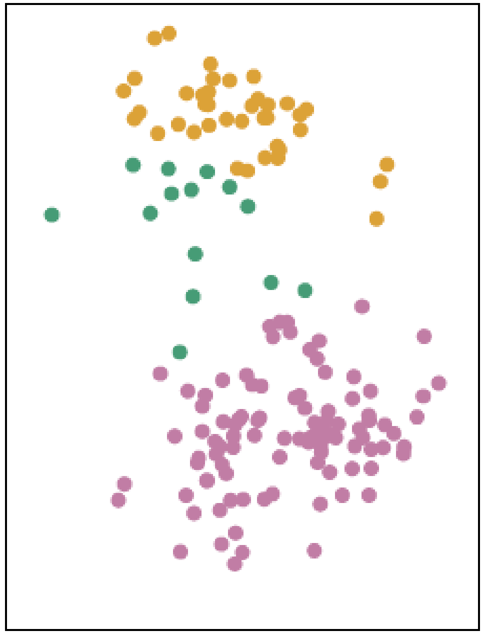
Clustering : Evaluation

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

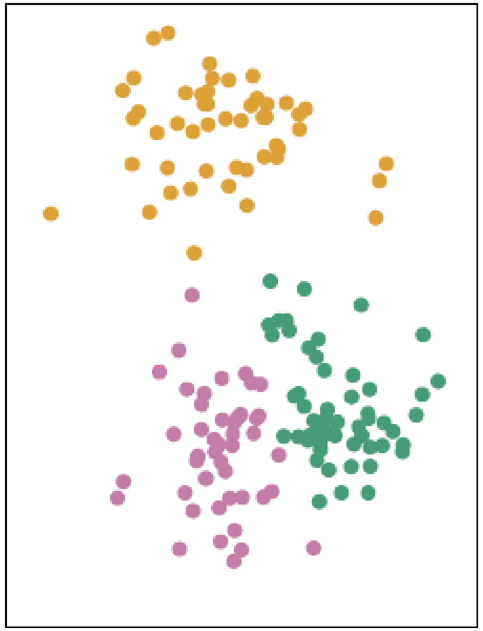
- Types of Clustering:
  - K-Means algorithm: Initial assignment matters



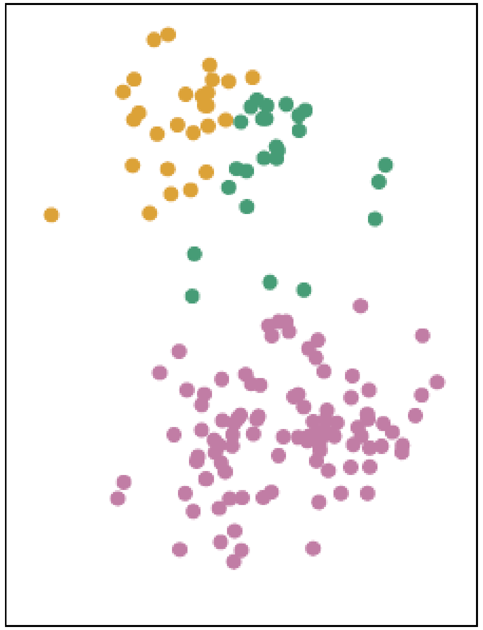
320.9



235.8



310.9



Clustering

- Types of Clustering:
  - K-Means

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Readings

Required Readings:

Introduction to Statistical Learning

- Chapter 4 – Section 4.4 page 158 – 164
- Chapter 12 – Section 12.4 page 521 - 525

Thank You

McMaster University