Machine Learning (19CSE305)

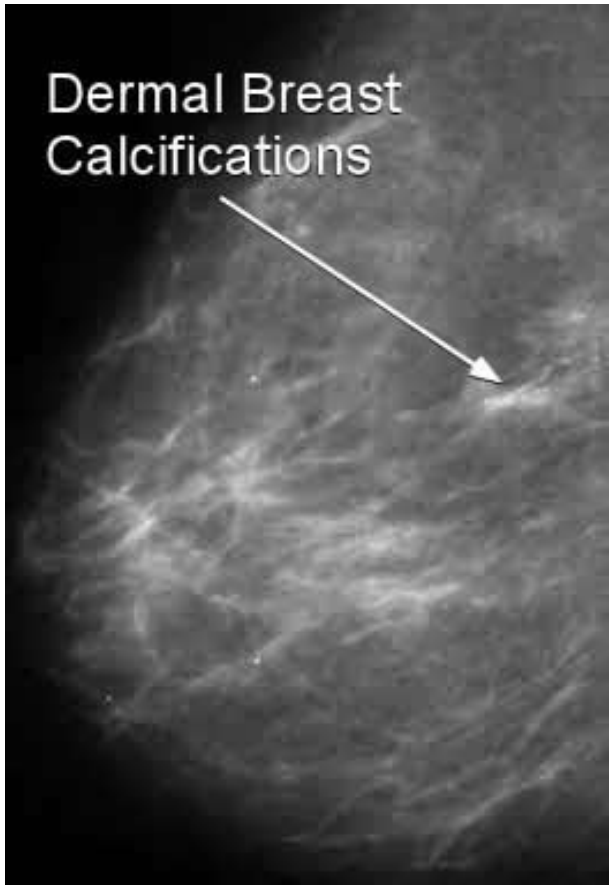
Confusion Matrix, ROC, F-Score

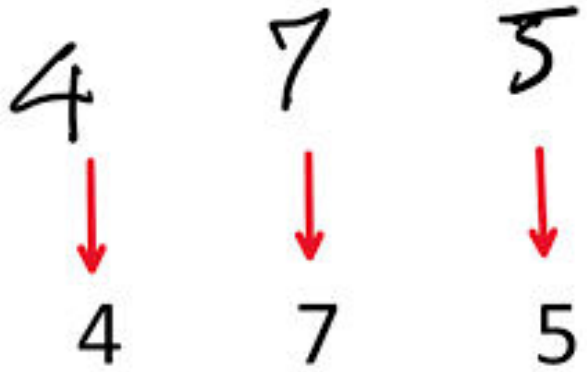
Dr. Peeta Basa Pati
Ms. Priyanka V
Department of Computer Science & Engineering,
Amrita School of Engineering, Bengaluru

Topics

- Confusion Matrix
- Scores to measure efficiency
- ROC & AUC

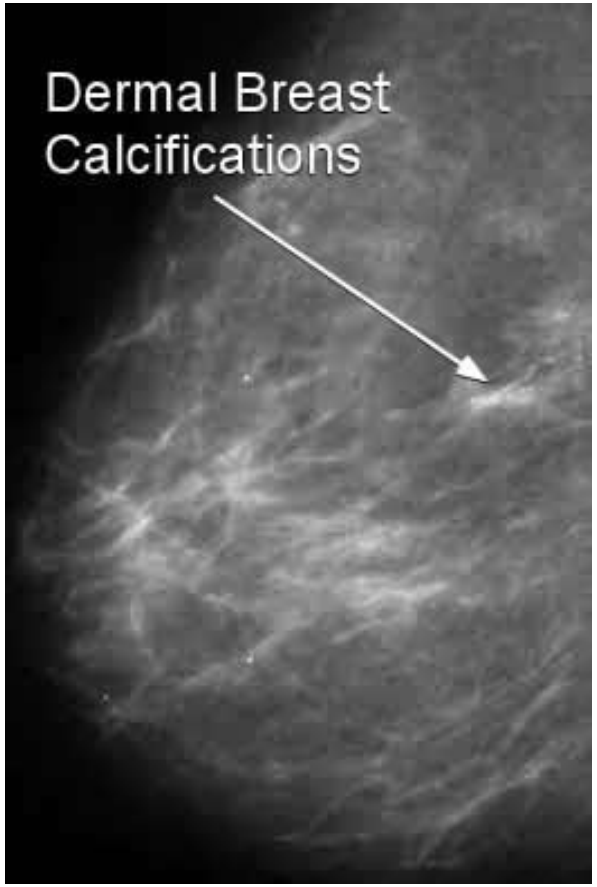
Example Cases





Source: Internet

Mammogram Analysis



- Calcification Present → +ve class
- Calcification Absent → -ve class

One in twenty-eight Indian women is likely to develop breast cancer during her lifetime. -Statistics of Breast Cancer In **India** | **Cytecare Hospitals**

Scenario	Calcification - Actual	Calcification - Detection
1	Yes	Yes
2	Yes	No
3	No	Yes
4	No	No

		DETECTION	
		YES	NO
PRESENCE	YES	TRUE POSITIVE	FALSE NEGATIVE
	NO	FALSE POSITIVE	TRUE NEGATIVE

Are the costs same?

Other Scenarios

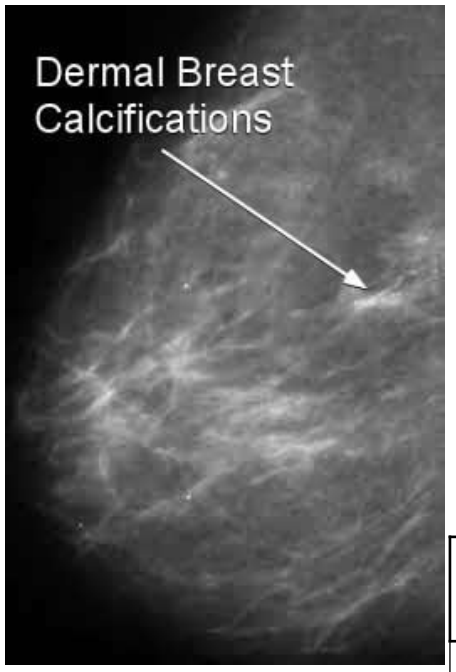
COVID Infection

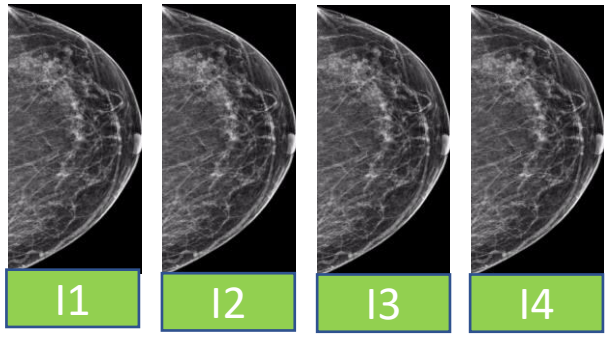
		DETECTION	
		YES	NO
PRESENCE	YES	TRUE POSITIVE	FALSE NEGATIVE
	NO	FALSE POSITIVE	TRUE NEGATIVE

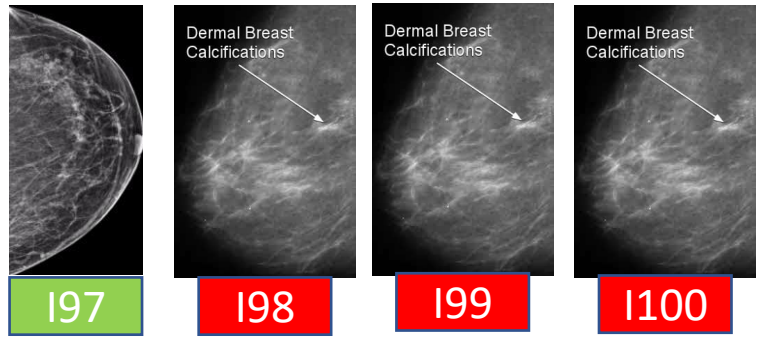
Spam Mail

		DETECTION	
		YES	NO
PRESENCE	YES	TRUE POSITIVE	FALSE NEGATIVE
	NO	FALSE POSITIVE	TRUE NEGATIVE

Concept of Accuracy







97 images are from healthy women

3 images are with tumor

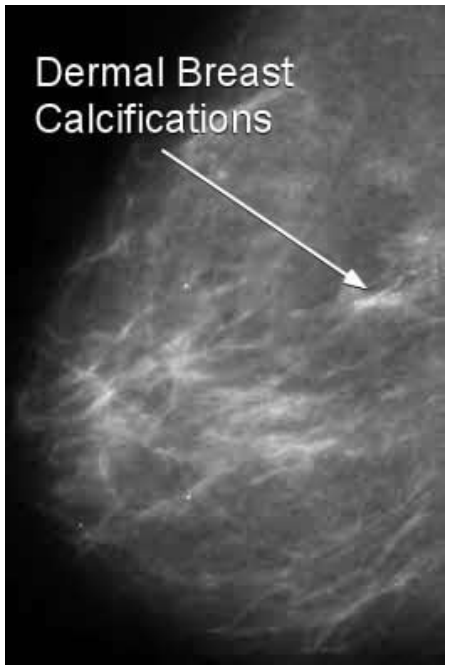
		DETECTION	
		YES	NO
PRESENCE	YES	TRUE POSITIVE (TP)	FALSE NEGATIVE (FN)
	NO	FALSE POSITIVE (FP)	TRUE NEGATIVE (TN)

- Declare all healthy → 97% accurate
- Declare all tumorous → 3% accurate

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

This notion of accuracy is fallacious since the costs associated with the decision are not same.

Precision & Recall



PRESENCE	DETECTION: YES	DETECTION: NO
YES	27 (TP)	5 (FN)
NO	18 (FP)	950 (TN)

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{27}{27+18} = 60\%$$

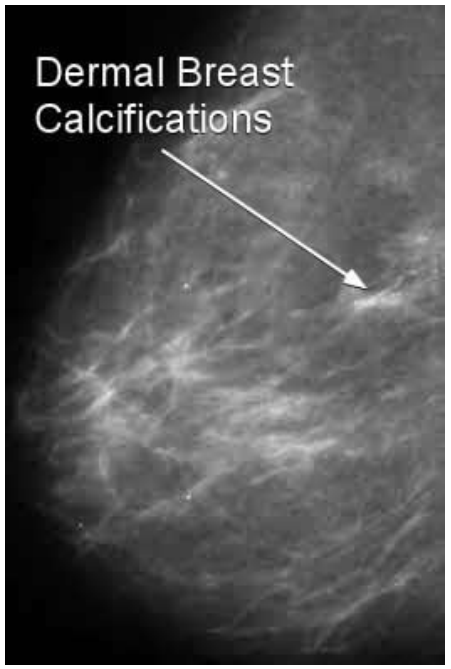
$$\text{Recall} = \frac{TP}{TP+FN} = \frac{27}{27+5} = 84\%$$

(Sensitivity)

$$\text{Specificity} = \frac{TN}{TN+FP} = \frac{950}{968} = 98\%$$

How do we measure these with a single number?

Precision & Recall



Precision = 60%

PRESENCE	DETECTION: YES	DETECTION: NO
YES	27 (TP)	5 (FN)
NO	18 (FP)	950 (TN)

Recall = 84%

How do we measure these with a single number?

We can take Average of the numbers ➔

Avg = (Precision + Recall) / 2 = 72%

Harmonic Mean = $\frac{2XY}{X+Y}$

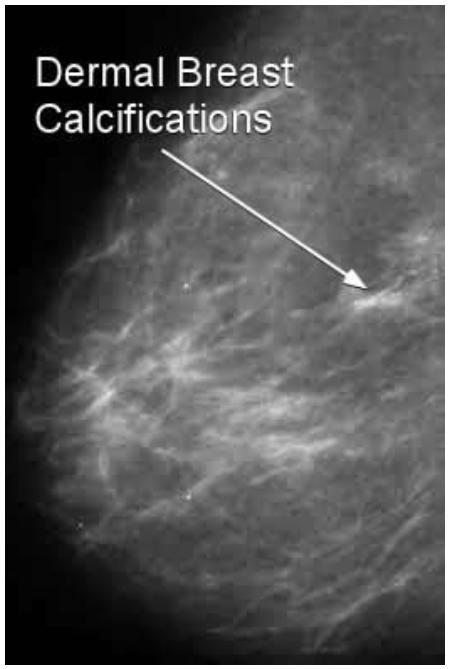
$= \frac{2*Precision*Recall}{Precision+Recall}$

$= 70\%$

Also called as F1-Score

$$F_\beta = (1+\beta^2)\frac{Precision * Recall}{(\beta^2 * Precision) + Recall}$$

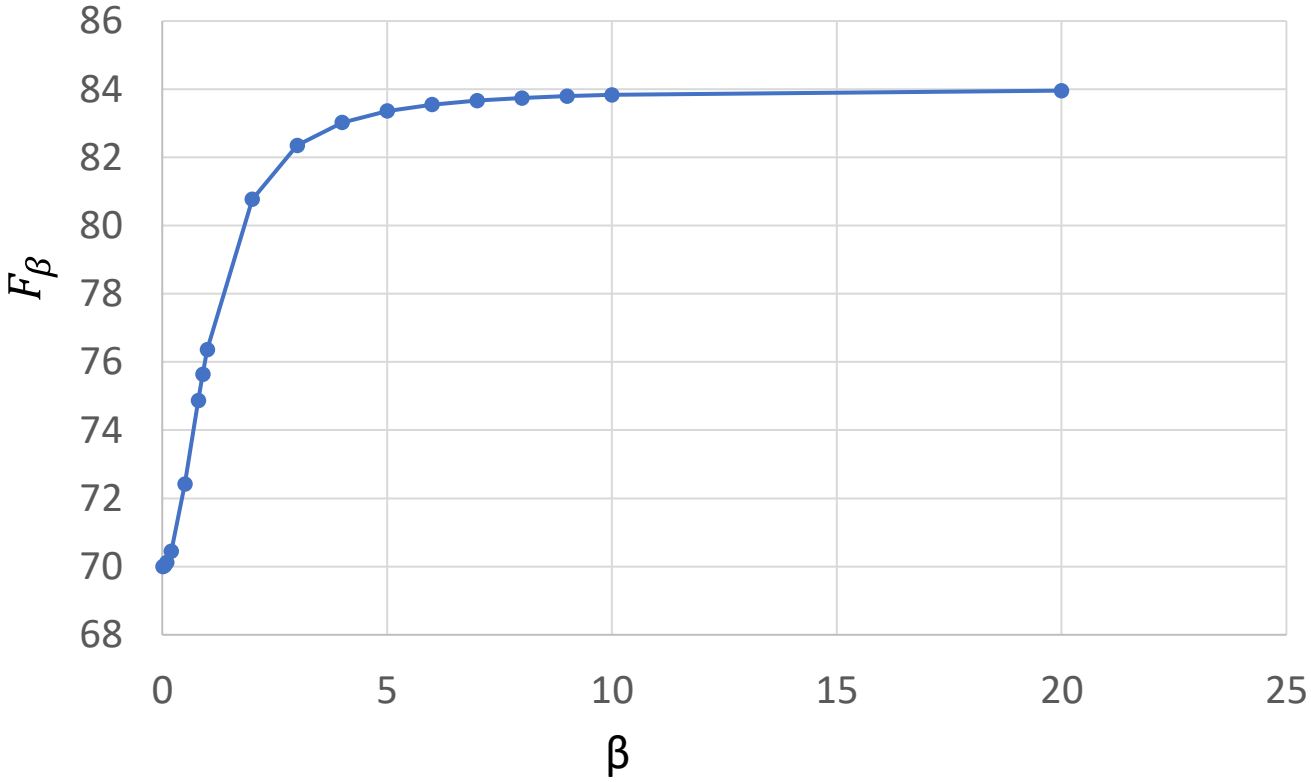
Precision & Recall



Precision = 60%

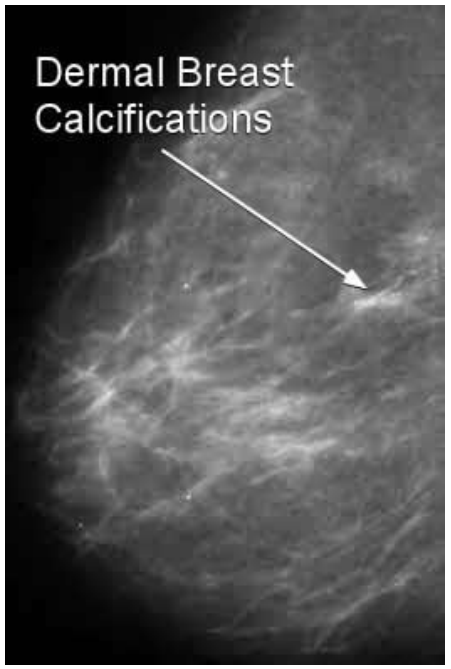
PRESENCE	DETECTION: YES	DETECTION: NO
YES	27 (TP)	5 (FN)
NO	18 (FP)	950 (TN)

Recall = 84%



$$F_\beta = (1+\beta^2)\frac{Precision * Recall}{(\beta^2 * Precision) + Recall}$$

Error Types



		DETECTION	
		YES	NO
PRESENCE	YES	27 (TP)	5 (FN)
	NO	18 (FP)	950 (TN)

Type-I Error:

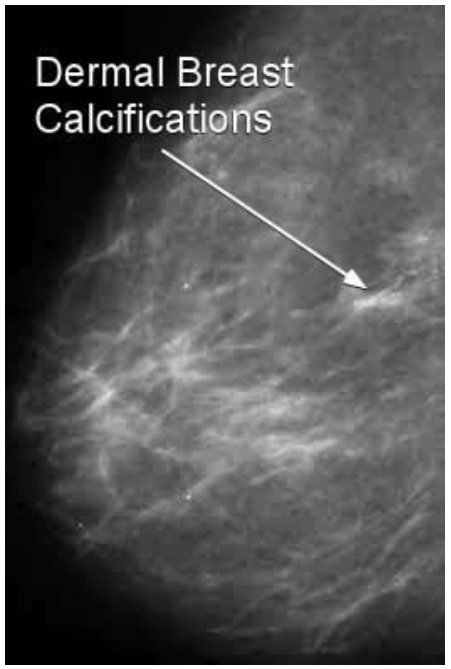
- Refers to error made in judgement leading to positive detection
- False Positives

Type-II Error:

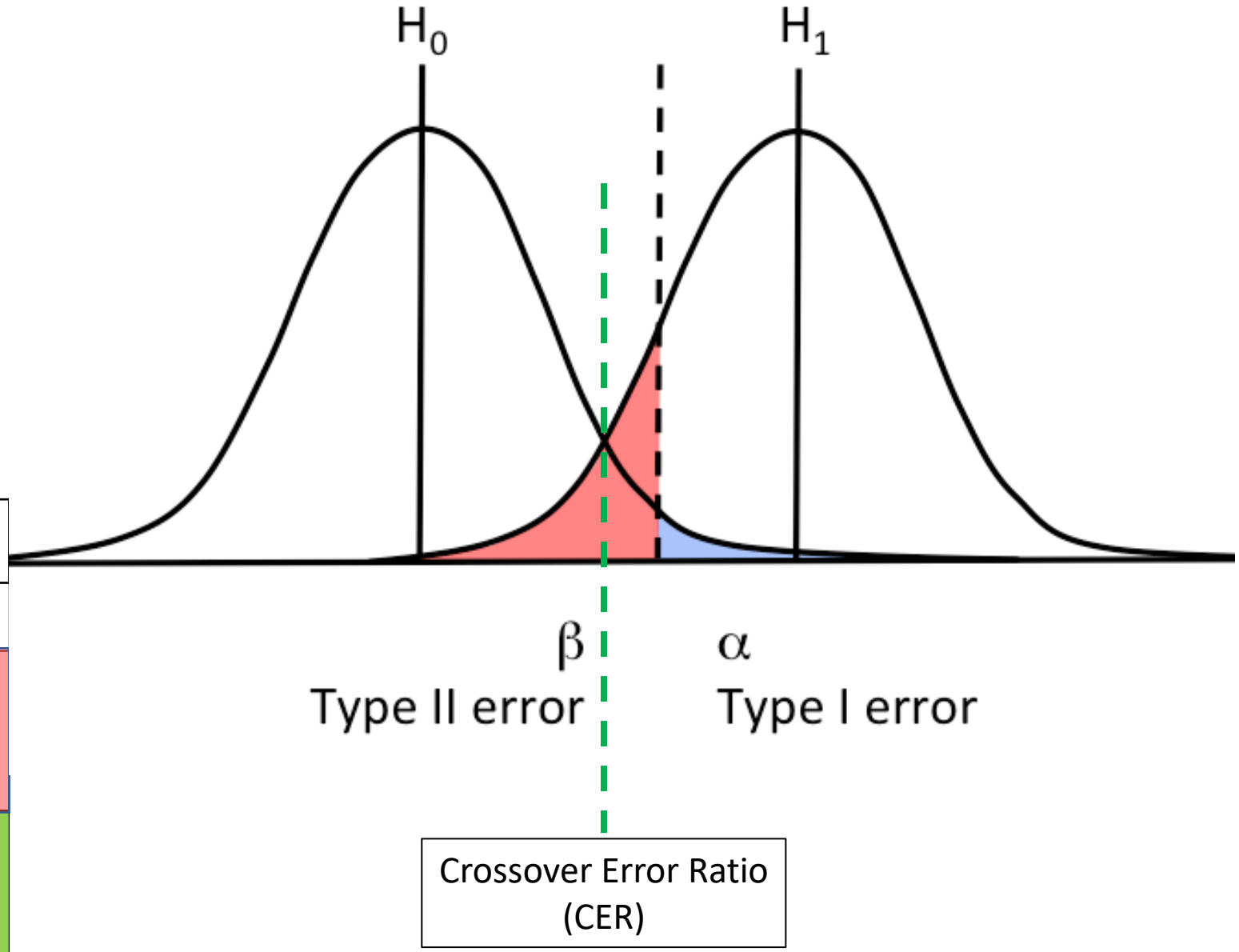
- Refers to error made in judgement leading to negative detection
- False Negatives

Error due to bias & error due to over-fitting

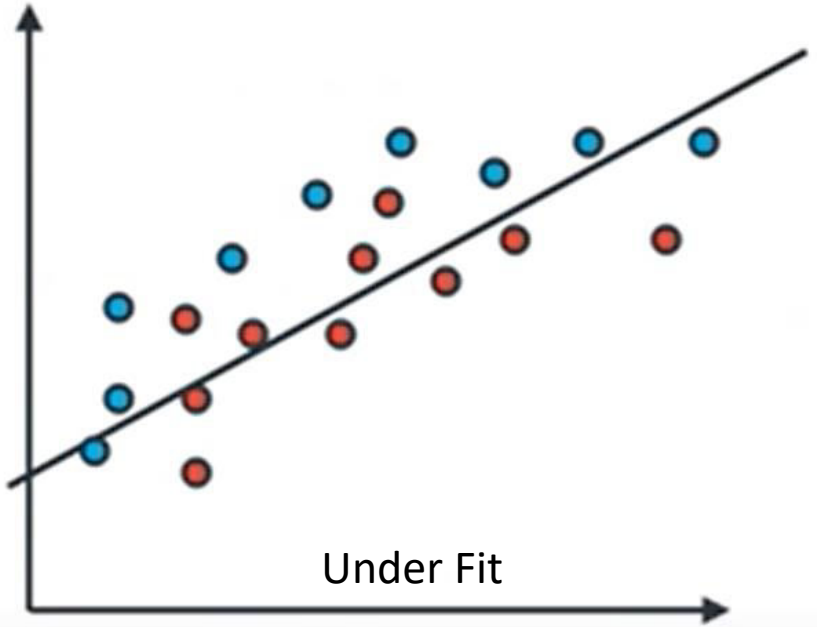
Error Types



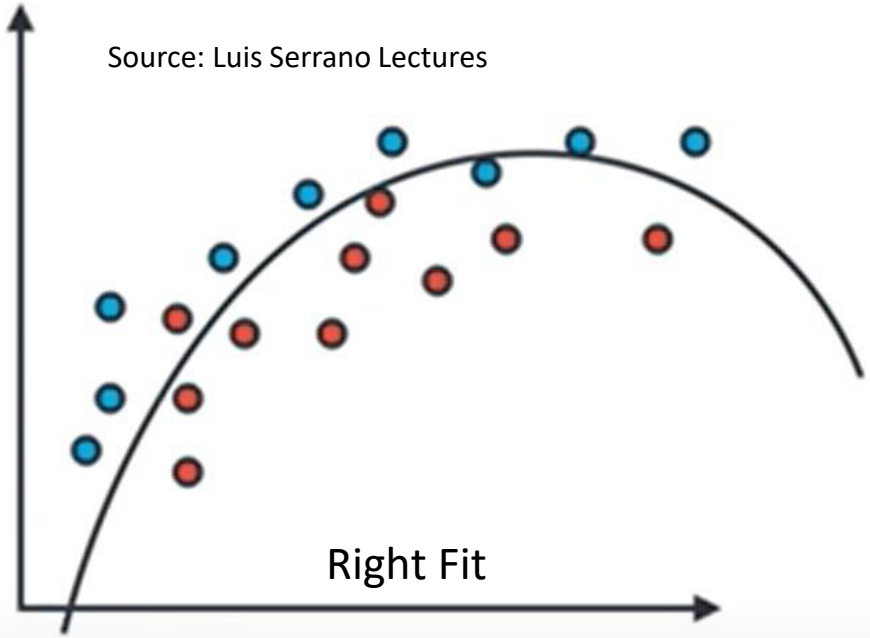
		DETECTION	
		YES	**NO**
PRESENCE	**YES**	**27 (TP)**	**5 (FN)**
	NO	**18 (FP)**	**950 (TN)**

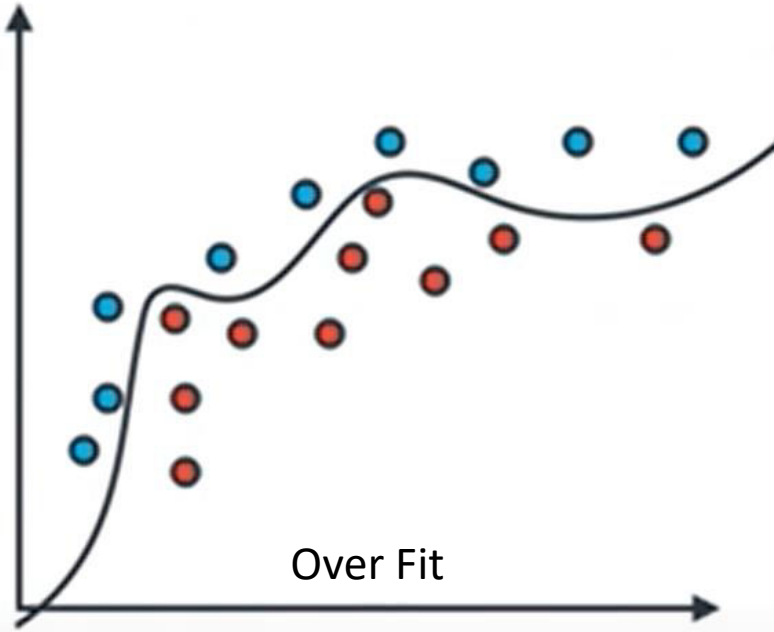


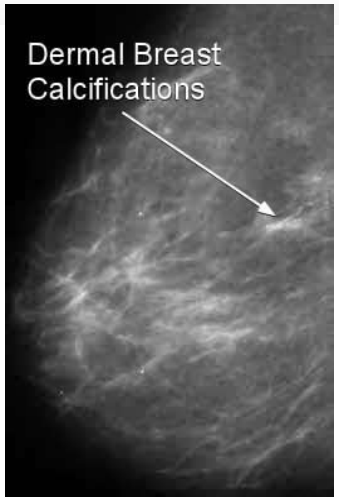
Model Complexity



Source: Luis Serrano Lectures





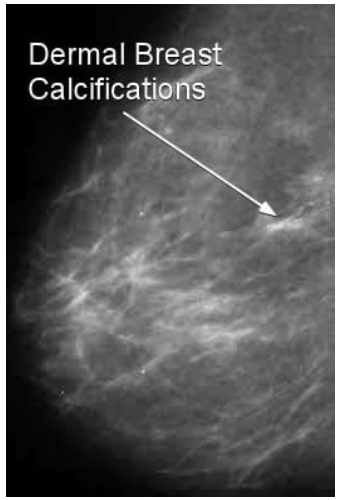


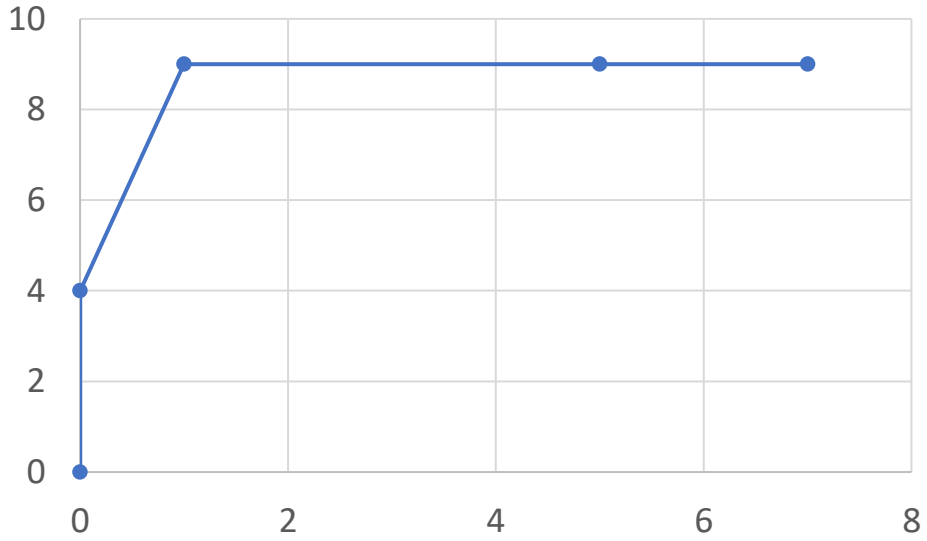
Under Fit – any bright spot (grey pixel value > 180) is malignant tissue

Right Fit – brightness, size, contrast with other similar objects

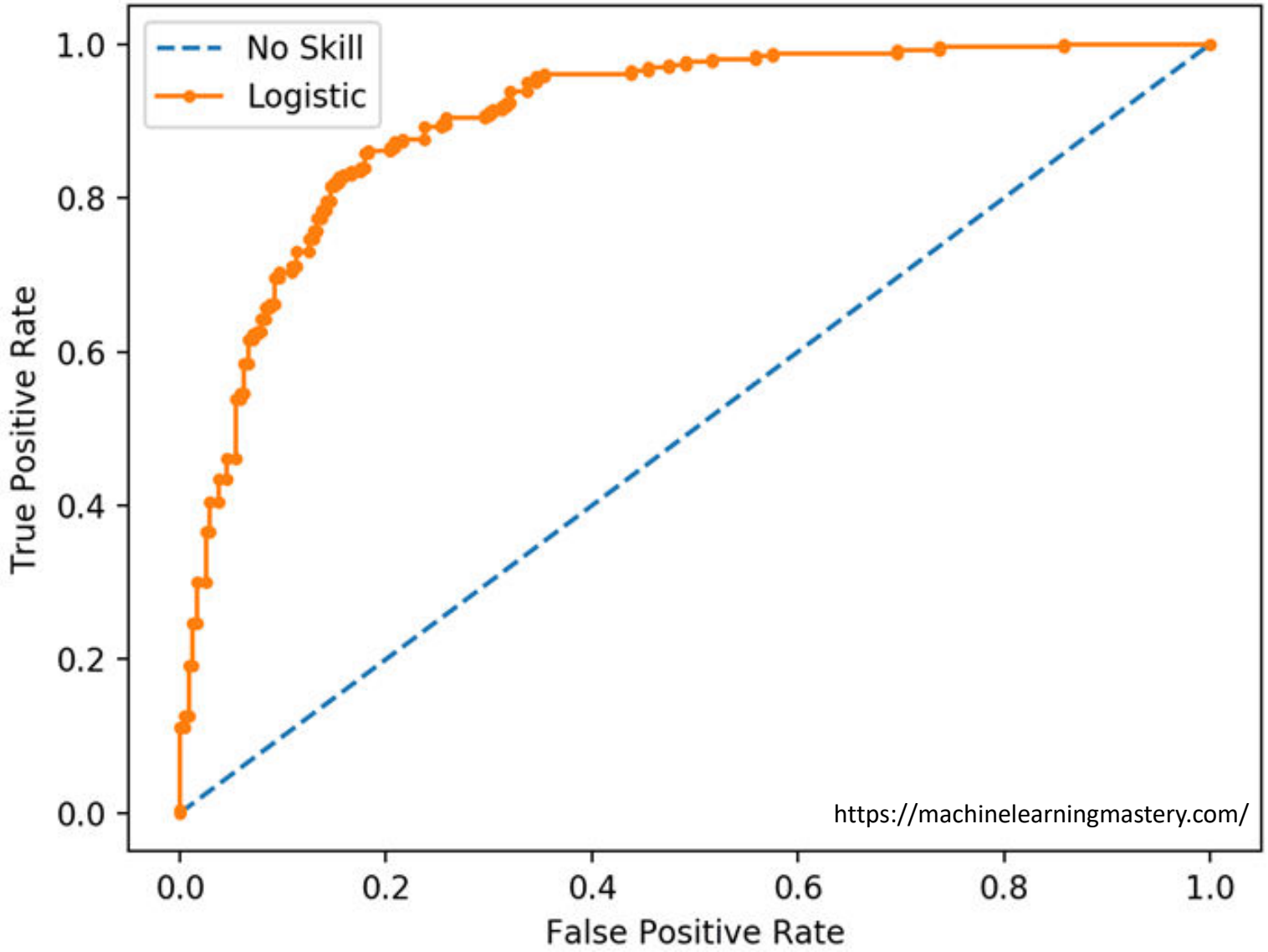
Over Fit – brightness, Contrast, elliptical shape, major/minor axis ratio, location, size etc.

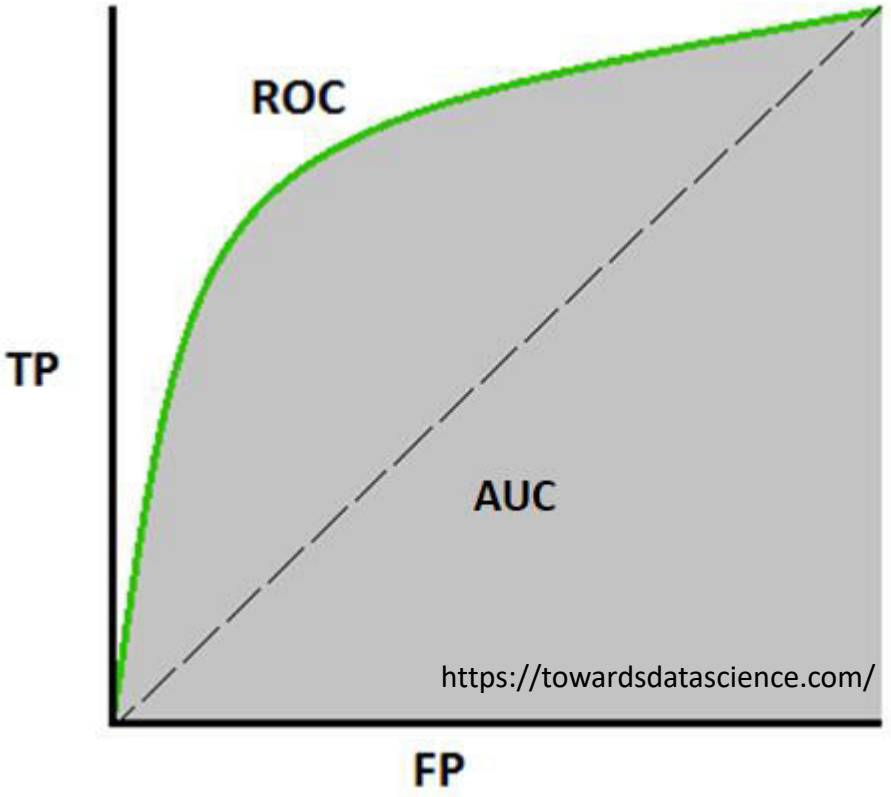
Receiver Operating Characteristic (ROC) Curve





Receiver Operating Characteristic (ROC) Curve, AUC





Thank you !!!!!

Machine Learning (19CSE305)

Session 32-33Clustering

Dr. Peeta Basa Pati
Ms. Priyanka V
Department of Computer Science & Engineering,
Amrita School of Engineering, Bengaluru

Topics

- Clustering.
- k-means clustering

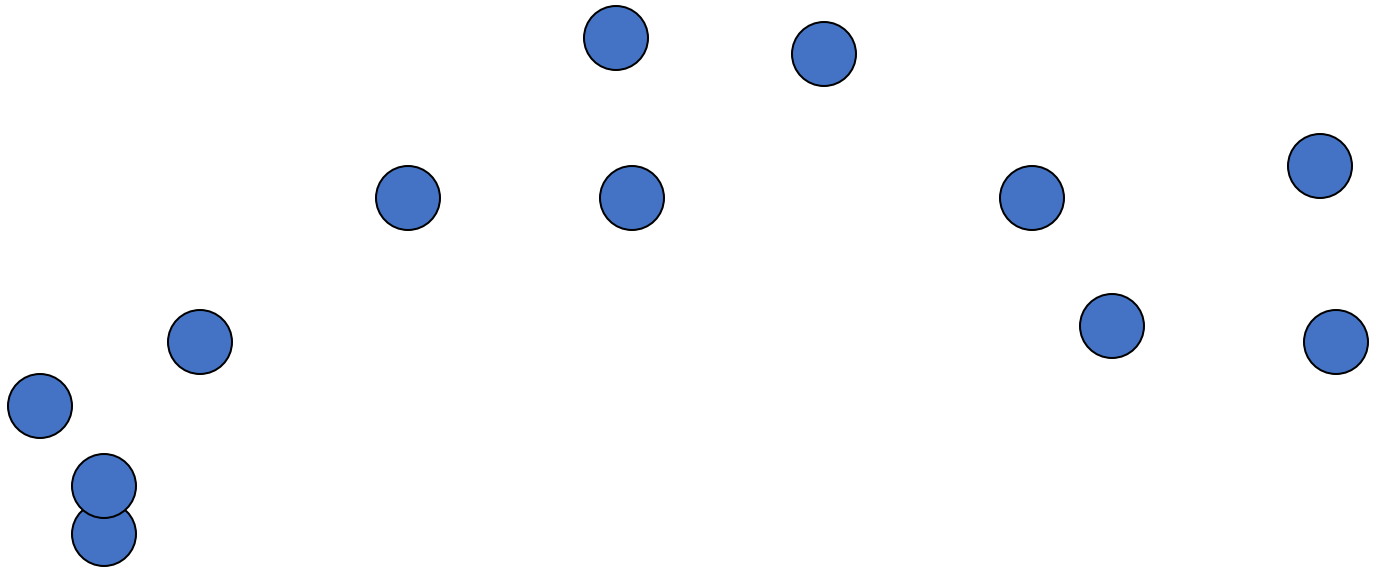
Clustering

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

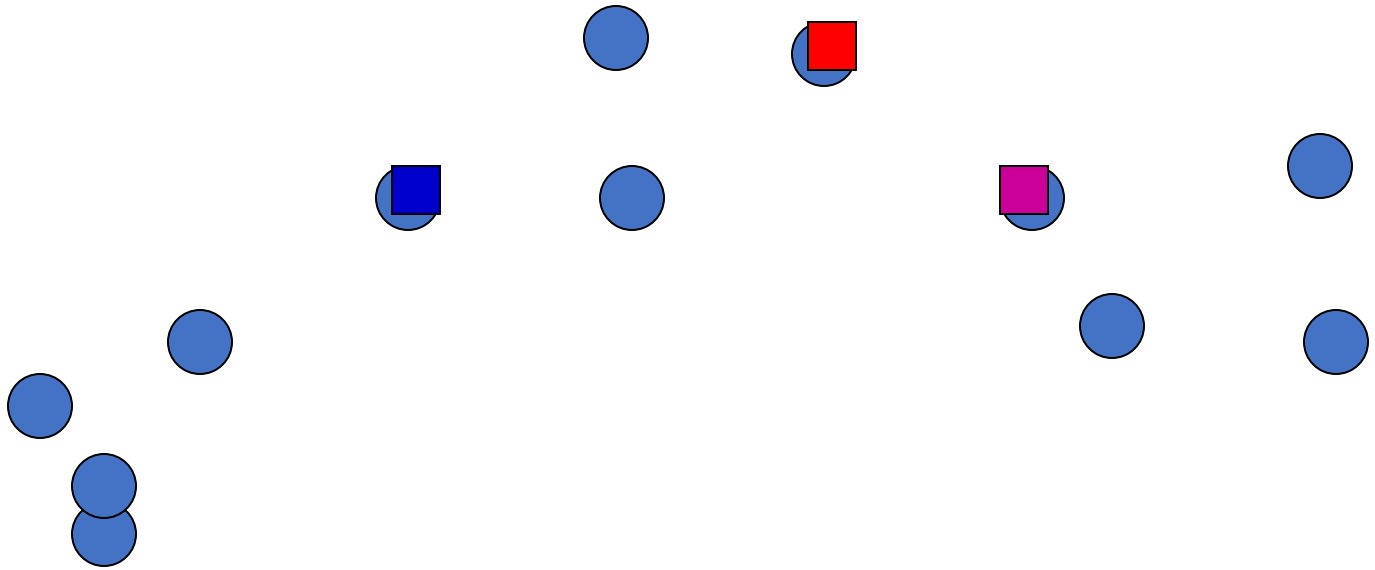
The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change

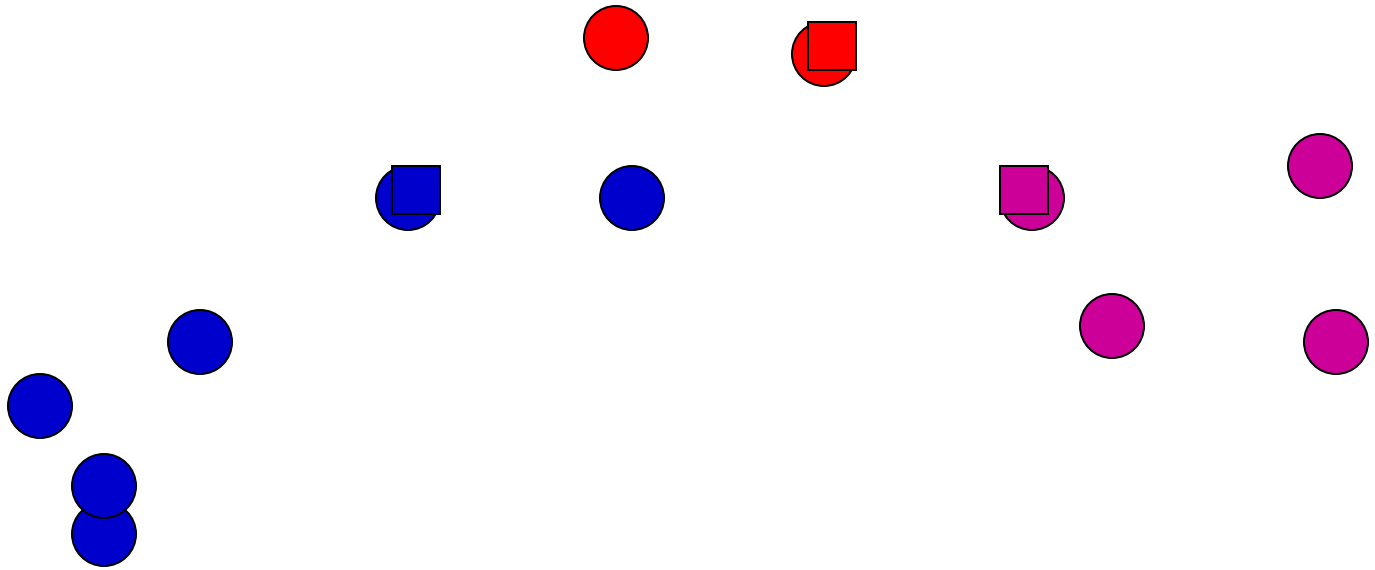
K-means: an example



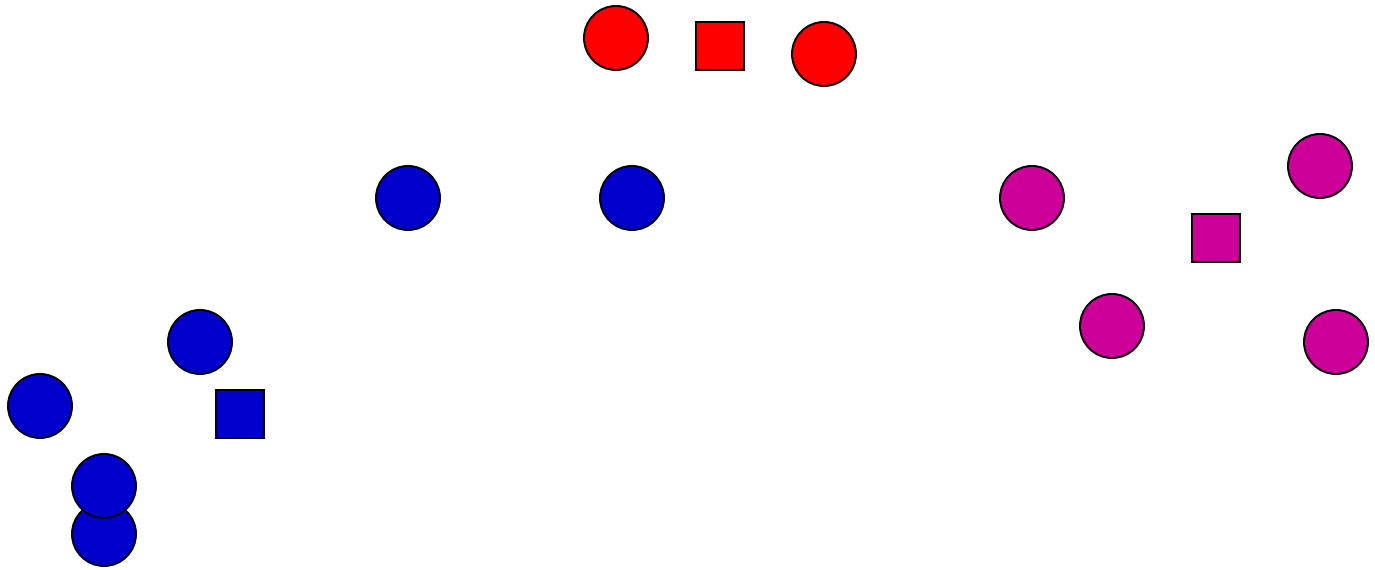
K-means: Initialize centers randomly



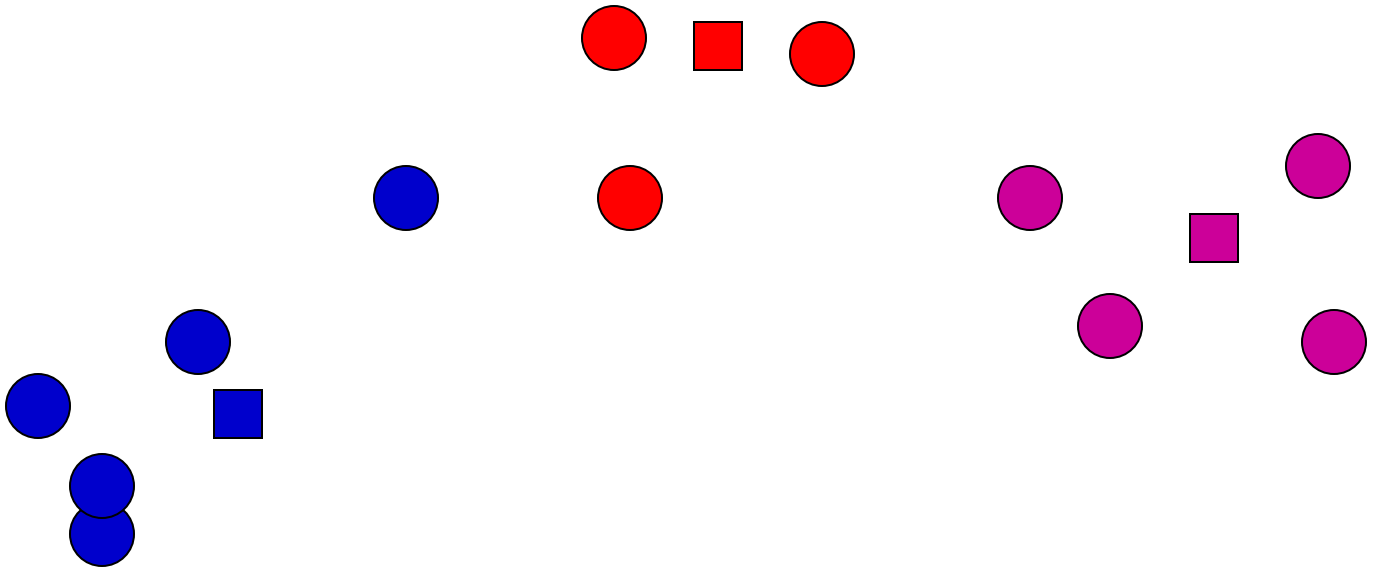
K-means: assign points to nearest center



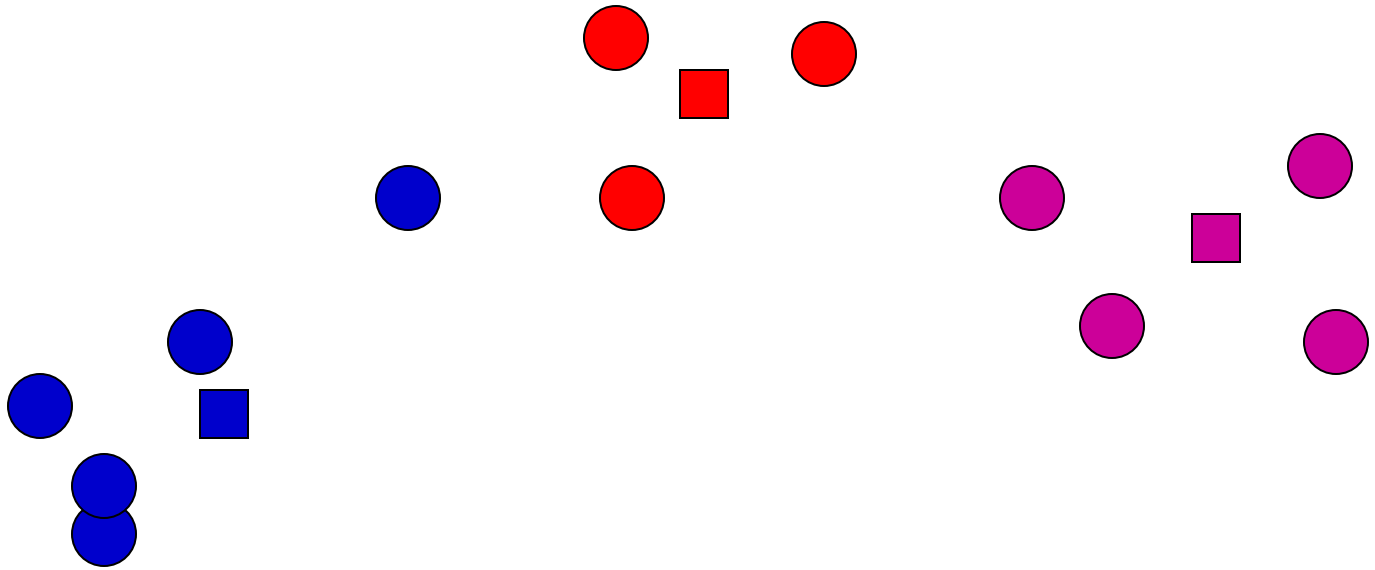
K-means: readjust centers



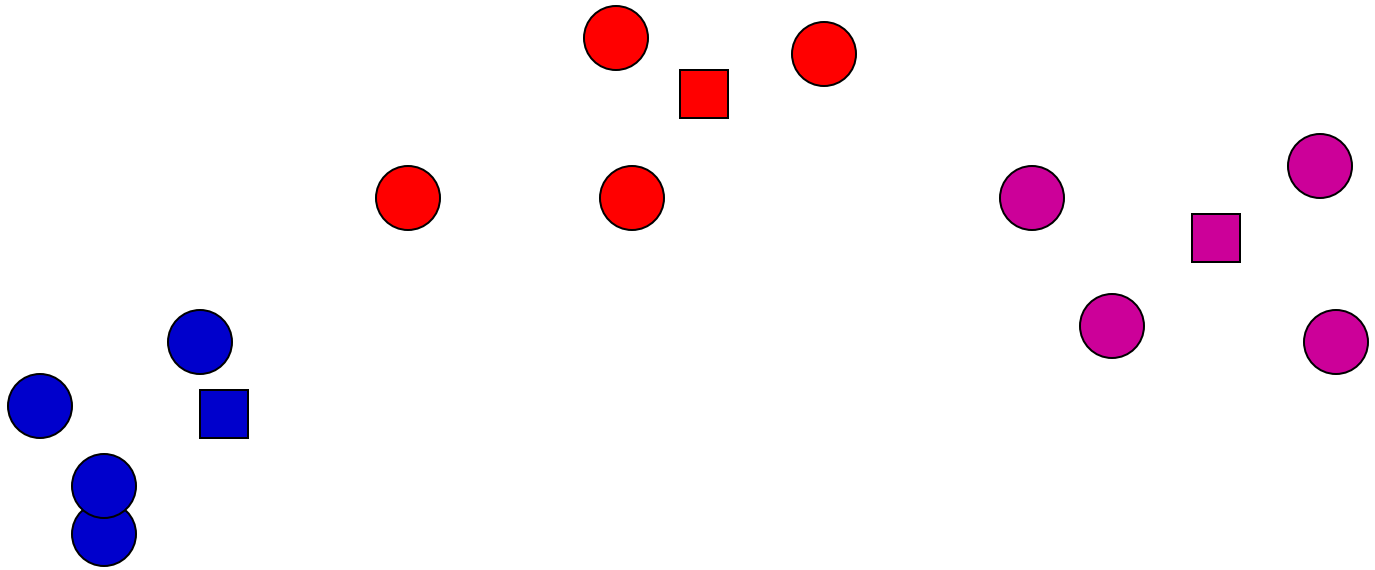
K-means: assign points to nearest center



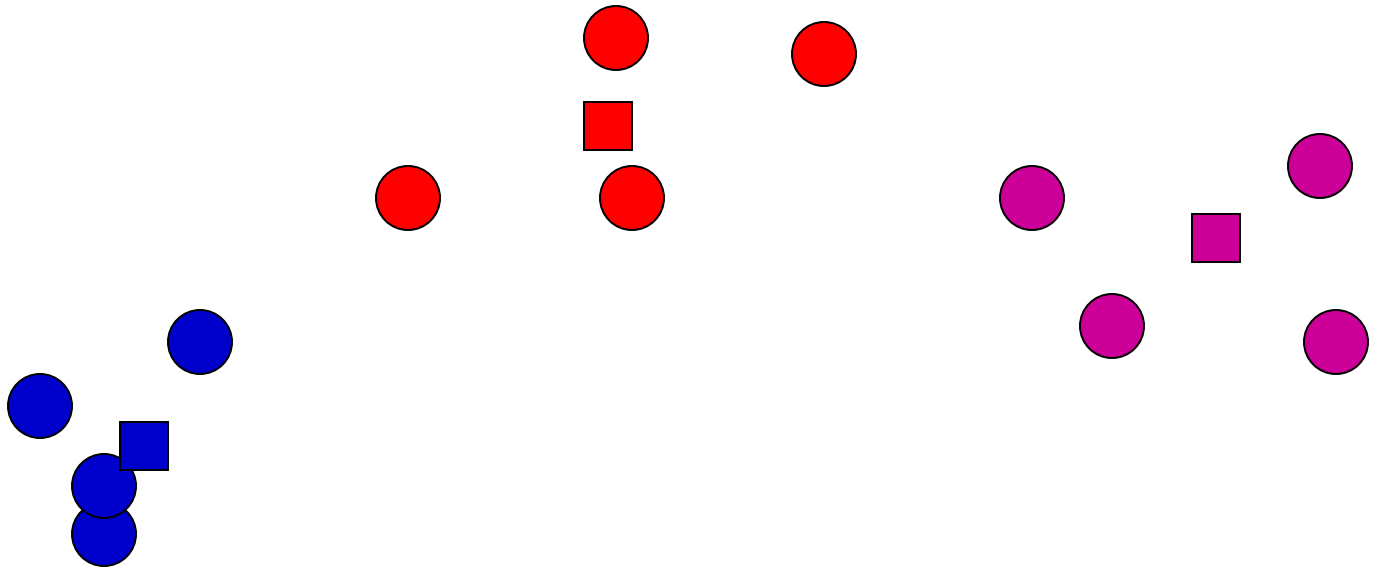
K-means: readjust centers



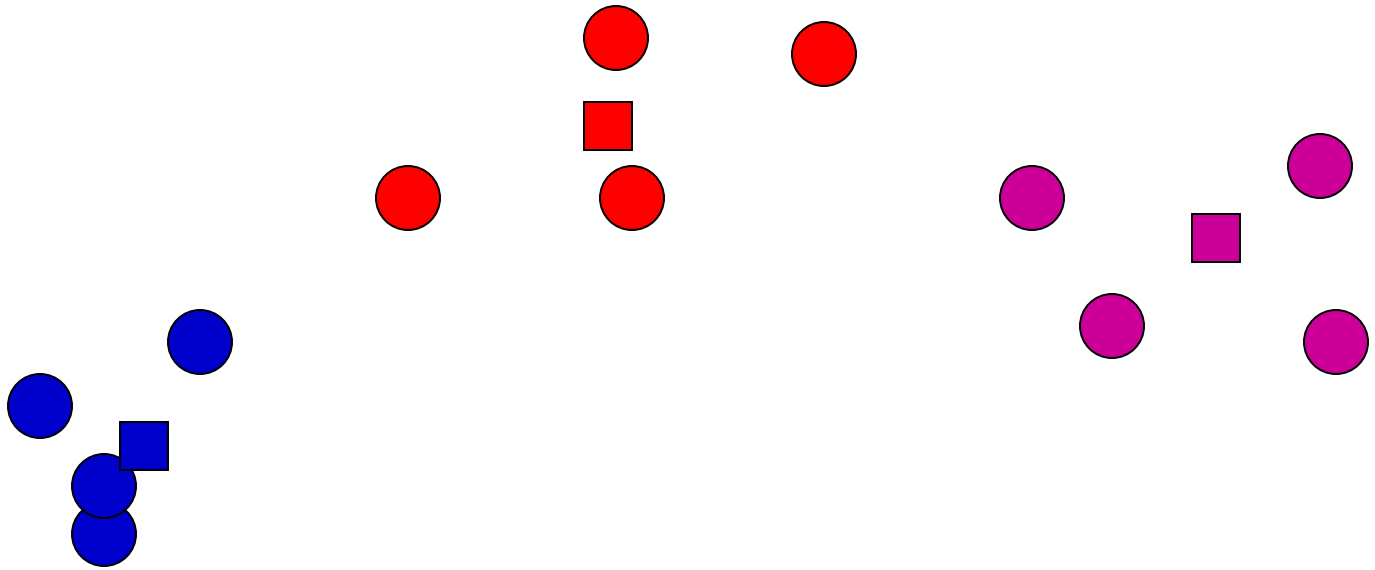
K-means: assign points to nearest center



K-means: readjust centers



K-means: assign points to nearest center

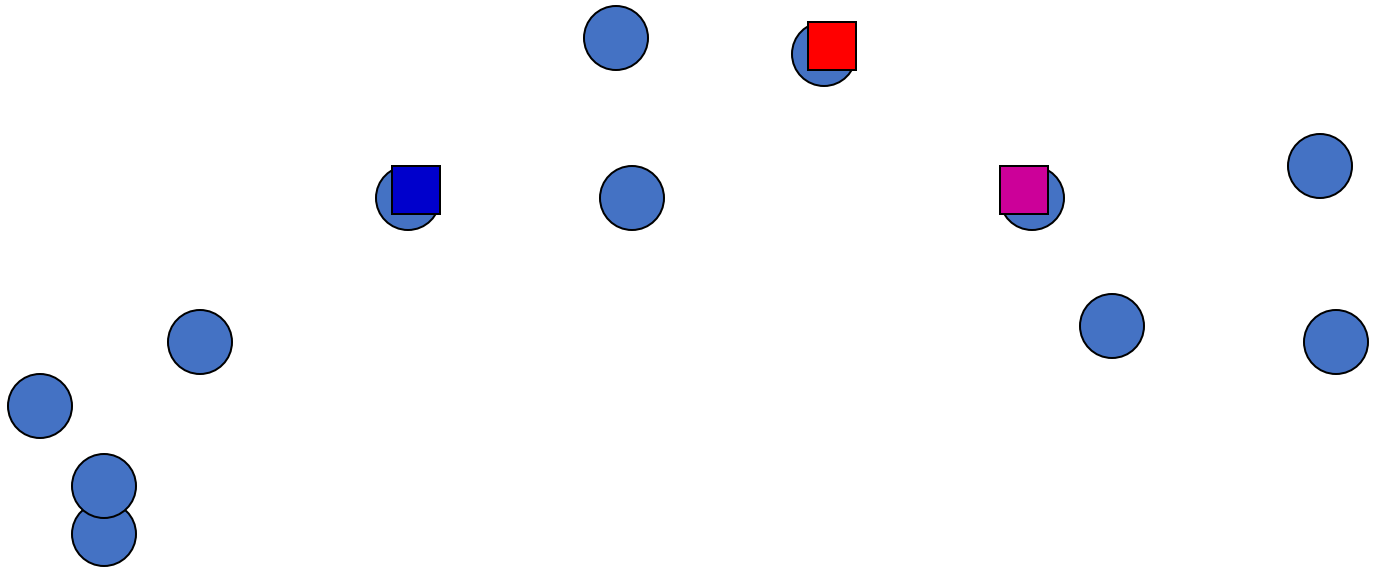


No changes: Done

K-means

Iterate:

- **Assign/cluster each example to closest center**
- Recalculate centers as the mean of the points in a cluster



How do we do this?

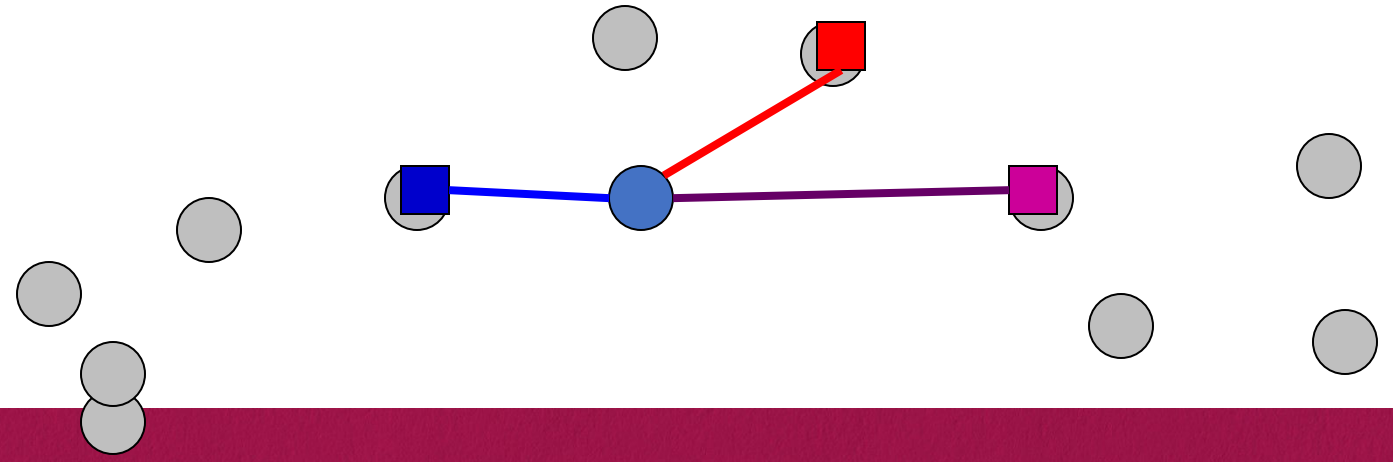
K-means

Iterate:

- **Assign/cluster each example to closest center**

  iterate over each point:
  - get distance to each cluster center
  - assign to closest center (hard cluster)

- Recalculate centers as the mean of the points in a cluster



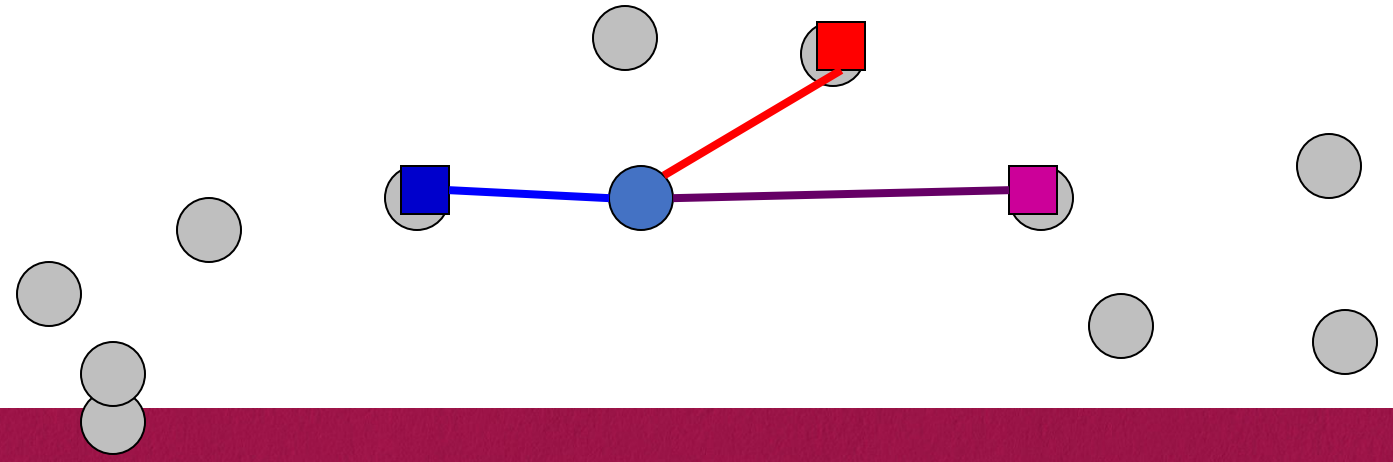
K-means

Iterate:

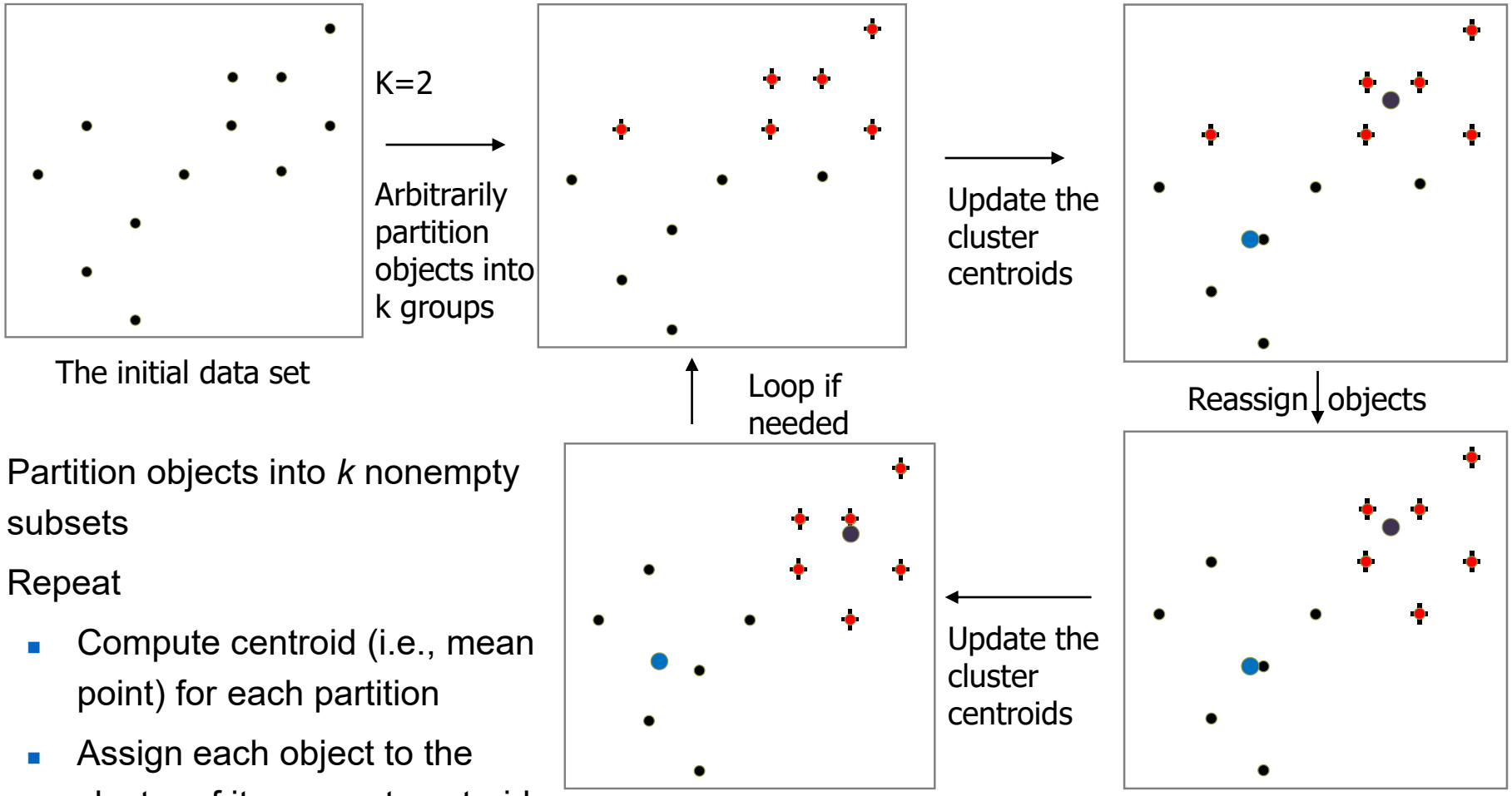
- **Assign/cluster each example to closest center**

  iterate over each point:
  - get **distance** to each cluster center
  - assign to closest center (hard cluster)

- Recalculate centers as the mean of the points in a cluster



An Example of *K-Means* Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

A Simple example showing the implementation of k-means algorithm (using K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 1:

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

- Thus, we obtain two clusters containing:

  {1,2,3} and {4,5,6,7}.
- Their new centroids are:

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$= (4.12, 5.38)$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.

- Therefore, the new clusters are:

  {1,2} and {**3**,4,5,6,7}

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Step 4 :

  The clusters obtained are:

  {1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

What Is the Problem of the K-Means Method?

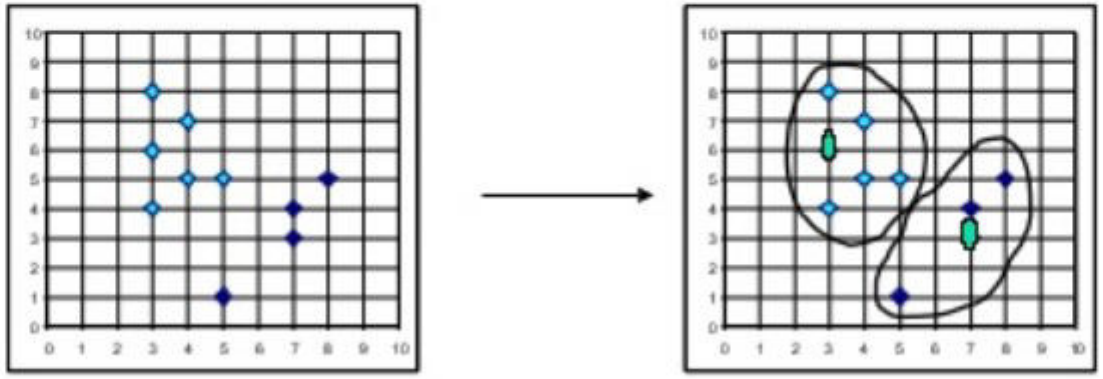
- Since an object with an extremely large value may substantially distort the distribution of the data –senisitve to outliers
- The k-means method is not guaranteed to converge to the global optimum and often terminates at a local optimum.
- The k-means method can be applied only when the mean of a set of objects is defined. This may not be the case in some applications such as when data with nominal attributes are involved.
- The necessity for users to specify k, the number of clusters, in advance

K-Med

What is the problem of k-Means Method?

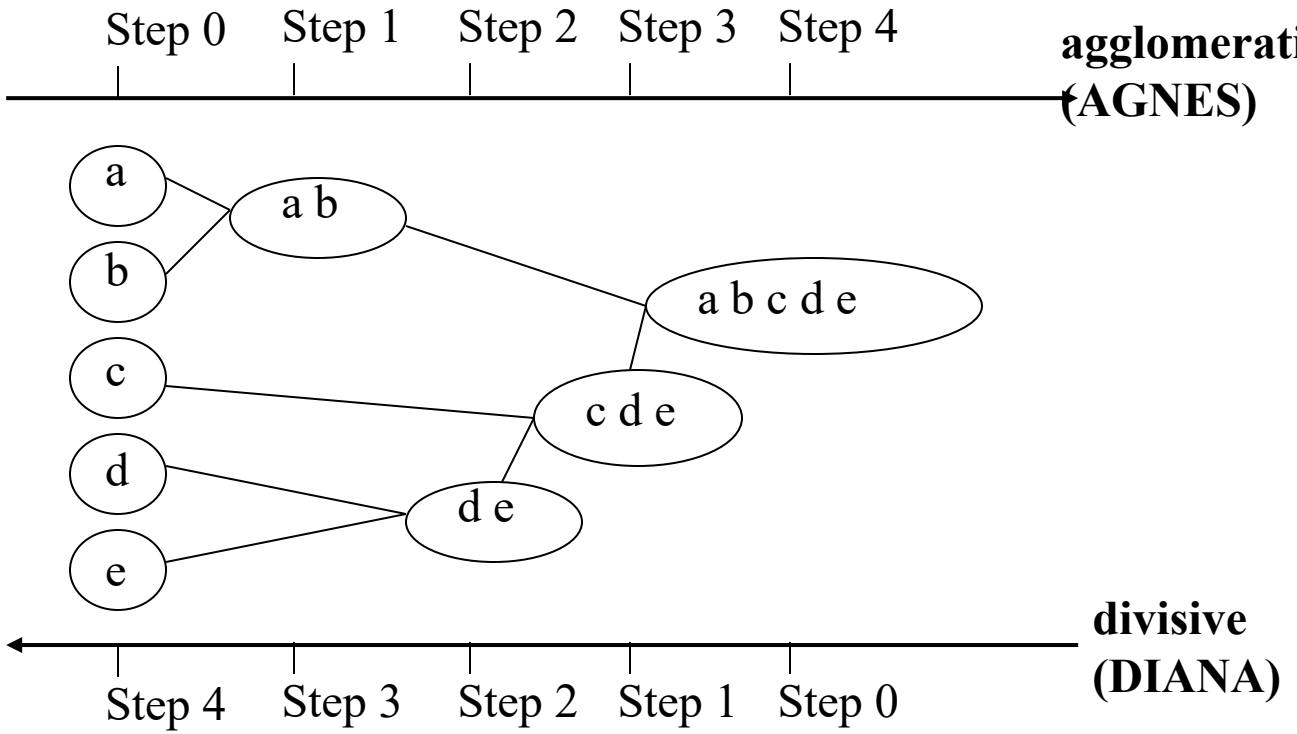
Clip slide

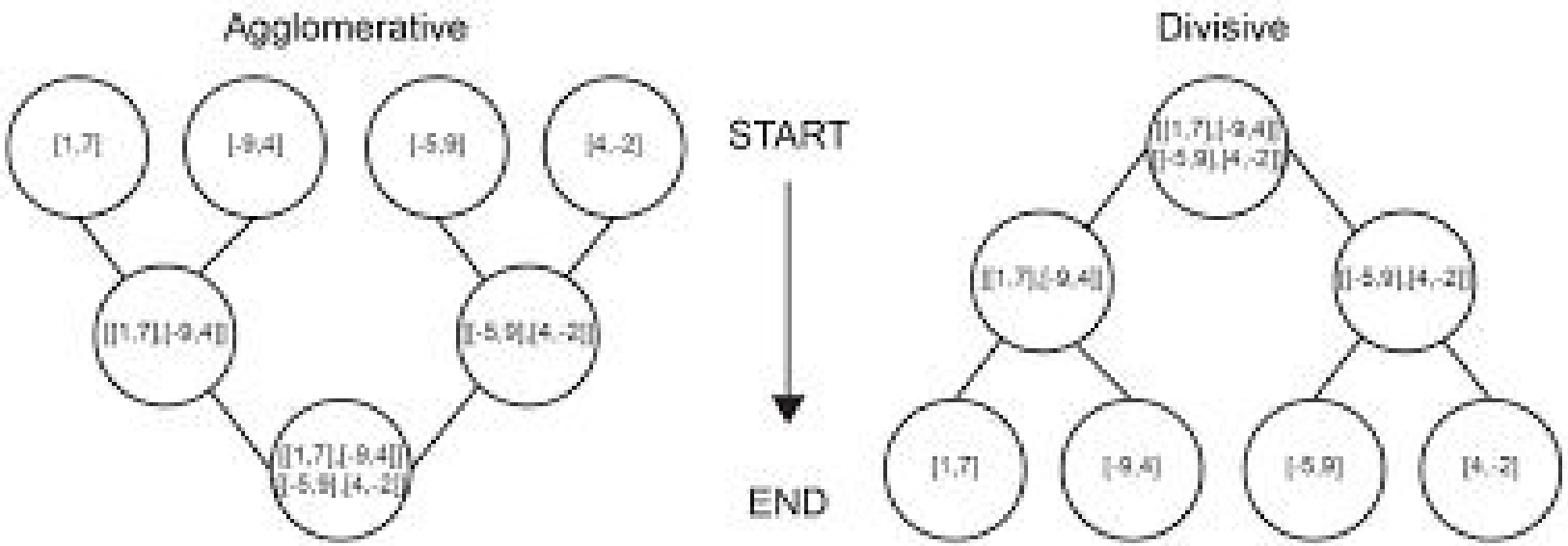
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input.

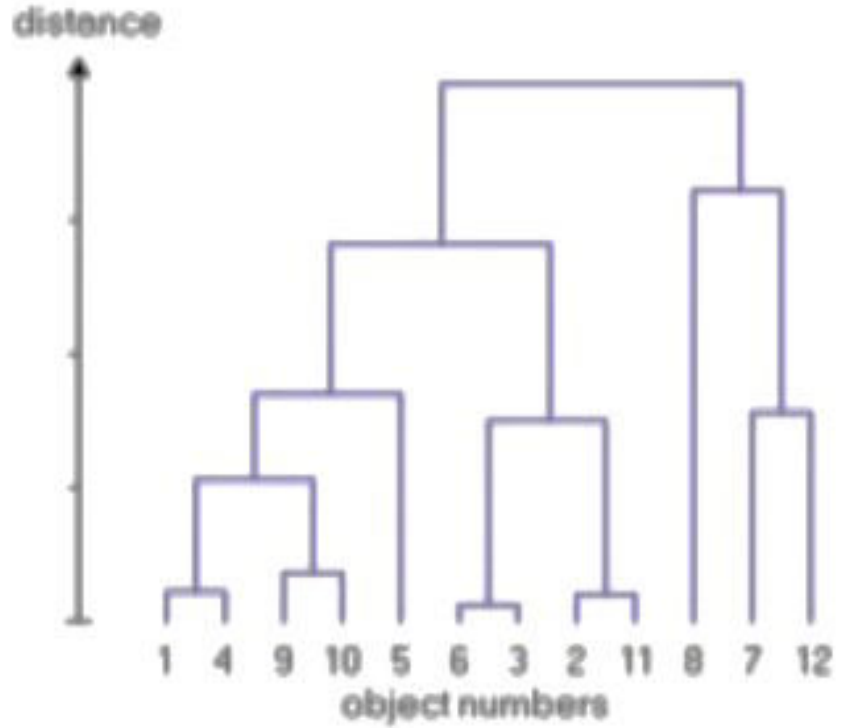




Types of hierarchical clustering

- Agglomerative (bottom up) clustering: It builds the dendrogram (tree) from the bottom level, and
  - merges the most similar (or nearest) pair of clusters
  - stops when all the data points are merged into a single cluster (i.e., the root cluster).
- Divisive (top down) clustering: It starts with all data points in one cluster, the root.
  - Splits the root into a set of child clusters. Each child cluster is recursively divided further
  - stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point

Dendrogram: Hierarchical Clustering



Dendrogram

- Each level of the tree represents a partition of the input data into several (nested) clusters or groups.
- May be cut at any level: Each connected component forms a cluster.

Single linkage

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

Distance Matrix

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Closest D, F so merge

Single Linkage

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

$$d_{(D,F)\mapsto A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

$$d_{(D,F)\mapsto B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

$$d_{(D,F)\mapsto C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$

	A	B	C	(D,F)	E
A	0	0.71	5.66	3.2	4.24
B	0.71	0	4.95	2.5	3.54
C	5.66	4.95	0	2.24	1.41
(D,F)	3.2	2.5	2.24	0	1
E	4.24	3.54	1.41	1	0

Min is 0.71 . Merge A & B

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

$$d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

$d_{(D,F)\to(A,B)} =$ min (d_{DFA},d_{DFB})=min(3.2,2.5)=2.5

$$d_{E\to(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Min Distance (Single Linkage)

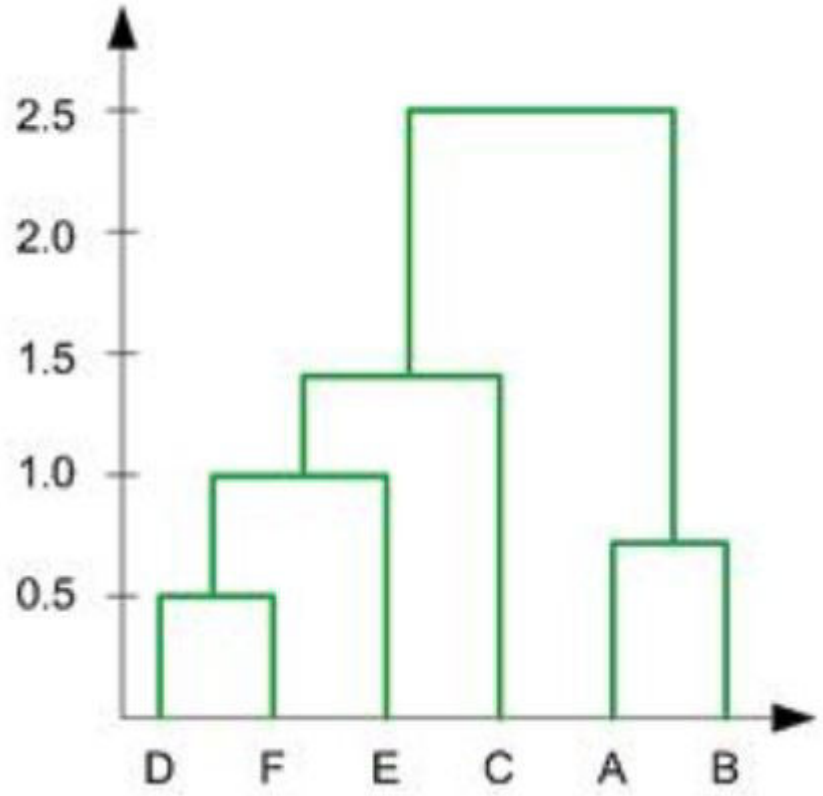
Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

Min Distance (Single Linkage)

Dist	(A,B)	(D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge cluster D and F into cluster (D, F) at distance **0.50**
3. We merge cluster A and cluster B into (A, B) at distance **0.71**
4. We merge cluster E and (D, F) into ((D, F), E) at distance **1.00**
5. We merge cluster ((D, F), E) and C into (((D, F), E), C) at distance **1.41**
6. We merge cluster (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance **2.50**
7. The last cluster contain all the objects, thus conclude the computation

Dendrogram



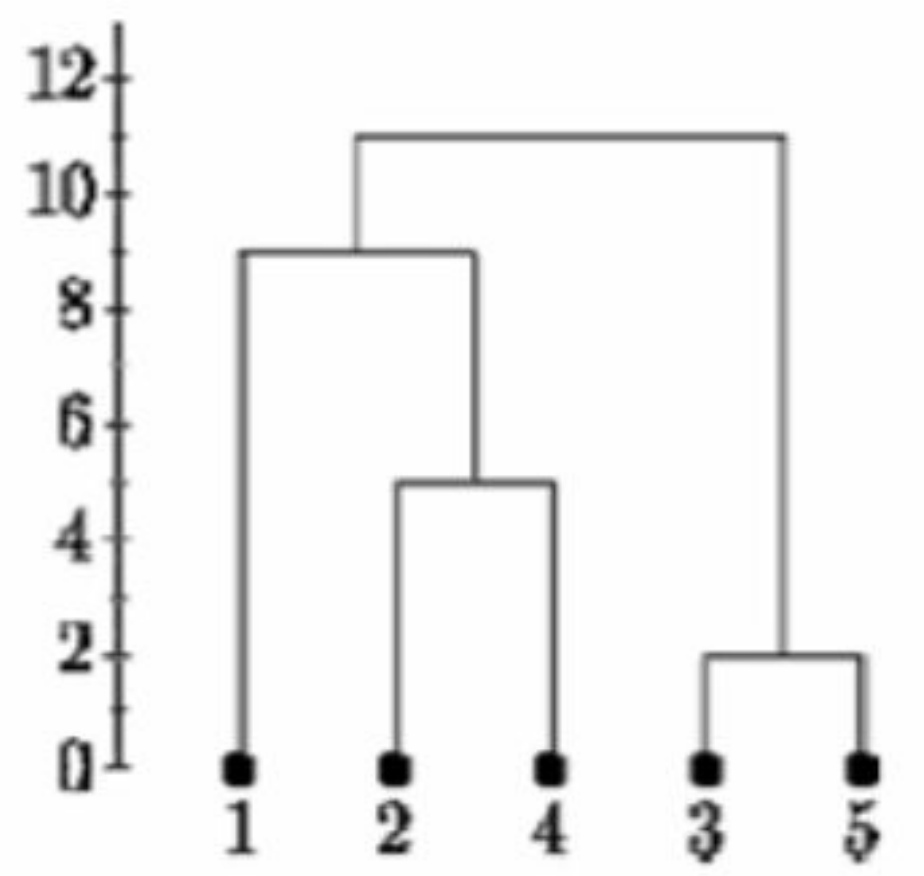
Complete Linkage

	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	2	8	0

Complete Linkage

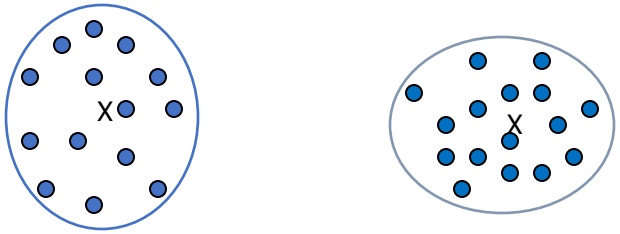
- Since we are using complete linkage clustering, the distance between "35" and every other item is the maximum of the distance between this item and 3 and this item and 5. For example, d(1,3)= 3 and d(1,5)=11. So, D(1,"35")=11.

	35	1	2	4
35	0			
1	11	0		
2	10	9	0	
4	9	6	5	0



Complete Linkage

Linkage criteria



- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$

Determine the Number of Clusters

- Empirical method
  - # of clusters ≈√n/2 for a dataset of n points
- Elbow method
  - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
- Cross validation method
  - Divide a given data set into *m* parts
  - Use *m* – 1 parts to obtain a clustering model
  - Use the remaining part to test the quality of the clustering
    - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
  - For any k > 0, repeat it *m* times, compare the overall quality measure w.r.t. different *k's*, and find # of clusters that fits the data the best

Measuring Clustering Quality

- External: supervised, employ criteria not inherent to the dataset
  - Compare a clustering against prior or expert-specified knowledge using certain clustering quality measure
- Internal: unsupervised, criteria derived from data itself
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are, e.g., Silhouette coefficient
- Relative: directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm

- K-means and K-medoids algorithms are popular partitioning-based clustering algorithms
- Birch and Chameleon are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms
- DBSCAN, OPTICS, and DENCLU are interesting density-based algorithms
- STING and CLIQUE are grid-based methods, where CLIQUE is also a subspace clustering algorithm

Thank you !!!!!

Machine Learning (19CSE305)

Feature Selection

AMRITA VISHWA VIDYAPEETHAM

DEEMED TO BE UNIVERSITY

Dr. Peeta Basa Pati
Ms. Priyanka V
Department of Computer Science & Engineering,
Amrita School of Engineering, Bengaluru

Topics

- Feature extraction vs selection
- Search techniques
- Criterion functions

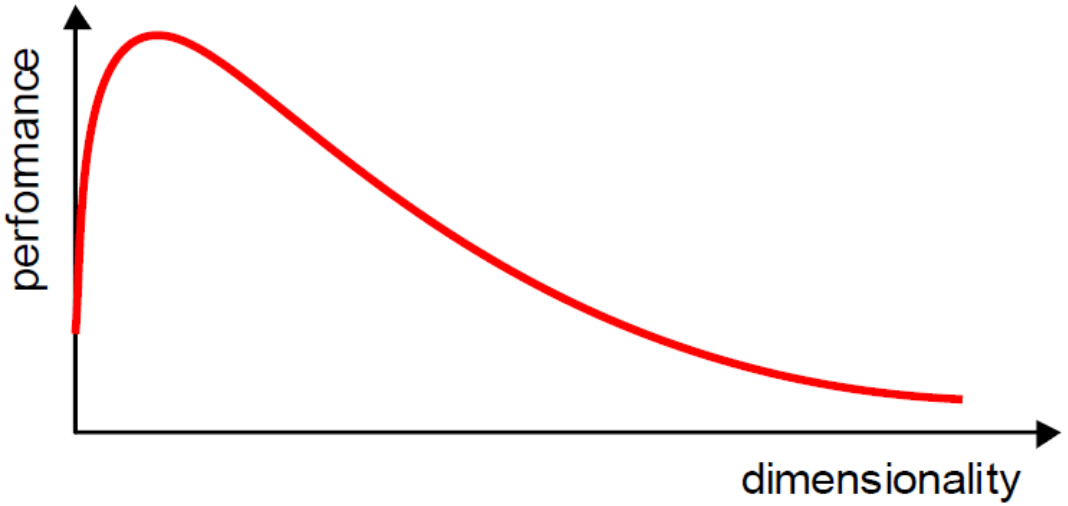
A very good reference book on this module:

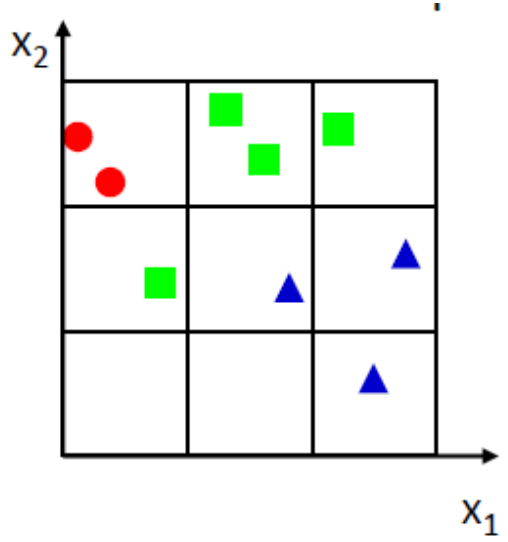
Pattern Recognition 4th Edition by **Theodoridis** and **Koutroumbas**

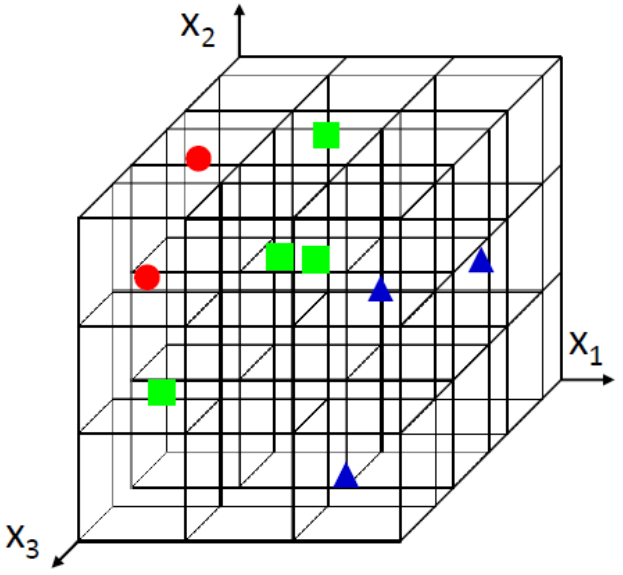
Images used in these slides are taken from multiple sources. All authors are acknowledged.

Curse of Dimensionality

- Increasing features initially increases accuracy but deteriorates after some point
- The number of training examples required increases exponentially with dimensionality **d** (i.e., $k^{\mathbf{d}}$).
- Other curses – visualization & performance challenges







Ref: www.cse.unr.edu/~bebis/CS479/Lectures/DimensionalityReduction.ppt

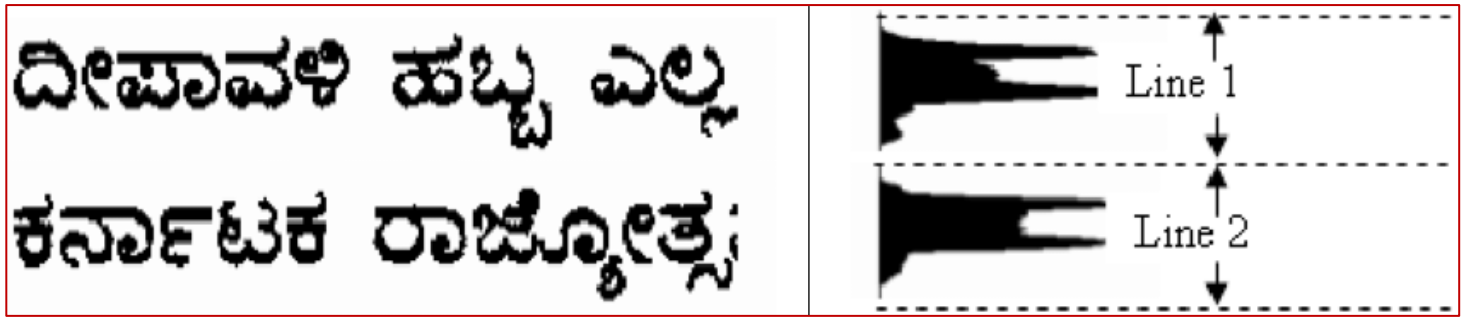
Feature Extraction

Extract a new set of features through a mapping function

$\mathbf{y} = f(\mathbf{X})$

Ex: projection profiles, DCT, MFCC etc.

Linear or non-linear combination

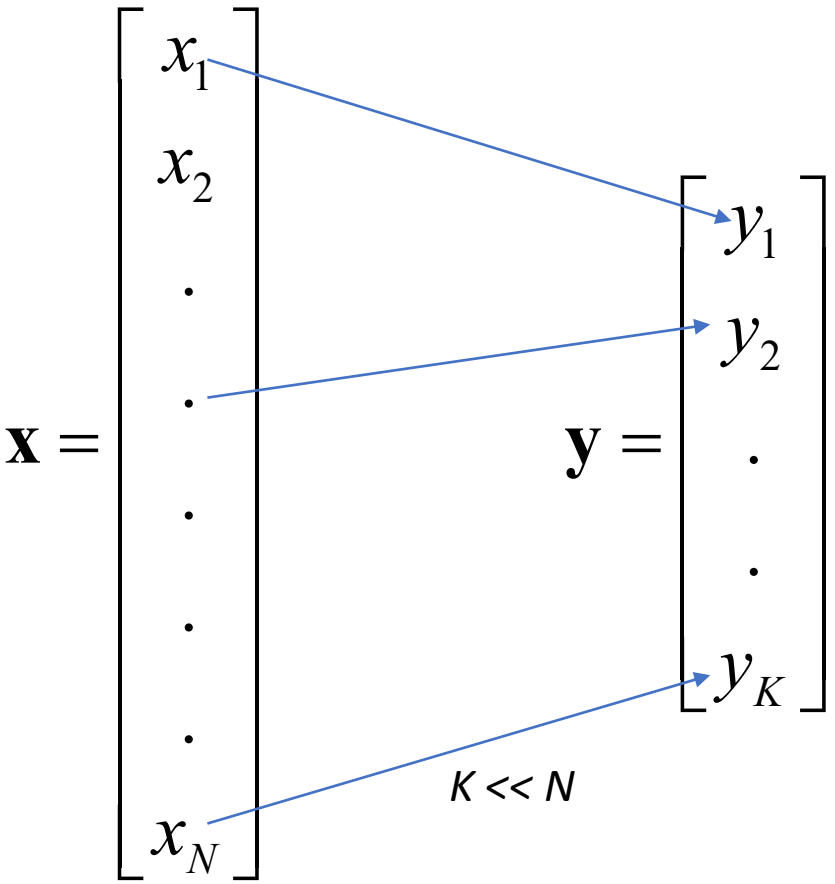


DOI: 10.1007/s12046-007-0039-1

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix} \xrightarrow{f(\mathbf{x})} \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_K \end{bmatrix}$$

$K << N$

Feature Selection

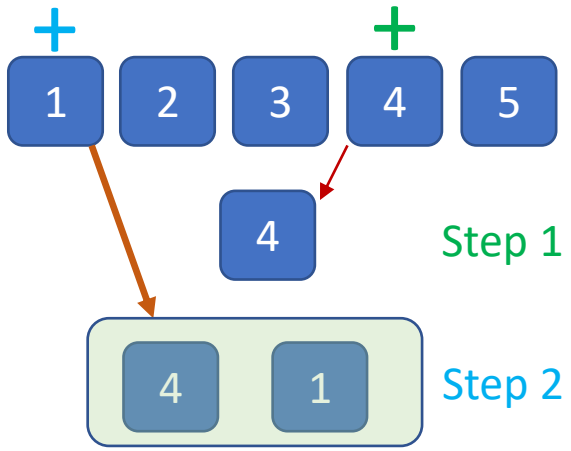


- Select an optimal subset of features from the original feature set
- Search problem
- Search Algorithms
  - Sequential forward search / generation
  - Sequential backward search / removal
  - Bidirectional search
  - Sequential forward-backward search
- Selection criterion
  - Entropy
  - Information Gain
  - Accuracy of classification etc.

Sequential forward / backward search

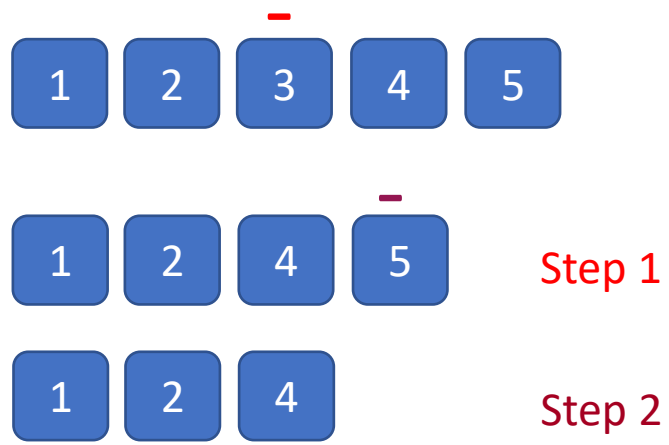
Algorithm 1 Sequential forward feature set generation - SFG.

function SFG(F - full set, U - measure)
  initialize: $S = \{\}$ ▷ S stores the selected features
  repeat
    f = FINDNEXT(F)
    $S = S \cup \{f\}$
    $F = F - \{f\}$
  until S satisfies U *or* $F = \{\}$
  return S
end function



Algorithm 2 Sequential backward feature set generation - SBG.

function SBG(F - full set, U - measure)
  initialize: $S = \{\}$ ▷ S holds the removed features
  repeat
    f = GETNEXT(F)
    $F = F - \{f\}$
    $S = S \cup \{f\}$
  until S does not satisfy U *or* $F = \{\}$
  return $F \cup \{f\}$
end function



Bidirectional search

- Begins the search in both directions – SF & SB simultaneously
- Stop if:
  - one search finds optimal subset before it reaches the middle
  - both searches reach middle; select the better of 2 subsets from SF & SB search

Algorithm 3 Bidirectional feature set generation - BG.

function BG(F_f, F_b - full set, U - measure)
  initialize: $S_f = \{\}$ ▷ S_f holds the selected features
  initialize: $S_b = \{\}$ ▷ S_b holds the removed features
  repeat
    $f_f = \text{FINDNEXT}(F_f)$
    $f_b = \text{GETNEXT}(F_b)$
    $S_f = S_f \cup \{f_f\}$
    $F_b = F_b - \{f_b\}$
    $F_f = F_f - \{f_f\}$
    $S_b = S_b \cup \{f_b\}$
  until (a) S_f satisfies U *or* $F_f = \{\}$ *or* (b) S_b does not satisfy U *or* $F_b = \{\}$
  return S_f if (a) *or* $F_b \cup \{f_b\}$ if (b)
end function

Sequential Forward-Backward Search

- Start with either SF or SB search; accuracy is criterion function
- If SF:
  - Start by searching for the feature that gives maximum accuracy
  - Add a feature
  - Check if removal of any (subset) feature(s) improves accuracy; remove
  - Continue
- If SB:
  - Start by searching for the feature that gives minimum accuracy
  - Remove a feature
  - Check if addition if any removed features improves accuracy; add
  - Continue
- May get into loop if proper care is not taken
  - Ex: for SF → same feature gets added and removed; will form a loop
- Modified version – continue adding or removing till optimal set is attained

Thank you !!!!!

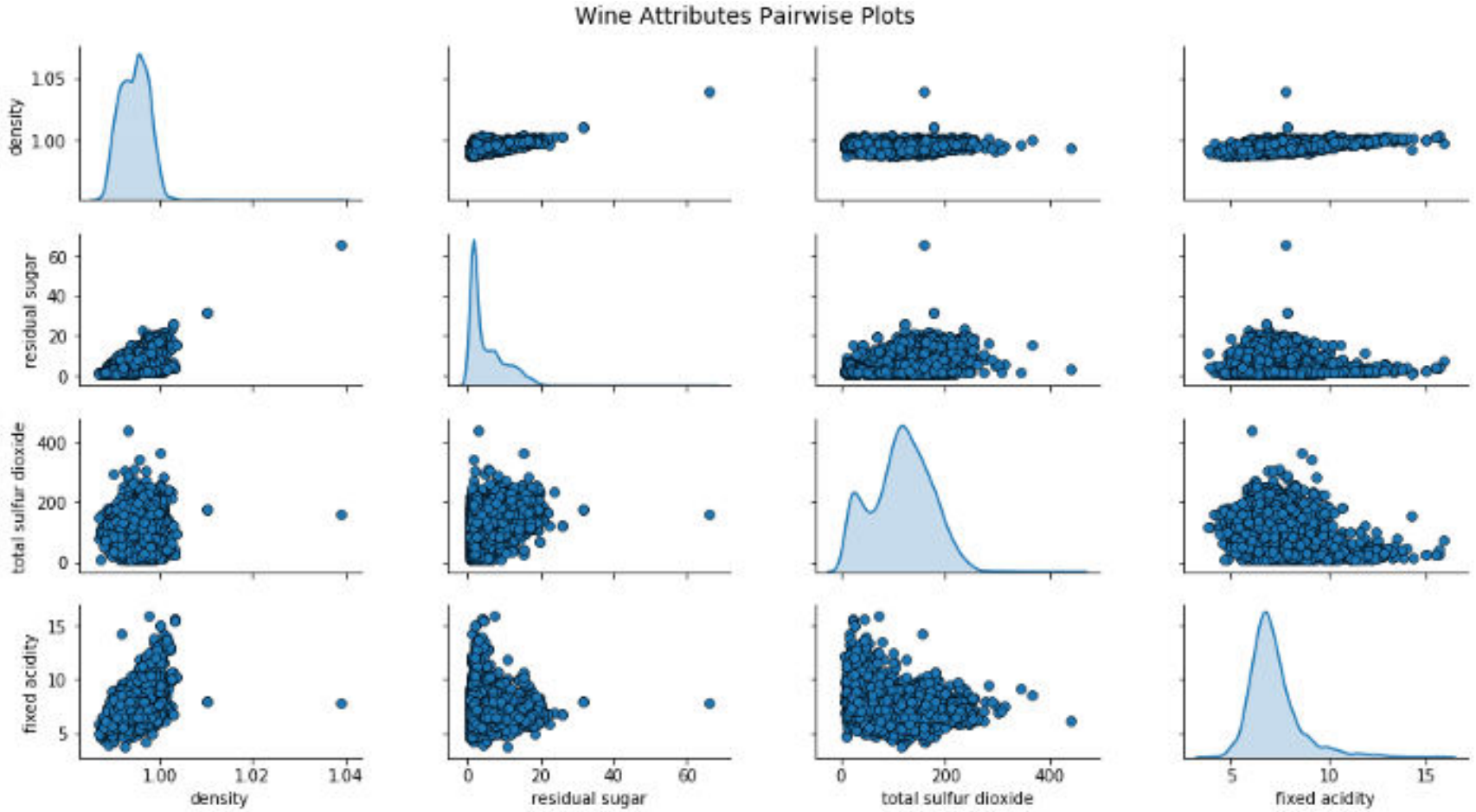
Is high dimension always good?

A bus starts from its source with 15 people. It stops after 35 minutes of travel where 7 people get down and 10 get in. After stopping for 10 mins, it moves for another 45 minutes. At this stop 5 people get down and 12 get in. After halting for 12 mins, the bus travels for 65 minutes and reaches the destination.

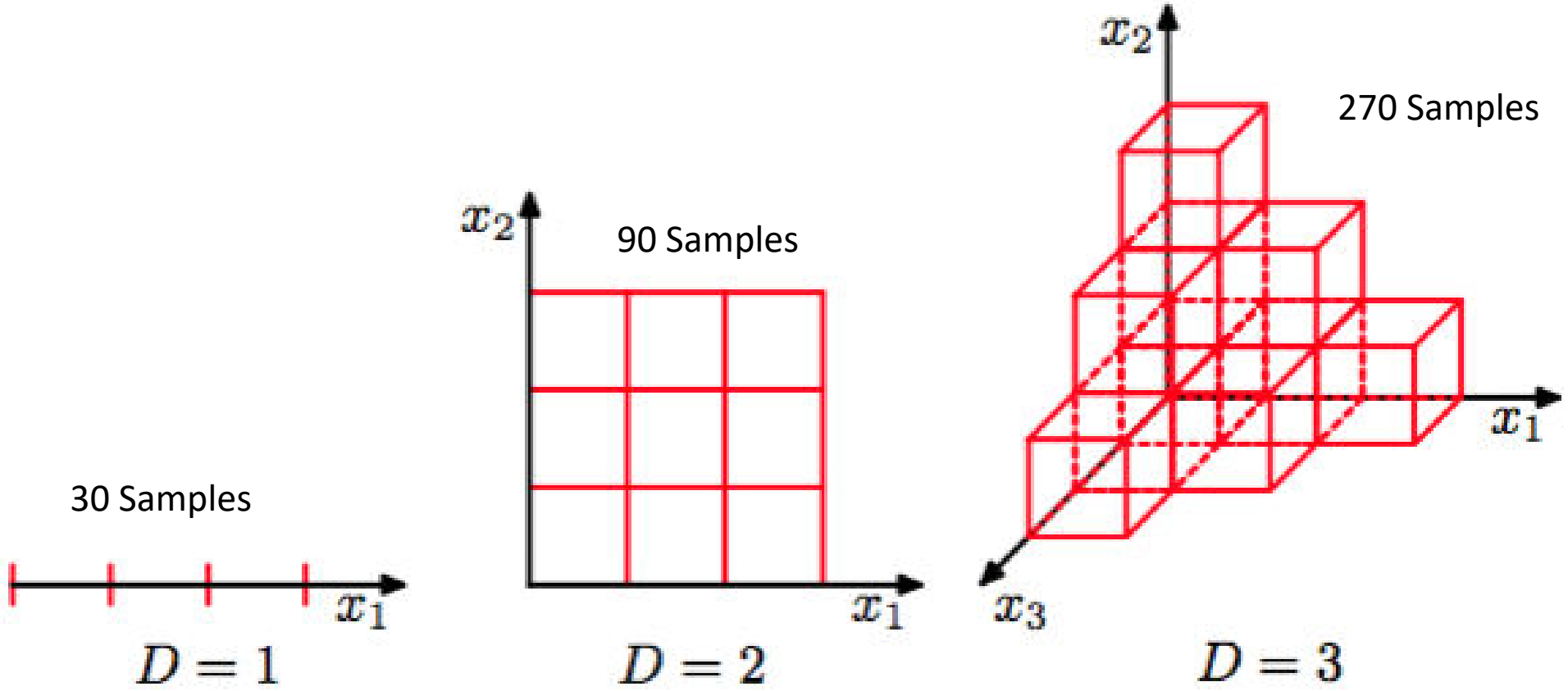
How many places did the bus stop?

Visualization Challenges

- Humans are tuned for visualization up to 3D



Density Estimation Challenges

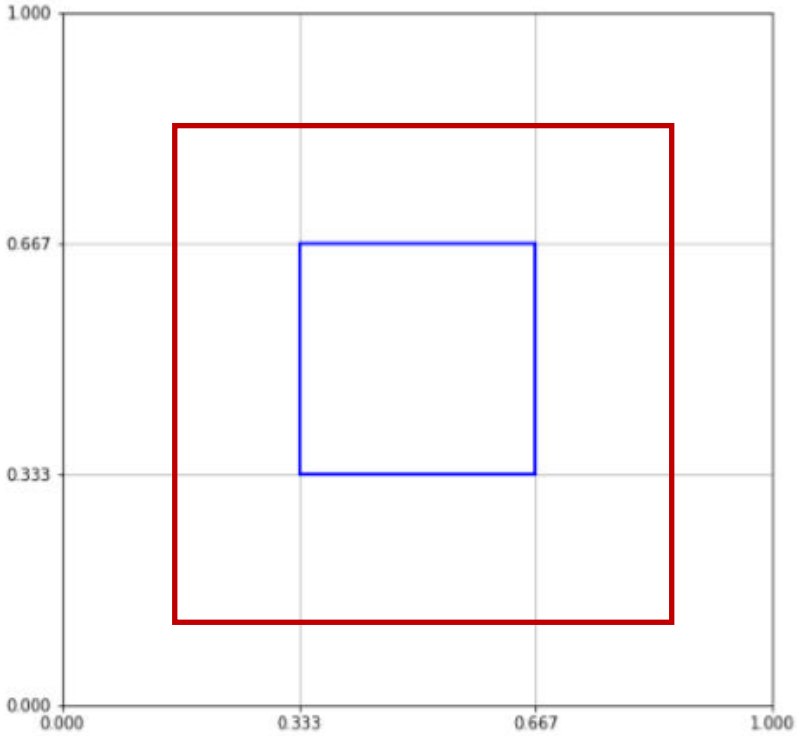


Assume 10 samples per each segment, we'd need 65,610 samples for a 8 dimensional space.

Space Coverage

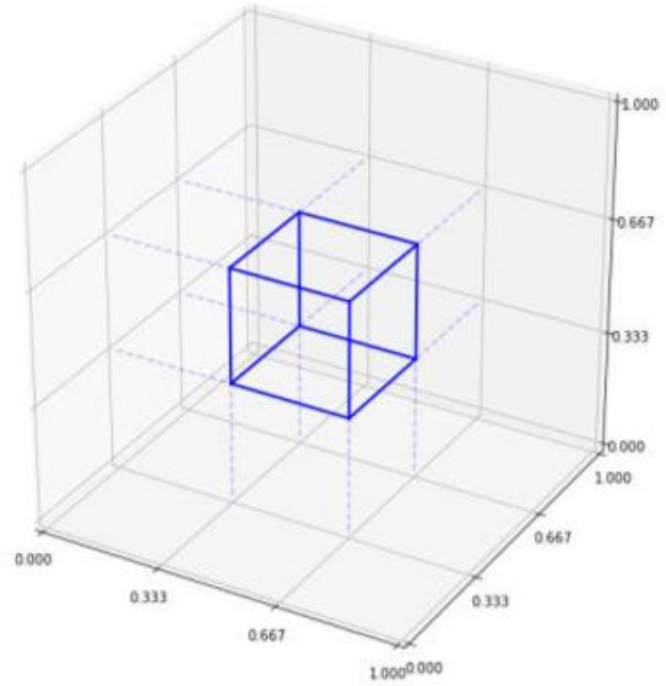
33%

To get 33% coverage, we'd need 57% length on each side.



11%

To get 33% coverage, we'd need 70% length on each side.



3.7%

To get 33% coverage in 8-D, we'd need 87% length on each side.

Performance Challenges

- Euclidean distance
  - Increase due to dimensions
  - Increase due to training vectors
- Density estimation
- Deep learning networks

How to Reduce Dimensionality?

- Feature Selection techniques
  - PCA
  - Identify based on statistical methods
  - Remove the dependent features
- Transformation Techniques
  - Spatio-temporal transformations such as DFT, DCT & Wavelets
    - DCT's reduce by 80% while Wavelets further
  - Representational features such as body temp, words spoken etc.

Thank you !!!!!