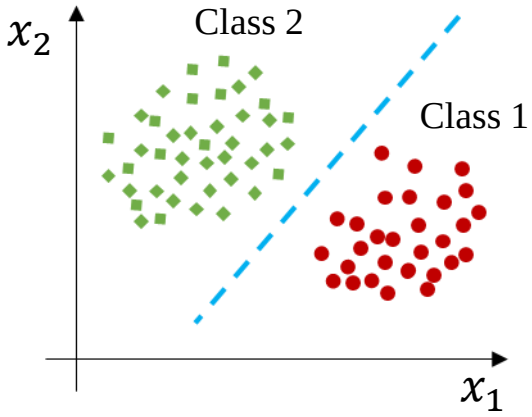
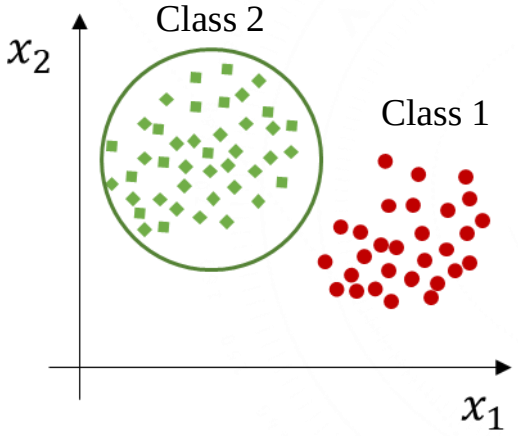
LOGISTIC REGRESSION

RESMI SURESH

ASSISTANT PROFESSOR, IIT GUWAHATI

INTRODUCTION





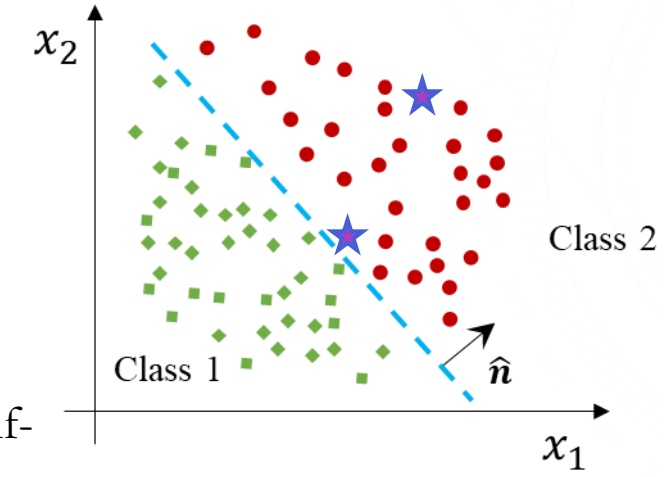
- Logistic regression is a supervised classification technique
- Performs well for binary classification, can be extended to multi-class classification
- Most common approach use a linear model, but can be extended to polynomial models to handle nonlinear cases
- Goal: Given a new data point, predict the class from which the data point is likely to have originated
- Input features can be both qualitative and quantitative
  - If the inputs are qualitative, then there has to be a systematic way of converting them to quantities
  - For example: A binary input like a "Yes" or "No" can be encoded as "1" and "0"

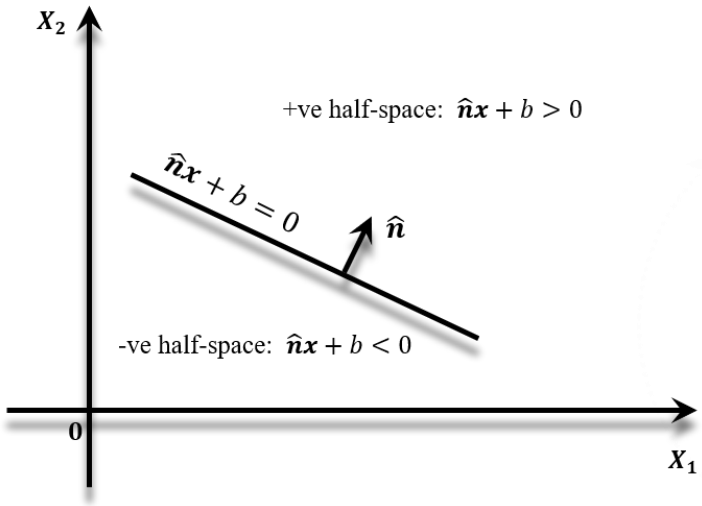
LINEAR CLASSIFIER

- Decision function is linear
- Binary classification can be performed depending on the side of the half-plane that the data falls in
- Let the hyper-plane that separates the two classes be $h(x) = n^T x + b$

  If $h(x) > 0$, sample belongs to class 2

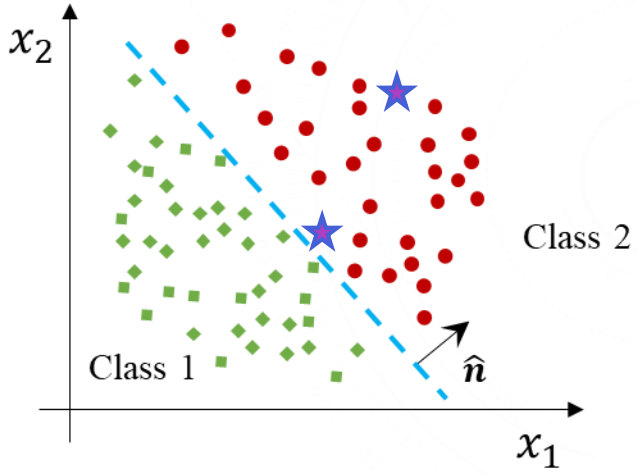
  If $h(x) < 0$, sample belongs to class 1
- However, simply guessing "Class 1" or "Class 2" is pretty crude
  - Both the new samples marked by ★ would be classified as class 2

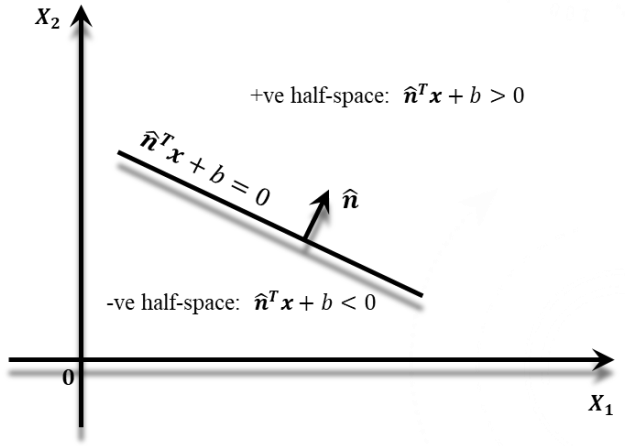




IMPORTANCE OF PROBABILITY

- The probability of a sample belonging to "Class 1" and "Class 2" give a better understanding of the sample's membership to a particular category
- Let p be the probability that the sample belongs to Class 2, the $1 - p$ would be the probability that the sample belongs to Class 1
- Samples close to boundary will have a probability close to 0.5 while those far off from boundary will have a probability close to 0 (if it belongs to class 1) or 1 (if it belongs to class 2)
- Estimating the binary outputs from the probabilities is straight forward through simple thresholding
  - $p > 0.5$ labelled as Class 2 and $p < 0.5$ labelled as Class 1

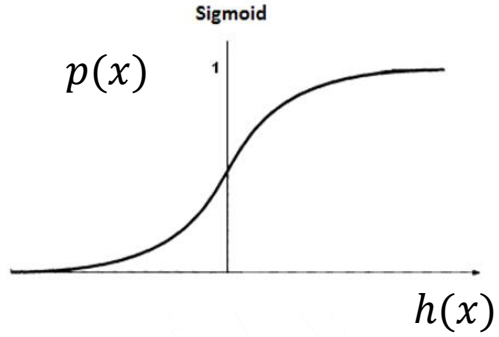




Aim is to convert the discrete class output to a continuous probability output

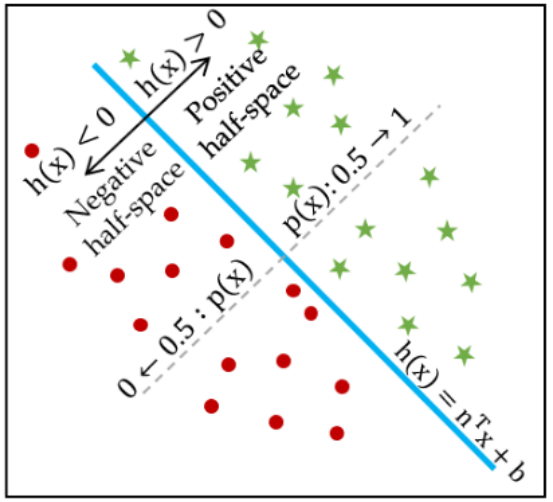
Shades of regression 'Logistic Regression'

LOGISTIC REGRESSION



- How does one model probability that a sample belongs to a specific class?
- A transformation that takes the values of the decision function $h(x)$ to be between 0 and 1 is required
- In Logistic Regression, following transformation is used

$$p(x) = \frac{e^{h(x)}}{1 + e^{h(x)}}$$



$p(x)$ is the probability that the sample belongs to class 2.

$$h(x) = 0 \Rightarrow \ p(x) = 0.5 \quad \text{(On the line)}$$
$$h(x) \ll 0 \Rightarrow \ p(x) \approx 0 \quad \text{(Class 1)}$$
$$h(x) \gg 0 \Rightarrow \ p(x) \approx 1 \quad \text{(Class 2)}$$

- Another viewpoint: Logistic regression can be thought of as modeling the log of odds ratio (probability of class 2 divided by probability of class 1)

$$h(x) = \ln\left(\frac{p(x)}{1 - p(x)}\right)$$

EXAMPLE

Q. For the data given, calculate the probabilities of passing the exam for each sample if a logistic regression model with decision boundary '$h(x) = 1 + 5x_1 - 10x_2 = 0$' is used.

No. of hours studied, x_1	*10.5*	*2.5*	*14*	*8.2*	*10.4*	*5*	*6.7*	*14.7*
No. of classes missed, x_2	*4*	*10*	*15*	*3*	*0*	*0*	*7*	*14*
y (Fail =0, Pass =1)	*1*	*0*	*0*	*1*	*1*	*1*	*0*	*1*

A. For the sample $x_1 = 10.5$ and $x_2 = 4$, $h(x) = 1 + 52.5 - 40 = 13.5$. Hence, $p(x) = \frac{e^{13.5}}{1+e^{13.5}} = 0.999$. Similary, we can evaluate $p(x)$ for all other samples.

No. of hours studied, x_1	*10.5*	*2.5*	*14*	*8.2*	*10.4*	*5*	*6.7*	*14.7*
No. of classes missed, x_2	*4*	*10*	*15*	*3*	*0*	*0*	*7*	*14*
y (Fail =0, Pass =1)	*1*	*0*	*0*	*1*	*1*	*1*	*0*	*1*
$h(x)$	*13.5*	*-86.5*	*-79*	*12*	*53*	*26*	*-35.5*	*-65.5*
$p(x)$	*0.99*	*0*	*0*	*0.99*	*1*	*1*	*0*	3.6×10^{-29}

PARAMETER ESTIMATION (TRAINING)

- Data label available: Class 1 and Class 2
- We do not have the true probabilities to compare the predicted probabilities
- One approach: Assume that $p(x) = 1$ for all samples labelled as Class 2 and $p(x) = 0$ for all samples labelled as Class 1 and run a standard regression type algorithm
  - Does not take cognizance of the notion of probabilities – hence, not preferred
- Need an alternate approach

PARAMETER ESTIMATION (TRAINING)

- Maximum likelihood estimation (MLE)
  - Find a decision boundary such that $p(x)$ for samples belonging to class 2 are maximized while $1 - p(x)$ for samples belonging to class 1 are maximized

$$\max_{n,b} L = \prod_{i=1}^{k} \left(p\left(x^i\right)\right)^{y^i} \left(1 - p\left(x^i\right)\right)^{(1-y^i)}$$

$y^i = 0$ for samples in Class 1 and $y^i = 1$ for samples in Class 2

  - For samples in Class 1, the term in L would be $1 - p\left(x^i\right)$ and for samples in Class 2, the term in L would be $p\left(x^i\right)$
  - Maximizing L is equivalent to maximizing $Log\ L$

$$\max_{n,b} \quad \log L = \sum_{i=1}^{k} y^i \log p\left(x^i\right) + \left(1 - y^i\right) \log\left(1 - p\left(x^i\right)\right)$$

  - Unconstrained nonlinear programming problem – solve using available nonlinear optimization solvers to get n and b

EXTENSION TO NONLINEAR PROBLEMS

- Logistic regression can be extended to nonlinear problems by changing $h(x)$
- Polynomial boundaries instead of linear boundary

$$h(x) = x^T Q x + n^T x + b$$

- Other aspects remains the same as before

$$p(x) = \frac{e^{h(x)}}{1 + e^{h(x)}}$$

REGULARIZATION

- Over-fitting can be avoided using regularization
- Increasing the number of parameters is penalized
- Achieved by modifying objective function

$$\max_{\theta} \; Log\, L \; - C \, ||\theta||$$

- C is a constant that can be used to increase or decrease the penalty
- Norm ($||\,.\,||$) can be chosen to be any p-norm, however, 1-norm and 2-norm are most commonly used

MULTI-CLASS PROBLEMS

- Logistic regression can be extended to multi-class classification problems by solving multiple binary classification problems
- One-versus-rest approach
- A decision logic is used to process the results of binary classification
- Example:

Training set

Sepal Length (x_1)	5.1	4.9	4.7	4.6	4.8	6.8	6.7	6.7	6.3	5.9	5.3	5.5
Petal Length (x_2)	2.4	1.4	1.3	1.5	1.3	5.9	5.7	5.2	5	3.1	2.9	3.4
Species (y)	2	1	1	1	1	0	0	0	0	2	2	2

Test set

Sample	1	2	3	4
Sepal Length (x_1)	4.5	6.2	5	5.2
Petal Length (x_2)	1.2	5.4	3.4	2.2
Species (y)	1	0	2	2

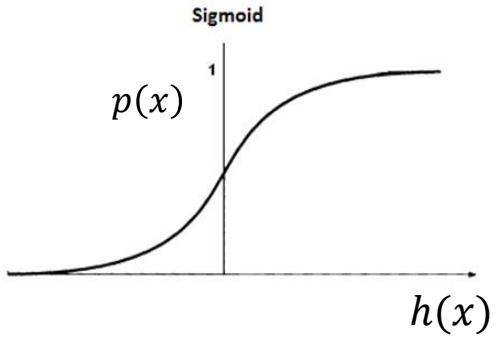
Classifier 1 – Class 0 vs (Class 1 & 2)
Classifier 2 – Class 1 vs (Class 0 & 2)
Classifier 3 – Class 2 vs (Class 1 & 0)

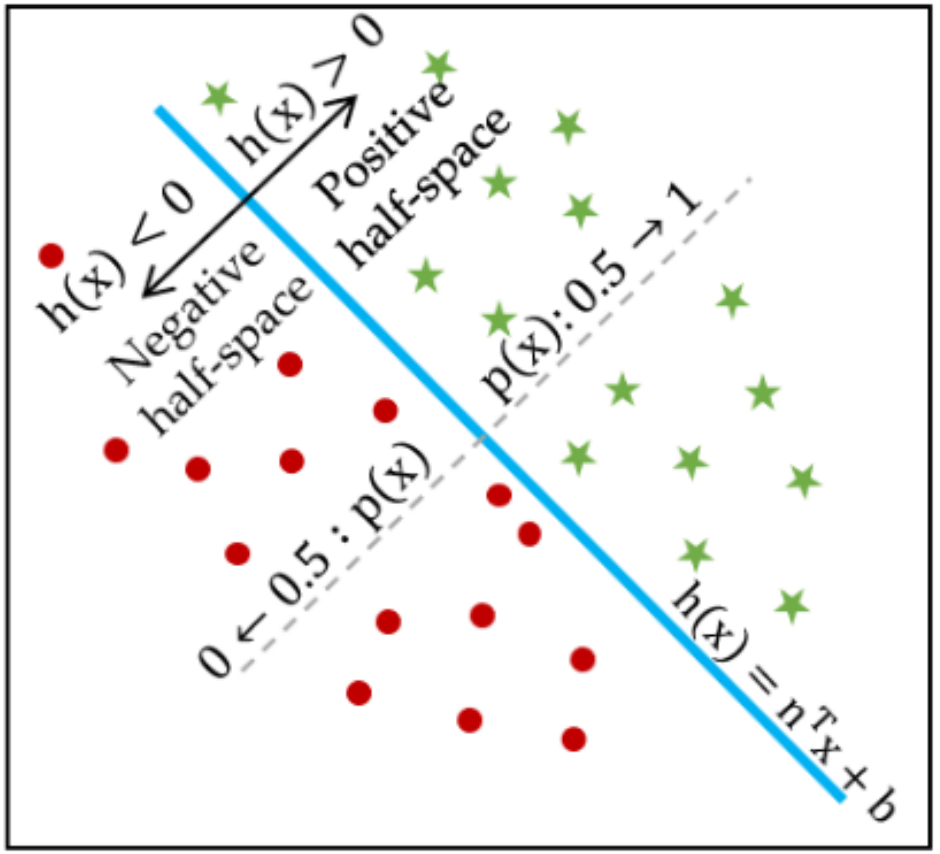
		Predicted Probability			
Sample	y_{true}	*Class 0*	*Class 1*	*Class 2*	y_{pred}
1	1	0.0061	0.667	0.3268	1
2	0	0.7493	0.005	0.2449	0
3	2	0.2351	0.240	0.524	2
4	2	0.047	0.529	0.422	1

CONCLUSIONS

- Logistic regression for supervised binary classification
- Parameter estimation using MLE
- Extension to multi-class classification

$$p(x) = \frac{e^{h(x)}}{1 + e^{h(x)}}$$





UNSUPERVISED LEARNING
K-MEANS CLUSTERING

RESMI SURESH

ASSISTANT PROFESSOR, IIT GUWAHATI

CLASSIFICATION

Examples:

Fraud detection in credit card transactions

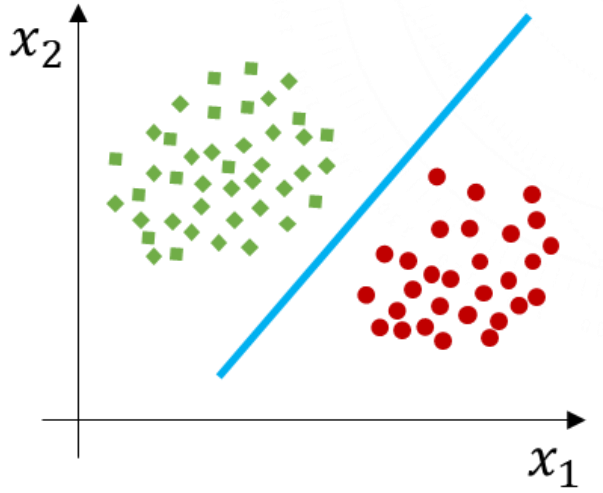
Distinguishing objects – "Self-driving cars"

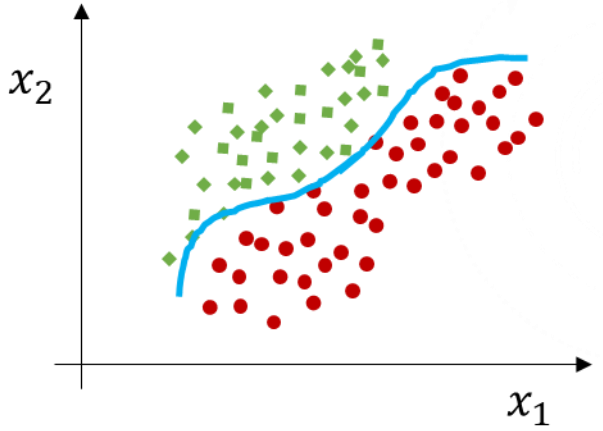
Detecting failures in built systems/equipment

Classifying emails as spam or genuine

Techniques:

Logistic regression, k-nearest neighbors, Neural network, Decision tree, Random forest, Support vector machines, LDA, QDA, Naïve Bayes, Hierarchical clustering, k-means clustering, …





CLUSTERING TECHNIQUES

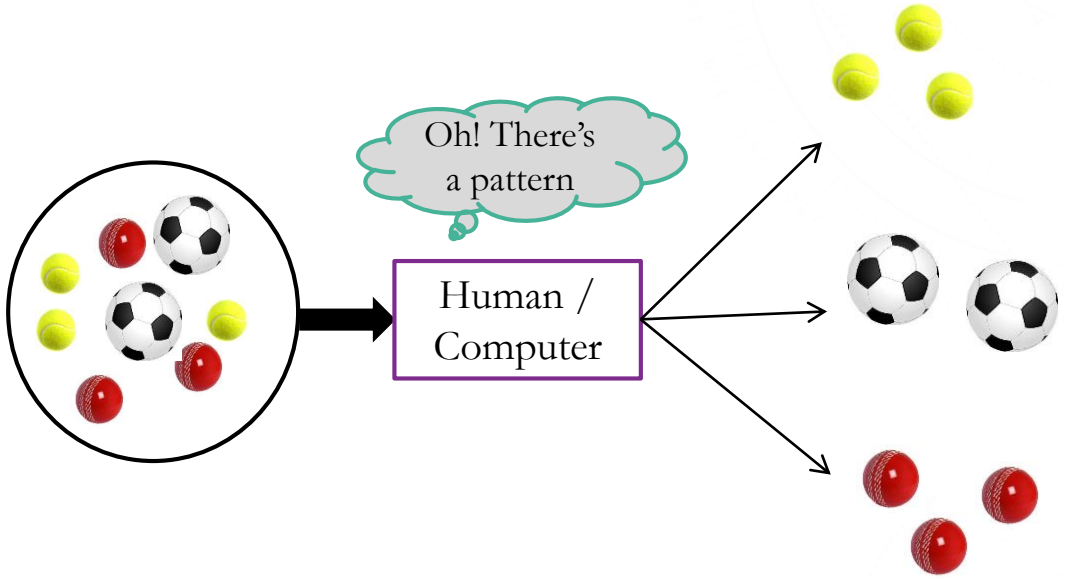
Unsupervised techniques

Technique is presented with unlabeled data and the intent is to group data so that some meaningful insights can be derived

Dissimilarity in terms of classes cannot be used to classify data unlike classification algorithms

Clustering techniques group datapoints by looking at the similarity between them

- Similarity-based learning techniques/unsupervised techniques
- Definition of similarity becomes critical



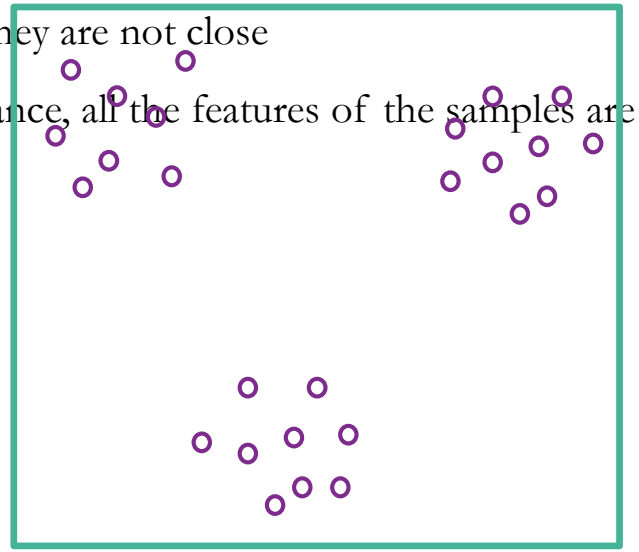
Goal is to partition datapoints into different clusters in a meaningful manner

K-MEANS CLUSTERING

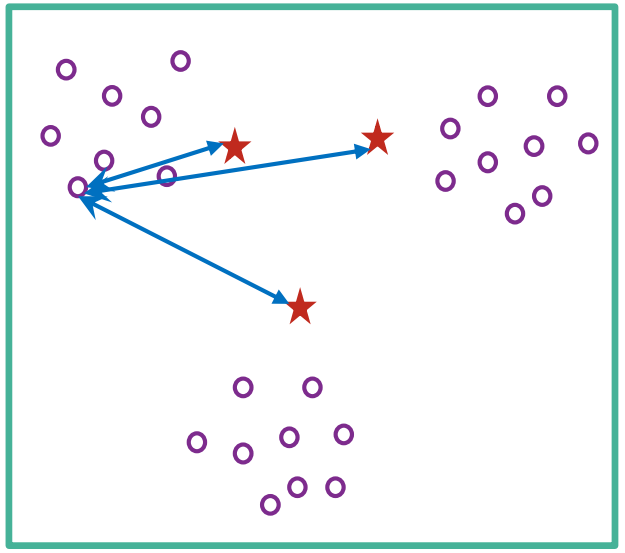
K-MEANS CLUSTERING

Similarity measure based on distance between two samples

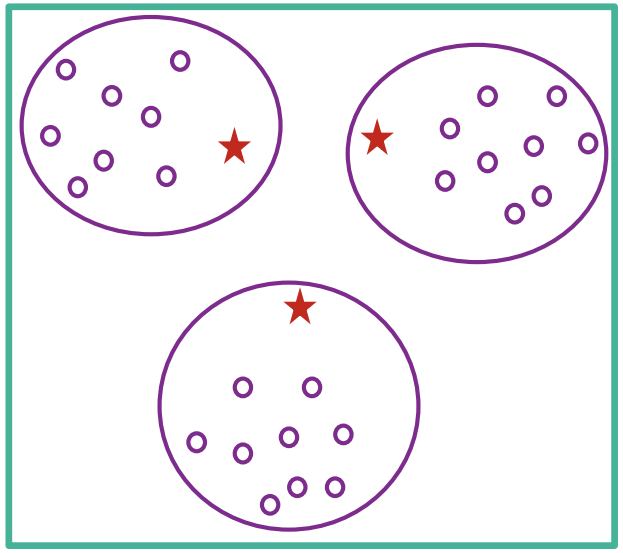
- If two samples are close to each other, they are similar and dissimilar if they are not close
- Multivariate scenario: if two samples are close together in Euclidean distance, all the features of the samples are close to each other, implying the samples are very similar
- Maximizing similarity equivalent to minimizing distance



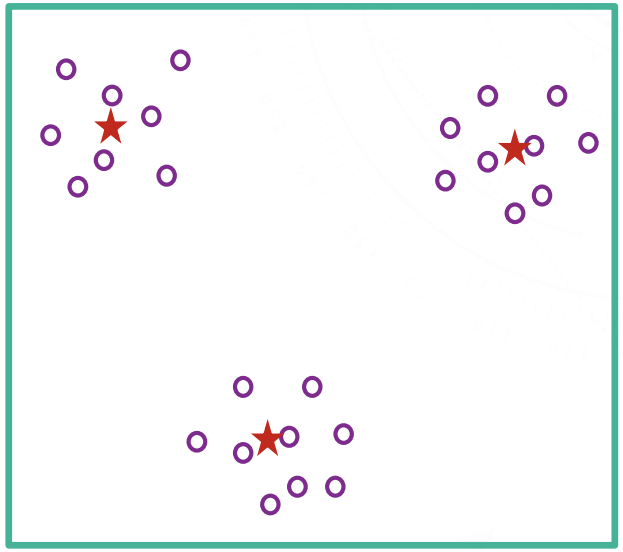
K-MEANS CLUSTERING



1. Randomly initialize the means of k clusters
2. Find the distances between all the samples and cluster centers



3. (Re)Assign samples to their closest cluster centers



4. Recompute centers as the mean of all samples assigned to that cluster

5. Repeat steps 2-4 till the cluster centers do not change and/or there are no reassignments

EXAMPLE

Consider the dataset containing the sepal length and petal length of 3 types of Iris species

Sepal length x_1	5.1	4.9	4.7	4.6	4.8	6.8	6.7	6.7	6.3	5.9	5.3	5.5
Petal length x_2	2.4	1.4	1.3	1.5	1.3	5.9	5.7	5.2	5	3.1	2.9	3.4

Solution:

Given 3 clusters. Initialize cluster centers as $C_1 = (4.7,1.3), C_2 = (5.9,3.1)$, and $C_3 = (4.9,1.4)$

Iteration 1: Compute Euclidean distance from all datapoints to these 3 centroids

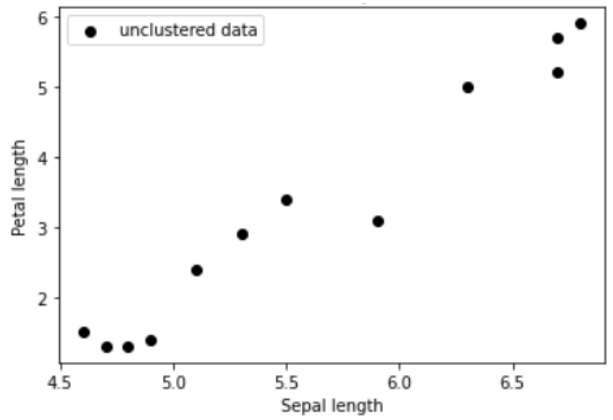
For sample $(5.1,2.4)$,

Distance from C_1, $d_1 = \sqrt{(5.1-4.7)^2 + (2.4-1.3)^2} = 1.17$

Distance from C_2, $d_2 = \sqrt{(5.1-5.9)^2 + (2.4-3.1)^2} = 1.06$

Distance from 3, $d_3 = \sqrt{(5.1-4.9)^2 + (2.4-1.4)^2} = 1.02$

The shortest distance among d_1, d_2 and d_3 is d_3. Hence, sample 1 is assigned to cluster 3.



Sepal length x_1	*5.1*	*4.9*	*4.7*	*4.6*	*4.8*	*6.8*	*6.7*	*6.7*	*6.3*	*5.9*	*5.3*	*5.5*
Petal length x_2	*2.4*	*1.4*	*1.3*	*1.5*	*1.3*	*5.9*	*5.7*	*5.2*	*5*	*3.1*	*2.9*	*3.4*

EXAMPLE

Iteration 1 (Contd.)

Repeating for all sample, we get the following clusters

Cluster 1	Cluster 2	Cluster 3
[4.7, 1.3] [4.6, 1.5] [4.8, 1.3]	[6.8, 5.9] [6.7, 5.7] [6.7, 5.2] [6.3, 5] [5.9, 3.1] [5.3, 2.9] [5.5, 3.4]	[5.1, 2.4] [4.9, 1.4]
[4.7, 1.37]	**[6.17, 4.45]**	**[5, 1.9]**

Average of all samples in a cluster

New cluster centers

Iteration 2: Reassign samples based on the new cluster centers

Sepal length x_1	*5.1*	*4.9*	*4.7*	*4.6*	*4.8*	*6.8*	*6.7*	*6.7*	*6.3*	*5.9*	*5.3*	*5.5*
Petal length x_2	*2.4*	*1.4*	*1.3*	*1.5*	*1.3*	*5.9*	*5.7*	*5.2*	*5*	*3.1*	*2.9*	*3.4*

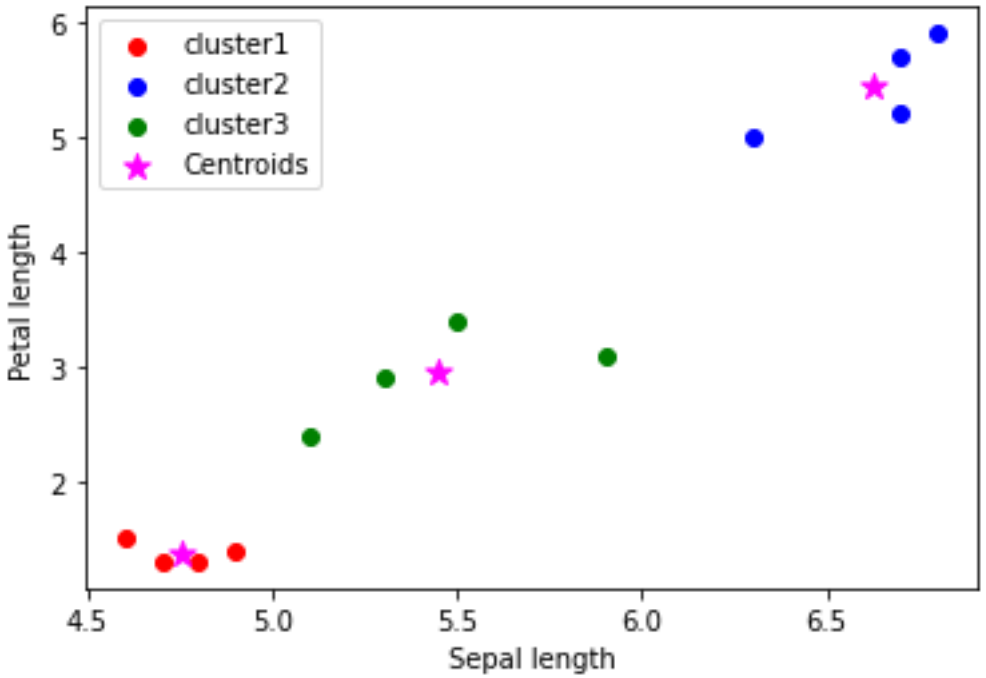
EXAMPLE

Algorithm converged in 4 iterations

Final centroids after 100 iterations, we get the following cluster centers

C_1	C_2	C_3
[4.75, 1.375]	[6.625, 5.45]	[5.45, 2.95]

Samples that are close to each other are assigned to the same cluster



K-MEANS CLUSTERING

OPTIMIZATION PROBLEM

Optimization perspective of the k-means algorithm is that the algorithm is attempting to minimize the total sum of squares or within cluster sum of squares

$$\min_{C_1, C_2, \ldots C_k} \sum_{j=1}^{k} \sum_{i \in S_j} \left|\left| x^i - C_j \right|\right|^2$$

S_j - set of sample indices belonging to cluster j

C_j - cluster center for j^{th} cluster

K-means algorithm can only identify locally optimum solutions

K-MEANS CLUSTERING

INITIALIZATION ISSUES

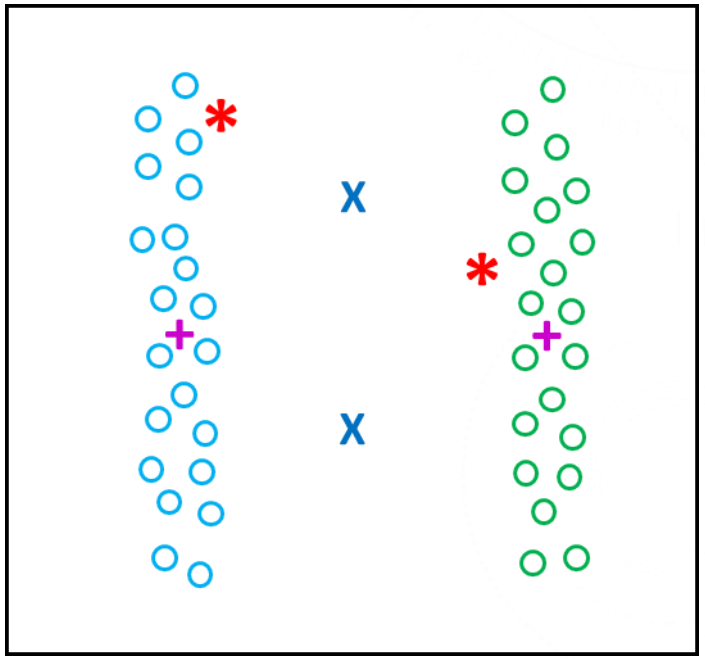
Final result of the algorithm will depend on the initialization

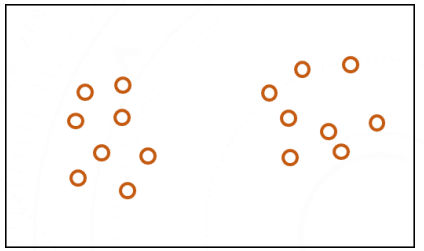
If one starts with 2 cluster centers marked by 'X', then,

- Samples on the upper half will belong to one class and samples on the lower half will belong to another class
- The cluster centers are likely to move very little and converge to the same location

How to avoid the pitfalls of local solutions?

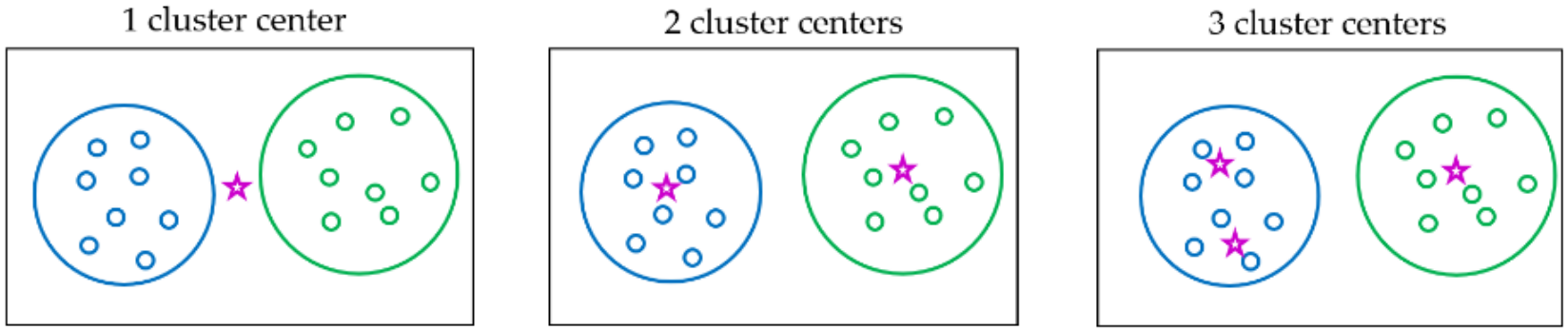
- Attempting different cluster initializations
  - If one starts with 2 cluster centers marked by '+' or '*', then the algorithm will be able to find the correct clusters
- Through stochastic optimization





K-MEANS CLUSTERING

OPTIMAL NUMBER OF CLUSTERS (KNEE/ELBOW PLOT)

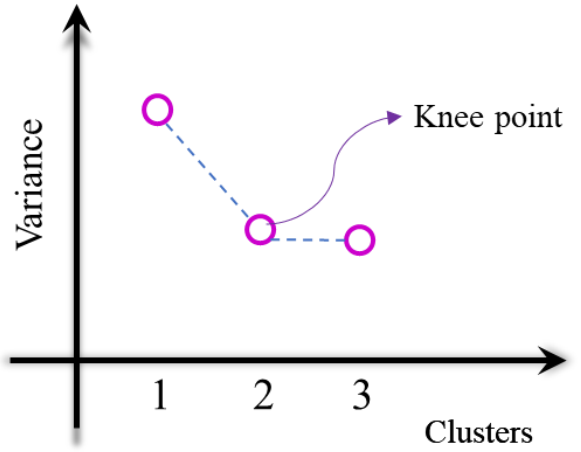


$$\min_{C_1, C_2, \ldots C_k} \sum_{j=1}^{k} \sum_{i \in S_j} \left|\left| x^i - C_j \right|\right|^2$$

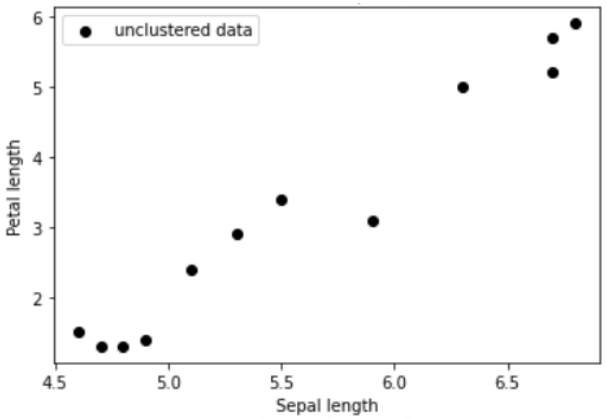
$k = m \Rightarrow$ objective function = 0

$k = 1 \Rightarrow$highest objective function value

m – total number of samples



EXAMPLE



Choose the optimal number of clusters for the given data.

Sepal length x_1	5.1	4.9	4.7	4.6	4.8	6.8	6.7	6.7	6.3	5.9	5.3	5.5
Petal length x_2	2.4	1.4	1.3	1.5	1.3	5.9	5.7	5.2	5	3.1	2.9	3.4

Solution:

K-means clustering is performed for different number of clusters ($k \in [1, 8]$)

Initialization: Choosing the first k cluster centers from the following list

C_i	1	2	3	4	5	6	7	8
x_1	4.7	5.9	4.9	4.8	5.3	4.6	6.7	6.8
x_2	1.3	3.1	1.4	1.3	2.9	1.5	5.7	5.9

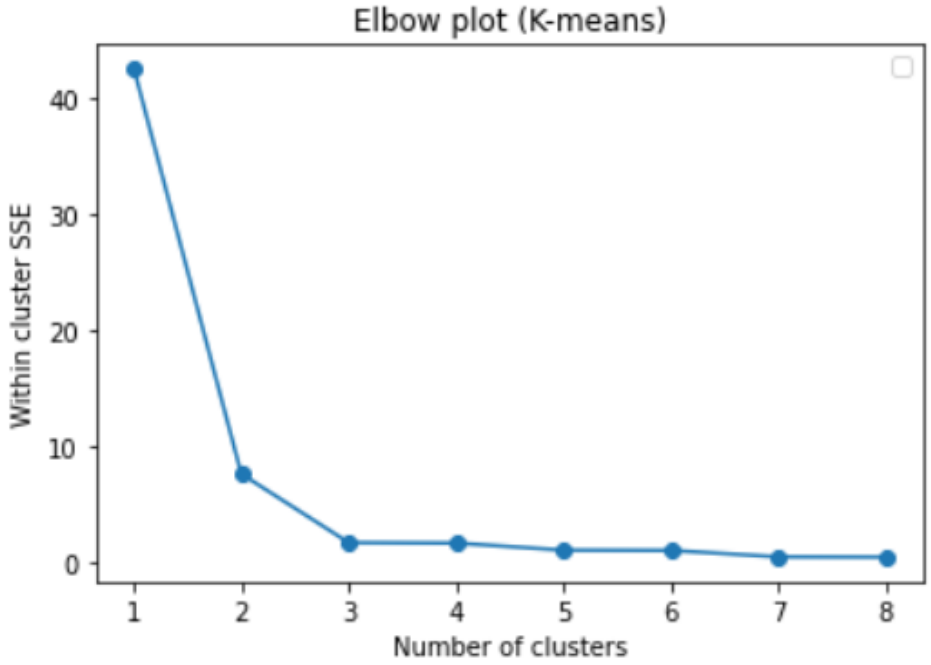
Evaluated within cluster sum square value

$$wcss = \sum_{j=1}^{k} \sum_{i \in S_j} (x^i - C_j)^2$$

EXAMPLE

For all $k > 3$, marginal decrease in $wcss$ for increase in k

Optimal number of clusters based on the elbow plot is 3



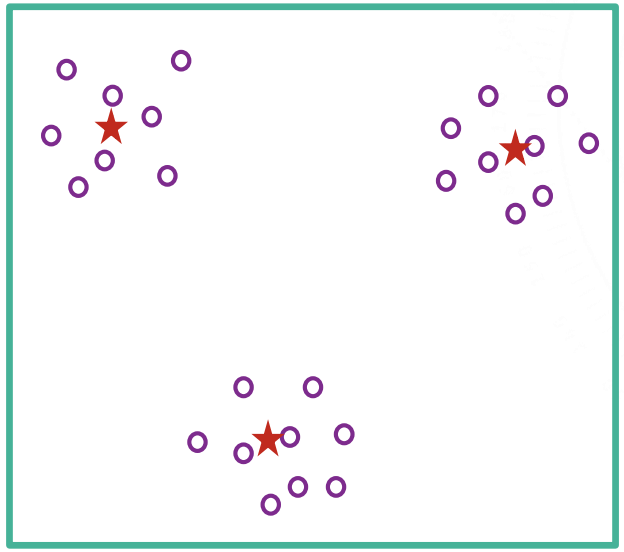
CONCLUSIONS

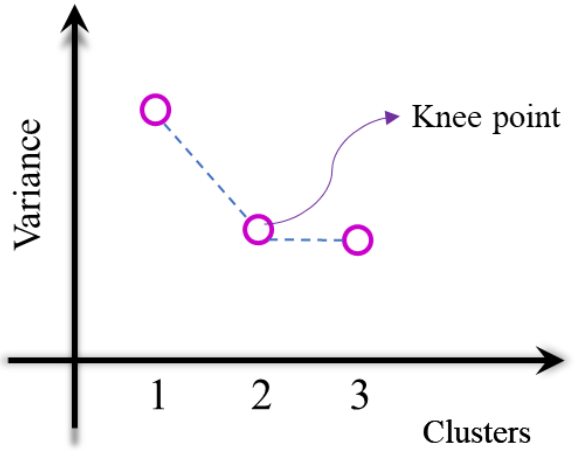
Unsupervised learning

Clustering techniques

K-Means clustering

- Optimization problem
- Initialization issues
- Optimal number of clusters






operation == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True
selection at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.acti
("Selected" + str(modifier
irror_ob.select = 0
bpy.context.selected_ob
ata.objects[one.name].se
int("please select exact

THANK YOU