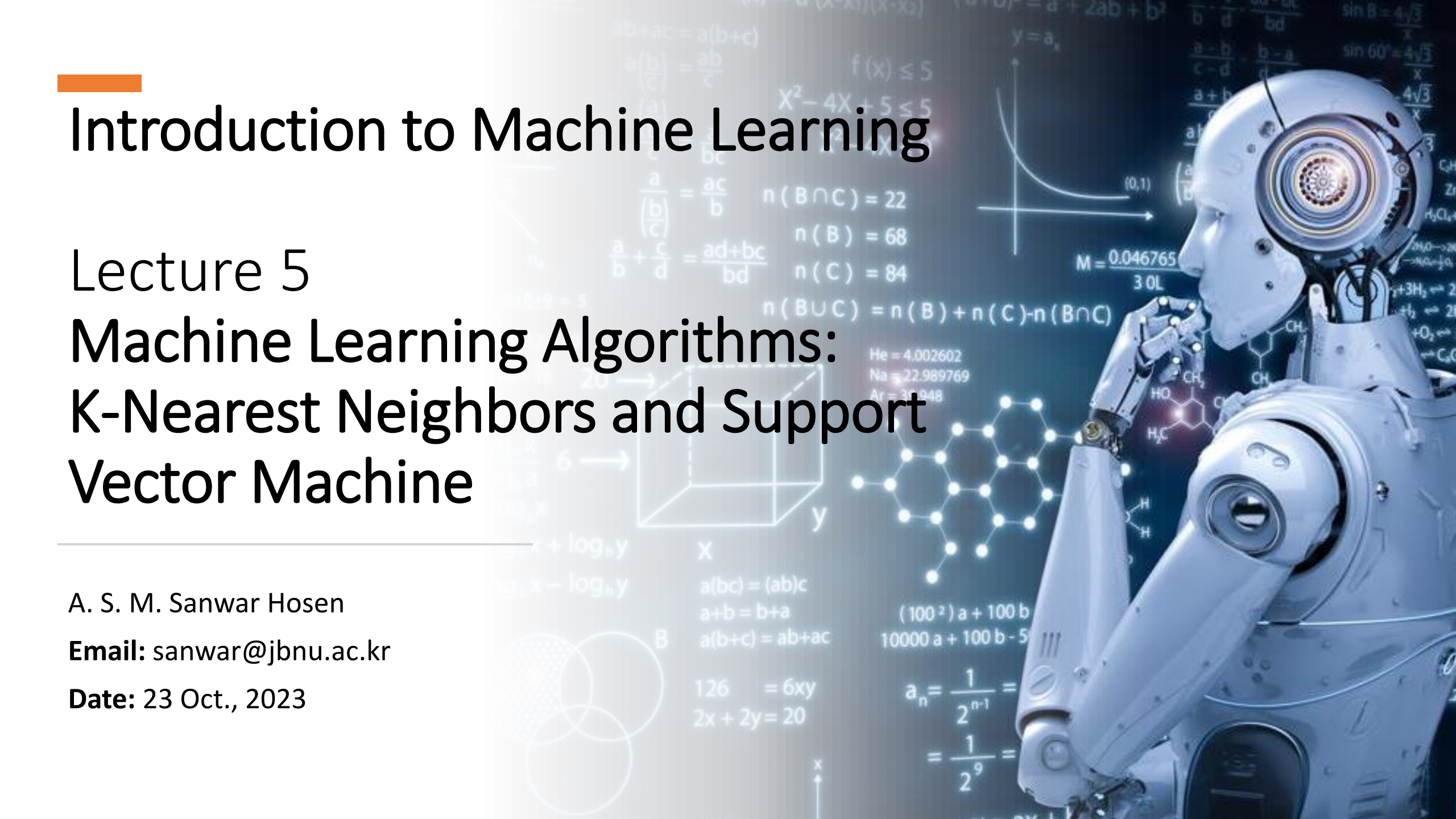
Introduction to Machine Learning

Lecture 5
Machine Learning Algorithms: K-Nearest Neighbors and Support Vector Machine

A. S. M. Sanwar Hosen

Email: sanwar@jbnu.ac.kr

Date: 23 Oct., 2023



Machine Learning Algorithms: KNN and SVM

❑ Supervised Machine Learning (ML) Algorithms

Some of the ML algorithms are:

- ✓ K-Nearest Neighbors (KNN)
- ✓ Support Vector Machine (SVM)
- ✓ Decision Tree (DT)
- ✓ Random Forest (RF)
- ✓ Gaussian Naïve Bayes
- ✓ Linear Regression (LR)

Machine Learning Algorithms: KNN and SVM

❑ K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm assumes similar things are near to each other. It stores all the available data and classifies a new data point based on the similarity. This means when a new data appears then it can be easily classified into a well-suite category by using K-NN, where K is the number of the nearest neighbors among the actual data.

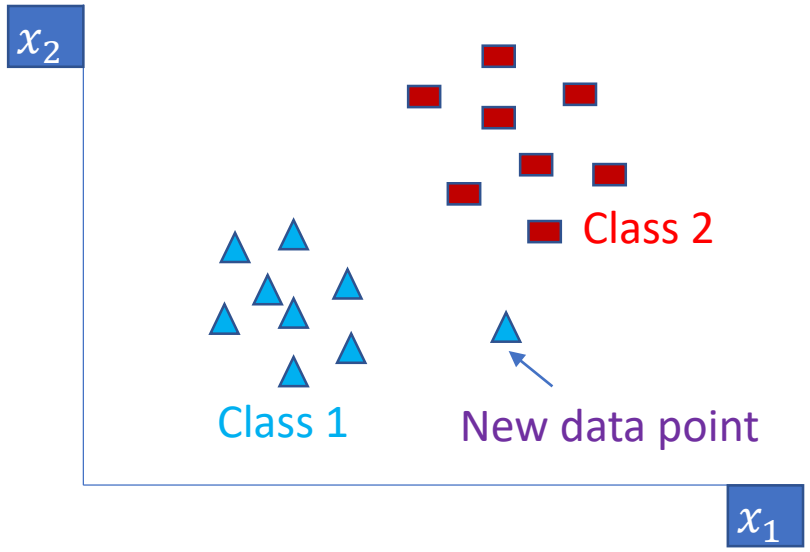


Fig: KNN algorithm working visualization.

Intuition Behind KNN: By visualizing the data points on a graph, we have the opportunity to identify potential clusters or groupings. Consequently, when confronted with an unclassified point, we can determine its group affiliation by examining the groups to which its closest neighbors belong. Essentially, if a point is in close proximity to a cluster of points categorized as 'triangle,' there is a greater likelihood that it will be categorized as 'triangle' rather than 'rectangle.'

❑ Distance Metrics Used in KNN

KNN algorithm helps to identify the nearest points or the groups for a query point. Different distance metrics are used to determine the closest groups or the nearest points for a query point as follows:

- ✓ Euclidean Distance
- ✓ Manhattan Distance
- ✓ Minkowski Distance
- ✓ Hamming Distance

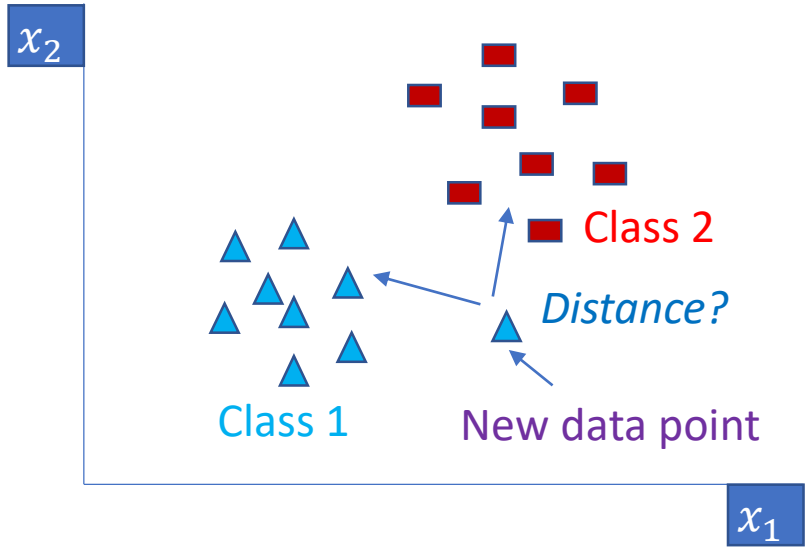


Fig: KNN algorithm working visualization.

Machine Learning Algorithms: KNN and SVM

❑ Euclidean Distance based KNN

Euclidean Distance: It measures the cartesian distance between two points which are in a plane or hyperplane. It is calculated as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

❑ How it Works?

The steps are as follows:

Step 1: Find the Euclidian distance d of the observed value and actual values in the samples.

Distance, $d = \sqrt{(XO_0 - XA_0)^2 + (XO_1 - XA_1)^2}$

Where $XO = \{XO_0, XO_1, \dots, XO_n\}$ is the observed value and $XA = \{XA_0, XA_1, \dots, XA_n\}$ is the actual value in the samples.

Step 2: Find the K nearest neighbors in the samples among distances derived.

Step 3: Find the highest probability of the results (output) belong to the sample of actual data. Probability,

$$P\ (class\ 1) = \frac{no.of\ class\ 1}{K} \text{ and } P\ (class\ 0) = \frac{no.of\ class\ 0}{K}$$

Step 4: The probability value is dominating, the given query is fall into the class.

Machine Learning Algorithms: KNN and SVM

❑ Euclidean Distance based KNN (cont..)

Example: Given a query $XO = (math\ (XO_0) = 6, AI(XO_1) = 8$ out of 10) and $K = 3$, nearest neighbors Classification Pass/Fail (1 or 0).

Dataset (Actual): Student results

ID	Math (XA_0)	AI (XA_1)	Result (Output)
1	4	3	F
2	6	7	P
3	7	8	P
4	5	5	F
5	8	8	P

KNN Algorithm: Classification using Euclidean distance based KNN in supervised learning

Step 1: Find the Euclidian distance d of the observed value XO, and actual values XA in the samples.

$$d_1 = \sqrt{(XO_0 - XA_0)^2 + (XO_1 - XA_1)^2} = \sqrt{(6-4)^2 + (8-3)^2} = \sqrt{29} = 5.38$$
$$d_2 = \sqrt{(6-6)^2 + (8-7)^2} = \sqrt{1} = 1$$
$$d_3 = \sqrt{(6-7)^2 + (8-8)^2} = \sqrt{1} = 1$$
$$d_4 = \sqrt{(6-5)^2 + (8-5)^2} = \sqrt{10} = 3.16$$
$$d_5 = \sqrt{(6-8)^2 + (8-8)^2} = \sqrt{4} = 2$$

Step 2: Find the nearest $K = 3$ neighbors among distances derived $(d_1, d_2, d_3, d_4, d_5) = 1, 1, 2$ belong to IDs $2, 3$ and 5 in the actual data.

Step 3: Find the highest probability of the results (output) belong to the sample of actual data corresponding to the IDs, $ID2 = P$, $ID3 = P$, $ID5 = P$, so there are $P + P + P = 3P$ (Pass) and there is no F(Fail) classes.

Step 4: The P(Pass) value is dominating ($Pass\ probability = \frac{3}{3} = 1$ or 100%) and $Faill\ probability = \frac{0}{3} = 0$ or 0%), so the given query is fall into P class.

Machine Learning Algorithms: KNN and SVM

❑ Manhattan Distance based KNN

Manhattan Distance: This metric is determined by adding up the absolute differences between the coordinates of points in n-dimensional space. It is calculated as:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

❑ How it Works?

The steps are as follows:

Step 1: Find the Manhattan distance d of the observed value and actual values in the samples.

Distance, $d = |(XO_0 - XA_0)| + |(XO_1 - XA_1)|$

Where $XO = \{XO_0, XO_1, \dots, XO_n\}$ is the observed value and $XA = \{XA_0, XA_1, \dots, XA_n\}$ is the actual value in the samples.

Step 2: Find the K nearest neighbors in the samples among distances derived.

Step 3: Find the highest probability of the results (output) belong to the sample of actual data. Probability,

$$P\ (class\ 1) = \frac{no.of\ class\ 1}{K} \text{ and } P\ (class\ 0) = \frac{no.of\ class\ 0}{K}$$

Step 4: The probability value is dominating, the given query is fall into the class.

Machine Learning Algorithms: KNN and SVM

❑ Manhattan Distance based KNN (cont..)

Example: Given a query $XO = (math\ (XO_0) = 6, AI(XO_1) = 8$ out of 10) and $K = 3$, nearest neighbors Classification Pass/Fail (1 or 0).

Dataset (Actual): Student results

ID	Math (XA_0)	AI (XA_1)	Result (Output)
1	4	3	F
2	6	7	P
3	7	8	P
4	5	5	F
5	8	8	P

KNN Algorithm: Classification using Manhattan distance based KNN in supervised learning

Step 1: Find the Manhattan distance d of the observed value XO, and actual values XA in the samples.

$d_1 = |(XO_0 - XA_0)| + |(XO_1 - XA_1)| = |(6-4)| + |(8-3)| = 7$
$d_2 = |(6-6)| + |(8-7)| = 1$
$d_3 = |(6-7)| + |(8-8)| = 1$
$d_4 = |(6-5)| + |(8-5)| = 4$
$d_5 = |(6-8)| + |(8-8)| = 2$

Step 2: Find the nearest $K = 3$ neighbors among distances derived $(d_1, d_2, d_3, d_4, d_5) = 1, 1, 2$ belong to IDs $2, 3$ and 5 in the actual data.

Step 3: Find the highest probability of the results (output) belong to the sample of actual data corresponding to the IDs, $ID2 = P$, $ID3 = P$, $ID5 = P$, so there are $P + P + P = 3P$ (Pass) and there is no F(Fail) classes.

Step 4: The P(Pass) value is dominating ($Pass\ probability = \frac{3}{3} = 1$ or 100%) and $Faill\ probability = \frac{0}{3} = 0$ or 0%), so the given query is fall into P class.

❑ Minkowski Distance based KNN

Minkowski Distance: This metric used to measure the distance between two points in n-dimensional space. Essentially, it serves as a generalization of both the Euclidean distance and the Manhattan distance. It is calculated as:

$$d(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$$

Where p is the number of orders $(p \in \{1,2, \dots, n\})$. If $p = 2$ then it is the same as the formula for the Euclidean distance, and when $p = 1$, it provides the formula for the Manhattan distance.

❑ How it Works?

The steps are as follows:

Step 1: Find the Minkowski distance d of the observed value and actual values in the samples, where given $p = 1$.

Distance, $d = |(XO_0 - XA_0)| + |(XO_1 - XA_1)|$, formula for the Manhattan distance

Where $XO = \{XO_0, XO_1, \dots, XO_n\}$ is the observed value and $XA = \{XA_0, XA_1, \dots, XA_n\}$ is the actual value in the samples.

Step 2: Find the K nearest neighbors in the samples among distances derived.

Step 3: Find the highest probability of the results (output) belong to the sample of actual data. Probability,

$$P\ (class\ 1) = \frac{no.of\ class\ 1}{K} \text{ and } P\ (class\ 0) = \frac{no.of\ class\ 0}{K}$$

Step 4: The probability value is dominating, the given query is fall into the class.

❑ Minkowski Distance based KNN (cont..)

Example: Given a query $XO = (math\ (XO_0) = 6, AI(XO_1) = 8$ out of 10) and $K = 3$, nearest neighbors Classification Pass/Fail (1 or 0).

Dataset (Actual): Student results

ID	Math (XA_0)	AI (XA_1)	Result (Output)
1	4	3	F
2	6	7	P
3	7	8	P
4	5	5	F
5	8	8	P

KNN Algorithm: Classification using Minkowski distance based KNN in supervised learning

Step 1: Find the Minkowski distance d of the observed value XO, and actual values XA in the samples for $p = 1$.

$$d_1 = |(XO_0 - XA_0)| + |(XO_1 - XA_1)| = |(6-4)| + |(8-3)| = 7$$
$$d_2 = |(6-6)| + |(8-7)| = 1$$
$$d_3 = |(6-7)| + |(8-8)| = 1$$
$$d_4 = |(6-5)| + |(8-5)| = 4$$
$$d_5 = |(6-8)| + |(8-8)| = 2$$

Step 2: Find the nearest $K = 3$ neighbors among distances derived $(d_1, d_2, d_3, d_4, d_5) = 1, 1, 2$ belong to IDs 2, 3 and 5 in the actual data.

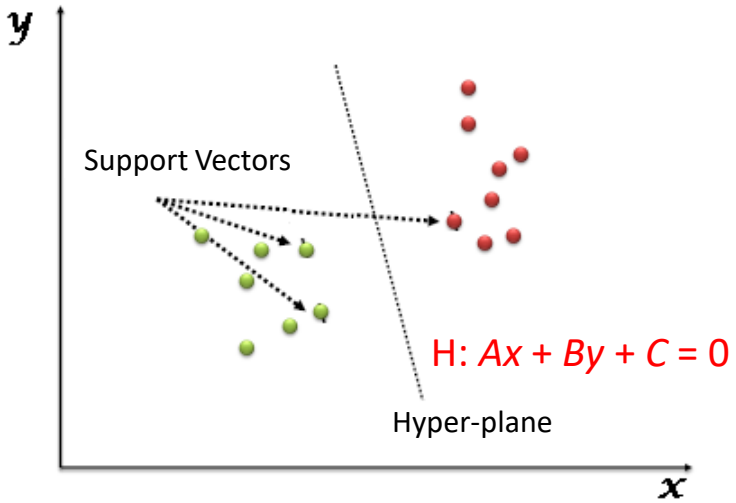
Step 3: Find the highest probability of the results (output) belong to the sample of actual data corresponding to the IDs, $ID2 = P$, $ID3 = P$, $ID5 = P$, so there are $P + P + P = 3P$ (Pass) and there is no F(Fail) classes.

Step 4: The P(Pass) value is dominating ($Pass\ probability = \frac{3}{3} = 1$ or 100%) and $Faill\ probability = \frac{0}{3} = 0$ or 0%), so the given query is fall into P class.

Machine Learning Algorithms: KNN and SVM

❑ Support Vector Machine

Support Vector Machine (SVM) is a supervised ML algorithm that can be used for both classification and regression problems. The SVM algorithm plots each data item as a point in n-dimensional space (where n is the number of features) with the value of each feature being the value of a particular coordinate. Then it performs the classification by finding the hyper-plane that differentiates two classes.



- ✓ Support Vectors: The coordinates of individual observation.
- ✓ Hyper-plane: A line between the data points (coordinated) that partitions/segregates the data points into two classes.

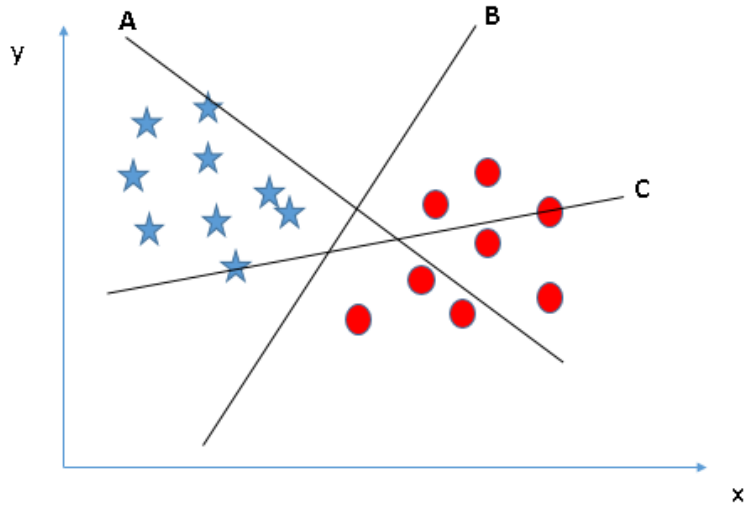
Machine Learning Algorithms: KNN and SVM

❑ Support Vector Machine (cont..)

How it works?

We know the process of segregating the two classes with a hyper-plane. Now the issue is how we can identify the right hyper-plane?

✓ Identify the right hyper-plane (scenario 1): Here, we have three hyper-planes (A, B, and C). How can we Identify the right hyper-plane to classify the stars and circles?



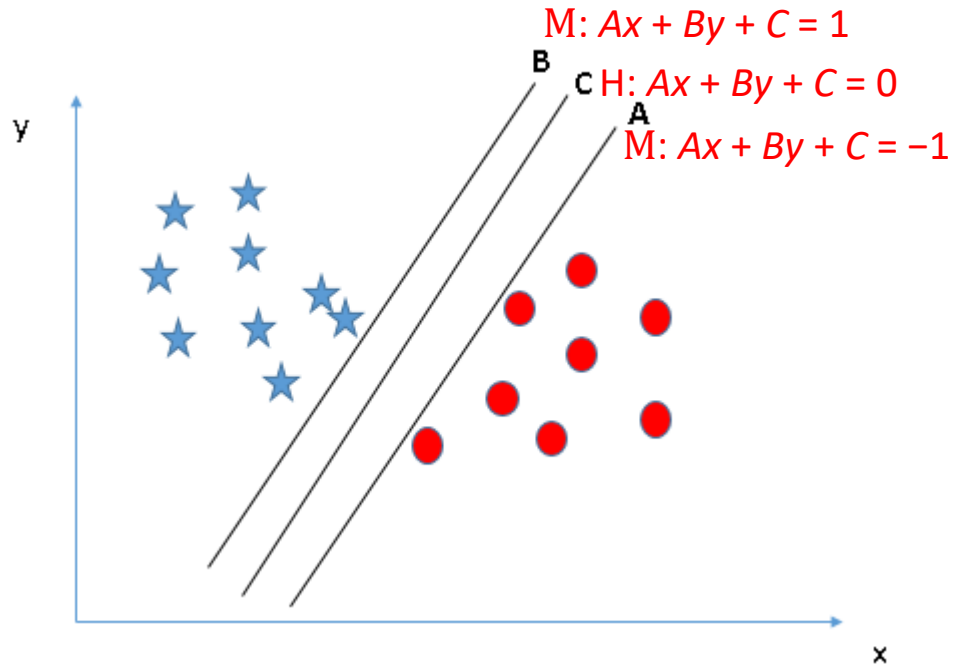
Solution: Select the right hyper-plane which segregates the two classes better.

Machine Learning Algorithms: KNN and SVM

❑ Support Vector Machine (cont..)

How it works?

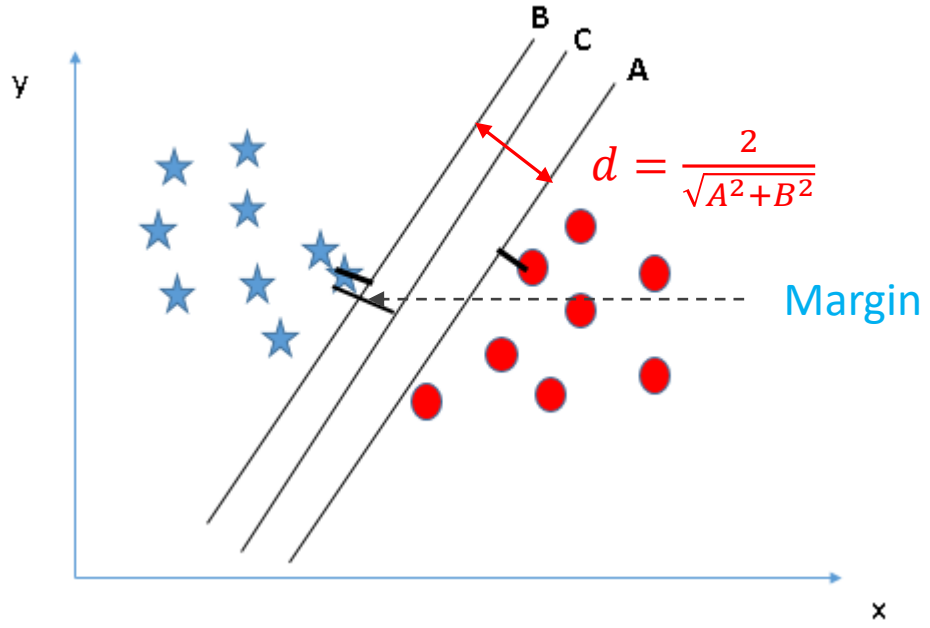
✓ Identify the right hyper-plane (scenario 2): Here, we have three hyper-planes (A, B, and C) and all are segregating the classes well. Now, how can we identify the right hyper-plane?



Solution: Here, maximizing the distances nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called Margin. Let's look at the next slide.

Machine Learning Algorithms: KNN and SVM

❑ Support Vector Machine (cont..)



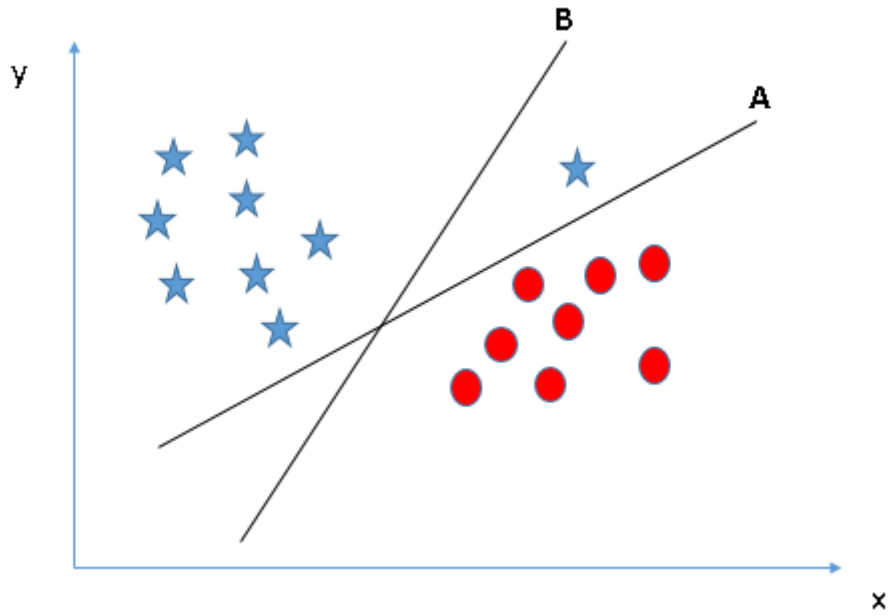
Above, we can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane is C. If we select a hyper-plane having low margin, then there is high chance of miss-classification.

Machine Learning Algorithms: KNN and SVM

❑ Support Vector Machine (cont..)

How it works?

✓ Identify the right hyper-plane (scenario 3): Here, we have two hyper-planes (*A* and *B*). Now, how can we identify the right hyper-plane?



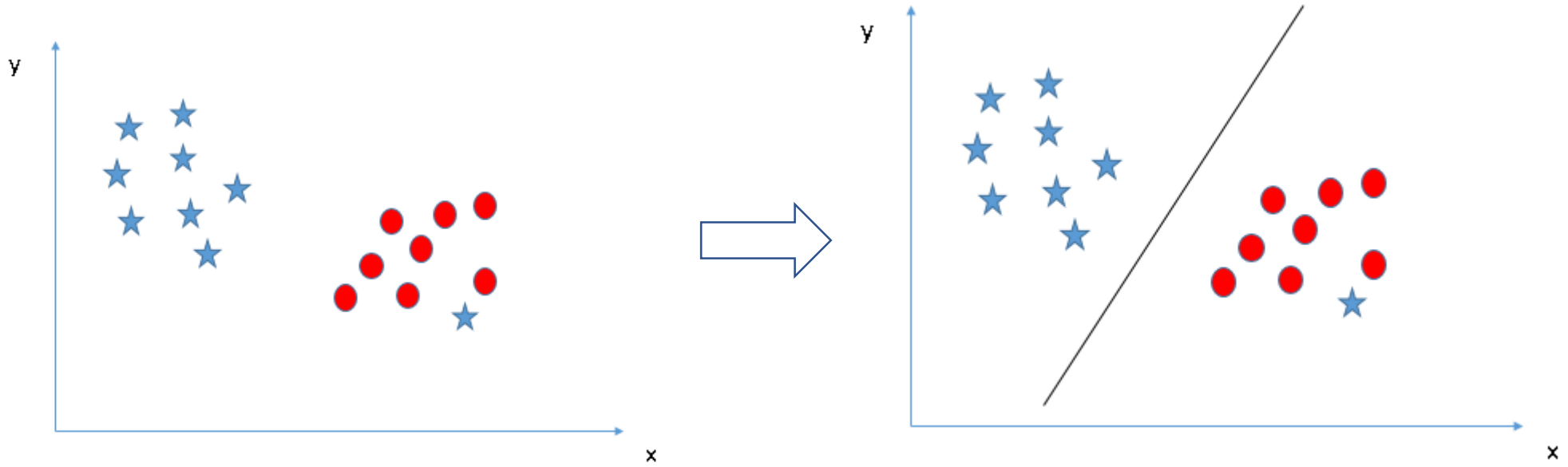
Hint: Use the rules as discussed in previous slides to identify the right hyper-plane.

Machine Learning Algorithms: KNN and SVM

❑ Support Vector Machine (cont..)

How it works?

✓ Identify the right hyper-plane (scenario 4): In the following scenario, we are unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other (circle) class as an outlier. Then, how can we identify the right hyper-plane?



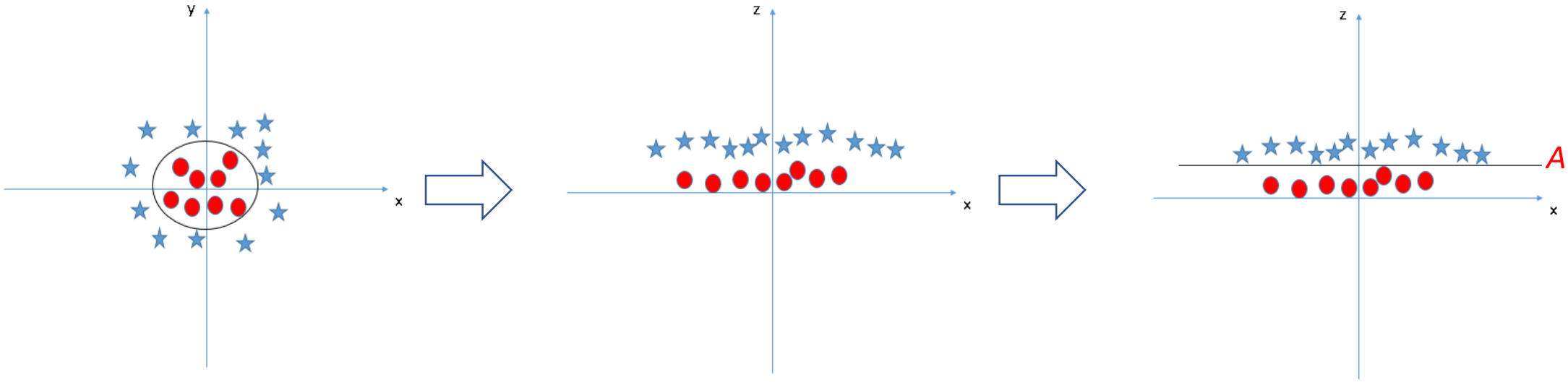
Solution: SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin.

Machine Learning Algorithms: KNN and SVM

❑ Support Vector Machine (cont..)

How it works?

✓ Identify the right hyper-plane (scenario 5): In the scenario below, we cannot have linear hyper-plane between two classes, so how does SVM classify these two classes?



Solution: SVM can solve this problem by using an additional function called Kernel Trick that takes a low dimensional input space and transforms to a higher dimensional space $z = x^2 + y^2$ and plots the data points on axis x and z.

Machine Learning Algorithms: KNN and SVM

❑ **Lecture Overview**

✓ Supervised ML Algorithms (for classification)

- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)