Support Vector Machines

Sachin Tripathi

IIT(ISM), Dhanbad

Topics to be covered

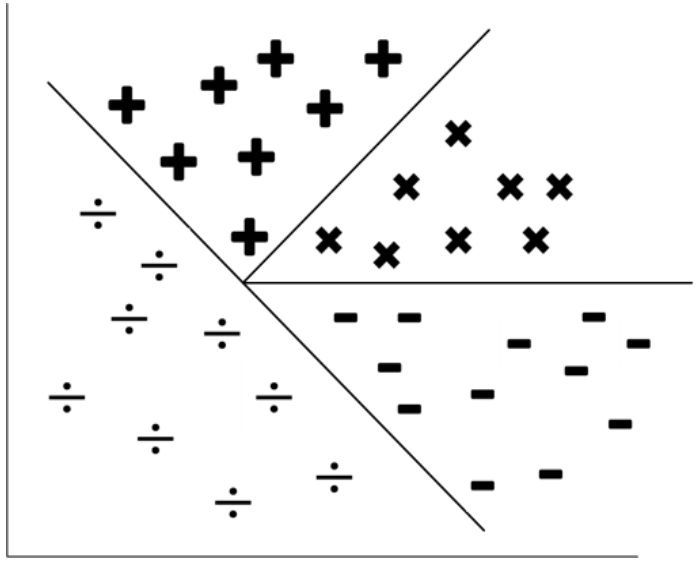
- Introduction to SVM
- Concept of maximum margin hyperplane
- Linear SVM

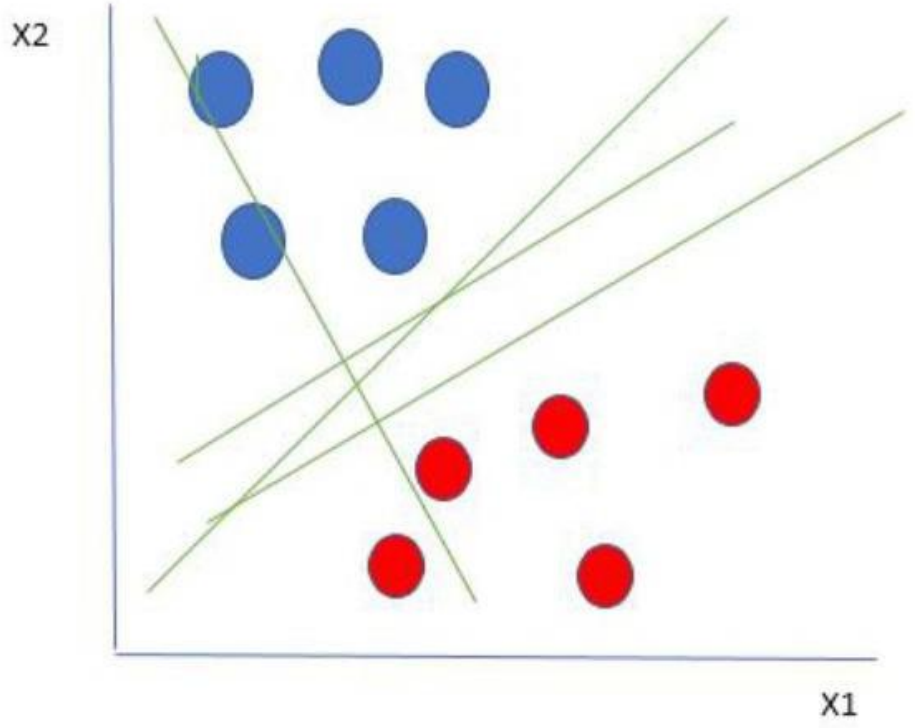
Introduction

- A classification that has received considerable attention is support vector machine and popularly abbreviated as SVM.
- This technique has its roots in statistical learning theory (Vlamidir Vapnik, 1992).
- As a task of classification, it searches for optimal Hyperplane separating the tuples of one class from another.
- SVM works well with higher dimensional data and thus avoids dimensionality problem.

- A support vector machine (SVM) is a classifier which intakes training data (supervised learning), the algorithm outputs an optimal hyperplane (it is also called decision surface) which categorizes new examples
- Although the SVM based classification (i.e., training time) is extremely slow, the result, is however highly accurate. Further, testing an unknown data is very fast.
- SVM is less prone to over fitting than other methods. It also facilitates compact model for classification.

Decision Boundary in SVM





Linearly Separable Data points

Terminology

- Margin is the perpendicular distance between the closest data points and the hyperplane (on the both sides)
- The best optimized line with maximum margin is termed as Margin Maximal Hyperplane
- The closest points where the marginal distance is calculated are termed as support vectors
- Support Vectors: Support vectors are the closest data points to the Hyperplane, which makes a critical role in deciding the Hyperplane and margin.

- SVM aims to find out a hyperplane that will maximize the marginal distance

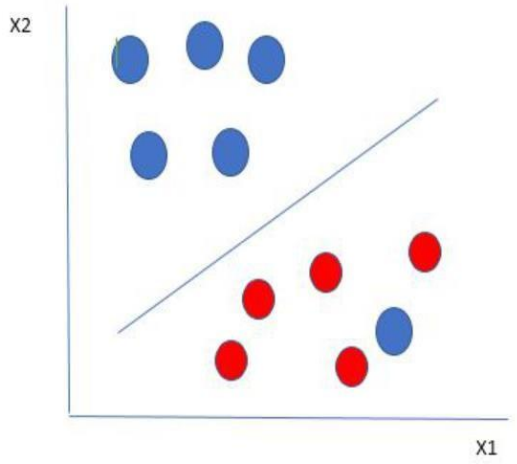
- Margin: Margin is the distance between the support vector and Hyperplane. The main objective of the support vector machine algorithm is to maximize the margin. The wider margin indicates better classification performance.
- Kernel: Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the Hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.

- Hyperplane: Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e. $wx+b = 0$.

- The shortest distance from a Hyperplane to one of its decision boundary is equal to the shortest distance from the Hyperplane to the decision boundary at its other side. Alternatively, Hyperplane is at the middle of its decision boundaries.

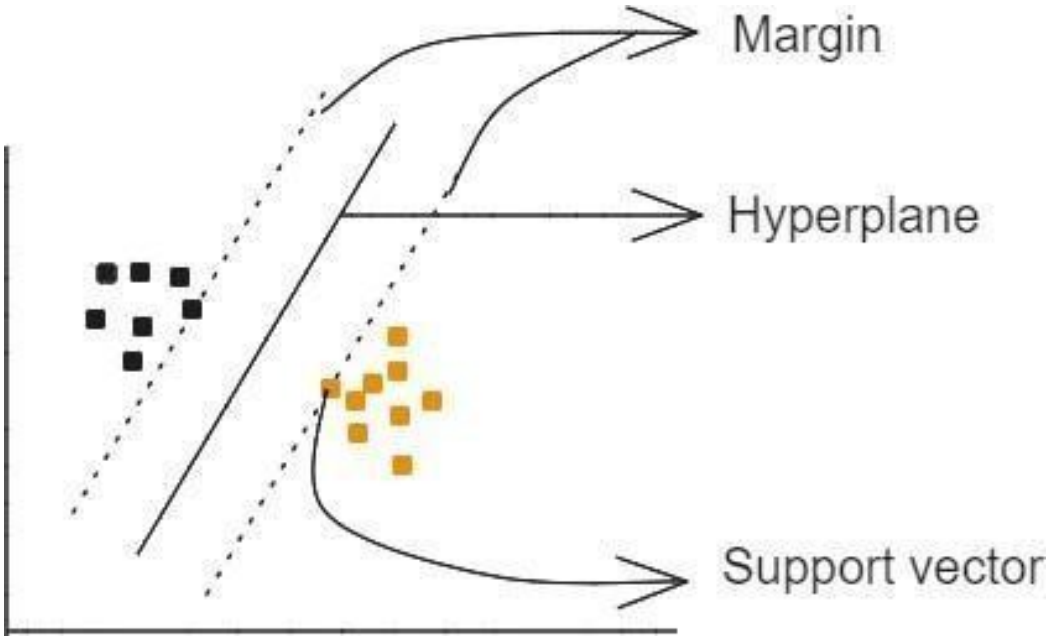
- Hard Margin: The maximum-margin Hyperplane or the hard margin Hyperplane is a Hyperplane that properly separates the data points of different categories without any misclassifications.

- Soft Margin: When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique.



Hyperplane which is the most optimized one

Terminology (contd)



- Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations.
- It discovers a compromise between increasing the margin and reducing violations.

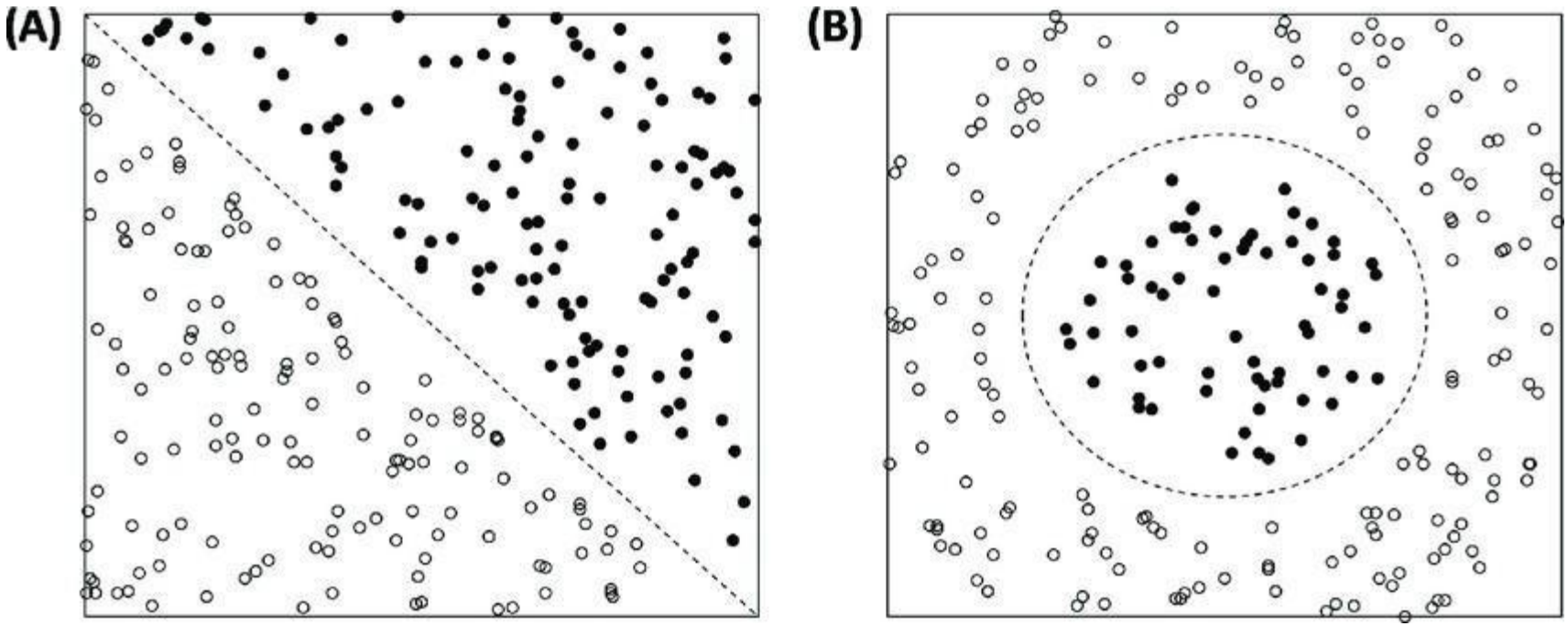
- So, in this type of data point what SVM does is, finds the maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So, the margins in these types of cases are called soft margins.

- When there is a soft margin to the data set, the SVM tries to minimize (1/margin+(∑penalty)).
- Hinge loss is a commonly used penalty. If no violations no hinge loss. If violations hinge loss proportional to the distance of violation.
- C: Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM.

- The penalty for going over the margin or misclassifying data items is decided by it. A stricter penalty is imposed with a greater value of C, which results in a smaller margin and perhaps fewer misclassifications.
- Hinge Loss: A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.

- Dual Problem: A dual Problem of the optimization problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM.
- The dual formulation enables the use of kernel tricks and more effective computing.

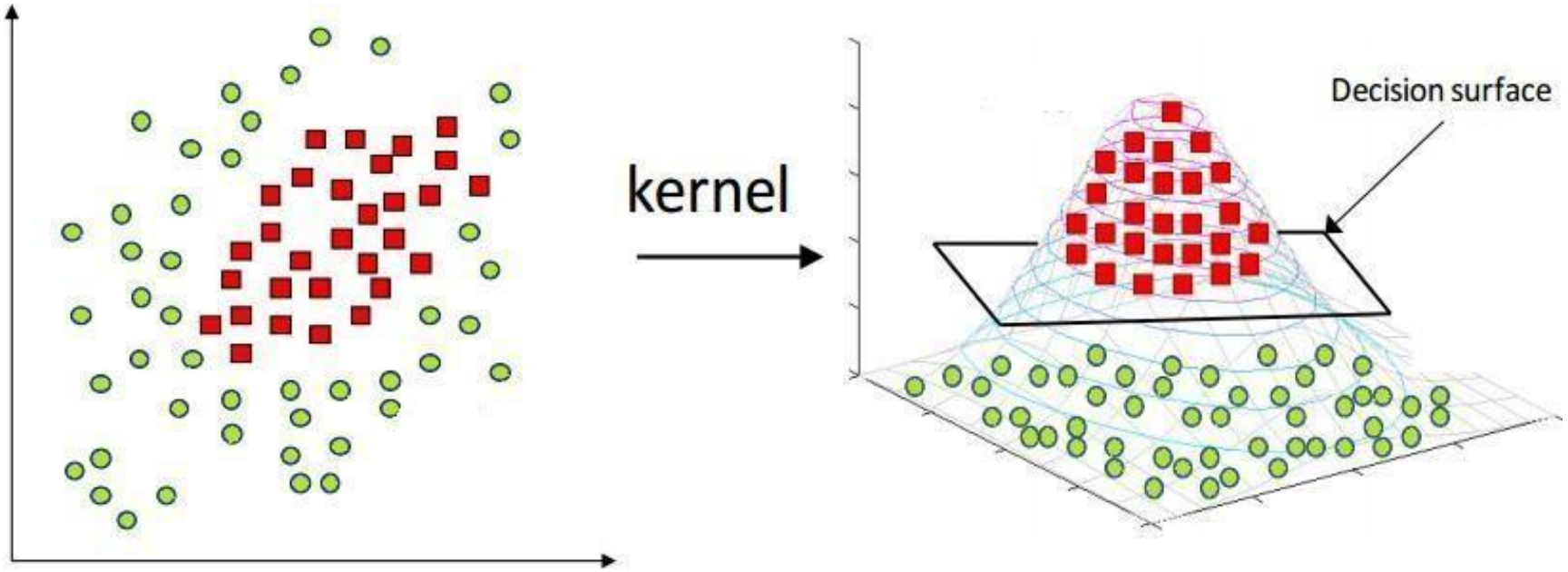
Linearly and Nonlinearly separable



Kernel in SVM

- For non-linearly separable dataset we need kernel to separate those dataset.
- Kernel essentially transforms the dataset from one space to another such that the dataset can be separated by the hyperplane
- It can transform the data with respect to the same dimension itself or it can be used to transform the lower dimensional data into higher dimensional data, thus drawing the hyperplane will become easier

Kernel in SVM (contd)



Pros of SVM

- It works really well with clear margin of separation
- It is effective in high dimension space
- It is effective in cases where number of dimensions is greater than the number of samples
- It uses the subset of data points (support vectors) in the decision function, so it is also memory efficient

Cons of SVM

- It doesn’t perform well when there is too much noise in the data set i.e. target classes are overlapped
- SVM doesn’t directly provide probability estimates

Application of SVM

- Face detection
- Bioinformatic analysis

Finding MMH for a Linear SVM

- Assuming a case of binary classification problem consisting of n training data.

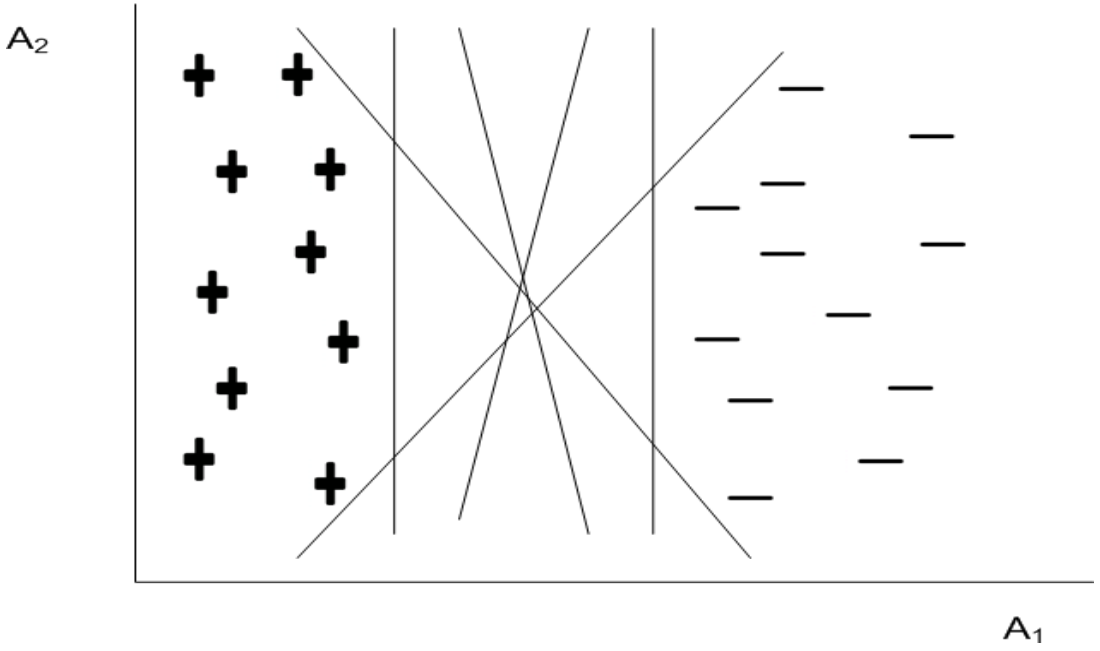
- Each tuple is denoted by (Xi ,Yi) where Xi = (xi1,xi2.....xim) corresponds to the attribute set for the ith tuple (data in m-dimensional space) and Yi ϵ [+,-] denotes its class label.

- Note that choice of which class should be labeled as + or - is arbitrary.

Maximum Margin Hyperplane

In our subsequent discussion, we shall assume a simplistic situation that given a training data D ={t_1,t_2,....t_n} with a set of n tuples, which belong to two classes either + or - and each tuple is described by two attributes say A_1, A_2.

A 2D data linearly separable by Hyperplane

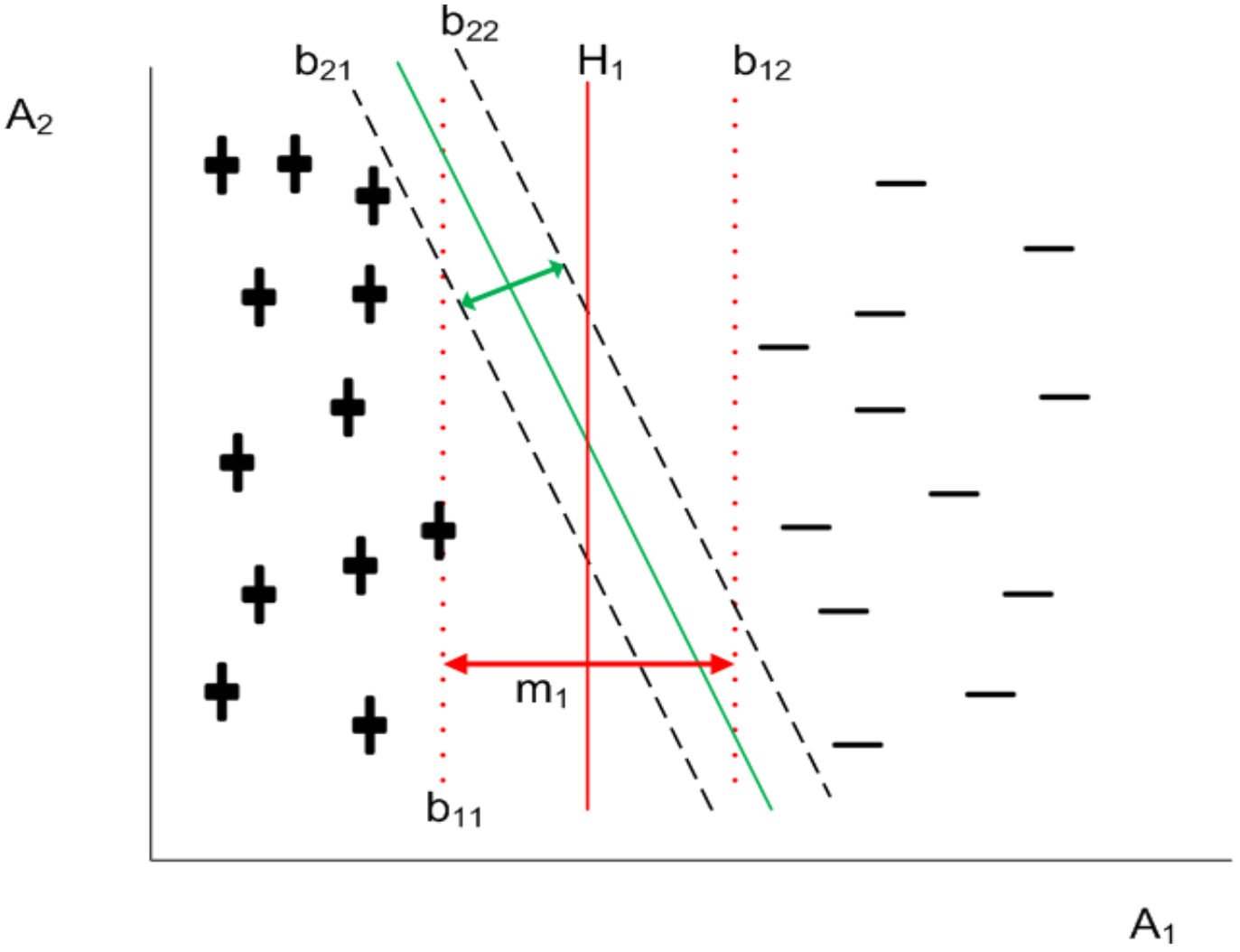


- Figure shows a plot of data in 2-D. Another simplistic assumption here is that the data is linearly separable, that is, we can find a Hyperplane (in this case, it is a straight line) such that all +'s reside on one side whereas all -'s reside on other side of the Hyperplane.
- From Fig., it can be seen that there are an infinite number of separating lines that can be drawn. Therefore, the following two questions arise:

  - Whether all Hyper planes are equivalent so far the classification of data is concerned?
  - If not, which Hyperplane is the best?

- We may note that so far the classification error is concerned (with training data), all of them are with zero error.
- However, there is no guarantee that all Hyper planes perform equally well on unseen (i.e., test) data.
- Thus, for a good classifier it must choose one of the infinite number of Hyper planes, so that it performs better not only on training data but as well as test data.
- To illustrate how the different choices of Hyperplane influence the classification error, consider any arbitrary two Hyper planes H1 and H2 as shown in figure (next slide).

Hyper planes with decision boundaries and their margins



- In Figure shown, two Hyper planes H_1 and H_2 have their own boundaries called decision boundaries(denoted as b_{11} and b_{12} for H_1 and b_{21} and b_{22} for H_2).
- A decision boundary is a boundary which is parallel to Hyperplane and touches the closest class in one side of the Hyperplane.

- The distance between the two decision boundaries of a Hyperplane is called the margin. So, if data is classified using Hyperplane H1, then it is with larger margin then using Hyperplane H2.

- The margin of Hyperplane implies the error in classifier. In other words, the larger the margin, lower is the classification error.

- Intuitively, the classifier that contains Hyperplane with a small margin are more susceptible to model over fitting and tend to classify with weak confidence on unseen data.

- Thus during the training or learning phase, the approach would be to search for the Hyperplane with maximum margin.
- Such a Hyperplane is called maximum margin Hyperplane and abbreviated as MMH.
- We may note the shortest distance from a Hyperplane to one of its decision boundary is equal to the shortest distance from the Hyperplane to the decision boundary at its other side.
- Alternatively, Hyperplane is at the middle of its decision boundaries

Linear SVM

- A SVM which is used to classify data which are linearly separable is called linear SVM.
- In other words, a linear SVM searches for a hyperplane with the maximum margin.
- This is why a linear SVM is often termed as a maximal margin classifier (MMC).

Finding MMH (Linear SVM)

- In the following, we shall discuss the mathematics to find the MMH given a training set.
- In our discussion, we shall consider a binary classification problem consisting of n training data.
- Each tuple is denoted by (Xi ,Yi) where Xi = (xi1,xi2.....xim) corresponds to the attribute set for the ith tuple (data in m-dimensional space) and Yi ϵ [+,-] denotes its class label.
- Note that choice of which class should be labeled as + or - is arbitrary.

- Thus, given $\{(X_i, Y_i)\}_{i=1}^{n}$, we are to obtain a hyperplane which separates all $X_i|_{i=1}^{n}$ into two sides of it (of course with maximum gap).

- Before, going to a general equation of a plane in *n*-dimension, let us consider first, a hyperplane in 2-D plane.

Equation of a Hyperplane in 2-D

- Let us consider a 2-D training tuple with attributes A1 and A2 as X =(x1,x2), where x1 and x2 are values of attributes A1 and A2, respectively for X .

- Equation of a plane in 2-D space can be written as
  w0 + w1x1 + w2x2 = 0 [e.g., ax + by + c = 0]
  - where w0, w1, and w2 are some constants defining the slope and intercept of the line.
- Any point lying above such a Hyperplane satisfies
  $w_0 + w_1x_1 + w_2x_2 > 0$

- Similarly, any point lying below the Hyperplane satisfies

$$w0 + w1x1 + w2x2 < 0$$

- An SVM Hyperplane is an n-dimensional generalization of a straight line in 2-D.
- It can be visualized as a plane surface in 3-D, but it is not easy to visualize when dimensionality is greater than 3
- In fact, Euclidean equation of a Hyperplane in R^m is

$$w1x1 + w2x2 + ... + wmxm = b$$

  - where wi ’s are the real numbers and b is a real constant (called the intercept, which can be positive or negative).

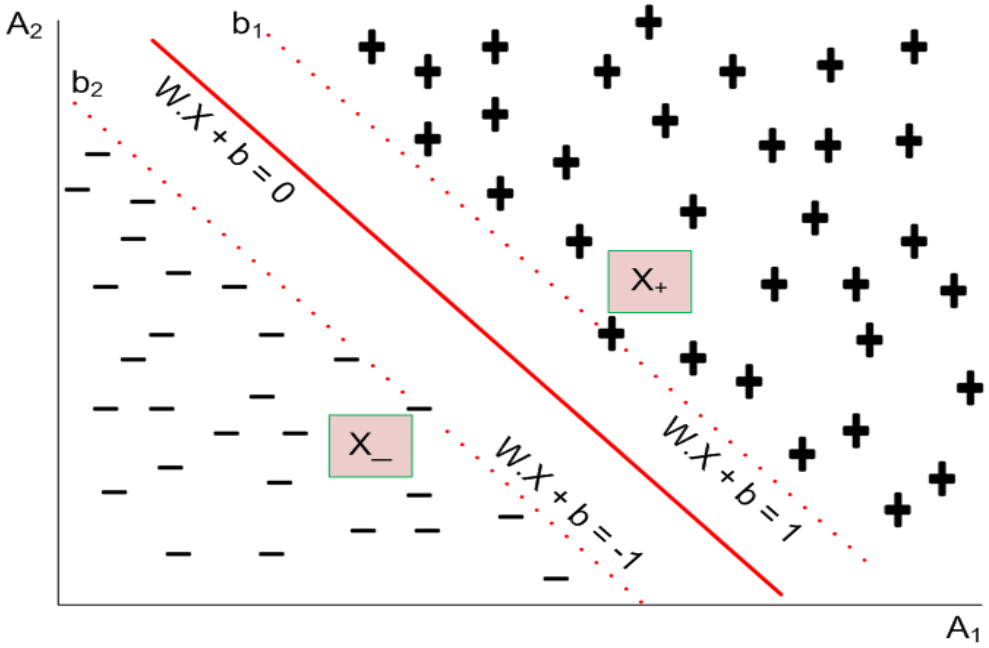
- In matrix form, a Hyperplane thus can be represented as: $W.X + b = 0$
  where $W = [w1, w2 \ldots\ldots wm]$ and $X = [x1, x2 \ldots\ldots xm]$ and b is a real constant.

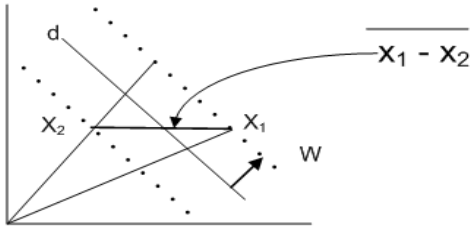
- Here, W and b are parameters of the classifier model to be evaluated given a training set D.

- Let us consider a two-dimensional training set consisting two classes + and - as shown in next slide
- Suppose, b1 and b2 are two decision boundaries above and below a Hyperplane, respectively.
- Consider any two points X+ and X— as shown in next slide.
- For X+ located above the decision boundary, the equation can be written as

$$W.X+ + b = K \quad \text{where } K > 0$$

Computation of the MMH





- Similarly, for any point X— located below the decision boundary, the equation is

W.X— + b = K r where K r < 0

- Thus, if we label all +'s are as class label + and all -'s are class label -, then we can predict the class label Y for any test data X as

$$(Y) = \begin{cases} + & \text{if } W.X + b > 0 \\ — & \text{if } W.X + b < 0 \end{cases}$$

Hyperplane and Classification

- Note that W.X + b = 0, the equation representing Hyperplane can be interpreted as follows.
  - Here, W represents the orientation and b is the intercept of the Hyperplane from the origin.
  - If both W and b are scaled (up or down) by dividing a non zero constant, we get the same Hyperplane.
  - This means there can be infinite number of solutions using various scaling factors, all of them geometrical representing the same Hyperplane.

- To avoid such a confusion, we can make W and b unique by adding a constraint that $W_r.X + b_r = \pm 1$ for data points on boundary of each class.
- It may be noted that $W_r.X + b_r = \pm 1$ represents two Hyperplane parallel to each other.
- For clarity in notation, we write this as $W.X + b = \pm 1$.
- Having this understating, now we are in the position to calculate the margin of a hyperplane.

Calculating Margin of a Hyperplane

- Suppose, x1 and x2 are two points on the decision boundaries b1 and b2, respectively. Thus,

$$W.x1 + b = 1$$

$$W.x2 + b = -1$$

or

$$W.(x1 - x2) = 2$$

This represents a dot (.) product of two vectors W and x1-x2. Thus taking magnitude of these vectors, the equation obtained is

This represents a dot (.) product of two vectors W and x_1-x_2. Thus taking magnitude of these vectors, the equation obtained is

$$d = \frac{2}{||W||}$$

where $||W|| = \sqrt{w_1^2 + w_2^2 + ...w_m^2}$ in an m-dimension space.

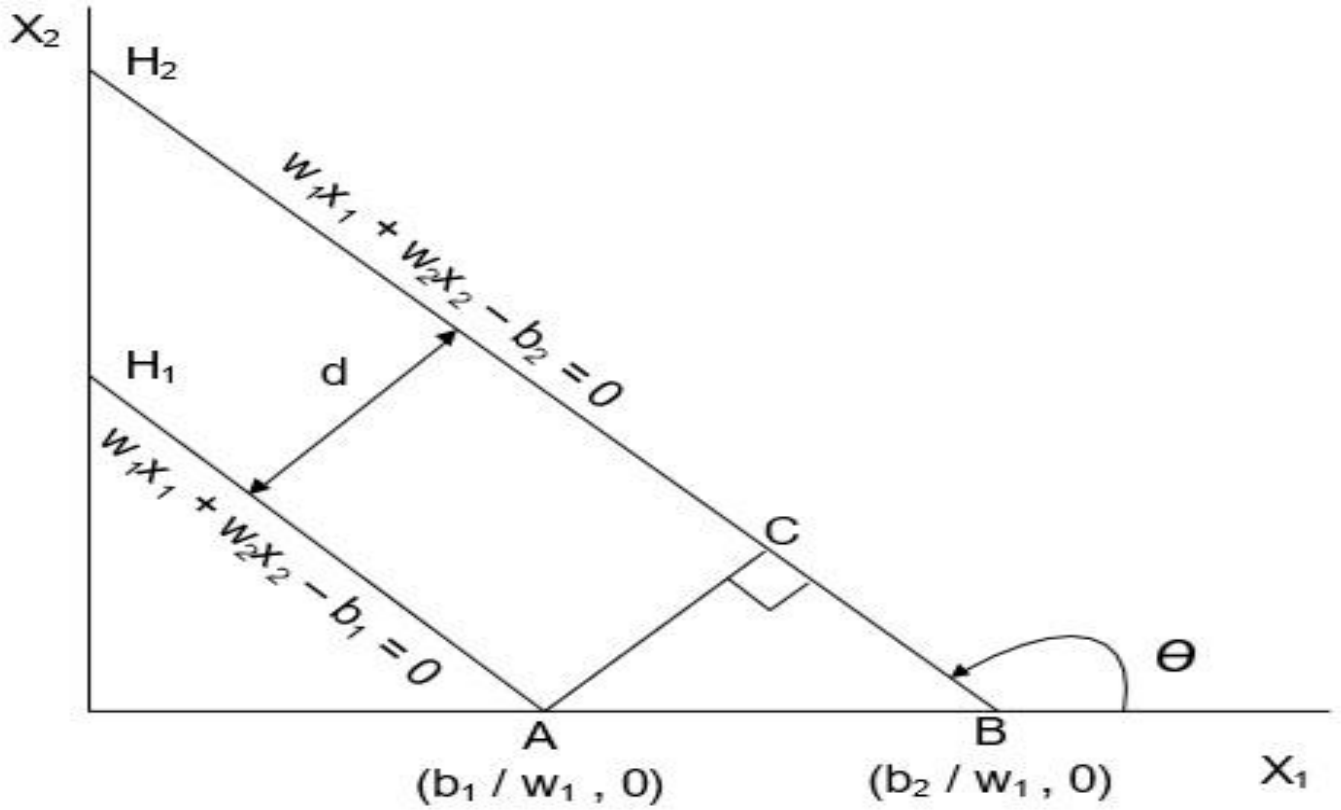
- We calculate the margin more mathematically, as in the following.
- Consider two parallel Hyper planes H1 and H2 as shown in next slide . Let the equations of Hyper planes be

$$H_1 : w_1x_1 + w_2x_2 - b_1 = 0$$

$$H_2 : w_1x_1 + w_2x_2 - b_2 = 0$$

To draw a perpendicular distance d between H_1 and H_2, we draw a right-angled triangle ABC as shown in next slide.

Details of Margin Calculation



Being parallel, the slope of H_1 (and H_2) is $tan\theta = -\frac{w_1}{w_2}$.

In triangle ABC, AB is the hypotenuse and AC is the perpendicular distance between H_1 and H_2.

Thus , $sin(180 - \theta) = \frac{AC}{AB}$ or $AC = AB \cdot sin\theta$.

$AB = \frac{b_2}{w_1} - \frac{b_1}{w_1} = \frac{|b_2 - b_1|}{w_1}$, $sin\theta = \frac{w_1}{\sqrt{w_1^2 + w_2^2}}$

(Since, $tan\theta = -\frac{w_1}{w_2}$).

Hence, $AC = \frac{|b_2 - b_1|}{\sqrt{w_1^2 + w_2^2}}$.

This can be generalized to find the distance between two parallel margins of any hyperplane in n-dimensional space as

$$d = \frac{|b_2 - b_1|}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}} \simeq \frac{|b_2 - b_1|}{\| W \|}$$

where, $\| W \| = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$.

In SVM literature, this margin is famously written as $\mu(W, b)$.

- The training phase of SVM involves estimating the parameters Wand b for a Hyperplane from a given training data.
- The parameters must be chosen in such a way that the following two inequalities are satisfied.

$$W.x_i + b \geq 1 \qquad if \quad y_i = 1$$

$$W.x_i + b \leq -1 \qquad if \quad y_i = -1$$

- These conditions impose the requirements that all training tuples from class Y = + must be located on or above the Hyperplane W.x + b = 1, while those instances from class Y = − must be located on or below the Hyperplane W.x + b = −1

- Both the inequalities can be summarized as

$$y_i (W.x_1 + b) \geq 1 \qquad \forall_i i = 1, 2, ..., n$$

Note that any tuples that lie on the hyper planes H_1 and H_2 are called support vectors.

- Essentially, the support vectors are the most difficult tuples to classify and give the most information regarding classification.
- In the following, we discuss the approach of finding MMH and the support vectors. The above problem is turned out to be an optimization problem, that is, to maximize μ(W, b) = $\frac{2}{\|w\|}$.

Searching for MMH

- Maximizing the margin is, however, equivalent to minimizing the following objective function

$$\mu`(W, b) = \frac{||W||}{2}$$

- In nutshell, the learning task in SVM, can be formulated as the following constrained optimization problem.

$$\textit{minimize} \quad \mu`(W, b)$$

$$\textit{subject to} \quad y_i(W.x_i + b) \geq 1, \quad i = 1, 2, 3.....n$$

- The above stated constrained optimization problem is popularly known as convex optimization problem, where objective function is quadratic and constraints are linear in the parameters W and b.

- The well known technique to solve a convex optimization problem is the standard Lagrange Multiplier method.

- First, we shall learn the Lagrange Multiplier method, then come back to the solving of our own SVM problem.

Lagrange Multiplier Method

- The Lagrange multiplier method follows two different steps depending on type of constraints.

Equality constraint optimization problem: In this case, the problem is of the form:

$$minimize\ f(x_1, x_2, ..., x_d)$$

$$subject\ to\ g_i(x) = 0, i = 1, 2, ..., p$$

Inequality constraint optimization problem: In this case, the problem is of the form:

$$minimize\ f(x_1, x_2, ..., x_d)$$

$$subject\ to\ h_i(x) \leq 0, i = 1, 2, ..., p$$

Equality constraint optimization problem solving

The following steps are involved in this case:

Define the Lagrangian as follows:

$$(X, \lambda) = f(X) + \sum_{i=1}^{p} \lambda_i . g_i(x)$$

where $\lambda_i' s$ are dummy variables called Lagrangian multipliers.

Set the first order derivatives of the Lagrangian with respect to x and the Lagrangian multipliers $\lambda_i' s$ to zero's. That is

$$\frac{\delta L}{\delta x_i} = 0, i = 1, 2, ..., d$$

$$\frac{\delta L}{\delta \lambda_i} = 0, i = 1, 2, ..., p$$

Solve the $(d + p)$ equations to find the optimal value of $X = [x_1, x_2....., x_d]$ and $\lambda_i' s$.

Example

Suppose, minimize $f(x, y) = x + 2y$
subject to $x^2 + y^2 - 4 = 0$

Lagrangian $L(x, y, \lambda) = x + 2y + \lambda(x^2 + y^2 - 4)$

$$\frac{\delta L}{\delta x} = 1 + 2\lambda x = 0$$
$$\frac{\delta L}{\delta y} = 1 + 2\lambda y = 0$$
$$\frac{\delta L}{\delta \lambda} = x^2 + y^2 - 4 = 0$$

Solving the above three equations for x, y and λ, we get $x = \mp\frac{2}{\sqrt{5}}$, $y = \mp\frac{4}{\sqrt{5}}$ and $\lambda = \pm\frac{\sqrt{5}}{4}$

When $\lambda = \frac{\sqrt{5}}{4}$,
$x = -\frac{2}{\sqrt{5}}$,
$y = -\frac{4}{\sqrt{5}}$,
we get f$(x, y, \lambda) = -\frac{10}{\sqrt{5}}$

Similarly, when $\lambda = -\frac{\sqrt{5}}{4}$,
$x = \frac{2}{\sqrt{5}}$,
$y = \frac{4}{\sqrt{5}}$,
we get f$(x, y, \lambda) = \frac{10}{\sqrt{5}}$

Thus, the function f(x, y) has its minimum value at
$x = -\frac{2}{\sqrt{5}}, y = -\frac{4}{\sqrt{5}}$

Inequality constraint optimization problem solving

The method for solving this problem is quite similar to the Lagrange multiplier method described above.

It starts with the Lagrangian

$$L = f(x) + \sum_{i=1}^{p} \lambda_i . h_i(x)$$

In addition to this, it introduces additional constraints, called **Karush-Kuhn-Tucker (KKT) constraints**, which are stated in the next slide.

$$\frac{\delta L}{\delta x_i} = 0, i = 1, 2, ..., d$$

$$\lambda_i \geq 0, i = 1, 2....., p$$

$$h_i(x) \leq 0, i = 1, 2....., p$$

$$\lambda_i . h_i(x) = 0, i = 1, 2....., p$$

Solving the above equations, we can find the optimal value of $f(x)$.

Example

Consider the following problem.

Minimize $f(x, y) = (x - 1)^2 + (y - 3)^2$
subject to $x + y \leq 2$,
$y \geq x$

- The Lagrangian for this problem is
  $L = (x - 1)^2 + (y - 3)^2 + \lambda_1(x + y - 2) + \lambda_2(x - y)$.

  subject to the KKT constraints, which are as follows:

$$\frac{\delta L}{\delta x} = 2(x-1) + \lambda_1 + \lambda_2 = 0$$

$$\frac{\delta L}{\delta y} = 2(y-3) + \lambda_1 - \lambda_2 = 0$$

$$\lambda_1(x+y-2) = 0$$

$$\lambda_2(x-y) = 0$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0$$

$$(x+y) \leq 2, y \geq x$$

To solve KKT constraints, we have to check the following tests:

Case 1: $\lambda_1 = 0, \lambda_2 = 0$

$$2(x-1) = 0 \mid 2(y-3) = 0 \Rightarrow x = 1, y = 3$$

since, $x + y = 4$, it violates $x + y \leq 2$; is not a feasible solution.

Case 2: $\lambda_1 = 0, \lambda_2 \neq 0$ $2(x-y) = 0 \mid$
$2(x-1) + \lambda_2 = 0 \mid$
$2(y-3) - \lambda_2 = 0$
$\Rightarrow x = 2, y = 2$ and $\lambda_2 = -2$
since, $x + y \leq 4$, it violates $\lambda_2 \geq 0$; is not a feasible solution.

Case 3: $\lambda_1 \neq 0, \lambda_2 = 0$ $2(x+y) = 2$
$2(x-1) + \lambda_1 = 0$
$2(y-3) + \lambda_1 = 0$

$\Rightarrow x = 0, y = 2$ and $\lambda_1 = 2$; this is a feasible solution.

Case 4: $\lambda_1 \neq 0, \lambda_2 \neq 0$ $2(x+y) = 2$
$2(x-y) = 0$
$2(x-1) + \lambda_1 + \lambda_2 = 0$
$2(y-3) + \lambda_1 - \lambda_2 = 0$

$\Rightarrow x = 1, y = 1$ and $\lambda_1 = 2$ $\lambda_2 = -2$
This is not a feasible solution.

LMM to Solve Linear SVM

The optimization problem for the linear SVM is inequality constraint optimization problem.

The Lagrangian multiplier for this optimization problem can be written as

$$L = \frac{||W||^2}{2} - \sum_{i=1}^{n} \lambda_i (y_i (W.x_i + b) - 1)$$

where the parameters $\lambda_i's$ are the Lagrangian multipliers, and $W = [w_1, w_2 ..... w_m]$ and b are the model parameters.

The KKT constraints are:

$$\frac{\delta L}{\delta w} = 0 \Rightarrow W = \sum_{i=1}^{n} \lambda_i . y_i . x_i$$

$$\frac{\delta L}{\delta b} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i . y_i = 0$$

$$\lambda \geq 0, i = 1, 2, ...n$$

$$\lambda_i [y_i (W.x_i + b) - 1] = 0, i = 1, 2, ...n$$

$$y_i (W.x_i + b) \geq 1, i = 1, 2, ...n$$

Solving KKT constraints are computationally expensive and can be solved using a typical linear/ quadratic programming technique (or any other numerical technique).

We first solve the above set of equations to find all the feasible solutions.

Then, we can determine optimum value of $\mu(W, b)$.

Note:

Lagrangian multiplier λ_i must be zero unless the training instance x_i satisfies the equation $y_i(W.x_i + b) = 1$. Thus, the training tuples with $\lambda_i > 0$ lie on the hyperplane margins and hence are support vectors.

The training instances that do not lie on the hyperplane margin have $\lambda_i = 0$.

Classifying a test sample

For a given training data, using SVM principle, we obtain MMH in the form of W, b and λ_i's. This is the machine (i.e., the SVM).

Now, let us see how this MMH can be used to classify a test tuple say X. This can be done as follows.

$$\delta(X) = W.X + b = \sum_{i=1}^{n} \lambda_i.y_i.x_i.X + b$$

Note that

$$W = \sum_{i=1}^{n} \lambda_i.y_i.x_i$$

This is famously called as “Representer Theorem” which states that the solution W always be represented as a linear combination of training data.

$$Thus, \delta(X) = W.X + b = \sum_{i=1}^{n} \lambda_i.y_i.x_i.X + b$$

The above involves a dot product of $x_i.X$,where x_i is a support vector (this is so because $(\lambda_i) = 0$ for all training tuples except the support vectors), we can check the sign of $\delta(X)$.

If it is positive, then X falls on or above the MMH and so the SVM predicts that X belongs to class label +. On the other hand, if the sign is negative, then X falls on or below MMH and the class prediction is -.

Note:

Once the SVM is trained with training data, the complexity of the classifier is characterized by the number of support vectors.

Dimensionality of data is not an issue in SVM unlike in other classifier.

Illustration

Consider the case of a binary classification starting with a training data of 8 tuples as shown in Table 1.

Using quadratic programming, we can solve the KKT constraints to obtain the Lagrange multipliers λ_i for each training tuple, which is shown in Table 1.

Note that only the first two tuples are support vectors in this case.

Let $W = (w_1, w_2)$ and b denote the parameter to be determined now. We can solve for w_1 and w_2 as follows:

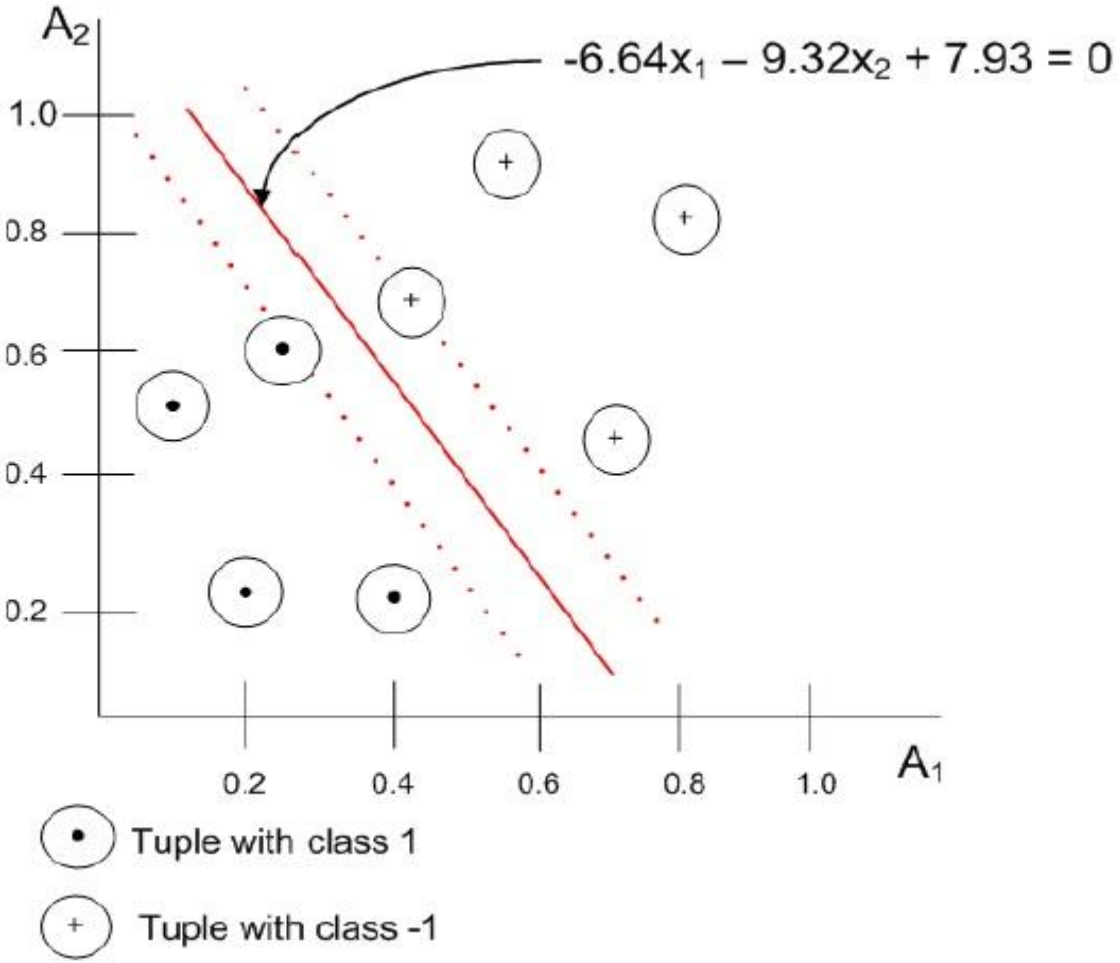
$$w_1 = \sum_i \lambda_i . y_i . x_{i1} = 65.52 \times 1 \times 0.38 + 65.52 \times -1 \times 0.49 = -6.64 \quad (22)$$

$$w_2 = \sum_i \lambda_i . y_i . x_{i2} = 65.52 \times 1 \times 0.47 + 65.52 \times -1 \times 0.61 = -9.32$$

Table 1: Training Data

A_1	A_2	y	λ_i
0.38	0.47	+	65.52
0.49	0.61	-	65.52
0.92	0.41	-	0
0.74	0.89	-	0
0.18	0.58	+	0
0.41	0.35	+	0
0.93	0.81	-	0
0.21	0.10	+	0

Linear SVM Example



The parameter b can be calculated for each support vector as follows

$b_1 = 1 - W.x_1$ // for support vector x_1

$= 1 - (-6.64) \times 0.38 - (-9.32) \times 0.47$ //using dot product

$= 7.93$

$b_2 = 1 - W.x_2$ // for support vector x_2

$= 1 - (-6.64) \times 0.48 - (-9.32) \times 0.611$ //using dot product

$= 7.93$

Averaging these values of b_1 and b_2, we get $b = 7.93$.

Thus, the MMH is $-6.64x_1 - 9.32x_2 + 7.93 = 0$

Suppose, test data is $X = (0.5, 0.5)$. Therefore,

$\delta(X) = W.X + b$

$= -6.64 \times 0.5 - 9.32 \times 0.5 + 7.93$

$= -0.05$

$= -ve$

This implies that the test data falls on or below the MMH and SVM classifies that X belongs to class label -.

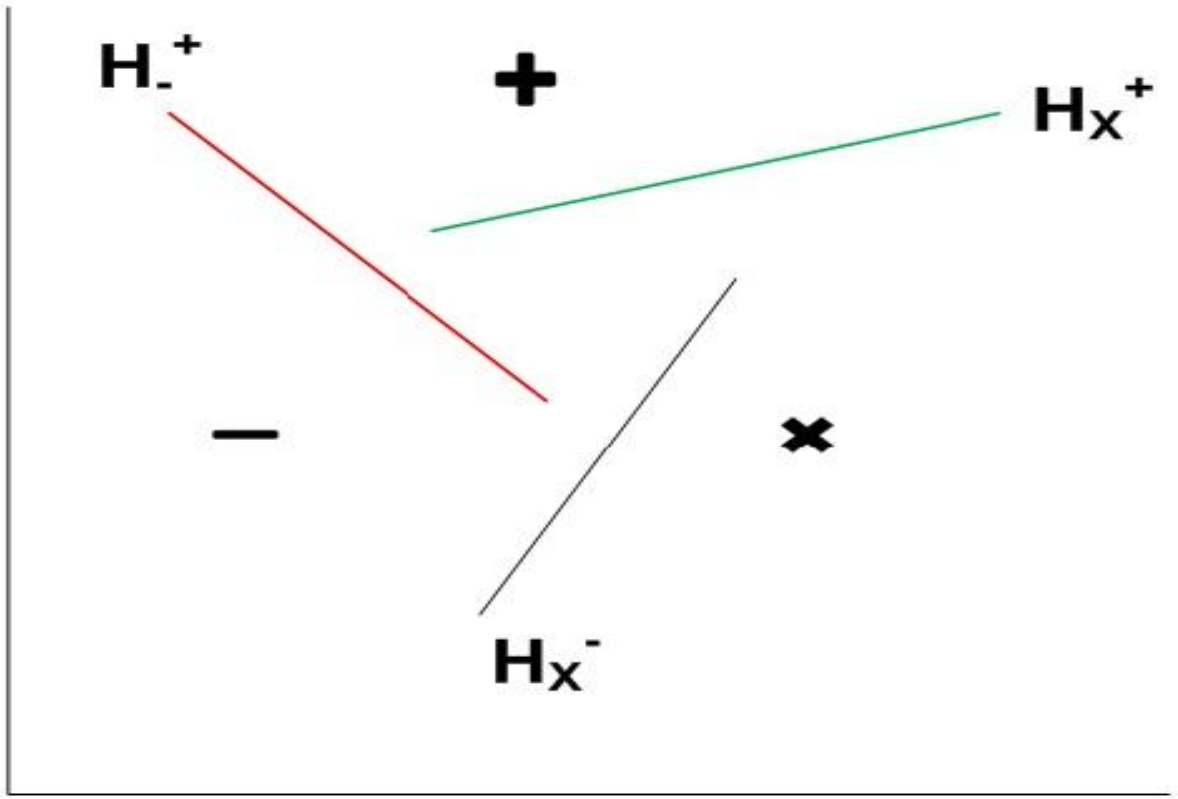
Classification of Multiple-class Data

- In the discussion of linear SVM, we have limited to binary classification (i.e., classification with two classes only).
- Note that the discussed linear SVM can handle any n-dimension, n greater than 2.
- There are two possibilities: all classes are pairwise linearly separable, or classes are overlapping, that is, not linearly separable.
- If the classes are pair wise linearly separable, then we can extend the principle of linear SVM to each pair. There are two strategies:
  - One versus one (OVO) strategy
  - One versus all (OVA) strategy

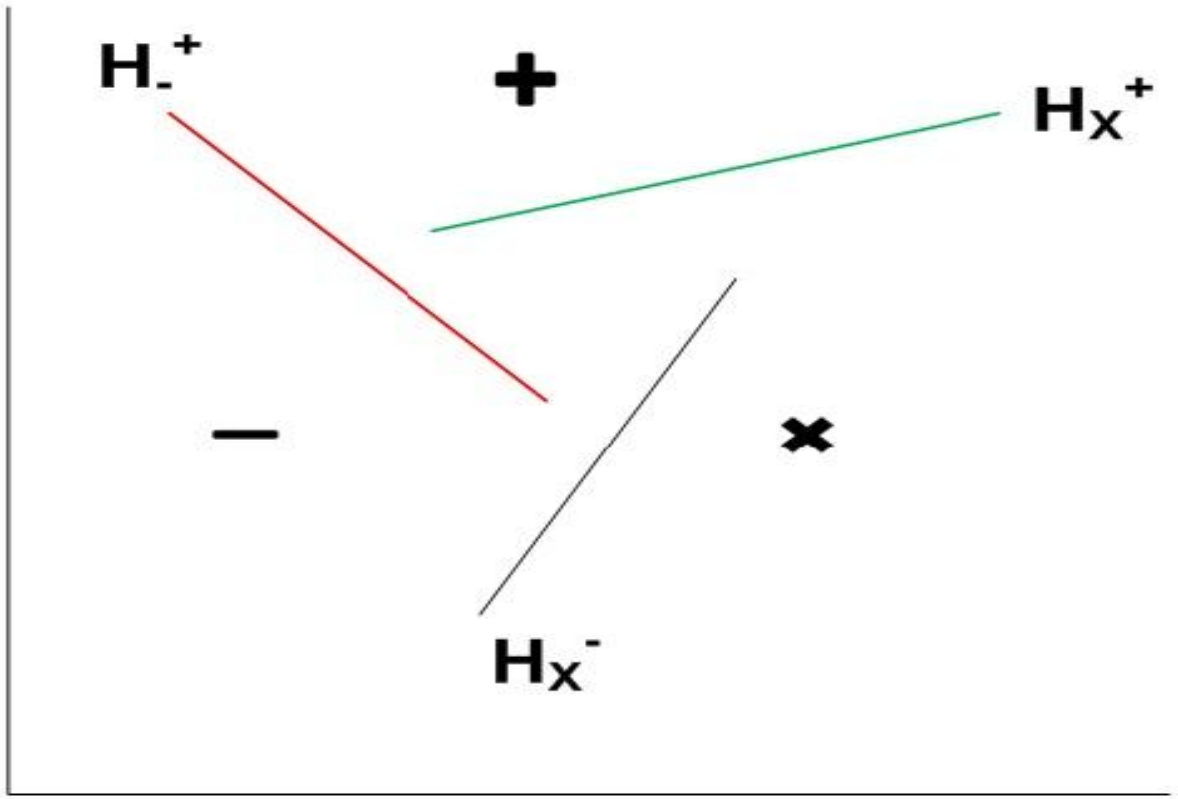
OVO Strategy

- In OVO strategy, we have to find MMHs for each pair of classes
- Thus, if there are n classes, then n (n-1)/2 pairs and hence so many classifiers possible (of course, some of which may be redundant).

3-pairwise linearly separable classes.



3-pairwise linearly separable classes.



OVO Strategy

With OVO strategy, we test each of the classifier in turn and obtain $\delta_i^j(X)$, that is, the count for MMH between i^{th} and i^{th} classes for test data X.

If there is a class i, for which $\delta_i^j(X)$ for all j(and $j \neq i$), gives same sign, then unambiguously, we can say that X is in class i.

OVA Strategy

- OVO strategy is not useful for data with a large number of classes, as the computational complexity increases exponentially with the number of classes.
- As an alternative to OVO strategy, OVA(one versus all) strategyhas been proposed.
- In this approach, we choose any class say Ci and consider that all tuples of other classes belong to a single class.

- This is, therefore, transformed into a binary classification problem and using the linear SVM discussed above, we can find the Hyperplane. Let the Hyperplane between Ci and remaining classes be MMHi .
- The process is repeated for each Ci [C1;C2; :::::Ck] and getting MMHi .
- Thus, with OVA strategies k classifiers obtained.

Multi-Class Classification: OVA

The unseen data X is then tested with each classifier so obtained.

Let $\delta_j(X)$ be the test result with MMH_j, which has the maximum magnitude of test values.

That is

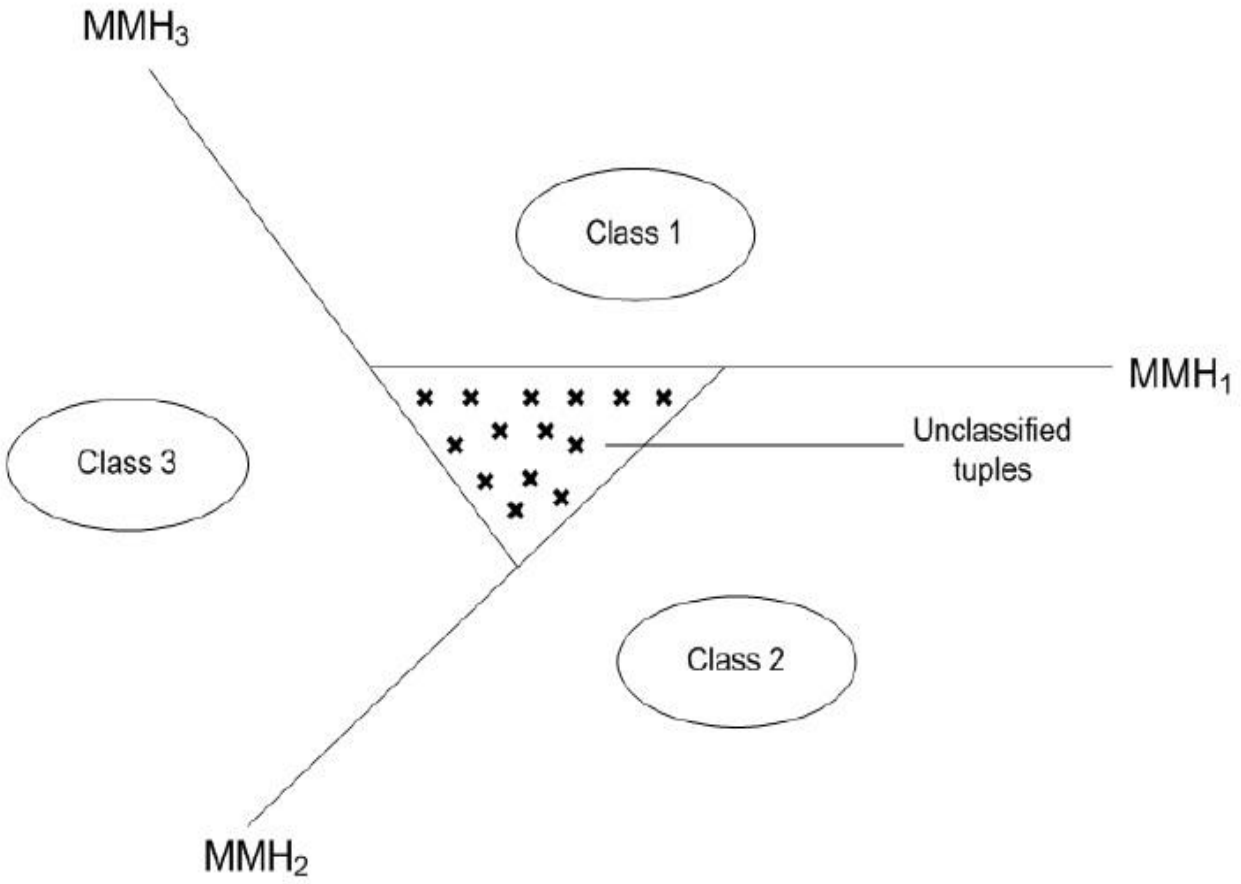
$$\delta_j(X) = max_{\forall i}\{\delta_i(X)\}$$

Thus, X is classified into class C_j.

Multi-Class Classification: OVA Strategy

- The linear SVM that is used to classify multi-class data fails, if all classes are not linearly separable.
- If one class is linearly separable to remaining other classes and test data belongs to that particular class, then only it classifies accurately.
- Further, it is possible to have some tuples which cannot be classified none of the linear SVMs
- There are some tuples which cannot be classified unambiguously by neither of the Hyperplane.
- All these tuples may be due to noise, errors or data are not linearly separable

Unclassifiable region in OVA



Thank You