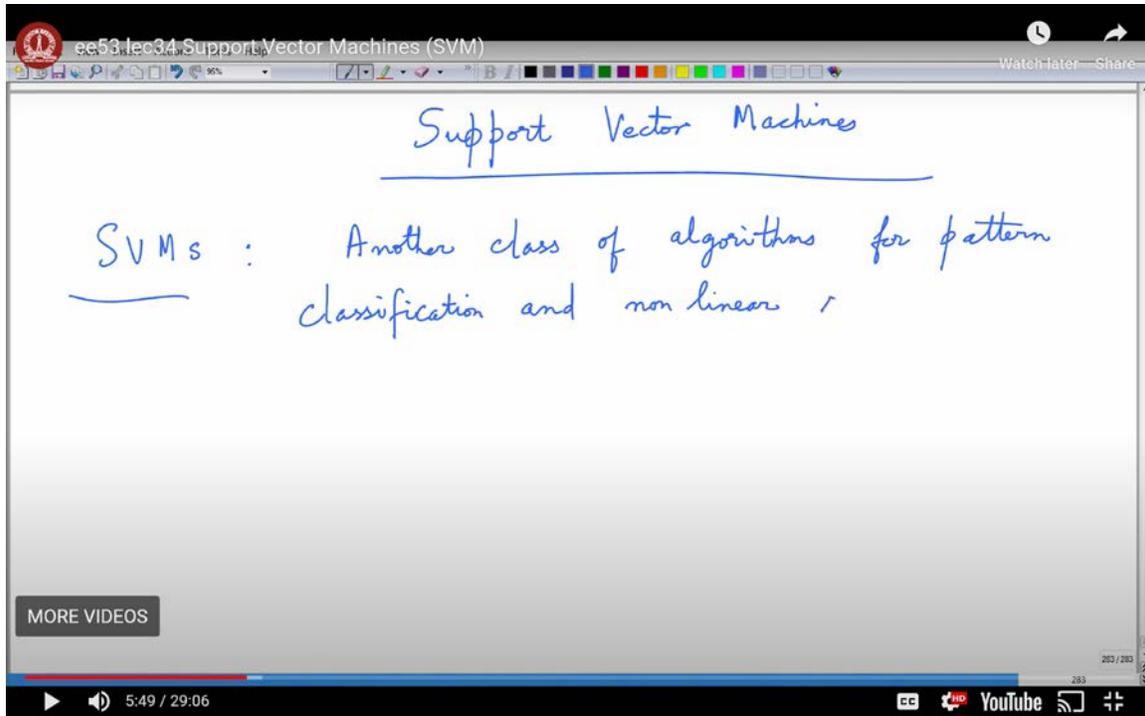


**Neural Networks for Signal Processing-I**  
**Prof. Shayan Srinivasa Garani**  
**Department of Electronic System Engineering**  
**Indian Institute of Science, Bengaluru**

**Lecture – 34**  
**Support Vector Machines (SVM)**

In the earlier modules, we explored Radial Basis Functions (RBFs), Multi-Layer Perceptrons (MLPs), and Perceptrons, each contributing to our understanding of pattern separability in different ways. As we review these concepts, our primary questions are: How can we determine the optimal hyperplane for linearly separable patterns? And if patterns are not linearly separable, is it possible to transform the problem into a higher dimension where we might achieve linear separability?

(Refer Slide Time: 05:49)



The image shows a screenshot of a video lecture slide. The slide title is "Support Vector Machines". The main text on the slide reads: "SVMs : Another class of algorithms for pattern classification and non linear". The slide is displayed within a video player interface. The video player shows the title "ee53. lec34 Support Vector Machines (SVM)" and a progress bar at the bottom indicating the current time is 5:49 out of 29:06. The video player also includes a "MORE VIDEOS" button and a "YouTube" logo.

To revisit the problem, let's start from where we left off. We initially worked with the Perceptron, which guaranteed convergence and could find a hyperplane to separate two linearly separable classes. However, the Perceptron had the drawback of not providing a unique solution; there could be infinitely many hyperplanes that separate the two classes, as long as they are linearly separable.

We then extended this concept to the Multi-Layer Perceptron (MLP), incorporating activation functions that allowed us to create decision boundaries that are non-linear. This enabled us to handle classes that are not linearly separable by using non-linear discriminant boundaries. We thoroughly discussed the architectures and the details of MLPs with non-linear activation functions.

(Refer Slide Time: 07:05)

Support Vector Machines

SVMs : Another class of algorithms for pattern classification and non linear regression.  
It is a linear machine

$w^T x + b$

Roots to SVMs : Vladimir Vapnik  
Very elegant theory with firm roots in Convex

MORE VIDEOS

7:05 / 29:06

YouTube

Following that, we examined Radial Basis Functions (RBFs), which also used non-linear functions in the hidden layer. These functions were combined linearly with weights between the hidden and output layers, providing a way to address non-linear separability effectively.

Now, we face a new question: What if we don't want non-linear decision boundaries, as they can be challenging to handle? Specifically, if we are in an  $m$ -dimensional space and the data points are not linearly separable, can we find a solution by mapping the problem to a higher dimension? This idea, rooted in concepts like Cover's Theorem, suggests that by transforming our data into a higher-dimensional space, we might find a hyperplane that can separate the classes.

Thus, our goal is to explore whether it's possible to use higher-dimensional spaces to achieve linear separability for problems where the data points are not separable in their original dimensions.

(Refer Slide Time: 10:12)

Idea: Construct a hyperplane as the decision surface in such a way that the margin of separation between the 2 classes is maximized

Idea of deriving the hyperplane stems from "structural risk minimization"

Class 1  
hyper plane  
Class 2

MORE VIDEOS

10:12 / 29:06

YouTube

In this module, we will delve into Support Vector Machines (SVMs), a powerful tool designed to address some of the challenges encountered with earlier algorithms, such as the perceptron. Specifically, SVMs offer solutions to issues related to the uniqueness of solutions and hyperplane geometry. Our focus will be on understanding how SVMs

improve upon the limitations of previous methods, particularly in achieving a unique solution when dealing with linear problems.

SVMs are a class of algorithms used for pattern classification and non-linear regression. They operate as linear machines, meaning they work with hyperplanes in the form  $W^T X + b$ . The foundations of SVMs are rooted in the work of Vladimir Vapnik and are deeply embedded in the theory of convex optimization.

The core idea behind SVMs is to construct a hyperplane that serves as a decision surface. Unlike Multi-Layer Perceptrons (MLPs) and Radial Basis Functions (RBFs), which can produce non-linear decision boundaries, SVMs aim for a hyperplane that maximizes the margin of separation between two classes. This means that the distance between the two classes should be as large as possible.

(Refer Slide Time: 13:47)

In the case of linearly separable patterns, we need to derive a hyperplane that solves our objective.

In the case of non-linearly separable patterns, we need to lift the data points to a higher dimension so that we can still derive a hyperplane that solves our objective.

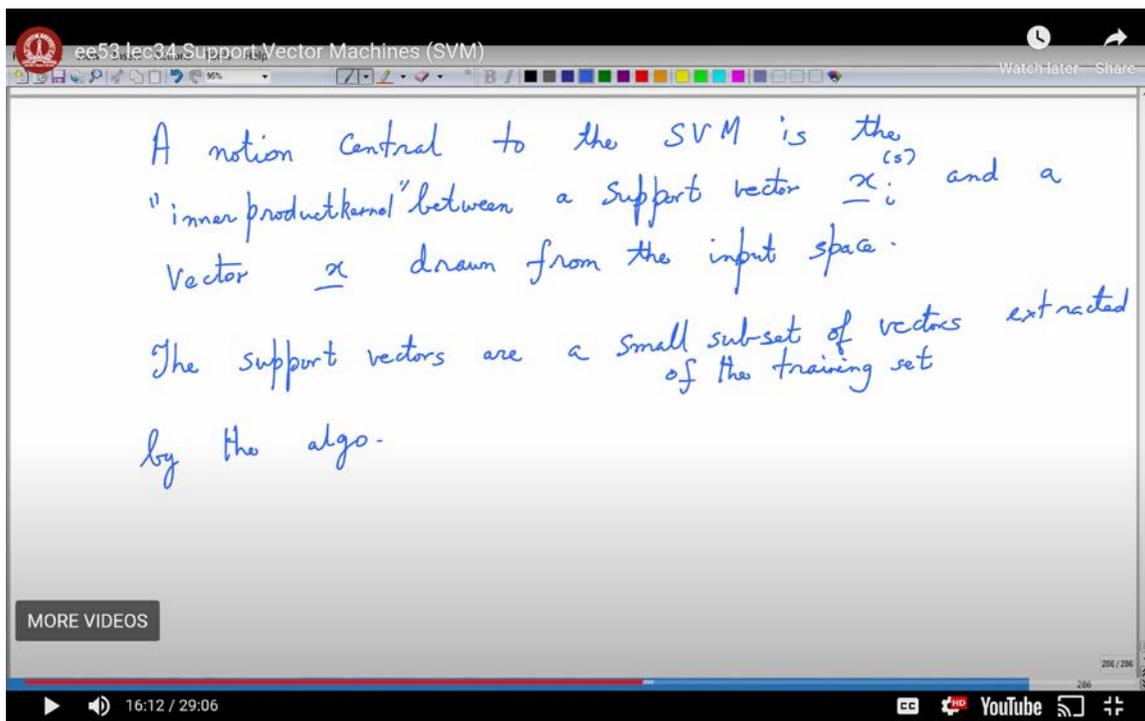
The diagram illustrates the mapping of data points from a 2D space ( $R^2$ ) to a 3D space ( $R^3$ ). In  $R^2$ , two classes of points (circles and crosses) are not linearly separable. In  $R^3$ , the points are lifted, and a hyperplane (indicated by a red line) successfully separates the two classes.

For linearly separable patterns, we need to find a hyperplane that achieves this objective. Multiple hyperplanes could theoretically separate the two classes, but we seek the one that maximizes the margin of separation. This is based on the principle of structural risk

minimization, where we formulate an objective function for each class, often labeled as +1 and -1. The goal is to derive the hyperplane that minimizes a risk function, which depends on the data points and their associated labels.

In cases where patterns are not linearly separable, we need to transform the data into a higher-dimensional space. By doing so, we can find a hyperplane in this new dimension that effectively separates the classes. For example, if points in a two-dimensional space ( $R_2$ ) are not separable, transforming them to a three-dimensional space ( $R^3$ ) might make it possible to find a separating hyperplane.

(Refer Slide Time: 16:12)



A notion central to the SVM is the "inner product kernel" between a support vector  $x_i^{(s)}$  and a vector  $x$  drawn from the input space.

The support vectors are a small sub-set of vectors extracted of the training set by the algo.

MORE VIDEOS

16:12 / 29:06

A key concept in SVMs is the computation of the inner product between support vectors and vectors from the input space. Support vectors are a small, crucial subset of the training data, selected by the algorithm. Instead of merely calculating the inner product, we use an inner product kernel to measure the relationship between a support vector and a vector from the input space. The role of support vectors and their geometry will be discussed in further detail in the following sections.

Let's delve into the geometry of our problem. Our goal is to identify an optimal hyperplane for linearly separable patterns. To achieve this, we start with a set of training samples. Specifically, let's denote these training samples as  $x_i$  and  $d_i$ , where  $i$  ranges from 1 to  $N$ . Here,  $x_i$  represents the input pattern for the  $i$ -th example, and  $d_i$  signifies the target or desired response. We assume that the patterns represented by  $d_i$ , which take values of  $+1$  or  $-1$ , are linearly separable. This assumption is crucial as we are focusing on finding optimal hyperplanes for linearly separable patterns.

The equation of the decision surface, or hyperplane, is given by  $W^T X + b = 0$ . For  $d_i = +1$ , we require  $W^T x_i + b \geq 0$ , while for  $d_i = -1$ , we need  $W^T x_i + b < 0$ .

(Refer Slide Time: 19:16)

Optimal hyperplane for linearly separable patterns

Consider the training samples  $\{x_i, d_i\}_{i=1}^N$

$x_i$  → i/p pattern for the  $i$ -th example  
 $d_i$  → target

Assume that the patterns represented by  $d_i = \{+1, -1\}$  is linearly separable

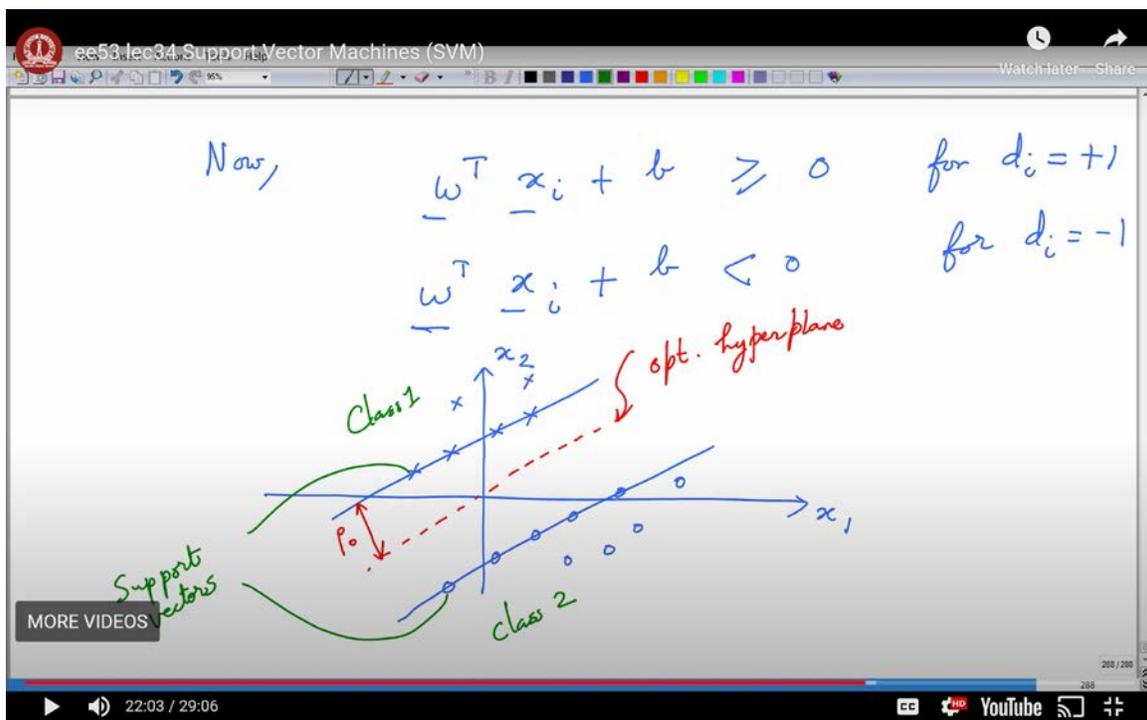
The eqn of the decision surface is  $W^T x + b = 0$

To visualize this, consider a two-dimensional plane where points are plotted as crosses and circles, representing two different classes. The optimal hyperplane is the one that maximizes the margin of separation, denoted as  $\rho_0$ , between these two classes. The points lying exactly on the boundaries of this margin are called support vectors. These support

vectors are crucial data points extracted by the algorithm that lie on the margins of the decision boundary.

While the perceptron algorithm can provide multiple hyperplanes separating the two classes, our objective is to identify the single optimal hyperplane that maximizes the margin. To achieve this, let  $W_0$  and  $b_0$  denote the optimal values for the weight vector and bias, respectively. The equation of the decision boundary is thus given by  $W_0^T x + b_0 = 0$ .

(Refer Slide Time: 22:03)



We can define a discriminant function,  $g(x) = W_0^T x + b_0$ , which helps determine the classification of any given point  $x$ . Depending on the value of  $g(x)$ , we can classify the point as either +1 or -1.

From coordinate geometry, any vector  $x$  can be expressed as  $x = x_p + r \frac{W_0}{|W_0|}$ , where  $x_p$  is a point on the hyperplane, and  $r$  is the algebraic distance from  $x$  to the hyperplane.

(Refer Slide Time: 25:45)

ee53 lec34 Support Vector Machines (SVM)

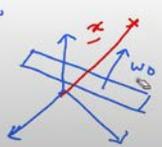
Let  $\underline{w}_0$  and  $b_0$  be the opt. values of the weights vector and the bias

$$\underline{w}_0^T \underline{x} + b_0 = 0 \quad \leftarrow \text{Eqn of the decision boundary}$$

Let us write the discriminant function as

$$g(\underline{x}) = \underline{w}_0^T \underline{x} + b_0$$

From our notion of the normal to a plane

$$\underline{x} = \underline{x}_p + r \frac{\underline{w}_0}{\|\underline{w}_0\|}$$


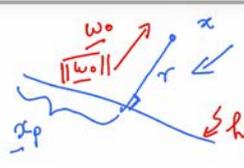
MORE VIDEOS

25:45 / 29:06

YouTube

(Refer Slide Time: 28:48)

ee53 lec34 Support Vector Machines (SVM)



algebraic distance of the point  $\underline{x}$  w.r.t plane

$$\underline{x} = \underline{x}_p + r \frac{\underline{w}_0}{\|\underline{w}_0\|}$$

normal projection of  $\underline{x}$  on to the hyperplane

$r$  is +ve if  $\underline{x}$  is on the +ve side of the hyperplane

-ve side of the hyperplane

MORE VIDEOS

28:48 / 29:06

YouTube

Here,  $W_0$  is the normal vector to the hyperplane, and  $r$  represents the perpendicular distance between the point  $x$  and the hyperplane. This geometric interpretation helps visualize how the point  $x$  is positioned relative to the hyperplane. Let's consider a two-dimensional plane where we have a point  $x$ . The algebraic distance of this point from the plane is along the direction of  $W_0$ .

To express any vector  $x$ , we can decompose it as  $x = x_p + r \frac{W_0}{|W_0|}$ . Here,  $x_p$  represents the projection of  $x$  onto the hyperplane. The vector  $\frac{W_0}{|W_0|}$  is the normalized direction of  $W_0$ , which gives us a unit vector in that direction. The scalar  $r$  is the distance from the point  $x$  to the hyperplane, scaled appropriately.

So,  $x_p$  is the projection of  $x$  onto the hyperplane, and the distance  $r$  is measured in the direction of  $W_0$ . If  $x$  lies on the positive side of the hyperplane,  $r$  is positive; if  $x$  lies on the negative side,  $r$  is negative. This completes our discussion on this topic.