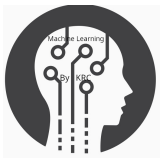
Machine Learning (Support Vector Machine (SVM))

Prof K R Chowdhary

CSE Dept., MBM University

December 03, 2024



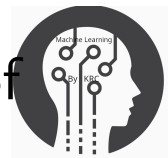
Introduction to SVM

$\Rightarrow$ SVM is a parametric supervisory learning model, solves same problem as logistic regression – a classification with two classes – and yields similar performance. SVMs are based on *associative learning*, these are geometrically motivated, and solve classification problems with attributes : -1 and $+1$.

$\Rightarrow$ Once trained, the algorithm builds a model that can assign new examples to -1 or $+1$ classes. SVM is used in predictive modeling applications, like, face detection, handwriting recognition, text classification.

$\Rightarrow$ Originally SVM was used to solve complex problems by simply reducing them to binary classification. Applications varied from text classifications, e.g., "Is this article related to my search query?", to bioinformatics, e.g., "Do these micro-array profiles indicate cancerous cells?"

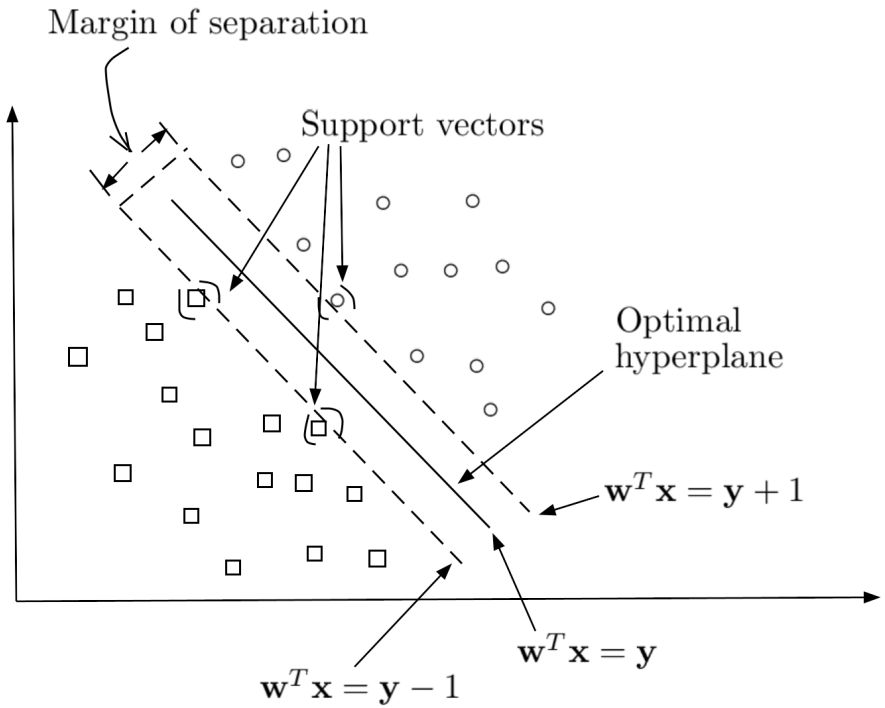
$\Rightarrow$ New approach of SVMs is based more on *statistical* foundations. Simple binary failed to exploit the structure of predicted objects.



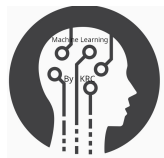
Introduction to SVM ...

$\Rightarrow$ Some examples of problems that SVM can solve:
- Are the symbols in a 2D plane bullet or box?
- Is review of a research paper positive or negative?
- Is the given image of a tiger or cat?

$\Rightarrow$ In 1st above, classification using SVMs is performed in a 2D space (Fig. 1), with *attributes* $+1$, labeled with circles, separating from those with *attribute* -1, labeled with boxes. The SVM works as a *linear classifier* (Fig 1) $\downarrow$.



SVMs are two-class classifier. The data consists objects labeled with one of two classes:$+1$ ($+$ve examples) or -1 (neg. examples).



SVM as Linear Classifier

$\Rightarrow$ Let variable $\mathbf{x}$ denotes a vector with components $\mathbf{x}_i$, and stands for i^{th} vector of a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where y_i is the label associated with $\mathbf{x}_i$.

$\Rightarrow$ Objects $\mathbf{x}_i$ are *patterns* or *examples*. The examples belong to some set $\mathbf{X}$. Initially examples are the vectors, but once we introduce *kernels*, this assumption will be relaxed, and they could be any discrete or continuous objects, for example, a protein/DNA sequence or some protein structure.

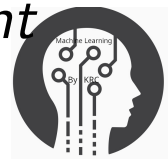
$\Rightarrow$ A key concept in a linear classifier is the *dot-product* between two vectors, referred as *scalar* product, and defined as,

$$\mathbf{w}^T\mathbf{x} = \sum_i w_i x_i. \qquad (1)$$

$\Rightarrow$ A linear classifier is based on a linear discriminant function of the form,

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b. \qquad (2)$$

$\Rightarrow$ The vector $\mathbf{w}$ is called *weight vector*, and b is called *bias*.



SVM as Linear Classifier...

$\Rightarrow$ For $b = 0$, set of points $\mathbf{x}$ such that $\mathbf{w}^T\mathbf{x} = 0$ are all points that are perpendicular to $\mathbf{w}$ and go through origin. It is a line in two dimensions, and a plane in three dimensions: a *hyperplane* in more than 3D. Bias b translates the hyperplane away from origin at distance b.

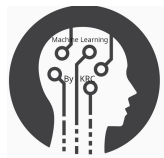
$$\{\mathbf{x} : f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0\} \quad (3)$$

divides the plane into two: sign of *discriminant function* $f(\mathbf{x})$ denotes side of hyperplane to which point belongs.

$\Rightarrow$ Boundary between regions "+" and "-" is *decision boundary* of classifier. Boundary is linear because it is linear in the input examples, as the equation (2) shows.

$\Rightarrow$ Classification using SVMs: Objects' attribute whose value is to be predicted, called *dependent attribute*, has two possible values: -1 and $+1$.

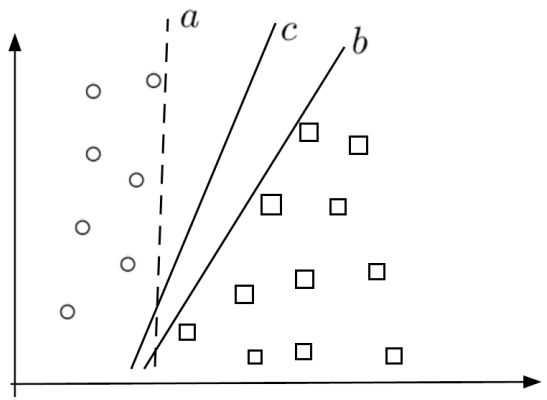
$\Rightarrow$ Optimal separating surface is computed by maximizing margin of separation (Fig. next page). Data point with $+1$ attribute are circle ($\circ$s) and with -1 are squares($\square$s).



SVM as Linear Classifier...

$\Rightarrow$ The fig. (page 3) above shows a problem separable in a 2-dimensional space, with margin of separation marked by broken lines. The margin of separation is a measure of safety (robustness) in separating the two sets of points, hence larger the margin, more robust is it, and less likely that noisy points get misclassified.

$\Rightarrow$ Fig. (right) shows three separator lines between set of data values of two categories, line c produces maximum margin. Computing the optimal separating surface in a standard SVM formulation requires solving an optimization problem, which is quadratic in nature.



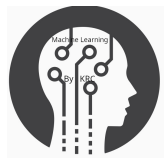
Linear Classifier Example

$\Rightarrow$ In the linear classifiers a straight line can be a model. Consider scenario where "height and weight" of number of people are available. Medical experts have found that some of them are over-weight or under-weight (see table 1). Accordingly, the data is divided into two classes, as shown in fig. 1: over-weight are marked as '$\square$' and under-weight by '$\circ$'.

$\Rightarrow$ We want to construct a model from available data. Then, for any new person, given weight and height, model could easily predict whether he/she is over- or under-weight.

Table 1: A training Set

Data$\rightarrow$	1	2	3	4	5	6
Weight(kg)	50	60	70	70	80	90
Height(cm)	1.6	1.7	1.9	1.5	1.7	1.6
Over-wighted	No	No	No	Yes	Yes	Yes



Linear Classifier Example...

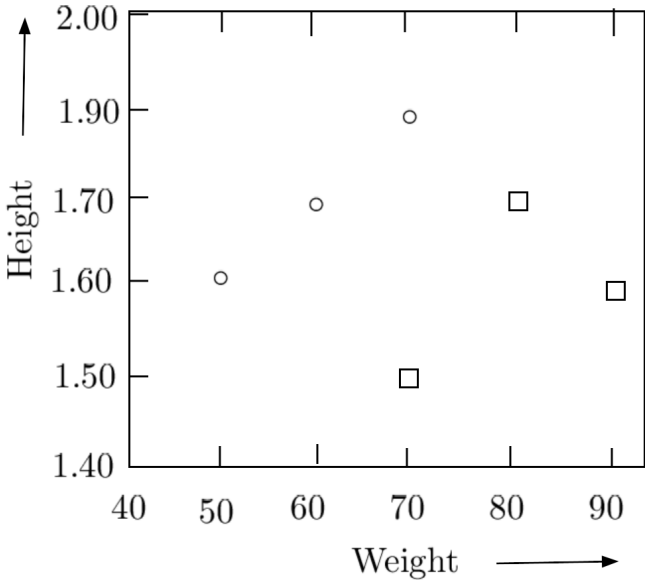


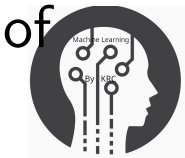
Figure 1: SVM training Set

$\Rightarrow$ A model can be a rule like: *If weight* ≥ 60, *then over-weight.*

$\Rightarrow$ However, it is not useful, as some tall people may be thin but their weights are > 60 kg. Better model is: *If weight/(height)*$^2 > 23$, *then over-weight.*

$\Rightarrow$ Classification objective: Identify a good model, so that future prediction are accurate. "weight" and "height" are *features* or *attributes*, (variables in statistics).

$\Rightarrow$ Each person is a data instance. Vector $\mathbf{x} = [weight, height]$ is a data instance, $y = 1$ or -1, a label of each instance.



Linear Classifier Example...

$\Rightarrow$ Six training instances are $x_1, \ldots, x_6$, and labels $y = [-1, -1, -1, 1, 1, 1]^T$. T stands for transpose[1] of matrix, -1 and 1 means under-weight ($\circ$), over-weight ($\square$), respectively.

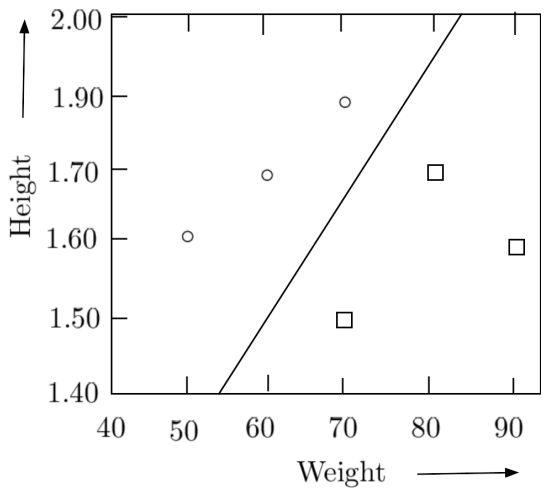


Figure 2: SVM as a linear Classifier

We will show that a straight line can be a model to classify the future data. Fig. 2 shows that a line, separating training data of over-weight and under-weight persons, can be represented by,

$$0.2 \times weight - 10 \times height + 3 = 0, \quad (4)$$

which is similar to the general form of a classifier we defined earlier, can be represented using vectors $\mathbf{w}$ and $\mathbf{x}$ as,

$$\mathbf{w}^T \mathbf{x} + b = 0, \quad (5)$$

[1] If $\mathbf{x} = [1, 2]$, then $\mathbf{x}^T$ is a column matrix of $\mathbf{x}$

Linear Classifier Example...

$\Rightarrow$ where $\mathbf{x} = [weight, height]^T$, $\mathbf{w} = [0.2, -10]^T$, and $b = 3$. Then, for any new data $\mathbf{x}$, we check whether it is on left or right side of the line. That is,

$$\begin{aligned} &if\ \mathbf{w}^T\mathbf{x} + b > 0 \\ &\quad \text{predict } \mathbf{x} \text{ as "over-wt.",} \\ &\quad < 0 \text{ predict } \mathbf{x} \text{ as "under-wt.} \end{aligned} \quad (6)$$

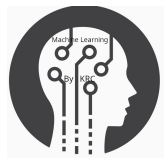
Above equation represents a hyperplane in a multidimensional space, where:

- $\mathbf{w}$: weight vector (coefficients for each dimension),
- $\mathbf{x}$: input vector (a point in the space), and
- b : *bias term* (a constant).

$\Rightarrow$ *Hyperplane* is a flat affine subspace in an n-dimensional space. In two dimensions, this would be a line; in three dimensions, a plane.

$\Rightarrow$ Term $\mathbf{w}^T\mathbf{x}$ is dot product of vectors $\mathbf{w}^T$ and $\mathbf{x}$, and calculates a single scalar value: the projection of $\mathbf{x}$ onto $\mathbf{w}$.

$\Rightarrow$ $\mathbf{w}^T$ is transpose of $\mathbf{w}$. Bias term b shifts the hyperplane away from the origin



Linear Classifier Example...

$\Rightarrow$ The equation $\mathbf{w}^T\mathbf{x} + b = 0$, describes the set of points $\mathbf{x}$ that lie exactly on the hyperplane. Points for which $\mathbf{w}^T\mathbf{x} + b > 0$ are on one side of the hyperplane, and for which $\mathbf{w}^T\mathbf{x} + b < 0$ are on the other side of the hyperplane.

$\Rightarrow$ Apart from support vector machines, the equation $\mathbf{w}^T\mathbf{x} + b = 0$ is also used in algorithms of linear regression, to separate or fit data in a high-dimensional space.

Example: Let the weight vector be $\mathbf{w} = [0.2, -10]^T$, bias $b = 3$, and attribute vectors be:

1. $\mathbf{x} = [80, 1.6]^T$, and
2. $\mathbf{x} = [70, 1.9]^T$.

Classify over/under weight.

Solution. We will make use of equation (6).

First case:

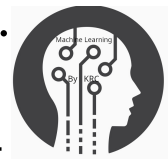
$$\mathbf{w}^T * \mathbf{x} + b$$

$$= [0.2, -10]^{T\,T} * [80, 1.6]^T + 3$$

$$= [0.2, -10] * \begin{bmatrix} 80 \\ 1.6 \end{bmatrix} + 3$$

$$= 0.2 \times 80 - 10 \times 1.6 + 3 = 3.$$

Hence, it is case of over-weight.



Linear Classifier Example...

Case two:

$$\mathbf{w}^T * \mathbf{x} + b$$

$$= [0.2, -10]^{T\,T} * [70, 1.9]^T + 3$$

$$= [0.2, -10] * \begin{bmatrix} 70 \\ 1.9 \end{bmatrix} + 3$$

$$= 0.2 \times 70 - 10 \times 1.9 + 3 = -2.$$

Hence, it is case of under-weight.