Support Vector Machine

Dr. Dinesh Kumar Vishwakarma

Professor,

Department of Information Technology,

Delhi Technological University, Delhi-110042

dinesh@dtu.ac.in

http://www.dtu.ac.in/Web/Departments/InformationTechnology/faculty/dkvishwakarma.php

Overview

- Introduction of Support Vector Machines (SVM)
- Linear Classifier
- Non Linear SVM
- Properties of SVM
- Multiclass SVM
- Weakness of SVM
- Applications
- Issues with SVM
- References

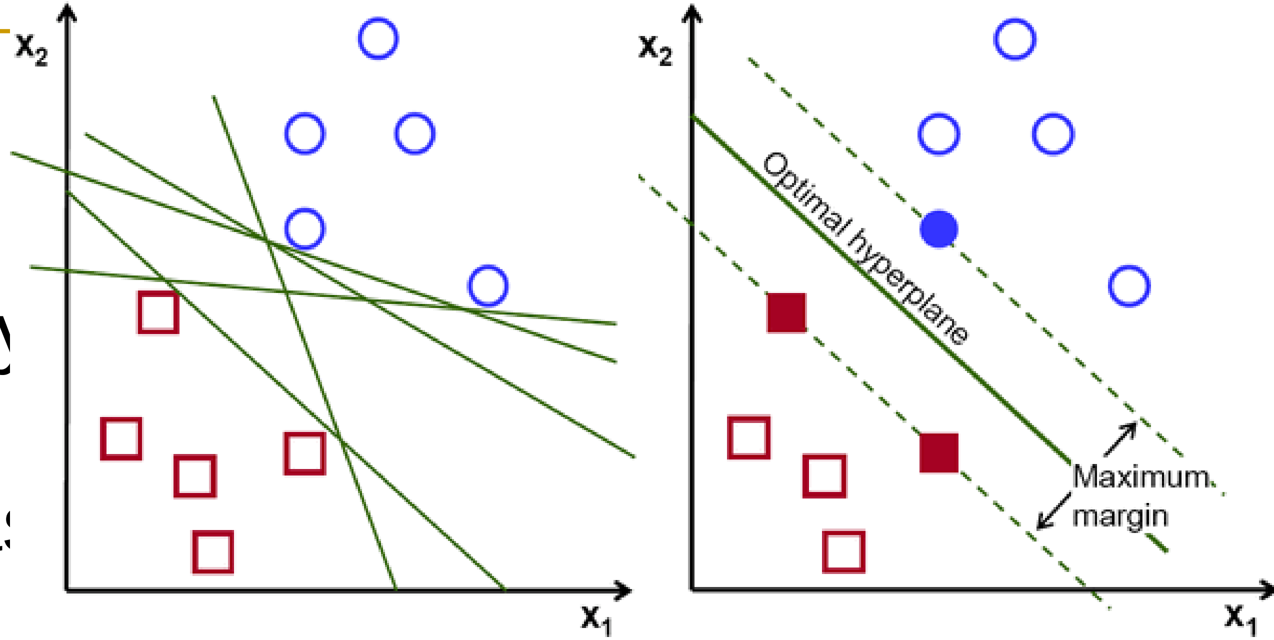
Introduction

Supervised learning algorithm

- SVM algorithm was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963.
- The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.
- Support vector machine is highly preferred because of high accuracy with less computation.
- It can be used as regression and classification but it is widely used as classifier.

Objectives

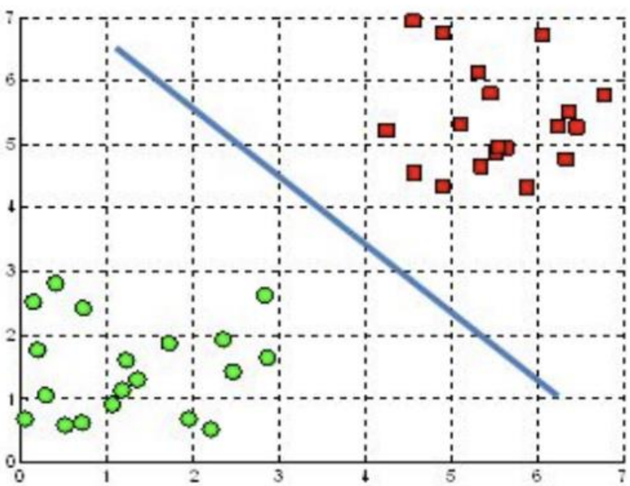
- To find a hy
space(N —
distinctly cla
classes.



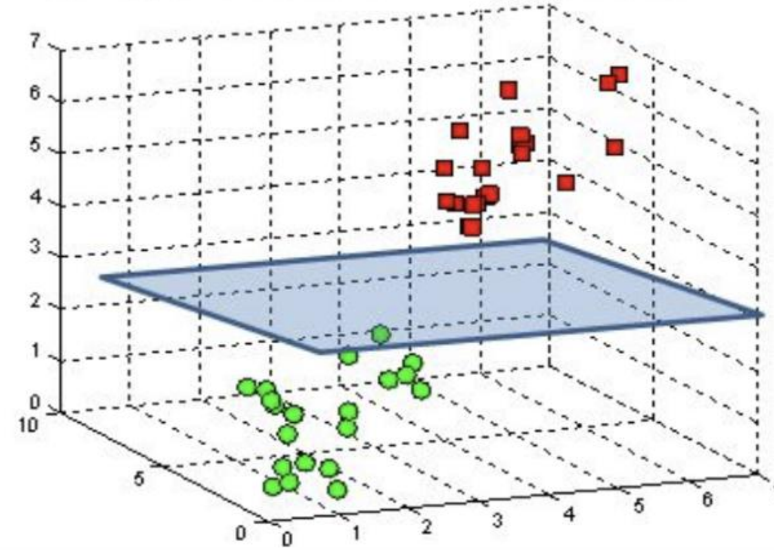
- To find a plane that has the maximum margin, i.e the maximum distance between data points of both classes.
- Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

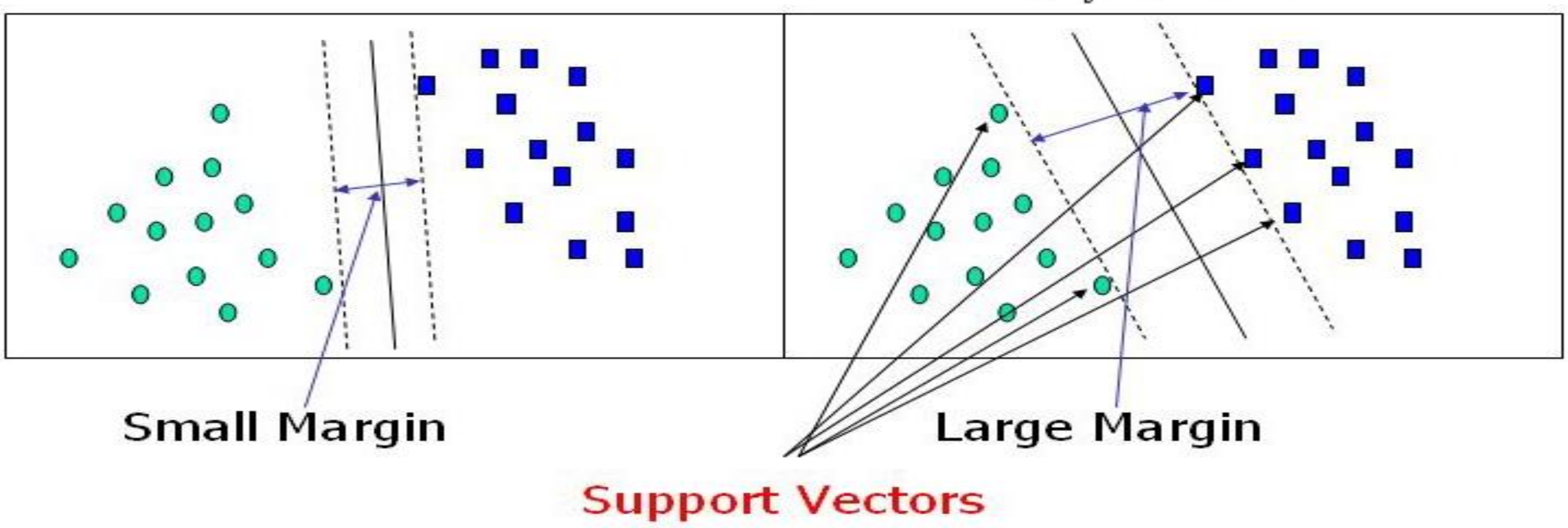
Hyperplanes and Support Vectors

A hyperplane in $\mathbb{R}^2$ is a line

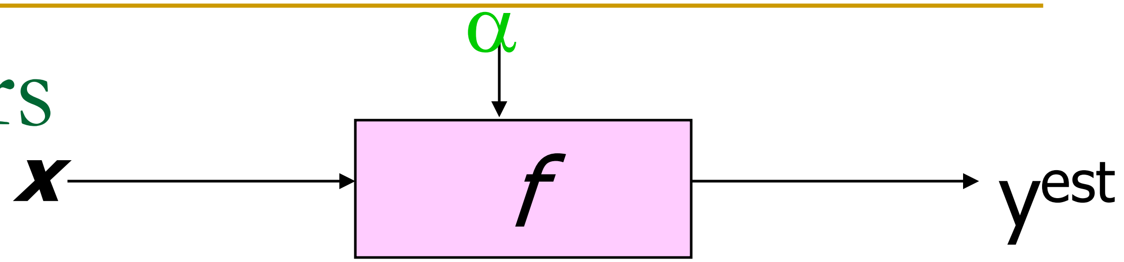


A hyperplane in $\mathbb{R}^3$ is a plane



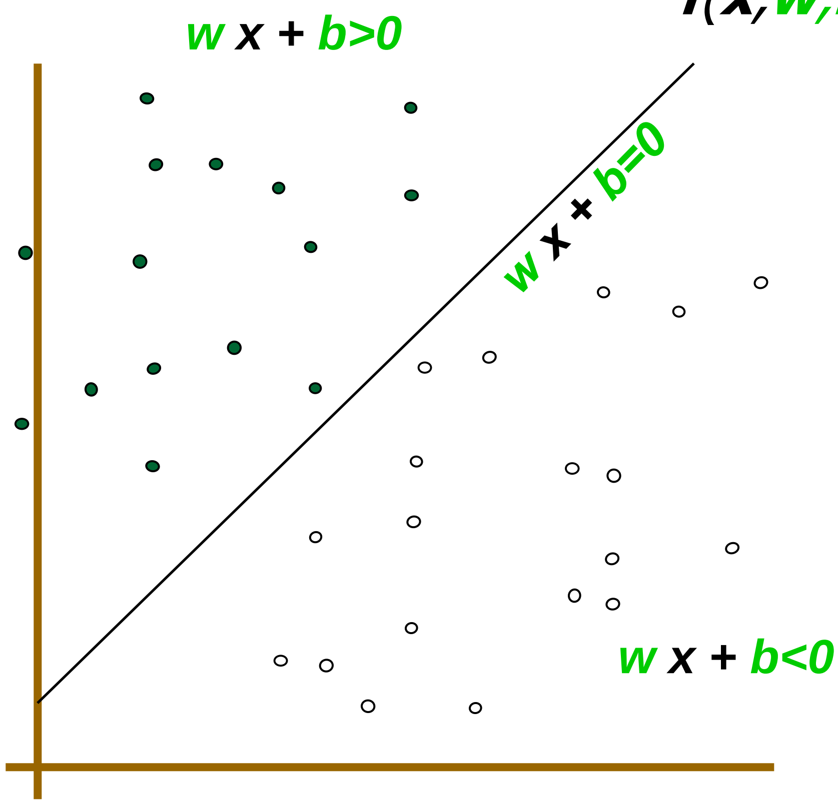


Linear Classifiers



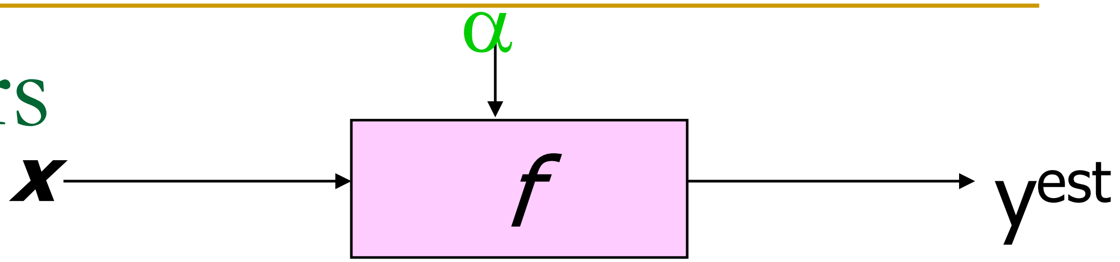
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- denotes +1
- denotes -1



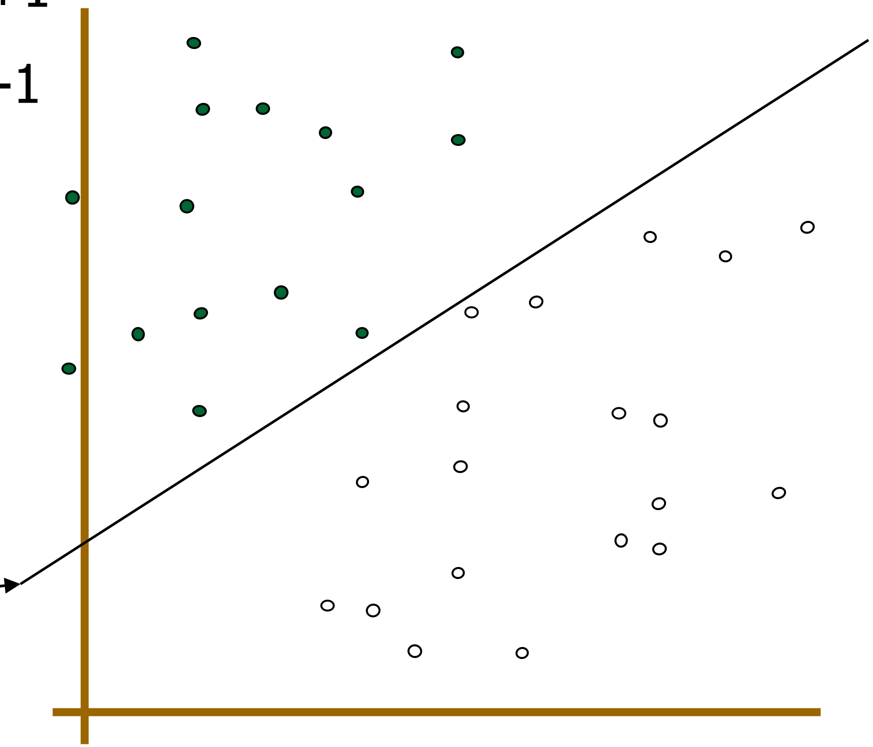
How would you classify this data?

Linear Classifiers



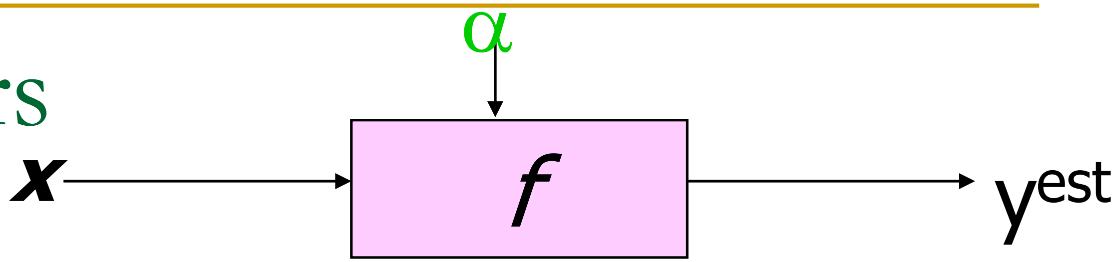
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$

- ● denotes +1
- ○ denotes -1

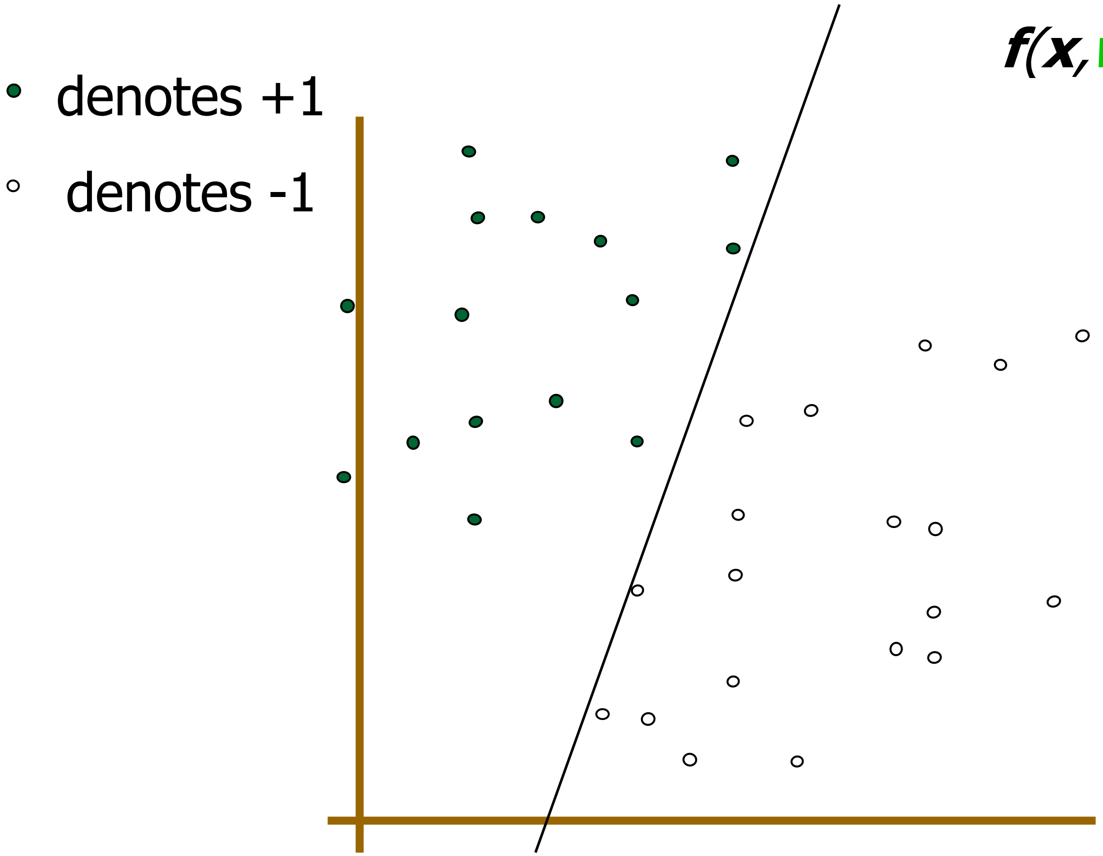


How would you classify this data?

Linear Classifiers

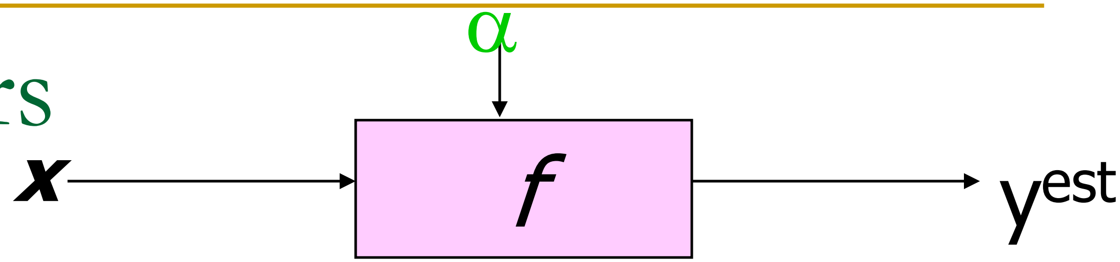


$$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\ \mathbf{x} + b)$$

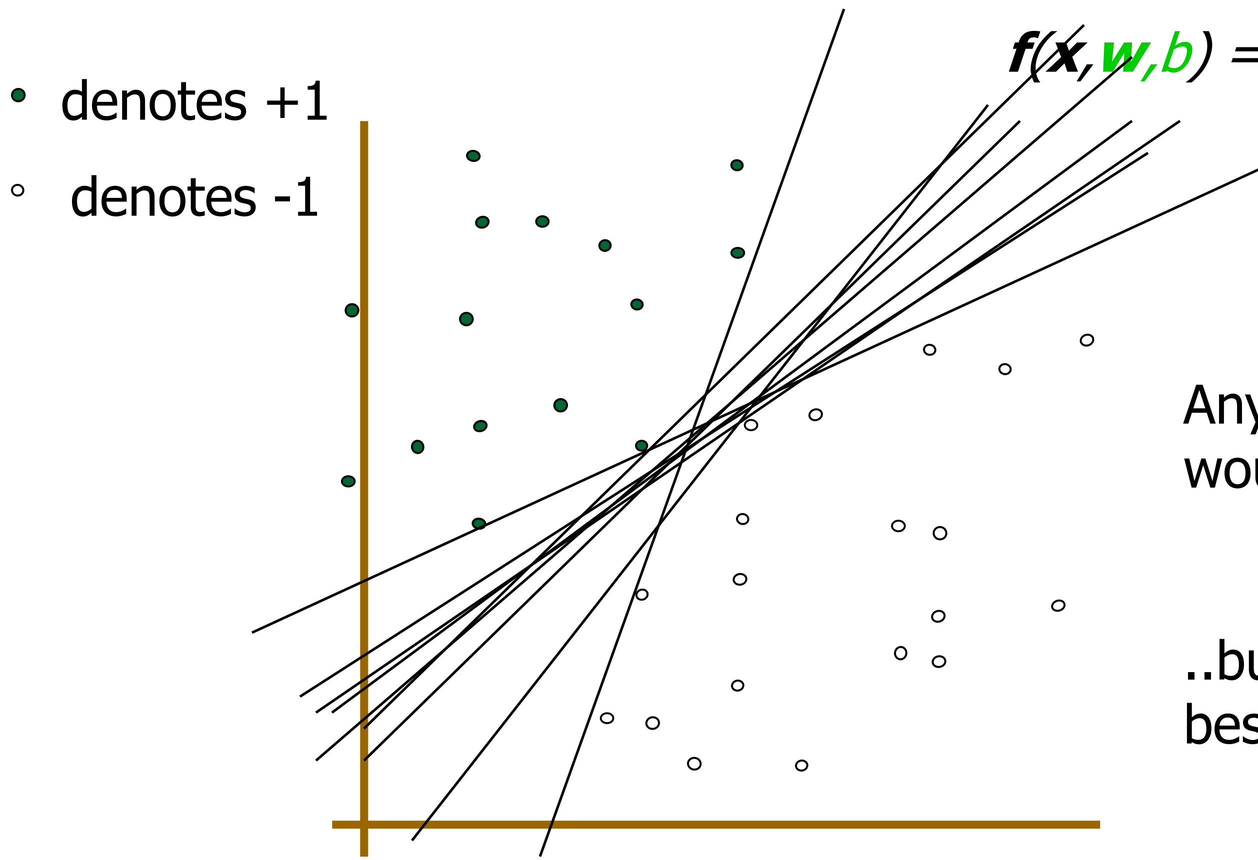


How would you classify this data?

Linear Classifiers



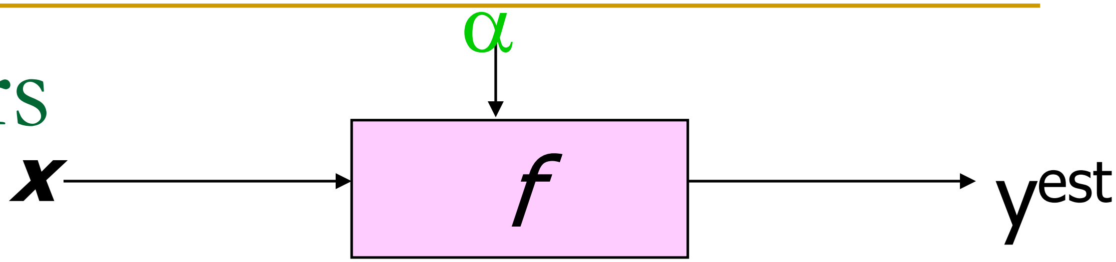
$$f(\boldsymbol{x}, \boldsymbol{w}, b) = sign(\boldsymbol{w}\, \boldsymbol{x} + b)$$



Any of these would be fine..

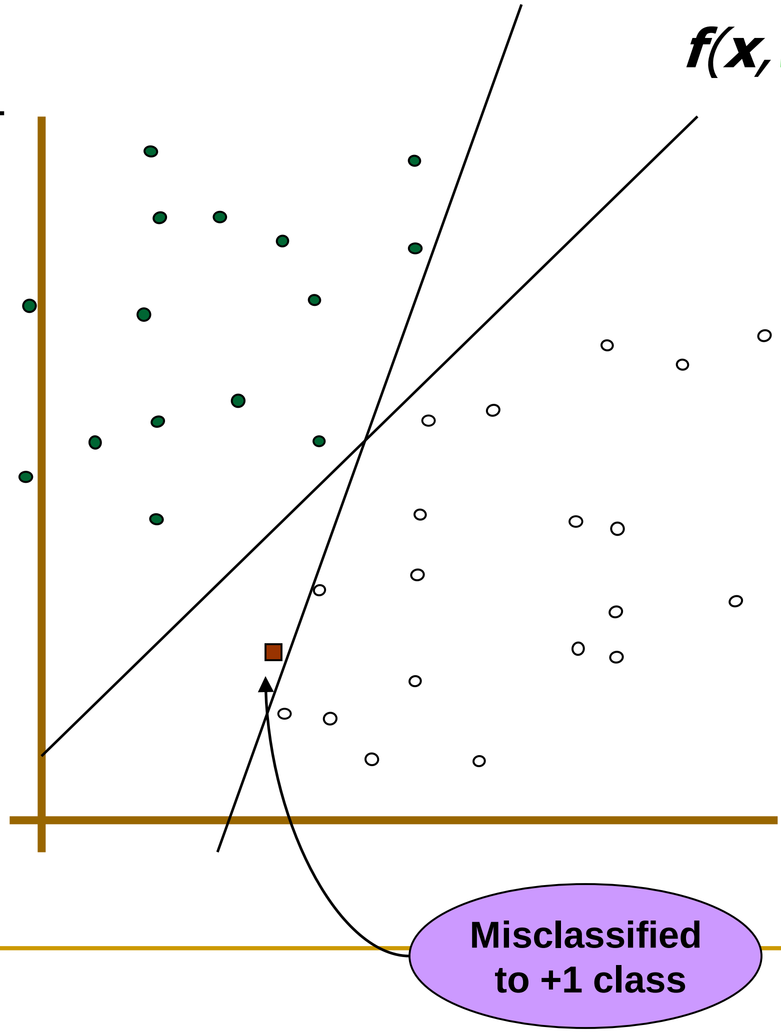
..but which is best?

Linear Classifiers



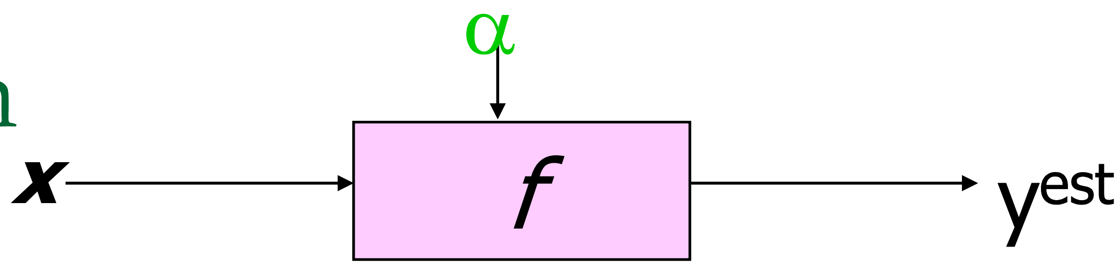
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$

- denotes +1
- denotes -1



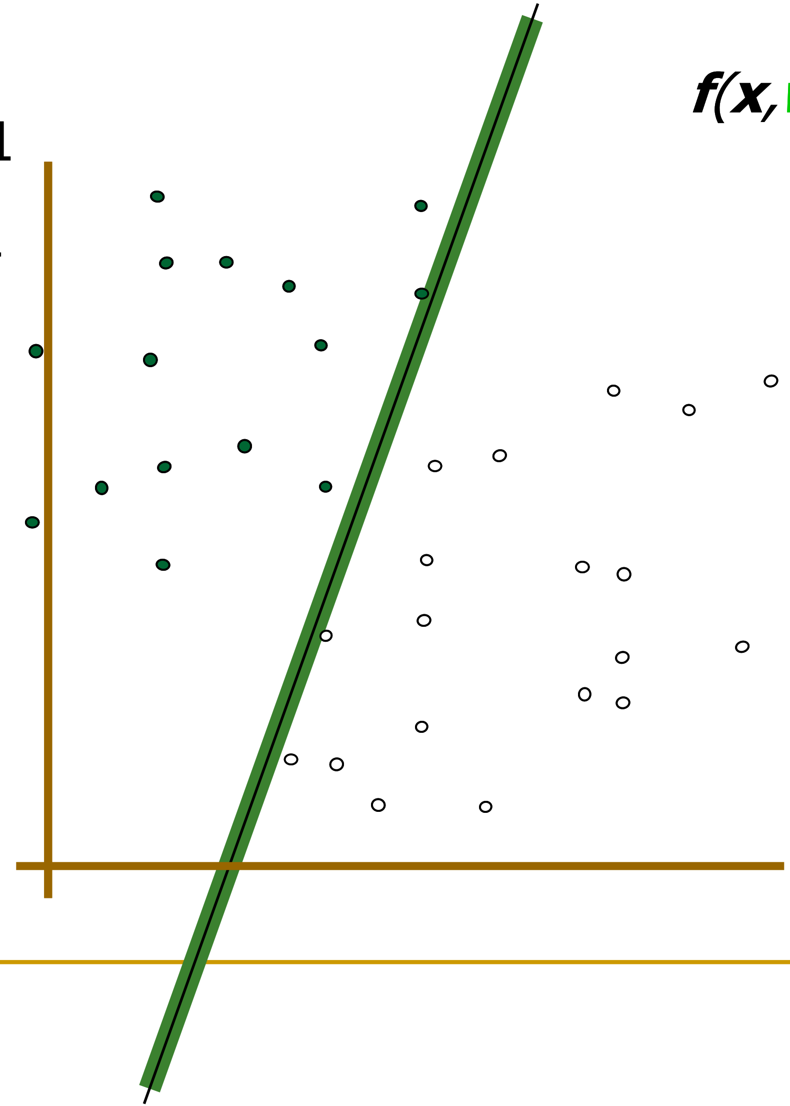
How would you classify this data?

Classifier Margin



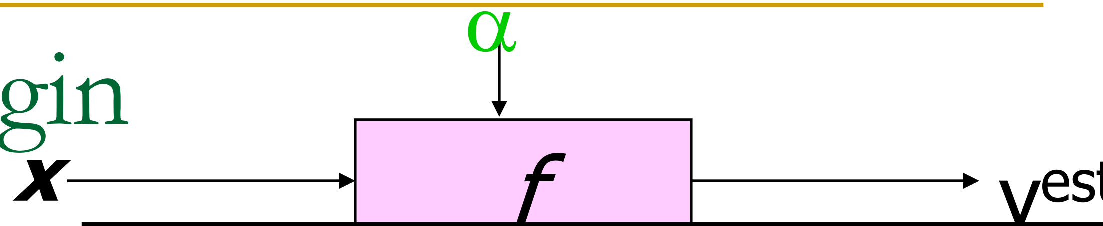
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$

- • denotes +1
- ∘ denotes -1



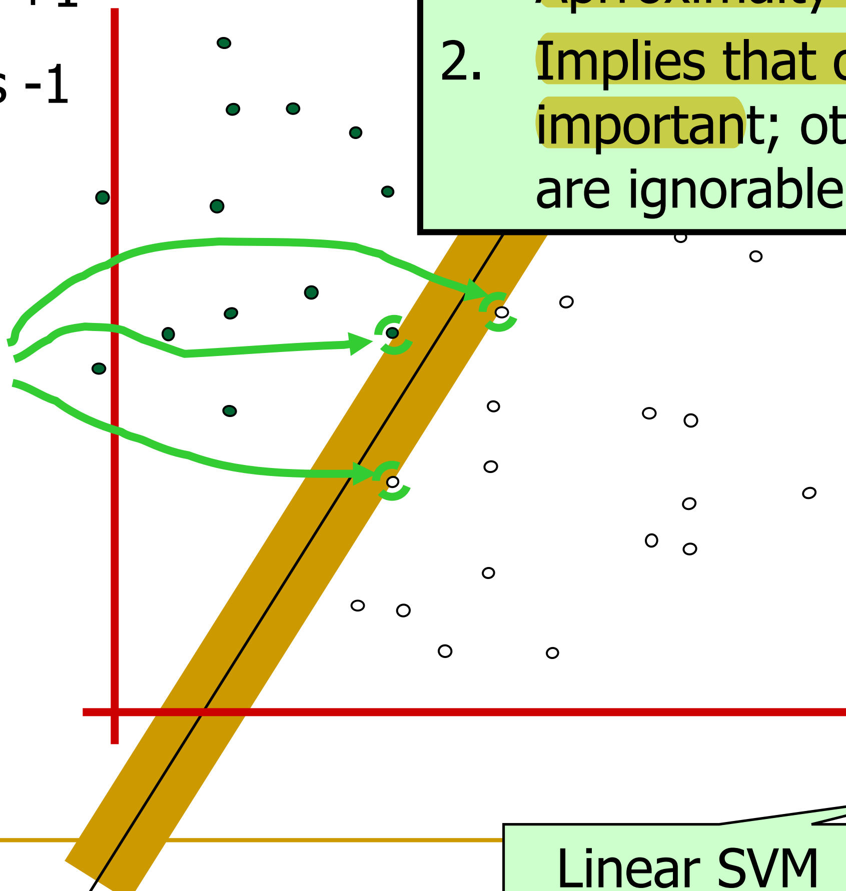
Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin



1. Maximizing the margin is good according to intuition and Probability Aprroximalty Correct.
2. Implies that only support vectors are important; other training examples are ignorable.

- denotes +1
- ◦ denotes -1



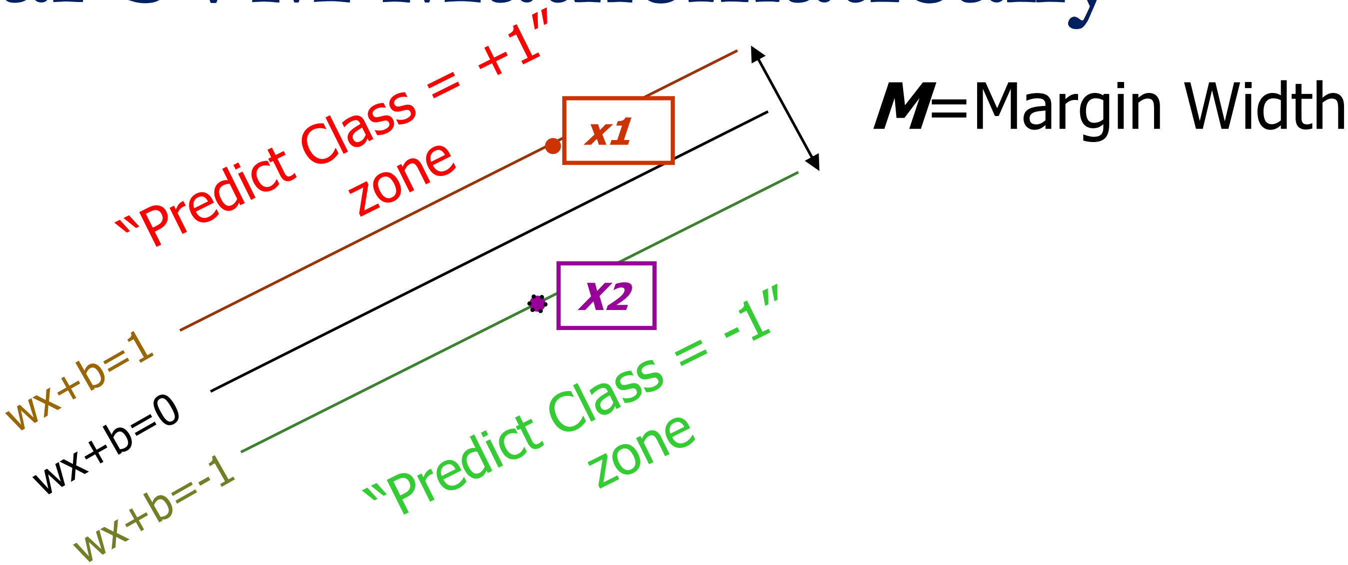
Support Vectors are those datapoints that the margin pushes up against

classifier is the linear classifier with the, um, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Linear SVM

Linear SVM Mathematically



$$\Rightarrow \mathbf{w}^T\mathbf{x_2} + b = 1 \text{ where } \mathbf{x_2} = \mathbf{x_1} + \lambda\mathbf{w}$$

$$\Rightarrow \mathbf{w}^T(\mathbf{x_1} + \lambda\mathbf{w}) + b = 1$$

$$\Rightarrow \mathbf{w}^T\mathbf{x_1} + b + \lambda\mathbf{w}^T\mathbf{w} = 1 \text{ where } \mathbf{w}^T\mathbf{x_1} + b = -1$$

$$\Rightarrow -1 + \lambda\mathbf{w}^T\mathbf{w} = 1$$

$$\Rightarrow \lambda\mathbf{w}^T\mathbf{w} = 2$$

$$\Rightarrow \lambda = \frac{2}{\mathbf{w}^T\mathbf{w}} = \frac{2}{\|\mathbf{w}\|^2}$$

And so, the distance $\lambda\|\mathbf{w}\|$ is $\frac{2}{\|\mathbf{w}\|^2}\|\mathbf{w}\| = \frac{2}{\|\mathbf{w}\|} = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$

$$\lambda = \frac{2}{w^Tw} = \frac{2}{\|w\|^2}$$

$$\text{Distance} = \lambda\|w\| = \frac{2}{\|w\|^2}\|w\| = \frac{2}{\|w\|} = \frac{2}{\sqrt{w^Tw}}$$

Linear SVM Mathematically

- Goal: **1) Correctly classify all training data**

$$wx_i + b \geq 1 \quad \textit{if } y_i = +1$$

$$wx_i + b \leq 1 \quad \textit{if } y_i = -1$$

$$y_i(wx_i + b) \geq 1 \quad \text{for all } i$$

2) Maximize the Margin $M = \dfrac{2}{|w|}$

same as minimize $\dfrac{1}{2} w^t w$

- **We can formulate a Quadratic Optimization Problem and solve for w and b**

- Minimize $\Phi(w) = \dfrac{1}{2} w^t w$

  subject to $y_i(wx_i + b) \geq 1 \qquad \forall i$

Solving the Optimization Problem

Find **w** and b such that
$\mathbf{\Phi(w)} = \frac{1}{2}\, \mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i (\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- **Need to optimize a *quadratic* function subject to *linear* constraints.**
- **Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.**
- **The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:**

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

The Optimization Problem Solution

- The solution has the form:

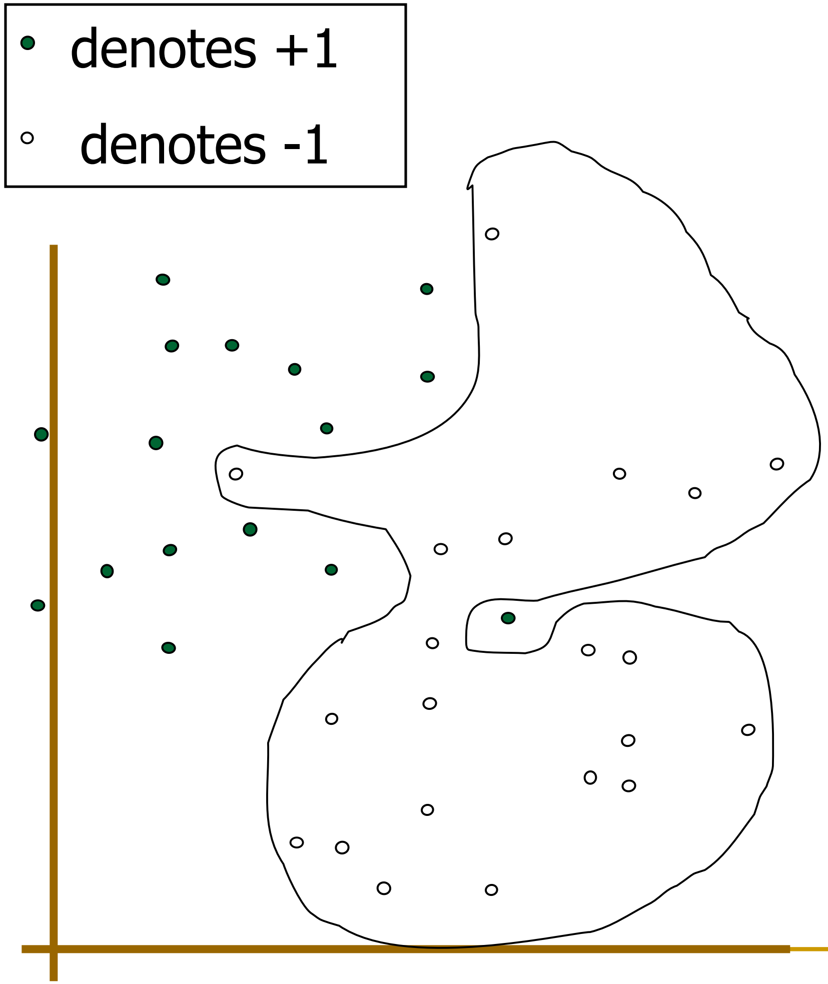
$$\boxed{\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w^T x_k} \text{ for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0}$$

- Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.
- Then the classifying function will have the form:

$$\boxed{f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i^T x} + b}$$

- Notice that it relies on an *inner product* between the test point $\mathbf{x}$ and the support vectors $\mathbf{x_i}$ – we will return to this later.
- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x_i^T x_j}$ between all pairs of training points.

Dataset with noise



- **Hard Margin: So far we require all data points be classified correctly**

  - No training error

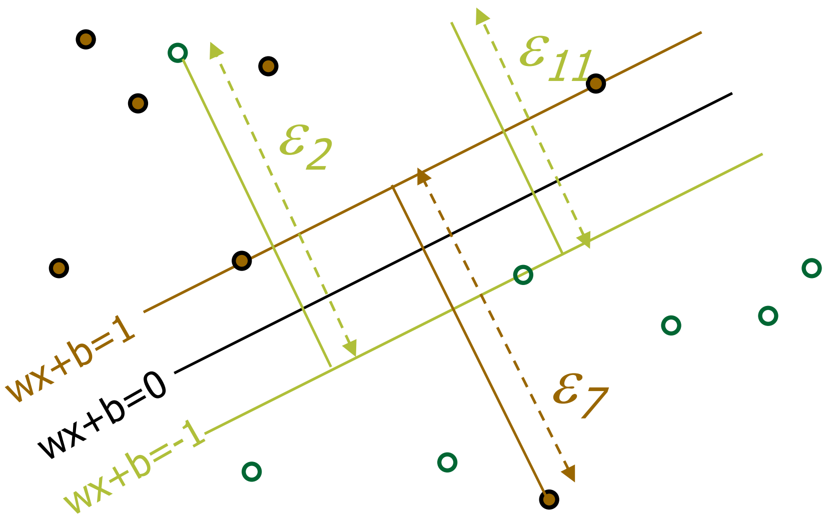
- **What if the training set is noisy?**

  - Solution 1: use very powerful kernels

OVERFITTING!

Soft Margin Classification

Slack variables ξi **can be added to allow misclassification of difficult or noisy examples.**



What should our quadratic optimization criterion be?

Minimize

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$

Hard Margin v.s. Soft Margin

- **The old formulation:**

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i\,(\mathbf{w^T x_i} + b) \geq 1$

- **The new formulation incorporating slack variables:**

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w} + C\sum\xi_i$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i\,(\mathbf{w^T x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

- **Parameter *C* can be viewed as a way to control overfitting.**

Linear SVMs: Overview

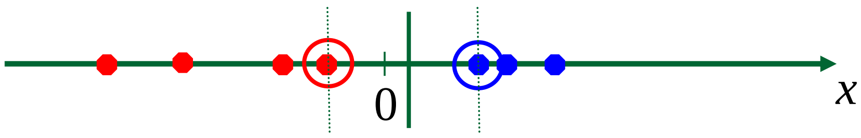
- **The classifier is a *separating hyperplane.***
- **Most "important" training points are support vectors; they define the hyperplane.**
- **Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers α_i.**
- **Both in the dual formulation of the problem and in the solution training points appear only inside dot products:**

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

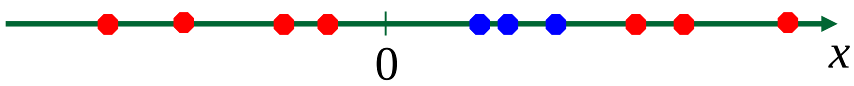
$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$$

Non-linear SVMs

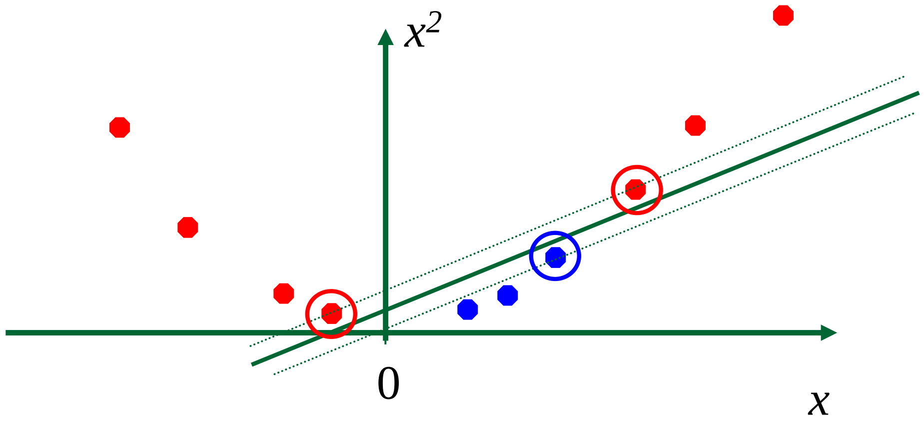
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

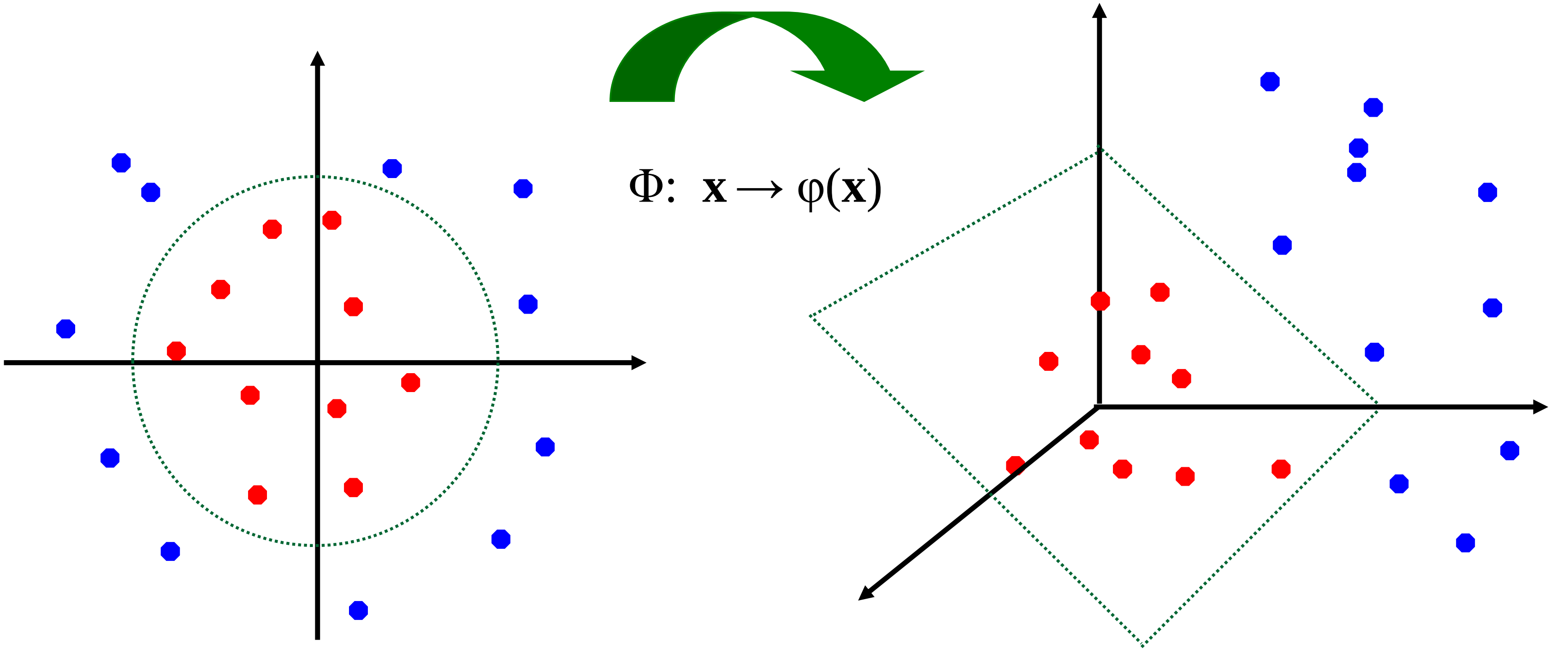


- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



The "Kernel Trick"

- **The linear classifier relies on dot product between vectors $K(x_i,x_j)=x_i^T x_j$**
- **If every data point is mapped into high-dimensional space via some transformation Φ: x → φ(x), the dot product becomes:**

$$K(x_i,x_j)= \varphi(x_i)^T\varphi(x_j)$$

- **A *kernel function* is some function that corresponds to an inner product in some expanded feature space.**
- **Example:**

2-dimensional vectors $x=[x_1\ \ x_2]$; let $K(x_i,x_j)=(1 + x_i^T x_j)^2$,

Need to show that $K(x_i,x_j)= \varphi(x_i)^T\varphi(x_j)$:

$$K(x_i,x_j)=(1 + x_i^T x_j)^2,$$
$$= 1+ x_{i1}^2 x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$
$$= [1\ \ x_{i1}^2\ \ \sqrt{2}\, x_{i1}x_{i2}\ \ x_{i2}^2\ \ \sqrt{2}x_{i1}\ \ \sqrt{2}x_{i2}]^T [1\ \ x_{j1}^2\ \ \sqrt{2}\, x_{j1}x_{j2}\ \ x_{j2}^2\ \ \sqrt{2}x_{j1}\ \ \sqrt{2}x_{j2}]$$
$$= \varphi(x_i)^T\varphi(x_j), \quad \text{where } \varphi(x) = [1\ \ x_1^2\ \ \sqrt{2}\, x_1x_2\ \ x_2^2\ \ \sqrt{2}x_1\ \ \sqrt{2}x_2]$$

What Functions are Kernels?

- **For some functions $K(x_i,x_j)$ checking that**

  $$K(x_i,x_j)= \varphi(x_i)^T\varphi(x_j) \text{ can be cumbersome.}$$

- **Mercer's theorem:**

  Every semi-positive definite symmetric function is a kernel

- **Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:**

K=

$K(\mathbf{x}_1,\mathbf{x}_1)$	$K(\mathbf{x}_1,\mathbf{x}_2)$	$K(\mathbf{x}_1,\mathbf{x}_3)$	…	$K(\mathbf{x}_1,\mathbf{x}_N)$
$K(\mathbf{x}_2,\mathbf{x}_1)$	$K(\mathbf{x}_2,\mathbf{x}_2)$	$K(\mathbf{x}_2,\mathbf{x}_3)$		$K(\mathbf{x}_2,\mathbf{x}_N)$
…	…	…	…	…
$K(\mathbf{x}_N,\mathbf{x}_1)$	$K(\mathbf{x}_N,\mathbf{x}_2)$	$K(\mathbf{x}_N,\mathbf{x}_3)$	…	$K(\mathbf{x}_N,\mathbf{x}_N)$

Examples of Kernel Functions

- Linear: $K(\mathbf{x_i},\mathbf{x_j}) = \mathbf{x_i}^T\mathbf{x_j}$

- Polynomial of power p: $K(\mathbf{x_i},\mathbf{x_j}) = (1+\mathbf{x_i}^T\mathbf{x_j})^p$

- Gaussian (radial-basis function network):

$$K(\mathbf{x_i},\mathbf{x_j}) = \exp\left(-\frac{\|\mathbf{x_i}-\mathbf{x_j}\|^2}{2\sigma^2}\right)$$

- Sigmoid: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0\mathbf{x_i}^T\mathbf{x_j} + \beta_1)$

Non-linear SVMs Mathematically

- **Dual problem formulation:**

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

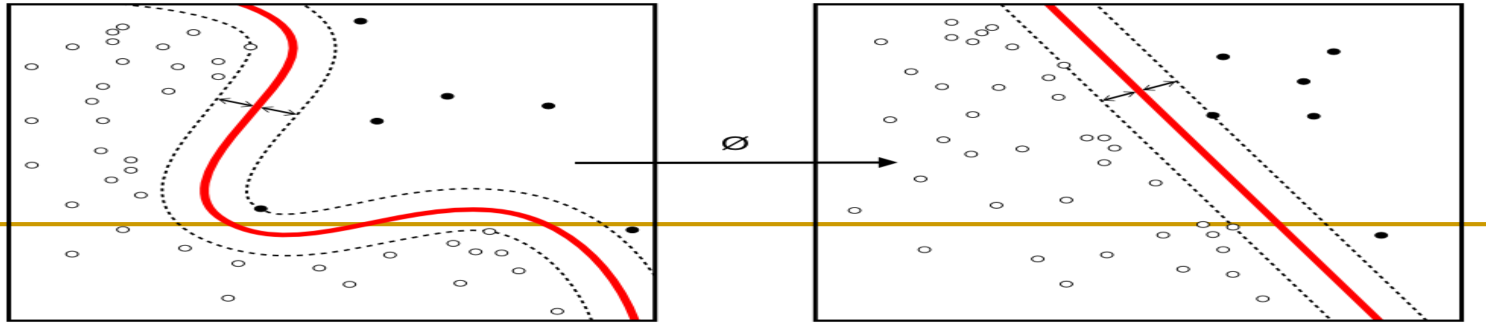
- **The solution is:**

$$\boxed{f(\mathbf{x}) = \Sigma\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b}$$

- **Optimization techniques for finding α_i's remain the same!**

Nonlinear SVM - Overview

- SVM locates a separating hyperplane in the feature space and classify points in that space
- It does not need to represent the space explicitly, simply by defining a kernel function
- The kernel function plays the role of the dot product in the feature space.



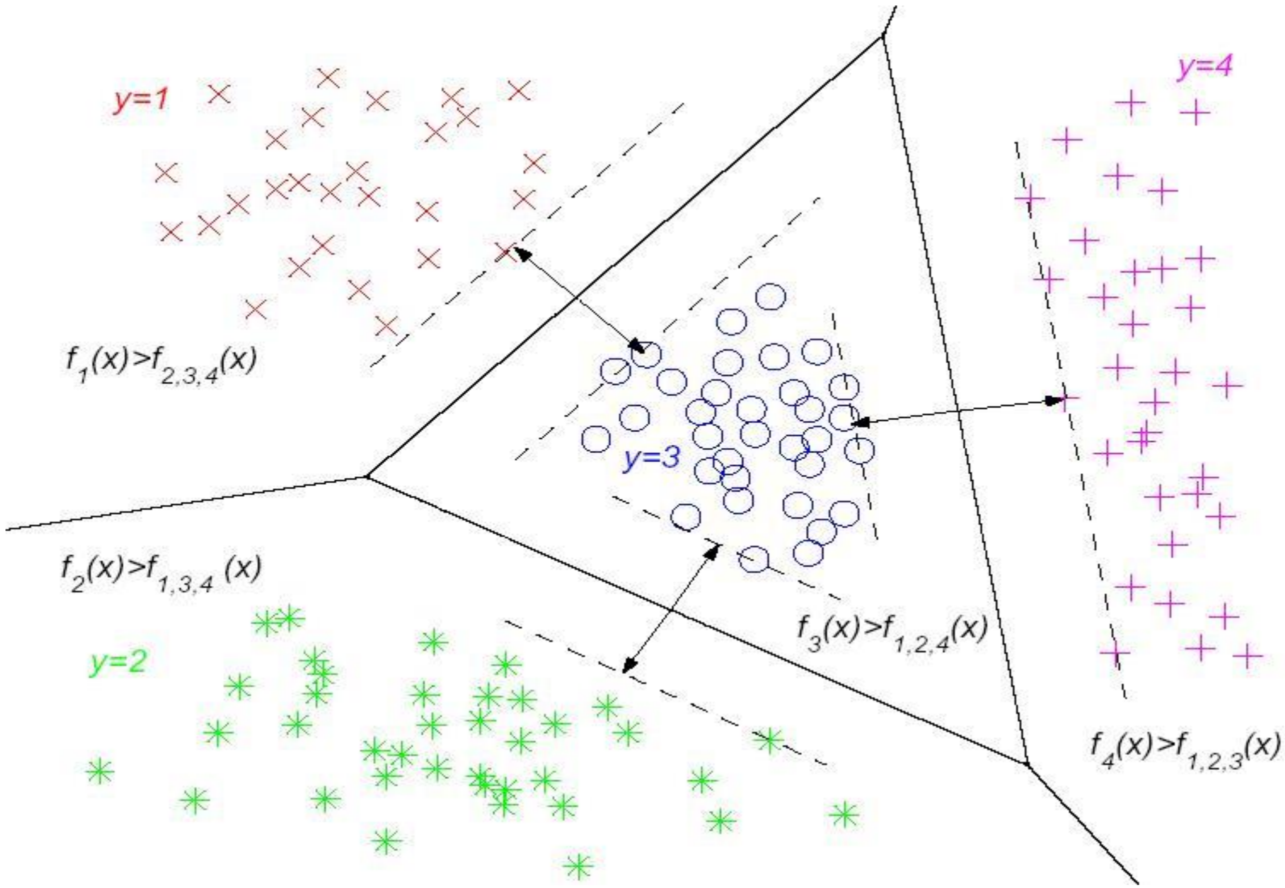
Properties of SVM

- **Flexibility in choosing a similarity function**
- **Sparseness of solution when dealing with large data sets**

  - only support vectors are used to specify the separating hyperplane
- **Ability to handle large feature spaces**

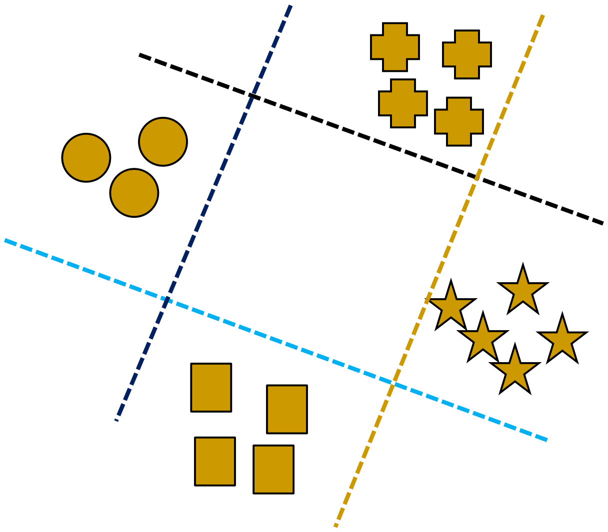
  - complexity does not depend on the dimensionality of the feature space
- **Overfitting can be controlled by soft margin approach**
- **Nice math property**
  - a simple convex optimization problem which is guaranteed to converge to a single global solution
- **Feature Selection**

Multi-class SVM



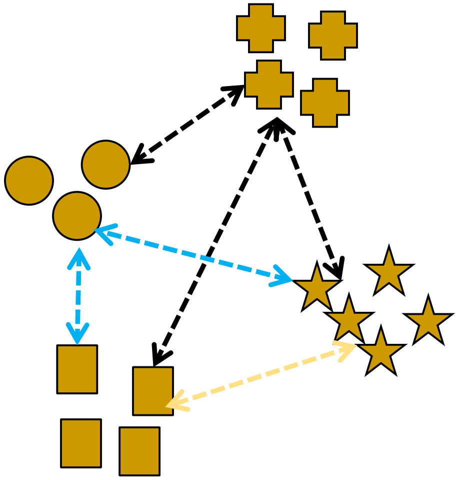
1 -against- All

- Or "one-against-rest", a tree algorithm
- Decomposed to a collection of binary classifications
- k decision functions, one for each class $(\mathbf{w}_k)^T \cdot \phi x + b_k,\ k \in Y$
- The kth classifier constructs a hyperplane between class n and the $\boldsymbol{k}$**-1** other classes
- Class of x = argmax$_i\{(w_i)^T \cdot \phi(x) + b_i\}$



1 -against- 1

- $\frac{\boldsymbol{k(k-1)}}{\mathbf{2}}$ classifiers where each one is trained on data from two classes
- For training data from i^{th} and j^{th} classes, run binary classification
- Voting strategy: If $sign\left(\left(w_{ij}\right)T.\varphi x + b_{ij}\right)$says x is in class i, then add 1 to class i else to j.
- Assign x to class with largest vote (Max wins)



$$\frac{4(4-1)}{2} = 6\ SVM$$

Multi-class SVM Approaches

1-against-all

- Each of the SVMs separates a single class from all remaining classes (Cortes and Vapnik, 1995)

1-against-1

- Pair-wise. $k(k\text{-}1)/2,\ k \in Y$ SVMs are trained. Each SVM separates a pair of classes (Fridman, 1996)

Performance similar in some experiments (Nakajima, 2000)

Time complexity similar: k evaluation in 1-all, k-1 in 1-1

Advantages of SVM

- Effective in high dimensional cases.
- Its memory efficient as it uses a subset of training points in the decision function called support vectors.
- Different kernel functions can be specified for the decision functions and its possible to specify custom kernels.
- Support vector machine works comparably well when there is an understandable margin of dissociation between classes.

Weakness of SVM

- It is sensitive to noise
  - A relatively small number of mislabeled examples can dramatically decrease the performance
  - SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
- In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
- As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.
- Support vector machine algorithm is not acceptable for large data sets.

SVM Applications

- **SVM has been used successfully in many real-world problems**

  - text (and hypertext) categorization

  - image classification

  - bioinformatics (Protein classification,
  Cancer classification)

  - hand-written character recognition

Application 1: Cancer Classification

- High Dimensional
  - p>1000; n<100

Genes				
Patients	g-1	g-2	......	g-p
P-1				
p-2				
....				
p-n				

- Imbalanced
  - less positive samples

$$K[x,x] = k(x,x) + \lambda \frac{n^+}{N}$$

- Many irrelevant features
- Noisy

FEATURE SELECTION

In the linear case, w_i^2 gives the ranking of dim i

SVM is sensitive to noisy (mis-labeled) data ☹

Application 2: Text Categorization

- Task: The classification of natural text (or hypertext) documents into a fixed number of predefined categories based on their content.

  - email filtering, web searching, sorting documents by topic, etc..

- A document can be assigned to more than one category, so this can be viewed as a series of binary classification problems, one for each category

Representation of Text

IR′s vector space model (aka bag-of-words representation)

- A doc is represented by a vector indexed by a pre-fixed set or dictionary of terms
- Values of an entry can be binary or weights

$$\phi_i(x) = \frac{\mathrm{tf}_i \log(\mathrm{idf}_i)}{\kappa},$$

- Normalization, stop words, word stems
- Doc x => $\boldsymbol{\varphi}(x)$

Text Categorization using SVM

- **The distance between two documents is $\varphi(x)\cdot\varphi(z)$**

- **$K(x,z) = \langle \varphi(x)\cdot\varphi(z)$ is a valid kernel, SVM can be used with $K(x,z)$ for discrimination.**

- **Why SVM?**

-High dimensional input space

-Few irrelevant features (dense concept)

-Sparse document vectors (sparse instances)

-Text categorization problems are linearly separable

Some Issues

- **Choice of kernel**
  - Gaussian or polynomial kernel is default
  - if ineffective, more elaborate kernels are needed.
  - domain experts can give assistance in formulating appropriate similarity measures
- **Choice of kernel parameters**
  - e.g. σ in Gaussian kernel
  - σ is the distance between closest points with different classifications.
  - In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters.
- **Optimization criterion – Hard margin v.s. Soft margin**
  - a lengthy series of experiments in which various parameters are tested
- **Requires full labeling of input data**
- **Parameters of a solved model are difficult to interpret.**
- **The SVM is only directly applicable for two-class tasks**

Additional Resources

- **An excellent tutorial on VC-dimension and Support Vector Machines:**

  C.J.C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.

- **The VC/SRM/SVM Bible:**

  Statistical Learning Theory by Vladimir Vapnik, Wiley-Interscience; 1998

http://www.kernel-machines.org/

Reference

- **Support Vector Machine Classification of Microarray Gene Expression Data**, Michael P. S. Brown William Noble Grundy, David Lin, Nello Cristianini, Charles Sugnet, Manuel Ares, Jr., David Haussler
- www.cs.utexas.edu/users/mooney/cs391L/svm.**ppt**
- **Text categorization with Support Vector Machines:**
  learning with many relevant features

  T. Joachims, ECML - 98

Thank You

Contact: dinesh@dtu.ac.in
Mobile: +91-9971339840

Questions

Problem 1

- Consider a hyperplane defined by a line $y = x_1 - 2x_2$.
- Determine the correct prediction of following data points.
  - (1, 0)
  - (1,1)

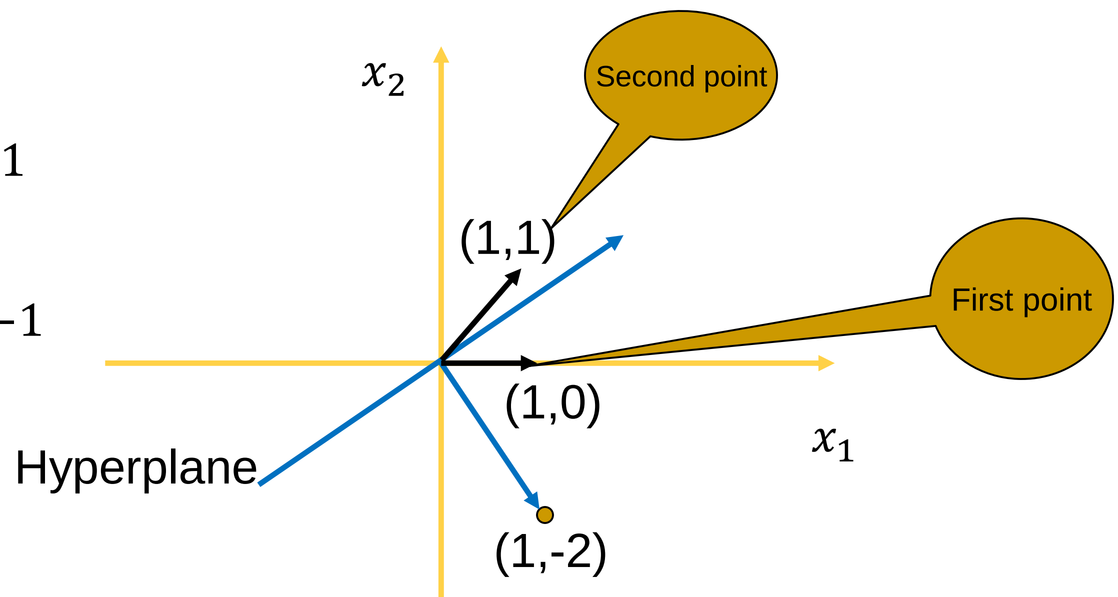
Solutions

- Draw the hyperplane $y = x_1 - 2x_2 = w^T x$.

Calculation

Case 1: $\begin{bmatrix} 1 \\ -2 \end{bmatrix} [1 \quad 0] = 1$

Case 2: $\begin{bmatrix} 1 \\ -2 \end{bmatrix} [1 \quad 1] = -1$



Correctly Classified the data point (1,0), since it is positive

Problem 2

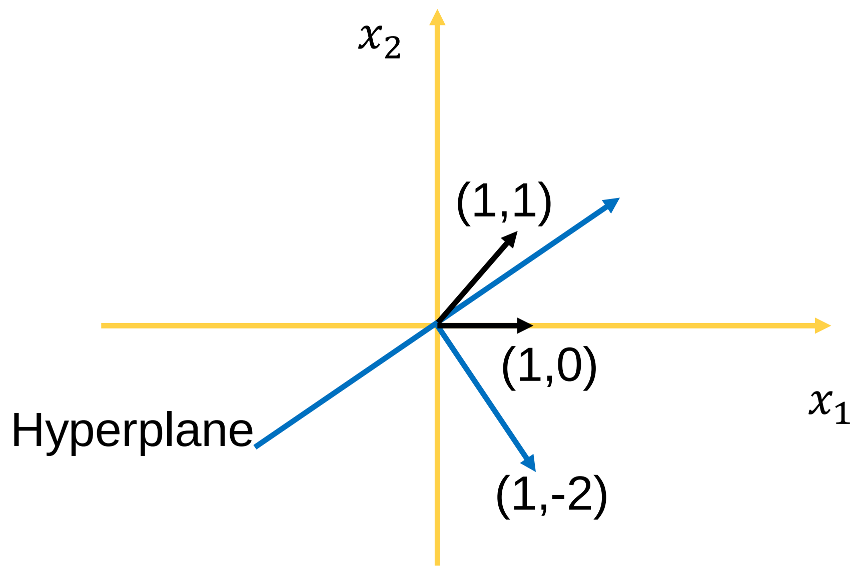
- Consider a hyperplane defined by a line $y = x_1 - 2x_2$.
- Determine the distance of the following points from the hyperplane.
  - (-1, 2)
  - (1,0)
  - (1,1)

Solutions

- Distance of a point from hyperplane is calculated using following formula:
  - $\frac{ax_1+bx_2}{\sqrt{a^2+b^2}}$ for this case it is written as $\frac{x_1-2x_2}{\sqrt{(1)^2+(-2)^2}}$
- (-1, 2)
- (1,0)
- (1,1)

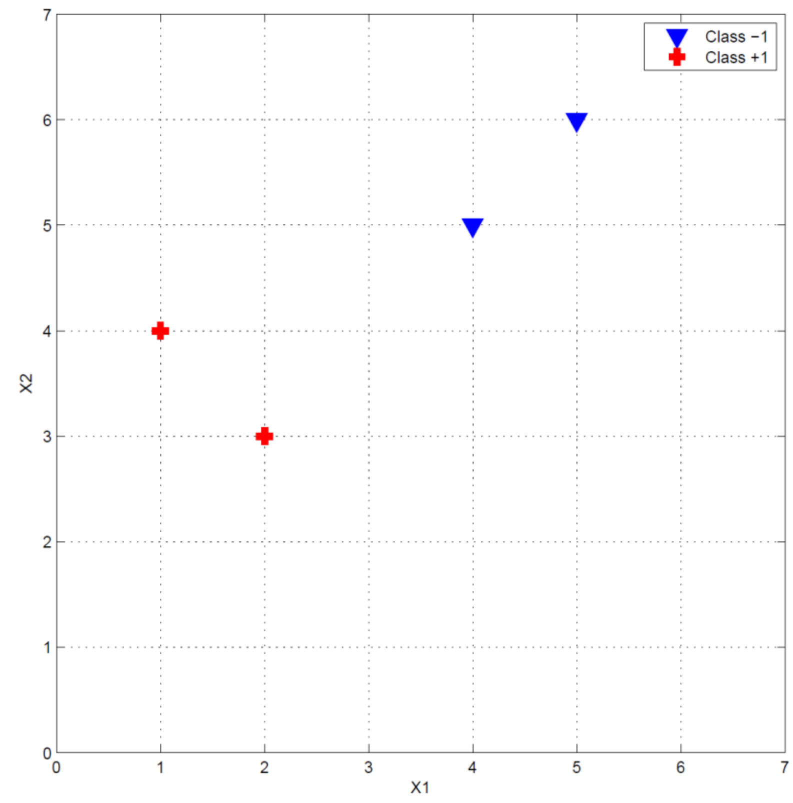
Distance of point (-1,2) is

$$\frac{1\times -1-2\times 2}{\sqrt{(1)^2+(-2)^2}}=-\frac{5}{\sqrt{5}}=-\sqrt{5}$$



Problem 3

- Support vector machines learn a decision boundary leading to the largest margin from both classes. You are training SVM on a tiny dataset with 4 points shown in Figure. This dataset consists of two examples with class label -1 (denoted with plus), and two examples with class label +1 (denoted with triangles). i. Find the weight vector w and bias b. What's the equation corresponding to the decision boundary?



Solution

- SVM tries to maximize the margin between two classes. Therefore, the optimal decision boundary is diagonal and it crosses the point (3,4).
- It is perpendicular to the line between support vectors (4,5) and (2,3), hence it is slope is m = -1.
- Thus the line equation is $x_2 - 4 = -1(x_1 - 3) = x_1 + x_2 = 7$. From this equation, we can deduce that the weight vector has to be of the form (w_1, w_2), where $w_1 = w_2$. It also has to satisfy the following equations:
  - $2w_1 + 3w_2 + b = 1$ and $4w_1 + 5w_2 + b = 1$
  - Hence, $w_1 = w_2 = -\frac{1}{2}$ and $\mathrm{b} = \frac{7}{2}$

Solution…