# Introduction to Machine Learning

# Support Vector Machine (SVM)

Inas A. Yassine
Systems and Biomedical Engineering Department,
Faculty of Engineering - Cairo University
*iyassine@eng.cu.edu.eg*

# Learning Objectives

- Support Vector Machine (SVM)
  - Introduction
  - Properties of SVM
  - SVM Applications
- Artificial Neural Network (ANN)
  - Key Concepts
  - Perceptron Learning
  - Learning by Error Minimization

# A Way to Choose a Model Class

- We want to get a low error rate on unseen data.
  - This is called "structural risk minimization"
- It would be really helpful if we could get a guarantee of the following form:

  Test error rate =< train error rate + f(N, h, p)

  where N = size of training set,

  h = measure of the model complexity,

  p = the probability that this bound fails

  We need p to allow for really unlucky test sets.
- Then we could choose the model complexity that minimizes the bound on the test error rate.

# 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

# Why Support Vector Machine (SVM)?

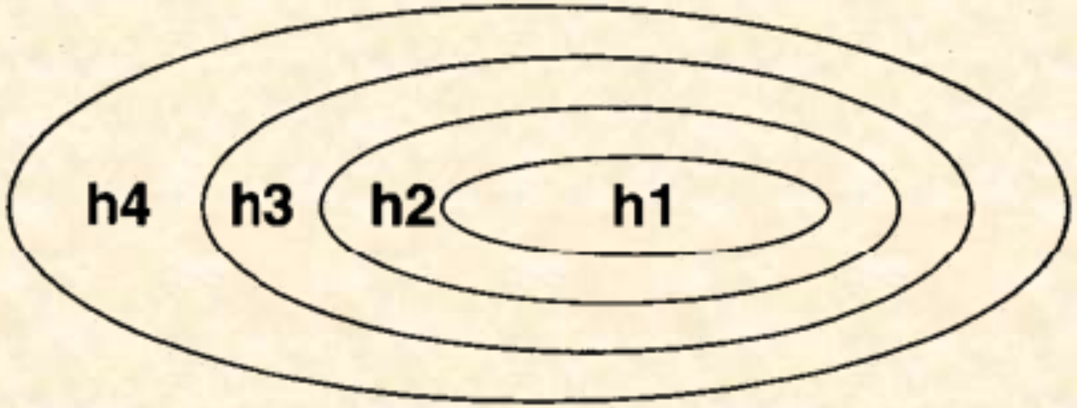
- Use a very big set of non-linear features that is task-independent.
- Have a clever way to:

  - prevent overfitting

  - Use a huge number of features without requiring nearly as much computation as seems to be necessary

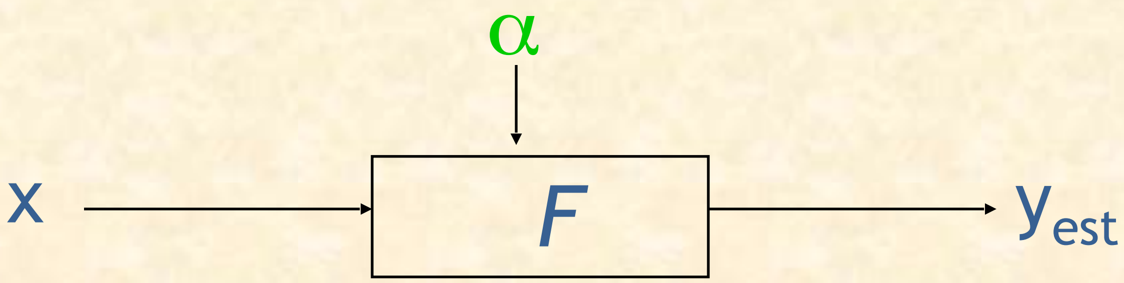
# A Hierarchy of Model Classes

- Some model classes can be arranged in a hierarchy of increasing complexity.
- How do we pick the best level in the hierarchy for modeling a given dataset?



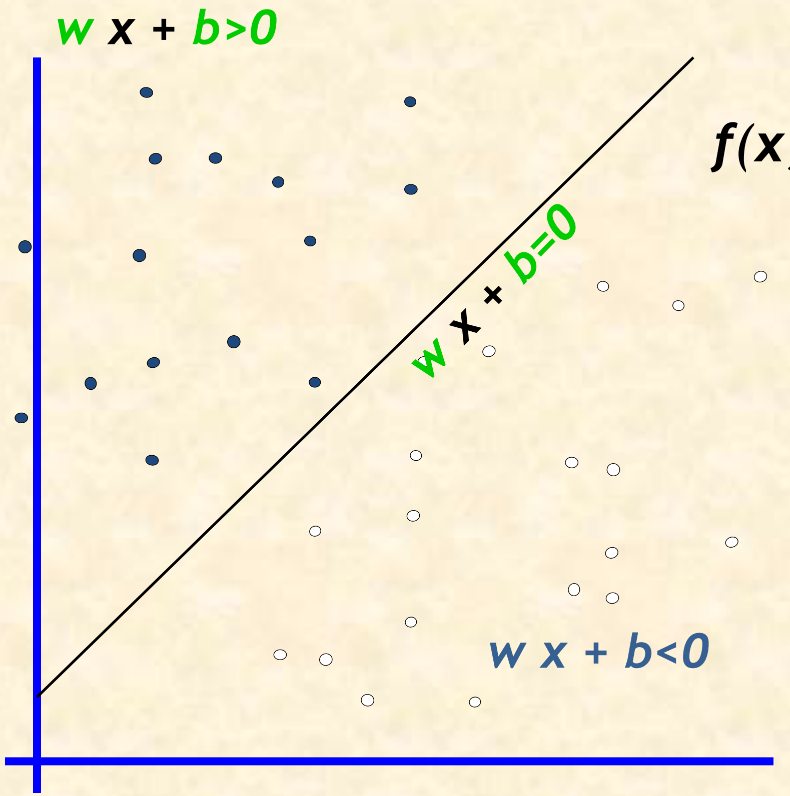

h1 < h2 < h3 ...

# Linear Classifiers




•denotes +1

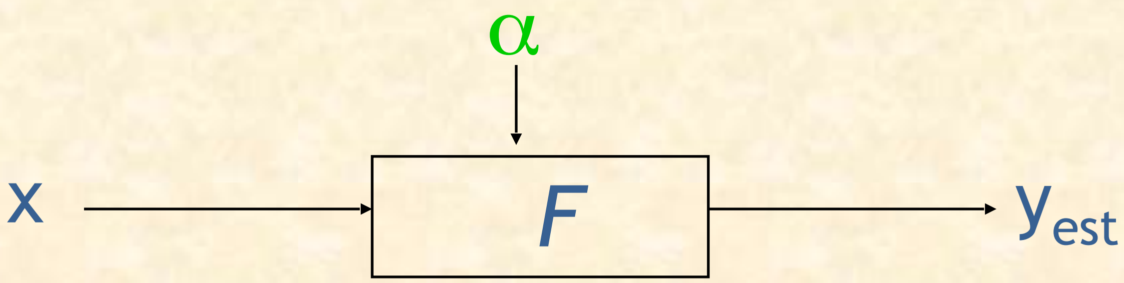
◦denotes -1



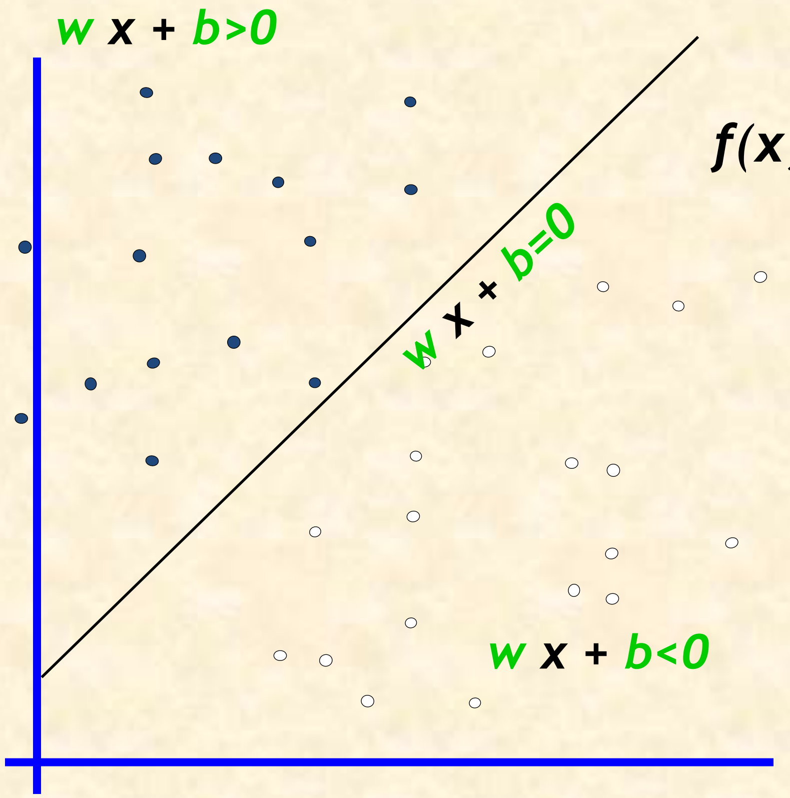

$f(\mathbf{x},w,b) = sign(w\ \mathbf{x} + b)$

How would you classify this data?

# Linear Classifiers



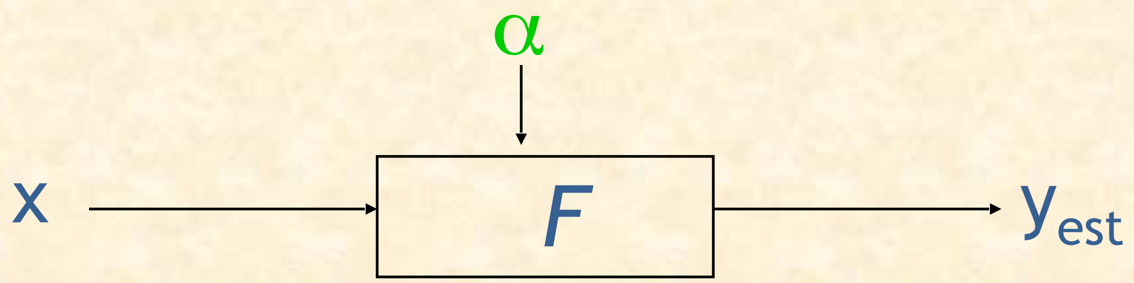

- •denotes +1
- ◦denotes -1




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

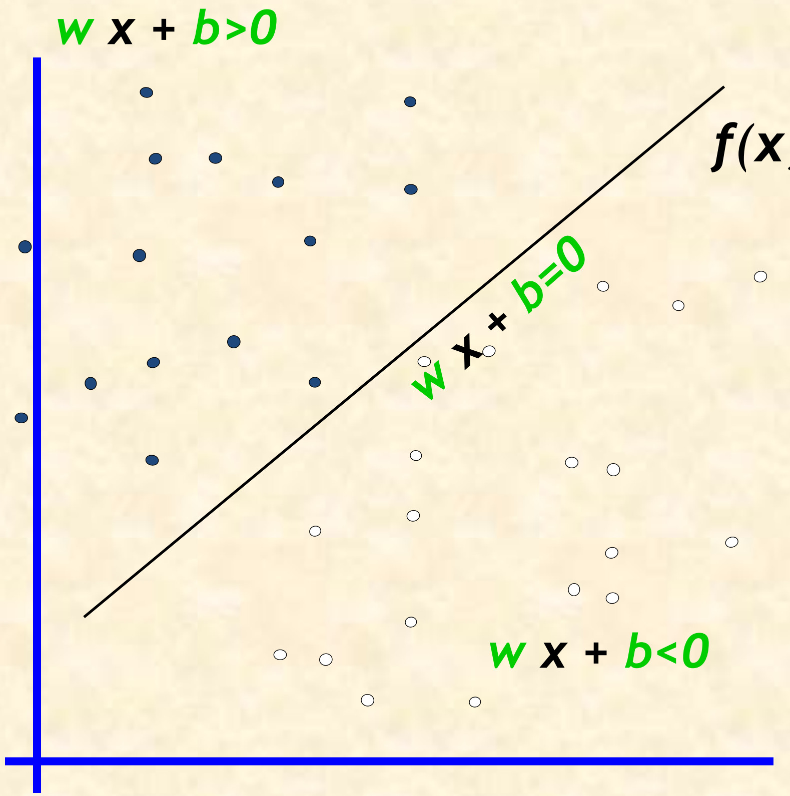
How would you classify this data?

# Linear Classifiers




•denotes +1

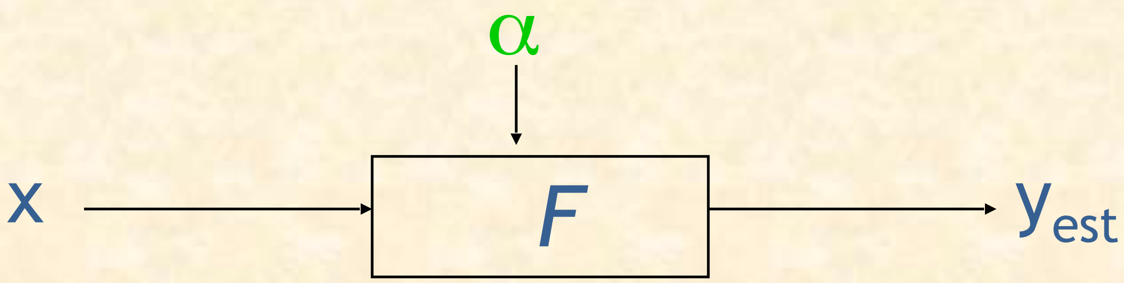
◦denotes -1



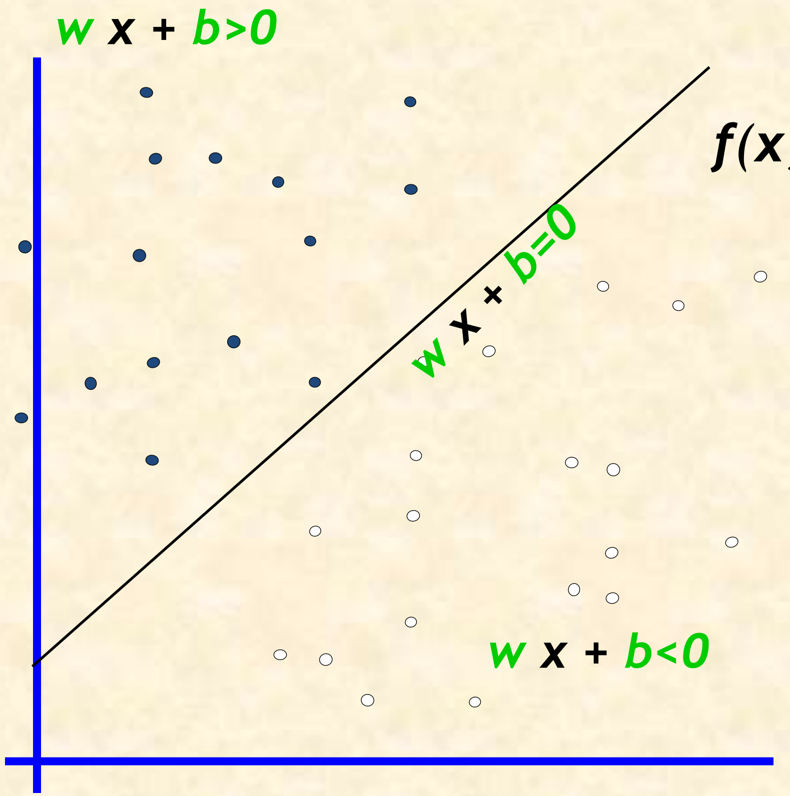

$f(\mathbf{x},w,b) = sign(w\ \mathbf{x} + b)$

How would you classify this data?

# Linear Classifiers



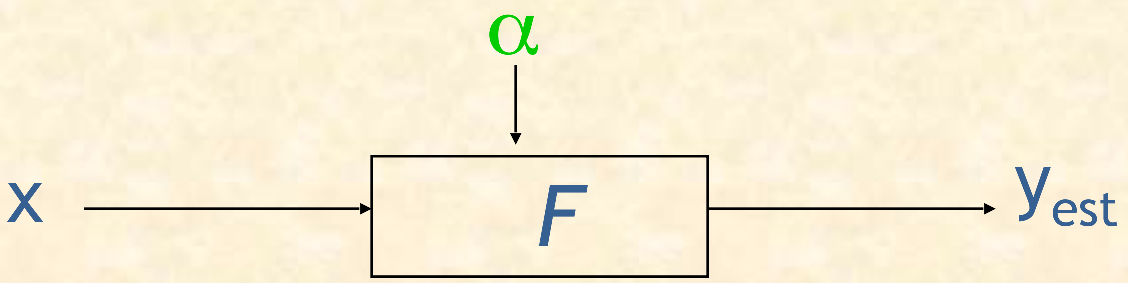

- denotes +1
- ◦ denotes -1




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

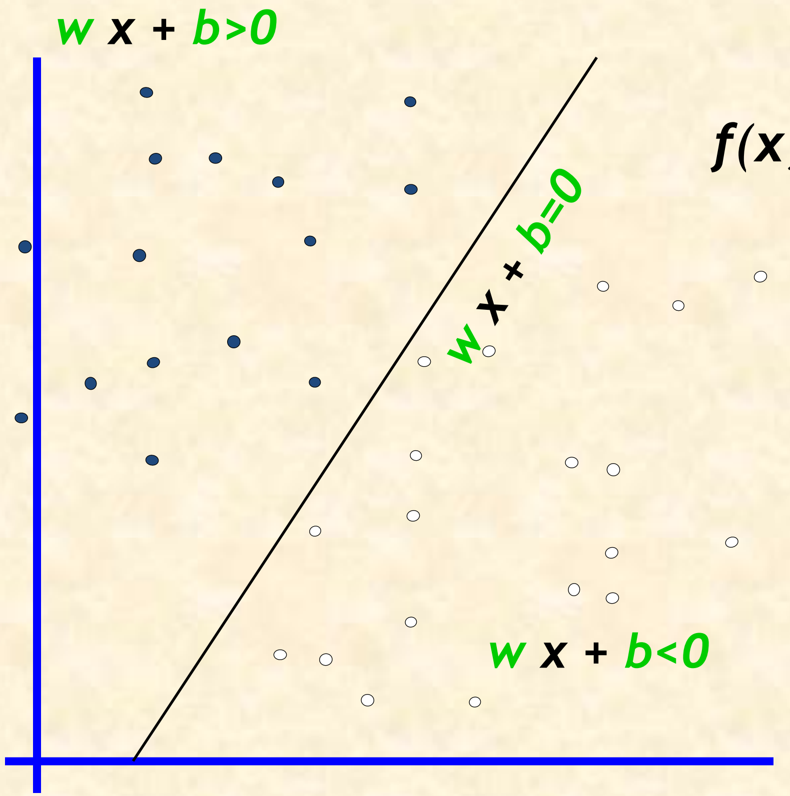
How would you classify this data?

# Linear Classifiers




• denotes +1

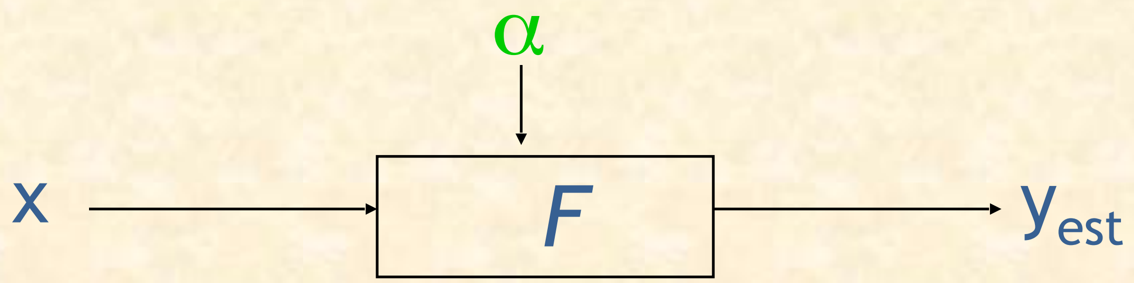
○ denotes -1



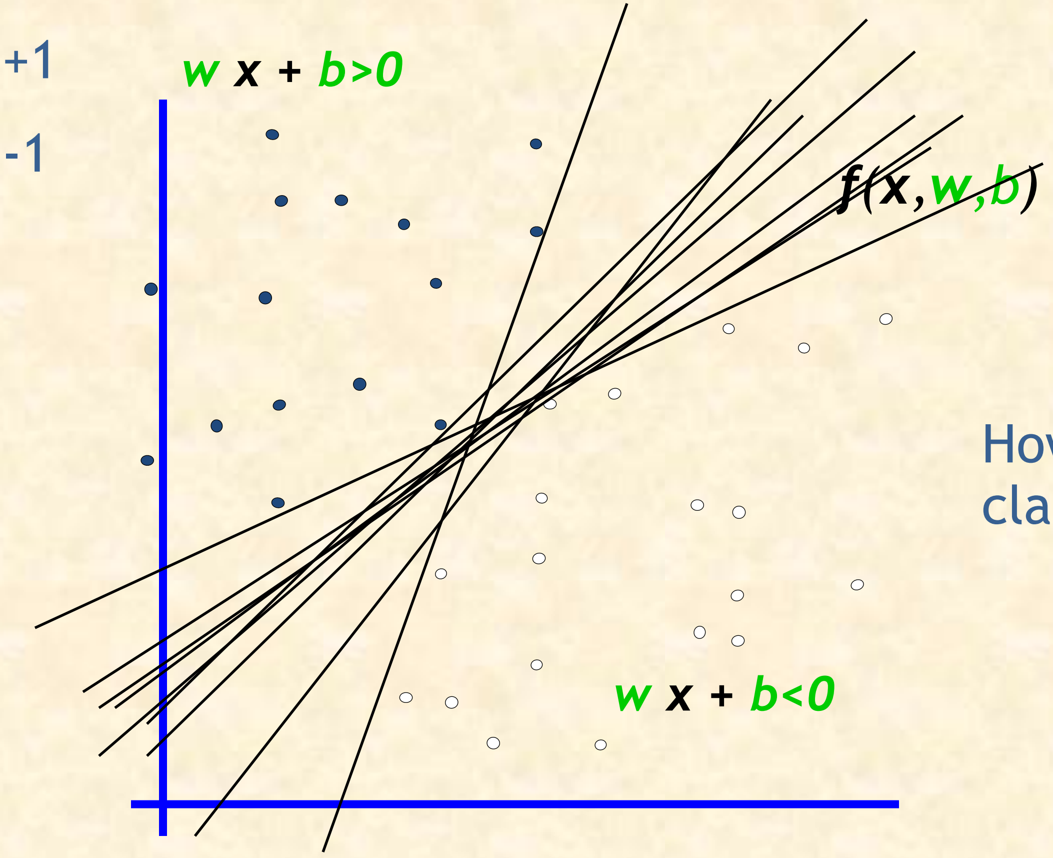

$$f(\mathbf{x},w,b) = sign(w\ \mathbf{x} + b)$$

How would you classify this data?

# Linear Classifiers



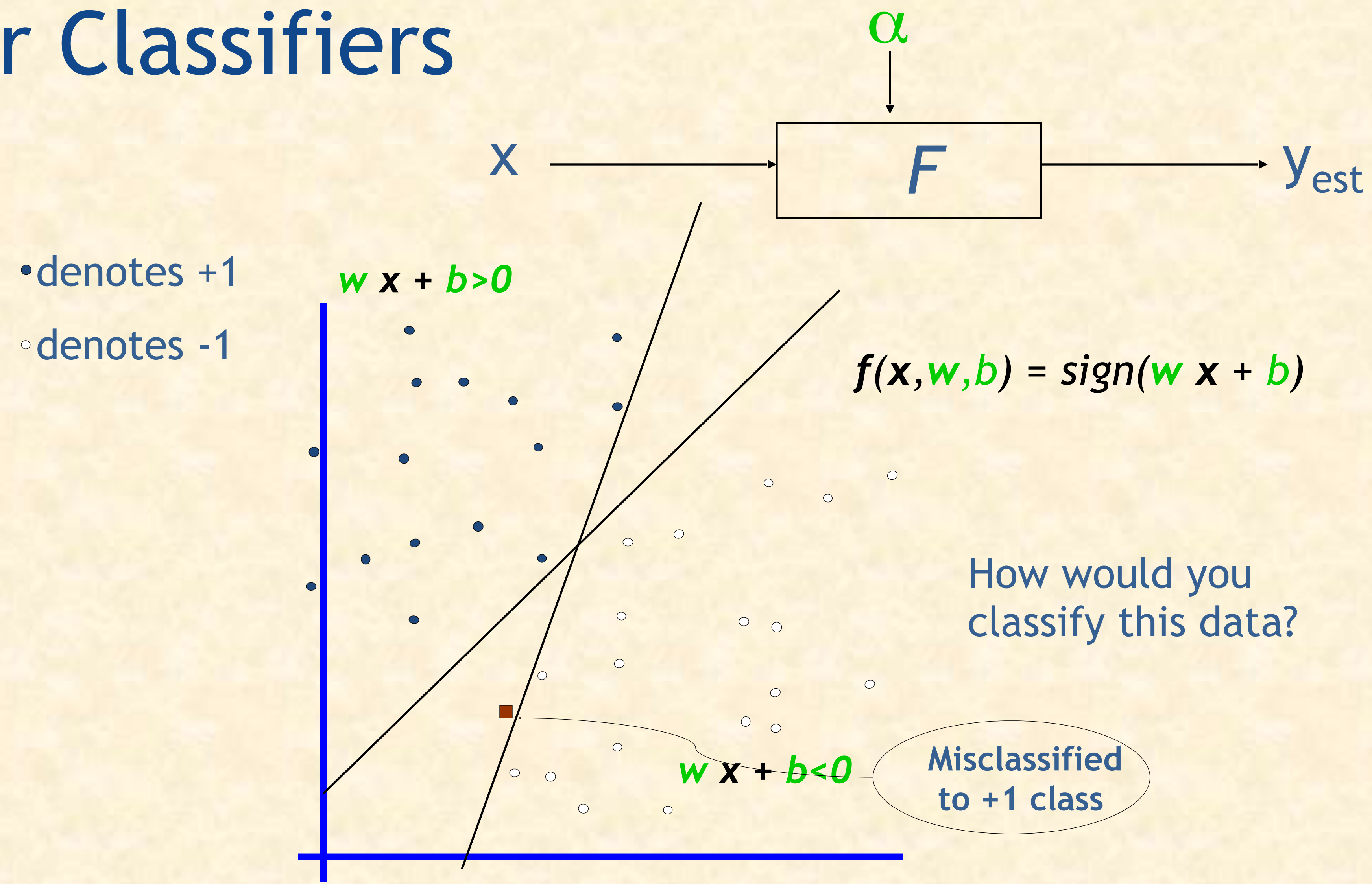

- •denotes +1
- ◦denotes -1




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

How would you classify this data?

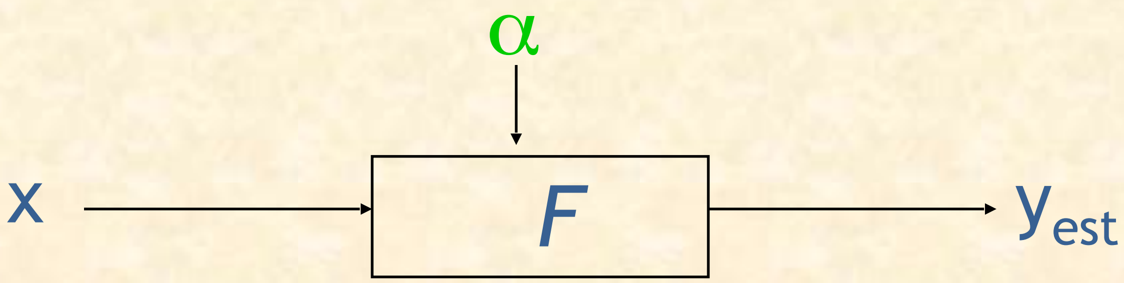
# Linear Classifiers




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

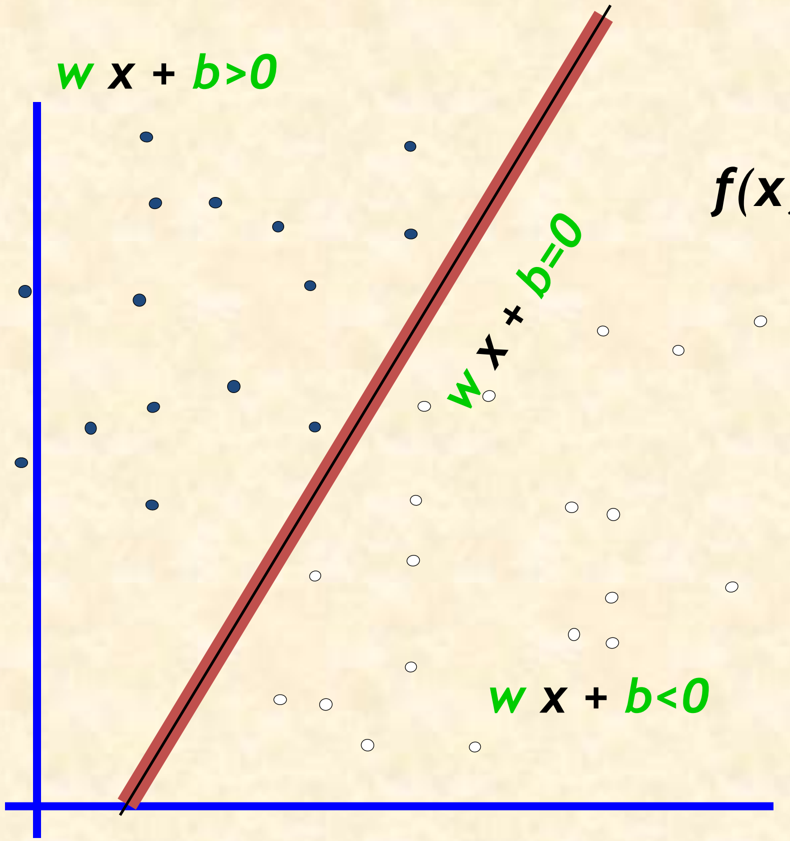
How would you classify this data?

# Linear Classifiers




•denotes +1

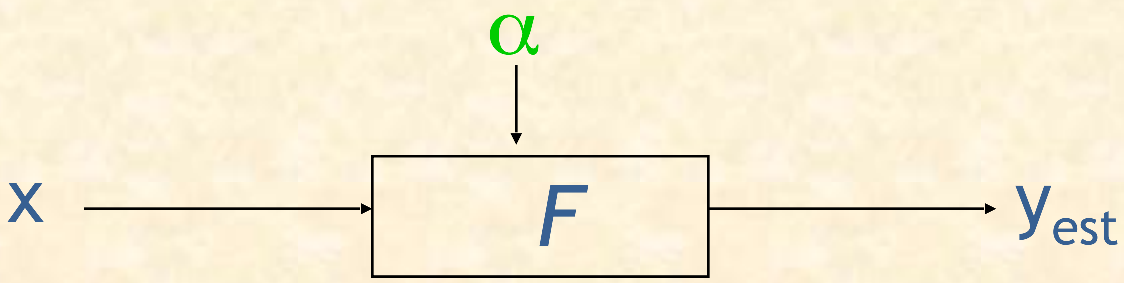
◦denotes -1




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

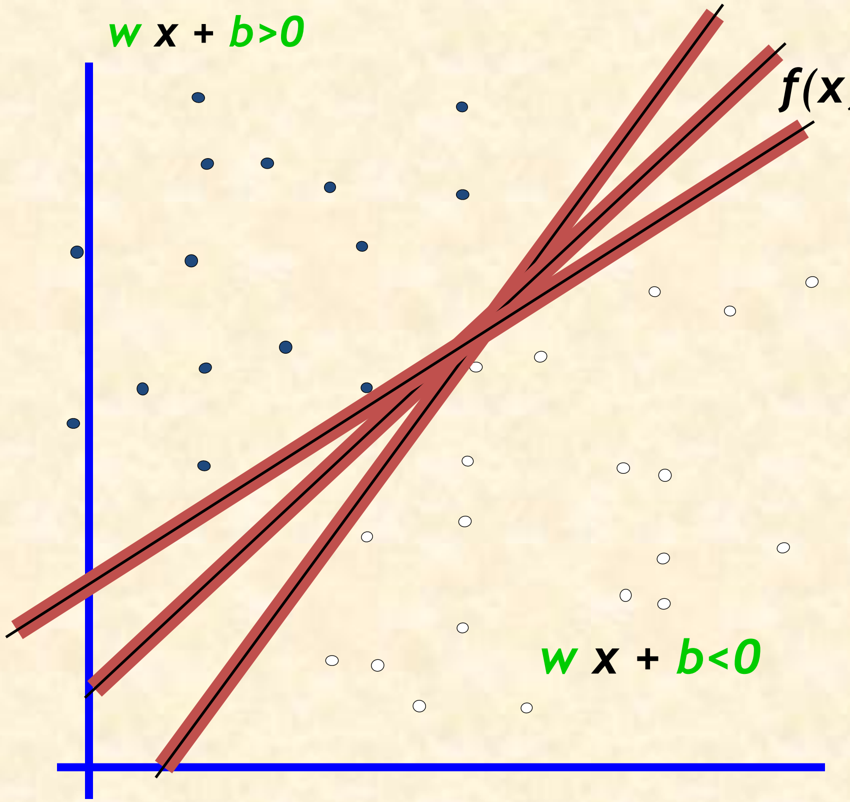
How would you classify this data?

# Linear Classifiers




• denotes +1

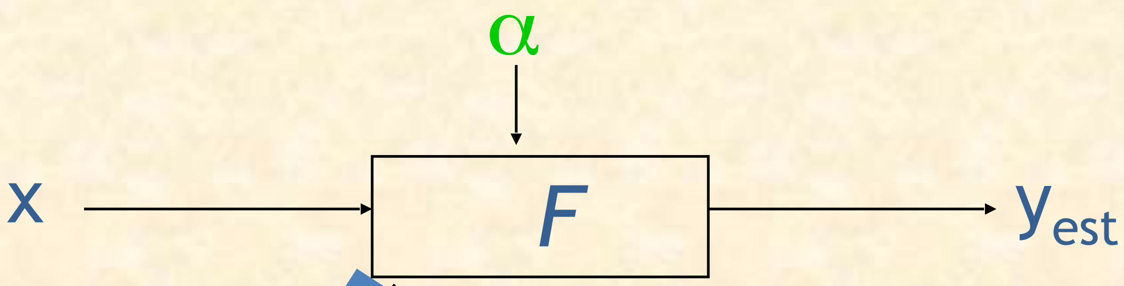
◦ denotes -1

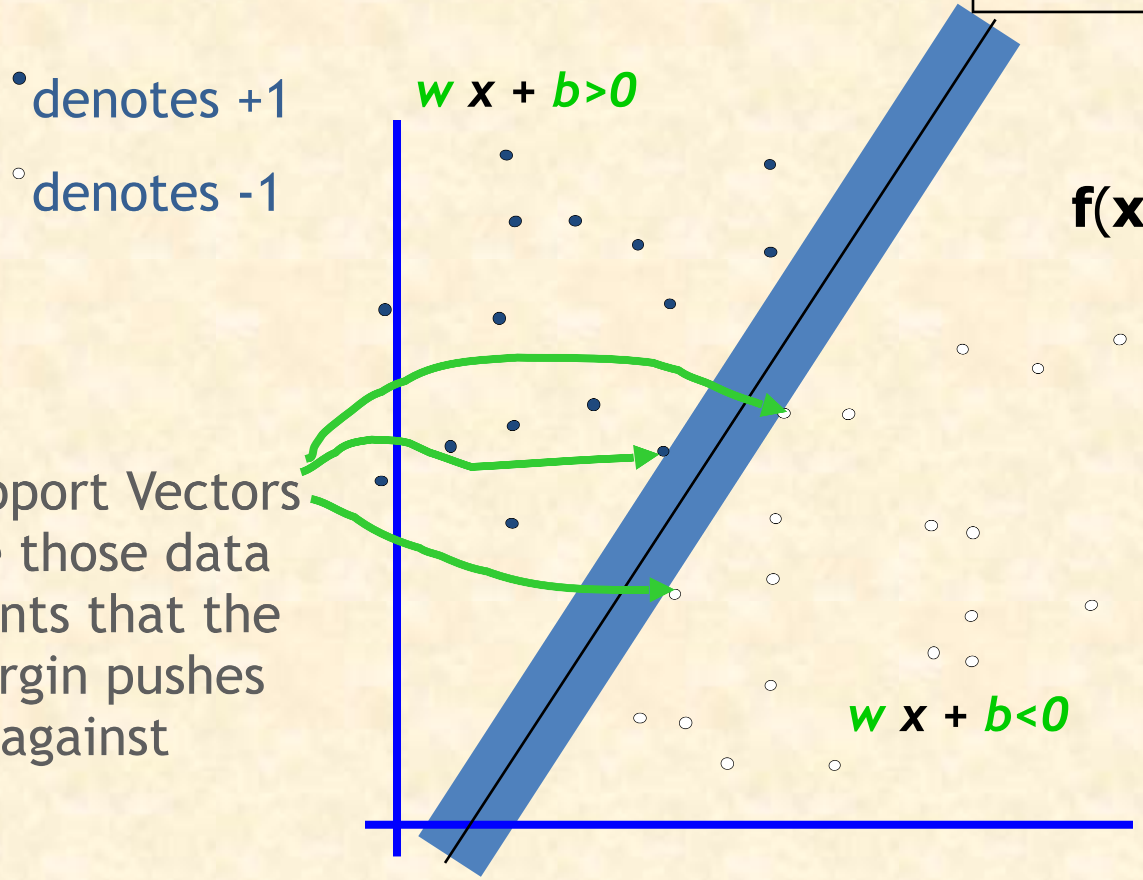



$f(\mathbf{x}, w, b) = sign(w\ \mathbf{x} + b)$

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

# Maximum Margin




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

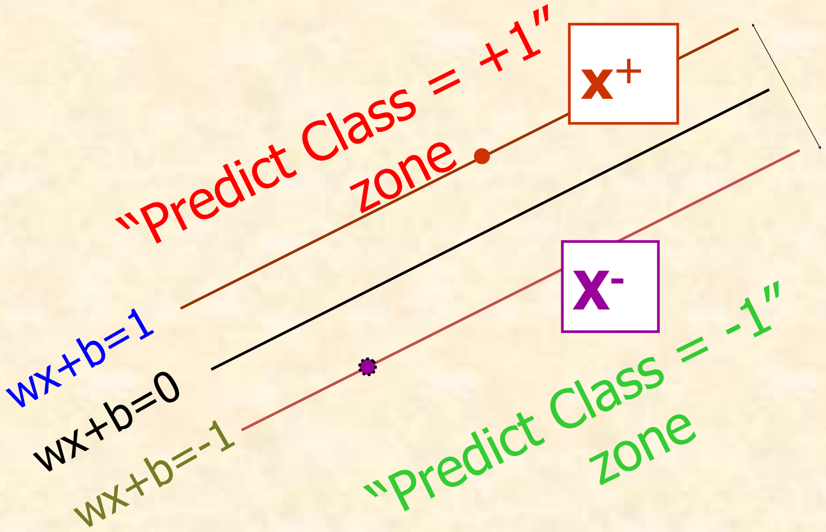
Support Vectors are those data points that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the maximum margin; called an LSVM.

# Linear SVM Mathematically

What we know:

- $w \cdot x^{+} + b = +1$
- $w \cdot x^{-} + b = -1$
- $w \cdot (x^{+} - x^{-}) = 2$




*M*=Margin Width

$$M = \frac{(x^{+} - x^{-}) \cdot w}{|w|} = \frac{2}{|w|}$$

# Linear SVM Mathematically

- Goal: 1) Correctly classify all training data

$$wx_i + b \geq 1 \qquad \text{if } y_i = +1$$

$$wx_i + b \leq 1 \qquad \text{if } y_i = -1$$

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

$$M = \frac{2}{|w|}$$

2) Maximize the Margin

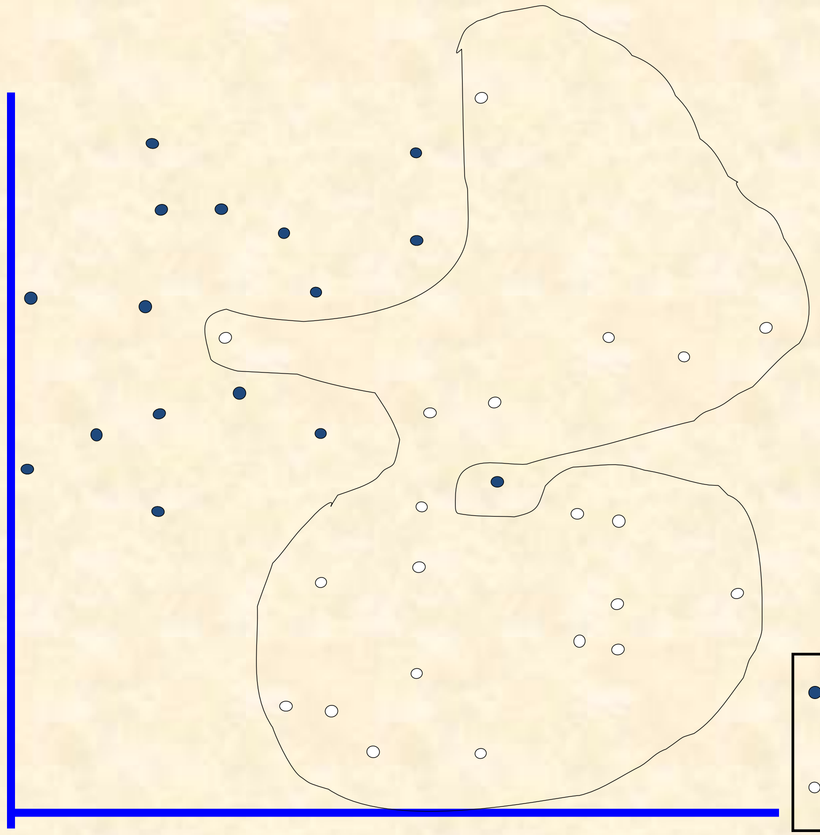
same as minimize $\Phi(w) = \frac{1}{2} w^t w$

- Quadratic Optimization Problem and solve for w and b

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

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

# Dataset With Noise



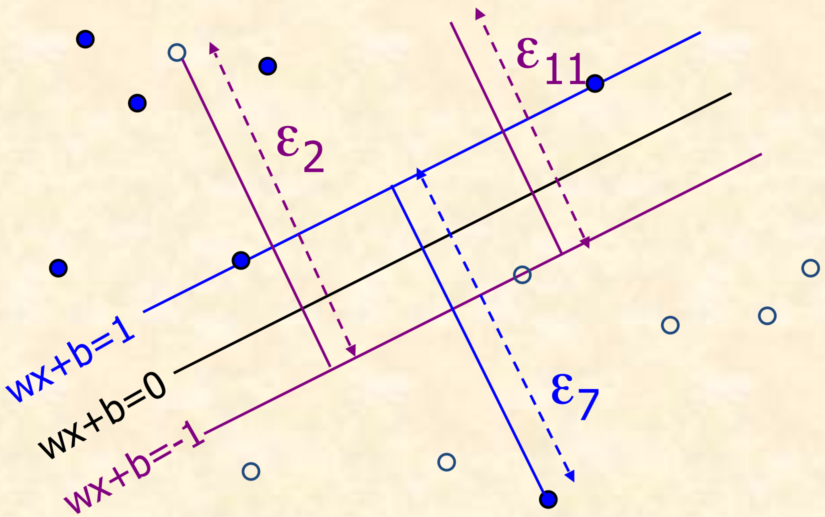
- Hard Margin: So far we require all data points be classified correctly
  - No training error
- What if the training set is noisy?

• denotes +1
◦ denotes -1

**OVERFITTING!**

# Soft Margin Classification

- Slack variables $\xi_i$ can be added to allow misclassification of difficult or noisy examples.




optimization criterion:

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 w and $b$ such that

$\Phi(\mathbf{w}) = \frac{1}{2}\, \mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x}_i, y_i)\}$

$y_i\, (\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- The new formulation incorporating slack variables:

Find w and $b$ such that

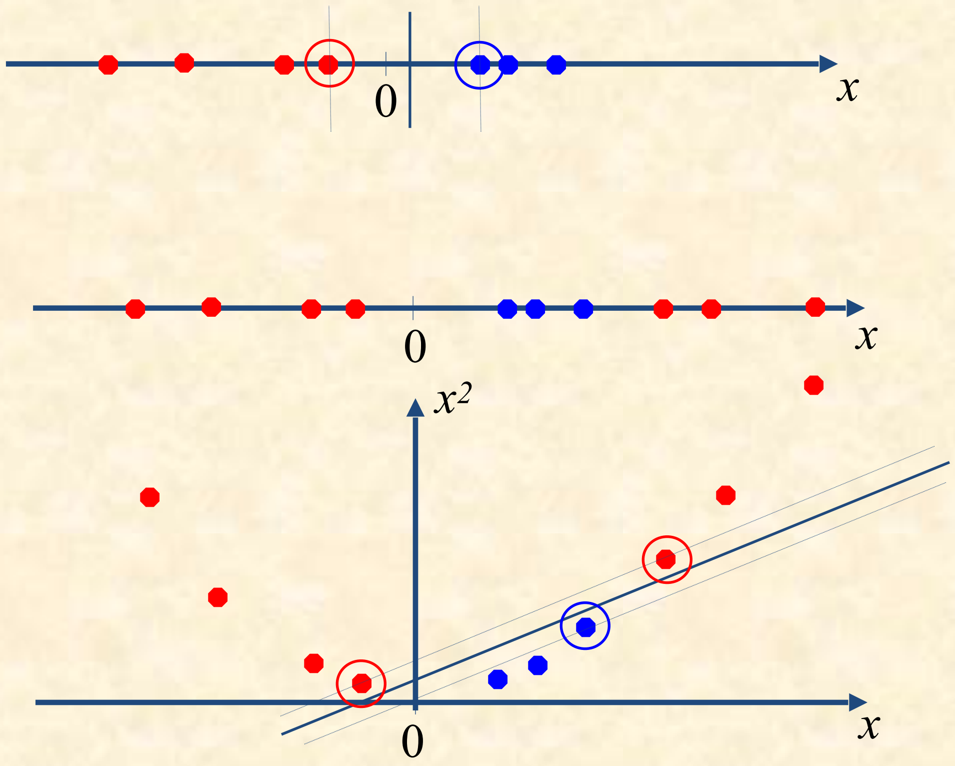
$\Phi(\mathbf{w}) = \frac{1}{2}\, \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized and for all $\{(\mathbf{x}_i, y_i)\}$

$y_i\, (\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all $i$

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

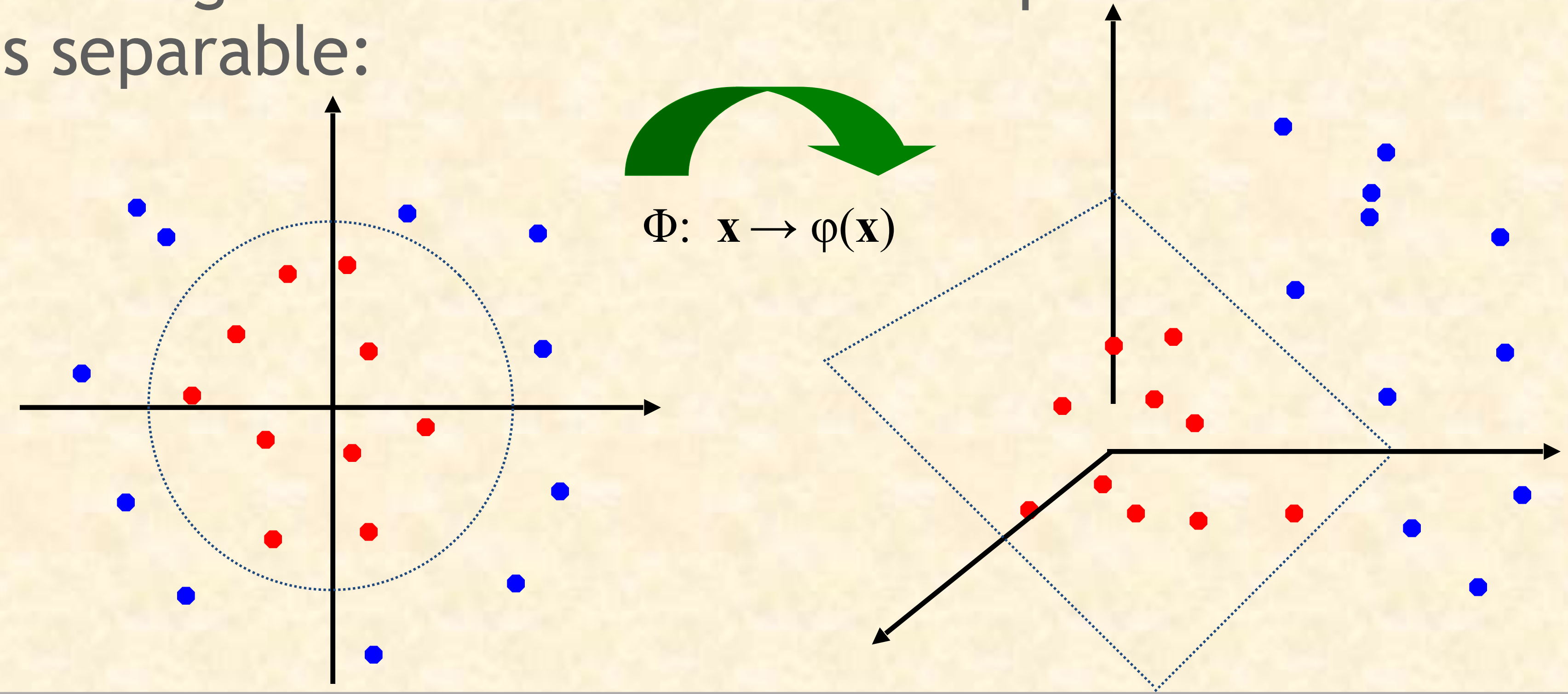
# Non-linear SVMs

- 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:



# Examples of Kernel Functions

The kernel function plays the role of the dot product in the feature space.

- 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): $K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$
- Sigmoid: $K(\mathbf{x}_i,\mathbf{x}_j)$: $\tanh(\beta_0\mathbf{x}_i^T\mathbf{x}_j + \beta_1)$

# SVM parameters choice

- 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

# 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

# Weakness of SVM

- It is sensitive to noise

  - A relatively small number of mislabeled examples can dramatically decrease the performance

- It only considers two classes

  - how to do multi-class classification with SVM?

  - Answer:

  1) with output similarity m, learn m SVM's

  - SVM 1 learns "Output==1" vs "Output != 1"
  - SVM 2 learns "Output==2" vs "Output != 2"
  - :
  - SVM m learns "Output==m" vs "Output != m"

  2)To predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

# Thank You ...