Processamento e Análise de Imagens

Support Vector Machines

Prof. Alexei Machado
PUC Minas

Support Vector Machines

- A classifier derived from statistical learning theory by Vapnik, et al. in 1992
- SVM became famous when, using images as input, it gave accuracy comparable to neural-network with hand-designed features in a handwriting recognition task
- Currently, SVM is widely used in object detection & recognition, content-based image retrieval, text recognition, biometrics, speech recognition, etc.
- Also used for regression

Support Vector Machines

- SVMs pick best separating hyperplane according to some criterion
  - e.g. maximum margin
- Training process is an optimisation
- Training set is effectively reduced to a relatively small number of support vectors

Discriminant Function

A classifier is said to assign a feature vector x to class w_i if

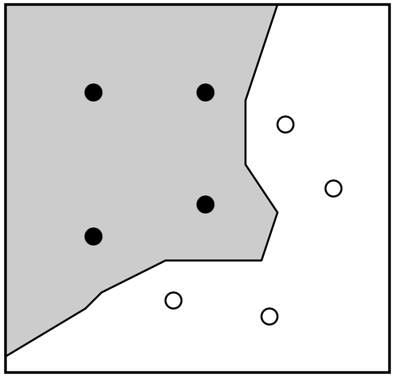
$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \qquad \text{for all} \quad j \neq i$$

- For two-category case, $g(\mathbf{x}) \equiv g_1(\mathbf{x}) - g_2(\mathbf{x})$

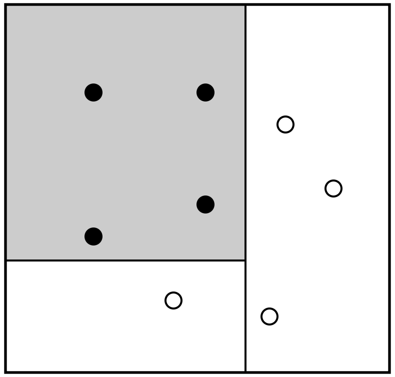
$$\text{Decide } \omega_1 \text{ if } g(\mathbf{x}) > 0; \text{ otherwise decide } \omega_2$$

Discriminant Function

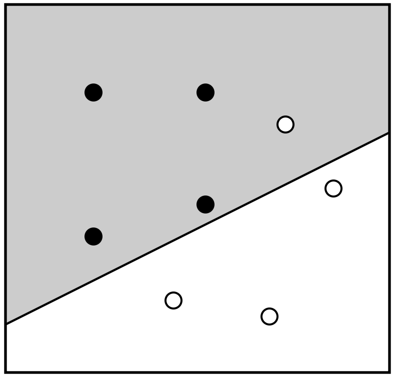
It can be arbitrary functions of *x*, such as:



Nearest Neighbor

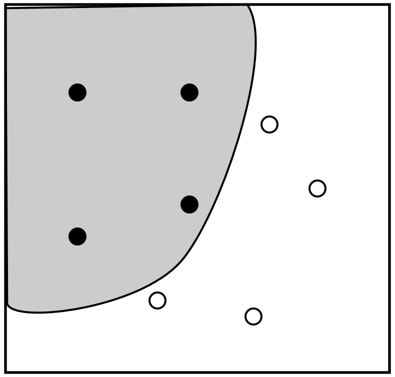


Decision Tree



Linear Functions

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$



Nonlinear Functions

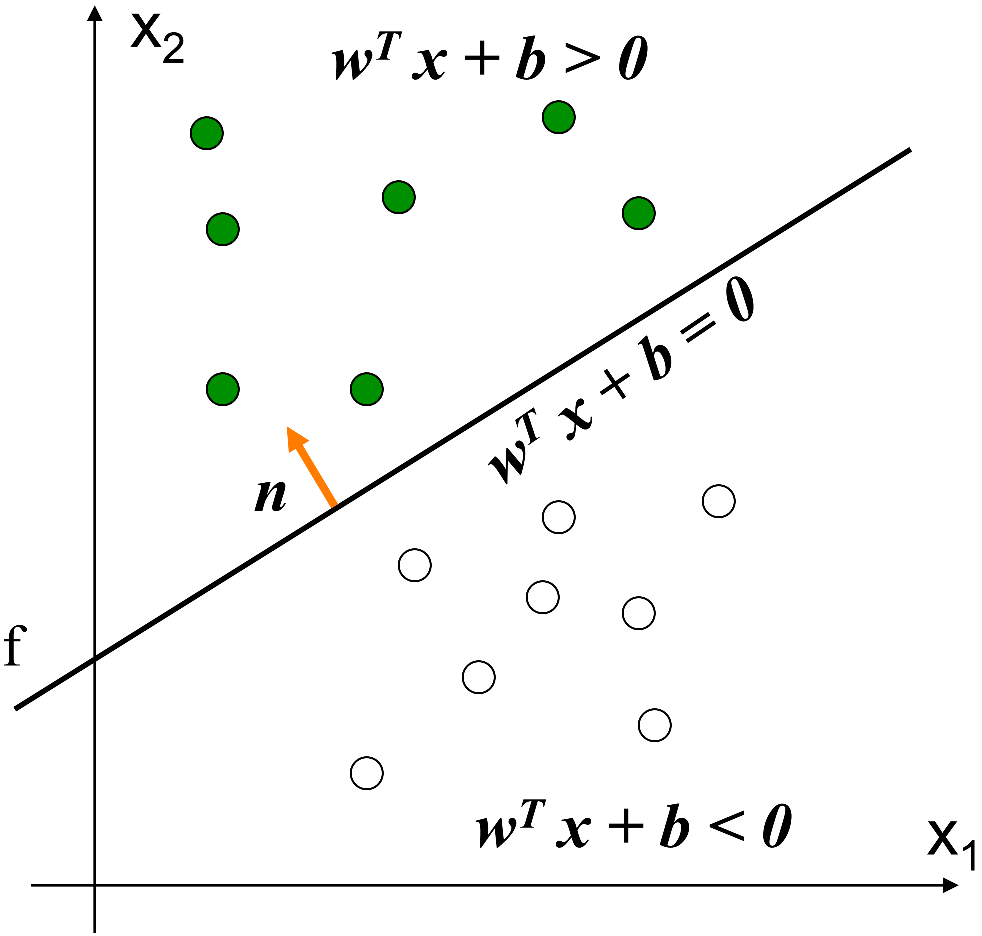
Linear Discriminant Function

g(x) is a linear function:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

- A hyper-plane in the feature space
- (Unit-length) normal vector of the hyper-plane:

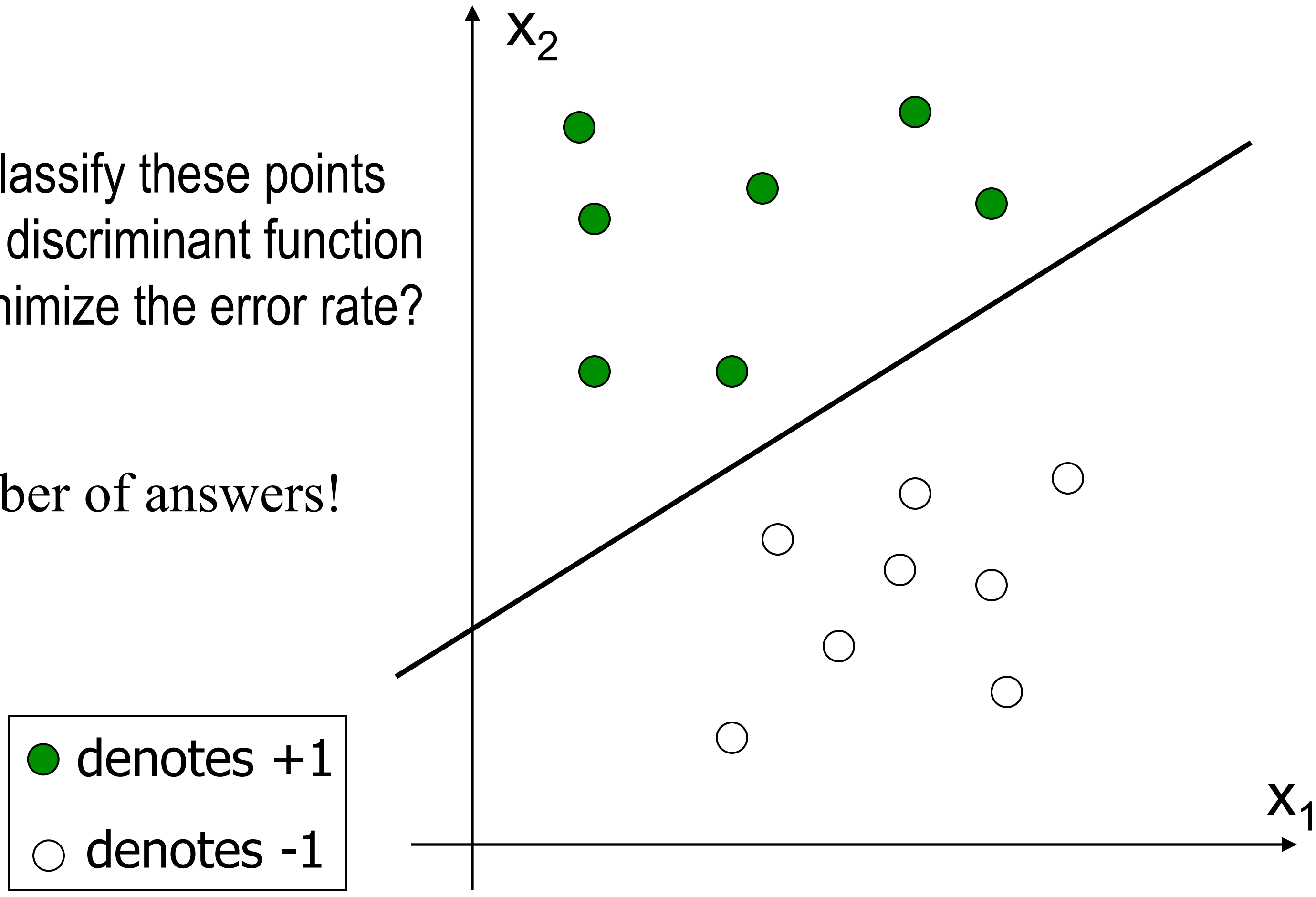
$$\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$



Linear Discriminant Function

How would you classify these points using a linear discriminant function in order to minimize the error rate?

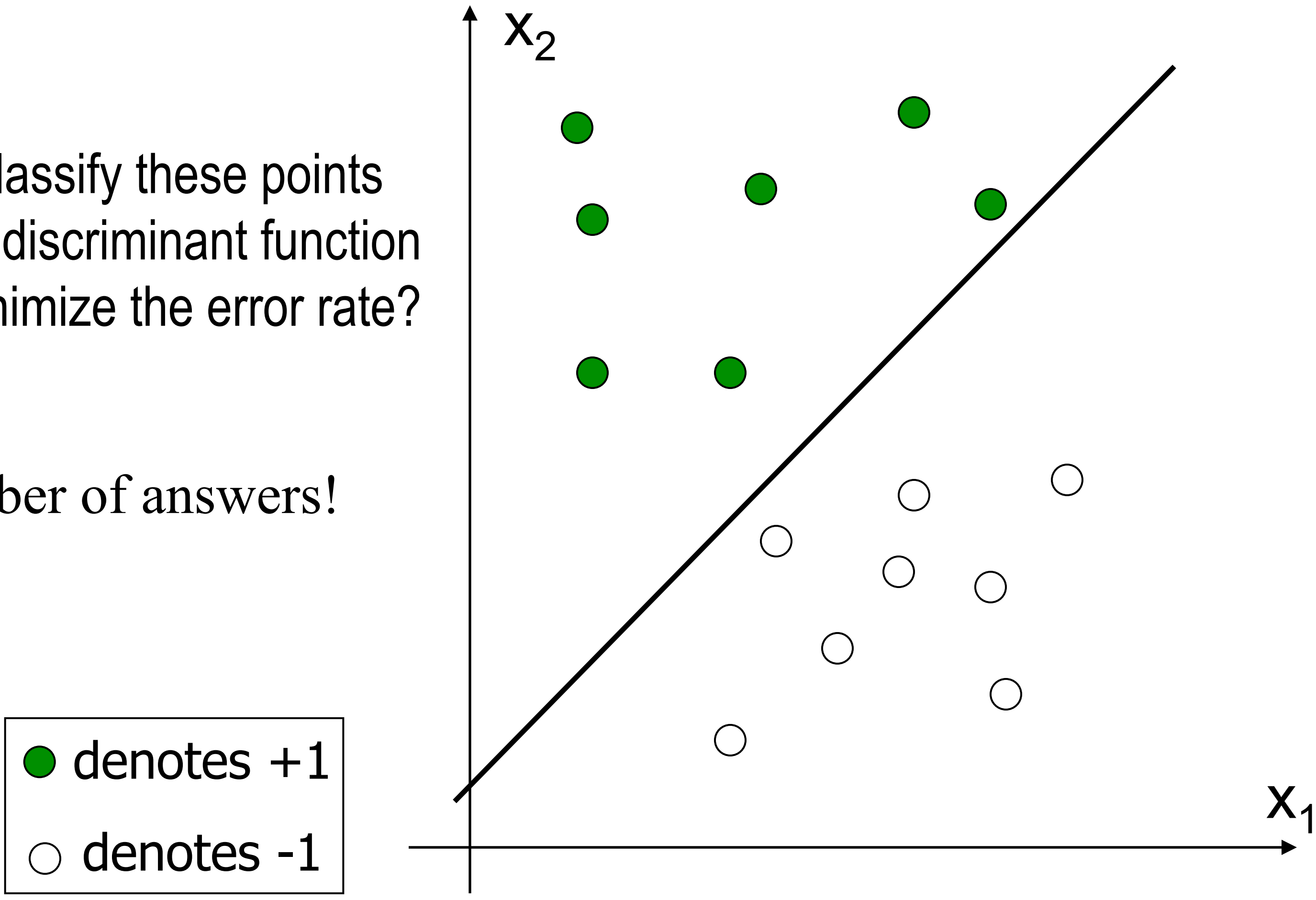
- Infinite number of answers!



Linear Discriminant Function

How would you classify these points using a linear discriminant function in order to minimize the error rate?

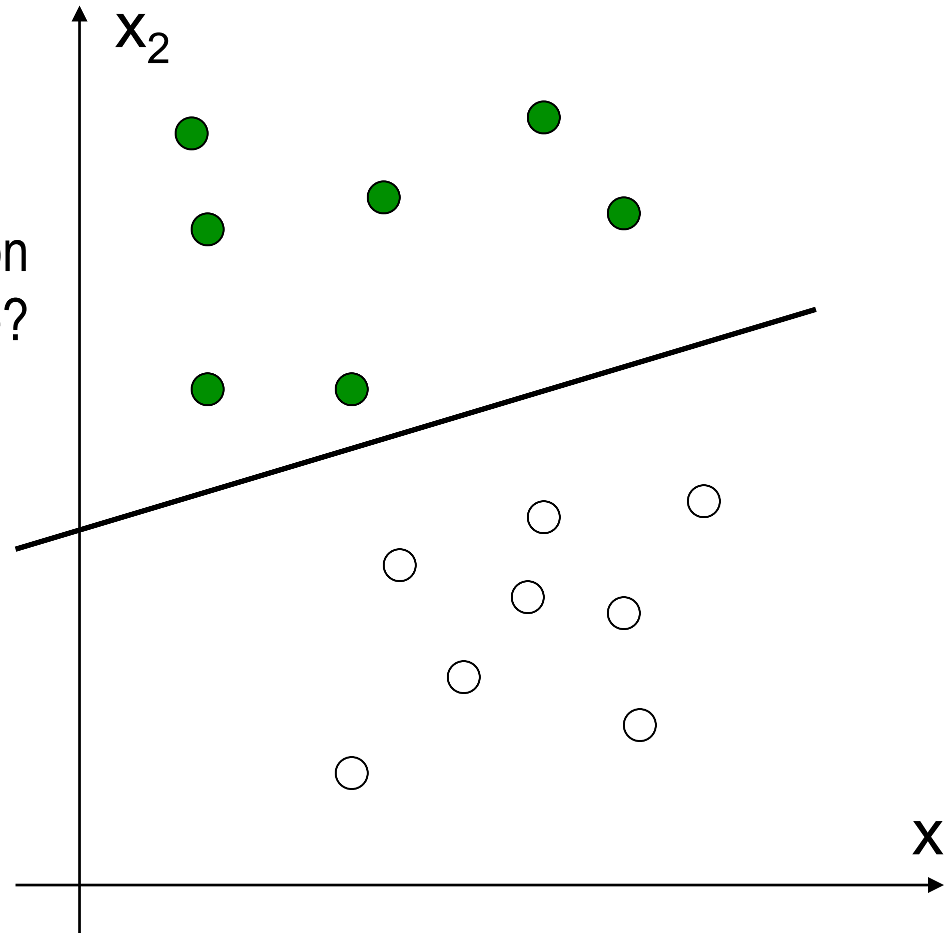
- Infinite number of answers!

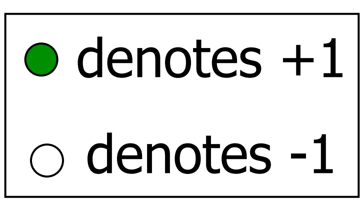


Linear Discriminant Function

How would you classify these points using a linear discriminant function in order to minimize the error rate?

- Infinite number of answers!

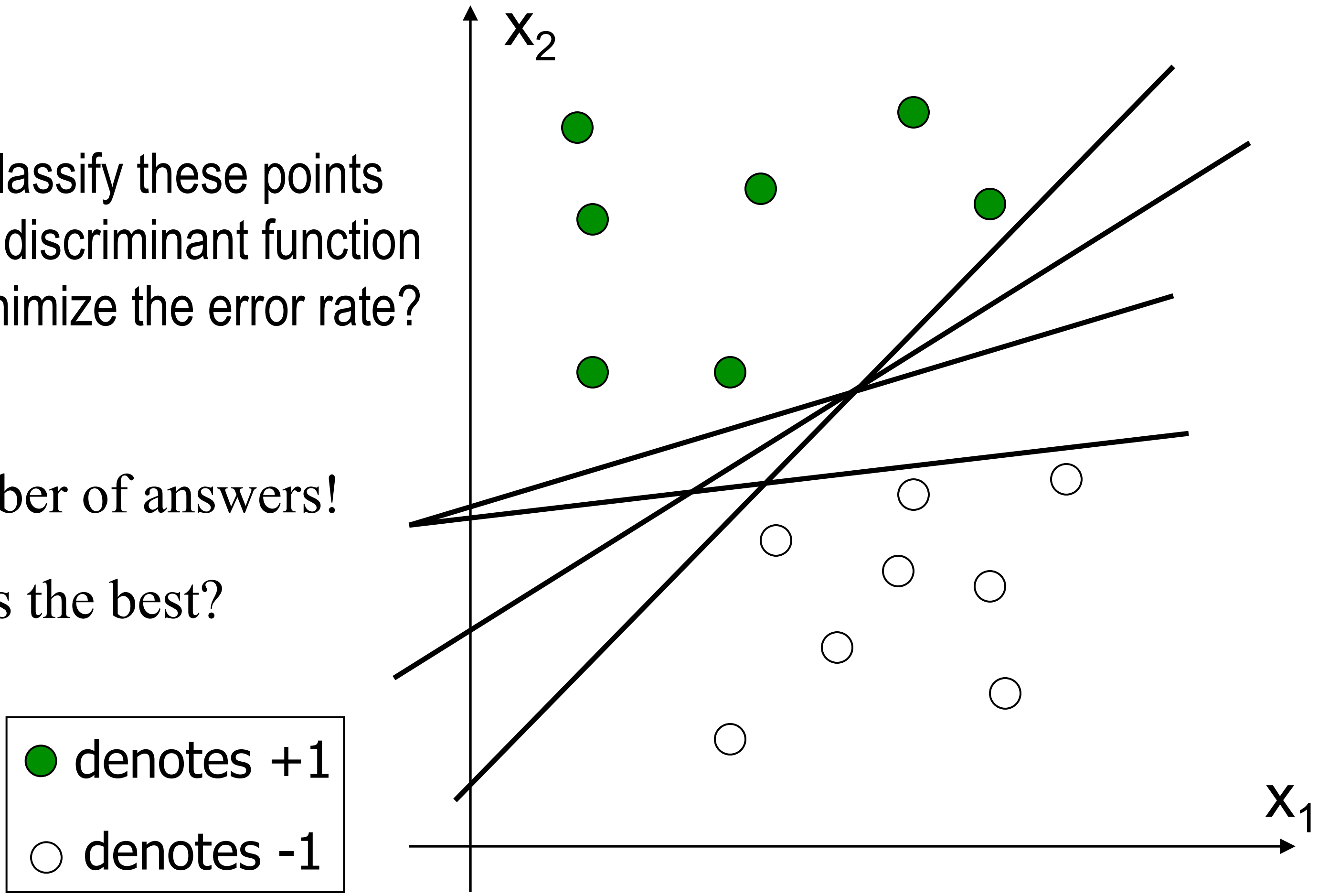




Linear Discriminant Function

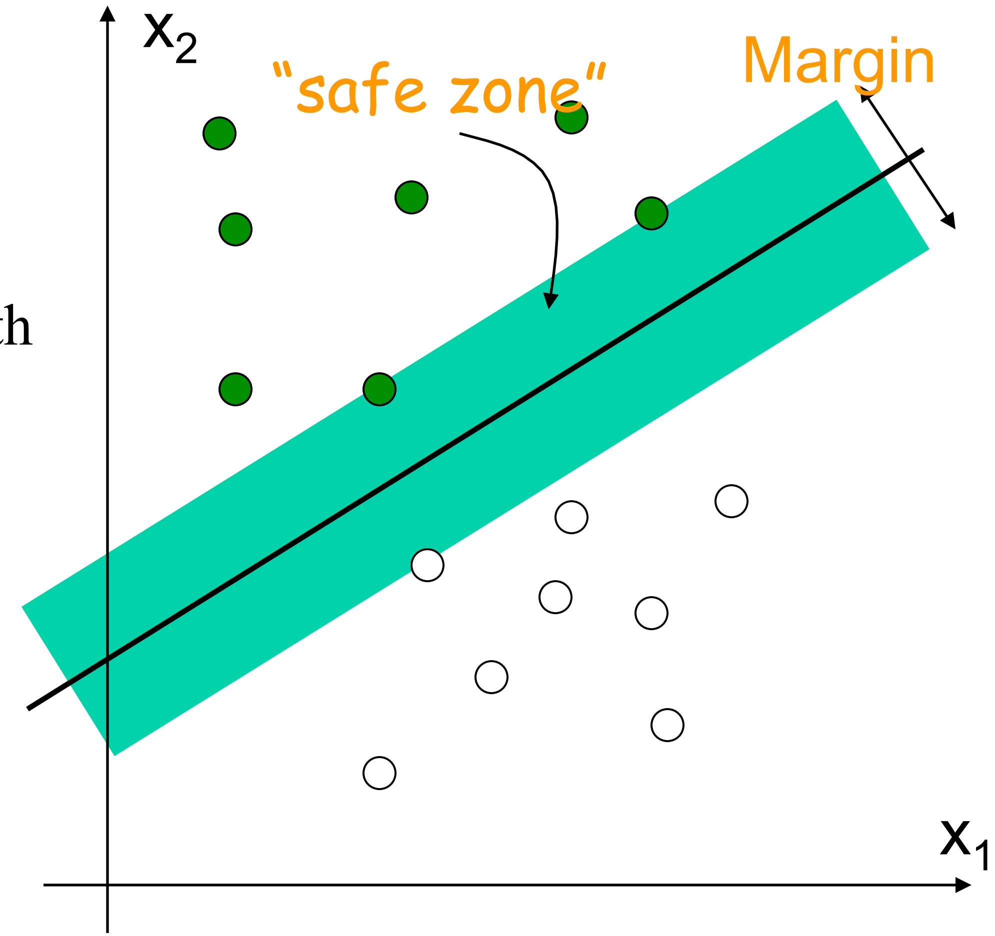
How would you classify these points using a linear discriminant function in order to minimize the error rate?

- Infinite number of answers!
- Which one is the best?



Large Margin Linear Classifier

The linear discriminant function (classifier) with the maximum margin is the best



- Margin is defined as the width that the boundary could be increased by before hitting a data point

- Why it is the best?
  - Robust to outliners and thus strong generalization ability

Large Margin Linear Classifier

Given a set of data points:

$\{(\mathbf{x}_i, y_i)\},\ i = 1, 2, \cdots, n,$ where

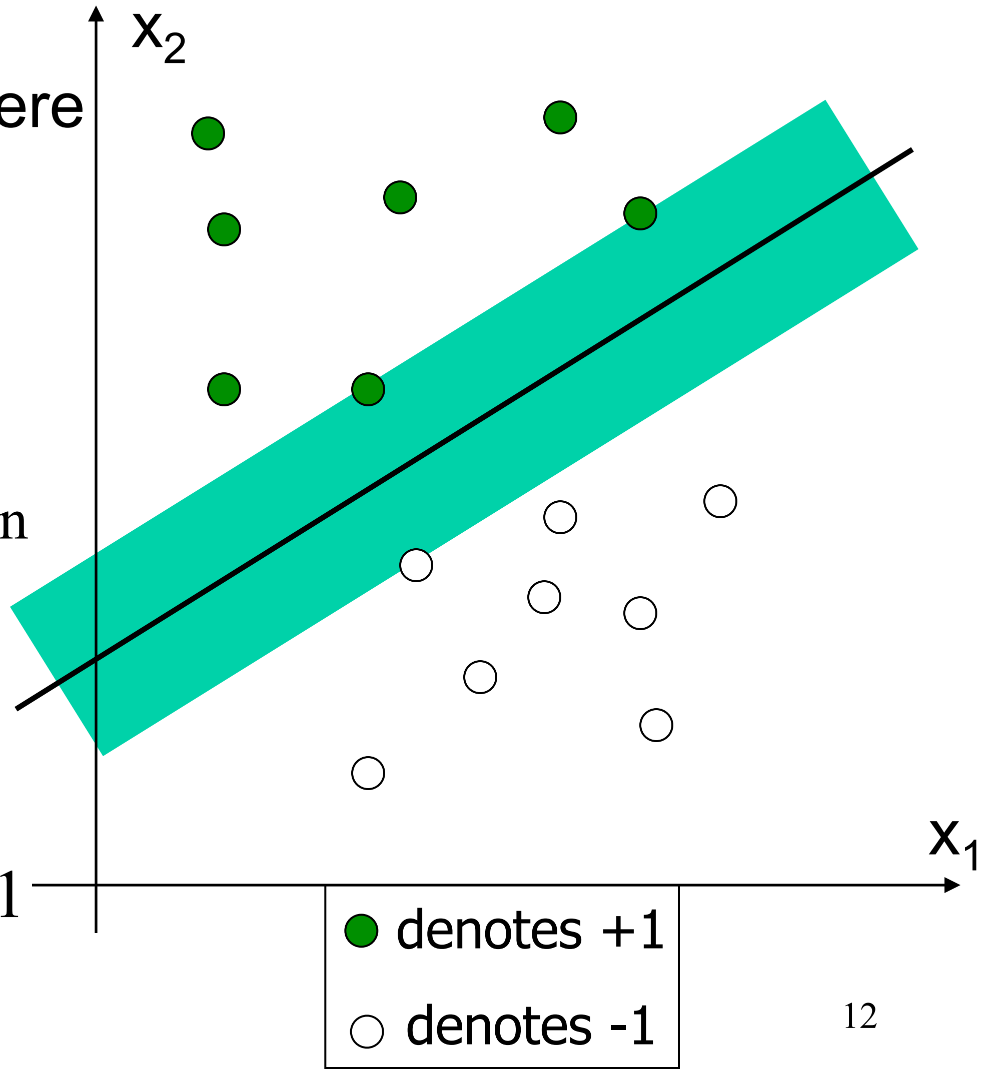
$$\text{For } y_i = +1, \quad \mathbf{w}^T \mathbf{x}_i + b > 0$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T \mathbf{x}_i + b < 0$$

- With a scale transformation on both w and b, the above is equivalent to

$$\text{For } y_i = +1, \quad \mathbf{w}^T \mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1$$



Large Margin Linear Classifier

We know that

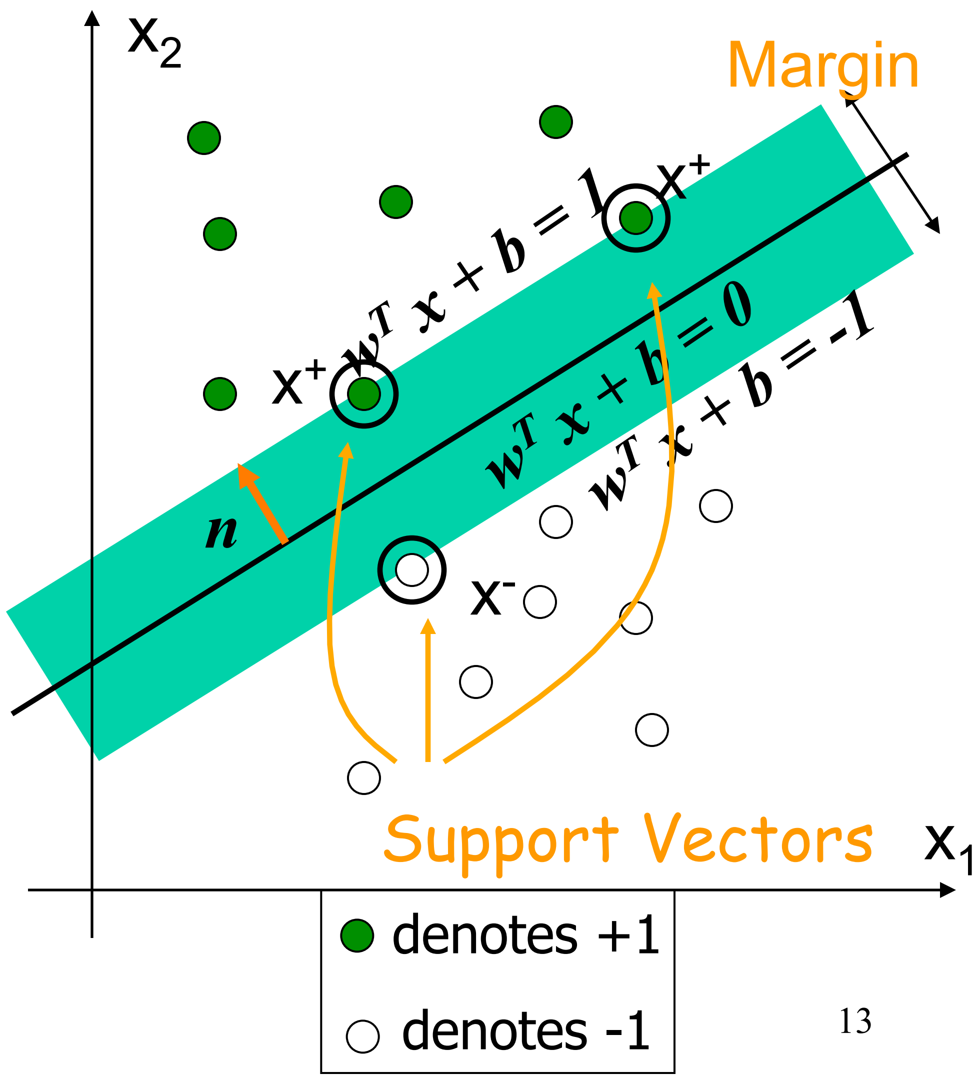
$$\mathbf{w}^T \mathbf{x}^+ + b = 1$$

$$\mathbf{w}^T \mathbf{x}^- + b = -1$$

- The margin width is:

$$M = (\mathbf{x}^+ - \mathbf{x}^-) \cdot \mathbf{n}$$

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



Large Margin Linear Classifier

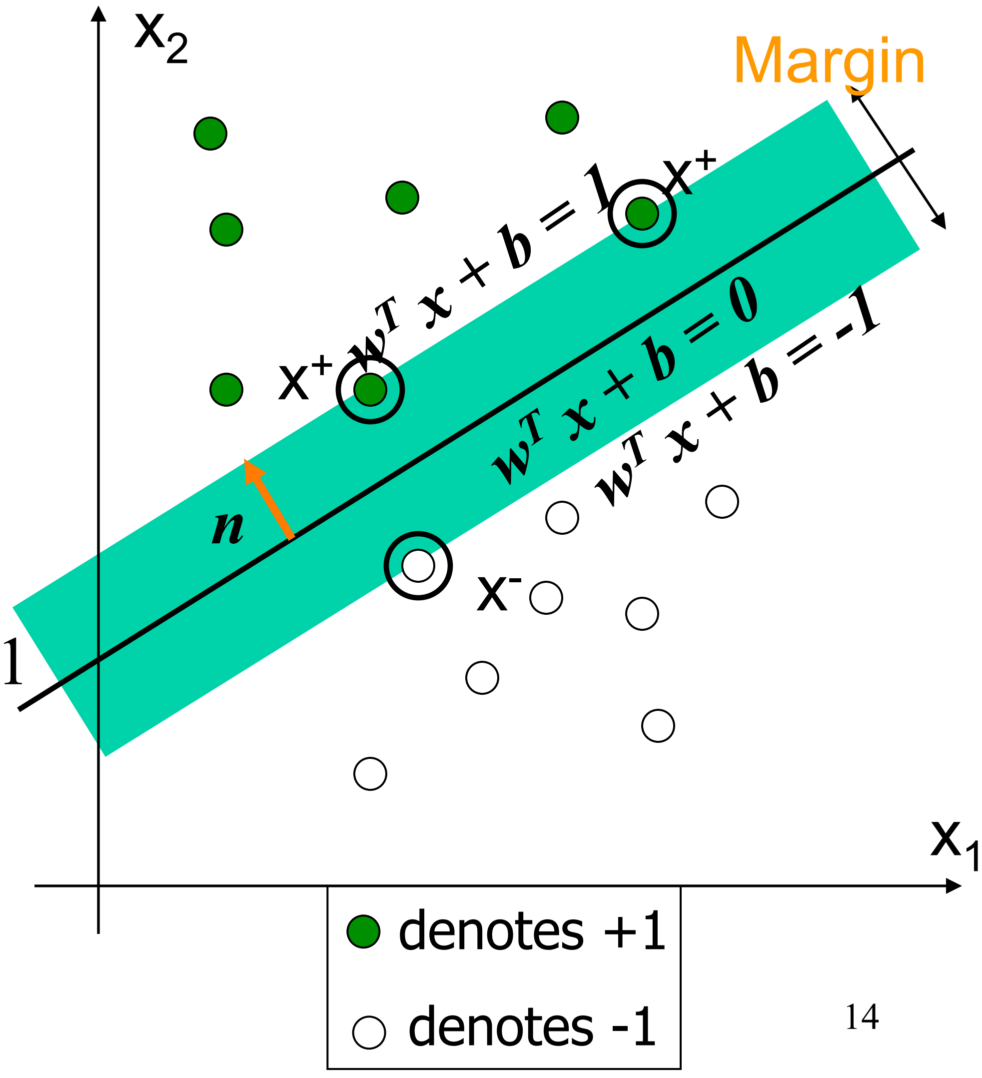
Formulation:

$$\text{maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

such that

$$\text{For } y_i = +1, \quad \mathbf{w}^T \mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1$$



Large Margin Linear Classifier

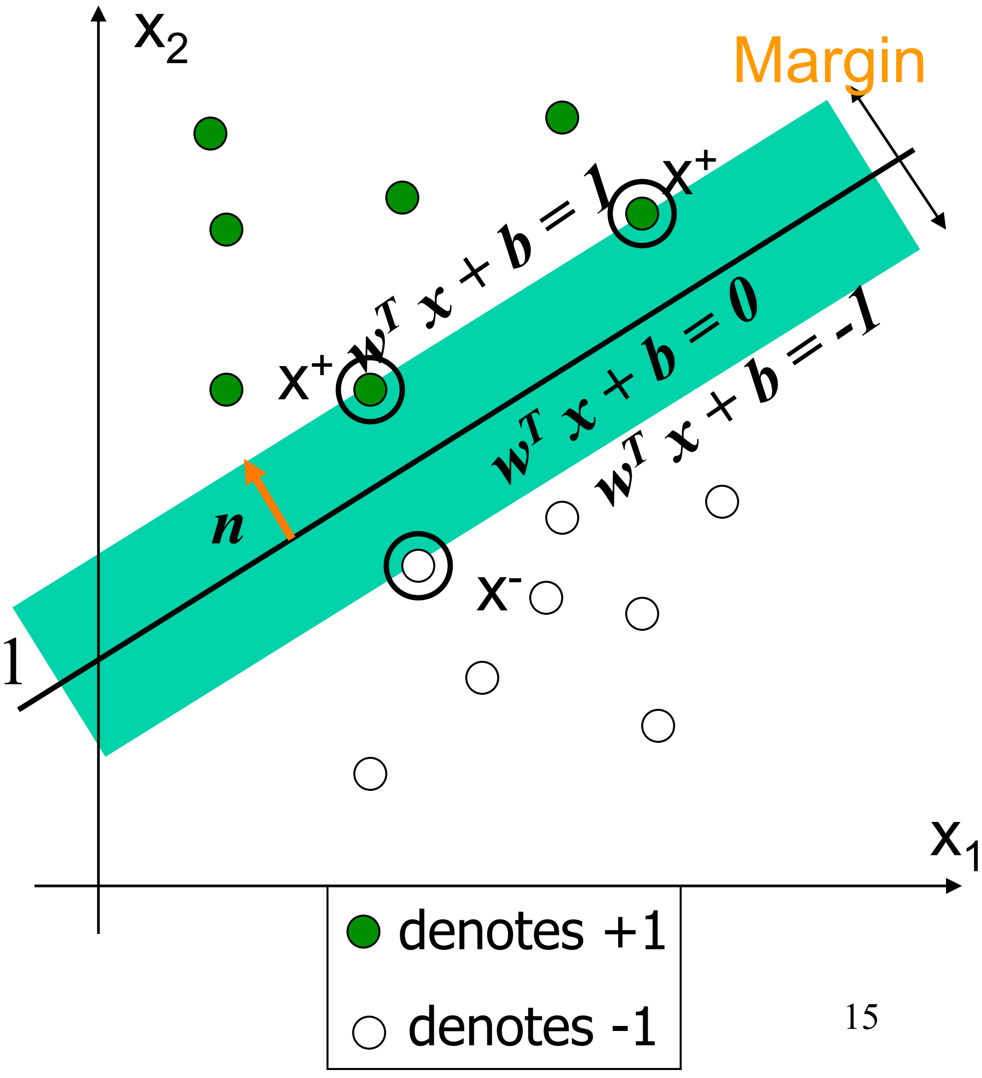
Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

such that

$$\text{For } y_i = +1, \quad \mathbf{w}^T\mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T\mathbf{x}_i + b \leq -1$$



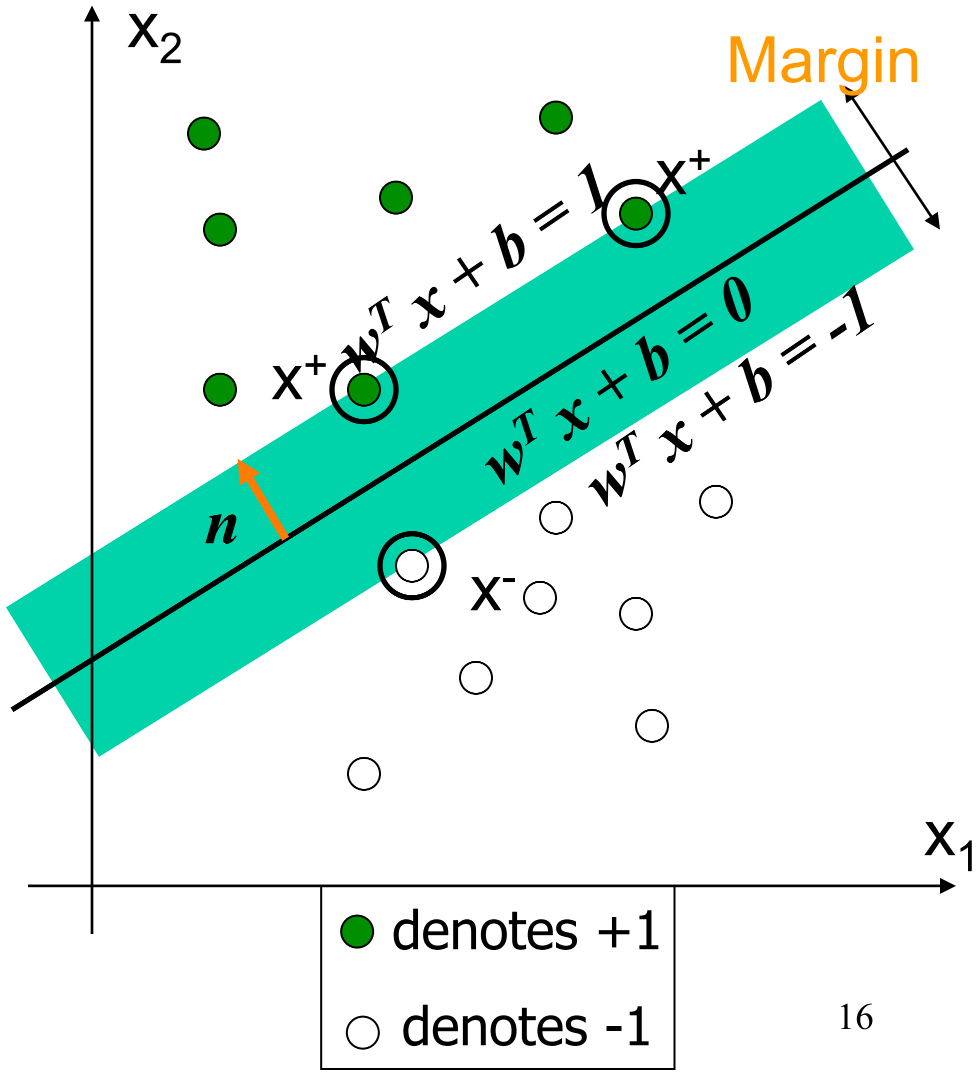
Large Margin Linear Classifier

Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

such that

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



Solving the Optimization Problem

Quadratic programming with linear constraints

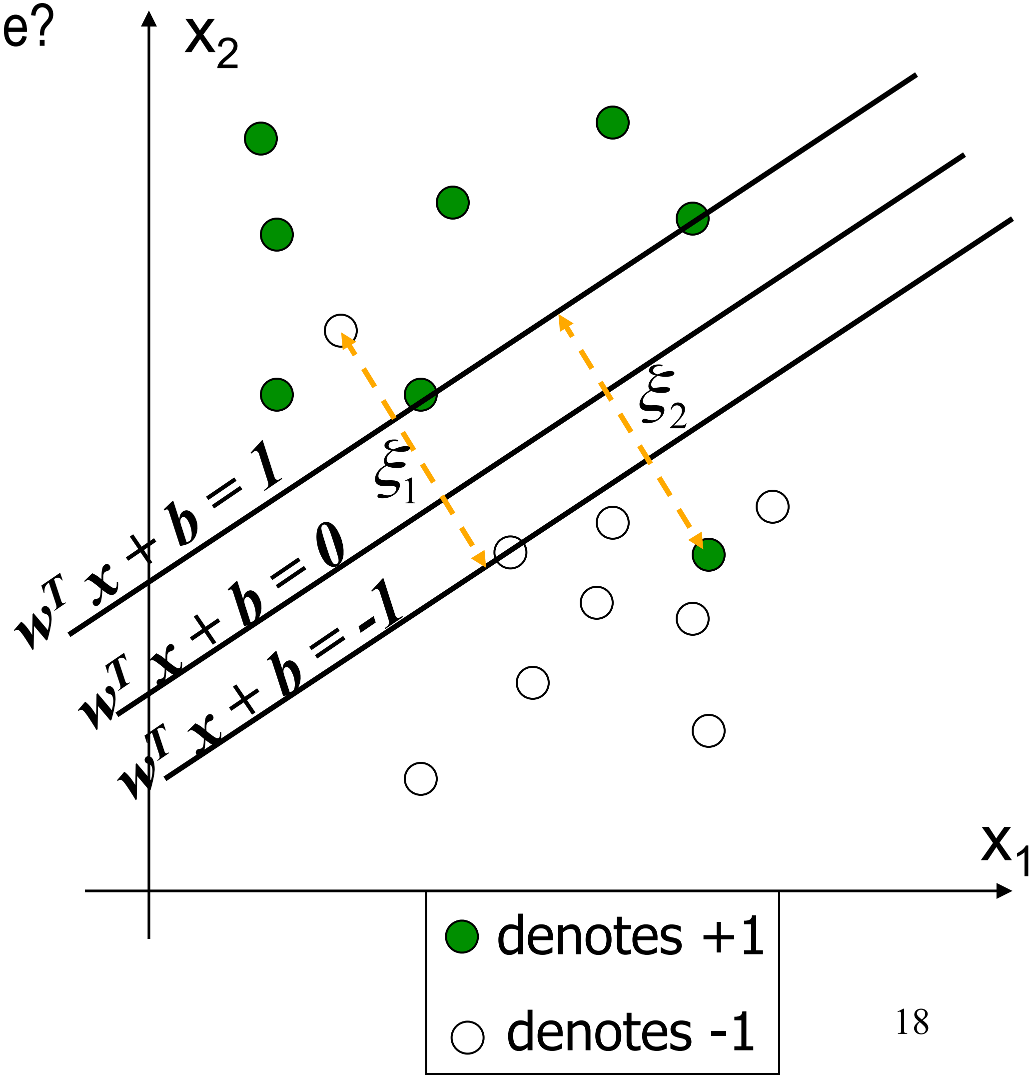
$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

Large Margin Linear Classifier

What if data is not linear separable?
(noisy data, outliers, etc.)

- Slack variables ξ_i can be added to allow mis-classification of difficult or noisy data points



Large Margin Linear Classifier

- Formulation:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

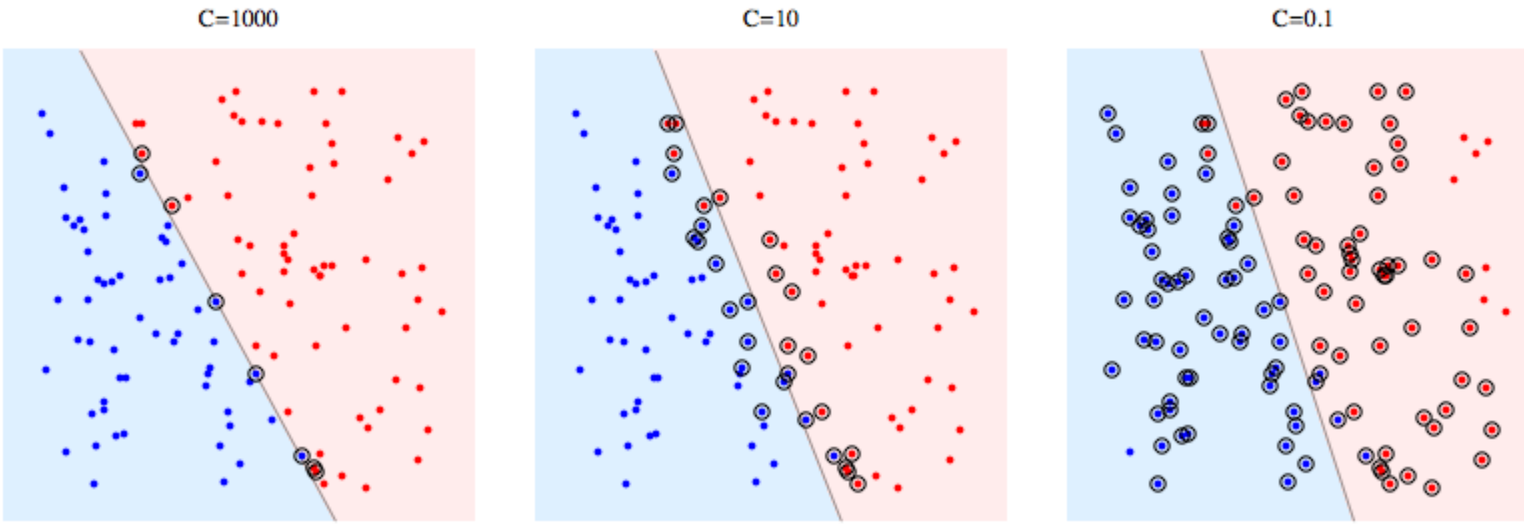
such that

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

$$\xi_i \geq 0$$

- Parameter C can be viewed as a way to control over-fitting.

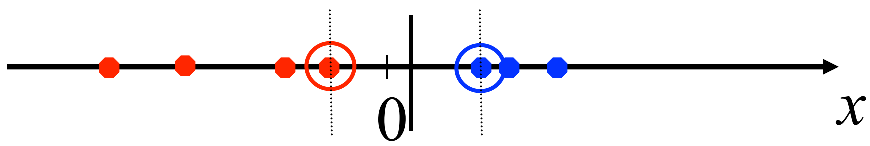
Soft and Hard margin



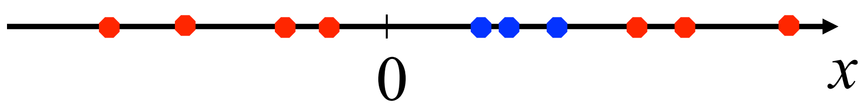
- Circled points show support vectors.
- Decreasing C causes classifier to sacrifice linear separability in order to gain stability, in a sense that influence of any single datapoint is now bounded by C.

Non-linear SVMs

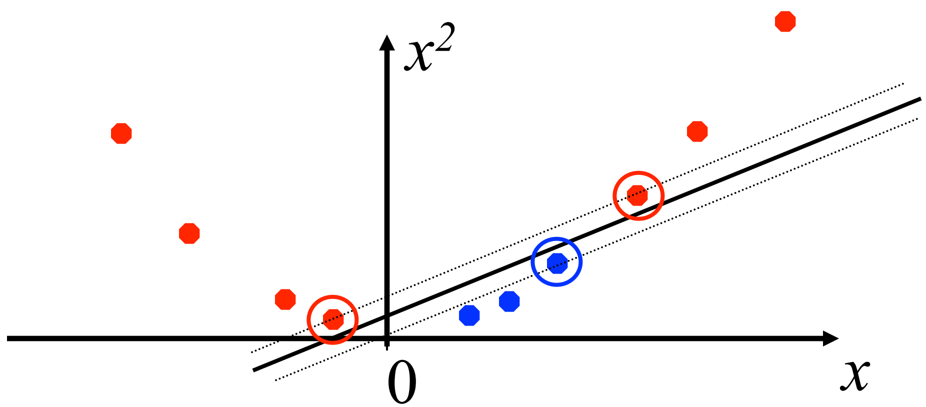
- Datasets that are linearly separable with noise work out great:



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

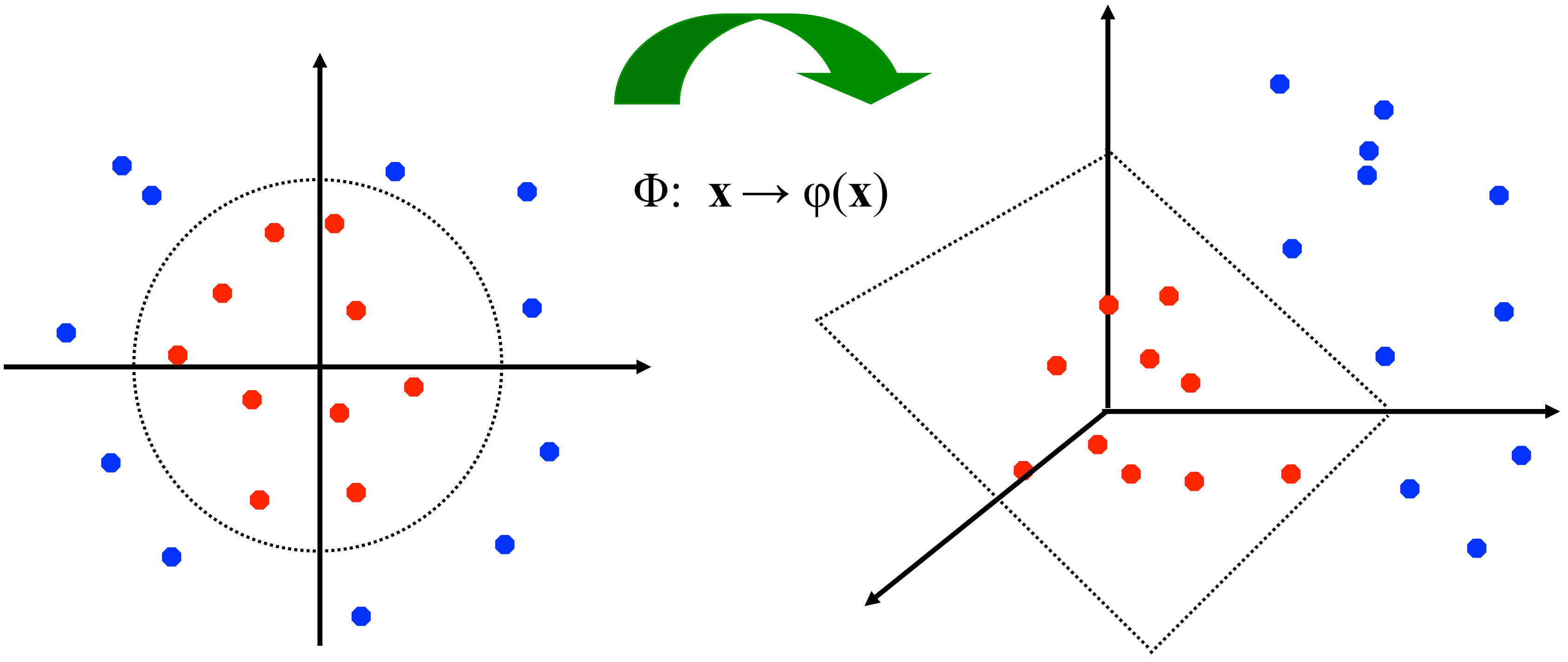


- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature Space

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



Nonlinear SVMs: The Kernel Trick

- Examples of commonly-used kernel functions:
  - Linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
  - Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
  - Gaussian (Radial-Basis Function (RBF)) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2})$$

  - Sigmoid:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$$

Support Vector Machine: Algorithm

1. Choose a kernel function

2. Choose a value for *C*

3. Solve the quadratic programming problem (many software packages available)

4. Construct the discriminant function from the support vectors

Multiclass classification

- Specify n(n-1)/2 classifiers of the form "one against one" and choose the "most voted" class.
- Specify n classifiers of the form "one against all" and choose the class with larger score.
- Specify a tree of classifiers of the form "one against the remaining" until a single class is selected.

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