

Support Vector Machines

Problem Definition

Consider a training set of n iid samples

$$(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_n, \mathbf{y}_n)$$

where \mathbf{x}_i is a vector of length m and

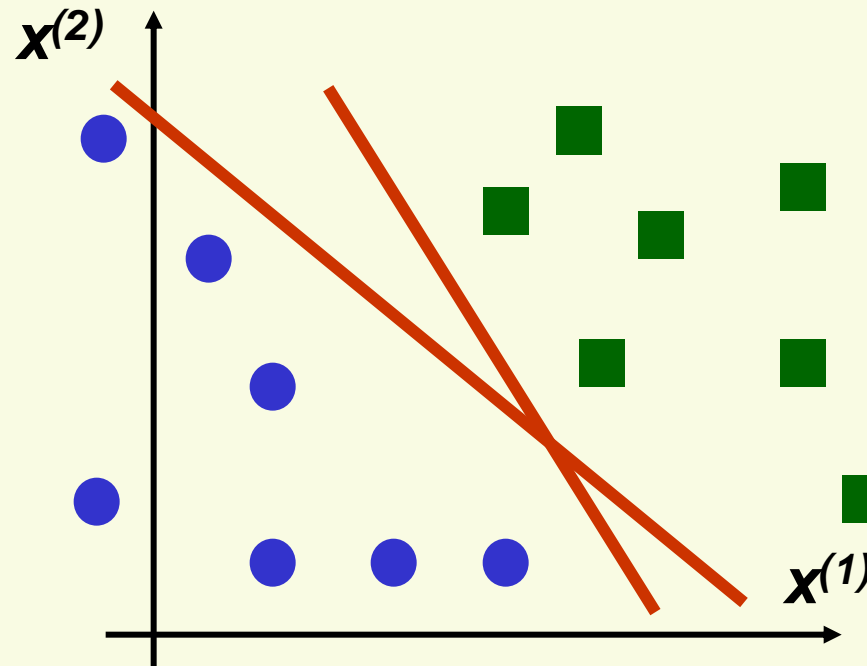
$\mathbf{y}_i \in \{+1, -1\}$ is the class label for data point \mathbf{x}_i .

Find a separating hyperplane $\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0$

corresponding to the decision function

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

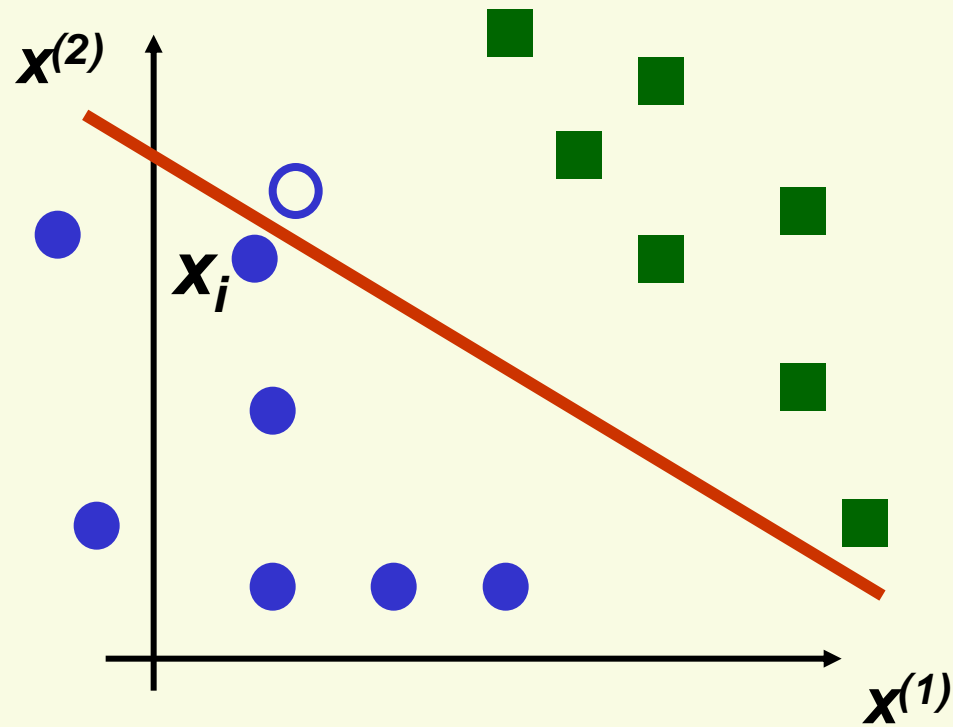
Separating Hyperplanes



- which separating hyperplane should we choose?

Separating Hyperplanes

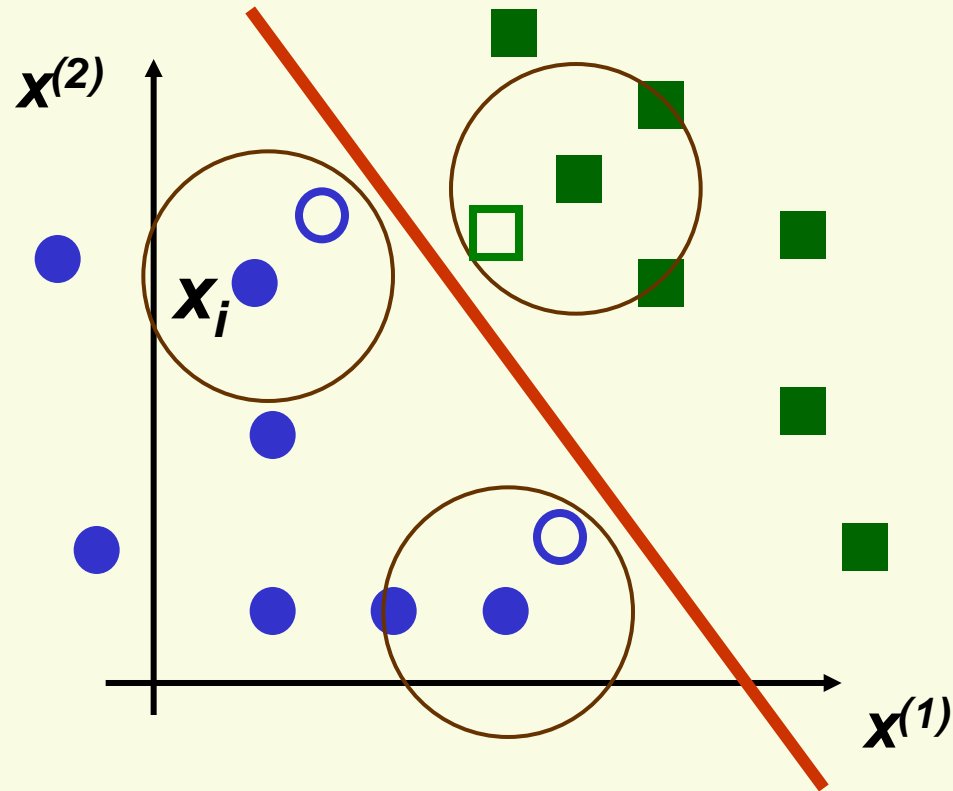
- Training data is just a subset of all possible data
- Suppose hyperplane is close to sample \mathbf{x}_i
- If we see new sample close to sample i , it is likely to be on the wrong side of the hyperplane



- Poor generalization (performance on unseen data)

Separating Hyperplanes

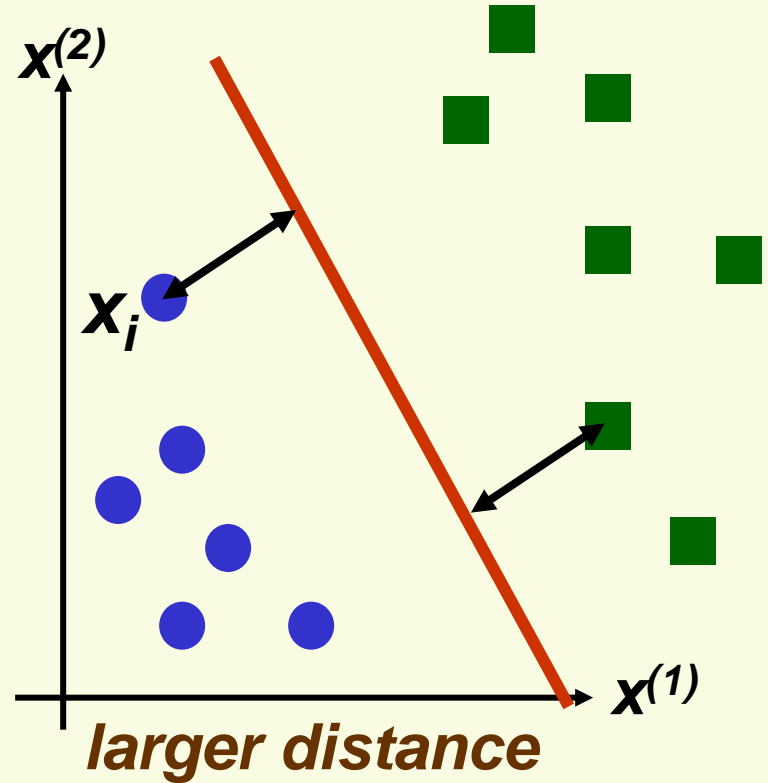
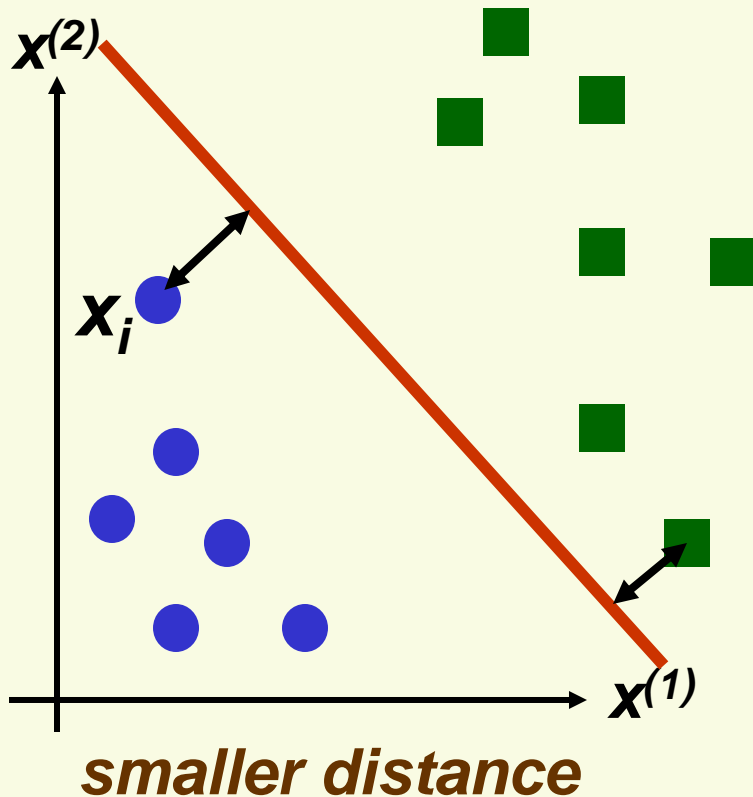
- Hyperplane as far as possible from any sample



- New samples close to the old samples will be classified correctly
- Good generalization

SVM

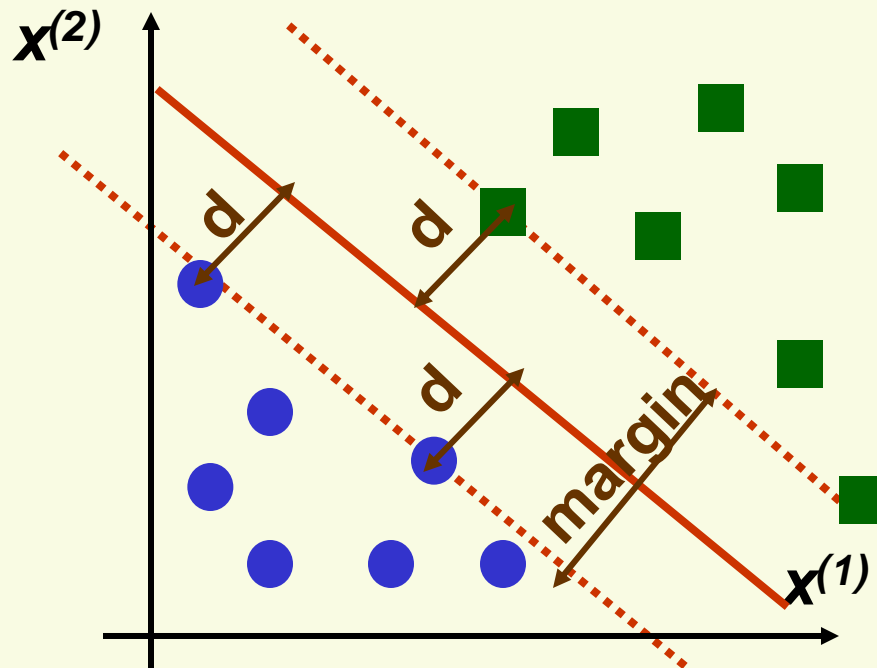
- Idea: maximize distance to the closest example



- For the optimal hyperplane
 - distance to the closest negative example = distance to the closest positive example

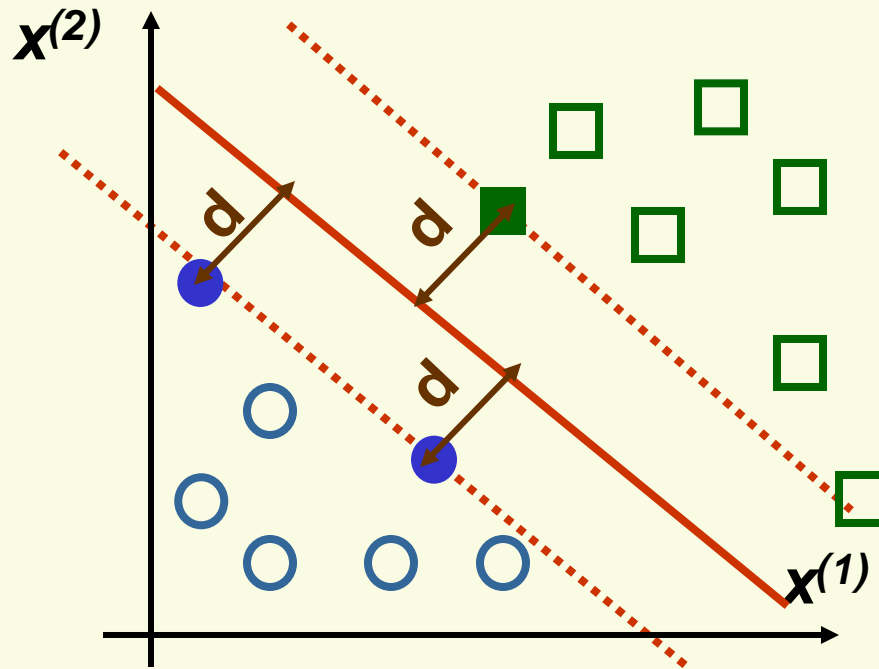
SVM: Linearly Separable Case

- SVM: maximize the *margin*



- *margin* is twice the absolute value of distance d of the closest examples to the separating hyperplane
- Better generalization (performance on test data)
 - in practice
 - and in theory

SVM: Linearly Separable Case

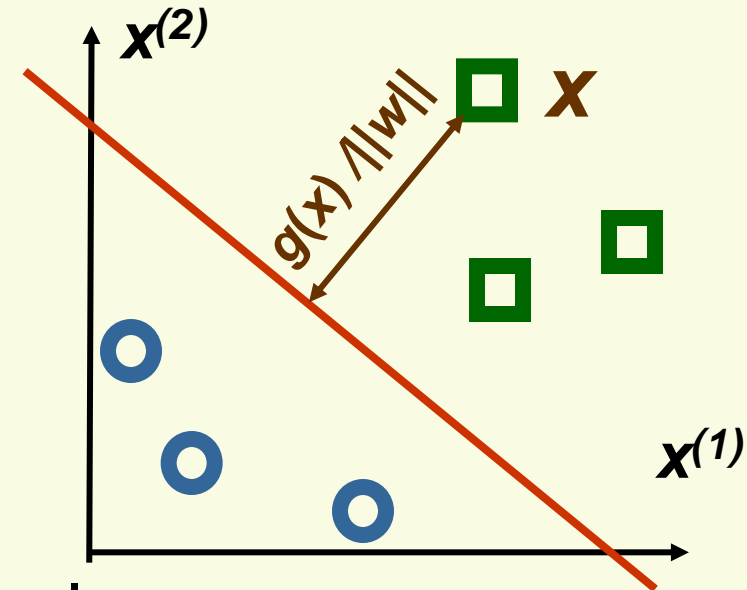


- **Support vectors** are the samples closest to the separating hyperplane
 - they are the most difficult patterns to classify
 - Optimal hyperplane is completely defined by support vectors
 - of course, we do not know which samples are support vectors without finding the optimal hyperplane

SVM: Formula for the Margin

- $g(\mathbf{x}) = \mathbf{w}^t \mathbf{x} + b$
- absolute distance between \mathbf{x} and the boundary $g(\mathbf{x}) = 0$

$$\frac{|\mathbf{w}^t \mathbf{x} + b|}{\|\mathbf{w}\|}$$



- distance is unchanged for hyperplane

$$g_1(\mathbf{x}) = \alpha g(\mathbf{x})$$

$$\frac{|\alpha \mathbf{w}^t \mathbf{x} + \alpha b|}{\|\alpha \mathbf{w}\|} = \frac{|\mathbf{w}^t \mathbf{x} + b|}{\|\mathbf{w}\|}$$

- Let \mathbf{x}_i be an example closest to the boundary. Set

$$|\mathbf{w}^t \mathbf{x}_i + b| = 1$$

- Now the largest margin hyperplane is unique

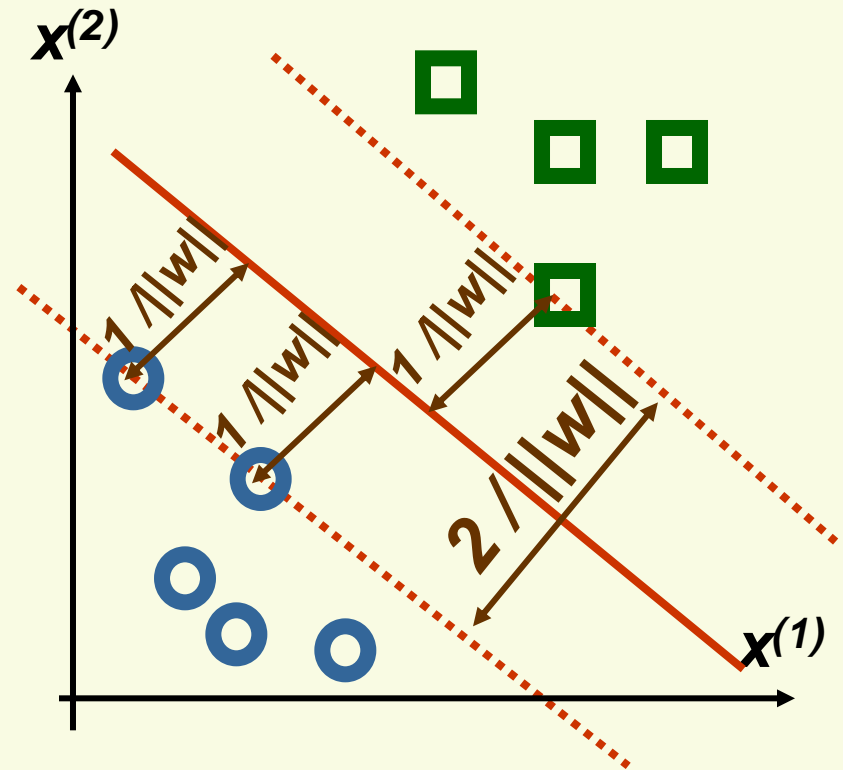
SVM: Formula for the Margin

- For uniqueness, set $|\mathbf{w}^t \mathbf{x}_i + b| = 1$ for any example \mathbf{x}_i closest to the boundary
- now distance from closest sample \mathbf{x}_i to $\mathbf{g}(\mathbf{x}) = 0$ is

$$\frac{|\mathbf{w}^t \mathbf{x}_i + b|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

- Thus the margin is

$$m = \frac{2}{\|\mathbf{w}\|}$$



SVM: Optimal Hyperplane

- Maximize margin $m = \frac{2}{\|\mathbf{w}\|}$

subject to constraints

$$\begin{cases} \mathbf{w}^t \mathbf{x}_i + b \geq 1 & y_i = 1 \\ \mathbf{w}^t \mathbf{x}_i + b \leq -1 & y_i = -1 \end{cases}$$

- Can convert our problem to

$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{s.t.} \quad y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$$

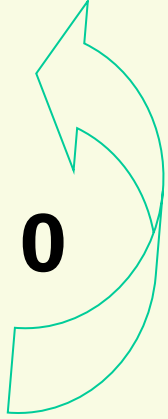
- $J(\mathbf{w})$ is a quadratic function, thus there is a single global minimum

Constrained Quadratic Programming

Primal Problem:

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

$$\text{subject to } \mathbf{y}_i (\mathbf{w} \cdot \mathbf{x}_i + \mathbf{b}) \geq 1, \quad \forall i$$

- Introduce Lagrange multipliers $\alpha_i \geq 0$ associated with the constraints
 - The solution to the primal problem is equivalent to determining the saddle point of the function
- 

$$L_P \equiv L(\mathbf{w}, \mathbf{b}, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (\mathbf{y}_i (\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b}) - 1)$$

Solving Constrained QP

- At saddle point, L_P has minimum requiring

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i \mathbf{y}_i \mathbf{x}_i = \mathbf{0} \Rightarrow \mathbf{w} = \sum_i \alpha_i \mathbf{y}_i \mathbf{x}_i$$

$$\frac{\partial L_P}{\partial \mathbf{b}} = \sum_i \alpha_i \mathbf{y}_i = \mathbf{0}$$

Primal-Dual

Primal:

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^n \alpha_i$$

minimize L_P with respect to \mathbf{w}, b ,
subject to $\alpha_i \geq 0$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \quad \sum_i \alpha_i y_i = 0 \quad \text{substitute}$$

Dual:

$$L_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

maximize L_D with respect to α
subject to $\alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0$

Solving QP using dual problem

$$\begin{aligned} &\text{maximize } L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j \\ &\text{constrained to } \alpha_i \geq 0 \quad \forall i \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

- $\alpha = \{\alpha_1, \dots, \alpha_n\}$ are new variables, one for each sample
- $L_D(\alpha)$ can be optimized by quadratic programming
- $L_D(\alpha)$ formulated in terms of α
 - it depends on \mathbf{w} and \mathbf{b} indirectly
- $L_D(\alpha)$ depends on the number of samples, not on dimension of samples

Threshold

- \mathbf{b} can be determined from the optimal α and Karush-Kuhn-Tucker (KKT) conditions

$$\alpha_i [\mathbf{y}_i (\mathbf{w} \cdot \mathbf{x}_i + \mathbf{b}) - 1] = 0, \quad \forall i$$

- $\alpha_i > 0$ implies

$$\mathbf{y}_i (\mathbf{w} \cdot \mathbf{x}_i + \mathbf{b}) = 1 \quad \Rightarrow \quad \mathbf{w} \cdot \mathbf{x}_i + \mathbf{b} = \mathbf{y}_i$$

$$\mathbf{b} = \mathbf{y}_i - \mathbf{w} \cdot \mathbf{x}_i$$

Support Vectors

- For every sample i , one of the following must hold
 - $\alpha_i = 0$
 - $\alpha_i > 0$ and $y_i(\mathbf{w} \cdot \mathbf{x}_i + b - 1) = 0$
- Many $\alpha_i = 0 \Rightarrow \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$ sparse solution
- Samples with $\alpha_i > 0$ are **Support Vectors** and they are the closest to the separating hyperplane
- Optimal hyperplane is completely defined by support vectors

SVM: Classification

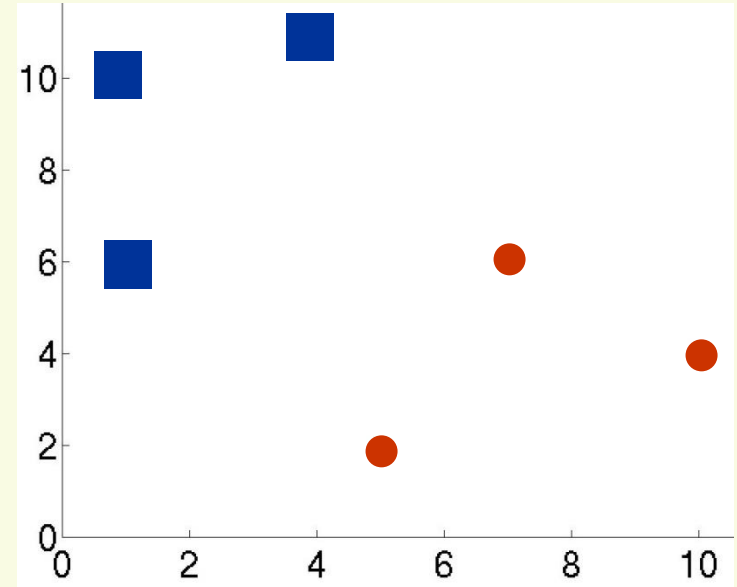
- Given a new sample x , finds its label y

$$***y = \text{sign}(w \cdot x + b)***$$

$$***w = \sum_{i=1}^n \alpha_i y_i x_i***$$

SVM: Example

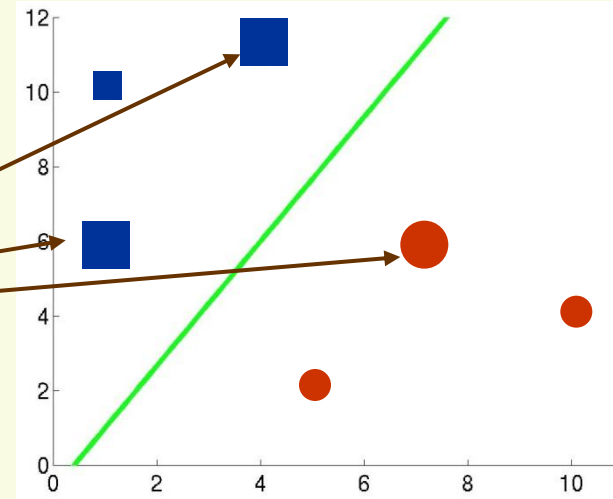
- Class 1: [1,6], [1,10], [4,11]
- Class 2: [5,2], [7,6], [10,4]



SVM: Example

■ Solution $\alpha = \begin{bmatrix} 0.036 \\ 0 \\ 0.039 \\ 0 \\ 0.076 \\ 0 \end{bmatrix}$

*support
vectors*



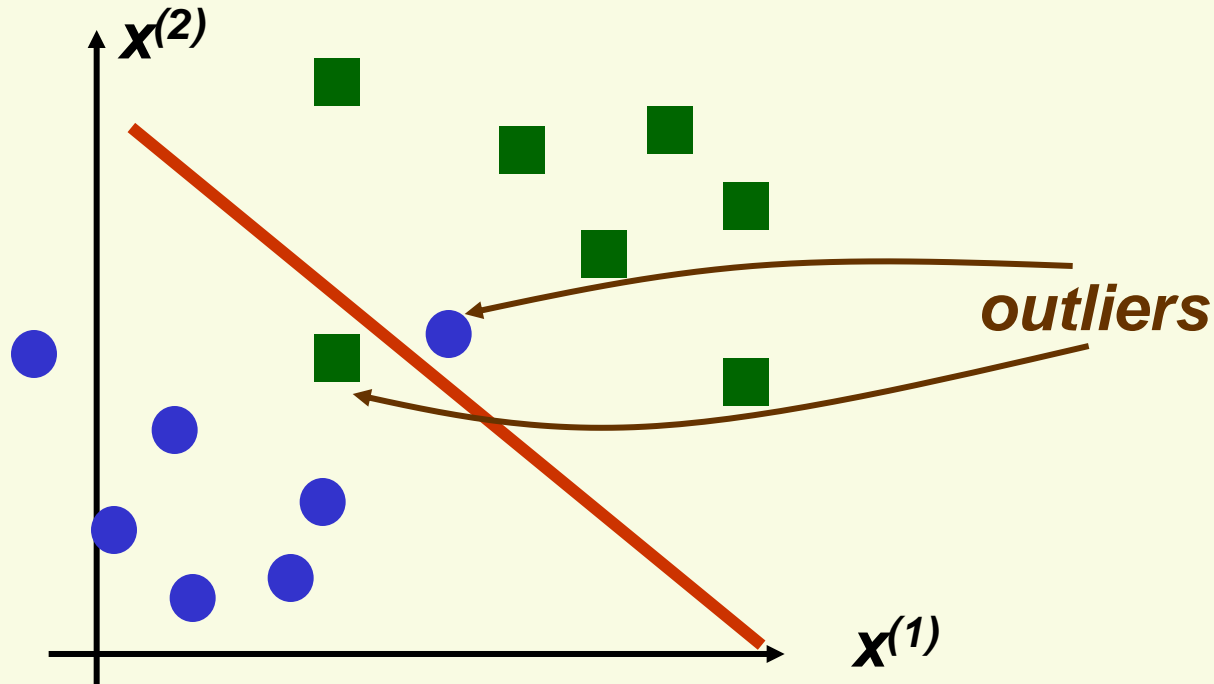
■ find w using $w = \sum_{i=1}^n \alpha_i y_i x_i = (\alpha \cdot y)^t x = \begin{bmatrix} -0.33 \\ 0.20 \end{bmatrix}$

■ since $\alpha_1 > 0$, can find b using

$$b = y_1 - w^t x_1 = 0.13$$

SVM: Non Separable Case

- Data is most likely to be not linearly separable, but linear classifier may still be appropriate



- Can apply SVM in non linearly separable case
 - data should be “almost” linearly separable for good performance

SVM with slacks

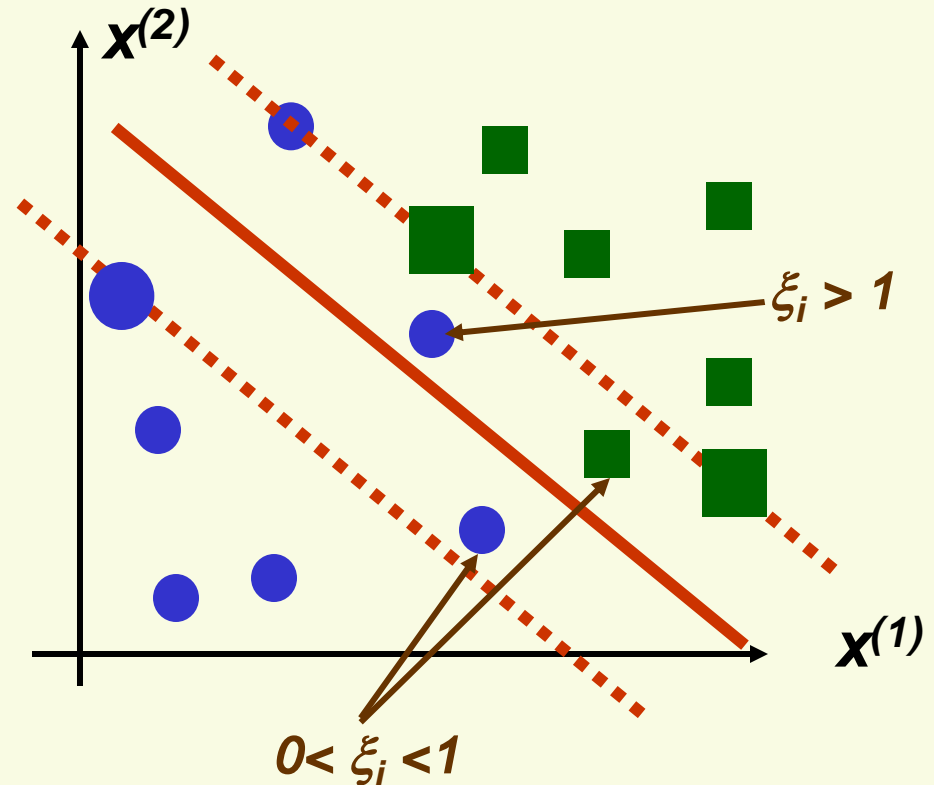
- Use nonnegative “slack” variables ξ_1, \dots, ξ_n (one for each sample)

- Change constraints from $y_i(\mathbf{w}^t \mathbf{x}_i + b) \geq 1 \quad \forall i$ to

$$y_i(\mathbf{w}^t \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i$$

- ξ_i is a measure of deviation from the ideal position for sample i

- $\xi_i > 1$ sample i is on the wrong side of the separating hyperplane
- $0 < \xi_i < 1$ sample i is on the right side of separating hyperplane but within the region of maximum margin



SVM with slacks

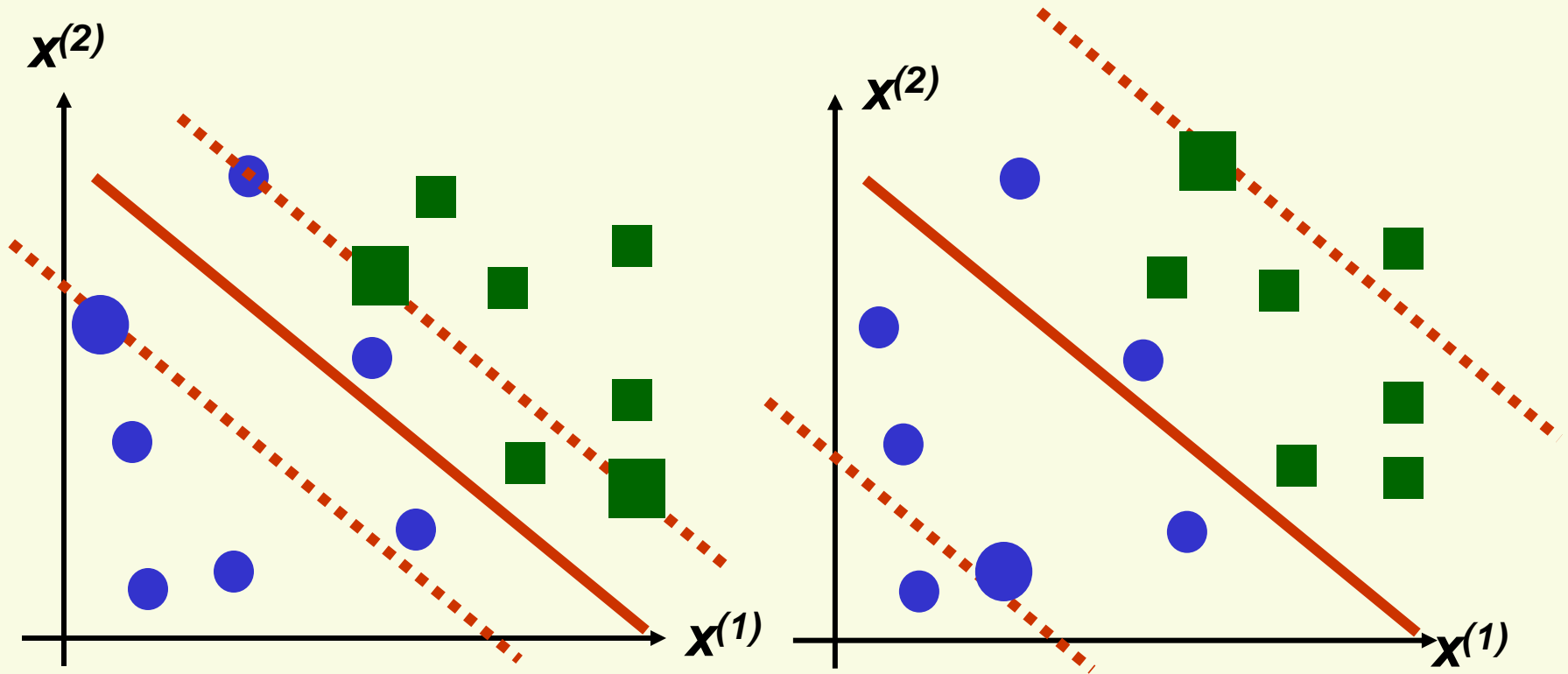
- Would like to minimize

$$J(\mathbf{w}, \xi_1, \dots, \xi_n) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$

- constrained to $\mathbf{y}_i(\mathbf{w}^t \mathbf{x}_i + \mathbf{b}) \geq 1 - \xi_i$ and $\xi_i \geq 0 \quad \forall i$
- $C > 0$ is a constant which measures relative weight of the first and second terms
 - if C is small, we allow a lot of samples not in ideal position
 - if C is large, we want to have very few samples not in ideal position

SVM with slacks

$$J(w, \xi_1, \dots, \xi_n) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$



large C, few samples not in ideal position

small C, a lot of samples not in ideal position

SVM with slacks– Dual Formulation

maximize $L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$

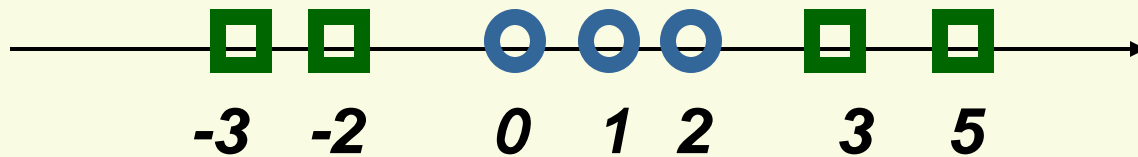
constrained to $0 \leq \alpha_i \leq C \quad \forall i \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0$

- find \mathbf{w} using $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$
- solve for \mathbf{b} using any $0 < \alpha_i < C$ and $\alpha_i [\mathbf{y}_i (\mathbf{w}^t \mathbf{x}_i + \mathbf{b}) - 1] = 0$

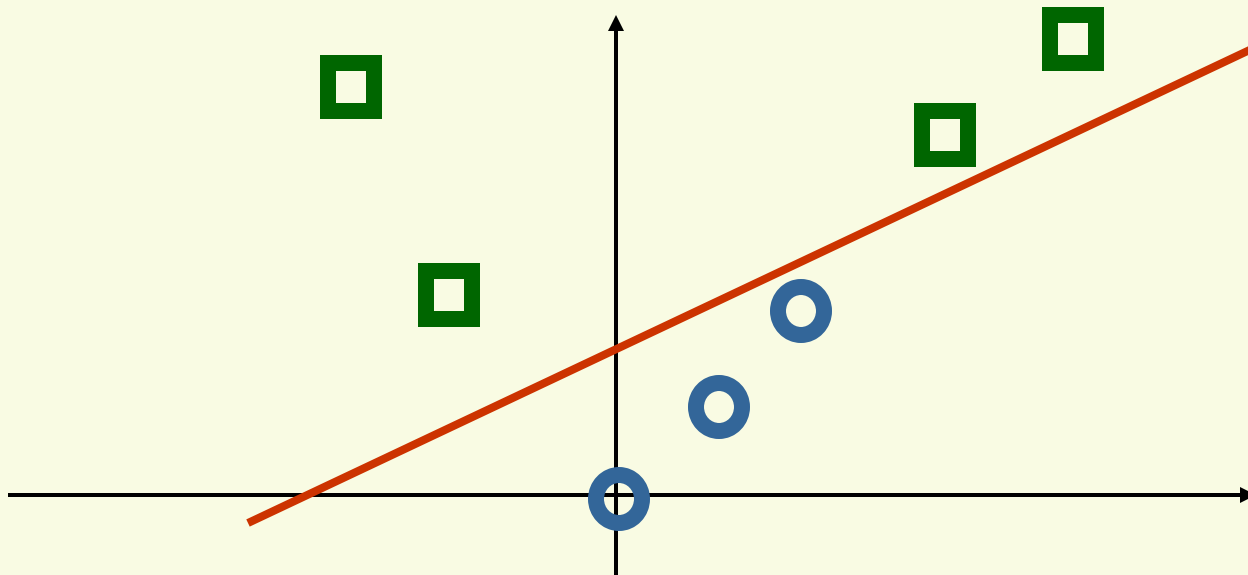
Non Linear Mapping

- Cover's theorem:
 - “*pattern-classification problem cast in a high dimensional space non-linearly is more likely to be linearly separable than in a low-dimensional space*”

- One dimensional space, not linearly separable

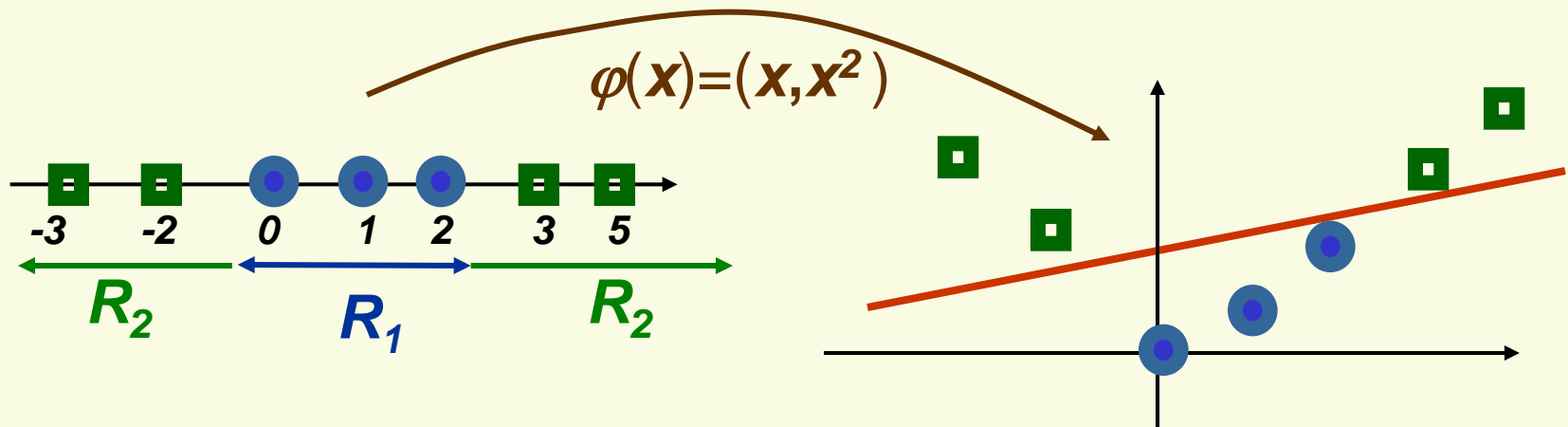


- Lift to two dimensional space with $\phi(\mathbf{x})=(\mathbf{x},\mathbf{x}^2)$



Non Linear Mapping

- Solve a non linear classification problem with a linear classifier
 1. Project data \mathbf{x} to high dimension using function $\varphi(\mathbf{x})$
 2. Find a linear discriminant function for transformed data $\varphi(\mathbf{x})$
 3. Final nonlinear discriminant function is $\mathbf{g}(\mathbf{x}) = \mathbf{w}^t \varphi(\mathbf{x}) + w_0$

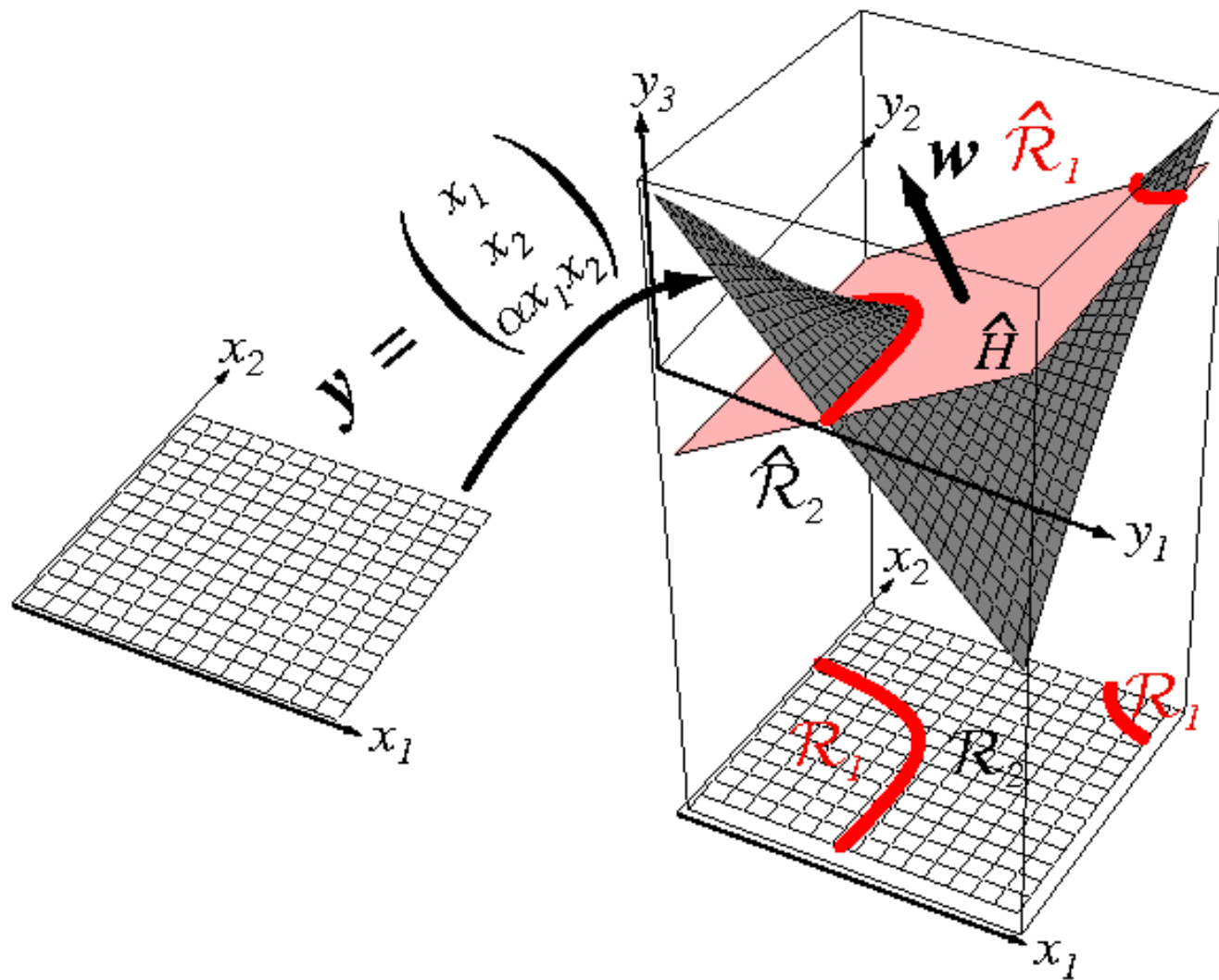


- In 2D, discriminant function is linear

$$\mathbf{g}\left(\begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{bmatrix}\right) = [\mathbf{w}_1 \quad \mathbf{w}_2] \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{bmatrix} + w_0$$

- In 1D, discriminant function is not linear $\mathbf{g}(\mathbf{x}) = \mathbf{w}_1 \mathbf{x} + \mathbf{w}_2 \mathbf{x}^2 + w_0$

Non Linear Mapping: Another Example



Non Linear SVM

- Can use any linear classifier after lifting data into a higher dimensional space. However we will have to deal with the “curse of dimensionality”
 1. poor generalization to test data
 2. computationally expensive
- SVM handles the “curse of dimensionality” problem:
 1. enforcing largest margin permits good generalization
 - It can be shown that generalization in SVM is a function of the margin, independent of the dimensionality
 2. computation in the higher dimensional case is performed only implicitly through the use of **kernel** functions

Non Linear SVM: Kernels

- Recall SVM optimization

$$\text{maximize } L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$

and classification $y = \text{sign}(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b)$

- Note that samples \mathbf{x}_i appear only through the dot products $\mathbf{x}_i^t \mathbf{x}_j$, $\mathbf{x}_i^t \mathbf{x}$.

- If we lift \mathbf{x}_i to high dimensional space F using $\varphi(\mathbf{x})$, need to compute high dimensional product $\varphi(\mathbf{x}_i)^t \varphi(\mathbf{x}_j)$

$$\text{maximize } L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \varphi(\mathbf{x}_i)^t \varphi(\mathbf{x}_j)$$

- The dimensionality of space F not necessarily important. May not even know the map φ .

Kernel

- A function that returns the value of the dot product between the images of the two arguments:

$$K(x, y) = \varphi(\mathbf{x})^t \varphi(\mathbf{y})$$

- Given a function K , it is possible to verify that it is a kernel.

$$\text{maximize } L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \varphi(\mathbf{x}_i)^t \varphi(\mathbf{x}_j)$$

$K(\mathbf{x}_i, \mathbf{x}_j)$

- Now we only need to compute $K(\mathbf{x}_i, \mathbf{x}_j)$ instead of $\varphi(\mathbf{x}_i)^t \varphi(\mathbf{x}_j)$
- “kernel trick”: do not need to perform operations in high dimensional space explicitly

Kernel Matrix

- (aka the Gram matrix):

$K =$

$K(1,1)$	$K(1,2)$	$K(1,3)$...	$K(1,m)$
$K(2,1)$	$K(2,2)$	$K(2,3)$...	$K(2,m)$
...
$K(m,1)$	$K(m,2)$	$K(m,3)$...	$K(m,m)$

- The central structure in kernel machines
- Contains all necessary information for the learning algorithm
- Fuses information about the data AND the kernel
- Many interesting properties:

Mercer's Theorem

- The kernel matrix is Symmetric Positive Definite
- Any symmetric positive definite matrix can be regarded as a kernel matrix, that is as an inner product matrix in some space

Every (semi)positive definite, symmetric function is a kernel: i.e. there exists a mapping φ such that it is possible to write:

$$K(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x})^t \varphi(\mathbf{y})$$

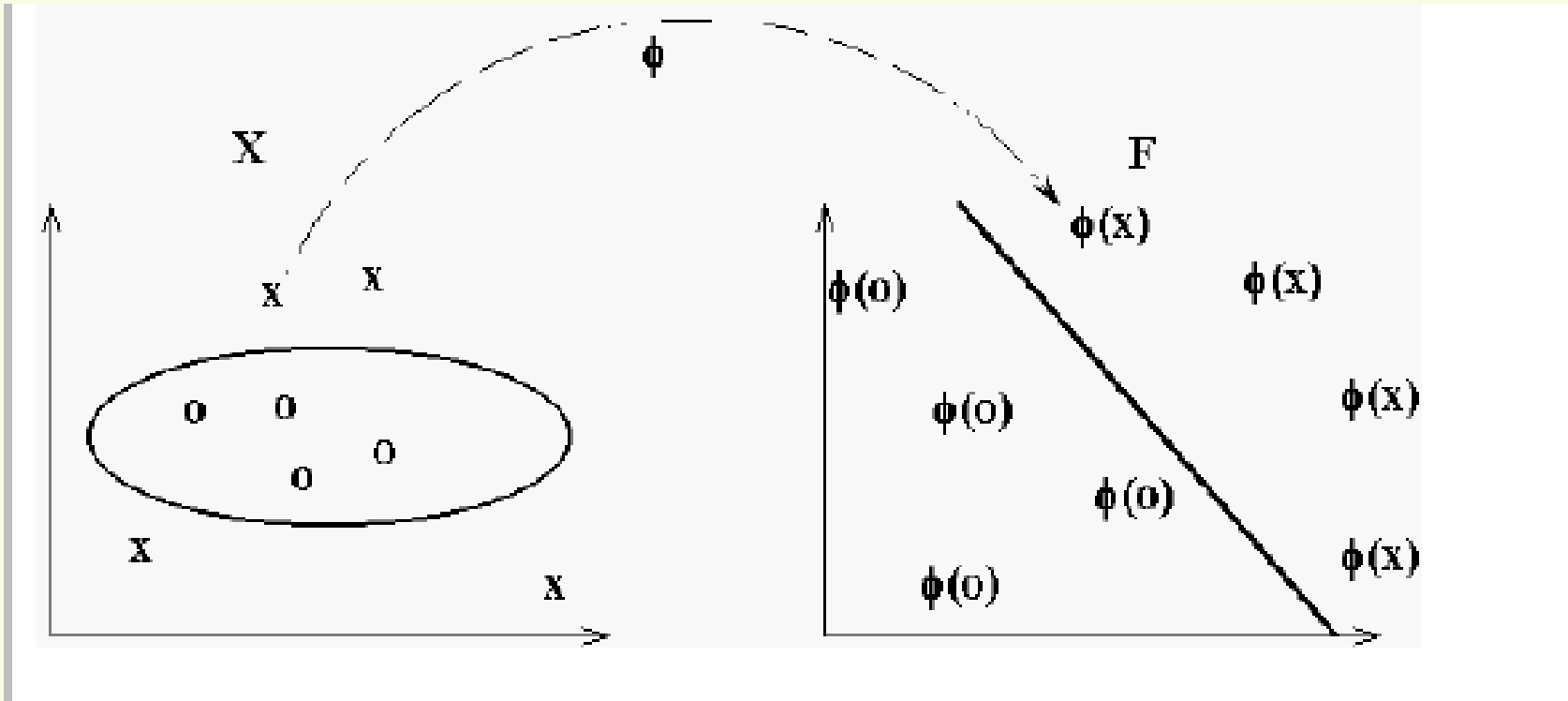
Positive definite $\int_{\forall f \in L_2} K(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0$

Examples of Kernels

- Some common choices (both satisfying Mercer's condition):
 - Polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^t \mathbf{x}_j + 1)^p$
 - Gaussian radial Basis kernel (data is lifted in infinite dimension)

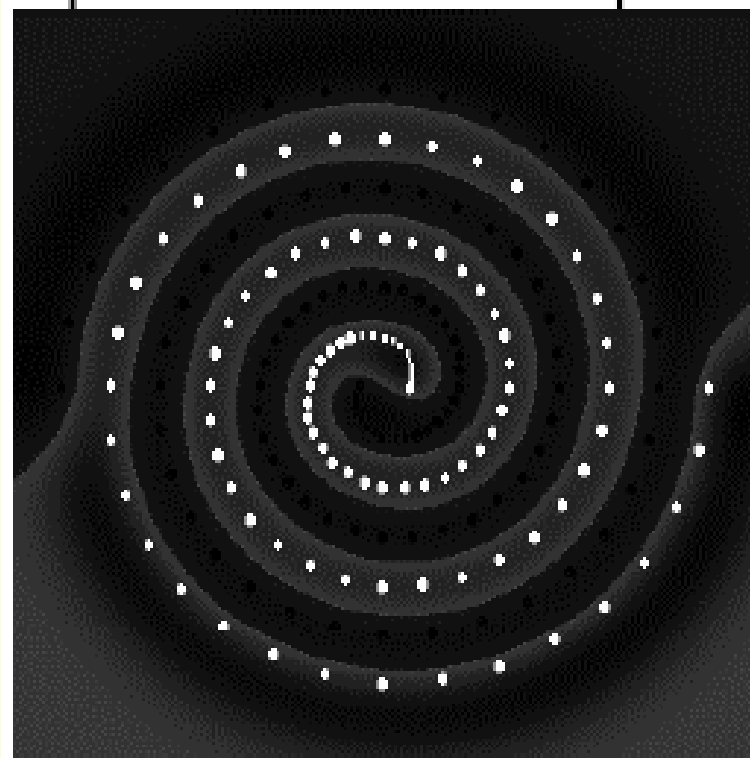
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

Example Polynomial Kernels



Example: the two spirals

- Separated by a hyperplane in feature space (gaussian kernels)



Making Kernels

- The set of kernels is closed under some operations. If K , K' are kernels, then:
- $K+K'$ is a kernel
- cK is a kernel, if $c>0$
- $aK+bK'$ is a kernel, for $a,b >0$
- Etc etc etc.....
- can make complex kernels from simple ones: modularity !

Non Linear SVM Recipe

- Start with data $\mathbf{x}_1, \dots, \mathbf{x}_n$ which lives in feature space of dimension d
- Choose kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ corresponding to some function $\varphi(\mathbf{x}_i)$ which takes sample \mathbf{x}_i to a higher dimensional space
- Find the largest margin linear discriminant function in the higher dimensional space by using quadratic programming package to solve:

$$\text{maximize } L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{constrained to } \mathbf{0} \leq \alpha_i \leq \mathbf{C} \quad \forall i \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = \mathbf{0}$$

Non Linear SVM Recipe

- Weight vector \mathbf{w} in the high dimensional space:

$$\mathbf{w} = \sum_i^n \alpha_i y_i \varphi(\mathbf{x}_i)$$

- Linear discriminant function of largest margin in the high dimensional space:

$$\mathbf{g}(\varphi(\mathbf{x})) = \mathbf{w}^t \varphi(\mathbf{x}) = \left(\sum_{\mathbf{x}_i \in \mathcal{S}} \alpha_i y_i \varphi(\mathbf{x}_i) \right)^t \varphi(\mathbf{x})$$

- Non linear discriminant function in the original space:

$$\mathbf{g}(\mathbf{x}) = \left(\sum_{\mathbf{x}_i \in \mathcal{S}} \alpha_i y_i \varphi(\mathbf{x}_i) \right)^t \varphi(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathcal{S}} \alpha_i y_i \varphi^t(\mathbf{x}_i) \varphi(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathcal{S}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})$$

- decide class 1 if $\mathbf{g}(\mathbf{x}) > 0$, otherwise decide class 2

Non Linear SVM

- Nonlinear discriminant function

$$g(\mathbf{x}) = \sum_{\mathbf{x}_i \in S} \alpha_i \mathbf{z}_i K(\mathbf{x}_i, \mathbf{x})$$

$$g(\mathbf{x}) = \sum \begin{array}{|l} \text{weight of support} \\ \text{vector } \mathbf{x}_i \end{array} \begin{array}{|l} \mp 1 \end{array} \begin{array}{|l} \text{"inverse distance"} \\ \text{from } \mathbf{x} \text{ to} \\ \text{support vector } \mathbf{x}_i \end{array}$$

most important training samples, i.e. support vectors

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x}_i - \mathbf{x}\|^2\right)$$

Higher Order Polynomials

Taken from Andrew Moore

Poly-nomial	$\phi(\mathbf{x})$	Cost to build H matrix traditionally	Cost if $d=100$	$\phi(\mathbf{a})^t\phi(\mathbf{b})$	Cost to build H matrix sneakily	Cost if $d=100$
Quadratic	All $d^2/2$ terms up to degree 2	$d^2 n^2 / 4$	2,500 n^2	$(\mathbf{a}^t\mathbf{b}+1)^2$	$d n^2 / 2$	50 n^2
Cubic	All $d^3/6$ terms up to degree 3	$d^3 n^2 / 12$	83,000 n^2	$(\mathbf{a}^t\mathbf{b}+1)^3$	$d n^2 / 2$	50 n^2
Quartic	All $d^4/24$ terms up to degree 4	$d^4 n^2 / 48$	1,960,000 n^2	$(\mathbf{a}^t\mathbf{b}+1)^4$	$d n^2 / 2$	50 n^2

n is the number of samples, d is number of features

SVM Summary

- Advantages:
 - Based on nice theory
 - excellent generalization properties
 - objective function has no local minima
 - can be used to find non linear discriminant functions
 - Complexity of the classifier is characterized by the number of support vectors rather than the dimensionality of the transformed space
- Disadvantages:
 - It's not clear how to select a kernel function in a principled manner
 - tends to be slower than other methods (in non-linear case).