Introduction to Support Vector Machines

Lee, Yuh-Jye

National Taiwan University of Science and Technology

December 23, 2005

Binary Classification Problem

Learn a Classifier from the Training Set

Given a training dataset

$$S = \{(x^i, y_i) \,|\, x^i \in R^n, y_i \in \{-1, 1\}, i = 1, \ldots, m\}$$

$$x^i \in A_+ \Leftrightarrow y_i = 1 \;\; \& \;\; x^i \in A_- \Leftrightarrow y_i = -1$$

Main goal:

Predict the unseen class label for new data

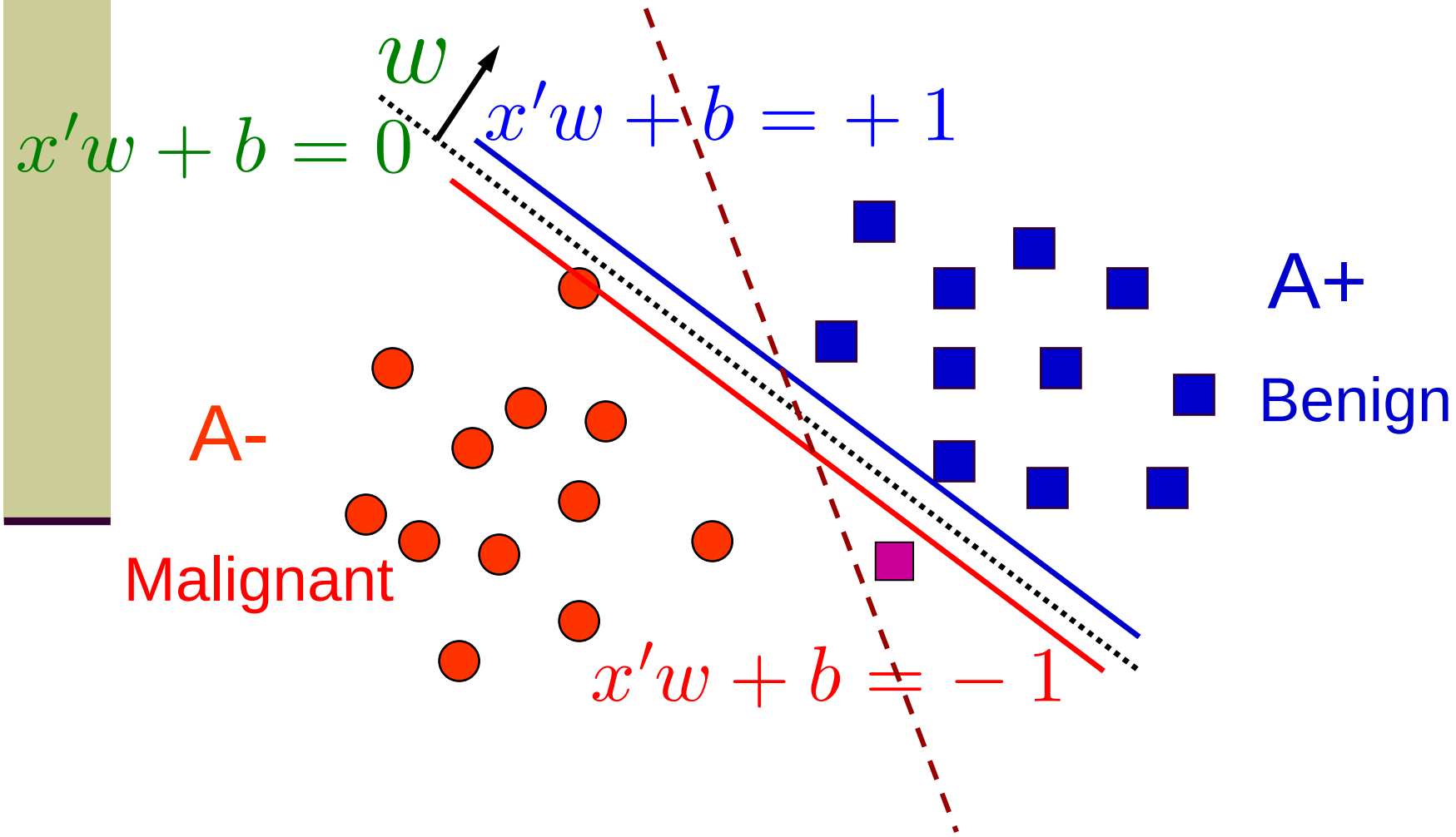
Find a function $f : R^n \rightarrow R$ by learning from data

$$f(x) > 0 \Rightarrow x \in A_+ \;\; and \;\; f(x) < 0 \Rightarrow x \in A_-$$

The simplest function is linear: $f(x) = w'x + b$

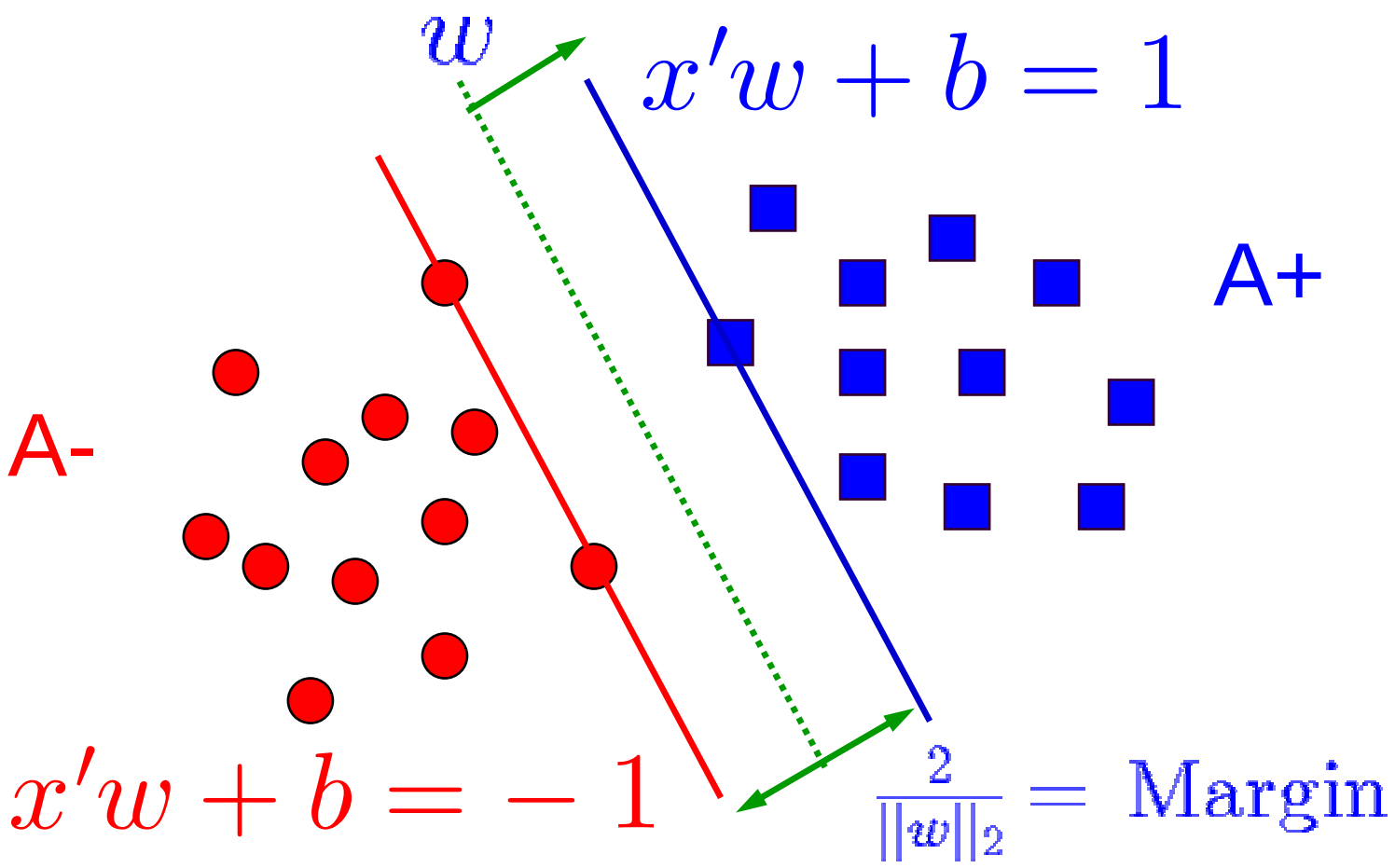
Binary Classification Problem

Linearly Separable Case



Support Vector Machines

Maximizing the Margin between Bounding Planes



Summary the Notations

Let $S = \{(x^1, y_1), (x^2, y_2), \ldots (x^m, y_m)\}$ be a training dataset and represented by matrices

$$A = \begin{bmatrix} (x^1)' \\ (x^2)' \\ \vdots \\ (x^m)' \end{bmatrix} \in R^{m \times n}, \; D = \begin{bmatrix} y_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & y_m \end{bmatrix} \in R^{m \times m}$$

$$\begin{array}{lcll} A_i w + b & > & +1, & for \quad D_{ii} = +1, \\ A_i w + b & 6 & -1, & for \quad D_{ii} = -1 \end{array}$$ equivalent to

$$D(Aw + eb) > e \text{ , where } \quad e = [1, 1, \ldots, 1]' \in R^m.$$

Support Vector Classification

(Linearly Separable Case, Primal)

The hyperplane (w, b) is determined by solving the minimization problem:

$$\min_{(w,b)\in R^{n+1}} \quad \frac{1}{2}||w||_2^2$$

$$D(Aw + eb) > e,$$

It realizes the maximal margin hyperplane with geometric margin $\gamma = \frac{1}{||w||_2}$

Support Vector Classification

(Linearly Separable Case, Dual Form)

The dual problem of previous MP:

$$\max_{\alpha \in R^l} \quad e'\alpha - \tfrac{1}{2}\alpha' DAA'D\alpha$$

subject to

$$e'D\alpha = 0, \quad \alpha \geq 0.$$

Applying the KKT optimality conditions, we have $w = A'D\alpha$. But where is b?

Don't forget $0 \leq \alpha \perp D(Aw + eb) - e \geq 0$

Dual Representation of SVM

(Key of Kernel Methods: $w = A'D\alpha^* = \sum_{i=1}^{\ell} y_i \alpha_i^* A_i'$)

The hypothesis is determined by (α^*, b^*)

$$h(x) = sgn(\langle x, A'D\alpha^* \rangle + b^*)$$

$$= sgn(\sum_{i=1}^{l} y_i \alpha_i^* \langle x^i, x \rangle + b^*)$$

$$= sgn(\sum_{\alpha_i^* > 0} y_i \alpha_i^* \langle x^i, x \rangle + b^*)$$

$$Remember: \quad A_i' = x^i$$

Soft Margin SVM

(Nonseparable Case)

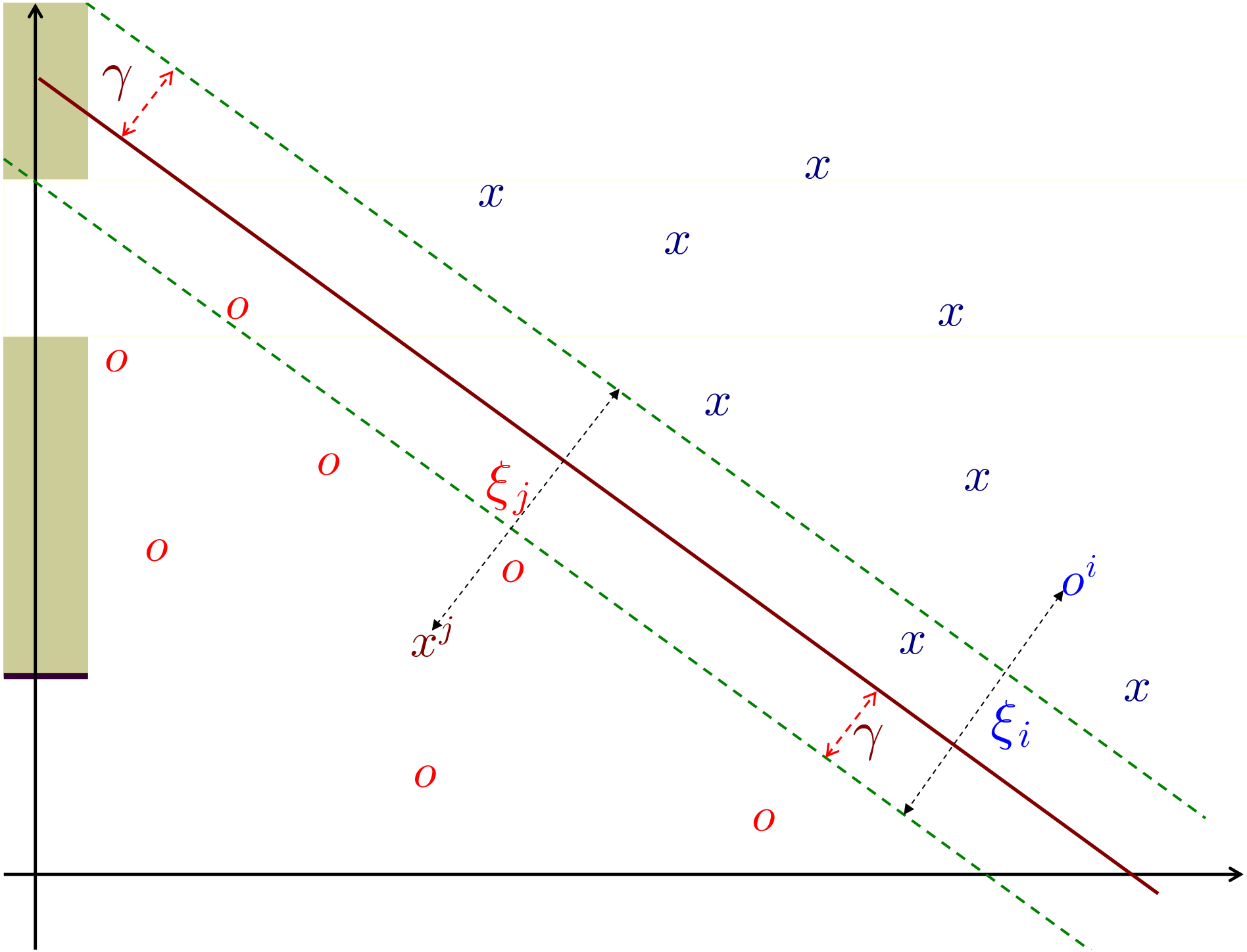
- If data are not linearly separable
  - Primal problem is infeasible
  - Dual problem is unbounded above
- Introduce the slack variable for each training point

$$y_i(w'x^i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall \ i$$

- The inequality system is always feasible

e.g.

$$w = \mathbf{0}, \ \ b = 0 \ \ \& \ \ \xi = e$$


γ
x
x
x
x
o
o
x
o
ξ_j
x
o
o
o^i
x^j
x
x
γ
ξ_i
o
o

Robust Linear Programming

Preliminary Approach to SVM

$$\min_{w,b,\xi} \quad e'\xi$$

$$\text{s.t.} \quad D(Aw + eb) + \xi > e \qquad \text{(LP)}$$

$$\xi > 0$$

where ξ : nonnegative slack (*error*) vector

- The term $e'\xi$, 1-norm measure of *error* vector, is called the *training error*.
- For the linearly separable case, at solution of (LP):

$$\xi = 0$$

Support Vector Machine Formulations

(Two Different Measures of Training Error)

2-Norm Soft Margin:

$$\min_{(w,b,\xi)\in R^{n+1+l}} \quad \frac{1}{2}||w||_2^2 + \frac{C}{2}||\xi||_2^2$$

$$D(Aw + eb) + \xi > e$$

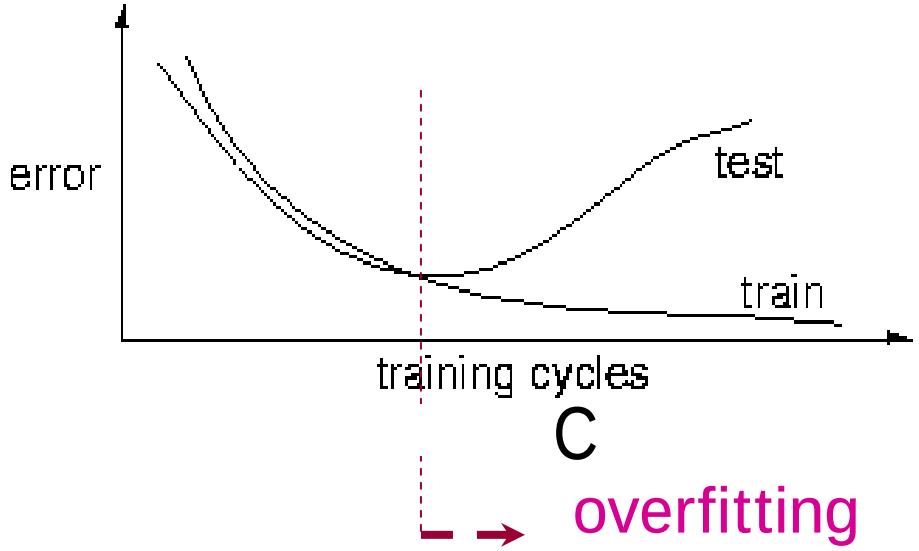
1-Norm Soft Margin (Conventional SVM):

$$\min_{(w,b,\xi)\in R^{n+1+l}} \quad \frac{1}{2}||w||_2^2 + Ce'\xi$$

$$D(Aw + eb) + \xi > e$$

$$\xi > 0$$

Tuning Procedure
How to determine C?



The final value of parameter is one with the maximum testing set correctness !

Lagrangian Dual Problem

$$\max_{\alpha,\beta} \min_{x \in \Omega} \mathcal{L}(x, \alpha, \beta)$$

subject to $\alpha \geqslant \mathbf{0}$

$$\max_{\alpha,\beta} \theta(\alpha, \beta)$$

subject to $\alpha \geqslant \mathbf{0}$

where $\theta(\alpha, \beta) = \inf_{x \in \Omega} \mathcal{L}(x, \alpha, \beta)$

1-Norm Soft Margin SVM
Dual Formulation

The Lagrangian for 1-norm soft margin:

$$\mathcal{L}(w, b, \xi, \alpha, r) = \tfrac{1}{2} w'w + Ce'\xi + \alpha'[e - D(Aw + eb) - \xi] - r'\xi$$

where $\alpha \geq 0$ & $r \geq 0$

The partial derivatives with respect to primal variables equal zeros

$$\frac{\partial \mathcal{L}(w,b,\xi,\alpha)}{\partial w} = w - A'D\alpha = 0$$

$$\frac{\partial \mathcal{L}(w,b,\xi,\alpha)}{\partial b} = e'D\alpha = 0, \quad \frac{\partial \mathcal{L}(w,b,\xi,\alpha)}{\partial \xi} = C - \alpha - r = 0$$

Substitute: $w = A'D\alpha, \;\; Ce'\xi = (\alpha + r)'\xi$ $e'D\alpha = 0,$ in $\mathcal{L}(w, b, \xi, \alpha, r)$

$$\mathcal{L}(w, b, \xi, \alpha, r) = \tfrac{1}{2}w'w + Ce'\xi +$$

$$\alpha'[e - D(Aw + eb) - \xi] - r'\xi$$

where $\alpha > 0 \;\; \& \;\; r > 0$

$$\theta(\alpha, r) = \tfrac{1}{2}\alpha'DAA'D\alpha + e'\alpha - \alpha'DA(A'D\alpha)$$

$$= -\tfrac{1}{2}\alpha'DAA'D\alpha + e'\alpha$$

$s.t. \;\; e'D\alpha = 0, \;\; \alpha - r = Ce$ and $\alpha > 0 \;\; \& \;\; r > 0$

Dual Maximization Problem for 1-Norm Soft Margin

Dual:

$$\max_{\alpha \in R^l} \quad e'\alpha - \tfrac{1}{2}\alpha' DAA'D\alpha$$

$$e'D\alpha = 0$$

$$0 \leq \alpha \leq Ce$$

- The corresponding KKT complementarity:

$$0 \leq \alpha \perp D(Aw + eb) + \xi - e \geq 0$$

$$0 \leq \xi \perp \alpha - Ce \leq 0$$

Slack Variables for 1-Norm Soft Margin SVM $f(x) = \sum_{\alpha_i^* > 0} y_i \alpha_i^* \langle x^i, x \rangle + b^*$

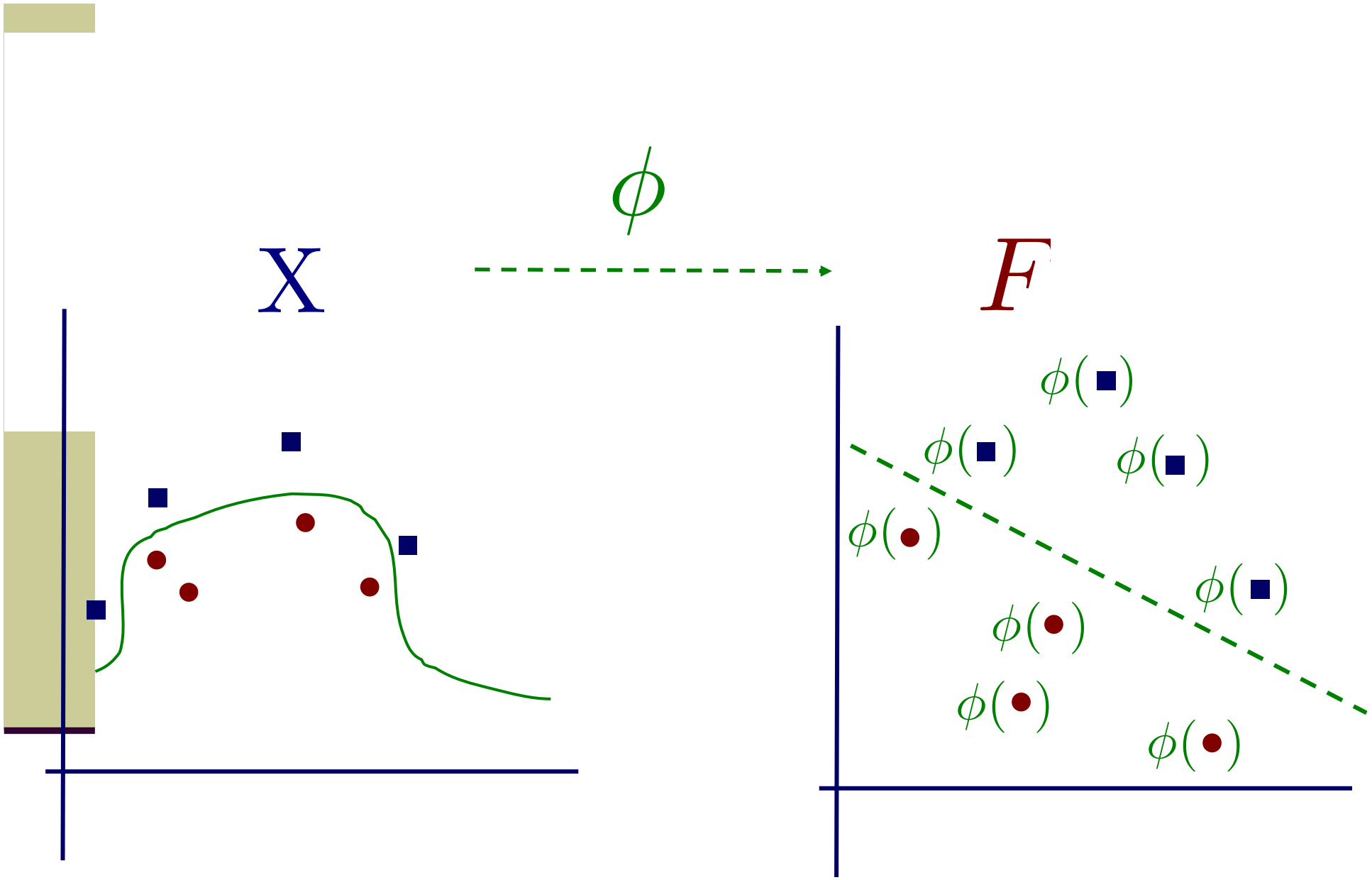
- Non-zero slack can only occur when $\alpha_i^* = C$
  - The contribution of outlier in the decision rule will be at most C
  - The trade-off between accuracy and regularization directly controls by C
- The points for which $0 < \alpha_i^* < C$ lie at the bounding planes
  - This will help us to find b^*

Two-spiral Dataset
(94 White Dots & 94 Red Dots)

Learning in Feature Space

(Could Simplify the Classification Task)

- Learning in a high dimensional space could degrade generalization performance
  - This phenomenon is called *curse of dimensionality*
- By using a *kernel function*, that represents the inner product of training example in feature space, we never need to explicitly know the nonlinear map.
  - Even do not know the dimensionality of feature space
- There is no free lunch
  - Deal with a huge and dense kernel matrix
    - Reduced kernel can avoid this difficulty


ϕ
X
F
$\phi(\blacksquare)$
$\phi(\blacksquare)$
$\phi(\blacksquare)$
$\phi(\bullet)$
$\phi(\blacksquare)$
$\phi(\bullet)$
$\phi(\bullet)$
$\phi(\bullet)$

Linear Machine in Feature Space

Let $\phi : X \rightarrow F$ be a nonlinear map from the input space to some feature space

The classifier will be in the form (*Primal*):

$$f(x) = \left(\sum_{j=1}^{?} w_j \phi_j(x) \right) + b$$

Make it in the *dual* form:

$$f(x) = \left(\sum_{i=1}^{l} \alpha_i y_i \langle \phi(x^i) \cdot \phi(x) \rangle \right) + b$$

Kernel: Represent Inner Product in Feature Space

Definition: A kernel is a function $K: \ X \times X \rightarrow R$ such that $for \ \ all \ \ x, z \in X$

$$K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle$$

where $\phi : X \rightarrow F$

The classifier will become:

$$f(x) = \left(\sum_{i=1}^{l} \alpha_i y_i K(x^i, x) \right) + b$$

A Simple Example of Kernel

Polynomial Kernel of Degree 2: $K(x,z) = \langle x, z \rangle^2$

Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in R^2$ and the nonlinear map

$\phi : R^2 \mapsto R^3$ defined by $\phi(x) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \end{bmatrix}$

Then $\langle \phi(x), \phi(z) \rangle = \langle x, z \rangle^2 = K(x,z)$

- There are many other nonlinear maps, $\psi(x)$, that satisfy the relation: $\langle \psi(x), \psi(z) \rangle = \langle x, z \rangle^2 = K(x,z)$

Power of the Kernel Technique

Consider a nonlinear map $\phi : R^n \mapsto R^p$ that consists of distinct features of all the *monomials* of degree d.

Then $p = \binom{n+d-1}{d}$.

$x_1^3 x_2^1 x_3^4 x_4^4 \Rightarrow \times\circ\circ\circ\times\circ\times\circ\circ\circ\circ\times\circ\circ\circ\circ$

For example: $n = 11,\ \ d = 10,\ \ p = 92378$

- Is it necessary? We only need to know $\langle \phi(x), \phi(z) \rangle$!
- This can be achieved $K(x, z) = \langle x, z \rangle^d$

Kernel Technique

Based on Mercer's Condition (1909)

- The value of kernel function represents the inner product of two training points in feature space

- Kernel functions merge two steps

  1. map input data from input space to feature space (might be infinite dim.)
  2. do inner product in the feature space

More Examples of Kernel

$$K(A, B) : R^{m\times n} \times R^{n\times l} \longmapsto R^{m\times l}$$

$A \in R^{m\times n}, a \in R^m, \mu \in R,$ d is an integer:

- Polynomial Kernel: $(AA' + \mu aa')^d_\bullet$

  (Linear Kernel AA': $\mu = 0, d = 1$)

- Gaussian (Radial Basis) Kernel:

$$K(A, A')_{ij} = \varepsilon^{-\mu\|A_i - A_j\|_2^2}, \quad i, j = 1, \ldots, m$$

  - The ij-entry of $K(A, A')$ represents the "similarity" of data points A_i and A_j

Nonlinear 1-Norm Soft Margin SVM In Dual Form

Linear SVM:

$$\max_{\alpha \in R^l} \quad e'\alpha - \tfrac{1}{2}\alpha' DAA'D\alpha$$

$$e'D\alpha = 0$$

$$0 \leq \alpha \leq Ce$$

Nonlinear SVM:

$$\max_{\alpha \in R^l} \quad e'\alpha - \tfrac{1}{2}\alpha' DK(A, A')D\alpha$$

$$e'D\alpha = 0$$

$$0 \leq \alpha \leq Ce$$

SVM as an Unconstrained Minimization Problem

$$\min_{w,b} \frac{C}{2}\|\xi\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2) \quad \text{(QP)}$$
$$\text{s. t.} \quad D(Aw + eb) + \xi > e$$

At the solution of (QP): $\xi = (e - D(Aw + eb))_+$

where $(\cdot)_+ = \max\{\cdot, 0\}$

Hence (QP) is equivalent to the nonsmooth SVM:

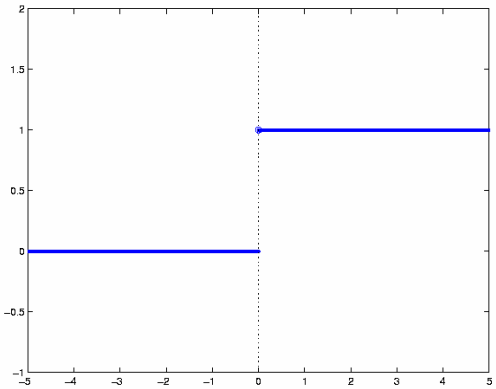
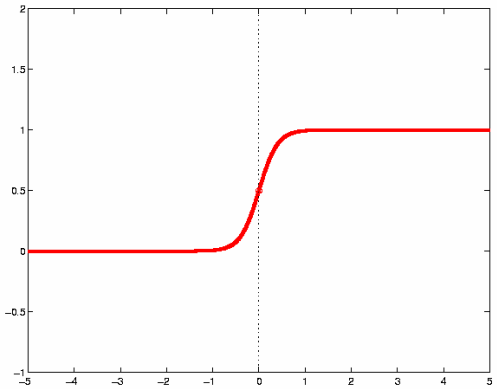
$$\min_{w,b} \frac{C}{2}\|(e - D(Aw + eb))_+\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2)$$

- Change (QP) into an unconstrained MP
- Reduce $(n+1+m)$ variables to $(n+1)$ variables

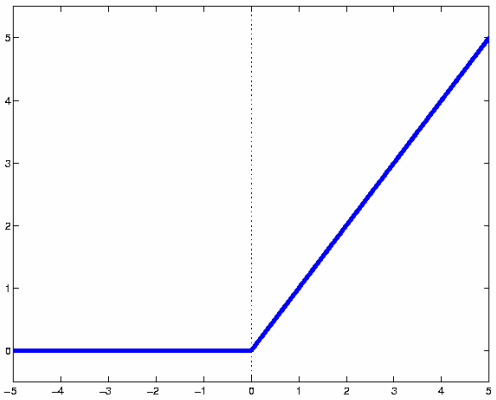
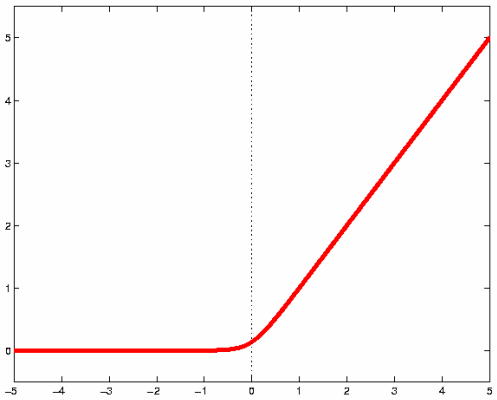
Smooth the Plus Function: Integrate $\frac{1}{(1+\varepsilon^{-\beta x})}$

$$p(x,\beta) := x + \frac{1}{\beta}\log(1+\varepsilon^{-\beta x})$$

Step function: x_* Sigmoid function: $\frac{1}{(1+\varepsilon^{-5x})}$

Plus function: x_+ p-function: $p(x,5)$

SSVM: Smooth Support Vector Machine

- Replacing the plus function $(\cdot)_+$ in the nonsmooth SVM by the smooth $p(\cdot\,, \beta)$, gives our SSVM:

$$\min_{(w,b) \in R^{n+1}} \frac{C}{2} \| p((e - D(Aw + eb)), \beta) \|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2)$$

- The solution of SSVM converges to the solution of nonsmooth SVM as β goes to infinity.

  (Typically, $\beta = 5$)

Newton-Armijo Method: Quadratic Approximation of SSVM

- The sequence $\{(w^i, b_i)\}$ generated by solving a quadratic approximation of SSVM, converges to the unique solution (w^*, b^*) of SSVM at a quadratic rate.
  - Converges in 6 to 8 iterations
- At each iteration we solve a linear system of:
  - n+1 equations in n+1 variables
  - Complexity depends on dimension of input space
- It might be needed to select a stepsize

Newton-Armijo Algorithm

$$\Phi_\beta(w,b) = \frac{C}{2}\|p((e - D(Aw + eb)), \beta)\|_2^2 + \frac{1}{2}(\|w\|_2^2 + b^2)$$

Start with any $(w^0, b_0) \in R^{n+1}$. Having (w^i, b_i),

stop if $\nabla\Phi_\beta(w^i, b_i) = 0$, else :

(i) Newton Direction :

$$\nabla^2\Phi_\beta(w^i, b_i)d^i = -\nabla\Phi_\beta(w^i, b_i)'$$

(ii) Armijo Stepsize :

$$(w^{i+1}, b_{i+1}) = (w^i, b_i) + \lambda_i d^i$$

$\lambda_i \in \{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$

such that Armijo's rule is satisfied

globally and quadratically converge to unique solution in a finite number of steps

Nonlinear Smooth SVM

Nonlinear Classifier: $K(x', A')D\alpha + b = 0$

- Replace AA' by a nonlinear kernel $K(A, A')$:

$$\min_{\alpha, b} \frac{C}{2}\|p(e - D(K(A, A')D\alpha + eb, \beta)\|_2^2 + \frac{1}{2}(\|\alpha\|_2^2 + b^2)$$

- Use Newton-Armijo algorithm to solve the problem
  - Each iteration solves m+1 linear equations in m+1 variables

- Nonlinear classifier depends on the data points with nonzero coefficients :

$$K(x', A')D\alpha + b = \sum_{\alpha_j > 0} \alpha_j y_j K(A_j, x) + b = 0$$

Conclusion

- An overview of SVMs for classification
- SSVM: A new formulation of support vector machine as a smooth unconstrained minimization problem
  - Can be solved by a fast Newton-Armijo algorithm
  - No optimization (LP, QP) package is needed
- There are many important issues did not address this lecture such as:
  - How to solve conventional SVM?
  - How to select parameters: C & μ
  - How to deal with massive datasets?