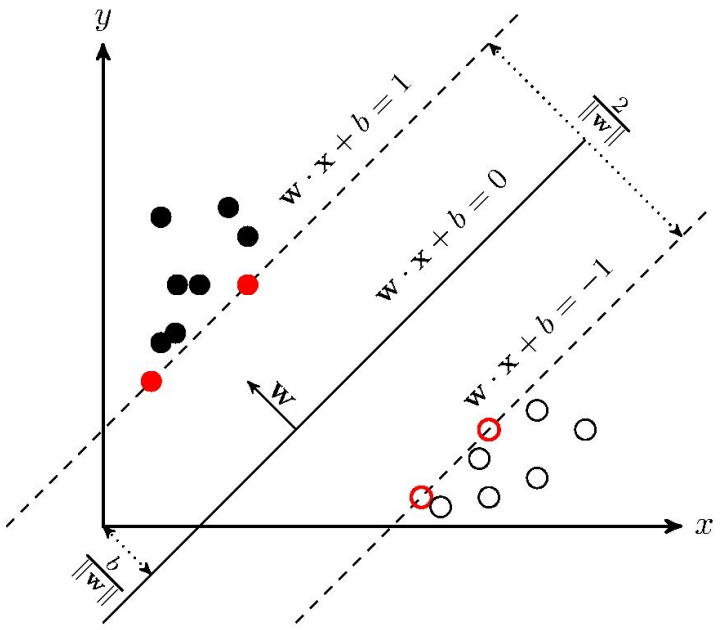
Support Vector Machine

Yanyan Lan

lanyanyan@ict.ac.cn

支持向量机

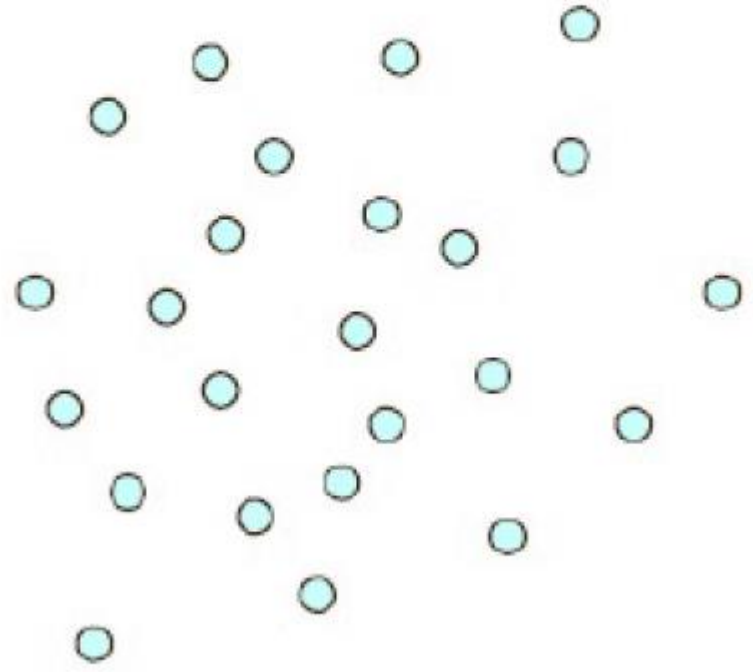
- [1995] V. Vapnik和C. Cortes两人发明了SVM， 它是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，即支持向量机的学习策略便是**间隔最大化**，最终可转化为一个**凸二次规划**问题的求解。

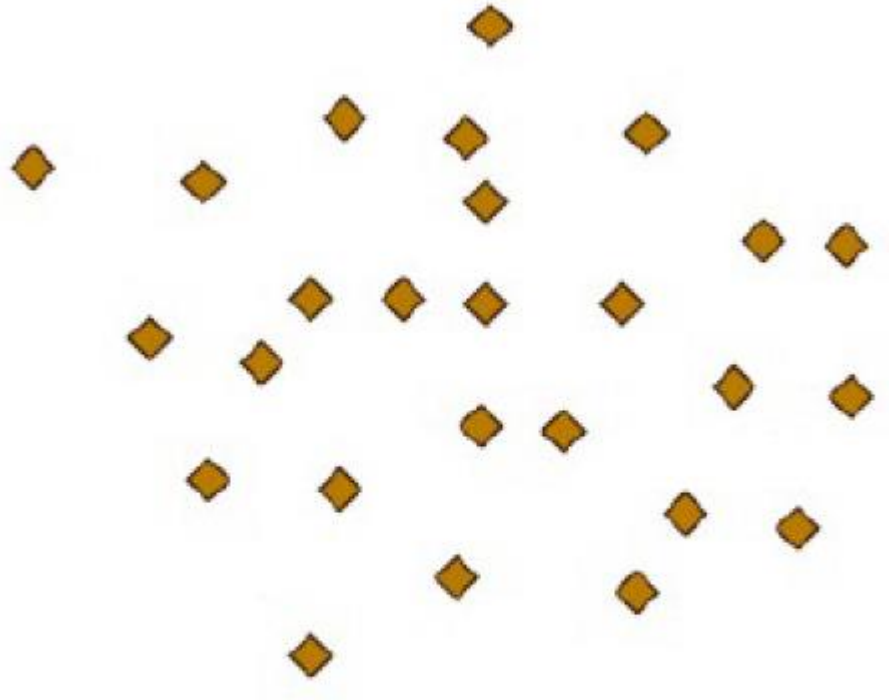


V. Vapnik

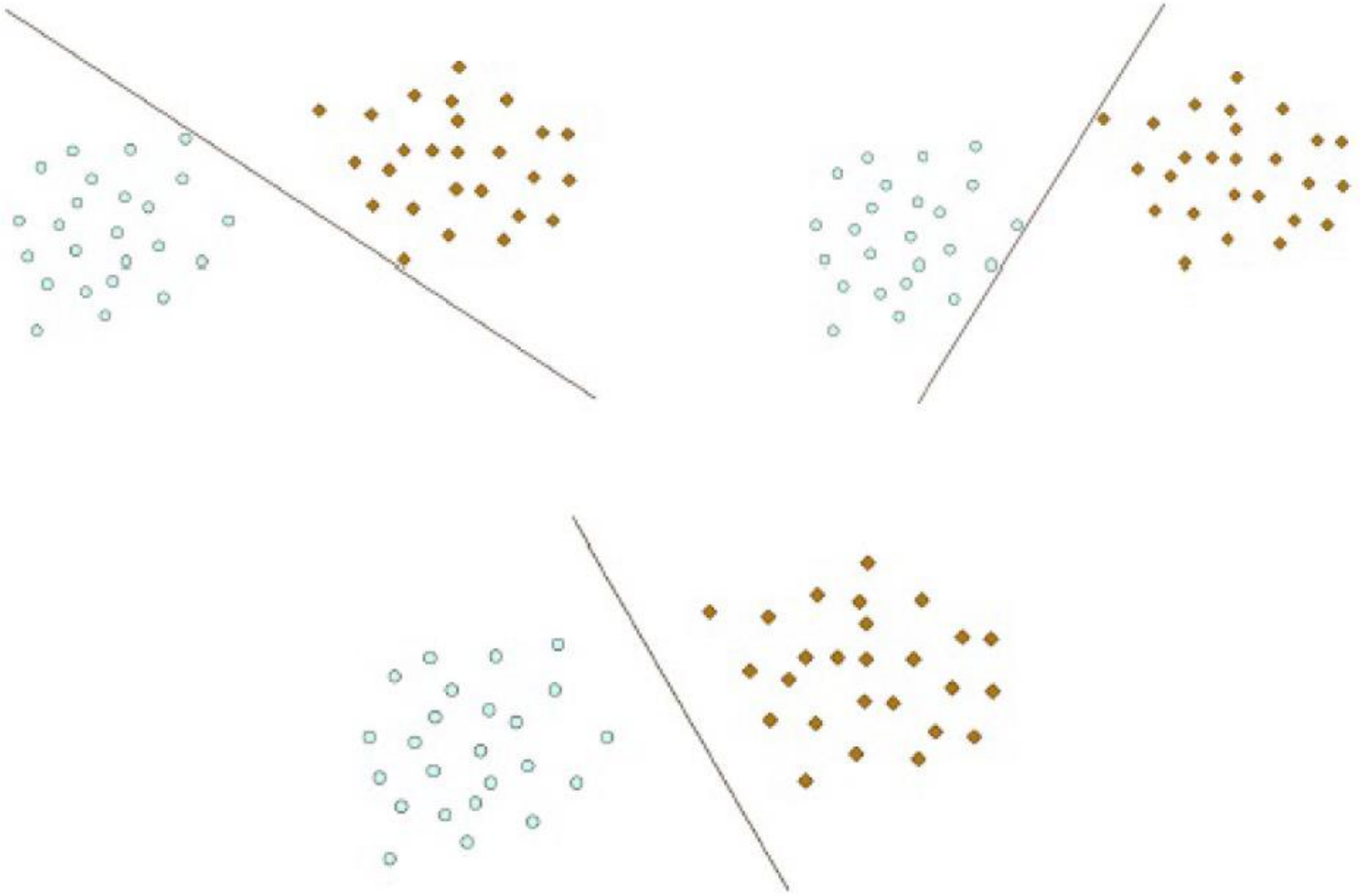
C. Cortes

Given a Data Set …



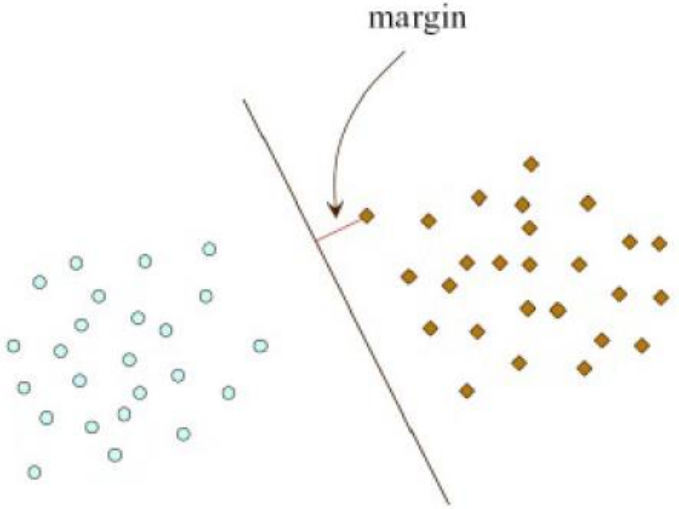


Which Separating Hyperplane is the Best?



Optimal Separating Hyperplane

- Margin of a separating hyperplane: distance to the separating hyperplane from the data point closest to it.



- Relationship between margin and generalization: There exist theoretical results from statistical learning theory showing that the separating hyperplane with the largest margin generalizes best (i.e., has smallest generalization error).

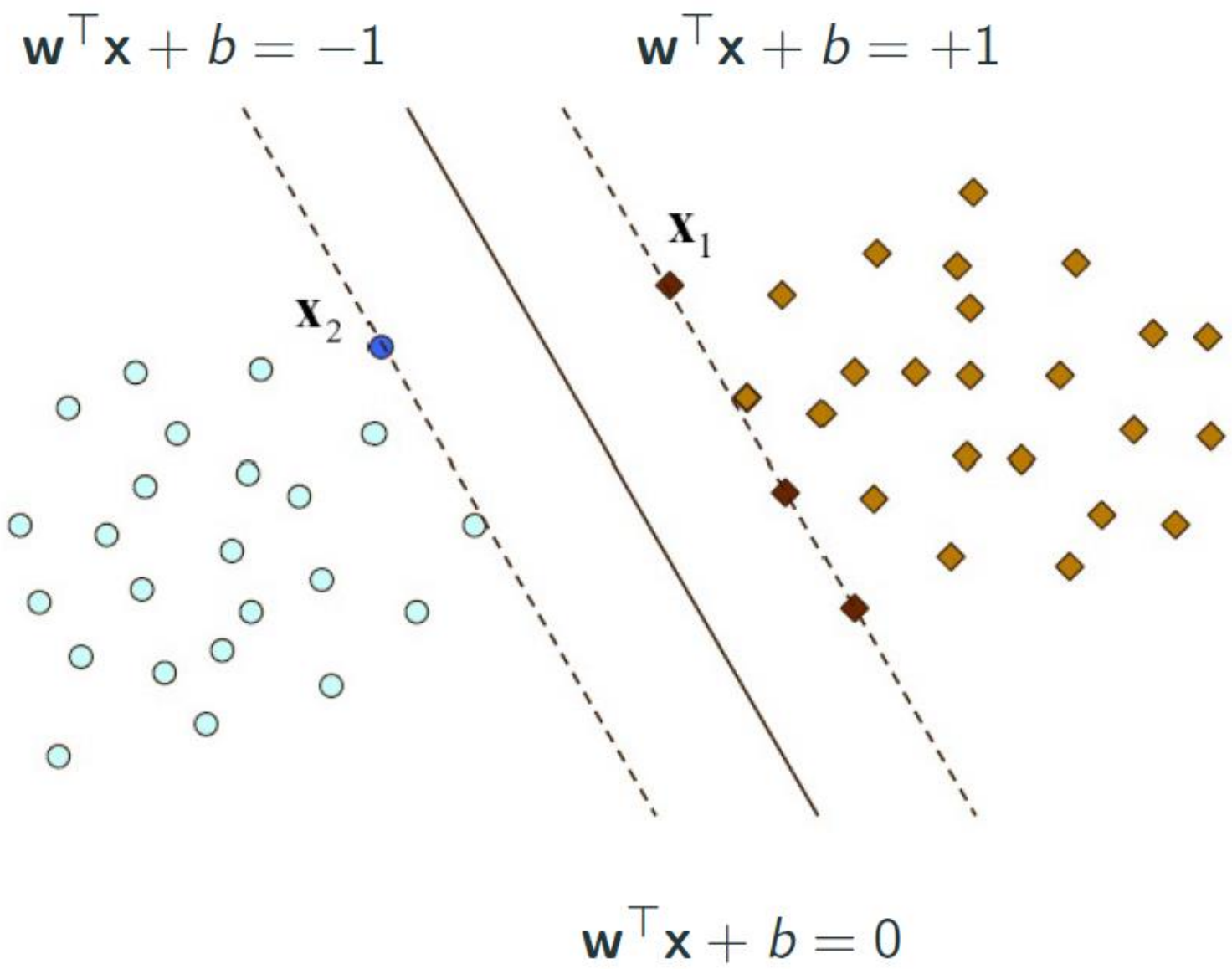
Canonical Optimal Separating Hyperplane

- Hard-margin case: data points from the two classes are assumed to be linearly separable.
- Let $\boldsymbol{w}$ denote a vector orthogonal to the decision boundary, and b denote a scalar "offset" term, then we can write the decision boundary as:

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0$$

- For $\lambda \neq 0, (\lambda\boldsymbol{w}, \lambda b)$ describes the same hyperplane as $(\boldsymbol{w}, b)$, i.e., $\{\boldsymbol{x}|\boldsymbol{w}^T\boldsymbol{x} + b = 0\} = \{\boldsymbol{x}|\lambda(\boldsymbol{w}^T\boldsymbol{x} + b) = 0\}$
- With proper scaling of $\boldsymbol{w}$ and b, the points closest to the hyperplane satisfy $|\boldsymbol{w}^T\boldsymbol{x} + b| = 1$. Such a hyperplane is called a canonical separating hyperplane.
- The one that maximizes the margin is called the canonical optimal separating hyperplane.

Canonical Optimal Separating Hyperplane



- A Canonical Optimal Separating Hyperplane case.
  - Points closest to the hyperplane are in dark color.
  - $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two closest points from each side, which satisfy $|\boldsymbol{w}^T\boldsymbol{x} + b| = 1$
- How can we calculate the margin?

Canonical Optimal Separating Hyperplane

- Let $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ be two closest points, one on each side of the hyperplane.
- Note that

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

Which imply

$$\boldsymbol{w}^T\left(\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)}\right) = 2$$

By projecting vector $\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)}$ on direction of $\boldsymbol{w}$, we can get the margin:

$$margin = \frac{1}{2}\frac{\boldsymbol{w}^T\left(\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)}\right)}{||\boldsymbol{w}||} = \frac{1}{||\boldsymbol{w}||}$$

- Maximizing the margin is equivalent to minimizing $||\boldsymbol{w}||$

Inequality Constraints

- For all data points in the sample set $D = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$, their distance to the hyperplane should not be less than the margin, which requires $\boldsymbol{w}$ and b to satisfy

$$\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b \begin{cases} \geq +1 \;\; if \; y^{(i)} = +1 \\ \leq -1 \;\; if \; y^{(i)} = -1 \end{cases}$$

- Equivalent form of inequality constraints:

$$y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right) \geq 1$$

Primal Optimization Problem

- Primal optimization problem:

$$\min \frac{1}{2}||\boldsymbol{w}||^2$$
$$s.t.\, y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right) \geq 1$$

- A quadratic programming (QP) problem
- In optimization theory, it is very common to turn a primal problem into a dual problem and then solve the latter instead.
- In our case, it also turns out to be more convenient to solve the dual problem (whose complexity depends on the sample size n) rather than the primal problem directly (whose complexity depends on the dimensionality d). The dual problem also makes it easy for a nonlinear extension using kernel functions.

Example

- 已知训练数据集包含三个数据点
- 其中
  - 正例点是 x1=(3,3) x2=(4,3)
  - 负例点是 x3=(1,1)
- 试求最大间隔分离超平面

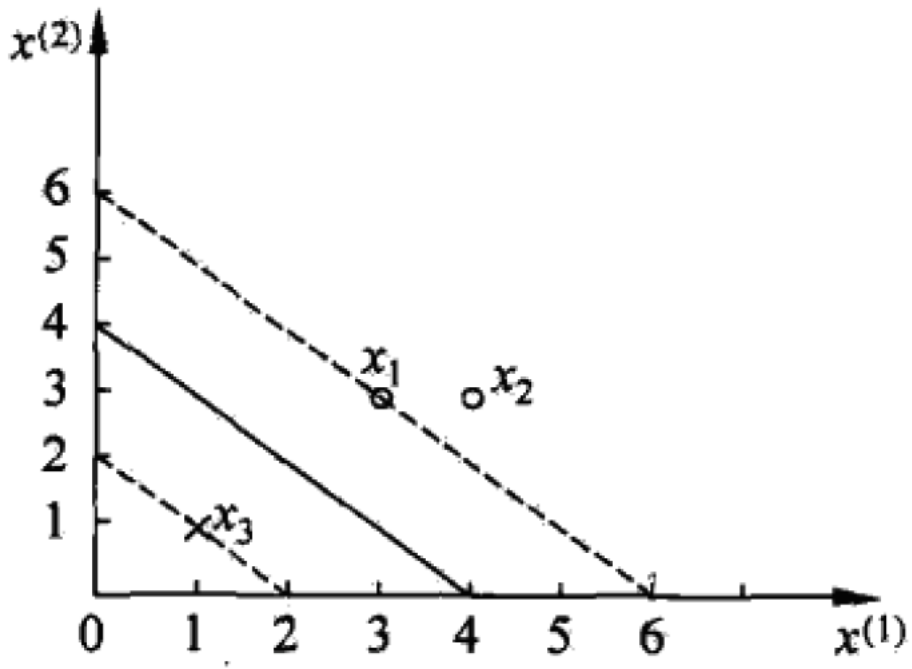


图 7.4 间隔最大分离超平面示例

根据训练数据集构造的约束最优化问题：

$$\min_{w,b} \quad \frac{1}{2}(w_1^2 + w_2^2)$$

$$\text{s.t.} \quad 3w_1 + 3w_2 + b \geqslant 1$$

$$4w_1 + 3w_2 + b \geqslant 1$$

$$-w_1 - w_2 - b \geqslant 1$$

由此最优化问题解得 $w_1 = w_2 = \frac{1}{2}$， $b = -2$.

因此最大间隔分离超平面为： $\frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)} - 2 = 0$

支持向量为： $x_1 = (3,3)^{\mathrm{T}}$ 与 $x_3 = (1,1)^{\mathrm{T}}$

Digression to Lagrangian Duality

- The Primal Problem:

$$\min_{w} f(w)$$
$$s.t.\ g_i(w) \leq 0, i = 1, \dots, k$$
$$h_i(w) = 0, i = 1, \dots, l$$

- The generalized Lagrangian:

$$L_p(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

where $\alpha_i \geq 0$ and β_i are called the lagrange multipliers.

- A re-written Primal:

$$\min_{w} \max_{\alpha, \beta, \alpha_i \geq 0} L_p(w, \alpha, \beta)$$

Note this is equivalent to original Primal

Primal to Dual

- The Primal Problem:

$$\min_{w} \max_{\alpha,\beta,\alpha_i \geq 0} L_p(w, \alpha, \beta)$$

$$L_p(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

- By switching the order of min and max operation, we get the Dual Problem:

$$\max_{\alpha,\beta,\alpha_i \geq 0} \min_{w} L_p(w, \alpha, \beta)$$

- Under certain conditions, solving the Dual Problem is equivalent to solving the Primal Problem

Strong Dual Theorem

Strong Dual Theorem:

For a Primal Problem:

$$\min_{w} f(w)$$
$$s.t.\ g_i(w) \leq 0, i = 1, \dots, k$$
$$h_i(w) = 0, i = 1, \dots, l$$

If following conditions are met, there exists w^*, α^*, β^*, where w^* is optimal solution to Primal and α^*, β^* are optimal solutions to Dual

- $f(w)$ and $g_i(w)$ are convex for w
- $h_i(w)$ is affine ($h(w) = aw + b$)
- $\exists w_0, \forall i, g_i(w_0) < 0$

The KKT Conditions

- Under the conditions of Strong Dual Theorem, solving the Dual Problem is equivalent to solving the Primal Problem, solutions of Primal and Dual problems satisfy the following "Karush-Kuhn-Tucker" (KKT) conditions
  - $\partial L_p(w, \alpha, \beta) = 0$ (stationarity)
  - $\alpha_i g_i(w) = 0$, for all i (complementary slackness)
  - $g_i(w) \leq 0$, for all i (primal feasibility)
  - $h_i(w) = 0$, for all i (primal feasibility)
  - $\alpha_i \geq 0$, for all i (dual feasibility)

- Also, if w^*, α^* and β^* satisfy the KKT conditions, then it is also a solution to the primal and the dual problems

Lagrangian for SVM

- Lagrangian for SVM Primal Problem: (α_i is Lagrange multiplier)

$$\min \frac{1}{2}||\boldsymbol{w}||^2$$
$$s.t.\, y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right) \geq 1$$
$$\Downarrow$$
$$\min_{\boldsymbol{w},b} \max_{\alpha} L_p(\boldsymbol{w}, b, \alpha)$$
$$L_p(\boldsymbol{w}, b, \alpha) = \frac{1}{2}||\boldsymbol{w}||^2 + \sum_{i=1}^{n} \alpha_i\left[1 - y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right)\right]$$

- The inequality constraints of the primal problem are incorporated into the second term of the Lagrangian.
- Conditions for Strong Dual Theorem are met here, we can solve its dual instead.

Dual Problem for SVM

- Primal Problem

$$\min_{\boldsymbol{w},b} \max_{\alpha} L_p(\boldsymbol{w}, b, \alpha)$$

- Dual Problem

$$\max_{\alpha} \min_{\boldsymbol{w},b} L_p(\boldsymbol{w}, b, \alpha) = \max_{\alpha} L_d(\alpha)$$

$$L_d(\alpha) \triangleq \min_{\boldsymbol{w},b} L_p(\boldsymbol{w}, b, \alpha)$$

- Next, we will eliminate Primal variables $\boldsymbol{w}, b$ by solving the inner min operation and get the final form of the Dual Problem.

$$L_d(\alpha) = \min_{\boldsymbol{w},b} L_p(\boldsymbol{w}, b, \alpha) = ?$$

Eliminating Primal Variables

$$L_p(\boldsymbol{w}, b, \alpha) = \frac{1}{2}||\boldsymbol{w}||^2 + \sum_{i=1}^{n} \alpha_i \left[1 - y^{(i)}\left(\boldsymbol{w}^T \boldsymbol{x}^{(i)} + b\right)\right]$$

- Setting the gradients of L_p w.r.t. $\boldsymbol{w}$ and b to 0:

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)}$$

$$\frac{\partial L_p}{\partial \mathbf{b}} = 0 \Longrightarrow \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

- Plugging above equations into L_p gives the objective function L_d for the dual problem:

$$L_d = -\frac{1}{2}\mathbf{w}^\top \mathbf{w} + \sum_i \alpha_i$$

$$= -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)} + \sum_i \alpha_i$$

Dual Optimization Problem

- Final form of Dual Problem

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y^{(i)}y^{(j)}(\mathbf{x}^{(i)})^\top\mathbf{x}^{(j)} + \sum_{i}\alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y^{(i)} = 0 \text{ and } \forall i, \alpha_i \geq 0$$

- This is also a QP problem, but its complexity depends on the sample size n (rather than the input dimensionality d):
  - Time complexity: $\mathcal{O}(n^3)$
  - Space complexity: $\mathcal{O}(n^2)$
- SMO, proposed by John Platt in 1998, gives an efficient way of solving the dual problem arising from the derivation of the SVM

Support Vectors

- Most of the dual variables vanish with $\alpha_i = 0$.
- From complementary slackness condition in KKT

$$\alpha_i[1 - y^{(i)}(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b)] = 0$$

  we know for all i that $\alpha_i > 0$, there is $1 - y^{(i)}(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b) = 0$, which means such points lie on the margin.
- The points lying beyond the margin have no effect on the hyperplane

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \boldsymbol{x}^{(i)} = \sum_{\alpha_i > 0} \alpha_i y^{(i)} \boldsymbol{x}^{(i)}$$

- Support vectors: $\boldsymbol{x}^{(i)}$ with $\alpha_i > 0$, hence the name Support Vector Machine (SVM).

Compute w and b

- Computation of primal variables:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^{(i)} \mathbf{x}^{(i)} = \sum_{\mathbf{x}^{(i)} \in \mathcal{SV}} \alpha_i y^{(i)} \mathbf{x}^{(i)}$$

where $\mathcal{SV}$ denotes the set of support vectors.

- The support vectors must lie on the margin, so they should satisfy

$$y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) = 1 \text{ or } b = y^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)}$$

For numerical stability, all support vectors are used to compute b:

$$b = \frac{1}{|\mathcal{SV}|} \sum_{\mathbf{x}^{(i)} \in \mathcal{SV}} (y^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)})$$

Discriminant Function

- Discriminant function:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$
$$= \Big(\sum_{\mathbf{x}^{(i)} \in \mathcal{SV}} \alpha_i y^{(i)} \mathbf{x}^{(i)} \Big)^\top \mathbf{x} + b$$

- Classification rule during testing:

$$\text{Choose} \begin{cases} C_1 & \text{if } f(\mathbf{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}$$

Example

- 已知训练数据集包含三个数据点
- 其中
  - 正例点是 x1=(3,3) x2=(4,3)
  - 负例点是 x3=(1,1)
- 试求最大间隔分离超平面

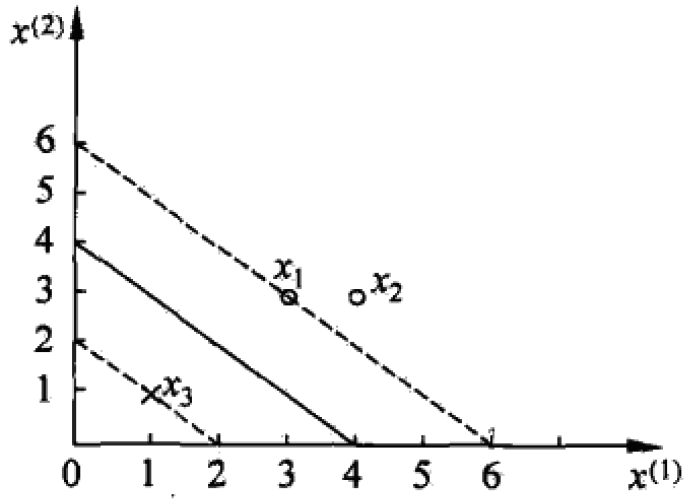


图 7.4 间隔最大分离超平面示例

根据训练数据集构造的约束最优化问题的对偶问题是：

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$= \frac{1}{2}(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$\text{s.t.} \quad \alpha_1 + \alpha_2 - \alpha_3 = 0$$

$$\alpha_i \geqslant 0, \quad i = 1,2,3$$

Example – Cont.

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$= \frac{1}{2}(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3$$

$$\text{s.t.} \quad \alpha_1 + \alpha_2 - \alpha_3 = 0$$

$$\alpha_i \geqslant 0, \quad i = 1,2,3$$

解这一最优化问题. 将 $\alpha_3 = \alpha_1 + \alpha_2$ 代入目标函数并记为

$$s(\alpha_1, \alpha_2) = 4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$$

对 α_1, α_2 求偏导数并令其为 0，易知 $s(\alpha_1, \alpha_2)$ 在点 $\left(\frac{3}{2}, -1\right)^{\mathrm{T}}$ 取极值，但该点不满足约束条件 $\alpha_2 \geqslant 0$，所以最小值应在边界上达到.

Example – Cont.

当$\alpha_1=0$时，最小值$s\left(0,\frac{2}{13}\right)=-\frac{2}{13}$；当$\alpha_2=0$时，最小值$s\left(\frac{1}{4},0\right)=-\frac{1}{4}$．于是$s(\alpha_1,\alpha_2)$在$\alpha_1=\frac{1}{4},\alpha_2=0$达到最小，此时$\alpha_3=\alpha_1+\alpha_2=\frac{1}{4}$.

这样，$\alpha_1^*=\alpha_3^*=\frac{1}{4}$对应的实例点$x_1,x_3$是支持向量．根据式(7.25)和式(7.26)计算得

$$w_1^*=w_2^*=\frac{1}{2}$$

$$b^*=-2$$

分离超平面为

$$\frac{1}{2}x^{(1)}+\frac{1}{2}x^{(2)}-2=0$$

分类决策函数为

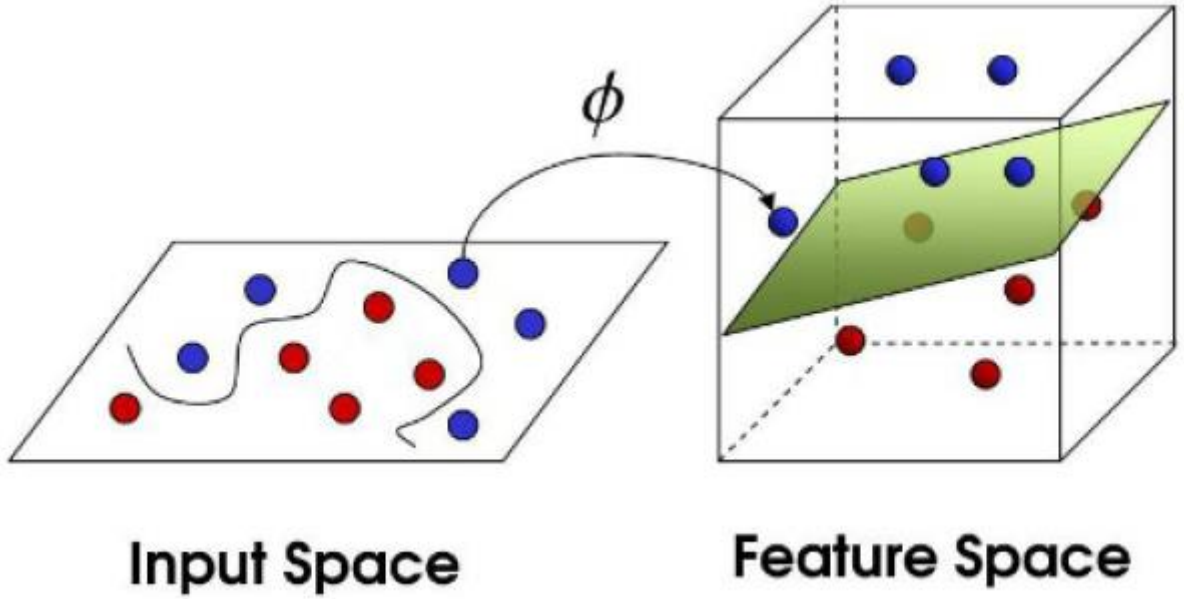
$$f(x)=\text{sign}\left(\frac{1}{2}x^{(1)}+\frac{1}{2}x^{(2)}-2\right)$$

■

Kernel

Key Ideas of Kernel Methods

What can we do if there is no hyperplane that can separate the data?



Mapping the data from the input space to a high dimensional feature space where the data can be separated.

Key Ideas of Kernel Methods

- Instead of defining a nonlinear model in the original (input) space, the problem is mapped to a new (feature) space by performing a nonlinear transformation.
- A linear model is then applied in the new space.
- The nonlinear transformations are often defined implicitly via defining kernel functions directly.

Feature Space

- Primal Optimization Problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y^{(i)}(\mathbf{w}^\top \phi(\mathbf{x}^{(i)}) + b) \geq 1 \end{aligned}$$

- Dual Optimization Problem:

$$\begin{aligned} \max \quad & \sum_i \alpha_{i=1}^N - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(\mathbf{x}^{(i)})^\top \phi(\mathbf{x}^{(j)}) \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y^{(i)} = 0 \\ & \forall i, \alpha_i \geq 0 \end{aligned}$$

- Prediction:

$$f(\mathbf{x}) = \sum_{\mathbf{x}^{(i)} \in \mathcal{SV}} \alpha_i y^{(i)} \phi(\mathbf{x}^{(i)})^\top \phi(\mathbf{x}) + b$$

Kernel methods

- In Dual Problem and Prediction, $\boldsymbol{x}$ appears nowhere else but in a inner-product expression. Thus knowing inner-product of $\boldsymbol{x}$ is sufficient without knowing ϕ. So we can do the following trick:
- Dual Optimization Problem:

$$\max \quad \sum_{i} \alpha_{i=1}^{N} - \frac{1}{2} \sum_{i} \sum_{j} \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(\mathbf{x}^{(i)})^\top \phi(\mathbf{x}^{(j)})$$

$$\phi(\mathbf{x}^{(i)})^\top \phi(\mathbf{x}^{(j)}) \rightarrow K(x^{(i)}, x^{(j)})$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

$$\forall i, \alpha_i \geq 0$$

- Prediction:

$$f(\mathbf{x}) = \sum_{\mathbf{x}^{(i)} \in \mathcal{SV}} \alpha_i y^{(i)} \phi(\mathbf{x}^{(i)})^\top \phi(\mathbf{x}) + b$$

$$\phi(\mathbf{x}^{(i)})^\top \phi(\mathbf{x}) \rightarrow K(x^{(i)}, x)$$

Kernel function

- A Kernel $\mathrm{K}: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a function that maps two points to a real value.
- *e.g.* $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{x}_2$
  or $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = ||\boldsymbol{x}_1 - \boldsymbol{x}_2||$
- Each kernel function corresponds to a transformation(usually nonlinear), but not all corresponding transformations have analytical expression.

Polynomial Kernel

- Polynomial kernel:

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left((\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)} + 1\right)^q$$

where q is the degree

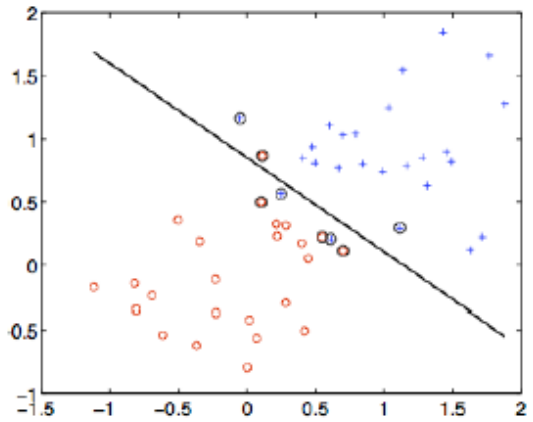
- E.g., $q = 2$ and $D = 2$,

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + 1)^2$$
$$= 1 + 2x_1x_1' + 2x_2x_2' + 2x_1x_2x_1'x_2' + (x_1)^2(x_1')^2 + (x_2)^2(x_2')^2$$

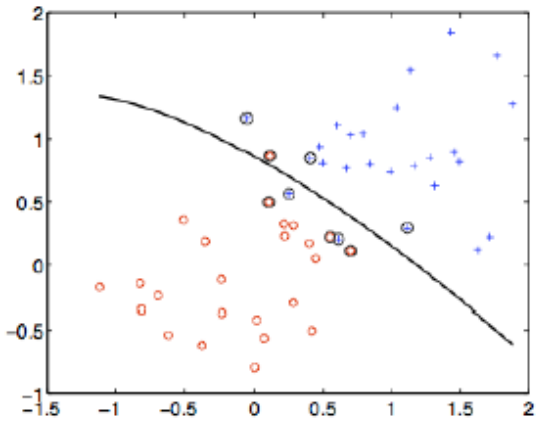
which corresponds to the inner product of the nonlinearly transformed vector

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, (x_1)^2, (x_2)^2)$$

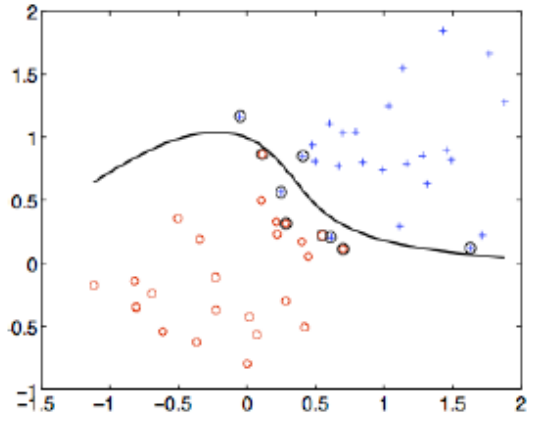
Polynomial Kernels with SVMs



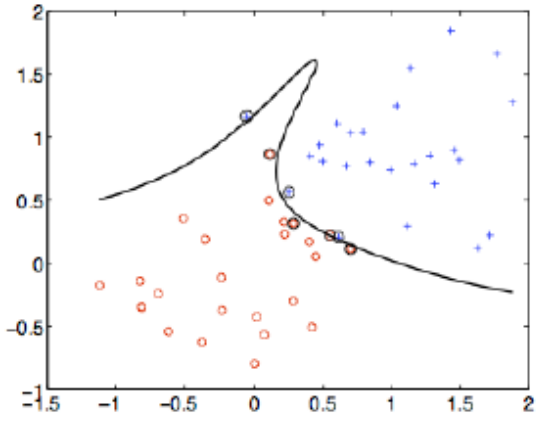
linear



2^{nd} order polynomial



4^{th} order polynomial



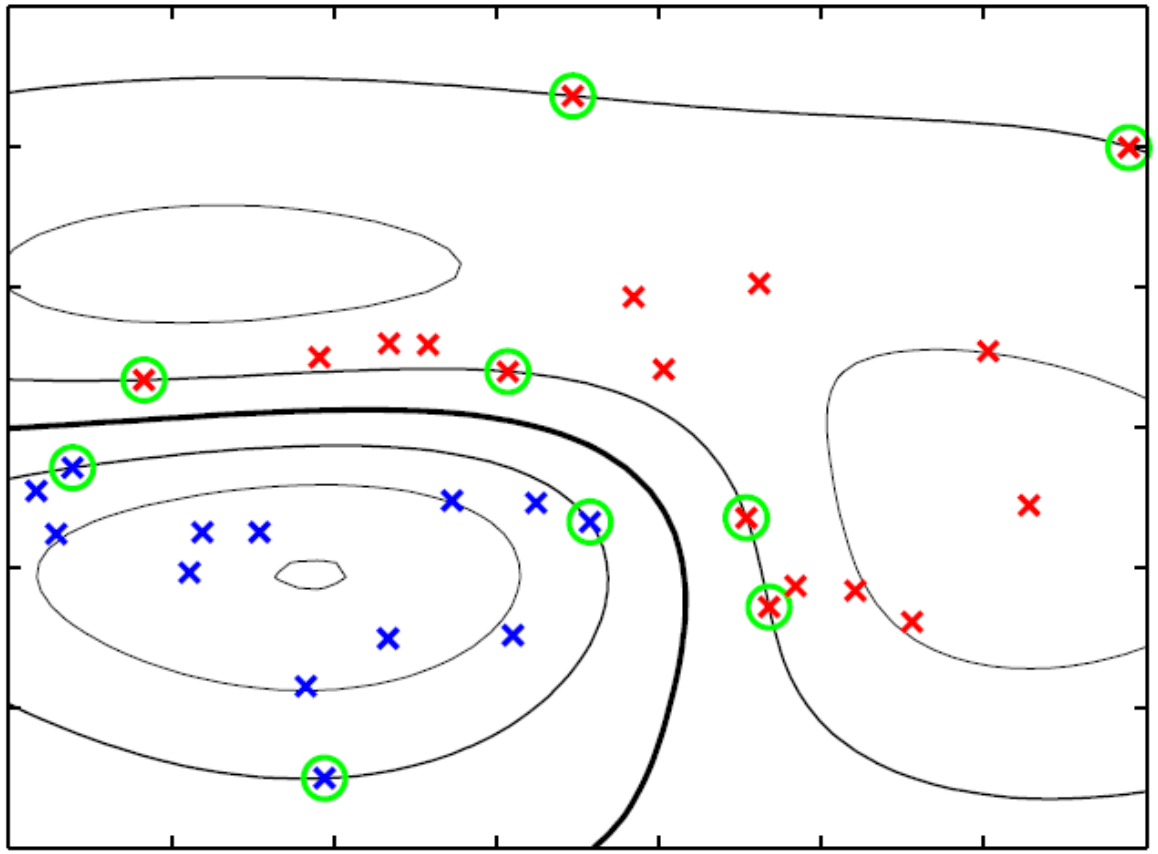
8^{th} order polynomial

From Tommi Jaakkola, MIT CSAIL

Gaussian Kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{2\sigma^2})$$

In this case the feature space is infinite dimensional function space.



Some Common Kernel Functions

Name	Formula
Linear Kernel	$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$
Polynomial kernel	$K(\mathbf{x}, \mathbf{x}') = ((\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)} + c)^q$
Gaussian Kernel	$K(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|}{2\sigma^2})$
Laplace Kernel	$K(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|}{\sigma})$
Sigmoid Kernel	$K(\mathbf{x}, \mathbf{x}') = \tanh(\beta \mathbf{x}^\top \mathbf{x}' + \theta)$

if K_1 and K_2 are kernel functions, then for any $\gamma_1 > 0$, $\gamma_2 > 0$ and function $g(x)$,

All are kernel functions

$$\begin{cases} \gamma_1 K_1 + \gamma_2 K_2 \\ K_1 \otimes K_2(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') K_2(\mathbf{x}, \mathbf{x}') \\ K(\mathbf{x}, \mathbf{x}') = g(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') g(\mathbf{x}') \end{cases}$$

For text data, linear kernel is a common choice.

The essence of kernel

Kernel design, any principle?

- $K(\boldsymbol{x}, \boldsymbol{x}')$ can be thought of as a similarity function between $\boldsymbol{x}$ and $\boldsymbol{x}'$
- This intuition can be well reflected in the following "Gaussian" function (Similarly one can easily come up with other $K(\cdot)$ in the same spirit)

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||}{2\sigma^2})$$

- Is this necessarily lead to a "legal" kernel?

Kernel Matrix

Design kernel function:

$$K(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) \equiv \phi(\boldsymbol{x}^{(i)})^T \phi(\boldsymbol{x}^{(j)})$$

Kernel matrix (a.k.a. Gram matrix):

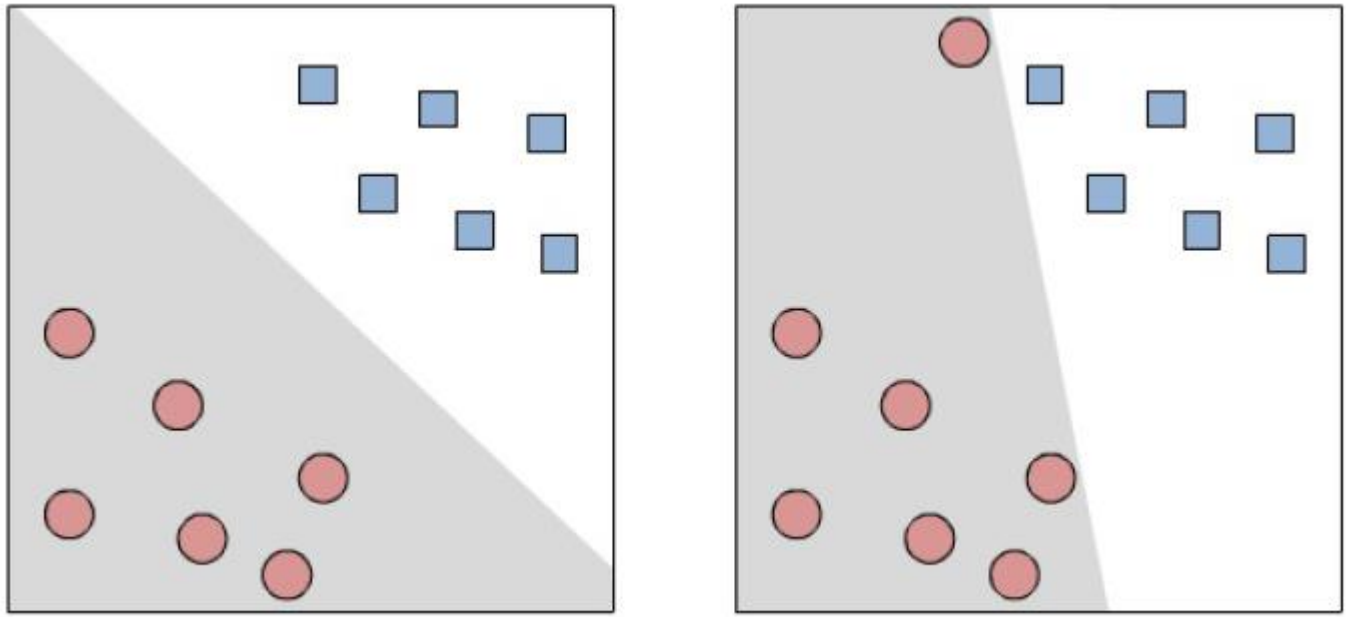
$$\boldsymbol{K} = [K(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})]_{i,j=1}^{N}$$

Mercer kernel

Let K: $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be given. Then for K to be a valid (Mercer) kernel, it is necessary and sufficient that for any $\{x_1, \cdots, x_m\}, (m < \infty)$, the corresponding kernel matrix is symmetric positive semi-definite.

Soft-Margin

The Need of Regularization -- Outliers

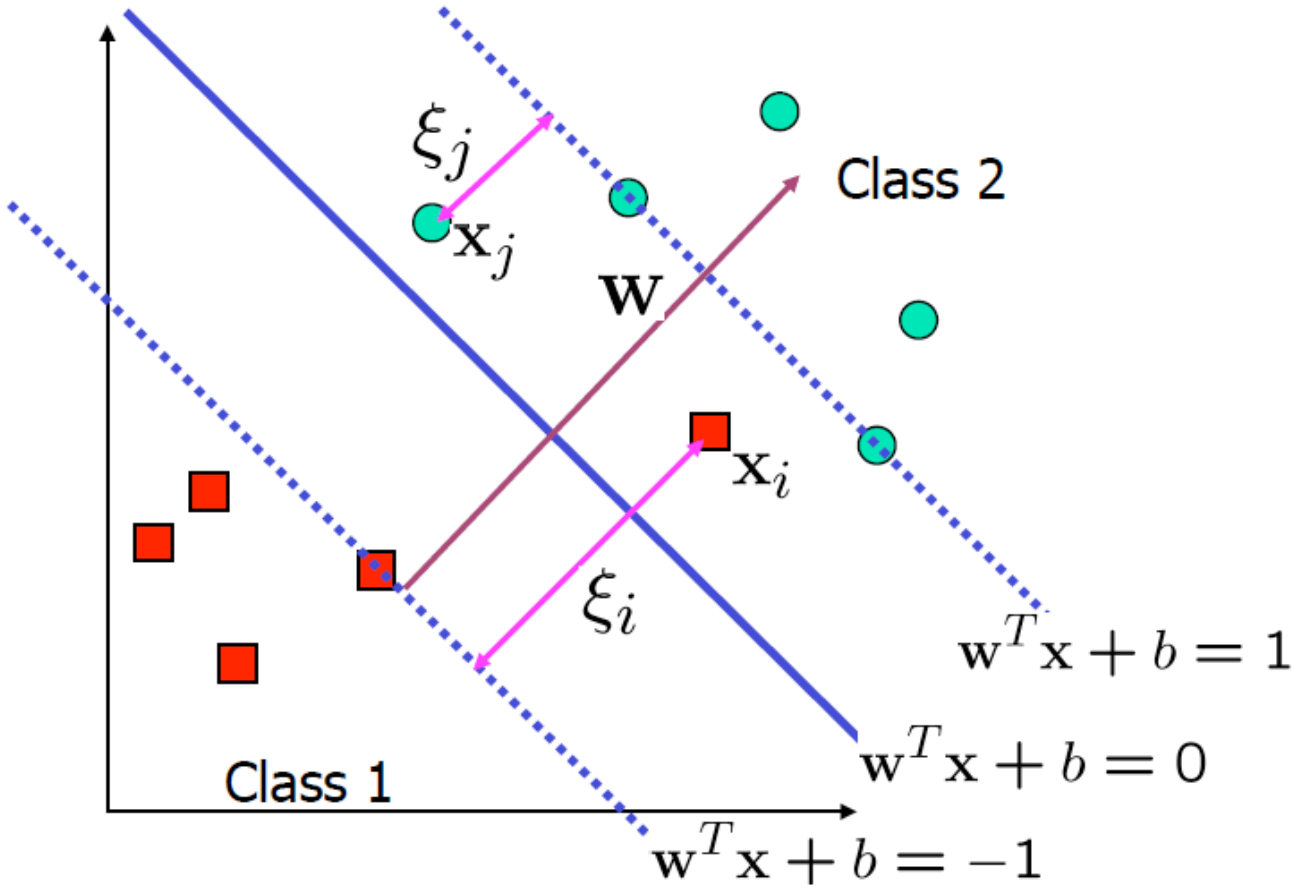


Sometimes finding a hard separating hyperplane is not exactly what we want.

Relaxing the Constraints

- In practice, a separating hyperplane may not exist, possibly due to a high noise level which causes a large overlap of the classes.
- Even if a separating hyperplane exists, it is not always the best solution to the classification problem when there exist outliers in the data.
- A mislabeled example can become an outlier which affects the location of the separating hyperplane.

Soft-Margin



- Slack Variables ξ_i can be added to allow misclassification of difficult or noisy examples, resulting margin called soft margin.

Slack Variables

- A soft-margin SVM allows for the possibility of violating the inequality constraints

$$y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right) \geq 1$$

by introducing slack variables

$$\xi_i \geq 0, i = 1, \ldots, N$$

which store the derivation from the margin.

- Relaxed separation constraints:

$$y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right) \geq 1 - \xi_i$$

Penalty

- By making ξ_i large enough, the constraint on $(\boldsymbol{x}^{(i)}, y^{(i)})$ can always be met. To avoid trivial solution, we should penalize ξ_i in the objective function.
- Three cases:
  - $\xi_i = 0$: no slack with $\boldsymbol{x}^{(i)}$ (no penalty)
  - $0 < \xi_i < 1$: $\boldsymbol{x}^{(i)}$ lies on the right side of the hyperplane but in the margin (small penalty)
  - $\xi_i > 1$: $\boldsymbol{x}^{(i)}$ lies on the wrong side of the hyperplane (large penalty)
- Number of misclassifications: $\#\{\xi_i > 1\}$
- Number of non-separable instances: $\#\{\xi_i > 0\}$
- Soft error as additional penalty term:

$$\sum_i \xi_i$$

Soft-Margin SVM Formulation

The goal now is

- To make the margin as large as possible
- To keep the number of points with $\xi_i > 0$ as small as possible

Reformulated primal optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i \\ & \forall i, \xi_i \geq 0 \end{aligned}$$

where C is a regularization parameter, which trades off between margin maximization and training error minimization.

Primal Optimization Problem

- Lagrangian:

$$\mathcal{L}_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{n}\alpha_i[y^{(i)}(\mathbf{w}^\top\mathbf{x}^{(i)} + b) - 1 + \xi_i] - \sum_{i=1}^{N}\mu_i\xi_i$$

  where μ_i are new Lagrange multipliers to guarantee that $\xi_i \geq 0$

- Both the misclassified instances and the ones in the margin are penalized for better generalization, though the latter ones would be correctly classified during testing.

Dual Optimization Problem

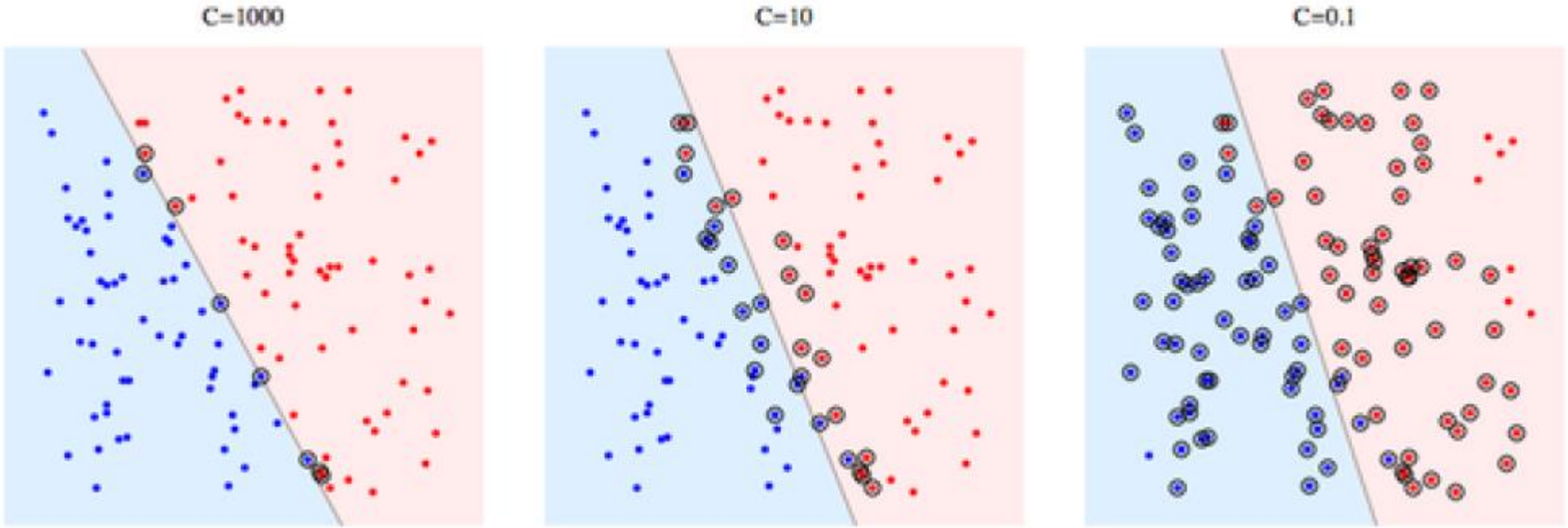
$$\max_{\boldsymbol{\alpha}} \quad -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)} + \sum_i \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

$$0 \leq \alpha_i \leq C, \forall i$$

- This is very similar to the optimization problem in the linear separable case, except that there is an upper bound C on α_i now.
- $\boldsymbol{w}$ and b can be computed similarly based on the support vectors.

Parameter C



- Circled points show support vectors.
- Decreasing C causes classifier to sacrifice linear separability in order to gain stability.

Loss Function of SVM

- Optimization Aspect:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i \\ & \forall i, \xi_i \geq 0 \end{aligned}$$

- Loss Function Aspect:

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} [1 - y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)]_+ + \lambda\|\mathbf{w}\|^2$$

- We can prove that the above two optimization problem are equivalent.

SVM vs. Logistic Regression

- When viewed from the point of view of regularized empirical loss minimization, SVM and logistic regression appear quite similar:

$$\text{SVM: } \underbrace{\sum_{i=1}^{n}[1 - y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)]_+}_{\text{loss}} + \underbrace{\lambda\|\mathbf{w}\|^2}_{\text{regularizer}}$$

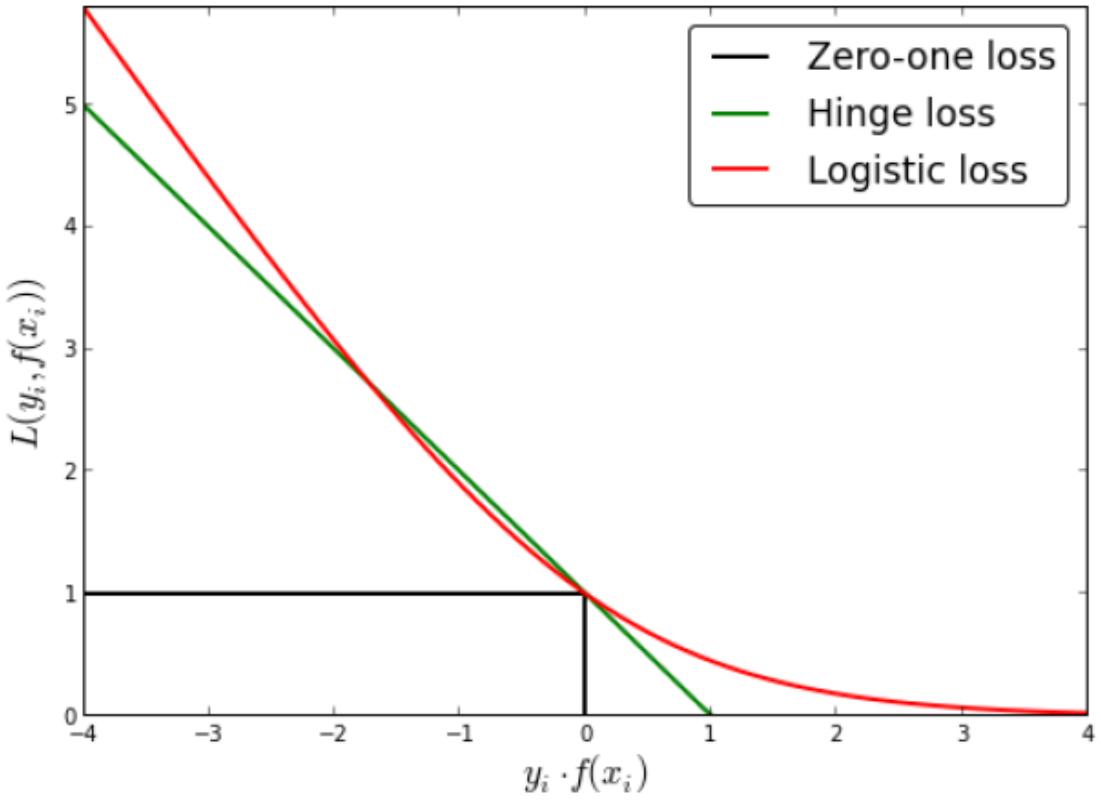
$$\text{LR: } \sum_{i=1}^{n} \underbrace{-\log \sigma(y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b))}_{-\log p(y^{(i)}|\mathbf{x}^{(i)})} + \lambda\|\mathbf{w}\|^2$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic function.

- Note that we have transformed the problem maximizing the penalized log-likelihood into minimizing negative penalized log-likelihood.

Hinge Loss as a Surrogate of the Binary Loss

- The loss function of SVM is:
  $\mathcal{L}(f, x, y) = [1 - y(\boldsymbol{w}^T \boldsymbol{x} + b)]_+$ Hinge Loss Function



0-1 loss is non-convex, non-smooth, and hard to optimize!

Tools for SVM

- LIBSVM
  http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- LIBLINEAR
  http://www.csie.ntu.edu.tw/~cjlin/liblinear/
- SVM-light, SVM-perf, SVM-struct
  http://svmlight.joachims.org/svm_struct.html
- Pegasos
  http://www.cs.huji.ac.il/~shais/code/index.html

Usage of LibSVM

- Let Us Try A Practical Example
- A problem from astroparticle physics

```
1.0 1:2.617300e+01 2:5.886700e+01 3:-1.894697e-01 4:1.251225e+02
1.0 1:5.707397e+01 2:2.214040e+02 3:8.607959e-02 4:1.229114e+02
1.0 1:1.725900e+01 2:1.734360e+02 3:-1.298053e-01 4:1.250318e+02
1.0 1:2.177940e+01 2:1.249531e+02 3:1.538853e-01 4:1.527150e+02
1.0 1:9.133997e+01 2:2.935699e+02 3:1.423918e-01 4:1.605402e+02
1.0 1:5.537500e+01 2:1.792220e+02 3:1.654953e-01 4:1.112273e+02
1.0 1:2.956200e+01 2:1.913570e+02 3:9.901439e-02 4:1.034076e+02
```

- Training and testing sets available: 3,089 and 4,000
- Data format is an issue

Usage of LibSVM

- http://www.csie.ntu.edu.tw/~cjlin/libsvm
- Now one of the most used SVM software

- Installation
- On Unix:
  - Download zip file and make
- On Windows:
  - Download zip file and make
  - c:nmake -f Makefile.win
  - Windows binaries included in the package

Usage of LibSVM

- Training
  - Usage: svm-train [options] training_set_file
  - options:
  - -s svm_type : set type of SVM (default 0)
    - 0 -- C-SVC
    - 1 -- nu-SVC
    - 2 -- one-class SVM
    - 3 -- epsilon-SVR
    - 4 -- nu-SVR
  - -t kernel_type : set type of kernel function
- Testing
  - Usage: svm-predict test_file model_file output

Usage of LibSVM

- Training and Testing

Training

```
$./svm-train train.1
......*
optimization finished, #iter = 6131
nu = 0.606144
obj = -1061.528899, rho = -0.495258
nSV = 3053, nBSV = 724
Total nSV = 3053
```

Testing

```
$./svm-predict test.1 train.1.model
  test.1.predict
Accuracy = 66.925% (2677/4000)
```

What does this Output Mean

- obj: the optimal objective value of the dual SVM
- rho: − b in the decision function
- nSV and nBSV: number of support vectors and bounded support vectors (i.e., $\alpha_i = C$).
- nu-svm is a somewhat equivalent form of C-SVM where C is replaced by ν.

Data Scaling

- Without scaling Attributes in greater numeric ranges may dominate

Example:

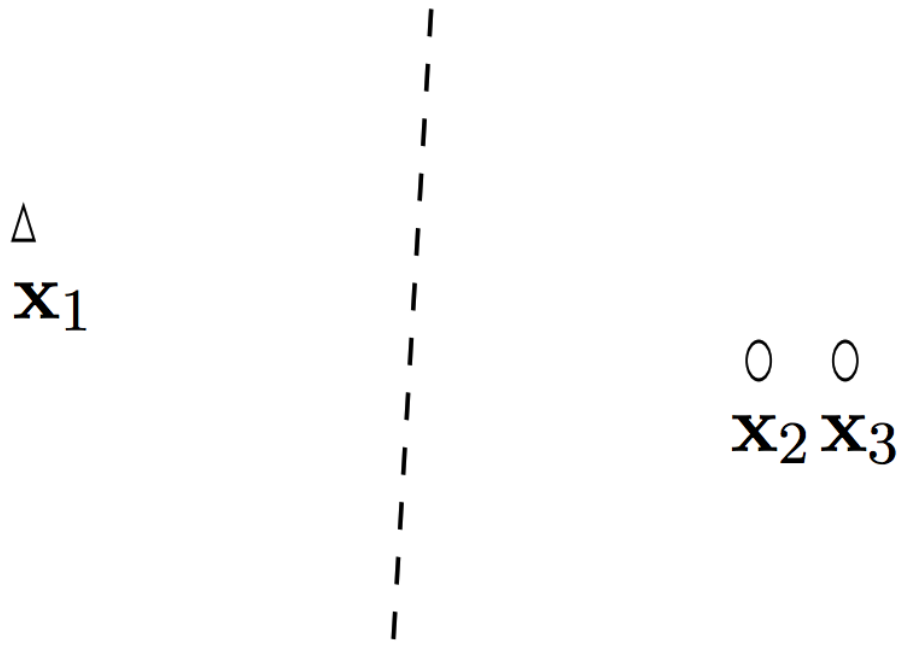
	height	sex
$\mathbf{x}_1$	150	F
$\mathbf{x}_2$	180	M
$\mathbf{x}_3$	185	M

and

$$y_1 = 0, y_2 = 1, y_3 = 1.$$

Usage of LibSVM

The separating hyperplane



Decision strongly depends on the first attribute

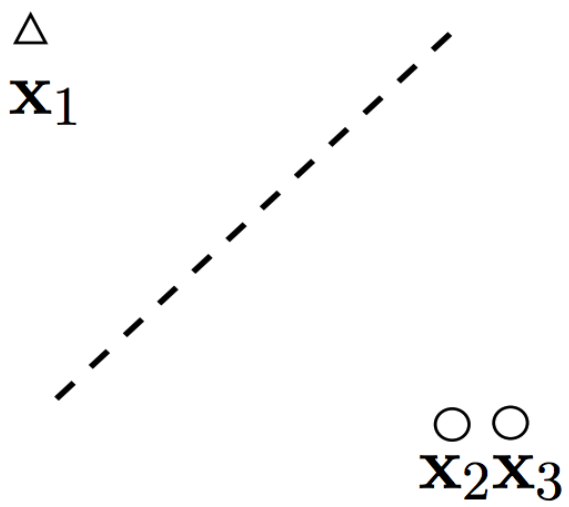
What if the second is more important

Usage of LibSVM

Linearly scale the first to $[0, 1]$ by:

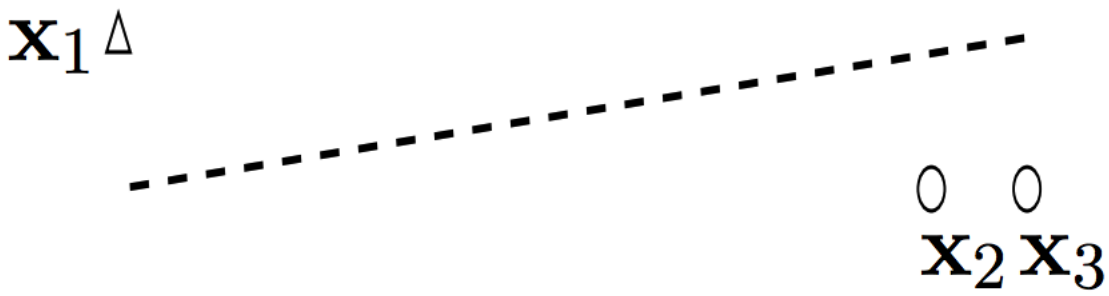
$$\frac{\text{1st attribute} - 150}{185 - 150},$$

New points and separating hyperplane



Usage of LibSVM

Transformed to the original space,



The second attribute plays a role