给定一个训练集，我们的目标是找到一个决策边界，使我们能够对训练示例做出所有正确和自信（意味着远离决策边界）的预测。

- Given a point (x_0, y_0), the distance from the point to the line $Ax + By + C = 0$:

$$distance = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

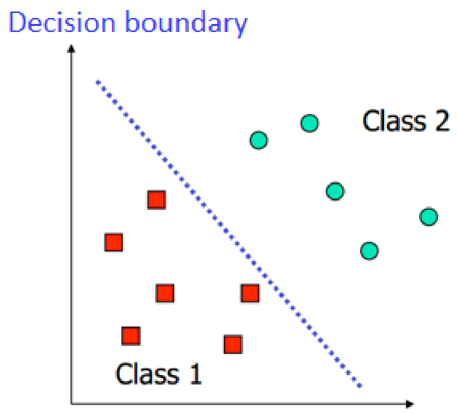
- Given a point $\boldsymbol{x}_i$, the distance from the point to the hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$:

$$distance = \frac{|\boldsymbol{w}^T\boldsymbol{x} + b|}{\|\boldsymbol{w}\|}$$

w是和直线垂直的向量

What Is a Good Decision Boundary?

- We aim to find the hyperplane (i.e., decision boundary) linearly separating our classes.
  我们的目标是找到线性分离类的超平面（即决策边界）。
- Our boundary will have equation: $\boldsymbol{w}^T\boldsymbol{x} + b = 0$



- Above the decision boundary should have label 1, i.e., for any $\boldsymbol{x}_i$ s.t. $\boldsymbol{w}^T\boldsymbol{x} + b > 0$, then $y_i = 1$.

- Below the decision boundary should have label -1, i.e., for any $\boldsymbol{x}_i$ s.t. $\boldsymbol{w}^T\boldsymbol{x} + b < 0$, then $y_i = -1$.

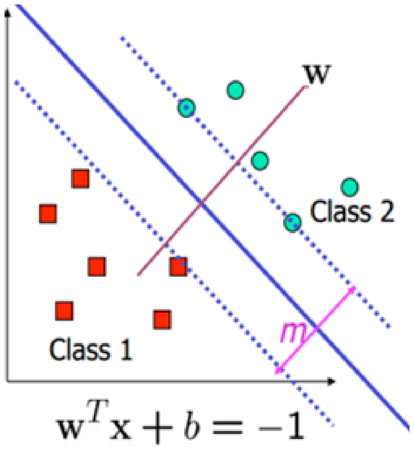
$$f(x) = sign(\boldsymbol{w}^T\boldsymbol{x} + b)$$

- Moreover, we hope the hyperplane lies in the middle

$$\begin{cases} (\boldsymbol{w}^T\boldsymbol{x}_i + b)/\|\boldsymbol{w}\| \geq \frac{m}{2} & \forall\, y_i = 1 \\ (\boldsymbol{w}^T\boldsymbol{x}_i + b)/\|\boldsymbol{w}\| \leq -\frac{m}{2} & \forall\, y_i = -1 \end{cases}$$

$distance = \frac{|\boldsymbol{w}^T\boldsymbol{x}+b|}{\|\boldsymbol{w}\|}$

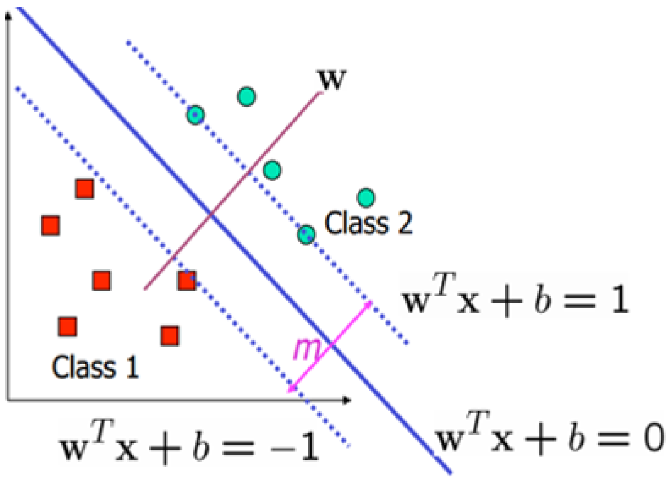
m is the margin



m is the margin，两类之间最近的距离

- Therefore,

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_i + b \geq 1 & \forall\, y_i = 1 \\ \boldsymbol{w}^T\boldsymbol{x}_i + b \leq -1 & \forall\, y_i = -1 \end{cases} \Rightarrow y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$



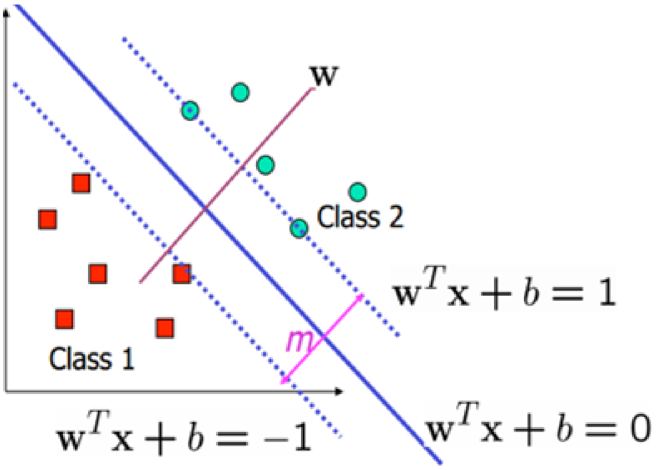
决策边缘应尽可能远离这两个类的数据，应该取m的最大值

- For the **support vectors** (data points nearest to the hyperplane)

$$Distance = |\boldsymbol{w}^T\boldsymbol{x}_i + b| / \|\boldsymbol{w}\|$$

$$= 1/\|\boldsymbol{w}\|$$

$$m = 2/\|\boldsymbol{w}\|$$



以上是一个具有凸二次目标和仅线性约束的优化问题。

Exercise

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$s.t.\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1,\ i = 1,2,\ldots,n$$

- Given the dataset consist of two positive samples $\boldsymbol{x}_1 = (3,3)^T$, $\boldsymbol{x}_2 = (4,3)^T$, and one negative sample $\boldsymbol{x}_3 = (1,1)^T$. Please write the objective function with SVM.

Answer

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}w_1^2 + \frac{1}{2}w_2^2$$

$$\begin{aligned} \text{s.t. } & 3w_1 + 3w_2 + b \geq 1, \\ & 4w_1 + 3w_2 + b \geq 1, \\ & -w_1 - w_2 - b \geq 1, \end{aligned}$$

拉格朗日对偶性

对偶形式将使我们能够推导出一种有效的算法来解决优化问题。

对偶形式将允许我们使用核来获得最优边缘分类器，以便在非常高维的空间中有效地工作。

Constrained Optimization

Consider a problem of the following form:

$$\min_{w} f(\boldsymbol{w})$$

$$\text{s.t. } h_i(\boldsymbol{w}) = 0,\ i = 1, \dots, l.$$

Lagrange multiplier method:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\beta}) = f(\boldsymbol{w}) + \sum_{i=1}^{l} \beta_i h_i(\boldsymbol{w})$$

β_i's are the Lagrange multipliers.

No constraint now.

Set the partial derivatives to zero:

$$\frac{\partial \mathcal{L}(\boldsymbol{w}, \boldsymbol{\beta})}{\partial w_j} = 0 \qquad \frac{\partial \mathcal{L}(\boldsymbol{w}, \boldsymbol{\beta})}{\partial \beta_i} = 0$$

不等式约束优化

将其推广到约束优化问题，在这些问题中，我们可能存在不等式和等式约束。

Consider the following primal optimization problem:

$$\min_{w} f(\boldsymbol{w})$$

$$\text{s.t. } g_i(\boldsymbol{w}) \le 0,\ i = 1, \dots, k$$

$$h_i(\boldsymbol{w}) = 0,\ i = 1, \dots, l.$$

Generalized Lagrangian

α_i's and β_i's are the Lagrange multipliers.

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

$\alpha_i \ge 0$ **Why?**

Optimization with Inequality Constraints

Consider the following primal optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} f(\boldsymbol{x})$$

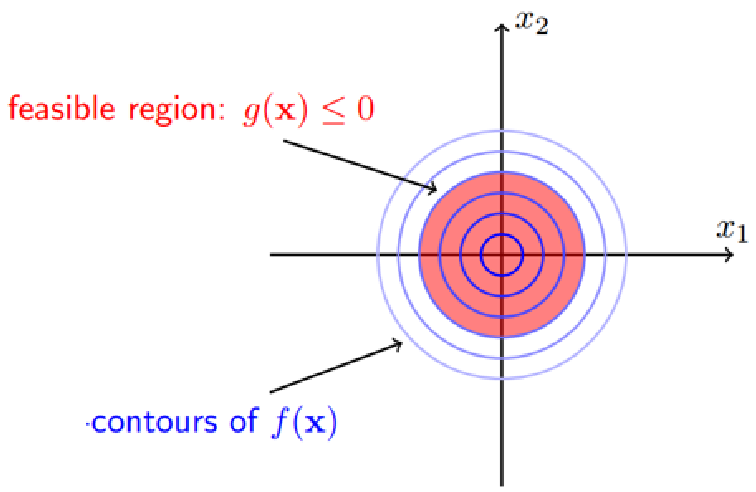
$$\text{s.t. } g(\boldsymbol{x}) \leq 0$$

Example1:

$$\boxed{f(\mathbf{x}) = x_1^2 + x_2^2 \textbf{ and } g(\mathbf{x}) = x_1^2 + x_2^2 - 1}$$

Optimization with Inequality Constraints

$$\boxed{f(\mathbf{x}) = x_1^2 + x_2^2 \textbf{ and } g(\mathbf{x}) = x_1^2 + x_2^2 - 1}$$



$$\boxed{g(\mathbf{x}) = x_1^2 + x_2^2 - 1}$$

Optimization with Inequality Constraints

Problem:

Our constrained optimization problem

$$\boxed{\min_{\mathbf{x}\in\mathbb{R}^2} f(\mathbf{x}) \text{ subject to } g(\mathbf{x}) \leq 0}$$

where

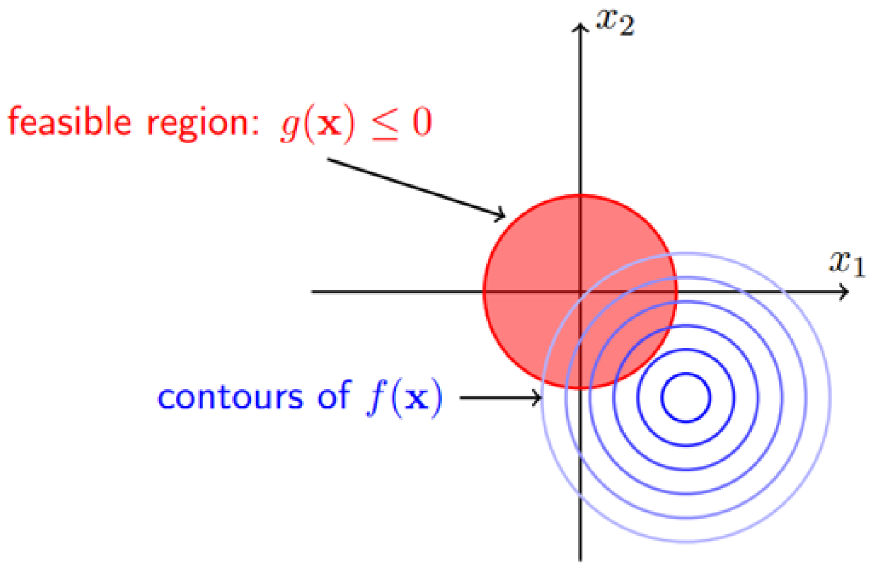
$$\boxed{f(\mathbf{x}) = x_1^2 + x_2^2 \textbf{ and } g(\mathbf{x}) = x_1^2 + x_2^2 - 1}$$

Constraint is not active at the local minimum ($g(\mathbf{x}^*) < 0$):

Therefore the local minimum is identified by the same conditions as in the unconstrained case. 有约束和无约束条件下的解一样

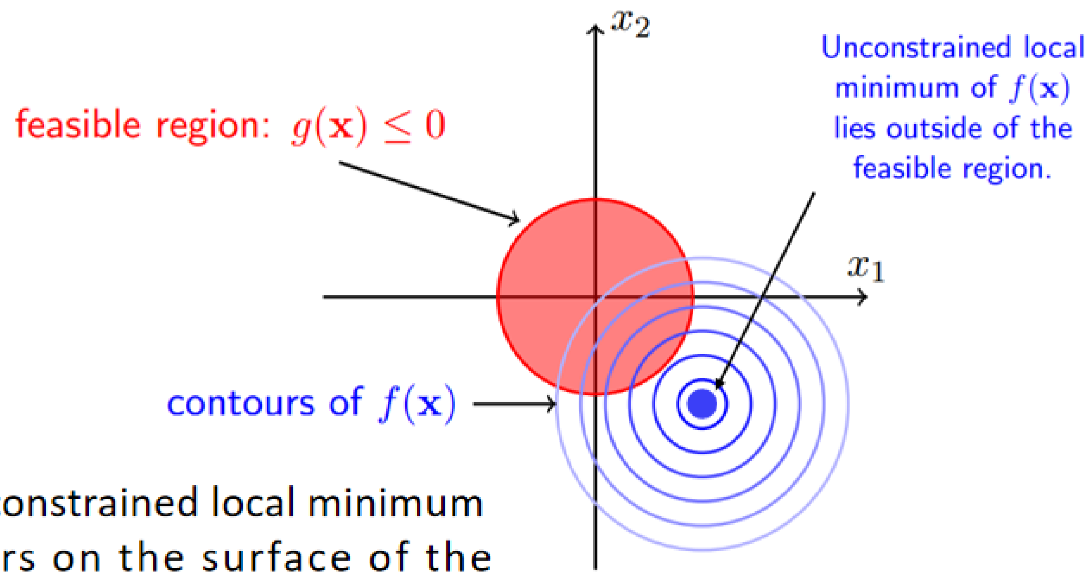
Optimization with Inequality Constraints

$$\boxed{f(\mathbf{x}) = (x_1 - 1.1)^2 + (x_2 + 1.1)^2 \textbf{ and } g(\mathbf{x}) = x_1^2 + x_2^2 - 1}$$



$$\boxed{g(\mathbf{x}) = x_1^2 + x_2^2 - 1}$$

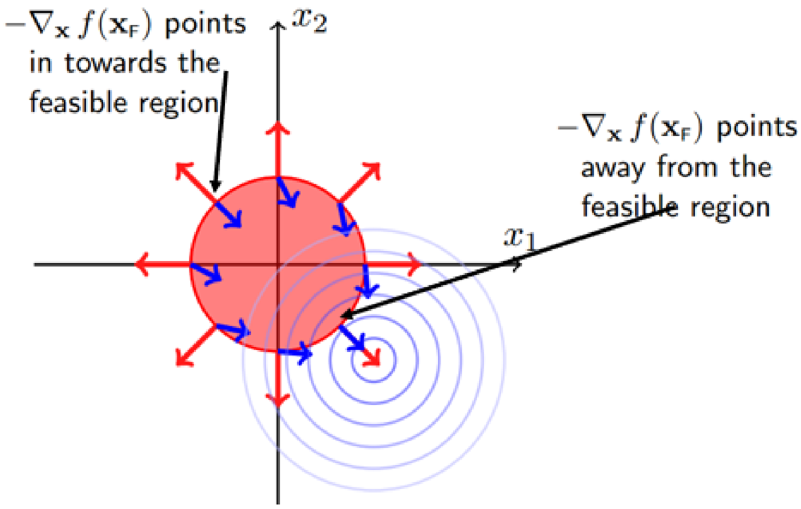
约束条件在g(x）=0时起作用



The constrained local minimum occurs on the surface of the constraint surface. $g(\boldsymbol{x}) = 0$

$$\boxed{g(\mathbf{x}) = x_1^2 + x_2^2 - 1}$$

Red: $\nabla_{\boldsymbol{x}} g(\boldsymbol{x})$
Blue: $-\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$



$\therefore$ Constrained local minimum occurs when $-\nabla_{\mathbf{x}} f(\mathbf{x})$ and $\nabla_{\mathbf{x}} g(\mathbf{x})$ point in the same direction:

$$\boxed{-\nabla_{\mathbf{x}} f(\mathbf{x}) = \lambda \nabla_{\mathbf{x}} g(\mathbf{x}) \quad \textbf{and} \quad \lambda > 0}$$

当f(x)与g(x)的导数不同号时，取得局部最小值

If $\mathbf{x}^*$ corresponds to a constrained local minimum then

Case 1:
Unconstrained local minimum occurs **in** the feasible region.

1. $g(\mathbf{x}^*) < 0$ 不起约束作用
2. $\nabla_{\mathbf{x}} f(\mathbf{x}^*) = \mathbf{0}$

Case 2:
Unconstrained local minimum lies **outside** the feasible region.

1. $g(\mathbf{x}^*) = 0$
2. $-\nabla_{\mathbf{x}} f(\mathbf{x}^*) = \lambda \nabla_{\mathbf{x}} g(\mathbf{x}^*)$ with $\lambda > 0$

起约束作用，在边界

Dual optimization problem

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{T}\boldsymbol{x}_j$$

$$s.t.\ \alpha_i \geq 0, i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

坐标上升方法

Coordinate Ascent 坐标上升法

- Consider trying to solve the unconstrained optimization problem

$$\max_{\alpha} L(\alpha_1, \alpha_2, \ldots, \alpha_l)$$

- Coordinate Ascent

Loop until convergence:{

For $i = 1, \ldots l${

$$\alpha_i := \arg\max_{\hat{\alpha}_i} L(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots \alpha_l)$$

}

}

在该算法的最内循环中，我们将保持除某些α_i之外的所有变量固定，并仅针对参数α_i重新优化L。

SMO最小序列方法

每次只优化一个参数，其他参数先固定住，仅求当前这个优化参数的极值。我们来看一下 SMO 算法在 SVM 中的应用。

假设我们有一组满足约束的α_i

假设我们固定α_2，…，α_n，我们可以采取坐标上升步骤并优化关于α_1的函数吗?

- **NO!!!** $\sum_{i=1}^{n} \alpha_i y_i = 0$ $\alpha_1 = -y_1 \sum_{i=2}^{n} \alpha_i y_i$

我们必须**同时更新至少两个α_i**。

重复，直到收敛{

选择一些对α_i和α_j进行下一次更新（使用启发式方式，尝试选择两个，使我们能够朝着全局最大值取得最大进展）。

关于α_i和α_j重新优化L（α），同时保持所有其他α_k（k≠i，j）固定

}

SMO是有效的，因为可以非常有效地计算对α_i和α_j的更新。

支持向量及其性质

$$\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{n} \alpha_i y_i = \zeta \quad \text{Constant，常数}$$

$$L(\alpha_1, \alpha_2, \ldots, \alpha_n) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\boldsymbol{T}} \boldsymbol{x}_j$$

$$L(\alpha_1, \alpha_2, \ldots, \alpha_n) = L(y_1(\zeta - \alpha_2 y_2), \alpha_2, \ldots, \alpha_n)$$

- 这是一个α2的二次函数
- Once we have α_2^{new}, we can obtain α_1^{new} with $\alpha_1 y_1 + \alpha_2 y_2 = \zeta$

不能没有支持向量

- Question: can we have no support vector? $\boldsymbol{\alpha}^* = \mathbf{0}$

- Answer: No.
- If $\boldsymbol{\alpha}^* = \mathbf{0}$, then $\boldsymbol{w}^* = \mathbf{0}$. (This is not the optimal solution for the primal optimization problem)这不是原始优化问题的最优解

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w}$$

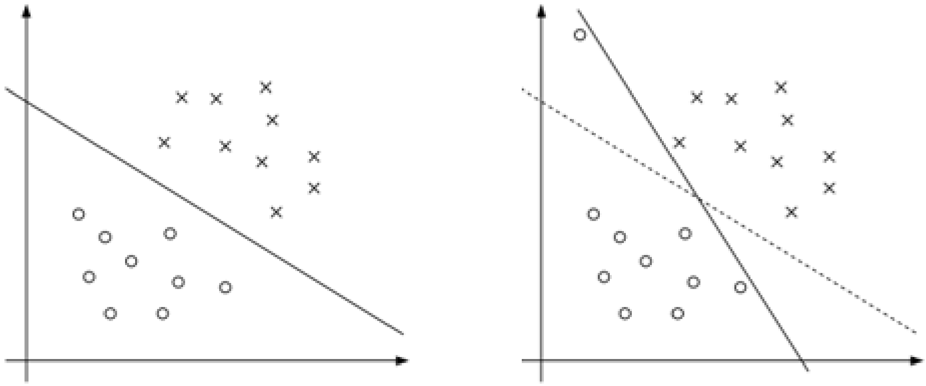
$$s.t.\ y_i(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_i + b) \geq 1,\ i = 1,2, \ldots, n$$

大多数的α是0.因为支持向量是少数

正则化与不可分情形

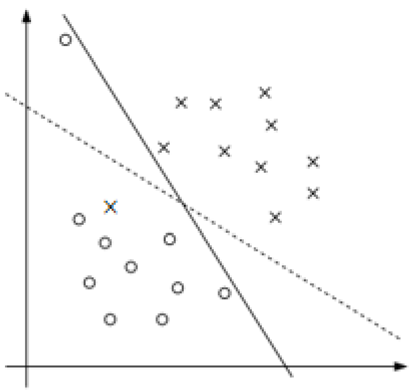
在某些情况下（由于异常值），我们不清楚找到一个分离超平面是否正是我们想要做的

图（a）显示了一个最优裕度分类器，当在左上角区域添加单个异常值时（图b），它会导致决策边界发生剧烈波动，结果分类器的裕度要小得多（对异常值敏感）。



(a) Linearly separable　(b) Linearly separable with outliers

在某些情况下（图c），数据不能完全线性分离。



(c) Non-linearly separable

软间隔

正松弛变量

如果一个例子的裕度为$1-\xi_i$（$\xi_i>0$），我们将付出目标函数增加$C\xi_i$的代价。

C控制两个目标之间的相对权重

- 使“w”变小（使裕度变大）；
- 确保大多数示例的裕度至少为1

非线性

我们不想使用原始输入空间x来应用SVM，而是想使用一些特征空间$\phi(x)$来学习

要做到这一点，我们只需要回顾我们以前的SVM算法，并将其中所有的x都替换为ξ（x）。

Train SVM:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\boldsymbol{T}} \boldsymbol{x}_j \qquad \boxed{\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)}$$

$$s.t.\ 0 \le \alpha_i \le C,\ i = 1, \dots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Test SVM:

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = \sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i$$

$$f(\boldsymbol{x}) = sign(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b)$$

$$= sign\left(\left(\sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i^{\boldsymbol{T}}\, \boldsymbol{x}\right) + b\right) \qquad \boxed{\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x})}$$

能计算在特征空间下的ϕ(xi)^Tϕ(xj)，就不需要求显式的ϕ(xi)

核函数

- Define the kernel function K by

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$$

Train SVM:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\boldsymbol{T}} \boldsymbol{x}_j \qquad \boxed{\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)} \qquad \boxed{K(\boldsymbol{x}_i, \boldsymbol{x}_j)}$$

$$s.t.\ 0 \le \alpha_i \le C,\ i = 1, \dots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Test SVM:

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i = \sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i$$

$$f(\boldsymbol{x}) = sgn(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b)$$

$$= sgn\left(\left(\sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i^{\boldsymbol{T}}\, \boldsymbol{x}\right) + b\right) \qquad \boxed{\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x})} \qquad \boxed{K(\boldsymbol{x}_i, \boldsymbol{x})}$$

Intuition:

- If $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$ are close, we might want $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to be large.
- If $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$ are far apart, we might want $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to be small.

我们可以把K（xi，xj）看作是对ξ（xi）和ξ（xj）相似程度的测量。

高斯核

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right) \Leftrightarrow K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right), \gamma > 0$$

γ越大，越容易过拟合

核函数的判定

Given the training dataset $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$.
Let $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ be the (i, j)-entry of $\boldsymbol{K} \in \mathbb{R}^{n \times n}$.
$\boldsymbol{K}$ is called the *Kernel matrix*.
If K is a valid kernel, then $K_{ij} = \phi(\boldsymbol{x}_i)^{\boldsymbol{T}}\phi(\boldsymbol{x}_j) = \phi(\boldsymbol{x}_j)^{\boldsymbol{T}}\phi(\boldsymbol{x}_i) = K_{ji}$.
$\boldsymbol{K}$ is symmetric. $\boldsymbol{K}$ is positive semi-definite.

要满足Kij==kji，K是对称矩阵，是半正定的（所有的特征值都是非负的）

SVM的评价

优点：

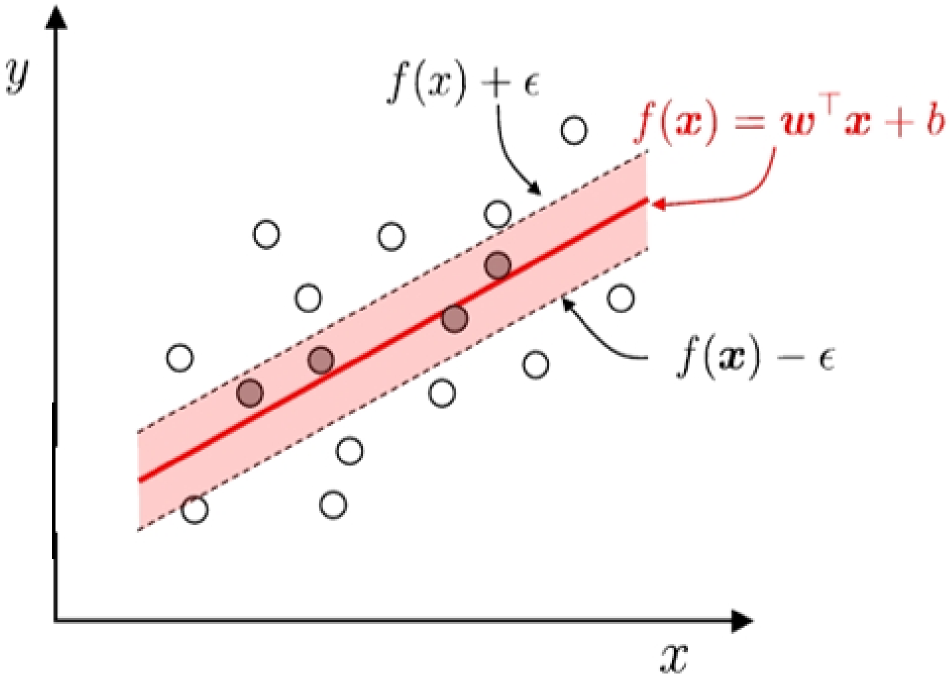
- **没有局部解**
- 它对高维数据的扩展性相对较好
- 分类器复杂度和误差之间的权衡可以明确控制
- **字符串和树等非传统数据可以用作SVM的输入**，而不是特征向量

缺点：需要选择一个好的核函数

SVM回归

特征空间中的线性回归

与最小二乘回归不同，误差函数为ε-不敏感损失函数，直觉上，小于ε的错误被忽略



y
$f(x) + \epsilon$
$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$
$f(\boldsymbol{x}) - \epsilon$
x