第3章　支持向量机

刘家锋

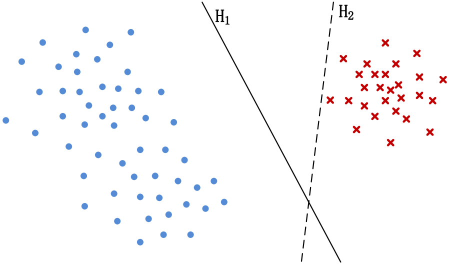
哈尔滨工业大学

第3章 支持向量机

3.1 最优超平面与支持向量机

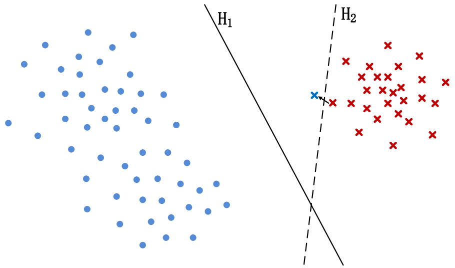
最优分类超平面

- **线性分类器的优劣**
  - 对于线性可分的训练样例来说，分类超平面H_1和H_2是等价的，都能正确分类所有样例；



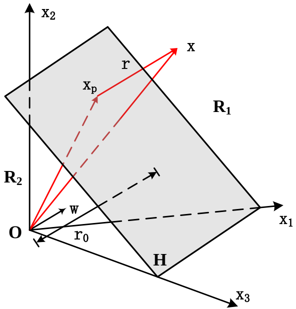
最优分类超平面

- **线性分类器的优劣**
  - 对于线性可分的训练样例来说，分类超平面H_1和H_2是等价的，都能正确分类所有样例；
  - 对于未来的测试样例来说，距离训练样例更近的超平面H_2产生错误分类的风险更大；



线性判别函数与超平面

- **分类超平面**
  - 给定线性判别函数：$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + b$
  - 分类超平面H上的点满足方程：$g(\mathbf{x}) = 0$
  - 分类超平面将空间划分为两个区域，分别代表两个类别；



样例到超平面的距离

- **空间中任意点x到超平面H的距离：** $r = g(\mathbf{x})/\|\mathbf{w}\|$
  - 首先证明权值向量$\mathbf{w}$垂直于超平面H：

    $\forall \mathbf{x}_p, \mathbf{x}_q \in H$，满足超平面方程：$\mathbf{w}^t\mathbf{x} + b = 0$

    $$\mathbf{w}^t(\mathbf{x}_p - \mathbf{x}_q) = \mathbf{w}^t\mathbf{x}_p + b - (\mathbf{w}^t\mathbf{x}_q + b) = 0$$

    因此有，$\mathbf{w}$正交于向量$\mathbf{x}_p - \mathbf{x}_q$；由$\mathbf{x}_p$和$\mathbf{x}_q$的任意性，证明$\mathbf{w}$正交于超平面$H$；
  - 计算点到超平面的距离：

    令$\mathbf{x}_p \in H$为$\mathbf{x}$向H所引垂线的垂足，$\mathbf{x}$到H的距离为r

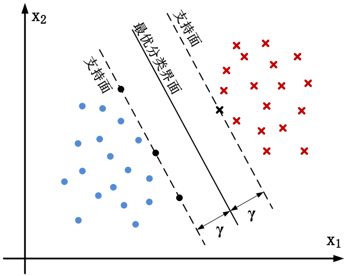
    $$\overrightarrow{O\mathbf{x}} = \overrightarrow{O\mathbf{x}_p} + \overrightarrow{\mathbf{x}_p\mathbf{x}} \quad \Longrightarrow \quad \mathbf{x} = r \cdot \mathbf{w}/\|\mathbf{w}\| + \mathbf{x}_p$$

    因此：

    $$\begin{aligned} g(\mathbf{x}) &= \mathbf{w}^t\left(r \cdot \mathbf{w}/\|\mathbf{w}\| + \mathbf{x}_p\right) + b \\ &= r\|\mathbf{w}\| + (\mathbf{w}^t\mathbf{x}_p + b) = r\|\mathbf{w}\| \end{aligned}$$

最优分类超平面

- **最优分类超平面**
  - 样例集与分类超平面之间的间隔γ：样例到分类超平面距离的最小值；
  - 最优分类超平面：能够将样例集区分开的最大间隔超平面；



最优分类超平面

- **支持向量**
  - 支持向量(Support Vector)：距离分类超平最近的训练样本；
  - 支持面：支持向量所在的平行于最优分类超平面的超平面；
  - 在最优分类超平面两侧各有一个支持面，与最优分类超平面的距离均为γ；
- **支持向量机(Support Vector Machine)的学习问题**
  - 给定二分类数据集：$D=\{(\mathbf{x}_1,y_1),\cdots,(\mathbf{x}_m,y_m)\}$，类别标记$y_i\in\{-1,+1\}$；
  - 学习判别函数$g(\mathbf{x})=\mathbf{w}^t\mathbf{x}+b$，能够正确区分训练样例集，并且与样例集的间隔最大，即最优分类界面；

SVM的优化目标函数

- **最优超平面的条件**
  - 当判别函数$g(\mathbf{x})$能够正确区分训练样例时，有下式成立：

$$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq \min_{1 \leq j \leq m} \left[y_j(\mathbf{w}^t\mathbf{x}_j + b) \right] = z_{min} > 0$$

  - 令$\mathbf{w} \leftarrow \mathbf{w}/z_{min}, b \leftarrow b/z_{min}$，不会改变分类超平面的位置；
  - 满足下述条件的判别函数，能够将训练样例正确分类：

$$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1, \quad i = 1, \cdots, m$$

SVM的优化目标函数

- **最大化间隔**
  - 最靠近分类超平面的样例到超平面的距离：$z_{min} = 1$，因此间隔：

$$\gamma = \frac{1}{\|\mathbf{w}\|}$$

  - 最大化间隔γ，等价于最小化权值向量的长度$\|\mathbf{w}\|$；
- **SVM的优化问题**

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

$$subject\ to$$

$$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1, \quad i = 1, \cdots, m$$

凸二次规划问题可以直接优化计算，但也存在更高效的办法；

3.2 对偶问题

约束优化问题的求解

- **Min-Max原理**
  - SVM是一个典型的线性不等式约束的二次优化问题;
  - 下面引用John von Neumann的Min-Max原理,来说明约束优化问题的求解方法;
  - 对于函数$F(\mathbf{u},\mathbf{v})$,有如下两个优化问题:

$$\text{Min-Max:} \quad \begin{cases} F^*(\mathbf{u}) = \max_{\mathbf{v}} F(\mathbf{u},\mathbf{v}) \\ \min_{\mathbf{u}} F^*(\mathbf{u}) = \min_{\mathbf{u}} \max_{\mathbf{v}} F(\mathbf{u},\mathbf{v}) \end{cases}$$

$$\text{Max-Min:} \quad \begin{cases} F^*(\mathbf{v}) = \min_{\mathbf{u}} F(\mathbf{u},\mathbf{v}) \\ \max_{\mathbf{v}} F^*(\mathbf{v}) = \max_{\mathbf{v}} \min_{\mathbf{u}} F(\mathbf{u},\mathbf{v}) \end{cases}$$

  - 如果两个问题的解存在,必在同一点取得最优解:

$$\min_{\mathbf{u}} \max_{\mathbf{v}} F(\mathbf{u},\mathbf{v}) = \max_{\mathbf{v}} \min_{\mathbf{u}} F(\mathbf{u},\mathbf{v}) = F(\mathbf{u}^*,\mathbf{v}^*)$$

Kuhn-Tucker构造法

- **SVM的原始优化问题**

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

$$subject\ to$$

$$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1, \quad i = 1, \cdots, m$$

- 构造Lagrange函数：

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left[y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1\right]$$

其中，$\alpha_i \geq 0$称为Lagrange乘数，第2项的最小值为0，因此有：

$$\max_{\{\alpha_i \geq 0\}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2$$

Kuhn-Tucker构造法

- **原始优化问题到对偶优化问题**
  - SVM的原始优化问题等价于：

$$\min_{\mathbf{w},b}\ \max_{\{\alpha_i\geq 0\}}\ L(\mathbf{w},b,\boldsymbol{\alpha})$$

  - 根据Min-Max原理：

$$\min_{\mathbf{w},b}\ \max_{\{\alpha_i\geq 0\}}\ L(\mathbf{w},b,\boldsymbol{\alpha}) = \max_{\{\alpha_i\geq 0\}}\ \min_{\mathbf{w},b} L(\mathbf{w},b,\boldsymbol{\alpha})$$

  - 优化次序可以互换，Lagrange函数先对$\mathbf{w}, b$优化；

Kuhn-Tucker构造法

- **原始优化问题到对偶优化问题**
  - Lagrange函数的最小值优化：

$$\min_{\mathbf{w},b} L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left[y_i(\mathbf{w}^t \mathbf{x}_i + b) - 1 \right]$$

  - 计算极值点条件：

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

Kuhn-Tucker构造法

- **原始优化问题到对偶优化问题**
  - 将$\mathbf{w}=\sum_{i=1}^m \alpha_i y_i \mathbf{x}_i$带入Lagrange函数：

$$
\begin{aligned}
L(\boldsymbol{\alpha}) &= \frac{1}{2}\mathbf{w}^t\mathbf{w} - \sum_{i=1}^m \left[\alpha_i y_i \mathbf{w}^t \mathbf{x}_i + \alpha_i y_i b - \alpha_i\right] \\
&= \frac{1}{2}\left(\sum_{i=1}^m \alpha_i y_i \mathbf{x}_i\right)^t \left(\sum_{i=1}^m \alpha_i y_i \mathbf{x}_i\right) \\
&\quad - \sum_{i=1}^m \left[\alpha_i y_i \left(\sum_{j=1}^m \alpha_j y_j \mathbf{x}_j\right)^t \mathbf{x}_i + \alpha_i y_i b - \alpha_i\right]
\end{aligned}
$$

  - 利用$\sum_{i=1}^m \alpha_i y_i = 0$，化简得到：

$$
L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j + \sum_{i=1}^m \alpha_i
$$

Kuhn-Tucker构造法

- **SVM的对偶优化问题**
  - 得到与原始优化问题等价的对偶优化问题：

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

$$subject\ to$$

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

$$\alpha_i \geq 0, \quad i = 1, \cdots, m$$

  - 优化对偶问题的好处：
    - 优化变量简单，只有Lagrange乘数$\boldsymbol{\alpha}$；
    - 只需要计算任意两个特征矢量的内积：$\mathbf{x}_i^t \mathbf{x}_j$

3.3 软间隔与正则化

软间隔

- **松弛变量**
  - 对于线性不可分的训练集，不等式约束不可能全部被满足；
  - 引入"松弛变量"$\xi_i \geq 0$，将不等式约束变为：

$$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

  - 只要选择一系列适合的松弛变量$\xi_1, \cdots, \xi_n$，不等式约束条件总是可以得到满足的；
  - 好的判别函数能够正确分类更多的样本，因此希望更多松弛变量$\xi_i = 0$；
  - 优化目标需要同时考虑分类界面的几何间隔，以及松弛变量的大小；

软间隔

- **软间隔SVM的原始优化问题**

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\begin{aligned} & \textit{subject to} \\ & \qquad y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \cdots, m \\ & \qquad \xi_i \geq 0, \quad i = 1, \cdots, m \end{aligned}$$

其中，参数C用来协调两个优化目标，需要设置；

- **构造Lagrange函数**

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\beta_i\xi_i$$

线性不可分的训练集

- **软间隔SVM的对偶优化问题**
  - 采用Kuhn-Tucker构造法，可以得到对偶优化问题：

$$\max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$

$$subject\ to$$

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

$$C \geq \alpha_i \geq 0, \quad i = 1, \cdots, m$$

与线性可分情况相比，对偶优化问题只是增加了α_i上界的约束；

SVM的优化求解

- **SVM的优化求解方法**
  - SVM的原始问题和对偶问题都是典型的二次凸优化，存在唯一解，一般选择比较简单的对偶问题求解；
  - 针对SVM的学习，存在专门的有效算法：序列最小化算法(SMO算法)；
  - 在大多数的机器学习软件包中都有相应的功能，例如Matlab 2019中的fitcsvm：

```
svm = fitcsvm( X,Z,'KernelFunction','linear','BoxConstraint',1 );
Label = predict( svm,x );
```

Lagrange乘数与支持向量

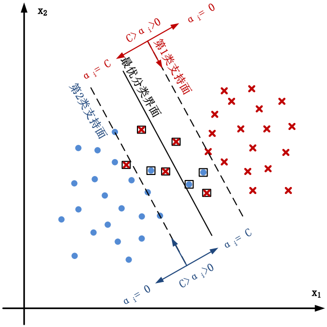
- **支持向量**
  - 对偶问题的解是m个Lagrange乘数$\{\alpha_1, \cdots, \alpha_m\}$;
  - 根据Kuhn-Tucker定理，有如下关系成立：

$$\begin{cases} y_i(\mathbf{w}^t\mathbf{x}_i + b) > 1, & \alpha_i = 0 \\ y_i(\mathbf{w}^t\mathbf{x}_i + b) = 1, & C > \alpha_i > 0 \\ y_i(\mathbf{w}^t\mathbf{x}_i + b) < 1, & \alpha_i = C \end{cases}$$

  - 其中，满足条件2，3的训练样本称为支持向量;

Lagrange乘数与支持向量

- 非支持向量
  - 支持面之外：$\alpha_i = 0$
- 支持向量
  - 支持面之上：$C > \alpha_i > 0$
  - 支持面之内：$\alpha_i = C$
  - 错误分类样本：$\alpha_i = C$



Lagrange乘数与支持向量

- **Kuhn-Tucker条件的简单解释**
  - 从Lagrange函数来看：

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}^t\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\beta_i\xi_i$$

  - 关于ξ_i的极值条件：

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$

Lagrange乘数与支持向量

- **Kuhn-Tucker条件的简单解释**
  - $y_i(\mathbf{w}^t\mathbf{x}_i + b) > 1$时
    - 只有$\alpha_i = 0$，$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$才能取得最大值；
  - $y_i(\mathbf{w}^t\mathbf{x}_i + b) = 1$时
    - 允许$\alpha_i > 0$；
    - $\xi_i = 0$，$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$取得最小值；
    - 对应$\beta_i > 0$，有$\alpha_i = C - \beta_i < C$；
  - $y_i(\mathbf{w}^t\mathbf{x}_i + b) < 1$时
    - $\xi_i > 0$，不等式条件才能满足；
    - 对应$\beta_i = 0$，有$\alpha_i = C - \beta_i = C$；

SVM的判别函数

- **Kuhn-Tucker条件的简单解释**
  - 通过对偶问题的求解，可以得到对应每个训练样本的最优Lagrange乘数：$\{\alpha_1, \cdots, \alpha_m\}$；
  - 判别函数的权值向量$\mathbf{w}$，可以由原始问题的极值条件得到：

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

  - 处于支持面上的样本$\mathbf{x}_i$与判别界面的函数间隔为1，并且对应的Lagrange乘数满足：$C > \alpha_i > 0$；
  - 判别函数的偏置可以由任意的支持面上的训练样本计算：

$$y_i(\mathbf{w}^t \mathbf{x}_i + b) = 1 \quad \Rightarrow \quad b = y_i - \mathbf{w}^t \mathbf{x}_i$$

损失函数和正则项

- **hinge损失函数**
  - 软间隔SVM的优化目标还可以用hinge loss函数表示：

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} l_{hinge}(y_i(\mathbf{w}^t\mathbf{x}_i + b))$$

其中：$l_{hinge}(z) = \max(0, 1 - z)$

- **正则项**
  - 统计学习模型的更一般形式：

$$\min_{f} \Omega(f) + C\sum_{i=1}^{m} l(f(\mathbf{x}_i), y_i)$$

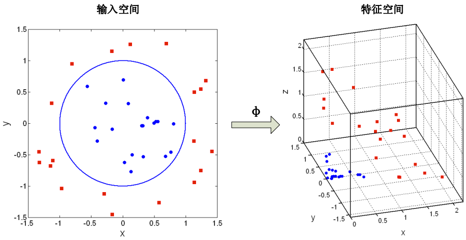
  - 第一项称为正则项，控制模型的复杂度；
  - 第二项称为经验风险，是模型在训练数据上的损失；
  - 参数C，调和两者的重要程度；

3.4 核函数

空间的非线性映射

- **非线性映射**
  - 低维空间线性不可分的数据，非线性地映射到高维特征空间，可以变为线性可分的数据；
  - 例如，$R^2 \to R^3$的非线性映射：

$$\phi : (x_1, x_2)^t \to (x_1^2, \sqrt{2}x_1x_2, x_2^2)^t$$



特征空间中的内积

- 高维特征空间中计算内积

$$\begin{aligned}\phi(\mathbf{x})^t\phi(\mathbf{y}) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)(y_1^2, \sqrt{2}y_1y_2, y_2^2)^t \\ &= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \\ &= (x_1y_1 + x_2y_2)^2 \\ &= (\mathbf{x}^t\mathbf{y})^2\end{aligned}$$

- 输入空间中计算特征空间内积
  - 定义核函数：$\kappa(\mathbf{x},\mathbf{y}) = (\mathbf{x}^t\mathbf{y})^2$
  - 特征空间中向量的内积可以直接在输入空间计算：

$$\phi(\mathbf{x})^t\phi(\mathbf{y}) = \kappa(\mathbf{x},\mathbf{y})$$

特征空间中的内积

- **特征空间中的计算**
  - 输入空间向量$\mathbf{x}, \mathbf{y} \in R^d$映射到特征空间$\phi(\mathbf{x}), \phi(\mathbf{y}) \in R^D$；
  - 特征空间中计算向量的内积$\phi(\mathbf{x})^t \phi(\mathbf{y})$；
  - 计算复杂度与特征空间的维数$D(\gg d)$相关；
  - 当$D \to +\infty$时，很难在特征空间中计算内积；
- **输入空间中的计算**
  - 只需计算定义在输入空间向量$\mathbf{x}, \mathbf{y}$上的核函数$\kappa(\mathbf{x}, \mathbf{y})$；
  - 计算复杂度只与输入空间的维数d相关；

核函数方法

- **核函数的应用**
  - 在输入空间定义核函数$\kappa(\mathbf{x}, \mathbf{y})$；
  - 线性方法中的向量内积，使用核函数计算；
- **应用条件**
  - 定义的核函数对应特征空间中的内积；
  - 线性方法不需要特征空间中的向量$\phi(\mathbf{x})$，只需要计算向量的内积$\phi(\mathbf{x})^t\phi(\mathbf{y})$；
- **特征空间中的内积都可以用核函数计算吗？**
  - 核函数的计算方式不是偶然的；
  - 但并不是所有函数都可以作为核函数，需要满足一定条件；

核函数

- **定理3.1（核函数）**
  - 令$\mathcal{X}$为输入空间，$\kappa(\cdot,\cdot)$是定义在$\mathcal{X}\times\mathcal{X}$上的对称函数，则$\kappa$是核函数当且仅当对于任意数据$D=\{\mathbf{x}_1,\cdots,\mathbf{x}_m\}$，核矩阵$K$总是半正定矩阵：

$$K=\begin{pmatrix}\kappa(\mathbf{x}_1,\mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_1,\mathbf{x}_m)\\ \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m,\mathbf{x}_1) & \cdots & \kappa(\mathbf{x}_m,\mathbf{x}_m)\end{pmatrix}$$

  - K为半正定矩阵，当且仅当对于任意向量$\mathbf{v}$：

$$\mathbf{v}^t K\mathbf{v}\geq 0$$

  - 半正定矩阵的特征值大于等于0；

核函数

- **常用核函数**

线性核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^t \mathbf{x}_j$	
多项式核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^t \mathbf{x}_j)^d$	$d \geq 1$为多项式的次数
高斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\lVert\mathbf{x}_i - \mathbf{x}_j\rVert^2}{2\sigma^2}\right)$	$\sigma \geq 0$为高斯核的带宽
拉普拉斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\lVert\mathbf{x}_i - \mathbf{x}_j\rVert}{\sigma}\right)$	$\sigma \geq 0$
Sigmoid核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^t \mathbf{x}_j + \theta)$	$\beta > 0, \theta < 0$

双曲正切函数： $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

非线性的支持向量机

- **特征空间中的SVM**
  - 原始问题优化：

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$subject\ to$$

$$y_i(\mathbf{w}^t\phi(\mathbf{x})_i + b) \geq 1 - \xi_i, \quad i = 1, \cdots, m$$
$$\xi_i \geq 0, \quad i = 1, \cdots, m$$

  - 判别函数：

$$f(\mathbf{x}) = \mathbf{w}^t\phi(\mathbf{x}) + b$$

  - 没有定义映射ϕ，特征空间中的向量$\mathbf{w}$无法直接求解；

非线性的支持向量机

- **特征空间中的SVM**
  - 对偶问题优化：

$$\max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^t \phi(\mathbf{x}_j)$$

subject to

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

$$C \geq \alpha_i \geq 0, \quad i = 1, \cdots, m$$

  - 特征空间的权值向量：$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i)$
  - 判别函数：

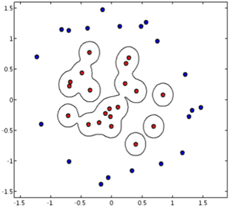
$$f(\mathbf{x}) = \mathbf{w}^t \phi(\mathbf{x}) + b = \sum_{i=1}^{m} \alpha_i \kappa(\mathbf{x}, \mathbf{x}_i) + b$$

非线性SVM的优化求解

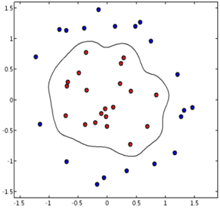
- **非线性SVM的优化求解方法**
  - 非线性SVM只能求解对偶问题，仍然是典型的二次凸优化，存在唯一解；；
  - 与线性SVM的差别仅在于引入了核函数代替内积的计算；
  - Matlab 2019中可以使用fitcsvm求解：

```
svm = fitcsvm( X,Z,'KernelFunction','rbf','KernelScale',0.1,...
      'BoxConstraint',1 );
Label = predict( svm,x );
```

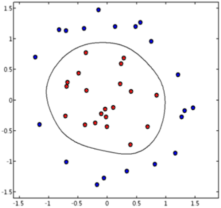
非线性SVM的参数选择：σ



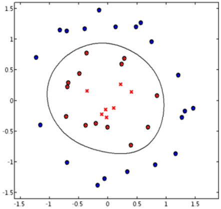
$\sigma = 0.1, C = 1$



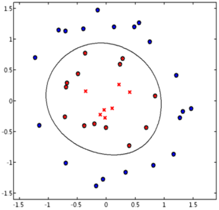
$\sigma = 0.25, C = 1$



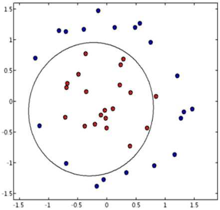
$\sigma = 0.5, C = 1$



$\sigma = 0.75, C = 1$

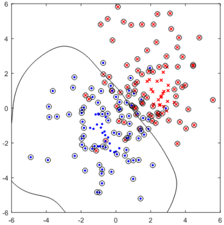


$\sigma = 1, C = 1$

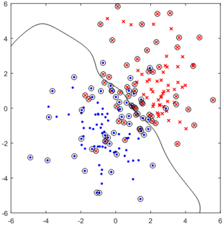


$\sigma = 3, C = 1$

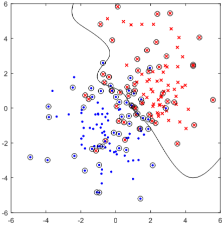
非线性SVM的参数选择：C



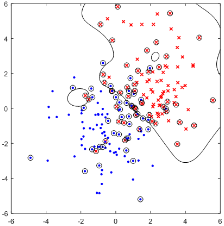
$\sigma = 0.5, C = 0.1$



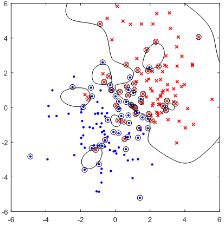
$\sigma = 0.5, C = 1$



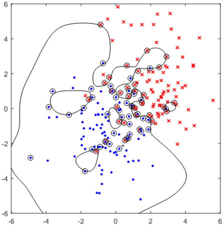
$\sigma = 0.5, C = 10$



$\sigma = 0.5, C = 100$



$\sigma = 0.5, C = 1000$

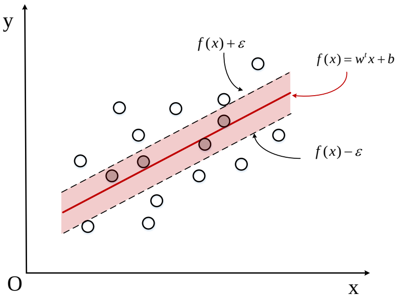


$\sigma = 0.5, C = 10000$

3.5 支持向量机回归

支持向量机回归

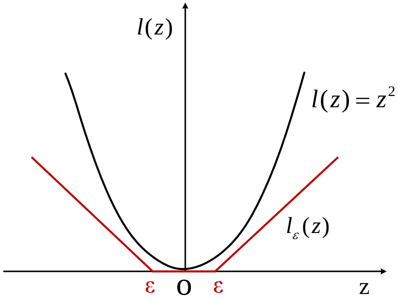
- **基本思路**
  - 在线性回归中使用$\|\mathbf{w}\|^2$作为正则项；
  - 损失函数中，允许预测值$f(\mathbf{x})$与真实之间存在最大ϵ的误差，误差范围之内不计损失；



损失函数

- ϵ-不敏感损失函数

$$l_\epsilon(z) = \begin{cases} 0, & |z| \leq \epsilon \\ |z| - \epsilon, & \text{otherwise} \end{cases}$$



支持向量机回归

- **原始优化问题**
  - 在正负两端引入松弛变量ξ_i和$\hat{\xi}_i$

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \hat{\xi}_i)$$

$subject\ to$

$$\mathbf{w}^t\mathbf{x}_i + b - y_i \leq \epsilon + \xi_i,$$
$$y_i - (\mathbf{w}^t\mathbf{x}_i + b) \leq \epsilon + \hat{\xi}_i,$$
$$\xi_i \geq 0, \quad \hat{\xi}_i \geq 0, \qquad i = 1, \cdots, m$$

支持向量机回归

- **对偶优化问题**
  - 类似于分类问题，由Lagrange乘子法可以得到：

$$\max_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \sum_{i=1}^{m} y_i(\hat{\alpha}_i - \alpha_i) - \sum_{i=1}^{m} \epsilon(\hat{\alpha}_i + \alpha_i) - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\mathbf{x}_i^t\mathbf{x}_j$$

subject to

$$\sum_{i=1}^{m}(\hat{\alpha}_i - \alpha_i) = 0$$

$$C \geq \alpha_i, \hat{\alpha}_i \geq 0, \quad i = 1, \cdots, m$$

支持向量机回归

- **SVR的解**
  - 线性回归的权值向量$\mathbf{w}$：

$$\mathbf{w} = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i)\mathbf{x}_i$$

  - 根据Kuhn-Turker条件，对应$0 < \alpha_i < C$的数据$\mathbf{x}_i$刚好满足$f(\mathbf{x}_i) = y_i + \epsilon$，因此：

$$b = y_i + \epsilon - \sum_{j=1}^{m} (\hat{\alpha}_j - \alpha_j)\mathbf{x}_j^t \mathbf{x}_i$$

  - 引入核函数$\kappa(\cdot,\cdot)$，非线性的SVR回归函数：

$$f(\mathbf{x}) = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i)\kappa(\mathbf{x}, \mathbf{x}_i) + b$$