支持向量机理论简介[1]

Apple Zhang

深圳大学

2021年4月9日

符号定义

- 第 i 个训练样本: $\mathbf{x}_i = [x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(d)}]^T \in \mathbb{R}^d$.
- 第 i 个训练样本的标签: $y_i \in \{-1, 1\}$.
- 训练样本矩阵: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$.
- 训练样本标签向量: $\mathbf{y} = [y_1, y_2, \cdots, y_n] \in \{-1, 1\}^n$.
- 超平面的法向量: $\mathbf{w} \in \mathbb{R}^d$.
- 超平面的偏移: $b \in \mathbb{R}$.
- L_2 范数: $\|\mathbf{x}\|_2$:

$$\|\mathbf{x}\|_2^2 = \sum_{i=1}^{d} x_i^2 = \mathbf{x}^T\mathbf{x}.$$

- 损失函数: $\mathcal{L}(\cdot)$.
- 目标函数: $J(\cdot)$.
- 拉格朗日函数: $L(\cdot)$.
- 函数 $f(x, y)$ 对 x 的梯度: $\nabla_x f$.

什么是超平面?

- 在$\mathbb{R}^2$空间中, 一条直线可以表示为

$$w_1 x^{(1)} + w_2 x^{(2)} + b = 0.$$

- 在$\mathbb{R}^3$空间中, 一个平面可以表示为

$$w_1 x^{(1)} + w_2 x^{(2)} + w_3 x^{(3)} + b = 0.$$

- $\cdots$
- 在$\mathbb{R}^d$空间中, 一个超平面可以表示为

$$w_1 x^{(1)} + w_2 x^{(2)} + \cdots + w_d x^{(d)} + b = 0,$$

即:

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

数据点到超平面的距离

- 在$\mathbb{R}^2$空间中，数据点$\mathbf{x}_i=[x_i^{(1)},x_i^{(2)}]^T$到一条直线的距离为

$$\mathcal{D}=\frac{|w_1x_i^{(1)}+w_2x_i^{(2)}+b|}{\sqrt{w_1^2+w_2^2}}.$$

- 在$\mathbb{R}^3$空间中，数据点$\mathbf{x}_i=[x_i^{(1)},x_i^{(2)},x_i^{(3)}]^T$到一个平面的距离是

$$\mathcal{D}=\frac{|w_1x_i^{(1)}+w_2x_i^{(3)}+w_2x_i^{(3)}+b|}{\sqrt{w_1^2+w_2^2+w_3^2}}.$$

- $\cdots$
- 在$\mathbb{R}^d$空间中，数据点$\mathbf{x}_i=[x_i^{(1)},x_i^{(2)},\cdots,x_i^{(d)}]^T$到超平面的距离是

$$\begin{aligned}\mathcal{D}&=\frac{|w_1x_i^{(1)}+w_2x_i^{(2)}+\cdots+w_dx_i^{(d)}+b|}{\sqrt{w_1^2+w_2^2+\cdots+w_d^2}}\\&=\frac{|\mathbf{w}^T\mathbf{x}_i+b|}{\|\mathbf{w}\|_2}.\end{aligned}$$

线性分类器

问题定义:
对于一个二分类问题，假设数据是线性可分的 (linear separatable), 如何找到一个超平面把两类数据分开?

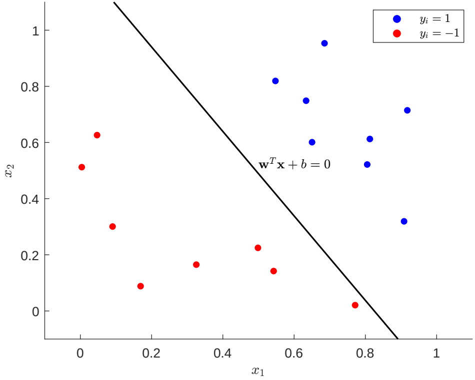


Figure: 二维平面中，用一条直线分隔两个类的实例

关键词:

- 超平面: $\mathbf{w}^T\mathbf{x} + b = 0$;
- 线性模型: $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$;
- 决策/预测函数:
  $h(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x} + b)$;
- 线性可分:
  存在超平面, 对于任意一个训练样本 $\mathbf{x}_i$, 有 $y_i = h(\mathbf{x}_i)$.

深圳大学 SHENZHEN UNIVERSITY

减少错误为目的: 感知机

感知机 (Percetron) 以减少每个样本分类错误为目标.

- 损失函数:

$$\mathcal{L}(\mathbf{x}, y; \mathbf{w}, b) = \max(0, -y(\mathbf{w}^T\mathbf{x} + b)) \tag{1}$$

$$= \begin{cases} -y(\mathbf{w}^T\mathbf{x} + b), & y(\mathbf{w}^T\mathbf{x} + b) < 0 : \text{misclassification}, \\ 0, & y(\mathbf{w}^T\mathbf{x} + b) \geq 0 : \text{correct classification}. \end{cases} \tag{2}$$

- 优化目标:

$$(\hat{\mathbf{w}}, \hat{b}) = \arg\min_{\mathbf{w}, b} J(\mathbf{w}, b), \tag{3}$$

其中, 目标函数 $J(\mathbf{w}, b)$ 定义为:

$$J(\mathbf{w}, b) = \sum_{i=1}^{n} \max(0, -y_i(\mathbf{w}^T\mathbf{x}_i + b)). \tag{4}$$

- 优化方法: 梯度下降.

减少错误为目的: 感知机

梯度下降法: 损失函数梯度: 当 $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 0$ 时, 梯度为 0，因此只有 $y_i(\mathbf{w}^T\mathbf{x}_i + b) < 0$ 时梯度存在:

$$\nabla_{\mathbf{w}}\mathcal{L} = -y_i\mathbf{x}_i, \quad \nabla_b\mathcal{L} = -y_i, \quad \text{only if} : y_i(\mathbf{w}^T\mathbf{x}_i + b) < 0. \tag{5}$$

所以目标函数的梯度:

$$\nabla_{\mathbf{w}}J = \sum_{y_i(\mathbf{w}^T\mathbf{x}_i+b)<0} -y_i\mathbf{x}_i, \tag{6}$$

$$\nabla_b J = \sum_{y_i(\mathbf{w}^T\mathbf{x}_i+b)<0} -y_i. \tag{7}$$

设学习率为 η，则梯度下降迭代式:

$$\mathbf{w} := \mathbf{w} + \sum_{y_i(\mathbf{w}^T\mathbf{x}_i+b)<0} \eta y_i\mathbf{x}_i,$$

$$b := b + \sum_{y_i(\mathbf{w}^T\mathbf{x}_i+b)<0} \eta y_i.$$

减少错误为目的: 感知机

Algorithm 1 Percetron Learning Algorithm

Input: Data $\mathbf{X}, \mathbf{y}$, iteration limit T, learning rate η.
Output: Hyperplane $\mathbf{w}, b$.

Randomly initialize $\mathbf{w}, b$.
for $i = 1$ to T **do**
    Find the set $M = \{(\mathbf{x}_i, y_i):\ y_i(\mathbf{w}^T\mathbf{x}_i + b) < 0\}$.
    if $M = \emptyset$ **then**
        Break.
    end if
    Update $\mathbf{w}, b$: $\mathbf{w} := \mathbf{w} + \sum_{(\mathbf{x}_i, y_i)\in M} \eta y_i \mathbf{x}_i, \quad b := b + \sum_{(\mathbf{x}_i, y_i)\in M} \eta y_i$.
end for
return $\mathbf{w}, b$.

- 虽然有全局最小值，但无法保证全局唯一解;
- 线性不可分时算法不收敛，需特殊处理 (pocket 算法);
- 模型的泛化性能弱.

最大化间隔分类器: 支持向量机

考虑线性可分的数据，支持向量机 (Support vector machine, SVM) 最大化两个类之间的**间隔**.

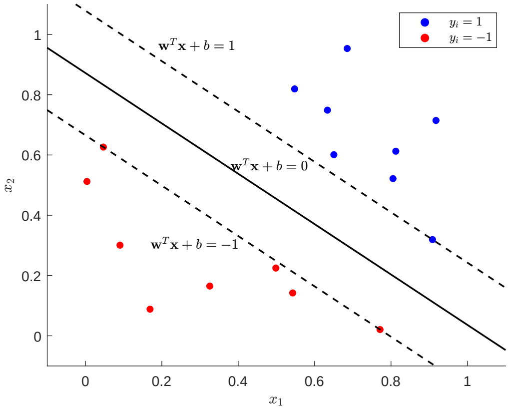


Figure: 硬间隔 SVM 中的分类超平面.

SVM 的优化目标

定义

某一类样本到超平面 $\mathbf{w}^T\mathbf{x}+b=0$ 的间隔,是该类样本到超平面的最小距离:

$$\rho = \min_{i=1,2,\cdots,n} \frac{|\mathbf{w}^T\mathbf{x}_i + b|}{\|\mathbf{w}\|_2} = \min_{i=1,2,\cdots,n} \frac{y_i(\mathbf{w}^T\mathbf{x}_i + b)}{\|\mathbf{w}\|_2}$$

线性可分的情形下，考虑它两侧的超平面 $\mathbf{w}^T\mathbf{x}+b=\pm 1$，满足所有样本都**刚刚好**在这两个超平面以外，即

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq 1, & \text{if } y_i = 1; \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1, & \text{if } y_i = -1. \end{cases} \Leftrightarrow y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1.$$

“刚刚好” 意味着两个类都至少有一个样本满足 $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$，这说明两个类到超平面 $\mathbf{w}^T\mathbf{x}+b=0$ 的间隔都是

$$\rho = \frac{1}{\|\mathbf{w}\|_2}.$$

SVM 的优化目标

因此要最大化两个类之间的间隔，就有

$$\max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|_2}, \quad \text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1. \tag{8}$$

由于:

$$\max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|_2} \Leftrightarrow \min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2.$$

所以最终可以得到一个标准的优化形式

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2, \text{ s.t. } \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1. \tag{9}$$

深圳大学
SHENZHEN UNIVERSITY

SVM 的对偶问题

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2, \text{ s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1.$$

拉格朗日函数:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \alpha_i \left[1 - y_i \left(\mathbf{w}^T\mathbf{x}_i + b\right)\right].$$

定理

上述*SVM*的优化问题等价于下面的形式

$$\max_{\alpha} \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha), \text{ s.t. } \alpha \geq \mathbf{0}.$$

深圳大学 SHENZHEN UNIVERSITY

SVM 的对偶问题

简要证明：

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2, \text{ s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1.$$

我们首先需要构造一个辅助函数$\theta(\mathbf{w}, b)$，且满足

$$\theta(\mathbf{w}, b) = \begin{cases} \frac{1}{2}\|\mathbf{w}\|_2^2 & \forall \mathbf{x}_i,\ 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0, \\ \infty & \exists \mathbf{x}_i,\ 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) > 0. \end{cases}$$

这样构造的原因是:

$$\begin{aligned} &\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2, \text{ s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \\ \Leftrightarrow &\min_{\mathbf{w},b} \theta(\mathbf{w}, b). \end{aligned}$$

由此就消去了优化问题中的不等式约束.

SVM 的对偶问题

$$\theta(\mathbf{w}, b) = \begin{cases} \frac{1}{2}\|\mathbf{w}\|_2^2 & \forall \mathbf{x}_i,\ 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0, \\ \infty & \exists \mathbf{x}_i,\ 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) > 0. \end{cases}$$

接下来将说明:

$$\theta(\mathbf{w}, b) = \max_{\alpha} L(\mathbf{w}, b, \alpha),\ \text{s.t. } \alpha \geq \mathbf{0}.$$

事实上，对于上面这个优化问题，由于

$$h(\mathbf{w}, b) = \max_{\alpha} \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \alpha_i \left[1 - y_i \left(\mathbf{w}^T\mathbf{x}_i + b\right)\right],\ \text{s.t. } \alpha \geq \mathbf{0}.$$

- $\forall \mathbf{x}_i,\ 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0 \Rightarrow h(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|_2^2;$
- $\exists \mathbf{x}_i,\ 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) > 0 \Rightarrow h(\mathbf{w}, b) = \infty.$

因此, $\theta(\mathbf{w}, b) = h(\mathbf{w}, b)$.

SVM 的对偶问题

记录一下, 我们现在得到了:

$$
\begin{aligned}
& \min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2, \text{ s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \\
\Leftrightarrow & \min_{\mathbf{w},b} \theta(\mathbf{w}, b) \\
\Leftrightarrow & \min_{\mathbf{w},b} \max_{\alpha} L(\mathbf{w}, b, \alpha), \text{ s.t. } \alpha \geq \mathbf{0}.
\end{aligned}
$$

事实上，我们可以做进一步化简

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

这个性质称为强对偶性, 一般的函数只满足弱对偶性, 即中间是小于等于号. SVM 拥有这个性质是因为它是凸优化问题，且满足 slater 条件.

SVM 的对偶问题

定理* (Slater 条件)

设定义在 $\mathcal{D}$ 上的函数 $f_i(\cdot), i=1,2,\cdots,n$ 为凸函数， $g_j(\cdot), j=1,2,\cdots,m$ 为仿射函数，考虑凸优化问题

$$\min_{\mathbf{x}} f_0(\mathbf{x}), \quad \text{s.t. } f_i(\mathbf{x}) \leq 0, g_i(\mathbf{x}) \leq 0. \tag{10}$$

如果存在点 $\mathbf{x} \in \operatorname{relint} \mathcal{D}$（即 $\mathcal{D}$ 的相对内点），则强对偶性成立.

SVM 的对偶问题

$$\max_{\alpha\geq\mathbf{0}}\min_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|_2^2+\sum_{i=1}^{n}\alpha_i\left[1-y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)\right].$$

首先对于$\mathbf{w}, b$最小化$L(\mathbf{w}, b, \alpha)$:

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

$$\frac{\partial L}{\partial b}=0\Rightarrow 0=\sum_{i=1}^{n}\alpha_i y_i.$$

回代，化简得到:

$$\max_{\alpha\geq\mathbf{0}}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i y_i\alpha_j y_j\mathbf{x}_i^T\mathbf{x}_j,\quad \text{s.t.}\ \sum_{i=1}^{n}\alpha_i y_i=0$$

SVM 的对偶问题

最后我们得到了一个二次规划问题：

$$\begin{aligned} &\max_{\alpha} \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\mathbf{Q}\alpha, \\ &\text{s.t. } \alpha \geq \mathbf{0},\ \mathbf{y}^T\alpha = 0. \end{aligned} \tag{11}$$

这里1是全1的列向量, $\mathbf{Q} \in \mathbb{R}^{n\times n}$的每一个元素是

$$\mathbf{Q}_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j.$$

实际实现的过程中，可以调用 Matlab 的函数: `quadprog` 求解二次规划问题[1]，对于 Python，可以使用 `cvxopt` 模块的 `solvers.qp` 函数[2].
最后需要说明，我们可以把预测函数写为下面的形式 (为什么要这么写?).

$$h(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}^T\mathbf{x} + b) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b\right)$$

[1] 需安装 Matlab Optimization toolbox.
[2] 需安装 numpy-mkl.

深圳大学 SHENZHEN UNIVERSITY

小贴士

$$\max_{\alpha} \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\mathbf{Q}\alpha,$$
$$\text{s.t. } \alpha \geq \mathbf{0},\ \mathbf{y}^T\alpha = 0.$$

- 上式解出的α是稀疏的. 其中对应的训练样本$\mathbf{x}_i$称为支持向量(**Support Vector**, SV), 如果它对应的$\alpha_i > 0$.
- b 可以通过支持向量来计算:

$$b = \frac{1}{\|\alpha\|_0}\sum_{\alpha_i>0}\left(y_i - \sum_{j=1}^{n}\alpha_j y_j \mathbf{x}_j^T\mathbf{x}_i\right)$$

其中$\|\cdot\|_0$表示L_0范数算子, 返回向量非零元素个数.

小贴士

$$\max_{\alpha} \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\mathbf{Q}\alpha,$$
$$\text{s.t. } \alpha \geq \mathbf{0},\ \mathbf{y}^T\alpha = 0. \tag{12}$$

Matlab函数：`x = quadprog(H, f, A, b, Aeq, Beq, lb, ub, x0);`

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{f}^T\mathbf{x},\ \text{s.t.} \begin{cases} \mathbf{A}\cdot\mathbf{x} \leq \mathbf{b} \\ \mathbf{Aeq}\cdot\mathbf{x} = \mathbf{beq} \\ lb \leq \mathbf{x} \leq ub \end{cases}$$

因此解SVM的二次规划问题可以按以下方式调用：

$$\texttt{alpha = quadprog}(\mathbf{Q},\ -\mathbf{1},\ \texttt{[]},\ \texttt{[]},\ \mathbf{y}^T,\ 0,\ \mathbf{0},\ \texttt{[]},\ \mathbf{x}_0)$$

注意：需要安装Matlab Optimization toolbox!

深圳大学 SHENZHEN UNIVERSITY

也许可以犯错: 软间隔 SVM

硬间隔SVM 的假设基于所有数据**线性可分**，大多数情况并不满足.
软间隔SVM 则将硬间隔 SVM 推广到更一般的形式.

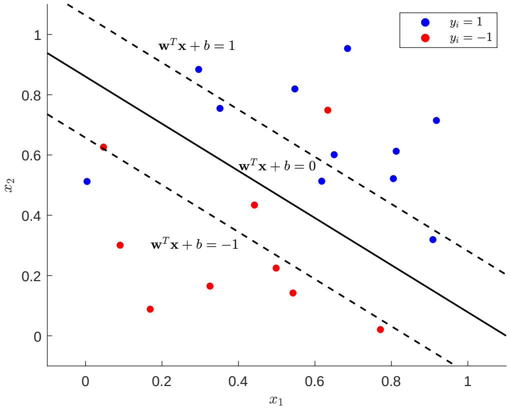


Figure: 软间隔 SVM 在线性不可分数据上的分类示意图.

也许可以犯错: 软间隔 SVM

软间隔 SVM 模型:

$$\min_{\mathbf{w},b} \quad \underbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}_{L_2\text{-regularization}} \quad + \quad \underbrace{c\sum_{i=1}^{n}\mathcal{L}_{\text{hinge}}(y_i(\mathbf{w}^T\mathbf{x}_i+b))}_{\text{Empirical risk}}. \tag{13}$$

此处 c 是惩罚参数, $\mathcal{L}_{\text{hinge}}(\cdot)$称为**铰链损失**, 其定义为

$$\mathcal{L}_{\text{hinge}}(z) = \max(0, 1-z).$$

可以看到这个优化问题具有最小化 “正则+经验风险” 的形式，这就是**最小化结构风险 (Structral Risk Minimization, SRM)** 的典型代表.

深圳大学 SHENZHEN UNIVERSITY

也许可以犯错: 软间隔 SVM

引入松弛变量 ξ_i，可以将原优化目标重写为:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|_2^2 + c\sum_{i=1}^{n}\xi_i,$$
$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0$$

其对应的拉格朗日函数:

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|_2^2 + c\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\left[\alpha_i\left(1 - \xi_i - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)\right] - \sum_{i=1}^{n}\beta_i\xi_i$$

对偶问题:

$$\max_{\alpha} \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\mathbf{Q}\alpha,$$
$$\text{s.t. } \mathbf{0} \leq \alpha \leq c\mathbf{1},\ \mathbf{y}^T\alpha = 0.$$

可以发现, 硬间隔 SVM 就是 $c \to \infty$ 的特殊情况.

软间隔 SVM 的泛化性能

定理 (SVM 的泛化性[2])

对于软间隔 *SVM*，其中 $\|\mathbf{w}\|_2 \leq \Lambda$，且任意一个训练数据 $\mathbf{x}_i$ 有 $\|\mathbf{x}_i\|_2 \leq r$，则下式以至少 $1-\delta\ (0<\delta<1)$ 的概率成立:

$$\textit{Generalization error} \leq \frac{1}{n}\sum_{i=1}^{n}\xi_i + 2\sqrt{\frac{r^2\Lambda^2}{n}} + 3\sqrt{\frac{-\ln\delta}{2n}}. \tag{14}$$

这个定理给出了一个泛化误差的上界:

- 当样本本量 n 增大时，泛化误差的上界变小.
- 软间隔 SVM 的泛化误差的上界与特征维数 d 没有直接的依赖关系.
- 软间隔 SVM 最小化结构风险恰好具有最小化该上界的形式.

深圳大学 SHENZHEN UNIVERSITY

非线性数据: 核 SVM

前面的讨论都使用的是纯线性模型. 如果数据是非线性的, 效果将明显变差.
核 SVM: 借助核方法完成非线性分类.

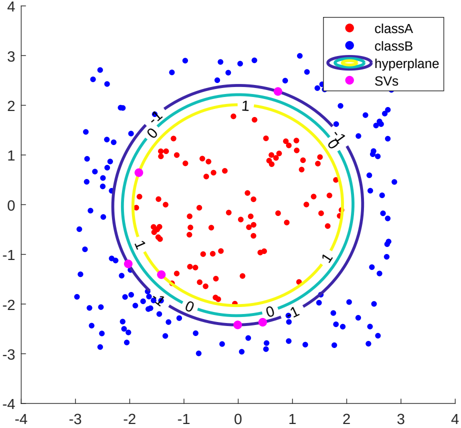
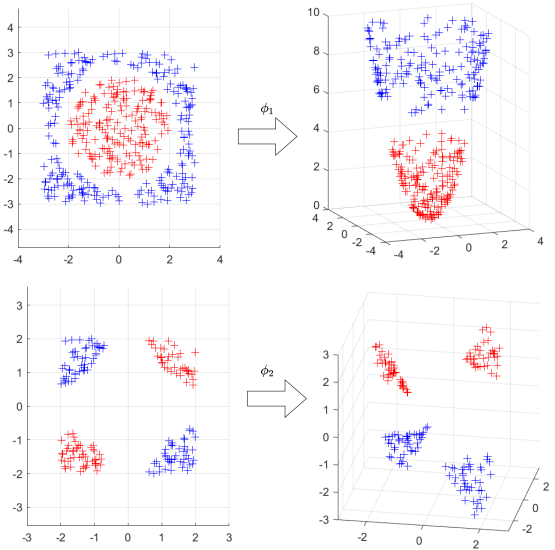


Figure: 核 SVM 对非线性的数据进行分类.

非线性数据: 核 SVM



非线性数据: 核 SVM

定义

函数 $\mathcal{K}:\mathbb{R}^d\times\mathbb{R}^d\mapsto\mathbb{R}$ 定义为在 $\mathbb{R}^d$ 上的核.

这里我们主要讨论 **正定对称核**，对于正定对称核，存在唯一的特征映射 $\phi:\mathbb{R}^d\mapsto\mathcal{H}$，使得

$$\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)=\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle=\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j). \tag{15}$$

其中，$\mathcal{H}$ 是一个特征空间. 核函数 $\mathcal{K}$ 是正定对称核的充要条件是满足 **Mercer 条件**.

- 正定对称核可以唯一确定一个非线性特征映射.
- 计算特征空间的内积时，利用核函数可以降低计算量.
- 通过核函数完成的非线性映射是隐式的，不需要知道特征映射 $\phi(\cdot)$ 的具体形式.

非线性数据: 核 SVM

对于一个由正定对称核确定的特征映射:

$$\phi(\mathbf{x}):\ \mathbb{R}^d \mapsto \mathcal{H}.$$

在映射后的特征空间 $\mathcal{H}$ 中，使用软间隔 SVM，此时所求的超平面变为:

$$\mathbf{w}^T\phi(\mathbf{x}) + b = 0.$$

而且$\mathbf{w} \in \mathcal{H}$，因此我们可以写出新的优化问题

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_{\mathcal{H}}^2 + c\sum_{i=1}^{n}\xi_i,$$
$$\text{s.t. } y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \xi_i \geq 0.$$

非线性数据: 核 SVM

拉格朗日函数:

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|_{\mathcal{H}}^2 + c\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\left[\alpha_i\left(1 - \xi_i - y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b)\right)\right] - \sum_{i=1}^{n}\beta_i\xi_i$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \phi(\mathbf{x}_i), \quad \frac{\partial L}{\partial b} = 0 \Rightarrow 0 = \sum_{i=1} S^n \alpha_i y_i,$$

$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow 0 = \sum_{i=1}^{n}(c\xi_i - \alpha_i - \beta_i).$$

非线性数据: 核 SVM

对偶问题:

$$\max_{\alpha} \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\mathbf{Q}^{\Phi}\alpha,$$
$$\text{s.t. } \mathbf{0} \leq \alpha \leq c\mathbf{1},\ \mathbf{y}^T\alpha = 0,$$

其中$\mathbf{Q}^{\Phi} \in \mathbb{R}^{n\times n}$的每一个元素是

$$\mathbf{Q}^{\Phi}_{ij} = y_i y_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) = y_i y_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j).$$

需要注意，**核 SVM 的决策函数:**

$$h(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}^T\phi(\mathbf{x}) + b) = \operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i \mathcal{K}(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{16}$$

正定对称核函数的实例: **径向基核 (Radial Basis Function Kernel)**

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right).$$

更多有关SVM...

高效的SVM解决方案(C/C++实现): **LIBSVM, LIBLINEAR**[3, 4].

- 已有C++, Java, MATLAB, Python接口.
- 贼快.

其他SVM的变种:

- (1999) Least square support vector machine (LSSVM) [5].
- (2003 **NIPS**) L_1-regularized support vector machine (L_1-SVM) [6].
- (2006 **JMLR**) Laplacian support vector machine [7].
- (2007 **TPAMI**) Twin support vector machine (TWSVM) [8].
- (2011 **TNNLS**) Twin bound support vector machine (TBSVM) [9].
- (2012 **Neural Networks**) Laplacian twin support vector machine [10].
- (2019 **Neural Networks**) Robust twin support vector machine [11].
- (2021 **AAAI**) Hash (binary embedding) + kernel SVM [12].

结束了!!!!!!!

谢谢!

参考文献

[1] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*.
The MIT Press, 2012.

[3] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBLINEAR: A library for large linear classification,” *Journal of Machine Learning Research*, vol. 9, p. 1871–1874, 2008.

[5] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[6] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, “1-norm support vector machines,” in *Proceedings of the 16th International Conference on Neural Information Processing Systems*, p. 49–56, MIT Press, 2003.

[7] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *Journal of Machine Learning Research*, vol. 7, no. 11, pp. 2399–2434, 2006.

[8] Jayadeva, R. Khemchandani, and S. Chandra, “Twin support vector machines for pattern classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, 2007.

[9] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, and N.-Y. Deng, “Improvements on twin support vector machines,” *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 962–968, 2011.

[10] Z. Qi, Y. Tian, and Y. Shi, “Laplacian twin support vector machine for semi-supervised classification,” *Neural Networks*, vol. 35, pp. 46–53, 2012.

[11] C. Wang, Q. Ye, P. Luo, N. Ye, and L. Fu, “Robust capped l1-norm twin support vector machine,” *Neural Networks*, vol. 114, pp. 47–59, 2019.

[12] Z. Lei and L. Lan, “Memory and computation-efficient kernel SVM via binary embedding and ternary model coefficients,” 2020.