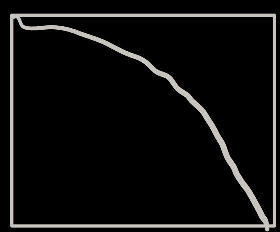
真正例率 $= \dfrac{TP}{TP+FN}$

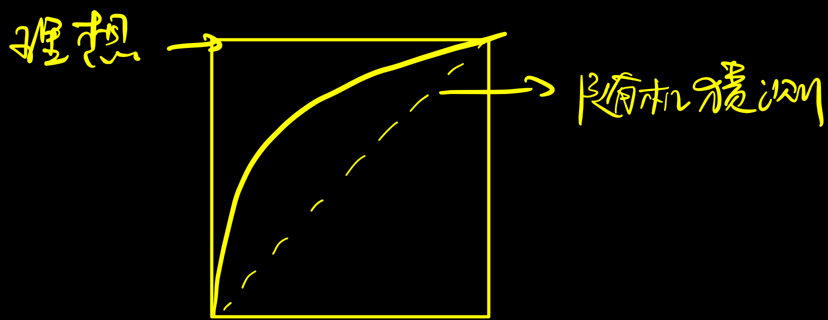
假正例率 $= \dfrac{FP}{FP+TN}$

P-R曲线：查准率为x，查全率为y

面积、和平衡点（P=R时R值）



ROC曲线：假正例率为x，真正例率为y



F1-score：R和P的调和平均

偏差：算法预测与真实结果偏离程度

欠拟合 = 高偏差

应对：引入更多特征减弱正则化

方差：数据扰动对学习性能的影响

高方差 = 过拟合

应对：增加样本，加强正则化

SVM：点 X 到超平面 (w, b) 距离：$\frac{|w^T x + b|}{\|w\|}$

$y(w^T x + b) \geq 1 \longrightarrow$ 最小化 $\|w\|$

由此我们可以得到平面$W^TX+b=1$和$W^TX+b=-1$之间的距离为

$$\gamma = \frac{2}{\|w\|}$$

不等式约束下最小化问题

$$L(X,\lambda,\mu) = f(X) + \sum_{j=1}^{m} \lambda_j h_j(X) + \sum_{k=1}^{n} \mu_k g_k(X)$$

对X求梯度

由不等式约束引入的 KKT 条件 $(j = 1, 2, \ldots, n)$为:

$$\begin{cases} \nabla f + \sum_{j=1}^{m} \lambda_j \nabla h_j(\boldsymbol{x}) + \sum_{k=1}^{n} \mu_k \nabla g_k(\boldsymbol{x}) = 0 \\ h_j(\boldsymbol{x}) = 0, i = 1, 2, \ldots, m \\ g_k(\boldsymbol{x}) \leqslant 0, j = 1, 2, \ldots, n \\ \mu_k \geqslant 0 \\ \mu_k g_k(\boldsymbol{x}) = 0. \end{cases}$$

无约束 直接求目标函数得0的点。如果没有解析解的话，可以使用梯度下降或牛顿方法等迭代的手段来使X沿负梯度方向逐步逼近极小值点。

$$\min_{x} f(x) \qquad \nabla_x f(x) = 0$$

等式约束:可以直接应用拉格朗日乘子法去求取最优值

不等式约束:可以转化为在满足 KKT 约束条件下应用拉格朗日乘子法求解。

拉格朗日求得的并不一定是最优解，只有在凸优化的情况下，才能保证得到的是**最优解.**

对偶求SVM：

$$L(w, b, \beta) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i(w^T x_i + b))$$

$\Downarrow$ 对 w, b 求偏导 $= 0$

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

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

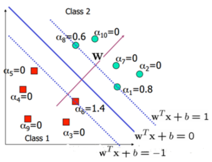
$$\sum_{i=1}^{m} \alpha_i y_i x_i^T x + b$$

- 最终模型：$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = \boxed{\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x}} + b$

- KKT条件：（满足此条件才能使用拉格朗日乘子法）

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\boldsymbol{x}_i) \geq 1, \\ \boxed{\alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0.} \end{cases}$$

$$\underset{\boldsymbol{w},b}{\arg\min} \ \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t.} \ y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i = 1, 2, \ldots, m.$$



$y_i f(\boldsymbol{x}_i) > 1$ ➡ $\alpha_i = 0$ 该样本对f(x)没有影响

$y_i f(\boldsymbol{x}_i) - 1 = 0$ ➡ 该样本位于边界，为支持向量

支持向量机解的稀疏性：训练完成后，
大部分的训练样本都不需保留，
最终模型仅与支持向量有关.

例：正例点：(3,3)，(4,3)

负例点：(1,1)

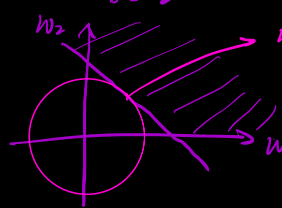
求超平面

最优化问题：$\min(\frac{1}{2}w_1^2 + \frac{1}{2}w_2^2)$

$\text{s.t.} \ y_i(w^T x_i + b) \geq 1$

$$\Rightarrow \begin{cases} 3w_1 + 3w_2 + b \geq 1 \\ 4w_1 + 3w_2 + b \geq 1 \\ w_1 + w_2 + b \leq -1 \end{cases}$$

$$\Rightarrow \begin{cases} w_1 \geq 0 \\ w_1 + w_2 \geq 1 \\ b \leq -2 \end{cases}$$



故 $w_1 = w_2 = \frac{1}{2}$，$b = -2$

法二. s.t. ⇒

$$\begin{cases} 1 - 3w_1 - 3w_2 - b \leq 0 \\ 1 - 4w_1 - 3w_2 - b \leq 0 \\ 1 + w_1 + w_2 + b \leq 0 \end{cases}$$

（不等式约束 g）

$$\Downarrow$$

$$L(X, \lambda, \mu) = f(X) + \sum_{j=1}^{m} \lambda_j h_j(X) + \sum_{k=1}^{n} \mu_k g_k(X)$$

对X求梯度

由不等式约束引入的 KKT 条件 $(j = 1, 2, \ldots, n)$ 为：

$$\begin{cases} \nabla f + \sum_{j=1}^{m} \lambda_j \nabla h_j(\boldsymbol{x}) + \sum_{k=1}^{n} \mu_k \nabla g_k(\boldsymbol{x}) = 0 \\ h_j(\boldsymbol{x}) = 0, i = 1, 2, \ldots, m \\ g_k(\boldsymbol{x}) \leqslant 0, j = 1, 2, \ldots, n \\ \mu_k \geqslant 0 \\ \mu_k g_k(\boldsymbol{x}) = 0. \end{cases}$$

$$L = \frac{1}{2}(w_1^2 + w_2^2) + \mu_1(1 - 3w_1 - 3w_2 - b) + \mu_2(1 - 4w_1 - 3w_2 - b) + \mu_3(1 + w_1 + w_2 + b)$$

$$\text{s.t.} \begin{cases} \mu_1(1 - 3w_1 - 3w_2 - b) = 0 \\ \mu_2(1 - 4w_1 - 3w_2 - b) = 0 \\ \mu_3(1 + w_1 + w_2 + b) = 0 \end{cases}$$

$$\begin{cases} \frac{\partial L}{\partial w_1} = w_1 - 3\mu_1 - 4\mu_2 + \mu_3 = 0 \\ \frac{\partial L}{\partial w_2} = w_2 - 3\mu_1 - 3\mu_2 + \mu_3 = 0 \\ \frac{\partial L}{\partial b} = -\mu_1 - \mu_2 + \mu_3 = 0 \end{cases}$$

$$\Downarrow$$

$w_1 = w_2 = \frac{1}{2}$，$b = -2$

核函数

核函数的基本作用就是接受两个低维空间里的向量，能够计算出经过某个变换后在高维空间里的向量内积值。

核支持向量机

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{n} \alpha_i$$

$$\text{s.t.} \ \sum_{i=1}^{n} \alpha_i y_i = 0 \quad (\alpha_i \geq 0, i = 1, 2.., n)$$

- 设样本 映射后的向量为 ，划分超平面为 .

原始问题

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

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

对偶问题

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boxed{\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)} - \sum_{i=1}^{m} \alpha_i$$

$$\text{s.t.} \ \sum_{i=1}^{m} \alpha_i y_i = 0,$$ 只以内积的形式出现

预测

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \boxed{\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x})} + b$$

软间隔

软间隔支持向量机

原始问题 $\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\max\left(0, 1 - y_i(\boldsymbol{w}^\top\phi(\boldsymbol{x}_i)+b)\right)$

引入松弛变量ξi≥0，可写成

$$\min_{[w,b]} \frac{1}{2}||w||^2 + C\sum \xi_i$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} l_{0/1}\left(y_i(\boldsymbol{w}^\top\phi(\boldsymbol{x}_i)+b)-1\right)$$

其中$l_{0/1}$ 是"0/1损失函数"

$$l_{0/1} = \begin{cases} 1 & z < 0 \\ 0 & otherwise \end{cases}$$

正则化

就是说给损失函数加上一些限制，通过这种规则去规范他们再接下来的循环迭代中，不要自我膨胀。

SVM

① 原理：

样本可分 — 硬间隔

" 近似可分 — 引入松弛变量，软间隔

" 不可分 — 核技巧 + 软间隔，非线性 SVM

② 为什么间隔最大化：

鲁棒性好

③ 为什么用对偶问题？

1. 易求解

2. 核引入

④ 优：

解决高维特征的分类回归问题

解稀疏，无需依赖全部数据

样本可较少

缺：

特征维度 >> 样本数，表现不好

样本量大时，映射维度高，计算量大

核函数难选

缺失敏感

决策树

基本流程

Algorithm 1 决策树学习基本算法

输入:

- 训练集 $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$;
- 属性集 $A = \{a_1, \ldots, a_d\}$.

过程: 函数 TreeGenerate(D, A)

```
生成结点 node;
if D 中样本全属于同一类别 C then
    将 node 标记为 C 类叶结点; return
end if
if A = ∅ OR D 中样本在 A 上取值相同 then
    将 node 标记叶结点，其类别标记为 D 中样本数最多的类; return
end if
从 A 中选择最优划分属性 a*;
for a* 的每一个值 a*^v do
    为 node 生成每一个分枝; 令 D_v 表示 D 中在 a* 上取值为 a*^v 的样本子集;
    if D_v 为空 then
        将分枝结点标记为叶结点, 其类别标记为 D 中样本最多的类; return
    else
        以 TreeGenerate(D_v, A − {a*}) 为分枝结点
    end if
end for
```

输出: 以 node 为根结点的一棵决策树

（1）当前结点包含的样本全部属于同一类别

（2）当前属性集为空，或所有样本在所有属性上取值相同

（3）当前结点包含的样本集合为空

信息熵 $Ent(D) = -\sum_{k=1}^{|y|} p_k \log p_k$

信息增益 $G(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$

增益率 $G\_r(D,a) = \frac{G(D,a)}{-\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log \frac{|D^v|}{|D|}}$

	优	缺
预剪枝	减小时间开销	欠拟合
后剪枝	减小欠拟合	时间开销大

连续值处理：

① 属性a取值从小到大排列：

$$a^1 \quad a^2 \quad a^3 \quad \dots \quad a^n$$

取划分点

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \;\middle|\; 1 \le i \le n-1 \right\}$$

② 选取最优划分点

$$\begin{aligned} Gain(D,a) &= \max_{t \in T_a} Gain(D,a,t) \\ &= \max_{t \in T_a} Ent(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda) \end{aligned}$$

缺失值处理

让同一样本以不同概率划入不同子结点中

贝叶斯分类器：

$$\frac{P(c)\,P(x|c)}{\quad} = \frac{P(c)}{\quad} \prod_{i=1}^{d} P(x_i|c)$$

$$P(c|x) = \frac{\quad}{p(x)} = \frac{\quad}{p(x)} \prod_{i=1}$$

$$h_{nb}(x) = \arg\max_{c\in y} \underbrace{P(c)}_{\frac{|D_c|}{|D|}} \prod_{i=1}^{d} \underbrace{P(x_i|c)}$$

离散属性：

$$\frac{|D_{c,x_i}|}{|D_c|}$$

连续属性：

$\mu_{c,i}$, $\sigma_{c,i}$: c类样本在第i属性上均值和方差

$$\frac{1}{\sqrt{2\pi}\,\sigma_{c,i}} e^{-\frac{(x_i-\mu_{c,i})^2}{2\sigma_{c,i}^2}}$$

避免数据被"抹去"：

拉普拉斯修正：

$$\hat{P}(c) = \frac{|D_c|+1}{|D|+N}$$

$$\hat{P}(x_i|c) = \frac{|D_{c,x_i}|+1}{|D_c|+N}$$

（N：D中类别数；

N_i：第i个属性取值数）