> Author By quinwu

最大间隔超平面

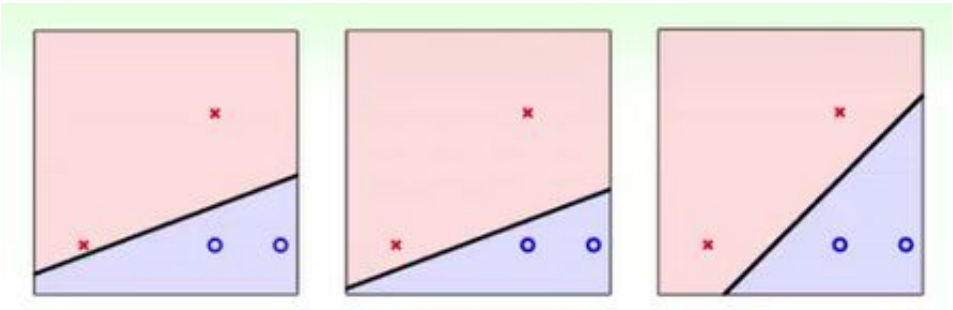
为何要使得间隔最大?

这里假设我们的训练数据**存在Gaussian-like的误差**

感知机和支持向量是一种东西吗?

最大间隔超平面的思想是让 SVM 最求的是**一个固定的、间隔最大的线（或超平面）**

一般的感知机算法的分割线却是随机的



凸二次规划

Convex Quadratic Programming 是一类典型的优化问题，目标函数是变量的二次函数，而约束条件是变量的线性不等式。

假设变量个数为d，约束条件的个数为m，则标准的二次规划问题为

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}x^T Q x + c^T x \\ s.t. \quad & Ax \leq b \end{aligned}$$

其中 x为d维向量，Q为实对称矩阵，A为实矩阵，b和c为实向量，约束条件每一行对应一个约束。

若Q为半正定矩阵，则上式目标函数是凸函数，相应的二次规划问题是凸二次优化问题。此时若约束条件定义的可行域不为空，且目标函数在此可行域有下界，则该问题将有全局最小值。若Q为正定矩阵，则该问题有唯一的全局最小值。若Q为非正定矩阵，则目标函数有多个平稳点跟局部极小点的NP Hard问题。

拉格朗日对偶

为什么SVM要做一次拉格朗日对偶?

目前处理的模型严重依赖与数据集的维度D，维度太高会严重的提升运算时间。拉格朗日对偶事实上把SVM从**依赖d个维度**转变到**依赖N个数据点**。考虑到最后计算时只有支持向量才是有意义的，这个实际上的计算量比N要小很多。

这里有没有发现加上拉格朗日条件的SVM非常像**正则化**?

拉格朗日函数

对于优化问题

$$\begin{aligned} &\min_u f(u) \\ s.t. \quad &g_i(u) \leq 0, \quad i = 1, 2, \dots, m \\ &h_j(u) = 0, \quad j = 1, 2, \dots, n \end{aligned}$$

定义其拉格朗日函数为

$$\mathcal{L}(u, \alpha, \beta) = f(u) + \sum_{i=1}^{m} \alpha_i g_i(u) + \sum_{j=1}^{n} \beta_j h_j(u)$$

其中 $\alpha_i \geq 0$

引入引理:

上述拉格朗日描述的优化问题等价于

$$\begin{aligned} &\min_u \max_{\alpha,\beta} \mathcal{L}(u, \alpha, \beta) \\ s.t. \quad &\alpha_i \geq 0, \quad i = 1, 2, \dots, m \end{aligned}$$

> 证明.
>
> $$\begin{aligned} &\min_u \max_{\alpha,\beta} \mathcal{L}(u, \alpha, \beta) \\ &= \min_u (f(u) + \max_{\alpha,\beta} (\sum_{i=1}^{m} \alpha_i g_i(u) + \sum_{j=1}^{n} \beta_i h_j(u))) \\ &= \min_u (f(u) + \begin{cases} 0 & \text{若}u\text{满足约束} \\ \infty & \text{否则} \end{cases}) \\ &= \min_u f(u), \quad \text{且}u\text{满足约束} \end{aligned}$$
>
> 其中, 当 g_i不满足约束时, 即$g_i(u) > 0$, 我们可以取$\alpha_i = \infty$, 使得 $\alpha_i g_i(u) = \infty$;当 h_j 不满足约束时, 即 $h_j(u) \neq 0$,我们可以取 $\beta_j = sign(h_j(u))\infty$, 使得 $\beta_j h_j(u) = \infty$. 当u 满足约束时, 由于 $\alpha_i \geq 0$,$g_i(u) \leq 0$, 则 $\alpha_i g_i(u) \leq 0$. 因此$\alpha_i g_i(u)$ 最大值为 0.

KKT条件

优化问题在最优值处必须满足如下条件

- 主问题可行：$g_i(u) \leq 0, h_i(u) = 0$;
- 对偶问题可行：$\alpha_i \geq 0$
- 互补松弛：$\alpha_i g_i(u) = 0$

> 证明.

> 由上述引理可知，u必须满足约束，即主问题可行.
>
> 对偶问题可行是上述优化问题的约束项.
>
> $a_i g_i(u) = 0$ 是在主问题和对偶问题都可行的条件下的最大值.

对偶问题

优化问题的对偶问题为：

$$\max_{\alpha,\beta} \min_{u} \mathcal{L}(u, \alpha, \beta)$$
$$s.t. \quad \alpha_i \geq 0, \quad i = 1, 2, \ldots, m$$

对偶问题是主问题的下界，即

$$\max_{\alpha,\beta} \min_{u} \mathcal{L}(u, \alpha, \beta) \leq \min_{u} \max_{\alpha,\beta} \mathcal{L}(u, \alpha, \beta)$$

如何理解?

这个不等式叫**弱对偶性质（Week Duality）**，最大值中最小的一个，也要大于等于最小值中最大的一个。

同时，我们可以得到一个**对偶间隙**，即$p^* - d^*$

话说回来，如果我们这里恰好能够取等号，即对偶间隙消失就好了。

在凸优化理论中，有一个Slater定理，当这个定理满足，那么对偶间隙就会消失，即

$$d^* = p^*$$

此时称为**强对偶性质（strong Duality）**。幸运的是，我们这里满足Slater定理。

Slater 条件

当主问题为凸优化问题时，即 f 和 g_i 为凸函数，h_j为仿射函数，且可行域中至少有一点使得不等式约束条件严格成立时，对偶问题等价于原问题。

> 证明.
>
> 略
>
> 推论：线性支持向量机满足Slater条件

Slater定理其实就是说，存在**x**，使不等式约束中的**小于等于号**要严格取到**小于号**。

线性支持向量机对偶型

线性支持向量机的拉格朗日函数为

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

等价于

$$\min_{w,b}\max_{\alpha} \quad \frac{1}{2}w^Tw+\sum_{i=1}^{m}\alpha_i(1-y_i(w^Tx_i+b)).$$
$$s.t. \quad \alpha_i\geq 0, \quad i=1,2,\dots,m.$$

其对偶问题为

$$\max_{\alpha}\min_{w,b} \quad \frac{1}{2}w^Tw+\sum_{i=1}^{m}\alpha_i(1-y_i(w^Tx_i+b)).$$
$$s.t. \quad \alpha_i\geq 0, \quad i=1,2,\dots,m.$$

因此，线性支持向量机的对偶问题等价于找到一组合适的参数α，使得

$$\begin{aligned}\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_jy_iy_j\boldsymbol{x}_i^T\boldsymbol{x}_j-\sum_{i=1}^{m}\alpha_i\\ s.t. \quad & \sum_{i=1}^{m}\alpha_iy_i=0,\\ & \alpha_i\geq 0, \quad i=1,2,\dots,m.\end{aligned}$$

> 证明
>
> $$\frac{\partial\mathcal{L}}{\partial w}=0 \quad \Rightarrow \quad w=\sum_{i=1}^{m}\alpha_iy_i\boldsymbol{x}_i$$
> $$\frac{\partial\mathcal{L}}{\partial b}=0 \quad \Rightarrow \sum_{i=1}^{m}\alpha_iy_i=0$$
>
> 带入可得上式

线性支持向量机对偶型中描述的优化问题属于二次规划问题，包括m个优化变量，$m+2$项约束条件

支持向量

线性支持向量机的KKT条件

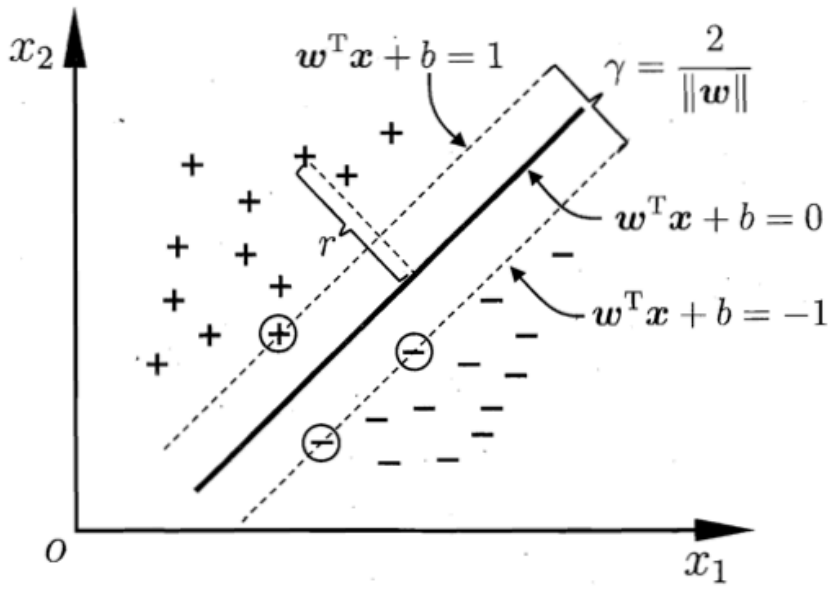
- 主问题可行：$1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\leq 0$
- 对偶问题可行：$\alpha_i\geq 0$
- 互补松弛：$\alpha_i(1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b))=0$

> （支持向量）. 对偶变量$\alpha_i>0$对应的样本

线性支持向量机中，支持向量是距离划分超平面最近的样本，落在最大间隔边界上。

> 由线性支持向量机的KKT条件可知，$\alpha_i(1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b))=0$.当$\alpha_i>0$时，
> $1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)=0$.即$1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)=1$.

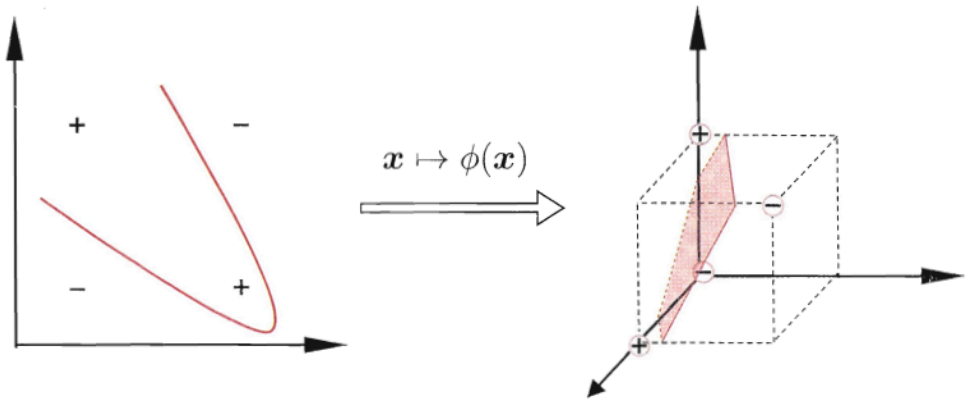
支持向量机的参数(w,b)仅由支持向量决定，与其他样本无关.



> 对偶变量 $\alpha_i > 0$对应的样本是支持向量
>
> $$\begin{aligned} w &= \sum_{i-1}^{m} \alpha_i y_i \boldsymbol{x}_i \\ &= \sum_{i:\alpha_i=0}^{m} 0 \cdot y_i \boldsymbol{x}_i + \sum_{i:\alpha_i>0}^{m} \alpha_i y_i \boldsymbol{x}_i \\ &= \sum_{i \in SV} \alpha_i y_i \boldsymbol{x}_i \end{aligned}$$
>
> 其中SV代表所有支持向量的集合.b可以由互补松弛算出，对于某一支持向量 $\boldsymbol{x}_s$及其标记y_s，由于$y_s(\boldsymbol{m}^T \boldsymbol{x}_s + b) = 1$,则
>
> $$b = y_s - \boldsymbol{w}^T \boldsymbol{x}_s = y_s - \sum_{i \in SV} \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x}_s$$
>
> 实践中，为了得到对b 更稳健的估值，通常使用对所有的支持向量求解得到b的平均值.

Kernel 核技巧

当假设训练数据线性可分时，即存在一个划分超平面能将属于不同标记的训练样本分开.在很多任务重，这样超平面是不存在的.支持向量机通过核技巧来解决样本不是线性可分的情况。



核函数的作用是什么?

核的作用一方面让SVM可以在高维上对数据集线性可分，另一方面是提高对内积的计算速度。

非线性可分

令 $\phi(\boldsymbol{x})$代表将样本$\boldsymbol{x}$映射到 $\mathbb{R}^{\tilde{d}}$ 中的特征向量，参数w的维数也要相应的变为$\tilde{d}$维.

支持向量机的基本型变为：

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \\ s.t. \quad & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) \geq 1, \quad i=1,2,\ldots,m; \end{aligned}$$

对偶型:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) - \sum_{i=1}^{m}\alpha_i \\ s.t. \quad & \sum_{i=1}^{m}\alpha_i y_i = 0, \\ & \alpha_i \geq 0, \quad i=1,2,\ldots,m. \end{aligned}$$

核技巧

在支持向量机的对偶性中，被映射到高维的特征向量总是以成对的内积的形式存在，即$\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$.如果先计算特征在$\mathbb{R}^{\tilde{d}}$ 空间的映射，再计算内积，复杂度是$\mathcal{O}(\tilde{d})$.当特征被映射到非常高维的空间，甚至是无穷空间的时候，计算的代价会非常的大。

核技巧旨在将特征映射和内积这两步运算压缩为一步，并且使得运算复杂度降为$\mathcal{O}(d)$.

即核技巧希望构造一个核函数$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)$使得

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

并且$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)$的计算复杂度是$\mathcal{O}(d)$.

核函数选择

- 当特征维数d超过样本数m时，使用线性核

- 当特征维数d比较小，样本数m中等时，使用RBF核
- 当特征维数d比较小，样本数m特别大时，支持向量机性能通常不如深度神经网络
- 自定义核，需满足 $Mercer$条件

> ($Mercer$ 条件). 核函数$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$对应的矩阵
>
> $$\mathcal{K} = [\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{m\times m}$$
>
> 是半正定的，反之亦然.

名称	形式	优点	缺点
线性核	$\boldsymbol{x}_i^T\boldsymbol{x}_j$	高效实现，不易过拟合	无法处理非线性可分问题
多项式核	$(\beta\boldsymbol{x}_i^T\boldsymbol{x}_j+\theta)^n$	比线性核更一般，n直接描述了被映射空间的复杂度	参数太多，当n过大时会导致计算不稳定
RBF核	$\exp(-\frac{\|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\|^2}{2\sigma^2})$	只有一个参数，没有计算不稳定的问题	计算慢，过拟合风险大

几种常见的核函数

> 线性核

$$\kappa(x_i, x_j) = x_i^T x_j$$

> 多项式核

$$\kappa(x_i, x_j) = (x_i^T x_j)^d$$

> 高斯核

$$\kappa(x_i, x_j) = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$$

> 拉普拉斯核

$$\kappa(x_i, x_j) = \exp(-\frac{||x_i - x_j||}{\sigma})$$

> Sigmoid核

$$\kappa(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$$

常见核函数的组合也是核函数

- 若κ_1和κ_2为核函数，对于任意正数γ_1、γ_2，其线性组合$\gamma_1\kappa_1 + \gamma_2\kappa_2$也是核函数。
- 若κ_1和κ_2为核函数，则核函数的直积$\kappa_1 \otimes \kappa_2(x,z) = \kappa_1(x,z)\kappa_2(x,z)$也是核函数。
- 若κ_1为核函数，对于任意函数$g(x)$，$\kappa(x,z) = g(x)\kappa_1(x,z)g(z)$也是核函数。

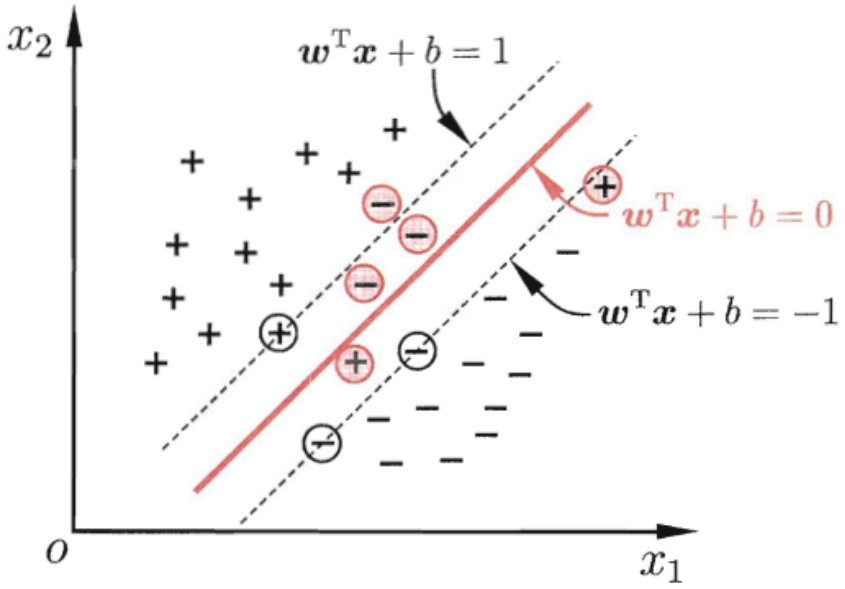
Soft-Margin & Hinge Loss

为什么要引入软间隔?

并不是所有的数据都是线性可分的，对于实在不可分的数据，我们要加上松弛约束条件。

不管直接在原特征空间，还是在映射的高维空间，我们都假设样本是线性可分的。在理论上，我们总能找到一个高维映射使数据线性可分，但在实际任务中，寻找到一个高维的核函数通常很难。

由于数据中常有噪声存在，一味追求数据线性可分可能会使模型陷入过拟合的泥沼。因此，我们需要放宽对样本的要求，即允许有少量的样本分类错误。



软间隔支持向量机基本型

我们希望在优化间隔的同时，允许分类错误的样本出现，但这类样本应尽可能的少：

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{m}\mathbb{I}(y_i \neq sign(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b)) \\ s.t. \quad & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) \geq 1, \quad if \quad y_i = sign(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) \end{aligned}$$

其中，$\mathbb{I}(\cdot)$是指示函数，C是个可调节参数用于权衡优化间隔和少量分类错误样本这两个目标。指示函数不连续，也不是凸函数，使得优化问题不再是二次规划问题。

指示函数只有两个离散值$0/1$，对应样本分类正确/错误。为了能使优化问题继续保持为二次规划问题，我们需要引入一个取值为连续的变量，刻画样本满足约束的程度。

我们引入松弛变量 ξ_i,用于度量样本违背约束的程度。当样本违背约束的程度越大，松弛变量值越大。

即

$$\xi_i = \begin{cases} 0 & if \quad y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) \geq 1 \\ 1-y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) & else \end{cases}$$

软间隔支持向量机基本型

软间隔支持向量机旨在找到一组合适的参数(w,b)，使得

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{m}\xi_i \\ s.t. \quad & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) \geq 1-\xi_i, \quad i=1,2,\ldots,m, \\ & \xi_i \geq 0, \quad i=1,2,\ldots,m. \end{aligned}$$

其中，C是个可调节参数用于权衡优化间隔和少量样本违背间隔约束这两个目标.当C比较大时，我们希望有更多的样本满足大间隔约束；当C比较小时，我们允许有一些样本不满足大间隔约束.

> 证明
>
> 当样本满足约束 $y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) \geq 1$时，$y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) \geq 1-\xi_i$对任意$\xi_i \geq 0$成立，而优化目标要最小化$\xi_i$，所以$\xi_i = 0$.当样本不满足约束时，$\xi_i \geq 1-y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b)$，而优化目标要最小化$\xi_i$，所以$\xi_i = 1-y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b)$

软间隔支持向量机对偶型

软间隔支持向量机的对偶问题等价于找到一组合适的α使得

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) - \sum_{i=1}^{m}\alpha_i \\ s.t. \quad & \sum_{i=1}^{m}\alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq \xi_i, \quad i=1,2,\ldots,m. \end{aligned}$$

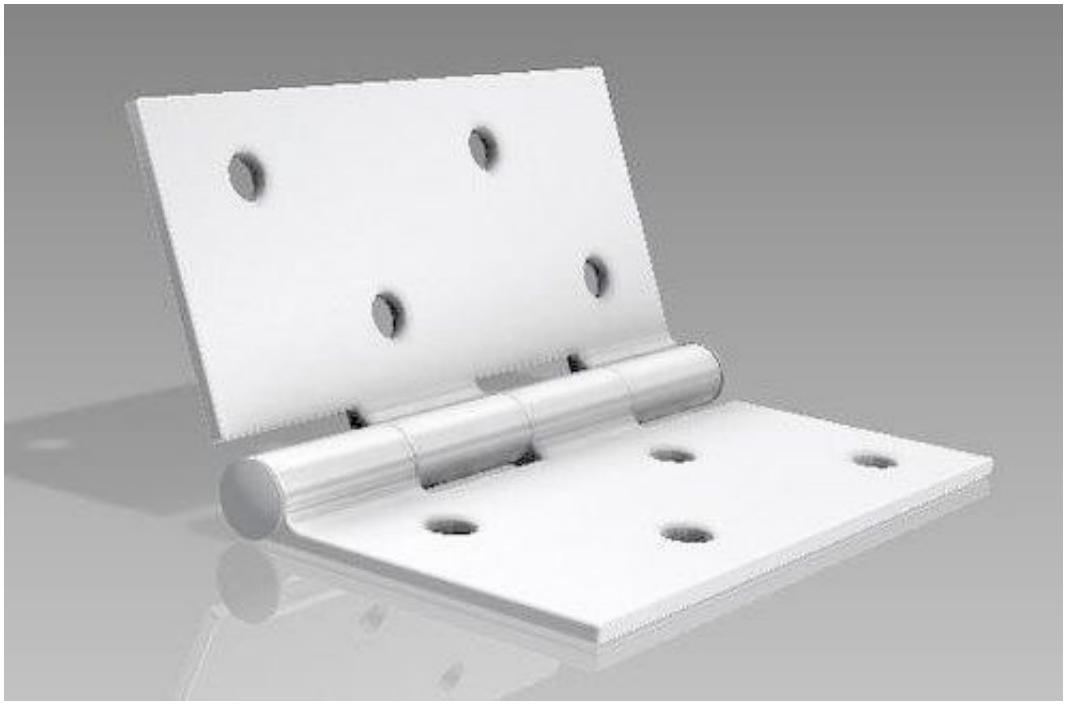
> 证明.
>
> 软间隔支持向量机的拉格朗日函数为
>
> $$\begin{aligned} \mathcal{L}(w,b,\xi,\alpha,\beta) = & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{m}\xi_i \\ & + \sum_{i=1}^{m}\alpha_i(1-\xi_i - y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b)) \\ & + \sum_{i=1}^{m}\beta_i(-\xi_i). \end{aligned}$$
>
> 其对偶问题为
>
> $$\begin{aligned} \max_{\alpha,\beta}\min_{w,b,\xi} \quad & \mathcal{L}(w,b,\xi,\alpha,\beta) \\ s.t. \quad & \alpha_i \geq 0, \quad i=1,2,\ldots,m, \\ & \beta_i \geq 0, \quad i=1,2,\ldots,m. \end{aligned}$$
>
> 我们通过令偏导等于零的方法得到(w,b,ξ)的最优值

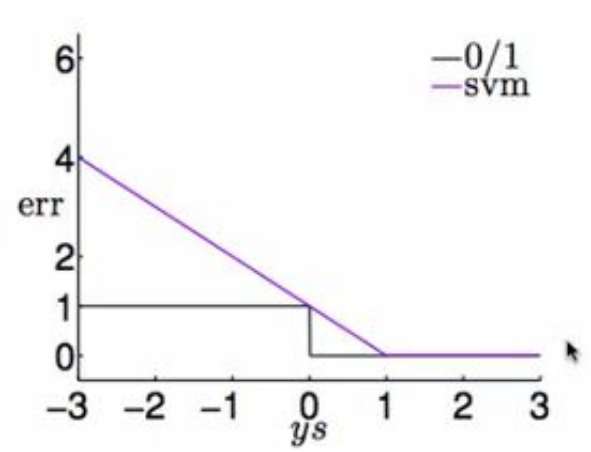
$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y_i \phi(\boldsymbol{x}_i),$$
$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y_i = 0,$$
$$\frac{\partial \mathcal{L}}{\partial \xi} = 0 \Rightarrow \alpha_i + \beta_i = C.$$

带入基本型可得到上述对偶型

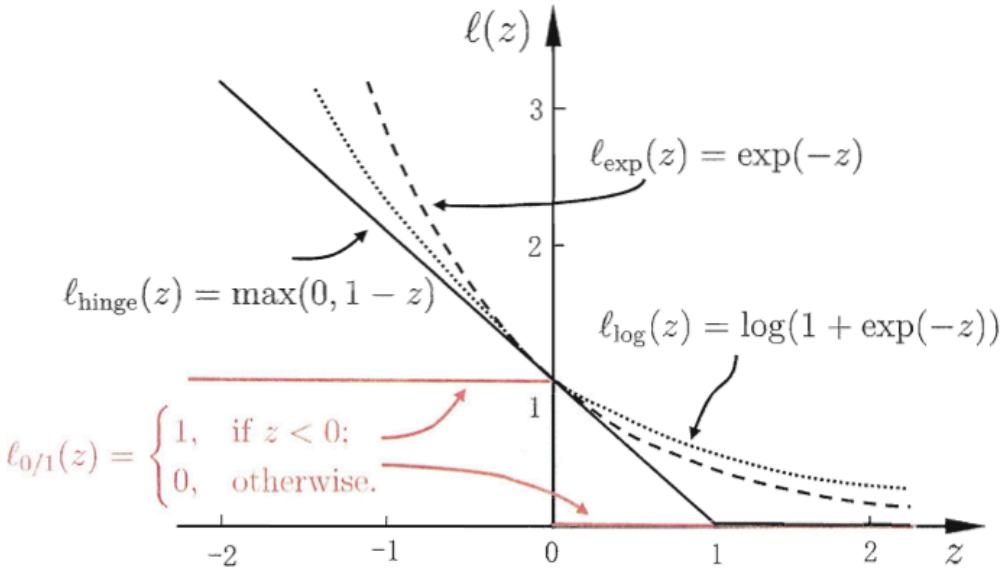
Hinge Loss ， 0-1 Loss

0-1是离散的不方便优化啊，而且Hinge Loss刚好是0-1 Loss的一个上界





其他的损失：



SMO

SMO的基本思路是坐标下降

为什么不能直接计算

$Q = [y_i y_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)]_{m \times m}$的存储开销与计算开销过大

坐标下降

通过循环使用不同的坐标方向，每次固定其他元素，只沿一个坐标方向进行优化，以达到目标函数的局部最小。

我们希望在支持向量机中的对偶型中，每次固定除α_i外的其他变量，求在α_i方向上的极值，但由于约束$\sum_{i=1}^{m} y_i \alpha_i = 0$，当其他变量固定时，$\alpha_i$也随之确定.这样，我们无法再不违背约束条件的前提下对α_i进行优化。因此，SMO每步同时选择两个变量α_i和α_j进行优化，并固定其他参数，以保证不违背约束。

SMO每步的优化目标

SMO每步的优化目标为：

$$
\begin{aligned}
\min_{\alpha_i, \alpha_j} \quad & \frac{1}{2}(\alpha_i^2 y_i^2 \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_i) + \alpha_i^2 y_i^2 \phi(\boldsymbol{x}_j)^T \phi(\boldsymbol{x}_j) \\
& + 2\alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_j)^T \phi(\boldsymbol{x}_i)) - (\alpha_i + \alpha_j) \\
s.t. \quad & \alpha_i y_i + \alpha_j y_j = c, \\
& 0 \le \alpha_i \le \xi_i, \\
& 0 \le \alpha_j \le \xi_j, \\
& \text{其中，} c = -\sum_{k \ne i,j} \alpha_k y_k.
\end{aligned}
$$

SMO每步的优化目标可等价为对α_i的单变量二次规划问题

> 证明
>
> $\alpha_j = y_j(c - \alpha_i y_i)$ 代入SMO每步优化目标，可以消去变量α_j。
>
> 此时优化目标函数是一个对于α_i的二次函数，约束是一个取值区间 $L \leq \alpha_i \leq H$。之后根据目标函数顶点与区间$[L, H]$的位置关系，可以得到α_i的最优值。

其他优化

Pegasos

Pegasos使用基于梯度的方法在线性支持向量机基本型

$$\min_{w,b} \quad \frac{1}{m}\sum_{i=1}^{m}\max(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)) + \frac{\lambda}{2}||w||^2$$

进行优化

$$\frac{\partial J}{\partial w} \leftarrow -\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \leq 1) \cdot y_i\boldsymbol{x}_i + \lambda\boldsymbol{w}$$

$$\frac{\partial J}{\partial b} \leftarrow -\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \leq 1) \cdot y_i$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta\frac{\partial J}{\partial w}$$

$$b \leftarrow b - \eta\frac{\partial J}{\partial b}$$

支持向量机的其他变体

- ProbSVM

  对数几率回归可以估计出样本属于正类的概率，而支持向量机只能判断出样本属于正类或者负类，无法得到概率。ProbSVM 先训练一个支持向量机，得到参数(w, b)，再令$s_i = y_i\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b$,将$\{(s_1, y_1), (s_2, y_2), \ldots, (s_m, y_m)\}$当做新的训练数据训练一个对数几率回归模型。

- 多分类支持向量机

  对于K分类问题，多分类支持向量机有K组参数$\{(\boldsymbol{w}_1, b_1), (\boldsymbol{w}_2, b_2), \ldots, (\boldsymbol{w}_k, b_k)\}$，并希望模型对于属于正确标记的结果以1的间隔高于其他类的结果，形式化如下：

  $$\min_{\boldsymbol{W},b} \quad \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\max(0, (\boldsymbol{w}_{y_i}^T\phi(\boldsymbol{x}_i) + b_{y_i}) - (\boldsymbol{w}_k^T\phi(\boldsymbol{x}_i) + b_k) + 1) + \frac{\lambda}{2}\sum_{k=1}^{K}\boldsymbol{w}_k^T\boldsymbol{w}_k$$

- 支持向量回归（SVR）

  经典回归模型的损失函数度量了模型预测$h(\boldsymbol{x}_i)$与y_i的差别，支持向量回归能够容忍$h(\boldsymbol{x}_i)$与y_i之间小于ϵ的偏差.令$s = y - (\boldsymbol{w}^T\phi(\boldsymbol{x}) + b)$，我们定义$\epsilon$不敏感损失为：

$$\begin{cases} 0 & if \quad |y-(\boldsymbol{w}^T\phi(\boldsymbol{x})+b)| \le \epsilon; \\ |s|-\epsilon & else \end{cases}$$

支持向量回归可形式化为：

$$\min_{\boldsymbol{w},b} \quad \frac{1}{m}\sum_{i=1}^{m}\max(0,|y-(\boldsymbol{w}^T\phi(\boldsymbol{x})+b)|-\epsilon)+\frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$$

> 证明
>
> ToDo

再谈SVM

正则化

- L1正则化的优点是优化后参数向量往往比较稀疏
- L2正则化的有点是其正则化项处处可导

SVM

SVM的训练被看做一个有约束的优化问题 目标函数是$\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$，后面的$C\sum_{i=1}^{N}\xi_i$反倒是比较像正则化

不过并不典型。它并不是为了解决过拟合问题，而是为了容错。而且它本身并不是限制模型参数本身的大小，而是限制容错量的大小。

不过我们可以把SVM的训练过程从另外一个角度理解

我们不再把margin的大小作为目标函数，而是考虑分类错误所带来的代价。

对于“+”类（$\boldsymbol{y_i}=\boldsymbol{1}$）的数据$\boldsymbol{x_i}$，我们希望$\boldsymbol{w^Tx>0}$（同样省略了bias）； 对于“-”类（$\boldsymbol{y_i}=\boldsymbol{-1}$）的数据$\boldsymbol{x_i}$，我们希望$\boldsymbol{w^Tx<0}$：

总之，我们希望

$$\boldsymbol{w^Tx_iy_i>0}$$

。

那么，如果实际上$\boldsymbol{w^Tx_iy_i}$符号为负，或者虽然符号为正但离0不够远，具体来说是$\boldsymbol{w^Tx_iy_i<1}$，我们就认为这个分类错误（或“不够正确”）带来了大小为$\boldsymbol{1-w^Tx_iy_i}$的损失。

于是目标函数（损失函数）就是$\boldsymbol{L=\sum_{i=1}^{n}\max(0,1-w^Tx_iy_i)}$，SVM的训练变成了这个目标函数下的**无约束优化问题**。

在数据线性可分的情况下，会有许多w使得L = 0。为了选择一个合理的w，也需要加入正则化。

加入L2正则化之后，目标函数就变成了

$$\boldsymbol{L=\sum_{i=1}^{n}\max(0,1-w^Tx_iy_i)+\frac{C}{2}||w||_2^2}$$

。

可以验证，这个目标函数跟题干中的目标函数是等价的。

而正则化项恰巧具有物理意义；最小化正则化项，就是最大化margin。

现在该如何理解SVM的过程?

- **最大化margin（目标函数），顺便容错；**
- **另一个角度，是最小化分类错误造成的损失（目标函数），顺便让margin尽可能大（正则化）。**

我们再来比较一下logistic regression和SVM的（第二种）目标函数：

$$L_{\mathrm{LR}}=\sum_{i=1}^{n}\log(1+\mathrm{e}^{-w^{T}x_{i}y_{i}})+\frac{C}{2}||w||_{2}^{2}\quad L_{\mathrm{SVM}}=\sum_{i=1}^{n}\max(0,1-w^{T}x_{i}y_{i})+\frac{C}{2}||w||_{2}^{2}$$

可以看出，二者的唯一区别就是对于单个数据点，计算损失的方法不同。

Logistic regression中的$f(x)=\log(1+\mathrm{e}^{-x})$称为**log loss**；

而SVM中的 $f(x)=\max(0,1-x)$ 称为**hinge loss**。

这样SVM中的损失函数就是可以被随便设计的