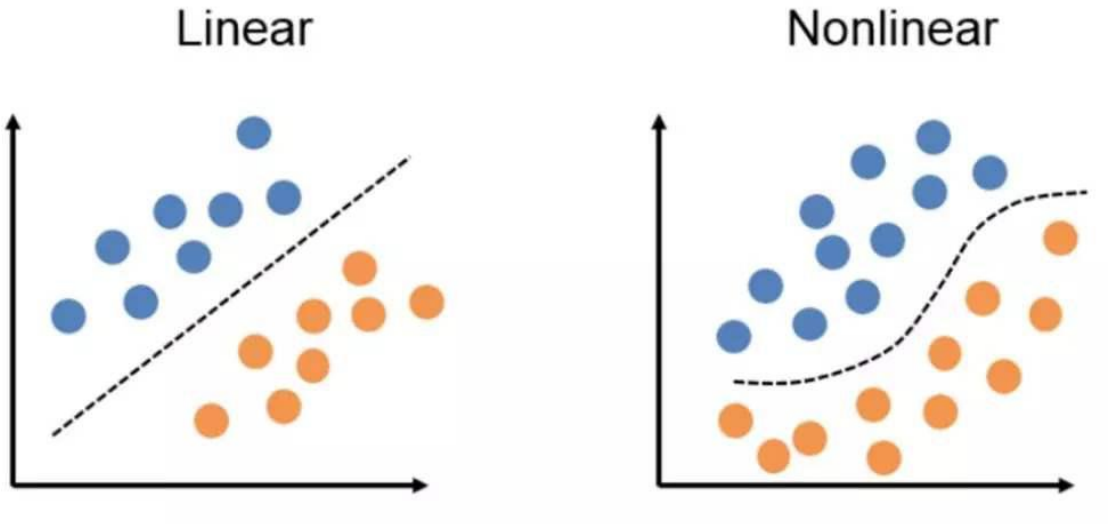
SVM 支持向量机

支持向量机(Support Vector Machine, SVM)本身是一个二元分类算法，是对感知器算法模型的一种扩展，现在的 SVM 算法支持线性分类和非线性分类的分类应用，并且也能够直接将 SVM 应用于回归应用中，同时通过 OvR 或者 OvO 的方式我们也可以将 SVM 应用在多元分类领域中。在不考虑集成学习算法，不考虑特定的数据集的时候，在分类算法中 SVM 可以说是特别优秀的。

感知器算法思想

Rosenblatt 他是一位心理医生，在神经感知科学背景下于 1958 提出了类似现在机器学习模型的感知机。罗森布拉特声称感知机不仅能识别图像，还能教机器行走，说话和做出表情。受时代的限制，感知机能做的事情还很有限。神经网络的研究经历了十多年的冷冻期。后来，2004 年，IEEE Frank Rosenblatt Award 成立，他被称为神经网络的创立者。

感知器算法是最古老的分类算法之一，原理比较简单，不过模型的分类泛化能力比较弱，不过感知器模型是 SVM、神经网络、深度学习等算法的基础。感知器的思想很简单：在任意空间中，感知器模型寻找的就是一个超平面，能够把所有的二元类别分割开。感知器模型的前提是：数据是线性可分的。



对于 m 个样本，每个样本 n 维特征以及一个二元类别输出 y，如下：

$$\{(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}), (x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(2)}), \ldots, (x_1^{(m)}, x_2^{(m)}, \ldots, x_n^{(m)})\}$$

目标是找到一个超平面

$$\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n = 0 \rightarrow \theta x = 0$$

让一个类别的样本满足：θx>0 ；另外一个类别的满足：θx<0
感知器模型为:

$$y = sign(\theta x) = \begin{cases} +1, & \theta x > 0 \\ -1, & \theta x < 0 \end{cases}$$

正确分类：y*θx>0，错误分类：y*θx<0；所以我们可以定义我们的损失函数为：期望使分类错误的所有样本到超平面的距离之和最小。

逻辑回归的几何意义，又何尝不是向量空间找到一个超平面，超平面一侧的点计算分数结果为负，另一侧结果分数为正，只不过最后不直接看 sign 符合，而是根据 sigmoid 函数将分数映射到 0-1 之间通过最大似然来赋予概率意义。

几何距离和函数距离

如何计算任意 xi，yi 到某平面 W 的距离？
高中知识，点到直线的距离：
二维平面中，点 xi，yi 到到直线 ax+by+c=0 的距离为：

$$d(x_i, y_i) = \frac{|ax_i + by_i + c|}{\sqrt{a^2 + b^2}}$$

推广到高维空间中，任意一个点 X0 其对应标签为 y0，到某平面$w^T x + b = 0$的距离为：

$$\gamma = \frac{|w^t x_0 + b|}{||w||}$$

其中

$$||w|| = \sqrt{w^T w}$$

对于那些正确分类的点，y0 必然与$w^t x_0 + b$同号，所以可以将距离表示为：

$$\gamma = \frac{y_0(w^t x_0 + b)}{||w||}$$

称这个距离为某点到平面的几何距离

称分子部分$y_0(w^t x_0+b)$为某点到平面的函数距离

损失函数求解

感知器损失函数的定义：

期望使分类错误的所有样本(m 条样本)到超平面的距离之和最小

对于那些错误的点，y0 必然与$w^t x_0+b$反号，要想使得损失函数是大于 0 的，所以分类错误的点距离在前面加个负号

$$L=\sum_{i=1}^{m}\frac{-y^i\theta*x^i}{||\theta||_2}$$

因为此时分子和分母中都包含了θ值，当分子扩大 N 倍的时候，分母也会随之扩大，也就是说分子和分母之间存在倍数关系，所以可以固定分子或者分母为 1，然后求另一个即分子或者分母的倒数的最小化作为损失函数，简化后的损失函数为（分母为 1）：

$$L=-\sum_{i=1}^{m}y^{(i)}\theta*x^{(i)}$$

直接使用梯度下降法就可以对损失函数求解，不过由于这里的 m 是分类错误的样本点集合，不是固定的，所以我们不能使用批量梯度下降法(BGD)求解，只能使用随机梯度下降(SGD)或者小批量梯度下降(MBGD)；一般在感知器模型中使用 SGD 来求解。

$$\frac{\delta L(\theta)}{\delta\theta}=-\sum_{i=1}^{m}y^{(i)}x^{(i)}$$

$$\theta^{k+1}=\theta^k+\alpha y^{(i)}x^{(i)}$$

SVM 算法思想

相同的地方

SVM 也是通过寻找超平面，用于解决二分类问题的分类算法

超平面一侧的点计算分数结果为负是负例，另一侧结果分数为正是正例

与感知机相同，通过 sign 给出预测标签，正例为+1，负例为-1，模型判别式同样：

$$y = sign(\theta x) = \begin{cases} +1, & \theta x > 0 \\ -1, & \theta x < 0 \end{cases}$$

不同的地方

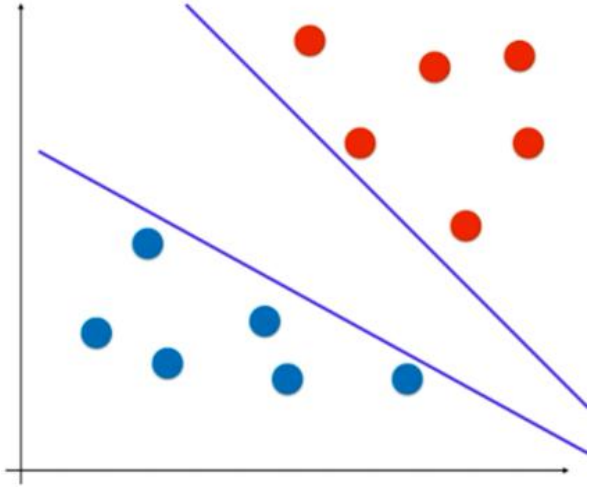
损失函数与感知机和逻辑回归都不同

感知机是通过判错的点寻找超平面，逻辑回归是通过最大似然寻找超平面，SVM 是通过支持向量寻找超平面，这也是 SVM 这个名字的由来，当然这也是损失函数不同的原因

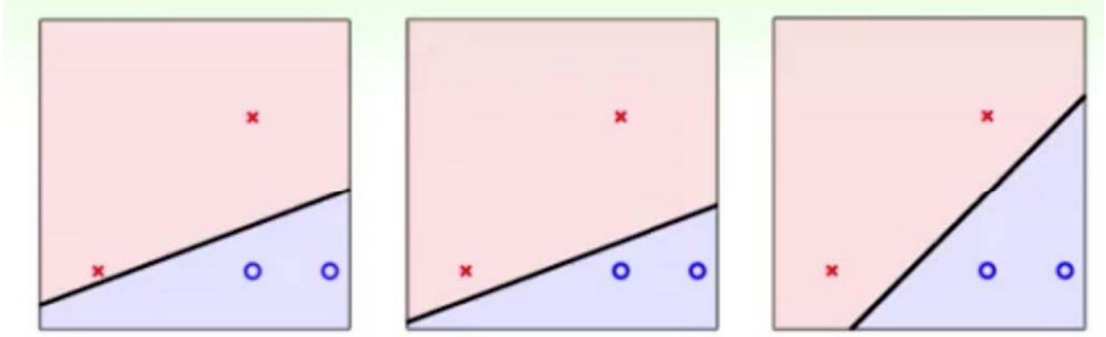
感知机和逻辑回归是直接最小化损失函数来得到θ，或者叫 W 和 b，SVM 有两种求解方式，一种是直接最小化损失函数来得到θ，另一种先寻找支持向量，找到支持向量超平面就自然找到了

高级的地方

当面对下图时，感知器可以分类开来，但是哪条分界是更好的呢？

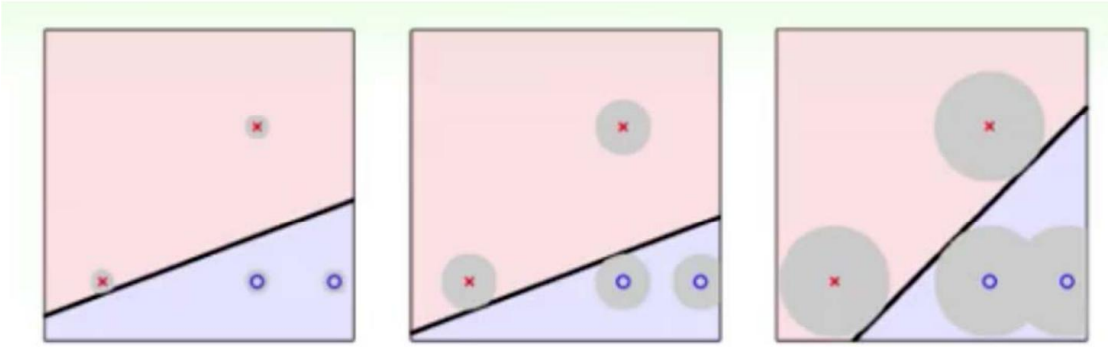


对于这些分界线哪个更好，这便是 SVM 要解决得问题。



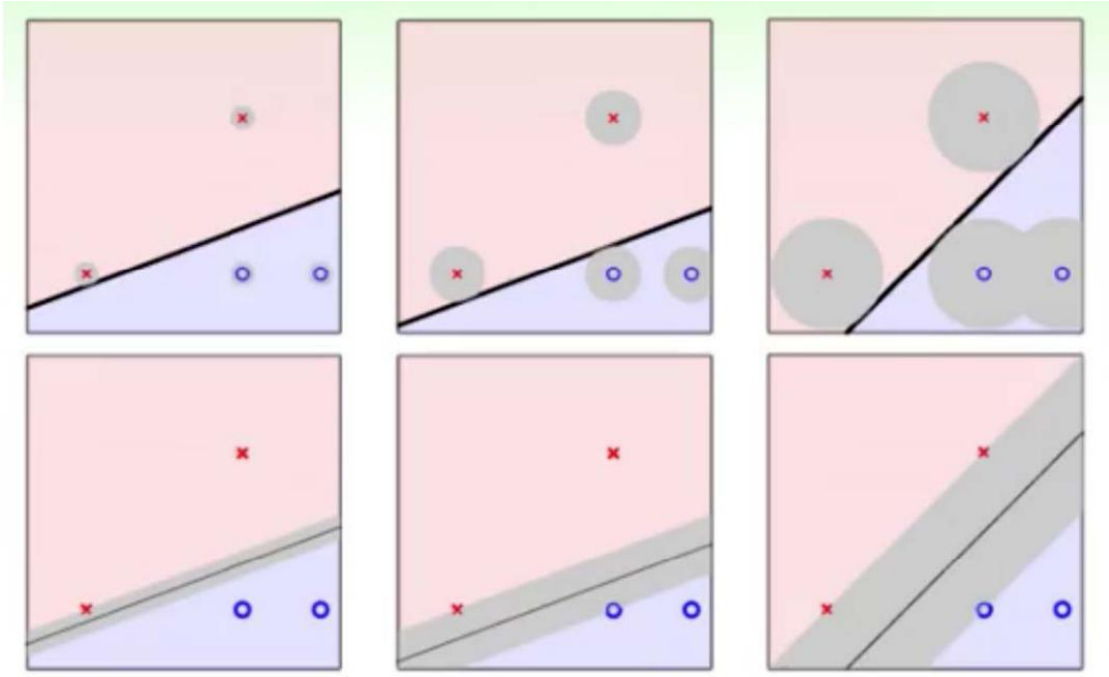
在感知器模型中，我们可以找到多个可以分类的超平面将数据分开，并且优化时希望所

有的点都离超平面尽可能的远，但是实际上离超平面足够远的点基本上都是被正确分类的，所以这个是没有意义的；反而比较关心那些离超平面很近的点，这些点比较容易分错。



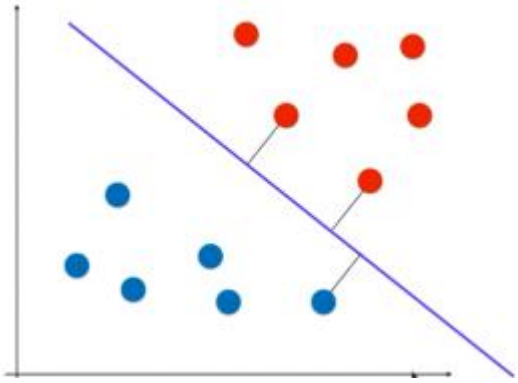
假设未来拿到的数据含有一部分噪声，那么不同的超平面对于噪声的容忍度是不同的，最右边的线是最 robust 的。

换一种角度考虑，找到最胖的超平面



所以说我们只要让离超平面比较近的点尽可能的远离这个超平面 ,那么我们的模型分类效果应该就会比较不错喽。SVM 其实就是这个思想。

就是在支持向量机中 ,距离超平面最近的且满足一定条件的几个训练样本点被称为支持向量。



SVM 尝试找到一个决策边界,距离两个类别最近的样本最远！

SVM 支持向量机

1. 线性可分支持向量机
   硬间隔最大化
2. 线性支持向量机
   软间隔最大化
3. 非线性支持向量机
   升维（核函数）

线性可分(Linearly Separable)：

在数据集中，如果可以找出一个超平面，将两组数据分开，那么这个数据集叫做线性可分数据。

线性不可分(Linear Inseparable)：

在数据集中，没法找出一个超平面，能够将两组数据分开，那么这个数据集就叫做线性不可分数据。分割超平面(Separating Hyperplane)：将数据集分割开来的直线/平面叫做分割超平面。

间隔(Margin)：

数据点到分割超平面的距离称为间隔。

支持向量(Support Vector)：

离分割超平面最近的那些点叫做支持向量。

线性可分支持向量机

硬间隔最大化

需要找到一个超平面：

1. 能够完美分类正负例
2. 距离最近的点越远越好

超平面怎么确定：

1. $y = \text{sign}(w^T x + b)$
2. $w^T x + b = 0$表示的就是分割超平面
3. 只要确认了 w 和 b 就确认了我们的分割超平面

目标：

能正确分类的平面中，距离越近的点越远越好

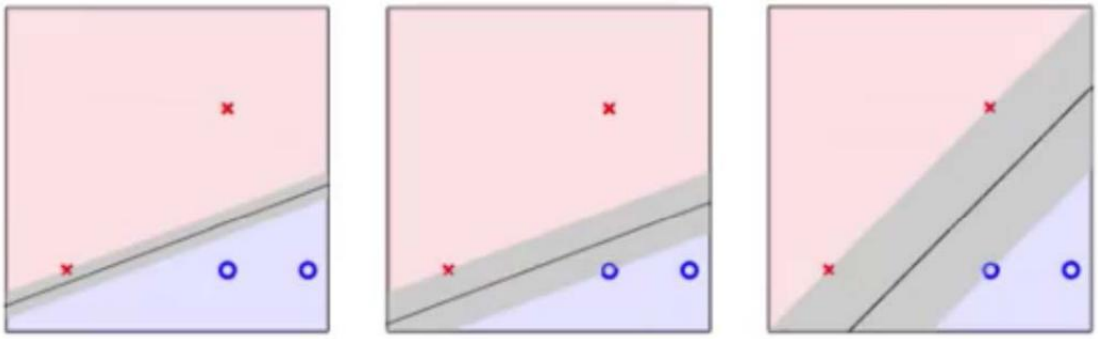
也就是说，找到一组最好的 w 和 b 固定一个超平面，使这个超平面在能完美区分正负例的基础上，距离最近的点间隔最大

转换为有约束的函数最优化问题就是：

$$\max_{w,b} \gamma_{min} = \frac{y_{min}(wx_{min}+b)}{||w||}$$

$$s.t \quad y_i(w^T x_i + b) = \gamma'^{(i)} \geq \gamma'(i = 1, 2, \ldots m)$$

其中γ′ 代表支持向量的函数距离



简化目标

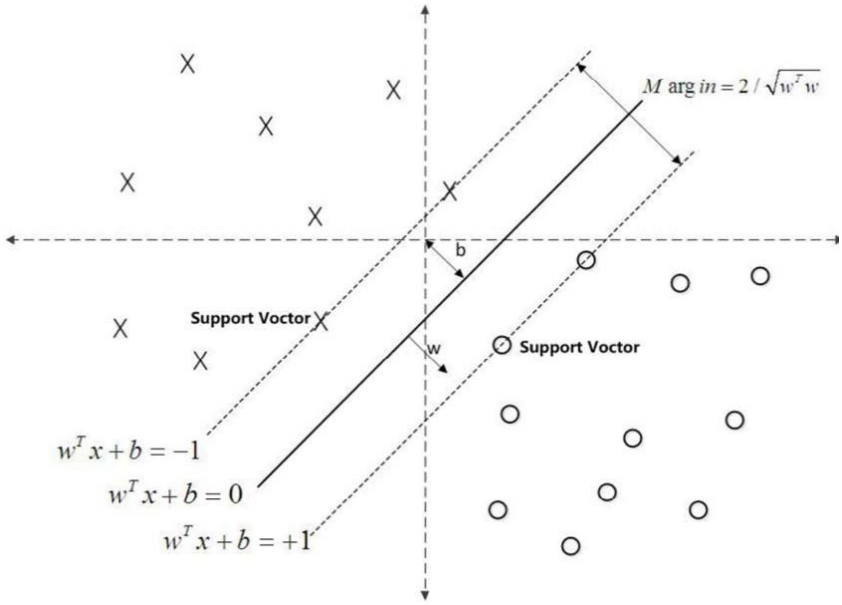
一组 w，b 只能固定一个超平面

一个超平面对应无数个 w,b，只要找到其中任意一个 w 符合条件的 w 就可以了

选择最好求的，令γ' =1 则原最优化问题为：

$$max \quad \frac{1}{||w||_2}$$

$$s.t \quad y_i(w^T x_i + b) \geq 1 (i = 1, 2, \ldots m)$$



等价于：

$$min \quad \frac{1}{2}||w||_2^2 \quad s.t \quad y_i(w^T x_i + b) \geq 1 (i = 1, 2, \ldots m)$$

函数最优化问题

给定一个函数 f(x)找到一个 x 使 f(x)最小

梯度下降、L-BFGS、SMO 算法都是解决函数最优化问题的

$$x = min_{x \in R^n} f(x)$$

由于上式这里对 x 的取值范围，并没有做任何限制所以也称为无约束条件的最优化问题

有约束条件的函数最优化问题

例子：$f(x) = 4x^2 + 5x + 10$

最优化问题就是，求得 x 使得 f(x)最小，结果：x=-5/8

假设给定约束条件 x> =0 呢？

x=0 的时候取得最小值！

如何通过一个统一的方法来求解这种带约束条件的函数最优化问题呢？

原始问题

带约束条件的最优化问题泛化表示方法：

$$\min_{x \in R^n} f(x)$$

$$\text{s.t.} \quad c_i(x) \leqslant 0, \quad i = 1, 2, \cdots, k$$

$$h_j(x) = 0, \quad j = 1, 2, \cdots, l$$

可以将约束条件表述为 k 个不等式约束条件，和 L 个等式约束条件

我们命名其为原始最优化问题

拉格朗日函数

定义某原始最优化问题的拉格朗日函数为：

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^{k} \alpha_i c_i(x) + \sum_{j=1}^{l} \beta_j h_j(x)$$

其中 ci 是第 i 个不等式约束函数（需要整理），bj 是第 j 个等式约束函数

αi 和βi 是拉格朗日乘子

拉格朗日函数的特性

令 $\theta_P(x) = \max_{\alpha, \beta; \alpha_i \geqslant 0} L(x, \alpha, \beta)$

若 x 不满足之前的约束条件：

$$\theta_P(x)=\max_{\alpha,\beta:\alpha_i\geq 0}\left[f(x)+\sum_{i=1}^{k}\alpha_i c_i(x)+\sum_{j=1}^{l}\beta_j h_j(x)\right]=+\infty$$

若 x 满足约束条件：

$$\theta_P(x)=f(x)$$

拉格朗日函数

$$\theta_P(x)=\begin{cases}f(x), & x\text{满足原始问题约束}\\ +\infty, & \text{其他}\end{cases}$$

如果对于$\theta_p(x)$ 进行极小化，$\min_x \theta_P(x)=\min_x \max_{\alpha,\beta:\alpha_i\geq 0} L(x,\alpha,\beta)$ 就相当于对原始最优化问题进行极小化，它们拥有相同的解

定义原始问题的最优解 $p^*=\min_x \theta_P(x)$

对偶问题

定义

$$\theta_D(\alpha,\beta)=\min_x L(x,\alpha,\beta)$$

此时极大化θ_D

$$\max_{\alpha,\beta:\alpha_i\geq 0}\theta_D(\alpha,\beta)=\max_{\alpha,\beta:\alpha_i\geq 0}\min_x L(x,\alpha,\beta)$$

称为拉格朗日的极大极小问题，也称为原始问题的对偶问题

设 $d^*=\max_{\alpha,\beta:\alpha_i\geq 0}\theta_D(\alpha,\beta)$ 为对偶最优化问题的最优解

当 f(x)和 ci 函数为凸函数，hj 函数为仿射函数时，有：

$$p^*=d^*=L(x^*,\alpha^*,\beta^*)$$

如何求解

KKT 条件：

$$\nabla_x L(x^*,\alpha^*,\beta^*)=0 \qquad \alpha_i^* c_i(x^*)=0,\quad i=1,2,\cdots,k$$

$$\nabla_\alpha L(x^*,\alpha^*,\beta^*)=0 \qquad c_i(x^*)\leqslant 0,\quad i=1,2,\cdots,k$$

$$\nabla_\beta L(x^*,\alpha^*,\beta^*)=0 \qquad \alpha_i^*\geqslant 0,\quad i=1,2,\cdots,k$$

$$h_j(x^*)=0 \quad j=1,2,\cdots,l$$

求解最优化问题

对于原始问题：

$$min \quad \frac{1}{2}||w||_2^2 \quad s.t \quad y_i(w^T x_i+b)\geq 1(i=1,2,...m)$$

构建拉格朗日函数：

$$L(w,b,\alpha)=\frac{1}{2}||w||_2^2-\sum_{i=1}^{m}\alpha_i[y_i(w^T x_i+b)-1] \text{ 满足}\alpha_i\geq 0$$

可将原始有约束的最优化问题转换为对拉格朗日函数进行无约束的最优化问题(也叫二次规划问题)

$$\underbrace{min}_{w,b}\ \underbrace{max}_{\alpha_i\geq 0}L(w,b,\alpha)$$

由于我们的原始问题满足 f(x)为凸函数，那么可以将原始问题的极小极大优化转换为对偶函数的极大极小优化进行求解：

对于原始问题：

$$\underbrace{min}_{w,b}\ \underbrace{max}_{\alpha_i\geq 0}L(w,b,\alpha)$$

对偶函数为：

$$\underbrace{max}_{\alpha_i\geq 0}\ \underbrace{min}_{w,b}\ L(w,b,\alpha)$$

下面就开始求解对偶函数的第一步

$$\underbrace{min}_{w,b} L(w,b,\alpha)$$

第一步求极小

对拉格朗日函数分别求 w 和 b 的偏导：

$$L(w,b,\alpha)=\frac{1}{2}||w||_2^2-\sum_{i=1}^{m}\alpha_i[y_i(w^Tx_i+b)-1] \text{ 满足}\alpha_i\geq 0$$

$$\frac{\partial L}{\partial w}=0 \Rightarrow w=\sum_{i=1}^{m}\alpha_i y_i x_i \qquad \frac{\partial L}{\partial b}=0 \Rightarrow \sum_{i=1}^{m}\alpha_i y_i=0$$

可以看出：我们已经求得了 w 和α的关系，下一步将 w 反代回原来的拉格朗日函数中就可以进行第二步求关于α的极大了

反代回去

$$=\frac{1}{2}||w||_2^2-\sum_{i=1}^{m}\alpha_i[y_i(w^Tx_i+b)-1]$$

$$=\frac{1}{2}w^Tw-\sum_{i=1}^{m}\alpha_i y_i w^Tx_i-\sum_{i=1}^{m}\alpha_i y_i b+\sum_{i=1}^{m}\alpha_i$$

$$=\frac{1}{2}w^T\sum_{i=1}^{m}\alpha_i y_i x_i-\sum_{i=1}^{m}\alpha_i y_i w^Tx_i-\sum_{i=1}^{m}\alpha_i y_i b+\sum_{i=1}^{m}\alpha_i$$

$$=\frac{1}{2}w^T\sum_{i=1}^{m}\alpha_i y_i x_i-w^T\sum_{i=1}^{m}\alpha_i y_i x_i-\sum_{i=1}^{m}\alpha_i y_i b+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}w^T\sum_{i=1}^{m}\alpha_i y_i x_i-\sum_{i=1}^{m}\alpha_i y_i b+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}w^T\sum_{i=1}^{m}\alpha_i y_i x_i-b\sum_{i=1}^{m}\alpha_i y_i+\sum_{i=1}^{m}\alpha_i$$

$$
\begin{aligned}
&= -\frac{1}{2}(\sum_{i=1}^{m}\alpha_i y_i x_i)^T(\sum_{i=1}^{m}\alpha_i y_i x_i) - b\sum_{i=1}^{m}\alpha_i y_i + \\
&= -\frac{1}{2}\sum_{i=1}^{m}\alpha_i y_i x_i^T\sum_{i=1}^{m}\alpha_i y_i x_i - b\sum_{i=1}^{m}\alpha_i y_i + \sum_{i=1}^{m} \\
&= -\frac{1}{2}\sum_{i=1}^{m}\alpha_i y_i x_i^T\sum_{i=1}^{m}\alpha_i y_i x_i + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i y_i x_i^T\alpha_j y_j x_j + \sum_{i=1}^{m}\alpha_i \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j
\end{aligned}
$$

整理对偶函数

第二步对对偶函数的优化问题：

$$\underbrace{max}_{\alpha} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j(x_i \bullet x_j) + \sum_{i=1}^{m}\alpha_i$$

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

$$\alpha_i \geq 0 \; i = 1, 2, \ldots m$$

去掉负号转换为求极小问题：

$$\underbrace{min}_{\alpha} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j(x_i \bullet x_j) - \sum_{i=1}^{m}\alpha_i$$

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

$$\alpha_i \geq 0 \; i = 1, 2, \ldots m$$

只要解决了这个问题，svm 的学习问题就完成了！

通常使用 SMO 算法进行求解，可以求得一组α* 使得函数最优化

得到最终的超平面

假设已经通过 SMO 算法，求得α*，此时求 w*很容易：

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

b*怎么求？对于任意支持向量，有：

$$y_s(w^T x_s + b) = y_s(\sum_{i=1}^{m} \alpha_i y_i x_i^T x_s + b) = 1$$

如何找到支持向量？根据 KKT 条件有：

$$\alpha_i^*(y_i(w^T x_i + b) - 1) = 0$$

那么所有α>0 时后面一项需要=0 也就是

$$y_i(w^T x_i + b) = 1$$

求 b 的过程：找到所有个支持向量带进去求出所有个 b，然后求平均

这样我们就得到了分割超平面

$$w^{*T} x + b^* = 0$$

硬分隔 svm 总结

流程

1. 原始目标： 求得一组 w 和 b 使得分隔 margin 最大
2. 转换目标： 通过拉格朗日函数构造目标函数，问题由求得 n 个 w 和 1 个 b 转换为求得 m 个α

$$\underbrace{min}_{\alpha} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \bullet x_j) - \sum_{i=1}^{m} \alpha_i$$

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

$$\alpha_i \geq 0 \; i = 1, 2, \ldots m$$

3. 利用 smo 算法求得 m 个α*
4. 利用求得的 m 个α*求得 w*和 b*

$$w^* = \sum_{i=1}^{m} \alpha_i^* y_i x_i \qquad b_s^* = y_s - \sum_{i=1}^{m} \alpha_i y_i x_i^T x_s \qquad b^* = \frac{1}{S} \sum_{i=1}^{S} b_s^*$$

举例

给定 3 个数据点：正例点 x1=(3,3)，x2=(4,3)，负例点 x3=(1,1)，求线性可分支持向量机

建立目标函数：

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$= \frac{1}{2}\left(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3\right) - \alpha_1 - \alpha_2 - \alpha_3$$

$$s.t. \ \ \alpha_1 + \alpha_2 - \alpha_3 = 0$$

$$\alpha_i \geq 0, \quad i = 1,2,3$$

求解α1，α2，α3

求解过程

由于α3=α1+α2 所以可以消去α3 得到关于α1、α2 的函数

$$f(\alpha 1,\ \alpha 2) = 4\alpha_1^2 + \frac{13\alpha_2^2}{2} + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$$

分别对α1、α2 求偏导，可以求得 f(α1,α2)在(2/3,-1)处求得极值

但是α2 不满足约束条件>0，所以次之α2=0 时出现约束条件下的极值α2=0 时，对α1 求偏导求得对应的α1=1/4

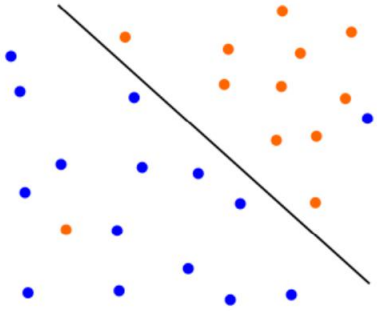
此时，α1=1/4，α2=0，α3=1/4

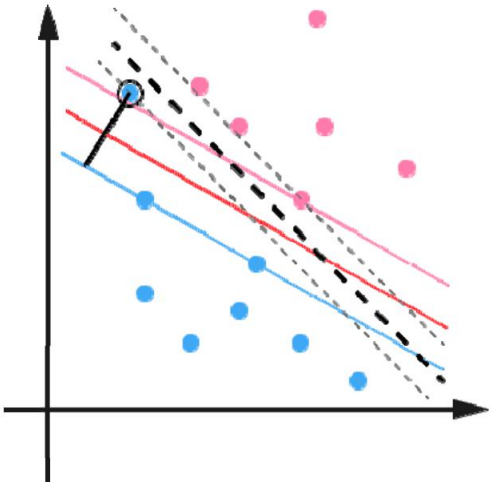
由此可知，α1 和α3 对应的 X1 和 X3 为该例子的支持向量

线性支持向量机

硬间隔面临的问题

有些时候，线性不可分是由噪声点决定的





软间隔 SVM

对于之前讲述的 线性可分 svm 可通过构造超平面令硬间隔最大化 ,从而求得最好的分隔超平面

条件：

1. 正负例完美分开（体现在约束条件>=1 上）
2. 找到能使间隔最大的点（有约束条件的函数优化问题）

如果数据集线性不可分，意味着找不到一个合格的超平面

体现在优化问题上，任何的 w 和 b 都无法满足优化条件

引入松弛变量

对于之前的问题，硬间隔不可分，体现在满足不了约束条件上，于是提出松弛变量ξi≥0（每个数据点自己有一个ξi）

我们将约束条件放松为：

$$y_i(w \bullet x_i + b) \geq 1 - \xi_i$$

这样就至少肯定有好多的 w 和 b 满足条件了 但是这相当于没有约束条件了 ,只要ξi 无穷大，那么所有 w 和 b 都满足条件

ξ代表异常点嵌入间隔面的深度 ,我们要在能选出符合约束条件的最好的 w 和 b 的同时，让嵌入间隔面的总深度越少越好

目标函数的优化

$$min \quad \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad (i = 1, 2, \ldots m)$$

$$\xi_i \geq 0 \quad (i = 1, 2, \ldots m)$$

1. 根据 f(x) 和约束条件构造拉格朗日函数：

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(w^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\mu_i\xi_i$$

- 其中要求μi 和αi>=0

2. 优化原始问题：

$$\underbrace{min}_{w,b,\xi} \ \underbrace{max}_{\alpha_i \geq 0, \mu_i \geq 0,} \ L(w, b, \alpha, \xi, \mu)$$

3. 对偶问题：

$$\underbrace{max}_{\alpha_i \geq 0, \mu_i \geq 0,} \ \underbrace{min}_{w,b,\xi} \ L(w, b, \alpha, \xi, \mu)$$

- 先求 L 函数对 w，b，ξ的极小值，再求其对α和μ的极大值

对偶问题求解

对 3 个参数分别求偏导得到一定的信息，反带回拉格朗日函数

$$\frac{\partial L}{\partial w} = 0 \ \Rightarrow w = \sum_{i=1}^{m}\alpha_i y_i x_i$$

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

$$\frac{\partial L}{\partial \xi} = 0 \ \Rightarrow C - \alpha_i - \mu_i = 0$$

带回拉格朗日函数

$$
\begin{aligned}
&= \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(w^Tx_i+b)-1+\xi_i] - \sum_{i=1}^{m}\mu_i\xi_i \\
&= \frac{1}{2}||w||_2^2 - \sum_{i=1}^{m}\alpha_i[y_i(w^Tx_i+b)-1+\xi_i] + \sum_{i=1}^{m}\alpha_i\xi_i \\
&= \frac{1}{2}||w||_2^2 - \sum_{i=1}^{m}\alpha_i[y_i(w^Tx_i+b)-1] \\
&= \frac{1}{2}w^Tw - \sum_{i=1}^{m}\alpha_iy_iw^Tx_i - \sum_{i=1}^{m}\alpha_iy_ib + \sum_{i=1}^{m}\alpha_i \\
&= \frac{1}{2}w^T\sum_{i=1}^{m}\alpha_iy_ix_i - \sum_{i=1}^{m}\alpha_iy_iw^Tx_i - \sum_{i=1}^{m}\alpha_iy_ib + \sum_{i=1}^{m}\alpha_i \\
&= \frac{1}{2}w^T\sum_{i=1}^{m}\alpha_iy_ix_i - w^T\sum_{i=1}^{m}\alpha_iy_ix_i - \sum_{i=1}^{m}\alpha_iy_ib + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}w^T\sum_{i=1}^{m}\alpha_iy_ix_i - \sum_{i=1}^{m}\alpha_iy_ib + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}w^T\sum_{i=1}^{m}\alpha_iy_ix_i - b\sum_{i=1}^{m}\alpha_iy_i + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}(\sum_{i=1}^{m}\alpha_iy_ix_i)^T(\sum_{i=1}^{m}\alpha_iy_ix_i) - b\sum_{i=1}^{m}\alpha_iy_i + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}\sum_{i=1}^{m}\alpha_iy_ix_i^T\sum_{i=1}^{m}\alpha_iy_ix_i - b\sum_{i=1}^{m}\alpha_iy_i + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}\sum_{i=1}^{m}\alpha_iy_ix_i^T\sum_{i=1}^{m}\alpha_iy_ix_i + \sum_{i=1}^{m}\alpha_i \\
&= -\frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_iy_ix_i^T\alpha_jy_jx_j + \sum_{i=1}^{m}\alpha_i \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i\alpha_jy_iy_jx_i^Tx_j
\end{aligned}
$$

整理约束条件

第二步极大问题：

$$\underbrace{max}_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

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

$$C - \alpha_i - \mu_i = 0$$

$$\alpha_i \geq 0 \ (i = 1, 2, \ldots, m)$$

$$\mu_i \geq 0 \ (i = 1, 2, \ldots, m)$$

与之前的目标函数一模一样，只不过约束条件不同了

由于目标函数中并没有出现 C，可将约束条件的第 2，3，4 项合并，消去 C

得到最终的待优化函数为：

$$\underbrace{min}_{\alpha} \frac{1}{2} \sum_{i=1,j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m} \alpha_i$$

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

$$0 \leq \alpha_i \leq C$$

与之前相比，只是多了个约束条件而已，仍然可以使用 SMO 来求解

分析软间隔问题的支持向量

结论：

•αi=0 -> 该点为分类正确的点

•0<αi<C -> 该点为软边界上的点

•αi=c -> 该点嵌入了软边界内

- 此时如果ξ<1 ，该点被正确分类
- 此时如果ξ=1 ，该点刚好落在超平面上
- 此时如果ξ>1，该点被错误分类

总结软间隔最大化算法

1.设定惩罚系数 C，构造优化问题

$$\underbrace{min}_{\alpha}\frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j-\sum_{i=1}^{m}\alpha_i$$

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

$$0\le\alpha_i\le C$$

2.用 SMO 算法求出α*

3.计算

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

4.找到全部的支持向量，计算出

$$b_s^*=y_s-\sum_{i=1}^{S}\alpha_i y_i x_i^T x_s$$

5.计算所有的 b_s^* 的平均值得到最终的

$$b^*=\frac{1}{S}\sum_{i=1}^{S}b_s^*$$

判别函数的另一种表达形式

对于线性 SVM 来说，判别函数为 $y=(w^{*T}x+b^*)$

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

由于

所以也有下面这种判别函数的形式：

$$y=(\sum_{i=1}^{m}a_i^* y_i(x_i\cdot x))+b$$

我们得到了一个很好的结论，每一次在计算判别函数结果时需要求得待判断点和所有训练集样本点的内积

非线性支持向量机

如何处理线性不可分问题

升维是一种处理线性不可分问题的方式，我们通过把原始的 x 映射到更高维空间φ(x)上
比如多项式回归：
可以将 2 元特征（x1,x2）映射为 5 元特征（x1,x2,x1*x2,x1^2,x2^2）这样在五元空间中有些二元空间里线性不可分的问题就变得线性可分了
将这个思想引入 SVM 如果对 svm 升维会怎样呢？

SVM 的升维

对于线性 SVM 来说，最优化问题为：

$$\underbrace{min}_{\alpha} \frac{1}{2} \sum_{i=1,j=1}^{m} \alpha_i \alpha_j y_i y_j x_i \bullet x_j - \sum_{i=1}^{m} \alpha_i$$

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

$$0 \le \alpha_i \le C$$

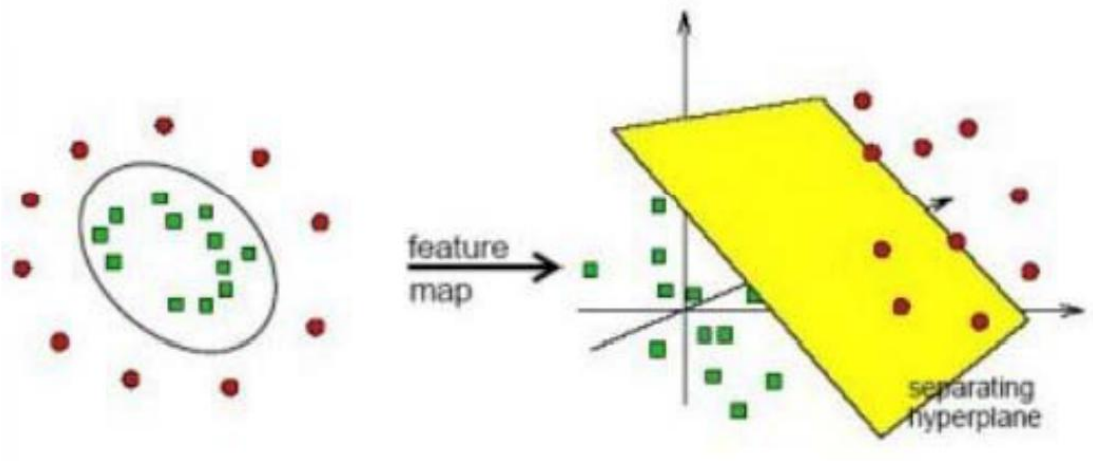
如果使用φ(x) 对训练集升维，最优化问题就变成了：

$$\underbrace{min}_{\alpha} \frac{1}{2} \sum_{i=1,j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(x_i) \bullet \phi(x_j) - \sum_{i=1}^{m} \alpha_i$$

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

$$0 \le \alpha_i \le C$$

升维示意



维度爆炸

看似这种升维方式已经完美解决了线性不可分问题，但是带来了一个新问题
假设就使用多项式回归的方式进行升维：对于二维 x1,x2 升维后的结果是：
x1,x2,x1*x2,x1^2,x2^2
假如是三维数据 x1,x2,x3 呢？
19 维！
假如是 10 维呢？ 维度爆炸
而且我们升维之后还需要做向量的内积，时间空间消耗就更可怕了

引入核函数

我们发现在 SVM 学习过程中

$$\underbrace{min}_{\alpha}\frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i\alpha_j y_i y_j \phi(x_i)\bullet\phi(x_j)-\sum_{i=1}^{m}\alpha_i$$

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

$$0 \le \alpha_i \le C$$

只需要求得 $\phi(x_i)\bullet\phi(x_j)$ 的结果，并不需要知道具体的φ(x)是什么

于是先驱们决定，跳过φ(x)直接定义 $\phi(x_i)\bullet\phi(x_j)$ 的结果，这样既可以达到升维的效果，又可以避免维度爆炸的问题

定义：

$$K(x,z) = \varphi(x)\cdot\varphi(z)$$

此时，对偶问题的目标函数变为了：

$$\underbrace{min}_{\alpha} \frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i,x_j) - \sum_{i=1}^{m}\alpha_i$$

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

$$0 \le \alpha_i \le C$$

判别函数变为了：

$$y = (\sum_{i=1}^{m} a_i^* y_i K(x_i,x)) + b$$

常用核函数

1.线性核函数

$$K(x,z) = x \bullet z$$

2.多项式核函数

$$K(x,z) = (\gamma x \bullet z + r)^d$$

3.高斯核函数

$$K(x,z) = exp(-\gamma||x-z||^2)$$

4.sigmoid 核函数

$$K(x,z) = tanh(\gamma x \bullet z + r)$$

SVM 算法流程总结

1. 选择某个核函数及其对应的超参数
2. 选择惩罚系数 C
3. 构造最优化问题

$$\underbrace{min}_{\alpha}\frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{m}\alpha_i$$

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

$$0\le\alpha_i\le C$$

4. 利用 SMO 算法求解出一组α*
5. 根据α*计算 w*
6. 根据α*找到全部支持向量，计算每个支持向量对应的 bs*
7. 对 bs*求均值得到最后的 b*

学得的超平面为：$\sum_{i=1}^{m}\alpha_i^* y_i K(x,x_i)+b^*=0$

最终的判别函数为：

$$f(x)=sign(\sum_{i=1}^{m}\alpha_i^* y_i K(x,x_i)+b^*)$$

SMO 算法

SMO 思路

我们首先来回顾一下我们要解决的问题，在将 SVM 原始问题转换成为对偶问题之后，我们先求的了 w 和 b 的值，带回到原式中并化简，得到了如下的最优化问题：

$$\min_{\vec{\alpha}}\Psi(\vec{\alpha})=\min_{\vec{\alpha}}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}y_i y_j K(\vec{x}_i,\vec{x}_j)\alpha_i\alpha_j-\sum_{i=1}^{N}\alpha_i$$
$$0\le\alpha_i\le C,\forall i,$$
$$\sum_{i=1}^{N}y_i\alpha_i=0.$$

可以看到，我们共有 N 个决策变量需要处理，每两个决策变量还会以乘积的形式出现在目标函数中，那么这个问题如何求解，就要用到我们 SMO 算法。

其中(xi,yi)表示训练样本数据，xi 为样本特征，yi∈{−1,1}为样本标签，C 为惩罚系数由自己设定。上述问题是要求解 N 个参数(α1,α2,α3,...,αN)，其他参数均为已知，有多种算法可以对上述问题求解，但是算法复杂度均很大。但 1998 年，由 Platt 提出的序列最小最优化算法(SMO)可以高效的求解上述 SVM 问题，它把原始求解 N 个参数二次规划问题分解

成很多个子二次规划问题分别求解，每个子问题只需要求解 2 个参数，方法类似于坐标上升，节省时间成本和降低了内存需求。每次启发式选择两个变量进行优化，不断循环，直到达到函数最优值。

概括来说，SMO 算法主要分为以下两步：

(1) 选择接下来要更新的一对 αi 和 αj:采用启发式的方法进行选择，以使目标函数最大程度地接近其全局最优值

(2) 将目标函数对 αi 和 αj 进行优化，保持其它所有的 αk (k<> i, j) 不变

视为一个二元函数

为了求解 N 个参数(α1,α2,α3,...,αN)，首先想到的是坐标上升的思路，例如求解α1,可以固定其他 N-1 个参数，可以看成关于α1 的一元函数求解，但是注意到上述问题的等式约束条件$\sum_{i=1}^{N} y_i\alpha_i = 0$,当固定其他参数时，参数α1 也被固定，因此此种方法不可用。

SMO 算法选择同时优化两个参数，固定其他 N-2 个参数，假设选择的变量为α1,α2,固定其他参数α3,α4,...,αN,由于参数α3,α4,...,αN 的固定 ,可以简化目标函数为只关于α1,α2 的二元函数，Constant 表示常数项(不包含变量α1,α2 的项)。

$$min\ \Psi(\alpha_1,\alpha_2) = \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_1y_2K_{12}\alpha_1\alpha_2 - (\alpha_1+\alpha_2) + y_1v_1\alpha_1 + y_2v_2\alpha_2 + Constant \quad (1)$$

其中$v_i = \sum_{j=3}^{N}\alpha_j y_j K(x_i,x_j), i = 1,2$

视为一元函数

由等式约束得：

$$\alpha_1y_1 + \alpha_2y_2 = -\sum_{i=3}^{N}\alpha_i y_i = \zeta$$

可见ζ为定值。

等式$\alpha_1y_1 + \alpha_2y_2 = \zeta$两边同时乘以 y1，且$y_1^2 = 1$，得

$$\alpha_1 = (\zeta - y_2\alpha_2)y_1 \quad (2)$$

(2) 式带回到(1)中得到只关于参数α2 的一元函数，由于常数项不影响目标函数的解，以下省略掉常数项 Constant

$$min\ \Psi(\alpha_2) = \frac{1}{2}K_{11}(\zeta-\alpha_2y_2)^2 + \frac{1}{2}K_{22}\alpha_2^2 + y_2K_{12}(\zeta-\alpha_2y_2)\alpha_2 - (\zeta-\alpha_2y_2)y_1 - \alpha_2 + v_1(\zeta-\alpha_2y_2) + y_2v_2\alpha_2 \quad (3)$$

对一元函数求极值点

上式中是关于变量α2 的函数，对上式求导并令其为 0 得：

$$\frac{\partial \Psi(\alpha_2)}{\partial \alpha_2} = (K_{11}+K_{22}-2K_{12})\alpha_2 - K_{11}\zeta y_2 + K_{12}\zeta y_2 + y_1 y_2 - 1 - v_1 y_2 + v_2 y_2 = 0$$

1.由上式中假设求得了α_2的解，带回到(2)式中可求得α_1的解，分别记为$\alpha_1^{new}, \alpha_2^{new}$,优化前的解记为$\alpha_1^{old}, \alpha_2^{old}$;由于参数$\alpha_3, \alpha_4, \ldots, \alpha_N$固定，由等式约束$\sum_{i=1}^{N} y_i \alpha_i = 0$有
$\alpha_1^{old} y_1 + \alpha_2^{old} y_2 = -\sum_{i=3}^{N} \alpha_i y_i = \alpha_1^{new} y_1 + \alpha_2^{new} y_2 = \zeta$

$$\zeta = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 \qquad (4)$$

2.假设SVM超平面的模型为$f(x) = w^T x + b$,上一篇中已推导出w的表达式,将其带入得
$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b$; $f(x_i)$表示样本x_i的预测值,y_i表示样本x_i的真实值，定义E_i表示预测值与真实值之差为

$$E_i = f(x_i) - y_i \qquad (5)$$

3.由于$v_i = \sum_{j=3}^{N} \alpha_j y_j K(x_i, x_j), i = 1, 2,$ 因此

$$v_1 = f(x_1) - \sum_{j=1}^{2} y_j \alpha_j K_{1j} - b \qquad (6)$$

$$v_2 = f(x_2) - \sum_{j=1}^{2} y_j \alpha_j K_{2j} - b \qquad (7)$$

把(4)(6)(7)带入下式中：

$$(K_{11}+K_{22}-2K_{12})\alpha_2 - K_{11}\zeta y_2 + K_{12}\zeta y_2 + y_1 y_2 - 1 - v_1 y_2 + v_2 y_2 = 0$$

化简得: 此时求解出的 α_2^{new} 未考虑约束问题，先记为$\alpha_2^{new,unclipped}$ ：

$$(K_{11}+K_{22}-2K_{12})\alpha_2^{new,unclipped} = (K_{11}+K_{22}-2K_{12})\alpha_2^{old} + y_2\left[y_2 - y_1 + f(x_1) - f(x_2)\right]$$

带入(5)式，并记η=K11+K22−2K12 得：

$$\alpha_2^{new,unclipped} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta} \qquad (8)$$

对原始解修剪

上述求出的解未考虑到约束条件：

$$0 \le \alpha_{i=1,2} \le C$$

$$\alpha_1 y_1 + \alpha_2 y_2 = \zeta$$

在二维平面上直观表达上述两个约束条件

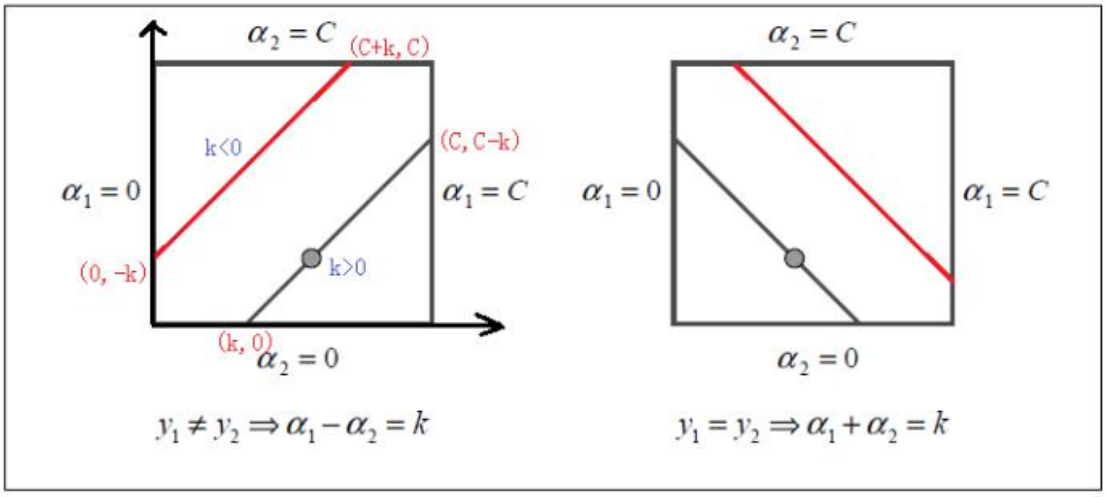


Figure 1. The two Lagrange multipliers must fulfill all of the constraints of the full problem. The inequality constraints cause the Lagrange multipliers to lie in the box. The linear equality constraint causes them to lie on a diagonal line. Therefore, one step of SMO must find an optimum of the objective function on a diagonal line segment.

最优解必须要在方框内且在直线上取得，因此

$$L \leq \alpha_2^{new} \leq H;$$

$$当y_1 \neq y_2时,\ L = max(0, \alpha_2^{old} - \alpha_1^{old}); H = min(C, C + \alpha_2^{old} - \alpha_1^{old})$$

$$当y_1 = y_2时,\ L = max(0, \alpha_1^{old} + \alpha_2^{old} - C); H = min(C, \alpha_2^{old} + \alpha_1^{old})$$

经过上述约束的修剪，最优解就可以记为α_2^{new}了

$$\alpha_2^{new} = \begin{cases} \mathrm{H}, \alpha_2^{new,unclipped} > H \\ \alpha_2^{new,unclipped}, L \leqslant \alpha_2^{new,unclipped} \leqslant H \\ \mathrm{L}, \alpha_2^{new,unclipped} < L \end{cases}$$

求解 α_1^{new}

由于其他 N-2 个变量固定，因此

$$\alpha_1^{old} y_1 + \alpha_2^{old} y_2 = \alpha_1^{new} y_1 + \alpha_2^{new} y_2$$

所以可求得

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new}) \qquad (9)$$

启发式选择变量

① 第一个变量的选择

第一个变量的选择称为外循环，首先遍历整个样本集，选择违反 KKT 条件的αi 作为第一个变量，接着依据相关规则选择第二个变量(见下面分析),对这两个变量采用上述方法进行优化。直到遍历整个样本集后，没有违反 KKT 条件αi，然后退出。

KKT条件

$$\begin{aligned} \alpha_i = 0 &\Rightarrow y^{(i)}(w^T x^{(i)} + b) \geq 1 \\ \alpha_i = C &\Rightarrow y^{(i)}(w^T x^{(i)} + b) \leq 1 \\ 0 < \alpha_i < C &\Rightarrow y^{(i)}(w^T x^{(i)} + b) = 1. \end{aligned}$$

② 第二个变量的选择

SMO 称第二个变量的选择过程为内循环，假设在外循环中找个第一个变量记为α1，第二个变量的选择希望能使α2 有较大的变化，由于α2 是依赖于|E1－E2|,当 E1 为正时，那么选择最小的 Ei 作为 E2,如果 E1 为负，选择最大 Ei 作为 E2，通常为每个样本的 Ei 保存在一个列表中，选择最大的|E1－E2|来近似最大化步长。

阈值 b 的计算

每完成对两个变量的优化后，要对 b 的值进行更新，因为 b 的值关系到 f(x)的计算，即关系到下次优化时 Ei 的计算。

1) 如果

$0 < \alpha_1^{new} < C$

由 KKT 条件$y_1(w^T x_1 + b) = 1$,得到

$\sum_{i=1}^{N} \alpha_i y_i K_{i1} + b = y_1$

由此得：

$$b_1^{new} = y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i1} - \alpha_1^{new} y_1 K_{11} - \alpha_2^{new} y_2 K_{21}$$

由(5)式得，上式前两项可以替换为：

$$y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i1} = -E_1 + \alpha_1^{old} y_1 K_{11} + \alpha_2^{old} y_2 K_{11} + b^{old}$$

得出：

$$b_1^{new} = -E_1 - y_1 K_{11}(\alpha_1^{new} - \alpha_1^{old}) - y_2 K_{21}(\alpha_2^{new} - \alpha_2^{old}) + b^{old}$$

2) 如果

$0 < \alpha_2^{new} < C$

$$b_2^{new} = -E_2 - y_1 K_{12}(\alpha_1^{new} - \alpha_1^{old}) - y_2 K_{22}(\alpha_2^{new} - \alpha_2^{old}) + b^{old}$$

3) 如果同时满足

$0 < \alpha_i^{new} < C$

则

$$b_1^{new} = b_2^{new}$$

4) 如果同时不满足

$$0 < \alpha_i^{new} < C$$

则

b_1^{new} 与 b_2^{new} 以及它们之间的数都满足 KKT 阈值条件，这时选择它们的中点。

SVM 概率化输出

标准的 SVM 的无阈值输出为

$$f(\mathbf{x}) = h(\mathbf{x}) + b.$$

其中

$$h(\mathbf{x}) = \sum_i y_i \alpha_i k(\mathbf{x}_i, \mathbf{x})$$

Platt 利用 sigmoid-fitting 方法，将标准 SVM 的输出结果进行后处理，转换成后验概率。

$$P(y = 1|f) = \frac{1}{1 + \exp(Af + B)}.$$

A,B 为待拟合的参数 ,f 为样本 x 的无阈值输出。sigmoid-fitting 方法的优点在于保持 SVM 稀疏性的同时，可以良好的估计后验概率。

拟合 sigmoid 模型

用极大似然估计来估计公式中的参数 A,B。

定义训练集为(fi,ti),ti 为目标概率输出值，定义为

$$t_i = \frac{y_i + 1}{2}.$$

yi 为样本的所属类别，取值{-1，1}

极小化训练集上的负对数似然函数

$$\min - \sum_i t_i \log(p_i) + (1 - t_i) \log(1 - p_i),$$

其中

$$p_i = \frac{1}{1 + \exp(Af_i + B)}$$

代码

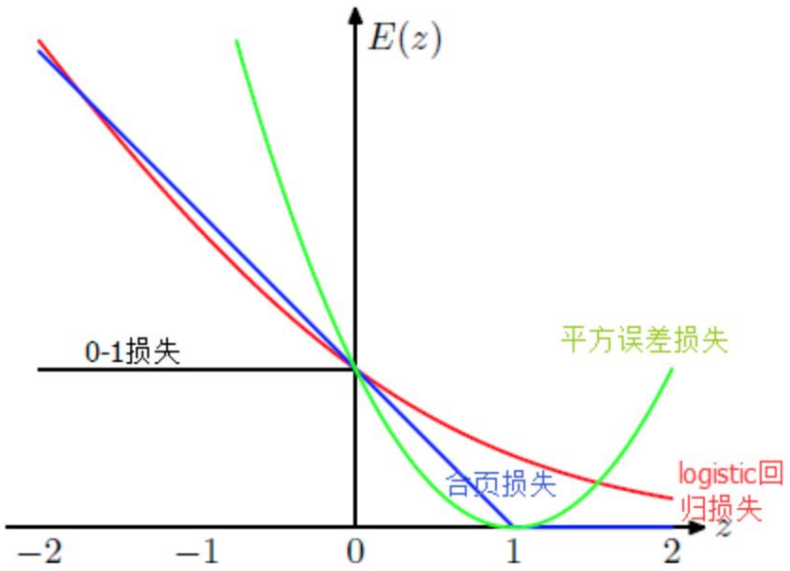
```
import numpy as np
from sklearn.svm import SVC

X = np.array([[-1, -1], [-2, -1], [1, 1], [2, 1]])
y = np.array([1, 1, 2, 2])

clt = SVC(probability = True)
clt.fit(X, y)

print clt.predict([[-0.8, -1]])
print clt.predict_proba([[-0.8, -1]])
```

SVM 合页损失



如图：0-1 损失是二分类问题的真正损失函数，合页损失与 logistic 损失是对 0-1 的损失函数的近似。

hinge loss function

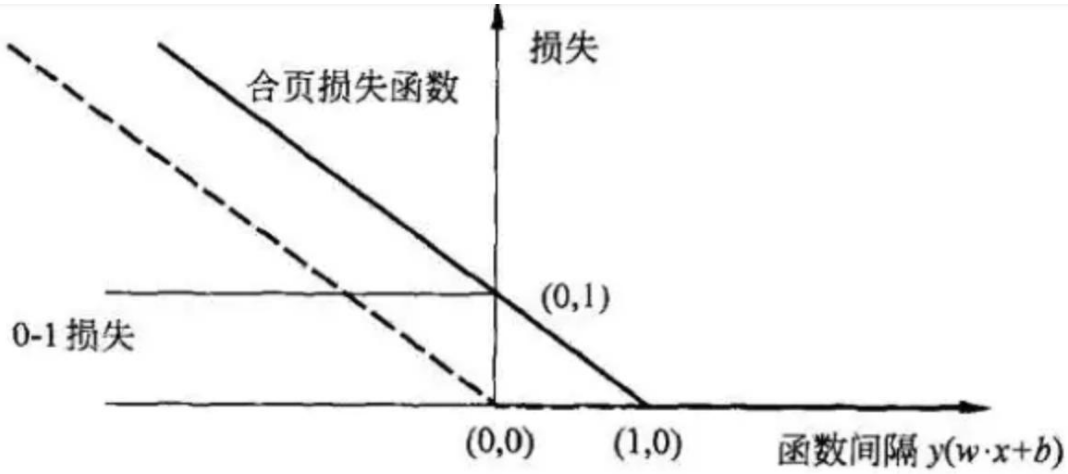
$$L(y \cdot (w \cdot x + b)) = [1 - y(w \cdot x + b)]_+$$

下标”+”表示以下取正值的函数，我们用 z 表示中括号中的部分：

$$[z]_+ = \begin{cases} z, z > 0 \\ 0, z \leq 0 \end{cases}$$

也就是说，数据点如果被正确分类，损失为 0，如果没有被正确分类，损失为 z。

合页损失函数如下图所示：



SVM 的损失函数就是合页损失函数加上正则化项：

$$min_{w,b} \sum_{i=1}^{m} [1 - y^{(i)}(w^T x^{(i)} + b)]_+ + \lambda \|w\|^2$$

上述目标函数中。当$\lambda = \frac{1}{2C}$时。等价于原目标函数$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i$

学习策略 $\min_{f \in H} \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f)$

损失函数. 正则项

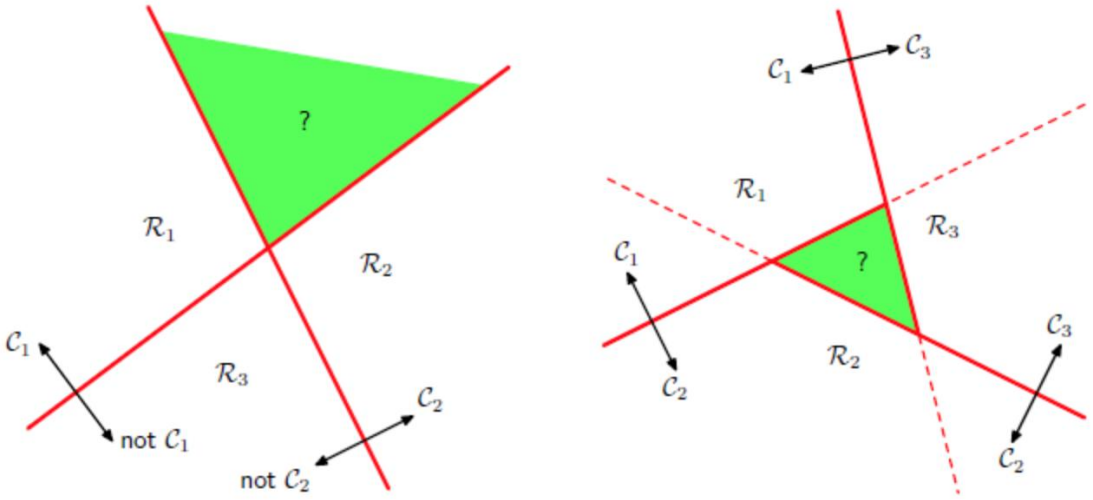
SVM: $\min_{w,b} \sum_{i=1}^{N} [1 - y_i(wx_i+b)]_+ + \lambda\|w\|^2$

hinge损失 L2范数.

令 $[1-y_i(wx_i+b)]_+ = \xi_i \Rightarrow \min_{w,b} \frac{1}{C}(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i)$ 即 SVM

$\lambda = \frac{1}{2C}$

SVM 多分类



one-versus-the-rest

对于 K 个类别的问题。在训练样本上，采用 SVM 训练出 K 个分类器。每一个分类器将训练样本分成 Ki 类与非 Ki 类，然后采用 SVM 训练出模型。如上图所看到的，每一个分类器仅仅能回答是否属于 Ki 的答案。此种方法会造成一个样本数据属于多个类别的情况，上左图阴影部分。

也能够采用：y(x)=maxk yk(x)，即采用最大的函数间隔的那个类别。但不同的分类器有可能尺度不同样，函数距离自然不能作为推断标准。

同一时候，训练样本的不平衡也可能造成分类器有误差。

one-versus-one

在 K 分类的情况下，训练出 K(K−1)2 个分类器，即每两个类别训练出一个分类器，然后依据 K(K−1)2 个分类器的结果，采用投票方法给出预测结果。

此种方法依旧造成部分数据不属于不论什么类的问题，上右图阴影部分所看到的。

SVM 算法小结

SVM 算法是一个很优秀的算法，在集成学习和神经网络之类的算法没有表现出优越性能之前，SVM 算法基本占据了分类模型的统治地位。目前在大数据时代的大样本背景下，SVM 由于其在大样本时超级大的计算量，热度有所下降，但仍然是一个常用的机器学习算法。

优点

1) 解决高维特征的分类问题和回归问题很有效，在特征维度大于样本数时依然有很好的效果。
2) 仅仅使用一部分支持向量来做超平面的决策，无需依赖全部数据。
3) 有大量的核函数可以使用，从而可以很灵活的来解决各种非线性的分类回归问题。
4) 样本量不是海量数据的时候，分类准确率高，泛化能力强。

缺点

1) 如果特征维度远远大于样本数，则 SVM 表现一般。
2) SVM 在样本量非常大，核函数映射维度非常高时，计算量过大，不太适合使用。
3) 非线性问题的核函数的选择没有通用标准，难以选择一个合适的核函数。
4) SVM 对缺失数据敏感。

SVM 对比逻辑回归

1、LR 采用 log 损失，SVM 采用合页损失。
2、LR 对异常值敏感，SVM 对异常值不敏感。
3、在训练集较小时，SVM 较适用，而 LR 需要较多的样本。
4、LR 模型找到的那个超平面，是尽量让所有点都远离它，而 SVM 寻找的那个超平面，是只让最靠近中间分割线的那些点尽量远离，即只用到那些支持向量的样本。
5、对非线性问题的处理方式不同，LR 主要靠特征构造，必须组合交叉特征，特征离散化。SVM 也可以这样，还可以通过 kernel。
6、SVM 更多的属于非参数模型，而 logistic regression 是参数模型，本质不同。其区别就可以参考参数模型和非参模型的区别

如何选择

那怎么根据特征数量和样本量来选择 SVM 和 LR 模型呢？Andrew NG 的课程中给出了以下建议：
如果 Feature 的数量很大 ,跟样本数量差不多 ,这时候选用 LR 或者是 Linear Kernel 的 SVM
如果 Feature 的数量比较小 ,样本数量一般 ,不算大也不算小 ,选用 SVM+Gaussian Kernel
如果 Feature 的数量比较小 ,而样本数量很多 ,需要手工添加一些 feature 变成第一种情况。
(LR 和不带核函数的 SVM 比较类似。)

SVM 回归

SVM 回归模型的损失函数度量

我们知道 SVM 分类模型的目标函数是

$$\min \frac{1}{2}\|\omega\|_2^2$$

同时要让训练集中的各个样本点尽量远离自己类别一侧的支持向量，即约束条件是

$$y_i(\omega \cdot \emptyset(x_i) + b) \geq 1$$

如果加上一个松弛变量

$$\varepsilon_i \geq 0$$

则目标函数变成

$$\min \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{m}\varepsilon_i$$

对应的约束条件变成

$$y_i(\omega \cdot \emptyset(x_i) + b) \geq 1 - \varepsilon_i$$

对于回归模型，优化目标函数和分类模型保持一致，依然是

$$\min \frac{1}{2}\|\omega\|_2^2$$

但是约束条件不同。我们知道回归模型的目标是让训练集中的每个样本点

$$(x_i, y_i)$$

尽量拟合到一个线性模型上。对于一般的回归模型，我们是用均方误差作为损失函数的，但 SVM 不是这样定义损失函数的。

$$y_i = \omega \cdot \emptyset(x_i) + b$$

SVM 需要定义一个常量

$$\epsilon > 0$$

对于某个样本点

$$(x_i, y_i)$$

如果

$$|y_i - \omega \cdot \emptyset(x_i) - b| \leq \epsilon$$

则完全没有损失；如果

$$|y_i - \omega \cdot \emptyset(x_i) - b| > \epsilon$$

则对应的损失为

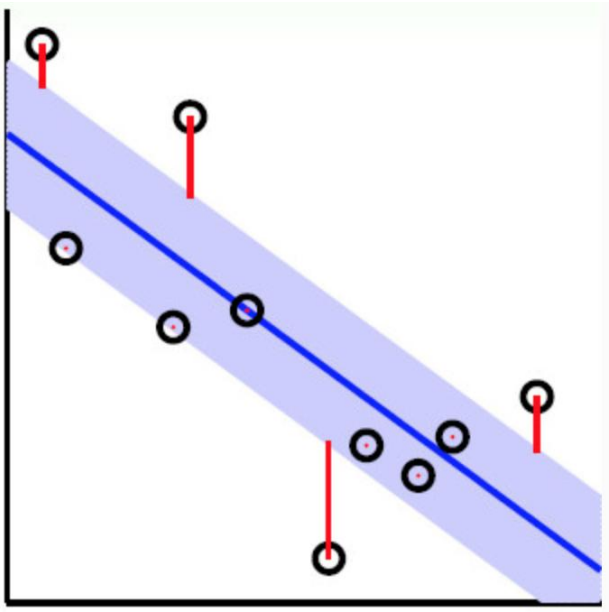
$$|y_i - \omega \cdot \emptyset(x_i) - b| - \epsilon$$

这个损失函数和均方误差不同，如果是均方误差，则只要

$$y_i - \omega \cdot \emptyset(x_i) - b \neq 0$$

就会有损失

如下图所示，在蓝色带里面的点都是没有损失的，但是外面的点是有损失的，损失大小为红色线的长度。



总结一下，SVM 回归模型的损失函数度量为：

$$err(x_i, y_i) = \begin{cases} 0, & |y_i - \omega \cdot \emptyset(x_i) - b| \leq \epsilon \\ |y_i - \omega \cdot \emptyset(x_i) - b| - \epsilon, & |y_i - \omega \cdot \emptyset(x_i) - b| > \epsilon \end{cases}$$

SVM 回归模型的目标函数的原始形式

前面我们已经得到了 SVM 回归模型的损失函数度量，现在可以定义目标函数了，如下所示：

$$\min \frac{1}{2}\|\omega\|_2^2 \ \text{s.t.} \ |y_i - \omega \cdot \emptyset(x_i) - b| \leq \epsilon (i = 1,2,\dots,m)$$

和 SVM 分类模型类似，SVM 回归模型也可以对每个样本点

$$(x_i, y_i)$$

加入松弛变量

$$\varepsilon_i \geq 0$$

但是我们这里使用的是绝对值，实际上是两个不等式，也就是说两边都需要松弛变量，我们

定义为

$\varepsilon_i^{\vee}, \varepsilon_i^{\wedge}$

则 SVM 回归模型的损失函数度量在加入松弛变量之后变为：

$$\min \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{m}(\varepsilon_i^{\vee} + \varepsilon_i^{\wedge})$$

$$s.t. \quad -\epsilon - \varepsilon_i^{\vee} \le y_i - \omega \cdot \emptyset(x_i) - b \le \epsilon + \varepsilon_i^{\wedge}$$

$$\varepsilon_i^{\vee} \ge 0, \varepsilon_i^{\wedge} \ge 0 (i = 1,2,\dots,m)$$

和 SVM 分类模型一样，我们也可以用拉格朗日函数将目标优化函数变成无约束的形式，即：

$$L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge})$$
$$= \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{m}(\varepsilon_i^{\vee} + \varepsilon_i^{\wedge}) + \sum_{i=1}^{m}\alpha_i^{\vee}(-\epsilon - \varepsilon_i^{\vee} - y_i + \omega \cdot \emptyset(x_i) + b)$$
$$+ \sum_{i=1}^{m}\alpha_i^{\wedge}(y_i - \omega \cdot \emptyset(x_i) - b - \epsilon - \varepsilon_i^{\wedge}) - \sum_{i=1}^{m}\mu^{\vee}\varepsilon_i^{\vee} - \sum_{i=1}^{m}\mu^{\wedge}\varepsilon_i^{\wedge}$$

其中$\alpha_i^{\vee} \ge 0, \alpha_i^{\wedge} \ge 0, \mu^{\vee} \ge 0, \mu^{\wedge} \ge 0$是拉格朗日系数。

SVM 回归模型的目标函数的对偶形式

前面我们讲到了 SVM 回归模型的目标函数的原始形式，那么我们的目标是：

$$\arg \min_{\omega, b, \varepsilon^{\vee}, \varepsilon^{\wedge}} \left\{ \max_{\mu^{\vee} \ge 0, \mu^{\wedge} \ge 0, \alpha^{\vee} \ge 0, \alpha^{\wedge} \ge 0} L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge}) \right\}$$

和 SVM 分类模型一样，这个优化目标也满足 KKT 条件，也就是说，我们可以通过拉格朗日对偶将优化问题转化为等价的对偶问题来求解，如下所示：

$$\arg \max_{\mu^{\vee} \ge 0, \mu^{\wedge} \ge 0, \alpha^{\vee} \ge 0, \alpha^{\wedge} \ge 0} \left\{ \min_{\omega, b, \varepsilon^{\vee}, \varepsilon^{\wedge}} L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge}) \right\}$$

我们首先求优化函数对于支持向量机（SVM）的回归模型的极小值，可以通过求偏导数得到：

$$\frac{\partial}{\partial \omega} L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge}) = \omega + \sum_{i=1}^{m} \alpha_i^{\vee} \phi(x_i) - \sum_{i=1}^{m} \alpha_i^{\wedge} \phi(x_i)$$

$$= \omega - \sum_{i=1}^{m} (\alpha_i^{\wedge} - \alpha_i^{\vee}) \phi(x_i) = 0 \Rightarrow \omega = \sum_{i=1}^{m} (\alpha_i^{\wedge} - \alpha_i^{\vee}) \phi(x_i)$$

$$\frac{\partial}{\partial b} L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge}) = \sum_{i=1}^{m} \alpha_i^{\vee} - \sum_{i=1}^{m} \alpha_i^{\wedge} = \sum_{i=1}^{m} (\alpha_i^{\vee} - \alpha_i^{\wedge}) = 0$$

$$\Rightarrow \sum_{i=1}^{m} (\alpha_i^{\wedge} - \alpha_i^{\vee}) = 0$$

$$\frac{\partial}{\partial \varepsilon^{\vee}} L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge}) = C - \alpha^{\vee} - \mu^{\vee} = 0$$

$$\frac{\partial}{\partial \varepsilon^{\wedge}} L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge}) = C - \alpha^{\wedge} - \mu^{\wedge} = 0$$

将上面 4 个式子带入 $\min L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge})$ 消去 $\omega, b, \varepsilon^{\vee}, \varepsilon^{\wedge}$ 得到

$$
\begin{aligned}
&\min L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge}) \\
&= \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{m}(\varepsilon_i^{\vee} + \varepsilon_i^{\wedge}) + \sum_{i=1}^{m}\alpha_i^{\vee}(-\epsilon - \varepsilon_i^{\vee} - y_i + \omega \cdot \emptyset(x_i) + b) \\
&+ \sum_{i=1}^{m}\alpha_i^{\wedge}(y_i - \omega \cdot \emptyset(x_i) - b - \epsilon - \varepsilon_i^{\wedge}) - \sum_{i=1}^{m}\mu^{\vee}\varepsilon_i^{\vee} - \sum_{i=1}^{m}\mu^{\wedge}\varepsilon_i^{\wedge} \\
&= \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{m}\varepsilon_i^{\vee} + C\sum_{i=1}^{m}\varepsilon_i^{\wedge} - \sum_{i=1}^{m}\alpha_i^{\vee}\epsilon - \sum_{i=1}^{m}\alpha_i^{\vee}\varepsilon_i^{\vee} - \sum_{i=1}^{m}\alpha_i^{\vee}y_i \\
&+ \sum_{i=1}^{m}\alpha_i^{\vee}\omega \cdot \emptyset(x_i) + b\sum_{i=1}^{m}\alpha_i^{\vee} + \sum_{i=1}^{m}\alpha_i^{\wedge}y_i - \sum_{i=1}^{m}\alpha_i^{\wedge}\omega \cdot \emptyset(x_i) - b\sum_{i=1}^{m}\alpha_i^{\wedge} \\
&- \sum_{i=1}^{m}\alpha_i^{\wedge}\epsilon - \sum_{i=1}^{m}\alpha_i^{\wedge}\varepsilon_i^{\wedge} - C\sum_{i=1}^{m}\varepsilon_i^{\vee} + \sum_{i=1}^{m}\alpha_i^{\vee}\varepsilon_i^{\vee} - C\sum_{i=1}^{m}\varepsilon_i^{\wedge} + \sum_{i=1}^{m}\alpha_i^{\wedge}\varepsilon_i^{\wedge} \\
&= \frac{1}{2}\omega^T\omega - \sum_{i=1}^{m}\alpha_i^{\vee}\epsilon - \sum_{i=1}^{m}\alpha_i^{\vee}y_i + \sum_{i=1}^{m}\alpha_i^{\vee}\omega \cdot \emptyset(x_i) + b\sum_{i=1}^{m}\alpha_i^{\vee} + \sum_{i=1}^{m}\alpha_i^{\wedge}y_i \\
&- \sum_{i=1}^{m}\alpha_i^{\wedge}\omega \cdot \emptyset(x_i) - b\sum_{i=1}^{m}\alpha_i^{\wedge} - \sum_{i=1}^{m}\alpha_i^{\wedge}\epsilon \\
&= \frac{1}{2}\omega^T\omega - \sum_{i=1}^{m}\alpha_i^{\vee}\epsilon - \sum_{i=1}^{m}\alpha_i^{\wedge}\epsilon - \sum_{i=1}^{m}\alpha_i^{\vee}y_i + \sum_{i=1}^{m}\alpha_i^{\wedge}y_i + \sum_{i=1}^{m}\alpha_i^{\vee}\omega \cdot \emptyset(x_i) \\
&+ b\sum_{i=1}^{m}(\alpha_i^{\vee} - \alpha_i^{\wedge}) - \sum_{i=1}^{m}\alpha_i^{\wedge}\omega \cdot \emptyset(x_i) \\
&= \frac{1}{2}\omega^T\omega - \sum_{i=1}^{m}\alpha_i^{\vee}\epsilon - \sum_{i=1}^{m}\alpha_i^{\wedge}\epsilon - \sum_{i=1}^{m}\alpha_i^{\vee}y_i + \sum_{i=1}^{m}\alpha_i^{\wedge}y_i - \omega \\
&\cdot \sum_{i=1}^{m}(\alpha_i^{\wedge} - \alpha_i^{\vee})\emptyset(x_i) \\
&= \sum_{i=1}^{m}(y_i - \epsilon)\alpha_i^{\wedge} - \sum_{i=1}^{m}(y_i + \epsilon)\alpha_i^{\vee} \\
&- \frac{1}{2}\sum_{i=1,j=1}^{m}(\alpha_i^{\wedge} - \alpha_i^{\vee})(\alpha_j^{\wedge} - \alpha_j^{\vee})\emptyset(x_i) \cdot \emptyset(x_j)
\end{aligned}
$$

接着再求拉格朗日乘子 $\mu^{\vee}, \mu^{\wedge}, \alpha^{\vee}, \alpha^{\wedge}$ 的极大值，得到：

$$
\begin{aligned}
&\max \min L(\omega, b, \alpha^{\vee}, \alpha^{\wedge}, \varepsilon^{\vee}, \varepsilon^{\wedge}, \mu^{\vee}, \mu^{\wedge}) \\
&= \max \sum_{i=1}^{m}(y_i - \epsilon)\alpha_i^{\wedge} - \sum_{i=1}^{m}(y_i + \epsilon)\alpha_i^{\vee} \\
&- \frac{1}{2}\sum_{i=1,j=1}^{m}(\alpha_i^{\wedge} - \alpha_i^{\vee})(\alpha_j^{\wedge} - \alpha_j^{\vee})\emptyset(x_i) \cdot \emptyset(x_j)
\end{aligned}
$$

对上式取负号求最小值，得到和 SVM 分类模型类似的求极小值的目标函数，如下所示：

$$\min \frac{1}{2}\sum_{i=1,j=1}^{m}(\alpha_i^{\wedge}-\alpha_i^{\vee})(\alpha_j^{\wedge}-\alpha_j^{\vee})\emptyset(x_i)\cdot\emptyset(x_j)-\sum_{i=1}^{m}(y_i-\epsilon)\alpha_i^{\wedge}+\sum_{i=1}^{m}(y_i+\epsilon)\alpha_i^{\vee}$$

对于这个目标函数，我们依然可以用 SMO 算法来求出$\alpha^{\vee},\alpha^{\wedge}$，进而求出我们的回归模型系数 w 和 b。

SVM 回归模型系数

在 SVM 分类模型中，KKT 条件的对偶互补条件为：

$$\alpha_i(y_i(\omega\cdot\emptyset(x_i)+b)-1)=0$$

而在回归模型中，我们的对偶互补条件变成了：

$$\alpha_i^{\vee}(\epsilon+\varepsilon_i{}^{\vee}+y_i-\omega\cdot\emptyset(x_i)-b)=0$$
$$\alpha_i^{\wedge}(\epsilon+\varepsilon_i{}^{\wedge}-y_i+\omega\cdot\emptyset(x_i)+b)=0$$

根据松弛变量定义条件，如果

$$|y_i-\omega\cdot\emptyset(x_i)-b|\le\epsilon,\ \text{则有}\varepsilon_i{}^{\vee}=0,\varepsilon_i{}^{\wedge}=0$$

此时

$$\epsilon+\varepsilon_i{}^{\vee}+y_i-\omega\cdot\emptyset(x_i)-b\neq 0,\epsilon+\varepsilon_i{}^{\wedge}-y_i+\omega\cdot\emptyset(x_i)+b\neq 0$$

那么要满足对偶互补条件，只有

$$\alpha_i^{\vee}=0,\alpha_i^{\wedge}=0$$

我们定义样本系数

$$\beta_i=\alpha_i^{\wedge}-\alpha_i^{\vee}$$

根据上面 w 的计算式

$$\omega=\sum_{i=1}^{m}(\alpha_i^{\wedge}-\alpha_i^{\vee})\emptyset(x_i)$$

我们发现此时

$$\beta_i=0$$

也就是说 w 不受这些在误差范围内的点的影响。对于在边界上或者在边界外的点

$$\alpha_i^{\vee}\neq 0,\alpha_i^{\wedge}\neq 0$$

此时

$$\beta_i\neq 0$$