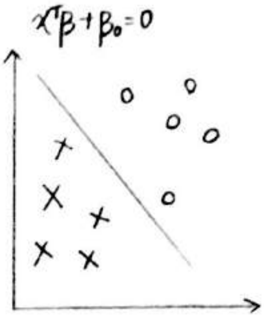
SVM

二类分类模型　　特征空间上的间隔最大的线性分类器

使间隔最大化 ⇒ 凸二次规划问题的求解



(Linear - Seperable)

例　现有一个二维平面，平面上有两种不同的数据，分别是圆圈和叉叉，由于这些数据是线性可分的，所以可以用一条直线将这两类数据分开，这条直线就相当于一个**超平面**。超平面一边的数据点所对应的y全是 -1，另一边全是1.

Define a hyperplane by

$$\{x: f(x) = x^T\beta + \beta_0 = 0\}$$

也就是说

$f(x)=0 \Rightarrow x$ 就是位于超平面上的点

$>0 \Rightarrow$ 对应 $y=1$ 的数据点

$<0 \Rightarrow$ 对应 $y=-1$ 的数据点

在预测时，遇到一个新的数据点 x，将 x 代入 $f(x)$，

$f(x)>0 \Rightarrow x$ 的类别为1

$f(x)<0 \Rightarrow x$ 的类别为-1

我们的问题就转变成了如何基于训练数据集找到一个**最好**的超平面.

最好：这个超平面离数据间隔最大

Functional Margin

$\hat{\gamma}_i = y_i f(x_i)$　　$i=1,\cdots,N$　　(x_i, y_i) 是训练集的数据

correct classification iff $y_i f(x_i) > 0$

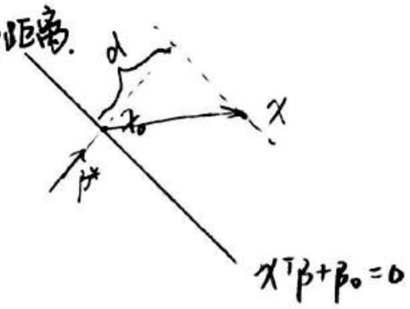
因此直观地，我们希望 $\hat{\gamma} = \min \hat{\gamma}_i$ $(i=1,\cdots,N)$ 越大越好

但是如果成比例改变 β 和 β_0，$f(x_i)$ 也会改变，而 Hyperplane 没有改变 ⇒ Functional Margin 还远远不够

⇒ Geometric Margin (真正定义点到超平面的距离)

(以二维平面为例)

图中 d 就是我们要求的平面内任意一点 x 到直线 $x^T\beta+\beta_0=0$ 的距离.



几何知识：

1. 对于直线 $x^T\beta+\beta_0=0$，我们有对应的单位法向量 $\beta^* = \frac{\beta}{\|\beta\|}$ (Unit Normal Vector)

2. d 就是向量 $(x-x_0)$ 在 β^* 方向上的投影

所以这段距离就可以表示为

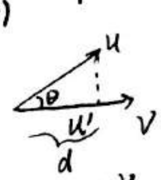
$$(x-x_0)^T\beta^* = (x-x_0)^T\frac{\beta}{\|\beta\|}$$

*单位向量模为1

x_0 是 $x^T\beta+\beta_0=0$ 上一点，$x_0^T\beta = -\beta_0$

$$= \frac{(x^T\beta+\beta_0)}{\|\beta\|}$$

$$= \frac{1}{\|\beta\|} f(x)$$



$u' = d\frac{v}{|v|}$

$$\begin{cases} d = u\cos\theta \\ \cos\theta = \frac{u \cdot v}{|u||v|} \end{cases}$$

$\Rightarrow u' = \frac{u \cdot v}{|v|^2} v$　　$|u'| = |u|\cos\theta = |u|\frac{u\cdot v}{|u||v|} = \frac{u \cdot v}{|v|}$

eg. $x+2y+2z=9$

$A=(1,2,2)$ normal vector

$\frac{A}{|A|} = \frac{(1,2,2)}{\sqrt{1^2+2^2+2^2}}$ unit normal vector

$\vec{a} \cdot \vec{b} = \vec{a}\vec{b}^T$

This was worked out for the case of a positive training example. (分界线的右边)

More generally, define Geometric Margin as

$$\tilde{\gamma} = \frac{y \cdot f(x)}{\|\beta\|} = \frac{\hat{\gamma}}{\|\beta\|} \qquad = \min \frac{y_i f(x_i)}{\|\beta\|} \rightarrow \frac{y_i f(x_i)}{\|\beta\|} \geq \tilde{\gamma}$$

因此我们的问题就转变成了

寻找 Maximum Margin Classifier

$$\max \tilde{\gamma} \quad s.t \quad \frac{y_i f(x_i)}{\|\beta\|} \geq \tilde{\gamma} \quad i=1,\cdots,N$$

令函数间隔等于1 (ie. $\hat{\gamma}=1$)

因为函数间隔的变化并不会移动平面
所以令其等于1不会影响目标函数的
优化

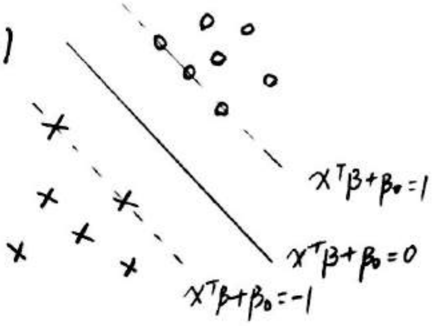
$$\Rightarrow \max \frac{1}{\|\beta\|} \quad s.t \quad y_i f(x_i) \geq 1 \quad i=1,\cdots,N$$

如图所示，中间的实线便是寻找到的最优超平面 (Optimal Hyperplane)

虚线边界上的点就是 Support Vector.

Support Vector 满足 $y f(x) = 1$ 而对于所有不是 Support Vector 的点

显然 $y f(x) > 1$.



具体求解

$$\max \frac{1}{\|\beta\|} \quad s.t. \quad y_i f(x_i) \geq 1 \quad i=1,\cdots,N$$

$$\Leftrightarrow \min \frac{1}{2}\|\beta\|^2 \quad s.t. \quad y_i f(x_i) \geq 1 \quad i=1,\cdots,N$$

现在的目标函数是二次的，约束条件是线性的 ⇒ 凸二次规划问题 (Quadratic Programming)

由于这个问题的特殊结构，可以通过拉格朗日对偶性 (Lagrange Duality) 变换到

对偶变量 (Dual variable) 的优化问题

因此我们的问题就转变成了求解对偶问题 (Dual Problem)

这就是线性可分条件下
支持向量机的对偶算法
这样做的优点在于
① 对偶问题往往更容易求解
② 自然的引入 kernel
⇒ 非线性分类

Lagrange Duality (通过 L 函数将约束条件融合到目标函数
只用一个函数表达式表达我们的问题)

$$L(\beta, \beta_0, \alpha) = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i (x_i^T\beta + \beta_0) - 1\right)$$

然后令 $\theta(\beta) = \max_{\alpha_i \geq 0} L(\beta, \beta_0, \alpha)$

容易验证，当某个约束条件不满足时，例如 $y_i(x_i^T\beta+\beta_0) < 1$ $\alpha_i > 0$ $\alpha_i y_i \cdots < 0$ 令 $\alpha_i = \infty$ $\theta(\beta) = \infty$

当所有条件都满足时 $\alpha_i y_i \cdots \geq 0$

则最优值为 $\frac{1}{2}\|\beta\|^2$ (即最初要最小化的量)

因此 在约束条件下 $\min \frac{1}{2}\|\beta\|^2$

$\Leftrightarrow$ 最小化 $\theta(\beta)$ $(\alpha_i \ge 0,\ i=1,\cdots,N)$　　因为如果约束条件不满足 $\Rightarrow \theta(\beta)=\infty$

$\Leftrightarrow \min\limits_{\beta,\beta_0} \theta(\beta) = \min\limits_{\beta,\beta_0} \max\limits_{\alpha_i \ge 0} L(\beta,\beta_0,\alpha) = p^*$　　p^*表示这个问题的最优解

把 min 和 max 互换 $\Rightarrow$ 原始问题的对偶问题

$\Leftrightarrow \max\limits_{\alpha_i \ge 0} \min\limits_{\beta,\beta_0} L(\beta,\beta_0,\alpha) = d^*$　　$d^* \le p^*$ 在满足某些条件的情况下 两者等价

求解对偶问题

1. 固定α，让L关于β,β_0最小化

$\frac{\partial L}{\partial \beta}=0 \Rightarrow \beta = \sum_{i=1}^N \alpha_i y_i x_i$　$\xrightarrow{\text{代入}L}$　$L(\beta,\beta_0,\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i,j=1}^N \alpha_i\alpha_j y_i y_j x_i^T x_j$　(具体推导跳过 太难了)

$\frac{\partial L}{\partial \beta_0}=0 \Rightarrow \beta_0 = \sum_{i=1}^N \alpha_i y_i = 0$

2. 求对α的极大

$$\max \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i,j=1}^N \alpha_i\alpha_j y_i y_j x_i^T x_j$$
$$\text{s.t.}\quad \alpha_i \ge 0 \quad \sum_{i=1}^N \alpha_i y_i = 0$$

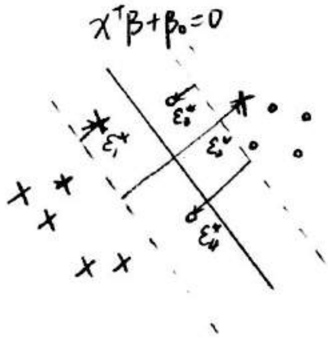
$\Rightarrow \hat{\alpha}_i$

$\hat{\beta} = \sum \hat{\alpha}_i y_i x_i$　　$\hat{\beta}_0 = -\frac{1}{2}\{\max_{i:y_i=-1} x_i^T\hat{\beta} + \min_{i:y_i=1} x_i^T\hat{\beta}\}$

3. 利用SMO算法求解α

到此为止呢，我们就能通过$\{x_i,y_i\}_{i=1:N}$找到$\hat{\alpha}_i$ (support vectors) $\Rightarrow$ 找到 Hyperplane $\{x: f(x) = x^T\hat{\beta}+\hat{\beta}_0 = 0\}$

(针对于线性可分的情况)



Non-Seperable Case

原来的约束条件是

$$y_i f(x_i) \ge 1 \quad i=1,\cdots,N$$

现在变成了

$$y_i f(x_i) \ge 1-\varepsilon_i \quad (i=1,\cdots,N) \quad \varepsilon_i \ge 0 \text{ slack variable}$$

自然而然，我们是想少犯错误 所以要给ε_i加限制

$\Rightarrow \min \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^N \varepsilon_i$　　C — cost 可以理解为犯错误的成本 需要调参

s.t. $y_i f(x_i) \ge 1-\varepsilon_i$　　$C \to 0$ 不介意犯错误 underfitting

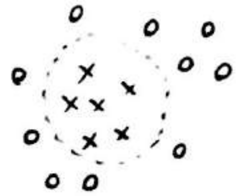
$\varepsilon_i \ge 0$ $i=1,\cdots,N$　　$C \to \infty$ 容易 overfitting (因为一点错误都不要犯)

$$\Leftrightarrow \max \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

$$\text{s.t.}\quad 0\le\alpha_i\le C \quad \sum_{i=1}^{N}\alpha_i y_i = 0$$

唯一区别是多了一个上限 C.

如果数据本身就是线性不可分的呢？比如说 ⇒



首先回顾一下

对一个数据点 x 进行分类，实际上是 plug in ⇒ $f(x)=x^T\beta+\beta_0$ 根据正负号来划分的

因为 $\beta=\sum_{i=1}^{N}\alpha_i y_i x_i$

$$\Rightarrow f(x) = \left(\sum\alpha_i y_i x_i\right)^T x + \beta_0$$

$$= \sum_{i=1}^{N}\alpha_i y_i \langle x_i, x\rangle + \beta_0$$

对于新点x的预测，只需要计算它与训练数据的内积即可（$\langle\cdot,\cdot\rangle$ 表示向量内积.）

另外，非支持向量对应的 α 等于 0。这是因为

$$\max L(\beta,\beta_0,\alpha) = \max \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N}\alpha_i\left(y_i(x^T\beta+\beta_0)-1\right)$$

对于非支持向量来说 $y_i(x^T\beta+\beta_0)-1>0$ $\alpha_i\ge 0$ $-\sum(\alpha_i y_i f(x_i)-1)\le 0$ 为了满足最大化 ⇒ $\alpha_i=0$.

对于非线性的情况 选择一个 kernel function $K(\cdot,\cdot)$ ⇒ 将数据映射到高维空间 ⇒ 在高维空间中构造出最优分离超平面

比如说，一堆数据在二维空间无法划分，从而映射到三维空间里划分.

引入 $K(\cdot,\cdot)$ 后，分类函数就转变成了

$$f(x) = \left(\sum_{i=1}^{N}\alpha_i y_i K(x_i,x)\right) + \beta_0$$

其中 α 由如下对偶问题计算得到

$$\max_{\alpha} \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$

$$\text{s.t.}\quad \alpha_i\ge 0$$

$$\sum\alpha_i y_i = 0$$

核函数的优点

1. 将特征进行从低维到高维的转换
2. 事先在低维上进行计算，将实质上的分类效果表现在高维上，避免了直接在高维空间中的复杂计算

不同的核函数

- 多项式 $K(x_1,x_2)=(\langle x_1,x_2\rangle+R)^d$
- 高斯 $K(x_1,x_2)=\exp\left(-\frac{\|x_1-x_2\|^2}{2\sigma^2}\right)$
- 线性核 $K(x_1,x_2)=\langle x_1,x_2\rangle$ （原始空间的内积，）为了方便一起算