SVM 调研笔记

预备知识

- 拉格朗日对偶原理
- 点到面的距离

SVM 简介

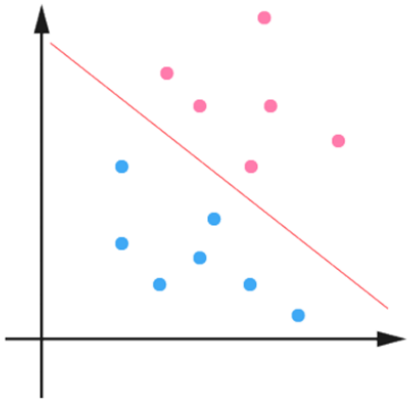
SVM（支持向量机），主要通过支持向量来进行分类（线性和非线性分类）或回归的算法，一直被认为是效果最好的现成可用的分类算法之一；

算法模型

寻找一个超平面，将两类数据分开；

$$f(x)=sign(wx + b)$$

（w.z+b>0 时为 1，<0 时为-1）

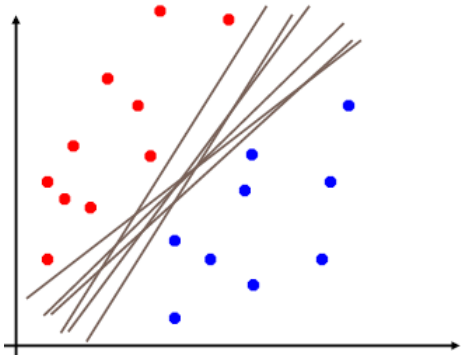


学习策略

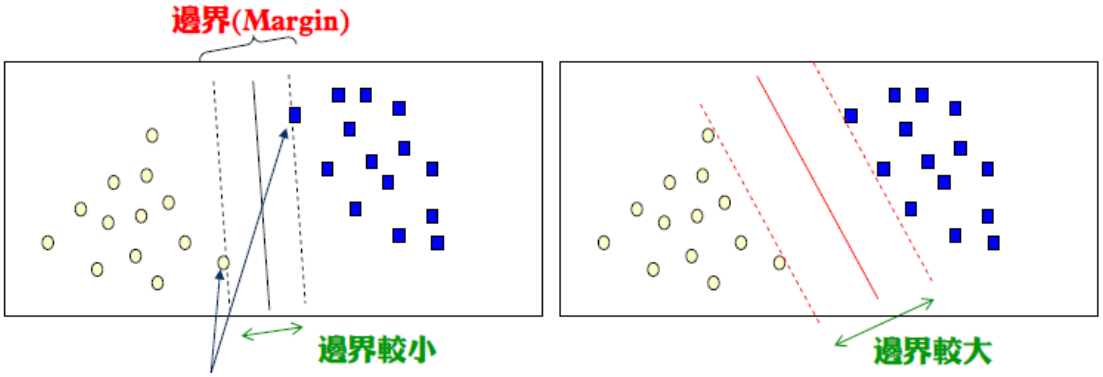
最大化分类间隔

选择哪个一个超平面

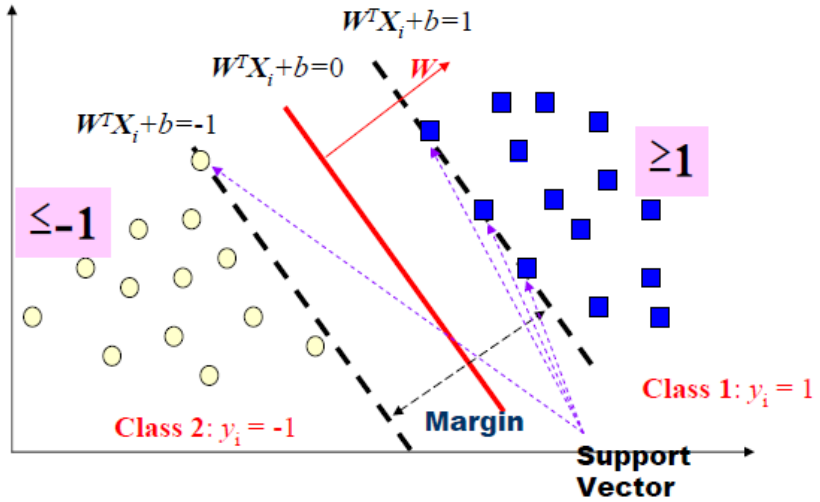
1. 有无数个超平面可以分隔



2. 选择间隔最大的超平面



分类间隔 Margin，分割超平面，支持超平面：

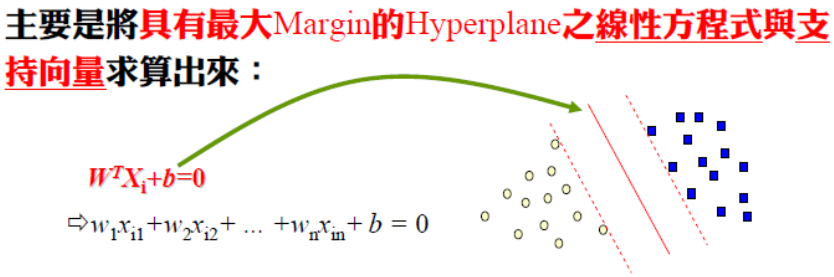


为什么最大化分类间隔

1. 直观上最有效
2. 概率的角度，就是使得置信度最小的点置信度最大
3. 即使我们在选边界的时候犯了小错误，使得边界有偏移，仍然有很大概率保证可以正确分类绝大多数样本

4. 很容易实现交叉验证，因为边界只与极少数的样本点有关
5. 有一定的理论支撑(如 VC 维)
6. 实验结果验证了其有效性

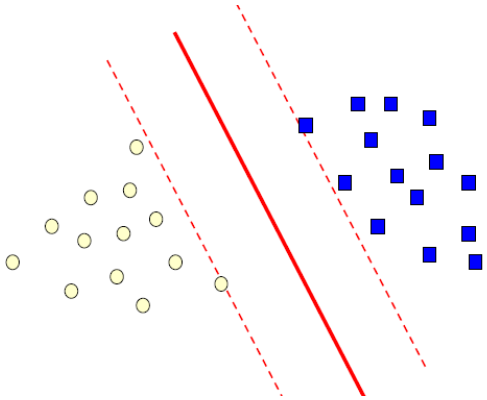
求解问题本质：



后面针对三种数据分布情况分别描述：线性可分，线性不可分，非线性；

线性可分

- 线性可分指通过分割超平面完全可以将数据分开，如下图所示：



- 学习策略：硬间隔（几何间隔）最大化

1. 定义点到超平面的函数间隔：

$$\hat{\gamma} = y(w^{\mathrm{T}} x + b) = y f(x)$$

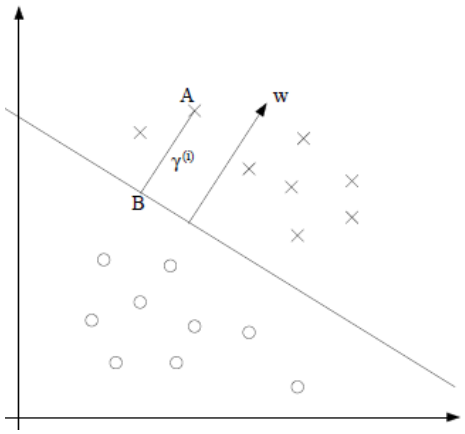
2. 定义样本集到超平面的函数间隔：

$$\hat{\gamma} = \min \hat{\gamma}_i, i = 1, 2, \cdots, n$$

3. 定义样本集到超平面几何间隔：

$$\tilde{\gamma} = y\gamma = \frac{\hat{\gamma}}{\|w\|}$$

（直观上点到超平面距离）



原始问题

1. 最大化几何间隔：

$$\max_{w,b} \frac{\hat{\gamma}}{\|w\|}$$
$$s.t. \quad y_i\left(w \cdot x_i + b\right) \geq \hat{\gamma}, \quad i = 1,2,\cdots N$$

2. 变换上式：

$$\max_{w,b} \frac{1}{\left\|\frac{w}{\hat{\gamma}}\right\|}, st. y_i\left(\frac{w}{\hat{\gamma}} \cdot x_i + \frac{b}{\hat{\gamma}}\right) \geq 1 => \max_{w,b} \frac{1}{\|w'\|}, \text{st.} \quad y_i\left(w' \cdot x_i + b'\right) \geq 1$$

$$w' = \frac{w}{\hat{\gamma}}, b' = \frac{b}{\hat{\gamma}}$$

3. 问题转化为：

$$\max \frac{1}{\|w\|}$$
$$s.t. \quad y_i(w^{\mathrm{T}} x_i + b) \geq 1, i = 1, 2, \cdots, n$$

4. 原始问题等价于：

$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$s.t. \quad y_i\left(w \cdot x_i + b\right) \geq 1, \quad i = 1,2,\cdots N$$

（方便求导）

问题求解

- 本质是二次优化问题（目标函数是二次的，约束条件是线性的）
- 求解方式:
  1. 一个是 QP 问题的常规方法；
  2. Lagrange 对偶方法，通过求解对偶问题得到最优 解，这样做的优点在于：一者对偶问题往往更容易求解；二者可以自然的引入核函数，进而推广到非线性分类问题；原问题通过对偶问题求解时，解需要满足 KKT 条件；关于对偶问题的解释参考<统计学习方法>

拉格朗日对偶问题：

1. 定义拉格朗日函数

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1\right)$$

2. L 函数特性

$$\max_{\alpha} L(w,b,\alpha) = \frac{1}{2}\|w\|^2 \text{（满足限制条件）}$$

$$\max_{\alpha} L(w,b,\alpha) = \infty \text{（不满足限制条件）}$$

3. 原始问题转化

$$\min_{w,b} \frac{1}{2}\|w\|^2 \; s.t. \quad y_i(w \cdot x_i + b) \geq 1 \Leftrightarrow \min_{w,b} \max_{\alpha} L(w,b,\alpha)$$

$$\min_{w,b} \max_{\alpha} L(w,b,\alpha) \leq \max_{\alpha} \min_{w,b} L(w,b,\alpha)$$

（满足 Slater 条件时相等）

4. 问题转化为：

$$\max_{\alpha} \min_{w,b} L(w,b,\alpha)$$

先对 w，b 求极小值，然后针对 a 求极大值。

5. 先对 w，b 求极小，对 w，b 求导等于 0，然后得到等式:

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

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

6. 将上式带入 L(w,b,a)：

$$\mathcal{L}(w, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)}.$$

7. 然后对上式求极大：

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$

$$s.t.\ \alpha_i \geq 0, i = 1, 2, \cdots, n$$

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

8. 添加负号后，为**最终对偶求解问题**：

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

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

$$\alpha_i \geq 0, \quad i = 1, 2, \ldots, N$$

9. 通过 SMO 求解 alpha，w，b 的值根据 alpha 得到：

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

$$b^* = y_i - \sum_{i=1}^{N} \alpha_i^* y_i \left(x_i \cdot x_j\right)$$

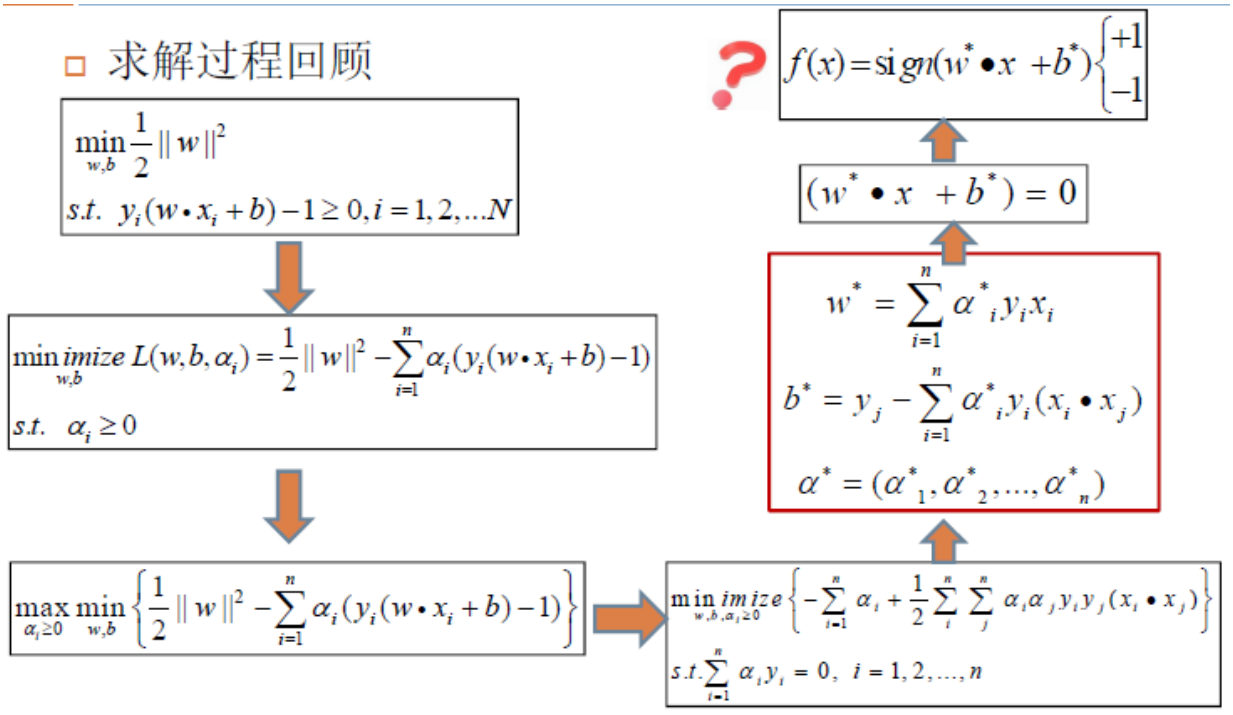
$$f(x)=sign(w^{*}x+b^{*})$$

线性可分支持向量：

- alpha 大于 0 时，根据 KKT 条件，Y(W*X + b) = 1，对应的实例一定在边界上，这些实例点称为支持向量；
- 线性可分情况下 KKT 条件：

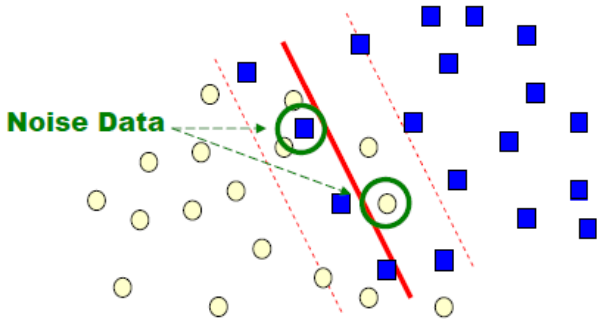
$$\alpha_i(y_i(w\cdot x_i+b)-1)=0$$
$$y_i(w\cdot x_i+b)\geq 1$$
$$\alpha_i\geq 0$$

求解问题回顾



线性不可分

- 线性不可分指的是无法通过一个线性超平面将数据完全分开，如下图：



- 学习策略：软间隔最大化

若数据线性不可分，可以容忍少部分分错，通过增加松弛因子 ξi≥0，使函数间隔加上松弛变量大于等于 1；

原始问题：

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

$$s.t.\ y_i(w^{\mathrm{T}}x_i + b) \geq 1 - \xi_i,\ i = 1, 2, \cdots, n$$

$$\xi_i \geq 0,\ i = 1, 2, \cdots, n$$

1. C 是一个参数（事先确定后），用于控制目标函数中两项（"寻找间隔最大的超平面”和“保证数据点偏差量最小”）之间的权重;
2. ξi (i = 1, 2, • • • , n) 称为松弛变量 Slack Variable ，对应数据点 xi 允许偏离的函数间隔的量。 当然，如果我 们允许 ξi 任意大的话，那任意的超平面都是符合条件的了。所以，我们在原来的目标函数后面加上一项，使得 这些 ξi 的总和也要最小；（对于线性可分，松弛变量 ξi 都为 0 即可）

对偶问题：

1. 拉格朗日函数：

$$L(w,b,\xi,\alpha,\mu) \equiv \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i(w\cdot x_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{N}\mu_i\xi_i$$

2. 类似之前先对 w,b, ξ 求导数，将得到的等式带入 L，得到目标函数：

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

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

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

$$\alpha_i \geq 0$$

$$\mu_i \geq 0, \quad i = 1,2,\ldots N$$

3. 整理上面的条件得到：

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

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

$$0 \leq \alpha_i \leq C, \quad i = 1,2,\ldots N$$

4. 对偶问题解：

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

$$b^* = y_i - \sum_{i=1}^{N}\alpha_i^* y_i\left(x_i\cdot x_j\right)$$

(计算 b*时，需要满足 0< α j<C)

alpha 与样本分布的关系：

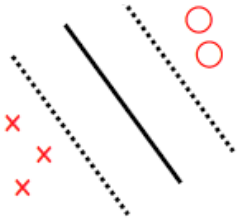
- 线性不可分情况下的 KTT 条件：

$$\begin{cases}\alpha_i^*\left(y_i\left(W^{*T}\Phi(X_i)+b^*\right)-1+\xi_i^*\right)=0\\ \beta_i^*\xi_i^*=0\end{cases}$$

- a 值三种情况：

當$\alpha^*=0$：

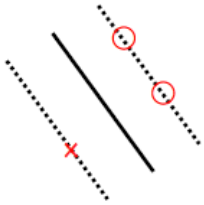
$$\alpha_i^* = 0 \Rightarrow \beta_i^* = C - \alpha_i^* = C$$
$$\Rightarrow \xi_i^* = 0$$



因此，$y_i(W^{*T}\Phi(X_i)+b^*)-1 \geq 0$ **(為了與下一情況做區隔，等號部份不看)⇨**
$y_i(W^{*T}\Phi(X_i)+b^*) > 1$**，代表資料落於支持超平面的兩邊。**

當 $0 < \alpha^* < C$ **時：**

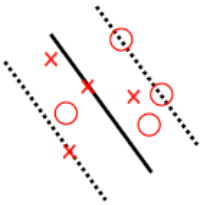
$$0 < \alpha_i^* < C \Rightarrow \begin{cases} 0 < \beta_i^* < C \Rightarrow \xi_i^* = 0 \\ y_i(W^{*T}\Phi(X_i)+b^*)-1+\xi_i^* = 0 \end{cases}$$
$$\Rightarrow y_i(W^{*T}\Phi(X_i)+b^*) = 1$$



代表資料一定是落於支持超平面上。

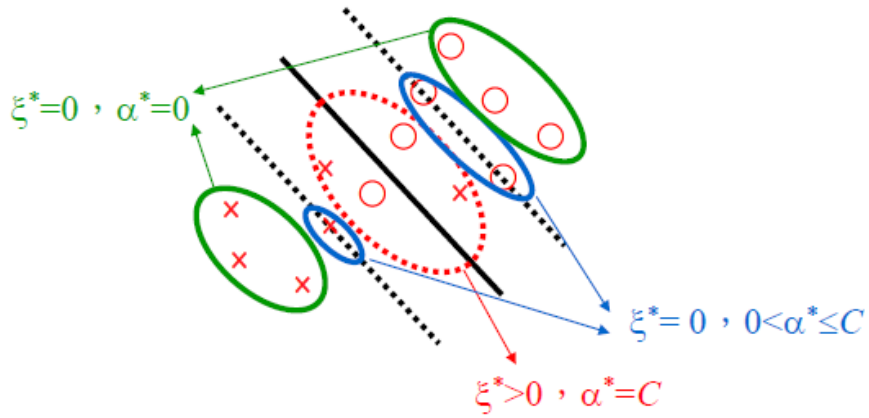
當 $\alpha^* = C$ **時：**

$$\alpha_i^* = C \Rightarrow \begin{cases} \beta_i^* = C - \alpha_i^* = 0 \Rightarrow \xi_i^* \geq 0 \\ y_i(W^{*T}\Phi(X_i)+b^*)-1+\xi_i^* = 0 \end{cases}$$
$$\Rightarrow y_i(W^{*T}\Phi(X_i)+b^*) \leq 1$$



代表資料可能是落於兩個支持超平面的中間($\because \xi_i^* > 0$**)或支持超平面上面(**$\because \xi_i^* = 0$**)。**

线性不可分支持向量：



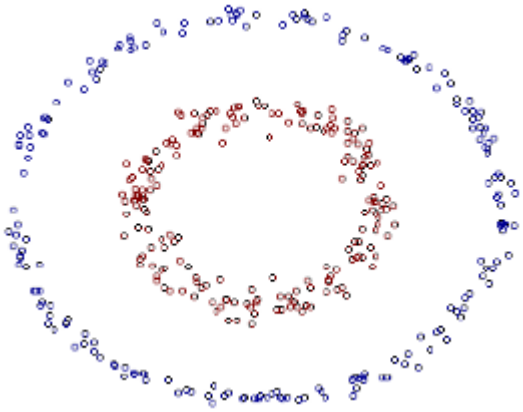
alpha 大于 0 的样本为支持向量，包括在支持超平面上的向量和支持超平面之间的向量，以及分错的向量；

线性 SVM 总结

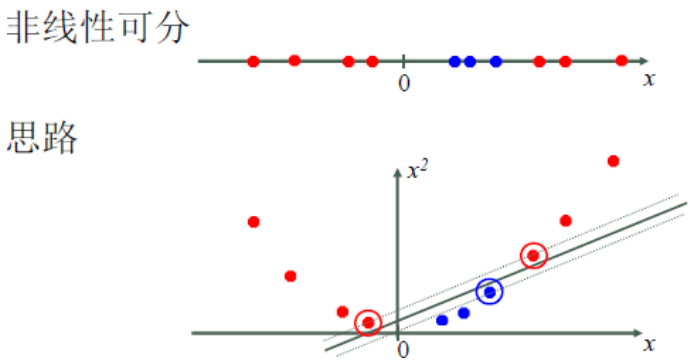
- 分类器是一个超平面 w • x + b = 0；
- “支持向量们”是最重要的训练样本，决定超平面的位 置；
- 通过发现对应 alpha 的是否为 0，二次规化算法可以发现哪些点是“支持向量”；
- 在问题的对偶形式及其解中，训练样本之间的关系都是线性的（内积）；

非线性可分

线性不可分是针对非线性数据，如下图所示：

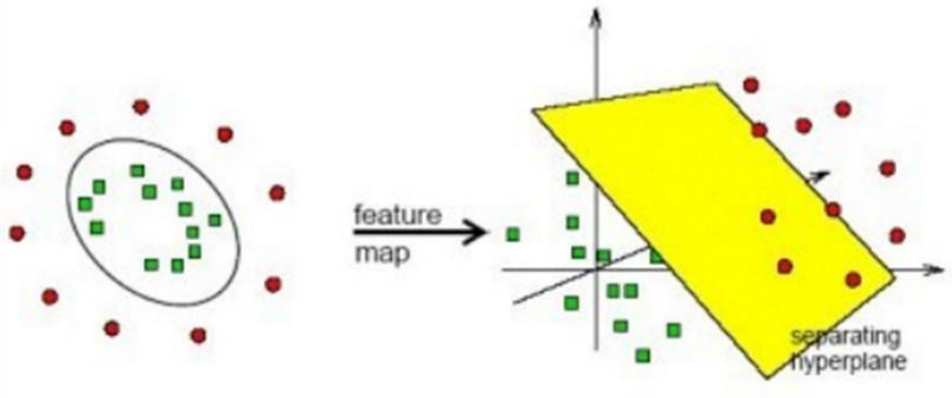


处理思路：



使用核函数，将输入空间映射到特征空间，从而使得原本线性不可分的样本可以在特征空间可分；实际应用中，往往依赖先验领域知识才能选择有效的核函数；

另一个例子：



问题模型

1. 对偶问题实际模型

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b$$

2. 高维空间模型：

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) \rangle + b$$

3. 目标问题：

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \alpha_i \alpha_j y_i y_j \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$$

$$s.t.\ \alpha_i \geq 0, i = 1, 2, \cdots, n$$

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

4. 核函数：
   - 计算两个向量在隐式映射过后的空间中的内积的函数叫做核函数；
   - $\kappa(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) \rangle$，避免将低维映射到高维计算内积，直接在低纬空间计算高维的内积；
   - 核函数满足条件：

Mercer定理

- 如果函数K是 $R^n * R^n -> R$ 上的映射（也就是从两个n维向量映射到实数域）。如果K是一个有效核函数（也称为Mercer核函数），那么当且仅当对于训练样例，其相应的核函数矩阵K是对称半正定的。

$$K_{ij} = K(x_i, x_j)$$

表明为了证明K是有效的核函数，那么我们不用去寻找投影函数ϕ，而只需要在训练集上求出各个K_{ij}，然后判断矩阵K是否是半正定（使用左上角主子式大于等于零等方法）即可

常用核函数

- 常见核函数

高斯核

$$\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left\{-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2}{2\sigma^2}\right\}$$

多项式

$$\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle + R)^d$$

常见问题：

1. 如何选择核函数 K
   - 默认选高斯（RBF）或多项式核函数，否则再尝试其它核函数
   - 使用专家先验知识，设计相应的样本内积计算方法
2. 如何选择参数
   - 例如高斯核中的参数
   - 使用交叉验证等方法进行参数选择

SVM 学习策略另一种解释

- 极小化合页损失（hinge loss)

$$\min \sum_{i=1}^{n} [1 - y_i(w \cdot x_i + b)]_+ + \lambda \|w\|^2$$

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

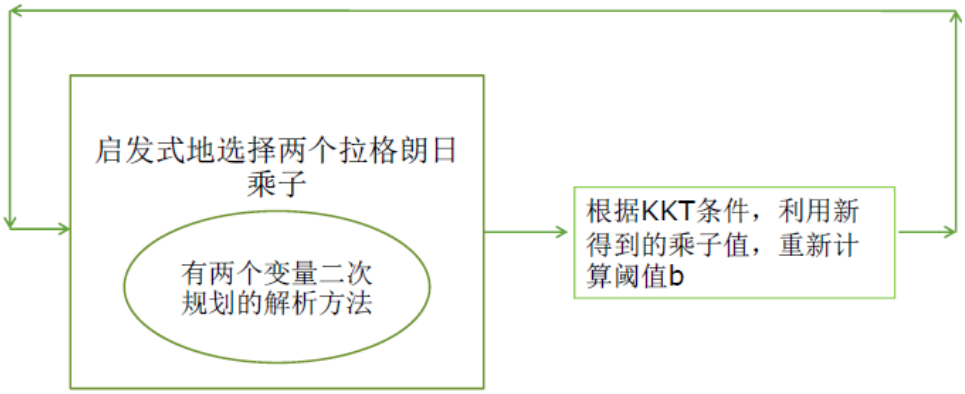
合页损失(hinge loss)函数，即当样本(x_i, y_i)被正确分类且函数间隔$y_i(w \cdot x_i + b)$大于1时，损失为0，否则损失为$1 - y_i(w \cdot x_i + b)$

- 具体证明过程参考<统计学习方法>

学习算法

SMO 算法的启发式思想：

- 如果所有变量都满足此优化问题的 KKT 条件，那么这个最优化问题的解就得到了；
- 不断将原二次规划问题分解为只有两个变量的二次规划问题；并对子问题求解，直到所有变量满足 KKT 条件；



SMO 关键步骤

1. 选择两个变量
   a) 先“扫描”所有乘子，把第一个违反 KKT 条件的作为更新对象，令为 α j；KKT 条件：

$$\begin{aligned} \alpha_i = 0 &\Leftrightarrow y_i u_i \geq 1, \\ 0 < \alpha_i < C &\Leftrightarrow y_i u_i = 1, \\ \alpha_i = C &\Leftrightarrow y_i u_i \leq 1. \end{aligned}$$

   b) 在所有不违反 KKT 条件的乘子中，选择使|Ei – Ej | 最大的 α i;
2. 求解两个变量的二次规划问题（解析求解）

$$\alpha_j^{\text{new}} = \alpha_j^{\text{old}} + \frac{y_j(E_i - E_j)}{\eta}$$

$$\alpha_i^{\text{new}} = \alpha_i + y_i y_j (\alpha_j - \alpha_j^{\text{new, clipped}})$$

$$g(x) = \sum_{i=1}^{N} y_i \alpha_i K(x_i, x) + b$$

$$E_i = g(x_i) - y_i = \left(\sum_{j=1}^{N} y_j \alpha_j K(x_j, x_i) + b \right) - y_i, \quad i = 1,2$$

3. 求解退出条件

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

$$0 \le \alpha_i \le C, \quad i = 1,2,\ldots N$$

$$y_i \cdot g(x_i) = \begin{cases} \ge 1, & \{x_i | \alpha_i = 0\} \\ = 1, & \{x_i | 0 < \alpha_i < C\} \\ \le 1, & \{x_i | \alpha_i = C\} \end{cases}$$

$$g(x_i) = \sum_{j=1}^{N} y_j \alpha_j K(x_j, x_i) + b$$

SVM 分类流程

1. 选择一个核函数 K（用以计算样本内积），并选择对应的参数
2. 选择一个 C 值(参数，控制软间隔程度以及防止过拟合)
3. 利用训练样本，求解二次规划问题(可以使用大量软件包)
4. 根据支持向量与切分面构造切分函数 sign(.)
5. 根据切分函数，对测试样本进行分类

与其他分类算法对比

LR VS SVM

从目标函数来看，区别在于逻辑回归采用的是 logistical loss，svm 采用的是 hinge loss。这两个损失函数的目的都是增加对分类影响较大的数据点的权重，减少与分类关系较小的数据点的权重。SVM 的处理方法是只考虑 support vectors，也就是和分类最相关的少数点，去学习分类器。而逻辑回归通过非线性映射，大大减小了离分类平面较远的点的权重，相对提升了与分类最相关的数据点的权重。两者的根本目的都是一样的。此外，根据需要，两个方法都可以增加不同的正则化项，如 l1,l2 等等。所以在很多实验中，两种算法的结果是很接近的。

但是逻辑回归相对来说模型更简单，好理解，实现起来，特别是大规模线性分类时比较方便。而 SVM 的理解和优化相对来说复杂一些。但是 SVM 的理论基础更加牢固，有一套结构化风险最小化的理论基础，虽然一般使用的人不太会去关注。还有很重要的一点，SVM 转化为对偶问题后，分类只需要计算与少数几个支持向量的距离，这个在进行复杂核函数计算时优势很明显，能够大大简化模型和计算量。

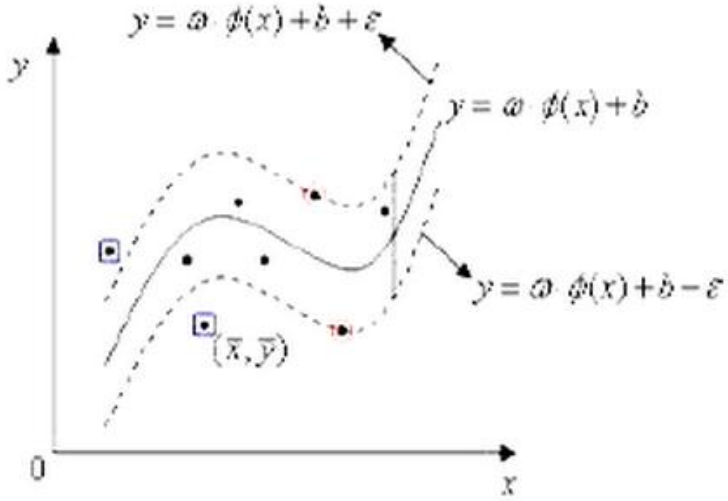
多分类

- 上面所谈到的分类都是 2 分类的情况，当 N 分类的情况下，主要有两种方式，一种是 1 vs (N－1)，一种是 1 vs 1；
  - 前一种方法我们需要训练 N 个分类器，第 i 个分类器是看看是属于分类 i 还是属于分类 i 的补集（出去 i 的 N-1 个分类）；
  - 后一种方式我们需要训练 N * (N － 1) / 2 个分类器，分类器(i, j)能够判断某个点是属于 i 还是属于 j；

1. 这种处理方式不仅在 SVM 中会用到，在很多其他的分类中也是被广泛用到，从林教授（libsvm 的作者）的结论来看，1 vs 1 的方式要优于 1 vs (N － 1)；

SVM 回归

SVM 回归又称 SVR，SVM 试图寻找将两类样本分得最开的超平面，而 SVR 则是试图寻找能准确预测样本点分布的超平面；



模型

$$f(x) = w*x + b$$

与 SVM 不同，SVR 中 f(x)为连续值而不是 1 或-1；

学习策略

所有样本点离目标超平面的“总偏差”最小；

线性支持回归

1. 参考之前 SVM，实际上需要解决的问题为：

$$\text{minimize} \quad \tfrac{1}{2}\|w\|^2$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b & \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i & \leq \varepsilon \end{cases}$$

ε ≥ 0，表示预测值与实际值的最大误差（看图）

2. 类似线性不可分情况，可以容忍部分样本落在 ε 之外：

$$\text{minimize} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\text{subject to} \begin{cases} y_i - w \cdot x_i - b \leq \varepsilon + \xi_i \\ w \cdot x_i + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i,\ \xi_i^* \geq 0 \end{cases}$$

3. 问题求解
   - 采用拉格朗日对偶原理求解，此处不详述；

$$L = \frac{1}{2}\omega \cdot \omega + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) - \sum_{i=1}^{n}\alpha_i[\xi_i + \varepsilon - y_i + f(x_i)]$$

$$-\sum_{i=1}^{n}\alpha_i^*[\xi_i^* + \varepsilon - y_i + f(x_i)] - \sum_{i=1}^{n}(\xi_i\gamma_i + \xi_i^*\gamma_i^*)$$

   - 求得拉格朗日成子后，得到实际拟合函数：

$$f(x) = \omega \cdot x + b = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)x_i \cdot x + b$$

非线性回归

非线性 SVR 的基本思想是通过事先确定的非线性映射将输入向量映射的一个高维特征空间（Hilbert 空间）中，然后在此高维空间中再进行线性回归，从而取得在原空间非线性回归的效果；核心还是利用核函数。

使用核函数后，拟合函数变为：

$$f(x) = \omega \cdot \Phi(x) + b$$
$$= \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x, x_i) + b$$

参考

1. http://blog.csdn.net/v_july_v/article/details/7624837 推荐
2. http://www.cnblogs.com/LeftNotEasy/archive/2011/05/02/basic-of-svm.html
3. http://www.cnblogs.com/wangbogong/p/3148668.html
4. http://www.svms.org/
5. http://jacoxu.com/?p=118
6. http://www.blogjava.net/zhenandaci/archive/2009/02/13/254519.html svm 入门
7. http://blog.csdn.net/techq/article/details/6171688 svm 实现
8. http://blog.csdn.net/nwpuwyk/article/details/41309007 手写公式
9. http://wenku.baidu.com/view/aeba21be960590c69ec3769e.html svm 方法的实现和证明
10. http://wenku.baidu.com/view/dd266fa6941ea76e59fa0438.html svr