支持向量机

机器学习研究室

计算机科学与技术学院
吉林大学

大纲

- 支持向量机(SVM)的概念与原理
- 线性可分支持向量机——硬间隔最大化
- 支持向量机的优化算法
- 线性支持向量机——软间隔最大化
- 非线性支持向量机——核方法
- 支持向量机回归

支持向量机(SVM)的概念与原理

支持向量机(SVM)的概念

- 支持向量机（Support Vector Machines, SVM）
- 二类分类模型。它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机。
- 支持向量机还包括核技巧，这使它成为实质上的非线性分类器。
- 支持向量机的学习策略就是间隔最大化，可形式化为一个求解凸二次规划(convex quadratic programming)的问题，也等价于正则化的Hinge损失函数的最小化问题。支持向量机的学习算法是求解凸二次规划的最优化算法。

支持向量机(SVM)的概念

- **支持向量机的类型**
- 线性可分支持向量机(linear support vector machine in linearly separable case).
  - 硬间隔最大化(hard margin maximization);
- 线性支持向量机(linear support vector machine)
  - 训练数据近似线性可分时，通过软间隔最大化(soft margin maximization);
- 非线性支持向量机(non-linear support vector machine)
  - 当训练数据线性不可分时，通过使用核技巧(kernel trick)及软间隔最大化。

支持向量机(SVM)的概念

- 当输入空间为欧氏空间或离散集合、特征空间为希尔伯特空间时，核函数(kernel function)表示将输入从输入空间映射到特征空间得到的特征向量之间的内积;
- 通过使用核函数可以学习非线性支持向量机，等价于隐式地在高维的特征空间中学习线性支持向量机，这样的方法称为核技巧;
- 核方法(kernel method)是比支持向量机更为一般的机器学习方法。

SVM的理论基础

- 《说文》中记载，模，法也；式，法也。
  - 简单来说就是一种规律。
- 英文中模式pattern这个词的意思有两层
  - 第一层是代表事物（个体或一组事物）的模板或原型；
  - 第二层是表征事物特点的特征或性状的组合。
- 模式可以看做是对象的组成成分或影响因素间存在的规律性关系，或者是因素间存在的确定性或随机性规律的对象、过程或事件的集合。因此，也有人把模式成为模式类，模式识别也被称作为模式分类（Pattern Classification）。
- **模式识别**研究的重点是如何通过一系列数学方法**让机器来实现类人的识别（认知）能力。**

SVM的理论基础

- 目标：确定最优分类器$y = f(x, w)$
- 满足条件：期望风险最小

$$R(w) = \int L(y, f(x, w)) dF(x, y)$$

其中，$L(y, f(x, w))$称为损失函数

- 由于数据实际分布$F(x, y)$不可知，因此从观察数据中最小化风险→经验风险最小(大数定律)
- 对于N个观测样本，$(x_1, y_1), (x_2, y_2), \ldots, (x_N, yN)$ $x_i \in R^n, y_i \in \{-1, +1\}$，经验风险定义如下：

$$R_{emp}(w) = \sum_i^N L(y_i, f(x_i, w))$$

SVM的理论基础

- 传统的统计模式识别方法只有在样本趋向无穷大时，其性能才有理论的保证。统计学习理论（SLT）研究有限样本情况下的机器学习问题。
- 统计学习理论是SVM的理论基础
- 传统的统计模式识别方法在进行机器学习时，强调经验风险最小化。而单纯的经验风险最小化会产生“过学习问题”，其推广能力较差。
- **推广能力**：模型(即预测函数,或称学习函数、学习模型)对未来输出进行正确预测的能力。
- **过学习问题**：模型训练精度高，而测试精度低、推广能力差的现象。

SVM的理论基础

- SVM如何克服过拟合问题

什么叫小样本？

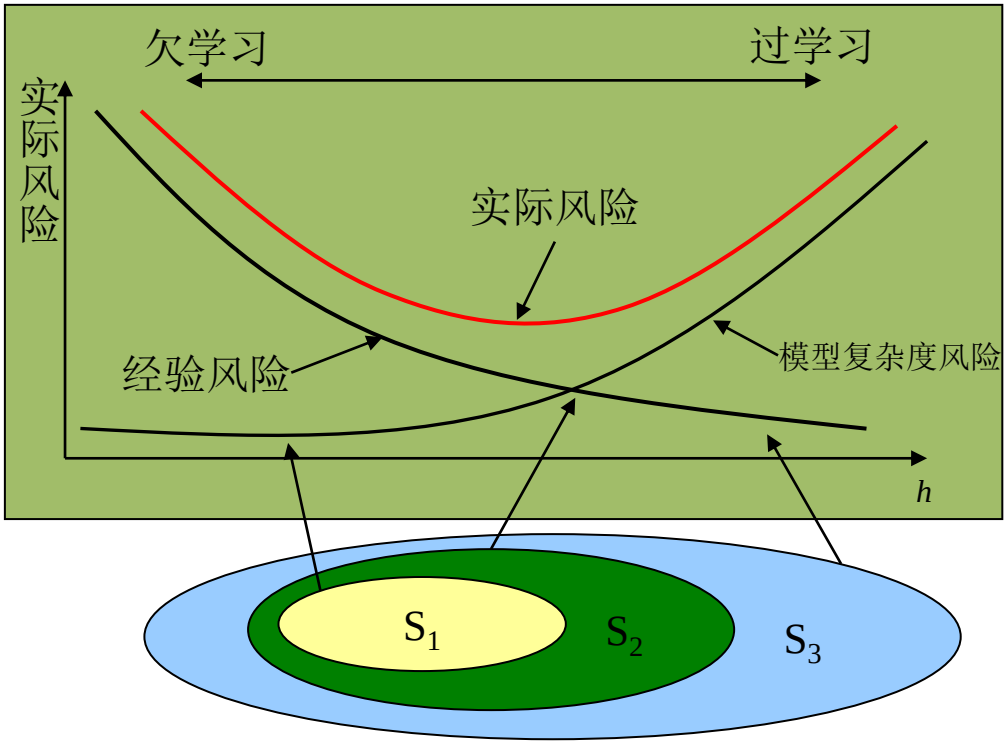
  - 统计学习理论：针对小样本问题而提出的学习理论
  - 期望风险：模型在统计意义上的学习误差，$R(w)$
  - 经验风险：由观测样本计算的学习误差，$R_{emp}(w)$
  - VC维(模型复杂度)：模型复杂程度的衡量，记为h
  - 模型复杂度风险：与VC维和学习样本数相关的风险，组成期望风险上界的一部分
  - 结构风险：在一定的VC维下，由经验风险和模型复杂度风险组成的期望风险的上界，称为结构风险

$$R(w) \le R_{emp}(w) + \sqrt{\frac{h(\ln(2h/n)+1) - \ln(\eta/4)}{n}} \equiv R_{emp}(w) + \Phi(h/n)$$

期望风险　经验风险　结构风险

SVM的理论基础

- 结构风险最小化示意图



函数子集：$S_1 \subset S_2 \subset S_3$

VC维：$h_1 \leq h_2 \leq h_3$

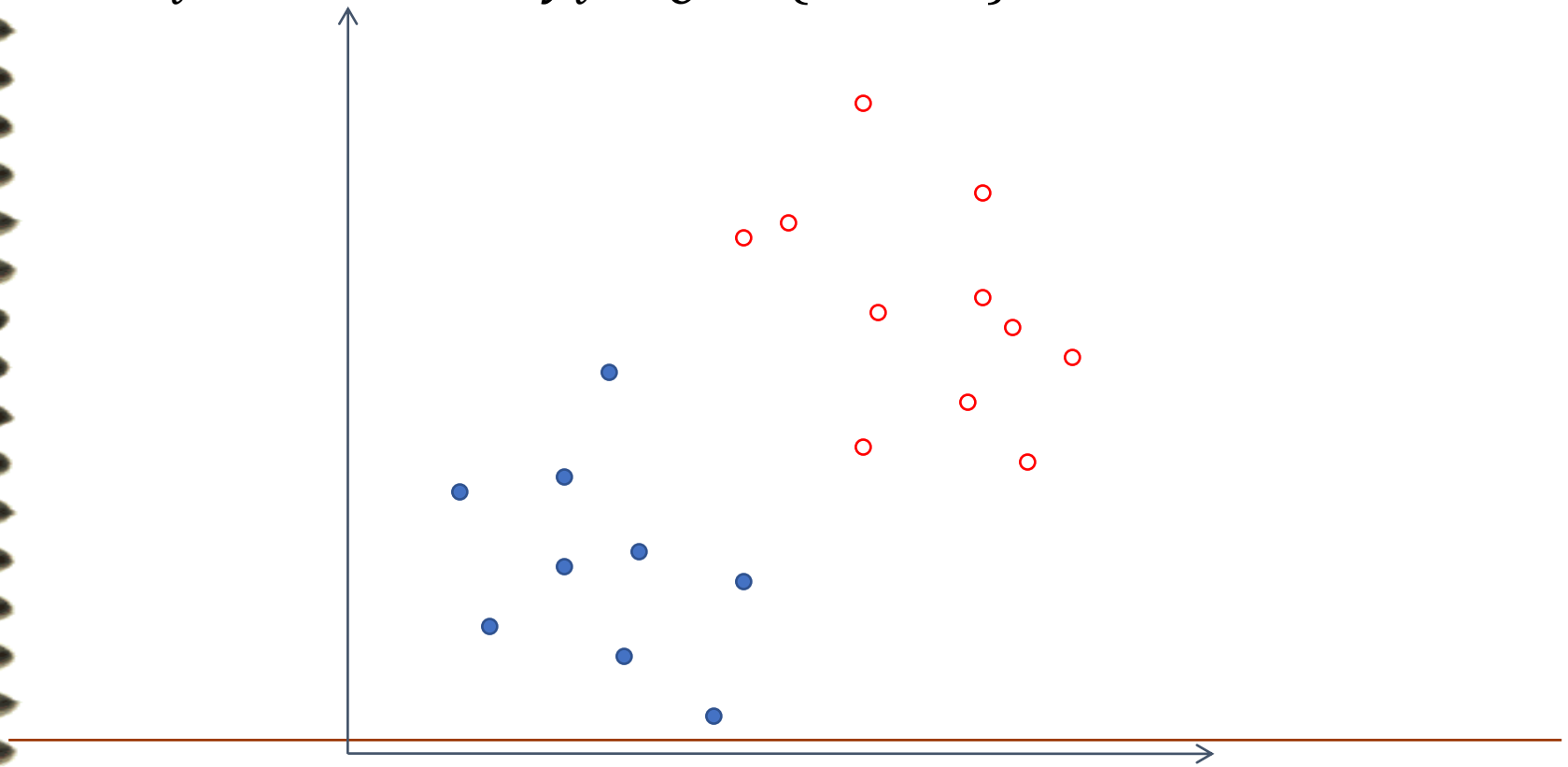
线性可分支持向量机

线性可分支持向量机

- 假设特征空间上的训练数据集：

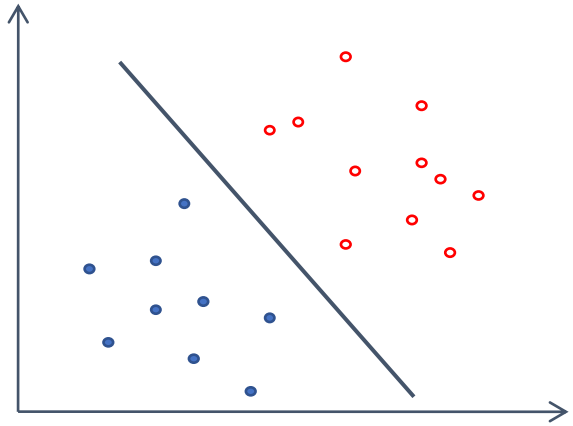
$$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

$$x_i \in \mathcal{X} = R^n, \quad y_i \in \mathcal{Y} = \{+1, -1\}, \quad i = 1,2, \cdots, N$$



线性可分支持向量机

- 线性可分：在二维空间上，如果两类点可以被一条直线（高维空间叫超平面）完全分开叫做线性可分。



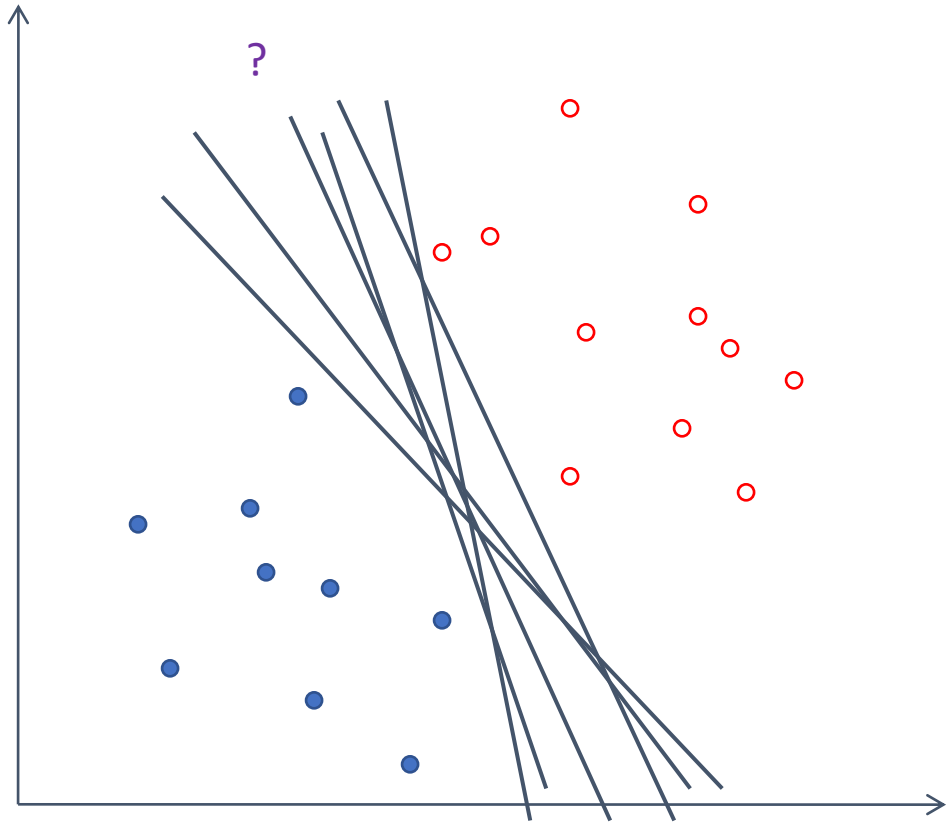
- 严格的数学定义如下：

设D_0 和 D_1 是 n 维欧氏空间中的两个点集，如果存在 n 维向量 w 和实数 b，使得：

1. 所有属于 D_0 的点 x_i 都有 $wx_i + b > 0$
2. 而对于所有属于 D_1 的点 x_j 则有 $wx_j + b < 0$，则我们称 D_0 和 D_1 **线性可分**
3. 从二维扩展到多维空间中时，将 D_0 和 D_1 完全正确地划分开的 $wx + b = 0$ 就成了一个超平面。

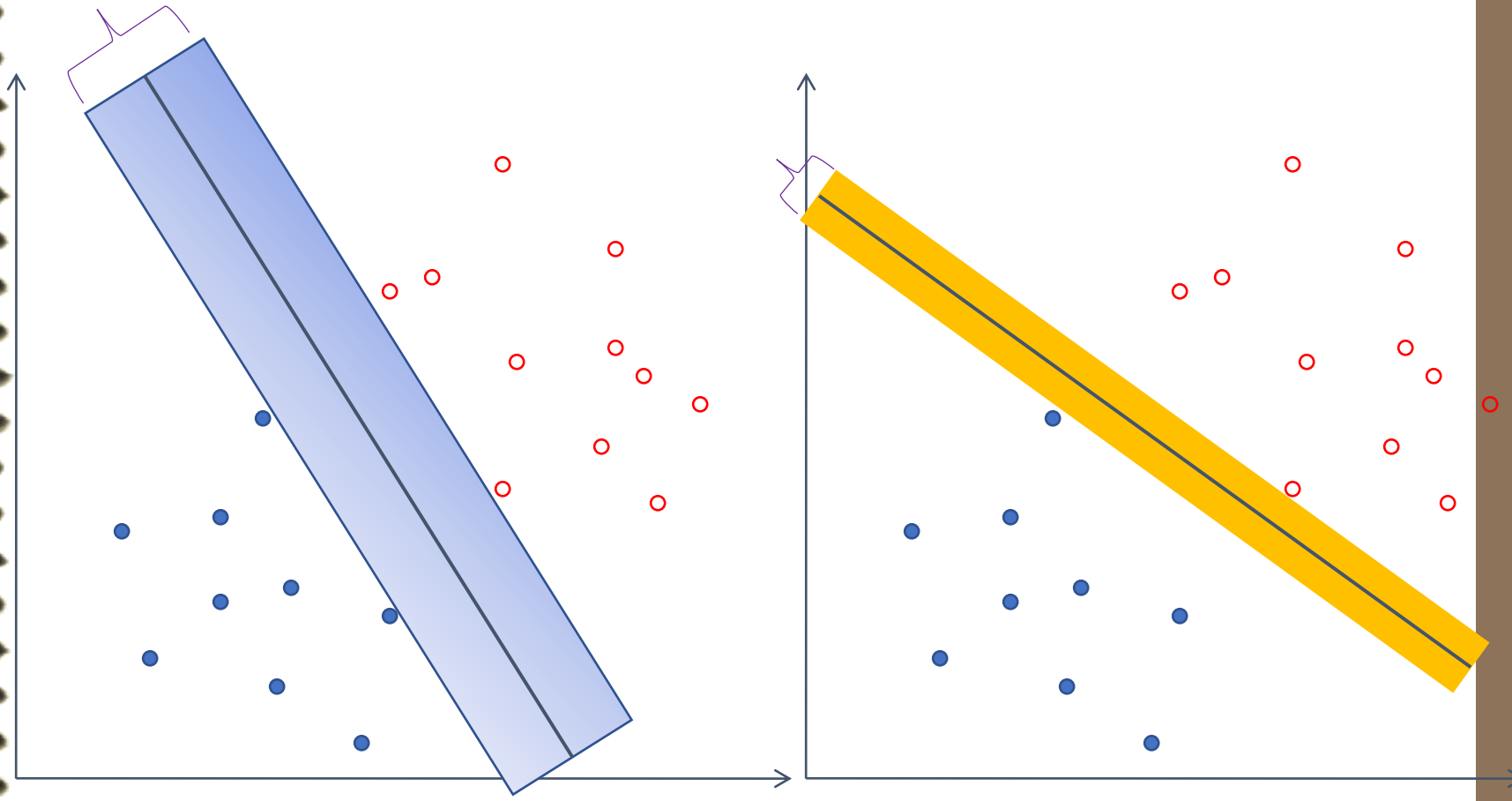
线性可分支持向量机

- 如何选择超平面?



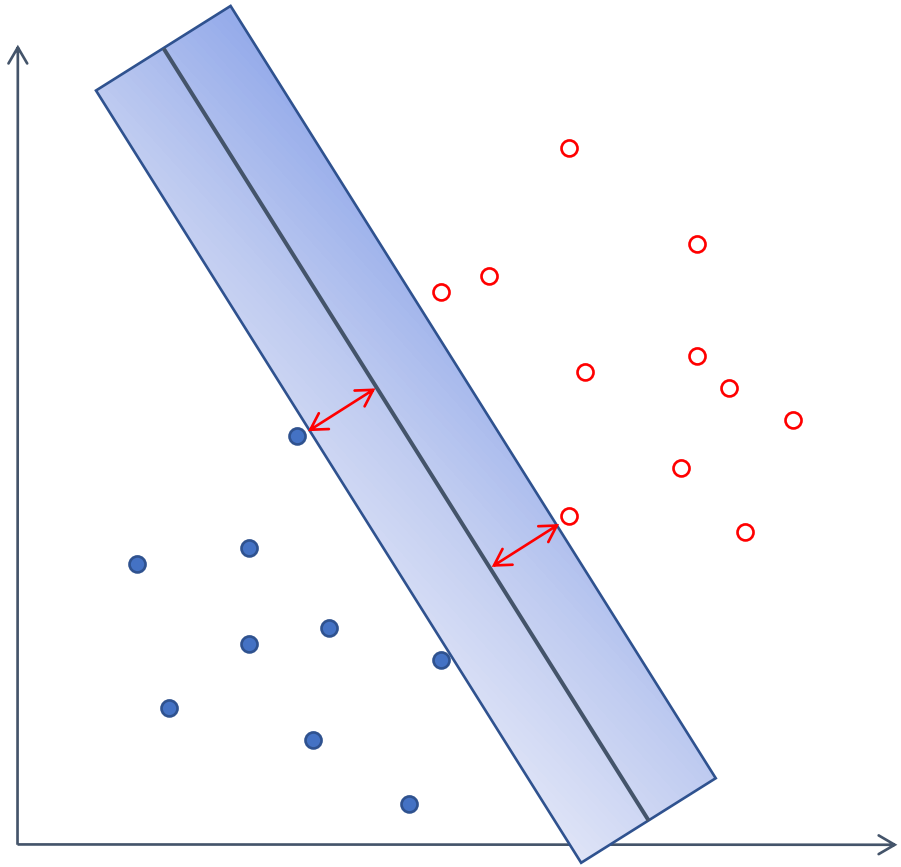
线性可分支持向量机

- Margins（边界）



线性可分支持向量机

- 如何找到最大Margins对应的平面?
  最大化离分类平面最近的点的距离



线性可分支持向量机

- 任意点x_i到平面$w \cdot x + b = 0$的距离为：

$$\mathrm{d_i} = \frac{|w \cdot x_i + b|}{\|w\|}$$

- 由于数据线性可分，平面$w \cdot x + b = 0$对于任意点(x_i, y_i)都满足： $y_i(w \cdot x_i + b) > 0$

- 因此，

$$\mathrm{d_i} = \frac{y_i(w \cdot x_i + b)}{\|w\|}$$

线性可分支持向量机

- 定义$\hat{\gamma} = \min\limits_{1 \le i \le N} y_i(w \cdot x_i + b)$，则：

$$y_i(w \cdot x_i + b) \ge \hat{\gamma} > 0$$

- 对任给的$k \neq 0$ 来说，(w, b)和(kw, kb) 所表示的是同一超平面，因此可选择合适的(w, b) (上式同除$\hat{\gamma}$即可），使得：

$$y_i(w \cdot x_i + b) \ge 1$$

- 离平面最近的点到该平面的距离：

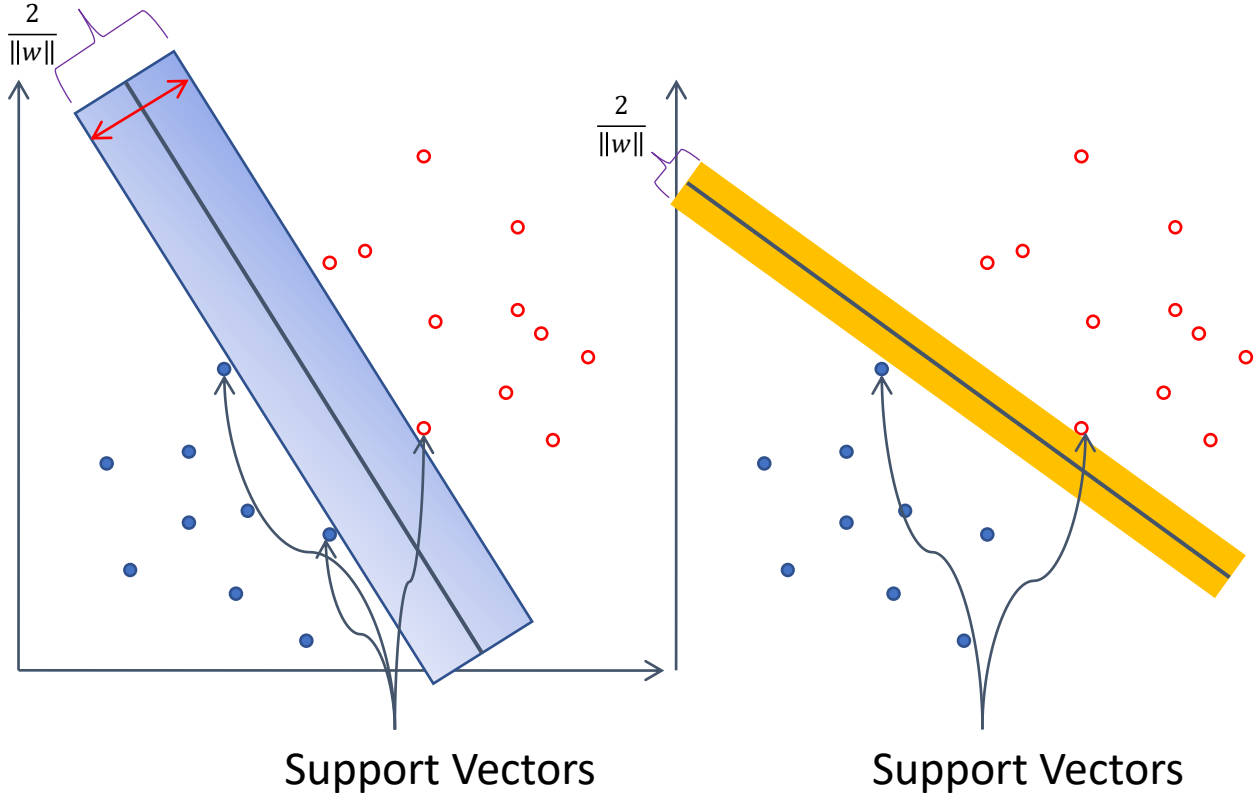
$$d^* = \min_{1 \le i \le N} d_i = \min_{1 \le i \le N} \frac{y_i(w \cdot x_i + b)}{\|w\|} = \frac{1}{\|w\|}$$

线性可分支持向量机

- 到平面距离最小的样本点，满足：

$$y_i(w \cdot x_i + b) = 1$$

称为支持向量（Support Vectors）



线性可分支持向量机

- 支持向量机求的最优分离超平面，不仅要分类正确，而且要使得间隔最大化，这里称之为硬间隔最大化，即：

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

- 最大化$\frac{1}{\|w\|}$和最小化$\frac{1}{2}\|w\|^2$是等价的，因此上式可写为如下的凸二次规划问题：

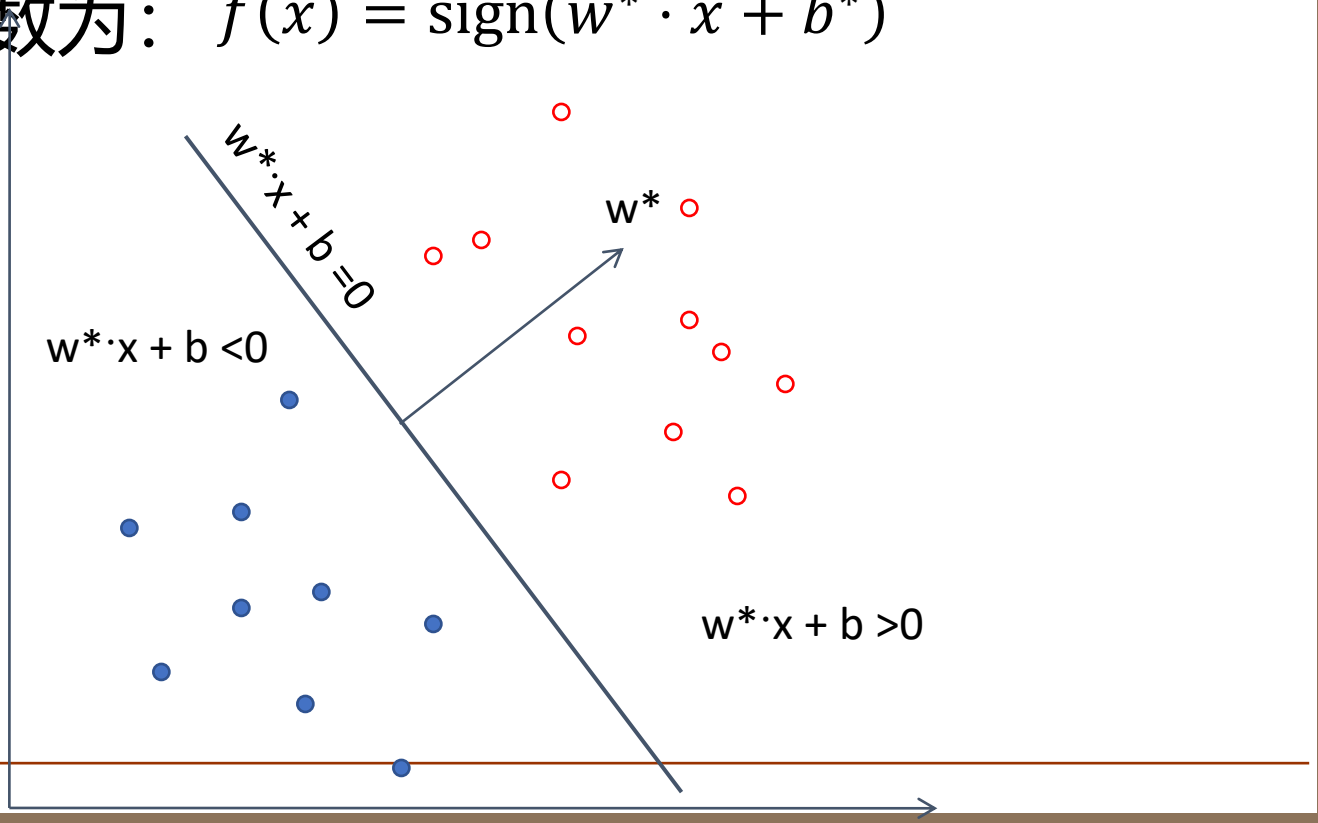
$$\min_{w,b} \frac{1}{2} \| w \|^2,$$
$$s.t. \quad y_i\left(w \cdot x_i + b\right) \geq 1, i = 1,2,\cdots,N$$

线性可分支持向量机

- 通过间隔最大化或等价方法求解相应的凸二次规划问题学习得到的分离超平面为：

$$w^* \cdot x + b^* = 0$$

- 决策函数为：$f(x) = \text{sign}(w^* \cdot x + b^*)$



支持向量机的优化算法

支持向量机的优化算法

- 若数据是线性可分的，则上述凸二次规划问题的解存在且唯一。二次规划问题有很多算法可以求解，也有很多求解器能直接解决二次规划问题。但是一般我们不直接求解原问题，而是求解其对偶问题，原因有几点：

1. 对偶问题更简单，更容易求解，效率更高；
2. 对偶问题将原始问题中的不等式约束转为了对偶问题中的等式约束；
3. 对偶问题方便核函数的引入，进而可以推广到非线性分类问题。

支持向量机的优化算法

- 线性可分支持向量机的优化目标如下：

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

$$s.t. \quad 1 - y_i \left(w \cdot x_i + b \right) \leq 0, i = 1,2, \cdots, N$$

- 求解线性可分支持向量机的步骤为：

1. 利用拉格朗日乘子法，构造拉格朗日函数；
2. 利用强对偶性(KKT条件)将优化问题进行转化，并求解；
3. 利用最优的w^*和b^*构建分类器。

支持向量机的优化算法

1. 利用拉格朗日乘子法，构造拉格朗日函数

$$L(w,b,\alpha)=\frac{1}{2}||w||^2+\sum_{i=1}^{N}\alpha_i[1-y_i(w\cdot x_i+b)]$$

$$s.t.\ \alpha_i\geq 0$$

优化目标变为：$\min_{w,b}\max_{\alpha}L(w,b,\alpha)$

2. 利用强对偶性将优化问题进行转化，并求解

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

为了得到对偶问题的解，需要先$L(w,b,\alpha)$对w,b的极小，再求对α的极大。

支持向量机的优化算法

2.1 求解$\min_{w,b} L(w,b,\alpha)$

将拉格朗日函数 $L(w,b,\alpha)$分别对 w,b求偏导数并令其等于 0:

$$\nabla_w L(w,b,\alpha) = w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0$$

$$\nabla_b L(w,b,\alpha) = -\sum_{i=1}^{N} \alpha_i y_i = 0$$

可得: $w = \sum_{i=1}^{N} \alpha_i y_i x_i$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$

支持向量机的优化算法

2.1 求解$\min_{w,b} L(w,b,\alpha)$

将$w=\sum_{i=1}^{N}\alpha_i y_i x_i$带入拉格朗日函数:

$$\min_{w,b} L(w,b,\alpha)=\frac{1}{2}||w||^2+\sum_{i=1}^{n}\alpha_i[1-y_i(w\cdot x_i+b)] \quad (\| w \|^2= w^T\cdot w)$$

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

由$\sum_{i=1}^{N}\alpha_i y_i=0$可得:

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

支持向量机的优化算法

2.2 求解$\max\limits_{\alpha}\min\limits_{w,b}L(w,b,\alpha)$得到最优的$\alpha^*$

将所得到的$\min\limits_{w,b}L(w,b,\alpha)$带入可得:

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

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0,\ \alpha_i \geqslant 0, i = 1,2,\cdots,N$$

将目标函数由求极大转换成求极小，可得:

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

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0,\ \alpha_i \geqslant 0, i = 1,2,\cdots,N$$

支持向量机的优化算法

2.2 求解最优的α^*的分解算法

- 求解如下问题：

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

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0,\ \alpha_i \geqslant 0, i = 1,2,\cdots,N$$

$$D = \{ d_{ij} | y_i y_j (x_i \cdot x_j) \}$$

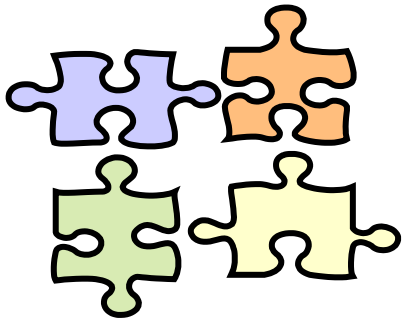
D的存储代价=0.5 × (训练样本数)^2 ×单元存储空间

$$0.5 \times (7000)^2 \times 8 = 196 \text{ MByte}$$

支持向量机的优化算法

2.2 求解最优的α^*的分解算法

- Edgar Osuna(Cambridge ,MA)等人在IEEE NNSP'97发表了An Improved Training Algorithm for Support Vector Machines，提出了SVM的分解算法



支持向量机的优化算法

2.2 求解最优的α^*的分解算法

- 将α向量分成两个集合,工作集α_B，固定集α_N。即

$$\alpha = \{\alpha_B, \alpha_N\}$$

- 每次对α_B解决一个小的二次规划问题，保持α_N中的值不变
- 每次迭代选择不同的α_B和α_N，每解决一个小规模优化问题，都在原来的基础上向最终的解集前进一步。
- 每次迭代检查当前结果，满足优化条件，则找到了优化问题的解，算法结束。

支持向量机的优化算法

2.2 求解最优的α^*的分解算法

- Original QP:

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

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0,\ \alpha_i \geqslant 0, i = 1,2,\cdots,N$$

- Small QP:

$$min \quad \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{1}$$

$$\mathbf{0} \leq \boldsymbol{\alpha},\ \boldsymbol{\alpha}^T \boldsymbol{y} = 0$$

$$\boldsymbol{\alpha} = \begin{Bmatrix} \boldsymbol{\alpha}_B \\ \boldsymbol{\alpha}_N \end{Bmatrix}$$

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{BB} & \mathbf{D}_{BN} \\ \mathbf{D}_{NB} & \mathbf{D}_{NN} \end{pmatrix}$$

支持向量机的优化

2.2 求解最优的α^*的分解算法

- SVM^{light}

Thorsten Joachims

(UniversityDortmund ,Informatik, AI-Unit)

http://svmlight.joachims.org/

http://www.programsalon.com/

参考文献：Make Large-Scale SVM Learning Practical

- SMO

John C. Platt

(Microsoft Research)

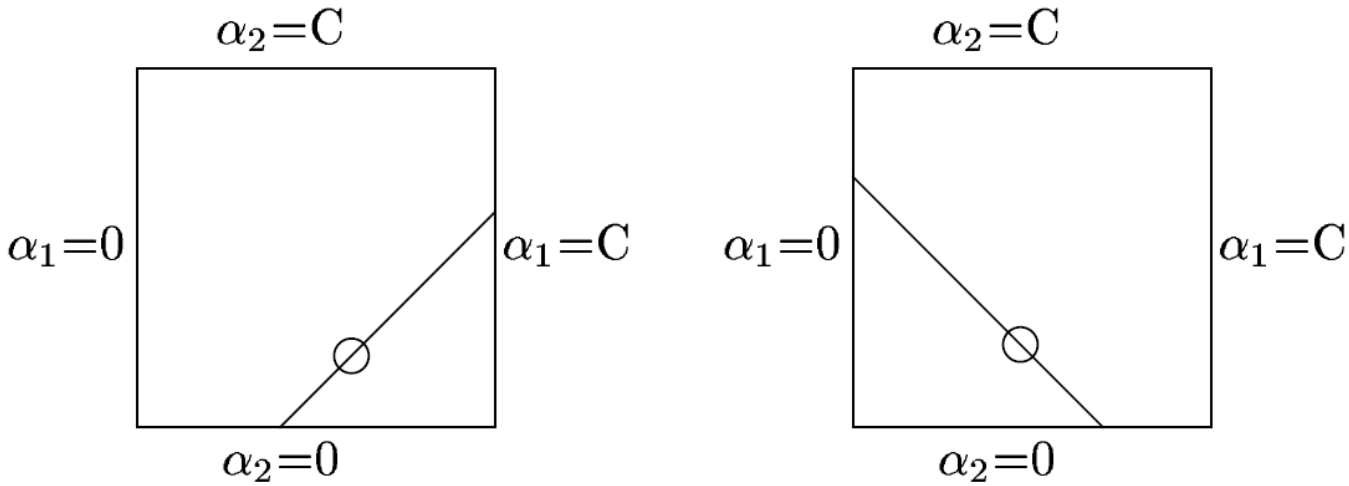
http://theoval.sys.uea.ac.uk/svm/toolbox/

参考文献：Fast Training of Support Vector Machines using Sequential Minimal Optimization

支持向量机的优化算法

2.2 求解最优的α^*的分解算法

- 序列最小优化算法(SMO)
  - 每次只优化2个参数



  - 由约束条件$\sum \alpha_i y_i = 0$，可知

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

- 参考文献：Platt 等: Sequential Minimal Optimization for SVM. In: svm-smo source code package.

支持向量机的优化

2.2 SMO求解$\max_{\alpha} \min_{w,b} L(w,b,\alpha)$得到最优的$\alpha*$

1. 选择两个需要更新的参数 α_i 和 α_j，固定其他参数，可以得到以下约束：

$$\alpha_i y_i + \alpha_j y_j = c \quad \alpha_i \geq 0, \alpha_j \geq 0$$

其中$c = -\sum_{k \neq i,j} \alpha_k y_k$，由此可以得出$\alpha_j = \frac{c - \alpha_i y_i}{y_j}$，因此目标问题转化成了仅有一个约束条件$\alpha_i \geq 0$的最优化问题。

2.对于仅有一个约束条件的最优化问题，可以在 α_i 上对优化目标求偏导，令导数为零，从而求出变量值 α_{i_new}，然后根据 α_{i_new} 求出 α_{j_new}。

3. 多次迭代直至收敛，可以求出所有样本对应的最优α^*。

支持向量机的优化

3. 利用 SMO 算法求解得到α^*，利用最优的w^*和b^*构建分类器

由2.1可得最优的$w^* = \sum_{i=1}^{N} \alpha_i y_i x_i$

由于所有$\alpha_i^* > 0$对应的点都是支持向量，可以找任意支持向量(x_s, y_s)，带入$1 - y_i(w \cdot x_i + b) = 0$，得到$y_s(w \cdot x_s + b) = 1$。等式两边同时乘以$y_s$，可得$y_s^2(w \cdot x_s + b) = y_s$。由于$y_s^2 = 1$，得到$b^* = y_s - w \cdot x_s$。

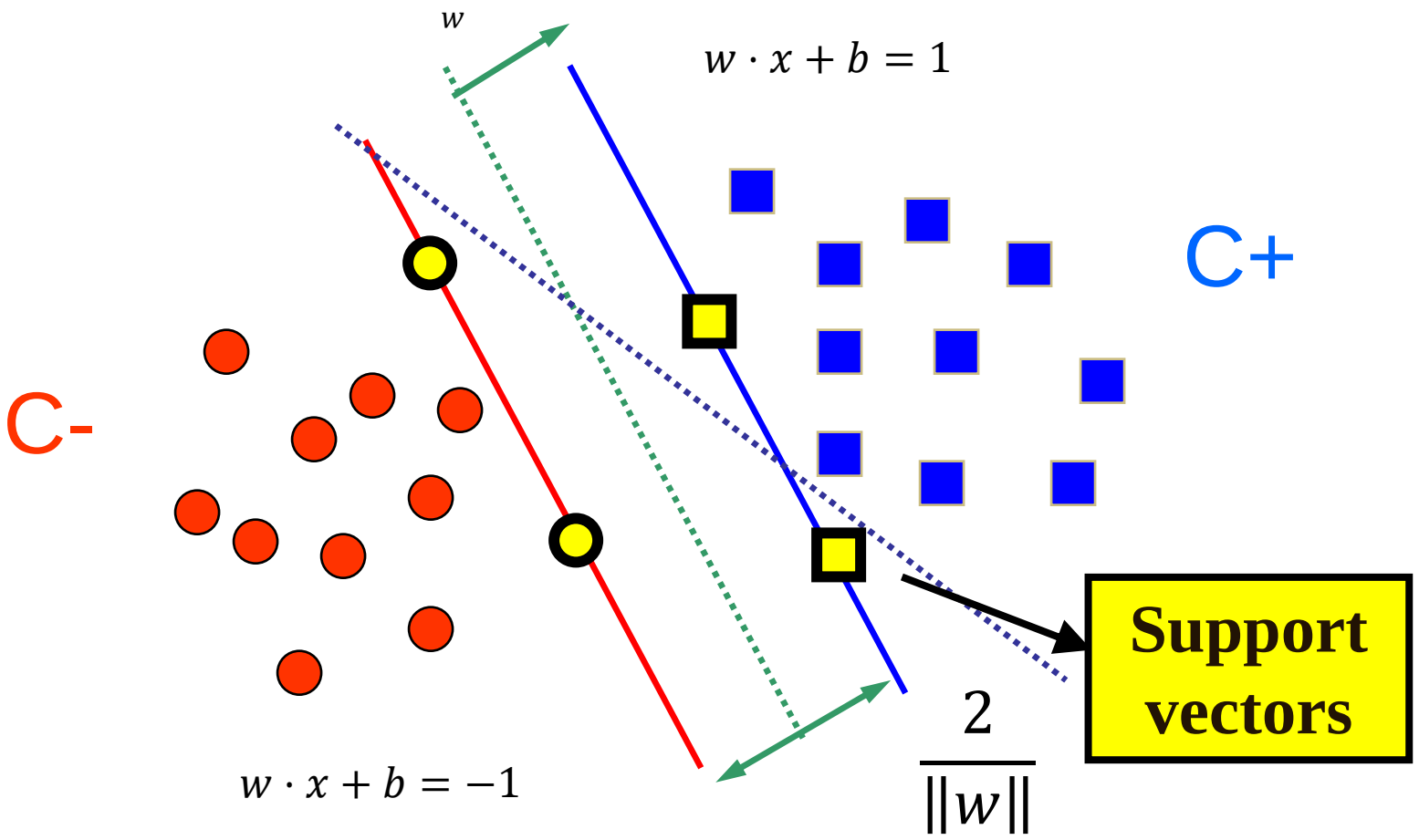
为了使结果具有更好的鲁棒性，可以使用支持向量点的均值得到：

$$b^* = \frac{1}{|S|} \sum_{s \in S} (y_s - w x_s)$$

线性支持向量机—软间隔最大化

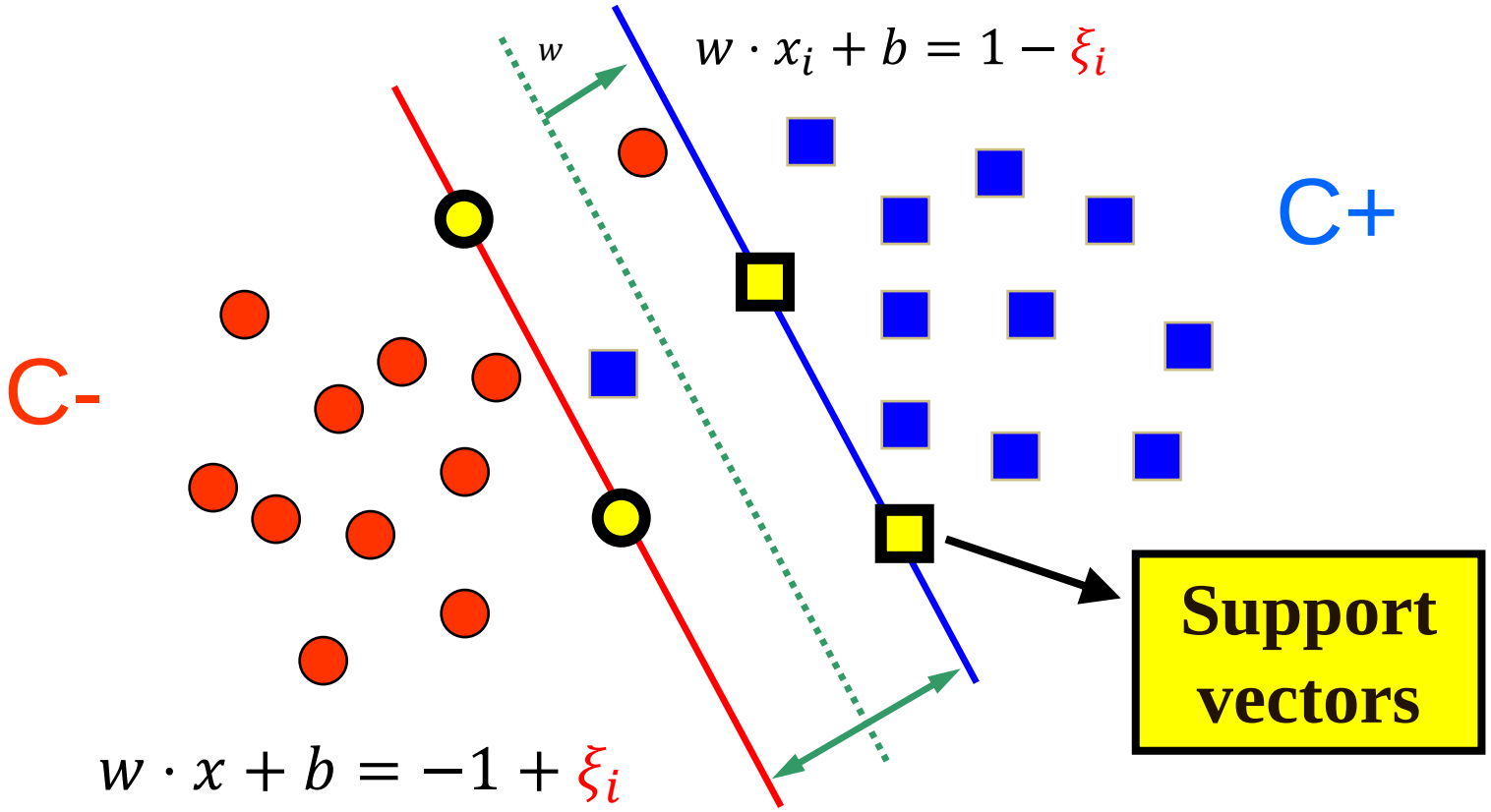
线性支持向量机—软间隔最大化

- 线性可分支持向量机



线性支持向量机—软间隔最大化

- 近似线性可分数据



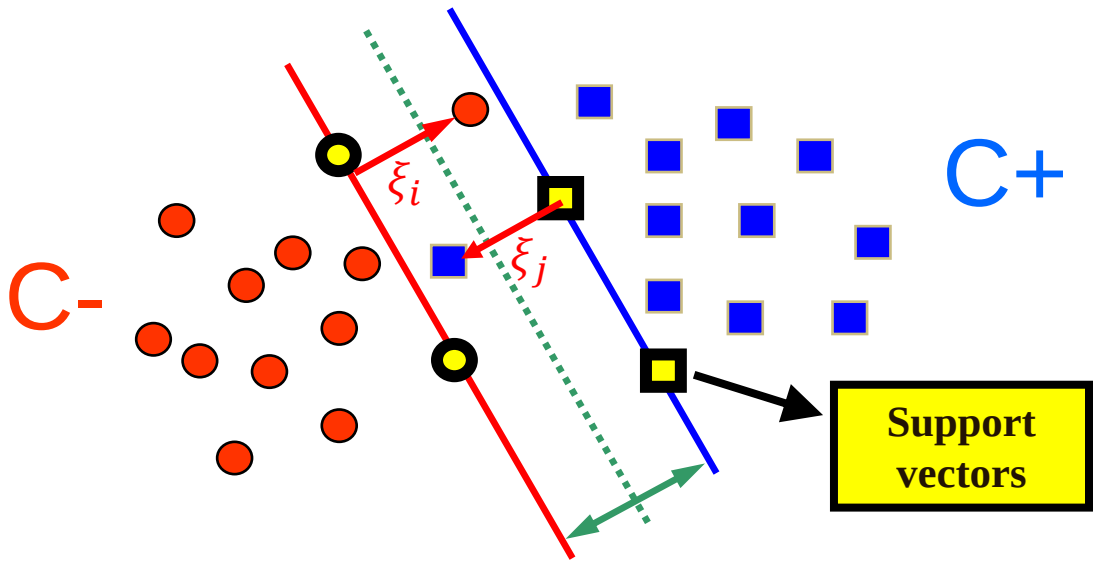
线性支持向量机—软间隔最大化

- 软间隔允许部分样本点不满足约束条件：

$$1 - y_i(w \cdot x_i + b) \leq 0$$

- 为了度量这个间隔的程度，为每个样本引入一个松弛变量ξ_i（$\xi_i \geq 0$），约束条件变为：

$$1 - y_i(w \cdot x_i + b) - \xi_i \leq 0$$



线性支持向量机—软间隔最大化

- 增加软间隔后，优化目标变成：

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i$$

$$s.t.\ \ 1 - y_i(w \cdot x_i + b) - \xi_i \leq 0, \xi_i \geq 0, i = 1,2,\dots,N$$

其中$C > 0$称为惩罚参数，由实际问题决定。

- 优化目标使$\frac{1}{2}||w||^2$尽量小即间隔尽量大，同时使误分类点的个数尽量少。
- 相对于硬间隔最大化，它称为软间隔最大化。线性不可分的线性支持向量机的学习问题同线性可分支持向量机相同。

线性支持向量机—软间隔最大化

1. 利用拉格朗日乘子法，构造拉格朗日函数：

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

$$= \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i[1-\xi_i - y_i(w\cdot x_i + b)] - \sum_{i=1}^{N}\beta_i\xi_i$$

$$s.t.\quad \alpha_i \geq 0 \quad \beta_i \geq 0$$

其中α_i和β_i是拉格朗日乘子，w, b和ξ_i是主问题参数。根据强对偶性，转换为对偶问题：

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

线性支持向量机—软间隔最大化

2.1 分别对主问题参数w, b和ξ_i求偏导数，并令偏导数为 0，得出如下关系：

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

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

$$C = \alpha_i + \beta_i$$

将这些关系带入拉格朗日函数中，得到：

$$\min_{w,b,\xi} L(w, b, \xi, \alpha, \beta) = \sum_{j=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

线性支持向量机—软间隔最大化

2.2 带入到$\max\limits_{\alpha,\beta}\min\limits_{w,b,\xi} L(w,b,\xi,\alpha,\beta)$中可得：

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

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

$$C-\alpha_i-\beta_i=0$$

$$\alpha_i\geqslant 0$$

$$\beta_i\geqslant 0, i=1,2,\cdots,N$$

利用$\alpha_i+\beta_i=C$消去β_i，得到如下对偶问题：

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

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

$$0\leq\alpha_i\leq C, i=1,2,\cdots,N.$$

和硬间隔的一样，只是多了个约束条件

线性支持向量机—软间隔最大化

3. 利用 SMO 算法求解得到α^*，利用最优的w^*和b^*构建分类器

由2.1可得最优的$w^* = \sum_{i=1}^{N} \alpha_i y_i x_i$

由于所有$\alpha_i^* > 0$ 对应的点都是支持向量，可以找任意支持向量(x_s, y_s)，带入$1 - y_i(w \cdot x_i + b) = 0$，得到$y_s(w \cdot x_s + b) = 1$。等式两边同时乘以$y_s$，可得$y_s^2(w \cdot x_s + b) = y_s$。由于$y_s^2 = 1$，得到$b^* = y_s - w \cdot x_s$。

为了使结果具有更好的鲁棒性，可以使用支持向量点的均值得到：

$$b^* = \frac{1}{|S|} \sum_{s \in S} (y_s - w x_s)$$

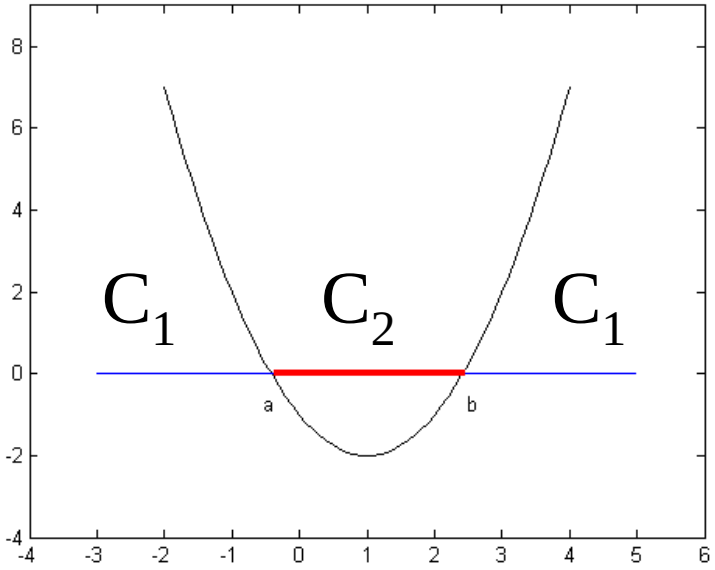
非线性支持向量机——核方法

非线性支持向量机——核方法

- 线性分类器的局限：在二维空间中，没有任何一个线性函数能解决下述划分问题（黑红各代表一类数据）。

非线性支持向量机——核方法

- 线性分类器的局限（举例说明）
  - 如果建立一个二次判别函数$g(x)=(x-a)(x-b)$（**非线性分类器**），则可以很好的解决上述分类问题。
  - 决策规则仍是：如果$g(x)>0$，则判定x属于C_1，如果$g(x)<0$，则判定x属于C_2，如果$g(x)=0$，则可以将x任意分到某一类或者拒绝判定。



非线性支持向量机——核方法

- 二次判别函数$g(x)=(x-a)(x-b)$可以写成如下的一般形式：

$$g(x)=c_0+c_1x+c_2x^2$$

- 如果选择 $x\to z$的映射：

$$X\mapsto R^m:z=\varphi(x)\ (x\in R^n,z\in R^m,n<<m)$$

- 则可以把二次函数化成关于y的线性函数

$$g(x)=a^T\varphi(x)=\sum_{i=1}^{3}a_iz_i$$

- 其中

$$y=\begin{bmatrix}z_1\\z_2\\z_3\end{bmatrix}=\begin{bmatrix}1\\x\\x^2\end{bmatrix},a=\begin{bmatrix}a_1\\a_2\\a_3\end{bmatrix}=\begin{bmatrix}c_0\\c_1\\c_2\end{bmatrix}$$

因此，$g(x)$也被称为广义线性分类器

非线性支持向量机——核方法

- 世界真奇妙
  - 一般地，对于任意高次的非线性分类器$g(x)$都可以通过适当的线性变换，使之转化为广义线性分类器。
  - $a^T z$不是x的线性函数，但它是z的线性函数。
  - $a^T z = 0$在Z空间确定了一个通过原点的超平面。
  - $g(x)$也可以看作任意分类器的判别式在原点的展开形式。

非线性支持向量机——核方法

- 终点又回到了起点
  - 因此，理论上可以利用线性分类器简单的特性来解决任意复杂的问题。

?

非线性支持向量机——核方法

- 使用广义线性分类器的困难
  - 一般来讲，确定$x -> z$的映射函数$\varphi(x)$困难
  - 转换之后，存在维数灾难问题。

- SVM巧妙地克服了这两个困难
  - 特征映射函数

$$\varphi(x): X \to R^m \ (m \gg n)$$

  - 核函数

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

非线性支持向量机——核方法

- 目标：找到一个曲面，使得它能够尽可能地将两类数据点正确的分开，并且使两类数据点边界曲面之间的距离最大。
- 解决方法：构造一个有约束条件的优化问题。具体的说是一个具有线性不等式约束的二次规划问题(Quadratic Programing，QP)。
- 通过求解该QP问题，得到最优的分类曲面，也称分类器。

非线性支持向量机——核方法

- SVM问题的数学表示
  - 已知两类模式识别问题的一组数据：
    $$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$
  - 目标：确定一个广义最优分类超平面
    $$w \cdot \varphi(x) + b = 0,$$

  使得
  $$y = f(x, w) = \mathrm{sgn}(g(x)) = \mathrm{sgn}(w \cdot \varphi(x) + b)$$
- 满足条件：
  - 经验风险($R_{emp}(f)$最小（错分最少）√
  - 推广能力最大（间隔最大2/||w||^2）

非线性支持向量机——核方法

- 非线性支持向量机的数学表示
  - 分类曲面方程满足条件

对于任意观测样本(x_i, y_i)，最优分类曲面

$$g(x) = w \cdot \varphi(x) + b$$

应满足

$$g(x_i) = \begin{cases} w \cdot \varphi(x_i) + b \geq 1 & y_i = +1 \\ w \cdot \varphi(x_i) + b \leq -1 & y_i = -1 \end{cases}$$

即

$$y_i(w \cdot \varphi(x_i) + b) \geq 1 \quad (i = 1,2 \cdots, N)$$

非线性支持向量机——核方法

- 非线性支持向量机的数学表示
  - SVM对应的优化问题
    - ✓已知：N个观测样本，$(x_1, y_1), (x_2, y_2) \ldots\ldots (x_N, yN)$
    - ✓求解：

$$\begin{cases} \min \dfrac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i \\ s.t.\ y_i(w \cdot \varphi(x_i) + b) \geq 1 - \xi_i, \\ \xi_i \geq 0\ (i = 1,2,\ldots,l) \end{cases} \quad \text{(P)}$$

  - 目标：确定最优分类曲面

$$g(x) = w \cdot \varphi(x) + b\ = 0$$

非线性支持向量机——核方法

利用拉格朗日乘子法，构造拉格朗日函数：

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

$$= \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i[1-\xi_i - y_i(w\cdot\varphi(x_i)+b)] - \sum_{i=1}^{N}\beta_i\xi_i$$

$$s.t.\quad \alpha_i \geq 0 \quad \beta_i \geq 0$$

其中α_i和β_i是拉格朗日乘子，w, b和ξ_i是主问题参数。根据强对偶性，转换为对偶问题：

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

非线性支持向量机——核方法

最优性条件($\min_{w,b,\xi}\max_{\alpha,\beta} L(w,b,\xi,\alpha,\beta)=\max_{\alpha,\beta}\min_{w,b,\xi} L(w,b,\xi,\alpha,\beta)$)

Karush-Kuhn-Tucker条件(KKT Condition)

$$\frac{\partial L}{\partial w}=\nabla|_w=w-\sum_i \alpha_i y_i \varphi(x_i)=0$$

$$\frac{\partial L}{\partial \xi}=\nabla|_\xi=C-\alpha-\beta=0$$ （原始问题可行性）

$$\frac{\partial L}{\partial b}=\sum_i \alpha_i y_i=0$$

$\alpha_i \geq 0, \beta_i \geq 0$ （对偶问题可行性）

$\alpha_i[1-\xi_i-y_i(w\cdot\varphi(x_i)+b)]=0$

$\beta_i\xi_i=0$ （松弛互补条件）

非线性支持向量机——核方法

分别对主问题参数w, b和ξ_i求偏导数，并令偏导数为0，得出如下关系：

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

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

$$C = \alpha_i + \beta_i$$

非线性支持向量机——核方法

- 将上述条件代入L中

$$L = \frac{1}{2}||\boldsymbol{w}||^2 + \sum_i \alpha_i - \sum_i \alpha_i y_i \varphi(x_i) \cdot \boldsymbol{w} + b\sum_i \alpha_i y_i + \sum_i (C - \beta_i - \alpha_i)\xi_i$$

- 得到原优化问题的对偶问题(Dual Problem)

$$\begin{cases} \max L(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \varphi(x_i) \cdot \varphi(x_j) \\ s.t.\ \ 0 \le \alpha_i \le C, \sum_i \alpha_i y_i = 0 \end{cases} \quad (D)$$

使用函数$K(x_i, x_j)$代替$\varphi(x_i) \cdot \varphi(x_j)$

非线性支持向量机——核方法

- 什么是核函数
  - 设X是输入空间(欧氏空间R^n的子集或离散集合)，又设H为特征空间(希尔伯特空间)
  - 如果存在一个从X到H的映射$\phi(x): \mathcal{X} \to \mathcal{H}$，使得对所有$x, z \in \mathcal{X}$，函数$K(x, z)$满足条件$K(x, z) = \phi(x) \cdot \phi(z)$，则称$K(x, z)$为核函数，$\phi(x)$为映射函数
  - 式中$\phi(x) \cdot \phi(z)$为$\phi(x)$和$\phi(z)$的内积

非线性支持向量机——核方法

- 常用的几种核函数
  - 线性核

$$K(x,y)=x\cdot y$$

  - Gauss径向基核

$$K(x,y)=\exp\{-\|x-y\|^2/\sigma^2\},(\sigma\in R,\sigma\neq 0)$$

  - 多项式核

$$K(x,y)=[x\cdot y+c]^q,(c\geq 0,q\in \mathrm{N}^+)$$

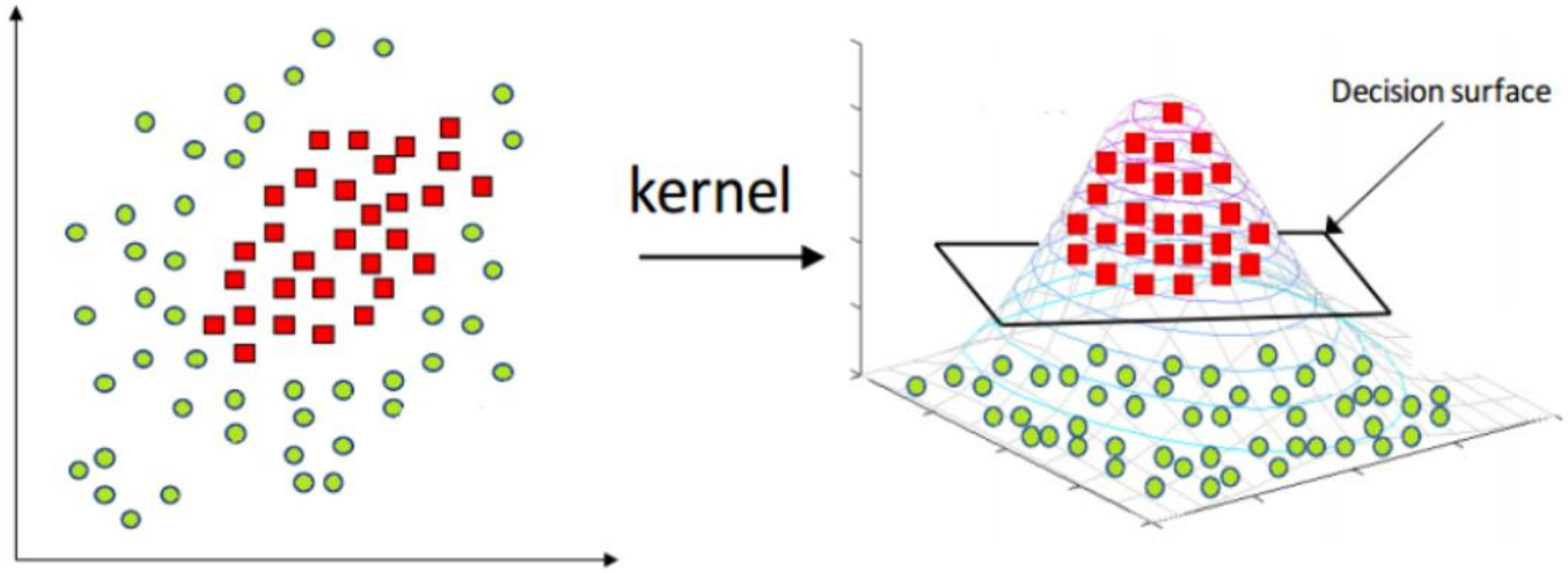
  - S形核(双曲正切核)

$$K(x_i,x_j)=\tanh(\upsilon(x_i\cdot x_j)+c),(\upsilon>0,c<0)$$

非线性支持向量机——核方法

- 核函数的效果
  - 特征映射函数

$$\varphi(x): X \to R^m \ (m \gg n)$$



非线性支持向量机——核方法

- 核函数形式下的对偶优化问题

$$\begin{cases} \max L(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ s.t.\ \ 0 \le \alpha_i \le C, \sum_i \alpha_i y_i = 0 \end{cases}$$

非线性支持向量机——核方法

SVM的最优分类超曲面

$$g(x) = w \cdot \varphi(x) + b = 0$$

$$w = \sum_i \alpha_i y_i \varphi(x_i)$$

$$g(x) = \sum_i \alpha_i y_i \varphi(x_i) \cdot \varphi(x) + b = 0$$

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

$$g(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b = 0$$

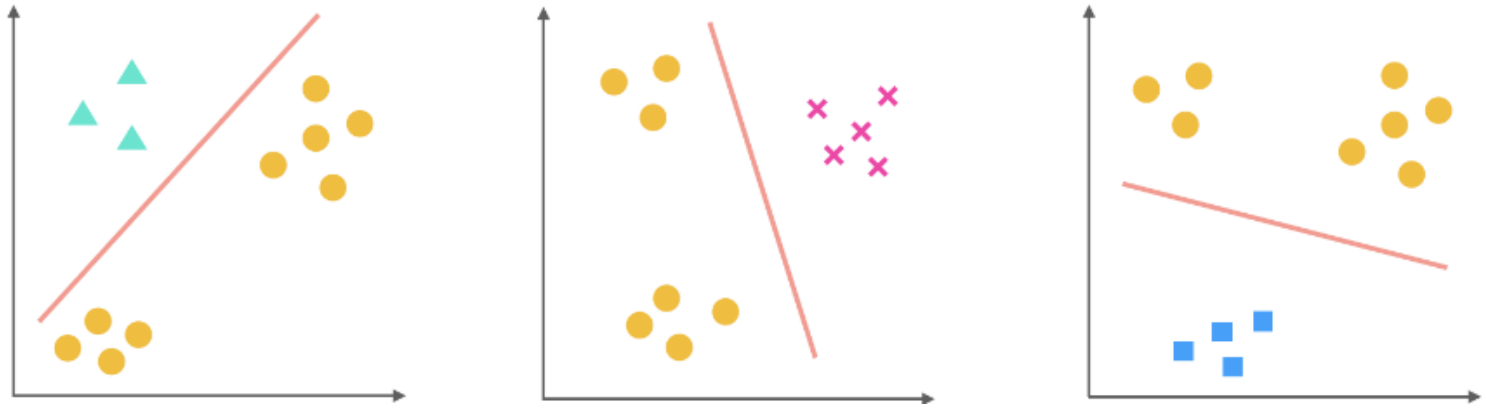
$$y = f(x, \mathbf{w}) = \mathrm{sgn}(g(x)) = \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i K(x, x_i) + b\right)$$

支持向量机分类问题

- 多类模式识别

  利用SVM解决多类模式识别问题有多种方法。例如

  - one-vs-all：可以把k类问题转化为k个两类问题，其中第i个问题是用标准SVM分类器把属于C_i类和不属于C_i类的模式区分开。

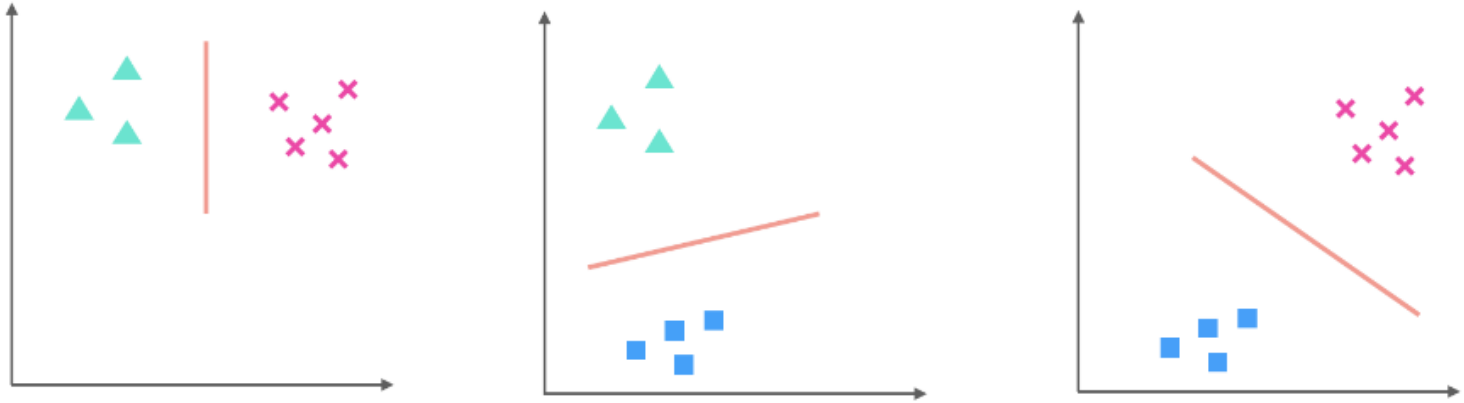


支持向量机分类问题

- 多类模式识别

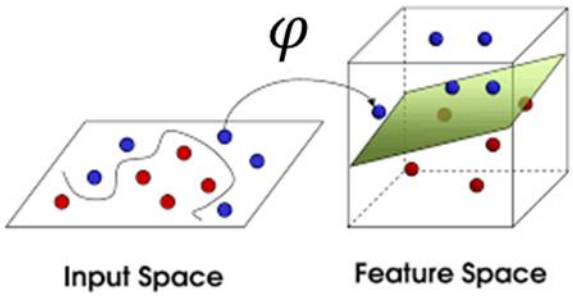
  利用SVM解决多类模式识别问题有多种方法。例如

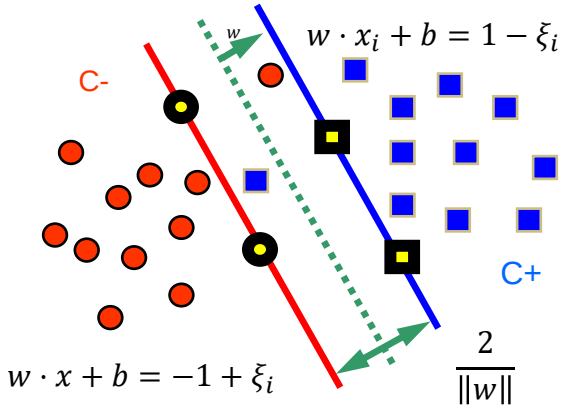
  - one-vs-one：更复杂一点的方法是训练k(k-1)/2个标准SVM分类器，采用委员会投票的方式确定待测模式所属的类别。



支持向量机分类问题

- SVM分类器小结：
  - 分类间距最大化：在理论上最大程度的保证了分类器的推广能力。
  - 特征映射函数$\varphi(x)$：把原始空间的分类问题变换到更高维的特征空间，使得在特征空间可以用线性分类器进行模式分类。
  - 核函数$K(x,y)$：把高维特征空间的内积计算转化成原始空间对称函数的计算问题，避免了特征空间的复杂计算。





支持向量机

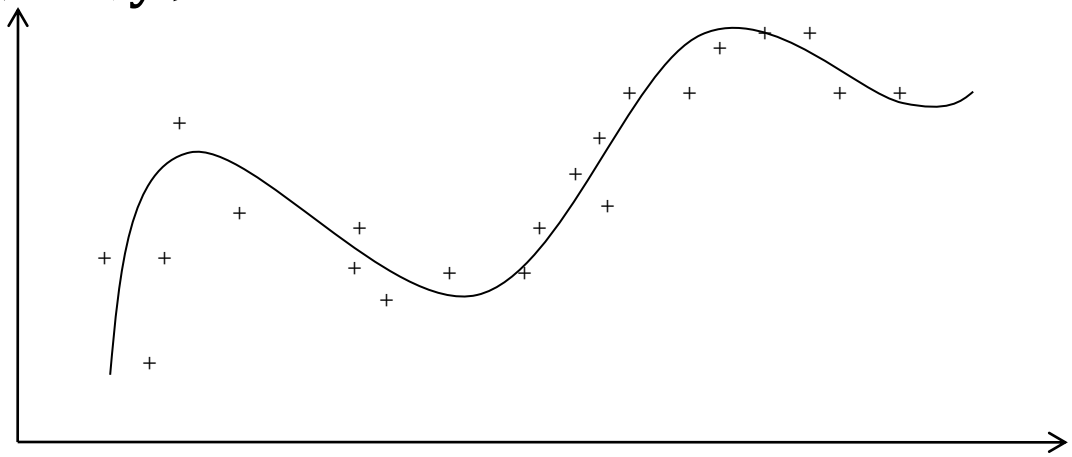
回归问题

回归问题

- 回归问题描述
  - 根据给定的训练集

$$T = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_l, y_l)\}$$

其中，$\boldsymbol{x}_i \in R^n, y_i \in R, i = 1, \cdots, l$，确定$R^n$上的一个实值函数$f(\boldsymbol{x})$，以便用$\mathrm{y} = f(\boldsymbol{x})$来推断任意模式$\boldsymbol{x}$对应的y值。



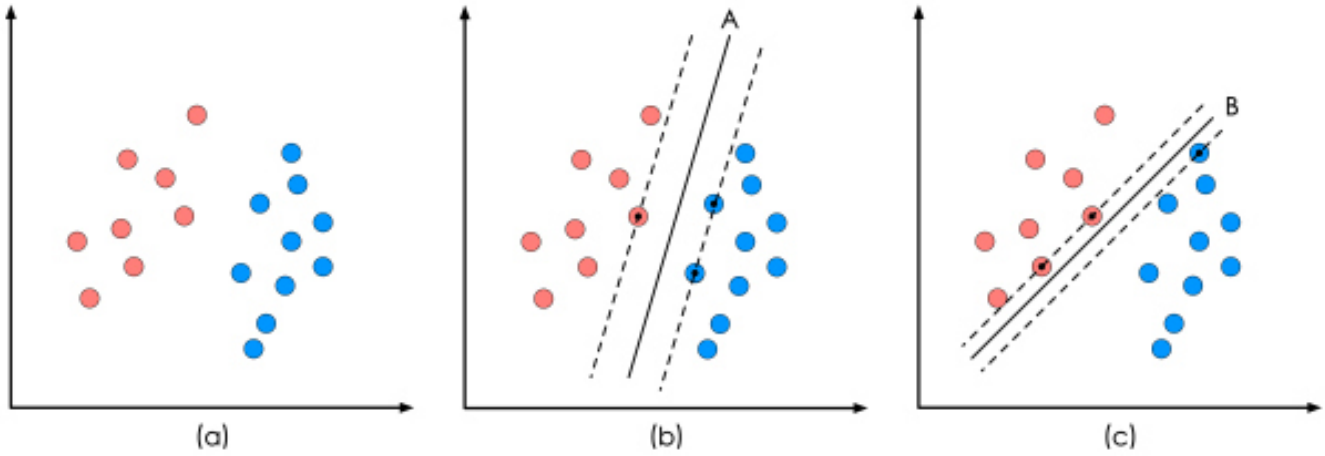
n=1时函数回归的示意图

回归问题

- **回忆分类中的情形**
  - 使靠超平面最近的样本点之间的间隔最大，最终可转化为一个凸二次规划问题的求解。

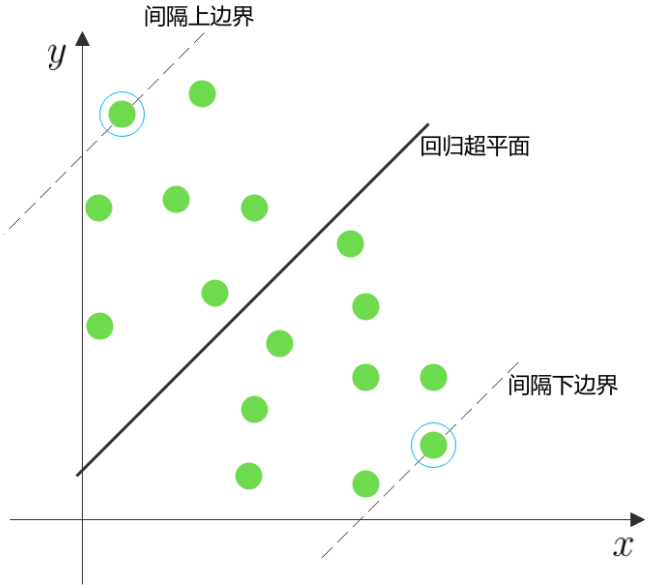
$$\min_{w,b} \frac{1}{2} \| w \|^2,$$

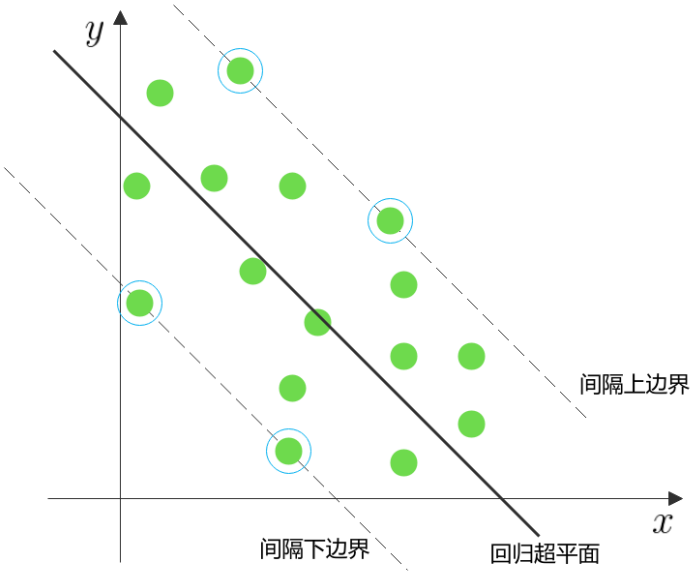
$$s.t. \quad y_i(w \cdot x_i + b) \geq 1, i = 1,2,\cdots,N$$



回归问题

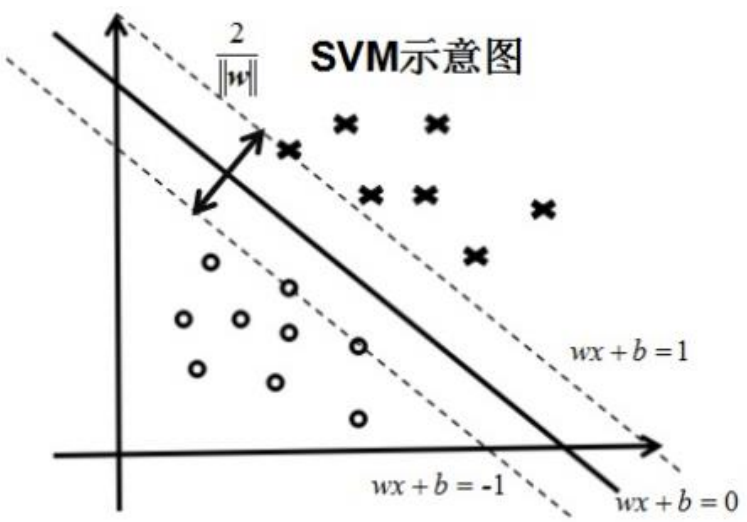
- 在Support Vector Regression(SVR)中，同样也是计算间隔，不同的是使靠超平面最远的样本点之间的间隔最小。（也就是包含了所有样本点，在两条虚线中间的线就是超平面）

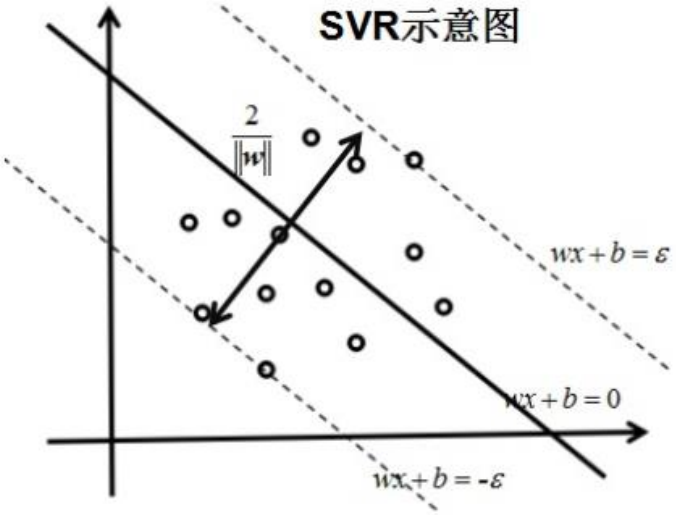




回归问题

- SVR与SVM对比



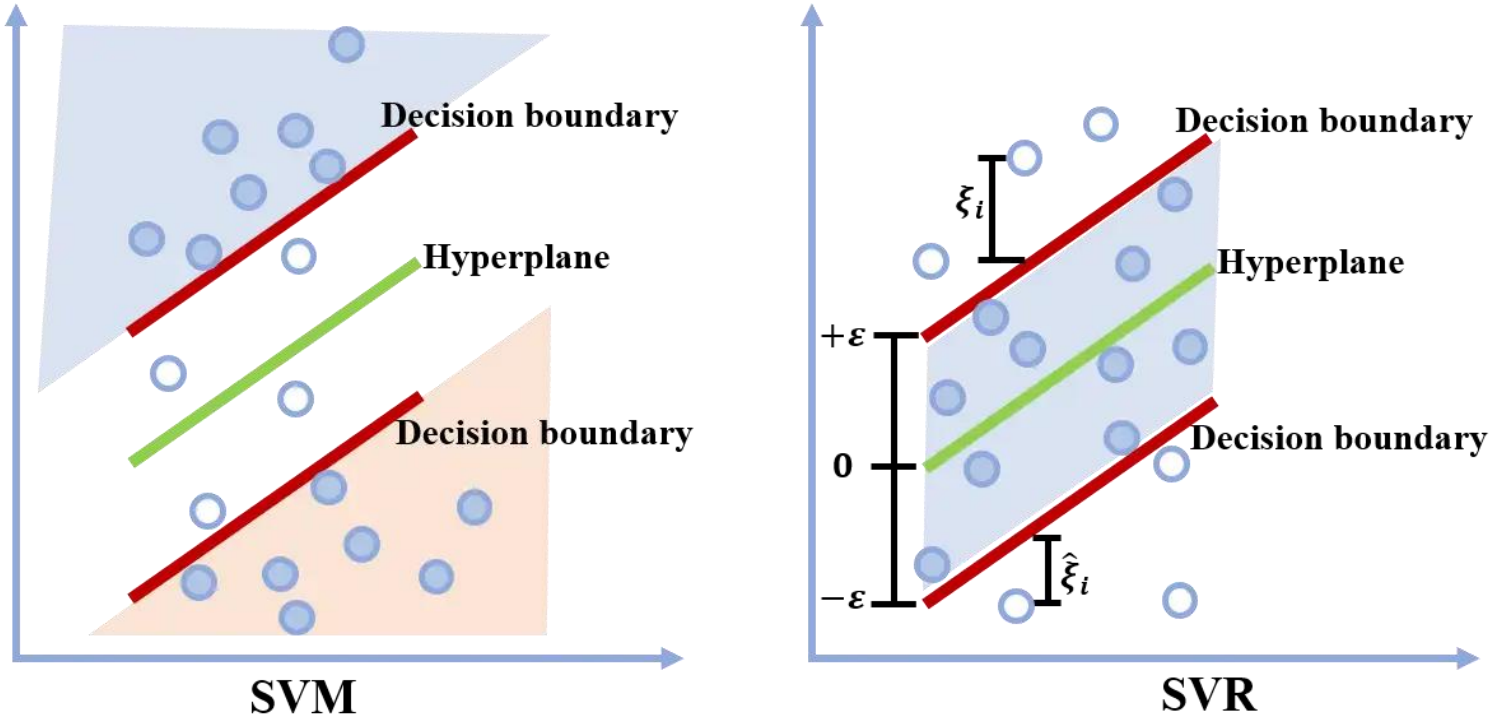


$$\min_{w,b}\frac{1}{2}\|w\|^2,$$
$$s.t.\ \ y_i(w\cdot x_i+b)\geq 1, i=1,2,\cdots,N$$

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

回归问题

- 为了去除异常数据对于结果的影响，SVR 同样可以加入松弛因子ξ。



回归问题

- SVR问题的数学表示
  - SVR对应的优化问题
    - ✓已知：l个观测样本 $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_l, y_l)$
    - ✓求解：

$$\begin{cases} \min \dfrac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ s.t. w \cdot \varphi(x_i) + b - y_i \le \varepsilon + \xi_i, \\ \quad y_i - w \cdot \varphi(x_i) - b \le \varepsilon + \xi_i^* \\ \quad \xi_i, \xi_i^* \ge 0 \ (i = 1,2,\ldots,l) \end{cases} \quad \text{(P)}$$

其中ξ_i和ξ_i^*为样本点松弛变量。SVR 只对间隔外的样本进行惩罚，当样本点位于间隔内时，则不计算其损失。

  - 目标：确定最优回归函数

$$f(\boldsymbol{x}) = w \cdot \varphi(\boldsymbol{x}) + b$$

回归问题

- 相应的Lagrange函数

$$L = \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*)$$
$$- \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i + y_i - w\cdot\varphi(x_i) - b)$$
$$- \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \xi_i^* - y_i + w\cdot\varphi(x_i) + b)$$

- 其中，Lagrange乘子满足：

$$\alpha_i, \alpha_i^* \geq 0, \eta_i, \eta_i^* \geq 0$$

回归问题

- 最优解的条件为：

$$\nabla_b L = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0$$

$$\nabla_w L = w - \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\varphi(x_i)\cdot\varphi(x_j) = 0$$

$$\nabla_\xi L = C - \alpha_i - \eta_i = 0$$
$$\nabla_{\xi^*} L = C - \alpha_i^* - \eta_i^* = 0$$

回归问题

- 原始问题的对偶问题

$$\max_{\alpha_i,\alpha_i^*} -\frac{1}{2}\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)\varphi(x_i)\cdot\varphi(x_j)$$
$$-\sum_{i=1}^{l}y_i(\alpha_i^*-\alpha_i)-\varepsilon\sum_{i=1}^{l}(\alpha_i^*+\alpha_i)$$
$$s.t.\quad \sum_{i=1}^{l}(\alpha_i^*-\alpha_i)=0,$$
$$0\le\alpha_i,\alpha_i^*\le C,(i=1,2,\cdots,l)$$

回归问题

- 对偶问题引入核函数，即

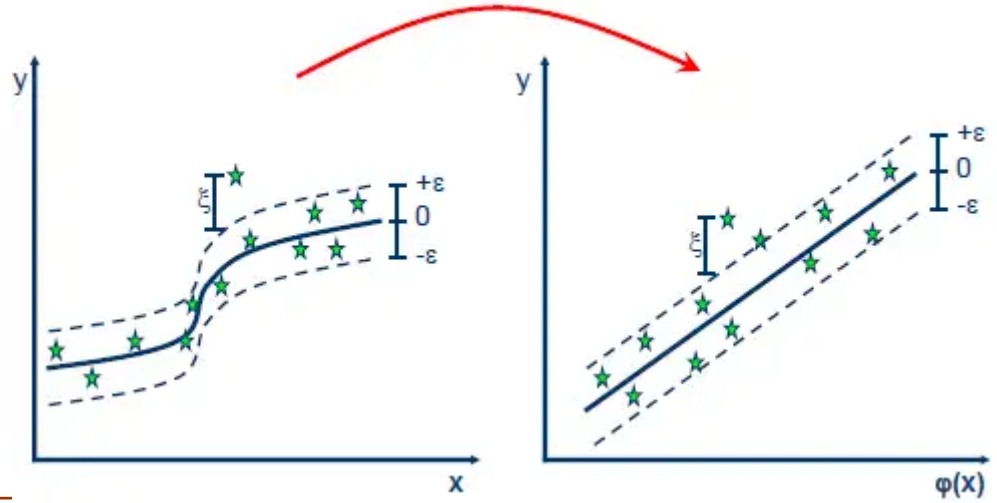
$$\max_{\alpha_i,\alpha_i^*} -\frac{1}{2}\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(x_i,x_j)$$
$$-\sum_{i=1}^{l}y_i(\alpha_i^*-\alpha_i)-\varepsilon\sum_{i=1}^{l}(\alpha_i^*+\alpha_i)$$
$$s.t.\quad \sum_{i=1}^{l}(\alpha_i^*-\alpha_i)=0, 0\le\alpha_i,\alpha_i^*\le C, (i=1,2,\cdots,l)$$

回归问题

- SVM回归器的决策函数

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

- **注**：事实上，大部分拉格朗日乘子都为0，只有少数样本对应的拉格朗日乘子非0，并满足：$\alpha_i\alpha_i^* = 0$



谢谢

附件：阅读材料

阅读材料

- An Overview of Statistical Learning Theory

  http://www.mit.edu/~6.454/www_spring_2001/emin/slt.pdf

阅读材料

- SVM是一种基于统计学习理论的模式识别方法。
- 它是由Boser,Guyon,Vapnik在COLT-92上首次提出，从此迅速的发展起来，现在已经在许多领域（生物信息学，文本和手写识别等）都取得了成功的应用。

COLT(Computational Learning Theory)

阅读材料

- 经验风险的计算
  - 期望风险$R(w)$要依赖联合概率$F(x,y)$的信息，实际问题中无法计算。
  - 一般用经验风险$R_{emp}(w)$代替期望风险$R(w)$

$$R_{emp}(w) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i, w))$$

错分的样本数

阅读材料

- 经验风险最小不等于期望风险最小，不能保证分类器的推广能力.
- 经验风险只有在样本数无穷大趋近于期望风险，需要非常多的样本才能保证分类器的性能。
- 需要找到经验风险最小和推广能力最大的平衡点。

阅读材料

- SVM的优势
  - 传统机器学习的基础：数理统计理论
    - ✓不足：过拟合，例如，神经网络
  - SVM的基础：统计学习理论
    - ✓优势1：不需要样本数量趋于无穷大
    - ✓优势2：能够衡量学习函数集的复杂程度
    - ✓优势3：把传统期望风险的估计拓展成结构风险