東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

人工智能概论

第六章：支持向量机

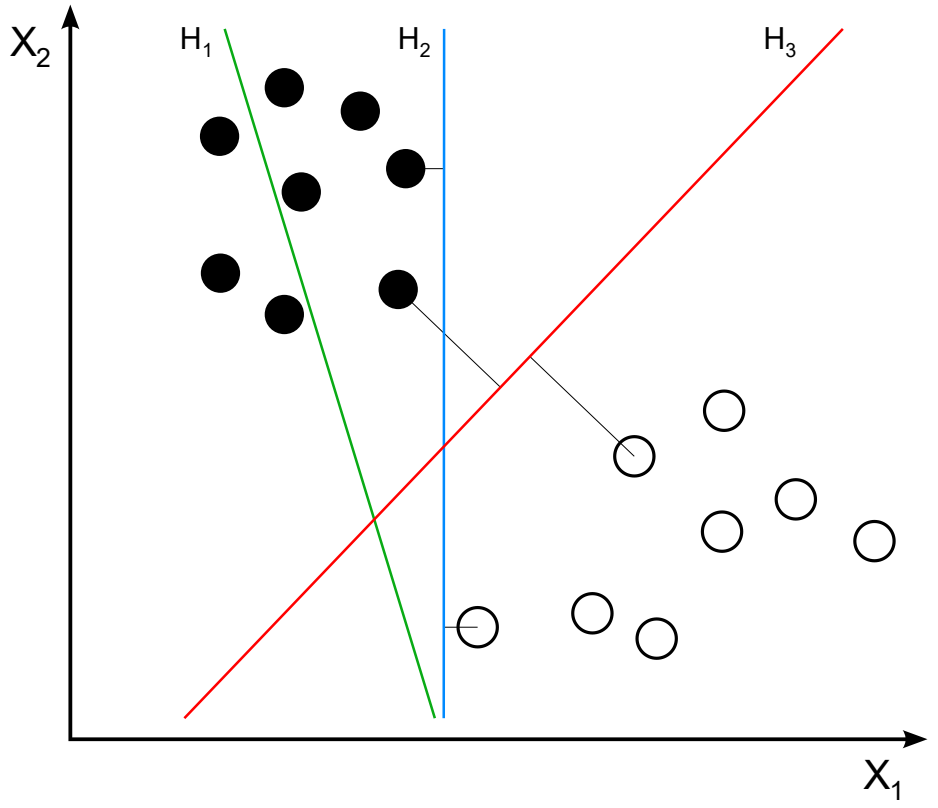
丁烨，计算机科学与技术学院
dingye@dgut.edu.cn

目录

基本概念

间隔与支持向量

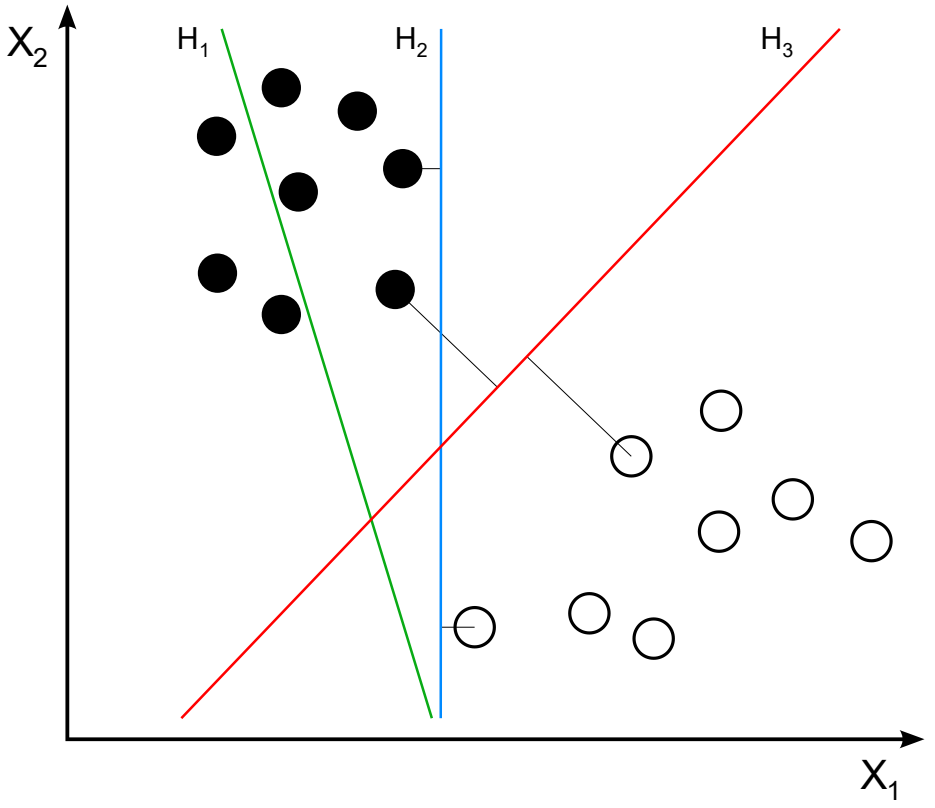
- 给定一个训练样本集
- 分类学习的目的是在样本空间中找到一个划分超平面
- 将不同类别的样本分开
- 但能将训练样本分开的划分超平面可能有很多
- 我们应该努力去找到哪一个呢？



基本概念

间隔与支持向量

- 直观上看，应该去找位于两类训练样本“正中间”的划分超平面
- 因为该划分超平面对训练样本局部扰动的“容忍”性最好
- 换言之，这个划分超平面所产生的分类结果是最鲁棒（Robust）的，对未见示例的泛化能力最强



基本概念

间隔与支持向量

- 鲁棒性（Robustness）
- 又称：健壮性、稳健性
- 指执行过程中处理错误，遭遇异常时继续正常运行的能力

基本概念

间隔与支持向量

- 在样本空间中，划分超平面可通过如下线性方程来描述：

$$w^T x + b = 0$$

- 其中 $w = (w_1, w_2, \cdots, w_d)$ 为法向量，决定了超平面的方向
- b 为位移项，决定了超平面与原点之间的距离
- 显然，划分超平面可被法向量 w 和位移 b 确定

基本概念

间隔与支持向量

- 样本空间中任意点 x 到超平面 (w, b) 的距离可写为：

$$r = \frac{\left|w^T x + b\right|}{\|w\|}$$

基本概念

间隔与支持向量

- 假设超平面 (w, b) 能将训练样本正确分类，即对于 $(x_i, y_i) \in D$：
- 若 $y_i = +1$，则有 $w^T x_i + b > 0$
- 若 $y_i = -1$，则有 $w^T x_i + b < 0$
- 那么：

$$\begin{cases} w^T x_i + b \geq +1, y_i = +1 \\ w^T x_i + b \leq -1, y_i = -1 \end{cases}$$

基本概念

间隔与支持向量

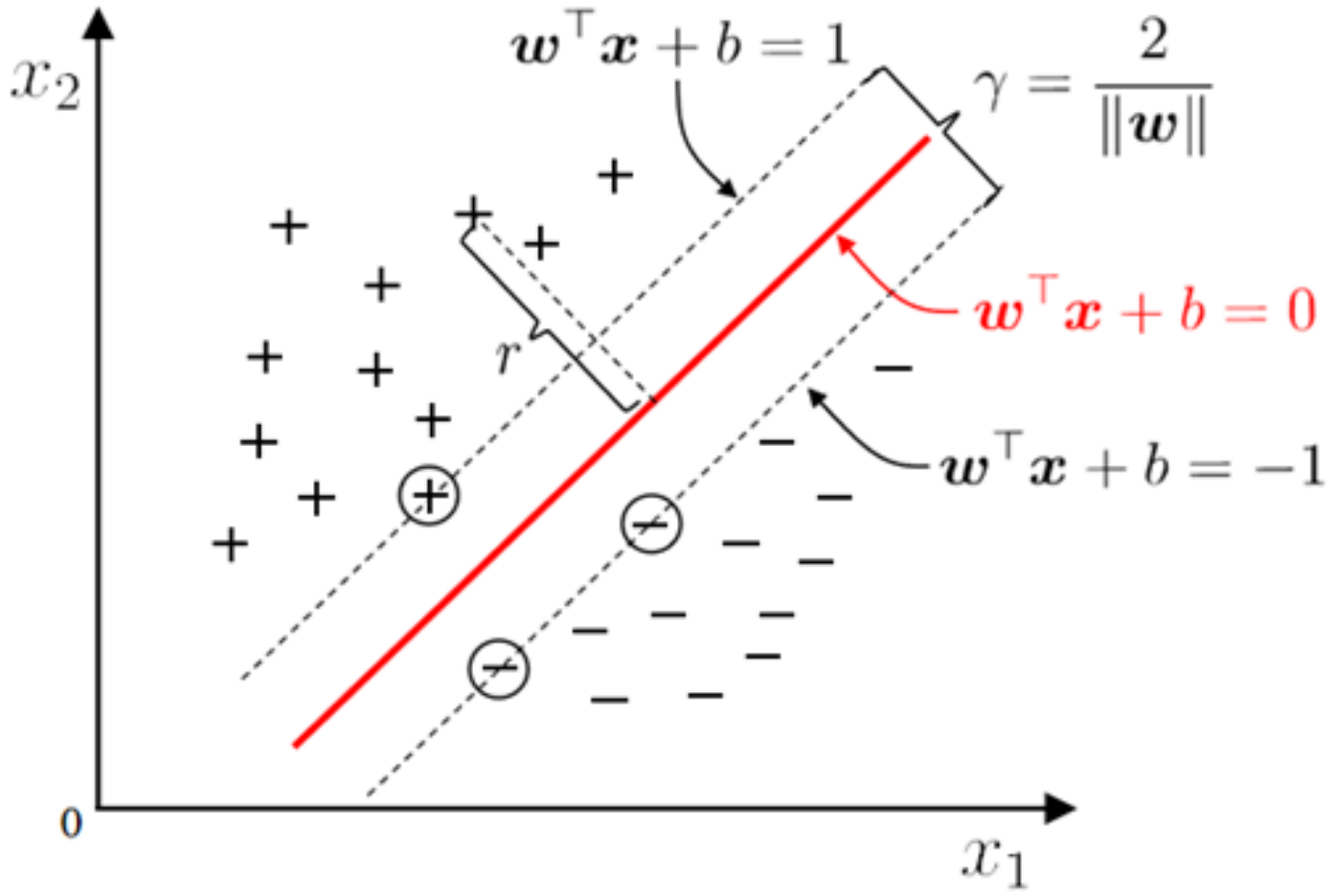
- 此时，距离超平面最近的这几个训练样本点使上述公式的等号成立
- 它们被称为“支持向量（Support Vector）”
- 两个异类支持向量到超平面的距离之和为：

$$\gamma = \frac{2}{\|w\|}$$

- 它被称为“间隔（Margin）”

基本概念

间隔与支持向量



基本概念

间隔与支持向量

- 欲找到具有“最大间隔（Maximum Margin）”的划分超平面
- 也就是要找到能满足约束的参数 w 和 b，使得 γ 最大
- 这就是支持向量机（Support Vector Machine，SVM）的基本原理

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

$$\text{s.t.} \quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1, \quad i = 1, 2, \ldots, m.$$

基本概念

对偶问题

- 支持向量机是一个：
- 凸二次规划（Convex Quadratic Programming）问题
- 这个问题可以直接用现成的优化计算求解，但效率很低
- 我们可以使用拉格朗日乘数法得到其“对偶问题（Dual Problem）”

基本概念

对偶问题

- 对偶问题（Dual Problem）
- 每一个线性规划问题都伴随有另一个线性规划问题，称为对偶问题
- 原来的线性规划问题则称为原始线性规划问题，简称原始问题
- 对偶问题有许多重要的特征
- 例如：目标函数对原始问题是极大化，对对偶问题则是极小化
- 它的变量能提供关于原始问题最优解的许多重要资料
- 有助于原始问题的求解和分析

基本概念

对偶问题

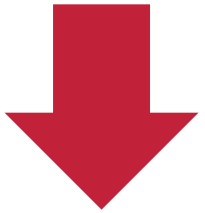
- 拉格朗日乘数法（Lagrange Multiplier）
- 寻找多元函数在其变量受到一个或多个条件的约束时的局部极值的方法
- 可以将一个有 n 个变量与 k 个约束条件的最优化问题转换为一个解有 $n+k$ 个变量的方程组的解的问题
- 这种方法中引入了一个或一组新的未知数，即拉格朗日乘数
- 它们是在转换后的方程，即约束方程中作为梯度（Gradient）的线性组合中各个向量的系数

基本概念

对偶问题

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

$$\text{s.t.} \quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1, \quad i = 1, 2, \ldots, m.$$

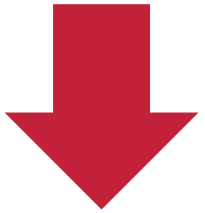


$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b)\right) \ ,$$

基本概念

对偶问题

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



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

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

$$\alpha_i \geqslant 0 , \quad i = 1, 2, \ldots, m .$$

基本概念

解的稀疏性

- 从对偶问题解出的 α_i 是拉格朗日乘数，对应着训练样本 (x_i, y_i)
- 解决对偶问题时需满足 KKT 条件：

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\boldsymbol{x}_i) \geq 1, \\ \alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0. \end{cases}$$

基本概念

解的稀疏性

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\boldsymbol{x}_i) \geq 1, \\ \alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0. \end{cases}$$

- 若 $\alpha_i = 0$，则该样本不会对模型有任何影响
- 若 $\alpha_i > 0$，则必有 $y_i f(x_i) = 0$，所对应样本点位于最大间隔边界上，是一个支持向量

基本概念

解的稀疏性

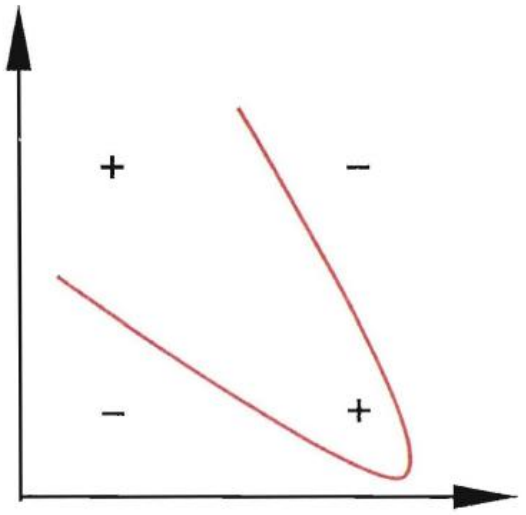
- 支持向量机的对偶问题是一个二次规划问题
- 可使用通用的二次规划算法来求解
- 然而，该问题的规模正比于训练样本数，开销很大
- 为了降低开销，人们通过利用问题本身的特性，提出了很多高效算法
- SMO（Sequential Minimal Optimization）是其中一个著名的代表
- 模型训练完后，大部分的训练样本都不需要保留
- 最终模型仅仅与支持向量有关

目录

核函数

基本概念

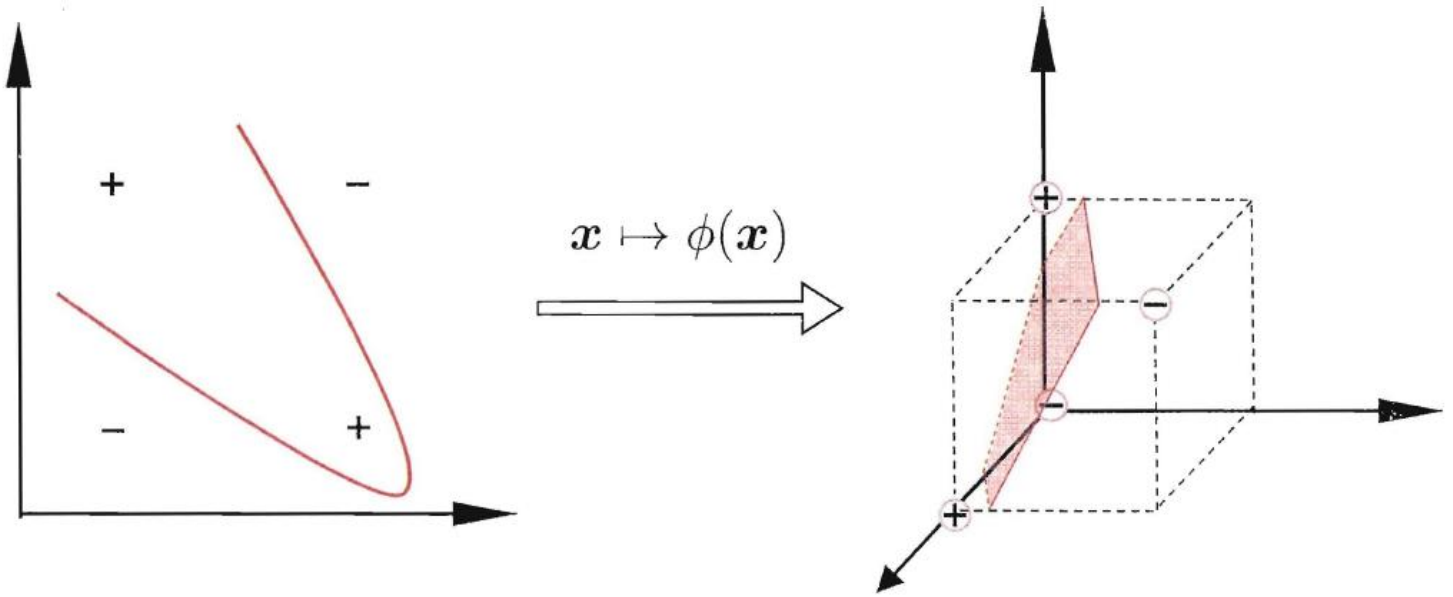
- 在前面的讨论中，我们假设训练样本是线性可分的
- 即存在一个划分超平面能将训练样本正确分类
- 然而在现实任务中，例如“异或”问题就不是线性可分的



核函数

基本概念

- 对这样的问题，可将样本从原始空间映射到一个更高维的特征空间
- 使得样本在这个特征空间内线性可分
- 例如将原始“异或”问题的二维空间映射到一个合适的三维空间



核函数

基本概念

- 令 $\phi(x)$ 表示将 x 映射后的特征向量
- 在特征空间中划分超平面所对应的模型可表示为：

$$f(x) = w^T \phi(x) + b$$

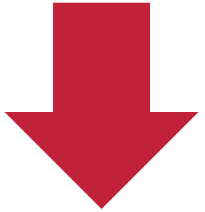
- 其中 w 和 b 是模型参数

核函数

基本概念

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

$$\text{s.t.} \quad y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b) \geqslant 1, \quad i=1,2,\ldots,m.$$



$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$$

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

$$\alpha_i \geqslant 0\ , \quad i=1,2,\ldots,m\ .$$

核函数

基本概念

- 这个对偶问题的求解涉及到计算 $\phi(x_i)^T\phi(x_j)$
- 这是样本 x_i 与 x_j 映射到特征空间之后的内积
- 由于特征空间维数可能很高，甚至可能是无穷维
- 因此直接计算 $\phi(x_i)^T\phi(x_j)$ 通常是很困难的

核函数

基本概念

- 为了避开这个障碍，可以设想这样一个函数：

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

- 即 x_i 与 x_j 在特征空间的内积等于它们在原始样本空间中
- 通过函数 $\kappa(\cdot,\cdot)$ 即“核函数（Kernel Function）”计算的结果
- 有了核函数，我们就不必直接去计算高维甚至无穷维特征空间中的内积

核函数

基本概念

- 显然，若已知合适映射 $\phi(\cdot)$ 的具体形式
- 则可写出核函数 $\kappa(\cdot,\cdot)$
- 但在现实任务中我们通常不知道 $\phi(\cdot)$ 是什么形式
- 那么，合适的核函数是否一定存在呢？
- 什么样的函数能做核函数呢？

核函数

基本概念

- 只要一个对称函数所对应的核矩阵半正定，它就能作为核函数使用

$$\mathbf{K} = \begin{bmatrix} \kappa(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_i, \boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_i, \boldsymbol{x}_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_m, \boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_m, \boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_m, \boldsymbol{x}_m) \end{bmatrix}.$$

核函数

基本概念

❖ 常用的核函数：

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j)^d$	$d \geq 1$为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\delta^2}\right)$	$\delta > 0$为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\delta}\right)$	$\delta > 0$
Sigmoid核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^\top \boldsymbol{x}_j + \theta)$	tanh为双曲正切函数, $\beta > 0, \theta < 0$

核函数

基本概念

- 此外，核函数还可通过函数组合得到，例如：
- 核函数的线性组合也是核函数
- 则核函数的直积也是核函数
- 若 κ 为核函数，则对于任意函数 g，下列函数也是核函数：

$$\kappa(x,z) = g(x)\kappa(x,z)g(z)$$

目录

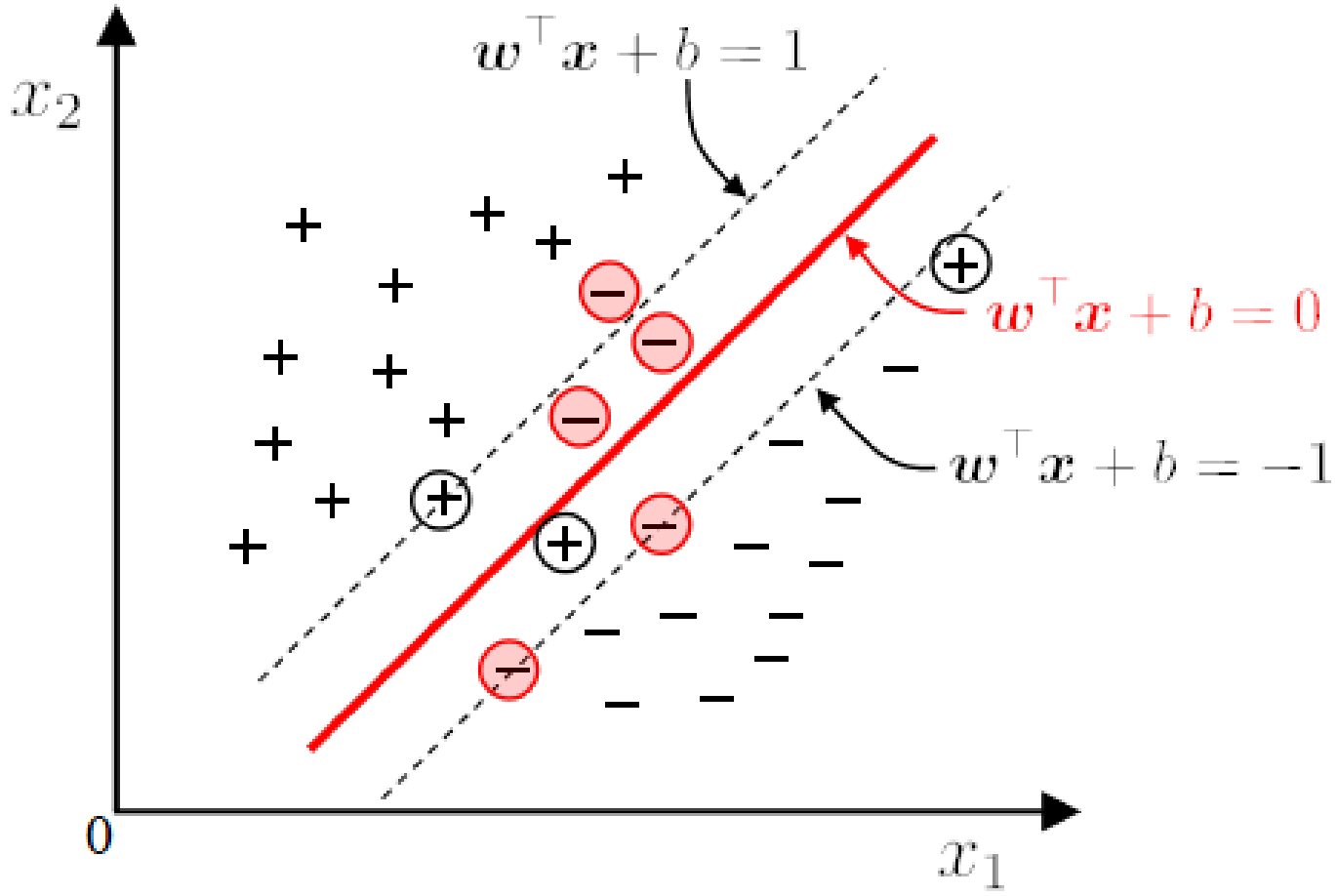
软间隔与正则化

软间隔

- 现实中，很难确定合适的核函数使得训练样本在特征空间中线性可分
- 同时一个线性可分的结果也很难断定是否是有过拟合造成的
- 我们可以引入“软间隔（Soft Margin）”的概念
- 允许支持向量机在一些样本上不满足约束

软间隔与正则化

软间隔



软间隔与正则化

软间隔

- 具体来说，前面介绍的支持向量机形式是要求所有样本都必须划分正确
- 这称为“硬间隔（Hard Margin）”
- 而软间隔则是允许某些样本不满足约束：

$$y_i(w^T x_i + b) \geq 1$$

软间隔与正则化

软间隔

- 当然，在最大化间隔的同时，不满足约束的样本应尽可能少
- 于是，优化目标可写为：

$$\min_{\boldsymbol{w},b} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} l_{0/1}\left(y_i(\boldsymbol{w}^\top\phi(\boldsymbol{x}_i)+b)-1\right)$$

- 其中 $C>0$ 是一个常数
- $l_{0/1}$ 是 “0/1 损失函数”

$$l_{0/1} = \begin{cases} 1 & z<0 \\ 0 & otherwise \end{cases}$$

软间隔与正则化

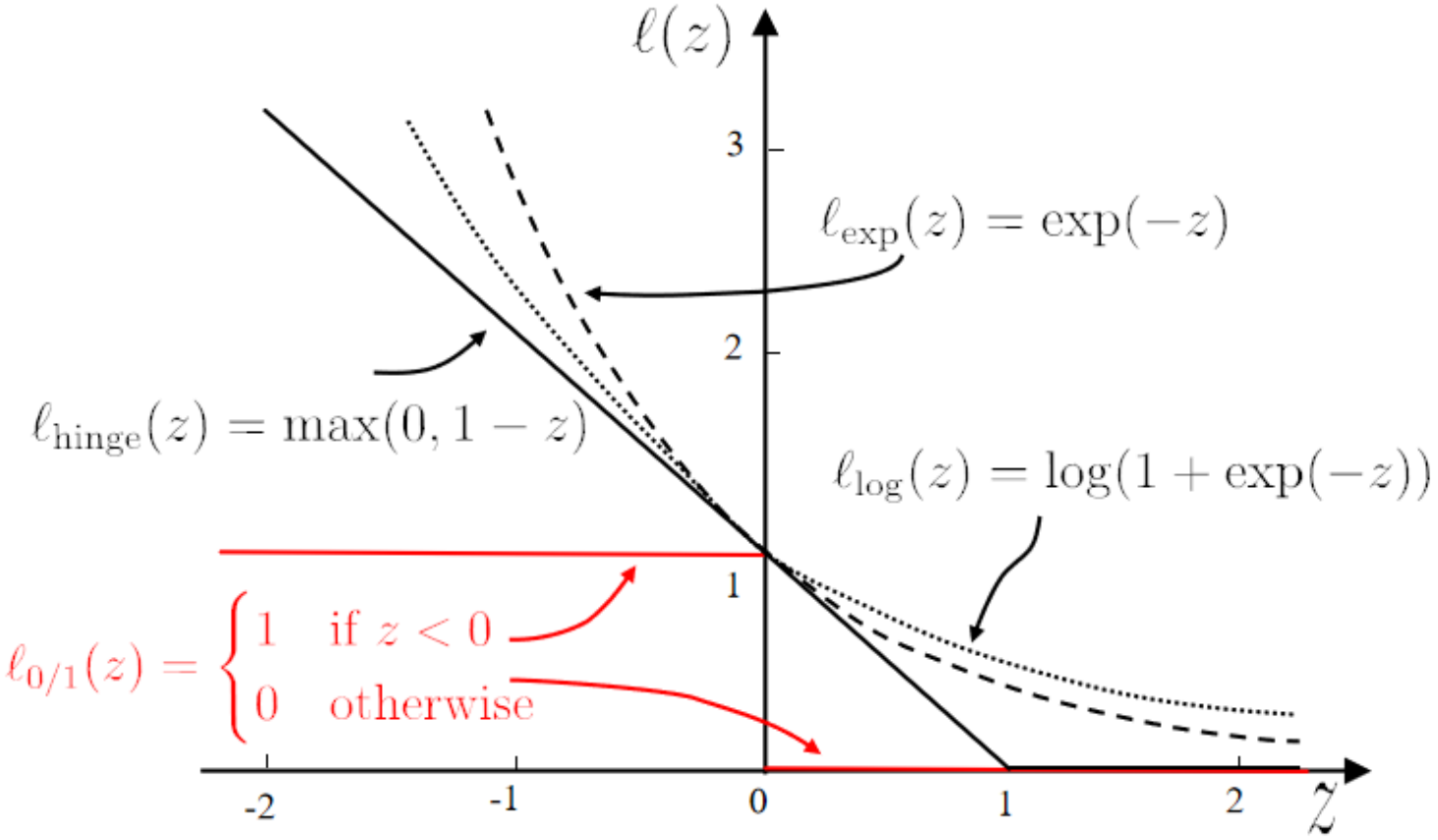
软间隔

- 然而，$l_{0/1}$ 非凸、非连续，数学性质不太好，不易直接求解
- 于是，通常用其他一些函数来代替 $l_{0/1}$
- 称为“替代损失（Surrogate Loss）”
- 替代损失函数一般具有较好的数学性质
- 如它们通常是凸的连续函数且是 $l_{0/1}$ 的上界

软间隔与正则化

软间隔

❖ 三种常用的替代损失函数：



软间隔与正则化

正则化

- 我们还可以把 $l_{0/1}$ 换成别的替代损失函数以得到其他学习模型
- 这些模型的性质与所用的替代函数直接相关，但它们具有一个共性：
- 优化目标中的第一项用来描述划分超平面的“间隔”大小
- 另一项用来表述训练集上的误差，可写为更一般的形式：

$$\min_{f} \ \Omega(f) + C \sum_{i=1}^{m} l(f(\boldsymbol{x}_i), y_i)$$

软间隔与正则化

正则化

$$\min_f \ \Omega(f) + C\sum_{i=1}^{m} l(f(\boldsymbol{x}_i), y_i)$$

- 其中 $\Omega(f)$ 称为“结构风险（Structural Risk）”
- 用于描述模型 f 的某些性质
- 第二项 $\sum_{i=1}^{m} l(f(x_i), y_i)$ 称为“经验风险（Empirical Risk）”
- 用于描述模型与训练数据的契合程度
- C 用于对二者进行折中

软间隔与正则化

正则化

- 从经验风险最小化的角度来看
- $\Omega(f)$ 表述了我们希望获得具有何种性质的模型
- 例如：希望获得复杂度较小的模型
- 另一方面，该信息有助于削减假设空间
- 从而降低了最小化训练误差的过拟合风险
- 从这个角度来说，上式被称为“正则化（Regularization）”问题
- $\Omega(f)$ 称为正则化项，C 则称为正则化常数
- L_p 范数（Norm）是常用的正则化项

目录

支持向量回归

基本概念

- 支持向量的原理也可以用来解决回归问题
- 给定训练样本集 $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$
- 我们希望学得一个回归模型
- 使得 $f(x)$ 与 y 尽可能接近
- w 和 b 是待确定的模型参数

支持向量回归

基本概念

- 对样本 (x, y)，传统回归模型通常直接基于模型输出 $f(x)$ 与真实输出 y 之间的差别来计算损失
- 当且仅当 $f(x)$ 与 y 完全相同时，损失才为零
- 与此不同，支持向量回归（Support Vector Regression，SVR）
- 假设我们能容忍 $f(x)$ 与 y 之间最多有 ϵ 的偏差
- 即仅当 $f(x)$ 与 y 之间的差别绝对值大于 ϵ 时才计算损失

支持向量回归

基本概念

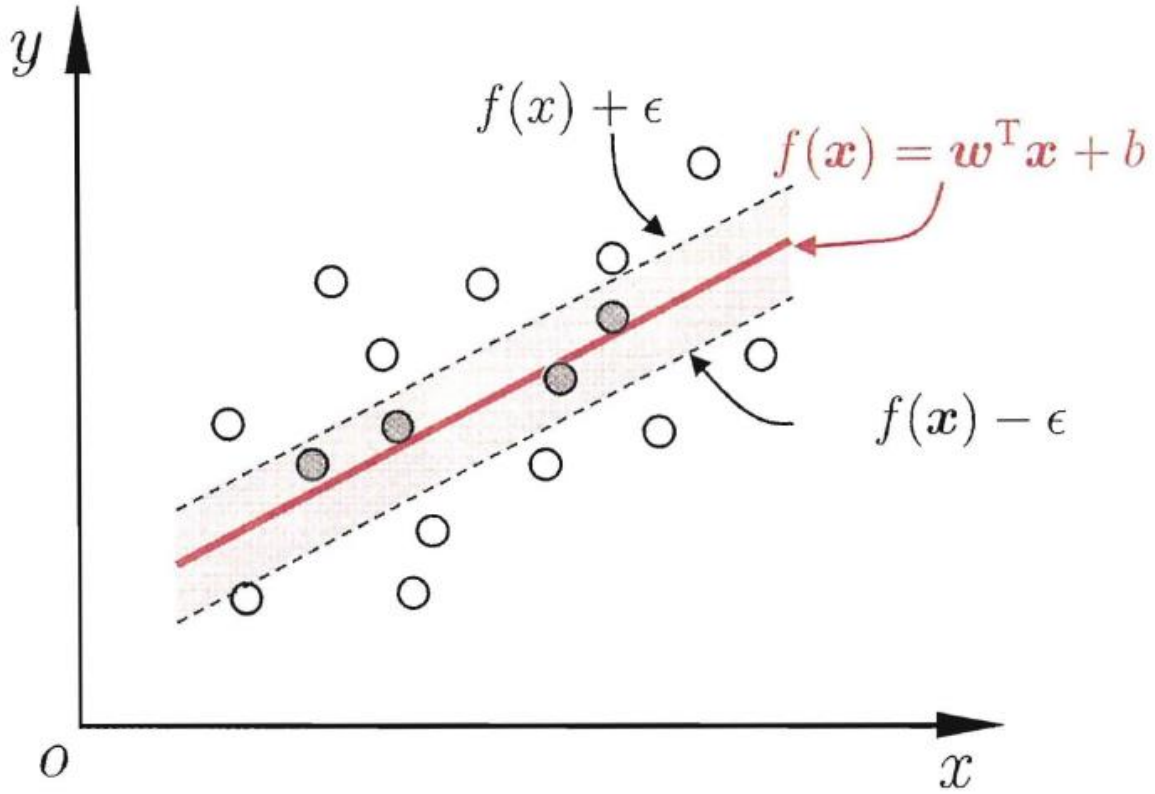


图 6.6 支持向量回归示意图. 红色显示出 ϵ-间隔带, 落入其中的样本不计算损失.

支持向量回归

基本概念

- 于是，SVR 问题可形式化为：

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\ell_\epsilon(f(\boldsymbol{x}_i)-y_i) ,$$

- 其中 C 为正则化常数
- l_ϵ 是 ϵ-不敏感损失（ϵ-insensitive Loss）函数

$$\ell_\epsilon(z) = \begin{cases} 0, & \text{if } |z| \leqslant \epsilon ; \\ |z|-\epsilon, & \text{otherwise.} \end{cases}$$

支持向量回归

基本概念

- 于是，SVR 问题可形式化为：

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\ell_\epsilon(f(\boldsymbol{x}_i)-y_i) ,$$

- 其中 C 为正则化常数
- l_ϵ 是 ϵ-不敏感损失（ϵ-insensitive Loss）函数

$$\ell_\epsilon(z) = \begin{cases} 0, & \text{if } |z| \leqslant \epsilon ; \\ |z|-\epsilon, & \text{otherwise.} \end{cases}$$

总结

- 「西瓜书」周志华《机器学习》，清华大学出版社
- https://item.jd.com/12762673.html

- 「南瓜书」谢文睿、秦州《机器学习公式详解》，人民邮电出版社
- https://github.com/datawhalechina/pumpkin-book/

总结

- Scikit-Learn
- https://scikit-learn.org

- TensorFlow
- https://www.tensorflow.org

東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

Thank You!

丁烨，计算机科学与技术学院
dingye@dgut.edu.cn