支持向量机

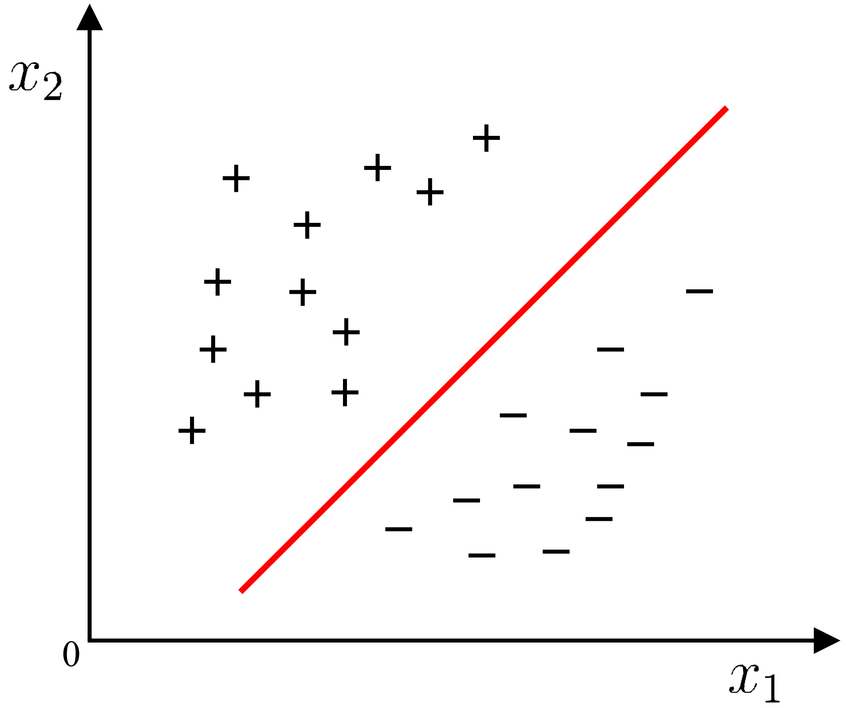
杜逆索

大纲

- 间隔与支持向量
- 对偶问题
- 核函数
- 软间隔与正则化
- 支持向量回归
- 核方法

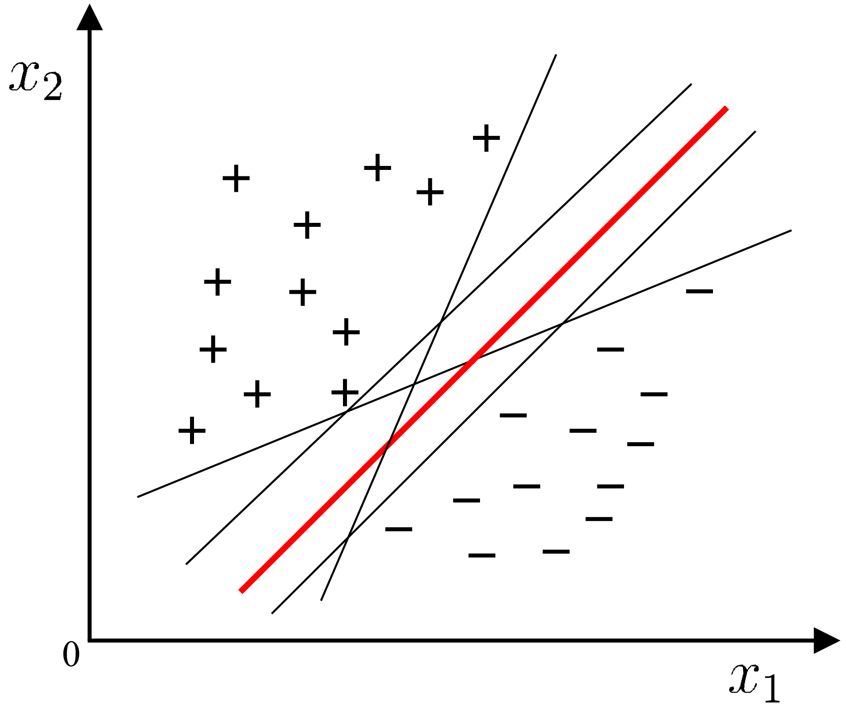
引子

线性模型：在样本空间中寻找一个超平面，将不同类别的样本分开.



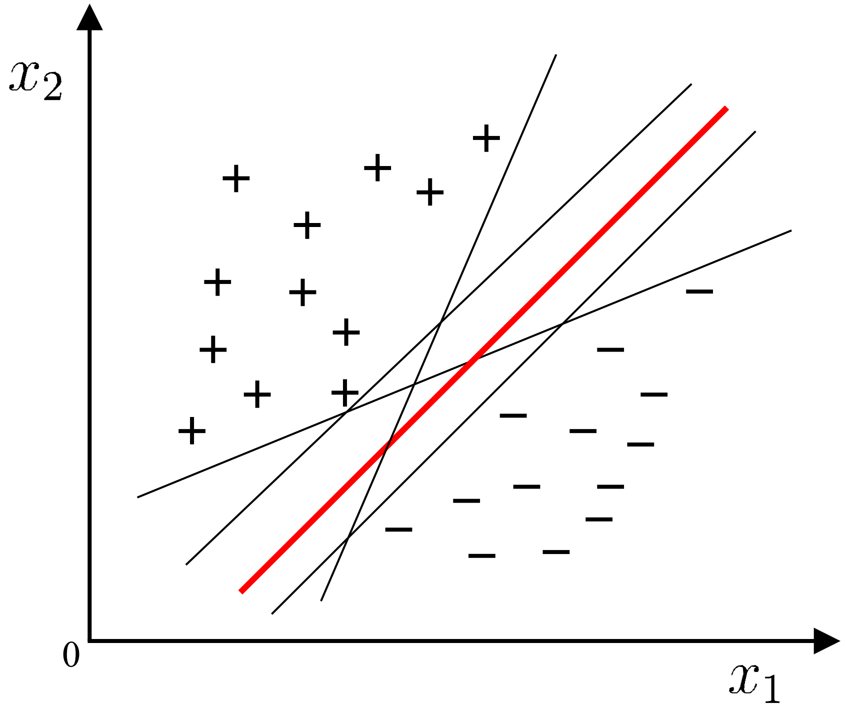
引子

-Q:将训练样本分开的超平面可能有很多, 哪一个好呢?



引子

-Q:将训练样本分开的超平面可能有很多, 哪一个好呢?

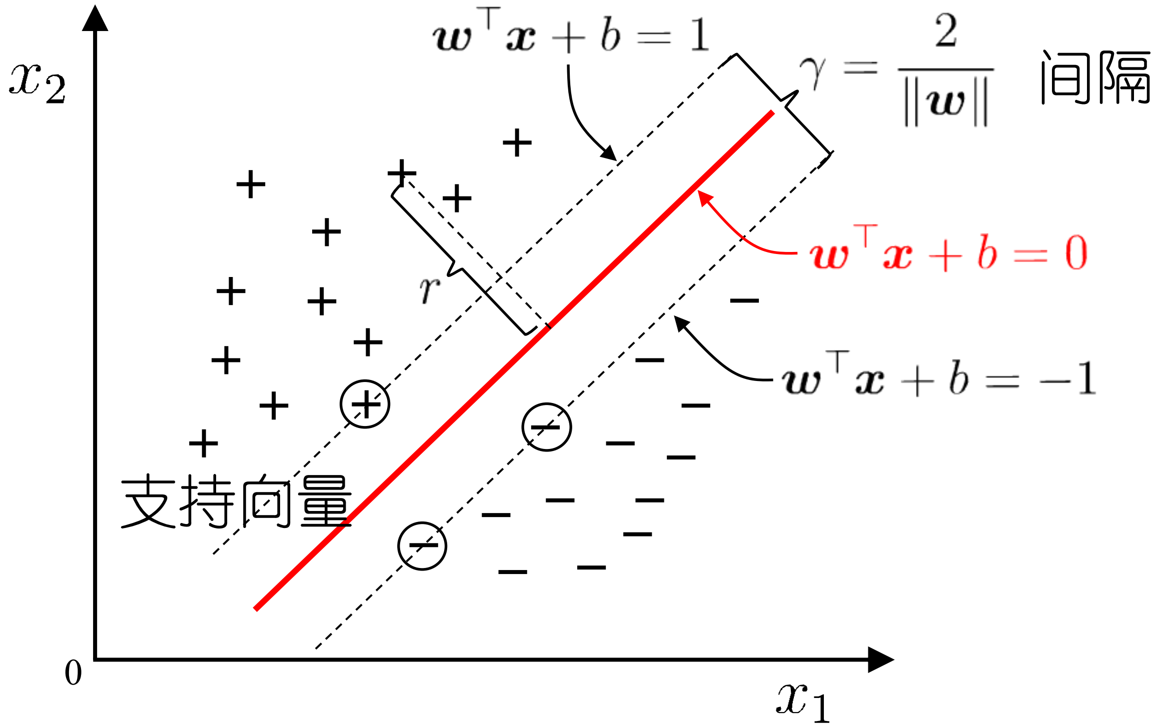


-A:应选择"正中间", 容忍性好, 鲁棒性高, 泛化能力最强.

间隔与支持向量

超平面方程: $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$

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



课堂练习

- 试证明样本空间中任意点 x 到超平面 (w, b) 的距离为式

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

支持向量机基本型

- 最大间隔：寻找参数 $\boldsymbol{w}$ 和 b, 使得 γ 最大.

$$\arg\max_{\boldsymbol{w},b} \quad \frac{2}{\|\boldsymbol{w}\|}$$
$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \; i = 1, 2, \ldots, m.$$

$$\arg\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \; i = 1, 2, \ldots, m.$$

SVM的基本型

大纲

- 间隔与支持向量
- 对偶问题
- 核函数
- 软间隔与正则化
- 支持向量回归
- 核方法

对偶问题

- 拉格朗日乘子法
  - 第一步：引入拉格朗日乘子 $\alpha_i \geq 0$ 得到拉格朗日函数

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

  - 第二步：令 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ 对 $\boldsymbol{w}$ 和 b 的偏导为零可得

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i, \quad \sum_{i=1}^{m} \alpha_i y_i = 0.$$

  - 第三步：回代

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

$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \ \alpha_i \geq 0, \ i = 1, 2, \ldots, m.$$

解的稀疏性

- 最终模型：$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = \boxed{\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x}} + b$

- KKT条件：

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

$$y_i f(\boldsymbol{x}_i) > 1 \quad \Rightarrow \quad \alpha_i = 0$$

支持向量机解的稀疏性：训练完成后，大部分的训练样本都不需保留，最终模型仅与支持向量有关.

求解方法 - SMO

- 基本思路：不断执行如下两个步骤直至收敛.
  - 第一步：选取一对需更新的变量α_i和α_j.
  - 第二步：固定α_i和α_j以外的参数，求解对偶问题更新α_i和α_j.

- 仅考虑 α_i和α_j 时，对偶问题的约束变为

$$\alpha_i y_i + \alpha_j y_j = -\sum_{k \neq i,j} \alpha_k y_k, \quad \alpha_i \geq 0, \quad \alpha_j \geq 0.$$

> 用一个变量表示另一个变量，回代入对偶问题可得一个单变量的二次规划，该问题具有闭式解.

- 偏移项b：通过支持向量来确定.

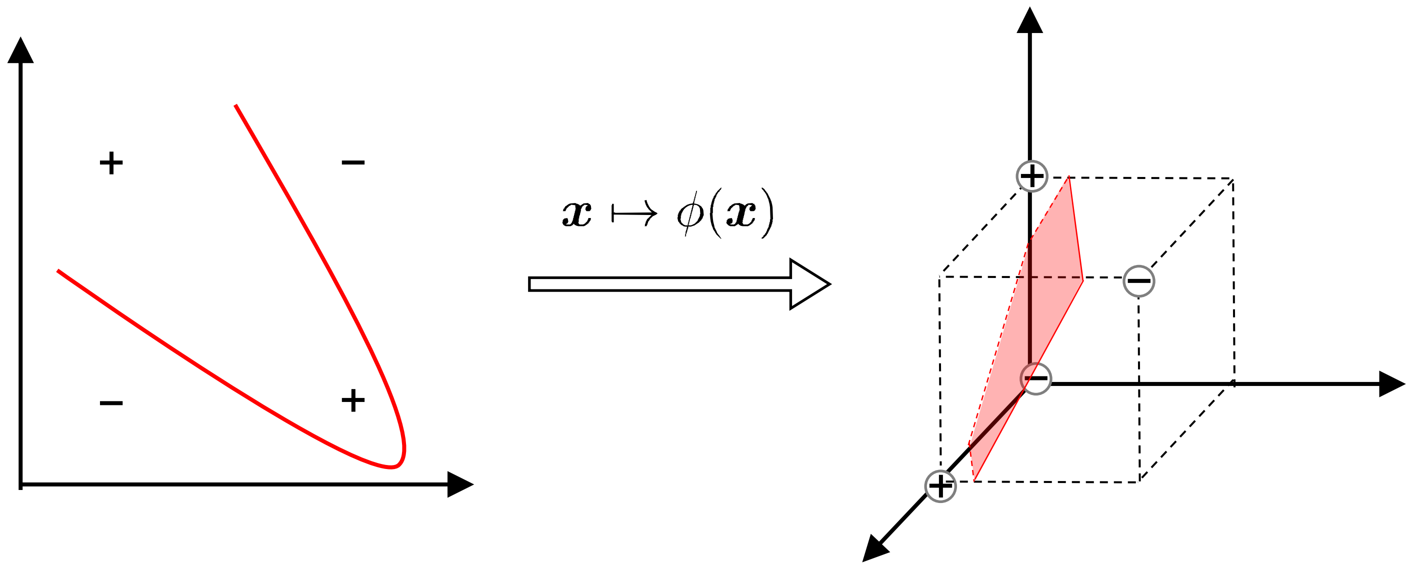
大纲

- 间隔与支持向量
- 对偶问题
- 核函数
- 软间隔与正则化
- 支持向量回归
- 核方法

线性不可分

-Q:若不存在一个能正确划分两类样本的超平面, 怎么办?

-A: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.



核支持向量机

- 设样本 $\boldsymbol{x}$ 映射后的向量为 $\phi(\boldsymbol{x})$，划分超平面为 $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b$.

原始问题

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

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

对偶问题

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boxed{\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)} - \sum_{i=1}^{m} \alpha_i$$

$$\text{s.t.} \ \ \sum_{i=1}^{m} \alpha_i y_i = 0,$$ 只以内积的形式出现.

预测

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \boxed{\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x})} + b$$

核函数

- 基本想法：不显式地设计核映射，而是设计核函数.

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$$

- Mercer定理(充分非必要)：只要一个对称函数所对应的核矩阵半正定，则它就能作为核函数来使用.

- 常用核函数：

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，$\beta > 0$, $\theta < 0$

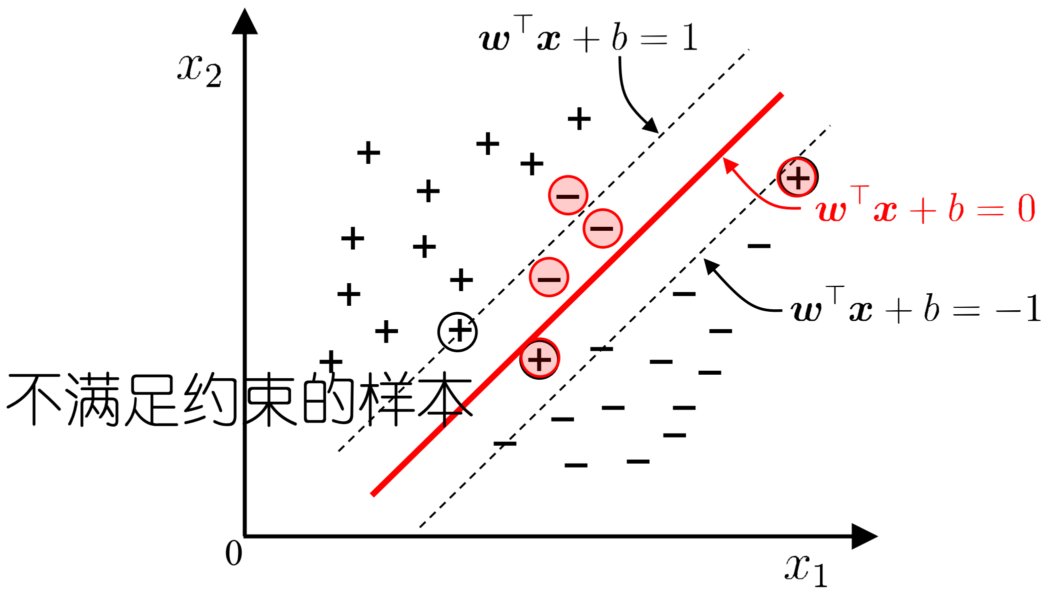
大纲

- 间隔与支持向量
- 对偶问题
- 核函数
- 软间隔与正则化
- 支持向量回归
- 核方法

软间隔

-Q: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.

-A:引入"软间隔"的概念，允许支持向量机在一些样本上不满足约束.



0/1损失函数

- 基本想法：最大化间隔的同时，让不满足约束的样本尽可能少.

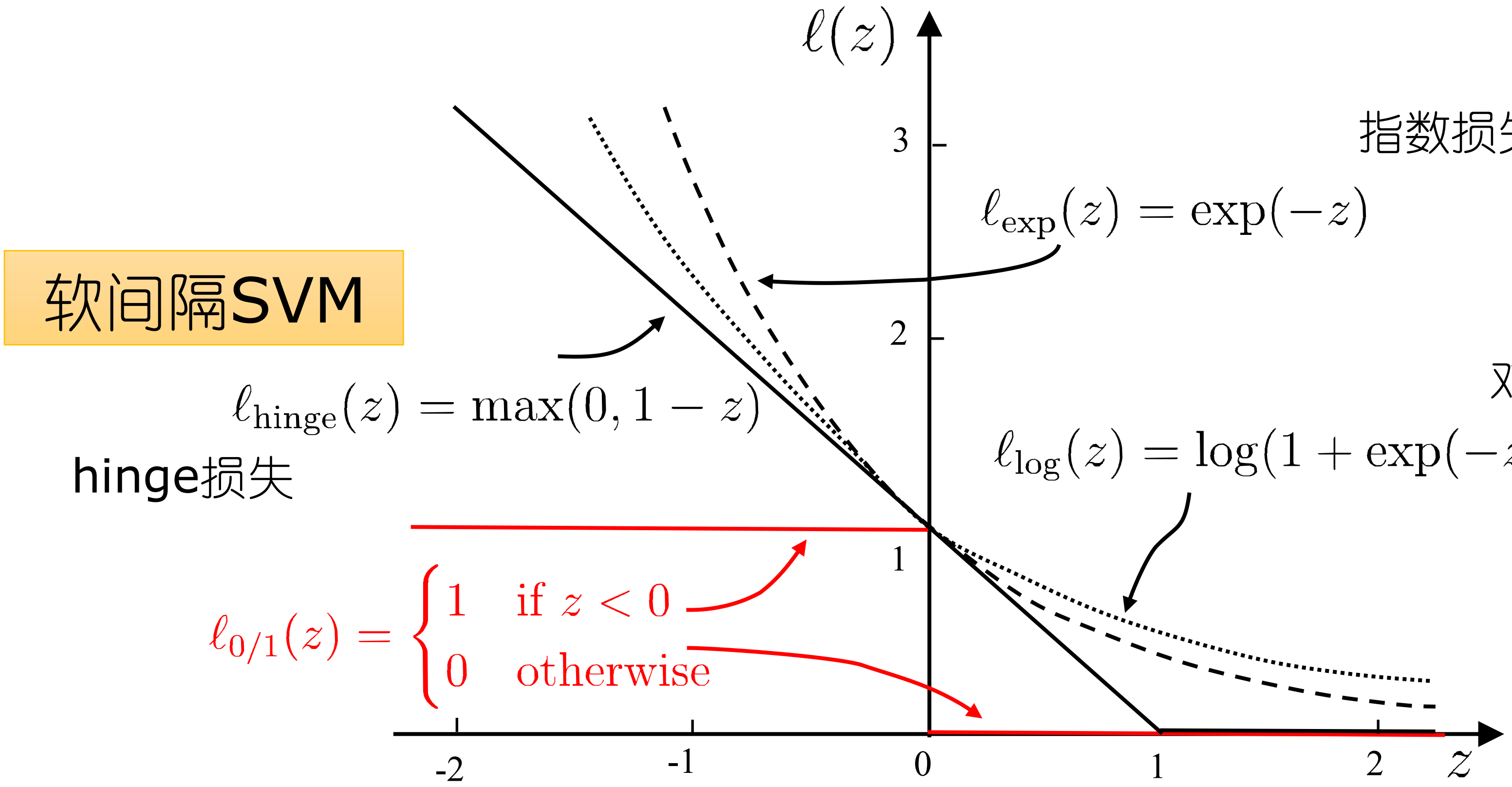
$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} l_{0/1}\left(y_i(\boldsymbol{w}^\top\phi(\boldsymbol{x}_i)+b)-1\right)$$

其中 $l_{0/1}$ 是"0/1损失函数"

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

- 存在的问题：0/1损失函数非凸、非连续，不易优化！

替代损失



替代损失函数数学性质较好，一般是0/1损失函数的上界

软间隔支持向量机

原始问题

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

对偶问题

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

$$\text{s.t.} \ \sum_{i=1}^{m}\alpha_i y_i = 0, \ \ 0 \le \alpha_i \le C, \ \ i = 1, 2, \ldots, m.$$

根据KKT条件可推得最终模型仅与支持向量有关，也即hinge损失函数依然保持了支持向量机解的稀疏性.

正则化

- 支持向量机学习模型的更一般形式

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

结构风险，描述模型的某些性质

经验风险，描述模型与训练数据的契合程度

可理解为一种“罚函数法”，对不希望得到的结果施以惩罚，从而使得优化过程趋向于希望目标

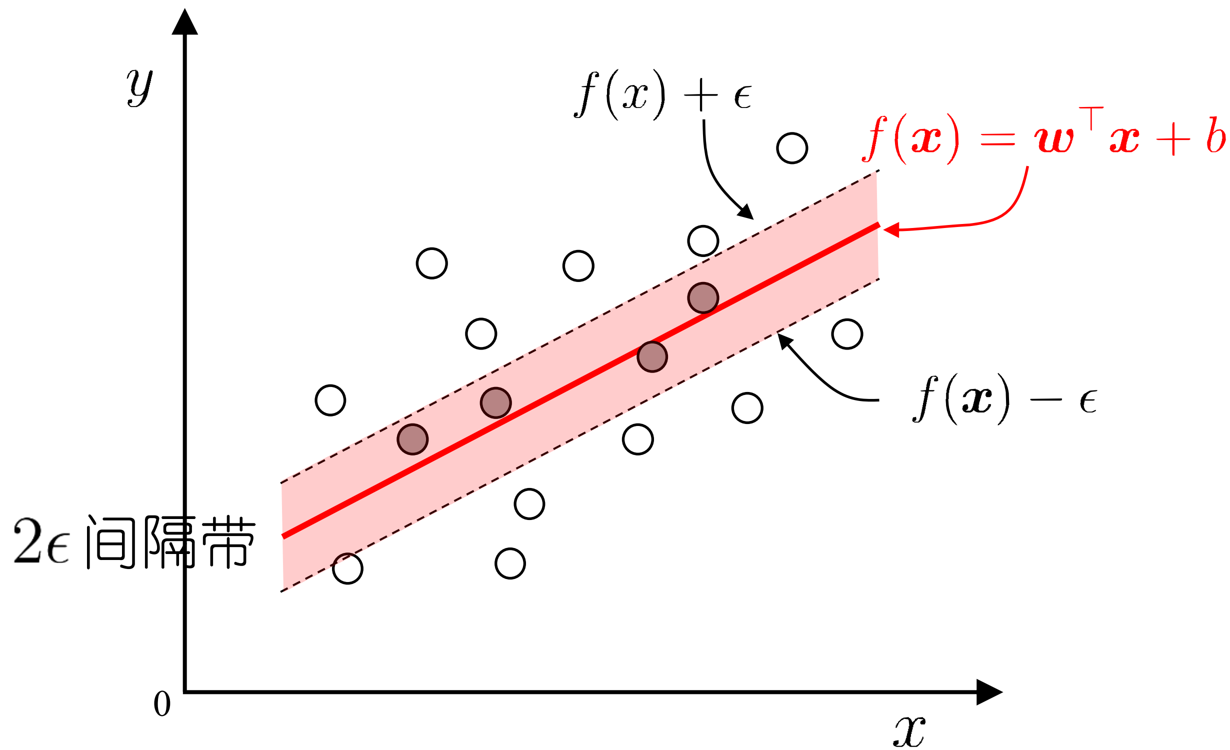
- 通过替换上面两个部分，可以得到许多其他学习模型
  - 对数几率回归(Logistic Regression)
  - 最小绝对收缩选择算子(LASSO)
  - ……

大纲

- 间隔与支持向量
- 对偶问题
- 核函数
- 软间隔与正则化
- 支持向量回归
- 核方法

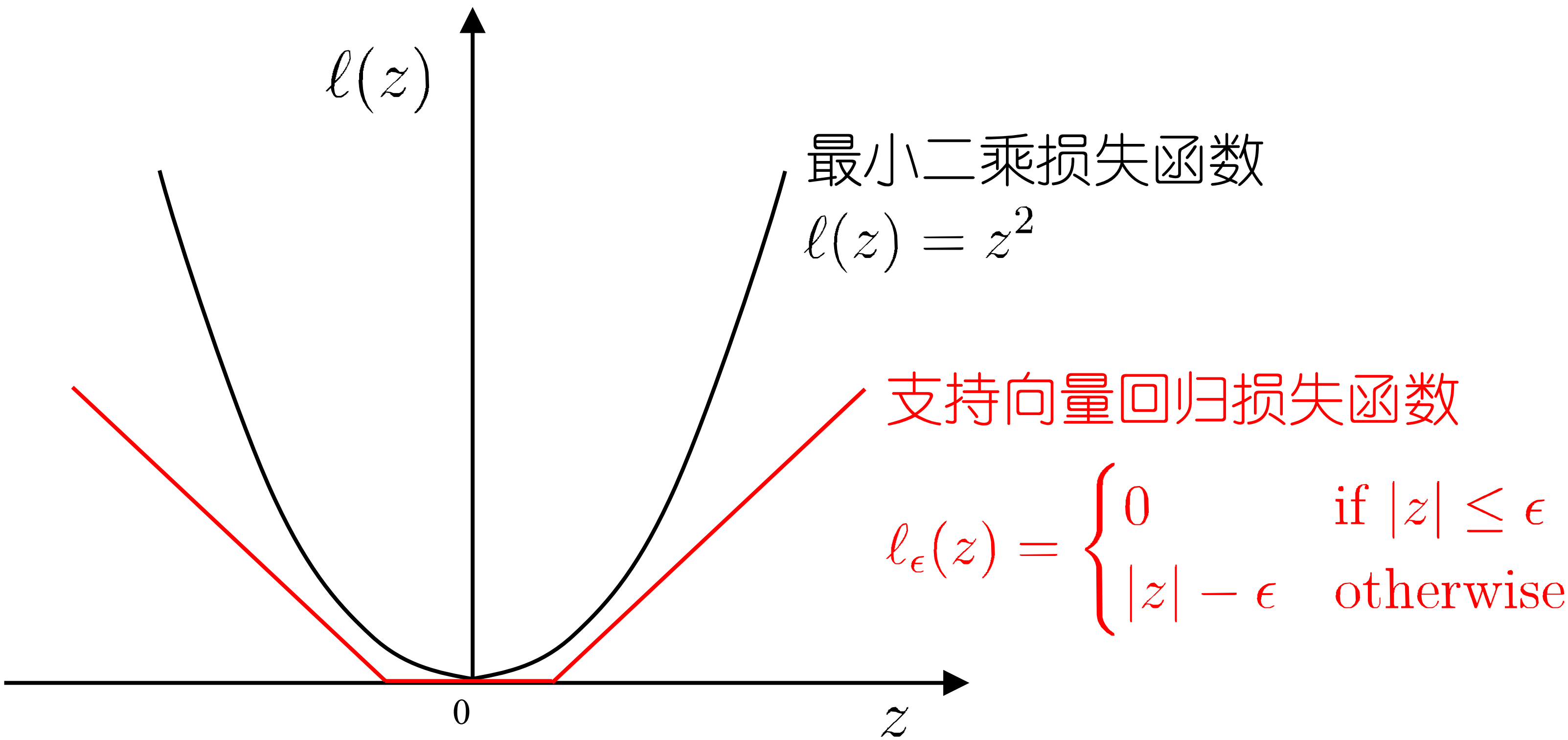
支持向量回归

特点：允许模型输出和实际输出间存在 2ϵ 的偏差.



损失函数

落入中间 2ϵ 间隔带的样本不计算损失，从而使得模型获得稀疏性.



形式化

原始问题

$$
\begin{aligned}
\min_{\boldsymbol{w},b,\xi_i,\hat{\xi}_i} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \hat{\xi}_i) \\
\text{s.t.} \quad & y_i - \boldsymbol{w}^\top\phi(\boldsymbol{x}_i) - b \leq \epsilon + \xi_i, \\
& y_i - \boldsymbol{w}^\top\phi(\boldsymbol{x}_i) - b \geq -\epsilon - \hat{\xi}_i, \\
& \xi_i \geq 0,\ \hat{\xi}_i \geq 0,\ i = 1, 2, \ldots, m.
\end{aligned}
$$

对偶问题

$$
\begin{aligned}
\min_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j)\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{m}(\alpha_i(\epsilon - y_i) + \hat{\alpha}_i(\epsilon + y_i)) \\
\text{s.t.} \quad & \sum_{i=1}^{m}(\alpha_i - \hat{\alpha}_i) = 0, \\
& 0 \leq \alpha_i \leq C,\ 0 \leq \hat{\alpha}_i \leq C.
\end{aligned}
$$

预测

$$
f(\boldsymbol{x}) = \boldsymbol{w}^\top\phi(\boldsymbol{x}) + b = \sum_{i=1}^{m}(\hat{\alpha}_i - \alpha_i)y_i\kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b
$$

大纲

- 间隔与支持向量
- 对偶问题
- 核函数
- 软间隔与正则化
- 支持向量回归
- 核方法

表示定理

支持向量机 $$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

支持向量回归 $$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i) y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

结论：无论是支持向量机还是支持向量回归，学得的模型总可以表示成核函数的线性组合.

更一般的结论(表示定理)：对于任意单调增函数Ω和任意非负损失函数l，优化问题

$$\min_{h \in \mathbb{H}} \quad F(h) = \Omega(\|h\|_{\mathbb{H}}) + l(h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_m))$$

的解总可以写为 $h^* = \sum_{i=1}^{m} \alpha_i \kappa(\cdot, \boldsymbol{x}_i)$.

核线性判别分析

- 通过表示定理可以得到很多线性模型的"核化"版本
  - 核SVM
  - 核LDA
  - 核PCA
  - ......
- 核LDA：先将样本映射到高维特征空间，然后在此特征空间中做线性判别分析

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_b^\phi \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_w^\phi \boldsymbol{w}}$$

$$\Big\downarrow \quad h(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x})$$

$$\max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^\top \boldsymbol{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top \boldsymbol{N} \boldsymbol{\alpha}}$$

Take Home Message

- 支持向量机的"最大间隔"思想
- 对偶问题及其解的稀疏性
- 通过向高维空间映射解决线性不可分的问题
- 引入"软间隔"缓解特征空间中线性不可分的问题
- 将支持向量的思想应用到回归问题上得到支持向量回归
- 将核方法推广到其他学习模型

成熟的SVM软件包

- LIBSVM
  http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- LIBLINEAR
  http://www.csie.ntu.edu.tw/~cjlin/liblinear/
- SVM^{light}、SVM^{perf}、SVM^{struct}
  http://svmlight.joachims.org/svm_struct.html
- Pegasos
  http://www.cs.huji.ac.il/~shais/code/index.html