

# Machine

## Learning - P46

支持向量机

对于训练集  $T = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$

其中输入空间为  $D$  维欧几里得空间  $X \subseteq \mathbb{R}^D$ , 输出空间为二分类标记  $\{-1, 1\}$

目标是找到超平面  $\vec{w}^\top \vec{x} + b = 0$  划分两个分类

其中  $\vec{w}$  为  $D$  维列向量  $\vec{w} \in \mathbb{R}^D$ ,  $b$  为实数参数

且有  $\min_{\vec{w}, b} \frac{1}{2} \vec{w}^\top \vec{w}$ , s.t.  $y_i (\vec{w}^\top \vec{x}_i + b) \geq 1$

原始问题是等价于  $\min_{\vec{w}, b} (\max_{y_i \geq 0} L(\vec{w}, b, \vec{x}))$

其中拉格朗日函数  $L(\vec{w}, b, \vec{x}) = \frac{1}{2} \vec{w}^\top \vec{w} + \sum_{i=1}^N \alpha_i [1 - y_i (\vec{w}^\top \vec{x}_i + b)]$

$\vec{x} \in \mathbb{R}^N$ , 且  $\forall i \alpha_i \geq 0$

有拉格朗日对偶问题  $\max_{\alpha_i \geq 0} (\min_{\vec{w}, b} L(\vec{w}, b, \vec{x}))$

KKT 条件 (Karush - Kuhn - Tucker condition),

非线性规划 (nonlinear programming) 最优解的必要条件

将拉格朗日乘子法 (Lagrange multiple) 的等式约束优化问题推广至不等式约束

对于多个约束等式和约束不等式的情况

$\min f(\vec{x}), \text{s.t. } g_j(\vec{x}) = 0, h_k(\vec{x}) \leq 0, j=1, 2, \dots, M, k=1, 2, \dots, P$

则有拉格朗日函数  $L(\vec{x}, \lambda, \mu) = f(\vec{x}) + \sum_{j=1}^M \lambda_j g_j(\vec{x}) + \sum_{k=1}^P \mu_k h_k(\vec{x})$

其中  $\lambda_j$  为对应  $g_j(\vec{x}) = 0$  的 Lagrange 乘数,  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_M)^\top$

$\mu_k$  为对应  $h_k(\vec{x}) \leq 0$  的 Lagrange 乘数,  $\mu = (\mu_1, \mu_2, \dots, \mu_P)^\top$

则有约束条件  $\nabla_{\vec{x}} L(\vec{x}, \lambda, \mu) = 0$

$\nabla_{\vec{x}} L(\vec{x}, \lambda, \mu) = 0, \quad j=1, 2, \dots, M$

$\nabla_{\vec{x}} L(\vec{x}, \lambda, \mu) = 0, \quad k=1, 2, \dots, P$

注意当优化问题是凸优化问题, 且满足 KTT 条件时,

拉格朗日对偶问题是等价于原问题是

对于 SVM 则有  $\max_{\alpha_i \geq 0} (\min_{\vec{w}, b} L(\vec{w}, b, \vec{x})) = \min_{\vec{w}, b} (\max_{\alpha_i \geq 0} L(\vec{w}, b, \vec{x}))$

于是首先固定  $\vec{x}$  并对  $\vec{w}, b$  求偏导数, 并令偏导数为 0

$$\frac{\partial}{\partial \vec{w}} L(\vec{w}, b, \vec{x}) = \vec{w} - \sum_{i=1}^N \alpha_i y_i \vec{x}_i = 0$$

$$\frac{\partial}{\partial b} L(\vec{w}, b, \vec{x}) = - \sum_{i=1}^N \alpha_i y_i = 0$$

于是有  $\vec{w} = \sum_{i=1}^N \alpha_i y_i \vec{x}_i, \sum_{i=1}^N \alpha_i y_i = 0$

$$\begin{aligned} L(\vec{w}, b, \vec{x}) &= \frac{1}{2} \vec{w}^\top \vec{w} - \vec{w}^\top \sum_{i=1}^N \alpha_i y_i \vec{x}_i - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i \\ &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \vec{x}_i^\top \vec{x}_j \end{aligned}$$

转化为最优化问题  $\max_{\alpha_i \geq 0} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \vec{x}_i^\top \vec{x}_j$

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

可以使用序列最小优化算法 (sequential minimal optimization, SMO) 求解

# Machine

## Learning - P47

支持向量机

对于训练集  $T = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$

其中输入空间为  $D$  维欧几里得空间，即  $\vec{x} \in \mathbb{R}^D$

输出空间为二分类标记  $\{-1, 1\}$

空间  $\vec{x}$  为对输入进行非线性变换， $\vec{x}$  空间为  $d$  维欧几里得空间

目标是用线性SVM找到  $\vec{x}$  空间的超平面  $\vec{w}^T \vec{x} + b = 0$  对样本进行划分

则  $\vec{x}$  空间中的线性分类对应于原来  $\vec{x}$  空间中的非线性分类

有最优化问题  $\max_{\vec{w}, b} \sum_{i=1}^N d_i - \frac{1}{2} \sum_{i=1}^N d_i d_j y_i y_j \vec{x}_i^T \vec{x}_j$

s.t.  $\vec{w}^T \vec{x}_i + b \geq 1, \sum_{i=1}^N d_i y_i = 0$

取  $N$  维列向量  $\vec{y} = (y_1, y_2, \dots, y_N)^T$ ,  $\vec{1} = (1, 1, \dots, 1)^T$

$N \times N$  矩阵  $Q$ , 其中  $q_{ij} = y_i y_j \vec{x}_i^T \vec{x}_j$

于是最优化问题转换为  $\min_{\vec{w}} \frac{1}{2} \vec{w}^T Q \vec{w} - \vec{1}^T \vec{w}$

s.t.  $\vec{w}^T \vec{1} = 0, \vec{w}^T \vec{x}_i \geq 1$

注意在这个过程中  $\vec{x}_i^T \vec{x}_j$  的运算依旧是依赖于  $\vec{x}$  空间的维数  $d$  的

而注意从原问题转换为拉格朗日对偶问题即为了摆脱对维数  $d$  的依赖

从  $\vec{x}$  空间看,  $\vec{x}_i^T \vec{x}_j$  是两个  $d$  维向量的内积计算

但从  $\vec{x}$  空间看, 首先通过  $\phi$  变换, 即  $\phi: \vec{x} \rightarrow \vec{x}$ , 进行非线性变换  $\vec{x}_i = \phi(\vec{x}_i)$

再进行内积的计算  $\vec{x}_i^T \vec{x}_j = \phi(\vec{x}_i)^T \phi(\vec{x}_j)$

kernel trick, 即将“ $\phi$  变换”和“内积”两个步骤聚合并为一个步骤

从而将  $O(d^2)$  的时间复杂度降低到  $O(D)$ , 由于非线性变换中通常有维数  $d \gg D$

对于某个给定的非线性变换  $\phi: \vec{x} \rightarrow \vec{x}$

有对应的核函数 (kernel function)  $K(\vec{x}, \vec{x}')$

使得 transform  $\phi \leftrightarrow$  kernel function:  $K_\phi(\vec{x}, \vec{x}') = \phi(\vec{x})^T \phi(\vec{x}')$

于是矩阵  $Q$  中的元素  $q_{ij} = y_i y_j \vec{x}_i^T \vec{x}_j = y_i y_j K(\vec{x}_i, \vec{x}_j)$

Kernel trick 核心是通过一个  $\vec{x}$  空间的高效 kernel function 计算

来映射到经过特征转换到  $\vec{x}$  空间的向量 内积结果

由于 kernel function 的计算在  $\vec{x}$  空间中, 从而摆脱对  $\vec{x}$  空间维数  $d$  的依赖

kernel hard-margin SVM algorithm

$Q = (q_{ij}) = (y_i y_j K(\vec{x}_i, \vec{x}_j))$ ,  $\vec{p} = -\vec{1}_N$ ,  $(A, \vec{c})$  for equal/bound constraint

$\vec{z} \leftarrow QP(Q, \vec{p}, A, \vec{c})$

$b \leftarrow (y_s - \sum_{i \in SV} \alpha_i y_i K(\vec{x}_i, \vec{x}_s))$  with  $SV(\vec{x}_s, y_s)$

return SVs and  $\alpha_i$  as well as  $b$  for new  $\vec{x}$

$y_{SVM}(\vec{x}) = \text{Sign}(\sum_{i \in SV} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b)$

# Machine

## Learning - P48

贝叶斯决策理论 (Bayesian decision theory) 假设在一个博弈中

存在一个自然选择的状态 / 参数 / 标签  $y \in Y$

nature pick state / parameter / label

并且生成了一次观察  $x \in X$  (generate an observation)

需要从行动空间中选择一个行动  $a \in A$

choose action from action space

对应于自然选择的  $y$  和主动选择的  $a$ , 会产生损失 (loss)  $L(y, a)$

用于衡量行动  $a$  与状态  $y$  之间的兼容性

measure how compatible action  $a$  with nature's hidden state  $y$

如 错误分类 (misclassification)  $L(y, a) = I(y \neq a)$ ,  $y, a \in \{1, 2, \dots, C\}$

平方误差 (squared loss)  $L(y, a) = (y - a)^2$ ,  $y, a \in R$

目标是得到一个决策过程 (decision procedure / policy),  $\delta: X \rightarrow A$

对每个可能的观察输入指派一个最优的行动选择

specify optimal action for each possible input

即最小化期望损失 (minimize expected loss) 的行动

$$\delta(x) = \operatorname{argmin}_{a \in A} E[L(y, a)]$$

在经济学中, 定义效用函数 (utility function) 为负损失 (negative loss)

$$\text{即 } U(y, a) = -L(y, a)$$

$$\text{则有 } \delta(x) = \operatorname{argmax}_{a \in A} E[U(y, a)]$$

称为最大期望效用原则 (maximum expected utility principle)

也称为理性行为 (rational behavior)

与最大后验概率类似, 对于给定的观察  $x$ , 指派  $(a)$  其后

最优选择为最小化后验期望损失 (posterior expected loss) 的行动

$$\text{即有 } p(a|x) = E_{p(y|x)}[L(y, a)] = \sum_y L(y, a) p(y|x)$$

$$\delta(x) = \operatorname{argmin}_{a \in A} p(a|x) = \operatorname{argmin}_{a \in A} \sum_y L(y, a) p(y|x)$$

称为贝叶斯预测 (Bayes estimator) 或贝叶斯决策规则 (Bayes decision rule)

如对于 0-1 损失 (0-1 loss)

$$L(y, a) = I(y \neq a) = \begin{cases} 0, & a=y \\ 1, & a \neq y \end{cases}$$

	$\hat{y} = 1$	$\hat{y} = 0$
$y = 1$	0	1
$y = 0$	1	0

$$\text{则有 } p(a|x) = p(y=a|x) \cdot 0 + p(y \neq a|x) \cdot 1 = 1 - p(y|x)$$

则  $\delta(x) = \operatorname{argmin}_{a \in A} p(a|x)$  等价于 最大后验概率估计

$$\text{即 } \delta(x) = \operatorname{argmin}_{a \in A} p(a|x) = \operatorname{argmax}_{a \in A} p(y=a|x)$$

# Machine

## Learning - P49

ICF - TFA.87

贝叶斯决策理论，对于一个博弈中的自然选择的隐藏状态  $y \in Y$ ，从行动空间中选择行动  $a \in A$ ，从而最大化后验期望损失， $\delta(x) = \operatorname{argmax}_{a \in A} E_{p(y|x)}[L(y, a)]$

考虑在分类问题中所有的  $p(y|x)$  都不足有多大时

存在行动为拒绝所有分类 (reject action)

适用于经济学或医学中风险厌恶 (risk averse) 的情况

如对于  $y \in \{1, 2, \dots, C\}$ ,  $a \in \{1, 2, \dots, C+1\}$ , 其中  $a=C+1$  表示拒绝

则有损失函数为

$y=a, y \in \{1, 2, \dots, C\}$	$a=1$	$a=2$	$\dots$	$a=C$
$\lambda_s$	$0$	$\lambda_s$	$\dots$	$\lambda_s$
$\lambda_r$	$a=c+1$	$\lambda_s$	$\dots$	$\lambda_s$

classification       $a=C$        $\lambda_s$        $\lambda_s$        $\dots$        $0$

reject       $a=c+1$        $\lambda_r$        $\lambda_r$        $\dots$        $\lambda_r$

于是有  $P(a \neq c+1|x) = \sum_{y \neq a} P(y|x) \cdot \lambda_s = [1 - p(y=a|x)] \lambda_s$

$P(a = c+1|x) = P(y=p(y|x)) \cdot \lambda_r = \lambda_r$

假如选择  $a=k$ ,  $k \in \{1, 2, \dots, C\}$ , 即  $\delta(x) = k$

则必须满足  $\forall c \in \{1, 2, \dots, C\} [1 - p(y=c|x)] \lambda_s \geq [1 - p(y=k|x)] \lambda_s$

即  $\forall c P(y=k|x) \geq P(y=c|x)$

且  $[1 - p(y=k|x)] \lambda_s \leq \lambda_r$ , 即  $p(y=k|x) \geq 1 - \frac{\lambda_r}{\lambda_s}$

即有  $\forall c P(y=k|x) \geq P(y=c|x) \wedge p(y=k|x) \geq 1 - \frac{\lambda_r}{\lambda_s}$

考虑  $\frac{\lambda_r}{\lambda_s}$  的取值变化对最优选择的影响

当  $\frac{\lambda_r}{\lambda_s} = 0$  时, 即拒绝的损失  $\lambda_r = 0$

于是选择  $a=k$  需要满足  $p(y=k|x) \geq 1 - \frac{\lambda_r}{\lambda_s} = 1$

即仅当  $p(y=k|x) = 1$  时, 选择  $a=k$ , 否则拒绝所有分类

当  $\frac{\lambda_r}{\lambda_s} \geq 1$  时, 即拒绝的损失  $\lambda_r \geq \lambda_s$

于是选择  $a=k$  时,  $p(y=k|x) \geq 1 - \frac{\lambda_r}{\lambda_s}$ , 其中  $1 - \frac{\lambda_r}{\lambda_s} \leq 0$

则实际上等价于最大后验概率估计, 即始终不选择拒绝

即有  $\forall c P(y=k|x) \geq P(y=c|x)$ ,  $\delta(x) = \operatorname{argmax}_{a \in \{1, 2, \dots, C\}} P(y=a|x)$

当  $\frac{\lambda_r}{\lambda_s}$  从 0 增加到 1 时,  $0 < 1 - \frac{\lambda_r}{\lambda_s} < 1$

则当  $\operatorname{argmax}_{a \in \{1, 2, \dots, C\}} P(y=a|x) \geq 1 - \frac{\lambda_r}{\lambda_s}$  时, 选择  $y=a$

否则选择拒绝所有分类, 即选择  $a=c+1$

当  $\frac{\lambda_r}{\lambda_s}$  从 0 增加到 1 时, 选择拒绝的可能越来越低

# Machine

## Learning - P50

009 - second

贝叶斯决策理论 false positive vs. false negative trade off

对于二元决策问题 (binary decision problem)

即有自然选择的隐藏状态为二分类标记集合 {0, 1}

而行动空间一一对应于对隐藏状态的预测  $\hat{y} \in \{0, 1\}$

则存在两类错误:

当  $\hat{y}=1$  但  $y=0$  时, 为第一类错误 (type I error)

称为 false positive (FP), 或称 false alarm

当  $\hat{y}=0$  但  $y=1$  时, 为第二类错误 (type II error)

称为 false negative (FN), 或称 missed detection

于是有  $P(\hat{y}=0|\vec{x}) = L_{FN} P(y=1|\vec{x})$

$P(\hat{y}=1|\vec{x}) = L_{FP} P(y=0|\vec{x}) = L_{FP}(1 - P(y=1|\vec{x}))$

则选择  $\hat{y}=1$  当且仅当  $P(\hat{y}=1|\vec{x}) < P(\hat{y}=0|\vec{x})$

再取  $L_{FN} = c L_{FP}$ , 其中  $c$  为实数常数

则有  $c L_{FP} P(y=1|\vec{x}) > L_{FP}(1 - P(y=1|\vec{x}))$

$P(y=1|\vec{x}) > \frac{1}{1+c}$

再取  $\gamma = \frac{1}{1+c}$ , 称为阈值 (threshold)

则有选择  $\hat{y}=1$  当且仅当  $P(y=1|\vec{x}) > \gamma$

confusion matrix (混淆矩阵), for given threshold  $\gamma$  and apply decision rule

count number of true positive (TP), FP, true negative (TN), FN

estimate \ truth	$y=1$	$y=0$	
$\hat{y}=1$	TP	FP	$\hat{N}_+ = TP + FP$
$\hat{y}=0$	FN	TN	$\hat{N}_- = FN + TN$
$\Sigma$	$N_+ = TP + FN$	$N_- = FP + TN$	$N = TP + FP + FN + TN$

则有不同的评价指标:

accuracy (准确率):  $acc = \frac{TP + TN}{TP + FP + FN + TN} = \frac{TP + TN}{N}$

error rate (错误率):  $err = 1 - acc = \frac{FP + FN}{TP + FP + FN + TN} = \frac{FP + FN}{N}$

precision (精确度):  $precision = \frac{TP}{TP + FP} = \frac{TP}{\hat{N}_+}$

true positive rate (TPR), 也称灵敏度 (sensitivity), 召回率 (recall), hit rate

$TPR = \frac{TP}{TP + FN} = TP/N_+ \approx P(\hat{y}=1|y=1)$

specificity (特异度):  $specificity (CTNR) = \frac{TN}{FP + TN} = TN/N_-$

miss rate (type II error rate):  $FNR = \frac{FN}{TP + FN} = FN/N_+$

false positive rate (FPR), 也称 false alarm rate, type I error rate

$FPR = \frac{FP}{FP + TN} = FP/N_- \approx P(\hat{y}=1|y=0)$

# Machine

## Learning - P51

贝叶斯决策理论，对于二元决策问题，有自然选择的隐藏状态  $y \in \{0, 1\}$

行动空间一一对应于对隐藏状态的预测  $\hat{y} \in \{0, 1\}$

则有 confusion matrix of estimating  $P(\hat{y}|y)$

estimate \ truth

$y=1$

$y=0$

$\hat{y}=1$

$TP/N_+ = TPR$  (sensitivity/recall)

$FP/N_- = FPR$  (type I)

$\hat{y}=0$

$FN/N_+ = FNR$  (miss rate/type II)

$TN/N_- = TNR$  (specificity)

其中

$N_+ = TP + FN$

$N_- = FP + TN$

ROC 曲线 (receiver operating characteristic curve)

对于给定的阈值  $\gamma \in [0, 1]$ , 可以计算 TPR 和 FPR

则可以画出 TPR vs FPR 的曲线

plot TPR vs FPR as implicit function of  $\gamma$

当  $\gamma=1$  时, 则始终选择  $\hat{y}=0$

即有  $TP=FP=0$ , 于是  $TPR=FPR=0$

当  $\gamma=0$  时, 则始终选择  $\hat{y}=1$  (几乎)

即有  $TN=FN=0$ , 于是  $TPR=FPR=1$

则 ROC 曲线从左下  $(0, 0)$  到右上  $(1, 1)$  描述为阈值从 1 降至 0 的过程

system perfectly separate positive from negative

has one threshold can achieve top left corner ( $FPR=0, TPR=1$ )

面积量度在 ROC 曲线的评价标准为 AUC (area under the curve)

AUC 越高越好, 且上限为 1

EER (equal error rate, cross over rate), value satisfy  $FNR=FPR=1-TPR$

EER 越低越好, 且下限为 0

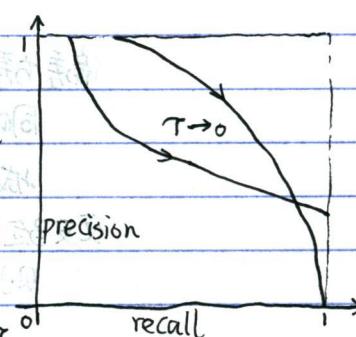
F-score

对于非负实数  $\beta$ , 有  $F_\beta = \frac{(1+\beta^2) \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$ , 为分类问题的衡量指标

其中  $\beta$  可以描述为召回率 (recall) 相对于精确率 (precision) 的重要度

当  $\beta=0$  时,  $F_\beta = \text{precision}$ , 当  $\beta \rightarrow \infty$  时,  $F_\beta = \text{recall}$

当  $\beta=1$  时, 称为 F1-score, 有  $F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$



PR 曲线 (precision-recall curve), plot precision vs recall as vary threshold  $\gamma$

则有  $\text{precision} = TP/\hat{N}_+ = P(y=1|\hat{y}=1) = \sum_i y_i \hat{y}_i / \sum_i \hat{y}_i$

$\text{recall} = TP/N_+ = P(\hat{y}=1|y=1) = \sum_i y_i \hat{y}_i / \sum_i y_i$

当数据极不平衡 (positive 样本较少) 时, PR 曲线可能优于 ROC 曲线

# Machine

## Learning - P52

偏差方差权衡 (bias-variance tradeoff), 对于训练集  $T = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$

其中输入空间为  $P$  维欧几里得空间  $\vec{x} \in R^P$ , 输出空间为实数  $y \in R$

假设存在函数关系  $y = f(\vec{x}) + \varepsilon$ , 其中函数  $f: R^P \rightarrow R$

其中噪音 (noise)  $\varepsilon \sim N(0, \sigma^2)$

目标是找到函数  $\hat{f}(\vec{x})$  来尽可能好地拟合  $y = f(x) + \varepsilon$

评价标准为均方误差 (mean square error, MSE):  $E[(y - \hat{f}(\vec{x}))^2]$

对均方误差进行偏差方差分解 (bias-variance decomposition)

又由于  $f(\vec{x})$  是确定的 (deterministic), 则  $E[f] = f$

且有  $E[y] = E[f + \varepsilon] = E[f] + E[\varepsilon] = f$

$$\text{Var}[y] = E[(y - E[y])^2] = E[(f + \varepsilon - f)^2] = E[\varepsilon^2] = \text{Var}[\varepsilon] + E[\varepsilon]^2 = \sigma^2$$

又函数估计  $\hat{f}(\vec{x})$  与噪音  $\varepsilon$  是独立的

$$\begin{aligned} \text{则 } E[(y - \hat{f})^2] &= E[(f + \varepsilon - \hat{f})^2] = E[(f + \varepsilon - \hat{f} + E[\hat{f}] - E[\hat{f}])^2] \\ &= E[(f - E[\hat{f}])^2] + E[\varepsilon^2] + E[(E[\hat{f}] - \hat{f})^2] \\ &\quad + 2E[(f - E[\hat{f}])\varepsilon] + 2E[\varepsilon(E[\hat{f}] - \hat{f})] + 2E[(E[\hat{f}] - \hat{f})(f - E[\hat{f}])] \end{aligned}$$

$$\text{又 } E[(f - E[\hat{f}])\varepsilon] = (f - E[\hat{f}])E[\varepsilon] = 0$$

$$E[\varepsilon(E[\hat{f}] - \hat{f})] = E[\varepsilon]E[E[\hat{f}] - \hat{f}] = 0$$

$$E[(E[\hat{f}] - \hat{f})(f - E[\hat{f}])] = (E[\hat{f}] - \hat{f})(f - E[\hat{f}]) = 0$$

$$\text{于是 } E[(y - \hat{f})^2] = E[(f - E[\hat{f}])^2] + E[\varepsilon^2] + E[(\hat{f} - E[\hat{f}])^2]$$

$$= (f - E[\hat{f}])^2 + \sigma^2 + \text{Var}[\hat{f}], \text{ 令 } \text{Bias}[\hat{f}] = (f - E[\hat{f}])^2$$

于是有三部分: 偏差  $\text{Bias}[\hat{f}]$ : error from erroneous assumption in learning algorithm

(欠拟合) high bias cause algorithm to miss relevant relation between feature and output

方差  $\text{Var}[\hat{f}]$ : error from sensitivity to small fluctuation in training set

(过拟合) high variance cause to model random noise rather than intended output

不可消除的误差 (irreducible error)  $\sigma^2$ : 由于随机噪音

偏差方差矛盾 (bias-variance dilemma): 在监督学习过程中无法同时最小化偏差与方差

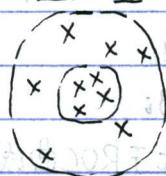
同时减少 bias, variance: 增加训练样本 / 修改模型结构 / 根据误差分析调整输入特征

减少 bias / variance: 提高 / 降低模型复杂度, 减少或去除 / 增加正则化, 一 / 提前终止

交叉验证 (cross validation): 将样本划分为  $K$  个子样本, 分别以每个子样本为 validation, 其他为 training

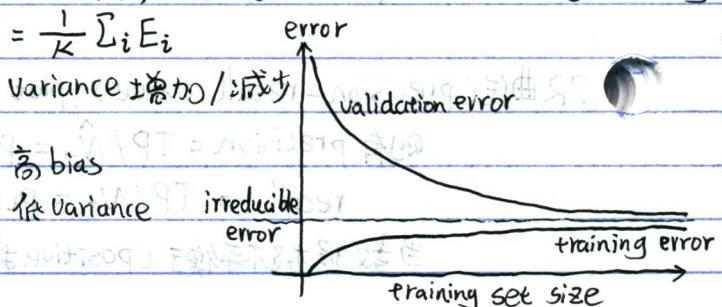
则有误差  $E_1, E_2, \dots, E_k$ , 则有误差  $E = \frac{1}{k} \sum_i E_i$

且  $K$  值大 / 小, 有 bias: 减少 / 增加, variance: 增加 / 减少



bias  
高 variance

高 variance  
bias



# Machine

## Learning - P53

正则化 (regularization), 为了解决模型中的过拟合现象, 同时保留所有的特征  
将部分不重要的特征权值变小甚至置0, 使得特征的参数矩阵变得稀疏  
从而减少测试误差, 有时候也会增加训练误差

对于权重向量  $\vec{w}$  的损失函数  $L_0(\vec{w})$ , 加入一个惩罚项/正则化项

即有  $L(\vec{w}) = L_0(\vec{w}) + \frac{\lambda}{n} \Omega(\vec{w})$ , 其中  $\lambda$  为一个常数

(ex. p=1)  $n$  为训练集的样本个数,  $\rho$  为实数常数,  $\lambda$  为超参数, 用于控制正则化程度

通常惩罚项  $\Omega(\vec{w})$  使用权重向量  $\vec{w}$  的  $L_p$  范数的类似倍数

即对于  $p > 0$ , 有  $\|\vec{w}\|_p = (\sum_i |w_i|^p)^{1/p}$

(ex. p=0) 则取  $\Omega(\vec{w}) = \sum_i |w_i|^0$  (此时  $\Omega(\vec{w})$  为零)

如果令  $0^0 = 0$ , 则可以扩展定义到  $p=0$  时,

$\Omega(\vec{w}) = \sum_i |w_i|^0 = \#\{w_i \neq 0 \mid w_i \in \vec{w}\}$ , 即非零参数的个数

常用的正则化有  $L_0$ ,  $L_1$ ,  $L_2$ ,  $L_p$  (其中  $p \neq 0, 1, 2$ )

$L_0$  正则化使用的惩罚项为  $\|\vec{w}\|_0 = \#\{i \mid w_i \neq 0\}$

$L_0$  可以找到最少最优的特征选择, 实现特征稀疏的结果, 即选择参数非零的特征  
(ex.  $\Omega(\vec{w}) = \#\{i \mid w_i \neq 0\}$ ) 但是  $L_0$  的最优化是 NP-hard 的问题

$L_1$  正则化使用的惩罚项为  $\|\vec{w}\|_1 = \sum_i |w_i|$ , 即特征权重的绝对值之和

$L_1$  通常用于代替直接优化  $L_0$ ,  $L_1$  范数为  $L_0$  的最优凸近似

$L_2$  正则化使用的惩罚项为  $\|\vec{w}\|_2 = \sqrt{\sum_i w_i^2}$

而对于其他的  $L_p$  范数, 其中  $p \in (0, +\infty)$

通常认为范数可以理解为满足非负, 自反, 三角不等式的距离 (度量空间)

但是注意在  $p \in (0, 1)$  时, 实际上并不是范数, 由于违反了三角不等式

直观的表现为  $p \in (0, 1)$  时单单位球 (unit ball) 不是凸集

(ex.  $\|\vec{w}\|_p = (\sum_i |w_i|^p)^{1/p}$ ) 即无法使用梯度下降来得到最优解

而当  $p \rightarrow +\infty$  的过程中, 会 给绝对值大的特征权重更大的惩罚, 而绝对值小的惩罚  $\rightarrow 0$

