第一部分 监督学习 / Supervised Learning

翻译&校正 | 韩信子@ShowMeAI 编辑 | 南乔@ShowMeAI 原文作者 | https://stanford.edu/~shervine 本节原文超链

[1]监督学习简介 / Introduction to Supervised Learning

给定一组数据点 $\{x^{(1)},...,x^{(m)}\}$ 及对应的输出 $\{y^{(1)},...,y^{(m)}\}$，构建一个预估器[1]，学习如何从 x 预测 y。

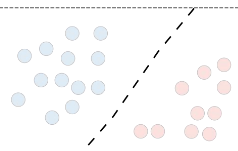
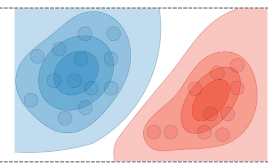
预测类型 Type of prediction

下表总结了不同类型的预测模型：

	回归	分类
输出	连续值	离散类别
例子	线性回归	Logistic 回归，SVM，朴素贝叶斯

模型类型 Type of model

下表总结了不同类型的模型：

	判别模型	生成模型
目标	直接估计 P(y|x)	估计 P(x|y)，然后推导 P(y|x)
所学内容	决策边界	数据的概率分布
例图		
示例	回归，SVMs	GDA，朴素贝叶斯

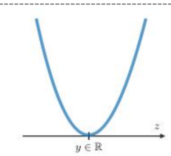
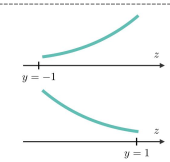
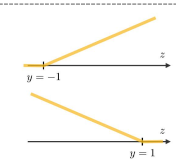
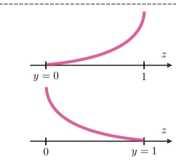
[2]数学符号和常见概念 / Notations and General Concepts

假设 Hypothesis

选定模型 h_θ 。对于给定的输入数据 $x^{(i)}$，该模型预测的输出是 $h_\theta(x^{(i)})$。

损失函数 Loss function

损失函数 $L:(z,y)\in\mathbb{R}\times Y\mapsto L(z,y)\in\mathbb{R}$，以 y [实际数据值] 和 z [预测值]为输入，输出二者之间的差异程度。下表总结了常见的损失函数：

最小二乘误差	Logistic 损失	合页损失	交叉熵
$\frac{1}{2}(y-z)^2$	$\log(1+\exp(-yz))$	$\max(0,1-yz)$	$-[y\log(z)+(1-y)\log(1-z)]$
			
线性回归	Logistic 回归	SVM	神经网络

成本函数[2] Cost function

成本函数 J 通常用于评估模型的性能。定义如下[L 为损失函数]：

$$J(\theta)=\sum_{i=1}^{m}L\left(h_\theta(x^{(i)}),\ y^{(i)}\right)$$

梯度下降 Gradient descent

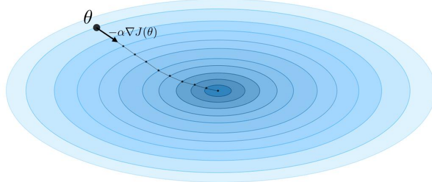
梯度下降的更新规则表示如下 [$\alpha\in R$ 为学习率，J 为成本函数]：

$$\theta\leftarrow\theta-\alpha\nabla J(\theta)$$

[1] 译者注：原文 Classifier，意为分类器。但监督学习覆盖分类回归不同的问题，因此译者将其改成了预估器（estimator）。

[2] 译者注：也常被译作 代价函数。

备注：随机梯度下降（SGD）是根据每个训练样本进行参数更新，而批量梯度下降是在一批训练样本上进行更新。



似然 Likelihood

以 θ为参数的模型 L(θ) 的似然函数，可以用于寻找使得函数最大化的最佳参数 θ:

$$\theta^{\mathrm{opt}} = \underset{\theta}{\arg\max}\ L(\theta)$$

备注：实际上，通常使用更容易优化的对数似然 $\ell(\theta) = \log(L(\theta))$ 。

牛顿算法 Newton's algorithm

牛顿算法是一种数值方法，目的是找到一个 θ，使得 $\ell'(\theta) = 0$， 更新规则如下:

$$\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$

备注：多维泛化，也称为 Newton-Raphson 方法，更新规则如下:

$$\theta \leftarrow \theta - \left(\nabla_\theta^2 \ell(\theta)\right)^{-1} \nabla_\theta \ell(\theta)$$

[3]线性模型 / Linear Models

3.1 线性回归 / Linear regression

假设 $y|x;\theta \sim \mathcal{N}(\mu, \sigma^2)$

正规方程 Normal equations

我们把设计函数记作 X，使得成本函数最小的参数θ是符合下式的闭式解:

$$\theta = (X^T X)^{-1} X^T y$$

最小均方算法 LMS algorithm

m 个数据的训练集，其最小均方（Least Mean Squares，LMS）算法的更新规则，也被称为“ Widrow- Hoff 学习规则”。形式如下[α 为学习率]:

$$\forall j, \quad \theta_j \leftarrow \theta_j + \alpha \sum_{i=1}^{m} \left[y^{(i)} - h_\theta\left(x^{(i)}\right)\right] x_j^{(i)}$$

备注：更新规则是梯度上升的特定情况。

局部加权回归 LWR

局部加权回归（Locally Weighted Regression，LWR），是线性回归的变形。通过参数 $\tau \in \mathbb{R}$ 对成本函数中每个训练样本 x 进行加权，形式如下：

$$w^{(i)}(x) = \exp\left(-\frac{\left(x^{(i)} - x\right)^2}{2\tau^2}\right)$$

3.2 分类和逻辑回归 / Classification and Logistic Regression

Sigmoid 函数 Sigmoid function

sigmoid 函数 g（也称 Logistic Function[3]），定义如下:

$$\forall z \in \mathbb{R}, \quad g(z) = \frac{1}{1 + e^{-z}} \in [0,1]$$

逻辑回归 Logistic regression

假设 $y|x;\theta \sim \mathrm{Bernoulli}(\phi)$。则 y 取值为 1 的概率$\phi$的计算公式如下:

$$\phi = p(y = 1|x;\theta) = \frac{1}{1 + \exp(-\theta^T x)} = g(\theta^T x)$$

备注：对于逻辑回归的情况，没有闭式解。

Softmax 回归 Softmax regression

Softmax 回归，也称作多分类逻辑回归，是逻辑回归在大于 2 个类别的分类场景下的拓展。我们设 $\theta_K = 0$，则每个类 i 的概率（伯努利参数 ϕ_i）等于:

[3] 译者注：没有统一的中文翻译，可被译作“逻辑函数”或音译为“逻辑斯蒂函数”。

$$\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^{K} \exp(\theta_j^T x)}$$

3.3 广义线性模型/ Generalized Linear Models

指数族 Exponential family

如果一类分布可以用 η [自然参数，也称表标准数或链接函数]、 $T(y)$ [充分统计量]、 $a(\eta)$ [对数配分函数]来表示，那么它属于指数族，形式如下：

$$\mathrm{p}(\mathrm{y};\eta) = \mathrm{b}(\mathrm{y}) \exp\big(\eta \mathrm{T}(\mathrm{y}) - \mathrm{a}(\eta)\big)$$

备注：经常会有 $\mathrm{T(y) = y}$。此外，$\exp\big(-\mathrm{a}(\eta)\big)$ 可以看作是归一化参数，以确保概率总和为 1。

下表总结了的最常见的指数分布：

	η	$T(y)$	$a(\eta)$	$b(y)$
伯努利	$\log\left(\frac{\phi}{1-\phi}\right)$	y	$\log(1+\exp(\eta))$	1
高斯	μ	y	$\frac{\eta^2}{2}$	$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y^2}{2}\right)$
泊松	$\log(\lambda)$	y	e^η	$\frac{1}{y!}$
几何	$\log(1-\phi)$	y	$\log\left(\frac{e^\eta}{1-e^\eta}\right)$	1

广义线性模型的假设 Assumptions of GLMs

广义线性模型（ Generalized Linear Models，GLM ），是将 $x \in \mathbb{R}^{n+1}$ 预测为随机变量 y 的函数，依赖以下 3 个假设：

$$(1)\quad y|x;\theta \sim \mathrm{ExpFamily}(\eta) \qquad (2)\quad h_\theta(x) = E[y|x;\theta] \qquad (3)\quad \eta = \theta^T x$$

备注：普通最小二乘法和逻辑回归是广义线性模型的特例。

[4]支持向量机 Support Vector Machines

支持向量机的目标是找到一条线，可以最大化[决策边界和训练样本之间的最小距离]。

最优间隔分类器 Optimal margin classifier

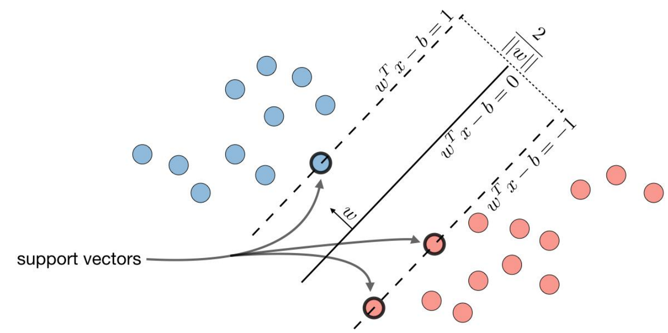
最优间隔分类器 h 定义如下：

$$h(x) = \mathrm{sign}(w^T x - b)$$

其中，$(\mathrm{w,b}) \in \mathbb{R}^n \times \mathbb{R}$ 是以下**最优化问题**的解：

$$\min \frac{1}{2}||w||^2 \qquad 使得 \quad y^{(i)}\big(w^T x^{(i)} - b\big) \geq 1$$

备注：支持向量中间的分割线定义为 $\mathrm{w^T x - b = 0}$。



合页损失 Hinge loss

SVM 中使用了合页损失，定义如下：

$$\mathrm{L(z,y)} = [1 - \mathrm{yz}]_+ = \max(0,1 - \mathrm{yz})$$

核 Kernel

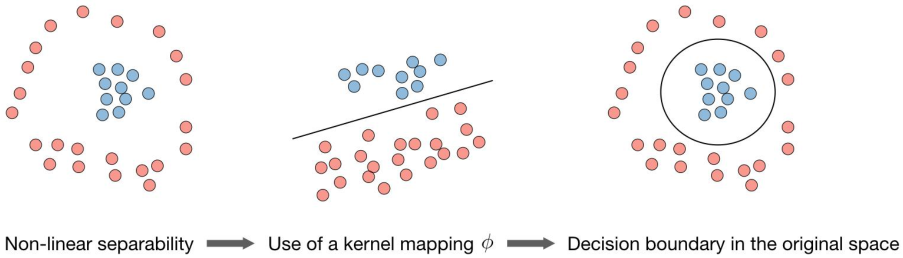
给定特征映射 ϕ，核 K 定义如下：

$$K(x,z) = \phi(x)^T \phi(z)$$

实际上，高斯核更为常用，定义如下：

$$K(x,z) = exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$$

备注：由显示映射 ϕ 计算成本函数是非常复杂的。而使用“核技巧”计算成本函数，则只需要知道 $\mathrm{K(x,z)}$的值。



拉格朗日 Lagrangian

将拉格朗日 $\mathcal{L}(w,b)$ 定义如下[β_i 为拉格朗日乘子]:

$$\mathcal{L}(w,b)=f(w)+\sum_{i=1}^{l}\beta_i h_i(w)$$

[5]生成学习 / Generative Learning

生成模型先通过预估计算 $P(x|y)$来学习数据分布，然后使用贝叶斯法则估计 $P(y|x)$。

5.1 高斯判别分析 Gaussian Discriminant Analysis

前提假设 Setting

高斯判别分析的假设如下:

$$(1)\quad y\sim\text{Bernoulli}(\phi)\quad(2)\quad x|y=0\sim\mathcal{N}(\mu_0,\Sigma)\quad(3)\quad x|y=1\sim\mathcal{N}(\mu_1,\Sigma)$$

估计 Estimation

下表总结了使得似然函数最大时的估计值:

$\hat{\phi}$	$\hat{u}_j\ (j=0,1)$	$\hat{\Sigma}$
$\frac{1}{m}\sum_{i=1}^{m}1_{\{y^{(i)}=1\}}$	$\frac{\sum_{i=1}^{m}1_{\{y^{(i)}=j\}}x^{(i)}}{\sum_{i=1}^{m}1_{\{y^{(i)}=j\}}}$	$\frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}-\mu_{y^{(i)}}\right)\left(x^{(i)}-\mu_{y^{(i)}}\right)^T$

5.2 朴素贝叶斯 Naive Bayes

假设 Assumption

朴素贝叶斯模型假设每个数据点的特征是相互独立的:

$$P(x|y)=P(x_1,x_2,\dots|y)=P(x_1|y)P(x_2|y)\dots=\prod_{i=1}^{n}P(x_i|y)$$

解 Solutions

最大化对数似然的解，形式如下【$k\in\{0,1\}$, $l\in[1,L]$】:

$$P(y=k)=\frac{1}{m}\times\#\{j|y^{(j)}=k\}$$

$$P(x_i=l|y=k)=\frac{\#\{j|y^{(j)}=k\text{ 和}x_i^{(j)}=l\}}{\#\{j|y^{(j)}=k\}}$$

备注：朴素贝叶斯广泛应用于**文本分类**和**垃圾邮件检测**。

[6]基于树模型的集成方法 / Tree-based and Ensemble Methods

适用于**回归问题**和**分类问题**。

分类回归树 CART

分类回归树（Classification and Regression Trees，CART），也称决策树，可以表示为二叉树。其优点是具备可解释性。

随机森林 Random forest

是一种基于树模型的技术，它使用大量随机选择的特征集构建决策树并集成。与决策树相反，它具备高度不可解释性，但其普遍良好的表现使其成为一种流行的算法。

备注：随机森林是一种集成方法。

提升/ Boosting

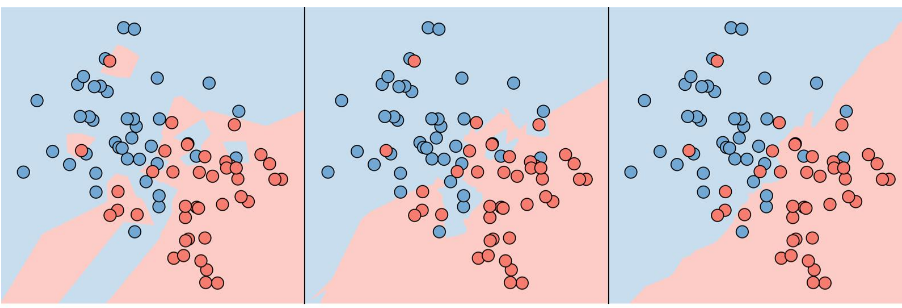
Boosting 的思路，是将几个弱学习器结合起来，形成一个更强大的学习器。见下方:

自适应增强	梯度提升
在下一轮提升步骤中，错误的样本会被置于高权重 最常见的是 Adaboost	训练弱学习器拟合残差 最常见的比如 Xgboost

[7]其他非参数方法 / Other Non-parametric Approaches

k-近邻 k-nearest neighbors

k-近邻算法（也称 k-NN），是一种非参数方法。一个预估样本的结果，是基于特征空间中 k 个最相似（即特征空间中 K 近邻）的样本的取值来确定的。适用于分类问题和回归问题[4]。



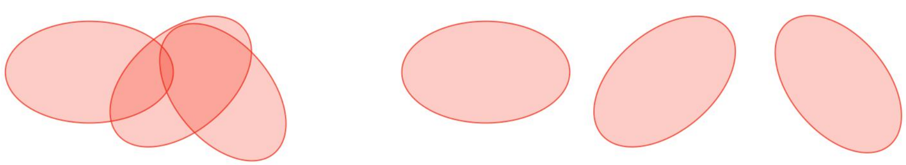
$k=1$　　$k=3$　　$k=11$

备注： k 越大，误差越大； k 越小，方差越大。

[8]学习理论 / Learning Theory

并集的上界 Union bound

$A_1,\ldots$， A_k 为 k 个事件，则$P(A_1\cup\ldots\cup A_k)\leq P(A_1)+\ldots+P(A_k)$



$A_1\cup A_2\cup A_3$　　A_1　　A_2　　A_3

[4] 译者注：在分类问题中取 k 近邻中最多的类别，在回归问题中取 k 近邻的取值均值。

霍夫丁不等式 Hoeffding's inequality

$Z_1,\ldots,Z_m$ 是 m 个独立同分布变量，取自参数 ϕ 的伯努利分布。设 $\widehat{\phi}$ 为样本均值，固定$\gamma>0$，则有：

$$P(|\phi-\widehat{\phi}|>\gamma)\leq 2\exp(-2\gamma^2 m)$$

备注：这个不等式也被称为**切诺夫界**（Chernoff bound）。

训练误差 Training error

给定分类器 h，定义 $\hat{\epsilon}(h)$为训练误差（也称经验风险或经验误差），形式如下：

$$\hat{\epsilon}(h)=\frac{1}{m}\sum_{i=1}^{m}1_{\{h(x^{(i)})\neq y^{(i)}\}}$$

概率近似正确 PAC

在概率近似正确（Probably Approximately Correct， PAC）的框架下，许多学习理论的成果得以证明。PAC 具有以下假设：

1) 训练集和测试集遵循相同的分布

2) 训练样本是相互独立的

打散 Shattering

$S=\{x^{(1)},\ldots,x^{(d)}\}$ 为集合，$\mathcal{H}$ 为一组分类器。如果任意一组标签 $\{y^{(1)},\ldots,y^{(d)}\}$ 都能满足以下条件，则称$\mathcal{H}$打散 S：

$$\exists h\in\mathcal{H},\forall i\in[\![1,d]\!],\quad h(x^{(i)})=y^{(i)}$$

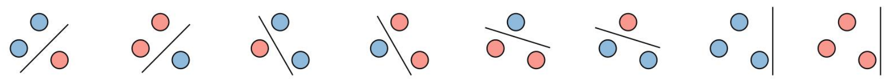
上限定理 Upper bound theorem

$\mathcal{H}$ 是有限假设类且 $|\mathcal{H}|=k$， δ 、样本大小 m 均为固定值。在概率至少为 $1-\delta$的情况下，则有：

$$\epsilon(\hat{h})\leq\left(\min_{h\in\mathcal{H}}\epsilon(h)\right)+2\sqrt{\frac{1}{2m}\log\left(\frac{2k}{\delta}\right)}$$

VC 维[VC dimension]

$\mathcal{H}$ 为无限假设类， VC($\mathcal{H}$) 为其 **VC 维**（ **Vapnik-Chervonenkis dimension**， **VC dimension**）， 注意 VC($\mathcal{H}$)是由 $\mathcal{H}$打散的最大集合。



备注：$\mathcal{H} = \{2\ 维线性分类器集\}$，其 VC 维数为 3。

定理 Theorem (Vapnik)

设 $\mathcal{H}$ 且 VC($\mathcal{H}$) = d， m为训练样本数。在概率至少为 $1-\delta$ 的情况下，有：

$$\epsilon(\hat{h}) \leq \left(\min_{h\in\mathcal{H}} \epsilon(h)\right) + O\left(\sqrt{\frac{d}{m}\log\left(\frac{m}{d}\right) + \frac{1}{m}\log\left(\frac{1}{\delta}\right)}\right)$$

Awesome AI Courses Notes Cheat Sheets

Machine Learning CS229 Stanford	Deep Learning CS230 Stanford	Natural Language Processing CS224n Stanford	Computer Vision CS231n Stanford	Deep Reinforcement Learning CS285 UC Berkeley	Neural Networks for NLP CS11-747 CMU	DL for Self-Driving Cars 6.S094 MIT	... ...

是 ShowMeAI 资料库的分支系列，覆盖最具知名度的 TOP20+门 AI 课程，旨在为读者和学习者提供一整套高品质中文速查表，可以点击【这里】查看。

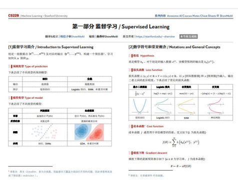
斯坦福大学（Stanford University）的 Machine Learning（CS229）和 Deep Learning（CS230）课程，是本系列的第一批产出。

本批两门课程的速查表由斯坦福大学计算机专业学生 Shervine Amidi 总结整理。原速查表为英文，可点击【这里】查看，ShowMeAI 对内容进行了翻译、校对与编辑排版，整理为当前的中文版本。

有任何建议和反馈，也欢迎通过下方渠道和我们联络 (*￣3￣)

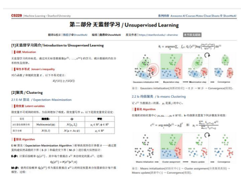
CS229 | Machine Learning @ Stanford University

监督学习
Supervised Learning



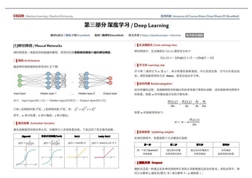
中文速查表链接

无监督学习
Unsupervised Learning



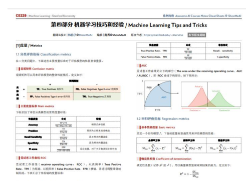
中文速查表链接

深度学习
Deep Learning



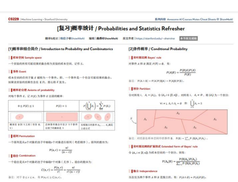
中文速查表链接

机器学习技巧和经验
Tips and Tricks



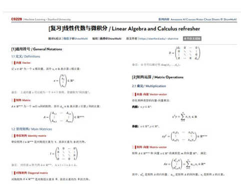
中文速查表链接

概率统计
Probabilities /Statistics



中文速查表链接

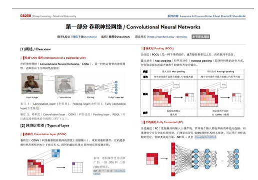
线性代数与微积分
Linear Algebra and Calculus



中文速查表链接

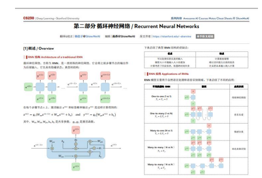
CS230 | Deep Learning @ Stanford University

卷积神经网络
CNN



中文速查表链接

循环神经网络
RNN



中文速查表链接

深度学习技巧与建议
Tips and Tricks

中文速查表链接

GitHub

ShowMeAI

https://github.com/ShowMeAI-Hub/

ShowMeAI 研究中心

扫码回复"速查表"

下载最新全套资料