

# VIP Refresher: Linear Algebra and Calculus

Afshine AMIDI and Shervine AMIDI

October 13, 2018

翻译: 朱小虎

## 通用符号

□ **向量** – 我们记为一个  $n$  维的向量, 其中  $x_i \in \mathbb{R}$  是第  $i$  维的元素:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

□ **矩阵** – 我们记  $A \in \mathbb{R}^{m \times n}$  为一个  $m$  行  $n$  列的矩阵, 其中  $A_{i,j} \in \mathbb{R}$  是第  $i$  行  $j$  列的元素:

$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

注意: 如上定义的向量  $x$  可以被看做是一个  $n \times 1$  的矩阵, 常被称为一个列向量。

□ **单位矩阵** – 单位矩阵  $I \in \mathbb{R}^{n \times n}$  是一个方阵其对角线上均是1 其余位置均为0。

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

注: 对所有矩阵  $A \in \mathbb{R}^{n \times n}$ , 我们有  $A \times I = I \times A = A$ 。

□ **对角阵** – 对角阵  $D \in \mathbb{R}^{n \times n}$  是一个方阵其对角线上元素均是非零值, 其余位置均为0。

$$D = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & d_n \end{pmatrix}$$

注: 我们记  $D$  为  $\text{diag}(d_1, \dots, d_n)$ 。

## 矩阵运算

□ **向量-向量** – 存在两种类型的向量-向量乘积:

- 内积: 对  $x, y \in \mathbb{R}^n$ , 我们有:

$$x^T y = \sum_{i=1}^n x_i y_i \in \mathbb{R}$$

- 外积: 对  $x \in \mathbb{R}^m, y \in \mathbb{R}^n$ , 我们有:

$$xy^T = \begin{pmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & & \vdots \\ x_m y_1 & \cdots & x_m y_n \end{pmatrix} \in \mathbb{R}^{m \times n}$$

□ **矩阵-向量** – 矩阵  $A \in \mathbb{R}^{m \times n}$  和向量  $x \in \mathbb{R}^n$  的乘积是一个大小为  $\mathbb{R}^m$  的向量, 满足:

$$Ax = \begin{pmatrix} a_{r,1}^T x \\ \vdots \\ a_{r,m}^T x \end{pmatrix} = \sum_{i=1}^n a_{c,i} x_i \in \mathbb{R}^m$$

其中  $a_{r,i}^T$  是行向量,  $a_{c,j}$  是  $A$  的列向量,  $x_i$  是  $x$  的元素。

□ **矩阵-矩阵** – 矩阵  $A \in \mathbb{R}^{m \times n}$  和  $B \in \mathbb{R}^{n \times p}$  的乘积是一个大小为  $\mathbb{R}^{m \times p}$  的矩阵, 满足:

$$AB = \begin{pmatrix} a_{r,1}^T b_{c,1} & \cdots & a_{r,1}^T b_{c,p} \\ \vdots & & \vdots \\ a_{r,m}^T b_{c,1} & \cdots & a_{r,m}^T b_{c,p} \end{pmatrix} = \sum_{i=1}^n a_{c,i} b_{r,i}^T \in \mathbb{R}^{m \times p}$$

其中  $a_{r,i}^T, b_{r,i}^T$  是行向量,  $a_{c,j}, b_{c,j}$  分别是  $A$  和  $B$  的列向量

□ **转置** – 矩阵  $A \in \mathbb{R}^{m \times n}$  的转置, 记作  $A^T$ , 是其中元素的翻转

$$\forall i, j, \quad A_{i,j}^T = A_{j,i}$$

注: 对矩阵  $A, B$ , 我们有  $(AB)^T = B^T A^T$

□ **逆** – 可逆方阵  $A$  的逆记作  $A^{-1}$  和唯一满足下列要求的矩阵:

$$AA^{-1} = A^{-1}A = I$$

注: 不是所有方阵都是可逆的。同样, 对矩阵  $A, B$ , 我们有  $(AB)^{-1} = B^{-1}A^{-1}$

□ **迹** – 方阵  $A$  的迹, 记作  $\text{tr}(A)$ , 是对角线元素的和:

$$\text{tr}(A) = \sum_{i=1}^n A_{i,i}$$

注: 对矩阵  $A, B$ , 我们有  $\text{tr}(A^T) = \text{tr}(A)$  和  $\text{tr}(AB) = \text{tr}(BA)$

□ **行列式** – 方阵的行列式, 记作  $|A|$  或者  $\det(A)$  采用去掉第  $i$  行  $j$  列的矩阵  $A_{\setminus i, \setminus j}$  递归表达为如下形式:

$$\det(A) = |A| = \sum_{j=1}^n (-1)^{i+j} A_{i,j} |A_{\setminus i, \setminus j}|$$

注:  $A$  可逆当且仅当  $|A| \neq 0$ 。同样, 有  $|AB| = |A||B|$  和  $|A^T| = |A|$ 。

## 矩阵的性质

□ **对称分解** – 一个给定矩阵  $A$  可以用其对称和反对称部分进行表示:

$$A = \underbrace{\frac{A + A^T}{2}}_{\text{对称}} + \underbrace{\frac{A - A^T}{2}}_{\text{反对称}}$$

□ **范数** – 一个范数是一个函数  $N : V \rightarrow [0, +\infty]$  其中  $V$  是一个向量空间, 满足对所有  $x, y \in V$ , 有:

- $N(x + y) \leq N(x) + N(y)$
- 对一个标量  $a$ , 有  $N(ax) = |a|N(x)$
- 若  $N(x) = 0$ , 则  $x = 0$

对  $x \in V$ , 最常用的范数列在下表中:

范数	符号	定义	用例
曼哈顿, $L^1$	$\ x\ _1$	$\sum_{i=1}^n  x_i $	LASSO
欧几里德, $L^2$	$\ x\ _2$	$\sqrt{\sum_{i=1}^n x_i^2}$	Ridge
$p$ -范数, $L^p$	$\ x\ _p$	$\left(\sum_{i=1}^n x_i^p\right)^{\frac{1}{p}}$	赫尔德不等式
无穷, $L^\infty$	$\ x\ _\infty$	$\max_i  x_i $	一致收

□ **线性相关** – 向量集合被称作线性相关的当其中一个向量可以被定义为其线性组合。

注: 若无向量可以按照此法表示, 则这些向量被称为线性无关。

□ **矩阵的秩** – 给定矩阵  $A$  的秩记作  $\text{rank}(A)$  是由列向量生成的向量空间的维度。这等于  $A$  的线性无关列向量的最大数目。

□ **半正定矩阵** – 矩阵  $A \in \mathbb{R}^{n \times n}$  是半正定矩阵 (PSD), 记作  $A \succeq 0$ , 当我们有:

$$A = A^T \quad \text{和} \quad \forall x \in \mathbb{R}^n, \quad x^T A x \geq 0$$

注: 类似地, 矩阵  $A$  被称作正定, 记作  $A \succ 0$ , 当它是一个 PSD 矩阵且满足所有非零向量  $x$ ,  $x^T A x > 0$ 。

□ **特征值, 特征向量** – 给定矩阵  $A \in \mathbb{R}^{n \times n}$ ,  $\lambda$  被称作  $A$  的一个特征值当存在一个向量  $z \in \mathbb{R}^n \setminus \{0\}$  称作特征向量, 满足:

$$Az = \lambda z$$

□ **谱定理** – 令  $A \in \mathbb{R}^{n \times n}$ , 若  $A$  是对称的, 则  $A$  可以被一个实正交矩阵  $U \in \mathbb{R}^{n \times n}$  对角化。记  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ , 我们有:

$$\exists \Lambda \text{ 对角阵, } A = U \Lambda U^T$$

□ **奇异值分解** – 对一个给定矩阵  $A$ , 其维度为  $m \times n$ , 奇异值分解 (SVD) 是一个因子分解技巧, 能保证存在酉矩阵  $U \in \mathbb{R}^{m \times m}$ , 对角阵  $\Sigma \in \mathbb{R}^{m \times n}$  和酉矩阵  $V \in \mathbb{R}^{n \times n}$ , 满足:

$$A = U \Sigma V^T$$

## 矩阵的微积分

□ **梯度** – 令  $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$  一个函数  $A \in \mathbb{R}^{m \times n}$  一个矩阵。  $f$  关于  $A$  的梯度是一个  $m \times n$  的矩阵, 记作  $\nabla_A f(A)$ , 满足:

$$\left( \nabla_A f(A) \right)_{i,j} = \frac{\partial f(A)}{\partial A_{i,j}}$$

注:  $f$  的梯度仅当  $f$  是返回一个标量的函数时有定义。

□ **Hessian** – 令  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  一个函数,  $x \in \mathbb{R}^n$  一个向量。  $f$  的关于  $x$  的 Hessian 是一个  $n \times n$  的对称阵, 记作  $\nabla_x^2 f(x)$ , 满足:

$$\left( \nabla_x^2 f(x) \right)_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

注:  $f$  的 Hessian 仅当  $f$  是一个返回标量的函数时有定义。

□ **梯度运算** – 对矩阵  $A, B, C$ , 下列梯度性质值得记住:

$$\nabla_A \text{tr}(AB) = B^T$$

$$\nabla_A^T f(A) = (\nabla_A f(A))^T$$

$$\nabla_A \text{tr}(ABA^T C) = CAB + C^T A B^T$$

$$\nabla_A |A| = |A| (A^{-1})^T$$

# VIP Refresher: Probabilities and Statistics

Afshine AMIDI and Shervine AMIDI

October 27, 2018

翻译: 朱小虎

## 概率和组合导引

□ **样本空间** – 一个实验的所有可能结果的集合称为实验的样本空间，记作  $S$ 。

□ **事件** – 样本空间的任何子集  $E$  被称为一个事件。即，一个事件是一个包含可能结果的集合。如果该实验的结果包含在  $E$  内，那么我们称  $E$  发生。

□ **概率论公理** – 对每个事件  $E$ ，我们记  $P(E)$  为事件  $E$  出现的概率。

$$(1) \quad 0 \leq P(E) \leq 1 \quad (2) \quad P(S) = 1 \quad (3) \quad P\left(\bigcup_{i=1}^n E_i\right) = \sum_{i=1}^n P(E_i)$$

□ **置换** – 一个置换是从  $n$  个对象的池中按照给定次序安置  $r$  个对象。这样的安置的数目由  $P(n, r)$  表示，定义为：

$$P(n, r) = \frac{n!}{(n-r)!}$$

□ **组合** – 一个组合是从  $n$  个对象的池中无序安置  $r$  个对象。这样的安置的数目由  $C(n, r)$  表示，定义为：

$$C(n, r) = \frac{P(n, r)}{r!} = \frac{n!}{r!(n-r)!}$$

注：对  $0 \leq r \leq n$ ，我们有  $P(n, r) \geq C(n, r)$

## 条件概率

□ **贝叶斯规则** – 对事件  $A$  和  $B$  满足  $P(B) > 0$ ，我们有：

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

注：我们有  $P(A \cap B) = P(A)P(B|A) = P(A|B)P(B)$

□ **分划** – 令  $\{A_i, i \in [1, n]\}$  对所有  $i$ ， $A_i \neq \emptyset$ 。我们称  $\{A_i\}$  为一个分划，当有：

$$\forall i \neq j, A_i \cap A_j = \emptyset \quad \text{和} \quad \bigcup_{i=1}^n A_i = S$$

注：对任意在样本空间中的事件  $B$  我们有  $P(B) = \sum_{i=1}^n P(B|A_i)P(A_i)$ 。

□ **贝叶斯规则的扩展形式** – 令  $\{A_i, i \in [1, n]\}$  为样本空间的一个分划，我们有：

$$P(A_k|B) = \frac{P(B|A_k)P(A_k)}{\sum_{i=1}^n P(B|A_i)P(A_i)}$$

□ **独立** – 两个事件  $A$  和  $B$  是独立的当且仅当我们有：

$$P(A \cap B) = P(A)P(B)$$

## 随机变量

□ **随机变量** – 一个随机变量，通常记作  $X$ ，是一个将在一个样本空间中的每个元素映射到一个实值的函数。

□ **累积分布函数 (CDF)** – 累积分布函数  $F$ ，是单调不减的，其

$$\lim_{x \rightarrow -\infty} F(x) = 0 \quad \text{和} \quad \lim_{x \rightarrow +\infty} F(x) = 1$$

定义为：

$$F(x) = P(X \leq x)$$

注：我们有  $P(a < X \leq b) = F(b) - F(a)$ 。

□ **概率密度函数 (PDF)** – 概率密度函数  $f$  是  $X$  取值在两个相邻随机变量的实现间的概率。

□ **PDF 和 CDF 的关系** – 这里是离散和连续场景下的重要性质。

类型	CDF $F$	PDF $f$	PDF 的性质
(D)	$F(x) = \sum_{x_i \leq x} P(X = x_i)$	$f(x_j) = P(X = x_j)$	$0 \leq f(x_j) \leq 1$ and $\sum_j f(x_j) = 1$
(C)	$F(x) = \int_{-\infty}^x f(y) dy$	$f(x) = \frac{dF}{dx}$	$f(x) \geq 0$ and $\int_{-\infty}^{+\infty} f(x) dx = 1$

□ **方差** – 随机变量的方差通常记作  $\text{Var}(X)$  或者  $\sigma^2$ ，是分布函数的扩散性的一个度量函数。定义如下：

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

□ **标准差** – 随机变量的标准差，通常记作  $\sigma$ ，是分布函数扩散性的一个和实际随机变量值单位相当的度量函数。定义如下：

$$\sigma = \sqrt{\text{Var}(X)}$$

□ **分布的期望和矩** – 这里是期望值  $E[X]$ 、一般期望值  $E[g(X)]$ 、第  $k$  阶矩  $E[X^k]$  和特征函数  $\psi(\omega)$  在离散和连续场景下的表达式：

Case	$E[X]$	$E[g(X)]$	$E[X^k]$	$\psi(\omega)$
(D)	$\sum_{i=1}^n x_i f(x_i)$	$\sum_{i=1}^n g(x_i) f(x_i)$	$\sum_{i=1}^n x_i^k f(x_i)$	$\sum_{i=1}^n f(x_i) e^{i\omega x_i}$
(C)	$\int_{-\infty}^{+\infty} x f(x) dx$	$\int_{-\infty}^{+\infty} g(x) f(x) dx$	$\int_{-\infty}^{+\infty} x^k f(x) dx$	$\int_{-\infty}^{+\infty} f(x) e^{i\omega x} dx$

□ **随机变量的变换** – 令变量  $X$  和  $Y$  由某个函数联系在一起。记  $f_X$  和  $f_Y$  分别为  $X$  和  $Y$  的分布函数，我们有：

$$f_Y(y) = f_X(x) \left| \frac{dx}{dy} \right|$$

□ **莱布尼兹积分法则** – 令  $g$  为  $x$  和  $c$  的函数， $a, b$  是可能依赖于  $c$  的边界。我们有：

$$\frac{\partial}{\partial c} \left( \int_a^b g(x) dx \right) = \frac{\partial b}{\partial c} \cdot g(b) - \frac{\partial a}{\partial c} \cdot g(a) + \int_a^b \frac{\partial g}{\partial c}(x) dx$$

□ **切比雪夫不等式** – 令  $X$  为随机变量期望值为  $\mu$ 。对  $k, \sigma > 0$ ，我们有下列不等式：

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

## 联合分布随机变量

□ **条件密度** –  $X$  关于  $Y$  的条件密度通常记作  $f_{X|Y}$ ，定义如下：

$$f_{X|Y}(x) = \frac{f_{XY}(x, y)}{f_Y(y)}$$

□ **独立性** – 两个随机变量  $X$  和  $Y$  被称为独立的当我们有：

$$f_{XY}(x, y) = f_X(x) f_Y(y)$$

□ **边缘密度和累积分布** – 从联合密度概率函数  $f_{XY}$ ，我们有：

类型	边缘密度函数	累积函数
(D)	$f_X(x_i) = \sum_j f_{XY}(x_i, y_j)$	$F_{XY}(x, y) = \sum_{x_i \leq x} \sum_{y_j \leq y} f_{XY}(x_i, y_j)$
(C)	$f_X(x) = \int_{-\infty}^{+\infty} f_{XY}(x, y) dy$	$F_{XY}(x, y) = \int_{-\infty}^x \int_{-\infty}^y f_{XY}(x', y') dx' dy'$

□ **协方差** – 我们定义两个随机变量  $X$  和  $Y$  的协方差，记作  $\sigma_{XY}^2$  或者更常见的  $\text{Cov}(X, Y)$ ，如下：

$$\text{Cov}(X, Y) \triangleq \sigma_{XY}^2 = E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - \mu_X \mu_Y$$

□ **相关性** – 记  $\sigma_X, \sigma_Y$  为  $X$  和  $Y$  的标准差，我们定义随机变量  $X$  和  $Y$  的相关性，记作  $\rho_{XY}$ ，如下：

$$\rho_{XY} = \frac{\sigma_{XY}^2}{\sigma_X \sigma_Y}$$

注：对任何随机变量  $X, Y$ ，我们有  $\rho_{XY} \in [-1, 1]$ 。

□ **主要的分布** – 这里是主要需要记住的分布：

类型	分布	PDF	$\psi(\omega)$	$E[X]$	$\text{Var}(X)$
(D)	$X \sim \mathcal{B}(n, p)$ Binomial	$P(X = x) = \binom{n}{x} p^x q^{n-x}$ $x \in \llbracket 0, n \rrbracket$	$(pe^{i\omega} + q)^n$	$np$	$npq$
	$X \sim \text{Po}(\mu)$ Poisson	$P(X = x) = \frac{\mu^x}{x!} e^{-\mu}$ $x \in \mathbb{N}$	$e^{\mu(e^{i\omega} - 1)}$	$\mu$	$\mu$
(C)	$X \sim \mathcal{U}(a, b)$ Uniforme	$f(x) = \frac{1}{b-a}$ $x \in [a, b]$	$\frac{e^{i\omega b} - e^{i\omega a}}{(b-a)i\omega}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
	$X \sim \mathcal{N}(\mu, \sigma)$ Gaussien	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ $x \in \mathbb{R}$	$e^{i\omega\mu - \frac{1}{2}\omega^2\sigma^2}$	$\mu$	$\sigma^2$
	$X \sim \text{Exp}(\lambda)$ Exponentiel	$f(x) = \lambda e^{-\lambda x}$ $x \in \mathbb{R}_+$	$\frac{1}{1 - \frac{i\omega}{\lambda}}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$

## 参数估计

□ **随机采样** – 一个随机采样是  $n$  个和  $X$  独立同分布的随机变量  $X_1, \dots, X_n$  的集。

□ **估计器** – 估计器是一个用来推断一个统计模型中未知参数值的关于数据的函数。

□ **偏差** – 估计器  $\hat{\theta}$  的偏差定义为  $\hat{\theta}$  分布的期望值和真实值间的差距，即：

$$\text{Bias}(\hat{\theta}) = E[\hat{\theta}] - \theta$$

注：估计器被称为无偏的当我们有  $E[\hat{\theta}] = \theta$ 。

□ **中央极限定理** – 令一个随机采样  $X_1, \dots, X_n$  满足一个给定分布均值  $\mu$  方差  $\sigma^2$ ，我们有：

$$\bar{X}_{n \rightarrow +\infty} \sim \mathcal{N}\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

# VIP Cheatsheet: Machine Learning Tips

Afshine AMIDI and Shervine AMIDI

October 13, 2018

翻译: *hujinsen*. 由 *spin6lock* 审阅.

## 分类问题的度量

在二分类问题中, 下面这些主要度量标准对于评估模型的性能非常重要。

□ **混淆矩阵** – 混淆矩阵可以用来评估模型的整体性能情况。它的定义如下:

		预测类别	
		+	-
实际类别	+	<b>TP</b> True Positives	<b>FN</b> False Negatives Type II error
	-	<b>FP</b> False Positives Type I error	<b>TN</b> True Negatives

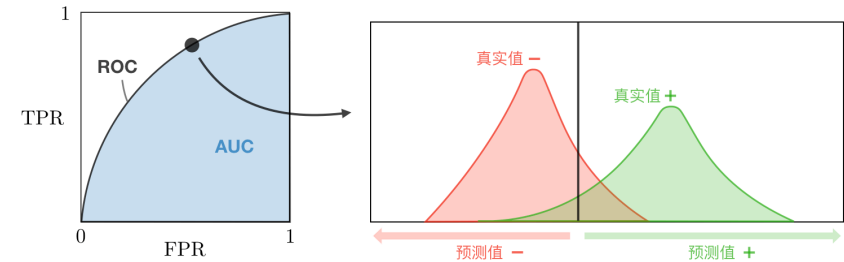
□ **主要度量标准** – 通常用下面的度量标准来评估分类模型的性能:

性能度量	公式	说明
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	模型总体性能
Precision	$\frac{TP}{TP + FP}$	预测为正样本的准确度
Recall Sensitivity	$\frac{TP}{TP + FN}$	真正样本的覆盖度
Specificity	$\frac{TN}{TN + FP}$	真负样本的覆盖度
F1 score	$\frac{2TP}{2TP + FP + FN}$	混合度量, 对于不平衡类别非常有效

□ **ROC** – 受试者工作曲线, 又叫做ROC曲线, 它使用真正例率和假正例率分别作为纵轴和横轴并且经过调整阈值绘制出来。下表汇总了这些度量标准:

性能度量	公式	等价形式
True Positive Rate TPR	$\frac{TP}{TP + FN}$	Recall, sensitivity
False Positive Rate FPR	$\frac{FP}{TN + FP}$	1-specificity

□ **AUC** – 受试者工作曲线的之下的部分, 又叫做AUC或者AUROC, 如下图所示ROC曲线下的部分:



## 回归指标

□ **基本性能度量** – 给定一个回归模型 $f$ , 下面的度量标准通常用来评估模型的性能

全部平方和	解释平方和	残差平方和
$SS_{\text{tot}} = \sum_{i=1}^m (y_i - \bar{y})^2$	$SS_{\text{reg}} = \sum_{i=1}^m (f(x_i) - \bar{y})^2$	$SS_{\text{res}} = \sum_{i=1}^m (y_i - f(x_i))^2$

□ **确定性系数** – 确定性系数, 记作 $R^2$  或  $r^2$ , 提供了模型复现观测结果的能力, 定义如下:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

□ **主要性能度量** – 以下性能度量通过考虑变量 $n$ 的数量, 常用于评估回归模型的性能:

Mallow's CP	AIC	BIC	Adjusted $R^2$
$\frac{SS_{\text{res}} + 2(n+1)\hat{\sigma}^2}{m}$	$2[(n+2) - \log(L)]$	$\log(m)(n+2) - 2\log(L)$	$1 - \frac{(1-R^2)(m-1)}{m-n-1}$

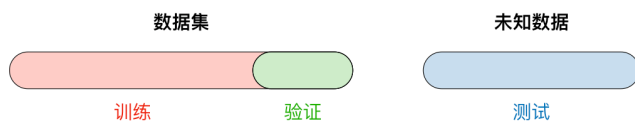
$L$  代表似然,  $\hat{\sigma}^2$  代表方差

## 模型选择

□ **词汇** – 在选择模型时，我们将数据分为的3个不同部分：

训练集	验证集	测试集
<ul style="list-style-type: none"> <li>- 模型训练</li> <li>- 一般数据集中的80</li> </ul>	<ul style="list-style-type: none"> <li>- 模型评估</li> <li>- 一般数据集中的20</li> <li>- 又叫做留出集或者开发集</li> </ul>	<ul style="list-style-type: none"> <li>- 模型预测</li> <li>- 未知数据</li> </ul>

一旦选择了模型，就会在整个数据集上进行训练，并在测试集上进行测试。如下图所示：



□ **交叉验证** – 交叉验证，记为CV，是一种不必特别依赖于初始训练集的模型选择方法。下表汇总了几种不同的方式：

$k$ -fold	Leave- $p$ -out
<ul style="list-style-type: none"> <li>- 在 <math>k - 1</math> 个子集上训练，在剩余的一个子集中评估</li> <li>- 通常 <math>k = 5</math> 或 <math>10</math></li> </ul>	<ul style="list-style-type: none"> <li>- 在 <math>n - p</math> 个子集上训练，在剩余的 <math>p</math> 个子集评估模型</li> <li>- <math>p = 1</math> 时又叫做留一法</li> </ul>

最常用的模型选择方法是  $k$  折交叉验证，将训练集划分为  $k$  个子集，在  $k - 1$  个子集上训练模型，在剩余的一个子集上评估模型，用这种划分方式重复训练  $k$  次。交叉验证损失是  $k$  次  $k$  折交叉验证的损失均值。

子集	数据集	验证错误	交叉验证错误
1		$\epsilon_1$	$\frac{\epsilon_1 + \dots + \epsilon_k}{k}$
2		$\epsilon_2$	
$\vdots$	$\vdots$	$\vdots$	
$k$		$\epsilon_k$	
	训练      验证		

□ **正则化** – 正则化方法可以解决高方差问题，避免模型对于训练数据产生过拟合。下表展示了常用的正则化方法：

LASSO	Ridge	Elastic Net
<ul style="list-style-type: none"> <li>- 将系数收缩为0</li> <li>- 有利于变量选择</li> </ul>	使系数更小	在变量选择和小系数之间进行权衡
$\dots + \lambda   \theta  _1$ $\lambda \in \mathbb{R}$	$\dots + \lambda   \theta  _2^2$ $\lambda \in \mathbb{R}$	$\dots + \lambda \left[ (1 - \alpha)   \theta  _1 + \alpha   \theta  _2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0, 1]$

## 诊断

□ **偏差** – 模型的偏差是模型预测值和真实值之间的差距

□ **方差** – 模型的方差是给定数据点的模型预测的可变性

□ **偏差/方差权衡** – 模型越简单，偏差越高，模型越复杂，方差越高。

	Underfitting	Just right	Overfitting
症状	<ul style="list-style-type: none"> <li>- 高训练误差</li> <li>训练误差接近测试误差</li> <li>- 高偏差</li> </ul>	<ul style="list-style-type: none"> <li>- 训练误差略低于测试误差</li> </ul>	<ul style="list-style-type: none"> <li>- 极低训练误差</li> <li>训练误差远低于测试误差</li> <li>- 高方差</li> </ul>
回归图			

分类图			
深度学习插图			
可能的补救措施	<ul style="list-style-type: none"> <li>- 模型复杂性</li> <li>- 添加更多特征</li> <li>- 训练更长时间</li> </ul>		<ul style="list-style-type: none"> <li>- 实施正则化</li> <li>- 获得更多数据</li> </ul>

❑ **错误分析** – 错误分析分析当前模型和完美模型之间性能差异的根本原因

❑ **烧蚀分析** – 烧蚀分析可以分析当前和基线模型之间性能差异的根本原因

# VIP Cheatsheet: Supervised Learning

Afshine AMIDI and Shervine AMIDI

October 27, 2018

翻译: Wang Hongnian. 由朱小虎, Chaoying Xue and Z 审阅

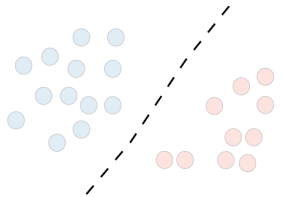
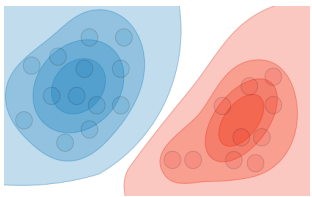
## 监督学习简介

给定一组数据点  $\{x^{(1)}, \dots, x^{(m)}\}$  和与其对应的输出  $\{y^{(1)}, \dots, y^{(m)}\}$ , 我们想要建立一个分类器, 学习如何从  $x$  预测  $y$ 。

□ **预测类型** – 不同类型的预测模型总结如下表:

	回归	分类
输出	连续	类
例子	线性回归	Logistic回归, SVM, 朴素贝叶斯

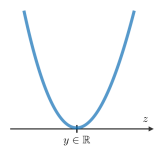
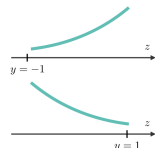
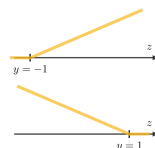
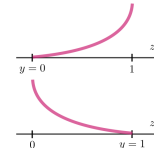
□ **型号类型** – 不同型号总结如下表:

	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

## 符号和一般概念

□ **假设** – 假设我们选择的模型是  $h_\theta$ 。对于给定的输入数据  $x^{(i)}$ , 模型预测输出是  $h_\theta(x^{(i)})$ 。

□ **损失函数** – 损失函数是一个  $L: (z, y) \in \mathbb{R} \times Y \mapsto L(z, y) \in \mathbb{R}$  的函数, 其将真实数据值  $y$  和其预测值  $z$  作为输入, 输出它们的不同程度。常见的损失函数总结如下表:

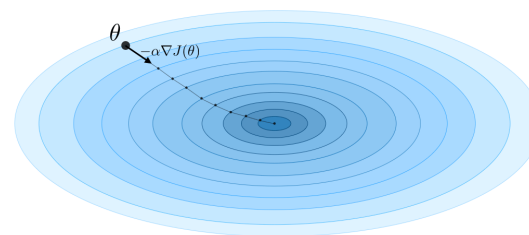
最小二乘误差	Logistic损失	铰链损失	交叉熵
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$	$-\left[y \log(z) + (1 - y) \log(1 - z)\right]$
			
线性回归	Logistic回归	SVM	神经网络

□ **成本函数** – 成本函数  $J$  通常用于评估模型的性能, 使用损失函数  $L$  定义如下:

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

□ **梯度下降** – 记学习率为  $\alpha \in \mathbb{R}$ , 梯度下降的更新规则使用学习率和成本函数  $J$  表示如下:

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



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** – 给定参数  $\theta$  的模型  $L(\theta)$  的似然性用于通过最大化似然性来找到最佳参数  $\theta$ 。在实践中, 我们使用更容易优化的对数似然  $\ell(\theta) = \log(L(\theta))$ 。我们有:

$$\theta^{\text{opt}} = \arg \max_{\theta} L(\theta)$$

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

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

备注: 多维泛化, 也称为 *Newton-Raphson* 方法, 具有以下更新规则:

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



## 线性回归

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

□ **正规方程** – 通过设计  $X$  矩阵，使得最小化成本函数时  $\theta$  有闭式解：

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

□ **LMS算法** – 通过  $\alpha$  学习率，训练集中  $m$  个数据的最小均方（LMS）算法的更新规则也称为Widrow-Hoff学习规则，如下：

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

备注：更新规则是梯度上升的特定情况。

□ **LWR** – 局部加权回归，也称为LWR，是线性回归的变体，通过  $w^{(i)}(x)$  对其成本函数中的每个训练样本进行加权，其中参数  $\tau \in \mathbb{R}$  定义为：

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

## 分类和逻辑回归

□ **Sigmoid函数** – sigmoid 函数  $g$ ，也称为逻辑函数，定义如下：

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

□ **逻辑回归** – 我们假设  $y|x; \theta \sim \text{Bernoulli}(\phi)$ 。我们有以下形式：

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

备注：对于逻辑回归的情况，没有闭式解。

□ **Softmax回归** – 当存在超过2个结果类时，使用softmax回归（也称为多类逻辑回归）来推广逻辑回归。按照惯例，我们设置  $\theta_K = 0$ ，使得每个类  $i$  的伯努利参数  $\phi_i$  等于：

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

## 广义线性模型

□ **指数分布族** – 如果可以用自然参数  $\eta$ ，也称为规范参数或链接函数，充分统计量  $T(y)$  和对数分割函数  $a(\eta)$  来表示，则称一类分布在指数分布族中，函数如下：

$$p(y; \eta) = b(y) \exp(\eta^T T(y) - a(\eta))$$

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

下表是总结的最常见的指数分布：

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

□ **GLM的假设** – 广义线性模型（GLM）是旨在将随机变量  $y$  预测为  $x \in \mathbb{R}^{n+1}$  的函数，并依赖于以下3个假设：

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

备注：普通最小二乘法和逻辑回归是广义线性模型的特例

## 支持向量机

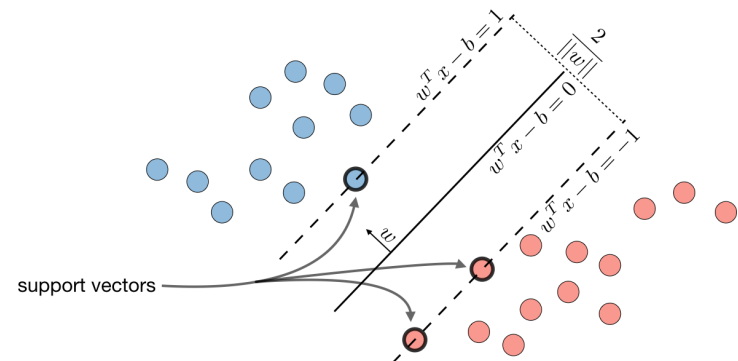
支持向量机的目标是找到使决策界和训练样本之间最大化最小距离的线。

□ **最优间隔分类器** – 最优间隔分类器  $h$  是这样的：

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

其中  $(w, b) \in \mathbb{R}^n \times \mathbb{R}$  是以下优化问题的解决方案：

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



备注：该线定义为  $w^T x - b = 0$ 。

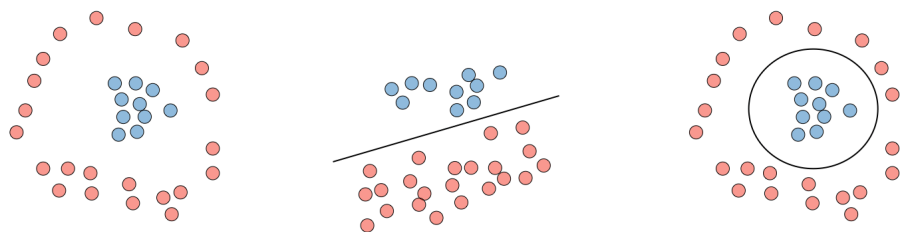
□ **合页损失** – 合页损失用于SVM，定义如下：

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

□ **核** – 给定特征映射 $\phi$ ，我们定义核 $K$ 为：

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

在实践中，由 $K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$  定义的核 $K$  被称为高斯核，并且经常使用这种核。



非线性可分性

核映射的使用  $\phi$

原始空间中的决策边界

备注：我们说我们使用“核技巧”来计算使用核的成本函数，因为我们实际上不需要知道显式映射 $\phi$ ，通常，这非常复杂。相反，只需要 $K(x, z)$  的值。

□ **拉格朗日** – 我们将拉格朗日 $\mathcal{L}(w, b)$  定义如下：

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

备注：系数 $\beta_i$  称为拉格朗日乘子。

## 生成学习

生成模型首先尝试通过估计 $P(x|y)$  来模仿如何生成数据，然后我们可以使用贝叶斯法则来估计 $P(y|x)$

## 高斯判别分析

□ **设置** – 高斯判别分析假设 $y$  和 $x|y = 0$  且 $x|y = 1$  如下：

$$y \sim \text{Bernoulli}(\phi)$$

$$x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma)$$

$$\text{和 } x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$$

□ **估计** – 下表总结了我们在最大化似然时的估计值：

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

## 朴素贝叶斯

□ **假设** – 朴素贝叶斯模型假设每个数据点的特征都是独立的：

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

□ **解决方案** – 最大化对数似然给出以下解， $k \in \{0, 1\}, l \in [1, L]$

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

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

备注：朴素贝叶斯广泛用于文本分类和垃圾邮件检测。

## 基于树的方法和集成方法

这些方法可用于回归和分类问题。

□ **CART** – 分类和回归树（CART），通常称为决策树，可以表示为二叉树。它们具有可解释性的优点。

□ **随机森林** – 这是一种基于树模型的技术，它使用大量的由随机选择的特征集构建的决策树。与简单的决策树相反，它是高度无法解释的，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。

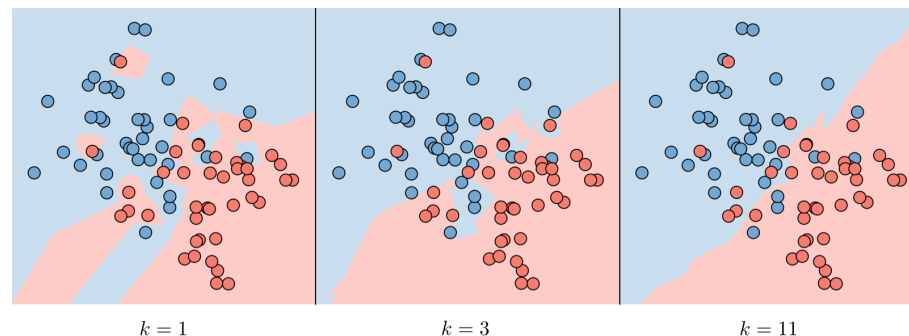
□ **提升** – 提升方法的思想是将一些弱学习器结合起来形成一个更强大的学习器。主要内容总结在下表中：

自适应增强	梯度提升
在下一轮提升步骤中，错误的会被置于高权重	弱学习器训练剩余的错误

## 其他非参数方法

□ **k-最近邻** – k-最近邻算法，通常称为k-NN，是一种非参数方法，其中数据点的判决由来自训练集中与其相邻的k个数据的性质确定。它可以用于分类和回归。

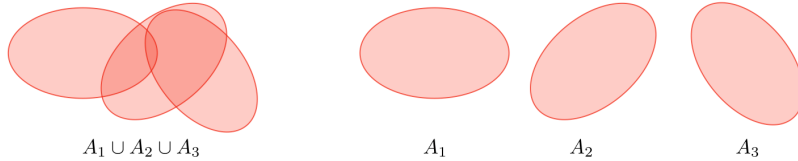
备注：参数 $k$  越高，偏差越大，参数 $k$  越低，方差越大。



## 学习理论

□ **联盟** – 让  $A_1, \dots, A_k$  成为  $k$  个事件。我们有：

$$P(A_1 \cup \dots \cup A_k) \leq P(A_1) + \dots + P(A_k)$$



□ **Hoeffding不等式** – 设  $Z_1, \dots, Z_m$  是从参数  $\phi$  的伯努利分布中提取的  $m$  iid 变量。设  $\phi$  为其样本均值，固定  $\gamma > 0$ 。我们有：

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

备注：这种不平等也被称为 *Chernoff* 界限。

□ **训练误差** – 对于给定的分类器  $h$ ，我们定义训练误差  $\hat{\epsilon}(h)$ ，也称为经验风险或经验误差，如下：

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

□ **可能近似正确(PAC)** – PAC是一个框架，在该框架下证明了许多学习理论的结果，并具有以下假设：

- 训练和测试集遵循相同的分布
- 训练样本是相互独立的

□ **打散** – 给定一个集合  $S = \{x^{(1)}, \dots, x^{(d)}\}$  和一组分类器  $\mathcal{H}$ ，如果对于任意一组标签  $\{y^{(1)}, \dots, y^{(d)}\}$  都能对分，我们称  $\mathcal{H}$  打散  $S$ ，我们有：

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

□ **上限定理** – 设  $\mathcal{H}$  是有限假设类，使得  $|\mathcal{H}| = k$  并且使  $\delta$  和样本大小  $m$  固定。然后，在概率至少为  $1 - \delta$  的情况下，我们得到：

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

□ **VC维** – 给定无限假设类  $\mathcal{H}$  的Vapnik-Chervonenkis(VC) 维，注意  $\text{VC}(\mathcal{H})$  是由  $\mathcal{H}$  打散的最大集合的大小。

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



□ **定理(Vapnik)** – 设  $\mathcal{H}$ ,  $\text{VC}(\mathcal{H}) = d$ ,  $m$  为训练样本数。概率至少为  $1 - \delta$ ，我们有：

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

# VIP Cheatsheet: Unsupervised Learning

Afshine AMIDI and Shervine AMIDI

October 27, 2018

翻译: 朱小虎

## 无监督学习导引

□ **动机** – 无监督学习的目标是找到在未标记数据  $\{x^{(1)}, \dots, x^{(m)}\}$  中的隐含模式。

□ **Jensen 不等式** – 令  $f$  为一个凸函数而  $X$  为一个随机变量。我们有下列不等式:

$$E[f(X)] \geq f(E[X])$$

## E-M 算法

□ **隐变量** – 隐变量是隐含/不可观测的变量，使得估计问题变得困难，通常被表示成  $z$ 。这里是包含隐变量的常见设定:

设定	隐变量 $z$	$x z$	评论
$k$ 元混合高斯分布	Multinomial( $\phi$ )	$\mathcal{N}(\mu_j, \Sigma_j)$	$\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$
因子分析	$\mathcal{N}(0, I)$	$\mathcal{N}(\mu + \Lambda z, \psi)$	$\mu_j \in \mathbb{R}^n$

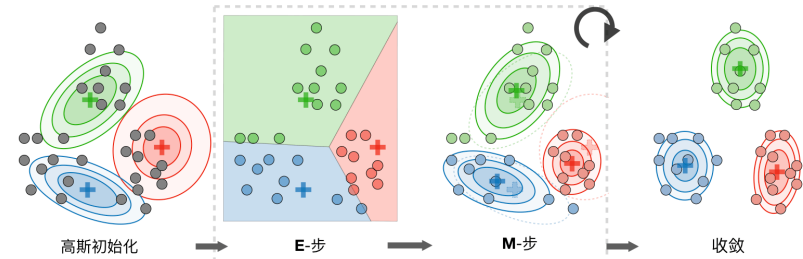
□ **算法** – E-M 算法给出了通过重复构建似然函数的下界 (E-步) 和最优化下界 (M-步) 进行极大似然估计给出参数  $\theta$  的高效估计方法:

- **E-步**: 计算后验概率  $Q_i(z^{(i)})$ , 其中每个数据点  $x^{(i)}$  来自特定的簇  $z^{(i)}$ , 过程如下:

$$Q_i(z^{(i)}) = P(z^{(i)} | x^{(i)}; \theta)$$

- **M-步**: 使用后验概率  $Q_i(z^{(i)})$  作为簇在数据点  $x^{(i)}$  上的特定权重来分别重新估计每个簇模型, 过程如下:

$$\theta_i = \operatorname{argmax}_{\theta} \sum_i \int_{z^{(i)}} Q_i(z^{(i)}) \log \left( \frac{P(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right) dz^{(i)}$$



## $k$ -均值聚类

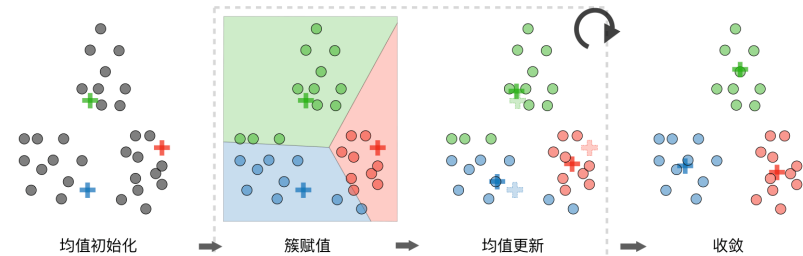
我们记  $c^{(i)}$  为数据点  $i$  的簇,  $\mu_j$  是簇  $j$  的中心。

□ **算法** – 在随机初始化簇中心  $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$  后,  $k$ -均值算法重复下列步骤直至收敛:

$$c^{(i)} = \operatorname{argmin}_j \|x^{(i)} - \mu_j\|^2$$

和

$$\mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}$$



□ **失真函数** – 为了看到算法是否收敛, 我们看看如下定义的失真函数:

$$J(c, \mu) = \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

## 层次化聚类

□ **算法** – 结合聚合层次化观点的聚类算法, 按照逐次构建嵌套簇的方式进行。

□ **类型** – 存在不同的层次化聚类算法, 解决不同的目标函数优化问题, 在下表中总结列出:

内链	均链	全链
最小化簇内距离	最小化簇对平均距离	最小化簇对的最大距离

## 聚类评测度量

在一个无监督学习设定中，通常难以评测一个模型的性能，因为我们没有像监督学习设定中那样的原始真实的类标。

□ **Silhouette 系数** – 通过记  $a$  和  $b$  为一个样本和在同一簇中的其他所有点之间的平均距离和一个样本和在下一个最近簇中的所有其他点的平均距离，针对一个样本的 Silhouette 系数  $s$  定义如下：

$$s = \frac{b - a}{\max(a, b)}$$

□ **Calinski-Harabaz 指标** – 通过记  $k$  为簇的数目， $B_k$  和  $W_k$  分别为簇间和簇内弥散矩阵，定义为：

$$B_k = \sum_{j=1}^k n_{c(i)} (\mu_{c(i)} - \mu)(\mu_{c(i)} - \mu)^T, \quad W_k = \sum_{i=1}^m (x^{(i)} - \mu_{c(i)})(x^{(i)} - \mu_{c(i)})^T$$

Calinski-Harabaz 指标  $s(k)$  表示一个聚类模型定义簇的好坏，分数越高，簇就越稠密和良好分隔。其定义如下：

$$s(k) = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{N - k}{k - 1}$$

## 主成分分析

这是一种维度降低的技巧，找到投影数据到能够最大化方差的方向。

□ **特征值，特征向量** – 给定矩阵  $A \in \mathbb{R}^{n \times n}$ ， $\lambda$  被称为  $A$  的一个特征值当存在一个称为特征向量的向量  $z \in \mathbb{R}^n \setminus \{0\}$ ，使得：

$$Az = \lambda z$$

□ **谱定理** – 令  $A \in \mathbb{R}^{n \times n}$ 。如果  $A$  是对称阵，那么  $A$  可以被一个实正交矩阵  $U \in \mathbb{R}^{n \times n}$  对角化。通过记  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  我们有：

$$\exists \Lambda \text{ 为对角阵, } A = U \Lambda U^T$$

注：关联于最大的特征值的特征向量被称为矩阵  $A$  的主特征向量。

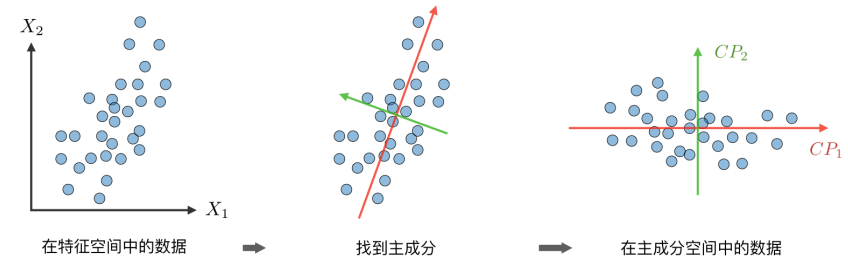
□ **算法** – 主成分分析（PCA）过程就是一个降维技巧，通过最大化数据的方差而将数据投影到  $k$  维上：

- 步骤1: 规范化数据使其均值为0 方差为1。

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j} \quad \text{哪里} \quad \mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \quad \text{和} \quad \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

- 步骤2: 计算  $\Sigma = \frac{1}{m} \sum_{i=1}^m x^{(i)} x^{(i)T} \in \mathbb{R}^{n \times n}$ ，其为有实特征值的对称阵

- 步骤3: 计算  $\Sigma$  的  $k$  个正交的主特征向量  $u_1, \dots, u_k \in \mathbb{R}^n$ ，即对应  $k$  个最大特征值的正交特征向量。
- 步骤4: 投影数据到  $\text{span}_{\mathbb{R}}(u_1, \dots, u_k)$  上。这个过程最大化所有  $k$  维空间的方差



## 独立成分分析

这是旨在找到背后生成源的技术。

□ **假设** – 我们假设数据  $x$  已经由  $n$ -维源向量  $s = (s_1, \dots, s_n)$  生成出来，其中  $s_i$  是独立的随机变量，通过一个混合和非奇异矩阵  $A$  如下方式产生：

$$x = As$$

目标是要找到去混合矩阵  $W = A^{-1}$ 。

□ **Bell-Sejnowski ICA 算法** – 该算法找出去混合矩阵  $W$ ，通过下列步骤：

- 记概率  $x = As = W^{-1}s$  如下：

$$p(x) = \prod_{i=1}^n p_s(w_i^T x) \cdot |W|$$

- 记给定训练数据  $\{x^{(i)}, i \in [1, m]\}$  对数似然函数其中  $g$  为 sigmoid 函数如下：

$$l(W) = \sum_{i=1}^m \left( \sum_{j=1}^n \log \left( g'(w_j^T x^{(i)}) \right) + \log |W| \right)$$

因此，随机梯度下降学习规则是，对每个训练样本  $x^{(i)}$ ，我们如下更新  $W$ ：

$$W \leftarrow W + \alpha \left( \begin{pmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{pmatrix} x^{(i)T} + (W^T)^{-1} \right)$$

# VIP Cheatsheet: Deep Learning

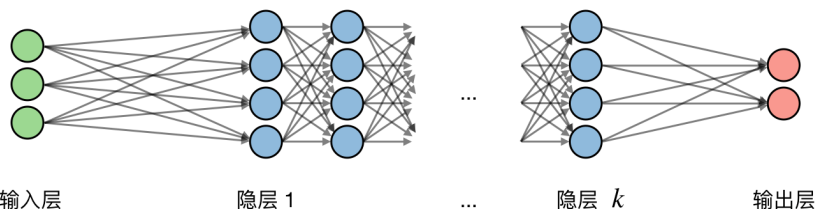
Afshine AMIDI and Shervine AMIDI

October 13, 2018

## 神经网络

神经网络是一类按层结构搭建的模型。常用的神经网络包括卷积神经网络和递归神经网络。

□ **架构** – 下表列举了用来描述神经网络架构的词汇：



已知 $i$  是网络的第 $i$  层,  $j$  是网络的第 $j$  层, 我们有:

$$z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]}$$

我们用 $w, b, z$  分别表示权重, 偏差和输出。

□ **激活函数** – 激活函数被用在隐含单元之后来向模型引入非线性复杂度。比较常见的如下所示:

逻辑函数(Sigmoid)	双曲正切函数(Tanh)	ReLU	Leaky ReLU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$

□ **交叉熵损失** – 在神经网络中, 交叉熵损失 $L(z, y)$  通常如下定义:

$$L(z, y) = - \left[ y \log(z) + (1 - y) \log(1 - z) \right]$$

□ **学习率** – 学习率, 通常记为 $\alpha$  或 $\eta$ , 表示权重更新的速度。它可以被修复或自适应改变。现阶段最常用的方法是一种调整学习率的算法, 叫做Adam。

□ **反向传播** – 反向传播是一种通过考虑实际输出和期望输出来更新神经网络中权重的方法。权重 $w$  的导数由链式法则计算, 模式如下:

$$\frac{\partial L(z, y)}{\partial w} = \frac{\partial L(z, y)}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w}$$

作为结果, 权重更新如下:

$$w \leftarrow w - \eta \frac{\partial L(z, y)}{\partial w}$$

□ **更新权重** – 在一个神经网络中, 权重如下所示更新:

- 第一步: 分出第一批次的训练数据。
- 第二步: 通过前向传播来得到相关损失。
- 第三步: 通过反向传播损失来得到梯度。
- 第四步: 利用梯度更新网络的权重。

□ **随机丢弃(Dropout)** – 随机丢弃是一种通过丢弃神经网络单元来防止训练数据过拟合的技术。实际上, 神经元以概率 $p$  被丢弃或以概率 $1 - p$  被保留。

## 卷积神经网络

□ **卷积神经网络要求** – 记 $W$  为输入图像尺寸,  $F$  为卷积层神经元尺寸,  $P$  为零填充的大小, 那么给定的输入图像能够容纳的神经元数目 $N$  为:

$$N = \frac{W - F + 2P}{S} + 1$$

□ **批量规范化** – 它是超参数 $\gamma, \beta$  标准化样本批 $\{x_i\}$  的一个步骤。将我们希望修正这一批样本的均值和方差记作 $\mu_B, \sigma_B^2$ , 会得到:

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

它通常在线性层之前和全连接/卷积层后应用, 目的是允许更高的学习率和减少初始化的强相关。

## 递归神经网络

□ **门控的种类** – 在一个典型的递归神经网络中有多种门控结构:

输入门	忘记门	门控	输出门
要不要写入单元?	要不要清空单元?	在单元写入多少?	从单元泄露多少?

□ **LSTM** – 长短期记忆网络是递归神经网络的一种, 它可以通过增加“遗忘”门控来避免梯度消失问题。

## 强化学习和控制

强化学习的目标是让代理学习如何在环境中进化。

□ **马尔可夫决策过程** – 一个马尔可夫决策过程(MDP)是一个5维元组 $(\mathcal{S}, \mathcal{A}, \{P_{sa}\}, \gamma, R)$ ，即：

- $\mathcal{S}$  是状态的集合
- $\mathcal{A}$  是动作的集合
- $\{P_{sa}\}$  是对于  $s$  属于  $\mathcal{S}$  并且  $a$  属于  $\mathcal{A}$  的状态转换概率
- $\gamma \in [0, 1]$  是折扣系数
- $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  或  $R : \mathcal{S} \rightarrow \mathbb{R}$  是算法希望最大化的回报函数

□ **策略** – 策略  $\pi$  是映射状态到动作的  $\pi : \mathcal{S} \rightarrow \mathcal{A}$  函数。

注意：如果对于一个指定的状态  $s$  我们完成了行动  $a = \pi(s)$ ，我们认为执行了一个指定的策略  $\pi$ 。

□ **价值函数** – 对于一个指定的策略  $\pi$  和指定的状态  $s$ ，我们定义如下价值函数  $V^\pi$ ：

$$V^\pi(s) = E \left[ R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots | s_0 = s, \pi \right]$$

□ **贝尔曼方程** – 最优贝尔曼方程描述了最优策略  $\pi^*$  的价值方程  $V^{\pi^*}$ ：

$$V^{\pi^*}(s) = R(s) + \max_{a \in \mathcal{A}} \gamma \sum_{s' \in \mathcal{S}} P_{sa}(s') V^{\pi^*}(s')$$

注意：我们注意到对于一个特定的状态  $s$  的最优策略  $\pi^*$  是：

$$\pi^*(s) = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{sa}(s') V^*(s')$$

□ **值迭代算法** – 值迭代算法分为两步：

- 首先我们初始化值：

$$V_0(s) = 0$$

- 我们通过之前的值进行迭代：

$$V_{i+1}(s) = R(s) + \max_{a \in \mathcal{A}} \left[ \sum_{s' \in \mathcal{S}} \gamma P_{sa}(s') V_i(s') \right]$$

□ **极大似然估计** – 状态转移概率的极大似然估计如下：

$$P_{sa}(s') = \frac{\text{\#状态 } s \text{ 下进行动作 } a \text{ 并且进入状态 } s' \text{ 的次数}}{\text{\#状态 } s \text{ 下进行动作 } a \text{ 的次数}}$$

□ **Q 学习** – Q 学习是一种 Q 的无模型(model-free)估计，如下所示：

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[ R(s, a, s') + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$