2016 Spring Notes

Yue Yu

February 23, 2016

Contents

1 各种相关背景知识

1.1 概率论

1.1.1 条件概率期望:

$$E(X) = E(E(X|Y)) \tag{1}$$

1.1.2 Correlation coefficient(相关系数)

$$\begin{aligned} \rho_{X,Y} &= \frac{Cov(X,Y)}{\sigma_X \sigma_Y} && (2)\\ &= \frac{E(X-E(X))(Y-E(Y))}{\sigma_X \sigma_Y} && (3)\\ &= \frac{E(XY)-E(X)E(Y)}{\sigma_X \sigma_Y} && (4) \end{aligned}$$

1.1.3 协方差矩阵

$$\begin{aligned} \mathrm{X} &= [X_1, \ldots, X_n]^T && (5)\\ \Sigma &= \mathrm{E}\Big[(\mathrm{X}-\mathrm{E}(\mathrm{X}))(\mathrm{X}-\mathrm{E}(\mathrm{X}))^T\Big] && (6)\\ \Sigma_{i,j} &= Cov(X_i, X_j) && (7)\\ &= \mathrm{E}\Big[(X_i-\mu_i)(X_j-\mu_j)\Big] && (8) \end{aligned}$$

1.1.4 Multivariate normal distribution

- 多维高斯联合分布:

$$f_X(x_1, \ldots, x_n) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$$

- 当为二维高斯分布时（ρ 为相关系数）

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\Big(-\frac{1}{2(1-\rho)^2}\Big[\frac{(x-\mu_x)}{\sigma_x^2} + \frac{(y-\mu_y)}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\Big]\Big)$$

条件概率服从:

$$P(X_1 \big| X_2 = x_2) \sim N(\mu_1 + \frac{\sigma_1}{\sigma_2}\rho(x_2-\mu_2), (1-\rho^2)\sigma_1^2)$$

1.1.5 Laplace Distribution:

- 概率密度函数

$$f(x|\mu,b)=\frac{1}{2b}\exp\big(-\frac{|x-\mu|}{b}\big)$$

- 期望

$$\mu$$

- 方差

$$2b^2$$

1.2 线性代数

- Trace

$$\mathrm{Tr}(\mathbf{ABC})=\mathrm{Tr}(\mathbf{BCA})=\mathrm{Tr}(\mathbf{CAB})$$

1.3 矩阵求导法则

- 雅各比矩阵
  假设某函数从 $\mathbb{R}^n$ 映射到 $\mathbb{R}^m$

$$J_F(x_1,\ldots,x_n)=\frac{\partial(y_1,\ldots,y_m)}{\partial(x_1,\ldots,x_n)}=\begin{pmatrix}\frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n}\\ \vdots & \ddots & \vdots\\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n}\end{pmatrix} \tag{9}$$

- Chain Rules:
  Suppose $\mathbf{y}=f(\mathbf{u}),\mathbf{u}=g(\mathbf{x})$

$$\frac{\partial(y_1,\ldots,y_m)}{\partial(x_1,\ldots,x_n)}=\frac{\partial(y_1,\ldots,y_m)}{\partial(u_1,\ldots,u_k)}\frac{\partial(u_1,\ldots,u_k)}{\partial(x_1,\ldots,x_n)} \tag{10}$$

- 对向量 (Vector Function) 求导 (定义列向量对标量求导得到的是行向量)：

$$\frac{\partial \mathbf{u^T A v}}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{A v} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{A^T u} \tag{11}$$

$$\frac{\partial(\mathbf{u(x)} + \mathbf{v(x)})}{\partial \mathbf{x}} = \frac{\partial \mathbf{u(x)}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v(x)}}{\partial \mathbf{x}} \tag{12}$$

$$\frac{\partial \mathbf{Ax}}{\partial \mathbf{x}} = \mathbf{A^T} \tag{13}$$

$$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} = \mathbf{0} \tag{14}$$

$$\frac{\partial \mathbf{x^T A b}}{\partial \mathbf{x}} = \mathbf{Ab} \tag{15}$$

$$\frac{\partial \mathbf{x^T A x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A^T})\mathbf{x} \tag{16}$$

$$= \mathbf{2Ax} \quad \text{如果 A 为对称阵} \tag{17}$$

$$\tag{18}$$

- 矩阵行列式对矩阵求导：

$$\frac{\partial |\mathbf{X}|}{\partial \mathbf{X}} = |\mathbf{X}|(\mathbf{X^{-1}})^\mathbf{T} \tag{19}$$

$$\frac{\partial \ln |\mathbf{X}|}{\partial \mathbf{X}} = (\mathbf{X^{-1}})^\mathbf{T} \tag{20}$$

- 矩阵求导

$$\frac{\partial \mathbf{a^T X b}}{\partial \mathbf{X}} = \mathbf{ab^T} \tag{21}$$

$$\frac{\partial \mathbf{a^T X^T b}}{\partial \mathbf{X}} = \mathbf{ba^T} \tag{22}$$

$$\partial \mathbf{X^T} = (\partial \mathbf{X})^\mathbf{T} \tag{23}$$

$$\partial(\mathrm{Tr}(\mathbf{X})) = \mathrm{Tr}(\partial \mathbf{X}) \tag{24}$$

$$\tag{25}$$

2 Information Theory

2.1 Differential Entropy

2.1.1 常见 Differential Entropy

- *Uniform distribution(from 0 to a)*

$$h(X) = \log(a)$$

- *Normal Distribution*

$$X \sim N(0, \sigma^2)$$

$$h(X) = \frac{1}{2} \log 2\pi e\sigma^2$$

- *Multivariate Normal Distribution*

$$N_n \sim (\mu, K)$$

μ is mean and K is covariance matrix

$$h(X_1, \ldots, X_n) = \frac{1}{2} \log(2\pi e)^n |K|$$

2.1.2 Properties

- $h(X + c) = h(X)$
- $h(aX) = h(X) + \log |a|$
- $h(AX) = h(X) + \log \left|\det(A)\right|$

3 Machine Learning

3.1 MAXIMUM LIKELIHOOD ESTIMATION (MLE)

3.1.1 GAUSSIAN MLE

$$\hat{\mu}_{ML} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\Sigma}_{ML} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu}_{ML})(x_i - \hat{\mu}_{ML})^T$$

3.2 Linear Regression

3.2.1 Least Squares

Usually, for linear regression (and classification) we include an intercept term w_0 that doesn't interact with any element in the vector x. It will be convenient to attach a 1 to the first dimension of each vector x_i.

$$x_i = \begin{bmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{id} \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1d} \\ 1 & x_{21} & \dots & x_{2d} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \dots & x_{nd} \end{bmatrix}$$

这种情况下，得到的解为:

$$w_{\text{ML}} = (X^T X)^{-1} X^T y$$

预测新的点为:

$$y_{\text{new}} = x_{\text{new}}^T w_{\text{ML}}$$

3.3 Classification

3.3.1 Bayes Classifier

The Bayes classifier has the smallest prediction error of all classifiers.
假设 (X, Y) 独立同分布，那么 optimal classifier is:

$$f(x) = \arg\max_{y \in \mathcal{Y}} P(Y = y | X = x) \tag{26}$$

Using Bayes rule and ignoring $P(X = x)$ we equivalently have:

$$f(x) = \arg\max_{y \in \mathcal{Y}} P(Y = y) \times P(X = x | Y = y) \tag{27}$$

其中 $P(Y = y)$ 叫做 class prior，$P(X = x|Y = y)$ 叫做 data likelihood given class.

3.3.2 THE PERCEPTRON ALGORITHM

- Suppose there is a linear classifier with zero training error:

$$y_i = \text{sign}(x_i^T w)$$

Then the data is linearly 'separable'

- By using the linear classifier $y = \text{sign}(x^T w)$ the Perceptron seeks to minimize

$$\mathcal{L} = -\sum_{i=1}^{n} (y_i \cdot x_i^T w) \mathbb{1}\{y_i \neq \text{sign}(x_i^T w)\}$$

Because $y \in \{-1, +1\}$,

$$y_i \cdot x_i^T w \quad \text{is} \quad \begin{cases} > 0 & y_i = x_i^T w \\ < 0 & y_i \neq x_i^T w \end{cases}$$

By minimizing $\mathcal{L}$ we're trying to always predict the correct label.

- $\nabla_w \mathcal{L} = 0$ 没有办法直接解出来，所以使用 gradient descent 的方法迭代求解 ($\mathcal{M}_t$ 表示在第 t 步被分错类的数据下标的集合)：

$$\nabla_w \mathcal{L} = \sum_{i \in \mathcal{M}_t} -y_i x_i \tag{28}$$

$$w' \leftarrow w - \eta \nabla_w \mathcal{L} \tag{29}$$

$$\mathcal{L}(w') < \mathcal{L}(w) \tag{30}$$

- Perceptron 算法的一些问题：
When the data is separable, there are an infinite number of hyperplanes.
This algorithm. doesn't take "quality"into consideration. It converges to the first one it finds.
When the data isn't separable, the algorithm doesn't converge. The hyperplane of w is always moving around.

3.3.3 LOGISTIC REGRESSION

- The distance of x to a hyperplane $x^T w + w_0$ is:

$$\Big| \frac{x^T w}{||w||_2} + \frac{w_0}{||w||_2} \Big|$$

- 鉴于 hyperplane 与 log odds 有很多相似的性质，我们可以直接用 hyperplane 来表示 log odds(注意，与前面 Bayes 得到的公式不同的是，前面是通过先验概率和 class prior 算出来的 w,w_0，这里我们对于这些先验概率没有约束 Discriminative classifier)：

$$\ln \frac{P(y = +1|x)}{P(y = -1|x)} = x^T w + w_0$$

$$P(y=+1|x)=\frac{\exp(x^Tw+w_0)}{1+\exp(x^Tw+w_0)}=\sigma(x^Tw+w_0)$$

其中 σ 叫做 *sigmoid function*, in this case x^Tw+w_0 is the *link function* for the log odds.

- 与 linear regression 类似，我们可以通过给 x 加入一维全为 1 的数据，将 w_0 并入到 w ;

$$w\leftarrow\begin{bmatrix}w_0\\w\end{bmatrix},\quad x\leftarrow\begin{bmatrix}1\\x\end{bmatrix}$$

$$x^Tw+w_0\leftarrow x^Tw$$

- Data likelihood
  $y_1,\ldots,y_n$ 的联合分布就是把每个的概率乘起来 (默认 $P(Y=y_i|X=x_i)$ 独立同分布)，对联合分布取对数的到目标函数 $\mathcal{L}$，算法的目标是找到参数 w 使得目标函数值最大 (实际上就是 Maximum likelihood)

$$\mathcal{L}=\sum_{i=1}^{n}\ln\sigma_i(y_i\cdot w)$$

$$w_{ML}=\arg\max_w\mathcal{L}$$

- 与 Perceptron 算法类似，使用梯度进行极值的计算 (其中 $\sigma_i(y_i\cdot w)$ 表示 $P(Y=y_i|X=x_i)$)

$$w^{(t+1)}=w^t+\eta\nabla_w\mathcal{L},\qquad\nabla_w\mathcal{L}=\sum_{i=1}^{n}\{1-\sigma_i(y_i\cdot w)\}y_ix_i$$

  有时为了避免“over fitting” 使用下述公式:

$$\mathcal{L}=\sum_{i=1}^{n}\ln\sigma_i(y_i\cdot w)-\lambda w^Tw$$

- 上述算法叫做 *steepest ascent*, 还有一种 Newton 算法利用的是 $\mathcal{L}$ 的二阶导数, 参见 lecture_9 第 14 页

- Laplace approximation for logistic regression 参见 lecture_9 第 24 页