# Linear Classification 01 Introduction

Chen Gong

29 October 2019

本节的主要目的是，有关于机器学习的导图。对频率派的有关统计学习方法做一个大致的梳理。而在贝叶斯派的学习中，是使用有关于概率图的模型。在频率派的有关统计学习方法中，我们可以大致的分为，线性回归和线性分类。

## 1 线性回归

在前文中已经提到了，我们的线性回归模型可以写为 $f(w,b) = w^T x + b$。线性回归主要有三条性质：线性，全局性和数据未加工。而我们从每一条入手，打破其中的一条规则就是一个新的算法。

### 1.1 线性

线性可以分为，属性非线性，全局非线性和系数非线性。

#### 1.1.1 属性非线性

所谓的属性非线性也就是从未知数入手，比如特征变换的方法还有将变量从一维，变换到高维。有点类似于引入二次型的思想，使用 $x_1^2 + x_2^2 + x_1 x_2 + \cdots$，的方法打破属性的线性。

#### 1.1.2 全局非线性

全局非线性的方法，是通过对函数的运算结果增加一个函数，来将线性函数改造成非线性函数。比如，神经网络中的激活函数，还有阈值函数来将软分类函数变成硬分类函数。

#### 1.1.3 系数非线性

所谓系数非线性，感觉就是系数的生成结果并不是单一的，固定的。就像神经网络算法一样。算法的收敛结果是一个分布，也就是位于一个区间之中，这样的算法的结果一定不是线性的，这样通过了不确定的方法来引入非线性。

### 1.2 全局性

所谓全局性，也就是将所有的数据看成一个整体来进行拟合。而打破的方法很简单，也就是将数据之间分隔开，分段进行拟合。典型的方法有线性样条回归，决策树等方法。

### 1.3 数据未加工

从字面的意义上理解非常的简单，那就是输入数据不经过加工直接的输入模型中。有一系列类似的方法来打破，比如主成分分析法 (PCA)，流形等方法来对输入数据进行预处理。

## 2 线性分类

线性回归和线性分类之间有着很大的联系。从某种意义上说，线性分类就是线性回归函数使用激活函数的结果，同时也可以看成是线性回归降维的结果。对于一个线性回归函数，我们可以通过添加全局函数的形式来将其转换为线性分类函数。也就是

$$y = w^T x + b \longrightarrow y = f(w^T x + b) \tag{1}$$

这样就可以将值域从 $[0,1]$ 转换为 $\{0,1\}$。其中 $f$ 被定义为 activation function，$f^{-1}$ 定义为 link function。那么这个 $f$ 实现了这样一个功能，也就是将 $w^T x + b \mapsto \{0,1\}$。而 $f^{-1}$ 恰好是反过来的，也就是将 $\{0,1\} \mapsto w^T x + b$。

而线性分类，大致上可以划分成硬分类和软分类两个部分。

### 2.1 硬分类

所谓硬分类，也就是 $y \in [0,1]$，大致上可以分成线性判别分析，也就是 Fisher 判别分析和感知机这两类。

### 2.2 软分类

所谓硬分类，也就是 $y \in \{0,1\}$，大致上可以分成生成式模型，Gaussian Discriminate Analysis 和著名的判别式模型，Logistic Regression。

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y) \tag{2}$$

也就是在求解 $p(y=0|x)$ 或 $p(y=1|x)$ 的时候，我们不直接求谁大谁小，而是转向求 $p(x|y=0)p(y=0)$ 和 $p(x|y=1)p(y=1)$，即求联合概率。

## 3 总结

通过这节的学习，我们已经大体上建立了有关于统计学习方法的知识的框架，包括线性分类和线性回归的内容，并作出了一定的梳理。

# Linear Classification 02 Perceptron

Chen Gong

30 October 2019

本节的主要内容是描述两类硬分类模型，也就是感知机模型和线性判别模型 (Fisher 判别模型) 的算法原理和推导过程。

## 1 感知机模型

感知机模型是一类错误驱动的模型，它的中心思想也就是" 错误驱动"。什么意思呢？也就是哪些数据点分类错误了，那么我们就进行调整权值系数 $w$，直到分类正确为止。

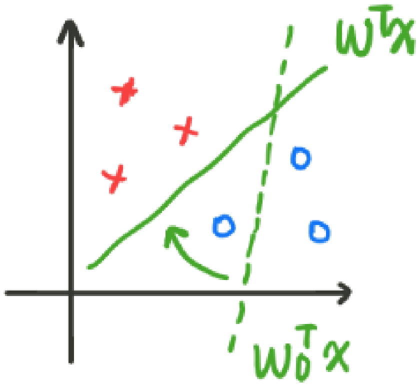


图 1: 感知机概念模型图

感知机可以做如下的描述：

$$f(x) = sign\{w^T x\} \quad x \in \mathbb{R}^p \ w \in \mathbb{R}^p \tag{1}$$

$$sign(a) = \begin{cases} +1 & a \geq 0 \\ -1 & a < 0 \end{cases} \tag{2}$$

其中 $D:$ { 被错误分类的样本 }，样本集为：$\{(x_i, y_i)\}_{i=1}^N$。

### 1.1 感知机模型的迭代过程

我们将损失函数定义为：

$$\mathcal{L}(w) = \sum_{i=1}^N I\left\{y_i w^T x_i < 0\right\} \tag{3}$$

而其中 $y_i w^T x_i < 0$ 就代表分类错误的类，为什么这么理解呢？因为：

$$\begin{cases} w^T x_i \geq 0 & y_i = +1 \\ w^T x_i < 0 & y_i = -1 \end{cases} \tag{4}$$

那么当分类正确时，必然有 $w^T x_i y_i > 0$。只有当错误分类的时候，才会出现 $w^T x_i y_i < 0$ 的情况。而在上述的函数中，$I$ 干了一个什么事，那就是将函数的值离散化，令 $\mathcal{L}$ 的值等于错误分类的点的个数，也就是这样一个映射 $I \mapsto 0, 1$。加这个函数的目的是得到损失函数的值，和普通的梯度下降法的过程一样。显然这不是一个连续的函数，无法求得其梯度来进行迭代更新。那么，我们需要想的办法是将离散的梯度连续。那么，我们将损失函数改写为：

$$\mathcal{L}(w) = \sum_{x_i \in D} -y_i w^T x_i \tag{5}$$

那么，梯度可以表示为:

$$\nabla_w \mathcal{L} = -\sum_{x_i \in D} y_i x_i \tag{6}$$

很显然，有关于 $w$ 的迭代公式，可以表示为：

$$w^{(t+1)} \longleftrightarrow w^{(t)} - \lambda \nabla_w L \tag{7}$$

代入可得，权值参数 $w$ 的更新过程为：

$$w^{(t+1)} \longleftrightarrow w^{(t)} + \lambda \sum_{x_i \in D} y_i x_i \tag{8}$$

那么，通过上述的推导，我们就得到了感知机中 $w$ 的更新过程。那么，感知机算法的推导过程就已经完成了。

# Linear classification 03 LDA

Chen Gong

31 October 2019

本小节为线性分类的第三小节，主要推导了线性判别分析算法，也就是 Fisher 算法。Fisher 算法的主要思想是：类内小，类间大。这有点类似于，软件过程里的松耦合，高内聚的思想。这个思想转换成数学语言也就是，同一类数据之间的方差要小，不同类数据之间的均值的差距要大。那么，我们对数据的描述如下所示：

$$X = (x_1, x_2, \cdots, x_N)^T = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{32} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Np} \end{pmatrix}_{N\times P} \tag{1}$$

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}_{N\times 1} \tag{2}$$

那么，我们的数据集可以记为 $\{(x_i, y_i)\}_{i=1}^N$，其中，$x_i \in \mathbb{R}^p$，$y_i \in \{+1, -1\}$，且 $\{y = +1\}$ 为 $C_1$ 类，且 $\{y = -1\}$ 为 $C_2$ 类。那么，$X_{c_1}$ 被定义为 $\{x_i|y_i = +1\}$，$X_{c_2}$ 被定义为 $\{x_i|y_i = -1\}$。所以，很显然可以得到 $|X_{c_1}| = N_1$，$|X_{c_2}| = N_2$，并且 $N_1 + N_2 = N$。

## 1 Fisher 线性判别分析

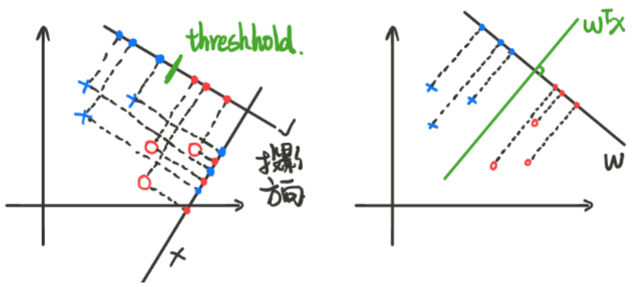


图 1: Fisher 线性判别分析模型图

在左图中，我们设置了两个投影方向，很显然上面那个投影方向可以更好的将两个点之间分开。我们可以在投影方向上找一个点作为两个类别的分界点，也就是阈值 (Threshhold)。首先，我们先引入

一个有关投影算法的小知识。

### 1.1 投影算法

首先，我们需要设定一个投影向量 $w$，为了保险起见，对这个投影向量 $w$ 作出约束，令 $||w|| = 1$。那么，在空间中的一个数据点，也就是一个向量，在投影向量上的投影长度可以表述为：

$$x_i \cdot w = |x_i||w|\cos\theta = |x_i|\cos\theta = \triangle \tag{3}$$

### 1.2 Fisher 判别分析的损失函数表达式

在这个部分，主要是要得出 Fisher 判别分析的损失函数表达式求法。对于，投影的平均值和方差，我们可以分别表述为:

$$\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i = \frac{1}{N}\sum_{i=1}^{N} w^T x_i \tag{4}$$

$$S_z = \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})(z_i - \bar{z})^T \tag{5}$$

那么对于第一类分类点 $X_{c_1}$ 和第二类分类点 $X_{c_2}$ 可以表述为：

$$C_1: \qquad \bar{z_1} = \frac{1}{N_1}\sum_{i=1}^{N_1} w^T x_i \qquad S_1 = \frac{1}{N_1}\sum_{i=1}^{N}(z_i - \bar{z_1})(z_i - \bar{z_1})^T \tag{6}$$

$$C_2: \qquad \bar{z_2} = \frac{1}{N_2}\sum_{i=1}^{N_2} w^T x_i \qquad S_2 = \frac{1}{N_2}\sum_{i=1}^{N}(z_i - \bar{z_2})(z_i - \bar{z_2})^T \tag{7}$$

那么类间的距离我们可以定义为：$(\bar{z_1} - \bar{z_2})^2$, 类内的距离被我们定义为 $S_1 + S_2$。那么我们的目标函数 Target Function $\mathcal{J}(w)$，可以被定义为:

$$\mathcal{J}(w) = \frac{(\bar{z_1} - \bar{z_2})^2}{S_1 + S_2} \tag{8}$$

因为，我们的目的是使方差越小越好，均值之间的差越大越好。

### 1.3 损失函数表达式的化简

#### 1.3.1 $(\bar{z_1} - \bar{z_2})^2$

分子的化简过程如下所示：

$$\begin{aligned}
(\bar{z_1} - \bar{z_2})^2 &= \left(\frac{1}{N_1}\sum_{i=1}^{N_1} w^T x_i - \frac{1}{N_2}\sum_{i=1}^{N_2} w^T x_i\right)^2 \\
&= \left(w^T(\frac{1}{N_1}\sum_{i=1}^{N_1} x_i - \frac{1}{N_2}\sum_{i=1}^{N_2} x_i)\right)^2 \\
&= \left(w^T(\bar{X}_{c_1} - \bar{X}_{c_2})\right)^2 \\
&= w^T(\bar{X}_{c_1} - \bar{X}_{c_2})(\bar{X}_{c_1} - \bar{X}_{c_2})^T w
\end{aligned} \tag{9}$$

### 1.3.2 $S_1 + S_2$

分母的化简过程如下所示：

$$\begin{aligned} S_1 =& \frac{1}{N_1}\sum_{i=1}^{N}(z_i - \bar{z}_1)(z_i - \bar{z}_1)^T \\ =& \frac{1}{N_1}\sum_{i=1}^{N}(w^T x_i - \frac{1}{N_1}\sum_{i=1}^{N_1} w^T x_i)(w^T x_i - \frac{1}{N_1}\sum_{i=1}^{N_1} w^T x_i)^T \\ =& w^T \frac{1}{N_1}\sum_{i=1}^{N}(x_i - \frac{1}{N_1}\sum_{i=1}^{N_1} x_i)(x_i - \frac{1}{N_1}\sum_{i=1}^{N_1} x_i)^T w \\ =& w^T S_{c_1} w \end{aligned} \tag{10}$$

同理可得，

$$S_1 = w^T S_{c_2} w \tag{11}$$

所以，

$$S_1 + S_2 = w^T (S_{c_1} + S_{c_2}) w \tag{12}$$

### 1.3.3 $\mathcal{J}(w)$ 的最简表达形式

$$\mathcal{J}(w) = \frac{w^T(\bar{X}_{c_1} - \bar{X}_{c_2})(\bar{X}_{c_1} - \bar{X}_{c_2})^T w}{w^T(S_{c_1} + S_{c_2})w} \tag{13}$$

令 $S_b$ 为 between-class 类间方差，$S_w$ 为 within-class，也就是类内方差。那么有

$$S_b = (\bar{X}_{c_1} - \bar{X}_{c_2})(\bar{X}_{c_1} - \bar{X}_{c_2})^T \qquad S_w = (S_{c_1} + S_{c_2}) \tag{14}$$

于是，我们可以得到进一步化简的表达式；

$$\mathcal{J}(w) = \frac{w^T S_b w}{w^T S_w w} \tag{15}$$

## 1.4 损失函数 $\mathcal{J}(w)$ 的梯度

为了方便求导，我们令 $\mathcal{J}(w) = (w^T S_b w)(w^T S_w w)^{-1}$。

$$\begin{aligned} \frac{\partial \mathcal{J}(w)}{\partial w} = 2S_b w(w^T S_w w)^{-1} + & (-1)(w^T S_b w)(w^T S_w w)^{-2} \cdot 2S_w w = 0 \\ S_b w (w^T S_w w)^{-1} =& (w^T S_b w)(w^T S_w w)^{-2} S_w w \end{aligned} \tag{16}$$

显然，$w$ 的维度是 $p \times 1$，$w^T$ 的维度是 $1 \times p$，$S_w$ 的维度是 $p \times p$，所以，$w^T S_w w$ 是一个实数，同理可得，$w^T S_w w$ 是一个实数所以，可以得到

$$S_b w = \frac{(w^T S_b w)}{(w^T S_w w)} S_w w \tag{17}$$

我们主要是需要求 $w$ 的方向，大小不是很重要了。并且根据我们的定义，$w^T S_b w$ 和 $w^T S_w w$ 都是正的。所以，我们可得

$$w = \frac{(w^T S_b w)}{(w^T S_w w)} S_b^{-1} S_w w \propto S_b^{-1} S_w w \tag{18}$$

$$S_w w = (\bar{X}_{c_1} - \bar{X}_{c_2})(\bar{X}_{c_1} - \bar{X}_{c_2})^T w \tag{19}$$

而 $(\bar{X}_{c_1} - \bar{X}_{c_2})^T w$ 是一个实数，不会改变 $w$ 的方向，所以汇总可得:

$$S_b^{-1} S_w w \propto S_w^{-1}(\bar{X}_{c_1} - \bar{X}_{c_2}) \tag{20}$$

那么，我们就可以求得 $w$ 的方向为 $S_w^{-1}(\bar{X}_{c_1} - \bar{X}_{c_2})$。如果，$S_w^{-1}$ 是一个各向同性的对角矩阵，那么 $S^{-1} \propto I$。所以，$w \propto (\bar{X}_{c_1} - \bar{X}_{c_2})$。既然，求得了 $w$ 的方向，其实 $w$ 的大小就不重要的。

# Linear Classification 04 Logistic Regression

Chen Gong

1 November 2019

在前面的两小节中我们, 我们讨论了有关于线性分类问题中的硬分类问题, 也就是感知机和 Fisher 线性判别分析。那么，我们接下来的部分需要讲讲软分类问题。软分类问题，可以大体上分为概率判别模型和概率生成模型，概率生成模型也就是高斯判别分析 (Gaussian Discriminate Analysis)，朴素贝叶斯 (Naive Bayes)。而线性判别模型也就是本章需要讲述的重点，Logistic Regression。

## 1 从线性回归到线性分类

线性回归的问题, 我们可以看成这样一个形式, 也就是 $w^Tx$。而线性分类的问题可以看成是 $\{0,1\}$ 或者 $[0,1]$ 的问题。其实，从从线性回归到线性分类之间通过一个映射，也就是 Activate Function 来实现的，通过这个映射我们可以实现 $w^Tx \longmapsto \{0,1\}$。

而在 Logistic Regression 中，我们将激活函数定义为:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{1}$$

那么很显然会有如下的性质:

1.$\lim_{z\longrightarrow +\infty} \sigma(z) = 1$

2.$\lim_{z\longrightarrow 0} \sigma(z) = \frac{1}{2}$

3.$\lim_{z\longrightarrow -\infty} \sigma(z) = 0$

那么，通过这样一个激活函数 $\sigma$，我们就可以将实现 $\mathbb{R} \longrightarrow (0,1)$。那么我们会得到以下的表达式:

$$p(y|x) = \begin{cases} p_1 = p(y=1|x) = \sigma(w^Tx) = \frac{1}{1+exp\{-w^tx\}} & y=1 \\ p_2 = p(y=0|x) = 1-p(y=1|x) = \frac{exp\{-w^tx\}}{1+exp\{-w^tx\}} & y=0 \end{cases} \tag{2}$$

而且，我们可以想一个办法来将两个表达式合二为一，那么有:

$$p(y|x) = p_1^y \cdot p_0^{1-y} \tag{3}$$

## 2 最大后验估计

$$
\begin{aligned}
MLE = & \hat{w} = \arg\max_{w} \log p(y|x) \\
& = \arg\max_{w} \log p(y_i|x_i) \\
& = \arg\max_{w} \sum_{i=1}^{N} \log p(y_i|x_i) \\
& = \arg\max_{w} \sum_{i=1}^{N} y \log p_1 + (1-y) \log p_2
\end{aligned} \tag{4}
$$

我们令，

$$
\frac{1}{1+exp\{-w^T x\}} = \varphi(x,w) \qquad \frac{exp\{-w^T x\}}{1+exp\{-w^T x\}} = 1-\varphi(x,w) \tag{5}
$$

那么，

$$
MLE = argmax_w \sum_{i=1}^{N} y \log \varphi(x,w) + (1-y) \log(1-\varphi(x,w)) \tag{6}
$$

实际上 $y \log \varphi(x,w) + (1-y) \log(1-\varphi(x,w))$ 就是一个交叉熵 (Cross Entropy)。那么，我们成功的找到了我们的优化目标函数，可以表述为 MLE (max) $\longrightarrow$ Loss function (Min Cross Entropy)。所以，这个优化问题就转换成了一个 Cross Entropy 的优化问题，这样的方法就很多了。

交叉熵是用来衡量两个分布的相似程度的，通过如下公式进行计算，其中 $p(x)$ 为真实分布，$q(x)$ 为预测分布：

$$
H(p,q) = \sum_{x} -p(x) log\, q(x) \tag{7}
$$

$$
H(p,q) = \int_{x} -p(x) log\, q(x)\, dx = \mathbb{E}_{x\sim p(x)}[-log\, q(x)] \tag{8}
$$

# Linear Classification 05 Gaussian Discriminate Analysis

Chen Gong

03 November 2019

前面讲的方法都是概率判别模型，包括，Logistic Regression 和 Fisher 判别分析。接下来我们将要学习的是概率生成模型部分，也就是现在讲到的 Gaussian Discriminate Analysis。数据集的相关定义为：

$$X = (x_1, x_2, \cdots, x_N)^T = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{32} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Np} \end{pmatrix}_{N\times P} \tag{1}$$

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}_{N\times 1} \tag{2}$$

那么，我们的数据集可以记为 $\{(x_i, y_i)\}_{i=1}^N$，其中，$x_i \in \mathbb{R}^p$，$y_i \in \{+1, -1\}$。我们将样本点分成了两个部分：

$$\left\{ \begin{array}{l} C_1 = \{x_i | y_i = 1, i = 1, 2, \cdots, N_1\} \\ C_2 = \{x_i | y_i = 0, i = 1, 2, \cdots, N_2\} \end{array} \right. \tag{3}$$

并且有 $|C_1| = N_1$，$|C_2| = N_2$，且 $N_1 + N_2 = N$。

## 1 概率判别模型与生成模型的区别

什么是判别模型？所谓判别模型，也就是求

$$\hat{y} = \arg\max_y \; p(y|x) \qquad y \in \{0, 1\} \tag{4}$$

重点在于求出这个概率来，知道这个概率的值等于多少。而概率生成模型则完全不一样。概率生成模型不需要知道概率值具体是多大，只需要知道谁大谁小即可，具体是对联合概率进行建模。举例即为 $p(y=0|x)$ 和 $p(y=1|x)$，谁大谁小的问题。而概率生成模型的求法可以用贝叶斯公式来进行求解，即为：

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x, y)}{p(x)} \propto p(x, y) \tag{5}$$

因为在这个公式中，比例大小 $p(x)$ 与 $y$ 的取值无关，所以它是一个定值。所以，概率生成模型实际上关注的就是一个求联合概率分布的问题。那么，总结一下

$$p(y|x) \propto p(x|y)p(y) \propto p(x,y) \tag{6}$$

其中，$p(y|x)$ 为 Posterior function，$p(y)$ 为 Prior function，p(x|y) 为 Likelihood function。所以有

$$\hat{y} = \arg\max_{y\in\{0,1\}} p(y|x) \propto \arg\max_{y\in\{0,1\}} p(x|y)p(y) \tag{7}$$

# 2 Gaussian Discriminate Analysis 模型建立

在二分类问题中，很显然可以得到，我们的**先验概率**符合，$p(y)$ ∼Bernoulli Distribution。也就是，

| $y$ | 1 | 0 |
|---|---|---|
| $p$ | $\varphi$ | $1-\varphi$ |

表 1: Bernoulli 分布的概率分布表

所以，可以写出：

$$p(y) = \begin{cases} \varphi^y & y=1 \\ (1-\varphi)^{1-y} & y=0 \end{cases} \Rightarrow \varphi^y(1-\varphi)^{1-y} \tag{8}$$

而随后是要确定**似然函数**，我们假设他们都符合高斯分布。对于不同的分类均值是不同的，但是不同变量之间的协方差矩阵是一样的。那么我们可以写出如下的形式：

$$p(x|y) = \begin{cases} p(x|y=1) \sim \mathcal{N}(\mu_1,\Sigma) \\ p(x|y=0) \sim \mathcal{N}(\mu_2,\Sigma) \end{cases} \Rightarrow \mathcal{N}(\mu_1,\Sigma)^y \mathcal{N}(\mu_2,\Sigma)^{1-y} \tag{9}$$

那么我们的 Likelihood function 可以被定义为：

$$\begin{aligned}
\mathcal{L}(\theta) &= \log \prod_{i=1}^N p(x_i,y_i) \\
&= \sum_{i=1}^N \log p(x_i,y_i) \\
&= \sum_{i=1}^N \log p(x_i|y_i)p(y_i) \\
&= \sum_{i=1}^N [\log p(x_i|y_i) + \log p(y_i)] \\
&= \sum_{i=1}^N \left[\log \mathcal{N}(\mu_1,\Sigma)^{y_i}\mathcal{N}(\mu_2,\Sigma)^{1-y_i} + \log \varphi_i^y (1-\varphi)^{1-y_i}\right] \\
&= \sum_{i=1}^N \log \mathcal{N}(\mu_1,\Sigma)^{y_i} + \sum_{i=1}^N \log \mathcal{N}(\mu_2,\Sigma)^{1-y_i} + \sum_{i=1}^N \log \varphi^{y_i} + \sum_{i=1}^N \log(1-\varphi)^{1-y_i}
\end{aligned} \tag{10}$$

为了方便后续的推演过程，所以，我们将 Likelihood function 写成，

$$\mathcal{L}(\theta) = ①+②+⑧$$

并且，我们令：$① = \sum_{i=1}^N \log \mathcal{N}(\mu_1, \Sigma)_i^y$，$② = \sum_{i=1}^N \log \mathcal{N}(\mu_2, \Sigma)^{1-y_i}$，$③ = \sum_{i=1}^N \log \varphi^{y_i} + \sum_{i=1}^N \log(1-\varphi)^{1-y_i}$。那么上述函数我们可以表示为：

$$\theta = (\mu_1, \mu_2, \Sigma, \varphi) \qquad \hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta) \tag{11}$$

# 3 Likelihood function 参数的极大后验估计

Likelihood function 的参数为 $\theta = (\mu_1, \mu_2, \Sigma, \varphi)$，下面我们分别用极大似然估计对这四个参数进行求解。下面引入几个公式：

$$tr(AB) = tr(BA) \tag{12}$$

$$\frac{\partial tr(AB)}{\partial A} = B^T \tag{13}$$

$$\frac{\partial |A|}{\partial A} = |A|A^{-T} \tag{14}$$

$$\frac{\partial \log |A|}{\partial A} = A^{-T} \tag{15}$$

## 3.1 求解 $\varphi$

$$③ = \sum_{i=1}^N \log \varphi^{y_i} + \sum_{i=1}^N \log(1-\varphi)^{1-y_i} = \sum_{i=1}^N y_i \log \varphi + \sum_{i=1}^N (1-y_i) \log(1-\varphi)$$

$$\frac{\partial ③}{\partial \varphi} = \sum_{i=1}^N y_i \frac{1}{\varphi} - \sum_{i=1}^N (1-y_i) \frac{1}{1-\varphi} = 0 \tag{16}$$

$$\sum_{i=1}^N y_i(1-\varphi) - (1-y_i)\varphi = 0 \tag{17}$$

$$\sum_{i=1}^N (y_i - \varphi) = 0 \tag{18}$$

$$\hat{\varphi} = \frac{1}{N} \sum_{i=1}^N y_i \tag{19}$$

又因为 $y_i = 0$ 或 $y_i = 1$，所以 $\hat{\varphi} = \frac{1}{N} \sum_{i=1}^N y_i = \frac{N_1}{N}$。

## 3.2 求解 $\mu_1$

$$\begin{aligned} ① &= \sum_{i=1}^N \log \mathcal{N}(\mu_1, \Sigma)^{y_i} \\ &= \sum_{i=1}^N y_i \log \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}(x_i - \mu_1)^T \Sigma^{-1} (x_i - \mu_1)\right\} \end{aligned}$$

那么求解过程如下所示：由于到对 $\mu_1$ 求偏导，我们只需要关注公式中和 $\mu_1$ 有关的部分。那么我们可以将问题简化为：

$$\max_{\mu_1} \sum_{i=1}^N y_i \log exp\left\{-\frac{1}{2}(x_i - \mu_1)^T \Sigma^{-1} (x_i - \mu_1)\right\} \tag{20}$$

然后将 $exp$ 和 $\log$ 抵消掉，再将括号打开，我们可以得到最终的化简形式：

$$\max_{\mu_1} -\frac{1}{2}\sum_{i=1}^{N} y_i \left\{x_i^T\Sigma^{-1}x_i - 2\mu_1^T\Sigma^{-1}x_i + \mu_1^T\Sigma^{-1}\mu_1\right\} \tag{21}$$

为了方便表示，我们令① = △。所以，极大似然法求解过程如下：

$$\begin{aligned}
\frac{\partial\triangle}{\partial\mu_1} =& -\frac{1}{2}\sum_{i=1}^{N} y_i(-2\Sigma^{-1}x_i + 2\Sigma^{-1}\mu_1) = 0 \\
=& \sum_{i=1}^{N} y_i(\Sigma^{-1}x_i - \Sigma^{-1}\mu_1) = 0 \\
=& \sum_{i=1}^{N} y_i(x_i - \mu_1) = 0 \\
=& \sum_{i=1}^{N} y_i x_i = \sum_{i=1}^{N} y_i\mu_1 \\
\mu_1 =& \frac{\sum_{i=1}^{N} y_i x_i}{\sum_{i=1}^{N} y_i} = \frac{\sum_{i=1}^{N} y_i x_i}{N_1}
\end{aligned} \tag{22}$$

### 3.3 求解 $\mu_2$

$\mu_2$ 的求解过程与 $\mu_1$ 的基本保持一致性。区别点从公式 (22) 开始，我们有：

$$\max_{\mu_2} -\frac{1}{2}\sum_{i=1}^{N} (1-y_i) \left\{x_i^T\Sigma^{-1}x_i - 2\mu_2^T\Sigma^{-1}x_i + \mu_2^T\Sigma^{-1}\mu_2\right\} \tag{23}$$

极大似然法的求解过程如下所示：

$$\begin{aligned}
\frac{\partial\triangle}{\partial\mu_2} =& -\frac{1}{2}\sum_{i=1}^{N} (1-y_i)(-2\Sigma^{-1}x_i + 2\Sigma^{-1}\mu_2) = 0 \\
=& \sum_{i=1}^{N} (1-y_i)(x_i - \mu_2) = 0 \\
=& \sum_{i=1}^{N} x_i - \sum_{i=1}^{N} y_i x_i = N\mu_2 - \sum_{i=1}^{N} y_i\mu_2 \\
\mu_2 =& \frac{\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} y_i x_i}{N - \sum_{i=1}^{N} y_i} = \frac{\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} y_i x_i}{N - N_1} \\
=& \frac{\sum_{i=1}^{N}(1-y_i)x_i}{N_2}
\end{aligned} \tag{24}$$

也可以对于求 $\mu_1$ 来说，求 $\mu_2$ 可以类比，将其中的 $N_1$ 换成 $N_2$，其中的 $y_i$ 换成 $1-y_i$，可以得到同样的结果。

### 3.4 求解 $\Sigma$

如果要使用极大似然估计来求解 $\Sigma$，这只与 $\mathcal{L}(\theta)$ 中的①和②有关。并且①+②的表达式为：

$$\sum_{i=1}^{N} \log\mathcal{N}(\mu_1,\Sigma)^{y_i} + \sum_{i=1}^{N} \log\mathcal{N}(\mu_2,\Sigma)^{1-y_i} \tag{25}$$

那么，按照分类点的方法，我们可以将其改写为：

$$\hat{\Sigma} = \arg\min_{\Sigma} \sum_{x\in C_1} \log \mathcal{N}(\mu_1, \Sigma) + \sum_{x\in C_2} \log \mathcal{N}(\mu_2, \Sigma) \tag{26}$$

公式加号前后都是一样的，所以，为了方便计算我们暂时只考虑一半的计算：

$$\begin{aligned}
\sum_{i=1}^{N} \log \mathcal{N}(\mu, \Sigma) =& \sum_{i=1}^{N} \log \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} exp\left\{ -\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) \right\} \\
=& -\sum_{i=1}^{N} \frac{p}{2} \log 2\pi - \sum_{i=1}^{N} \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) \\
=& C - \frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) \\
=& C - \frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{N} tr\left((x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\right) \\
=& C - \frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{N} tr\left((x_i-\mu)(x_i-\mu)^T\Sigma^{-1}\right)
\end{aligned} \tag{27}$$

而且，

$$S = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T \tag{28}$$

所以，

$$\sum_{i=1}^{N} \log \mathcal{N}(\mu, \Sigma) = C - \frac{N}{2}\log|\Sigma| - \frac{N}{2}tr(S\Sigma^{-1}) \tag{29}$$

那么代入公式 (27) 中，我们可以得到：

$$\begin{aligned}
\hat{\Sigma} =& \arg\max_{\Sigma}\ C - \frac{N_1}{2}\log|\Sigma| - \frac{N_1}{2}tr(S_1\Sigma^{-1}) + C - \frac{N_2}{2}\log|\Sigma| - \frac{N_2}{2}tr(S_2\Sigma^{-1}) \\
=& \arg\max_{\Sigma}\ -\frac{N}{2}\log|\Sigma| - \frac{N_1}{2}tr(S_1\Sigma^{-1}) - \frac{N_2}{2}tr(S_2\Sigma^{-1}) \\
=& \arg\min_{\Sigma}\ N\log|\Sigma| + N_1 tr(S_1\Sigma^{-1}) + N_2 tr(S_2\Sigma^{-1})
\end{aligned} \tag{30}$$

我们令函数 $N\log|\Sigma| + N_1 tr(S_1\Sigma^{-1}) + N_2 tr(S_2\Sigma^{-1}) = \triangle$，那么对 $\Sigma$ 求偏导并令其等于 0 可得：

$$\frac{\partial\triangle}{\partial\Sigma} = N\Sigma^{-1} - N_1\Sigma^{-1}S_1\Sigma^{-1} - N_2\Sigma^{-1}S_2\Sigma^{-1} = 0 \tag{31}$$

对上式左乘 $\Sigma$，又乘 $\Sigma$ 得到 $N\Sigma - N_1S_1 - N_2S_2 = 0$。
解得：

$$\Sigma = \frac{N_1S_1 + N_2S_2}{N} \tag{32}$$

其中对 $tr(S_1\Sigma^{-1})$ 求偏导的过程如下（由于 $\Sigma\Sigma^{-1} = \mathbb{I}$，所以 $d\,(\Sigma^{-1}\Sigma) = \mathbb{O} \Rightarrow (d\,\Sigma)\Sigma^{-1} + \Sigma d\,(\Sigma^{-1}) = 0 \Rightarrow d\,\Sigma^{-1} = -\Sigma^{-1}(d\,\Sigma)\Sigma^{-1}$：

$$\begin{aligned}
d\,tr(S_1\Sigma^{-1}) =& tr(S_1 d\,\Sigma^{-1}) \\
=& tr(-S_1\Sigma^{-1}(d\,\Sigma)\Sigma^{-1}) \\
=& tr(-\Sigma^{-1}S_1\Sigma^{-1}d\,\Sigma)
\end{aligned} \tag{33}$$

于是 $\frac{\partial tr(S_1\Sigma^{-1})}{\partial\Sigma} = -\Sigma^{-1}S_1\Sigma^{-1}$。同理可以知道 $\frac{\partial tr(S_2\Sigma^{-1})}{\partial\Sigma} = -\Sigma^{-1}S_2\Sigma^{-1}$。

# 4 总结

下面对 Gaussian Discriminate Analysis 做一个简单的小结。我们使用模型为:

$$\hat{y} = \underset{y\in\{0,1\}}{\arg\max}\, p(y|x) \propto \underset{y\in\{0,1\}}{\arg\max}\, p(x|y)p(y) \tag{34}$$

$$\begin{cases} p(y) = \varphi^y(1-\varphi)^{1-y} \\ p(x|y) = \mathcal{N}(\mu_1, \Sigma)^y \mathcal{N}(\mu_2, \Sigma)^{1-y} \end{cases} \tag{35}$$

利用极大似然估计得到的结果为:

$$\theta = (\mu_1, \mu_2, \Sigma, \varphi) = \begin{cases} \hat{\varphi} = \frac{N_1}{N} \\ \mu_1 = \frac{\sum_{i=1}^N y_i x_i}{N_1} \\ \mu_2 = \frac{\sum_{i=1}^N (1-y_i) x_i}{N_2} \\ \Sigma = \frac{N_1 S_1 + N_2 S_2}{N} \end{cases} \tag{36}$$

# Linear Classification 06 Naive Bayes

Chen Gong

04 November 2019

本节主要是介绍一下 Naive Bayes Classification，也就是朴素贝叶斯分类。朴素贝叶斯分类器的核心思想也就是，条件独立性假设。这是一种最简单的概率图模型，也就是一种有向图模型。

## 1 条件独立性假设

条件独立性假设用简单的图来进行表述，可以表示为如下图所示的形式：

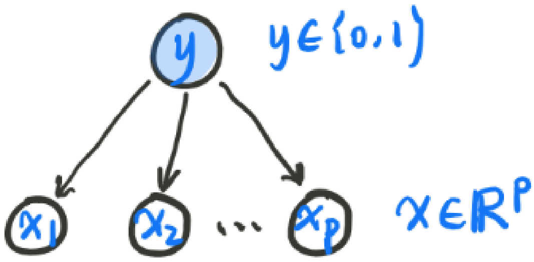


图 1: 条件独立性假设

我们可以将其定义为 $x_i \perp x_j | y \ (i \neq j)$。根据贝叶斯公式可以得：

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x,y)}{p(x)} \propto p(x,y) \tag{1}$$

而做条件独立性假设的最终目的，是为了简化运算。因为对于一个数据序列 $x = (x_1, x_2, \cdots, x_p)$。如果 $x_i$ 和 $x_j$ 之间有关系的话，这个计算难度可能会变得很难，所以就假设各个变量之间是相互独立的。而且，马尔可夫决策链也就是这样类似的思想。

## 2 Naive Bayes Classification

朴素贝叶斯算法的优化目的即为：

$$\begin{aligned} \hat{y} &= \arg\max_{y\in\{0,1\}} p(y|x) \\ &= \arg\max_{y\in\{0,1\}} p(x|y)p(y) \end{aligned} \tag{2}$$

其中，

$$p(x|y) = \prod_{i=1}^{N} p(x_i|y) \tag{3}$$

对于 $p(y)$ 这个先验概率密度函数的确定，对于二分类问题，也就是 $y \sim$Bernoulli Distribution，而对于多分类问题，先验概率为 $y \sim$Categorial Distribution。而对于，$p(x|y) = \prod_{i=1}^{N} p(x_i|y)$。如果 $x$ 是离散的，那么 $x_i|y \sim$Categorial Distribution；如果 $x$ 是连续的，那么 $x_i|y \sim \mathcal{N}(\mu_y, \Sigma_y^2)$。对于每一类都有一个高斯分布。

而有关于 $p(x|y)$ 用极大似然估计 MLE，估计出来就行。因为分布的形式我们已经知道了，那么只要利用数据来进行学习，使用极大似然估计就可以得到想要的结果了。其实对于多分类的情况，Naive Bayes Classification 和 Guassian Discriminate Analysis 很像的。