第三章、线性分类与广义线性模型

赵涵

2022 年 3 月 18 日

对于分类任务，线性回归模型就无能为力了，但是我们可以在线性模型的函数进行后再加入一层激活函数，这个函数是非线性的，激活函数的反函数叫做链接函数。我们有两种线性分类的方式：

1. 硬分类，我们直接需要输出观测对应的分类。这类模型的代表为：
   (a) 线性判别分析（Fisher 判别）
   (b) 感知机
2. 软分类，产生不同类别的概率，这类算法根据概率方法的不同分为两种：
   (a) 生成式（根据贝叶斯定理先计算参数后验，再进行推断）：
      i. 高斯判别分析
      ii. 朴素贝叶斯模型
   (b) 判别式（直接对条件概率进行建模）：logistic回归

下面的内容，我们主要讨论上面介绍的每个模型。

1 线性分类-硬分类

所谓的激活函数，就是对线性输出结果，做一个非线性变换，原则上只要是非线性函数，都可以作为激活函数，依照不同的模型要求，选择适当的激活函数，是建立模型的关键。

1.1 感知机模型

对于感知机模型，是我们类比大脑的神经元，在1960年代，感知机作为一个粗略的模型去解释大脑离神经元的工作机制。我们选择如下的激活函数：

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

我们把线性回归的结果，通过这个函数映射，就能实现二分类的效果了。

我们定义损失函数为分错的样本数目，比较直观的一种方式，是采用指示函数，但是指示函数不可导，因此可以定义：

$$J(\theta) = \sum_{\mathcal{D}_{wrong}} -y^{(i)}\theta^T x^{(i)} \tag{2}$$

其中，$\mathcal{D}_{wrong}$是错误分类集合，实际在每一次训练的时候，我们采用梯度下降的算法。损失函数对参数θ的梯度为：

$$\nabla_\theta J(\theta) = \frac{\partial}{\partial \theta} J(\theta) = \sum_{\mathcal{D}_{wrong}} -y^{(i)} x^{(i)} \tag{3}$$

但是如果样本非常多的情况下，计算复杂度较高，可以采用随机梯度下降或者小批量梯度下降，我们给出随机梯度下降的更新规则，每次从分错的样本中随机挑选一个：

$$\theta := \theta + \alpha y^{(i)} x^{(i)} \tag{4}$$

我们也把这种更新规则称为**感知机学习算法**（perceptron learning algorithm）。我们需要注意，感知机是我们模仿大脑里神经元的活动提出的模型，所以并没有概率基础，或者是似然函数这样的最大估计。

1.2 线性判别分析

在线性判别分析（Linear Discrimination Analysis）中，我们的基本思想是选定一个方向，将样本顺着这个方向投影，投影后的数据需要满足两个条件，从而可以更好的分类：

1. 相同类内部的样本距离接近。
2. 不同类别之间的距离较大。

总结来说，就是高内聚，低耦合。我们首先引入一个投影向量，假定样本是向量x，投影方向的向量为w，那么顺着w方向的投影就是标量：

$$z = w^T x = |w| \cdot |x| \cos\theta \tag{5}$$

最后面的等式，是在二维平面直角坐标系下的表达式。对于高内聚，我们首先假定属于两类的样本数量分别为N_1 和N_2。那么我们采用方差矩阵来表征每一个类内的总体分布，这里我们使用方差的定义，用S表示原始数据的方差，先看第一类的方差：

$$\begin{aligned}
Var_z[C_1] &= \frac{1}{N_1}\sum_{i=1}^{N_1}(z^{(i)} - \overline{z_{c1}})(z^{(i)} - \overline{z_{c1}})^T && (6)\\
&= \frac{1}{N_1}\sum_{i=1}^{N_1}(w^T x^{(i)} - \frac{1}{N_1}\sum_{j=1}^{N_1} w^T x^{(i)})(w^T x^{(i)} - \frac{1}{N_1}\sum_{j=1}^{N_1} w^T x^{(i)})^T && (7)\\
&= w^T \frac{1}{N_1}\sum_{i=1}^{N_1}(x^{(i)} - \overline{x_{c1}})(x^{(i)} - \overline{x_{c1}})^T w && (8)\\
&= w^T S_1 w && (9)
\end{aligned}$$

同理我们可以得到第二类的方差：

$$Var_z[C_2] = w^T S_2 w \tag{10}$$

所以类内距离可以记为：

$$Var_z[C_1] + Var_z[C_2] = w^T(S_1 + S_2)w \tag{11}$$

对于低耦合，我们可以采用两类的均值表示这个距离：

$$\begin{aligned}
(\overline{z_{c1}} - \overline{z_{c2}})^2 &= (\frac{1}{N_1}\sum_{i=1}^{N_1} w^T x^{(i)} - \frac{1}{N_2}\sum_{i=1}^{N_2} w^T x^{(i)})^2 && (12)\\
&= (w^T(\overline{x_{c1}} - \overline{x_{c2}}))^2 && (13)\\
&= w^T(\overline{x_{c1}} - \overline{x_{c2}})(\overline{x_{c1}} - \overline{x_{c2}})^T w && (14)
\end{aligned}$$

综合这两点，由于方差是一个矩阵，于是我们将这两个值相除来定义我们的损失函数，并最大化这个值：

$$\begin{aligned}\hat{w} &= \arg\max_w J(w) && (15)\\ &= \arg\max_w \frac{(\overline{z_{c1}} - \overline{z_{c2}})^2}{Var_z[C_1] + Var_z[C_2]} && (16)\\ &= \arg\max_w \frac{w^T(\overline{x_{c1}} - \overline{x_{c2}})(\overline{x_{c1}} - \overline{x_{c2}})^T w}{w^T(S_1 + S_2)w} && (17)\\ &= \arg\max_w \frac{w^T S_a w}{w^T S_w w} && (18)\end{aligned}$$

这里引入了两个记号，S_a和S_w分别代表等式中间的两个值。这样，我们就把损失函数和数据集以即参数结合起来了。下面对这个损失函数求梯度，注意我们对w的绝对值并没有任何要求，只有方向的要求，因此只需要一个方程就可以求解：

$$\nabla_w J(w) = 2S_a w(w^T S_w w)^{-1} - 2w^T S_a w(w^T S_w w)^{-2} S_w w = 0 \quad (19)$$

$$\Rightarrow S_a w(w^T S_w w) = (w^T S_a w) S_w w \quad (20)$$

$$\Rightarrow w \propto S_w^{-1} S_a w = S_w^{-1}(\overline{x_{c1}} - \overline{x_{c2}})(\overline{x_{c1}} - \overline{x_{c2}})^T w \propto S_w^{-1}(\overline{x_{c1}} - \overline{x_{c2}}) \quad (21)$$

最后，我们得到$S_w^{-1}(\overline{x_{c1}} - \overline{x_{c2}})$ 就是我们需要寻找的方向。最后可以通过归一化，确定下来w的值。

2 线性分类-软分类

在第一章我们介绍了机器学习模型，按照模型输出分类，可以分为判别式和生成式，如果通过贝叶斯规则来表示，即：

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \quad (22)$$

对于这个表达式来说，我们需要让左边的概率值最大：

$$\begin{aligned}\arg\max_y p(y|x) &= \arg\max_y \frac{p(x|y)p(y)}{p(x)} && (23)\\ &= \arg\max_y p(x|y)p(y) && (24)\end{aligned}$$

我们把左边的$p(y|x)$称为判别模型，即在数据x的基础上，去推测标签y的值，而等式右边的$p(x|y)$ 我们把它称为生成式模型，即已知标签，我们建立$p(x|y)$的模型，通过对概率采样，可以产生新的数据。我们接下来介绍两种简单的生成式模型。

2.1 高斯判别分析

对于二分类任务，我们采用如下的假设：

1. $y \sim Bernoulli(\phi)$
2. $x|y=1 \sim N(\mu_1, \Sigma)$
3. $x|y=0 \sim N(\mu_0, \Sigma)$

那么独立同分布的数据集，最大后验概率可以表示为：

$$\arg\max_{\phi,\mu_0,\mu_1,\Sigma} \log p(X|Y)p(Y) = \arg\max_{\phi,\mu_0,\mu_1,\Sigma} \sum_{i=1}^{N} (\log p(x^{(i)}|y^{(i)}) + \log p(y^{(i)})) \quad (25)$$

我们把Bernoulli分布的表达式$p(y)=\phi^y(1-\phi)^{1-y}$带入，得到：

$$\arg\max_{\phi,\mu_0,\mu_1,\Sigma}\sum_{i=1}^{N}((1-y^{(i)})\log N(\mu_0,\Sigma)+y^{(i)}\log N(\mu_1,\Sigma)+y^{(i)}\log\phi+(1-y^{(i)})\log(1-\phi)) \tag{26}$$

求解方法依旧是分别对参数求导取0，首先看ϕ 的结果：

$$\sum_{i=1}^{N}\frac{y^{(i)}}{\phi}+\frac{y^{(i)}-1}{1-\phi}=0 \tag{27}$$

$$\Rightarrow\phi=\frac{\sum_{i=1}^{N}y^{(i)}}{N}=\frac{N_1}{N} \tag{28}$$

然后求解μ_1：

$$\hat{\mu_1} = \arg\max_{\mu_1}\sum_{i=1}^{N}y^{(i)}\log N(\mu_1,\Sigma) \tag{29}$$

$$= \arg\min_{\mu_1}\sum_{i=1}^{N}y^{(i)}(x^{(i)}-\mu_1)^T\Sigma^{-1}(x^{(i)}-\mu_1) \tag{30}$$

把等式右边的乘积分解，得到如下的表达式：

$$\sum_{i=1}^{N}y^{(i)}(x^{(i)}-\mu_1)^T\Sigma^{-1}(x^{(i)}-\mu_1)=\sum_{i=1}^{N}y^{(i)}{x^{(i)}}^T\Sigma^{-1}x^{(i)}-2y^{(i)}\mu_1^T\Sigma^{-1}x^{(i)}+y^{(i)}\mu_1^T\Sigma^{-1}\mu_1 \tag{31}$$

对等式右边求梯度，并令其为0，我们得到：

$$\sum_{i=1}^{N}-2y^{(i)}\Sigma^{-1}x^{(i)}+2y^{(i)}\Sigma^{-1}\mu_1=0 \tag{32}$$

$$\Rightarrow\mu_1=\frac{\sum_{i=1}^{N}y^{(i)}x^{(i)}}{\sum_{i=1}^{N}y^{(i)}}=\frac{\sum_{i=1}^{N}y^{(i)}x^{(i)}}{N_1} \tag{33}$$

由于μ_0与μ_1是类似的，所以我们就不再求解，直接给出答案：

$$\mu_0=\frac{\sum_{i=1}^{N}(1-y^{(i)})x^{(i)}}{N_0} \tag{34}$$

最后我们求解协方差矩阵Σ。同样还是先化简方程：

$$\sum_{i=1}^{N}\log N(\mu,\Sigma) = \sum_{i=1}^{N}\log(\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}})+(-\frac{1}{2}(x^{(i)}-\mu)^T\Sigma^{-1}(x^{(i)}-\mu)) \tag{35}$$

$$= Const-\frac{1}{2}N\log|\Sigma|-\frac{1}{2}Trace((x^{(i)}-\mu)^T\Sigma^{-1}(x^{(i)}-\mu)) \tag{36}$$

$$= Const-\frac{1}{2}N\log|\Sigma|-\frac{1}{2}Trace((x^{(i)}-\mu)(x^{(i)}-\mu)^T\Sigma^{-1}) \tag{37}$$

$$= Const-\frac{1}{2}N\log|\Sigma|-\frac{1}{2}NTrace(S\Sigma^{-1}) \tag{38}$$

在第三个等号，我们利用了迹的运算规则，括号里面的元素满足轮换关系。对于包含绝对值和迹的表达式的导数，我们有：

$$\frac{\partial}{\partial A}(|A|)=|A|A^{-1} \tag{39}$$

$$\frac{\partial}{\partial A}Trace(AB)=B^T \tag{40}$$

首先把方程（26），只留下于Σ有关的，即：

$$-\frac{1}{2}(N_1+N_2)\log|\Sigma|-\frac{1}{2}N_1Trace(S_1\Sigma^{-1})-\frac{1}{2}N_2Trace(S_2\Sigma^{-1}) \tag{41}$$

其中S_1与S_2分别为两个类别数据内部的协方差矩阵，其次利用公式（39），得到：

$$N\Sigma^{-1}-N_1S_1^T\Sigma^{-2}-N_2S_2^T\Sigma^{-2}=0 \tag{42}$$

$$\Rightarrow \Sigma=\frac{N_1S_1+N_2S_2}{N} \tag{43}$$

这里应用了协方差矩阵的对称性。至此我们求出了高斯判别分析的所有参数。

2.2 朴素贝叶斯模型

以上的高斯判别分析的是对数据集的分布做出高斯分布的假设，同时要求两类数据为伯努利分布，从而利用最大后验求得这些假设中的参数。

朴素贝叶斯对数据之间的特征，作出假设，一般地，我们有需要得到$p(x|y)$这个概率值，由于x有p个维度，因此需要对这么多的维度的联合概率进行采样，但是我们知道这么高维度的空间中采样需要的样本数量非常大才能获得较为准确的概率近似。

在一般的有向概率图模型中，对各个特征维度之间的条件独立关系作出了不同的假设，其中最为简单的一个假设就是在朴素贝叶斯模型描述中的条件独立性假设。

$$p(x|y)=\prod_{i=1}^{p}p(x_i|y) \tag{44}$$

即：

$$x_i\perp x_j|y,\forall i\neq j \tag{45}$$

于是利用贝叶斯定理，对于单次观测：

$$p(y|x)=\frac{p(x|y)p(y)}{p(x)}=\frac{\prod_{i=1}^{p}p(x_i|y)p(y)}{p(x)} \tag{46}$$

对于单个维度的条件概率以及类先验做出进一步假设：

1. x_i是连续变量：$p(x_i|y)=N(\mu_i,\sigma_i^2)$
2. x_i是离散变量：类别分布（categorical）$p(x_i=i|y)=\theta_i$，$\sum_{i=1}^{K}\theta_i=1$
3. $p(y)=\phi^y(1-\phi)^(1-y)$

下面举一个例子进行说明，对垃圾邮件进行分类（选自CS229），一共两类，一类是工作邮件，一类是垃圾邮件（广告，诈骗，邪教等）。这一类任务被称为文本分类。目前来说，**自然语言处理**（Natural Language Process）是非常热门的一个领域，主要任务包括：文本分类、情感分析、机器翻译、舆情监测、自动摘要、观点提取、问题回答、文本语义对比、语音识别等众多分支。最热的模型词嵌入（Word Embedding）和BERT模型，曾一度占领所有自然语言处理的竞赛榜一。我们在此仅用文本分类作为一个例子，讨论一下朴素贝叶斯模型，我们在这用英文邮件举例。我们把英语常用词做成一个列表，当邮件里面出现的每个词语出现，就在该邮件对应的向量的元素记为1，否则是0。我们假定$y=1$是工作邮件。那么$\phi_{j|y=1}=p(x_j=1|y=1)$，$\phi_{j|y=0}=p(x_j=1|y=0)$，且$\phi_y=p(y=1)$。所以对于一个数据集$\{(x^{(i)},y^{(i)});i=1,...,n\}$，我们可以写下似然函数：

$$L(\phi_y,\phi_{j|y=0},\phi_{j|y=1})=\prod_{i=1}^{n}p(x^{(i)},y^{(i)}) \tag{47}$$

对似然函数采用MLE估计，同样可以得到三个参数的极值，这里我们就不在计算，直接给出结果：

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{n} \mathbb{I}\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum_{i=1}^{n} \mathbb{I}\{y^{(i)} = 1\}} \tag{48}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{n} \mathbb{I}\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum_{i=1}^{n} \mathbb{I}\{y^{(i)} = 0\}} \tag{49}$$

$$\phi_y = \frac{\sum_{i=1}^{n} \mathbb{I}\{y^{(i)} = 1\}}{n} \tag{50}$$

这里的∧是且的意思。通过以上的信息，那么我们就能对一个新邮件，进行预测了：

$$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x)} \tag{51}$$

$$= \frac{(\prod_{j=1}^{d} p(x_j|y=1))p(y=1)}{(\prod_{j=1}^{d} p(x_j|y=1))p(y=1) + (\prod_{j=1}^{d} p(x_j|y=0))p(y=0)} \tag{52}$$

这里我们有一个值得主义的地方，如果邮件当中存在数字，那么我们采用连续变量离散化来处理，即把连续变量分段，做成一个向量，出现的数字落在哪一段，就认为该段对应的元素值为1，否则为0。

上面建立的模型存在了一个问题，如果某一个词汇在数据集中，一直没有出现，那么会发生分母为0的情况，这是非常不好的情况。我们为了避免这种情况，采用Laplace 平滑技术，认为所有的词汇都已经出现过一次，即：

$$\phi_j = \frac{1 + \sum_{i=1}^{n} \mathbb{I}\{z^{(i)} = j\}}{k + n} \tag{53}$$

这里注意到分母多了一个k，表示所有的词均出现过一次，所有词汇的总数进行了扩充，这也是概率归一的必然结果$\sum_{j=1}^{k} \phi_j = 1$。那么同理，另外两个条件概率就有如下的形式：

$$\phi_{j|y=1} = 1 + \frac{\sum_{i=1}^{n} \mathbb{I}\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{2 + \sum_{i=1}^{n} \mathbb{I}\{y^{(i)} = 1\}} \tag{54}$$

$$\phi_{j|y=0} = \frac{1 + \sum_{i=1}^{n} \mathbb{I}\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{2 + \sum_{i=1}^{n} \mathbb{I}\{y^{(i)} = 0\}} \tag{55}$$

对上面的模型还可以做一些改进，比如有一些词汇，会在一封邮件当中多次出现，所以我们把模型进一步改进，提出事件模型（even model）。具体的内容，我们在此不做介绍，我们作为课外阅读材料，参看吴恩达讲义notes2。

2.3 逻辑斯蒂回归

有时候我们只要得到一个类别的概率，那么我们需要一种能输出[0, 1]区间内任意实数的激活函数。考虑到两分类模型，我们利用判别模型，希望对$p(C|X)$ 建模，这里C 代表类别。利用贝叶斯定理：

$$p(C_1|x) = \frac{p(x|C_1)p(C_1)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)} \tag{56}$$

取$z = \ln \frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)}$，于是：

$$p(C_1|x) = g(z) = \frac{1}{1 + \exp(-z)} \tag{57}$$

我们把这个函数称为Sigmoid函数，其参数表示两类联合概率分布比值的对数。在判别式中，不关心这个参数的具体值，模型假设直接对z进行，所以简记为$g(z)$。我们可以观察这个函数的变化情况，如图（1）所示：

Logistic回归的模型假设是：

$$z = \theta^T x \tag{58}$$

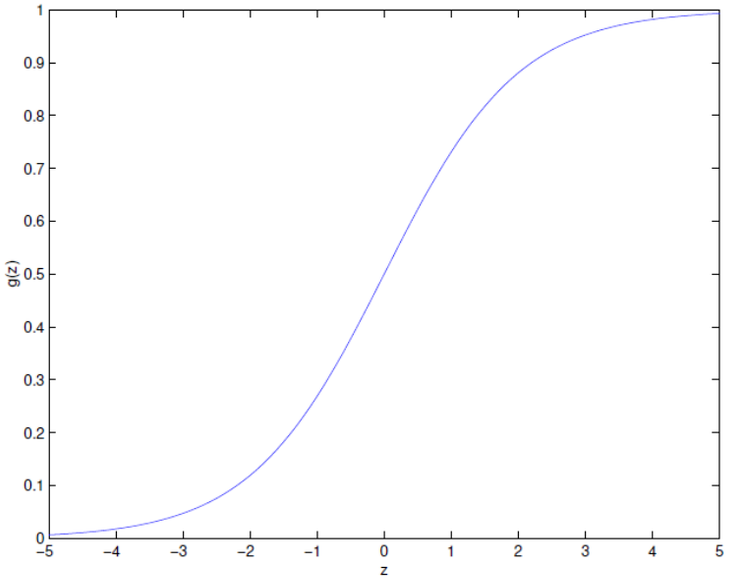

Figure 1: 横坐标是z，纵坐标是函数的值。

于是，我们的模型就是$h_\theta(x) = g(\theta^T x)$。通过寻找$\theta$ 的最佳值可以得到在这个模型假设下的最佳模型。概率判别模型常用最大似然估计的方式来确定参数。

对于一次观测，获得分类y的概率为（假定$C_1 = 1, C_2 = 0$）：

$$p(y|x) = p_1^y p_0^{1-y} \tag{59}$$

那么对于N次IID的观测，联合概率密度函数为：

$$\begin{aligned} L(\theta) &= p(Y|X;\theta) && (60)\\ &= \prod_{i=1}^{n} p(y^{(i)}|x^{(i)};\theta) && (61)\\ &= \prod_{i=1}^{n} (h_\theta(x^{(i)}))^{y^{(i)}} (1 - h_\theta(x^{(i)}))^{1-y^{(i)}} && (62) \end{aligned}$$

进而，我们通过MLE，得到：

$$\begin{aligned} \hat{\theta} &= \arg\max_\theta \log L(\theta) = \arg\max_\theta \ell(\theta) && (63)\\ &= \arg\max_\theta \sum_{i=1}^{N} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) && (64) \end{aligned}$$

模型与策略我们都已经建立起来了，下面就是算法了，我们依旧可以采用梯度下降算法。假设只有一个样本，我们先求一下对数似然函数的梯度：

$$\begin{aligned} \frac{\partial}{\partial \theta_j}\ell(\theta) &= (y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1 - g(\theta^T x)})\frac{\partial}{\partial \theta_j} g(\theta^T x) && (65)\\ &= (y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1 - g(\theta^T x)}) g(\theta^T x)(1 - g(\theta^T x))\frac{\partial}{\partial \theta_j}\theta^T x && (66)\\ &= (y(1 - g(\theta^T x)) - (1-y) g(\theta^T x)) x_j && (67)\\ &= (y - h_\theta(x)) x_j && (68) \end{aligned}$$

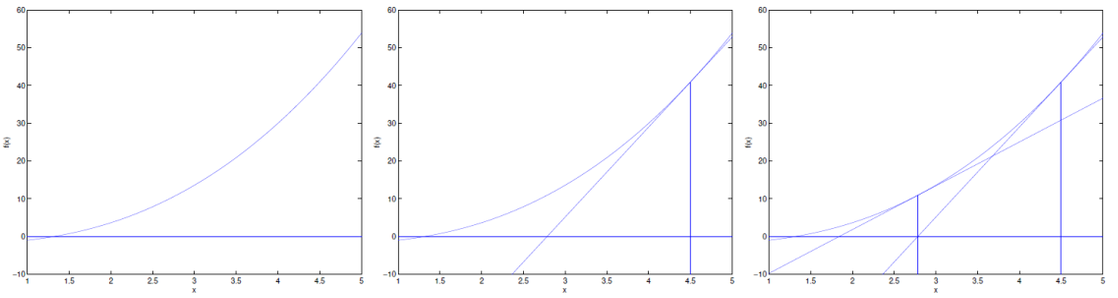

Figure 2: 牛顿方法寻找函数零点。

然后带入到随机梯度上升算法（这里是求最大值，所以是是上升），有如下的更新规则：

$$\theta_j := \theta_j + \alpha(y^{(i)} + h_\theta(x^{(i)}))x_j^{(i)} \tag{69}$$

当然可以把随机梯度下降扩展至批量梯度下降，或者小批量梯度下降进行更新。

3 牛顿方法

这一节，我们介绍一个新的算法，可以用于数据量不是很大的时候，一个二阶更新方法——**牛顿方法**（Newton’s Method）。我们从一个标量函数出发，去寻找它的零点。假定$f : \mathbb{R} \to \mathbb{R}$，我们希望寻找到一点$\theta$，使得$f(\theta) = 0$。这里的$\theta \in \mathbb{R}$是个实数。牛顿方法的更新规则为：

$$\theta := \theta - \frac{f(\theta)}{f'(\theta)} \tag{70}$$

我们对这个公式，有一个很自然的解释，我们观察图片（2）。我们通过做切线的方式，用当前的θ 减去切线对应的距离，得到更新值。通过多次更新找到函数的零点。 对于我们的问题，极大似然估计希望找到函数$\ell'(\theta) = 0$的解，所以我们令$f(\theta) = \ell'(\theta)$，然后带入到方程70，得到如下的方程：

$$\theta := \theta - \frac{\ell'(\theta)}{\ell''(\theta)} \tag{71}$$

我们这里是最大化似然函数（如果是最小化似然函数，牛顿更新公式是否发生变化？）。最后我们强调，对于我们的Logistic回归，参数θ是一个向量，所以我们需要推广牛顿方法。我们可以得到如下的方程：

$$\theta := \theta - H^{-1}\nabla_\theta \ell(\theta) \tag{72}$$

这里$\nabla_\theta \ell(\theta)$是对数似然函数的一阶导数，对应是一个列向量；$H$是海塞矩阵，$H \in \mathbb{R}^{(d+1)\times(d+1)}$，其中的矩阵元为：

$$H_{ij} = \frac{\partial \ell(\theta)}{\partial \theta_i \partial \theta_j} \tag{73}$$

牛顿方法作为一个二阶方法，会更快的收敛到极值点。但是换来的代价是，需要求二阶导数，如果d不是很大的情况下，是可行的，一旦d很大，每次迭代就会很慢，计算复杂度是$o(d^2)$。当牛顿方法应用在最大化Logistic回归的$\ell(\theta)$时，也被称为费舍尔得分（Fisher scoring）。

4 广义线性模型

到目前为止，我们看到了一个先验是高斯分布的回归模型，一个先验分布是Bernoulli分布的分类模型，在这一节，我们把这些模型进一步推广，得到一系列模型，我们把它称为广义线性模型。在介绍广义线性模型之前，我们先引入指数族分布。

4.1 指数族分布

指数族是一类分布，包括高斯分布，伯努利分布，二项分布，泊松分布，Beta分布，Dirichlet分布，Gamma分布等一系列分布。指数族分布可以写成统一的形式：

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - A(\eta)) \tag{74}$$

其中，η是参数向量，$T(y)$被称为充分统计量（一般情况下，我们取$T(y) = y$），$A(\eta)$是对数配分函数，$\exp(A(\eta))$它一般扮演着归一化因子的位置。充分统计量在在线学习中有应用。对于一个数据集，只需要记录样本的充分统计量即可。对于一个模型分布假设（似然），那么我们在求解中，常常需要寻找一个共轭先验，使得先验与后验的形式相同，例如选取似然是二项分布，可取先验是Beta分布，那么后验也是Beta分布。指数族分布常常具有共轭的性质，于是我们在模型选择以及推断具有很大的便利。共轭先验的性质便于计算，同时，指数族分布满足最大熵的思想（无信息先验），也就是说对于经验分布利用最大熵原理导出的分布就是指数族分布。在更复杂的概率图模型中，例如在无向图模型中如受限玻尔兹曼机中，指数族分布也扮演着重要作用。在推断的算法中，例如变分推断中，指数族分布也会大大简化计算。下面我们先通过几个例子来看一下指数族分布，首先看Bernoulli分布：

$$\begin{aligned} p(y;\phi) &= \phi^y(1-\phi)^{1-\phi} && (75)\\ &= \exp(y\log\phi + (1-y)\log(1-\phi)) && (76)\\ &= \exp((\log(\frac{\phi}{1-\phi})) + \log(1-\phi)) && (77) \end{aligned}$$

因此，参数向量η为$\eta = \log(\frac{\phi}{1-\phi})$。有趣的是，如果求解$\phi$的反函数，得到的是$\phi = 1/(1+e^{-\eta})$。这就是Sigmoid函数。我们可以看到，Logistic回归是指数族分布的一个具体实现。我们可以把每个量都具体对应：

$$\begin{aligned} T(y) &= y && (78)\\ A(\eta) &= -\log(1-\phi) = \log(1+e^{\eta}) && (79)\\ b(y) &= 1 && (80) \end{aligned}$$

通过以上的表达式，我们可以看出Bernoulli分布完全可以写成指数族分布的形式。

我们接下来再看一个例子，高斯分布。我们回忆一下，我们在对线性回归进行概率解释时，方差σ^2对我们最终的结果没有影响，所以我们简单起见，这里令$\sigma^2 = 1$[1]。接下来我们对$\sigma^2 = 1$的高斯分布做变形：

$$\begin{aligned} p(y;\eta) &= \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}(y-\mu)^2) && (81)\\ &= \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}y^2)\exp(\mu y - \frac{1}{2}\mu^2) && (82) \end{aligned}$$

因此，我们对照指数族分布的参数，给出各个参数的具体形式：

$$\begin{aligned} \eta &= \mu && (83)\\ T(y) &= y && (84)\\ a(\eta) &= \mu^2/2 && (85)\\ &= \eta^2/2 && (86)\\ b(y) &= (1/\sqrt{2\pi})\exp(-y^2/2) && (87) \end{aligned}$$

我们可以看到，高斯分布也是指数族分布的一个表现形式。接下来，我们在讨论一下指数族分布里充分统计量与对数配分函数的关系。

[1]如果考虑σ，依旧可以写成指数族的形式，只不过参数向量包含σ，充分统计量是一个向量$\phi(x) = (x\ x^2)^T$。

4.1.1 充分统计量与对数配分函数的关系

把对数配分函数放到概率密度函数以便，然后对概率密度函数求积分，因为积分对对数配分函数没有影响，所以有：

$$\exp(A(\eta)) = \int b(y)\exp(\eta^T T(y))dx \tag{88}$$

两边对参数求导：

$$\exp(A(\eta))A'(\eta) = \int b(y)\exp(\eta^T T(y))T(y)dx \tag{89}$$

$$A'(\eta) = E_{p(y;\eta)}[T(y)] \tag{90}$$

类似的，我们继续求导，可以得到：

$$A''(\eta) = Var_{p(y;\eta)}[T(y)] \tag{91}$$

由于方差为正，于是$A(\eta)$一定是凸函数。

4.1.2 充分统计量与最大似然估计

对于IID数据集$\mathcal{D} = \{y^{(1)}, ..., y^{(N)}\}$。

$$\eta_{MLE} = \arg\max_{\eta} \sum_{i=1}^{N} \log p(y^{(i)};\eta) \tag{92}$$

$$= \arg\max_{\eta} \sum_{i=1}^{N} (\eta^T T(y^{(i)}) - A(\eta)) \tag{93}$$

最后我们得到：

$$A'(\eta_{MLE}) = \frac{1}{N}\sum_{i=1}^{N} T(y^{(i)}) \tag{94}$$

由此可以烂到，为了估计参数，只需要知道充分统计量就可以了。

4.1.3 最大熵

熵（entropy）最初是在热力学引入的概念，描述了能量在转化过程中，必须遵循热力学第二定律，热力学第二定律由Clausius和Kelvin两个物理学家分别提出，二者等价，此时的熵，被称为Clausius熵，记为$S = \int_A^B \frac{dQ}{T}$。后来经过物理学家Boltzmann 提出的统计物理学完善，具有了熵的微观表达式$S = k_B \ln \Omega$，这里k_B称为玻尔兹曼常量，Ω是微观状态数，我们可以看到，微观状态数越多，熵越大，那么换据说，微观状态数越多，系统对应的混乱度越高，所以熵是描述系统混乱程度的物理量。热力学第二定律通过玻尔兹曼熵的解释后，又叫做熵增原理。把熵的概念推广到信息领域的科学家Shannon在1948年做出的工作，用来衡量信息量的多少。同样对于一个概率分布，同样可以衡量分布包含不确定性的多少。对于一个连续概率分布，信息熵的定义如下：

$$S = \int p(x)\log\frac{1}{p(x)}dx = \int -p(x)\log p(x)dx \tag{95}$$

信息熵在决策树模型上，具有非常重要的意义，感兴趣的同学，可以去查阅相关的文献与书籍。

我们假设，对于一个数据集来说，我们不知道任何先验信息，把这个假设称为最大熵假设。假设数据是离散分布的，k个特征的概率分别为p_k，最大熵原理可以表述为：

$$\max\{H(p)\} = \min\{\sum_{k=1}^{K} p_k \log p_k\} \ s.t. \sum_{k=1}^{K} p_k = 1 \tag{96}$$

利用拉格朗日乘子法：

$$L(p,\lambda)=\sum_{k=1}^{K}p_k\log p_k+\lambda(1-\sum_{k=1}^{K}p_k) \tag{97}$$

于是可得：

$$p_1=p_2=\cdots=p_K=\frac{1}{K} \tag{98}$$

因此等可能的情况熵最大，这是符合直觉的，等可能性意味着极大的不确定性，当一种情况的概率变大，意味着确定性越来越高，熵必然会减小。

一个数据集$\mathcal{D}$，在这个数据集上的经验分布为$\hat{p}=\frac{Count(x)}{N}$，实际不可能满足所有的经验概率相同，于是在上面的最大熵原理中还需要加入这个经验分布的约束。对任意一个函数，经验分布的经验期望可以求得为：

$$E_{\hat{p}}[f(x)]=\Delta \tag{99}$$

于是：

$$\max\{H(p)\}=\min\{\sum_{k=1}^{N}p_k\log p_k\}\ s.t.\ \sum_{k=1}^{N}p_k=1, E_p[f(x)]=\Delta \tag{100}$$

拉格朗日函数为：

$$L(p,\lambda_0,\Lambda)=\sum_{k=1}^{N}p_k\log p_k+\lambda_0(1-\sum_{k=1}^{N}p_k)+\lambda^T(\Delta-E_p[f(x)]) \tag{101}$$

求导得到：

$$\frac{\partial}{\partial p(x)}L=\sum_{k=1}^{N}(\log p(x)+1)-\sum_{k=1}^{N}\lambda_0-\sum_{k=1}^{N}\lambda^T f(x) \tag{102}$$

$$\Rightarrow\sum_{k=1}^{N}\log p(x)+1-\lambda_0-\Lambda^T f(x)=0 \tag{103}$$

由于数据集是任意的，对数据集求和也意味着求和项里面的每一项都是0：

$$p(x)=\exp(\lambda^T f(x)+\lambda_0-1) \tag{104}$$

我们可以看到，在具有先验知识后，通过最大熵原理，得到是指数族分布，也进一步印证了，当我们一旦假设数据集具有某种先验分布，那么一定会从指数族分布演化而来。

4.2 构造广义线性模型

广义线性模型（General Linear Model）假定你想建立一个模型，去评估到达一个商店的顾客数量y，特征x 可能是促销，广告，天气等。我们已经知道Possion 分布对这种计数问题能建立一个非常好的模型。幸运的是，Possion分布也是指数族分布的一员，所以自然地就想到广义线性模型。这里我们介绍如何对一个具体问题，构造广义线性模型。

更一般的来说，我们通过一个自变量x，去预测一个随机变量y的值。所以，基于此，我们通过如下的步骤构造广义线性模型：

1. $y|z;\theta\sim ExponentialFamily(\eta)$。换句话说，给定$x$ 和参数θ，y的分布应该服从指数族分布，参数为η。

2. 给定x，我们的目标是预测$T(y)$的期望。在大多数时，$T(y)=y$，所以这意味着我们模型的输出$h(x)$ 其实对应着的是$h(x)=E[y|x]$。

3. 最重要的是，要满足线性关系，即$\eta = \theta^T x$。

通过以上三个假设，我们就构造了一个非常优美的模型，叫做广义线性模型。进一步来说，我们再次回顾之前学过的两个模型，从广义线性模型出发，得到的假设是否与之前的结果一样。

4.2.1 回顾线性回归

对于我们之前的假设，我们说$h(x)$对应着概率的期望，所以我们由如下的表达式：

$$\begin{aligned} h_\theta &= E[y|x;\theta] && (105)\\ &= \mu && (106)\\ &= \eta && (107)\\ &= \theta^T x && (108) \end{aligned}$$

第一个等式来自于假设2，第二个等式采用了假设$y|x;\theta \sim N(\mu, \sigma^2)$，高斯分布的期望值是$\mu$，第三个等式来自假设1，最后一个等式来自假设3。

4.2.2 回顾逻辑斯蒂回归

我们再看Logistic回归，我们从哪个假设1出发：

$$\begin{aligned} h_\theta(x) &= E[y|x;\theta] && (109)\\ &= \phi && (110)\\ &= 1/(1+e^{-\eta}) && (111)\\ &= 1/(1+e^{-\theta^T x}) && (112) \end{aligned}$$

这里面我们运用了假设，y的分布是伯努利分布，伯努利分布是指数族分布的一员，所以换句话说，也是广义线性模型的体现。

4.2.3 softmax模型

最后我们讨论一个多分类问题，即类别数$K > 2$。那么这里y就对应着就不止0, 1两个取值，为$y \in \{1, 2, ..., k\}$。对于多分类任务，那么对应着的分布为**多项式分布**（multinomial distribution）。多项式分布，是伯努利分布的推广，意味着不止两类，比如掷骰子，一共六种可能性，每种可能性是1/6。我们接下来推导先验假设为多项式分布的广义线性模型。首先我们要强调，多项式分布也是指数族分布。因为一共有k种可能性，所以可以使用k个参数$\phi_1, ..., \phi_k$来对每种输出进行描述。然而，并不是所有的ϕ_i是独立的，因为要求概率归一$\sum_{i=1}^{k} \phi_i = 1$。所以我们参数化多项式分布仅需要$k-1$个参数，$\phi_1, ..., \phi_{k-1}$，这里$\phi_i = p(y = i; \phi)$并且$p(y = k; \phi) = 1 - \sum_{i=1}^{k-1} \phi_i$。我们一般也把$\phi_k = 1 - \sum_{i=1}^{k} \phi_i$，但是我们要清楚，这不是一个参数，参数只有$k-1$个。

我们定义充分统计量$T(y)$为一个列向量，$T(y)\mathbb{R}^{k-1}$，这样定义是因为不止两类。

$$T(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, T(2) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, T(3) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, ..., T(k-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, T(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (113)$$

我们通过定义可以看出，这里$T(y) \neq y$，已经不再是一个实数了。我们将用$(T(y))_i$来标记向量$T(y)$中i-th 的元素。我们在第零章介绍过一个特殊函数，指示函数。使用指示函数可以把$T(y)$与y之间的关系描述出来，即$(T(y))_i = \mathbb{I}\{y = i\}$。与Bernoulli分布类似，$E[T(y)_i] = p(y = i) = \phi_i$。

下一步，我们把多项式分布改称为指数族分布通式的形式：

$$
\begin{aligned}
p(y;\phi) &= \phi_1^{\mathbb{I}\{y=1\}}\phi_2^{\mathbb{I}\{y=2\}}\cdots\phi_k^{\mathbb{I}\{y=k\}} && (114)\\
&= \phi_1^{\mathbb{I}\{y=1\}}\phi_2^{\mathbb{I}\{y=1\}}\cdots\phi_k^{1-\sum_{i=1}^{k-1}\mathbb{I}\{y=i\}} && (115)\\
&= \phi_1^{(T(y))_1}\phi_2^{(T(y))_2}\cdots\phi_k^{1-\sum_{i=1}^{k-1}(T(y))_i} && (116)\\
&= \exp[(T(y))_1\log(\phi_1) + (T(y))_2\log(\phi_2) + \cdots + (1-\sum_{i=1}^{k-1}(T(y))_i\log(\phi_k)] && (117)\\
&= \exp[(T(y))_1\log(\phi_1/\phi_k) + (T(y))_2\log(\phi_2/\phi_k) + \cdots + (T(y))_{k-1}\log(\phi_{k-1}/\phi_k) + \log(\phi_k)] && (118)\\
&= b(y)\exp(\eta^T T(y) - A(\eta)) && (119)
\end{aligned}
$$

对应相等，可以给出：

$$
\begin{aligned}
\eta &= \begin{bmatrix} \log(\phi_1/\phi_k) \\ \log(\phi_2/\phi_k) \\ \vdots \\ \log(\phi_{k-1}/\phi_k) \end{bmatrix} && (120)\\
A(\eta) &= -\log(\phi_k) && (121)\\
b(y) &= 1 && (122)
\end{aligned}
$$

以上完成了把多项式分布转化为指数族分布的形式。其中参数向量的元素与概率值之间的联系为：

$$\eta_i = \log\frac{\phi_i}{\phi_k} \tag{123}$$

方便起见，我们定义$\eta_k = \log(\phi_k/\log_k) = 0$。求出$\eta_i$与$\phi_i$之间的关系：

$$
\begin{aligned}
e^{\eta_i} &= \frac{\phi_i}{\phi_k} && (124)\\
\phi_k e^{\eta_i} &= \phi_i && (125)\\
\phi_k\sum_{i=1}^{k} e^{\eta_i} &= \sum_{k=1}^{k}\phi_i = 1 && (126)
\end{aligned}
$$

这里面暗含了$\phi_k = 1 - \sum_{i=1}^{k} e^{\eta_i}$。所以把这个关系进一步带入，得到：

$$\phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}} \tag{127}$$

我们把上面这个函数，称为**softmax**函数。

为了完成我们的模型，还需要使用第三条假设，即$\eta_i = \theta_i^T x \ for\ i = 1, ..., k-1$，这里$\theta_1, ..., \theta_{k-1} \in \mathbb{R}^{d+1}$是我们模型的参数。为了与之前的记号对应，我们也定义$\theta_k = 0$，所以$\eta_k = \theta_k^T x = 0$。最后我们就可以写出模型的条件概率：

$$
\begin{aligned}
p(y = i|x;\theta) &= \phi_i && (128)\\
&= \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}} && (129)\\
&= \frac{e^{\theta_j^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}} && (130)
\end{aligned}
$$

这个模型可以处理多分类任务，即$y \in \{1, ..., k\}$，把这个模型称为**softmax回归**（softmax regression）。

根据假设，模型输出为：

$$
\begin{aligned}
h_\theta(x) &= E[T(y)|x;\theta] && (131)\\
&= E\left[\begin{array}{c|c} \mathbb{I}\{y=1\} & \\ \mathbb{I}\{y=1\} & x;\theta \\ \vdots & \\ \mathbb{I}\{y=1\} & \end{array}\right] && (132)\\
&= \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{k-1} \end{bmatrix} && (133)\\
&= \begin{bmatrix} \frac{\exp(\theta_1^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \\ \frac{\exp(\theta_2^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \\ \vdots \\ \frac{\exp(\theta_{k-1}^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \end{bmatrix} && (134)
\end{aligned}
$$

换句话说，我们的假设，输出的是一个概率分布$p(y = i|x;\theta)$，对于所有的$i = 1, 2, ..., k$。那么最后，我们还需要一个目标函数，即似然函数，与之前的类似，我们对n个样本，采用连乘的形式，把联合概率分布表达出来，然后取对数，得到对数似然函数：

$$
\begin{aligned}
\ell(\theta) &= \sum_{i=1}^n \log p(y^{(i)}|x^{(i);\theta}) && (135)\\
&= \sum_{i=1}^n \log \prod_{i=1}^k \left(\frac{e^{\theta_i^T x^{(i)}}}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}}\right)^{\mathbb{I}\{y^{(i)}=l\}} && (136)
\end{aligned}
$$

这里我们把方程（130）的表达式代入到了上式。至此softmax模型的建立已经完成，下面只需要采用MLE方法，采用梯度下降或者牛顿方法，去最大化似然函数就可以了。

References