CS229 Lecture Notes

Andrew Ng and Tengyu Ma

July 19, 2024

Part I

监督学习

Chapter 1

线性回归

为了让我们的住房案例更有趣，让我们考虑一个稍微复杂些的数据集，我们额外知晓每套住房的卧室数量：

居住面积（平方英尺）	# 卧室数量	价格（1000 美元）
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

其中，x 为属于 $\mathbb{R}^2$ 的二维向量。例如，$x_1^{(i)}$ 为训练集中第 i 套住房的居住面积，$x_2^{(i)}$ 为其卧室数量。（通常，当设计一个学习问题时，需要由你自己来决定选择哪些特征，因此，如果你在波特兰（Portland）收集住房数据，可能也会选择其他特征，比如，每套住房是否有壁炉及浴室的数量等。我们后续会讨论更多有关于特征选择的内容，但目前只考虑上面给出的特征。）

为了开展监督学习，我们必须决定如何在计算机中表示函数或假设 h。作为初始选择，我们将 y 近似为一个关于 x 的线性函数：

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

其中，θ_i 为参数（也称为权重），参数化从 $\mathcal{X}$ 映射到 $\mathcal{Y}$ 的线性函数空间。我们将 $h_\theta(x)$ 简写为 $h(x)$。为了简化我们的表示，我们引入 $x_0 = 1$（截距项，Intercept Term），可以得到如下等式：

$$h(x) = \sum_{i=0}^{d} \theta_i x_i = \theta^T x$$

其中，我们可以将上述等式右侧的 θ 与 x 视为向量，d 为输入变量的数量（不计算 x_0）。

1.1 LMS 算法

我们需要找到一个 θ 使 $J(\theta)$ 最小化。让我们使用一种搜索算法，该算法以一个对 θ 的初始猜测值开始，然后不断调整 θ 使 $J(\theta)$ 更小，直至收敛至某一个能够使 $J(\theta)$ 最小化的 θ。特别地，让我们考虑梯度下降（Gradient Descent）算法，以某个初始值 θ 开始，不断进行如下更新：

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

（同时对所有 $j = 0, \ldots, d$ 使用此更新。）其中，α 被称为学习率，这是一个非常自然的算法，每次向 J 最陡峭的衰减方向前进一步。

Chapter 2

分类与逻辑回归

2.1 多分类问题

考虑这样一个分类问题，响应值 y 满足 $y \in \{1, 2, \ldots, k\}$。举个例子，我们可能想要将邮件划分为三种类型，比如垃圾邮件、个人邮件及工作邮件，而不是只划分为垃圾邮件与非垃圾邮件（这是一个二分类问题）。标签或响应值仍然是离散的，但可以取两个以上的值。因此我们将使用多项式分布对这种问题进行建模。

在这种情况下，$p(y|x;\theta)$ 是基于 k 个离散值的分布，这是一个多项式分布。对于包含 k 个值的多项式分布，$\phi_1, \ldots, \phi_k$ 表示每一种可能的概率，必须满足 $\sum_{i=1}^{k} \phi_i = 1$。我们将设计一个参数化模型，在给定输入 x 的前提下，输出满足这个约束的 $\phi_1, \ldots, \phi_k$。

我们引入 k 组参数 $\theta_1, \ldots, \theta_k$，每一组参数都是空间 $\mathbb{R}^d$ 中的一个向量。根据直觉，我们应该可以使用 $\theta_1^T x, \ldots, \theta_k^T x$ 来表示 $\phi_1, \ldots, \phi_k$，即概率 $P(y = 1|x;\theta), \ldots, P(y = k|x;\theta)$。然而，采用这种直接的办法有两个问题，首先，$\theta_j^T x$ 不一定在 $[0, 1]$ 内，其次，$\sum_{j=1}^{k} \theta_j^T x$ 不一定为 1。因此，我们将使用 softmax 函数将向量 $(\theta_1^T x, \ldots, \theta_k^T x)$ 转化为每个元素都是非负的并且和为 1 的概率向量。

定义 softmax 函数 $\mathrm{softmax} : \mathbb{R}^{\mathrm{k}} \to \mathbb{R}^{\mathrm{k}}$ 为

$$\mathrm{softmax}(\mathrm{t}_1, \ldots, \mathrm{t}_{\mathrm{k}}) = \begin{bmatrix} \frac{\exp(t_1)}{\sum_{j=1}^{k} \exp(t_j)} \\ \vdots \\ \frac{\exp(t_k)}{\sum_{j=1}^{k} \exp(t_j)} \end{bmatrix} \tag{2.1.1}$$

softmax 函数的输入，向量 t 一般被称为 logits，在定义中，softmax 函数的输出必为每个元素都是非负的并且和为 1 的概率向量。

令 $(t_1, \ldots, t_k) = (\theta_1^T x, \ldots, \theta_k^T x)$，将 $(t_1, \ldots, t_k)$ 作为 softmax 函数的输入，将 softmax 函数的输出作为概率 $P(y = 1|x;\theta), \ldots, P(y = k|x;\theta)$，得到如下概率模型：

$$\begin{bmatrix} P(y = 1|x;\theta) \\ \vdots \\ P(y = k|x;\theta) \end{bmatrix} = \mathrm{softmax}(\mathrm{t}_1, \ldots, \mathrm{t}_{\mathrm{k}}) = \begin{bmatrix} \frac{\exp(\theta_1^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \\ \vdots \\ \frac{\exp(\theta_k^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \end{bmatrix} \tag{2.1.2}$$

为了表示方便，我们令 $\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)}$，上面等式可以简写为：

$$P(y = i|x;\theta) = \phi_i = \frac{\exp(t_i)}{\sum_{j=1}^{k} \exp(t_j)} = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^{k} \exp(\theta_j^T x)} \tag{2.1.3}$$

接下来，我们计算一个样例 (x, y) 的负对数-似然（log-likehood）。

$$-\log P(y|x,\theta) = -\log\left(\frac{\exp(t_y)}{\sum_{j=1}^{k}\exp(t_j)}\right) = -\log\left(\frac{\exp(\theta_y^T x)}{\sum_{j=1}^{k}\exp(\theta_j^T x)}\right) \tag{2.1.4}$$

因此，训练数据的负对数-似然，即损失函数可以写为：

$$\ell(\theta) = \sum_{i=1}^{n} -\log\left(\frac{\exp(\theta_{y^{(i)}}^T x^{(i)})}{\sum_{j=1}^{k}\exp(\theta_j^T x^{(i)})}\right) \tag{2.1.5}$$

通过模块化上面的等式，可以很方便地定义交叉熵损失（Cross-entropy Loss）$\ell_{ce} : \mathbb{R}^k \times \{1,\dots,k\} \to \mathbb{R}_{\le 0}$ 为：[1]

$$\ell_{ce}((t_1,\dots,t_k), y) = -\log\left(\frac{\exp(t_y)}{\sum_{j=1}^{k}\exp(t_j)}\right) \tag{2.1.6}$$

通过上述等式，我们可以将等式(2.1.5)简写为：

$$\ell(\theta) = \sum_{i=1}^{n} \ell_{ce}((\theta_1^T x^{(i)},\dots,\theta_k^T x^{(i)}), y^{(i)}) \tag{2.1.7}$$

并且交叉熵损失也具有一个简单的梯度表示。令 $t = (t_1,\dots,t_k)$，且 $\phi_i = \frac{\exp(t_i)}{\sum_{j=1}^{k}\exp(t_j)}$，通过基本微积分，我们可以推导出：

$$\frac{\partial \ell_{ce}(t,y)}{\partial t_i} = \phi_i - 1\{y = i\} \tag{2.1.8}$$

其中，$1\{\cdot\}$ 为指示函数（Indicator Function），即如果 $y = i$，$1\{y = i\} = 1$，如果 $y \neq i$，$1\{y = i\} = 0$。另外，基于矢量表示法我们有如下表示法，该表示在第 7 节非常有用：

$$\frac{\partial \ell_{ce}(t,y)}{\partial t_i} = \phi_i - e_s \tag{2.1.9}$$

其中，$e_s \in \mathbb{R}^k$ 为第 s 个自然基向量（Natural Basis Vector，向量的第 i 个元素为 1，其他元素为 0），使用链式法则，我们可以得到：

$$\frac{\partial \ell_{ce}((\theta_1^T x,\dots,\theta_k^T x), y)}{\partial \theta_i} = \frac{\partial \ell_{ce}(t,y)}{\partial t_i}\cdot\frac{t_i}{\theta_i} = (\phi_i - 1\{y = i\})\cdot x \tag{2.1.10}$$

因此，损失函数相对于参数 θ_i 的的梯度为：

$$\frac{\partial \ell(\theta)}{\partial \theta_i} = \sum_{j=1}^{n}(\phi_i^{(j)} - 1\{y^{(j)} = i\})\cdot x^{(j)} \tag{2.1.11}$$

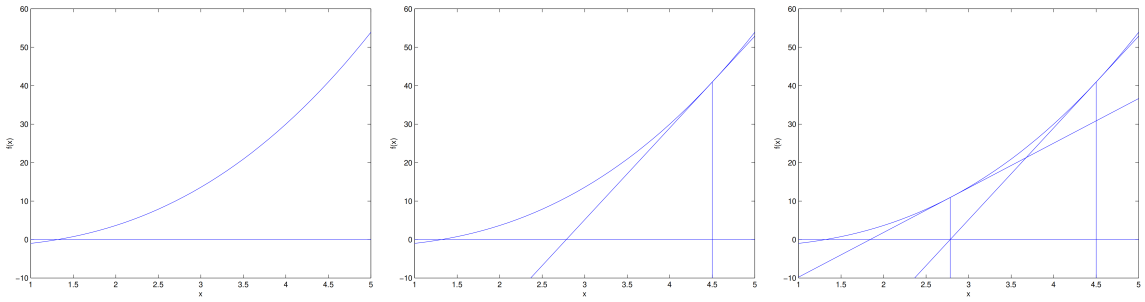
其中，$\phi_i^{(j)} = \frac{\exp(\theta_i^T x^{(j)})}{\sum_{s=1}^{k}\exp(\theta_s^T x^{(j)})}$ 为模型将样例 $x^{(j)}$ 预测为 i 的概率。根据上面的梯度公式，可以实现（随机）梯度下降以最小化损失函数 $\ell(\theta)$。

2.2 最大化 $\ell(\theta)$ 的另一种算法

回到逻辑回归问题，$g(z)$ 为 sigmoid 函数，让我们讨论另一种用于最大化 $\ell(\theta)$ 的算法。

让我们开始吧，考虑使用牛顿法寻找函数的零点。假设我们有一个函数 $f : \mathbb{R} \to \mathbb{R}$，期望找到一个

[1]这里的命名有些许歧义。一些人将交叉熵损失定义为将概率向量（在我们的定义中为 ϕ）与标签 y 映射为一个实数的函数，称我们的交叉熵损失为 softmax-交叉熵损失。我们选择这种命名习惯是因为它与大多数现代深度学习库是一致的，比如 PyTorch 与 Jax。



值 θ 满足 $f(\theta)=0$，其中 $\theta\in\mathbb{R}$ 为实数。牛顿法进行如下操作：

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

这个方法可以视为，在当前猜测值 θ 处，通过一个正切于 f 的线性函数近似 f，以求解该线性函数的零点，然后将下一个 θ 设置为线性函数的零点。

这里有一张展示牛顿法具体步骤的图片：

在最左边的图片中，可以看到函数 f 与直线 $y=0$ 绘制在一起，我们尝试寻找满足 $f(\theta)=0$ 的 θ，能够满足这个目标的 θ 约为 1.3。假设我们设置 θ 为 4.5 初始化此算法。然后牛顿法在 $\theta=4.5$ 处拟合一条正切于 f 的直线[2]，求解此直线的零点（中间的图片），并作为下一次迭代的猜测值，约为 2.8，最右面的图片展示了下一次迭代的结果，将 θ 更新为约 1.8，再经过几次迭代后，我们将接近 $\theta=1.3$。

牛顿法提供了寻找函数零点 $f(\theta)=0$ 的方法，那么求解某个函数 ℓ 的最大值该如何处理呢？函数 ℓ 的最大值与一阶导数 $\ell'(\theta)$ 的零点有关。因此，令 $f(\theta)=\ell'(\theta)$，我们可以用同样的方法获取 ℓ 的极大值点，更新规则如下：

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

（思考题：如果我们想使用牛顿法寻找函数的最小值而不是最大值，需要做哪些调整？）

最后，在我们逻辑回归的定义中，θ 为向量值，因此我们需要推广牛顿法以满足这种定义，将牛顿法推广至多维定义的方法如下（也称为 **Newton-Raphson** 法）：

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

其中，$\nabla_\theta\ell(\theta)$ 和通常一样，为 $\ell(\theta)$ 相对于 θ_i 的偏导数向量，H 为 $d\times d$ 矩阵（实际上为 $(d+1)\times(d+1)$，假设包括截距项），称为 **Hessian**，可以表示为：

$$H_{ij} = \frac{\partial^2 \ell(\theta)}{\partial\theta_i \partial\theta_j}$$

相较于（批量）梯度下降，牛顿法的收敛速度更快，且通过较少的迭代即可非常接近最小值。但牛顿法的一次迭代相较于梯度下降复杂度更大，因为需要计算和求逆 $d\times d$ Hession，但只要 d 不是很大，总体上还是快得多。当使用牛顿法最大化逻辑回归对数似然函数 $\ell(\theta)$ 时，这种方法称为 **Fisher Scoring**。

[2]译者注：此时直线为 $y=f'(4.5)x+f(4.5)-4.5f'(4.5)$，零点 $x=\frac{4.5f'(4.5)-f(4.5)}{f'(4.5)}=4.5-\frac{f(4.5)}{f'(4.5)}$。

Chapter 3

广义线性模型

目前，我们已经看过一个回归的案例，以及一个分类的案例。在回归的案例中，我们有 $y|x;\theta \sim \mathcal{N}(\mu,\sigma^2)$，在分类的案例中，我们有 $y|x;\theta \sim \text{Bernoulli}(\phi)$，其中，$\mu$ 与 ϕ 为 x 与 θ 的适当的函数定义。在这一章节，我们将展示这两种方法都是广义线性模型（Generalized Linear Models，GLMs）[1]的特例，我们也会展示广义线性模型中的其他模型是如何推导以及应用于其他回归与分类问题的。

3.1 指数族

为了进一步研究 GLMs，我们首先定义指数族，如果某种分布可以写为如下形式，那么我们认为它属于指数族。

$$p(y;\eta) = b(y)\exp(\eta^T T(y) - a(\eta)) \tag{3.1.1}$$

其中，η 被称为分布的**自然因数**（Natural Parameter）（也被称为**权威因数**（Canonical Parameter）），$T(y)$ 为**充分统计量**（Sufficient Statistic）（对于我们所考虑的分布，通常为 $T(y)=y$），$a(\eta)$ 为**对数分割函数**（Log Partition Function），$e^{a(\eta)}$ 的值通常起到归一化常数的作用，使得 $p(y;\eta)$ 基于 y 求和/积分结果为 1。

一组固定的 T、a、b 值定义了一个以 η 为参数的分布集合，当我们改变 η 时，我们得到这个集合中不同的分布。

现在我们将说明伯努利分布与高斯分布只是指数族中的个例。均值为 ϕ 的伯努利分布可以写为 $\text{Bernoulli}(\phi)$，表示 $y \in \{0,1\}$，且 $p(y=1;\phi)=\phi$，$p(y=0;\phi)=1-\phi$，改变 ϕ，我们将得到具有不同均值的伯努利分布。现在我们将说明这种通过改变 ϕ 得到的伯努利分布族属于指数族。例如，存在 T、a、b 使得等式（3.1.1）成为伯努利分布。

我们将伯努利分布写为：

$$\begin{aligned} p(y;\phi) &= \phi^y(1-\phi)^{1-y} \\ &= \exp(y\log\phi + (1-y)\log(1-\phi)) \\ &= \exp\left(y\log\left(\frac{\phi}{1-\phi}\right) + \log(1-\phi)\right) \end{aligned} \tag{3.1.2}$$

因此，自然因数为 $\eta = \log(\phi/(1-\phi))$，有趣地，如果我们逆转 η 的定义，通过 η 求解 ϕ，可以得到 $\phi = 1/(1+e^{-\eta})$，这是熟悉的 sigmoid 函数，当我们将推到逻辑回归推导为 GLM 时，这一点会再次

[1]这一章节的灵感来自于 Michael I. Jordan 所著的《Learning in graphical models》（未出版的草稿），以及 McCullagh 与 Nelder 所著的《Generalized Linear Models (第二版)》

出现。为了完成伯努利分布的指数族分布公式表达，我们有：

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

这说明使用合适的 T、a、b 可以将伯努利分布改写为等式（3.1.1）的形式。

现在我们开始考虑高斯分布。回想一下，当推导线性回归时，σ^2 的值对我们关于 θ 与 $h_\theta(x)$ 的最终选择是没有任何影响的。因此，对于 σ^2，我们可以选择任何值，为了简化后续推导，设 $\sigma^2 = 1^2$。然后，我们有

$$\begin{aligned} p(y;\mu) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y-\mu)^2\right) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \cdot \exp\left(\mu y - \frac{1}{2}\mu^2\right) \end{aligned}$$

因此，我们可以看出高斯分布属于指数族，其中

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

还有很多其他分布也属于指数族，比如多项式分布（稍后就会看到）、泊松分布（为计数数据建模；另见问题集）、Gamma 分布与指数分布（用于模拟连续的非负随机变量，如时间间隔）、Beta 分布与狄利克特分布（用于概率分布）等。在下一章节中，我们将介绍用于建模来自任何这些分布的 y（给出 x 与 θ）的一般方法。

3.2 构建 GLMs

假设你想构建一个模型，这个模型可以基于某些特征 x（例如，商店促销活动、最近的广告、天气、今天是周几等）来预测，在任一小时内，访问你的商店的顾客数量（或者你的网站的页面浏览数）y。我们知道对于访客数量泊松分布通常可以给出一个较好的模型。了解这个后，我们应该如何为我们的问题提出一个模型呢？幸运的是，泊松分布属于指数族，因此我们可以使用广义线性模型。在这一章节中，我们将介绍为这种问题构建广义线性模型的方法。

更一般地，考虑一个分类或回归问题，我们想将某个随机变量 y 的值作为 x 的函数进行预测。为推导出这个问题的 GLM，我们将对 y 的状态分布（在给出 x 的前提下）以及我们的模型做出以下三个假设：

[2]如果保留 σ^2 作为变量，同样可以展示高斯分布属于指数族，其中，$\eta \in \mathbb{R}^2$ 为取决于 μ 与 σ 的二维向量。为了 GLMs 的目的，然而，参数 σ^2 可以通过考虑更一般的指数族定义 $p(y;\eta,\tau) = b(a,\tau)\exp((\eta^T T(y) - a(\eta))/c(\tau))$ 来处理，其中，τ 被称为**分散因数**（Dispersion Parameter，译者注：这里翻译可能不准确），且对于高斯分布，$c(\tau) = \sigma^2$，但考虑到我们上面的简化，我们不再需要对后续例子进行更一般的定义。