极客大学

极客大学机器学习训练营
机器学习动手实战

王然

众微科技 AI Lab 负责人

二〇二一年一月二十五日

大纲

1 概览

2 逻辑回归的实现

大纲

- 上次讲座中我们讲了如何使用概率模型推导损失函数，并解释了如何对损失函数进行求导。
- 在这一讲中，我们将会讲如何将之前的内容转化为真实的生产力，及推导之前说过的模型。

从模型到代码的过程

极客大学

- 非常精细的写出模型当中的每一步；
- 检查是否有标记的错误；
- 使用推导当中的写法，无视 pep8 进行开发；
- 使用最笨的方式进行开发，不要考虑效率；
- 使用 Monte Carlo 检查简单的模型是否正常；

在开始本课前，注意复习…

极客大学

- 极大似然的概念；
- 矩阵求导的基本法则；

大纲

逻辑回归基本设定

极客大学

- 定义 $\sigma : x \mapsto \frac{1}{1+\exp(-x)}$.
- 逻辑回归的概率密度函数为 $p_\beta(x_i) = \sigma(x_i^t \beta)$, 其中 β 为未知参数，x_i 为解释变量；「
- 负的对数似然函数为 $-\sum_i y_i \log(p_\beta(x_i)) + (1-y_i)\log(1-p_\beta(x_i))$。
- 我们现在需要做的是求他的导数。

最麻烦的部分

极客大学

- 由于矩阵形式非常简单，所以最麻烦的部分其实是对一堆非线性的函数的推导。
- 我们当然可以手推，但是问题在于，手推很容易出错。
- 所以这时候 sympy 就可以出场了。
- 见 notebook。

使用 SymPy 之后...

我们可以写出对数似然函数为

$$-\sum_i \left(y_i \exp(-x_i^t\beta)/(1+\exp(-x_i^t\beta) - (1-y_i)\exp(x_i^t\beta)/(1+\exp(x_i^t\beta))\right) x_i$$

括号里面的东西还是有点复杂，所以我们不妨再试试看 sympy 是否能帮我们化简

化简结果如下

$$-\sum_{i}(y_i - \sigma(x_i^t\beta))x_i$$

下面让我们使用 jax 来实现自动求导的过程并测试整体的正确性

见 colab notebook。

关于 Jax 实践要点的回顾

- 请注意在这里，我们尽可能用接近于 Numpy 的形式进行了实现。并且我们通过 Jax 的 Autograd 机制判断了我们是否求导准确。这一点是非常重要的；
- 如果没有 Jax，我们可能只能用 PyTorch 或者 TF 计算 Autograd（会相对来说更麻烦一些）；
- 如果这些 Autograd 的机制都没有这时候我们面临的问题就更为麻烦。这时候通常使用 Finite Difference 进行调试。
- 请注意这种调试往往会造成很大的误差，所以有时候很难进行判断。

在继续实践之前...

极客大学

- 在我们上一讲当中，我们解释过如果我们得到了导数，接下来我们该使用一些优化方法来一步步进行优化了。
- 不幸的是，就目前对于 Python 来说，不论是 jax.scipy.opimitize.minimize 还是 scipy.minimize 都有巨大的问题。
- 就 Jax 来说，大部分优化算法都没有实现，实现的部分也有 bug。
- 就 scipy 来说，问题更大一些。

Scipy 的问题

极客大学

- Scipy 优化的最大的问题在于，scipy 包装的是 fortran 77 的优化路径；
- 在 fortran 77 的优化路径当中，其整体只使用了双精度，并对于各种存在的精度问题没有做任何优化。
- 这使得在实际使用中，scipy 几乎永远不会得到正确的结论，因为各种 numerical issue 都会出现，并且不能修复；
- 这也是为什么在科学计算中，大部分人还是使用 matlab 的原因。

但是...

- 从 scipy 的问题回来，其实在之前的讲解中，我们还有一个问题没有解答，那就是可识别性的问题。
- 什么叫可识别性呢？考虑以下问题。

思考题：请问以下模型是否可以正常优化求解

- 假设我们的目标是 y，我们有 x_1, x_2, x_3 三个变量，并且 $x_3 = 2x_1 + x_2$；
- 我们是否能找到 $\beta_1, \beta_2, \beta_3$ 使得 $\sum_i (y_i - \beta_1 x_{i1} - \beta_2 x_{i2} - \beta_3 x_{i3})^2$ 最小。
- 如果可能，我们能找到多少个？

以上问题称之为不可识别性的问题

- 简单来说，对于一个模型来说，存在（潜在）无穷多个解使得该模型对应的损失函数最小。
- 对于存在线性表达式的模型来说，这种情况是极其麻烦的。这里面一种很常见的情况，称之为多重共线性，值得是一些变量可以用其他变量的线性组合表达出来。
- 对于 R 来说，这些情况一般可以自动处理，即找到最大线性无关组；很不幸的是，在 python 中，scipy 的实现极烂（大约比正常 C++ 实现慢 100 万倍）；
- 我们不会对具体算法进行讲解，具体算法已经在第二章当中的 cython 例子当中给出。大家可以直接使用。

思考题：one-hot 编码输入逻辑回归之后是否可以正常求解？