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

二〇二一年六月七日

大纲

大纲

- 上次课程里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。
- 在这一章中，我们将会讲如何将之前的内容转化为真实的生产力，以及推导之前说过的模型。

- **这一章是我们的所有章节中，对于能力培养最核心的章节。**
- 在这一章中，我们会把第二章到第四章的所有核心知识点串联起来。

从模型到代码的过程

- 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，忽略 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…

极客时间

- 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

- 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 $-\sum_i y_i \log(p_\beta(x_i)) + (1-y_i)\log(1-p_\beta(x_i))$；
- 现在需要做的是求它的导数。

一个难点

- 由于矩阵形式非常简单，所以难点在于对一系列非线性函数的推导；
- 虽然可以手推，但是手推很容易出错，所以可以采用 sympy；
- 见 notebook。

使用 SymPy 之后...

我们可以写出对数似然函数的导数为

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

括号里的内容还是有些复杂，所以不妨再看看 sympy 是否能帮我们化简？

化简结果如下

$$-\sum_i \left(y_i - \sigma\left(x_i^t \beta\right)\right) x_i$$

使用 jax 实现自动求导的过程并测试整体的正确性（见 colab notebook）。

Jax 实践

- 尽可能用接近于 Numpy 的形式实现，通过 Jax 的 Autograd 机制来辅助判断求导的准确性；
- 如果没有 Jax，用 PyTorch 或者 TF 计算 Autograd（操作复杂度更高）；
- 在最原始的条件下，使用 Finite Difference 进行调试（有较大误差）。

Jax 实践

- 得到函数后，可以开始逐步优化
- jax.scipy.opimitize.minimize 与 scipy.optimize.minimize 的问题

Scipy 的问题

- scipy 包装的是 fortran 77 的优化路径；
- 在 fortran 77 的整体只使用了双精度；
- 这使得数值问题经常出现。

但是...

- 可识别性的问题还没有得到解答；
- 请思考以下问题。

思考题

- 请问以下模型是否可以正常优化求解?
  - 假设目标是 y，有 x_1, x_2, x_3 三个变量，并且 $x_3 = 2x_1 + x_2$;
  - 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，能找到多少个？

不可识别性的问题

- 对一个模型来说，存在（潜在）无穷多个解使得该模型对应的损失函数最小；
- 如果模型中有线性表达式，可能有多重共线性的情况出现，即一些变量可以用其他变量的线性组合表达出来；
- R 可以自动处理多重共线性（找到最大线性无关组）；
- python 的 scipy 的实现效果较差（大约比正常 C++ 实现慢 100 万倍）；
- 具体算法讲解见黑板。

思考题： one-hot 编码输入逻辑回归之后是否可以正常求解？

One-hot 和常数项之间的关系

- 如果有常数项的话，那么 one-hot 是不可以加入的，原因在于 one-hot 编码加起来等于 1；
- 如果没有任何常数项以及其他输入是可以的；
- 为什么要加常数项：假设我们用“受教育年限”对“工资”做回归，如果我们我们不加常数项，则等于我们认为未受过的教育的人的工资应该是 0，这显然是不符合实际的。

附录：关于 Tobit 模型的推导

见Cameron and Trivedi (2005) 16.3 的推导和实现

大纲

极客时间

1 概览

2 逻辑回归的实现

3 优化方法
- 牛顿法
- Proximal Methods 的原理（选学）
- Proximal Methods 的实现

4 补充习题（可选）

5 参考文献

大纲

1. 概览
2. 逻辑回归的实现
3. 优化方法
   - 牛顿法
   - Proximal Methods 的原理（选学）
   - Proximal Methods 的实现
4. 补充习题（可选）
5. 参考文献

牛顿法和拟牛顿法

- 目标：求解函数 $f(x)=0$，假设 f 可导；
- 对某一点 x_0 做泰勒展开，有 $0=f(x)\approx f(x_0)+f'(x_0)(x-x_0)$；
- 如果将之改写为迭代算法，则 $x\approx x_0-\frac{f(x_0)}{f'(x_0)}$；
- 问题：
  - 在优化问题中，这意味着需要求得 2 阶导数（Hessian），但在实际中一般采用各种近似的方法；
  - 步长问题：通常采用不同的步长，有一些方法采用固定步长，有些方法采用后向线性搜索（backtracking line search）。

拟牛顿法

- 目前最常用的方法为 BFGS 和 L-BFGS 方法；
- 推导比较复杂，故我们在这里省略，感兴趣的同学可以看附属材料；
- 目前 L-BFGS 最好的实现见软件库，请注意需要阅读源码！

大纲

1 概览

2 逻辑回归的实现

3 优化方法
- 牛顿法
- Proximal Methods 的原理（选学）
- Proximal Methods 的实现

4 补充习题（可选）

5 参考文献

见附件。

大纲

1. 概览
2. 逻辑回归的实现
3. 优化方法
   - 牛顿法
   - Proximal Methods 的原理（选学）
   - Proximal Methods 的实现
4. 补充习题（可选）
5. 参考文献

Proximal Methods 的实现

- 这一章，我们将会模仿真实的学习过程，及我们的讲义，直接尝试实现；
- 目标是实现对数似然函数加上 l_1 损失的情况；
- 在这里，假设对数似然函数为 $l_\beta(X, y)$，其中 β 为待估参数，而 X, y 为数据，目标是最小化 $-l_\beta(X, y) + \lambda\|\beta\|_1$，其中 $\lambda \geq 0$ 为惩罚参数；
- 具体实现过程是从练习开始，然后再开始实现。

实现细节

见 Colab Notebook。

思考题（进阶）：如何提升模型的效果

- 在上面的学习中，我们使用的 step size 都是固定的；
- 在这种情况，得到的结果类似于梯度下降；
- 那么是否有办法采用不同的 step size 呢？

如何对自己的算法 debug

- 首先检查数学推导是否正确：最好的方法是和其他材料做交叉验证；
- 其次检查每一步是否都有合适的结果；
- 最后运行整个算法的时候，需要注意：
  - 算法是否真的收敛了？
  - 是否有 overflow 和 underflow?
  - 在多大情况下，算法会运行到一个局部最优？
  - 是否可以通过调整初始值的方法加速收敛？
  - 是否可以改变 line_search 的方向？

关于 Proximal Methods 的一些应用的说明

极客时间

- Proximal methods 主要应用在 l_1 正则化的函数估计上；
- 这类方法还有很多，例如Efron et al. (2004) 和Garrigues and Ghaoui (2008) 等。目前在深度学习上也开始出现应用 (Yun, Lozano, and Yang 2020)。

大纲

基本要求

极客时间

- 实现时间为 24 小时（大学内的考核要求，在训练营中不规定实现时长）；
- 可通过任何一种优化方法（BFGS 或 Proximal Methods）实现；
- 根据模型内容，选择任何一种编程语言实现 100 万次以上模拟，并且根据该模拟研究该算法在不同情况下的可靠程度。

第一题：非参数 kernel 回归

- 请选择至少两种不同的和 y 存在非线性关系的 X 进行实验。
- 请实现逻辑回归中的 Kernel Regression 方法，见Cameron and Trivedi (2005) 第 9.5，并实现 Monte Carlo 估计。
- 请回答：
  - 不同的 bandwidth 对于问题的影响有多大？
  - 当 X 之间的相关性增加时，估计量效果如何？

第二题： Bayesian MCMC 估计

- 请复现Cameron and Trivedi (2005) 的 11.36 的内容。
- 请研究 Prior 在样本增加时对于 Posterior 的影响大小。

第三题：Nested Logic

- 阅读Cameron and Trivedi (2005) 的第 15.6 节，并实现 Nested Logic 模型的估计。
- 研究如果 Nested Structure 有问题时候，上一层估计量的影响。

第四题：Ordered Regression

- 阅读Cameron and Trivedi (2005) 的 15.9.1 节，并实现该模型。
- 研究如果 ϵ 来自于和 log-likehood 不同的分布时，估计量的性质。

第五题：Tobit 模型

- 阅读Cameron and Trivedi (2005) 的 16.3 节，并实现该模型。
- 检查当 ϵ 为柯西分布时对整个估计的影响。

第六题：Roy 模型

- 阅读Cameron and Trivedi (2005) 的 16.7 并实现 Roy Model。
- 检查当 16.47 式子中，当 σ 假定有错误的情况下，对于 Roy Model 的估计有什么影响。

第七题：Survival Analysis

- 阅读Cameron and Trivedi (2005) 的 17.6 节并且实现。
- 检查在 Hazard Function 指定错误的情况下模型的表现。

第八题：Finite Mixture of Count Regress

- 阅读Cameron and Trivedi (2005) 的 24.3 节，并实现模拟。
- 请检查当 latent class 数量指定错误时候，模型的结果。

第九题：Censored Count Regression

- 阅读Cameron and Trivedi (2005) 的 24.4 节，并实现 truncation 和 censored 中任选一种模型。
- 请检查当 truncation 或者 censoring 错误时候，其估计结果的正确性。

大纲

- Cameron, A Colin and Pravin K Trivedi (2005). *Microeconometrics: methods and applications*. Cambridge university press.
- Efron, Bradley et al. (2004). "Least angle regression". In: *Annals of statistics* 32.2, pp. 407–499.
- Garrigues, Pierre and Laurent Ghaoui (2008). "An homotopy algorithm for the Lasso with online observations". In: *Advances in neural information processing systems* 21, pp. 489–496.
- Yun, Jihun, Aurelie C Lozano, and Eunho Yang (2020). "A general family of stochastic proximal gradient methods for deep learning". In: *arXiv preprint arXiv:2007.07484*.