The-Elements-of-Statistical-Learning 学习笔记

李奥林

mrliaolin@outlook.com

目录

第一章
Model Inference and Averaging

本章为《The elements of Statistical Learning》第 8 章的笔记。

1.1 Introduction

1.2 The Bootstrap and Maximum Likelihood Methods

1.2.1 A Smoothing Example

自助法提供了一种评估不确定性的直接计算方法（置信区间）。

- 非参数自助法（与最小二乘法的置信区间类似）
- 参数自助法（对每个预测的 y 值加一个高斯噪声，参数为噪声的方差，此时估计出的函数的置信区间与最小二乘法的完全相同）

1.2.2 Maximum Likelihood Inference

参数自助法与最小二乘法是一致的，因为模型具有加法高斯误差。一般地，参数自助法并非与最小二乘法一致，而是与极大似然一致。

1.2.3 Bootstrap versus Maximum Likelihood

1.3 Bayesian Methods

在用于推理的贝叶斯方法中，$\Pr(\mathbf{Z}|\theta)$ 是采样模型，先验分布是 $\Pr(\theta)$，反映我们看到数据之前的关于 θ 的知识，后验分布为 $\Pr(\theta|\mathbf{Z})$ 是我们看到数据之后关于 θ 更新的知识。

与标准的“频率论”方法的区别是使用先验分布来表达看到数据之前的不确定性，并在看到数据之后允许残余的不确定性以后验分布形式来表示。

1.4 Relationship Between the Bootstrap and Bayesian Inference

自助法分布为我们的参数提供了一个（近似的）非参数的、无信息的后验分布。

1.5 The EM Algorithm

1.5.1 Two-Component Mixture Model

1.5.2 The EM Algorithm in General

公式 $\mathrm{E}(\ell_0(\theta';\mathbf{T})|\mathbf{Z},\hat{\theta}^{(j)})$ 的定义是

$$\mathrm{E}(\ell_0(\theta';\mathbf{T})|\mathbf{Z},\hat{\theta}^{(j)}) = \sum_{\mathbf{Z}^m} \ell_0(\theta';\mathbf{T})\Pr(\mathbf{Z}^m|\mathbf{Z},\hat{\theta}^{(j)}) \tag{1.1}$$

根据条件概率的链式法则

$$\Pr(x_2,...,x_n|x_1) = \prod_{i=2}^{n}\Pr(x_i|x_1,...,x_{i-1}) \tag{1.2}$$

得 $\Pr(\mathbf{Z}^m,\mathbf{Z}|\theta') = \Pr(\mathbf{Z}^m|\theta')\Pr(\mathbf{Z}|\mathbf{Z}^m,\theta')$，即

$$\Pr(\mathbf{Z}|\theta') = \frac{\Pr(\mathbf{Z}^m,\mathbf{Z}|\theta')}{\Pr(\mathbf{Z}^m|\mathbf{Z},\theta')} \tag{1.3}$$

用对数似然函数表示，$\ell(\theta';\mathbf{Z}) = \ell_0(\theta';\mathbf{Z}^m,\mathbf{Z}) - \ell_1(\theta';\mathbf{Z}^m|\mathbf{Z})$，其中 ℓ_1 基于条件概率密度 $\Pr(\mathbf{Z}^m|\mathbf{Z},\theta')$。关于参数 θ 支配的 $\mathbf{T}|\mathbf{Z}$ 取条件期望，得

$$\begin{aligned}\ell(\theta';\mathbf{Z}) &= \mathrm{E}[\ell_0(\theta';\mathbf{T})|\mathbf{Z},\theta] - \mathrm{E}[\ell_1(\theta';\mathbf{Z}^m|\mathbf{Z})|\mathbf{Z},\theta] \\ &\equiv Q(\theta',\theta) - R(\theta',\theta)\end{aligned} \tag{1.4}$$

其中，$R(\theta',\theta)$ 的定义是

$$R(\theta',\theta) = \sum_{\mathbf{Z}^m} \ell_1(\theta';\mathbf{Z}^m|\mathbf{Z})\Pr(\mathbf{Z}^m|\mathbf{Z},\theta) \tag{1.5}$$

is the expectation of a log-likelihood of a density(indexed by θ'), with respect to the same density indexed by θ，当 $\theta' = \theta$ 时，作为 θ' 的函数取最大值。因而，当极大化 $Q(\theta',\theta)$ 时，可以得出

$$\begin{aligned}\ell(\theta';Z) - \ell(\theta;Z) &= [Q(\theta',\theta) - Q(\theta,\theta)] - [R(\theta',\theta) - R(\theta,\theta)] \\ &\geq 0\end{aligned} \tag{1.6}$$

1.5.3 EM as a Maximization-Maximization Procedure

One does not need to maximize with respect to all of the latent data parameters at once, but could instead maximize over one of them at a time, 而可以在 M 步轮流一次极大化它们中的一个。

1.6 MCMC for Sampling from the Posterior

MCMC(Markov chain Monte Carlo) 马尔科夫链蒙特卡洛方法。

1.7 Bagging(装袋)

Bagging(Bootstrap aggregation) 对自助法样本集上的预测求平均，从而降低方差。真实装袋估计的定义是 $\mathrm{E}_{\hat{p}}\hat{f}^*(x)$，其中 $\hat{p}$ 表示经验分布，即从实际总体中而不是数据中抽取样本。

仅当原来的估计是非线性的，或者是数据的自适应函数时，装袋估计与 $\hat{f}(x)$ 不同。

对于数模型来说，类概率估计为在末端节点中的类比例，Bagging 平均类概率通常可降低方差。

1.7.1 Example: Trees with Simulated Data

由于预测子的相关性，这些树具有较高的方差。Bagging 成功地光滑了这种方差，从而降低了检验误差。

Bagging 可以降低均方误差，因为平均可以降低不稳定过程（如树）的方差，而保持偏倚不变。参考知乎：为什么说 bagging 是减少 variance，而 boosting 是减少 bias，由于子集样本集的相似性以及使用的是同种模型，因此各模型有近似相等的 bias 和 variance。由于 $\mathrm{E}[\sum x_i/n] = \mathrm{E}[x_i]$，以 bagging 后的 bias 和单个子模型的接近，一般来说不能显著降低 bias。另一方面，若各子模型独立，$\mathrm{var}(\sum x_i/n) = \mathrm{var}(x_i)/n$，此时可以显著降低 variance。

训练样本是从 P 分布中抽取的不相关的样本 (不重复)，而自助样本也是从 P 中采样得到的。

当 bag 一个模型时，模型中任何简单结构都将失去。如 bagged 树已不再是树，对于模型的解释，这显然是一个缺点。

由 bag 计算的期望类概率不能在任何一个 single replication 上实现，在这种其意义上，bag 一定程度上增大了各基分类器的模型空间。对于该例子 (single split 分类器 bag 拟合双

向拟合 $x_1 + x_2 = 1$) 或者其它例子，模型需要放大时，bag 没有帮助。

1.8 Model Averaging and Stacking

从非参数的贝叶斯角度分析，估计子的自助法值可看作对应参数的近似后验值，从这个角度看，bagged 值是一个后验贝叶斯均值 ($\mathrm{E}(Y|X = x)$)，因而可以减小均方误差。训练样本的估计对应于后验众数 (结构经验损失？)。

有时直接取均值的方法不能成功，原因是没考虑模型的复杂性，即没将模型置于相同的的立足点。

Stacked generalization 或者 stacking，则可以解决该问题。通过使用交叉验证预测 $\hat{f}_m^{-i}(x)$，可以避免将不合理的高权值赋予具有高复杂度的模型。通过限定权值非负并且和为 1 可以得到更好的结果。stacking 通常将导致更好的预测，但可解释性不如从 M 个模型中选取一个好。

1.9 Stochastic Search: Bumping

第二章

Additive Models, Trees, and Related Methods

2.1 Generalized Additive Models

2.2 Tree-Based Methods

2.2.1 Background

2.2.2 Regression Trees

2.2.3 Classification Trees

不同的节点非纯度度量 $Q_m(T)$，包括交叉熵或者 deviance(散离)。
交叉熵和基尼指数对结点概率的改变更加敏感，相对于错误率来说。

2.2.4 Other issues

Categorical Predictors

The Loss Matrix

观测的误分类后果对于某些类要比其他类严重。为了把损失引入到建模过程中，可以把 Gini 指数修改成 $\sum_{k\neq k'} L_{kk'}\hat{p}_{mk}\hat{p}_{mk'}$ 。该方法对多分类比较有效，对于二分类，系数不起作用，更好的办法是给 k 类中的样本加权 $L_{kk'}$。对于多分类来说，仅当 $L_{kk'}$ 与 k' 无关时才能使用。观测加权的作用是改变类的先验概率。

Missing Predictor Values

Why Binary Splits

多路分裂会很快地把数据分裂成碎片，导致下一层的数据不足。而且多路分裂也可以由一系列二叉分裂组成。

Other Tree-Building Procedures

CART(classification and regression tree)

Linear Combination Splits

线性组合分裂可能增强树的预测能力，但可能破坏其可解释性。在计算方面，分裂点搜索的离散性阻碍了权值光滑优化的使用。

Instability of Trees

树的方差较大，数据的一个较小变化将导致一系列完全不同的分裂，使得解释有些不稳定。这种不稳定性的主要原因是过程的分层本性，顶层分裂中的错误被传播到下面的所有分裂。

Lack of Smoothness

可以看到预测面缺乏光滑性 (分层)。在 0/1 损失的分类中，因为类概率估计中的偏倚的影响有限，因而不会产生太大伤害。然而，可能降低回归处理的性能。

Difficulty in Capturing Additive Structure

2.2.5 Spam Example(Continued)

交叉验证误差率由一系列 α 值来标引，而不是树的大小，因为对于同一个 α，不同折生成的树可能大小不一样。(图 9.4 的交叉验证结果只被 α 值索引，测试结果既可被 α，又可被剪枝后的树的大小索引)

医学分类问题中，术语敏感性 (sensitivity)(1 预测为 1) 和特效性 (specificity)(0 预测为 0) 用来刻画规则。

更好的方法不是仅在节点中修改贝叶斯规则（更改损失权重），而是在树增长过程中考虑不相等损失。

ROC 曲线下面的面积也被称为 c-statistic，当考虑一个额外的预测子加在标准模型上的影响时，其可能不是一个合理的度量。新的预测子可能在模型散离度的改变上影响很大，而在 c-statistic 上只会小量的增大。另一方面，c-statistic 在分析额外预测子对独立样本的分类的改变上有用。

2.3 Missing Data

第三章

Boosting and Additive Trees

3.1 Boosting Methods

3.1.1 Outline of This Chapter

3.2 Boosting Fits an Additive Model

3.3 Forward Stagewise Additive Modeling

3.4 Exponential Loss and AdaBoost

训练集误分类率大约在 250 次迭代后平稳下来，但是指数损失保持递减，因为它对估计类概率的变化更为敏感。

3.5 Why Exponential Loss?

对于加法建模，指数损失的主要吸引力在于计算。它引出简单的模再加权 AdaBoost 算法。

定义 $Y' = (Y+1)/2 \in \{0,1\}$，将 $\{-1,1\}$ 转换成 $\{0,1\}$。

3.6 Loss Functions and Roubustness

尽管应用于总体联合分布时指数损失和二项式散离产生相同的解，但都是 population（总体）意义上的，对于有穷数据集就不相同了。

Robust Loss Functions for Classification

在训练过程中，指数损失标准主要影响具有大的负边缘值的观测。二项式散离对这样的观测的影响相对较小，并更均匀地对所有数据散步这种影响。因此，在噪声处理中，它

的健壮性更强。

Robust Loss Functions for Regression

在拟合过程中，有限样本上的平方误差损失更重视具有较大绝对残差的观测。这样，它极其缺乏健壮性，而且对于长尾误差分布，特别是对于严重的误差度量值（“异常值”），它的性能会大幅度下降。

3.7 'Off-the-Shelf' Procedures for Data Mining

3.8 Example: Spam Data

3.9 Boosting Trees

提到的式子 (10.27) 对 (10.26) 的近似，是指 $\tilde{L}$ 对 L 的近似？

3.10 Numerical Optimization via Gradient Boosting

3.10.1 Steepest Descent

$$g_{im} = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)} \tag{3.1}$$

最速下降可以看作一个非常贪心的策略，因为在方向 $-\mathbf{g}_m$，$L(\mathbf{f})$ 在 $\mathbf{f} = \mathbf{f}_{m-1}$ 上下降最快。

3.10.2 Gradient Boosting

逐步前向提升也是一种非常贪心的策略，树预测 $T(x_i; \Theta_m)$ 类似于负梯度的分量。两者之间的主要区别是树分量 $\mathbf{t}_m = (T(x_1; \Theta_m), \cdots, T(x_N; \Theta_m))$ 不是独立的，它们被限制为一个 J_m 端点决策树的预测，而负梯度是无约束的最大下降方向。

使用梯度提升（利用负梯度拟合树），而不是残差进行拟合，主要原因是梯度提升可以针对特定的损失函数。

3.10.3 Implementation of Gradient Boosting

最原始的实现为 MART(multiple additive regression trees)，即多重加法回归树。

3.11 Right-Sized Trees for Boosting

对于很多实际当中遇到的问题，低阶交互效应趋于占支配地位。
基于树逼近的交互效应受树大小的限制。

3.12 Regularization

3.12.1 Shrinkage

3.12.2 Subsampling

3.13 Interpretation

3.13.1 Relative Importance of Predictor Variables

3.13.2 Partial Dependence Plots

$f(X)$ 对 X_S 的偏依赖的定义是：

$$f_S(X_S) = \mathrm{E}_{X_C} f(X_S, X_C) \tag{3.2}$$

它并不是忽略 X_C 的作用，而是考虑 X_C 的平均作用后的结果。

条件期望：

$$\mathrm{E}(f(X_S, X_C)|X_S) = \sum_{X_C} f(X_S, X_C)\,\Pr(X_c|X_S) \tag{3.3}$$

它是仅用 X_S 的函数对 $f(X)$ 的最佳最小二乘方逼近。

3.14 illustrations

3.14.1 California Housing

3.14.2 New Zealand Fish

抓住的尺寸大小有过多的 0，针对此有 zero-inflated Poisson 模型，一个更简单的方法是：

$$\mathrm{E}(Y|X) = E(Y|Y > 0, X) \cdot \Pr(Y > 0|X) \tag{3.4}$$