回归方法

1 逻辑斯蒂回归

因变量是定性变量（假设只取两个值），自变量可以定量、定性（哑变量）。输出结果在0到1之间，用逻辑斯蒂函数

$$p(X)=\frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

逻辑斯蒂函数可以等价地写成

$$\frac{p(X)}{1-p(X)}=e^{\beta_0+\beta_1 X}$$

称左边为“发生比"， 它的取值范围为 $(0,\infty)$
取对数，得:

$$\log\left(\frac{p(X)}{1-p(X)}\right)=\beta_0+\beta_1 X\ .$$

极大似然估计参数，似然函数：

$$L\left(\beta_0,\beta_1\right)=\prod_{i:y_i=1}p\left(x_i\right)\prod_{i':y_{i'}=0}\left(1-p\left(x_{i'}\right)\right)$$

梯度下降法或牛顿迭代法求系数估计。

2 判别分类

贝叶斯定理：

$$p_k(x):=P(Y=k\mid X=x)=\frac{\pi_k f_k(x)}{\sum_{l=1}^K\pi_l f_l(x)}$$

估计π_k：取随机样本，计算属于第k类的样本占总样本的比例。

将一个待判别的x分类到使得$p_k(x)$达到最大的那个类，这种方法被称为贝叶斯分类器。

2.1 线性判别（LDA)

假设预测变量只有1个。此外，假设 $f_k(x)$ 是正态的:

$$f_k(x)=\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left\{-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right\},\quad x\in R$$

再假设 $\sigma_1^2=\cdots=\sigma_K^2$, 简记为 σ^2, 那么:

$$p_k(x)=\frac{\pi_k\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu_k)^2\right\}}{\sum_{l=1}^K\pi_l\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu_l)^2\right\}}$$

对 $p_k(x)$ 两边取对数，可知贝叶斯分类器其实是将观测x分到使得

$$\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

达到最大的那一类。
假设$K = 2$, $\pi_1 = \pi_2$, 则当 $2x\left(\mu_1 - \mu_2\right) > {\mu_1}^2 - {\mu_2}^2$ 时，贝叶斯分类器将观测分入类1, 否则分入类2. 贝叶斯决策边界:

$$x = \frac{\mu_1^2 - \mu_2^2}{2\left(\mu_1 - \mu_2\right)} = \frac{\mu_1 + \mu_2}{2} .$$

2.2 二次判别分析（QDA）

假设来自第k类的随机观测服从$N\left(\mu_k, \sigma_k^2\right)$, 在这种假设下，贝叶斯 分类器把观测分入使得

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2\sigma_k^2}(x - \mu_k)^2 + \log \pi_k \\ &= -\frac{1}{2\sigma_k^2}x^2 + \frac{x\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + \log \pi_k\end{aligned}$$

达到最大的那一类。

当有p个自变量时, LDA的协方差矩阵有$p(p+1)/2$个参数, 而QDA的K个协方差矩阵有$Kp(p+1)/2$个参数。所以LDA没有QDA分类器光滑，LDA拥有更小的方差和更大的预测偏差。样本量小，用LDA；样本量多，用QDA。

3 K最近邻分类

对新的观测，根据其k个最近邻的训练数据的类别，通过多数表决等方式进行类别预判。因此，k最近邻方法不具有显式学习过程。

k最近邻法的三个基本要素：

- k的选择
- 距离度量
- 决策规则

涵盖最邻近k个点的 x 的邻域记作 $N_k(x)$ ，在 $N_k(x)$ 中根据分类决策规则(如多数表决)决定 x 的类别 y

$$y = \arg\max_{c_j} \sum_{x_i \in N_k(x)} I\left\{y_i = c_j\right\}, \quad i = 1, \cdots, N; j = 1, \cdots, L;$$

当$k = 1$时，最近邻分类器偏差较小但方差很大，决策边界很不规则。 当k变大时，方差较低但偏差却增大，将得到一个接近线性的决策边界。 在实际中，可用交叉验证的方法选择k的大小。

4 决策树

分而治之

因变量数值型 - 回归问题 - 回归树

因变量类别型 - 分类问题 - 分类树

- 模型具有可读性
- 预测的速度快

4.1 回归树

出于简化模型和增加模型的可解释性的考虑，通常将自变量空间划为高维矩形，或称为盒子。划分区域的目标是找到使模型的残差平方和RSS最小的矩形区域 $R_1, \ldots, R_J$. RSS的定义是:

$$\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}\left(R_j\right)\right)^2$$

其中, $\hat{y}\left(R_j\right)$ 是第j个矩形区域中训练观测的平均响应值。

将自变量空间划分为J个矩形区域，一般采用自上而下、贪婪的方法：递归二叉分裂。

在执行递归二叉分裂时，先选择自变量X_j 和分割点s，将自变量空间分为两个区域:

$$\{X \mid X_j < s\}$$
$$\{X \mid X_j \geq s\}$$

选择出来的两个区域要能够使RSS尽可能地减少。

重复至某个停止标准，如所有区域包含的观测个数都不大于5.

- 过拟合 -> 剪枝

成本复杂性剪枝

不是考虑每一棵可能的子树，而是考虑以非负调节参数 α 标记的一列子树。每一个 α 的取值对应一棵子树 $T \subset T_0$ 。当 α 值给定时，其对应的子树需使下式最小

$$\sum_{m=1}^{|T|} \sum_{i:\boldsymbol{x}_i \in R_m} \left(y_i - \hat{y}\left(R_m\right)\right)^2 + \alpha|T|$$

这里的$|T|$表示树的叶节点个数， R_m 是第m个叶节点对应的盒子, $\hat{y}\left(R_m\right)$ 是相应的响应预测值。

交叉验证挑选α，使均方预测误差达到最小，从而确定最优子树。

4.2 分类树

递归二叉分裂。RSS的替代指标：

- 分类错误率

$$E_m = 1 - \max_k \hat{p}_{mk}$$

其中， $\hat{p}_{\mathrm{mk}}$ 表示第m个区域的训练观测中第k类所占的比例， $\max_k \hat{p}_{mk}$ 是分类正确率。

但分类错误率在分类树的构建中不够敏感，常用基尼指数和互熵衡量节点纯度。

- 基尼指数

$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}\left(1 - \hat{p}_{mk}\right).$$

- 互熵

$$D_m = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

5 集成学习

- 构建并整合多棵分类树
- 个体分类树应“好而不同”

两大类集成树的产生方法：

- 个体分类树之间不存在强依赖关系、同时生成的并行方法，如Bagging和随机森林
- 个体分类树之间存在强依赖关系、串行生成的序列化方法，如Boosting（Adaboost）

5.1 Bagging

Bagging主要关注降低预测模型的方差。
给定n个独立随机变量$Z_1, \cdots, Z_n$, 假设它们的方差都为 σ^2, 那么样本 均值 $\bar{Z}=\frac{1}{n} \sum_{i=1}^{n} Z_i$ 的方差为 σ^2 / n。
启发：从总体中抽取多个训练集，对每个训练集分别建立预测模型，再对由此得到的全部预测模型求平均，从而降低方差，得到一个集成模型。

即，可以用B个独立的训练集训练出B个模型: $\hat{f}^1(x), \cdots, \hat{f}^B(x)$, 然 后求平均，得到一个低方差的模型:

$$\hat{f}_{\text {avg }}(\boldsymbol{x})=\frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(\boldsymbol{x}).$$

在实际中，不容易得到多个训练集。自助抽样法(Bootstrap)可以解决这个问题。

B足够大，可稳定预测准确率。

可以对某一自变量在一棵个体分类上因分裂导致的基尼指数减少量加总，再在所有B棵个体分类树上求平均值。平均值越大就说明这个自变量越重要。

5.2 随机森林

以决策树为基础构建Bagging分类树的基础上，进一步在决策树的训练过程中引入了**自变量的随机选择**，从而达到对树的去相关（decorrelating），实现对Bagging的改进。

在建立这些个体分类树时，每考虑树上的一个分裂点，都要从全部 的p个自变量中选出一个包含 $q\ (1 \leq q \leq p)$个自变量的随机样本作为候选变量。这个分裂点所用的自变量只能从这q个变量中选择。在每个分裂点处都重新进行抽样，选出q个自变量。若$q=p$，则随机森林就是Bagging. 通常取q为p的平方根。

若个体分类树有高度相关性，对高度相关的变量求平均无法大幅度减少方差。而随机森林对不相关变量求平均，可大幅度降低方差。

样本多样性来自

- Bagging：样本扰动
- 随机森林：样本扰动+自变量扰动

5.3 Adaboost

Boosting：可将弱分类器提升为强分类器。根据分类器的表现对训练样本分布进行调整，使先前分类器错分的训练样本在后续得到更多的关注。

Adaboost是Boosting中的一种，算法：

输入：训练集 $D=\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_m, y_m)\}$; 分类器算法 $\mathfrak{L}$; 训练轮数T;

过程:

(a)$\mathcal{D}_1(\boldsymbol{x})=1/m.$
(b) 对 $t=1, \cdots, T$, 执行:
(c) $h_t=\mathfrak{L}\left(D, \mathcal{D}_t\right)$
(d) $\epsilon_t=P_{\boldsymbol{x} \sim \mathcal{D}_t}\left(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x})\right)$;
(e) 如果 $\epsilon_t>0.5$, 则停止；否则，继续执行;

(f) $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;
(g) 令

$$\begin{aligned}\mathcal{D}_{t+1}(\boldsymbol{x}) &= \frac{\mathcal{D}_f(\boldsymbol{x})\exp(-\alpha_t f(\boldsymbol{x})h_t(\boldsymbol{x}))}{Z_t} \\ &= \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases}\exp(-\alpha_t), & \text{如果 } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{如果 } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x})\end{cases}\end{aligned}$$

其中 Z_t 是归一化常数;
(h) 循环结束.

输出: $H(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{t=1}^{T}\alpha_t h_t(\boldsymbol{x})\right)$.

从偏差-方差权衡角度，Adaboost更关注降低偏差。