

分类模型

授课教师：吴翔
邮箱：wuhsiang@hust.edu.cn

March 18-25, 2019

1 统计学习概述

2 基本分类模型

3 聚类模型

4 树模型

统计学习概述

统计学习方法

统计机器学习 (statistical machine learning) 可分为:

- 有监督学习 (supervised learning) vs 无监督学习 (unsupervised learning): 聚类分析即为典型的无监督学习
- 参数方法 (parametric methods) vs 非参数方法 (non-parametric methods)
- 回归 (regression) 问题 vs 分类 (classification) 问题: 分别针对连续变量和分类变量

测试均方误差的分解

测试均方误差的期望值 (expected test MSE) 可以分解为如下三个部分:

$$E(y - \hat{f}(x))^2 = \underbrace{\text{Var}(\hat{f}(x))}_{\text{variance}} + \underbrace{[\text{Bias}(\hat{f}(x))]^2}_{\text{bias}} + \underbrace{\text{Var}(\epsilon)}_{\text{irreducible}} .$$

- 模型方差 (variance): 针对不同的训练数据, \hat{f} 的变化程度。
- 模型偏误 (bias): 通过相对简化的模型来**近似**真实世界的问题时所引入的误差。

权衡模型偏误与方差

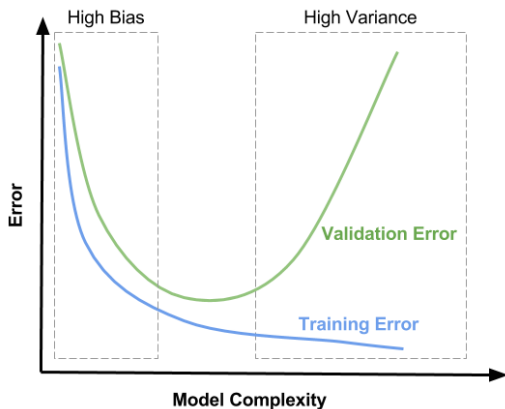


图 1: bias-variance trade-off

如何选择统计模型？

- 传统统计模型的局限：线性回归模型等统计模型通常最小化训练数据的均方误差，但是其测试均方误差 (test MSE) 却较大。换言之，传统统计模型执着于寻求“真实规律”，以致于将一些随机因素**误判**为 f 的真实性质。
- 权衡模型偏误与方差 (bias-variance trade-off)：随着模型灵活性 (或自由度) 的增加，模型方差随之增大，但模型偏误则相应减小 (过度拟合问题)。通过交叉验证 (cross-validation) 来实现两者的权衡。
- 权衡预测精度与可解释性 (accuracy-interpretability trade-off)：诸如 bagging、boosting、support vector machines 等非线性模型具有很高的预测精度，但不易解释；linear models 等易于解释，但预测精度不高。两者的权衡取决于研究目的。

交叉验证

交叉验证将原始数据集分为训练集 (training set) 和验证集 (validation set), 并以验证集的错误率选择最佳模型。

- 留一交叉验证法 (leave-one-out cross validation, LOOCV)
- k 折交叉验证法 (k -fold CV): 将观测集随机分为 k 个大小基本一致的组, 或说折 (fold)。每次选取其中一折作为验证集, 而剩余 $k - 1$ 折作为训练集。通常, 取 $k = 5$ 或 $k = 10$ 。

分类模型验证集错误率:

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^k \text{Err}_k = \frac{1}{k} \sum_{i=1}^k \frac{1}{m_k} \sum_{i=1}^{m_k} I(y_i \neq \hat{y}_i).$$

分类模型概述

预测分类响应变量 (categorical response variable):

- ① 基本分类模型 (basic classifier)
- ② 树模型 (tree-based models)
- ③ 支持向量机 (support vector machine, SVM)
- ④ 聚类模型 (clustering models)

分类模型的评价

Confusion Matrix and ROC Curve

| | | Predicted Class | |
|----------------|-----|-----------------|-----|
| | | No | Yes |
| Observed Class | No | TN | FP |
| | Yes | FN | TP |

TN True Negative
 FP False Positive
 FN False Negative
 TP True Positive

Model Performance

Accuracy = $(TN+TP)/(TN+FP+FN+TP)$

Precision = $TP/(FP+TP)$

Sensitivity = $TP/(TP+FN)$

Specificity = $TN/(TN+FP)$

图 2: confusion matrix

混淆矩阵及评价指标

- 准确率 (accuracy): 指被正确判断的样本的比例。
- 精确率 (precision): 指判断为阳性的样本中, 实际为阳性的比例。
- 灵敏度 (sensitivity): 也称为真阳性率、召回率 (recall rate)。指实际为阳性的样本中, 被正确判断为阳性的比例。
- 特异度 (specificity): 也称为真阴性率。指实际为阴性的样本中, 被正确判断为阴性的比例。

完美的分类器可以达到 100% 的灵敏度, 及 100% 的特异度。但是理论上所有的分类器都会有最小的误差范围, 称为贝叶斯错误率。

可以绘制接收者操作特征曲线 (receiver operating characteristic curve, ROC 曲线), 并结合 ROC 曲线下面积 (area under the curve of ROC, AUC) 来比较不同分类模型。

ROC 曲线与 AUC

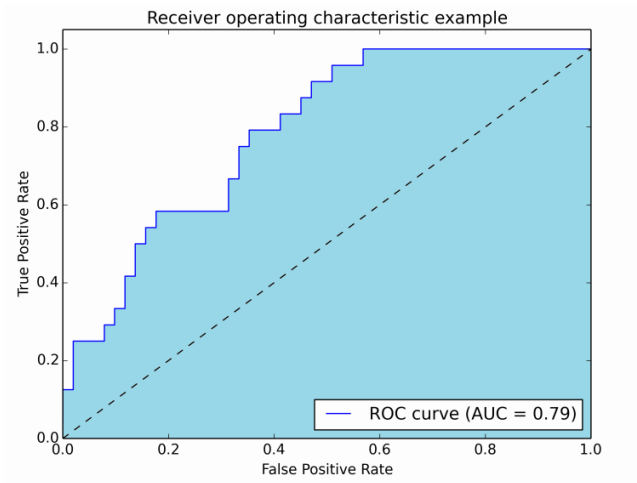


图 3: ROC curve and AUC

基本分类模型

基本分类模型 (basic classifier)

- ① 逻辑斯蒂回归 (logistic regression)
- ② 贝叶斯分类器 (bayes classifier)
- ③ 线性判别分析 (linear discriminant analysis, LDA)
- ④ 二次判别分析 (quadratic discriminant analysis, QDA)
- ⑤ K 最近邻 (K -nearest neighbor, KNN)

logistic 回归

给定 X 条件下事件 Y 发生的概率 $p(X) = \Pr(Y = 1|X)$, 据此可以将发生比 (odd) 的对数建模为 X 的线性函数

$$\log\left[\frac{p(X)}{1 - p(X)}\right] = \beta X.$$

上式左侧称为对数发生比 (log-odd) 或分对数 (logit), 其取值范围在 $(-\infty, \infty)$ 。

当类别 $K \geq 2$ 时, 则采用多类别 logistic 回归模型。

似然函数

可以通过**最大似然估计** (maximum likelihood estimation, MLE) 得到 logistic 回归的参数值。

参数记为 θ , 数据记为 D 。似然函数 (likelihood function) 是参数 θ 的函数, 且定义为给定参数 θ 时, 观测到数据 D 的概率:

$$l(\theta) = p(D|\theta).$$

例如, logistic 回归模型的似然函数

$$l(\beta) = \prod_{i=1}^n p(X_i)^{y_i} [1 - p(X_i)]^{1-y_i}.$$

贝叶斯定理

贝叶斯定理阐述了随机变量 X 和 Y 的条件概率之间的关系：

$$p(Y|X) = \frac{p(X, Y)}{p(X)} = \frac{p(Y) \cdot p(X|Y)}{p(X)}.$$

或从“数据-参数”的视角而言，参数 θ 的后验分布 $\pi(\theta) = p(\theta|D)$ 正比于参数的先验分布 $p(\theta)$ 和似然函数 $l(\theta)$ 之积：

$$\pi(\theta) = \frac{p(\theta)p(D|\theta)}{p(D)} = \frac{p(\theta)l(\theta)}{p(D)}.$$

课堂板书：贝叶斯定理推导及概念解释

贝叶斯定理与分类

对于分类 (categorical) 响应变量 Y 而言, 运用贝叶斯定理:

$$p(Y = k|X = x) = \frac{p(Y = k) \cdot p(X = x|Y = k)}{p(X = x)}.$$

假定 x 是 m 维向量 (即特征数量), 简写为

$$p(C_k|x) = \frac{p(C_k) \cdot p(x|C_k)}{p(x)} \propto p(C_k) \prod_{i=1}^m p(x_i|C_k)$$

贝叶斯分类器

贝叶斯分类器 (bayesian classifier) 选择后验概率 $p(C_k|x)$ 最大的类别, 作为分类结果, 即 $\operatorname{argmax} p(C_k|x)$ 。

可以证明, 贝叶斯分类器将产生最低的测试错误率, 亦即**贝叶斯错误率**。相应于分类的边界, 成为贝叶斯决策边界 (bayes decision boundary)。

问题在于, 如何推导出后验概率 $p(C_k|x)$? 我们需要更多**假设**。

LDA

线性判别分析 (linear discriminant analysis, LDA) 假定 $p(x|C_k) \sim N(\mu_k, \Sigma)$ 。
LDA 即是条件概率 $p(x|C_k)$ 为 (多元) 正态分布时的贝叶斯分类器, 其判别函数 $f(x)$ 为线性函数。

考虑 x 是一维的情况,

$$p(x|C_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x - \mu_k)^2\right],$$

由此根据后验概率 $p(C_k|x)$ 的对数, 得到如下判别函数

$$f_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log[p(C_k)].$$

课堂板书: 推导判别函数

LDA 示意图

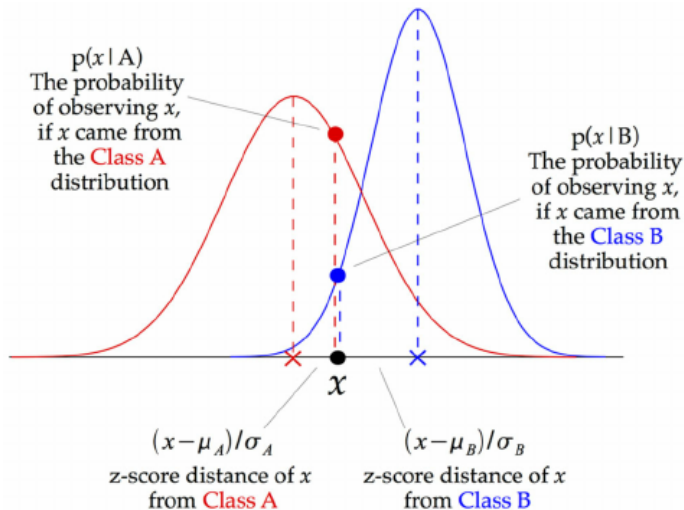


图 4: Illustration of LDA

QDA

二次判别分析 (quadratic discriminant analysis, QDA) 假定 $p(x|C_k) \sim N(\mu_k, \Sigma_k)$ 。QDA 即是条件概率 $p(x|C_k)$ 为 (多元) 正态分布时的贝叶斯分类器, 其判别函数 $f(x)$ 为二次函数。QDA 与 LDA 的差别在于, 协方差矩阵 Σ_k 是否假定相等。

x 为多维向量时, LDA 的判别函数为

$$f_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log[p(C_k)].$$

相应地, QDA 的判别函数为

$$f_k(x) = -\frac{1}{2} x^T \Sigma_k^{-1} x + x^T \Sigma_k^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma_k^{-1} \mu_k + \log[p(C_k)].$$

QDA 示意图

- 左图：对于两个类别，均有 $\rho(X_1, X_2) = 0.7$
- 右图：对于橙色类别， $\rho(X_1, X_2) = 0.7$ ；对于蓝色类别， $\rho(X_1, X_2) = -0.7$

LDA versus QDA

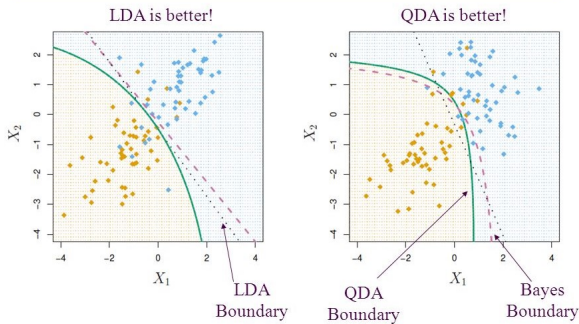


图 5: Illustration of QDA

KNN

通常难以知道 $p(C_k|X)$ 的分布。因而，可以设法估计条件分布 $p(C_k|X)$ 。

对给定正整数 K 和测试观测值 x_0 ， K 最近邻 (KNN) 分类器首先识别训练集中 K 个最靠近 x_0 的点集 A ，继而以集合 A 中的点估计条件概率：

$$p(C_j|x_0) = \frac{1}{K} \sum_{i \in A} I(y_i = j).$$

最后，运用贝叶斯规则将测试观测值 x_0 分到后验概率 $p(C_j|x_0)$ 最大的类中。

- KNN 作为典型的非参数方法 (non-parametric methods)，能够产生一个近似于最优贝叶斯分类器的效果
- K 值的选择对 KNN 分类器的效果有根本影响

KNN 示意图

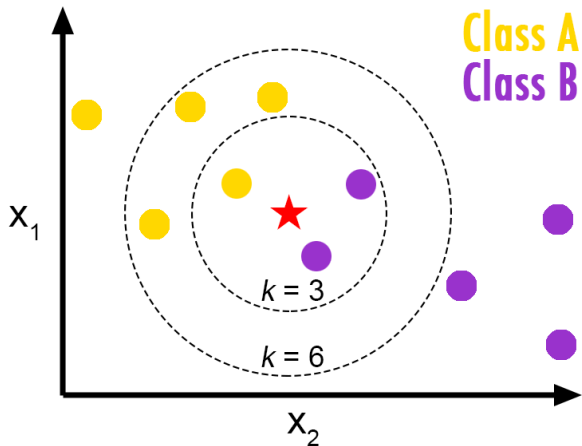


图 6: Illustration of KNN

模型讨论

- LDA 中的判别函数与 logistic 回归中的对数发生比 (log-odd) 均是 x 的线性函数, 因而二者都产生一个线性决策边界, 且分类结果相近。若 $p(x|C_k) \sim N(\mu_k, \Sigma)$ 近似成立, 则 LDA 优于 logistic 回归; 反之, logistic 回归优于 LDA。
- KNN 作为非参数方法, 对决策边界的形状没有做出任何假设。因而当决策边界高度非线性时, KNN 优于 LDA 和 logistic 回归。
- QDA 是线性决策边界 (LDA 和 logistic 回归) 和非参数 KNN 方法的折衷方案, 采用了二次函数形式的决策边界。

案例：股票市场走势预测

聚类模型

聚类模型 (clustering models)

聚类分析 (clustering) 试图从观测数据中寻找**同质子类**，属于**无监督学习** (unsupervised learning) 的范畴。基本聚类模型包括：

- ① K 均值聚类 (K -means clustering)
- ② 层次聚类 (hierarchical clustering)

原理：将观测样本分割到不同的类 (cluster) 中，使每个类内的观测彼此相似，而不同类中的观测彼此差异很大。

课堂讨论：比较聚类与 PCA、FA、ANOVA、线性回归

K 均值聚类

k 均值聚类通过**最小化类内差异**而得到聚类结果：

$$\min \sum_{k=1}^K W(C_k).$$

$W(\cdot)$ 衡量类内差异，例如可以采用欧氏距离计算。

k 均值聚类算法如下：

- ① 为每个观测样本随机分配一个初始类 $k(1 \leq k \leq K)$ 。
- ② 重复以下操作，直至类的重分配停止为止：
 - 分别计算 K 个类的中心。第 k 个类中心是其类内 p 维观测样本的均值向量。
 - 将每个观测样本分配到距离其最近的类中心所在的类中。

课堂讨论：是否存在其它思路？

K 均值聚类示意图

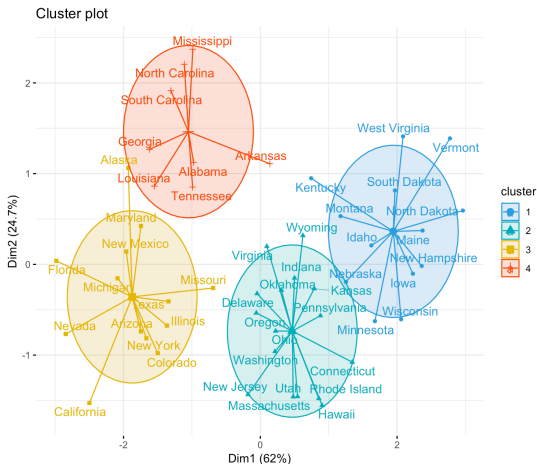


图 7: Illustration of K-means clustering

层次聚类

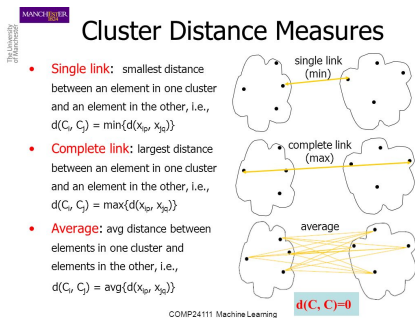
层次聚类 (hierarchical clustering) 算法如下:

- ① 每个观测样本自成一类, 共有 n 个初始类。计算所有 $n(n-1)/2$ 对观测样本 (类) 之间的相异度。
- ② 令 $i = n, n-1, \dots, 2$:
 - 在 i 个类中, 比较任意两类间的相异度, 找到相异度最小的两类, 将其合并起来。用两个类之间的相异度表示这两个类在谱系图中交汇的高度。
 - 计算剩下的 $i-1$ 个新类中, 每两个类间的相异度。

层次聚类采用逐步归并的方式, 构建了谱系图 (dendrogram), 从而允许任意的类别数量。

距离测度

通常采用聚类来衡量相异度，常见距离形式包括：最长 (complete) 距离法、类平均法 (average)、最短 (single) 距离法和重心法 (centroid)。



5

图 8: Distance measures in hierarchical clustering

层次聚类示意图

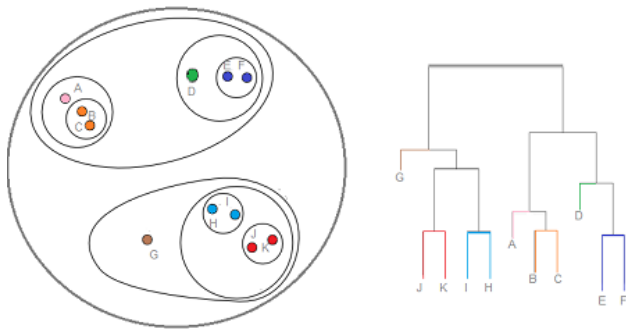


图 9: Illustration of hierarchical clustering

树模型

树模型 (tree-based models)

- ① 决策树 (decision tree)
- ② 装袋法 (bagging)
- ③ 随机森林 (random forest)
- ④ 提升法 (boosting)

决策树

- 步骤：(1) 根据分层 (stratifying) 或者分割 (segmenting) 的方式将预测变量空间划分为一系列的区域 (area); (2) 给定观测样本的预测值, 等于所属区域中训练集的平均值 (连续变量) 或者**众数** (分类变量)。
- 特征：划分预测变量空间的分割规则可以概括为一棵树
- 适用范围：回归问题及分类问题

课堂板书：与回归模型预测值比较

错误率的衡量

均方误差 MSE 适用于衡量回归问题的误差率。假定第 m 个区域第 k 个类别所占比例为 p_{mk} , 则分类问题的错误率衡量指标包括:

- 分类错误率 (classification error rate): 此区域的训练集中非最常见类所占的比例, $E = 1 - \max(p_{mk})$
- 基尼系数 (Gini index): $G = \sum_{k=1}^K p_{mk}(1 - p_{mk})$
- 互熵 (cross entropy): $D = -\sum_{k=1}^K p_{mk} \log(p_{mk})$

后两个指标对节点纯度更加敏感。

预测变量空间的划分

- 形状：理论上，预测变量空间可以划分为任意形状。但为了简化模型和增强可解释性，通常划分为**高维矩阵**，亦即盒子 (box)。
- 算法：采用**递归二叉分割** (recursive binary splitting) 算法将预测变量空间划分为不同的盒子。递归二叉分割从树的根节点开始依次分割预测变量空间；每个分割点都产生两个新的分支；每一次分割空间都是**局部**最优的。
- 步骤
 - ① 考虑所有预测变量 (X_1, \dots, X_p) ，从中选择预测变量 X_j 和分割点 s ，将预测变量空间分割为 $R_1(j, s) = X|X_j < s$ 和 $R_2(j, s) = X|X_j \geq s$ 两个盒子，使训练集的错误率最小。
 - ② 针对新的预测变量空间，重复步骤 1，直至符合某个停止规则（例如，每个盒子观测值个数小于等于 5）为止。

课堂讨论：(1) 分类变量的处理？(2) 对比向前逐步选择 (forward stepwise selection)

树的剪枝

递归二叉分割采用了局部最优算法，且有可能造成数据的过度拟合，从而在测试集上预测效果不佳。我们希望选择更简单的树，从而降低模型方差。可行的解决方案包括：仅当分割使训练集的 MSE 或 Err 的减少量超过某阈值时，才分割树的节点。

可以从树 T_0 开始，通过剪枝 (prune) 得到子树 (subtree)。**代价复杂性剪枝** (cost complexity pruning) 在训练集的误差/错误率衡量公式中加入调整系数 α ，以权衡模型的精确性和复杂性：

$$\sum_{m=1}^{m_T} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \text{ or } \sum_{m=1}^{m_T} \sum_{i: x_i \in R_m} [1 - \max(p_{mk})] + \alpha |T|$$

$|T|$ 为树 T 的节点个数。最后，可以通过交叉验证选择最佳调整系数 α 。

课堂讨论：比较 lasso 与代价复杂性剪枝

决策树示例

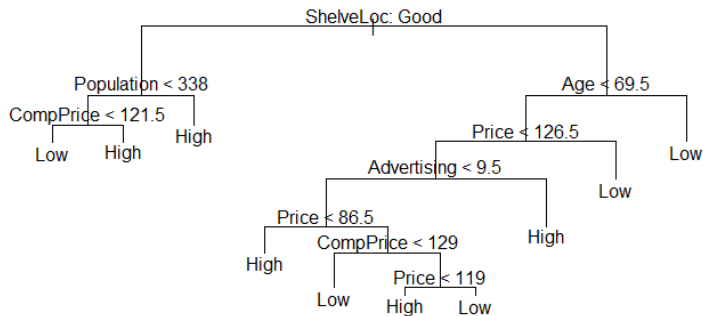


图 10: Illustration of decision tree

决策树的优缺点

- 优点：易于图示化，解释性较强，且更加接近人的决策模式
- 缺点：预测准确性通常低于其它回归和分类方法

因而，通常采用装袋法、随机森林、提升等方法组合大量决策树，从而显著提高树的预测效果。

装袋法

决策树有着高方差 (high variance), 而自助法 (bootstrapping) 可以用以降低方差, 其原理为: n 个独立观测值 Z_1, \dots, Z_n 的方差都为 σ^2 , 它们的均值 \bar{Z} 的方差为 σ^2/n .

装袋法 (bagging), 也称自助法聚集 (bootstrap aggregation), 从原始数据集中重抽样得到 B 个自助抽样训练集, 据此建立 B 棵回归树, 在计算相应预测值的均值 (连续变量) 或众数 (分类变量)。计算众数, 也称为**多数投票** (majority vote) 规则。当 B 增大到一定规模后, 就无法再降低模型误差了。因此, 只要 B 充分大即可。

装袋法的 B 颗树, 可以证明, 平均每棵树能利用约 $2/3$ 的观测样本。对特定树而言, 剩余 $1/3$ 的观测样本称为**袋外** (out-of-bag, OOB) 观测样本。可以用所有将第 i 个观测样本作为 OOB 的树来预测第 i 个观测样本的响应值。由此, 产生整体的 OOB 均方误差 (连续变量) 或 **OOB 分类误差** (分类变量)。实施 OOB 方法比交叉验证更为便利。

装袋法示意图

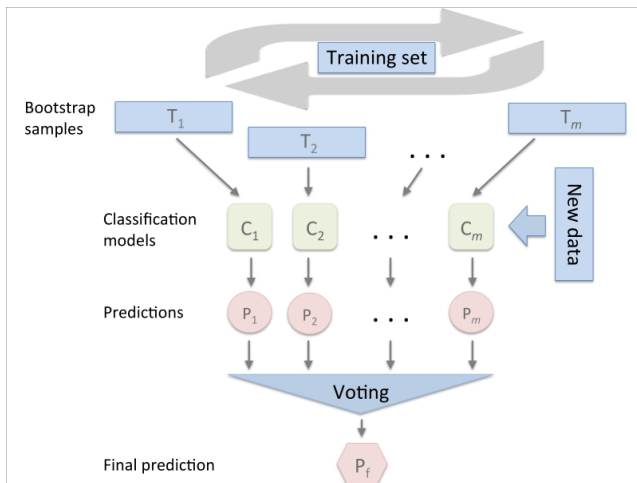


图 11: Illustration of bagging (bootstrap aggregation)

模型解释性

单棵决策树的结果易于解释，而采用多棵树得到的结果则难以解释。因而，装袋法以牺牲解释性为代价获得了更高的预测准确性。

可以计算**变量重要性** (variable importance)，从而获得装袋法的解释性。在装袋法建模过程中，记录下任一给定预测变量引发的分割而减少的误差量，并在所有 B 棵树上求平均。结果越大，标明变量越重要。

课堂讨论：回归模型中变异的分解与解释

随机森林

随机森林 (random forest) 沿袭了装袋法的思路，并进行了改进：每次分割的时候，从全部的 p 个预测变量中**随机**选择 $m \approx \sqrt{p}$ 个预测变量实施分割。

随机森林对树**去相关** (decorrelate)，从而减少 B 棵树的均值或众数的方差。尤其是存在较强预测变量，或者预测变量之间相关度较高时，随机森林能有效改进装袋法。

随机森林示意图

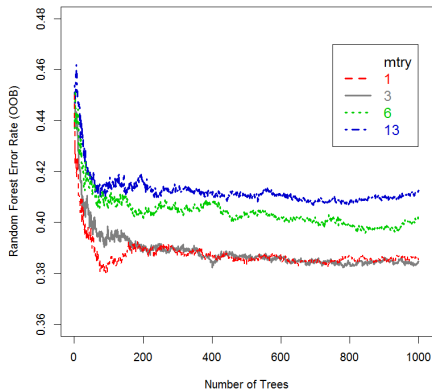


图 12: Number of trees

提升法

提升法 (boosting) 沿袭了装袋法的思路, 并进行了改进: B 棵树按**顺序** (sequentially) 生成, 每棵树的构建都需要用到之前生成的树的信息, 采用现有模型的**残差信息** (X, ϵ) 生成决策树。回归情形下, 提升法的算法如下:

- ① 对训练集中所有观测样本 i , 令 $\hat{f}(x) = 0$, $\epsilon_i = y_i$ 。
- ② 对 $b = 1, 2, \dots, B$, 重复以下过程:
 - 对训练数据 (X, ϵ) 建立一棵有 d 个分割点 (亦即 $d + 1$ 个节点) 的树 \hat{f}^b 。
 - 将压缩后的新树加入模型以更新 \hat{f} : $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$ 。
 - 更新残差: $\epsilon_i \leftarrow \epsilon_i - \lambda \hat{f}^b(x_i)$ 。
- ③ 输出经过提升的模型, $\hat{f}(x) = \sum_{b=1}^B \lambda \hat{f}^b(x)$ 。

提升法 (续)

提升法采用舒缓 (learning slowly) 训练模型的方法, 即利用现有残差而非原始响应变量 y 作为响应值。压缩参数 λ 使学习过程变得缓慢。通过充分利用之前生成的树的信息, 有效提高预测或分类准确度。

提升方法的调整参数包括:

- 树的总数 B : 与装袋法和随机森林不同, B 值过大时会出现过度拟合, 因而通常用交叉验证来选择 B 值。
- 压缩参数 λ : 控制提升法的学习速度, 通常取极小的值, 如 $\lambda = 0.01$ 或 $\lambda = 0.001$ 。压缩参数 λ 越小, 就需要越多的树, 即越大的 B 值。
- 每棵树的分割点数 d : 控制整个提升模型的复杂程度。

提升法通常需要的树更小 (d 值很小), 因为生成一棵特定的树已经考虑了其它已有的树。

提升法示意图

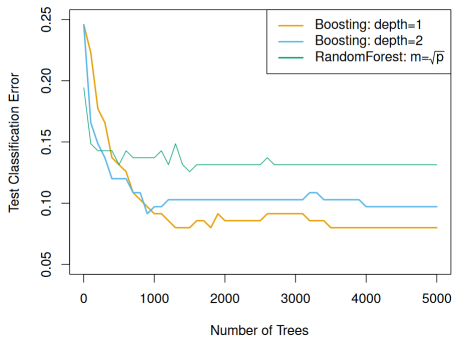


图 13: Illustration of boosting