法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象学院
  - 新浪微博：小象AI学院

机器学习模型用于评分卡模型

–GBDT模型

目录

GBDT模型简介

GBDT模型调参

变量重要性的衡量

GDBT模型的简介

集成模型的形式

三种常见的集成学习框架：bagging，boosting和stacking

Bagging

从训练集从进行(又放回地)抽样组成每个基模型所需要的子训练集，并且并行地训练基模型。最终对所有基模型预测的结果进行综合产生最终的预测结果

Boosting

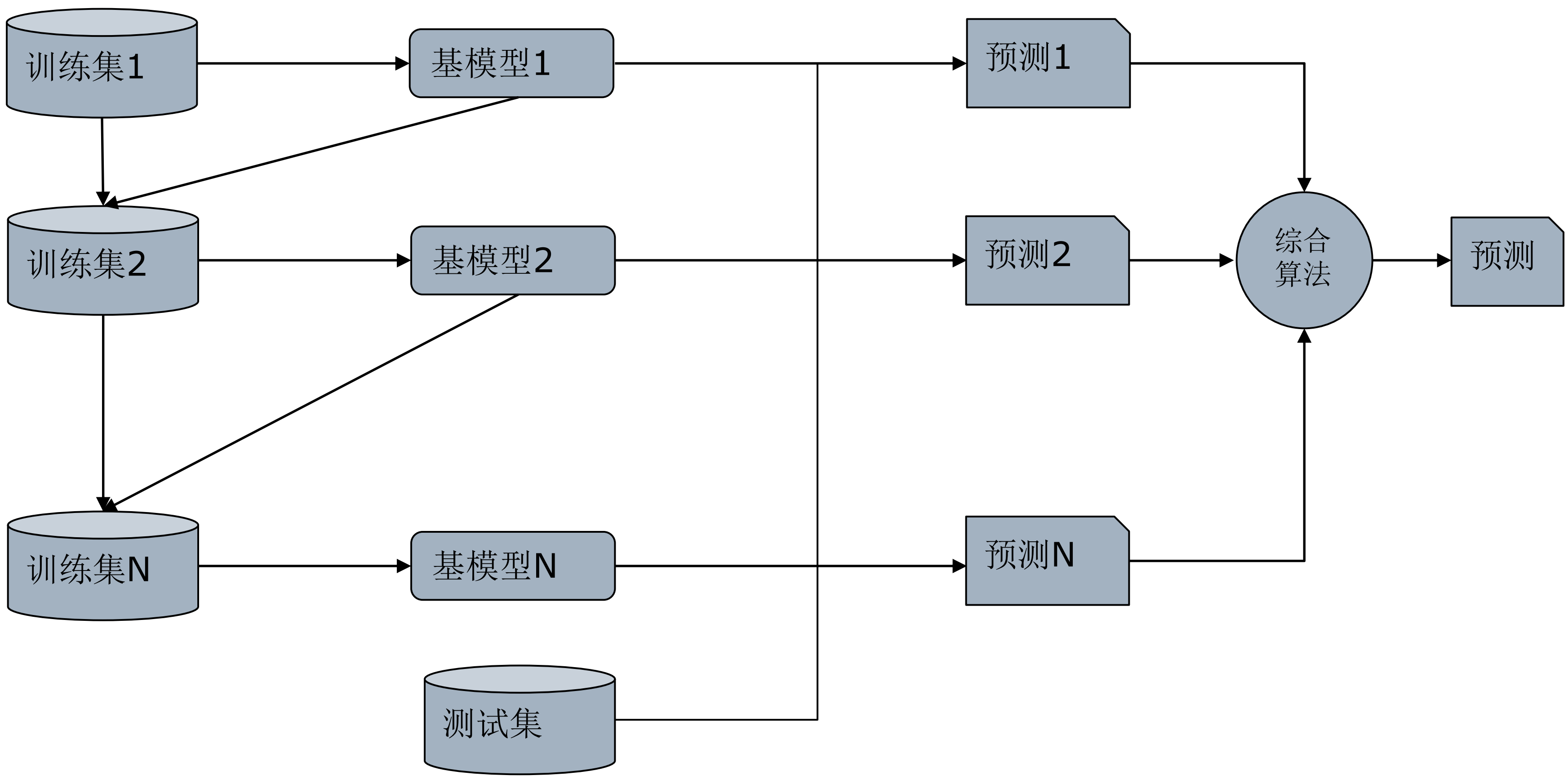
训练过程为串型，基模型按次序一一进行训练，基模型的训练集按照某种策略每次都进行一定的更新。对所有基模型预测的结果进行线性综合产生最终的预测结果。

Stacking

将训练好的所有基模型对训练基进行预测，第j个基模型对第i个训练样本的预测值将作为新的训练集中第i个样本的第j个特征值，最后基于新的训练集进行训练。同理，预测的过程也要先经过所有基模型的预测形成新的测试集，最后再对测试集进行预测

GDBT模型的简介

- 集成模型的形式之Boosting：训练与预测



GDBT模型的简介

- GDBT的原理

一般的有监督机器学习问题

假设训练数据

- $X = \{x_1, x_2 \dots, x_n\}$,n个样本
- $Y = \{y_1, y_2 \dots, y_n\}$

- 损失函数(loss function)

$$L(F(X), Y)$$

- 目标,寻找一个F

$$F^* = argmin_F \mathrm{L}(\mathrm{Y}, \mathrm{F}(\mathrm{x}))$$

GDBT模型的简介

常见的损失函数

Squared error(回归)

$$L(Y,F(X))=\sum_{i=1}^{n}(F(x_i)-y_i)^2$$

hinge loss(SVM)

$$L\big(Y,F(X)\big)=\sum_{i=1}^{n}\max(0,1-y_i*F(x_i))$$

Logistic regression loss

$$L\big(Y,F(X)\big)=\sum_{i=1}^{n}\log(1+\exp(-y_i*F(x_i))$$

GDBT模型的简介

- 参数估计的目标

对于参数化的模型F即$F(X;P)$,

$$F^* = argmin_F L\big(y, F(X)\big) = argmin_P\ L(Y, F(X;P))$$

- 参数P的求解

假设有一个初始解P_{m-1},如何寻找一个更优解 P_m

$$p_m = p_{m-1} + \rho * \Delta p$$

其中，ρ是个正数

GDBT模型的简介

- 梯度法

不妨令$L(Y, F(X;P)) = \varphi(P)$，根据(一阶)泰勒展开，有

$$L(Y, F(X;Pm)) = \varphi(p_m) = \varphi(p_{m-1} + \rho * \Delta p)$$

$$\approx \varphi(p_{m-1}) + \frac{\partial \varphi(P)}{\partial P}|_{P=P_{m-1}} * \rho * \Delta P$$

由于要求迭代的过程使得L下降，故$\varphi(p_m) < \varphi(p_{m-1})$，即

$$\frac{\partial \varphi(P)}{\partial P}|_{P=P_{m-1}} * \rho * \Delta P < 0$$

由于$\rho > 0$，可以选择

$$\Delta P = -\frac{\partial \varphi(P)}{\partial P}|_{P=P_{m-1}}$$

GDBT模型的简介

梯度提升法

假设最优解是$F^* = \sum_{i=1}^{M} f_i(X)$

如果有一个初始的$F_{m-1}(X)$,如何找到一个更优解$F_m(X)$

$$F_m(X) = F_{m-1}(X) + \rho * f(x)$$

令$L\big(Y, F(X)\big) = \varphi\big(F(X)\big)$

$$L\big(Y, F_m(X)\big) = \varphi\big(F_{m-1}(X) + \rho f(X)\big)$$

$$\approx \varphi\big(F_{m-1}(X)\big) + \frac{\partial \varphi\big(F(X)\big)}{\partial F(X)}\Big|_{F(X)=F_{m-1}(X)} * \rho f(x)$$

$$f(x) = -\frac{\partial \varphi\big(F(X)\big)}{\partial F(X)}\Big|_{F(X)=F_{m-1}(X)}$$

GDBT模型的简介

- 梯度提升

$$L(Y,F(X)) = \sum_{j=1}^{n} \log(1+\exp(-y_j * F(x_j)))$$

$$\frac{\partial L(Y,F(X))}{\partial F(X_i)} = \frac{\partial \sum_{j=1}^{n} \log(1+\exp(-y_j * F(x_j)))}{\partial F(x_j)}$$

$$= \frac{\partial \log(1+\exp(-y_i * F(x_i)))}{\partial F(x_i)} = \frac{\exp(-y_i * F(x_i)) * (-y_i)}{1+\exp(-y_i * F(x_i))}$$

$$f(x_i) = -\frac{\partial L(Y,F(X))}{\partial F(x_i)}\Big|_{F(X)=F_{m-1}(X)}$$

$$= -\frac{\exp(-y_i * F_{m-1}(x_i)) * (-y_i)}{1+\exp(-y_i * F_{m-1}(x_i))} = \frac{y_i}{1+\exp(y_i * F_{m-1}(x_i))}$$

GDBT模型的简介

- 梯度提升

若令$r_{mi}=\frac{y_i}{1+\exp(y_i*F_{m-1}(x_i))}$,则需要对$\{x_i, sign(r_{mi})\}$拟合分类树，且叶子节点的输出值为

$$c_{mj}=argmin\sum_{x\in R_{mj}}\log(1+\exp(-y_i(F_{m-1}(x_i)+c)))$$

由于上式较难优化，一般用近似解：

$$c_{mj}=sign(\sum_{x_i\in R_{mj}}r_{mi}\Big/\sum_{x_i\in R_{mj}}|r_{mi}|(1-|r_{mi}|))$$

目录

GBDT模型简介

GBDT模型调参

变量重要性的衡量

GDBT模型调参

- GDBT模型的几个重要参数

框架层面参数

n_estimators

弱学习器的最大迭代次数，或者说最大的弱学习器的个数。一般来说取值太小容易欠拟合；太大又容易过拟合，一般选择一个适中的数值。

Subsample

即子采样，取值为(0,1]。注意这里的子采样和随机森林不一样，随机森林使用的是放回抽样，而这里是不放回抽样。如果取值为1，则全部样本都使用；如果取值小于1，则只有一部分样本会去做GBDT的决策树拟合。选择小于1的比例可以减少方差，即防止过拟合，但是会增加样本拟合的偏差，因此取值不能太低。推荐在[0.5, 0.8]之间，默认是1.0，即不使用子采样。

GDBT模型调参

- GDBT模型的几个重要参数(续)

分类/回归树层面参数

最大特征数max_features

默认是“None”,即 考虑所有的特征数。如果是整数，代表考虑的特征绝对数。如果是浮点数，代表考虑特征百分比。一般来说，如果样本特征数不多，比如小于50，可以用默认的“None”，如果特征数非常多，需要进行网格搜索。

决策树最大深度max_depth:

默认可以不输入，此时决策树在建立子树的时候不会限制子树的深度。一般来说，数据少或者特征少的时候可以不管这个值。如果模型样本量多，特征也多则需要限制最大深度，取值取决于数据的分布。常用的可以取值10-100之间。

GDBT模型调参

- GDBT模型的几个重要参数(续)

分类/回归树层面参数

内部节点再划分所需最小样本数min_samples_split

如果某节点的样本数少于min_samples_split，则不会继续再进行划分。默认是2.如果样本量不大，不需要调节这个值。如果样本量数量级非常大，则推荐增大这个值。

叶子节点最少样本数min_samples_leaf

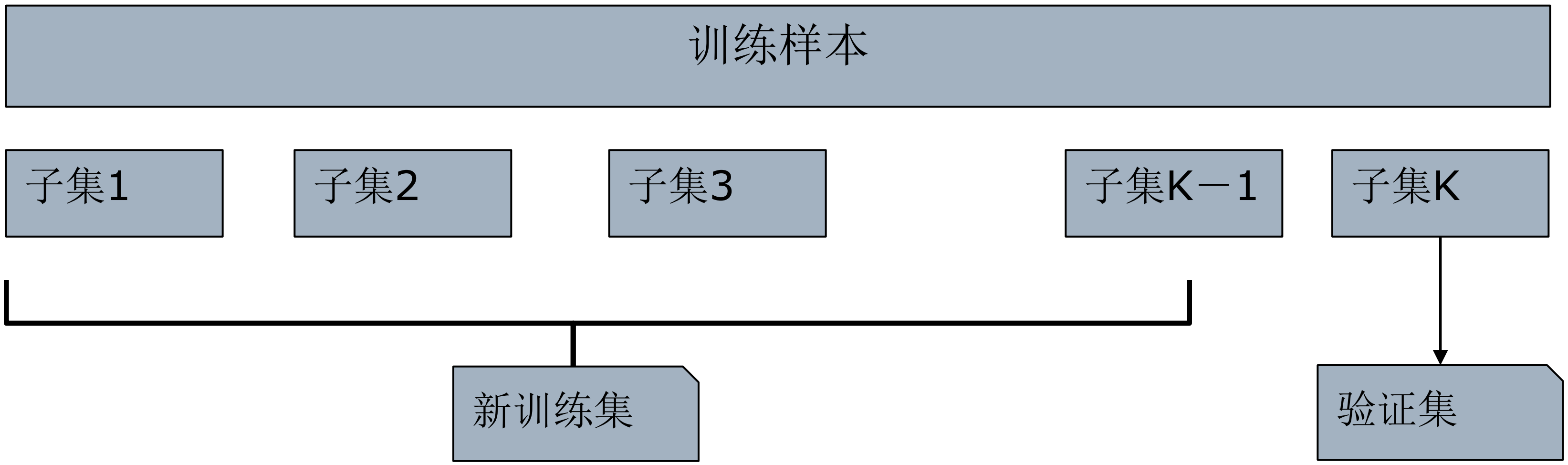
如果某叶子节点数目小于样本数，则会和兄弟节点一起被剪枝。默认是1,可以输入最少的样本数的整数，或者最少样本数占样本总数的百分比。如果样本量不大，不需要调节这个值。如果样本量数量级非常大，则推荐增大这个值。

GDBT模型调参

如何进行参数调节

K折交叉验证

- 选择K的值(比如10)，将数据集分成K等份
- 使用其中的K-1份数据作为训练数据进行模型的训练
- 使用一种度量测度在另外一份数据(作为验证数据)衡量模型的预测性能



GDBT模型调参

- 如何进行参数调节(续)

交叉验证的优点

- 交叉验证通过降低模型在一次数据分割中性能表现上的方差来保证模型性能的稳定性
- 交叉验证可以用于选择调节参数、比较模型性能差别、选择特征

交叉验证的缺点

- 交叉验证带来一定的计算代价，尤其是当数据集很大的时候，导致计算过程会变得很慢

GDBT模型调参

- 基于k折交叉验证的网格搜索法

GridSearchCV，其作用是自动调参。将每个参数所有可能的取值输入后可以给出最优化的结果和参数。但是该方法适合于小数据集，对于大样本很难得出结果。此时可以使用基于贪心算法的坐标下降进行快速调优：

先拿当前对模型影响最大的参数调优，直到最优化，再拿下一个影响最大的参数调优，如此下去，直到所有的参数调整完毕。这个方法的缺点就是可能会调到局部最优而不是全局最优，时间效率较高。

目录

GBDT模型简介

GBDT模型调参

变量重要性的衡量

变量重要性的衡量

- 变量重要性

特征的全局重要性通过其在单棵树的重要性的平均值来衡量。

$$importance\ of\ x = \sum_{t=1}^{L-1} i_t\, 1(v_t = x)$$

其中，L是叶子节点个数，则L－1是非叶子节点个数，v_t是和特征x相关联的节点，i_t是分裂后纯度比分裂前纯度的增加值。

疑问

- 小象问答官网
  - http://wenda.chinahadoop.cn

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：小象学院
- 新浪微博：小象AI学院