

Gradient Boost

深度学习专业班

欧阳若飞



全球人工智能学院

国内首家专注于AI技术职业化教育平台



主要内容

❑ Bagging (简介)

- 三个学渣一起考试

❑ Random Forest (简介)

- 三个偏科学生一起考试

❑ Boosting (重点)

- 一个学霸不停的刷错题考试超神
- AdaBoost
- XGBoost



Gradient Boost

Gradient Descend

在参数空间寻找最优的参数
最小化损失函数

调节的是参数

Gradient Boost

在函数空间寻找最优的函数
最小化损失函数

调节的是函数



Gradient Boost

$$H(x) = \sum_{k=1}^K h_k(x)$$

此时为了方便理解 我们讨论回归的情况

$$H_K(x) = \sum_{k=1}^{K-1} h_k(x) + h_K(x)$$
$$\sum_{i=1}^N \mathcal{L}(y_i, H_K(x_i)) = \sum_{i=1}^N (y_i - H_K(x_i))^2$$

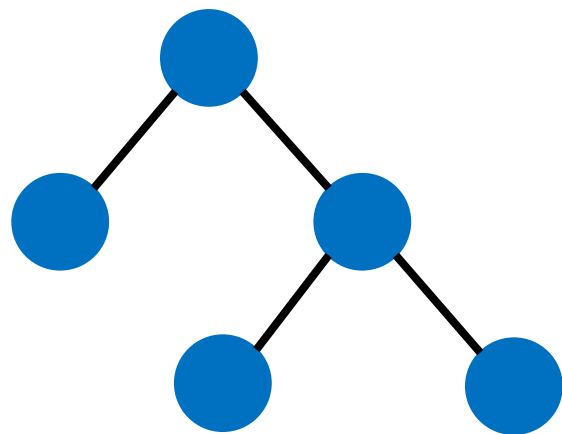
$$H_K(x) = H_{K-1}(x) + h_K(x)$$
$$\sum_{i=1}^N \mathcal{L}(y_i, H_{K-1}(x_i) + h_K(x_i)) = \sum_{i=1}^N (y_i - H_{K-1}(x_i) - h_K(x_i))^2$$

每次学习上一次结果的残差 $r_i^K = y_i - H_{K-1}(x_i)$



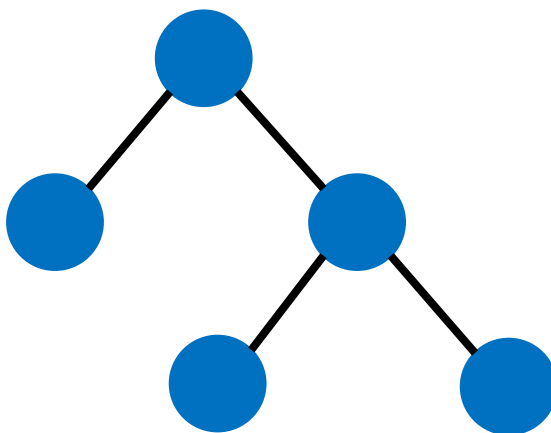
Gradient Boost

$$(X, Y)$$



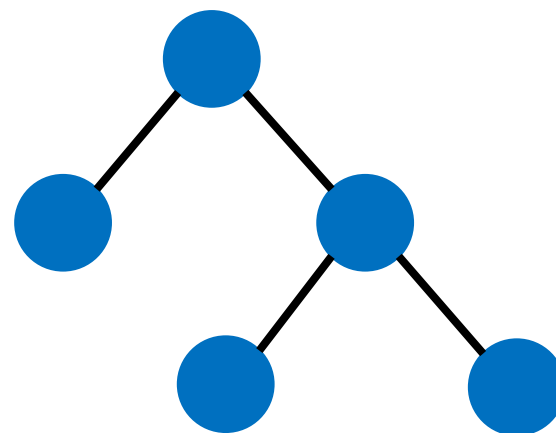
$$Y_1$$

$$(X, Y - Y_1)$$



$$Y_2$$

$$(X, Y - Y_1 - Y_2)$$

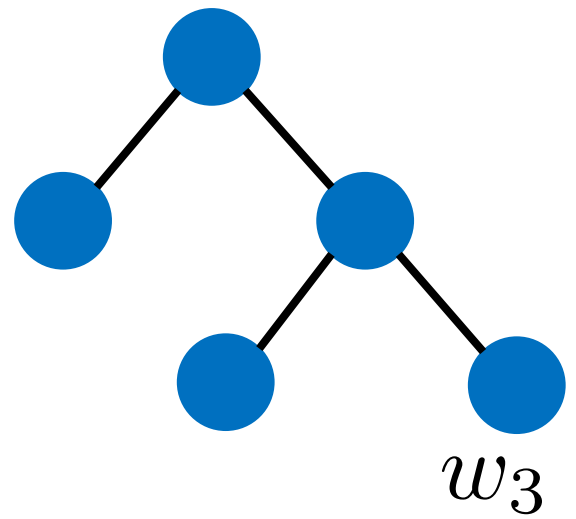
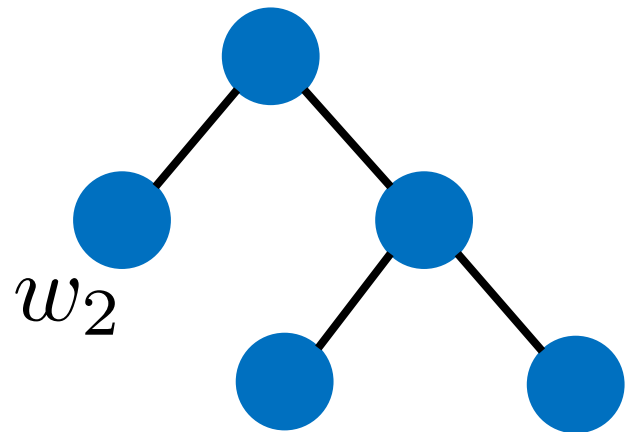
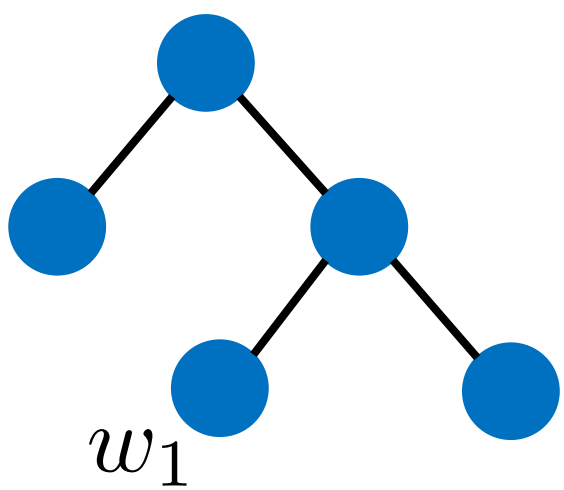


$$Y_3$$

$$\hat{Y} = Y_1 + Y_2 + Y_3$$



Gradient Boost



$$H(x) = h_1(x) + h_2(x) + h_3(x) = w_1 + w_2 + w_3$$



Gradient Boost

初始化 $H_0(x) = \operatorname{argmax} \sum_{i=1}^N \mathcal{L}(y_i, h_0(x))$

for k in range(1, K):

 计算残差 $r_i^k = \frac{\partial \mathcal{L}}{\partial f_{k-1}(x_i)}$

 训练分类器 $h_k(x)$

$$H_k(x) = H_{k-1}(x) + h_k(x)$$



XGBoost

最基本的Gradient Boost使用一阶导数近似残差

我们都知道加入二阶导数的近似效果好

所以就有了XGBoost

XGBoost是数据建模比赛的大杀器!



XGBoost

泰勒展开

$$f(x + \Delta x) \approx f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)\Delta x^2$$

$$\sum_{i=1}^N \mathcal{L}(y_i, \underbrace{H_{k-1}(x_i)}_x + \underbrace{h_k(x_i)}_{\Delta x}) = \sum_{i=1}^N \mathcal{L}(y_i, H_{k-1}(x_i)) + \underbrace{a_k}_{f'(x)} h_k(x_i) + \frac{1}{2} \underbrace{b_k}_{f''(x)} h_k(x_i)^2$$

最小化该目标函数等价于

$$\min \sum_{i=1}^N a_k h_k(x_i) + \frac{1}{2} b_k h_k(x_i)^2$$

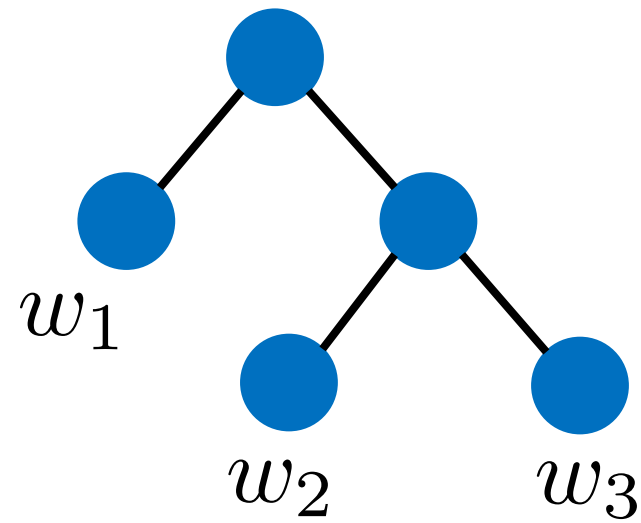


XGBoost

加入正则项

$$\min \sum_{i=1}^N a_k h_k(x_i) + \frac{1}{2} b_k h_k(x_i)^2 + \Omega(h_k)$$

$$\Omega(h_k) = \underset{\substack{\uparrow \\ \text{叶子节点个数}}}{\gamma T} + \frac{1}{2} \lambda \sum_{j=1}^T \underset{\substack{\uparrow \\ \text{叶子节点权重}}}{w_j^2}$$



$$T = 3$$



XGBoost

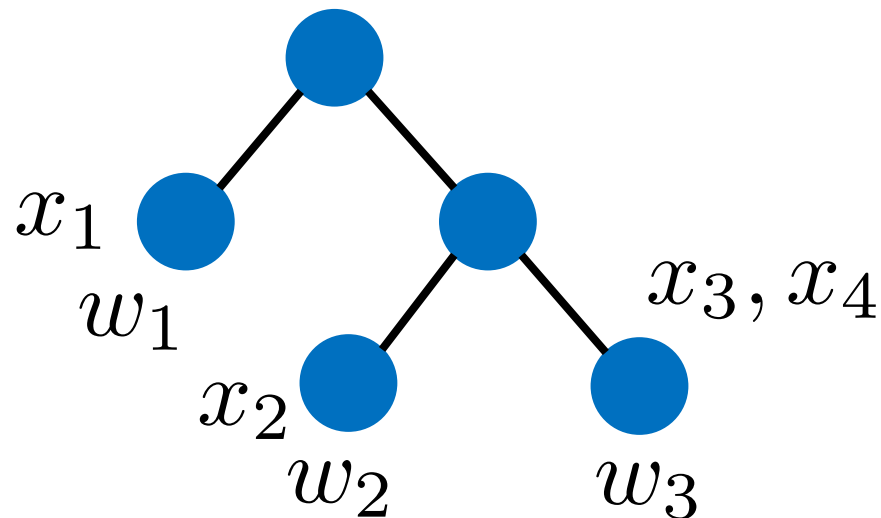
$$\sum_{i=1}^N a_k h_k(x_i) + \frac{1}{2} b_k h_k(x_i)^2 + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

$$\sum_{j=1}^T \left(\left(\sum_{i \in I_j} a_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} b_i + \lambda \right) w_j^2 \right) + \gamma T$$

数每个数据变成数每个叶子节点上的数据

对w求导：

$$\sum_{j=1}^T \left(\underbrace{\left(\sum_{i \in I_j} a_i \right)}_{A_j} + \underbrace{\left(\sum_{i \in I_j} b_i + \lambda \right)}_{B_j} w_j \right) = 0$$



$$w_j = -\frac{A_j}{B_j + \lambda}$$

$$\mathcal{L} = -\frac{1}{2} \sum_{j=1}^T \frac{A_j^2}{B_j + \lambda} + \gamma T$$



XGBoost

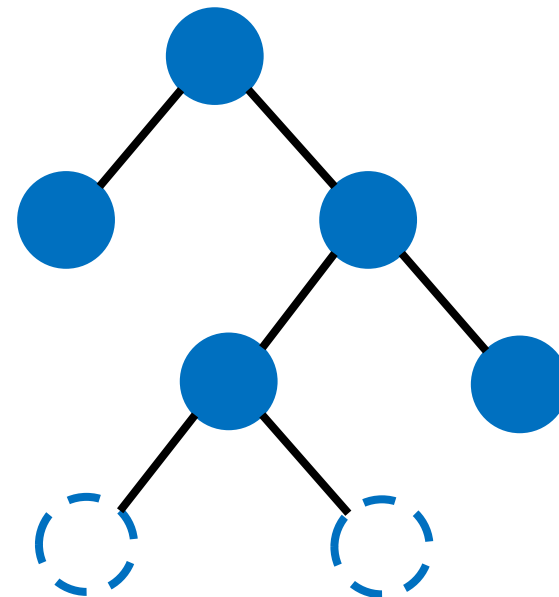
选择下个分裂点:

$$\mathcal{L} = -\frac{1}{2} \sum_{j=1}^T \frac{A_j^2}{B_j + \lambda} + \gamma T$$

分裂前 $\mathcal{L} = -\frac{1}{2} \frac{(A_L + A_R)^2}{(B_L + B_R) + \lambda} + \gamma T + \text{Rest}$

分裂后 $\mathcal{L} = -\frac{1}{2} \left(\frac{A_L^2}{B_L + \lambda} + \frac{A_R^2}{B_R + \lambda} \right) + \gamma(T + 1) + \text{Rest}$

$$\Delta \mathcal{L} = -\frac{1}{2} \left(\frac{(A_L + A_R)^2}{(B_L + B_R) + \lambda} - \frac{A_L^2}{B_L + \lambda} - \frac{A_R^2}{B_R + \lambda} \right) - \gamma$$



每次取cost下降最多的点去做分裂



XGBoost

XGBoost的优势

分类回归都能用 很多实际问题可以直接解决

树型模型的可解释型非常强

无需做过多的数据预处理

XGBoost的代码支持多种平台，自带GPU和分布式功能



XGBoost参数详解

eta 学习速率 learning rate

$$H_k(x) = H_{k-1} + \eta h_k(x) \quad \alpha_k \text{ in AdaBoost}$$

gamma 叶子节点个数的正则项

lambda 叶子节点权重的L2正则项

alpha 叶子节点权重的L1正则项

$$\Omega(h_k) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

max_depth 单棵树的深度 在Boosting模型里树不能大

min_child_weight 叶子节点最小权重

subsample 按行取样的概率 取数据的子集

col_sample_bytree 按列取样的概率 取特征的子集

scale_pos_weight 正样本的权重 调节不平衡样本



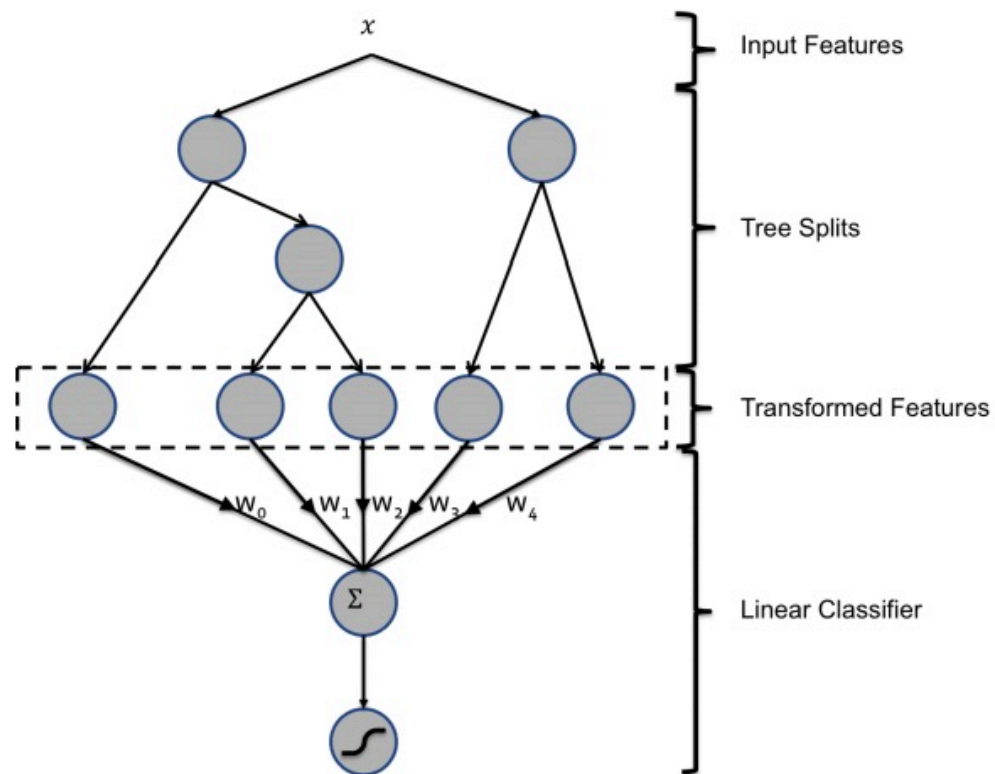
XGBoost参数详解

tree_method: hist 节点分裂是算法的速度瓶颈 hist相当于LightGBM的实现

在categorical feature上XGBoost的计算不如CatBoost



XGBoost生成新特征



Practical Lessons from Predicting Clicks on Ads at Facebook



XGBoost生成新特征

除了可以用逻辑回归在新特征上做分类，也可以使用其他模型

具有实战意义的模型有libfm

因为libfm擅长处理稀疏的特征，而决策树生成的特征比较稀疏

利用XGBoost生成的新特征做聚类分析

Topic Model能把句子按主题聚类

XGBoost对每个数据点生成的特征向量可以看做一个句子

这样可以很容易的获取数据的聚类结果

优势: 不用做繁琐的数据预处理，数值型和类别型特征都可以统一处理



XGBoost代码演示

<https://github.com/rfouyang/machine-learning.git>



小结

集成类别	集成现有模型	边学边集成
Uniform	投票	Bagging, Random Forest
Non-uniform	线性组合	AdaBoost, Gradient Boost
Conditional	stacking	Decision Tree



谢谢大家



助教微信: aischool1007