AugBoost:
Gradient Boosting Enhanced with Step-Wise Feature Augmentation

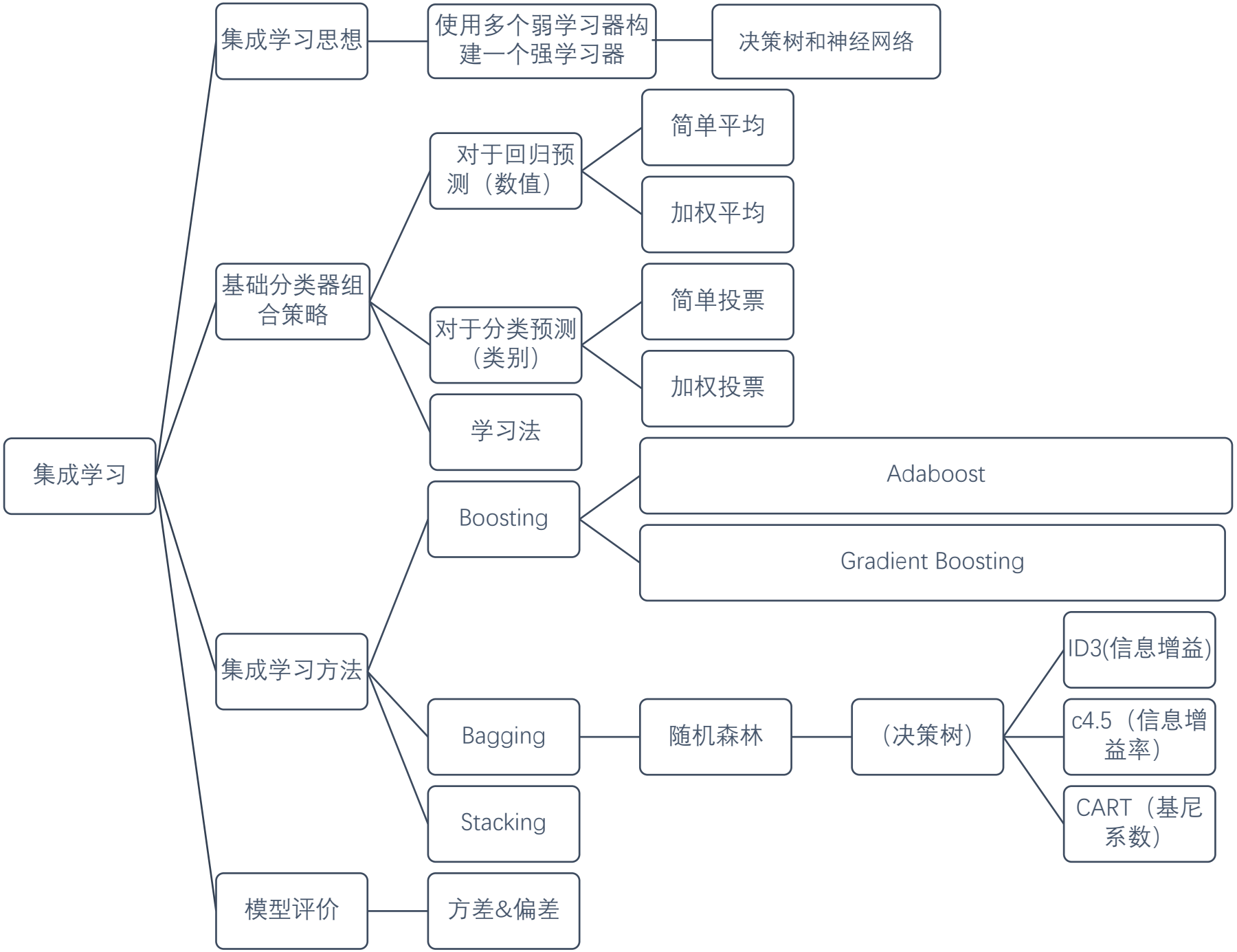
李轩 19212010022

Tannor P, Rokach L. Augboost: gradient boosting enhanced with step-wise feature augmentation[C].IJCAI. 2019: 3555-3561.

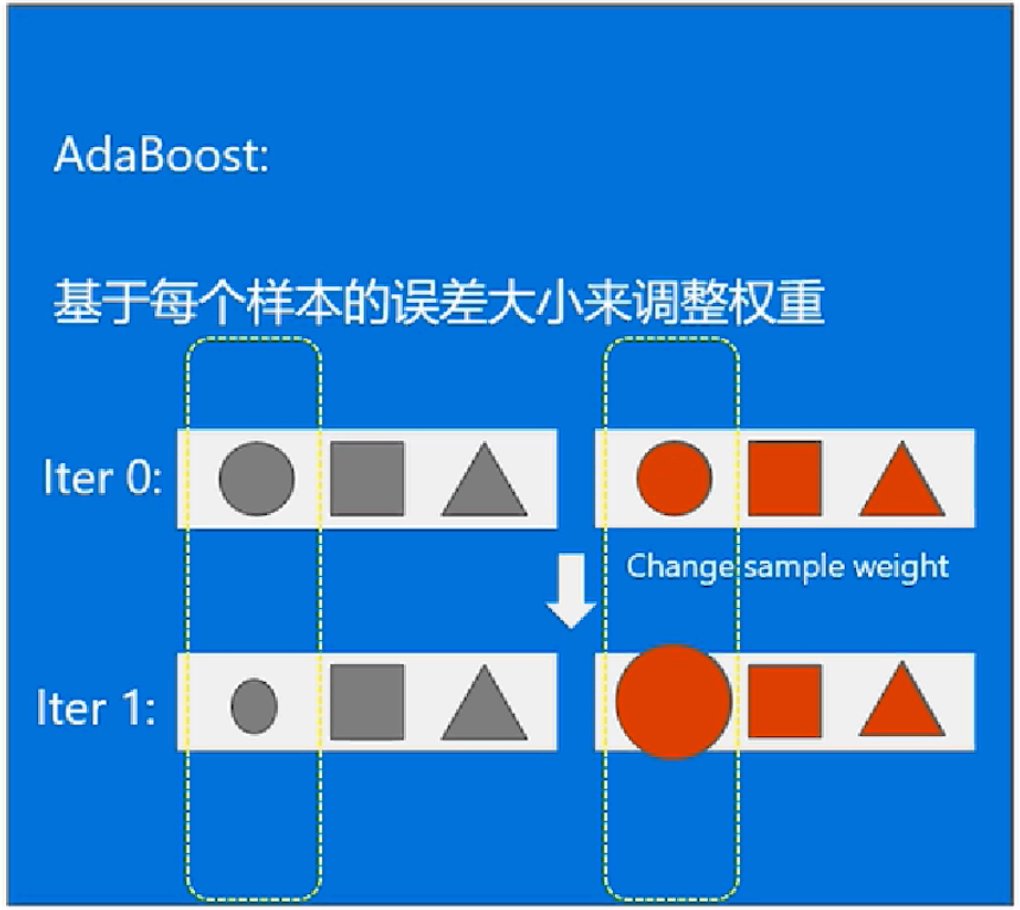
目录

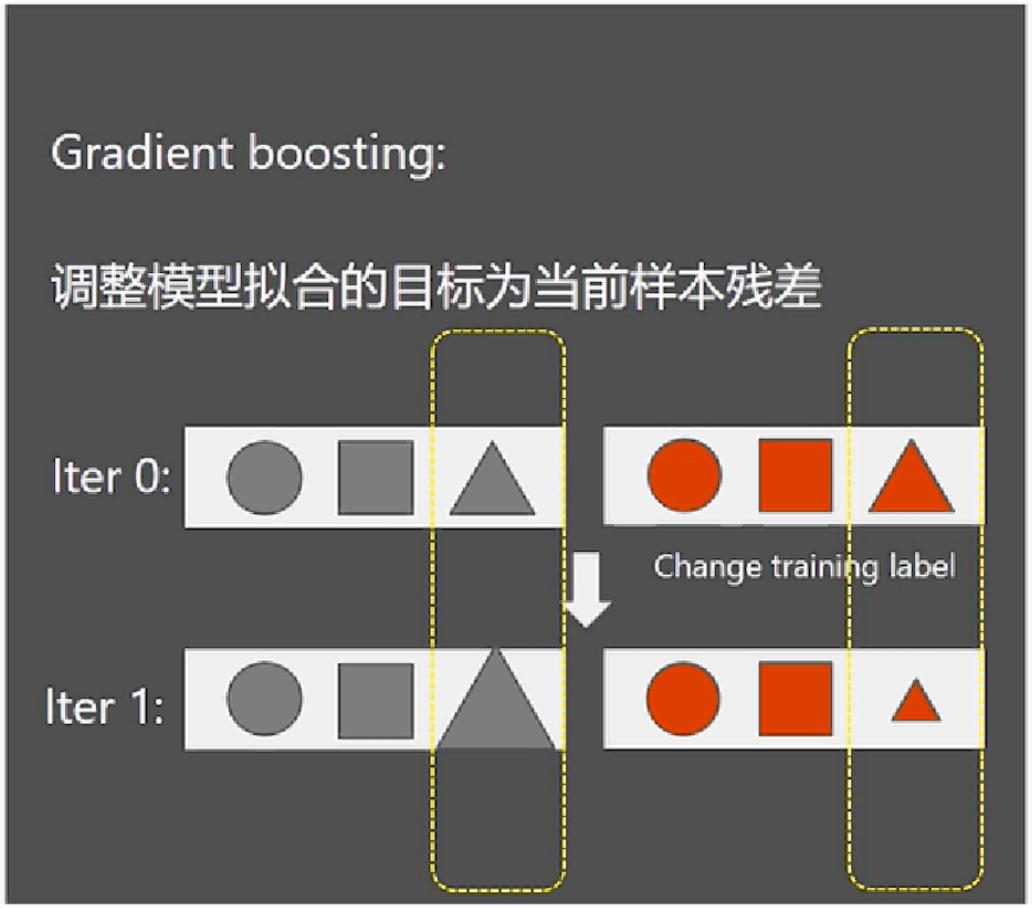
- 基模型GBDT(Gradient Boosting Decision Trees)
- 改进模型XGBoost 的方案与不足
- 基于GBDT改进的AugBoost的实现
  - 使用ANN做特征增强
  - 使用PCA/RP做特征增强
- AugBoost的实验结果
- 总结与改进

集成学习


集成学习
集成学习思想
使用多个弱学习器构建一个强学习器
决策树和神经网络
基础分类器组合策略
对于回归预测（数值）
简单平均
加权平均
对于分类预测（类别）
简单投票
加权投票
学习法
集成学习方法
Boosting
Adaboost
Gradient Boosting
Bagging
随机森林
（决策树）
ID3(信息增益)
c4.5（信息增益率）
CART（基尼系数）
Stacking
模型评价
方差&偏差

Boosting





:weight :feature :label

https://blog.cs

- Solution: **Additive Training (Boosting)**
  - Start from constant prediction, add a new function each time

$$
\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}
$$

New function

Model at training round t

Keep functions added in previous round

GBDT

- 1. 初始化，估计使损失函数极小化的常数值，它是只有一个根节点的树，即γ是一个常数值。
- 2.
  （a）计算损失函数的负梯度在当前模型的值，将它作为残差的估计
  （b）估计回归树叶节点区域，以拟合残差的近似值
  （c）利用线性搜索估计叶节点区域的值，使损失函数极小化
  （d）更新回归树
- 3. 得到输出的最终模型 f(x)

Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute

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

   (b) Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, $j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute

$$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$

   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

Gradient Boosting (Simplified)

Height	Weight	Male?	Likes Cake
1.02	35	True	False
1.82	90	True	False
1.67	50	False	True

Label
12
30
66

[Friedman, 2001]

GBDT

- 存在的问题：
  - Boosting算法所特有的性质，不断地降低bias，在应用中容易过拟合

- 现有的解决方法：
  - 调节CART的超参数，分裂的最大深度，叶子节点最少样本，最大叶子节点数等
  - Shrinkage 和 Column Subsampling
  - XGBoost和LightGBM的改进，优化目标的改变

XGBoost

- 改进
  优化目标

Taylor Expansion Approximation of Loss

- Goal $Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant$
  - Seems still complicated except for the case of square loss
- Take Taylor expansion of the objective
  - Recall $f(x+\Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2$
  - Define $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$

$$Obj^{(t)} \simeq \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant$$

- *If you are not comfortable with this, think of square loss*

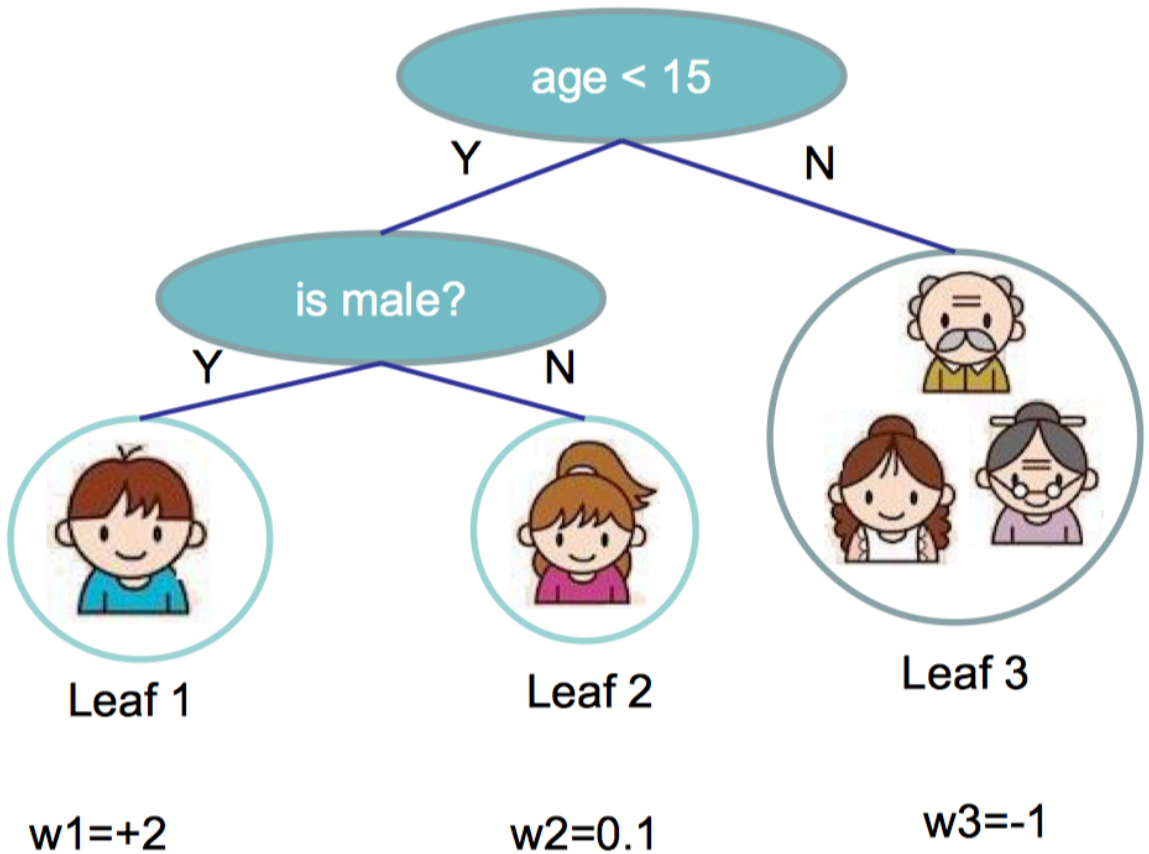
$$g_i = \partial_{\hat{y}^{(t-1)}} (\hat{y}^{(t-1)} - y_i)^2 = 2(\hat{y}^{(t-1)} - y_i) \quad h_i = \partial^2_{\hat{y}^{(t-1)}} (y_i - \hat{y}^{(t-1)})^2 = 2$$

Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C].Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016: 785-794.

- Define complexity as (this is not the only possible definition)

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

Number of leaves **L2 norm of leaf scores**



$$\Omega = \gamma 3 + \frac{1}{2}\lambda(4 + 0.01 + 1)$$

Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C].Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016: 785-794.

Define the instance set in leaf j as $I_j = \{i | q(x_i) = j\}$

- Regroup the objective by leaf

$$
\begin{aligned}
Obj^{(t)} &\simeq \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \\
&= \sum_{i=1}^{n} \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2 \right] + \gamma T + \lambda \frac{1}{2} \sum_{j=1}^{T} w_j^2 \\
&= \sum_{j=1}^{T} \left[(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2 \right] + \gamma T
\end{aligned}
$$

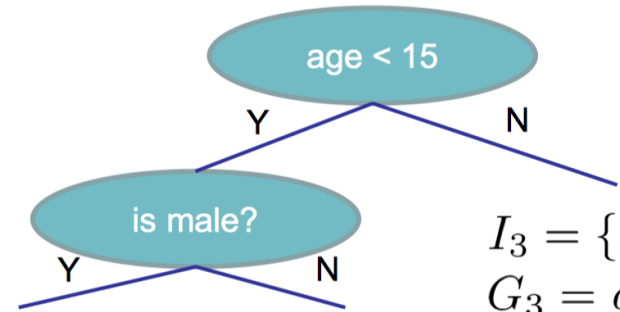
$$
w_j^* = -\frac{G_j}{H_j + \lambda} \qquad Obj = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T
$$

This measures how good a tree structure is!

Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C].Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016: 785-794.

The Structure Score Calculation

Instance index	gradient statistics
1	g1, h1
2	g2, h2
3	g3, h3
4	g4, h4
5	g5, h5



$$I_1 = \{1\} \qquad I_2 = \{4\} \qquad I_3 = \{2, 3, 5\}$$
$$G_1 = g_1 \qquad G_2 = g_4 \qquad G_3 = g_2 + g_3 + g_5$$
$$H_1 = h_1 \qquad H_4 = h_4 \qquad H_3 = h_2 + h_3 + h_5$$

$$Obj = -\sum_j \frac{G_j^2}{H_j+\lambda} + 3\gamma$$

The smaller the score is, the better the structure is

For each leaf node of the tree, try to add a split. The change of objective after adding the split is

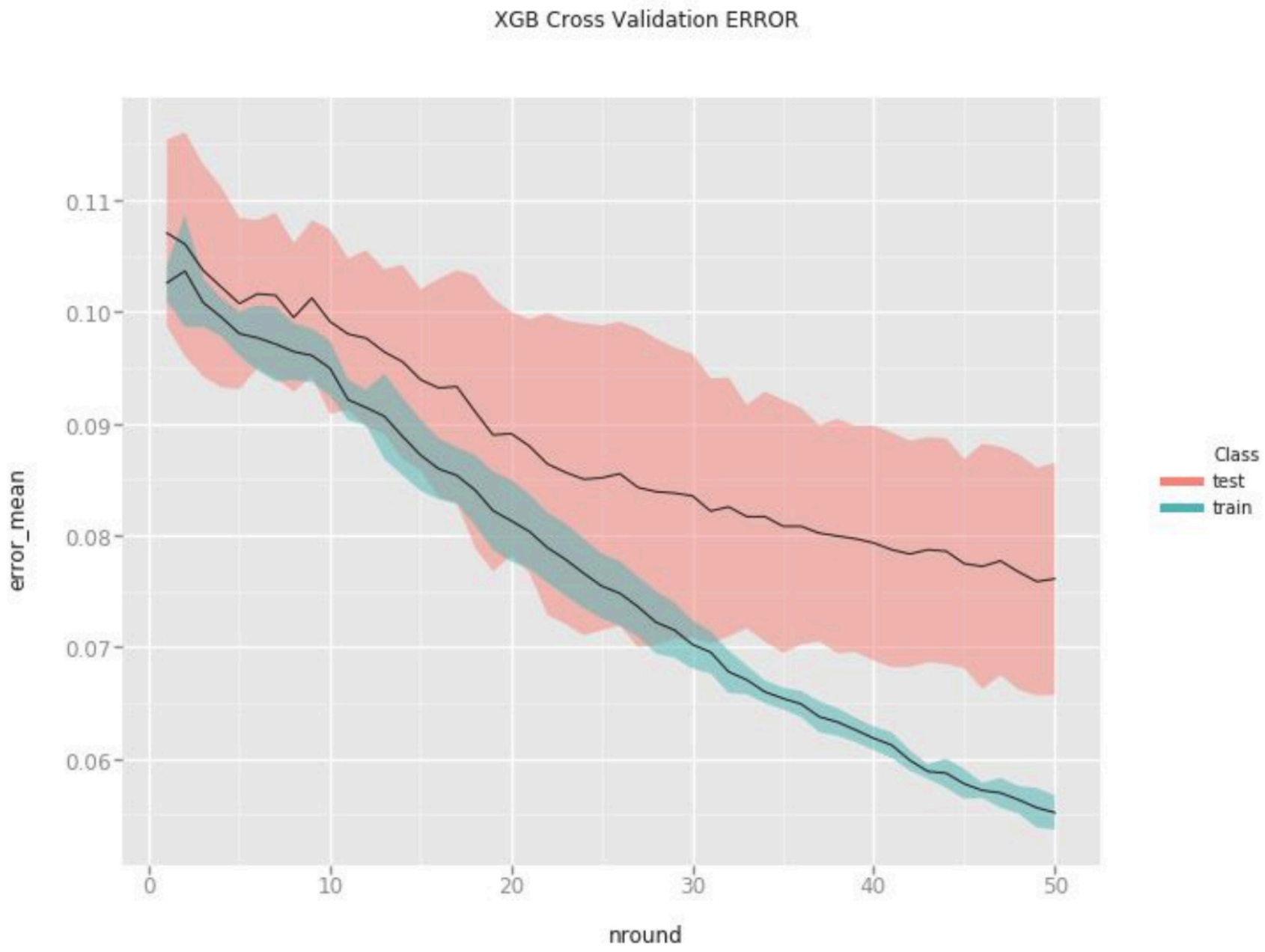
The complexity cost by introducing additional leaf

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$

the score of left child

the score of right child

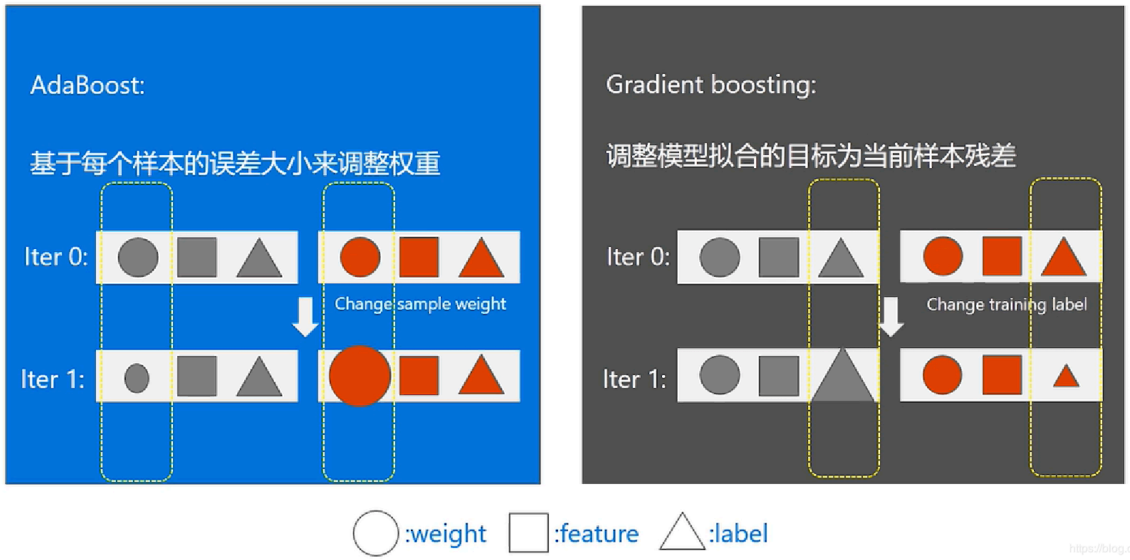
the score of if we do not split



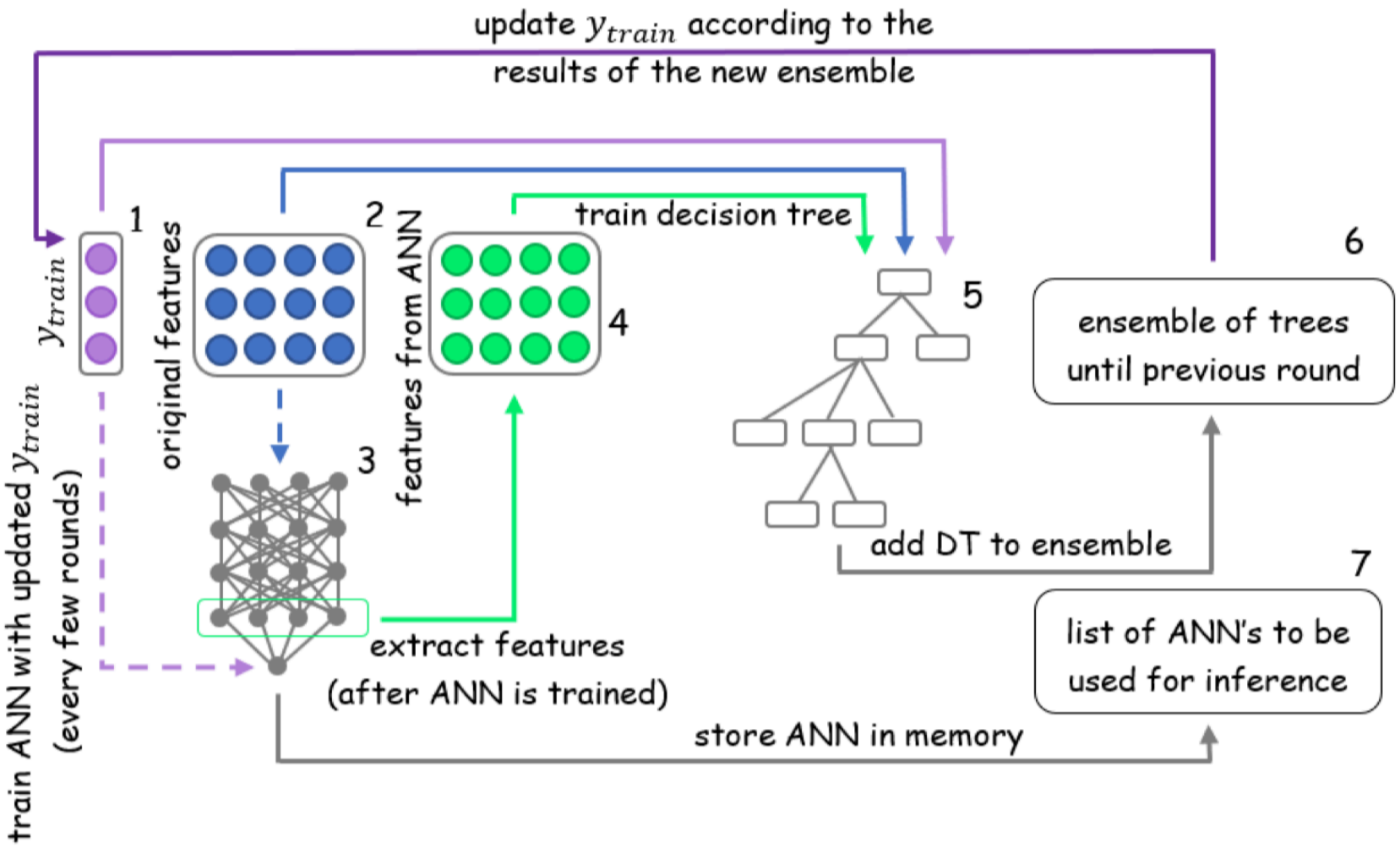
XGBoost虽然改进了GBDT的优化目标，但是还是比较依赖参数的选择

AugBoost

- XGBoost和LightGBM等算法都是从第i棵树的优化目标改进
- 主要贡献（提出了通用的机器学习特征增强方法）
  - 使用ANN做特征增强
  - 使用PCA做特征增强
  - 验证AugBoost的有效性，有效防止过拟合，并与LightGBM做精度对比



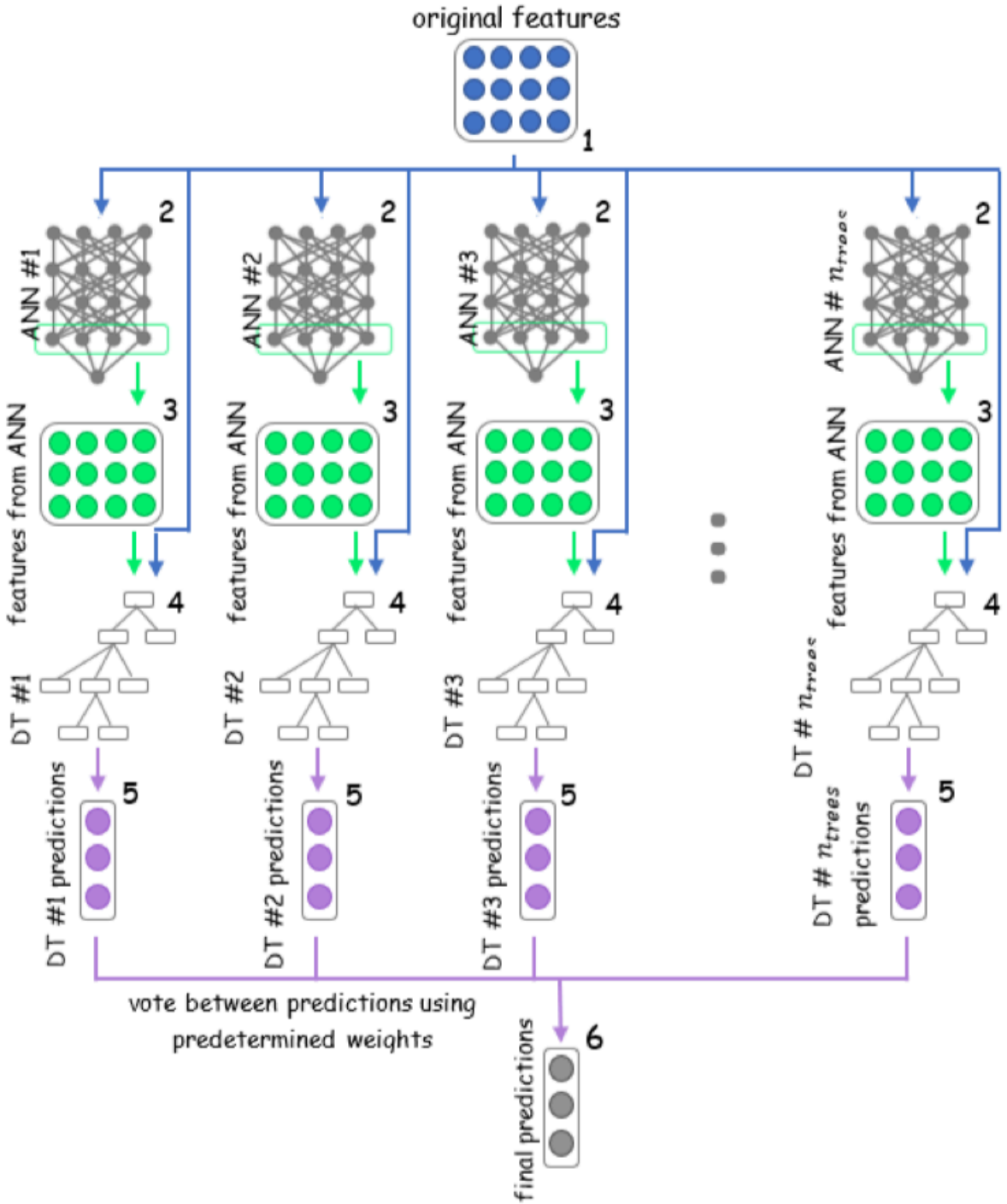
AugBoost-ANN



训练阶段：

首先训练ANN拟合y_train，然后将ANN保存ANN_list中在测试阶段使用，将ANN提取出的最后一层特征与原始特征拼接，送入GBDT。

训练技巧：利用迁移学习，首先训练好一个N层的抽取网络，然后固定前M层，fine-tunning后面层，抽取最后一层embedding。



- 测试阶段

  多个GBDT共同投票得到最后的预测结果

AugBoost-PCA

- PCA

设有m条n维数据。
1）将原始数据按列组成n行m列矩阵X
2）将X的每一行（一个属性字段）进行零均值化，即减去这一行的均值
3）求出协方差矩阵
4）求出协方差矩阵的特征值及对应的特征向量
5）将特征向量按对应特征值大小从上到下按行排列成矩阵，取前k行组成矩阵P
6）Y=PX即为降维到k维后的数据

PCA algorithm($\mathbf{X}$, k): top k eigenvalues/eigenvectors

% $\underline{\mathbf{X}}$ = N × m data matrix,
% … each data point $\mathbf{x}_i$ = column vector, i=1..m

- $\underline{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$
- $\mathbf{X} \leftarrow$ subtract mean $\underline{\mathbf{x}}$ from each column vector $\mathbf{x}_i$ in $\underline{\mathbf{X}}$
- $\Sigma \leftarrow \mathbf{X}\mathbf{X}^T$ … covariance matrix of $\mathbf{X}$
- $\{\lambda_i, \mathbf{u}_i\}_{i=1..N}$ = eigenvectors/eigenvalues of Σ
  … $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$
- Return $\{\lambda_i, \mathbf{u}_i\}_{i=1..k}$
  % top k principal components

Wold S, Esbensen K. Principal component analysis[J]. Chemometrics and intelligent laboratory systems, 1987, 2(1-3): 37-52

AugBoost-PCA

- PCA一般是利用最大投影方差原理来降维，但是如果在选择不丢失任何特征值时，就是相当于对特征进行重组。
- 举个例子：

x

$$\begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}$$

y

$$\begin{pmatrix} 3 & 2 & 4 & 6 & 1 \end{pmatrix}$$

$$C = \frac{1}{5}\begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} -1 & -2 \\ -1 & 0 \\ 0 & 0 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{6}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{6}{5} \end{pmatrix}$$

求出特征值和特征向量

$$\lambda_1 = 2, \lambda_2 = 2/5$$

$$c_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, c_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$PCP^{\mathsf{T}} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}\begin{pmatrix} 6/5 & 4/5 \\ 4/5 & 6/5 \end{pmatrix}\begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2/5 \end{pmatrix}$$

$$Y = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}\begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -3/\sqrt{2} & -1/\sqrt{2} & 0 & 3/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}$$

AugBoost-PCA

- 假设不做降维，

$$Y = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -3/\sqrt{2} & -1/\sqrt{2} & 0 & 3/\sqrt{2} & -1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 & -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

origin features X concatenated with enhanced features

$$\begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -3/\sqrt{2} & -1/\sqrt{2} & 0 & 3/\sqrt{2} & -1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 & -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

y

$$\begin{pmatrix} 3 & 2 & 4 & 6 & 1 \end{pmatrix}$$

AugBoost-PCA

- 按照新特征所包含的方差比例重组，抽取特征的按照重要程度排序，使得主要信息集中在前几维，在做列采样时包含这些重要的特征列，从而产生更加有鲁棒性的决策树，取得好的效果。

$$PCP^{\mathsf{T}} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 6/5 & 4/5 \\ 4/5 & 6/5 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2/5 \end{pmatrix}$$

origin features X concatenated with enhanced features

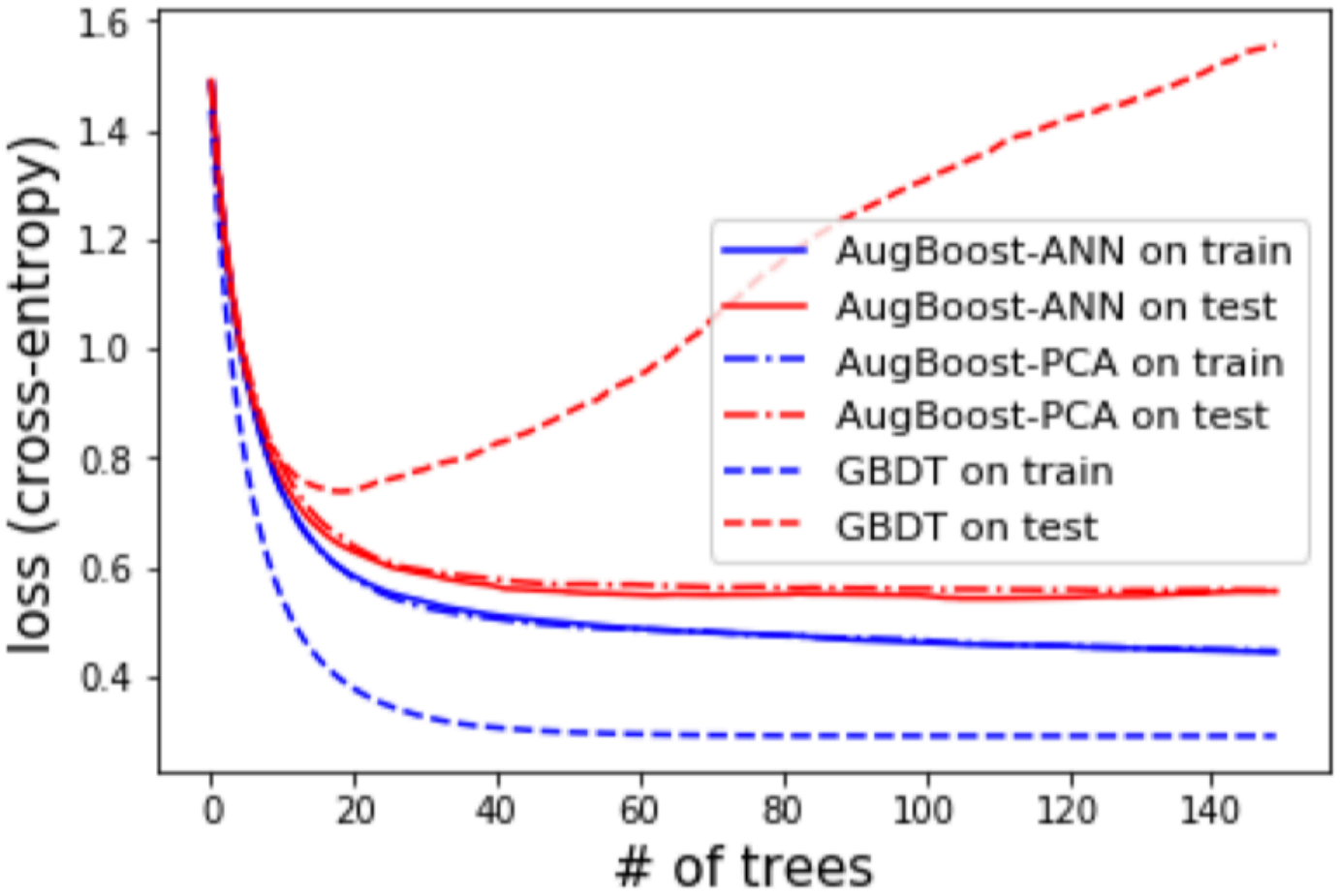
$$\begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \boxed{-3/\sqrt{2} \quad -1/\sqrt{2} \quad 0 \quad 3/\sqrt{2} \quad -1/\sqrt{2}} \\ -1/\sqrt{2} \quad 1/\sqrt{2} \quad 0 \quad -1/\sqrt{2} \quad 1/\sqrt{2} \end{pmatrix}$$

y

$$\begin{pmatrix} 3\,2\,4\,6\,1 \end{pmatrix}$$

实验结果



- 随着树数量的不断变多，GBDT过拟合的趋势越明显
- 使用AugBoost增强的方法，可以有效地防止过拟合

dataset	# samples	# features	# classes	loss (cross-entropy)				
				AugBoost-ANN	AugBoost-PCA	AugBoost-RP	RPBoost	LightGBM
cortex (mice protein)	1080	80	8	**1.04E-3 ± 2.37E-4**	5.79E-3 ± 8.57E-2	1.68E-2 ± 9.29E-3	9.71E-2±9.05E-3	1.75E-3 ± 4.34E-3
CRX (credit approval)	690	570	2	4.02E-1 ± 6.12E-2	**3.23E-1 ± 4.76E-2**	3.65E-1 ± 4.89E-2	3.73E-1±3.69E-2	4.83E-1 ± 1.45E-1
nursery	12960	8	5	**8.07E-2 ± 1.49E-2**	7.09E-2 ± 3.18E-2	2.21E-1 ± 1.56E-2	4.38E-1±1.17E-2	1.71E+0 ± 1.67E+0
heart disease	270	28	2	4.42E-1 ± 1.39E-1	5.55E-1 ± 2.42E-1	4.00E-1 ± 1.56E-1	**3.95E-1±1.31E-1**	5.54E-1 ± 2.18E-1
shuttle (statlog)	43500	9	7	8.06E-4 ± 6.80E-4	9.21E-4 ± 6.35E-4	**7.44E-4 ± 5.28E-4**	5.29E-2±3.42E-3	6.35E+0 ± 5.66E+0
wilt	4339	5	2	**2.55E-2 ± 1.56E-2**	2.73E-2 ± 1.61E-2	2.78E-2 ± 2.04E-2	3.25E-2±1.17E-2	3.40E-2 ± 2.41E-2
titanic	891	8	2	**4.34E-1 ± 4.36E-2**	4.56E-1 ± 5.18E-2	4.35E-1 ± 4.47E-2	4.45E-1±3.56E-2	6.50E-1 ± 7.55E-2
gender-by-voice	3168	20	2	6.29E-2 ± 2.32E-2	**6.06E-2 ± 2.55E-2**	6.26E-2 ± 2.37E-2	1.66E-1±2.30E-2	8.89E-2 ± 5.17E-2
adult	32560	14	2	2.97E-1 ± 5.39E-3	2.88E-1 ± 5.40E-3	3.06E-1 ± 7.21E-3	3.78E-1±5.70E-3	**2.77E-1 ± 4.74E-3**
frogs-MFCCs	7195	22	10	6.87E-2 ± 1.29E-2	**6.65E-2 ± 1.46E-2**	7.18E-2 ± 1.58E-2	1.37E-1±1.63E-2	6.25E-2 ± 2.17E-2
Wisconsin breast cancer	569	30	2	1.06E-1 ± 1.00E-1	1.23E-1 ± 1.10E-1	1.12E-1 ± 1.04E-1	**9.17E-2±4.83E-2**	1.37E-1± 1.65E-1
banknote authentication	1372	4	2	**1.32E-2 ± 1.69E-2**	1.37E-2 ± 1.43E-2	1.38E-2 ± 1.43E-2	5.69E-2±1.20E-2	1.31E-2 ± 2.76E-1
car evaluation	1727	6	4	1.41E+0 ± 1.45E-2	**1.31E+0 ± 3.13E-2**	1.38E+0 ± 1.55E-2	1.44E+0±1.71E-2	1.62E+0 ± 5.01E-2
CTG data (cardio)	2126	35	3	**4.08E-2 ± 2.28E-2**	5.20E-2 ± 2.99E-2	4.90E-2 ± 2.01E-2	9.37E-2±1.38E-2	7.86E-2 ± 4.66E-2
contraceptive method	1472	9	3	**9.25E-1 ± 3.13E-2**	9.38E-1 ± 3.28E-2	9.47E-1 ± 2.22E-2	9.55E-1±1.51E-2	1.08E+0 ± 5.84E-2
messidor (diabetes)	1151	19	2	**5.68E-1 ± 2.26E-2**	5.70E-1 ± 3.60E-2	5.82E-1 ± 1.54E-2	6.12E-1±2.36E-2	6.89E-1 ± 7.69E-2
parkinson	1040	27	2	2.98E-4 ± 5.00E-6	**2.95E-4 ± 4.89E-7**	2.98E-4 ± 8.00E-6	1.44E-1±1.80E-2	2.00E-6 ± 3.10E-9
pen digits	7494	16	10	4.39E-2 ± 1.05E-2	3.17E-2 ± 8.29E-3	5.32E-2 ± 5.68E-3	1.47E-1±1.39E-2	**3.15E-2 ± 1.11E-2**
phishing	1353	9	3	3.03E-1 ± 4.67E-2	**2.46E-1 ± 4.46E-2**	2.93E-1 ± 3.45E-2	3.55E-1±2.27E-2	3.08E-1 ± 9.51E-2
flare	1066	11	6	5.77E-1 ± 8.80E-2	6.67E-1 ± 1.12E-1	**5.61E-1 ± 7.57E-2**	5.83E-1±7.60E-2	7.03E-1 ± 1.48E-1
average ranking				2.25	2.5	2.65	3.9	3.7

数据来自 UCI, Kaggle datasets等开源数据集

AugBoost(ANN,PCA)几乎在所有数据集上取得远超LightGBM的效果。

总结

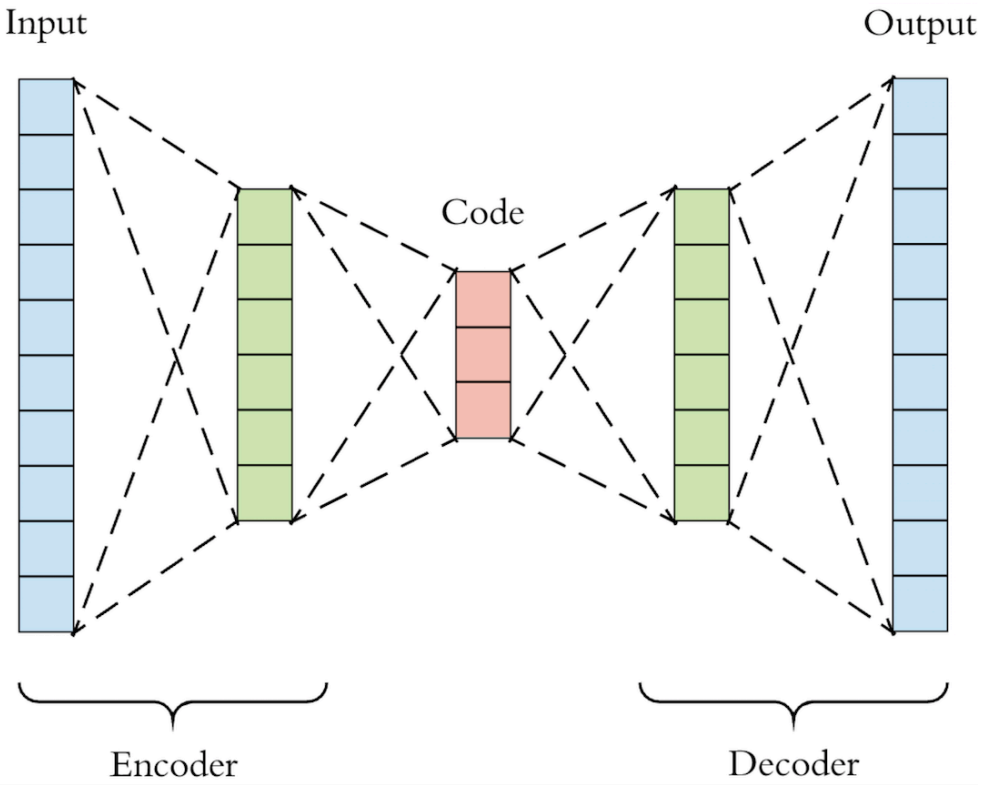
AugBoost能够使用特征增强的方法在机器学习任务中取得很好的分类和回归效果，从实验结果来看，可以认为基于ANN和PCA等特征增强方法能够更好地学习到数据的特征，从而能够更好地防止过拟合，有更好的泛化性能。

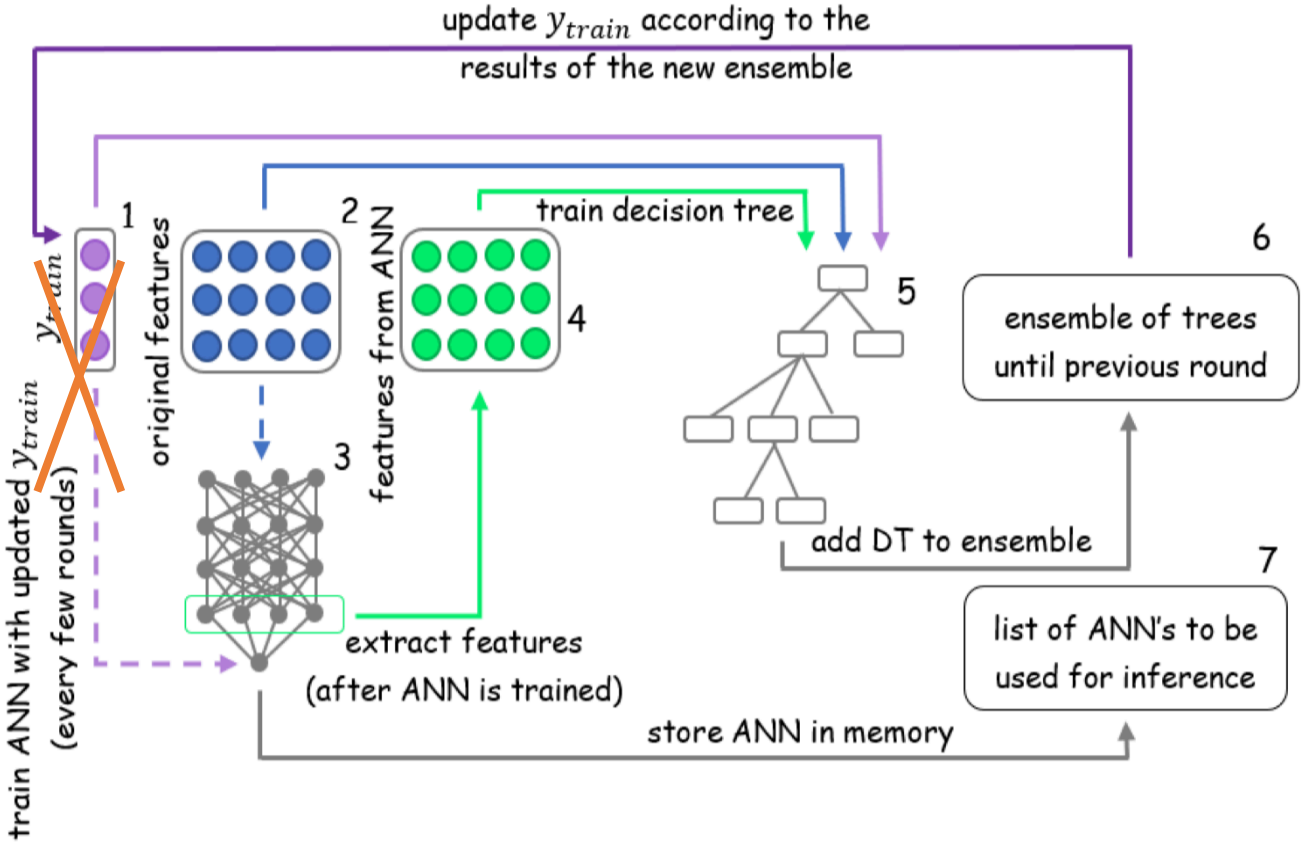
- 缺点：
  - 多种方法的特征融合是一种时间换精度的过程。

Future work

- 特征增强的改进：
  - 可以尝试 auto-encoder ANN, TSNE , LDA等方法进行特征的抽取

Auto-Encoder





- 基模型的改进
  - AugBoost是基于普通GBDT实现的，如果能使用XGBoost、LightGBM、CatBoost等思想对GBDT进行改进可能会提升集成效果，除了算法的改进（叶子节点和叶子节点数值的复杂度，Loss的泰勒二阶展开），还有工程实现上的改进，包括使用特征预排序或直方图算法快速寻找分裂点等并行计算的trick，可以提高速度和性能。

Thank You!