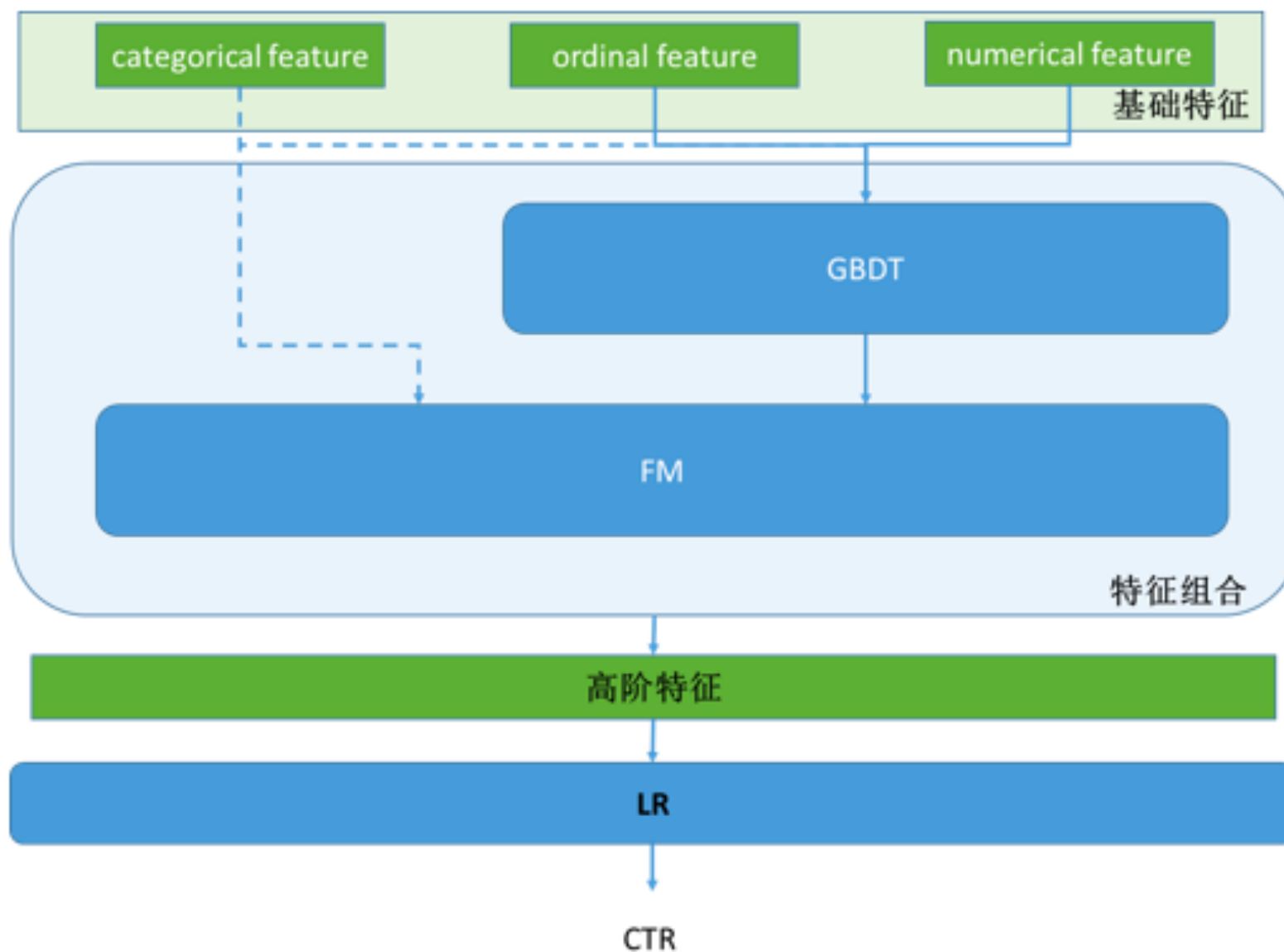


CTR预估-GBDT

CSDN学院

- CTR出现的背景
- Logistic回归 (LR)
- **GBDT**
- 因子分解机 (FM/FFM)
- 案例分析



► Facebook : GBDT+LR

- Practical Lessons from Predicting Clicks on Ads at Facebook [1]
- 用GBDT编码特征，然后再用LR做分类
 - GBDT可替代FM做特征编码
 - LR可用FTRL代替

[1] Xinran He et al. Practical Lessons from Predicting Clicks on Ads at Facebook, 2014.

- 用LR做CTR预估时，需做大量的特征工程 → 非线性特征
 - 连续特征离散化（+ One-Hot编码）
 - 特征进行二阶或者三阶的特征组合
- 问题：
 - 连续变量切分点如何选取？
 - 离散化为多少份合理？
 - 选择哪些特征交叉？
 - 多少阶交叉，二阶，三阶或更多？
- GBDT：一举解决了上面的问题
 - 确定切分点和切分数目不在是凭主观经验，而是根据信息增益/Gini指标
 - 每棵决策树从根节点到叶节点的路径，会经过不同的特征，此路径就是特征组合，而且包含了二阶，三阶甚至更多（所以GBDT提取特征时层数不用太深）

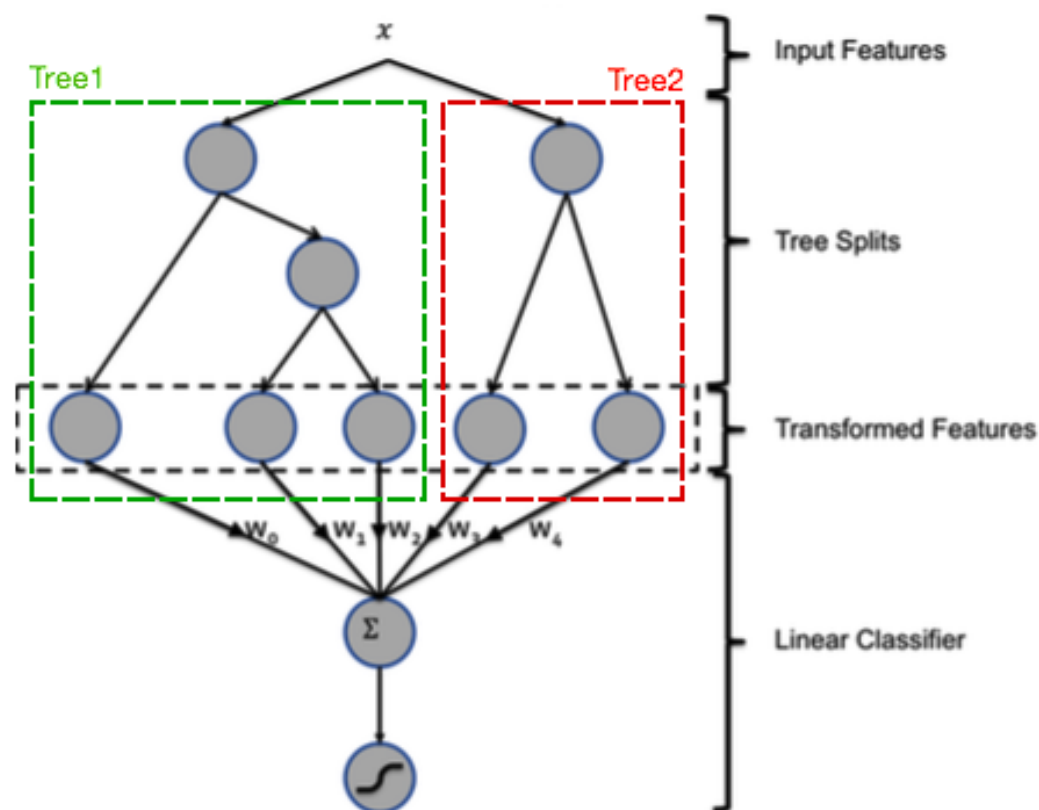
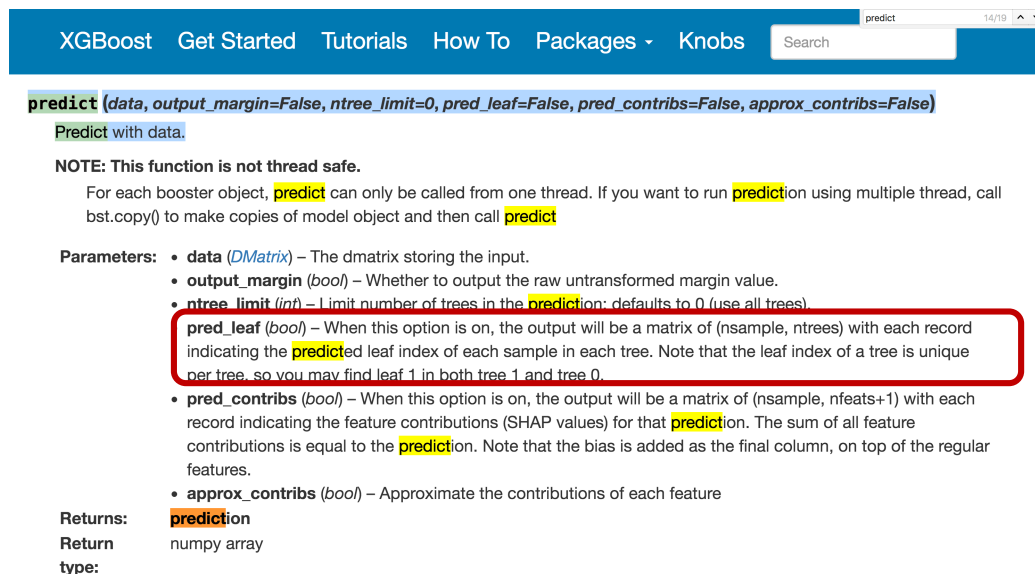


Figure 1: Hybrid model structure. Input features are transformed by means of boosted decision trees. The output of each individual tree is treated as a categorical input feature to a sparse linear classifier. Boosted decision trees prove to be very powerful feature transforms.

GBDT训练得到：
第一棵树有3个叶子结点
第二棵树有1个叶子结点

GBDT编码：对于一个输入样本点 x ，
如果它在第一棵树最后落在其中的第3个叶子结点，在第二棵树里最后落在第1个叶子结点，
则通过GBDT获得的新特征向量为
[0, 0, 1, 0, 1, 0]
向量中的前三位对应第一棵树的3个叶子结点
后两位对应第二棵树的1个叶子结点

- xgboost : predict函数
 - `predict(data, output_margin=False, ntree_limit=0, pred_leaf=False, pred_contribs=False, approx_contribs=False)`
- lightGBM: predict函数
 - `predict(data, output_margin=False, ntree_limit=0, pred_leaf=False, pred_contribs=False, approx_contribs=False)`



XGBoost Get Started Tutorials How To Packages - Knobs Search

`predict(data, output_margin=False, ntree_limit=0, pred_leaf=False, pred_contribs=False, approx_contribs=False)`

Predict with data.

NOTE: This function is not thread safe.

For each booster object, `predict` can only be called from one thread. If you want to run `prediction` using multiple thread, call `bst.copy()` to make copies of model object and then call `predict`

Parameters:

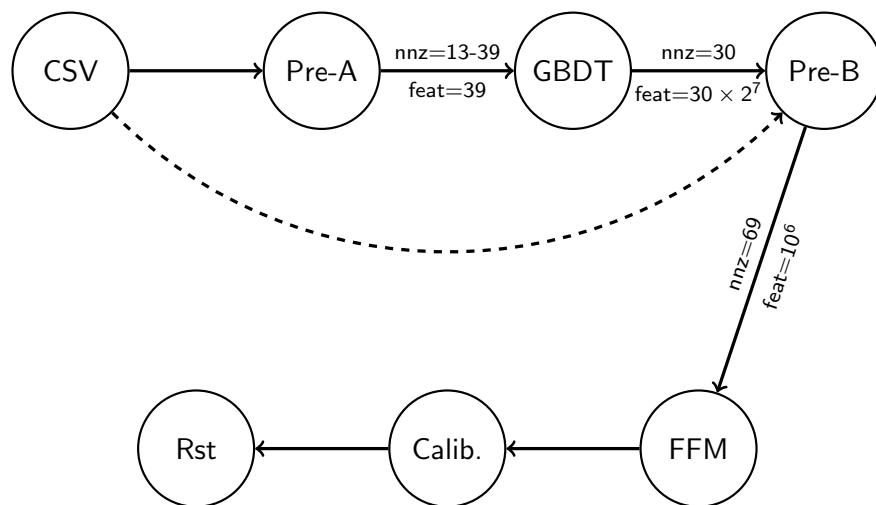
- `data (DMatrix)` – The dmatrix storing the input.
- `output_margin (bool)` – Whether to output the raw untransformed margin value.
- `ntree_limit (int)` – Limit number of trees in the `prediction`: defaults to 0 (use all trees).
- `pred_leaf (bool)` – When this option is on, the output will be a matrix of (nsample, ntrees) with each record indicating the `predicted` leaf index of each sample in each tree. Note that the leaf index of a tree is unique per tree, so you may find leaf 1 in both tree 1 and tree 0.
- `pred_contribs (bool)` – When this option is on, the output will be a matrix of (nsample, nfeats+1) with each record indicating the feature contributions (SHAP values) for that `prediction`. The sum of all feature contributions is equal to the `prediction`. Note that the bias is added as the final column, on top of the regular features.
- `approx_contribs (bool)` – Approximate the contributions of each feature

Returns: `prediction`

Return type: numpy array

```
xgb_leaves = xgb_test_basis_d6.predict(dtv, pred_leaf = True)
```

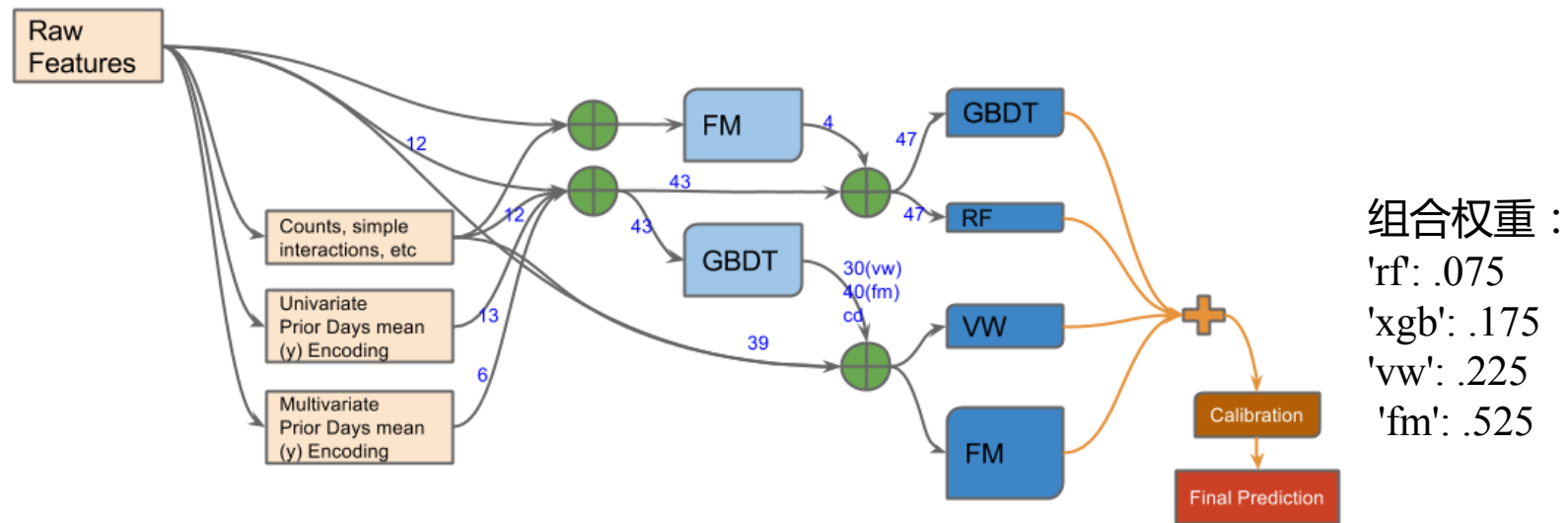
- Kaggle 2014年竞赛：Criteo Display Advertising Challenge
 - <https://www.kaggle.com/c/criteo-display-ad-challenge>
- Rank1解决方案：3 idiot's FM（FFM的发明者）



1. <https://github.com/guestwalk/kaggle2014criteo>

► GBDT+ FM & FM+GBDT

- Kaggle 2015年竞赛：Click-Through Rate Prediction
 - <https://www.kaggle.com/c/avazu-ctr-prediction>
- Rank2解决方案：



<https://github.com/owenzhang/kaggleavazu>

► 为什么不直接用GDBT？

- 因为GDBT在线预测比较困难，而且训练时间复杂度高于LR。
- 所以实际中，可以离线训练GDBT，然后将该模型作为在线ETL的一部分。

THANK YOU



AI100