

# 3 Idiots' Approach for Display Advertising Challenge

Yu-Chin Juan, Yong Zhuang, and Wei-Sheng Chin

NTU CSIE MLGroup

# What This Competition Challenges Us?

Predict the click probabilities of impressions.

# Dataset

Label	I1	I2	...	I13	C1	C2	...	C26
1	3	20	...	2741	68fd1e64	80e26c9b	...	4cf72387
0	7	91	...	1157	3516f6e6	cfc86806	...	796a1a2e
0	12	73	...	1844	05db9164	38a947a1	...	5d93f8ab
					⋮			
?	9	62	...	1457	68fd1e64	cfc86806	...	cf59444f

#Train:  $\approx 45\text{M}$

#Test:  $\approx 6\text{M}$

#Features after one-hot encoding:  $\approx 33\text{M}$

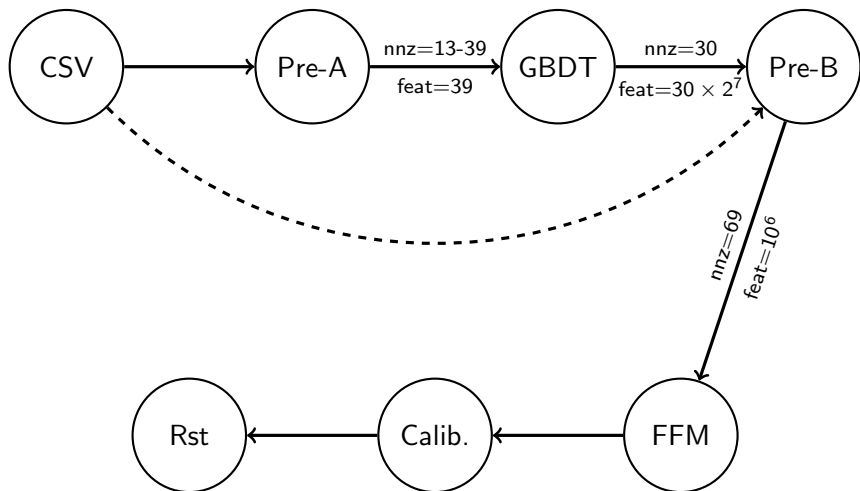
## Evaluation

$$\text{logloss} = -\frac{1}{L} \sum_{i=1}^L y_i \log \bar{y}_i + (1 - y_i) \log (1 - \bar{y}_i),$$

where  $L$  is the number of instances,  $y_i$  is the true label (0 or 1), and  $\bar{y}_i$  is the predicted probability.

This slide introduces our approach to achieve 0.44488 and 0.44479 on the public and private leaderboards, respectively.

## Flowchart



*"nnz" means the number of non-zero elements of each impression; "feat" represents the size of feature space.*

# Preprocessing-A

*Purpose: generate features for GBDT.*

- All numerical data are included. (13 features)
- Categorical features (after one-hot encoding) appear more than 4 million times are also included. (26 features)

# Gradient Boosting Decision Tree (GBDT)

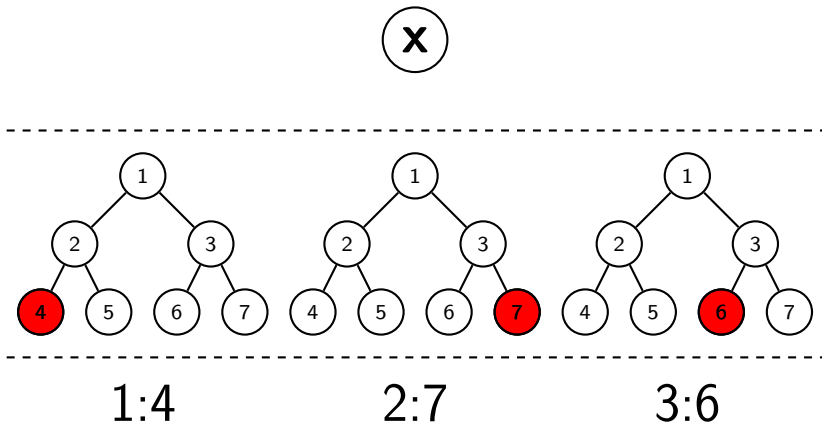
*Purpose: generate GBDT features.*

- We use trees in GBDT to generate features.
- 30 trees with depth 7 are used.
- 30 features are generated for each impression.
- This approach is proposed by [Xinran He et al.](#) at Facebook.



# Gradient Boosting Decision Tree (GBDT)

*Example: Assuming that we have already trained GBDT with 3 trees with depth 2. We feed an impression  $x$  into these trees. The first tree thinks  $x$  belong to node 4, the second node 7, and the third node 6. Then we generate the feature "1:4 2:7 3:6" for this impression.*



## Preprocessing-B

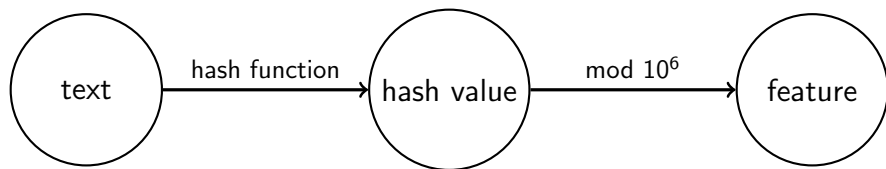
*Purpose: generate features for FFM.*

- **Numerical** features (I1-I13) greater than 2 are transformed by

$$v \leftarrow \lfloor \log(v)^2 \rfloor.$$

- **Categorical** features (C1-C26) appear less than 10 times are transformed into a special value.
- **GBDT** features are directly included.
- These three groups of features are hashed into 1M-dimension by hashing trick.
- Each impression has 13 (numerical) + 26 (categorical) + 30 (GBDT) = 69 features.

# Hashing Trick



l1:3

739920192382357839297

839297

C1-68fd1e64

839193251324345167129

167129

GBDT1:173

923490878437598392813

392813

# Field-aware Factorization Machine (FFM)

For the details of FFM, please check the following slides:

<http://www.csie.ntu.edu.tw/~r01922136/slides/ffm.pdf>

# Calibration

*Purpose: calibrate the final result.*

- The average CTRs on the public / private leaderboards are 0.2632 and 0.2627, respectively.
- The average CTR of our submission is 0.2663.
- There is a gap. So we minus every prediction by 0.003, and the logloss is reduced by around 0.0001.

# Running Time

Environment: A workstation with two 6-core CPUs  
All processes are parallelized.

Process	Time (min.)	Memory (GB)
Pre-A	8	0
GBDT	29	15
Pre-B	38	0
FFM	100	16
Calibration	1	0
Total	176	

## Comparison Among Different Methods

Method	Public	Private
LR-Poly2	0.44984	0.44954
FFM	0.44613	0.44598
FFM + GBDT	0.44497	0.44483
FFM + GBDT (v2)	0.44474	0.44462
FFM + GBDT + calib.	0.44488	0.44479
FFM + GBDT + calib. (v2)	0.44461	0.44449

v2: 50 trees and 8 latent factors