

GB-CENT

Gradient Boosted Categorical Embedding and Numerical Trees

Predictive Analytics Innovation Summit
February 22, 2017

Liangjie Hong
Head of Data Science, Etsy Inc.

Liangjie Hong

- **Head of Data Science at Etsy Inc.** since Aug. 2016.
- **Senior Manager of Research at Yahoo Research** in Sunnyvale, CA
Leading science efforts for personalization and search sciences
- Published papers in **SIGIR, WWW, KDD, CIKM, AAAI, WSDM, RecSys** and **ICML** (1800+ citations)
- **WWW 2011 Best Poster Paper Award**
WSDM 2013 Best Paper Nominated
RecSys 2014 Best Paper Award
- Program committee members in **KDD, WWW, SIGIR, WSDM, AAAI, EMNLP, ICWSM, ACL, CIKM, IJCAI** and various journal reviewers
- PhD in Machine Learning from Lehigh University

About This Paper

- Authors

Qian Zhao, PhD Student from **University of Minnesota**
Yue Shi, Research Scientist at **Facebook**, formerly at **Yahoo Research**
Liangjie Hong, Head of Data Science at **Etsy Inc.**, formerly at **Yahoo Research**
- Paper Venue

Full Research Paper in The 26th International World Wide Web Conference, 2017 (**WWW 2017**)

Why we need GB-CENT

Why we need GB-CENT

Two Families of Powerful Practical Data Mining and Machine Learning Tools

- **Tree-based Models**

Decision Trees, Random Forest, Gradient Boosted Decision Trees...

- **Matrix-based Embedding Models**

Matrix Factorization, Factorization Machines...

Why we need GB-CENT: Tree-based Models

- **Pros:**

Interpretability

Effectiveness in certain tasks: IR ranking models

Simple and easy to train

Handle numerical features well

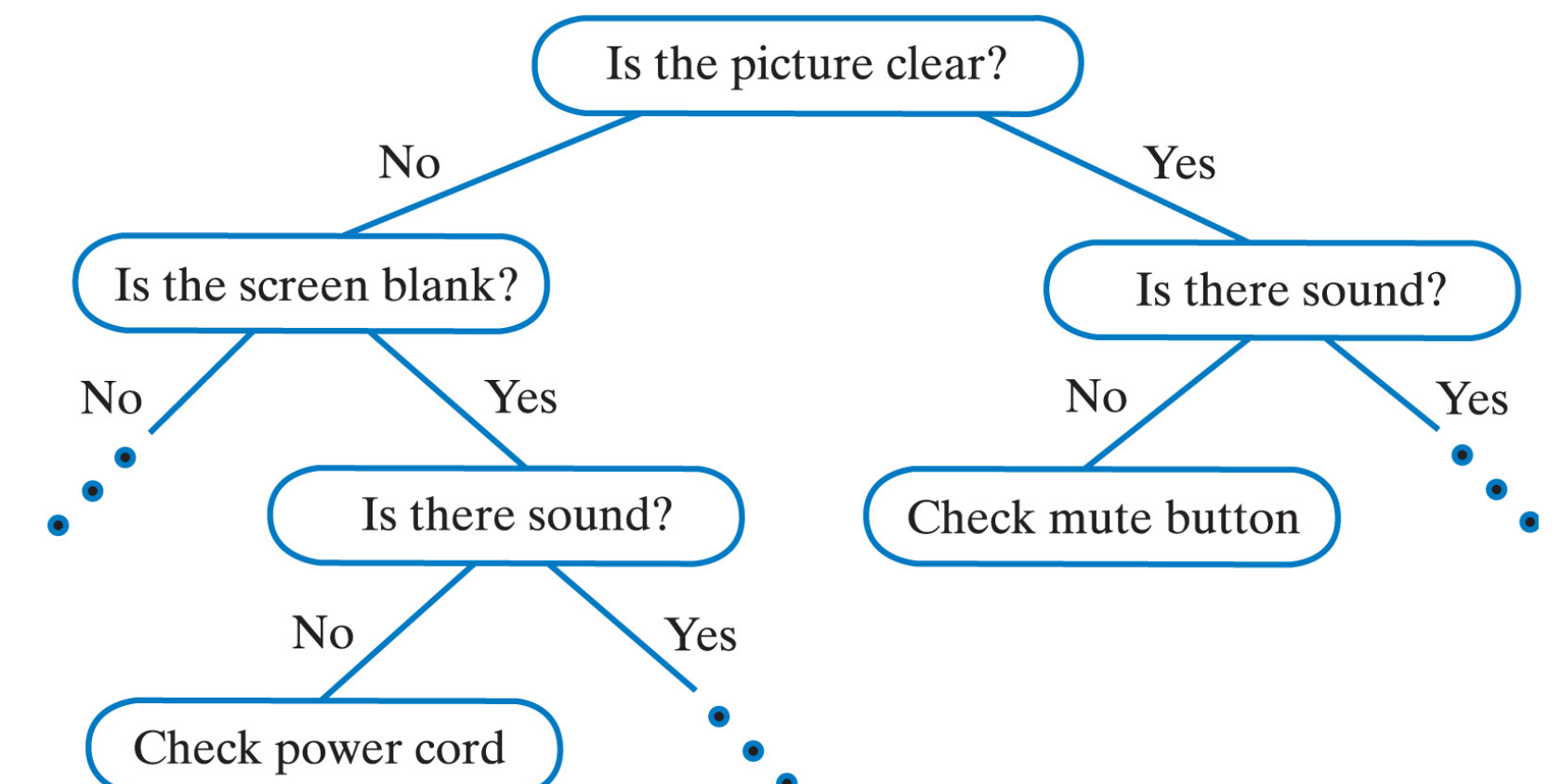
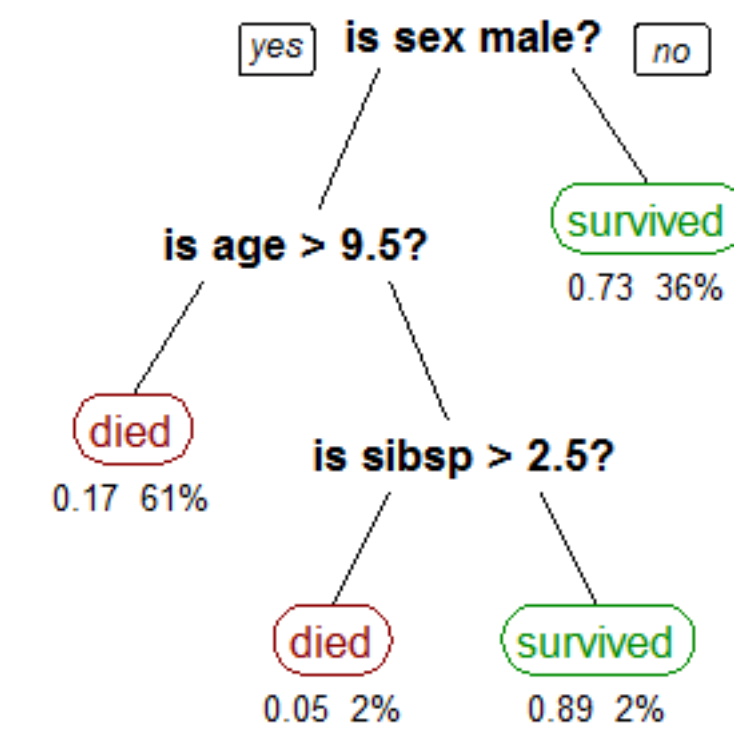
...

- **Cons:**

Cannot easily handle categorical features with large cardinality

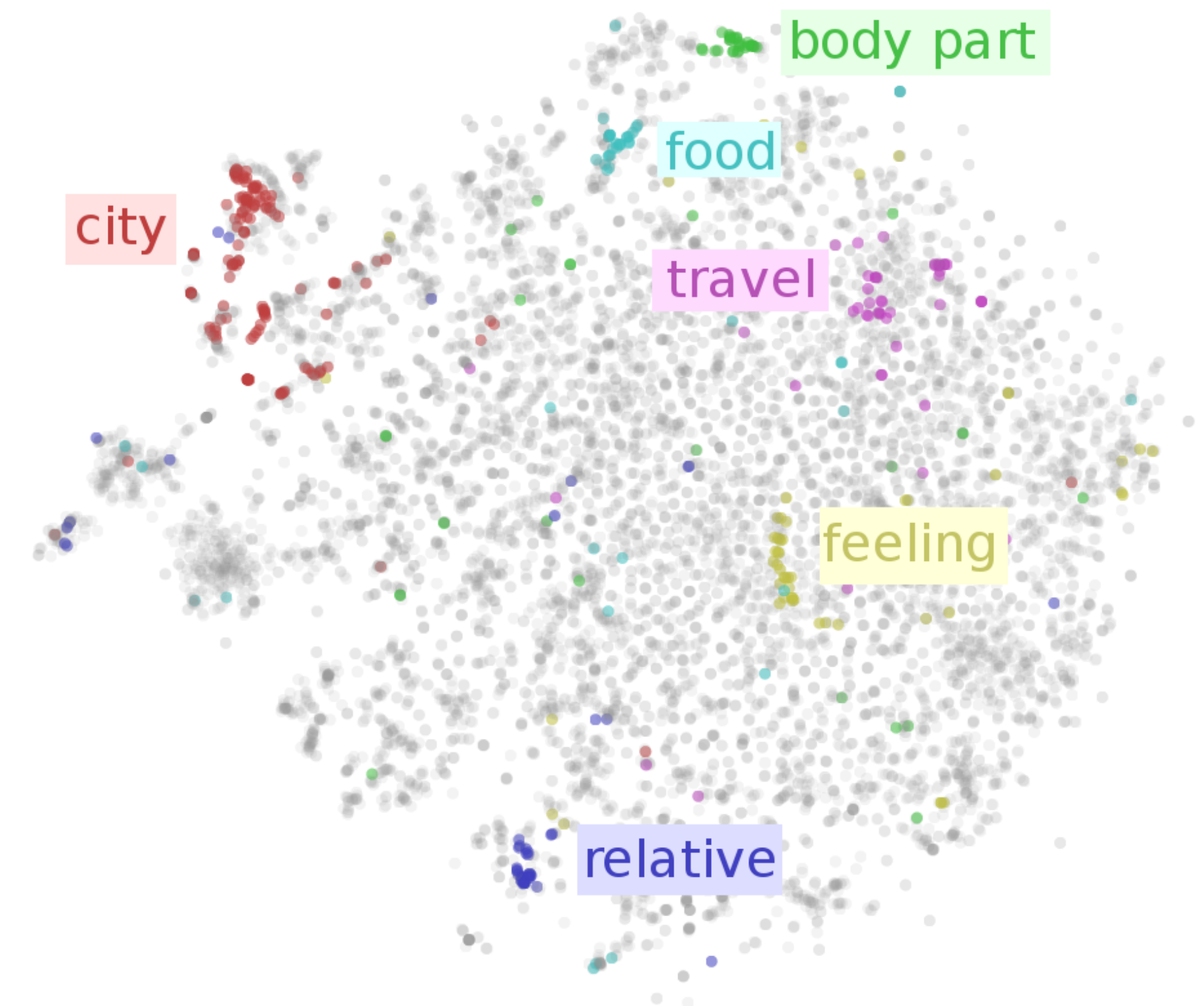
Hard to interpret complex trees

...



Why we need GB-CENT: Embedding-based Models

- **Pros:**
 - Predictive power
 - Effectiveness in certain tasks: recommender systems
 - Handle categorical features well
 - ...
- **Cons:**
 - Cannot easily handle numerical features
 - Hard to interpret in general
 - ...



Why we need GB-CENT

In practice,

- **We have both numerical features and categorical features.**
- **We need to utilize both models.**

What is GB-CENT

What is GB-CENT

In a nutshell, GB-CENT is to combine

- **Tree-based Models**
Handle numerical features...
- **Matrix-based Embedding Models**
Handle large-cardinality categorical features...

What is GB-CENT

$$y(\hat{x}) = \underbrace{\sum_{i=0}^k w_{a_i}}_{bias} + \underbrace{\left(\sum_{a_i \in U(a)} Q_{a_i} \right)^T \left(\sum_{a_i \in I(a)} Q_{a_i} \right)}_{factor} + \underbrace{\sum_{i=0}^k T_{a_i}(b)}_{CAT-NT}$$

$CAT-E$

A prediction is based on:

- **Bias terms from each categorical feature**
- **Dot-product of embedding features of two categorical features**
e.g., user-side v.s. item-side
- **Per-categorical decision trees based on numerical features**
ensemble of numerical decision trees where each tree is based on one categorical feature

What is GB-CENT

$$y(\hat{x}) = \underbrace{\sum_{i=0}^k w_{a_i}}_{bias} + \underbrace{\left(\sum_{a_i \in U(a)} Q_{a_i} \right)^T \left(\sum_{a_i \in I(a)} Q_{a_i} \right)}_{factor} + \underbrace{\sum_{i=0}^k T_{a_i}(b)}_{CAT-NT}$$

$CAT-E$

Two Ingredients:

- **Factorization Machines with Numerical Features**
- **GBDT without Categorical Features**

What is GB-CENT

$$y(\hat{x}) = \underbrace{\sum_{i=0}^k w_{a_i}}_{bias} + \underbrace{\left(\sum_{a_i \in U(a)} Q_{a_i} \right)^T \left(\sum_{a_i \in I(a)} Q_{a_i} \right)}_{factor} + \underbrace{\sum_{i=0}^k T_{a_i}(b)}_{CAT-NT}$$

$CAT-E$

Different from GBDT:

- The number of trees in GB-CENT depends on the cardinality of categorical features in the data set, while GBDT has a pre-specified number of trees M .
- Each tree in GB-CENT only takes numerical features as input while GBDT takes in both categorical and numerical features.
- Learning a tree for GBDT uses all N instances in the data set while the tree for a categorical feature in GB-CENT only involves its supporting instances.

What is GB-CENT

$$y(\hat{x}) = \underbrace{\sum_{i=0}^k w_{a_i}}_{bias} + \underbrace{\left(\sum_{a_i \in U(a)} Q_{a_i} \right)^T \left(\sum_{a_i \in I(a)} Q_{a_i} \right)}_{factor} + \underbrace{\sum_{i=0}^k T_{a_i}(b)}_{CAT-NT}$$

$CAT-E$

Training GB-CENT:

- **Train embedding part firstly**
- **Train GBDT part secondly**
 - Sort categorical features by their support and fit trees by that order
 - Use a validation set to see whether to stop

How does GB-CENT perform

How does GB-CENT perform

- **Datasets**

MovieLens: 240K users, 33K movies, 22M instances, 5 ratings

RedHat: 151K customers, 7 categories, 2M instances, binary response

- **Baselines**

GB-CENT variants: CAT-E, CAT-NT, GB-CENT

GBDT variants: GBDT-OH, GBDT-CE

FM variants: FM-S, FM-D

SVDFeature variants: SVDFeature-S, SVDFeature-D

- **Metrics**

AUC, Accuracy, Time (Empirically)

How does GB-CENT perform

Data Set	Metric	GBDT-OH	GBDT-CE	SVDFeature-S	SVDFeature-D	FM-S	FM-D	CAT-E	CAT-NT	GB-CENT
MovieLens	RMSE	0.883 (0.007) -%1.8	0.863 (0.006) +%0.4	0.877 (0.009) -%1.1	0.867 (0.006) +%0.0	0.913 (0.024) -%5.3	0.888 (0.005) -%2.4	0.886 (0.011) -%2.1	0.900 (0.006) -%3.8	0.867 (0.006)
	Time (s)	282 +1.08	1034 +6.65	68 -%49.6	66 -%51.1	73 -%45.9	60 -55.5	77 -%42.9	54 -%60.0	135
RedHat	AUC	0.955 (0.0005) -%3.6	0.981 (0.0003) -%1.0	0.975 (0.0002) -%1.6	0.976 (0.0003) -%1.5	0.986 (0.0009) -%0.5	0.987 (0.0003) -%0.4	0.967 (0.0002) -%2.4	0.942 (0.0006) -%4.9	0.991 (0.00006)
	Time (s)	857 +%35.8	3140 +3.97	130 -%79.3	241 -%61.8	204 -%67.6	181 -%71.3	561 -%11.0	98 -%84.4	631

How does GB-CENT perform

Table 3: The effect of minTreeSupport and maxTreeDepth on MovieLens data set. minTreeSupport is held to be 50 when varying maxTreeDepth; maxTreeDepth is held to be 3 when varying minTreeSupport.

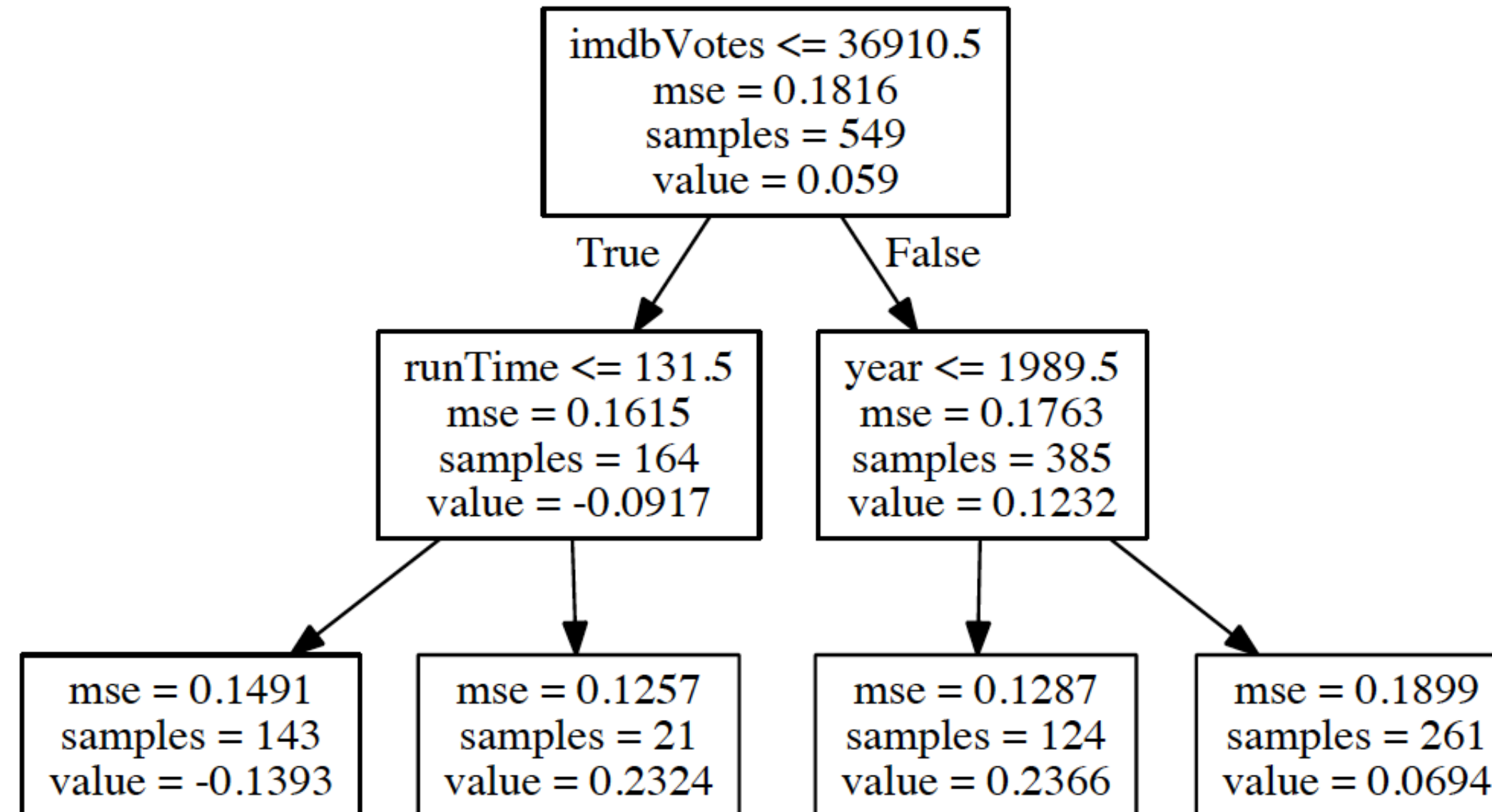
minTree-Support	RMSE	maxTree-Depth	RMSE
10	0.902	2	0.901
50	0.906	3	0.906
100	0.917	5	0.918
200	0.925	8	0.924
300	0.936	10	0.929
400	0.943	15	0.950

Table 4: The effect of tree regularization on MovieLens data set. minTreeSupport=50, maxTreeDepth=3.

Regulariza-tion	minTree-Gain	Number of Accepted Trees	RMSE
AAT	N.A.	7926	0.905
VSLR	0	7606	0.906
	1	7559	0.913
	3	7441	0.921
	5	6737	0.928
	8	6375	0.945

Main takeaway: Learn many shallow small trees

How does GB-CENT perform



Summary

GB-CENT

- Combine Factorization Machines and GBDT together
- Combine interpretable results and high predictive power
- Achieve high performance in real-world datasets

Questions