Alibaba.com

Job Recommendation with Hawkes Process

W. Xiao, X. Xu, K. Liang, J. Mao, and J. Wang

OneSearch Team, Alibaba Group

Boston, MA, USA

RECSYS / BOSTON / 2016

Outline

- **Introduction**
- **Problem Statement and Setting**
- **Our Solution**
  - Overall Framework
  - Feature Engineering
  - Learning to Rank with GBDT
  - Remedy for Cold Start
  - Temporal Descriptor with Hawkes Process
- **Results and Discussions**

Introduction

Introduction – OneSearch from Alibaba

OneSearch Team

- YunOS BU & Search BU of Alibaba Group
- OneSearch Project - Intelligent Data Platform for IOT

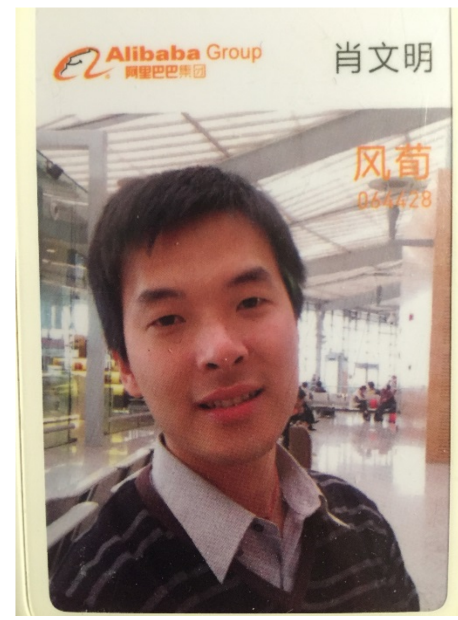
Introduction – OneSearch from Alibaba

Team for Challenges

- RecSys2015(Vienna)
  - The Second Place
- KDD Cup 2016 (San Francisco)
  - The Second Place (Phase 1)

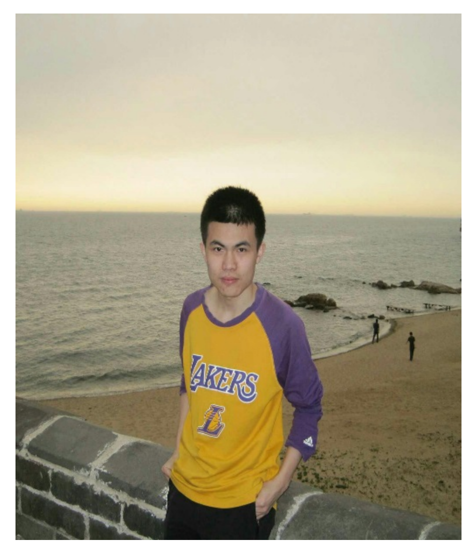
Boston

- RecSys2016
  - **The First Place**



Wenming Xiao

Xiao Xu



Kang Liang

Junkang Mao

Jun Wang

Problem Statement and Setting

Problem Statement

Task

- Given the *profile information of the users*, *the content of job posting*, and *the historical log of users activities*, the key task is to recommend a list of job posts, which the users might interact with in the next week

Evaluation Metric

$$score(R,T) = \sum_{i=1}^{N} s(u_i)$$

$$s(u_i) = 20\,(P_2 + P_4 + R + UserSuccess) + 10(P_6 + P_{20})$$

General Thinking

Item imp/int trend

Active during test

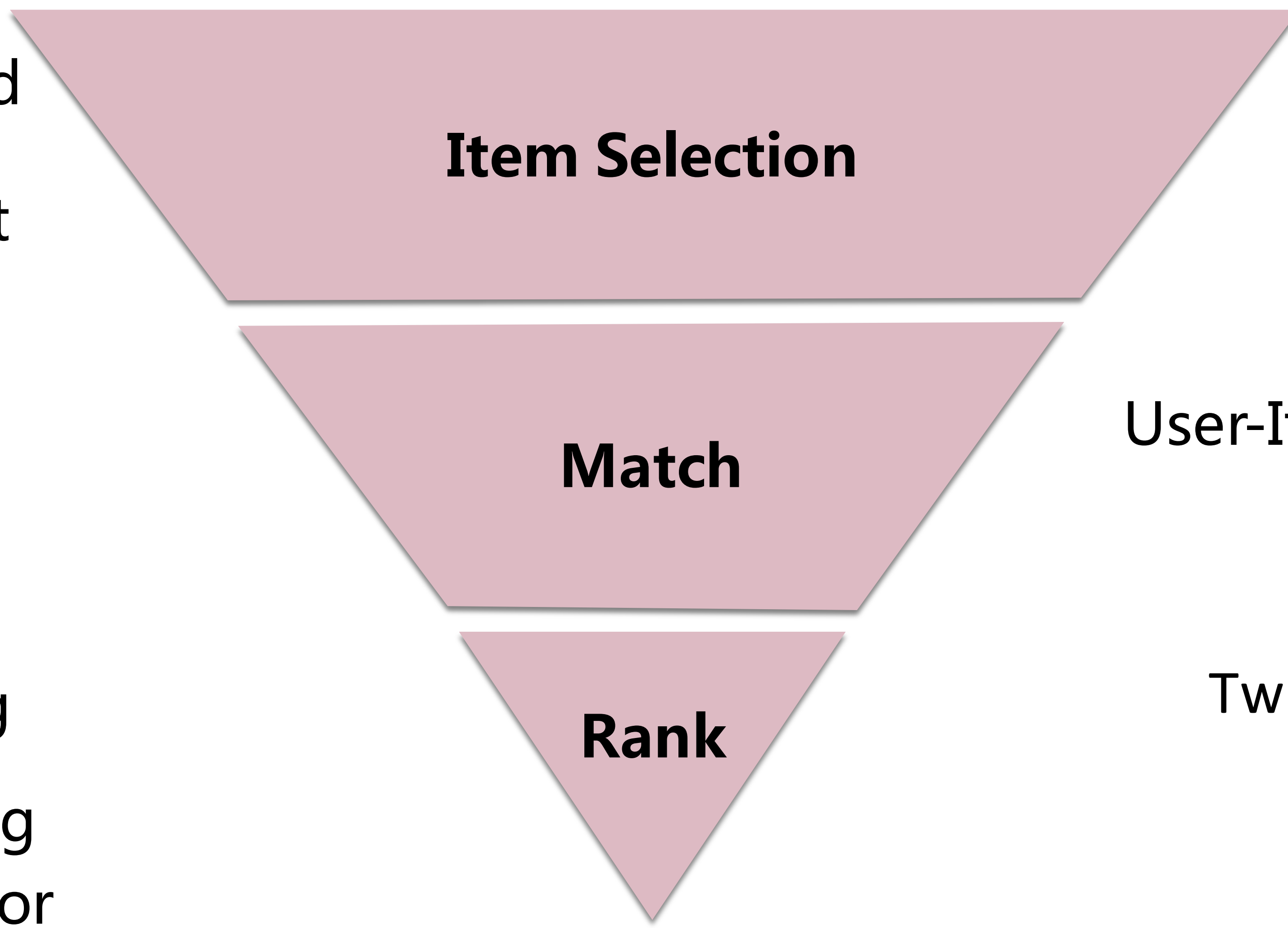
Item based CF

User based CF

Pairwise Learning

Ensemble Learning

Temporal Descriptor



Item profiling

User profiling

User-Item Profile Similarity

……

Two Stage Modeling

Our Solution

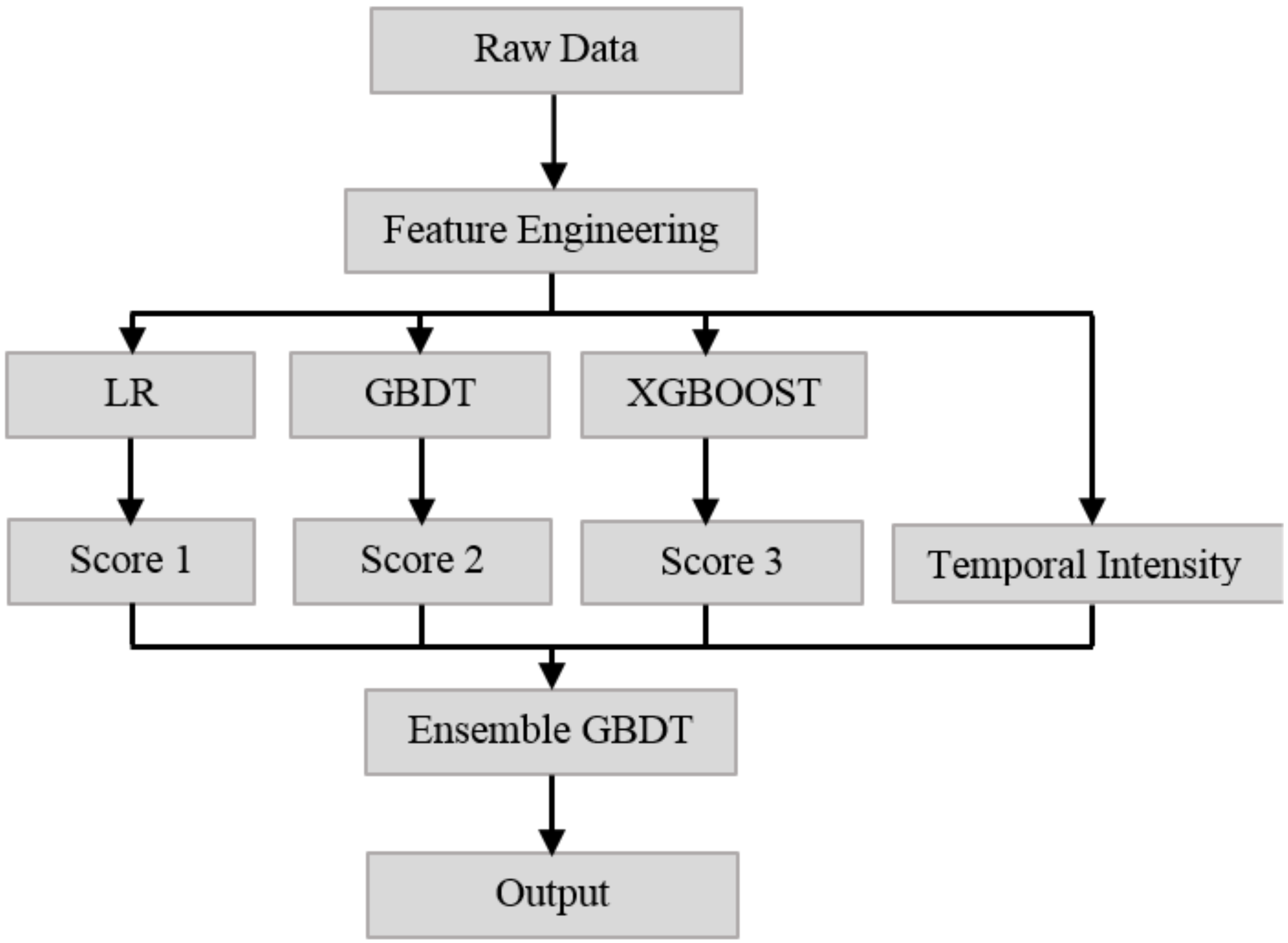
Overall Framework

An Hierarchical LTR framework

- LTR with various models to absorb diverse information
- Ensemble GBDT with the input from initial ranking
- Hawkes Process (self-exciting) to capture temporal patterns

Final Goal

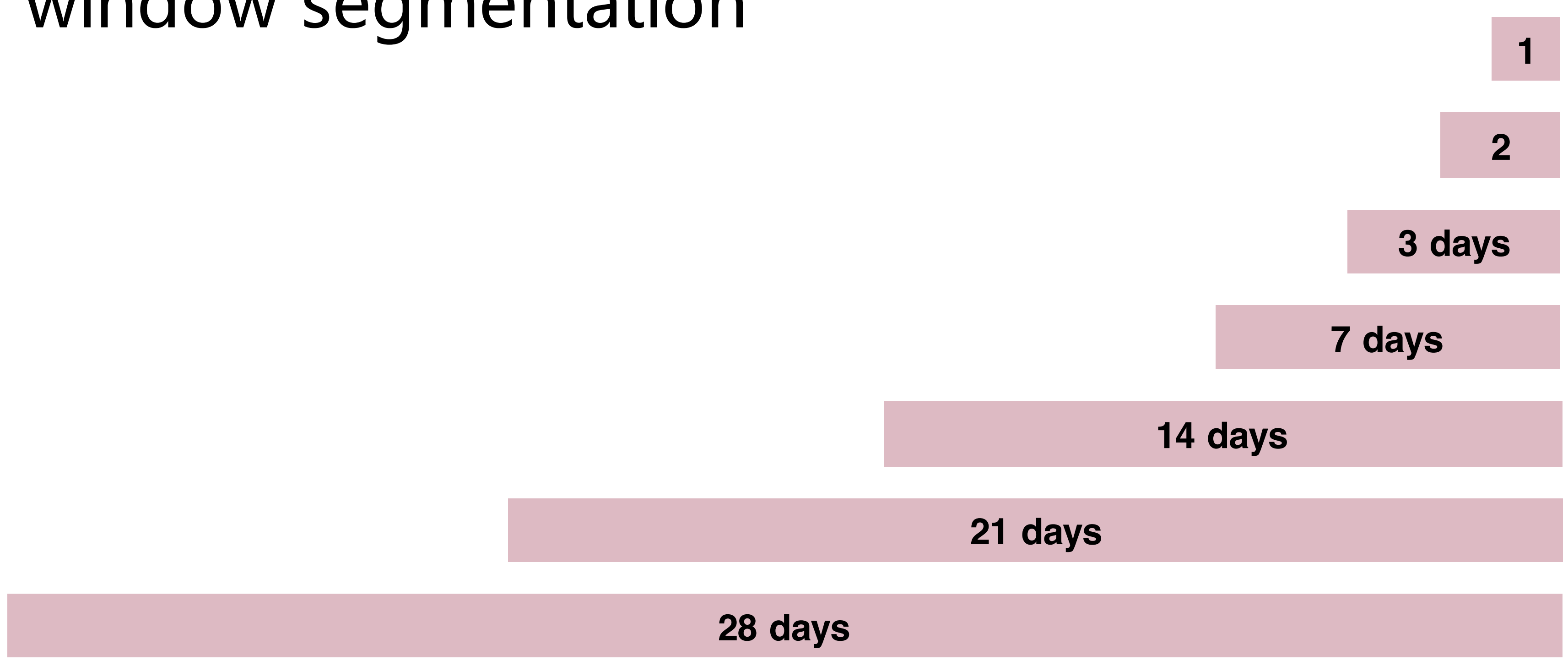
- Generate the right list of job posting to recommend in the right timing



Feature Engineering

Key characteristics of features

- Three basic feature categories: user, item, user-item interaction features
- Extended features with further analysis and statistics
- Aggregation is the remedy for sparsity
- Time window segmentation



Features from Low to High Level

user&item profile: 0/1

user/item/ui interaction statistics: (time window) 0/1, count, distinct count, days……

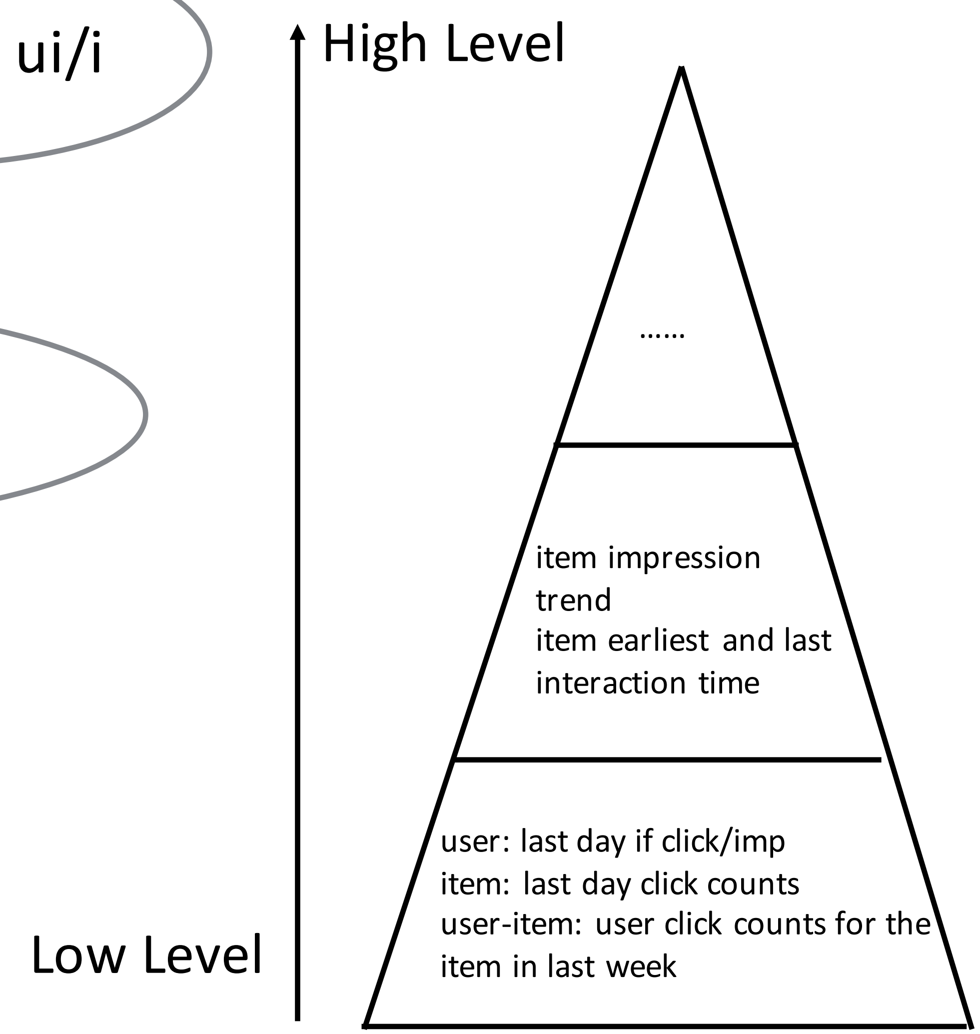
similarity(interaction&profiling): u2u, i2i, u2i

ratio: ui/u, ui/i

trend: item interaction counts

poi: item, user interaction

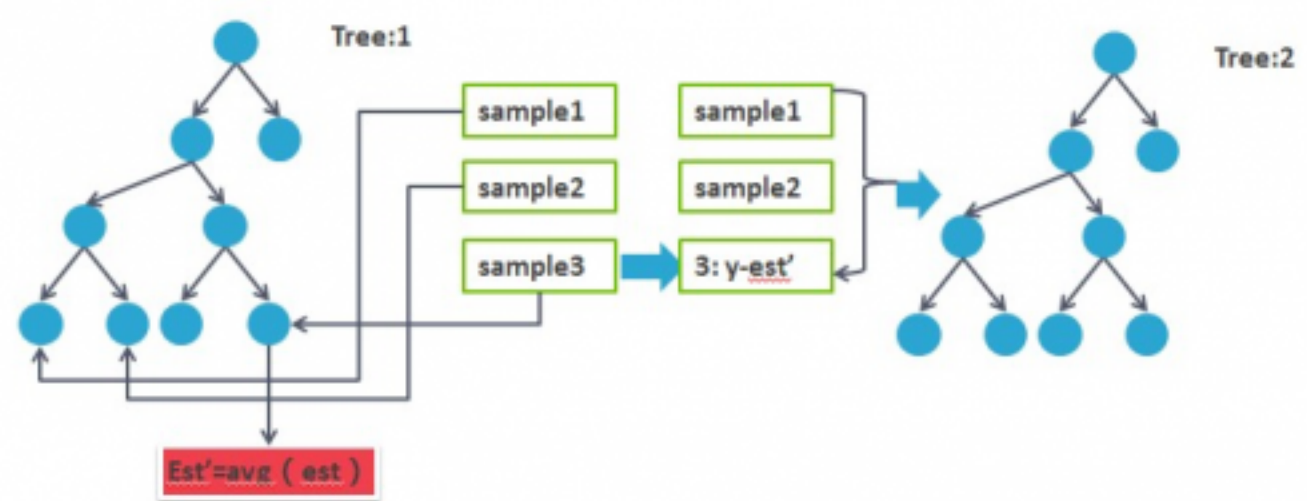
conditional probability: impression to interaction, interaction to interaction



Learning to Rank with GBDT

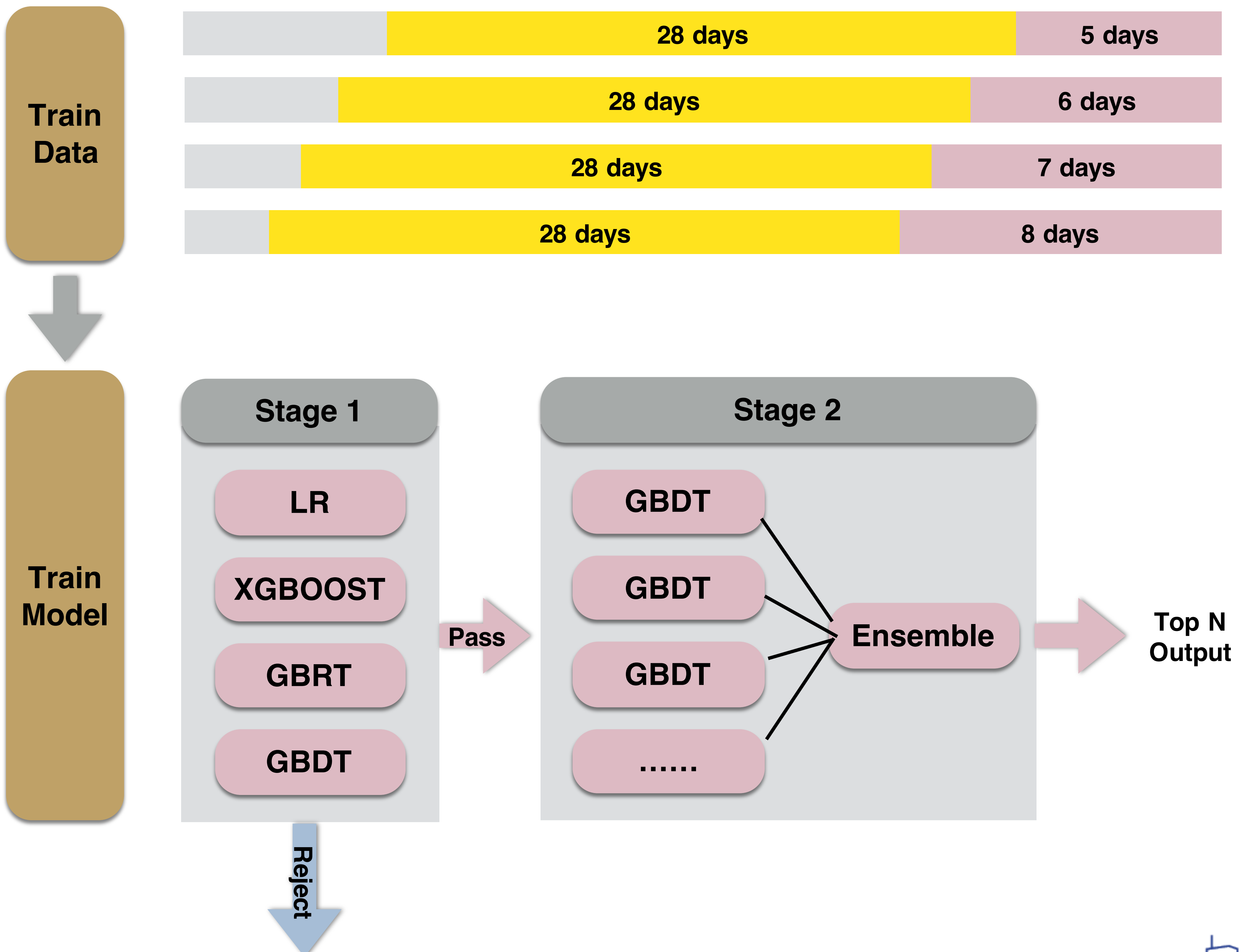
GBDT performs as a major ranking model

- Residual v.s. gradient
- Different loss functions for boosting
- Pointwise/Pairwise method



Name	Loss	Derivative	f^*	Algorithm
Squared error	$\frac{1}{2}(y_i - f(\mathbf{x}_i))^2$	$y_i - f(\mathbf{x}_i)$	$\mathbb{E}[y \vert \mathbf{x}_i]$	L2Boosting
Absolute error	$\vert y_i - f(\mathbf{x}_i) \vert$	$\operatorname{sgn}(y_i - f(\mathbf{x}_i))$	$\operatorname{median}(y \vert \mathbf{x}_i)$	Gradient boosting
Exponential loss	$\exp(-\tilde{y}_i f(\mathbf{x}_i))$	$-\tilde{y}_i \exp(-\tilde{y}_i f(\mathbf{x}_i))$	$\frac{1}{2}\log\frac{\pi_i}{1-\pi_i}$	AdaBoost
Logloss	$\log(1+e^{-\tilde{y}_i f_i})$	$y_i - \pi_i$	$\frac{1}{2}\log\frac{\pi_i}{1-\pi_i}$	LogitBoost

Model Training



Cold Start

Construct user self-introduction by raw features.

- Demographic data, education degree, career level, work experience, etc.
- Interactions between job roles and job title(or tag).

Training and recommendation

- Apply LDA to find the latent topic model
- Apply kmeans to get user clusters.
- Use KNN to get the top-n result.

Temporal Intensity with Hawkes Process

Self-exciting point process for recurrent event

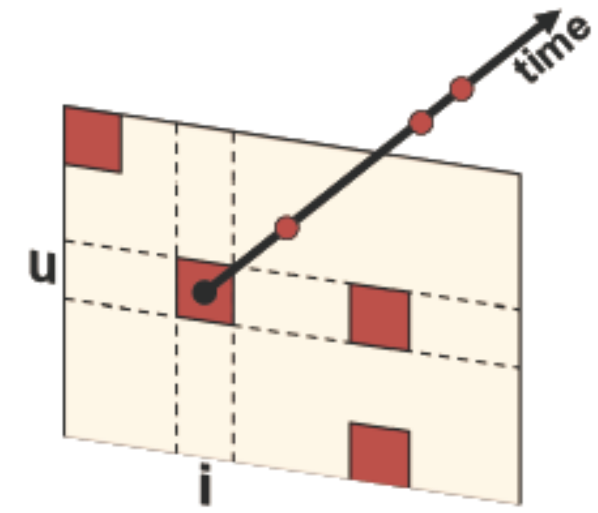
- The conditional probability that the next user-item event happens at time t

$$\lambda(t) = \lambda_0 + \alpha \sum_{j=1}^{n} \gamma(t, t_j)$$

$\{t_1, t_2, \cdots, t_n\}$ user-item event

λ_0 baseline intensity

$\gamma(t, t_j)$ temporal dependency



User-item event Model, Du, etal. 2015

- Homogenous point process

$$\lambda(t) = \lambda_0$$

Low Rank Hawkes Process

- Generalize the modeling the event of single user-item pair to all user-item pairs

$$\lambda^{u,i}(t) = \lambda_0^{u,i} + \alpha^{u,i} \sum_j \gamma(t, t_j^{u,i})$$

$\mathbf{\Lambda}_0$ baseline intensity matrix

$\boldsymbol{A}$ self exciting matrix

$\gamma(t, t_j^{u,i}) = \exp(-(t - t_j^{u,i})/\sigma).$ exponential form

- Low rank assumption: both users and items can be categorized into limited number clusters

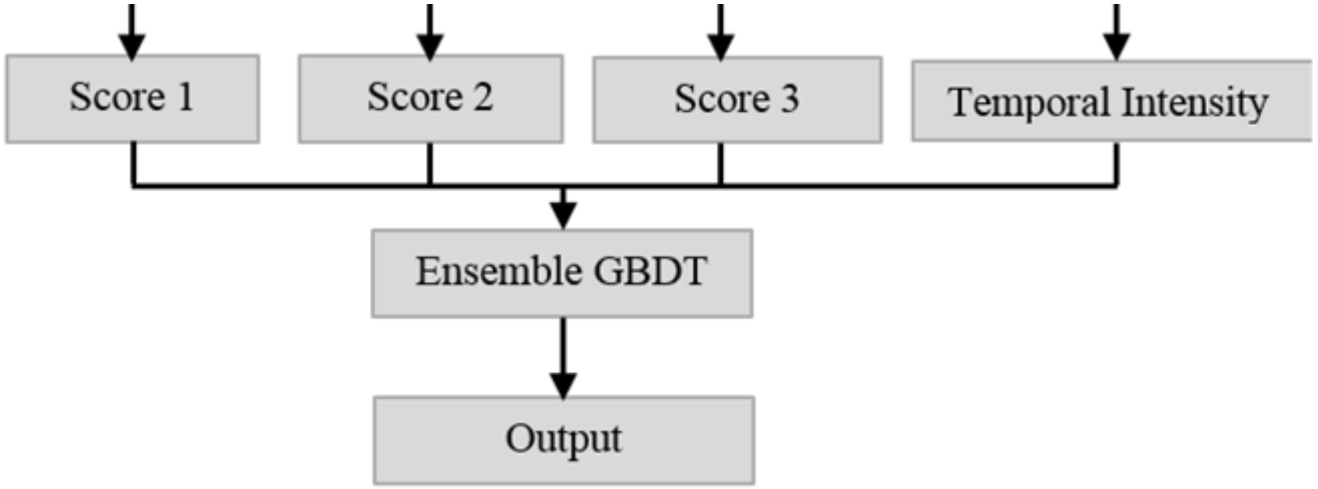
$$\|\mathbf{\Lambda}_0\|_* \leqslant \lambda', \|\boldsymbol{A}\|_* \leqslant \beta'$$

Ranking with Temporal Intensity

- Item recommendation with temporal intensity (Du, etal. 2015)

1. Calculate $\lambda^{u,i}(t)$ for each item i.
2. Sort the items by the descending order of $\lambda^{u,i}(t)$.
3. Return the top-k items.

- Temporal intensity as additional features



Results and Future Work

➤ Results in RecSys 2016

RecSys Challenge

leaders

Leaders

official results   unofficial leaderboard

Official Results

Rank	Team	Leaderboard Score	Full Score
1	YunOS-OneSearch	681707.38	2052185.54
2	mim-solutions	675985.03	2035964.16
3	DaveXster	665592.06	2005263.73

➤ Ongoing Studies:

- study the impact of the temporal patterns
- combine both contextual and temporal information
- investigate self-exciting and self-correcting process

Thank You!