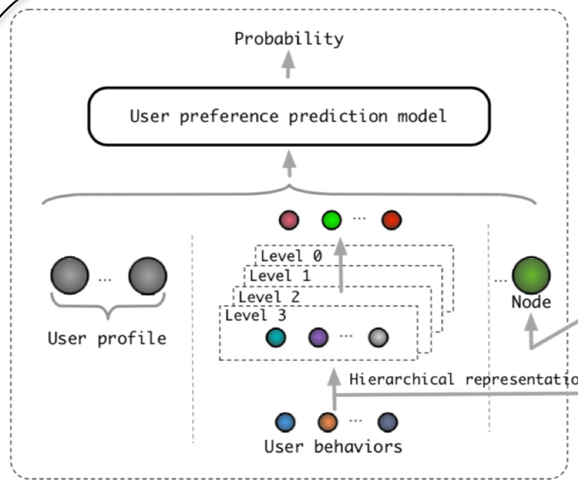
Joint Optimization of Tree-based Index and Deep Model for Recommender Systems

Han Zhu, Daqing Chang, Ziru Xu, Pengye Zhang, Xiang Li, Jie He, Han Li, Jian Xu, Kun Gai

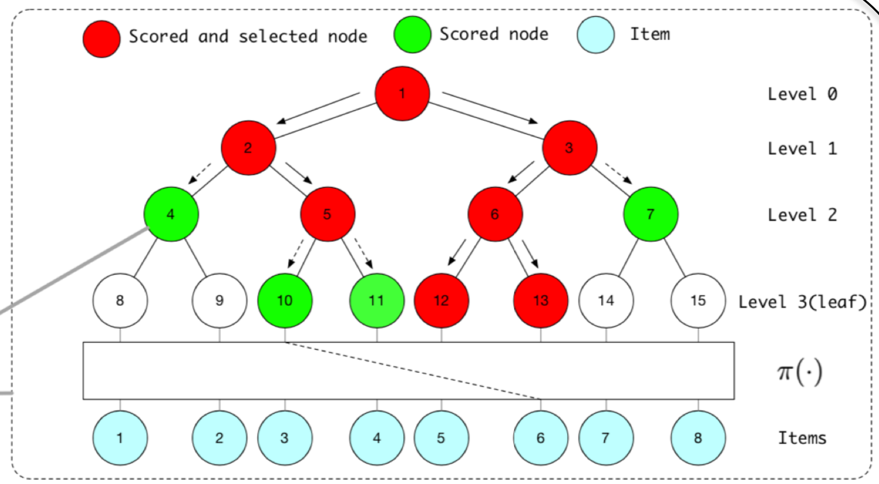
{zhuhan.zh, daqing.cdq, ziru.xzr, pengye.zpy, yushi.lx, jay.hj, lihan.lh, xiyu.xj, jingshi.gk}@alibaba-inc.com

Alibaba.com Global trade starts here.™

NeurIPS | 2019



(a)



(b)

Introduction

The previous work Tree-based Deep Model (TDM) creatively proposes to use tree index for large-scale recommendation. By using advanced deep model to retrieve user interests in the tree, recommendation accuracy is improved greatly. In this paper, we give a framework to jointly optimize the tree index and deep model with a unified global objective.

Max-heap preference modeling

$$p^{(l)}(n|u) = \frac{\max_{n_c \in \{n's \text{ children in level } l+1\}} p^{(l+1)}(n_c|u)}{\alpha^{(l)}}$$

From max-heap to samples

Samples from implicit feedback $\{(u^{(i)}, c^{(i)})\}_{i=1}^{n}$

Ground-truth preference $p\left(\pi(c)|u;\pi\right) = 1$

From max-heap $\{p(b_j(\pi(c))|u;\pi) = 1\}_{j=0}^{l_{max}}$

Unified global loss function

Denote the user preference model parameters as θ and the tree structure definition as π, the unified global loss w.r.t. (θ, π) is the cross-entropy loss of each sample in each level:

$$\mathcal{L}(\theta,\pi) = -\sum_{i=1}^{n}\sum_{j=0}^{l_{max}} \log \hat{p}\left(b_j(\pi(c^{(i)}))|u^{(i)};\theta,\pi\right)$$

Joint optimization framework

Algorithm 1: Joint learning framework of the tree index and deep model

Input: Loss function $\mathcal{L}(\theta,\pi)$, initial deep model $\mathcal{M}$ and initial tree $\mathcal{T}$

for $t = 0, 1, 2 \dots$ **do**
  Solve $\min_\theta \mathcal{L}(\theta,\pi)$ by optimizing the model $\mathcal{M}$.
  Solve $\max_\pi -\mathcal{L}(\theta,\pi)$ by optimizing the tree hierarchy with Algorithm 2
end for

Output: Learned model $\mathcal{M}$ and tree $\mathcal{T}$

Alternatively optimize the user preference model and tree structure. The optimization of preference model can be solved by standard back-propagation. The optimization of tree structure is equivalent to a maximum matching problem of weighted bipartite graph, which has no efficient solution when the corpus size is very large.

Approximate tree learning

$$\mathcal{L}_{c_k}^{s,e}(\pi) = \sum_{(u,c)\in\mathcal{A}_k}\sum_{j=s}^{e} \log \hat{p}\left(b_j(\pi(c))|u;\theta,\pi\right)$$

Algorithm 2: Tree learning algorithm

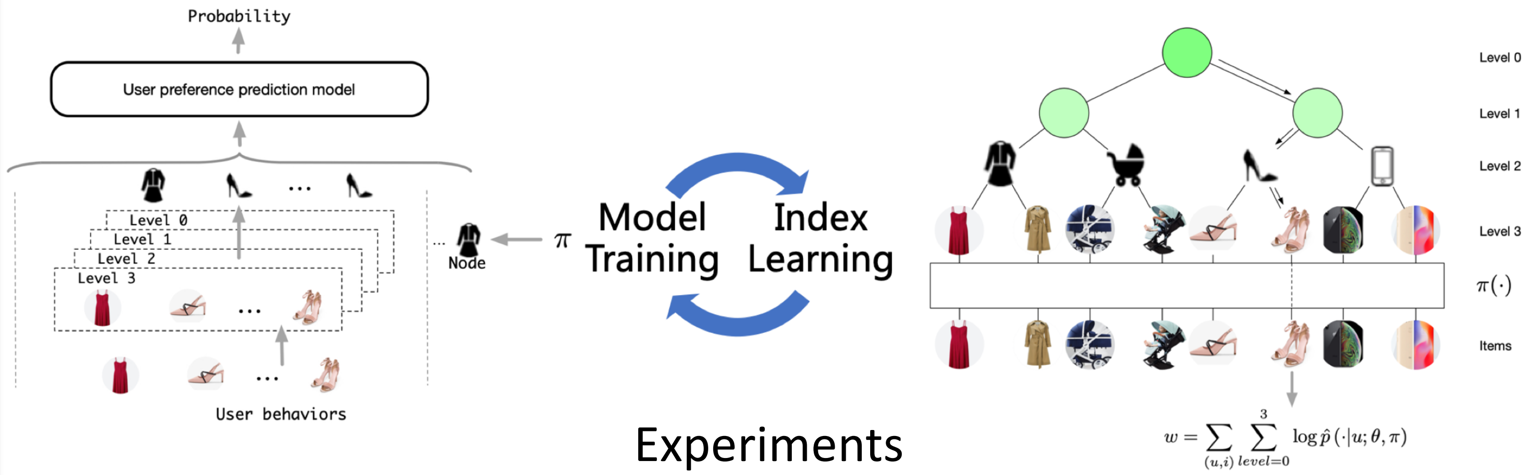
Input: Gap d, max tree level $l_{\max}$, original projection π_{old}

Output: Optimized projection π_{new}

Set current level $l \leftarrow d$, initialize $\pi_{new} \leftarrow \pi_{old}$
while $d > 0$ **do**
  for each node n_i in level $l - d$ **do**
    Denote $\mathcal{C}_{n_i}$ as the item set that $\forall c \in \mathcal{C}_{n_i}, b_{l-d}(\pi_{new}(c)) = n_i$
    Find π^* that maximize $\sum_{c\in\mathcal{C}_{n_i}} \mathcal{L}_c^{l-d+1,l}(\pi)$, s.t. $\forall c \in \mathcal{C}_{n_i}, b_{l-d}(\pi^*(c)) = n_i$
    Update π_{new}. $\forall c \in \mathcal{C}_{n_i}, \pi_{new}(c) \leftarrow \pi^*(c)$
  end for
  $d \leftarrow \min(d, l_{max} - l)$
  $l \leftarrow l + d$
end while

In order to tackle the corpus size problem in tree learning, we propose an approximate algorithm to learning the tree structure step-by-step.

Illustration of joint training and hierarchical user representation



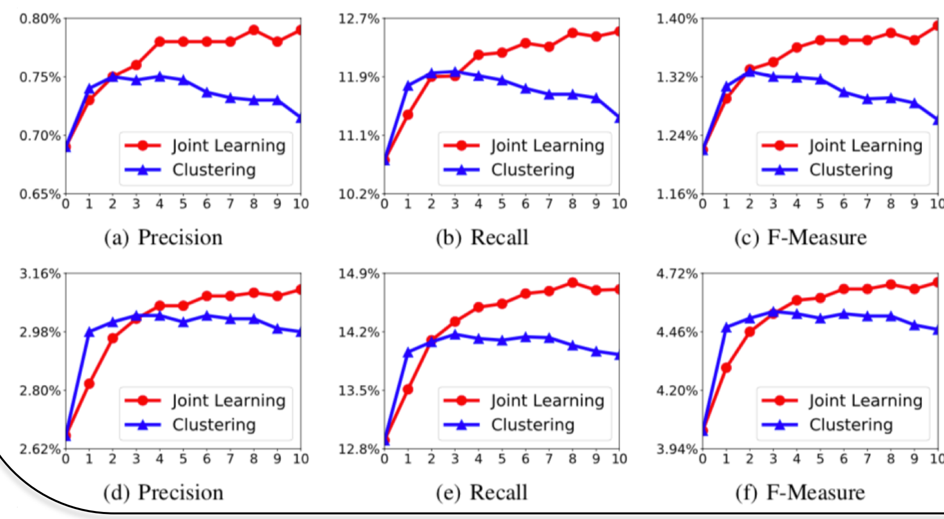
Experiments

Right: Dataset dimensions

Bottom: Overall offline results

	Amazon Books	UserBehavior
# of users	294,739	969,529
# of items	1,477,922	4,162,024
# of categories	2,637	9,439
# of records	8,654,619	100,020,395

Method	Amazon Books			UserBehavior		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Item-CF	0.52%	8.18%	0.92%	1.56%	6.75%	2.30%
YouTube product-DNN	0.53%	8.26%	0.93%	2.25%	10.15%	3.36%
HSM	0.42%	6.22%	0.72%	1.80%	8.62%	2.71%
TDM	0.50%	7.49%	0.88%	2.23%	10.84%	3.40%
DNN	0.56%	8.57%	0.98%	2.81%	13.45%	4.23%
JTM-J	0.51%	7.60%	0.89%	2.48%	11.72%	3.73%
JTM-H	0.68%	10.45%	1.19%	2.66%	12.93%	4.02%
JTM	**0.79%**	**12.45%**	**1.38%**	**3.11%**	**14.71%**	**4.68%**



(a) Precision (b) Recall (c) F-Measure

(d) Precision (e) Recall (f) F-Measure

Left: Jointly model and tree learning results

Bottom: Online A/B test results

Metric	Baseline	TDM	JTM
CTR	0.0%	+5.4%	+11.3%
RPM	0.0%	+7.6%	+12.9%