

AI在双11中的个性化搜索和决策实践

原创 云栖社区 阿里云云栖号 2017-12-16

12月13-14日，由云栖社区与阿里巴巴技术协会共同主办的《2017阿里巴巴双11技术十二讲》顺利结束，集中为大家分享了2017双11背后的黑科技。本文是《人工智能在搜索中的应用》演讲整理，本文首先介绍了AI如何在搜索中落地，进而重点介绍了个性化搜索，包括图搜索、个性化召回，接着又分享了智能决策的探索。内容如下。

分享嘉宾：



三桐：阿里巴巴资深算法专家，淘宝商品搜索算法方向，主持多个搜索技术项目，致力于推动大规模机器学习、深度学习等在商品搜索的应用。

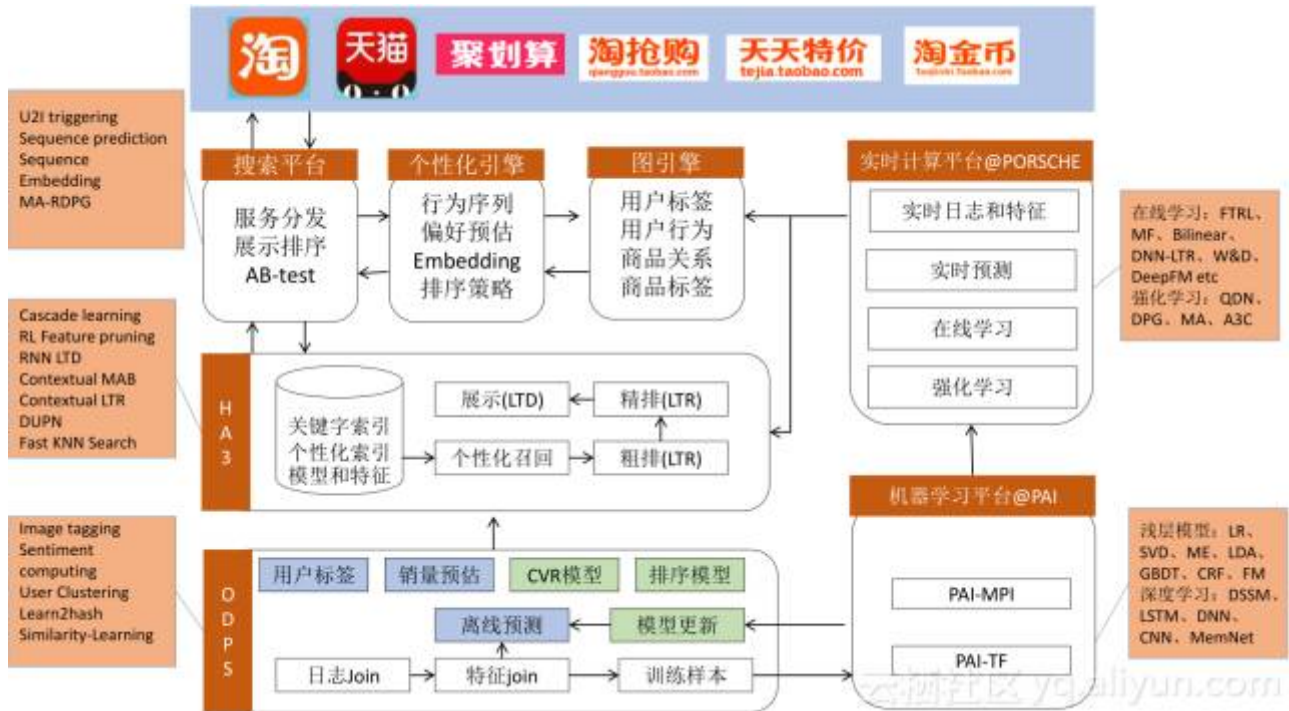
搜索智能化之路

搜索是有交互的个性化推荐产品，具体体现在query的引导，提升整体效率；其次是个性化，主要问题是行业冷启动、购物链路比较长，包括淘宝产品如搜索、推荐、广告的个性化，用户会在其中作不同的穿插；第三，电商搜索是商业平台的搜索产品，涉及到消费者、商家和买家三方市场；最后是数据闭环，淘宝数据链与网页不一样，用户行为（浏览、点击、成交、评价、物流）形成全链路闭环数据，商品数据也上传到淘宝平台中。

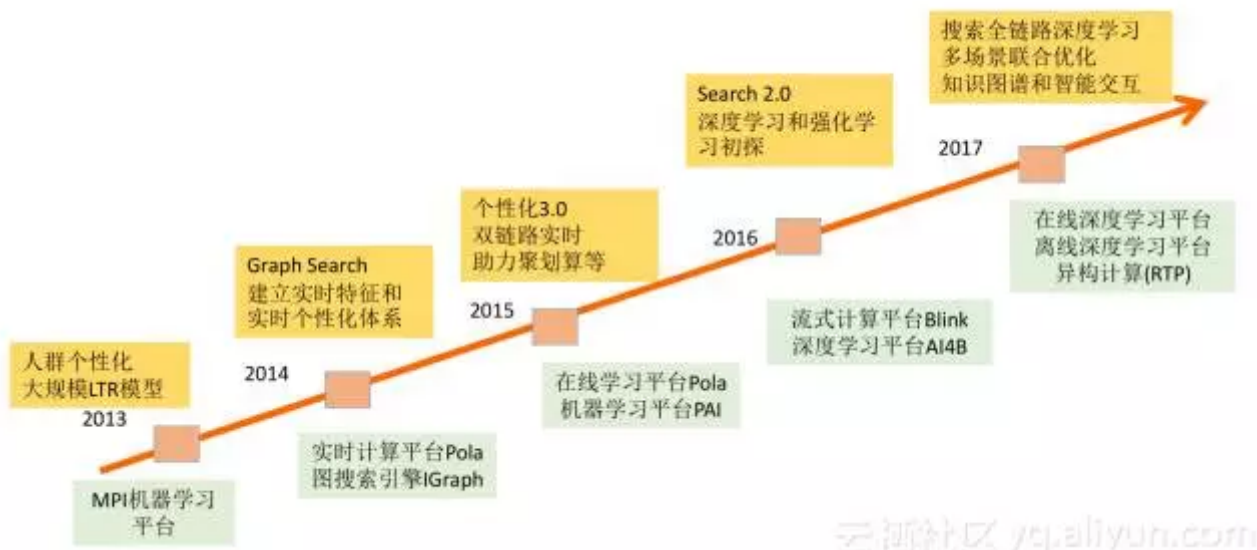
AI在搜索中如何落地？

在算法层面,需要考虑用户理解、个性化匹配和排序策略,还要对购物链路进行优化,也要对深度学习在线预估进行探索调研。学习(大数据+大模型+实时+在线预估)+决策(自动探索和学习)是在算法中所作的人工智能,学习具体体现在由单任务-多任务、由浅层-深层、由batch-sgd,决策是指探索未知领域、多任务联合优化等。

AI在搜索中的挑战主要是如何做超大规模机器学习平台,如何进行在线深度学习,如何进行稀疏深度模型处理和加速,如何进行异构计算,如何进行多模信息融合、多任务表征学习、多目标学习,以及如何处理长短期收益平衡?



图为算法系统框架,包括离线数据处理和机器学习平台和在线数据处理和机器学习平台,离线主要做预测和模型训练,在线主要做个性化引擎和搜索服务。算法层面我们实现了常见的强化学习、在线学习、深度学习等主要模型,我们还为搜索业务制定了特定模型。



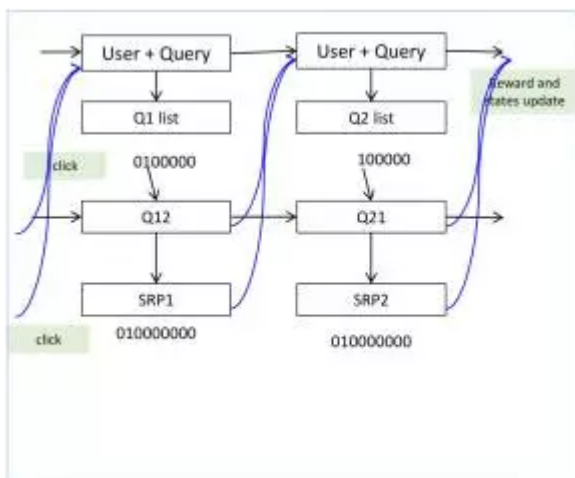
在系统演化方面我们取得了很多进展,比如:

- 流式计算Blink：实时日志Join，在线特征抽取和预测和在线学习 TF-PAI@Porsche；
- 图搜索引擎 IGRAPH：分布式超大规模，实现实时更新、在线查询、过滤、复合查询和推理；
- 在线排序算分服务 RTP：服务解耦，CPU + GPU异构计算；
- 机器学习平台：离线、增量和在线学习，混布和异构计算；
- 引擎：个性化索引和向量召回，多级排序和重排；
- 平台化：服务产品化，算法平台化。

整体来看，算法、系统和业务正在交织发展，机器学习平台、在线学习平台和实时计算平台是算法的前提条件，我们一直在探索深度学习和强化学习，并且在搜索全链路深度学习、多场景联合优化和知识图谱和智能交互方面取得一些进步。

个性化搜索

关键字推荐



Action: Queries

Reward: query + related SRP

States: User + Current Query

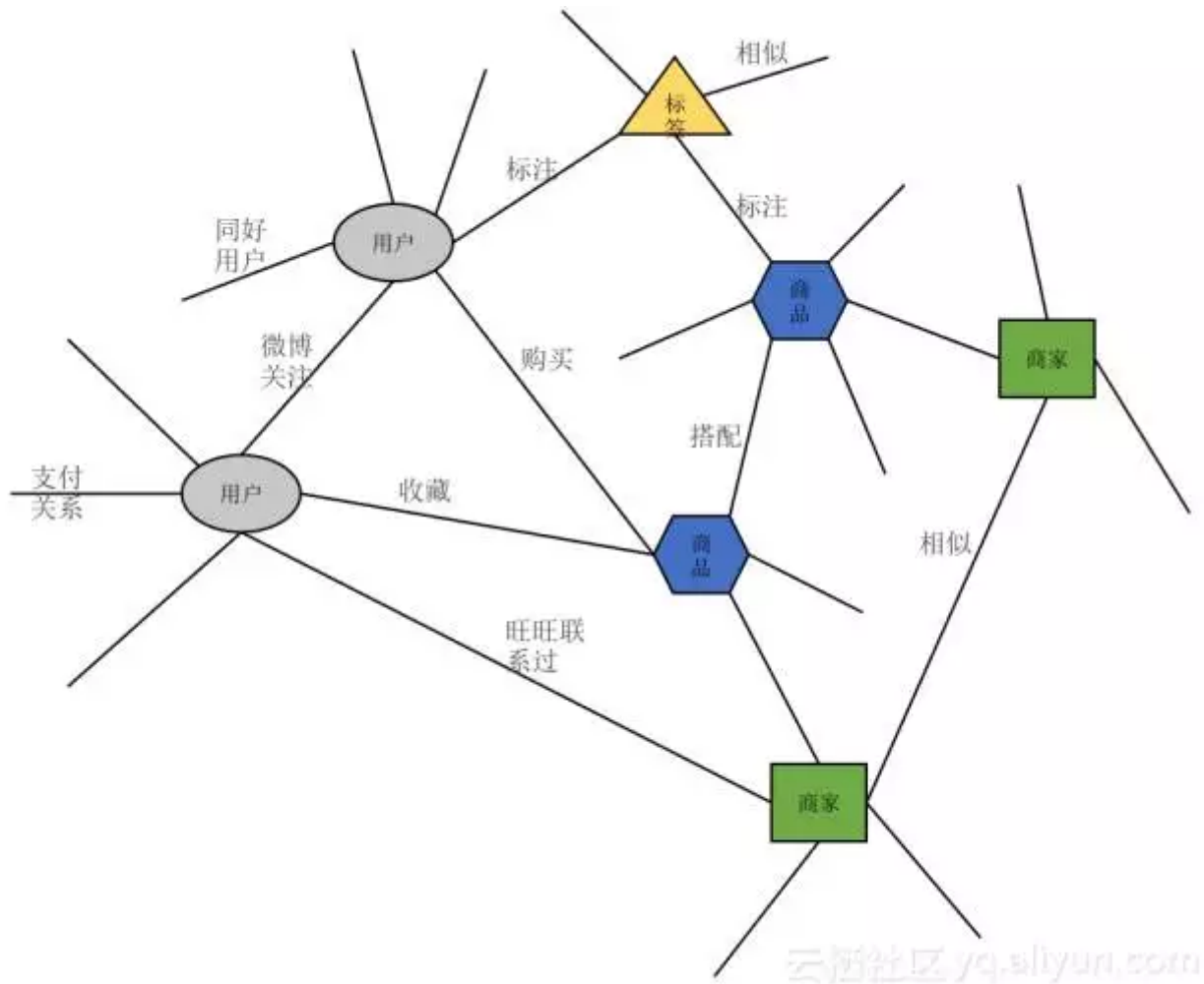
DQN

$$Q_{i+1}(s, a) = \mathbb{E}_{s'} \left[r + \gamma \max_{a'} Q_i(s', a') \mid s, a \right]$$

$$\mathcal{L}(w) = \mathbb{E} \left[\left(\underbrace{r + \gamma \max_{a'} Q(s', a', w)}_{\text{target}} - Q(s, a, w) \right)^2 \right]$$

推荐关键字主要参考用户和当前query，我们会给出关键字list，用户会选择想要的query进行下一步的搜索，那么，如何评价query list的好坏呢？我们从直接收益和间接收益当中学习，包括query点击和query引导的用户搜索页是否成交，以及用户选择query后，会产生新的query做query推荐。Query推荐是一个序列优化问题，适合使用强化学习来做，所以我们基于DQN做query推荐，每个query是一个action，状态是用户和当前query。

图搜索



个性化搜索数据体系叫电商图谱，是以用户为核心建立用户社交圈、用户生活圈、用户购物行为、用户购物行为、用户标签和辅助关系，通过数据挖掘挖掘出这些关系，基于此图做简单个性化，电商图谱约有10亿个实体、近千亿条边和几十种关系类型。

个性化召回

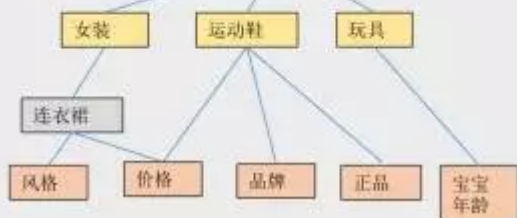
查询词

用户上下文

人口：年龄、性别、职业
场景：温度、LBS
标签：风格、购买力、车型
行为：浏览、点击、收藏、加购、评价
过商品、店铺、品牌
社交关系：好友

上下文分析

用户意图树



搜索引擎

关键字：商品list
商品：行为相似商品list、内容相似商品list、商品簇list
店铺：商品list
品牌：商品list
标签：商品list
场景：商品list

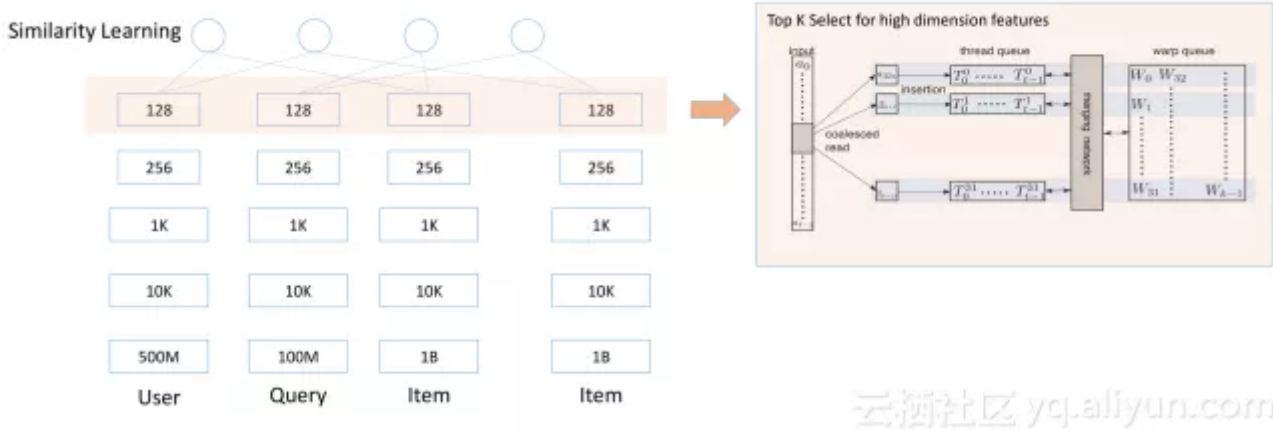
个性化召回

"关键字+商品关系+标签/场景混合索引"

+ "关键字+用户行为+用户标签/场景混合召回"

'query' OR ('query' AND (商品簇 OR 品牌 OR 店铺 OR 商品 OR rank 场景 or 标签))

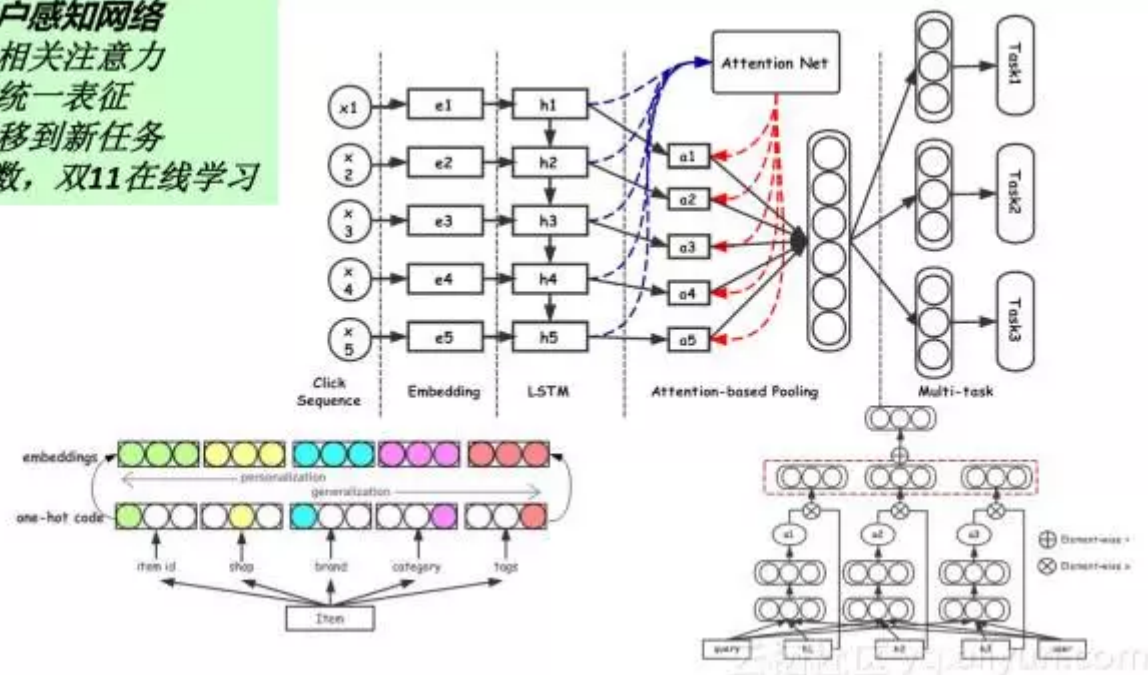
个性化搜索中较重要的是个性化召回，首先基于Igraph数据取出基本信息，因为搜索中是有query的，每个query中用户信息是不一样的，所以首先会基于上下文对基本信息进行过滤，再拿这个信息去引擎中召回结果。传统搜索引擎只是关键字，在这里除了关键字外，还有很多个性化辅助信息参与召回。



由于布尔召回的召回深度较浅，所以我们使用基于深度学习的向量化召回，向量化召回包括做similarity learning把query、user和商品变成低维的向量，再通过Top K高位向量的召回实现向量化召回，召回深度有所提高。

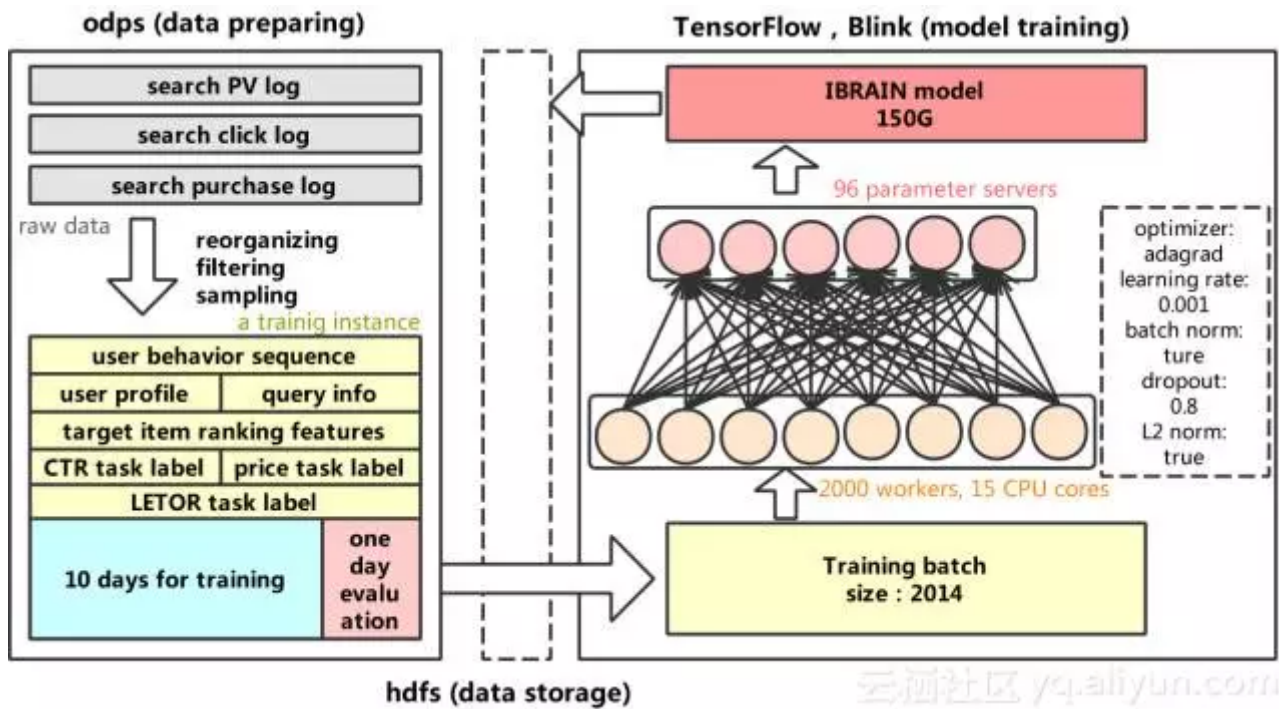
Deep User Perception Network

深度用户感知网络
上下文相关注意力
多任务统一表征
容易迁移到新任务
10B 参数，双11在线学习



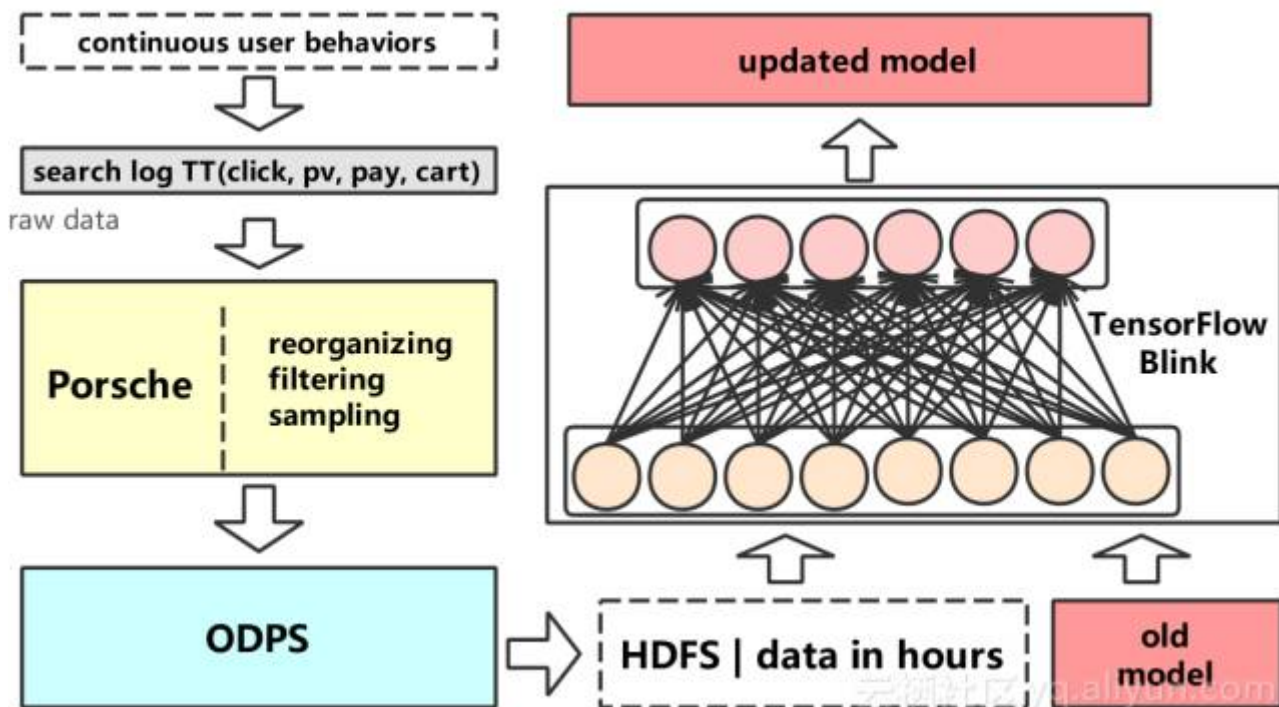
如何深度感知用户？从个性化角度说，一是年龄、性格等用户标签，一是用户行为序列；从搜索角度说，用户行为对于当前搜索的重要性是不一样的。如图，首先输入 $x_1 \sim x_n$ 的行为序列，首先做embedding，然后再经过LSTM做Attention，最后通过相似task学习统一用户表征。

offline training



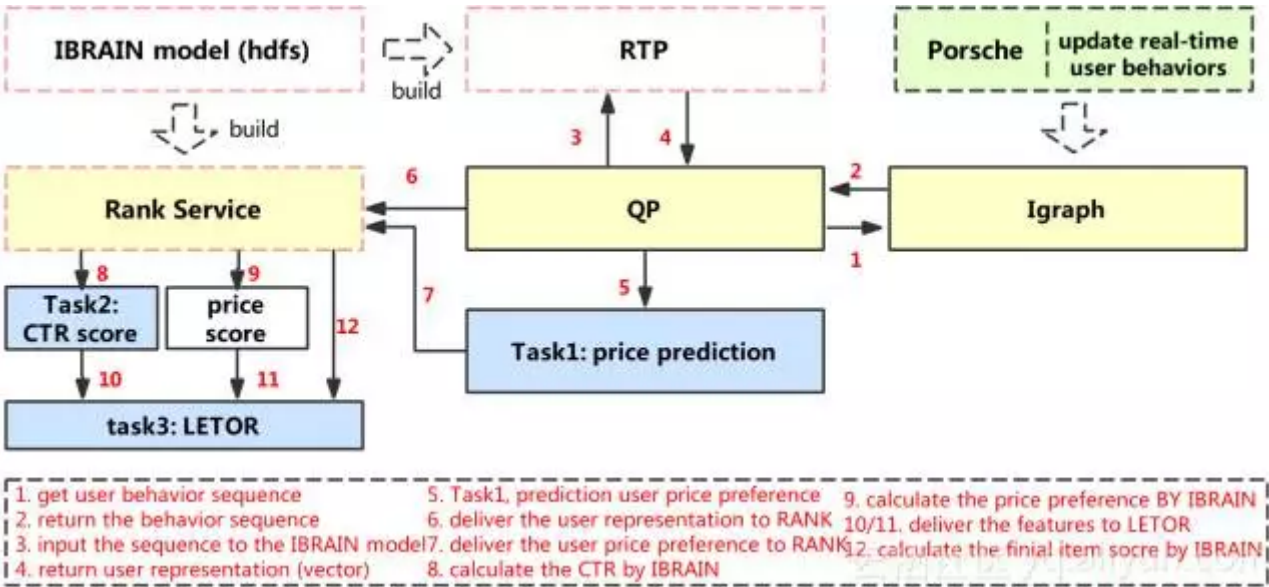
离线训练流程如图，主要解决如何做超大规模深度模型训练，我们在训练时候有2000个workers、96个parameter servers，每个worker应用15个GPU核，模型规模达到150G。在模型训练本身，针对稀疏 id 特征进行了专门处理。

real-time updating



在线学习流程如图，电商商品变化较快，如果模型无法快速适应变化，效果肯定会大打折扣，所以我们实现了端到端的在线模型学习更新。

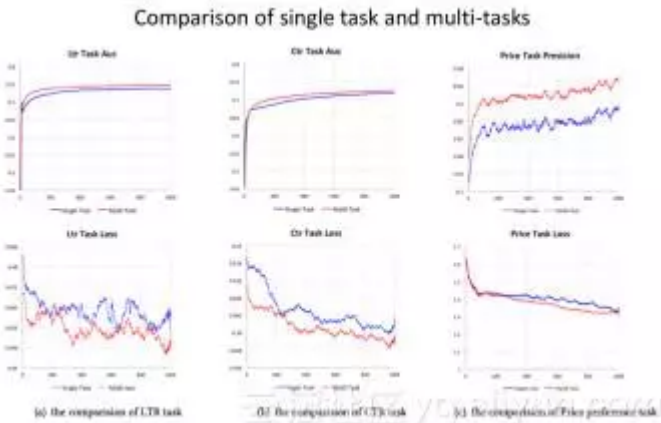
online processing



在线预测结果如图所示。

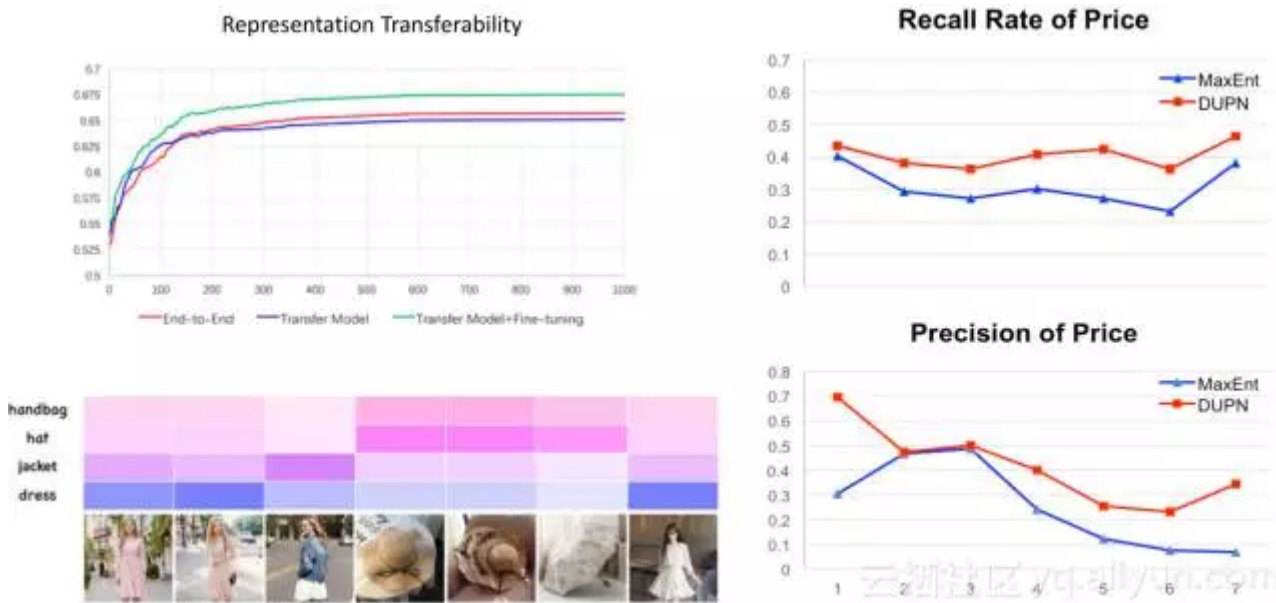
Comparison of Different Networks and Item Embedding

	LETOR AUC	CTR AUC	Price precision
DNN	0.71991	0.69266	36.404%
CNN	0.70557	0.67643	35.461%
LSTM	0.70569	0.67542	35.940%
pre-DUPN	0.73091	0.70127	38.079%
e2e-DUPN	0.75005	0.72519	44.011%



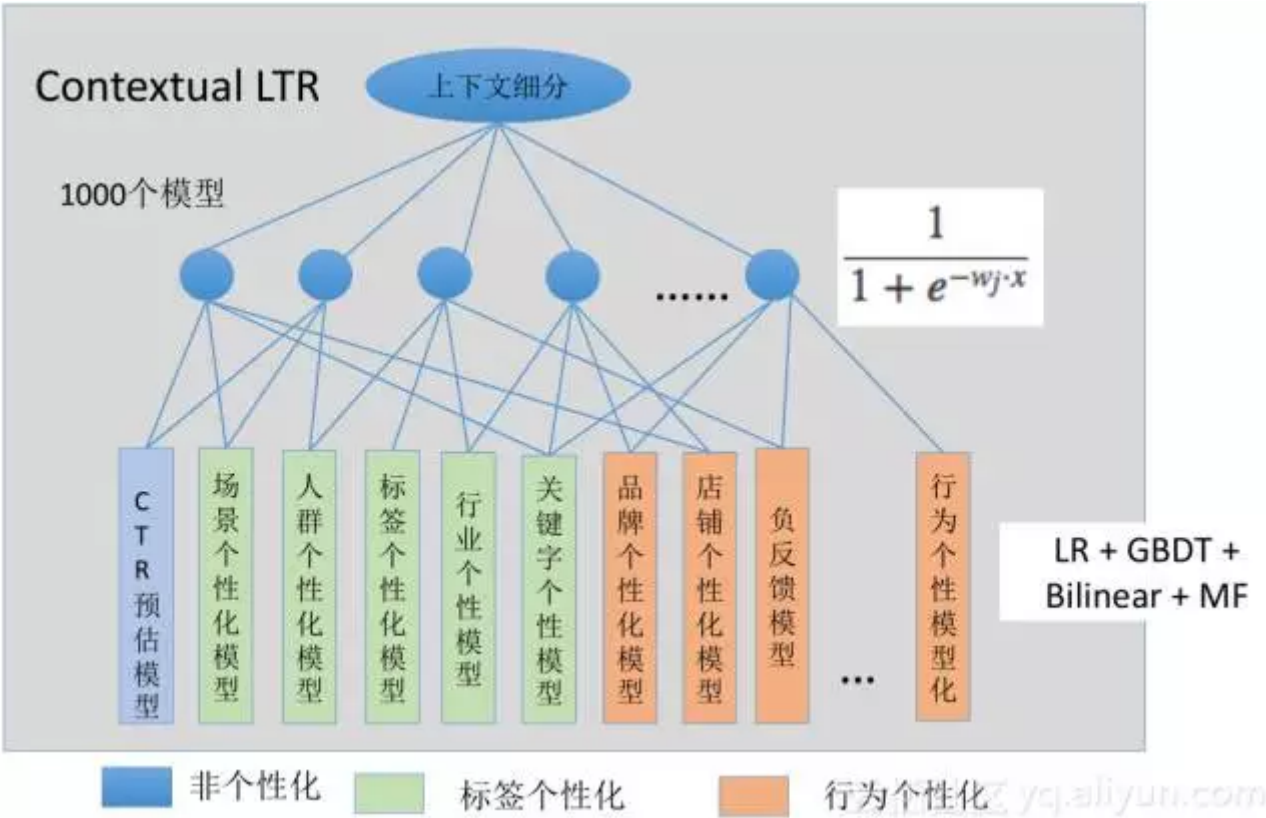
实验结果如图，我们会做不同的尝试实验，基于DUPN我们一是不做端到端训练，现将商品 embedding 部分用离线模型做预训练；二是做端到端模型训练。可以看到，端到端训练相比其它会有明显提升。

右图为多任务和单任务的对比，多任务准确率相对更高，对于一个任务来说，其它任务就是约束，得到用户表征更加通用。



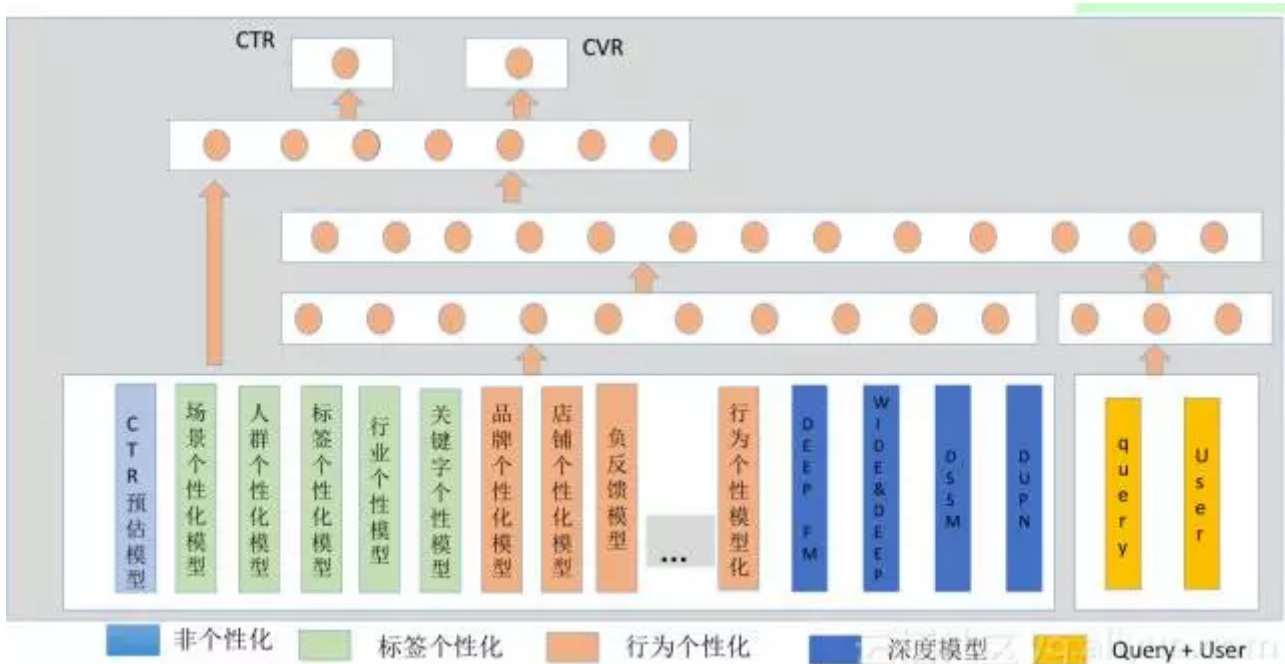
我们在做任务迁移时比传统迁移有明显优势，用户行为序列通过attention可以更好的学习与当前场景较相关的表征。

Contextual Linear LTR



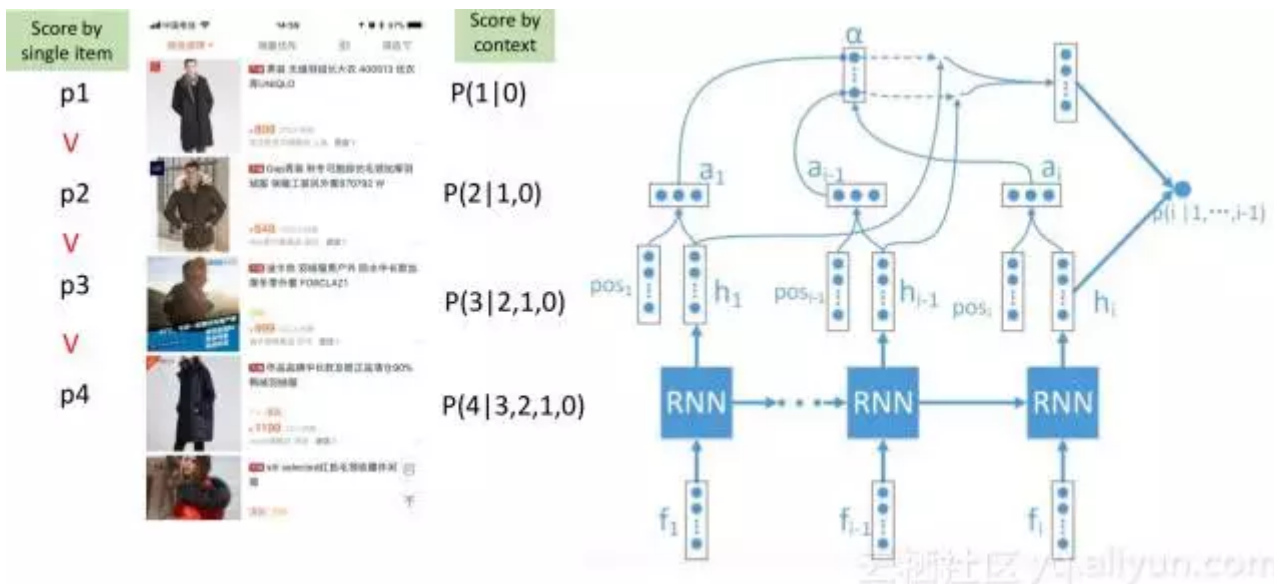
传统的LTR是指学习统一的模型，在电商搜索中不同行业不同用户下排序特征不一样，对此，我们做了基于上下文的LTR，第一版做了1000个排序模型，每个模型完全独立训练，其实际效果十分明显。但是，该场景划分是根据数据统计划分，具有很大不确定性，如果划分场景比较碎，做在线学习也有很大困难，模型训练稳定性较差。

Online Contextual Nonlinear LTR

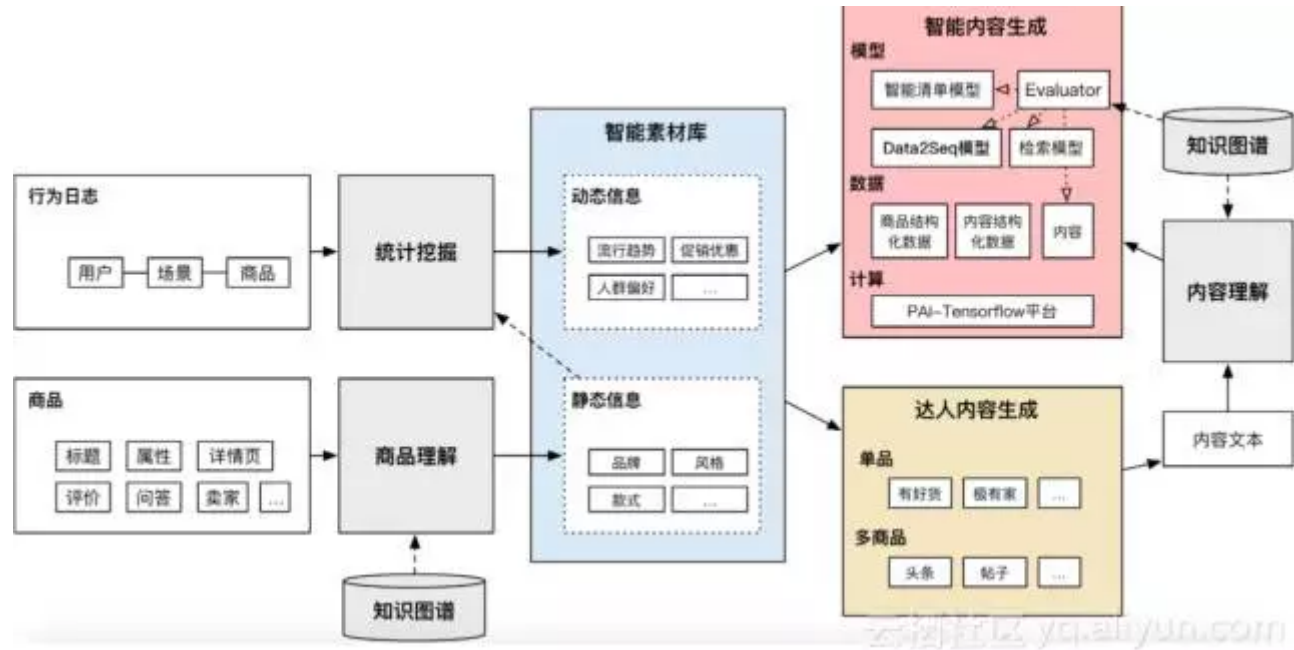


因此，我们使用深度学习得出在线上下文LTR模型。可以看到，在特征层面加入了深度特征，引入了query和user信息，首先会将商品特征进行结合再降维，用户也需进行结合再降维，再将商品特征和用户组合在一起再降维，最后输出时还会用原来商品特征和组合的降维特征再做CTR和CVR预估，通过深度学习非线性表达能力，实现基于用户和上下文感知的非线性排序模型。

Learn to Display



我们做了基于上下文的序列最优排序，利用前面已经排好的结果来预测下一个该排的商品，首先会对之前排好结果做embedding，基于embedding预测下一个商品，实现整页最优排序。



为了生成一些特色商品内容传递给消费者，也为了做商品聚合并生成主题，我们做了智能内容生成展示。

智能决策

智能决策与搜索的差异是：在搜索中无法给一些场景确定的样本，比如用户偏好的探索；其次是长短期受益的平衡。因此，实施在多目标环境下，根据环境反馈实时优化策略达到整体最优。

MAB 多样性探索



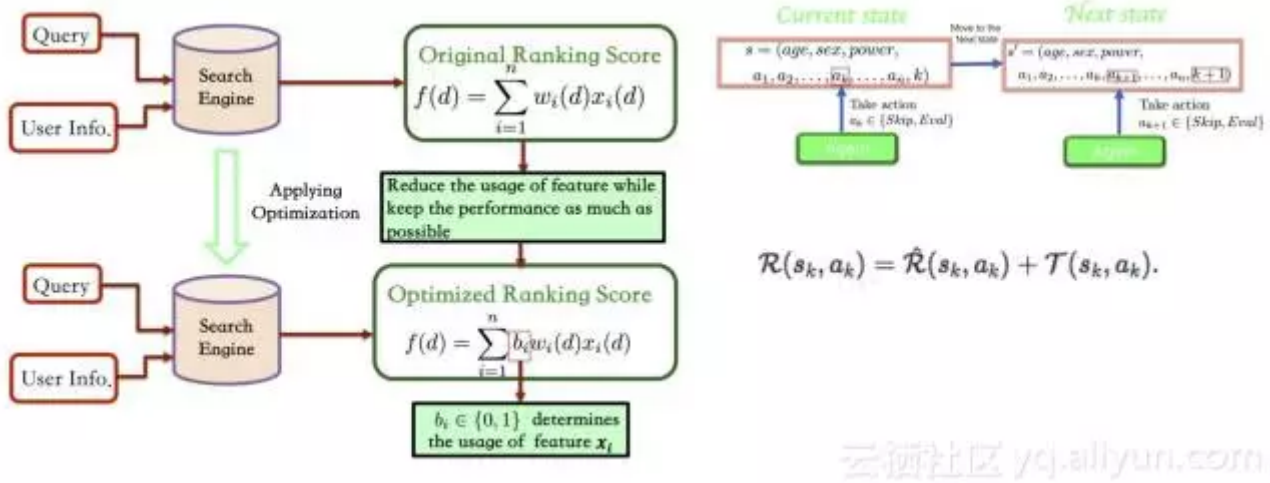
Initialization: $\forall \pi \in \Pi : w_t(\pi) = 1$
For each $t = 1, 2, \dots$:

- Observe x_t and let for $a = 1, \dots, K$
$$p_t(a) = (1 - K\rho_{min}) \frac{\sum_{\pi} \mathbf{1}[\pi(x_t) = a] w_t(\pi)}{\sum_{\pi} w_t(\pi)} + \rho_{min},$$

where $\rho_{min} = \sqrt{\frac{\ln |\Pi|}{KT}}$.
- Draw a_t from $p_{t,\cdot}$ and observe reward $r_t(a_t)$.
- Update for each $\pi \in \Pi$
$$w_{t+1}(\pi) = w_t(\pi) \exp \left(\frac{\rho_{min}}{2} \left(\mathbf{1}[\pi(x_t) = a_t] \frac{r_t(a_t)}{\rho_t(a_t)} + \frac{1}{\rho_t(\pi(x_t))} \sqrt{\frac{\ln N/\delta}{KT}} \right) \right)$$

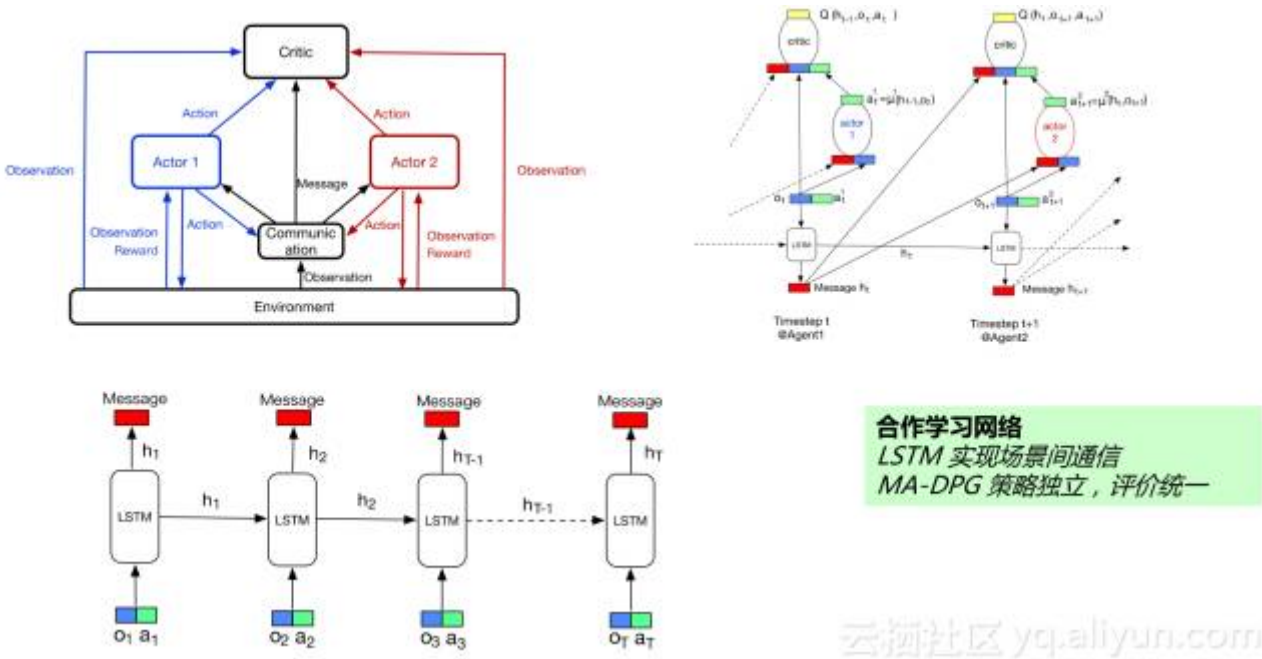
我们在搜索个性化结果中会有不同维度，使用MAB探索不同维度，分别去计算每个维度收益，使强个性化和弱个性化平衡，提升效率同时避免过度个性化。

feature pruning

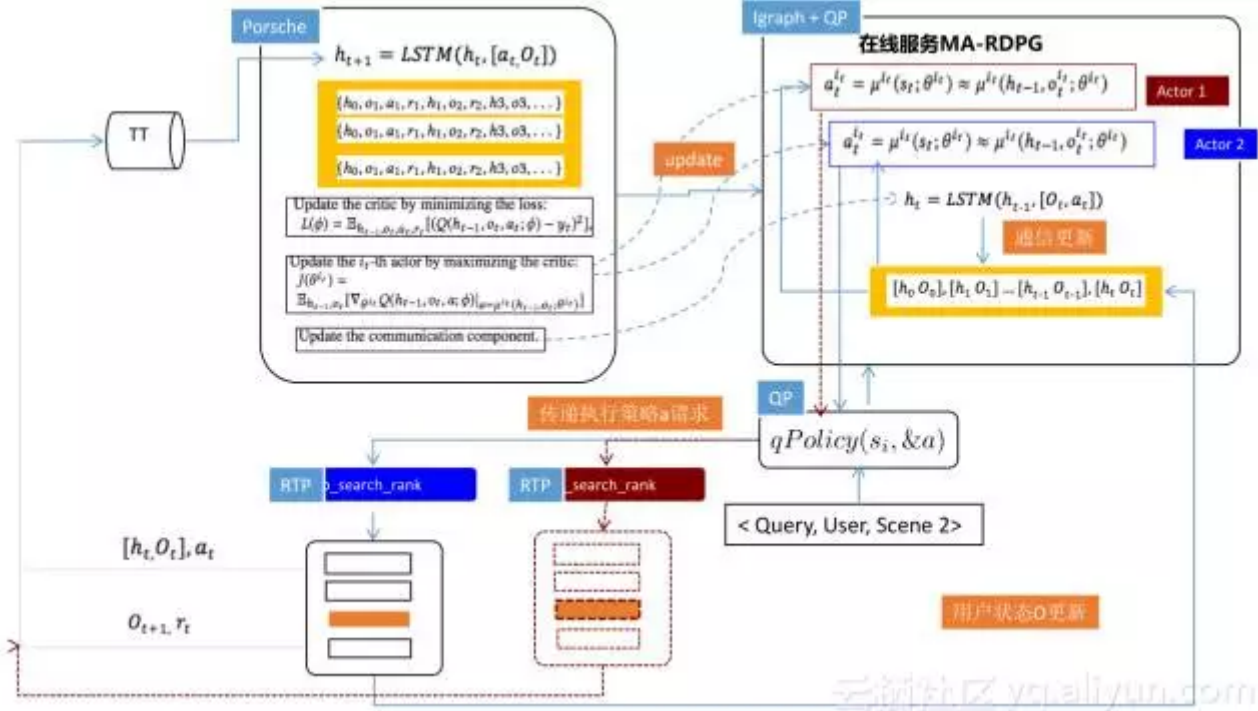


不同用户不同场景下，每个特征重要性不一样，如果不用一些特征，可以很好的提升性能。因此，我们用强化学习做用最少的特征学习用全部特征类似的排序结果，排序性能提升20%。

MA-RDPG



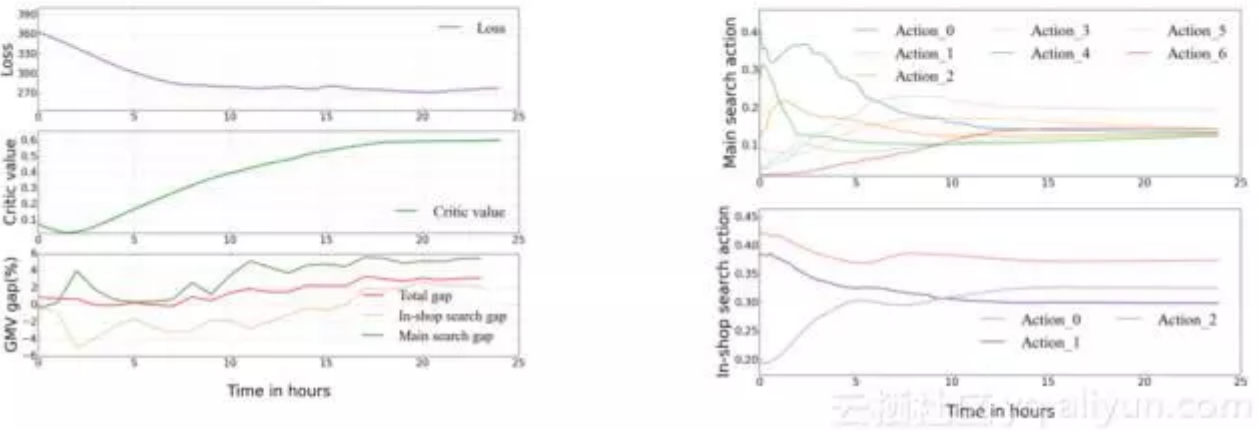
非合作竞争会带来负面影响，电商领域有很多类似问题，比如搜索和推荐都奔着GMV，会导致结果趋同，引发无效报关。对此，我们提出MA-RDPG模型，MA-DPG策略独立，评价统一。



实际应用架构如图，首先模型训练是在porsche中完成，搜集样本成立model，model会更新到在线服务，根据Actor场景产生排序policy再部署到我们的环境中，环境中的结果会反馈到porsche中，形成数据闭环。

day	EW + L2R			L2R + EW			L2R + L2R			MA-RDPG		
	main	in-shop	total	main	in-shop	total	main	in-shop	total	main	in-shop	total
1	0.04%	1.78%	0.58%	5.07%	-1.49%	3.04%	5.22%	0.78%	3.84%	5.37%	2.39%	4.45%
2	0.01%	1.98%	0.62%	4.96%	-0.86%	3.16%	4.82%	1.02%	3.64%	5.54%	2.53%	4.61%
3	0.08%	2.11%	0.71%	4.82%	-1.39%	2.89%	5.02%	0.89%	3.74%	5.29%	2.83%	4.53%
4	0.09%	1.89%	0.64%	5.12%	-1.07%	3.20%	5.19%	0.52%	3.74%	5.60%	2.67%	4.69%
5	-0.08%	2.24%	0.64%	4.88%	-1.15%	3.01%	4.77%	0.93%	3.58%	5.29%	2.50%	4.43%
6	0.14%	2.23%	0.79%	5.07%	-0.94%	3.21%	4.86%	0.82%	3.61%	5.59%	2.37%	4.59%
7	-0.06%	2.12%	0.62%	5.21%	-1.32%	3.19%	5.14%	1.16%	3.91%	5.30%	2.69%	4.49%
avg.	0.03%	2.05%	0.66%	5.02%	-1.17%	3.09%	5.00%	0.87%	3.72%	5.43%	2.57%	4.54%

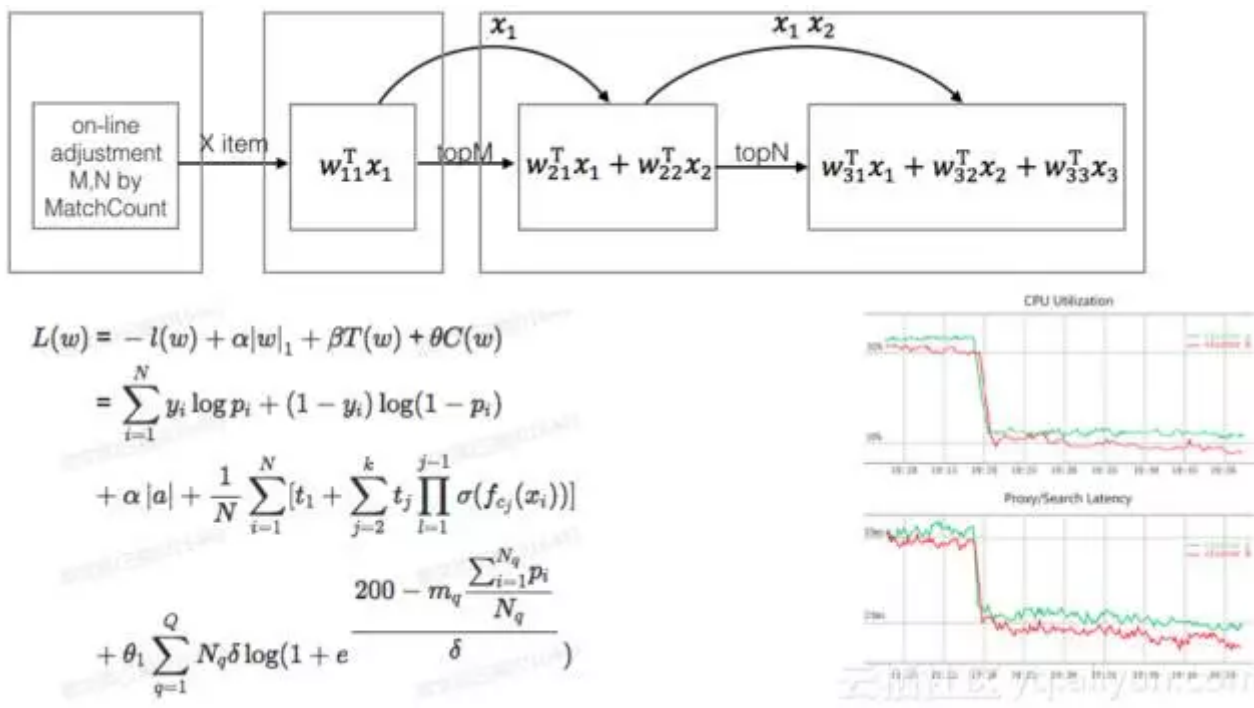
Gmv gaps, A/B test on taobao product search



实验结果如图，可以看到，我们的整体收益有明显提升。

性能优化

Cascade Learning



我们在排序时除了考虑模型军队外，还要考虑模型性能开销。我们会把每个特征性能的cascade带进来，样本进来后是否应该进入到下一轮排序，通过控制每一轮排序商品数量，达到整体模型精度。使用Cascade Learning使整体性能提升了30%。

除了业务角度的性能优化，针对深度学习我们也做了很多加速探索，包括降维、量化、剪枝和二值网络。

未来挑战仍有很多。包括在冷启动、多场景联合优化、多目标优化、超参数学习、多模交互和智能导购等多方面，仍有许多等待着我们去完成。

往期精彩文章

- 0. 盘点·GitHub最著名的20个Python机器学习项目
- 1. 机器学习必知的15大框架
- 2. 【逐云】阿里云高性能计算负责人何万青：做技术的一定要多跨界
- 3. 福利：热门技术看什么？这份书单告诉你！（内含PDF链接）
- 4. 基于TableStore构建简易海量Topic消息队列

-END-

云栖社区

ID: yunqiinsight

云计算 | 互联网架构 | 大数据 | 机器学习 | 运维