「AI大咖谈」阿里算法专家谈大规模推荐系统粗排层的设计与实现

王喆的笔记 王喆的机器学习笔记 5月11日

「AI大咖谈」阿里算法专家谈大规模推荐系统粗排层的设计与实现

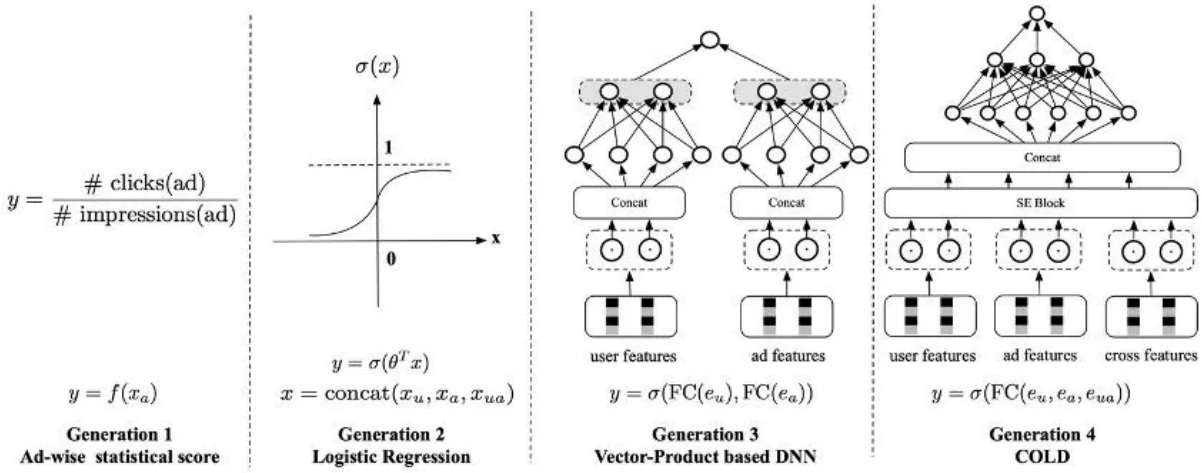
这里是「王喆的机器学习笔记」的第三十七篇文章。今天我们「AI大咖谈」邀请的大咖是阿里的算法专家王哲，所以今天是一次王喆对王哲的访谈。

王哲是上一届DLP-KDD workshop Best Paper Award的获得者，获奖paper COLD: Towards the Next Generation of Pre-Ranking System 深入探讨了阿里大规模推荐系统粗排层的设计和实现，是我非常推崇的业界实践文章。所以今天我们也围绕大规模推荐系统方向提出了十个问题，来看一下业界最前沿推荐系统的实践经验。

> 大咖简介
> 王哲，花名公渡，2017年中国科学技术大学计算机专业硕士毕业。曾在蚂蚁金服负责跨境游的推荐营销算法。目前为阿里妈妈展示广告团队算法专家，负责粗排及全链路联动的相关工作，有多篇顶会论文。
> 王哲同时也是一名知乎大V @萧瑟，对他的工作感兴趣的同学也可以直接跟他沟通

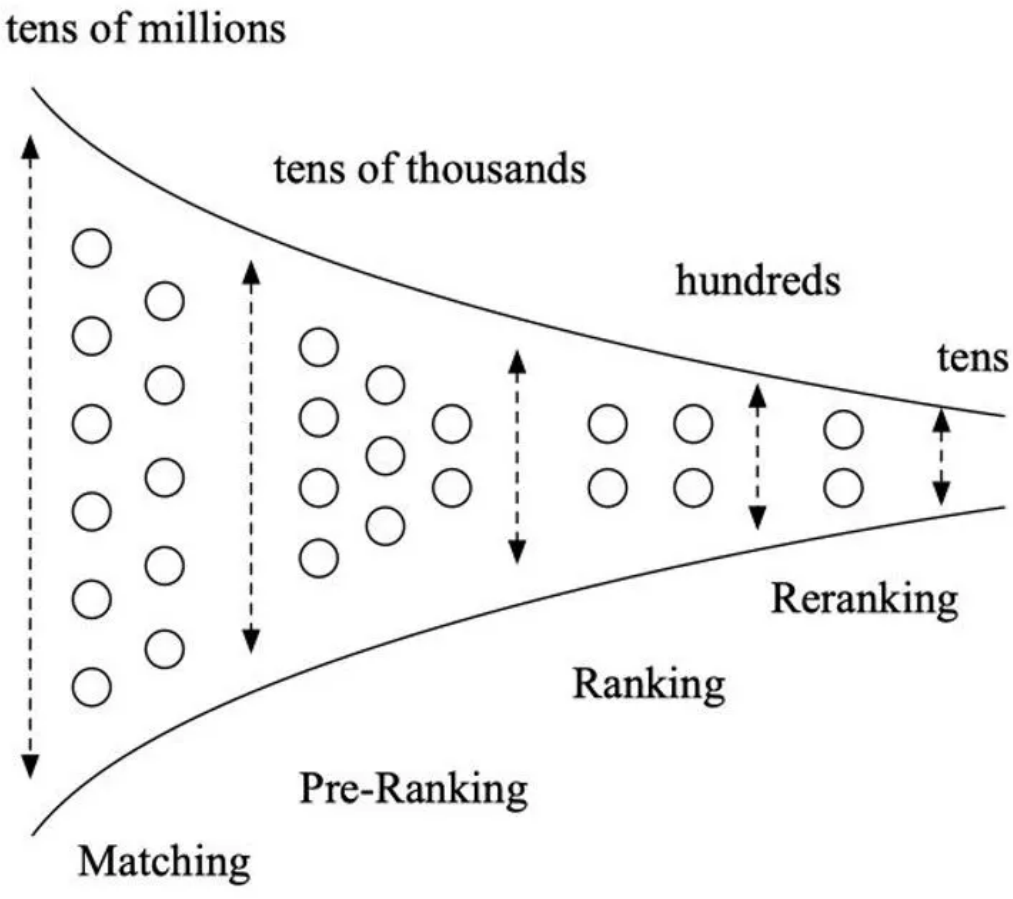
1. 去年DLP-KDD的best paper COLD的工作非常精彩，业界影响里非常大，能否简单的介绍一下它的主要思路？

COLD是一个非常典型的算法-系统Co-Design的工作。它没有限制模型结构，可以支持任意复杂的深度模型，COLD的网络结构是以拼接好的特征embedding作为输入，后面是7层全连接网络，包含交叉特征。整个系统是实时训练，实时打分，以应对线上分布的快速变化，对新广告冷启也更友好。当然，如果特征和模型过于复杂，算力和延时都会难以接受。因此我们一方面设计了一个灵活的网络架构可以进行效果和算力的平衡。另一方面进行了很多工程上的优化以节省算力。



粗排层在业界发展的主要阶段以及COLD的模型结构

2. 能否简单明了的介绍一下深度学习时代，召回、粗排和精排在阿里环境下的主要特点？



召回，粗排，精排，再排序层的结构

这里以阿里妈妈定向广告为例做一下介绍。

召回阶段一个主要特点是规模较大，阿里妈妈这边召回阶段的广告库规模在千万左右。另一个特点是召回的目标是选择符合后链路需要的集合，因此以多路召回方式为主，通过多种方式进行集合选择。不同路选择的广告分数往往不可比。

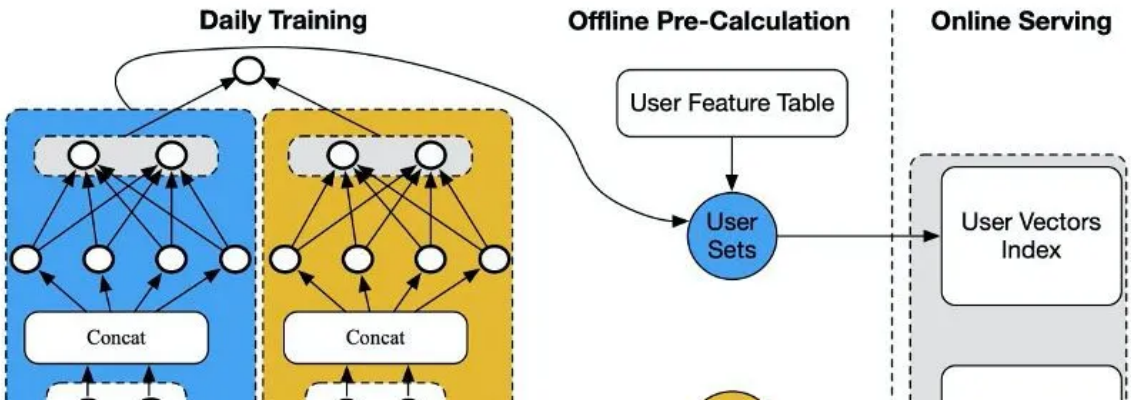
目前我们在召回使用了基于树结构的全库检索算法 TDM，向量近临检索算法以及基于用户行为触发的相似广告召回等。

粗排阶段是从上万个广告中选择上百个送给精排，实时性约束在十几 ms 以内，除了排序逻辑之外，也包含一些广告过滤逻辑。粗排是一个承上启下的过渡模块，既有召回的特点，可以采用多通道方式进行集合选择，只不过通道往往较少。也兼具精排统一排序的要求，以便对召回的多路集合用统一价值进行度量合并。粗排的天花板是精排，因此可以通过评估和精排的对齐程度来判断粗排的迭代空间。

精排阶段是非常复杂的深度模型，集中了较多的算力资源，延迟往往也较高，除了点击率 / 收藏加购率 / 成交率等多目标预估模型之外。后续还有调价模块基于广告主目标对价格进行调整以平衡广告主和平台收益，同时还有一些策略打散等重排逻辑。

如果把整个级联排序系统比做火车的话，精排就是火车头，是整个系统效果的天花板，是需要重兵投入的主战场。

3. 使用Embedding+简单运算（内积，简单网络）做快速召回/粗排的做法已经十分成熟，online inference的效率也很高，它能否代替COLD？COLD相比它的主要优势又在哪里？



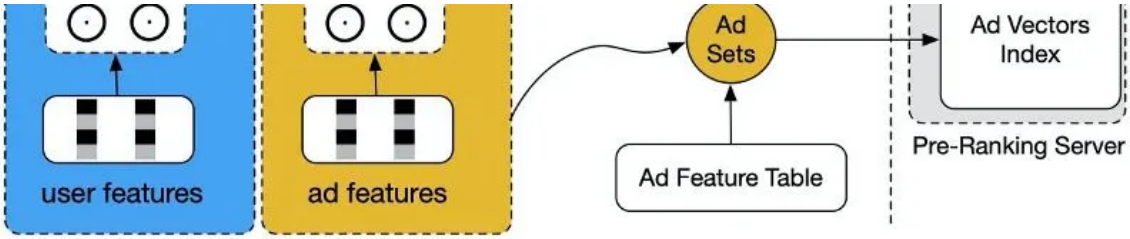


Figure 4: Infrastructure of pre-ranking system with vector-product based DNN model.

双塔模型的典型结构

使用Embedding+简单运算的方式虽然算力和RT（Reaction Time）消耗较低，但是在很多对模型效果和实时性要求较高的场景并不能完全代替COLD。COLD与之相比没有对模型结构进行限制，可以使用交叉特征和更复杂的网络结构，因此具有更强的拟合能力，同时可以对算力和效果进行灵活的平衡。COLD实时训练实时打分的架构可以更好的适应数据分布的快速变化，有利于快速迭代，在冷启动上也更为友好。

4. 现在业界有一个趋势是做分层的融合，比如粗排和精排的融合，甚至召回和排序的融合，你觉得这个趋势会不会发展下去，有哪些难点？

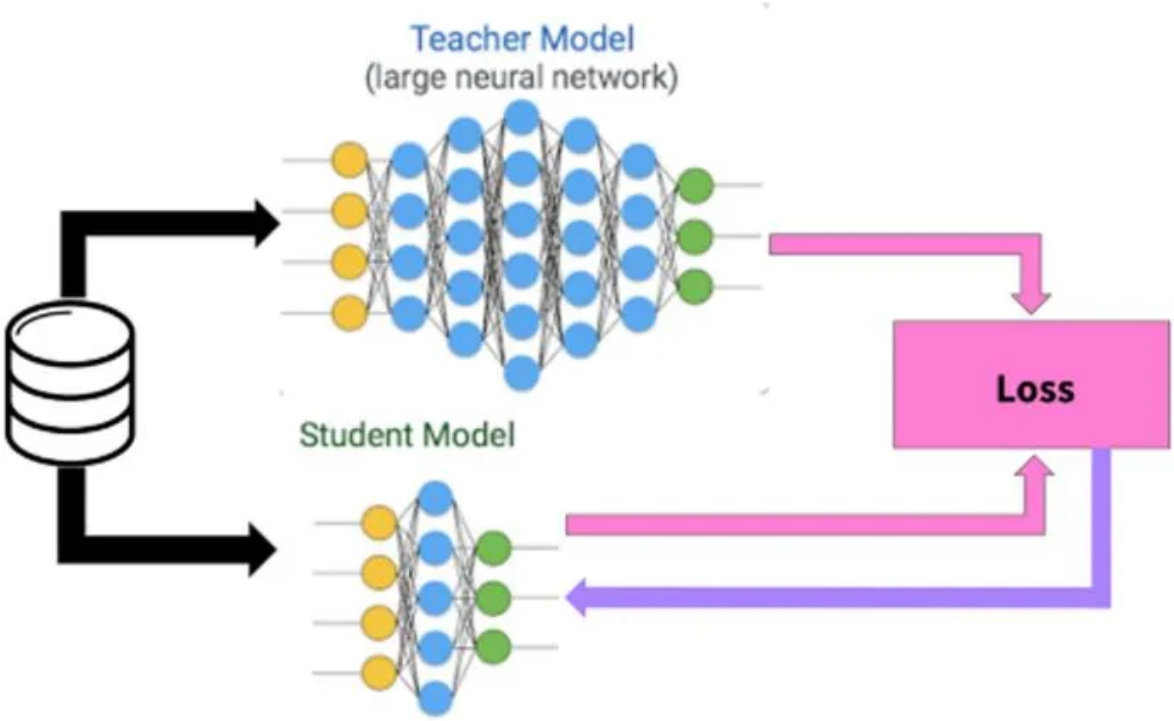
正所谓合久必分,分久必合。召回 - 粗排 - 精排这种级联排序架构，是当初算力 RT 不足情况下的一种折中。当前确实存在分层融合的趋势。例如在召回阶段，我们也在尝试用排序的方式，突破近似检索的瓶颈，直接进行全库打分。而粗排阶段也在探索粗排和精排的联合训练，在一次训练过程中同时产出多个不同结构的模型，粗排模型只是其中一个结构较为简化的版本。这个趋势一方面是因为深度学习时代算法技术的突破，使整个级联架构在模型结构上的统一成为了可能。

另一方面也要得益于 GPU/TPU/NPU 等硬件带来的算力红利释放。随着各模块技术水位不断增加，单点迭代的难度也越来越高，从整个级联排序架构的视角进行改进，进行模块间的融合，这个趋势未来会继续发展下去。而技术上的难点一方面在于样本选择偏差问题，即前链路（召回 / 粗排）的 inference 空间和训练空间存在较大差异，从而影响了融合的效果。另一方面则是在打分规模不断提升的情况下，如何控制算力和 RT 的增长。同时不同模块之间应该如何更好的交互，也值得进一步的研究。

5. 你如何对比COLD的工作和知识蒸馏的区别？

5. 你如何对比COLD的整体相对知识蒸馏的区别？

COLD是一种新的粗排实时排序架构，知识蒸馏是一种提升模型效果的技术手段，两者并不冲突。目前我们也在尝试，在当前COLD粗排模型的基础，通过和精排模型的联合训练以及知识蒸馏，取得了进一步的线上效果提升。



通过知识蒸馏缩小模型体积

6. COLD使用了online learning，能否介绍一下你们用到的具体方法？是在什么平台（TensorFlow？Flink？Server内部？）上进行的训练？

COLD使用的Online Learning技术，实时数据流是基于阿里妈妈自研的星云ODL系统，底层是Blink(Blink是阿里巴巴通过改进Flink项目而创建的阿里内部产品)。训练基于的是阿里妈妈内部自研的深度训练框架XDL。模型目前是实时训练，小时级更新（我们的业务场景小时级已经足够，整个系统可以支持更高的更新频率）。

阿里基于Flink开发自研数据流平台Blink

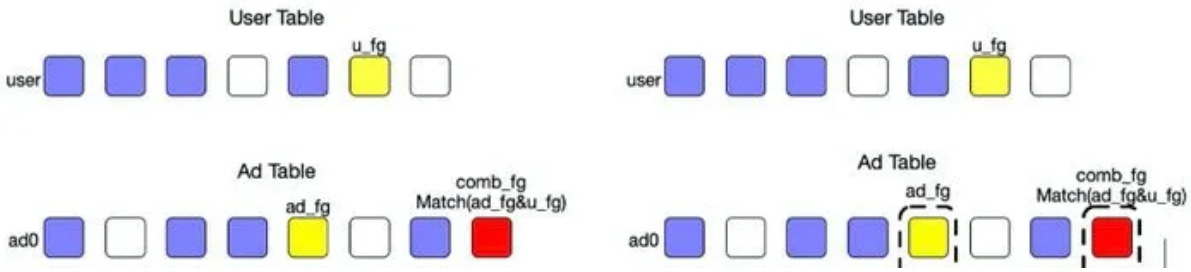
7. Model serving一直是业界的一个难点，COLD模型是如何部署到线上的？

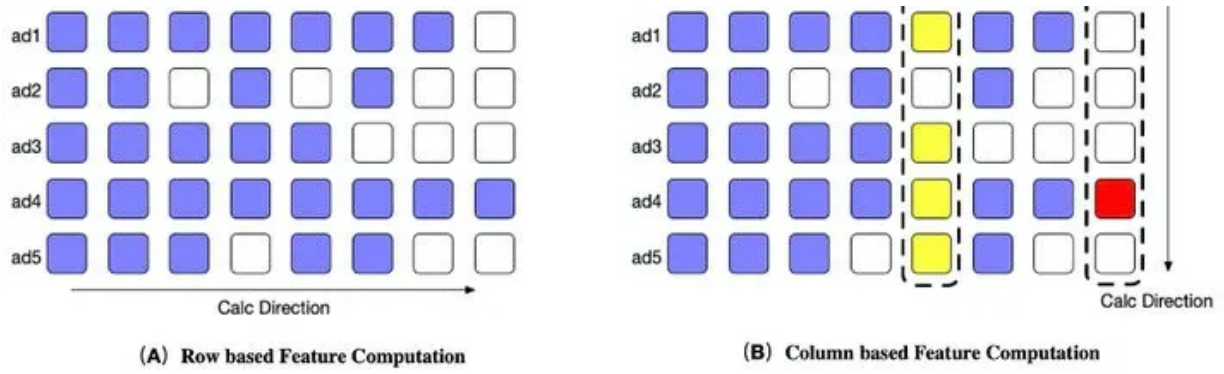
粗排的线上打分系统主要包含两部分：特征计算和网络计算。特征计算部分主要负责从索引中拉取用户和广告的特征并且进行交叉特征的相关计算。而网络计算部分，会将特征转成embedding向量，并将它们拼接进行网络计算。

为了将COLD部署到线上，工程上进行了很多优化，包括：

并行化：为了实现低时延高吞吐的目标，并行计算是非常重要的。而粗排对于不同的广告的计算是相互独立的，因此可以将计算分成并行的多个请求以同时进行计算，并在最后进行结果合并。特征计算部分使用了多线程方式进一步加速，网络计算部分使用了GPU以及和达摩院合作的NPU专用硬件。

行列转化：特征计算的过程可以抽象看做两个稀疏矩阵的计算，一个是用户矩阵，另一个是广告矩阵。矩阵的行是batch_size，对于用户矩阵来说batch_size为1，对于广告矩阵来说batch_size为广告数，矩阵的列是featue group的数目。常规计算广告矩阵的方法是逐个广告计算在不同feature group下特征的结果，这个方法符合通常的计算习惯，组合特征实现也比较简单，但是这种计算方式是访存不连续的，有冗余遍历、查找的问题。事实上，因为同一个feature group的计算方法相同，因此可以利用这个特性，将行计算重构成列计算，对同一列上的稀疏数据进行连续存储，之后利用MKL优化单特征计算，使用SIMD (Single Instruction Multiple Data)优化组合特征算子，以达到加速的目的。





行列转化示意图

Float16加速: 对于COLD来说，绝大部分网络计算都是矩阵乘法，而NVIDIA的Turning架构对Float16的矩阵乘法有额外的加速，因此我们将粗排模型做了Float16转化。使用Float16以后，CUDA kernel的运行性能有显著提升，同时kernel的启动时间成为了瓶颈。为了解决这个问题，我们使用了MPS (Multi-Process Service)来解决kernel启动的开销。Float16和MPS技术，可以带来接近2倍的QPS提升。

8. 现在大家越来越强调Algorithm-System Codesign，我想COLD应该是这方面非常成功的案例，能否介绍一下你们的经验？如何做到不同模型和工程团队/成员之间的良好配合？

在COLD之前，粗排和精排是两套独立的模型训练和线上打分的架构，迭代维护很不方便。COLD在架构上接耦了排序引擎和在线打分模块，统一了粗排和精排的ODL训练迭代体系和在线打分体系，一方面降低了迭代维护成本，另一方面也便于工程团队进行打分性能的专门优化。同时COLD的架构可以很灵活的对算力和效果进行平衡，不再受限于双塔结构。

团队成员配合上，首先我们团队很早就意识到了Algorithm-System Co-design的重要性，因此专门成立了效能团队，作为联接算法和工程的桥梁，以算力优化和迭代效率作为切入点，发挥了很重要的作用。算法团队，效能团队和工程团队相互配合，工作上各有侧重。算法团队主要负责模型的迭代维护和效果上的持续提升。效能团队会在算法迭代的早期就介入，进行算力和模型的平衡。同时效能团队也通过改进优化算法的迭代链路（特征加工/ODL流/训练框架等）来帮助提升迭代效率。工程团队会更侧重于线上引擎以及打分模块的性能优化和功能需求支持。通过这种合作模式保证了COLD的上线。

9. 就你个人而言，有没有遇到模型/系统改进的瓶颈期，如何破局？站在2021年，你如何预测未来算法发展的红利在哪？

有遇到过，这里还是以COLD作为例子。在COLD上线之后，粗排进一步的迭代到了瓶颈期，很多模型优化都难以取得进一步的效果提升。我破局的思路就是跳出粗排这个单一模块，站在整个排序链路的视角重新思考问题。通过在离线分析发现粗排和精排的对齐程度已经很高，粗排进一步迭代空间有限。因此需要从前链路的召回入手，通过提升召回进入粗排的广告数目和质量，来进一步打开粗排的迭代空间。因此我在召回从全链路目标对齐的视角出发，做了一些技术上的创新突破，构建了一些对齐后链路目标的召回通道，不仅打开了召回的迭代空间，也打开了粗排的空间。后面又回到粗排，一方面继续从全链路目标对齐的视角出发，对粗排做进一步的技术升级。同时也围绕样本选择偏差问题，把召回迭代过程中积累的技术经验迁移到粗排，取得了很好的线上效果。

至于未来算法发展的红利，这是个挺大也挺难回答的问题。这里我斗胆发表一点自己的看法。未来在下面几个方向，有可能存在一些红利：

1. 端智能：用户移动端设备的性能越来越强，这里潜藏着庞大的算力资源有待挖掘利用。同时端上有用户更实时更丰富的行为信息，对提升模型预估精度也有很大帮助。端上和服务端如何能更好的配合协同是非常值得研究的问题。

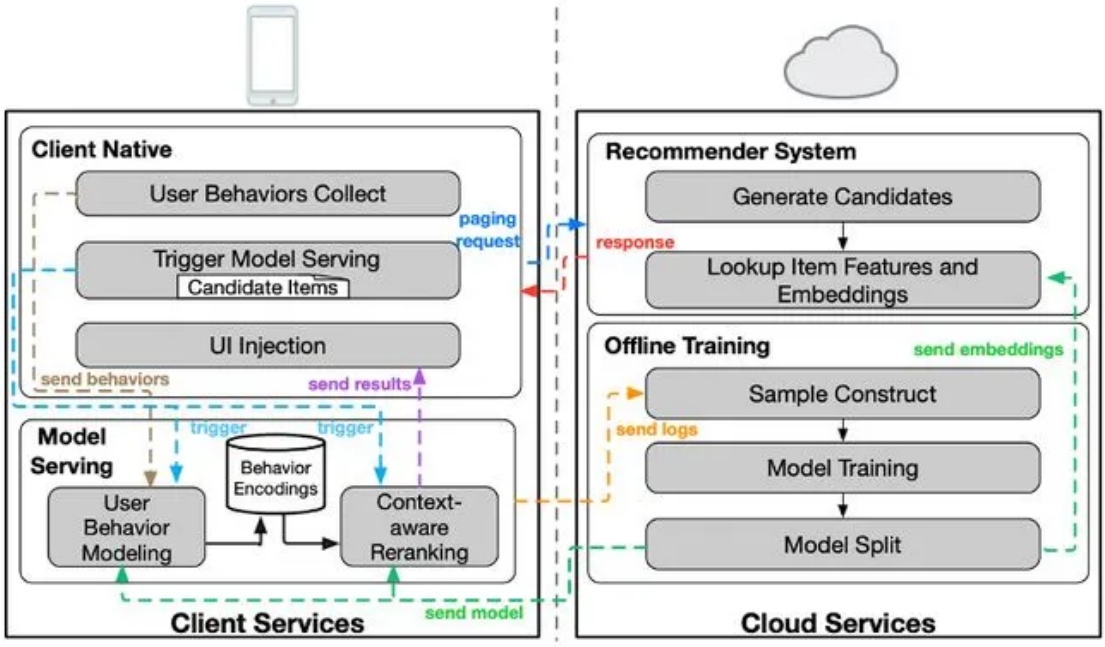


Figure 2: EdgeRec system overview. The left part of modules are deployed in mobile Taobao client and the right part of modules are serving on cloud.

端智能/边缘计算经典案例——EdgeRec

2. 排序架构升级：目前的级联排序架构，各模块之间一般独立迭代，在对齐最终目标的过程中很容易因为各模块的差异造成链路损耗影响最终效果。如何能更好地进行优化从而实现全链路的目标对齐，如何突破级联排序架构构建一个更优的排序架构体系是很值得探索的。

3. 算力的全局最优化分配：之前大家对算力的优化往往集中在单点。如果能真正把算力作为一个变量，在整个系统链路进行全局最优分配，是有可能进一步释放一部分算力空间的。

（推荐参考论文 DCAF: A Dynamic Computation Allocation Framework for Online Serving System ）

10. 给刚入行的同学说两句话吧，有哪些成功的经验和失败的教训可以对他们讲？

算法工程师的核心竞争力不仅仅在于对模型的理解，更关键的在于对业务的深入理解，在于能否帮助业务解决实际问题。有时候如果能跳出对模型等细节的关注，站在全局的视角，不仅有助于解决问题，也可以发现新的机会。就像COLD之后，我面临粗排后面如何进一步迭代的问题，在粗排和精排模型对齐程度已经很高的情况下，继续想办法去优化粗排模型就会面临很大困难，但是跳出粗排从整个链路的视角看待问题，就柳暗花明又一村。

结束跟王哲的谈话之后，我有两点感触是最强烈的，与你分享一下：

1.在业界经历了从简单的“排序层”到“召回-粗排-精排-重排”的越来越精细化的分层之后，随着一线公司对于算力的运用分配更加纯熟，对于深度学习架构的不断改进，业界有可能已经走上了“召回-粗排-精排层”的合并之路。正如王哲所说，天下大势，合久必分，分久必合。当初的分层拆分是由于算力限制，延迟限制的无奈之举，今天的合并也是由于我们冲破了这些限制，进而融合各层。
2.算法工程师的破局之路，离不开对于整体系统的理解。只有心中有Full Picture，我们才能够做到有利于全局的改进。在COLD的工作中，对于粗排层关键性的改进，也离不开对于整个推荐链路的理解，这是我们所有人值得借鉴的。

最后，再与大家分享一下2021年DLP-KDD的征稿信息。作为走出了COLD，DCAF，Res-Embedding等多个业界影响力非常大的工作的workshop，我们欢迎更多推荐、广告和搜索方向相关的同学加入进来，参与成果的分享。

> DLP-KDD Workshop介绍：在国际顶级会议KDD召开之际，在国际顶级会议KDD召开之际，来自阿里巴巴/微软/华为/Roku，以及上海交通大学/犹他大学等工业界/学术界资深同行，携手举办全球第三届面向高维稀疏数据的深度学习实践国际研讨会（The 3rd International Workshop on Deep Learning Practice for High-Dimensional Sparse Data with KDD 2021，简称DLP-KDD 2021），在此诚挚邀请学术界及工业界供稿。

2021年DLP-KDD的征稿结束日期是2021年5月20日，详细投稿信息请在阅读原文中查看。

最后欢迎大家关注我的微信公众号：王喆的机器学习笔记（wangzhenotes），跟踪计算广告、推荐系统等机器学习领域前沿。想进一步交流的同学也可以通过公众号加我的微信一同探讨技术问题。

王喆的机器学习笔记

推荐系统，计算广告，机器学习领域前沿进展

27篇原创内容

公众号

—END—

阅读原文

喜欢此内容的人还喜欢