

推荐系统技术演进趋势：重排篇

张俊林 炼丹笔记 1周前

↑↑↑关注后"星标"炼丹笔记

炼丹笔记干货

作者：张俊林

新浪微博 AI Lab 负责人

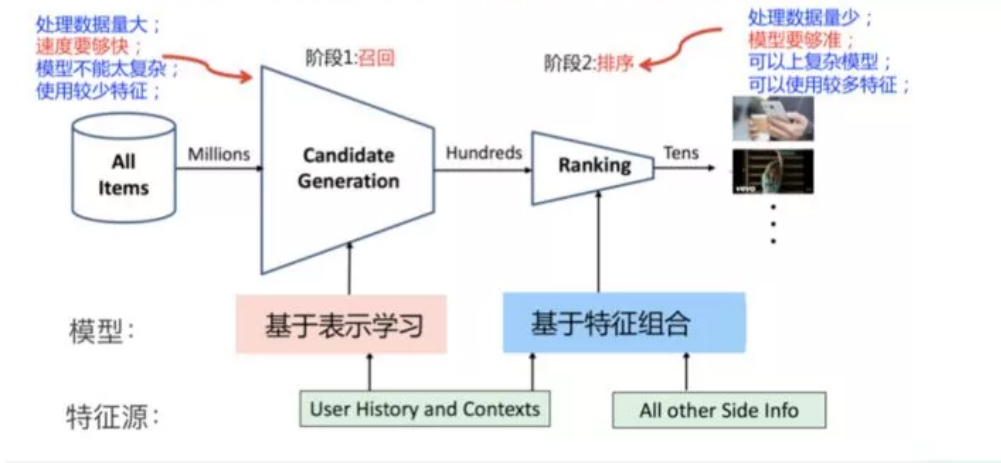
《推荐系统技术演进趋势》从召回篇、排序篇、重排篇依次更新，本文为**重排篇**。错过《**推荐系统技术演进趋势：召回篇**》和《**推荐系统技术演进趋势：排序篇**》的小伙伴可以点击链接跳转阅读。

背景

推荐系统技术，总体而言，与NLP和图像领域比，发展速度不算太快。不过最近两年，由于深度学习等一些新技术的引入，总体还是表现出了一些比较明显的技术发展趋势。这篇文章试图从推荐系统几个环节，以及不同的技术角度，来对目前推荐技术的比较彰显的技术趋势做个归纳。个人判断较多，偏颇难免，所以还请谨慎参考。

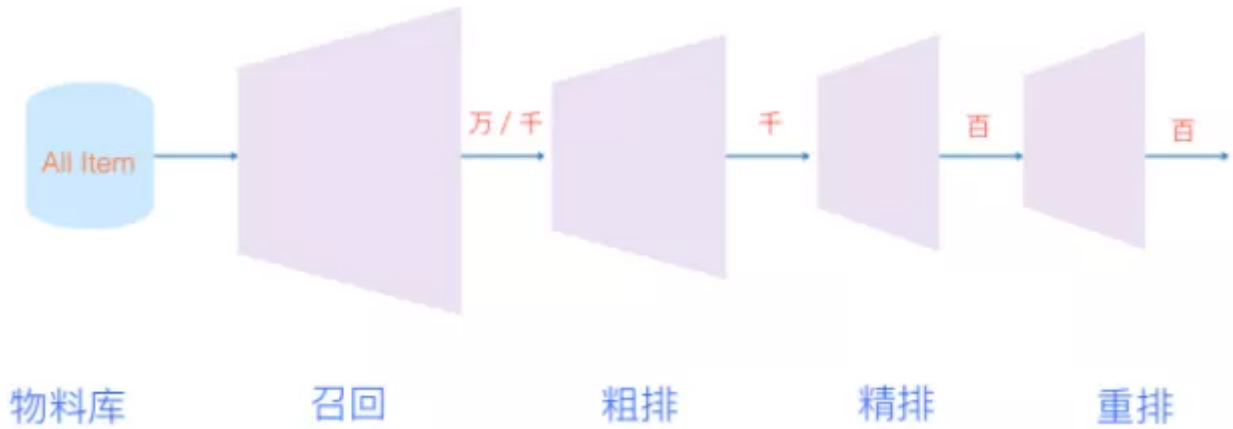
在写技术趋势前，照例还是对推荐系统的宏观架构做个简单说明，以免读者迷失在技术细节中。

推荐系统的两个阶段：召回+排序



实际的工业推荐系统，如果粗分的化，经常讲的有两个阶段。首先是召回，主要根据用户部分特征，从海量的物品库里，快速找回一小部分用户潜在感兴趣的物品，然后交给排序环节，排序环节可以融入较多特征，使用复杂模型，来精准地做个性化推荐。召回强调快，排序强调准。当然，这是传统角度看推荐这个事情。

但是，如果我们更细致地看实用的推荐系统，一般会有四个环节，如下图所示：



四个环节分别是：**召回、粗排、精排和重排**。

召回目的如上所述；有时候因为每个用户召回环节返回的物品数量还是太多，怕排序环节速度跟不上，所以可以在召回和精排之间加入一个粗排环节，通过少量用户和物品特征，简单模型，来对召回的结果进行个粗略的排序，在保证一定精准的前提下，进一步减少往后传送的物品数量，粗排往往是可选的，可用可不同，跟场景有关。

之后，是精排环节，使用你能想到的任何特征，可以上你能承受速度极限的复杂模型，尽量精准地对物品进行个性化排序。排序完成后，传给重排环节，传统地看，这里往往会上各种技术及业务策略，比如去已读、去重、打散、多样性保证、固定类型物品插入等等，主要是技术产品策略主导或者为了改进用户体验的。

那么，每个环节，从技术发展的角度看，都各自有怎样的发展趋势呢？下面我们分头说明。

重排技术演进趋势

在重排环节，常规的做法，这里是个策略出没之地，就是**集中了各种业务和技术策略**。比如为了更好的推荐体验，这里会加入去除重复、结果打散增加推荐结果的多样性、强插某种类型的推荐结果等等不同类型的策略。

按理说，这块没什么可讲的。但是，如果从技术发展趋势角度看，重排阶段上模型，来代替各种花样的业务策略，是个总体的大趋势。

→ List Wise重排序 ←

关于List Wise排序，可以从两个角度来说：

- 一个是**优化目标或损失函数**；
- 一个是**推荐模块的模型结构**。

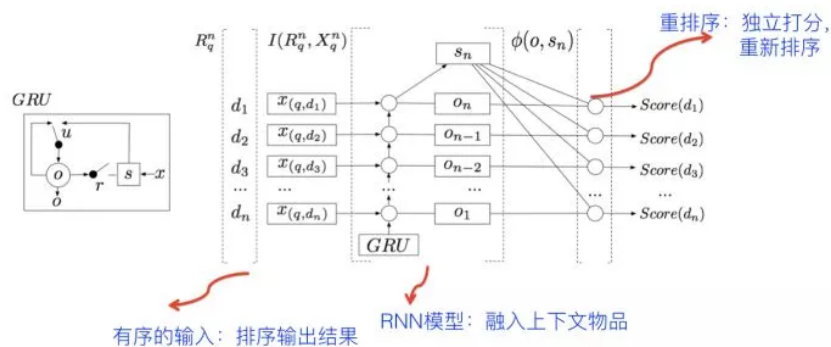
推荐系统里Learning to Rank做排序，我们知道常见的有三种优化目标：**Point Wise、Pair Wise和List Wise**。所以我们首先应该明确的一点是：List Wise它不是指的具体的某个或者某类模型，而是指的模型的优化目标或者损失函数定义方式，理论上各种不用的模型都可以使用List Wise损失来进行模型训练。

最简单的损失函数定义是**Point Wise**，就是输入用户特征和单个物品特征，对这个物品进行打分，物品之间的排序，就是谁应该在谁前面，不用考虑。明显这种方式无论是训练还是在线推理，都非常简单直接效率高，但是它的缺点是没有考虑物品直接的关联，而这在排序中其实是有用的。

Pair Wise损失在训练模型时，直接用两个物品的顺序关系来训练模型，就是说**优化目标是物品A排序要高于物品B**，类似这种优化目标。其实Pair Wise的Loss在推荐领域已经被非常广泛得使用，比如BPR损失，就是典型且非常有效的Pair Wise的Loss Function，经常被使用，尤其在隐式反馈中，是非常有效的优化目标。

List Wise的Loss**更关注整个列表中物品顺序关系**，会从列表整体中物品顺序的角度考虑，来优化模型。在推荐中，List Wise损失函数因为训练数据的制作难，训练速度慢，在线推理速度慢等多种原因，尽管用的还比较少，但是因为更注重排序结果整体的最优性，所以也是目前很多推荐系统正在做的事情。

ReRanker发展趋势：List Wise



因为输入有序，往往采用序列模型：RNN / Transformer etc.

从模型结构上来看。因为重排序模块往往是放在精排模块之后，而精排已经对推荐物品做了比较准确的打分，所以往往重排模块的输入是精排模块的Top得分输出结果，也就是说，是有序的。而精排模块的打分或者排序对于重排模块来说，是非常重要的参考信息。于是，这个排序模块的输出顺序就比较重要，而能够考虑到输入的序列性的模型，自然就是重排模型的首选。

我们知道，最常见的考虑时序性的模型是RNN和Transformer，所以经常把这两类模型用在重排模块，这是很自然的事情。

一般的做法是：排序Top结果的物品有序，作为RNN或者Transformer的输入，RNN或者Transformer明显可以考虑在特征级别，融合当前物品上下文，也就是排序列表中其它物品，的特征，来从列表整体评估效果。RNN或者Transformer每个输入对应位置经过特征融合，再次输出预测得分，按照新预测的得分重新对物品排序，就完成了融合上下文信息，进行重新排序的目的。

尽管目前还没看到CNN做重排的方法，但是从机制上来说，明显CNN也是比较适合用来做重排环节模型的，感兴趣的同学可以试一试。当然，前面说的强化学习，也是非常适合用在List Wise优化的，目前也有不少相关工作出现。

典型工作

1. Personalized Re-ranking for Recommendation
2. Learning a Deep Listwise Context Model for Ranking Refinement

参考资料

- <https://zhuanlan.zhihu.com/p/100019681>