

如何解决推荐系统中的Embedding冷启动问题？

机器学习与推荐算法 2月24日

以下文章来源于王喆的机器学习笔记，作者王喆的笔记



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

如何解决深度推荐系统中的Embedding冷启动问题？

今天我们聊一聊Embedding的冷启动问题。

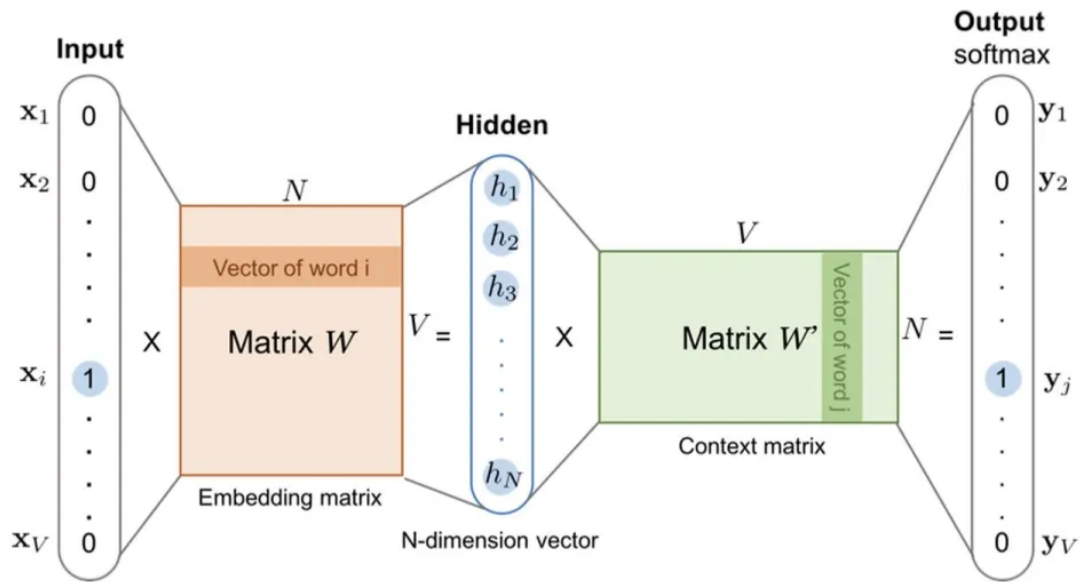
时至今日，深度学习的经典知识几乎已经是“显学”了，但是在实现深度学习推荐系统的过程中，还是充满了无数的细节和坑。所以接下来几篇文章会专门跟大家总结讨论课程中大家问题最多的，最感兴趣的话题。

今天的内容，就是“[如何解决Embedding的冷启动问题](#)”。我花了四五节的时间跟大家讨论word2vec，deepwalk，node2vec，GNN等Embedding知识，但大家最感兴趣的却是Embedding的冷启动。可见，这个问题在实践中处于一种什么样的地位。



Embedding冷启动问题出现的根源

在着手解决它之前，必须要搞清楚这个问题出现的根源在哪，为什么Embedding冷启动问题那么不好解决。我们以最简单的Word2vec为例（其他所有Embedding方法，不管多复杂，都遵循同样的原则），训练它的最终目的是要得到与onehot输入对应的向量，用这个Embedding向量来表示一个用户，或者一个物品，或者一个特定的特征。



Word2vec的模型结构图

为了生成这样一个Embedding向量，我们就必须完成整个神经网络的训练，拿上面的Word2vec的结构图来说，你必须在Embedding matrix W 训练完毕、收敛之后，才能够提取对应的Embedding。

这个时候冷启动的问题就来了，如果在模型训练完毕之后，又来了一个新的user或者item。怎么办？

要想得到新的Embedding，就必须把这个新的user/item加到网络中去，这就意味着你要更改输入向量的维度，这进一步意味着你要重新训练整个神经网络，但是，由于Embedding层的训练往往是整个网络中参数最大，速度最慢的，整个训练过程持续几个小时是非常常见的。这个期间，肯定又有新的item产生，难道整个过程就成一个死局了吗？这个所谓的“死局”就是棘手的Embedding的冷启动问题。



入手解决问题

清楚了问题的根源，我们开始入手分析和解决问题。从整个深度学习推荐系统的框架角度解决这个问题，我觉得可以从四个角度考虑：

- 1.信息 and 模型
- 2.补充机制
- 3.工程框架

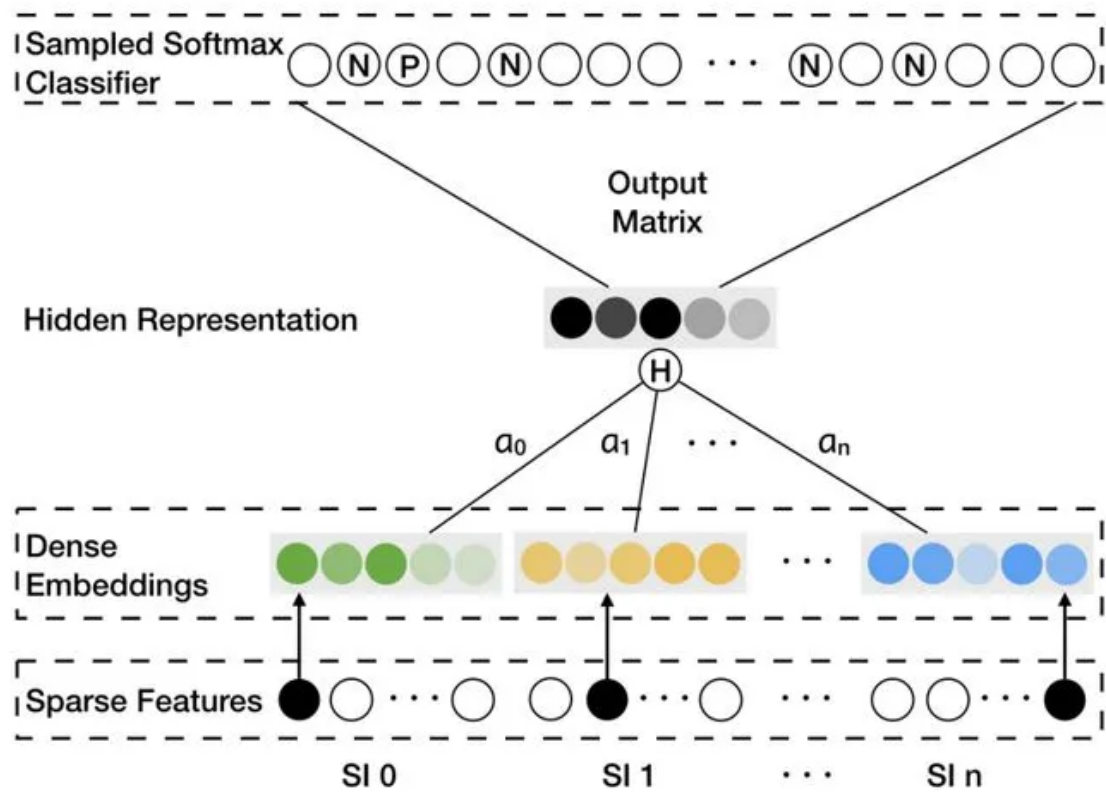
4.跳出固有思维

1、补充Side Information

第一个解决问题的思路我称为“大事化小”。对经典Embedding方法熟悉的同学一定知道，大多数Embedding方法是建立在用户的行为序列数据基础上的，用户的历史行为越多，训练出的Embedding越准确。但是，用户的历史行为并不是非常容易产生的，特别是购买、完成观看这类高质量的正向行为。也正因如此，如果仅基于历史行为来生成Embedding，必然会造成其覆盖率低下。

为了尽可能为更多的item和user生成Embedding，把Embedding冷启动这件事情“大事化小”。我们必须加入一些其他类型的特征，典型的用户侧特征是人口属性特征，典型的物品侧特征是一些内容型特征，我们一般统称side information。

其实，理论上来说，任何能够混合不同特征的深度学习模型都可以通过引入side information生成Embedding。我们这里还是举大家比较熟知的例子，淘宝团队的EGES模型（论文地址：<https://arxiv.org/pdf/1803.02349.pdf>）。

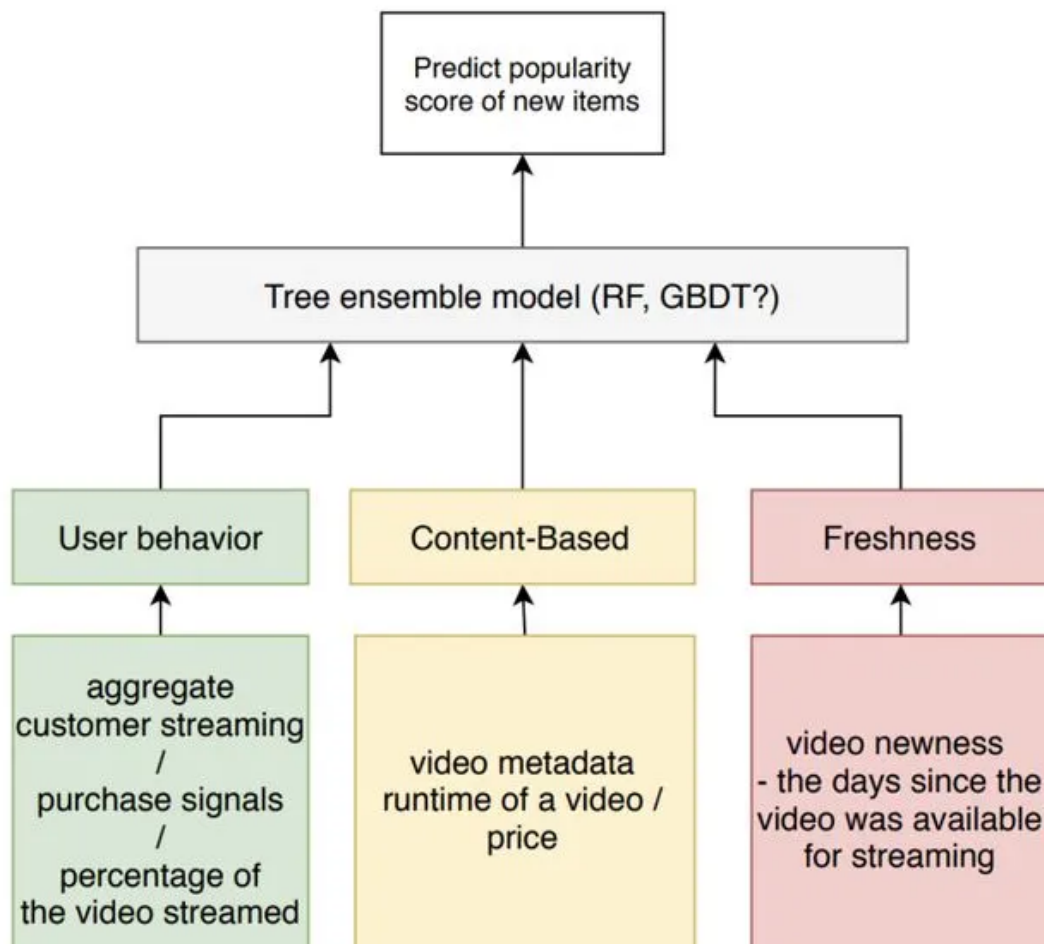


阿里的EGES模型

可以看到图中彩色的部分的Dense Embedding就是不同的特征域对应的Embedding层，每一个Embedding对应着一类特征，SI0就是Item id本身，SI1-SIn就是n个所谓的Side Information，比如Item的类别，价格，标签等等。这些Embedding经过一个加权平均的过程生成这个Item最终的Embedding。所以当有一个Item没有历史行为信息的时候，也就是说没有SI0时，还可以通过其他特征生成其Embedding，这就大大提高了Embedding的覆盖率。（[这里留一个问题给大家讨论，你觉得EGES生成Embedding的方式和Word2vec一样吗？我们应该从哪里拿到每个Item的Embedding？](#)）

而且我们要清楚的是，当不同特征的Embedding生成好之后，完全可以把组装生成Item Embedding的过程转移到线上，这样就能够彻底解决冷启动的问题，因为只要新的item生成，我们能够拿到它的特征，就能够组装出它的Embedding。

当然，[解决冷启动问题也没必要总是执着于从Embedding的角度解决](#)，因为Embedding也是作为一类特征输入到主推荐模型，或者主CTR预估模型之中的。既然这样，我们也完全可以直接把物品或者用户特征输入到主模型之中，只要这个主模型能够进行online inference，照样可以解决冷启动问题。



Amazon Video团队的推荐模型结构

Recsys2020上，Amazon Video团队就分享过类似的做法（论文地址：

<https://dl.acm.org/doi/10.1145/3383313.3411555>）。他们为了解决新加入视频的冷启动问题，把视频的一些metadata，流行度（popularity），新鲜度等一些列特征输入预测模型之中，即使没有模型提前生成好的Embedding，也照样可以做出“比较靠谱”的预测。

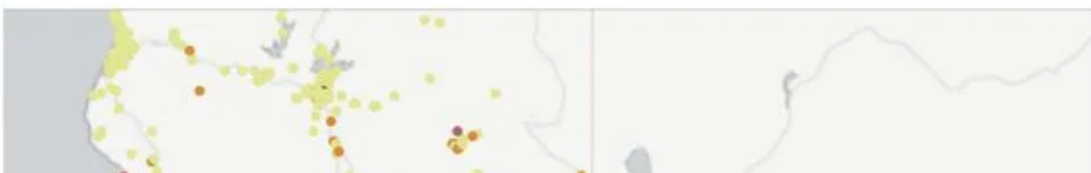
所以从“信息和模型”这个角度来解决这个问题，大致思路就是“聊胜于无”，整理好冷启动过程中你能获取到的可用的用户或者物品特征，把他们整合进模型，虽然刚开始可能不甚准确，但他们已经是你做出的最好的“菜”了，好过在一张白纸上预测。

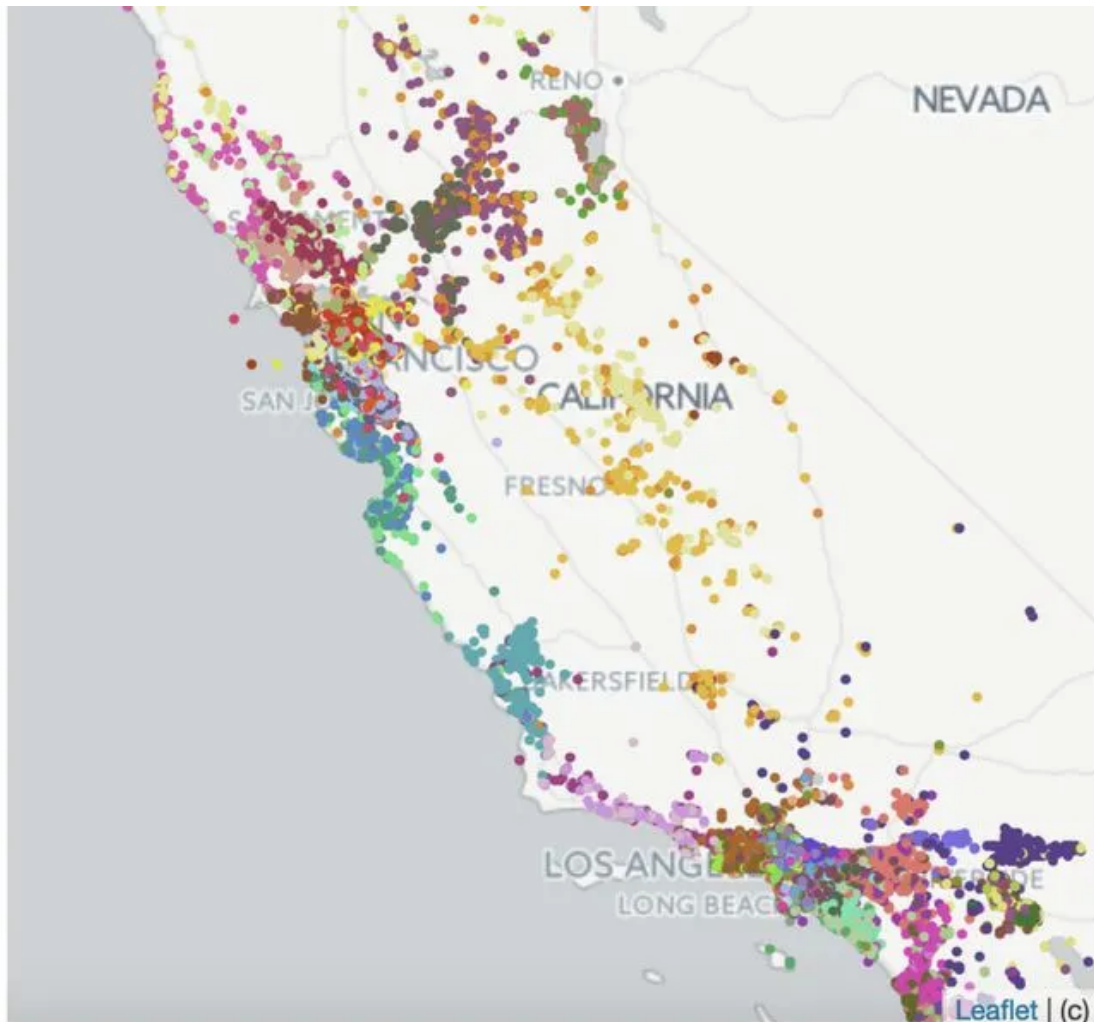
2、灵活的冷启动机制往往事半功倍

第二个解决问题的角度是“补充机制”。我们做模型的同学都有一个终极的梦想，“打造一个完美的End2End的模型，一个模型干净利索的解决所有问题”。这个想法当然是值得追求的，但是我们也不必落入“完美主义”的怪圈，自己给自己套上枷锁，在工业界的工作中，我们首要追求的还是整个推荐系统的效果。在主模型之上，适当的融入一些灵活的推荐机制，往往会取得事半功倍的效果。

大家如果有印象的话，在Airbnb那篇经典的Embedding文章中，所采用的冷启动机制就很实用。（论文地址：<https://www.kdd.org/kdd2018/accepted-papers/view/real-time-personalization-using-embeddings-for-search-ranking-at-airbnb>）

他们是怎么做的呢？首先肯定是要为大多数已有短租屋生成Embedding，那么在新的短租屋上架之后，会按照新的短租屋和已有短租屋的属性（房屋类型，价格，房间数等等）和地理位置距离，找到三个相似的有Embedding的短租屋，然后取其Embedding的平均。多么简单使用的冷启动策略。





AirBnb生成的出租屋聚类结果

当然，在冷启动机制的选取过程中，我们不应该忘记传统的机器学习模型，传统手艺不能丢啊。比如利用聚类快速定位新物品所在的cluster，找到相似物品。

再比如根据用户/物品的特征训练一颗决策树，再把冷启动的用户/物品根据有限的信息分配到决策树的某个分支中去，再根据分支对应的默认列表进行推荐等等。

我们在解决问题的时候还是遵循工程师工作的基本原则“在客观条件的制约下寻求最佳方案”，尽量利用好所有可用的信息和技术手段，寻求客观制约下的最优解。

3、推荐系统工程框架的改进

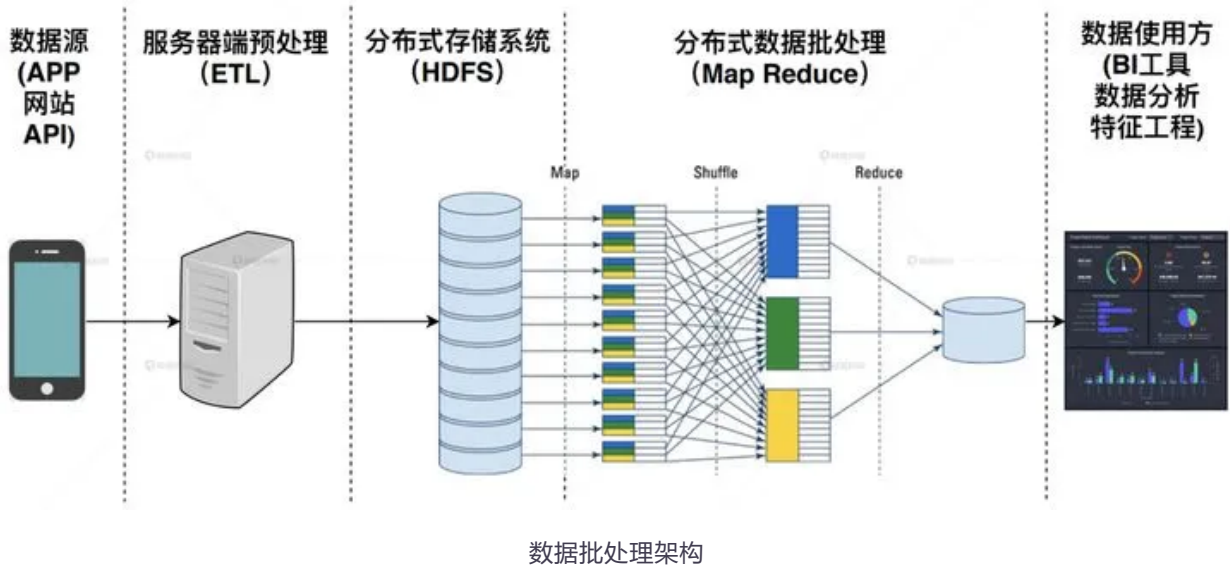
下面一个角度我想谈一谈通过“推荐系统工程架构上的改进”来解决冷启动问题。或者从更高的层面来说，冷启动的问题其实有一半是系统实时性的问题。

我们试想一个新闻APP新用户的交互过程，最开始用户的第一次登陆，这个新用户确实是“白板”一个。我们不得不使用我们刚才介绍的几种方法去做一些“聊胜于无”的推荐。但是一

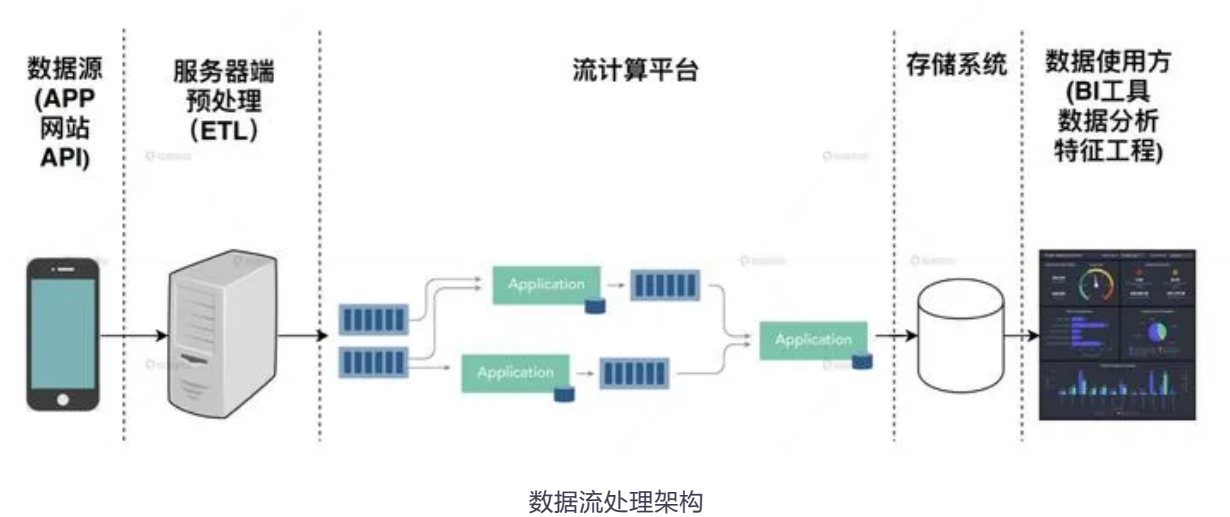
但如果你这个用户点击了第一条新闻，剩下的事情就是“速度”的比拼了。你的系统能不能实时的捕捉到这最新的珍贵的信号，决定了你能不能快速的让这个用户渡过冷启动的阶段。

旦这个用户点击过了第一条新闻，剩下的事情就是“速度”的比拼了。你的系统能不能实时的捕捉到这最新的珍贵的信号，决定了你能不能快速的让这个用户渡过冷启动的阶段。

有的同学说了，我们的系统是建立在批处理架构上的，每天训练一次，生成一次User Embedding，那么好，你的系统延迟是一天，这个用户明天才能看到新的推荐内容。

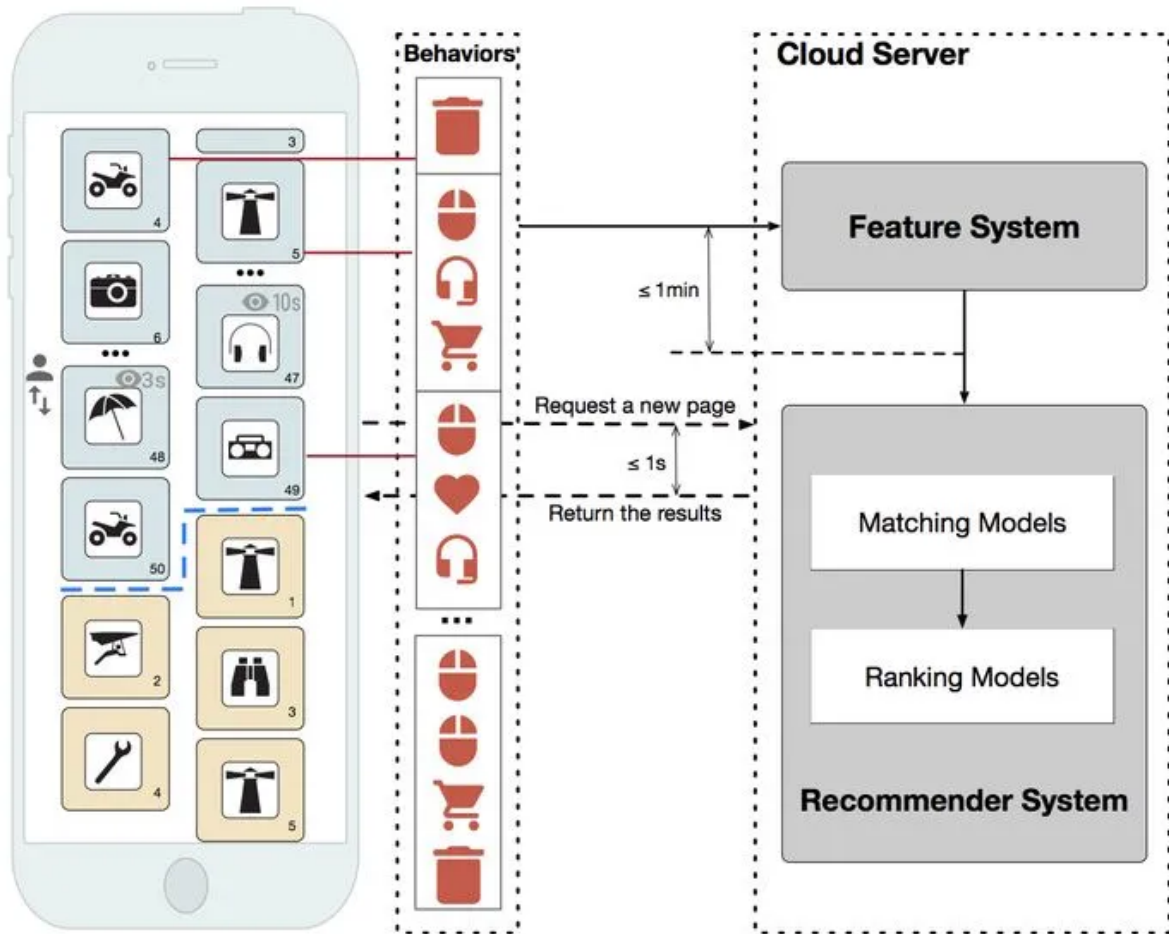


有的同学说了，我们的系统是基于流处理的，流处理的窗口大小5分钟。只要流处理平台看到了用户产生了正样本行为，就马上更新用户的embedding。很好，你的系统延迟是5-10分钟，用户大概率会在下次打开你的APP时看到更新。



有的同学还说了，我们的系统是实时的！用户产生新的行为之后，我们会在下次请求中发送这个行为记录到推荐服务器端，推荐模型直接把这个行为当新的特征输入，做实时推断，推荐结果实时更新，非常好，你的系统几乎就是实时的。

有的同学更牛，说我们的系统把推荐系统模型/策略做到了APP里面，就在客户端，就根据用户的新行为在设备内部做推荐，做结果的reranking，特别好！可以把“几乎”这两个字扔掉了，你的系统就是真正的实时推荐系统。



EdgeRec 边缘计算推荐系统

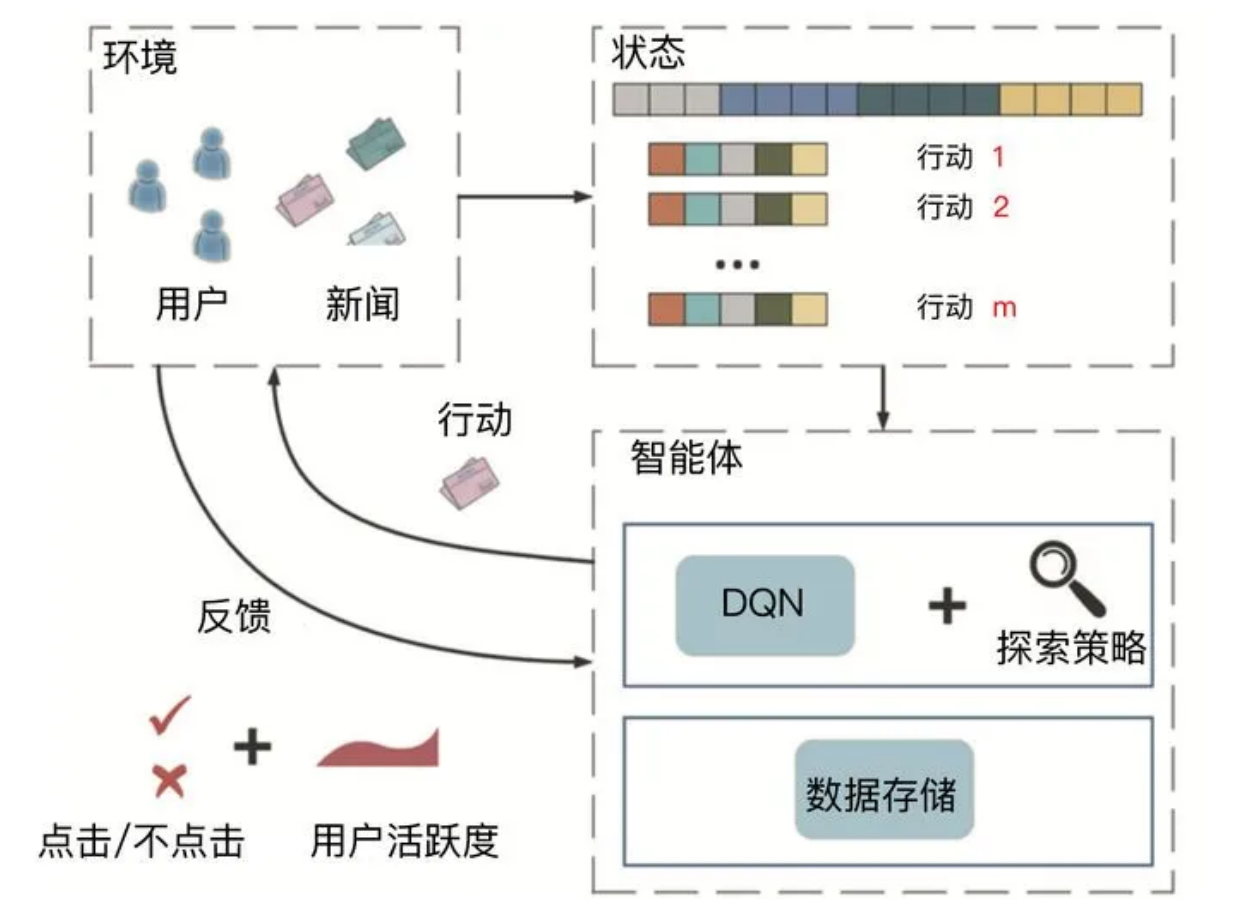
论文地址：EdgeRec: Recommender System on Edge in Mobile Taobao
<https://arxiv.org/pdf/2005.08416.pdf>

上面每一种做法，都有大量公司在使用，毫无疑问，最后一种的实时性最强，Taobao的推荐团队也已经在EdgeRec论文中披露了这一做法。那么这样基于“边缘计算”的推荐系统，无论在解决用户冷启动，还是物品冷启动，都可以实时处理新的信号，帮助用户或者物品以最快的速度渡过冷启动阶段。

4、跳出固有思维

有的同学可能还会说，我们团队实在是没有任何冷启动的数据可以用，难道就只能瞎猜吗？

你别说，瞎猜确实也是一种冷启动的手段，在探索与利用的诸多方法之中，epsilon greedy就是把一部分流量用于瞎猜，让冷启动的物品得到更多的展示，并快速收集数据。除此之外类似的机制还有主动学习，在线学习，强化学习等等，这里我就不展开讲了，但这些机制的原理是一致的，就是做到快速学习，主动探索，快速收集冷启动物品或者用户的数据，并且快速反馈到模型和特征中去。



强化学习中的反馈实时学习，实时调整推荐模型

但我这里主要想说的还不是这个。我想说的是，我们做推荐系统，有时候要跳出技术的固有思维，到更广阔空间去寻求团队合作。作为很多公司的核心产品，我们的意见其实是可以影响很多产品和运营的决策的。

冷启动问题就是一个非常好的点去做跨团队合作。

- 比如说跟产品团队的合作可以加入一些用户注册的冷启动信息页面。
- 跟数据团队合作看看能否从公司的DMP中获得更多可用于冷启动的用户、物品信息。
- 跟运营团队合作看看能不能推动更多跟第三方数据公司的合作，获得更多更全面的数据，等等。