# 检索

As Advanced search 高级搜索

Bs 基础索引 倒排索引

AS（Advanced Search）即高级搜索，2层架构的第一层，将WebPM.UI发送过来的检索请求发给BS，把BS的结果归并以及过滤后返回WebPM.UI。 AS是一个大模块，包含AC和BC两层模块。总体来说，AS接收US发送过来query，将query分发给BS计算，再merge BS的计算结果交给US，与US交互的是AC，与BS交互的是BC。

US（Universal Searcher） 通用搜索 是垂类检索资源整合的入口，主要功能包括两个部分：1、query阶段的query解析和分发，包括并行访问多个后端服务、合并及排序策略、缓存各个服务结果三个；2、数据返回阶段的结果整合，主要是将 aladdin 特殊需求结果，ecom 广告结果，ks 等搜索结果与大搜自然结果合并返回给前端。

BC（Basic searcher Controller），介于AS和BS之间的中间层。

模块定位：AC:Advanced Search Conroller，高级检索的控制模块（高级是相对于基础检索模块的BS而言的），高级体现在AC能在一定程度上理解用户的需求，返回内容除了具有文本相关性，还具有需求相关性

模块功能：AC将后端BC/BS返回的结果汇集起来，完成相关性排序并获取结果的摘要，将最相关的N条结果返回给US。

feedas load了所有笔记的一些特征和score之类的在内存 然后每天换库

冷启动是指在模型运行初期阶段，没有提前给模型赋予一定的初始条件，需要模型在整个运行期间实现各种指标数据从无到有的过程，比如模拟水位需要从0开始到趋近实际水位，模拟海流需要从0开始到趋近实际流速流向。就相当于这个初期就是让模型开始热起身来，先建立起该有的条件来，初期的数据是不准确的，是无法选用为研究结果的。如果模型模拟时间区段足够长，那么在模型运行中后期阶段，模型就已达到接近实际的各种条件了，此时模型才算是真正的开始模拟你的研究内容。

热启动是指在模型运行初期阶段，我们提前给模型赋予一定的初始条件，使模型从一开始就处于接近实际状况的状态，就相当于模型从一开始就已经热好身了，直接就可以上赛场开始干活了。热启动下的模型从模拟一开始就在真正的模拟你的研究内容了。

————————————————

版权声明：本文为CSDN博主「晏长街」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/m0_50132779/article/details/119236189>

<https://zhuanlan.zhihu.com/p/572998087>

除了倒排召回 还有ann召回

用户session召回

Ann（approximate nearest neighbor）是指一系列用于解决最近邻查找问题的近似算法。最近邻查找问题，即在给定的向量集合中查找出与目标向量距离最近的N个向量。

比较平凡的方法就是线性查找方法，也就是说每次检索都会遍历整个检索库，虽然准确率是100%，但是其效率是随着检索库的增加而线性增加的，当检索库的大小超过一定阈值之后，每次检索的时间开销将会变得巨大，不能忍受。在搜索集小的时候，还可以接受，但是实际中向量集合往往很大，线性查找方法的计算资源消耗较大且耗时较长，在工程上一般都不会采用这种方法。

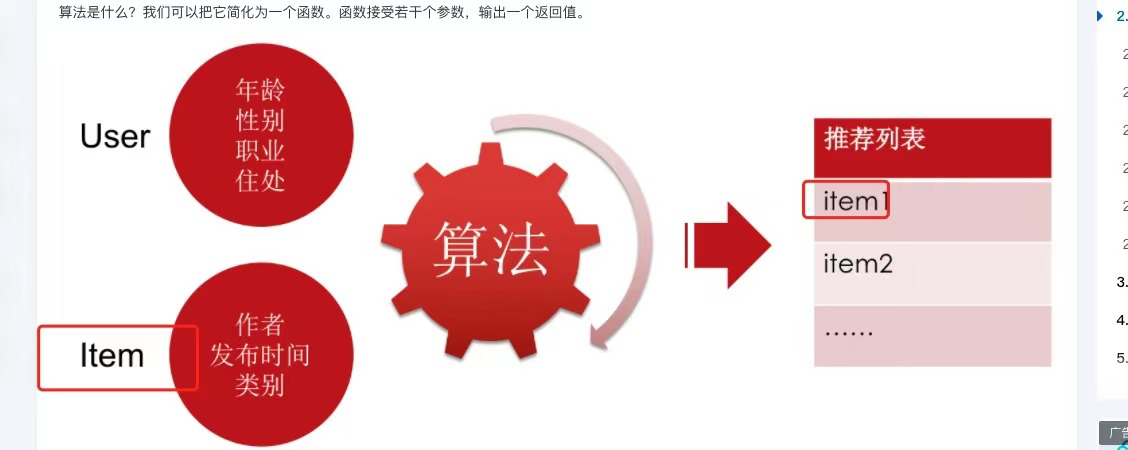
在 NN 的搜索上，提出了一些树的方法，比如说 KD-tree，以及进化的 ball-tree，像kd-tree等一些优化方法并没有提高高维空间里的搜索效率，效率甚至比线性扫描还要低，导致准确最近邻搜索难以直接用于实际问题。顾名思义，近似最近邻搜索找到的向量不要求一定是准确的最近邻，只要求距离准确的最近邻很近就可以了，可以看到近似近邻实际上是效果与性能的折衷

新闻推荐系统从海量新闻中推荐出你感兴趣的新闻，百度从海量的搜索结果中找到最优的结果，短视频推荐出你每天都停不下来的视频流，这些里面都包含有今天要介绍的ANN方法

Approximate Nearest Neighbor

ANN

## CF



协同过滤算法

协同过滤算法(Collaborative Filtering, CF)是很常用的一种算法，在很多电商网站上都有用到。CF算法包括基于用户的CF(User-based CF)和基于物品的CF(Item-based CF)。

基于用户的CF原理如下：

分析各个用户对item的评价（通过浏览记录、购买记录等）；

依据用户对item的评价计算得出所有用户之间的相似度；

选出与当前用户最相似的N个用户；

将这N个用户评价最高并且当前用户又没有浏览过的item推荐给当前用户

基于流行度的算法