【开源工具】和推荐系统有关的开源工具及框架介绍

我们懂得了原理，知道了实际推荐系统需要考虑哪些元素之后。正当你摩拳擦掌之际，如果发现要先从挖地基开始，你整个人可能是崩溃的。

轮子不要重复造

但是事实上你没必要这样做也不应该这样做。大厂研发力量雄厚，业务场景复杂，数据量大，自己从挖地基开始研发自己的推荐系统则是非常常见的，然而中小厂职工们则要避免重复造轮子。这是因为下面的原因。

1. 中小企业，或者刚刚起步的推荐系统，要达成的效果往往是基准线，通用的和开源的已经能够满足；
2. 开源的轮子有社区贡献，经过若干年的检验后，大概率上已经好于你自己从零开始写一个同样功能的轮子；
3. 对于没有那么多研发力量的厂来说，时间还是第一位的，先做出来，这是第一要义。

既然要避免重复造轮子，就要知道有哪些轮子。

有别于介绍一个笼统而大全的“推荐系统”轮子，我更倾向于把粒度和焦点再缩小一下，介于最底层的编程语言API和大而全的”推荐系统”之间，本文按照本专栏的目录给你梳理一遍各个模块可以用到的开源工具。

这里顺带提一下，选择开源项目时要优先选择自己熟悉的编程语言、还要选有大公司背书的，毕竟基础技术过硬且容易形成社区、除此之外要考虑在实际项目中成功实施过的公司、最后还要有活跃的社区氛围。

内容分析

基于内容的推荐，主要工作集中在处理文本，或者把数据视为文本去处理。文本分析相关的工作就是将非结构化的文本转换为结构化。主要的工作就是三类。

1. 主题模型；
2. 词嵌入；
3. 文本分类。

可以做这三类工作的开源工具有下面的几种。

开源项目名	用途	接口语言	单机/分布式	支持方
LightLDA	主题模型	C++	分布式	Microsoft
gensim	主题模型，词嵌入	python	单机多线程	adimrehurek.com
plda	主题模型	C++	单机多线程/分布式	Google
DMWE	词嵌入	C++	分布式	Microsoft
tensorflow-word2vec	词嵌入	Python	分布式或单机	Google
FastText	词嵌入，文本分类	C++/Python	单机多线程	Facebook
liblinear	文本分类	C++，Java，Python等	单机	台湾大学

由于通常我们遇到的数据量还没有那么大，并且分布式维护本身需要专业的人和精力，所以请慎重选择分布式的，将单机发挥到极致后，遇到瓶颈再考虑分布式。

这其中FastText的词嵌入和Word2vec的词嵌入是一样的，但FastText还提供分类功能，这个分类非常有优势，效果几乎等同于CNN，但效率却和线性模型一样，在实际项目中久经考验。LightLDA和DMWE都是微软开源的机器学习工具包。

协同过滤和矩阵分解

基于用户、基于物品的协同过滤，矩阵分解，都依赖对用户物品关系矩阵的利用，这里面常常要涉及的工作有下面几种。

1. KNN相似度计算；
2. SVD矩阵分解；
3. SVD++矩阵分解；
4. ALS矩阵分解；
5. BPR矩阵分解；
6. 低维稠密向量近邻搜索。

可以做这些工作的开源工具有下面几种。

开源项目名	用途	接口语言	单机/分布式	支持方
kgraph	KNN相似度计算和搜索	C++，Python	单机多线程	aaalgo(Wei Dong)
annoy	稠密低维向量的KNN相似搜索	C++，Python	单机多线程	Spotify
faiss	稠密低维向量的KNN相似搜索，聚类	C++，Python	单机多线程， 支持GPU加速	Facebook
nmslib	稠密低维向量的KNN相似搜索	C++，Python	单机	nmslib
Spark.RowMatrix.columnSimilarities	基于用户/基于物品协同过滤	Scala， Java，Python	单机多线程， 分布式	Twitter
lightfm	SVD矩阵分解，SVD++矩阵分解， BPR矩阵分解	Python	单机多线程	lyst
implicit	基于用户/物品的协同过滤， ALS矩阵分解，BPR矩阵分解	Python	单机多线程， 支持GPU加速	benfrederickson.com
QMF	加权ALS矩阵分解，BPR矩阵分解	C++，Python	单机多线程	Quora

这里面的工作通常是这样：基础协同过滤算法，通过计算矩阵的行相似和列相似得到推荐结果。

矩阵分解，得到用户和物品的隐因子向量，是低维稠密向量，进一步以用户的低维稠密向量在物品的向量中搜索得到近邻结果，作为推荐结果，因此需要专门针对低维稠密向量的近邻搜索。

同样，除非数据量达到一定程度，比如过亿用户以上，否则你要慎重选择分布式版本，非常不划算。

模型融合

模型融合这部分，有线性模型、梯度提升树模型。

开源项目名	用途	接口语言	单机/分布式	支持方
LightGBM	GBDT等树模型	C++	分布式	Microsoft
XGBoost	GBDT等树模型	C++，Python，R	单机多线程， 分布式	Distributed (Deep) Machine Learning Community
Tensorflow-wide and deep	Wide&Deep模型	Python	单机多线程， 分布式	Google
LibFFM	因子分解机， 场敏感的因子分解机	C++，Python	单机	台湾大学
vowpal_wabbit	线性模型	C++，Python，Java， C#等	单机多线程， 分布式	Microsoft

线性模型复杂在模型训练部分，这部分可以离线批量进行，而线上预测部分则比较简单，可以用开源的接口，也可以自己实现。

其他工具

Bandit算法比较简单，自己实现不难，这里不再单独列举。至于深度学习部分，则主要基于TensorFlow完成。

存储、接口相关开源项目和其他互联网服务开发一样，也在对应章节文章列出，这里不再单独列出了。

完整推荐系统

这里也梳理一下有哪些完整的推荐系统开源项目，可以作为学习和借鉴。 所谓完整的推荐系统是指：包含推荐算法实现、存储、接口。

开源项目名	说明	开发语言
PredictionIO	基于Spark(算法)、HBase（存储）、Spray（接口）开发	Scala
recommendationRaccoon	以协同过滤为核心算法的推荐系统，Redis作为存储	Node.js
easyrec	存储采用MySQL，接口基于Tomcat	Java
hapiger	存储采用PostgreSQL，接口采用Hapi.js框架	Node.js

总结

你可能注意到了，这里的推荐系统算法部分以Python和C++为主，甚至一些Python项目，底层也都是用C++开发而成。

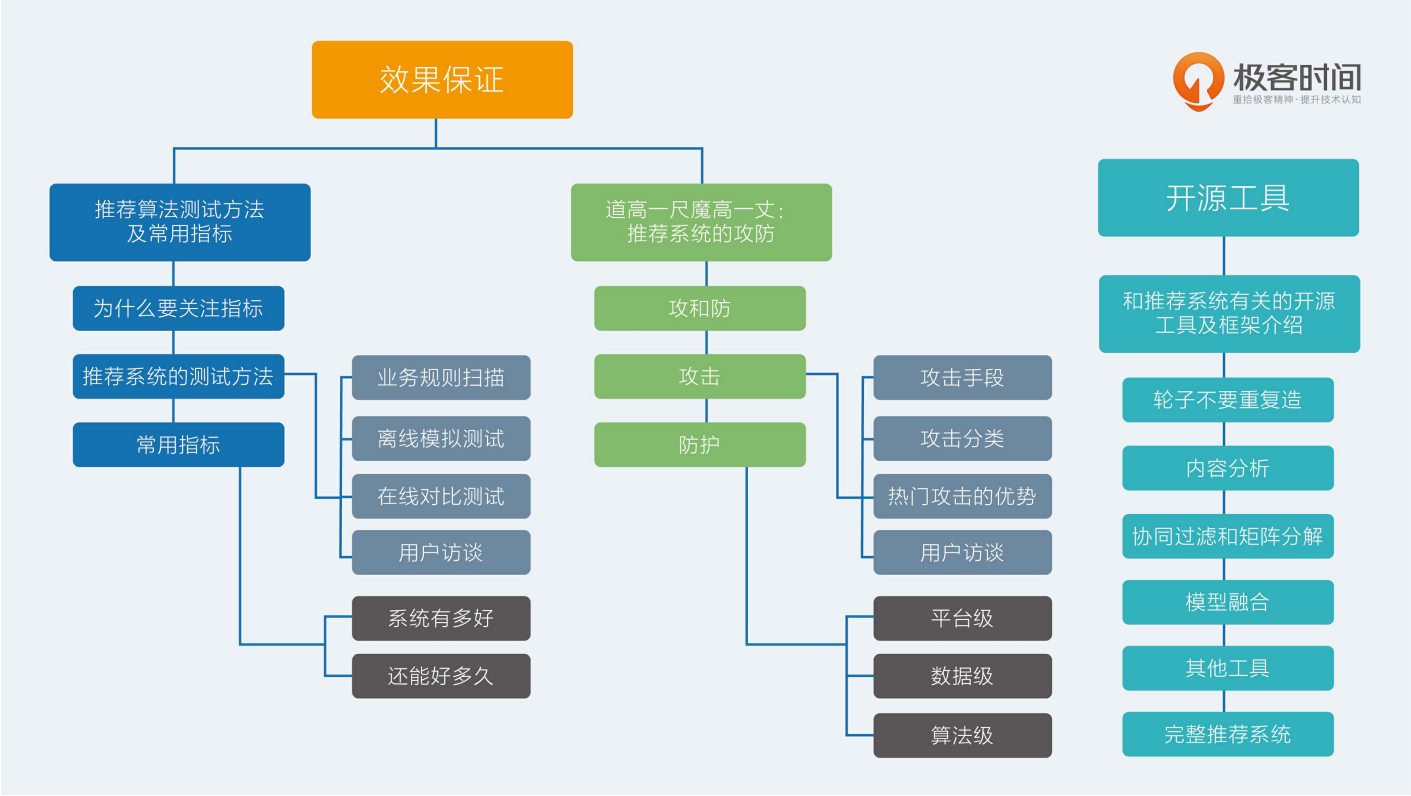
因此在算法领域，以Python和C++作为开发语言会有比较宽泛的选择范围。

至于完整的推荐系统开源项目，由于其封装过于严密，比自己将大模块组合在一起要黑盒很多，因此在优化效果时，不是很理想，需要一定的额外学习成本，学习这个系统本身的开发细节，这个学习成本是额外的，不是很值得投入。

因此，我倾向于选择各个模块的开源项目，再将其组合集成为自己的推荐系统。这样做的好处是有下面几种。

1. 单个模块开源项目容易入手，学习成本低，性能好；
2. 自己组合后更容易诊断问题，不需要的不用开发；
3. 单个模块的性能和效果更有保证。

当然，还是那句话，实际问题实际分析，也许你在你的情境下有其他考虑和选择。如果还有哪些开源项目，你觉得值得推荐，也欢迎留言分享。



精选留言

slvher

总结很赞！

topic model 可选的还有 Baidu Familia

embedding 可选的还有 FAIR starspace

2018-05-24 11:15

云学

非常实用，谢谢

2018-05-19 11:15

芭蕾小丑

如果是电商的推荐系统，能不能把以上几个模块帮忙组合一下，给个可行的方向吧？

2018-06-23 21:36

Sin0

不错，很实用！

2018-05-18 09:35

华仔

Spark MLlib和Mahout这两个好像是全家桶，大神如何评价？

2018-07-11 15:03

作者回复

工具当然要用专精的。

2018-07-23 20:51

风的轨迹

真是盼星星盼月亮，这篇文章真是解决了初学者如何能够对核心算法有一个快速的感性认识的问题，感谢陈老师

2018-05-30 08:43

贾贵源

很棒，已分享。会有更多的同事想订阅

2018-09-08 10:37

作者回复

谢谢！

2018-11-28 01:15

曾阿牛

老师,使用java语言的librec开源框架用来做推荐算法怎么样呢？谢谢

2018-08-03 22:56

爱看球的领带

老师好，ElasticSearch作存储和计算，推荐一下怎么学习吧，感谢

2018-08-03 15:11