

今天来学习下美团推荐算法实践：机器学习重排序模型

转载

2015年11月14日 20:38:15

标签：机器学习 / hbase / storm / spark / 美团

5445

转载，来自小象学院。

美团推荐算法实践：机器学习重排序模型

文章介绍了美团推荐系统的构架和优化过程，包括数据层，触发层，融合过滤层和排序层，采用了Hbase，Hive，Storm，Spark和机器学习等技术。

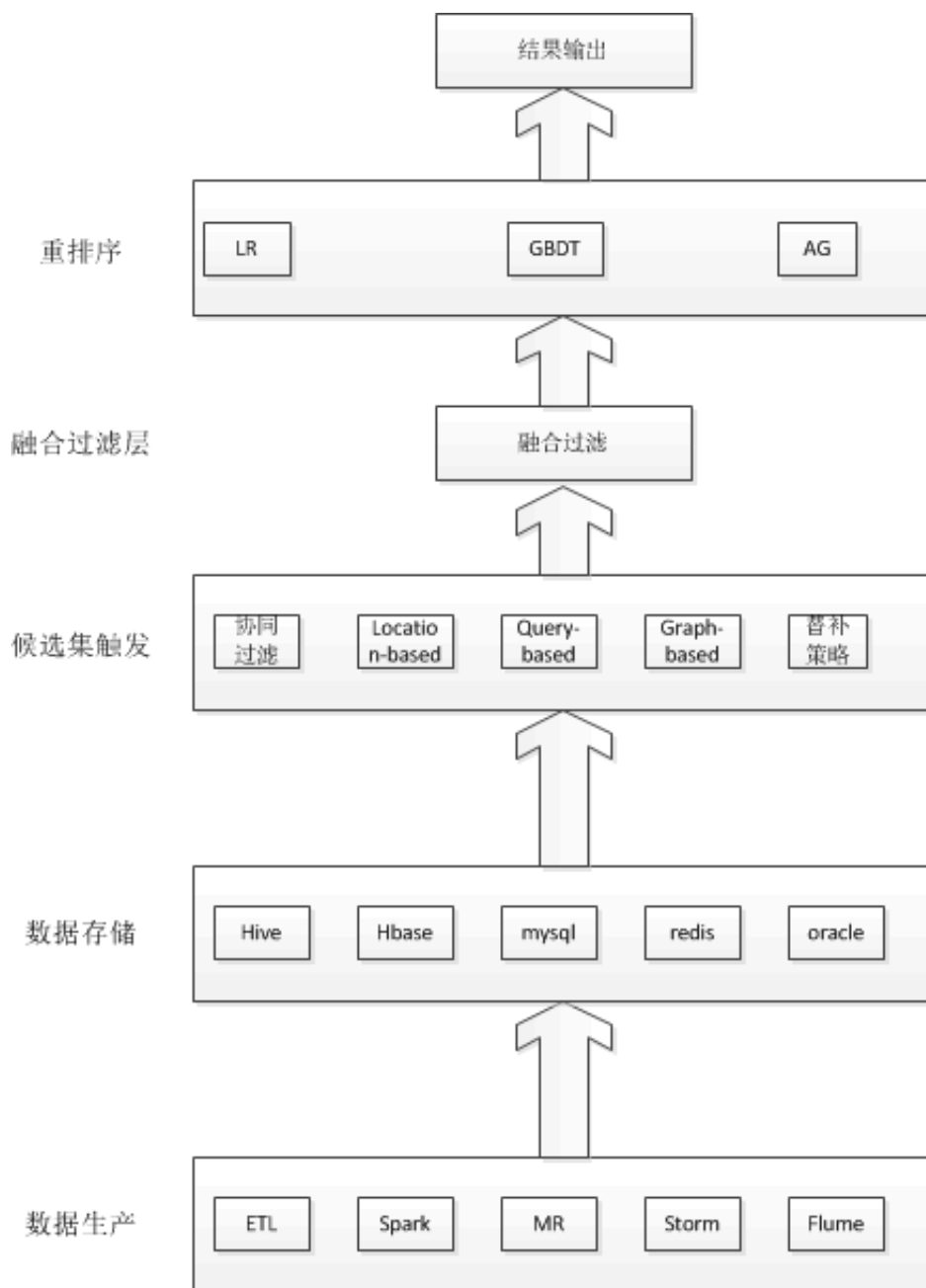
两个优化点将是候选集进行融合与引入重排序模型。

看的到这里顿时觉得高大上了，有木有.....

在用户意图明确时，我们通过用搜索引擎来结局互联网时代的信息过载问题，但当用户的意图不明确的时候或则很难用清洗的语义表达的时候，搜索引擎就无能为力。此时，借助推荐系统通过用户行为的分析理解其意图，为其推送个性化的结果，变成了好的选择。美团作为国内发展较快的o2o网站，有着大量的用户和丰富的用户行为，这些为推荐系统的应用和优化提供了很好的条件。

1：从框架来看

从框架的角度看，推荐系统基本可以分为数据层，触发层，融合过程和排序层。数据层包括了数据生产层和数据存储层，主要是利用各种数据处理工具对原始的日志进行清洗，处理成格式化的数据，落地到不同类型的存储系统中，供下游的算法和模型使用。候选集触发层主要是从用户的历史行为，实时行为，地理位置等角度利用各种触发策略产生推荐的候选集。候选集融合和过滤层有2个功能，一是对不同的候选集进行融合，提高推荐策略的覆盖度和精度；2是承担一定的过滤职责，从产品，运营角度确定一些人工规则，过滤掉不符合调教的item。排序层主要是利用机器学习的模型对触发层筛选出来的候选集进行重排序。



LR:是逻辑回归，GBDT是随机森林（这里错误了，我把gbdt和随机森林搞混淆了以前），AG不知道。

location-based基于位置。Query-based基于查询。Graph-based基于图计算。替补策略是热门推荐么？*

PS:一般来说，到候选集的时候，然后通过一些方法（对物品兴趣度排序，从高往下取item，人工定义规则之后就玩了），但是美团新增加了一个使用算法重排序的过程。为什么？，后面讲到了。

同时，对与候选集触发和重排序两层而言，为了效果迭代是需要频繁修改的两层。一次需要支持ABTest(强行插入，ABTest就是，那两个模型的效果去

上线测试，看那个的好)。为了支持高效率的迭代，我们对候选集触发和重排序两层经行了解耦，这两层的结果是正交的，因此可以分别进行对比试验，不会互相影响。同时在每一层的内部，我们会根据用户的流量划分为多份，支持多个策略同时在线对比。

PS:这段话的意思是，候选集层和排序层是分离的，这两个要做对比试验。而对于候选集层也是更加用户量分离成多个的，对于排序层也一样。组类对比，组外对比。

2数据应用

数据乃算法，模型的根本。美团做为一个交易平台，同时具有快速增长的用户量，因此产生了海量丰富的用户行为数据。当然，不同类型的数据价值和反映的用户意图的强索也有所不同。

行为类别	行为详情
主动行为数据	搜索，筛选，点击，收藏，下单，支付，评分
UGC	文本评价，上传图片
负反馈数据	左滑删除，取消收藏，取消订单，退款，差评，低评
用户画像	人口属性，美团NDA，品种偏好，消费水平，工作地和居

1用户主动行为数据记录了用户在美团平台上不同的环节的各种行为，这些行为一方面用于候选集触发算法中的离线运算，另一方面这些行为代表的意图强弱不同，因此在训练重排序模型时可以针对不同的行为设定不同的回归目标值，以更细地刻画用户的行为强弱。此外用户对deal的这些行为还可以作为重排序模型的交叉特征每用户模型的离线训练和在线预测。

这句话“因此在训练重排序模型时可以针对不同的行为设定不同的回归目标值”，每没看懂，LR逻辑回顾是2分类问题，GBTD是可以处理分类和回归问题的。它的意思是确定y值，求出各个行为的系数，也就是各个行为的意图强弱么，这种到时可以。

2负反馈数据反映了当前的结果可能在某些方面不能满足用户的需求，因此在后续的候选集触发过程中需要考虑对特定的因素经行过滤或者降权，降低负面因素的再次出现的几率，提高用户体验，同时在重排序的模型训练中，负反馈数据可以作为不可多的的负列参加模型训练，这样负列要比那些展示后未点击，未下单的样本显著的多。

- 3用户画像是刻画用户属性的基础数据，其中，有些是直接获取的原始数据，有些是经过挖掘的二次加工的数据，这些属性一方面可以用户候选集触发过程对deal进行加权或者降权，另外一方面可以作为重排序中的用户纬度特征。
- 4对UGC的挖掘可以提取一些关键词，然后使用这些关键词给deal打标签，用于deal的个性化展示。

3策略触发

上文中提到数据的重要性，但是数据的落脚点还是算法和模型。单纯的数据只是一些字符的堆积，我们必须通过对数据的清洗和除去数据中的噪声，然后通过算法和模型学习其中的规律，才能将数据的价值最大化。在本节中，将介绍推荐过程中候选集触发中用到的相关算法。

1协同过滤

提到推荐，就不得不说协同过滤，它几乎在每个推荐系统中都会用到。基本的算法非常简单，但是要获得好的效果，往往需要根据业务做一定差异化的处理。

清除作弊，刷单，代购等噪声数据。这些数据的存在会严重影响算法的效果，因此第一步把噪声数据清除。

合理选取训练数据。选取的训练数据时间床头不宜过长，也不宜果短。具体的窗口周期需要经过多次的实验来确定。同时可以考虑引入时间衰减，因为近期的用户行为更能反映用户接下来的行为动作。

null	群体/个体	计算代价	使用场景	冷启动
User-based	更依赖与当前用户相似的用户群体的社会化行为	适用于用户数较少的场合	时效性强，用户个性化兴趣不太显著场景	新加入的物品能很快进入推荐列表中
Item-based	更侧重与自身的个体行为	适用于物品数较少的场合	长尾物品丰富，用户个性化需求强烈的场合	新加入的用户能很快得到推荐

尝试不同的相似度计算。在实践中，我们采用了一种称作loglikedhood相似度计算方法。

下表表示了EventA和EventB之间的相互关系，其中

k11表示：EventA和EventB共出现的次数

K12表示：EventB发生，EventA未发生次数

k21表示：EventA发生，EventB未发生次数

k22表示：EventA和EventB都不发生的次数

null	Event A	Everything
Event B	K11 A and B together	
Everything but B	K21,A but not B	

则 $\log\text{LikelihoodRatio}=2*(\text{matrixEntropy}-\text{rowEntropy}-\text{columnEntropy})$

其中 $\text{rowEntropy}=\text{entropy}(k11,k12)+\text{entropy}(k21,k22)$

$\text{columnEntropy}=\text{entropy}(k11,k21)+\text{entropy}(k12,k22)$

$\text{matrixEntropy}=\text{entropy}(k11,k12,k21,k22)$

entropy为几个元素组成的系统的香农熵。

2location-based

对于移动设备而轻言，与PC端最大的区别之一就是移动设备的位置经常变化。不同的地理位置反映了不同的用户场景，在具体的业务中可以充分利用用户所在的地理位置。在推荐的候选集触发中，我们会更具用户的实时地理位置，工作地，居住地等地理位置触发相应的策略。

根据用户的历史消费，历史浏览等，挖掘出某一粒度的区域（比如商业圈）的区域消费热单和区域购买热单。

当新的上线用户请求到达时，根据用户的地理位置和相对应地理位置的区域消费热单和区域热单经行加权，最终得到一个推荐列表。此外还可以通过用户出现的地理位置，采用协同过滤方式计算用户的相似度。

3 query-based

收索是一种强的用户行为，比较明确的反映了用户的意愿，但是在很多情况下，因为各种各样的原因，没有形成最终的转换。尽管如此我们人为，这种情景还是代表了一定能的用户意愿么可以加以利用，具体做法如下：

1对用户过去一段时间的收索无法转换行为进行挖掘，计算每个用户对不同query的权重。2计算每个query下不同deal的权重。2当用户再次提交请求时，根据用户对不同query的权重及其query下不同deal的权重经行加权，取出权重最大的TopN推荐。

这段意思是，1计算的是每个用户的query，而且该query中没有转化成订单

的，把这些query的权重确定好。2在计算所有的query中，里面有订单的，订单的权重。3在2个权重求和，取topn。

4 graph=based

对于协同过滤而言，user之间或者deal之间的图距离是两跳，对于更远距离的关系则不能考虑在内。而图算法可以打破这一限制，讲user与deal的关系视作一个二部图，互相间的关系可以在图上传播。simrank是一种平衡对等实体相似度的图算法。它的基本思想是，如果两个实体与另外的相似实体有相关关系，那他们也是相似的，即相似性可以传播。计算出相似矩阵之后，就是用相似协同过滤的方式推荐。

不清楚图计算，但是主要还是借助图计算，求相似

5用户实时行为

目前我们的业务会产生包括收缩，筛选，收藏，浏览，下单等丰富的用户行为，这些是我们进行效果优化的重要基础。我们当然希望每个用户行为流都能到达转换环节，但是事实远非如此。

当用户产生了下单行为为上游的某些行为时，会有相当一部分因为各种原因使行为流没有形成转化。但是，用户的这些上游行为对我们而言是非常重要的先验知识。很多情况下，用户当时没有转化不代表用户对当前的item不感兴趣。当用户再次到达我们的推荐展位，我们根据用户之前产生的先验行为理解并识别用户的真正意图，将符合用户意图的相关deal再次展现给用户，引导用户沿着行为流下游进行，最后达到下单这个终极目标。目前引入的实时用户行为包括：实时浏览，实时收藏。

6替补策略

虽然我们有一系列基于用户历史行为的候选集触发算法，但对于某部分新用户后者历史行为不太丰富的用户，上述算法触发的候选集太小，所以需要使用一些替补策略进行填充。

热销单：一定时间内销量最多的item，可以考虑时间衰减情况。

好评单：用户产生的评价中，评分较高的item。

城市单：满足基本的限定条件，在用户的请求城市内。

4子策略融合

为了结合不同触发算法的有点，同时提高候选集的多样性和覆盖率，需要将不同的触发算法融合在一起。常见的融合方法有以下几种：

加权型：最简单的融合方法就是根据经验值对不同算法赋予不同的权重，对各个算法的候选集按照给定的权重进行加权，然后按照权重排序。

分级型：优先采用效果较好的算法，当产生的候选集太小不足以满足目标值时，再使用效果次好的候选集，然后叠加产生最终总的候选集。

调制型：不同的算法按照不同的比例产生一定量的候选集，然后叠加产生最终总的候选集。

过滤性：当前的算法对前一级算法产生的候选集进行过滤，一次类推，候选集一级一级被过滤，最终产生一个小而精的候选集。

目前我们使用的方式集成了调制和分级两种融合方法，不同的算法根据历史效果表现给定不同的候选集比例，同时优选采用效果好的算法触发，如果候选集不够大，在采用效果次之的算法触发，以此类推。

5候选集重新排序

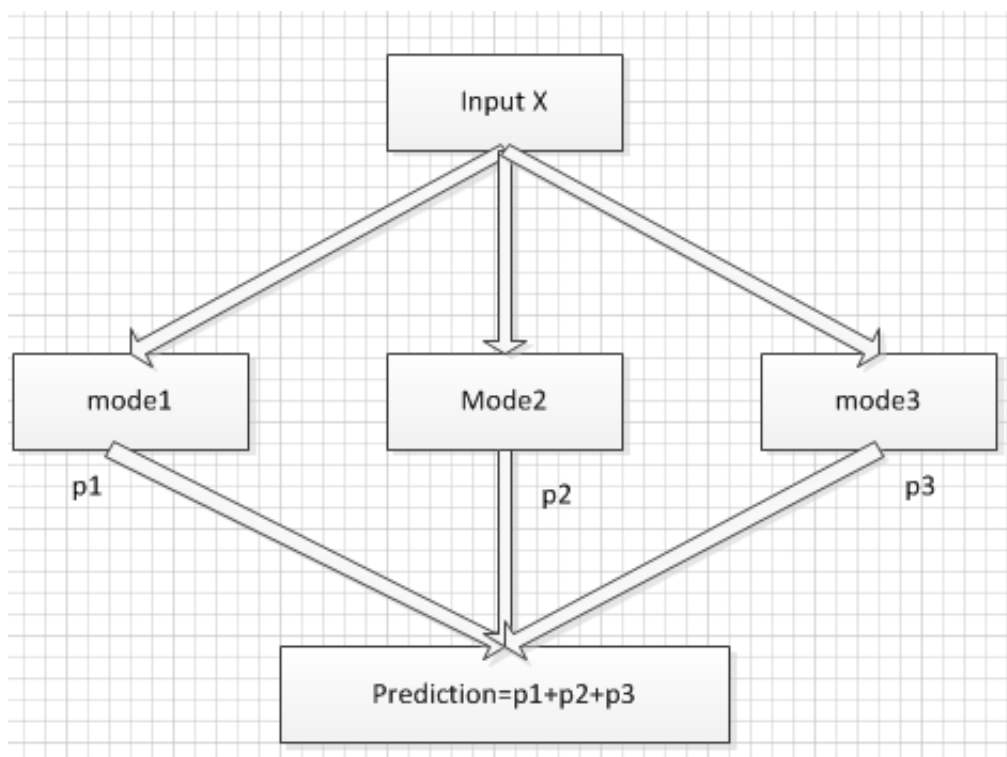
如上所述，对于不同算法触发出来的候选集，只是根据算法的历史效果决定算法产生的item的位置显得有点简单粗暴了，同时在每个算法的内部，不同item的顺序也只是简单的由一个或者几个因素决定，这些排序结果需要借助机器学习的方法，使用相关的排序模型，综合多方面的因素来决定。

1模型

非线性模型能较好的捕捉特征中的非线性关系，但训练和预测的代价相对线性模型要求高一些，这也导致了非线性模型的更新周期相对要长。反之，线性模型对特征的处理要求比较高，需要凭借领域知识和经验人工对特征做一些先期处理，但因为线性模型简单，在训练和预测时效较高。因此在更新周期上可以更短，还可以结合业务做一些在线学历的尝试。我们的实践中，非线性模型和线性模型都在应用。

1非线性模型

目前我们主要采用了非线性的树模型，相对线性模型，非线性模型可以更好的处理特征中的非线性关系，不必像性模型那样在特征处理和特征组合上化比较大的精力。AG是一个加性模型，有很多个Grove组成，不同的Grivoe之间经行bagging得出最后的预测结果，由此可以减小过拟合的影响。



每个Grove有多颗树组成，在训练时每棵树的拟合目标为真实值与其他树预测结果之后之间的残差。当达到给定数目的树时，重新训练的树会逐颗替代以前的树。经过多次迭代之后收敛。

PS：这就是随机森林嘛，不过他改进了，当达到数据给定的树时，会重新训练替换原来的树，多次之后，收敛。这个想法太好了，以前我也用过，但是没想到在随机森林也可以用。

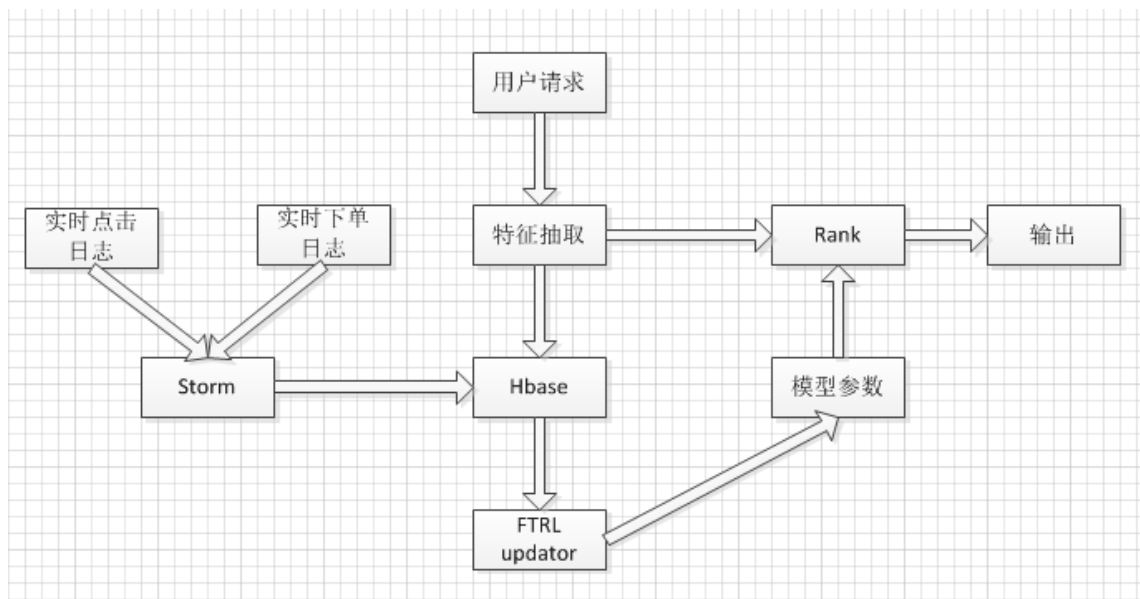
2线性模型

目前应用的比较多的线性模型非Logistic Regression莫属。为了能实时捕捉数据分布的变化，我们引入了online learning，接入实时数据流，使用google提出的FTRL方法对模型进行在线更新。

PS：在线，以前都只是听过。

主要步骤如下：

- 1.在线写特征向量到Hbase中
- 2.Storm解析实时点击和下单日志流，改写Hbase中对应特征向量的label
- 3.通过FTRL更新模型权重
- 4.将新的模型参数应用与线上



2数据

采样：对于点击率预估而言，正负样本严重不平衡，所以需要对负样本做一些采样。

负列：正列一般是用户产生点击，下单等转化行为的样本，但是用户没有转换行为的样本是否就一定是负列呢？（这样考虑很好啊，学习了）其实不然，很多展现其实用户根本没有看到，所以把这样样本视为负列是不合理的，也会影响模型的效果。比较常用的方法是skip-above，即用户点击的item位置以上的展现才可能视为负列。除此以外，我们还有用户主动少出的显示的负反馈数据，这些数据是高质量的负列。

去噪：对数据混杂的刷单等作弊行为数据，要将其排除，否则会直接影响模型的效果。

PS：哪有那么容易排除刷单，我朋友就有搞刷单的，大部分都是根据淘宝的规矩来的，还有人专门培训。

3特征

在我们目前的重排序模型中，大概分为以下几类特征：

deal(团购单)：主要是deal的本身属性，包括价格，折扣，销量，评分，类别，点击率等。

user纬度：用户等级，用户的人口统计，用户的客户端类型等

user, ideal交叉特征：包括用户对deal的点击，收藏，购买等。

距离特征：包括用的的实时地理位置，常去地理位置在，工作地，对特征值做一些分桶，归一化处理，使特征量纲统一。

6**总结**

以数据为基础，用算法去雕琢，只有将二者有机结合，才能带来效果的提升。对我们而言，一下两个节点是我们优化过程中的里程碑：

1将候选集融合：提高了推荐的覆盖率，多样性和精度。

2引入重排序模型：解决了候选集增加以后deal之间排列顺序的问题。

PS：以上剔除的算法，我以后会开一个专题和大家讨论.....今天就到这里了，如果有关推荐有好的点子的，请留言哈.....我想学习下.....^_^