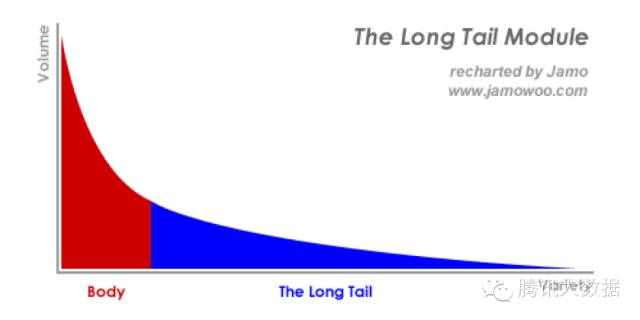
推荐算法与AutoML

* 算法提出背景
  + The long tail

经济学理论 The Long Tail

互联网领域：最热的那一小部分资源得到绝大部分的关注，而剩下的很大一部分资源却鲜少有人问津，这不仅成了资源利用上的浪费，也让很多口味偏小众的用户无法找到自己感兴趣的内容。

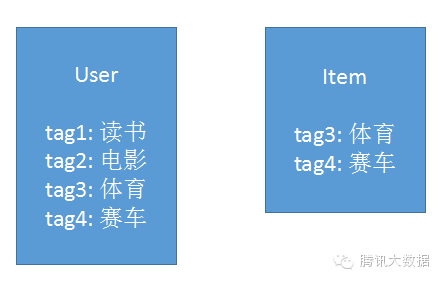
* 传统推荐算法小结
  + 协同过滤推荐算法（CF算法）
    - 简介
      * 通过用户的一系列行为，寻找特定模式来产生针对用户的特殊推荐
    - 参数
      * 惯用数据（评价，购买，下载等用户偏好）
    - 类型
      * 基于用户的CF(User-based CF)

1. 分析各个用户对item的评价（通过浏览记录、购买记录等）；
2. 依据用户对item的评价计算得出所有用户之间的相似度；
3. 选出与当前用户最相似的N个用户；
4. 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。
   * + - 基于物品的CF(Item-based CF)
5. 分析各个用户对item的浏览记录。
6. 依据浏览记录分析得出所有item之间的相似度；
7. 对于当前用户评价高的item，找出与之相似度最高的N个item；
8. 将这N个item推荐给用户。
   * + 评价

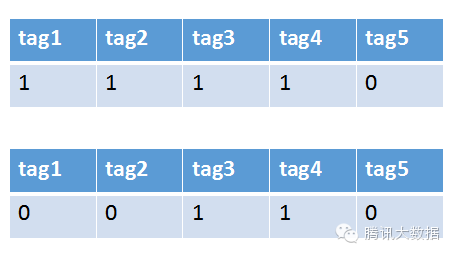
CF算法确实简单，而且很多时候推荐也是很准确的。然而它也存在一些问题：

* + - * + 依赖于准确的用户评分；
        + 在计算的过程中，那些大热的物品会有更大的几率被推荐给用户；
        + 冷启动问题。当有一名新用户或者新物品进入系统时，推荐将无从依据；
        + 在一些item生存周期短（如新闻、广告）的系统中，由于更新速度快，大量item不会有用户评分，造成评分矩阵稀疏，不利于这些内容的推荐。
  + 基于内容的推荐算法
    - 简介
      * 通过用户行为（阅读新闻，书籍）获取关键字作为参数，检索相关内容提供推荐。
    - 参数
      * 用户行为的关键字总结
    - 算法举例

现在系统里有一个用户和一条新闻。通过分析用户的行为以及新闻的文本内容，我们提取出数个关键字，如下图：



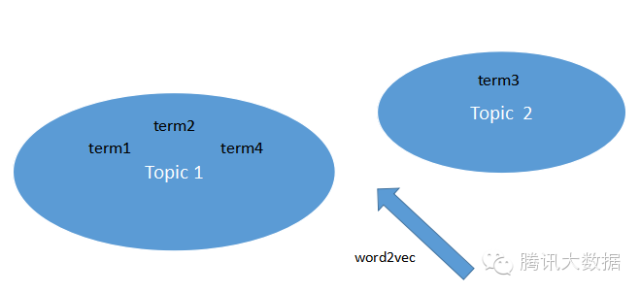
　　将这些关键字作为属性，把用户和新闻分解成向量，如下图：



之后再计算向量距离，便可以得出该用户和新闻的相似度了。这种方法很简单，如果在为一名热爱观看英超联赛的足球迷推荐新闻时，新闻里同时存在关键字体育、足球、英超，显然匹配前两个词都不如直接匹配英超来得准确，系统该如何体现出关键词的这种“重要性”呢？这时我们便可以引入词权的概念。在大量的语料库中通过计算（比如典型的TF-IDF算法），我们可以算出新闻中每一个关键词的权重，在计算相似度时引入这个权重的影响，就可以达到更精确的效果。

　　sim(user, item) = 文本相似度(user, item) \* 词权

然而，经常接触体育新闻方面数据的同学就会要提出问题了：要是用户的兴趣是足球，而新闻的关键词是德甲、英超，按照上面的文本匹配方法显然无法将他们关联到一起。在此，我们可以引用话题聚类：



利用word2vec一类工具，可以将文本的关键词聚类，然后根据topic将文本向量化。如可以将德甲、英超、西甲聚类到“足球”的topic下，将lv、Gucci聚类到“奢侈品”topic下，再根据topic为文本内容与用户作相似度计算。

* + - 评价

基于内容的推荐算法能够很好地解决冷启动问题，并且也不会囿于热度的限制，因为它是直接基于内容匹配的，而与浏览记录无关。然而它也会存在一些弊端，比如过度专业化(over-specialisation)的问题。这种方法会一直推荐给用户内容密切关联的item，而失去了推荐内容的多样性。

* + 流行度推荐算法
    - 简介
      * 根据热度，流行度进行内容的推荐，简单暴力
    - 参数
      * 内容热度排行
    - 评价

这种算法的优点是简单，适用于刚注册的新用户。缺点也很明显，它无法针对用户提供个性化的推荐。基于这种算法也可做一些优化，比如加入用户分群的流行度排序，例如把热榜上的体育内容优先推荐给体育迷，把政要热文推给热爱谈论政治的用户。

* + 基于模型的算法
    - 简介

基于模型的协同过滤推荐就是基于样本的用户喜好信息，训练一个推荐模型，然后根据实时的用户喜好的信息进行预测，计算推荐。

它不需要对物品或者用户进行严格的建模，而且不要求物品的描述是机器可理解的，所以这种方法也是领域无关的。这种方法计算出来的推荐是开放的，可以共用他人的经验，很好的支持用户发现潜在的兴趣偏好

* + - 参数
      * 样本：大量用户喜好信息
    - 评价

基于模型的算法由于快速、准确，适用于实时性比较高的业务如新闻、广告等，而若是需要这种算法达到更好的效果，则需要人工干预反复的进行属性的组合和筛选，也就是常说的Feature Engineering。而由于新闻的时效性，系统也需要反复更新线上的数学模型，以适应变化。

方法的核心是基于历史数据，所以对新物品和新用户都有“冷启动”的问题。

推荐的效果依赖于用户历史偏好数据的多少和准确性。

在大部分的实现中，用户历史偏好是用稀疏矩阵进行存储的，而稀疏矩阵上的计算有些明显的问题，包括可能少部分人的错误偏好会对推荐的准确度有很大的影响等等。

对于一些特殊品味的用户不能给予很好的推荐。

由于以历史数据为基础，抓取和建模用户的偏好后，很难修改或者根据用户的使用演变，从而导致这个方法不够灵活。

* 推荐算法和机器学习
* 根据现有的模型预测代码结合AutoML方案设计