《推荐系统实践》学习笔记系列(四)

推荐系统

第四章

4.1 UGC标签系统

1. 定义：
   让普通用户给物品打标签，也就是UGC（User Generated Content，用户生成的内容）的标签应用。
2. UCG标签系统代表应用
   Delicious
   CiteULike
   Last.fm
   豆瓣
   Hulu

4.2 标签系统中的推荐问题

1. 标签系统中的推荐问题主要有以下两个：
   如何利用用户打标签的行为为其推荐物品（基于标签的推荐）？
   如何在用户给物品打标签时为其推荐适合该物品的标签（标签推荐）？
2. 对1.的回答
   2.1 用户为什么进行标注：
   i. 社会维度
   有些用户标注是给内容上传者使用的（便于上传者组织自己的信息），而有些用户标注是给广大用户使用的（便于帮助其他用户找到信息）。
   ii. 功能维度
   功能维度，有些标注用于更好地组织内容，方便用户将来的查找，而另一些标注用于传达

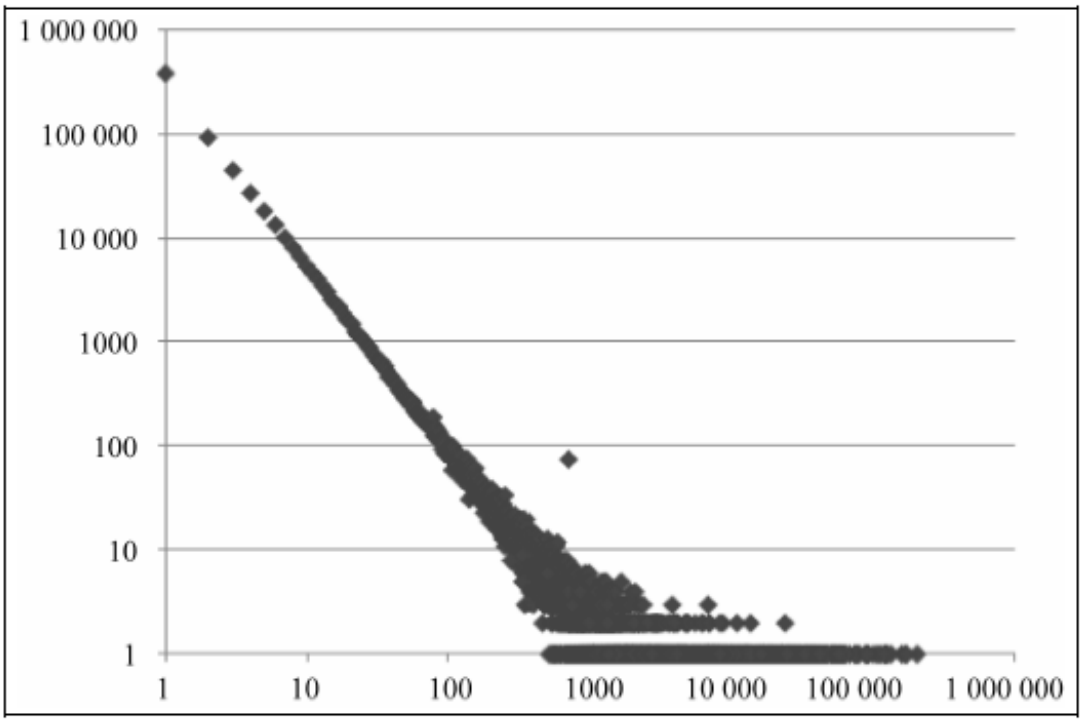
某种信息，比如照片的拍摄时间和地点等。

2.2 用户如何打标签

横坐标是流行度k，纵坐标是数据集中流行度为k的标签总数$n(k)$。标签的流行度分布也呈现非常典型的长尾分布，它的双对数曲线几乎是一条直线。

$$logn(k) = \alpha logk + \beta = logk^{\alpha} \cdot e^{\beta}$$

$$n(k) = k^{\alpha} \cdot e^{\beta}$$



标签流行度的长尾分布

2.3 用户打什么样的标签

不同网站的标签类型不同。

4.3 基于标签的推荐系统

1. 用户用标签来描述对物品的看法

   一个用户标签行为的数据集一般由一个**三元组**（u,i,b）组成，其中记录（u，i，b）表示用户u给物品i打上了标签b。
2. 一个最简单的算法

（1）统计每个用户最常见的标签。

（2）对于每个标签，统计被打过这个标签次数最多的物品。

（3）对于一个用户，首先找到他常用的标签，然后找到具有这些标签的最热门的物品推荐给这个用户。

对上面的算法，用户u对物品i的兴趣公式如下：

$$p(u,i)=\sum_b n_{u,b}n_{b,i}$$

这里，$B(u)$是被物品i打过的标签集合，$n_{u,b}$是用户u打过标签b的次数，$n_{b,i}$是物品i被打过标签b的次数。

3. 2.中算法的改进

   3.1

   - 前面这个公式倾向于给热门标签对应的热门物品很大的权重，因此会造成推荐热门的物品给用户，从而降低推荐结果的新颖性。

     另外，这个公式利用用户的标签向量对用户兴趣建模，其中每个标签都是用户使用过的标签，而标签的权重是用户使用该标签的次数。这种建模方法的缺点是给热门标签过大的权重，从而不能反应用户个性化的兴趣。惩罚热门标签：

     $$p(u,i)=\sum_b \frac{n_{u,b}}{log(1+n_b^{(u)})}n_{b,i}$$

     这里，$n_b^{(u)}$记录了标签b被多少个不同的用户使用过。这个算法记为TagBasedTFIDF。

   - 惩罚热门物品

     我们可以借鉴上面的思路，对热门的物品进行惩罚，从而得到公式：

     $$p(u,i)=\sum_b \frac{n_{u,b}}{log(1+n_b^{(u)})}\cdot\frac{n_{b,j}}{log(1+n_i^{(u)})}$$

     其中，$n_i^{(u)}$记录了物品i被多少个不同的用户打过标签。这个算法记为TagBasedTFIDF++。

   - 数据的稀疏性

     考虑到新用户或者新物品的标签很少，需要进行标签扩展。

     标签扩展的本质是对每个标签找到和它相似的标签，也就是计算标签之间的相似度。

     如果认为同一物品上的不同标签具有某种相似度，那么当两个标签同时出现在很多物品标签集合中时，我们就可以认为这两个标签具有较大的相似度。对于标签b，令N(b)为

有标签b的物品的集合，$n_{b,i}$为给物品i打上标签b的用户数，我们可以通过如下余弦相似度计算公式计算标签b和标签$b^{'}$的相似度：

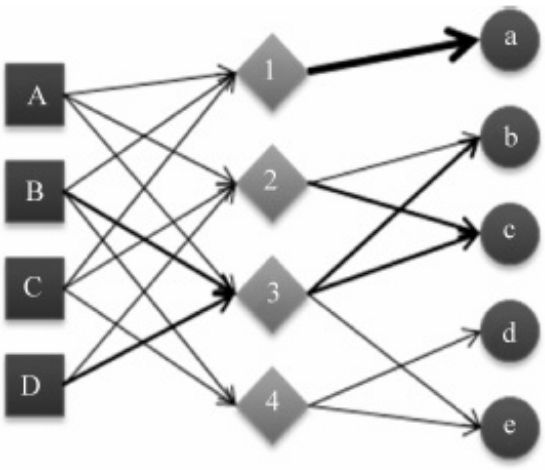
$$sim(b,b^{'})=\frac{\sum_{i\epsilon N(b)\bigcap N(b^{'})}n_{b,i}n_{b^{'},i}}{\sqrt{\sum_{i\epsilon N(b)}n_{b,i}^{2}\sum_{i\epsilon N(b^{'})}n_{b^{'},i}^{2}}}$$

- 标签清理
- 去除词频很高的停止词。
- 去除因词根不同造成的同义词，比如 recommender system和recommendation system。
- 去除因分隔符造成的同义词，比如collaborative_filtering和collaborative-filtering 为了控制标签的质量。
- 为了控制标签的质量，很多网站也采用了让用户进行反馈的思想，即让用户告诉系统某个标签是否合适。

3.3 基于图的推荐算法

用户对物品的兴趣公式如下：

$$P(i|u)=\sum_{b}P(i|b)p(b|u)$$



(A, a, 1)(A, c, 2)(A, c, 3)
(B, a, 1)(B, b, 3)(B, e, 3)(B, e, 4)
(C, a, 1)(C, b, 2)(C, d, 4)
(D, b, 3)(D, c, 2)(D, c, 3)

$SimpleGraph$的例子

3.4 基于标签的推荐解释

- 用户对标签的兴趣对帮助用户理解为什么给他推荐某个物品更有帮助。
- 用户对标签的兴趣和物品标签相关度对于帮助用户判定自己是否喜欢被推荐物品具有同样的作用。
- 物品标签相关度对于帮助用户判定被推荐物品是否符合他当前的兴趣更有帮助。
- 客观事实类标签相比主观感受类标签对用户更有作用。

4.4. 给用户推荐标签

1. 为什么要给用户推荐标签
   - 方便用户输入标签。
   - 提高标签质量。
2. 如何给用户推荐标签
   - 给用户u推荐整个系统里最热门的标签
   - 是给用户u推荐物品i上最热门的标签
   - 给用户u推荐他自己经常使用的标签
   - 前面两种的融合（**HybridPopularTags**）：该方法通过一个系数将上面的推荐结果线性加权，然后生成最终的推荐结果。**（在将两个列表线性相加时都将两个列表按最大值做了归一化，这样的好处是便于控制两个列表对最终结果的影响，而不至于因为物品非常热门而淹没用户对推荐结果的影响，或者因为用户非常活跃而淹没物品对推荐结果的影响。）**
3. 基于图的标签推荐算法
   图模型同样可以用于标签推荐。在根据用户打标签的行为生成图之后我们可以利用PersonalRank算法进行排名。但这次遇到的问题和之前不同。这次的问题是，当用户u遇到物品i时，会给物品i打什么样的标签。因此，我们可以重新定义顶点的启动概率，如下所示：

$$r_{v(k)} = \begin{cases} \alpha & (v(k) = v(u)) \\ 1-\alpha & (v(k) = v(i)) \\ 0 & \text{其他} \end{cases}$$

   也就是说，只有用户u和物品i对应的顶点有非0的启动概率，而其他顶点的启动概率都为0。在上面的定义中，v(u)和v(i)的启动概率并不相同，v(u)的启动概率是α，而v(i)的启动概率是$1-\alpha$。参数α可以通过离线实验选择
4. 其他

前面提到的基于统计用户常用标签和物品常用标签的算法有一个缺点，就是对新用户或者不热门的物品很难有推荐结果。解决这一问题有两个思路。

第一个思路是从物品的内容数据中抽取关键词作为标签。这方面的研究很多，特别是在上下文广告领域。

第二个思路是针对有结果，但结果不太多的情况。比如《MongoDB权威指南》一书只有一个用户曾经给它打过一个标签nosql，这个时刻可以做一些关键词扩展，加入一些和nosql相关的标签，比如数据库、编程等。实现标签扩展的关键就是计算标签之间的相似度。

作者[钱昊达]

2018年8月22日