《推荐系统实践》读书笔记系列(三)

推荐系统

第三章

3.1 冷启动问题简介

1. 冷启动问题的主要分类
   i. 用户冷启动
   ii. 物品冷启动
   iii. 系统冷启动
2. 解决方案
   i. **提供非个性化的推荐** ：非个性化推荐的最简单例子就是热门排行榜，我们可以给用户推荐热门排行榜，然后等到用户数据收集到一定的时候，再切换为个性化推荐。
   ii. 利用用户注册时提供的年龄、性别等数据做粗粒度的个性化。
   iii. 利用用户的社交网络账号登录（需要用户授权），导入用户在社交网站上的好友信息，然后给用户推荐其好友喜欢的物品。
   iv. 要求用户在登录时对一些物品进行反馈，收集用户对这些物品的兴趣信息，然后给用户推荐那些和这些物品相似的物品。
   v. 对于新加入的物品，可以利用内容信息，将它们推荐给喜欢过和它们相似的物品的用户。
   vi. 在系统冷启动时，可以引入专家的知识，通过一定的高效方式迅速建立起物品的相关度表。

3.2 利用用户注册信息

1. 用户注册信息分类
   i. 人口统计学信息
   ii. 用户兴趣描述

iii. 从其他网站导入的用户站外行为数据

2. 基于注册信息的个性化推荐流程基本如下:

   (1)获取用户注册信息

   (2)根据用户的注册信息对用户进行分类

   (3)给用户推荐他所属分类中用户喜欢的物品

3. 计算每种特征的用户喜欢的物品。

   基于用户注册信息的推荐算法其核心问题是计算每种特征的用户喜欢的物品。

   对于每种特征f，计算具有这种特征的用户对各个物品的喜好程度$p(f,i)$。

   $p(f,i)$可以简单地定义为物品i在具有f的特征的用户中的热门程度：

$$p(f,i) = |N(i) \bigcap U(f)|$$

   其中$N(i)$是喜欢物品i的用户集合,$U(f)$是具有特征f的用户集合。

上面这种定义可以比较准确地预测具有某种特征的用户是否喜欢某个物品。然而往往热门物品$N(i)$较高，因此$p(f,i)$也较高。因此，我们可以将$p(f,i)$定义为喜欢物品i的用户中具有特征f的比例：

$$p(f,i) = \frac{|N(i) \bigcap U(f)|}{|N(i)| + \alpha}$$

这里分母中使用参数α的目的是解决数据稀疏问题。比如有一个物品只被1个用户喜欢过，而这个用户刚好就有特征f，那么就有 $p(f,i) = 1$。但是，这种情况并没有统计意义，因此我们为分母加上一个比较大的数，可以避免这样的物品产生比较大的权重。

3.3 选择适合的物品启动用户的兴趣

1. 简介

   解决用户冷启动问题的另一个方法是在新用户第一次访问推荐系统时，不立即给用户展示推荐结果，而是给用户提供一些物品，让用户反馈他们对这些物品的兴趣，然后根据用户反馈给提供个性化推荐。很多推荐系统采取了这种方式来解决用户冷启动问题。能够用来启动用户兴趣的物品需要具有以下特点：

   i. 比较热门

   ii. 具有代表性和区分性

   iii. 启动物品集合需要有多样性：在冷启动时，我们不知道用户的兴趣，而用户兴趣的可能性非常多，为了匹配多样的兴趣，我们需要提供具有很高覆盖率的启动物品集合，这些

物品能覆盖几乎所有主流的用户兴趣。

如何选择启动物品集合呢？可以用决策树解决这个问题。

首先，给定一群用户，Nadav Golbandi用这群用户对物品评分的方差度量这群用户兴趣的一致程度。如果方差很大，说明这一群用户的兴趣不太一致，反之则说明这群用户的兴趣比较一致。令$\sigma_u \epsilon U'$ 为用户集合U' 中所有评分的方差，Nadav Golbandi的基本思想是通过如下方式度量一个物品的区分度$D(i)$:

$$D(i) = \sigma_{u \epsilon N^+(i)} + \sigma_{N^-(i)} + \sigma_{u \epsilon \bar{N}(i)}$$

其中，$N^+(i)$是喜欢物品i的用户集合,$N^-(i)$是不喜欢物品i的用户集合,$\bar{N}(i)$是没有对物品i评分的用户集合，$\sigma_{u \epsilon N^+(i)}$是喜欢物品i的用户对其他物品评分的方差,$\sigma_{N^-(i)}$是不喜欢物品i的用户对其他物品评分的方差,$\sigma_{u \epsilon \bar{N}(i)}$是没有对物品i评分的用户对其他物品评分的方差。

对于物品i，Nadav Golbandi将用户分成3类——喜欢物品i的用户、不喜欢物品i的用户和不知
道物品i的用户（即没有给i评分的用户）。如果这3类用户集合内的用户对其他的物品兴趣很不一致，说明物品i具有较高的区分度。

- 算法描述

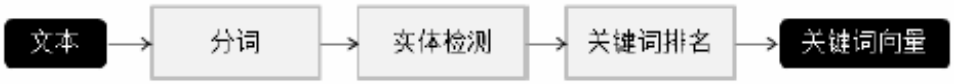
  Nadav Golbandi的算法首先会从所有用户中找到具有最高区分度的物品i，然后将用户分成3类。然后在每类用户中再找到最具区分度的物品，然后将每一类用户又各自分为3类，也就是将总用户分成9类，然后这样继续下去，最终可以通过对一系列物品的看法将用户进行分类。

  在冷启动时，我们从根节点开始询问用户对该节点物品的看法，然后根据用户的选择将用户放到不同的分枝，直到进入最后的叶子节点，此时我们就已经对用户的兴趣有了比较清楚的了解，从而可以开始对用户进行比较准确地个性化推荐。

3.4 利用物品的内容信息

1. ;一般来说，物品的内容可以通过向量空间模型①表示，该模型会将物品表示成一个关键词向量。

   如果物品的内容**实体**，可以直接将这些实体作为关键词。但如果内容是**文本**，则需要引入一些理解自然语言的技术抽取关键词。



关键词向量的生成过程

对于物品d，它的内容表示成一个关键词向量如下：

$$d_i = \{(e_1, w_1), (e_2, w_2)\}$$

其中，e_i就是关键词，w_i是关键词对应的权重。如果物品是文本，我们可以用信息检索领域著名的TF-IDF公式计算词的权重：

$$w_i = \frac{TF(e_i)}{logIDF(e_i)}$$

如果物品是电影，可以根据演员在剧中的重要程度赋予他们权重。向量空间模型的优点是简单，缺点是丢失了一些信息，比如关键词之间的关系信息。不过在绝大多数应用中，向量空间模型对于文本的分类、聚类、相似度计算已经可以给出令人满意的结果。

在给定物品内容的关键词向量后，物品的内容相似度可以通过向量之间的余弦相似度计算：

$$w_{ij} = \frac{d_i \cdot d_j}{\sqrt{|d_i||d_j|}}$$

2. LDA（Latent Dirichlet Allocation）话题模型
   - 基本思想

     一个人在写一篇文档的时候，会首先想这篇文章要讨论哪些话题，然后思考这些话题应该用什么词描述，从而最终用词写成一篇文章。因此，文章和词之间是通过话题联系的。
   - LDA中有3种元素

     文档

     话题

     词语

     每一篇文档都会表现为词的集合，这称为**词袋模型(bag of words)**。

     每个词在一篇文章中属于一个话题。令D为文档集合，$D[i]$是第i篇文档。$w[i][j]$是第i篇文档中的第j个词。$z[i][j]$是第i篇文档中第j个词属于的话题。
   - 算法步骤：

     LDA的计算过程包括**初始化**和**迭代**两部分。

首先要对z进行初始化，而初始化的方法很简单，假设一共有K个话题，那么对第i篇文章中的第j个词，可以随机给它赋予一个话题.同时，用$NWZ(w,z)$记录词w被赋予话题z的次数，$NZD(z,d)$记录文档d中被赋予话题z的词的个数。

```
foreach document i in range(0,|D|):
    foreach word j in range(0, |D(i)|):
        z[i][j] = rand() % K
        NZD[z[i][j], D[i]]++
        NWZ[w[i][j], z[i][j]]++
        NZ[z[i][j]]++
```

在初始化之后，要通过迭代使话题的分布收敛到一个合理的分布上去。

```
while not converged:
    foreach document i in range(0, |D|):
        foreach word j in range(0, |D(i)|):
            NWZ[w[i][j], z[i][j]]--
            NZ[z[i][j]]--
            NZD[z[i][j], D[i]]--
            z[i][j] = SampleTopic()
            NWZ[w[i][j], z[i][j]]++
            NZ[z[i][j]]++
            NZD[z[i][j], D[i]]++
```

LDA可以很好地将词组合成不同的话题。

3.5 发挥专家的作用

为了在推荐系统建立时就让用户得到比较好的体验，很多系统都利用专家进行标注。

作者[钱昊达]
2018年8月19日