算法概要

1. 构造 文章-标签 的 0-1 矩阵
2. 构造 用户-文章 评分矩阵
3. 计算相似度（余弦相似性）
4. 推荐

变量说明

- $users[i=1:S]$：用户集
- $articles[j=1:M]$：文章集
- $features[k=1:N]$：已经得到的特征集
- MA_j：向量，$1*N$，文章 j 特征向量，初始为全0
  - $MA_{j,k}$：表示文章 $articles[j]$ 在特征 $features[k]$ 的值
- MP_i：向量，$1*M$，用户 i 评分向量，初始为全0
  - $MP_{i,j}$：表示用户 $users[i]$ 对文章 $articles[j]$ 的评分
- MU_i：向量，$1*N$，用户 i 特征向量，初始为全0
  - $MU_{i,k}$：表示用户 $users[i]$ 在特征 $features[k]$ 的值

预处理 文章特征01矩阵

```
for all articles j = 1 to M
    for all features k = 1 to N
        if (articles[j] 拥有 features[k] 属性)
            MA_{j}[k] = 1;
        else
            MA_{j}[k] = 0;
```

构造 用户-文章评分矩阵

- 对于某一个用户 $users[i]$，在阅读某篇文章 $articles[j]$ 之后，评分为 $score_{i,j}$，则 $MP_{i,j}=score_{i,j}$
- $score_{i,j}$ 的计算如下：

$$score_{i,j}=\frac{readtime}{words}$$

其中 $readtime$ 为阅读时间，$words$ 为文章的字数

计算相似度

对于某一个用户 $users[i]$：

- 计算 $users[i]$ 所有评分的均值：

$$Avg_i=\frac{\sum_{j\in Scored}MP_{i,j}}{|Scored|}$$

其中 $Scored$ 为用户 $users[i]$ 已评分的文章集

- 计算用户 $users[i]$ 对 $features[k]$ 的喜好程度

$$MU_{i,k} = \frac{\sum(x_k - Avg_i)}{n}$$

这里，x_k 为所有包含 $features[k]$ 且用户 $users[i]$ 已评过分的文章的评分，n 为所有包含 $features[k]$ 的文章的数量
至此，对于用户 $users[i]$，得到了一个 $1 * N$ 的向量 MU_i

- 计算 $users[i]$ 和 $article[j]$ 的相似度

$$\cos(i,j) = \frac{\sum(MU_{i,k} * MA_{j,k})}{\sqrt{\sum MU_{i,k}^2} * \sqrt{\sum MA_{j,k}^2}}$$

推荐

对于用户 $users[i]$ ，遍历整个文章集，计算 $users[i]$ 和每个文章的相似度（推荐度），选择相似度最高的前若干个文章，推荐给用户 $users[i]$

算法说明

- 预处理所有文章的特征01矩阵
- 在用户 $users[i]$ 注册后，该用户的 MU_i 被初始化为全0
- 在用户 $users[i]$ 需要获取文章时，运用上述【算法概要】中【推荐】的做法，选择若干篇文章推荐给用户
- 在用户 $users[i]$ 阅读文章时，获取参数【阅读时间】和【文章字数】。在阅读完文章时，根据参数依次处理：
  - 更新用户评分向量 MP_i
  - 更新用户特征向量 MU_i