WebHW2

PB18000268 曾勇程

1.计算题

PB18000268 曾勇程 Web HW2

一. 计算题

1.1 (1) $w_{t,d} = wf_{t,d} \cdot \log_{10}\frac{N}{df_t}$ $N = 811400$

$$wf_{t,d} = \begin{cases} 1+\log_{10} tf_{t,d} & tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Doc1 中: 对于 Car: $w_{t,d} = (1+\log_{10}34)\cdot\log_{10}\frac{811400}{18871} = 4.135$

Auto: $w_{t,d} = (1+\log_{10}3)\cdot\log_{10}\frac{811400}{3597} = 3.476$

Insurance: $\because tf_{t,d} = 0 \quad \therefore wf_{t,d} = 0$

Best: $wf_{t,d} = (1+\log_{10}181)\cdot\log_{10}\frac{811400}{40014} = 2.948$

按同样的方法计算其他的 tf-idf 值.

$\therefore$

	tf-idf @Doc1	tf-idf @Doc2	tf-idf @Doc3
Car	4.135	3.109	4.092
Auto	3.476	5.601	0.000
Insurance	0.000	4.404	2.892
Best	2.948	0.000	2.763

(2) Doc1 对应的向量 $\vec{v_1} = (4.135, 3.476, 0, 2.948)$

0向量为: $\vec{0} = (0, 0, 0, 0)$

欧式距离: $dist(\vec{v_1}, \vec{0}) = \sqrt{4.135^2 + 3.476^2 + 2.948^2}$

≈ 6.154

采用欧式为一化方法，处理后的向量代表为:

$\vec{v_{n1}} = \frac{1}{dist(\vec{v_1}, \vec{0})} \cdot \vec{v_1}$

$= (0.672, 0.565, 0, 0.479)$

Doc2: $\vec{v_2} = (3.109, 5.601, 4.404, 0)$

$\vec{v_{n2}} = \frac{\vec{v_2}}{dist(\vec{v_2}, \vec{0})} = (0.400, 0.720, 0.567, 0)$

Doc3: $\vec{v_3} = (4.092, 0, 2.892, 2.763)$

$\vec{v_{n3}} = \frac{\vec{v_3}}{dist(\vec{v_3}, \vec{0})} = (0.715, 0, 0.505, 0.483)$

处理后的向量表示为:

Doc1: $\vec{v_{n1}} = (0.672, 0.565, 0, 0.479)$

Doc2: $\vec{v_{n2}} = (0.400, 0.720, 0.567, 0)$

Doc3: $\vec{v_{n3}} = (0.715, 0, 0.505, 0.483)$

(3) 查询 "car insurance"

向量表示：$\vec{Q} = (1, 0, 1, 0)$

$$\cos(\vec{q}, \vec{d}) = \frac{\sum_{i=1}^{|V|} q_i d_i}{dist(\vec{q}, \vec{0}) \cdot dist(\vec{d}, \vec{0})}$$

处理后的向量表示为：

Doc1： $\vec{V_{n1}} = (0.672, 0.565, 0, 0.479)$

Doc2： $\vec{V_{n2}} = (0.400, 0.720, 0.567, 0)$

Doc3： $\vec{V_{n3}} = (0.715, 0, 0.505, 0.483)$

$$Cos(\vec{Q}, \vec{V_{n1}}) = \frac{1 \times 0.672 + 1 \times 0}{\sqrt{2}} = 0.475$$

$$Cos(\vec{Q}, \vec{V_{n2}}) = \frac{1 \times 0.400 + 0.567}{\sqrt{2}} = 0.684$$

$$cos(\vec{Q}, \vec{V_{n3}}) = \frac{0.715 + 0.505}{\sqrt{2}} = 0.863$$

$$0.863 > 0.684 > 0.475$$

∴对于查询 "car insurance"，Doc3的得分(相似度)大于Doc2的得分(相似度)大于Doc1的得分(相似度)

1.2 1)

跳转矩阵：

$$M=R=\begin{pmatrix} 0 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/2 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1/2 & 1/3 \end{pmatrix}$$

∵随机跳转概率为 $0.15 = 1-d \Rightarrow d = 0.85$

$$dM=\begin{pmatrix} 0 & 0 & 17/60 & 0 & 0 & 0 & 0 \\ 0 & 17/40 & 0 & 0 & 6 & 0 & 0 \\ 0.85 & 17/40 & 17/60 & 0 & 0 & 0 & 0 \\ 0 & 0 & 17/60 & 17/40 & 0 & 0 & 17/60 \\ 0 & 0 & 0 & 17/40 & 0 & 0 & 17/60 \\ 0 & 0 & 0 & 0 & 0 & 17/40 & 0 \\ 0 & 0 & 0 & 0 & 0.85 & 17/40 & 17/60 \end{pmatrix}$$

∴(随机)转移概率矩阵 $A=$

$$\begin{pmatrix} \frac{3}{140} & \frac{3}{140} & \frac{32}{105} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} \\ \frac{3}{140} & \frac{25}{56} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} \\ \frac{61}{70} & \frac{25}{56} & \frac{32}{105} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} \\ \frac{3}{140} & \frac{3}{140} & \frac{32}{105} & \frac{25}{56} & \frac{3}{140} & \frac{3}{140} & \frac{32}{105} \\ \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{25}{56} & \frac{3}{140} & \frac{3}{140} & \frac{32}{105} \\ \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{25}{56} & \frac{3}{140} \\ \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{3}{140} & \frac{61}{70} & \frac{25}{56} & \frac{32}{105} \end{pmatrix}$$

2) 求该Markov链的平稳分布 $\vec{\pi}=(\pi_0,\pi_1,\pi_2,\pi_3,\pi_4,\pi_5,\pi_6)^T$

∴ 该Markov链处于稳态时有

$$\begin{cases}\vec{\pi}=A\vec{\pi}\\ \sum_{i=0}^{6}\pi_i=1\\ \pi_i\geq 0,\ 0\leq i\leq 6\end{cases}$$

解得：$\vec{\pi}=(0.055, 0.037, 0.117, 0.244, 0.211, 0.037, 0.302)^T$

即该矩阵所对应的PageRank向量为 $(0.055, 0.037, 0.117, 0.244, 0.211, 0.037, 0.302)^T$

3)将邻接矩阵中"节点 2 指向节点 3"和"节点 6 指向节点 3"的两条边权重设为 0，其它不 变。请计算该矩阵所对应的 Hub 值和 Authority 值向量。

答：法一：手算法：

$$
\begin{aligned}
&\text{由} \quad \vec{h} \leftarrow M\vec{a} \\
&\quad\;\; \vec{a} \leftarrow M^T\vec{h} \\
&\text{得到} \quad \vec{h} \leftarrow MM^T\vec{h} \\
&\quad\;\; \vec{a} \leftarrow M^TM\vec{a} \\
&\text{将} \leftarrow \text{替换成} = \text{号，得} \vec{h} = \frac{1}{\lambda_h} MM^T\vec{h} \\
&\quad\;\; \vec{a} = \frac{1}{\lambda_a} M^TM\vec{a} \\
&\text{其中，} \lambda_h \text{和} \lambda_a \text{分别为矩阵} MM^T \text{和} M^TM \text{的特征值}
\end{aligned}
$$

因此，计算过程为：

1）得到邻接矩阵M，计算MM^T和M^TM;

2）计算MM^T和M^TM的特征值，从而求得他们的主特征向量，从而得到最后的Hub值向量和Authority 值向量。

[但由于本题7阶矩阵太大，即使它可以拆成一个3阶矩阵和一个4阶矩阵，但4阶矩阵特征值的精确值也求不出来，故采用下面的第二种方法：编程法]

法二：编程法：

邻接矩阵M为：

0	**0**	**1**	**0**	**0**	**0**	**0**
0	1	1	0	0	0	0
1	0	1	0	0	0	0
0	0	0	1	1	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	1
0	0	0	0	1	0	1

$$
\begin{aligned}
&\text{计算方法：} a_{k+1} = M^T h_k \\
&\quad\;\; h_{k+1} = M a_{k+1} \\
&\quad\;\; a_0 = h_0 = (1,1,1,1,1,1,1)^T
\end{aligned}
$$

代码如下：

```
import numpy as np
m = [
 [0, 0, 1, 0, 0, 0, 0],
 [0, 1, 1, 0, 0, 0, 0],
 [1, 0, 1, 0, 0, 0, 0],
 [0, 0, 0, 1, 1, 0, 0],
 [0, 0, 0, 0, 0, 0, 1],
 [0, 0, 0, 0, 0, 1, 1],
 [0, 0, 0, 0, 1, 0, 1],
]
h = [[1], [1], [1], [1], [1], [1], [1]]
a = [[1], [1], [1], [1], [1], [1], [1]]
```

```
M = np.array(m)
H = np.array(h)
A = np.array(a)
for i in range(10000):
    A = np.matmul(np.transpose(M), H) # 根据公式迭代计算 Authority 值向量
    Len_A = np.sqrt((A*A).sum())
    A = A/Len_A                       # 归一化
    H = np.matmul(M, A)               # 根据公式迭代计算 Hub 值向量
    Len_H = np.sqrt((H*H).sum())
    H = H/Len_H                       # 归一化
print("Authority:", A)
print("Hub:", H)
```

最终的迭代10000次后的结果为：

```
Authority: [[3.95252517e-323]
 [3.95252517e-323]
 [1.03753786e-322]
 [1.68457870e-001]
 [4.98011193e-001]
 [2.72570559e-001]
 [8.05799037e-001]]
Hub: [[5.43472210e-323]
 [7.41098469e-323]
 [7.41098469e-323]
 [3.35070080e-001]
 [4.05118802e-001]
 [5.42154779e-001]
 [6.55495991e-001]]

Process finished with exit code 0
```

即$Authority = (0, 0, 0, 0.168, 0.498, 0.273, 0.806)$

$Hub = (0, 0, 0, 0.335, 0.405, 0.542, 0.655)$

1.3 a): P@10 = 4/10 = 2/5

P@20 = 4/20

b) $F_1 = \frac{2PR}{P+R}$ R@10 = 4/10 = 2/5 R@20 = 4/10

前10篇: $F_1 = \frac{2\times\frac{2}{5}\times\frac{2}{5}}{\frac{2}{5}+\frac{2}{5}} = \frac{2}{5}$

前20篇: $F_1 = \frac{2\times\frac{4}{20}\times\frac{4}{20}}{\frac{4}{20}+\frac{4}{20}} = \frac{4}{20}$

c) 简易AP = (1/1 + 2/2 + 3/5 + 4/9 + 5/11 + 6/15 + 7/20)/7

= 0.607

d) 最大情况下，第21, 22, 23篇文档为相关文档.

则最大简易AP = $\frac{(1/1 + 2/2 + 3/5 + 4/9 + 5/11 + 6/15 + 7/20 + 8/21 + 9/22 + 10/23)}{10}$

= 0.547

e) 最小情况下，第9998, 9999, 10000篇文档为相关文档.

最小简易AP = $\frac{1/1 + 2/2 + 3/5 + 4/9 + 5/11 + 6/15 + 7/20 + 8/9998 + \frac{9}{9999} + \frac{10}{10000}}{10}$

= 0.625

2.问答题

2.1

请简述解决以下问题的思路：

a) 如何从多源情境信息（如手机的多种传感器信息）中，抽象出用户当前所处的状态或行为模式?

答：例如，可以根据**感知查询的上下文**(搜索历史)、查询时的**环境信息**，判断用户所处的状态（真实意图）。

现如今，随着移动终端的发展（手机、平板、笔记本等），移动情境数据的感知和收集逐渐依赖于这些移动终端的多种传感器，如GPS、Wi-Fi、蓝牙、麦克风等系统。因此可以利用这些感知器抽象出用户当前所处的状态或行为模式。例如

1. **显著地点和轨迹挖掘**：
    - **显著地点挖掘**：利用GPS传感器，挖掘用户的个性化显著地点（个人生活基本地点：家或单位等）和根据大众记录挖掘特定公共区域（旅游名胜或商业中心等）。根据室内无GPS信号这一特点，利用GPS信号消失或重现的规律，结合已有的人工标注和时间的**情境信息**，标注用户所处的个性化显著地点。而基于多用户轨迹序列，和GPS标注，推理出用户所处的特定公共区域。
    - **轨迹模式挖掘**：利用GPS感知器和第三方数据源（如谷歌地图等），结合每个地点的相关语义信息，推理出用户的行为模式，例如获得3个停顿地点分别表示"购物广场→餐厅→电影院"结合时间、语义可理解为用户正在周末度假。
2. **用户行为模式挖掘**（揭示了用户的生活规律和个人偏好的基础性信息）：
    - **关联规则挖掘**：针对**多用户情境日志**挖掘公共的行为规律，再将这些公共规律映射到单个用户上，例如，下午在咖啡厅浏览社交网站（也可根据Wi-Fi、蓝牙等传感器推理出用户每天所处的位置和所干的事）。
    - **序列模式挖掘**：在**序列化的数据库**中发现**频繁子序列**（时间序列模式），分析和预测用户的当前和下一步行为变化趋势。

b) 在上述过程中，如何既体现用户的个性化因素，又减少用户个人记录稀疏的负面影响?

答:一种折中的方案是，以主题（行为的共同规律）为中介，基于**多用户情境日志**，提取出公共的行为规律，再将这些公共规律映射到单个用户的个性化因素上。或者说是聚类。类中心体现共性，类之间的差异体现个性。例如，基于大众的评价和建议，结合用户的个性化因素，从数据库中提取出相关度比较高的主题（行为的共同规律），从而对该用户给出特定的推荐。

2.2

用户在浏览网页时，可能通过点击“后退”按钮回到上一次浏览的页面。用户的这种回退行为（包括连续回退行为）能否用马尔科夫链进行建模？为什么?

答：不能，用户的回退行为不能被建模为马尔科夫链。这是因为用户的回退行为在语义上为释放了到达那个点(网页)的路径，并且连续的回退行为要涉及到之前的若干状态，而这不是马尔科夫过程的。

2.3

如何在网页排序的同时提升结果的多样化水平？如何在实现这一目的的同时保障算法的效率?

答：多样化排序的过程总体上可以看作一个“顺序选择”的过程。 采用MMR(最大边界相关性)方法：采取贪心策略，生成 `top K` 结果列表。第一次，先选取相关度最高的物品。然后，每次选取和查询 query 匹配度高、和已经选取的物品最大相似度低的物品。可以定义一个新的排序函数，它被定义为**文档**与**查询**的**相关性**得分和在给定已选择文档的情况下当前文档的**新颖性**得分的总和。并依赖于这个排序函数进行多样化排序，以提升结果的多样化水平。

保障算法的效率：可以考虑采用贪心算法，比如首先生成 `top K` 结果列表，选取与查询相关度最高的网页，然后更新迭代排序函数(这里可以当作是每个文档的属性)的值，再次贪心选择和查询匹配度高、和已经选取的网页最大相似度低的网页。