**Chp.6** **Ranking**

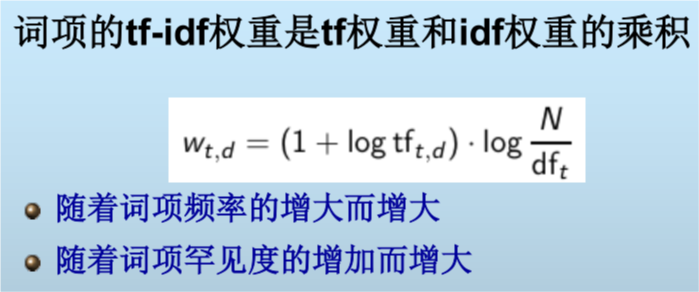
Hw4

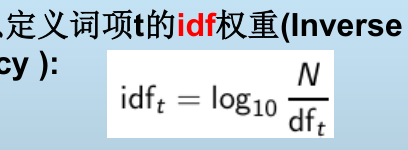
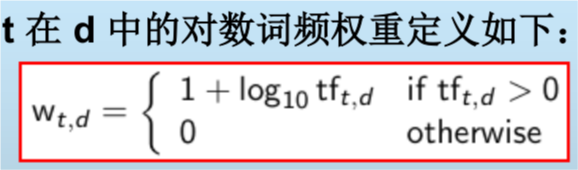
# PB16080377聂雷海

1. **假定已知文档d1和d2和查询q的词项以及词频如下：  
   d1：(<2010,2>,<世博会,3>,<中国,2>,<举行,1>)   
   d2：(<2005,1>,<世博会,2>,<1970,1>,<日本,2>,<举行,1>)   
   q：(<2010,1>,<世博会,2>)   
   请给出文档d1、d2以及查询q的基于tf-idf权值的向量表示，然后分别计算q和d1、d2的余弦相似度，并说明q和哪个文档更相关。**

Answer:

|  |  |  |  |
| --- | --- | --- | --- |
|  | d1 | d2 | q |
| 2010 | 2 | 0 | 1 |
| 1970 | 0 | 1 | 0 |
| 2005 | 0 | 1 | 0 |
| 世博会 | 3 | 2 | 2 |
| 中国 | 2 | 0 | 0 |
| 举行 | 1 | 1 | 0 |
| 日本 | 0 | 2 | 0 |



  
由上图公式，先计算idft(dft取出现该词项的文档数)

|  |  |  |
| --- | --- | --- |
|  | df \_t | idf\_t |
| 2010 | 1 | 0.30103 |
| 1970 | 1 | 0.30103 |
| 2005 | 1 | 0.30103 |
| 世博会 | 2 | 0 |
| 中国 | 1 | 0.30103 |
| 举行 | 2 | 0 |
| 日本 | 1 | 0.30103 |

再计算出tf-idf权值表

|  |  |  |  |
| --- | --- | --- | --- |
|  | d1 | d2 | q |
| 2010 | 0.391649 | 0 | 0.30103 |
| 1970 | 0 | 0.30103 | 0 |
| 2005 | 0 | 0.30103 | 0 |
| 世博会 | 0 | 0 | 0 |
| 中国 | 0.391649 | 0 | 0 |
| 举行 | 0 | 0 | 0 |
| 日本 | 0 | 0.391649 | 0 |

则按文档表示为向量（取4位小数）：

d1=(0.3916,0,0,0,0.3916,0,0)

d2=(0,0.3010,0.3010,0,0,0,0.3916)

q=(0.3010,0,0,0,0,0,0)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | d1 | d2 | q | q\*d1 | q\*d2 | d1\*d1 | d2\*d2 |
| 2010 | 0.3916 | 0.0000 | 0.3010 | 0.1179 | 0.0000 | 0.153389 | 0 |
| 1970 | 0.0000 | 0.3010 | 0.0000 | 0.0000 | 0.0000 | 0 | 0.090619 |
| 2005 | 0.0000 | 0.3010 | 0.0000 | 0.0000 | 0.0000 | 0 | 0.090619 |
| 世博会 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0 | 0 |
| 中国 | 0.3916 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.153389 | 0 |
| 举行 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0 | 0 |
| 日本 | 0.0000 | 0.3916 | 0.0000 | 0.0000 | 0.0000 | 0 | 0.153389 |
| Sum |  |  |  | 0.1179 | 0.0000 |  |  |
| Cosine |  |  |  | 0.3680 | 0 |  |  |

Cosine(q,d1)=0.3680 Cosine(q,d2)=0

结论：查询q与文档d1更相关

1. **基于tf-idf的相关度计算方法有什么缺点？请给出两点以上，并加以解释。**

Answer:

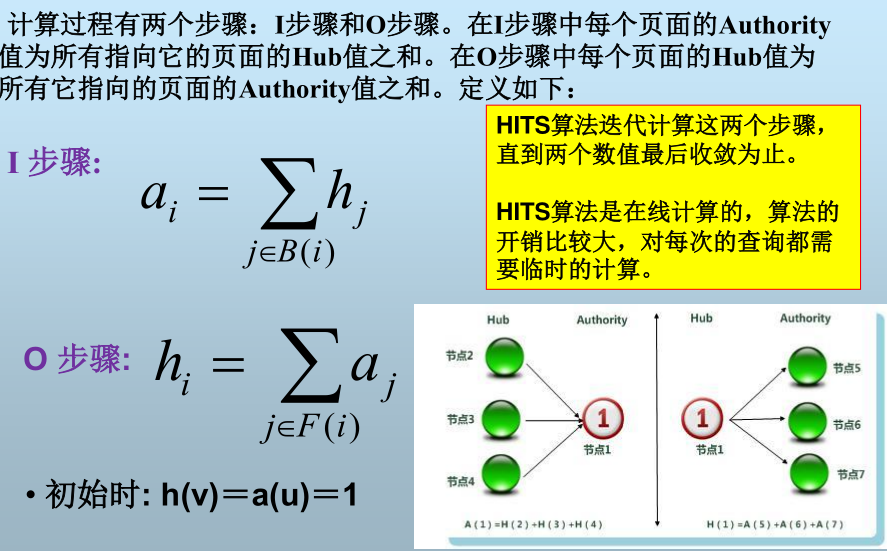
①无法描述词项关系。因为tf-idf的相关度计算是对单一词项的处理结果，计算步骤都是主要是考虑该词项在不同文档以及查询向量中的频率和信息，没有计算过词与词之间的绑定关系。

②查询关键字不符合向量空间模型的假设，即关键字并不一定相互独立。

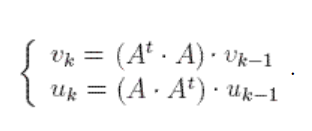
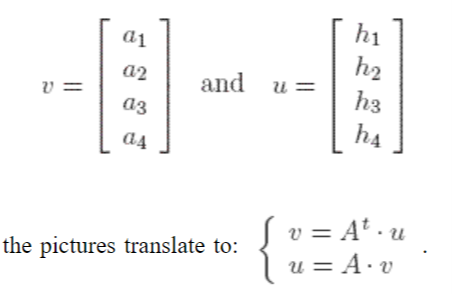
因为关键字间很可能存在相关的关系，比如“王励勤”和“乒乓球”。

1. **在微博平台上每天都会出现一些热门微博和活跃用户，假设我们借鉴HITS算法的思想来实时检测热门微博和活跃用户，应该如何实现？请给出基本的算法思路，并给出算法伪码（须有适当注释）。**

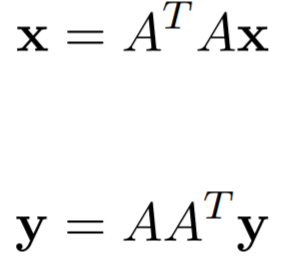
Answer:



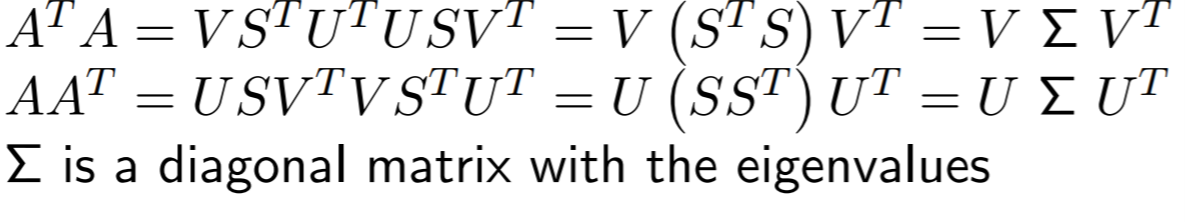
参考HITS算法,由于微博和微博之间、用户和用户之间的关系难以作为衡量是否活跃与热门的指标。因此我们可以假定，在I步骤微博的authority值为参与该微博的用户的hub值之和，在O步骤每个用户的hub值为它所参与的微博的authority值之和。



由上图运算，我们可以得到下图



其中x,y分别为在某个子图中对应的一个节点的authority和hub值



对A奇异值分解，可以提高运算速度

算法思路： 已知子图的邻接矩阵，来表示微博和用户的关系。初始化x,y向量。然后迭代计算x=A­TAx,y=AATy，当两个向量达到迭代次数，即迭代结束。然后在x,y向量中分别排序，得到各自排名靠前的下标，即得到热门微博和活跃用户。

Source code有一些不精确的用法.鉴于是伪代码，因此未深究。

algorithm:

input : adjacency matrix A of n\*m and number of iterations h.

​       name\_1 is a string array about 微博名字. name\_2 is a string

​       array about user\_name.

output:

​ authority and hub score vectors x and y respectively

HITS\_algorithms(A,h,name\_1,name\_2):

​   create x[],y[]

//  x[] is a m-dimension vector, y is a n-dimension vector

​   initial(x)

​   initial(y)

//then  x=(1,1,...,1)$\in R^{m}$ y=(1,1,...,1)$\in R^{n}$

​   create C,D

//  C is m\\*m dimension matrix, D is n\\*n dimension matrix.

​   $C = A^{T}A$

​   $D=AA^{T}$

//x=Cx,y=Dy . as we said before.

   while h--

​       for i = 1..m do

​           for j = 1..m do

​               x[i] += C[i][j]\*x[j]

​       for i =1..n do

​           for j =1..n do

​               y[i] +=D [ i ][ j ]\*y[j]

​   map <int,string> weibo(y,name\_1),users(x,name\_2)

// we can sort the array and get the element’s name.

​   sort(x[1..m])

​   sort(y[1..n])

​   for i = 1.. m do

​       print users[i].second

​   for j=1 .. n do

​       print weibo[j].second

​   return x,y