现代信息检索
Modern Information Retrieval

第18讲 隐性语义索引
Latent Semantic Indexing

授课人：王斌

http://ir.ict.ac.cn/~wangbin

*改编自"An introduction to Information retrieval"网上公开的课件，地址 http://nlp.stanford.edu/IR-book/

提纲

❶ 上一讲回顾

❷ 隐性语义索引

❸ 空间降维处理

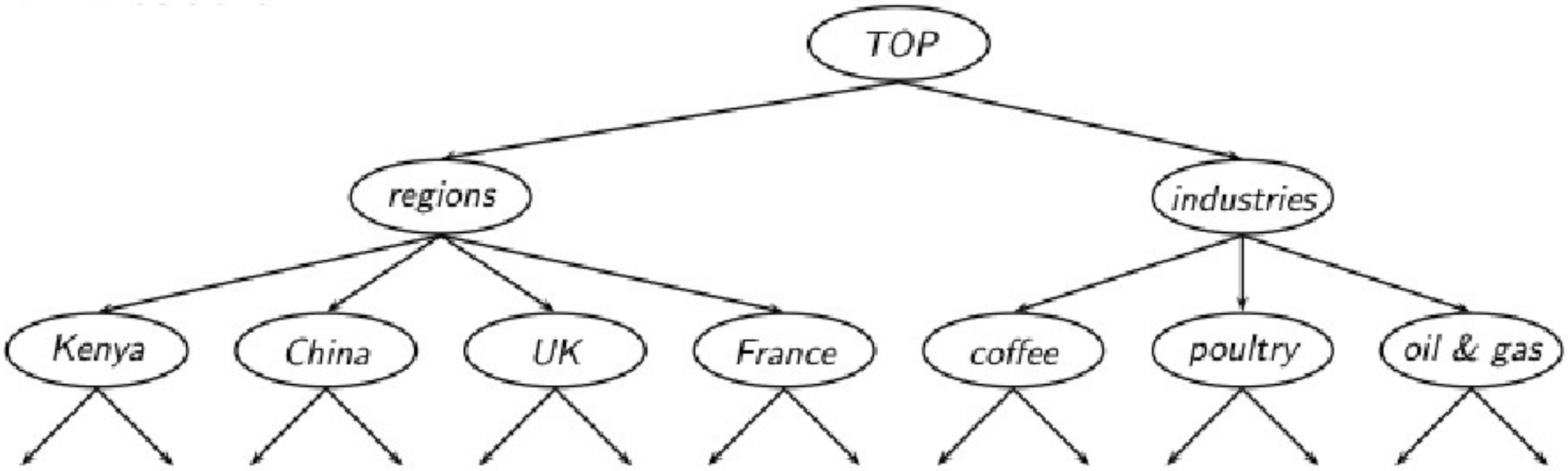
❹ LSI 在IR中的应用

提纲

1. 上一讲回顾
2. 隐性语义索引
3. 空间降维处理
4. LSI 在IR中的应用

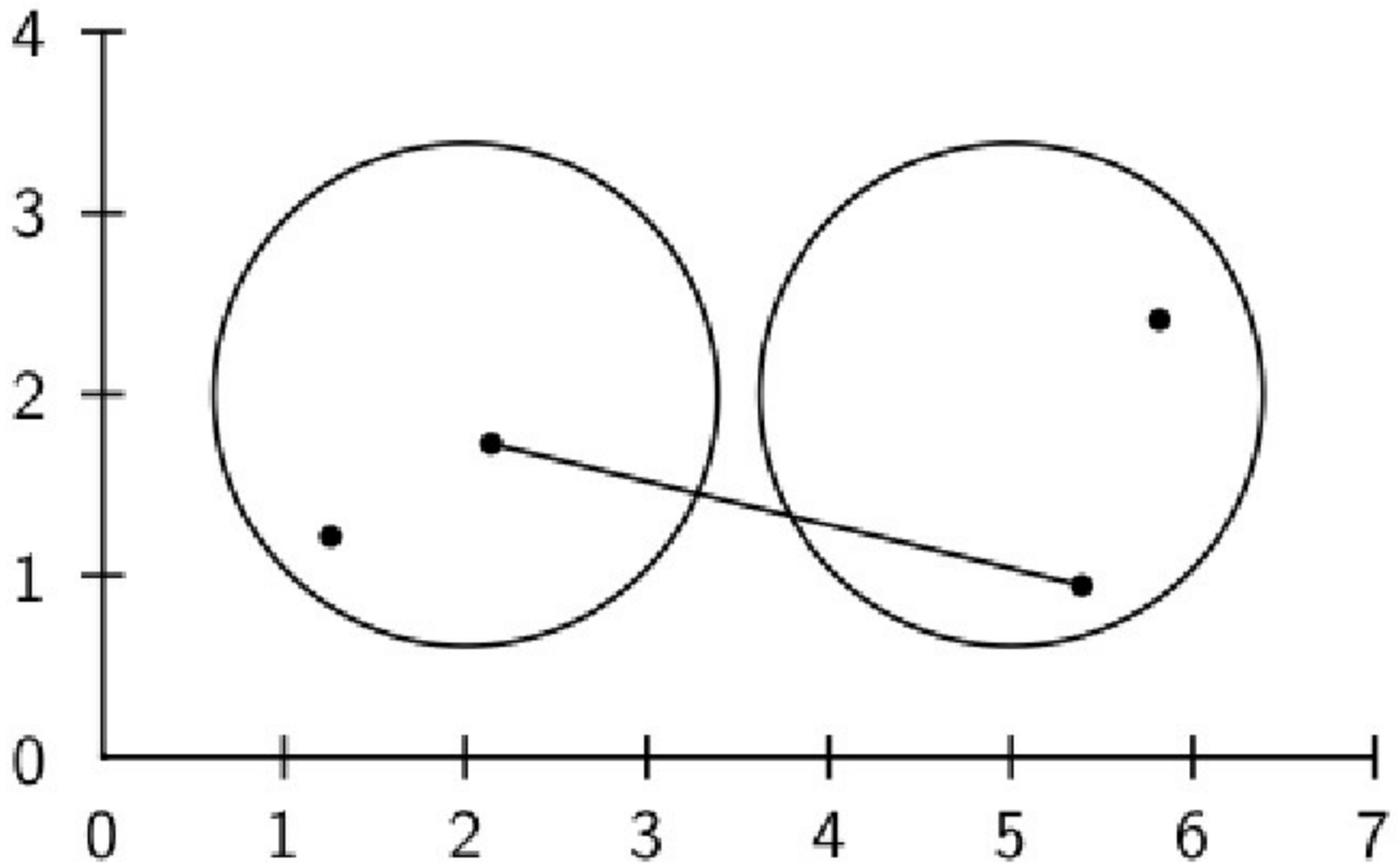
层次聚类

层次聚类的目标是生成类似于前面提到的Reuters目录的一个层次结构：

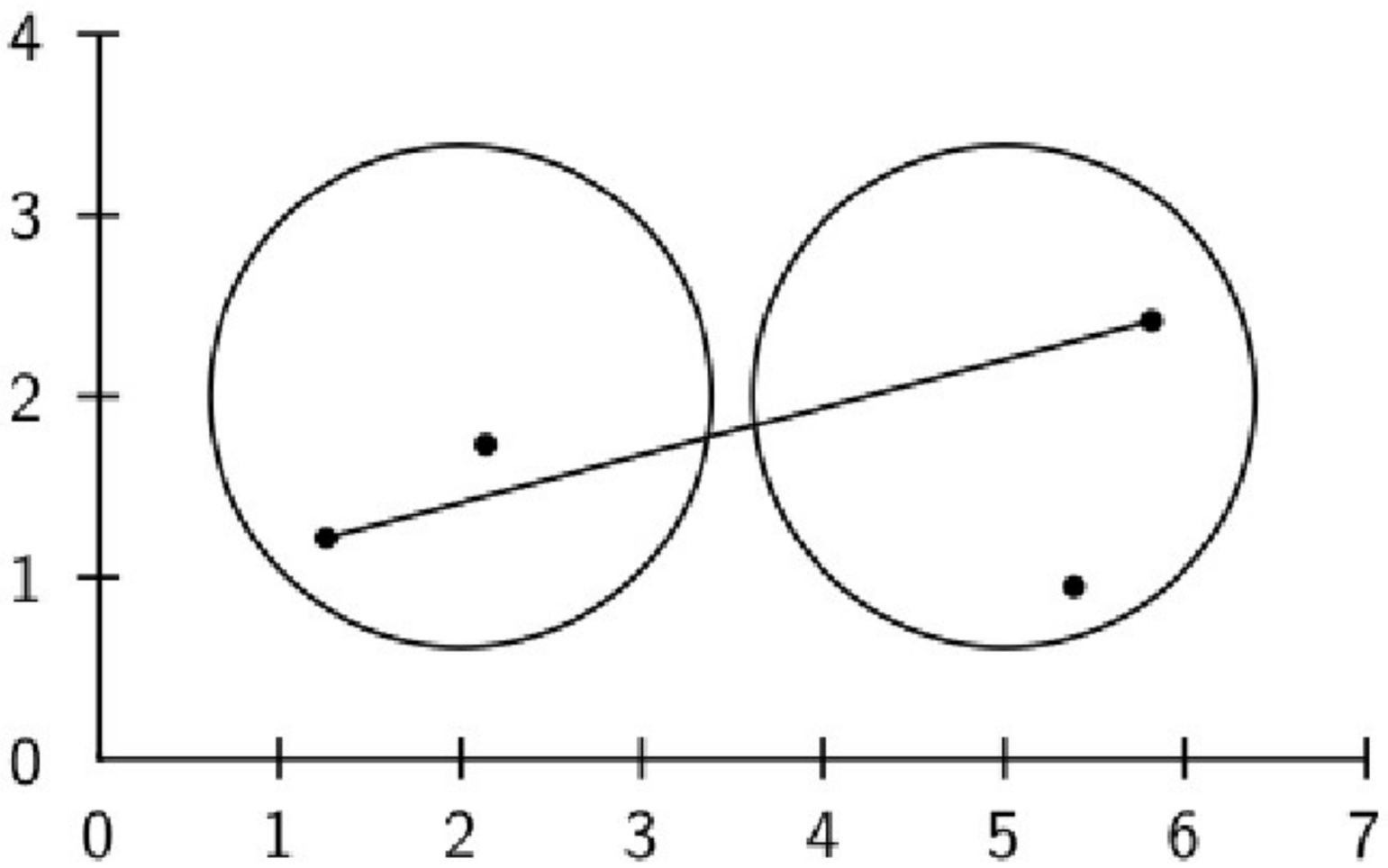


这个层次结构是自动创建的，可以通过自顶向下或自底向上的方法来实现。最著名的自底向上的方法是层次凝聚式聚类(hierarchical agglomerative clustering，HAC)。

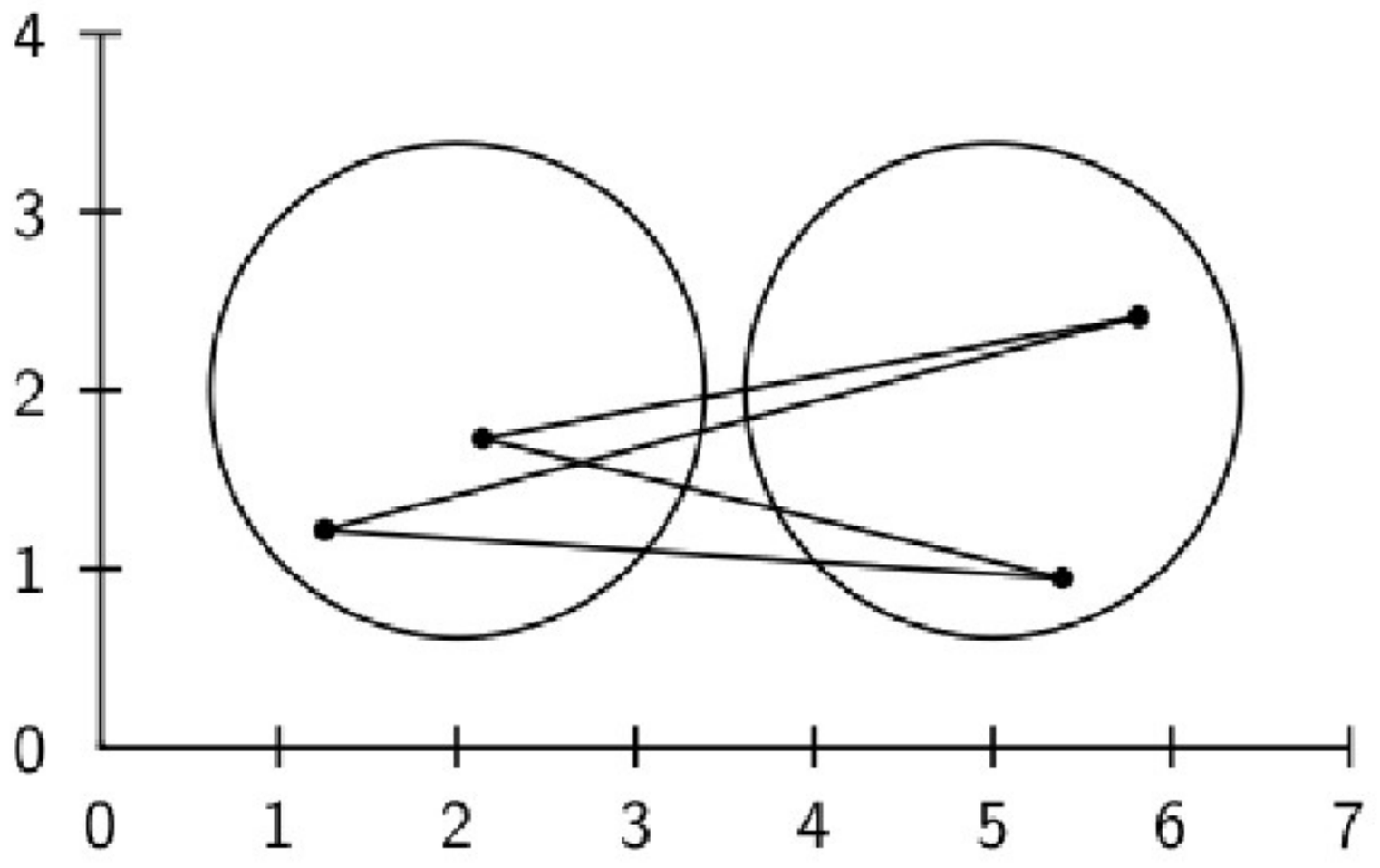
单连接: 最大相似度(最短距离)



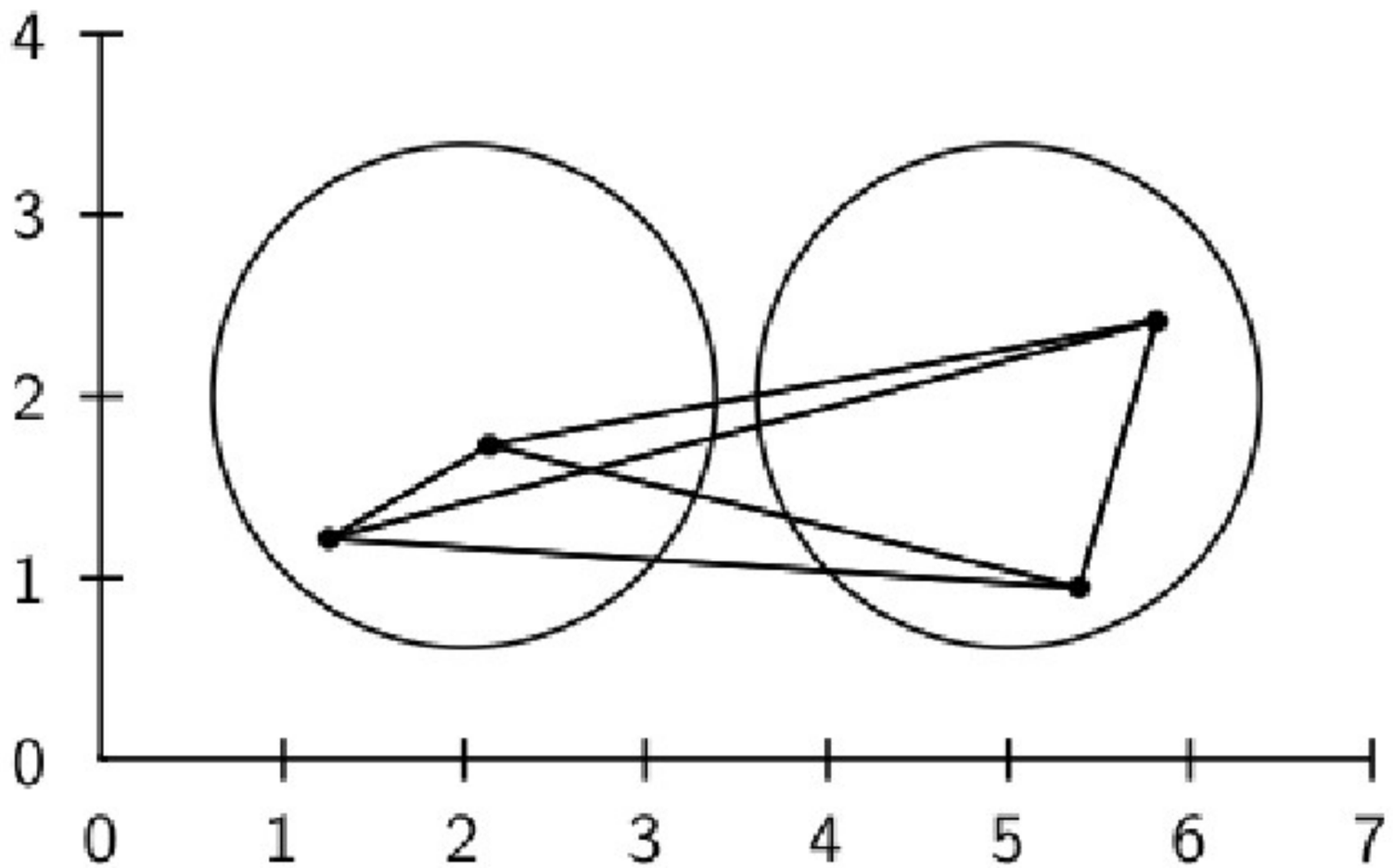
全连接: 最小相似度



质心法



组平均



四种HAC算法的比较

方　　法	结合相似度	时间复杂度	是否最优？	注　　释
单连接	簇间文档的最大相似度	$\Theta(N^2)$	yes	链化效应
全连接	簇间文档的最小相似度	$\Theta(N^2 \log N)$	no	对离群点敏感
组平均	所有文档相似度的平均值	$\Theta(N^2 \log N)$	no	大部分应用中的最佳选择
质心法	所有簇间相似度的平均值	$\Theta(N^2 \log N)$	no	相似度颠倒

簇标签生成的例子

	文档数目	簇标签生成方法		
		质心	互信息	标题
4	622	oil plant mexico production crude **power 000 refinery gas** bpd	plant oil production **barrels** crude bpd mexico **dolly capacity petroleum**	MEXICO: Hurricane Dolly heads for Mexico coast
9	1017	police security **russian** people military peace killed told **grozny court**	police killed military security peace told **troops forces rebels** people	RUSSIA: Russia's Lebed meets rebel chief in Chechnya
10	1259	00 000 tonnes traders futures wheat prices **cents september** tonne	**delivery** traders futures tonne tonnes **desk** wheat prices 000 00	USA: Export Business - Grain/oilseeds complex

- 三种方法：选择质心向量中的突出词项，使用MI的差别式标签，使用离质心最近的文档的标题
- 三种方法的结果都不错

本讲内容

- 矩阵SVD分解
- 隐性语义索引LSI(Latent Semantic Indexing)
- LSI在IR中的应用

提纲

1. 上一讲回顾
2. 隐性语义索引
3. 空间降维处理
4. LSI 在IR中的应用

回顾一下词项-文档矩阵

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
anthony	5.25	3.18	0.0	0.0	0.0	0.35
brutus	1.21	6.10	0.0	1.0	0.0	0.0
caesar	8.59	2.54	0.0	1.51	0.25	0.0
calpurnia	0.0	1.54	0.0	0.0	0.0	0.0
cleopatra	2.85	0.0	0.0	0.0	0.0	0.0
mercy	1.51	0.0	1.90	0.12	5.25	0.88
worser	1.37	0.0	0.11	4.15	0.25	1.95

. . .

该矩阵是计算文档和查询相似度的基础，接下来我们要介绍，能否通过对该矩阵进行转换来获得文档和查询之间的一个更好的相似度计算方法？

隐性语义索引LSI简介

- 我们将词项-文档矩阵转换成多个矩阵的乘积
- 这里我们使用的是一个特定的分解方法： 奇异值分解(singular value decomposition， 简称SVD)
- SVD: $C = U\Sigma V^T$ (其中 C = 词项-文档矩阵)
- 利用SVD分解的结果我们来构造一个新的、改进的词项-文档矩阵 C'
- 通过C' 我们可以得到一个更好的相似度计算方法(相对于 C而言)
- 为了这种目的使用SVD被称为隐性语义索引(latent semantic indexing)或者简称 LSI。

例子$C = U\Sigma V^T$：矩阵C

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

上面给出了一个标准的词项-文档矩阵，为简单起见，这里使用了布尔矩阵。

例子 $C = U\Sigma V^T$: 矩阵U

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

每个词项对应一行，每个$min(M,N)$对应一列，其中M为词项的数目，N是文档的数目。 这是个正交矩阵：

(i) 列向量都是单位向量；

(ii) 任意两个列向量之间都是互相正交的。可以想象这些列向量分别代表不同的“语义”维度，比如政治、体育、经济等主题。矩阵元素 u_{ij} 给出的是词项i和第j个“语义”维度之间的关系强弱程度。

例子 $C = U\Sigma V^T$: 矩阵 Σ

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

这是个$\min(M,N) \times \min(M,N)$的对角方阵。对角线上是矩阵C的奇异值。奇异值的大小度量的是相应“语义”维度的重要性。我们可以通过忽略较小的值来忽略对应的“语义”维度

例子$C = U\Sigma V^T$：矩阵V^T

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

每篇文档对应一列，每 $\min(M,N)$ 对应一行。同样，这也是一个正交矩阵：

(i) 每个行向量都是单位向量；

(ii) 任意两个行向量之间都是正交的；

同样每个行向量代表的是一个语义维度，矩阵元素v_{ij}代表的是文档 i 和语义维度j的关系强弱程度

例子 $C = U\Sigma V^T$：所有的四个矩阵

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

=

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

×

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

×

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

LSI: 小结

- 词项-文档矩阵可以分解成3个矩阵的乘积
- 词项矩阵 U – 每个词项对应其中的一个行向量
- 文档矩阵 V^T – 每篇文档对应其中的一个列向量
- 奇异值矩阵 Σ – 对角方阵，对角线上的奇异值代表的是每个“语义”维度的重要性
- 接下来我们要介绍这样做的原因。

提纲

1. 上一讲回顾
2. 隐性语义索引
3. 空间降维处理
4. LSI 在IR中的应用

为什么在LSI中使用SVD分解

- 最关键的性质：每个奇异值对应的是每个“语义”维度的权重
- 将不太重要的权重置为0，可以保留重要的信息，去掉一些信息“枝节”
- 这些“枝节”可能是：
  - 噪音 – 这种情况下，简化的LSI 噪音更少，是一种更好的表示方法
  - 枝节信息可能会使本来应该相似的对象不相似，同样简化的LSI 由于其能更好地表达相似度，因而是一种更优的表示方式
- “细节越少越好”的一个类比
  - 鲜红色花朵的图像
  - 红黑花朵的图像
  - 如果忽略颜色，将更容易看到两者的相似性

将空间维度降为 2

U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

实际上，我们只需将矩阵Σ中相应的维度置为0即可。此时，相当于矩阵U和V^T的相应维度被忽略，然后计算$C_2 = U\Sigma_2 V^T$.

维度降为 2

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

=

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

×

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

×

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

回顾原始未分解的矩阵 $C=U\Sigma V^T$

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

=

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

×

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

×

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

原始矩阵 C vs. 简化的矩阵 $C_2 = U\Sigma_2 V^T$

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

C_2 可以看成矩阵C的一个二维表示。

我们将表示的维度缩减至2维。

为什么新的低维空间更好？

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

在原始空间中， d2和d3的相似度为0；但是在新空间下， d2和 d3的相似度为：

$0.52 * 0.28 + 0.36 * 0.16 + 0.72 * 0.36 + 0.12 * 0.20 + - 0.39 * - 0.08 \approx 0.52$

为什么新的低维空间更好？

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

“boat” 和“ship” 语义上相似。低维空间能够反映出这一点。

SVD的什么性质会导致相似度计算有所提高？

提纲

1. 上一讲回顾
2. 隐性语义索引
3. 空间降维处理
4. LSI 在IR中的应用

LSI在IR中使用的原因

- LSI 能够发现文档的语义上的关联 . . .
- . . . 但是在原始向量空间中这些文档相似度不大 (因为它们使用不同的词语) . . .
- . . . 于是通过LSI可以将它们映射到新的低维向量空间中 .
- . . . 在新的空间下，两者相似度较高
- 因此， LSI能够解决一义多词(synonymy) 和语义关联问题
- 在标准向量空间下，同义词对文档相似度计算没有任何贡献
- LSI所期望的效果：同义词对文档相似度贡献很大

LSI是如何解决一义多词和语义关联问题的

- 降维迫使我们忽略大量“细节”
- 我们将原始空间下不同的词映射到低维空间的同一维中
- 将同义词映射到同一维的“开销”远小于无关词的聚集
- SVD选择开销最小的映射方法
- 因此，SVD会将同义词映射到同一维
- 但是，它同时能避免将无关词映射到同一维

LSI与其它方法的比较

- 如果查询和文档没有公共词项时，前面我们介绍的相关反馈和查询扩展可以用于提高IR的召回率
- LSI会提高召回率但是损害正确率
- 因此，它和相关反馈查询扩展解决的是同一问题...
- ...同样它们的缺点也一致

LSI实现

- 对词项-文档矩阵进行SVD分解
- 计算在新的低维空间下的文档表示
- 将查询映射到低维空间中 $\vec{q}_2^T = \Sigma_2^{-1} U_2^T \vec{q}^T$.
- 上述公式来自： $C_2 = U\Sigma_2 V^T \Rightarrow \Sigma_2^{-1} U^T C = V_2^T$
- 计算 q_2 和 V_2 中的所有文档表示的相似度
- 像以往一样按照相似度高低输出文档结果
- 课堂练习：上述做法的最基本问题是什么？

最优性

- SVD 在下面的意义上说是最优的：
- 保留 k 个最大的奇异值并将其他奇异值置为0，这种做法得到是原始矩阵C的最佳逼近(参考Eckart-Young 定理)
- 最优性：不存在其它同秩的矩阵能够更加逼近C
- 逼近的度量指标F范数(Frobenius norm):

$$\|C\|_F = \sqrt{\sum_i \sum_j c_{ij}^2}$$

- 于是，LSI 得到最可能的矩阵
- 警告：F范数和文档的余弦相似度之间关系不大。

参考资料

- 《信息检索导论》第 18 章
- http://ifnlp.org/ir
  - Deerwester等人写的第一篇LSI的文章
  - Thomas Hofmann提出的概率LSI (PLSI)
  - 利用LSI来得到此空间

课后练习

- 习题18-5
- 习题18-11