建 模

# LSI潜在语义信息检索模型

何 伟

(中央民族大学信息与计算科学系, 北京 100081)

摘要: 本文介绍了基于向量空间的信息检索方法,检索词和文件之间的关系表示成一个矩阵,查寻信息表示为检索词权重的向量,通过求查寻和文件向量的夹角余弦确定出数据库中的相关文件. 使用矩阵的 QR分解和奇异值分解(SVD)来处理数据库本身的不确定性,本文的目的是说明线性代数中的基本概念可以很好解决信息检索(IR)问题.

关键词: 向量空间;信息检索(IR); QR分解;奇异值分解(SVD)

## 1 引 言

大型的学术会议上,组委会要对收到的论文组织分组会,即阅读论文的摘要将相关的论文分成一组;图书馆里工作人员要把大量的图书和杂志进行分类,以便读者能够检索到所需要的资料;互连网上每天要有几亿个网页要更新,等等. 所有这些都涉及到对文件的检索问题,而如此大量的工作利用手工检索是非常困难的.

目前,大多数信息检索(information retrieval,简称 IR)方法依据关键词匹配,即检索到的有关信息与使用者的信息需求(查询)具有共同使用的术语,但是,由于多词一意(同义词)的大量存在,使用者的查询可能不与相关文件匹配;另外,许多词具有一词多意(多义词),因此使用者的查询又可能与不相关的文件匹配. 所以,关键词匹配方法是不精确的.

潜在语义信息检索系统(Latent Semantic Indexing,简称 LSI)是一种将检索词和文件表示成矩阵(称为检索词—— 文件矩阵)的向量空间模型[1],所谓以语义为基础的检索,是指被检索到的有关信息与使用者的查询不一定具有共同使用的术语,通过对检索词—— 文件矩阵降秩,可以去掉矩阵表示的数据库中的无关信息和噪声.

本文首先介绍在早期的信息检索中使用的正交分解方法—— QR分解,虽然这种方法已不再使用,但可以用来说明降秩的意义;然后介绍比 QR分解更复杂的正交分解—— 奇异值分解(singular value decomposition,简称 SVD)在信息检索中的应用.

## 2 信息检索的向量空间模型

1) 信息的向量空间表示

在 LSI模型中,检索词和文件间的关系由矩阵 $A=(a_{ij})_{k\times d}$(称为检索词—— 文件矩阵)表示,元素 $a_{ij}$表示检索词 $i$在文件 $j$中出现的次数或频率,因为一个文件中检索词的数目远比整个文件集中的检索词要少,因此,$A$是稀疏矩阵. 实际应用中,一般使用局部权重和整

收稿日期: 2001-05-14

体权重来增加或降低检索词在文件中的重要程度,如取 $a_{ij}=l_{ij}g_i$,其中 $g_i$ 是检索词 $i$ 在整个文件集的整体权重,$l_{ij}$ 是检索词 $i$ 在文件 $j$ 中的局部权重. 例如考虑有关计算机文章中的"计算机"这个词,它的整体权重 $g_i$ 就应该比较小. 因子 $l_{ij}$ 反映了检索词 $i$ 在文件 $j$ 中的重要程度,有时取 0或 1,或者取检索词频率的函数,例如 $l_{ij}=\log(f_{ij}+1)$,其中 $f_{ij}$ 是检索词 $i$ 在文件 $j$ 中出现的频率.

2) 查询匹配

$A$ 的第 $j$ 列 $a_j$ 表示一个文件向量,将查询写成文件向量 $q=(q_i)_{t\times 1}$ 的形式,若查询中有第 $i$ 个检索词,则 $q_i$ 取 1,否则取 0,称 $q$ 为查询向量. 相关文件的查找通过计算查询向量 $q$ 和 $A$ 中的文件向量 $a_j$ 的夹角 $\theta_j$ 的余弦来确定,计算公式为:

$$\cos\theta_j=\frac{a_j^{T}q}{\|a_j\|_2\|q\|_2}=\frac{\sum_{i=1}^{t}a_{ij}q_i}{\sqrt{\sum_{i=1}^{t}a_{ij}^2}\sqrt{\sum_{i=1}^{t}q_i^2}} \tag{1}$$

仅当余弦值大于某个给定的阈值时,该文件作为与查询向量相关的文件反馈给使用者.

例 **1** 下表是一个小型的书目数据库.

$t=$ 6个检索词:

$T_1$: 股票; $T_2$: 债券; $T_3$: 期货; $T_4$: 期权; $T_5$: 理论; $T_6$: 应用.

$d=$ 7个文件:

$D_1$: 股票、债券、期货知识;

$D_2$: 金融工程核心工具—— 期权;

$D_3$: 期货分析预测学;

$D_4$: 期权定价理论;

$D_5$: 股票、债券、期货、期权的理论及其应用;

$D_6$: 期权、期货和衍生证券;

$D_7$: 期货期权市场.

让 $\tilde{a}_{ij}$ 表示检索词 $i$ 在文件 $j$ 中出现的次数,得到 $6\times 7$ 矩阵 $\tilde{A}$:

$$\tilde{A}=\begin{pmatrix}1&0&0&0&1&0&0\\1&0&0&0&1&0&0\\1&0&1&0&1&1&1\\0&1&0&1&1&1&1\\0&0&0&1&1&0&0\\0&0&0&0&1&0&0\end{pmatrix}$$

为计算方便,将其列按 $\|\cdot\|_2$ 模归一化,得到的矩阵作为检索词—— 文件矩阵 $A$:

$$A=\begin{pmatrix}0.5774&0&0&0&0.4082&0&0\\0.5774&0&0&0&0.4082&0&0\\0.5774&0&1&0&0.4082&0.7071&0.7071\\0&1&0&0.7071&0.4082&0.7071&0.7071\\0&0&0&0.7071&0.4082&0&0\\0&0&0&0&0.4082&0&0\end{pmatrix}$$

实际应用中一般采用 0.9这样比较强的截断值,但在这个比较小的例子里,取 0.45作为阈值.

假定用户要查询有关“股票”和“债券”的书,则查询向量为:

$$q^{(1)} = (1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0)^T$$

它与七个文件向量的夹角余弦为: 0.8166, 0, 0, 0, 0.5774, 0, 0,因此第一和第五本书作为相关文件反馈给用户.

但如果只查找有关“债券”的书,此时,查询向量为:

$$q^{(2)} = (0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0)^T$$

其相应的余弦值为: 0.5774, 0, 0, 0, 0.4082, 0, 0,只有第一本书作为与债券有关的书反馈给用户,而第五本书没有作为相关文件,这显然是错误的.

为此,人们研究了各种各样的方法来处理向量空间模型的这种缺陷,其典型特征是改变数据在检索词—— 文件矩阵中的表示方式, LSI采用低秩近似来替代矩阵 $A$,下面介绍 QR分解和 SVD分解两种降秩方法. LSI的研制者认为,降低矩阵 $A$的维数,可以揭示文件间的基本语义关系,消除无关信息或噪声.

## 2 QR分解

1) QR分解的定义

首先求矩阵 $A$的 QR分解:

$$A = QR$$

其中 $Q$是 $t \times t$正交矩阵, $R$是 $t \times d$上三角矩阵. 可以证明,任何矩阵的 QR分解都是存在的[2].

假设矩阵 $A$的秩为 $r_A$,则 $A$的 $d$个列向量都可以用列空间的 $r_A$个基向量来表示,从而可由 $Q$的 $r_A$个列向量线性表示,即 $Q$的这 $r_A$个列向量作为 $A$的一个基. 本例中的 $Q$和 $R$分别为:

$$Q = \left(\begin{array}{cccccc|c} -0.5774 & 0 & 0.4082 & 0 & 0 & & 0.7071 \\ -0.5774 & 0 & 0.4082 & 0 & 0 & & -0.7071 \\ -0.5774 & 0 & -0.8165 & 0 & 0 & & 0 \\ 0 & -1.0000 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & -1.0000 & 0 & & 0 \\ 0 & 0 & 0 & 0 & -1.0000 & & 0 \end{array}\right)$$

$$R = \left(\begin{array}{ccccccc} -1.0001 & 0 & -0.5774 & 0 & -0.7070 & -0.4082 & -0.4082 \\ 0 & -1.0000 & 0 & -0.7071 & -0.4082 & -0.7071 & -0.7071 \\ 0 & 0 & -0.8165 & 0 & 0 & -0.5773 & -0.5773 \\ 0 & 0 & 0 & -0.7071 & -0.4082 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.4082 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

将矩阵进行如上分块,则有:

$$A = QR = (Q_A, Q_A^{\perp}) \begin{pmatrix} R_A \\ 0 \end{pmatrix} = Q_A R_A$$

$Q_A$是 $6 \times 5$的矩阵, $R_A$是 $R$的非零行, $Q_A^{\perp}$对 $A$不起作用. $Q_A$的五列构成了 $A$的一个

基. 这里 $R$ 中全为零的行出现在其底部,否则可以先对 $A$ 作列置换,再进行 QR分解,即:

$$AP = QR$$

其中 $P$ 是一个置换矩阵. 但是对 $A$ 进行列置换,即交换 $A$ 的列并没有改变数据库,只是对文件向量进行重新排序,因此,为简单起见,仍用 $A$ 代替 $AP$. 假设 $R_A$ 的第 $j$ 列为 $r_j$,则 $A$ 的第 $j$ 列 $a_j = Q_A r_j$,于是 (1)式为:

$$\cos\vartheta_j = \frac{(Q_A r_j)^T q}{\| Q_A r_j \|_2 \| q \|_2} = \frac{r_j^T (Q_A^T q)}{\| r_j \|_2 \| q \|_2} \tag{2}$$

重新计算本例的余弦值,对于 $q^{(1)}$,其余弦值为: 0. 8166, 0, 0, 0. 5774, 0, 0. 与原来的计算结果完全一致,没有信息丢失.

2) QR分解对信息检索的意义

下面解释 QR分解的几何意义. 对正交矩阵 $Q$ 有:

$$I = QQ^T = (Q_A, Q_A^{\perp})(Q_A, Q_A^{\perp})^T = Q_A Q_A^T + Q_A^{\perp}(Q_A^{\perp})^T$$

于是查询向量 $q$ 可以分解成下面两项和:

$$\begin{aligned} q &= Iq = QQ^T q = [Q_A Q_A^T + Q_A^{\perp}(Q_A^{\perp})^T]q \\ &= Q_A Q_A^T q + Q_A^{\perp}(Q_A^{\perp})^T q = q_A + q_A^{\perp} \end{aligned} \tag{3}$$

$q_A = Q_A Q_A^T q$ 称为 $q$ 到由 $Q_A$ 的列构成空间的正交投影. 并且可以证明 $q_A$ 是 $q$ 在 $A$ 的列空间的最佳近似[1],即

$$\| q - q_A \|_2 = \min\{\| q - x \|_2, x \text{ 在 } A \text{ 的列空间中}\}$$

将 (3)带入 (1),重新计算查询向量和文件向量的余弦:

$$\begin{aligned} \cos\vartheta_j &= \frac{a_j^T q_A + a_j^T q_A^{\perp}}{\| a_j \|_2 \| q \|_2} = \frac{a_j^T q_A + a_j^T Q_A^{\perp}(Q_A^{\perp})^T q}{\| a_j \|_2 \| q \|_2} \\ &= \frac{a_j^T q_A + 0 \cdot (Q_A^{\perp})^T q}{\| a_j \|_2 \| q \|_2} = \frac{a_j^T q_A}{\| a_j \|_2 \| q \|_2} \end{aligned} \tag{4}$$

在上式中,因为 $a_j$ 是 $A$ 的一列,故它与 $Q_A^{\perp}$ 的列正交,因此,$a_j^T Q_A^{\perp} = 0$. 上式说明,在计算点乘 $a_j^T q$ 时,查询向量 $q$ 用它在数据库中的最佳估计 $q_A$ 代替,不考虑 $q_A^{\perp}$. 进一步,在 (4)式的分母中,用 $q_A$ 代替 $q$,得到新的相似性度量:

$$\cos\vartheta_j' = \frac{a_j^T q_A}{\| a_j \|_2 \| q_A \|_2} \tag{5}$$

显然, $\cos\vartheta_j' \geqslant \cos\vartheta_j$,即如果使用 (5)式,则文件 $j$ 更有可能与查询相关,从而能识别更多的相关文件,但无关文件也可能被错误地识别为相关文件,因此,检索率 (被检索到的相关文件的比例)有可能提高,但准确度 (反馈给使用者的相关文件比例)可能会下降,而检索率和准确度是评价一种检索方法好坏的重要指标,一般采用几种检索率水平上的平均准确度作为衡量检索方法的总体检索性能.

## 3 降秩估计

$QR$ 分解还可以处理数据库中的不确定性. 因为数据库和它的矩阵表示可能是由许多人经过较长的时间建立起来的,而这些人对数据库内容的分类会有不同的经验和观点. 因此,$A$ 的列空间并不一定是数据库语义内容的最好描述,它有不确定性,于是检索词—— 文件矩阵应该写成 $A+U$, $U$ 是不确定性矩阵,反映了文件中丢失的或不完全信息,以及文件

与某个主题的相关性,例如,在例 1中,有人认为第一个文件与检索词"理论"相关,则在矩阵 $\tilde{A}$ 中,应有 $\tilde{a}_{15}=1$. 由于我们所关心的是不相关的、独立的信息,故可以认为 $A+U$ 的秩应比 $A$ 的秩小. 下面说明怎样对 $A$ 降秩,以便去除相关的信息或噪声.

首先,给出有关矩阵大小的概念,称为罗贝尼乌斯(Frobenius)范数,简称 $F$ 范数,对于 $m\times n$ 矩阵 $X$,其 $F$ 范数为:

$$\|X\|_F=\sqrt{Trace(X'X)}=\sqrt{Trace(XX')}$$

其中 $Trace(X'X)$ 等于 $X'X$ 的对角元素之和. 对于一个 $m\times m$ 正交矩阵 $Y$,有:

$$\|YX\|_F=\sqrt{Trace((YX)'(YX))}=\sqrt{Trace(X'Y'YX)}=\sqrt{Trace(X'X)}=\|X\|_F$$

类似的,对于 $n\times n$ 正交矩阵 $Z$,有 $\|ZX\|_F=\|X\|_F$.

下面对矩阵 $A$ 找一个合理的低秩近似,我们知道在矩阵的 $QR$ 分解中,$A$ 与 $R$ 有相同的秩,而 $R$ 的秩容易确定,等于 $R$ 中对角线的非零元素个数. 在本例中,对 $R$ 进行如下分块:

$$R=\left(\begin{array}{cccc|ccc}
-1.0001 & 0 & -0.5774 & 0 & -0.7070 & -0.4082 & -0.4082\\
0 & -1.0000 & 0 & -0.7071 & -0.4082 & -0.071 & -0.7071\\
0 & 0 & -0.8165 & 0 & 0 & -0.5773 & -0.5773\\
0 & 0 & 0 & -0.7071 & -0.4082 & 0 & 0\\
\hline
0 & 0 & 0 & 0 & -0.4082 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)=\left(\begin{array}{cc}R_{11} & R_{12}\\ 0 & R_{22}\end{array}\right)$$

在这种分块下,子矩阵 $R_{22}$ 是矩阵 $R$ 中比较小的部分,$\|R_{22}\|_F/\|R\|_F=0.4082/2.6457\approx 0.1543$. 现在在矩阵 $R$ 中将小矩阵 $R_{22}$ 用零矩阵代替,得到一个新的上三角矩阵 $\tilde{R}$,其秩为 4,让 $A+U=Q\tilde{R}$,则 $A+U$ 的秩为 4,且不确定矩阵 $U$ 为:

$$U=(A+U)-A=Q\left(\begin{array}{cc}R_{11} & R_{12}\\ 0 & 0\end{array}\right)-Q\left(\begin{array}{cc}R_{11} & R_{12}\\ 0 & R_{22}\end{array}\right)=Q\left(\begin{array}{cc}0 & 0\\ 0 & R_{22}\end{array}\right)$$

由于

$$\|U\|_F=\left\|\left(\begin{array}{cc}0 & 0\\ 0 & R_{22}\end{array}\right)\right\|_F=\|R_{22}\|_F,\|A\|_F=\|R\|_F$$

故 $\|U\|_F/\|A\|_F\approx 0.1543$. 即对 $R$ 和 $A$ 的 $F$ 范数进行了大约 15%的相对改变,使得它们的秩都降了 1阶. 在用(2)式计算余弦值时,用 $A+U$ 代替 $A$,但不必将 $A+U$ 计算出来,此时 $r_j$ 是 $\tilde{R}$ 的第 $j$ 列,$Q_4$ 是 $Q$ 的前四列.

现在返回来看例 1,用 $A+U$ 代替 $A$,对 $q^{(1)}$,七个文件的余弦值为: 0.8166, 0, 0, 0, 0.6325, 0, 0,对 $q^{(2)}$,其余弦值为: 0.5774, 0, 0, 0, 0.4472, 0, 0. 文件的相关性都得到了提高,即秩为 4的 $A+U$ 比原来的检索词——文件矩阵 $A$ 更好的描述了该数据库.

进一步降秩,使 $R$ 中的第四行、第四列也包括在 $R_{22}$ 中,此时,$\|R_{22}\|_F/\|R\|_F=0.9129/2.6457\approx 0.3450$,即对 $R$ 和 $A$ 的值进行了大约 35%的相对改变,检索词——文件矩阵的秩变成了 3,此时,对 $q^{(1)}$,余弦值为: 0.8166, 0, 0, 0, 0.7072, 0, 0,对 $q^{(2)}$,余弦值为: 0.5774, 0, 0, 0, 0.5000, 0, 0,结果又进一步得到了改善,而且,对 $q^{(2)}$,第五本书作为相关文

件反馈给使用者.

再一次降秩,这时,$\| R_{22}\|_F / \| R\|_F = 1.5811/2.6457 \approx 0.5976$. 即对 $R$ 和 $A$ 的值进行了大约 60% 的相对改变,检索词——文件矩阵的秩降到 2,对 $q^{(1)}$,余弦值为: 0.8166, 0, 0.8166, 0, 0.7072, 0.4083, 0.4083. 这时不相关文件反馈给了用户,即 $R$ 和 $A$ 的 60% 的相对改变太大了,不可接受.

本例中,35% 的相对改变是可以接受的,在实际应用中,一般选择更小的相对值改变.

## 4 奇异值分解 (SVD)

下面介绍目前普遍采用的降秩方法——SVD分解方法,它有比较好的数学性质,即对给定的 $k$,在最小二乘意义下,$A$ 的 $k$ 秩近似是对 $A$ 的最小改变.

$A$ 的 SVD分解是:

$$A = U\Sigma V^T$$

其中 $U$ 是 $t\times t$ 正交矩阵,它的每一列是 $A$ 的左奇异向量,$V$ 是 $d\times d$ 正交矩阵,它的每一列是 $A$ 的右奇异向量,$\Sigma$ 是 $t\times d$ 对角矩阵,对角线元素是 $A$ 的奇异值,$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(t,d)}$ 按大小顺序排列. 可以证明,对任意矩阵 $A$,这种分解都是存在的[2].

$A$ 的秩 $r_A$ 等于非零的奇异值数,$A$ 的 $F$ 范数 $\| A\|_F$ 为:

$$\| A\|_F = \| U\Sigma V^T\|_F = \| \Sigma V^T\|_F = \| \Sigma\|_F = \sqrt{\sum_{j=1}^{r_A} \sigma_j^2}$$

SVD分解与 QR分解有许多相似之处,正如 $Q$ 的前 $r_A$ 列是 $A$ 的列空间的一个基,$U$ 的前 $r_A$ 列是 $A$ 的列空间的一个基,且 $V^T$ 的前 $r_A$ 行是 $A$ 的行空间的一个基;在 QR分解中,$A$ 的 $k$ 秩近似是让 $R$ 的除了前 $k$ 行以外的其它元素都为零,在 SVD分解中,$A$ 的 $k$ 秩近似 $A_k$ 是让 $A$ 的除了前 $k$ 个最大奇异值以外的奇异值都为零.

可以证明[3] $A$ 与 $A$ 的近似 $A_k$ 的距离满足以下式子:

$$\| A - A_k\|_F = \min_{rank(X)\leq k} \| A - X\|_F = \sqrt{\sigma_{k+1}^2 + \cdots \sigma_{r_A}^2}$$

这里 $A_k = U_k\Sigma_k V_k^T$,其中 $U_k$ 是 $U$ 的前 $k$ 列形成的 $t\times k$ 矩阵,$V_k$ 是 $V$ 的前 $k$ 列形成的 $d\times k$ 矩阵,$\Sigma_k$ 是 $A$ 的 $k$ 个最大奇异值形成的 $k\times k$ 对角矩阵.

例 1的 SVD分解为:

$$U = \begin{pmatrix} 0.1599 & -0.2596 & 0.5293 & -0.3276 & 0.1399 & -0.7071 \\ 0.1599 & -0.2596 & 0.5293 & -0.3276 & 0.1399 & 0.7071 \\ 0.6542 & -0.6231 & -0.3469 & 0.2519 & -0.0020 & 0 \\ 0.6921 & 0.6468 & -0.0814 & -0.3097 & -0.0079 & 0 \\ 0.1884 & 0.2402 & 0.4946 & 0.7877 & 0.2039 & 0 \\ 0.0796 & -0.0287 & 0.2611 & 0.0740 & -0.9588 & 0 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 2.0117 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.2842 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9406 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5787 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2903 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$V=\begin{pmatrix} 0.2795 & -0.5136 & 0.4369 & -0.4025 & 0.5524 & 0 & 0 \\ 0.3441 & 0.5037 & -0.0865 & -0.5351 & -0.0273 & -0.5773 & 0 \\ 0.3252 & -0.4852 & -0.3688 & 0.4352 & -0.0070 & -0.5773 & 0 \\ 0.3095 & 0.4884 & 0.3107 & 0.5841 & 0.4774 & 0 & 0 \\ 0.3925 & -0.0903 & 0.6015 & 0.1049 & -0.6819 & 0 & 0 \\ 0.4732 & 0.0131 & -0.3219 & -0.0706 & -0.0242 & 0.4083 & 0.7071 \\ 0.4732 & 0.0131 & -0.3219 & -0.0706 & -0.0242 & 0.4083 & -0.7071 \end{pmatrix}$$

$\| A-A_4\|_F=T_5=0.2903$,又$\| A\|_F=2.6457$,则$\| A-A_4\|_F/\| A\|_F\approx 0.1097$,类似的,$\| A-A_3\|_F/\| A\|_F\approx 0.2447$,$\| A-A_2\|_F/\| A\|_F\approx 0.4316$. 即 24% 的相对改变使 $A$ 的秩由 5降到 3, 43% 的相对改变使 $A$ 的秩由 5降到 2.

计算查询向量 $q$ 与 $A$ 的近似 $A_k$ 第 $j$ 列的夹角余弦,$e_j$ 表示 $d\times d$ 单位矩阵的第 $j$ 列,则,

$$\cos\theta_j=\frac{(A_ke_j)^Tq}{\| A_ke_j\|_2\| q\|_2}=\frac{(U_kE_kV_k^Te_j)^Tq}{\| U_kE_kV_k^Te_j\|_2\| q\|_2}=\frac{e_j^TV_kE_k(U_k^Tq)}{\| E_kV_k^Te_j\|_2\| q\|_2},\quad j=1,\cdots d$$

定义向量 $s_j=E_kV_k^Te_j$,则上式化简为:

$$\cos\theta_j=\frac{s_j^T(U_k^Tq)}{\| s_j\|_2\| q\|_2},\quad j=1,\cdots,d \tag{6}$$

(6)式的几何意义是: 向量 $s_j$ 的 $k$ 个元素是以 $U_k$ 的列为基的 $A_k$ 第 $j$ 列的坐标,向量 $U_k^Tq$ 的 $k$ 个元素是查询向量 $q$ 到 $A_k$ 列空间投影 $U_kU_k^Tq$ 的以 $U_k$ 的列为基的坐标. 这一解释与(5)式类似,于是相应的得到新的相似性度量:

$$\cos\theta_j'=\frac{s_j^T(U_k^Tq)}{\| s_j\|_2\| U_k^Tq\|_2},\quad j=1,\cdots,d \tag{7}$$

显然,$\cos\theta_j'\geqslant\cos\theta_j$,当用(7)式代替(6)式时,检索率可能会提高,其成本是降低准确度.

再来看例 1,使用 $A_3$,对 $q^{(1)}$,余弦值是: 0.7057, -0.1492, 0.1209, 0.1389, 0.6589, 0.0176, -0.0176. 第一与第五本书被正确的识别了,只是其余的余弦值不再是零了,但与阈值相比很小,对 $q^{(2)}$,余弦值是: 0.4990, -0.1055, 0.0855, 0.0982, 0.4659, -0.0124, -0.0124.

使用 $A_2$,对 $q^{(1)}$,余弦值是: 0.4261, -0.0854, 0.4169, -0.1012, 0.2771, 0.2196, 0.2196. 对 $q^{(2)}$,余弦值是: 0.3013, -0.0604, 0.2948, -0.0716, 0.1959, 0.1553. 这个结果不合理,说明 $A_2$ 不是 $A$ 的合理近似.

从以上分析看到,降秩可以消除许多噪声,但维数太小,又会丢失重要的信息,选择多大的秩,需要依据不同的数据库,一般凭经验和实验来决定.

从例 1看到,$SVD$ 分解中降秩所要求的相对改变小于 $QR$ 分解的相应改变;另外,不象 $QR$ 分解,$SVD$ 分解是对矩阵 $A$ 的行和列空间同时降秩,因此,SVD分解即能对查询和文件进行比较,也能对检索词之间进行比较,故 SVD分解优于 QR分解.

## 5 处理动态数据

数据库的信息可能每秒中都发生着变化,新的检索词和文件不断的进入到数据库,因此,如果不断的计算检索词—— 文件矩阵的 SVD分解,对一个大型的数据库,这个过程对计

算机在时间和空间上的成本都太大了. 下面简单介绍两种解决这个问题的方法: 合并(folding-in)和 SVD-更新(SVD-updating)[3],两种方法都利用已有的降秩 SVD分解. 前一种方法计算量小,但对数据库的描述不精确,比较适合只是偶尔有新的文件进入的情形. 后一种方法计算量大,效果好.

1. 合并

假设收到一个新的 $t\times 1$文件向量 $P$,为了将其添加到 $A_k$ 中,首先将其投影到检索词——文件矩阵列空间的基 $U_k$ 上,得到向量 $P^{(0)}$:

$$P^{(0)} = U_k U_k^T P$$

将 $P^{(0)}$在基 $U_k$ 下的坐标表示 $U_k^T P$ 作为矩阵 $\mathrm{E}_k V_k^T$ 的一列添加上去,即在 $V_k$ 中添加新的一行 $P^T U_k \mathrm{E}_k^{-1}$,得到新的矩阵 $V_k^{(0)}$,这样新的文件就被引入了. 但这里 $V_k^{(0)}$ 不再是正交的, $V_k^{(0)}$ 的行空间也不表示新的检索词——文件矩阵的行空间,并且,如果 $P^{(0)}$与 $U_k$ 的行接近正交,那么,新文件的大部分信息在投影时会被丢失.

类似的,当有一个新的 $d\times 1$检索向量 $c$时,将其投影到 $A_k$ 的行空间,得投影向量 $C^{(0)}$:

$$C^{(0)} = V_k V_k^T c$$

将 $C^{(0)}$的坐标 $V_k^T c$作为一行添加到 $U_k$ 中去,得 $U_k^{(0)}$ 即为所求,同样, $U_k^{(0)}$ 也不是正交的.

2. SVD-更新

前一种方法 $U_k^{(0)}$ 和 $V_k^{(0)}$ 的正交性受到了破坏,丢失的信息多,下面的方法克服了这一缺点,但计算量大,这里只作简单介绍.

1) 更新检索词

假设有 $s$ 个新的检索词向量. 用 $S$ 表示 $s$ 个新的检索词形成的 $s\times d$ 矩阵,将 $S$ 添加到 $k$ 秩 $t\times d$ 矩阵 $A_k = U_k \mathrm{E}_k V_k^T$ 中,得到新的检索词——文件矩阵 $B$:

$$B = \begin{pmatrix} A_k \\ S \end{pmatrix}$$

$B$ 是 $(t+s)\times d$ 矩阵,其秩大于等于 $A_k$ 的秩 $k$,再求 $B$ 的 $k$ 秩近似 $B_k = U_k \mathrm{E}_k V_k^T$,这几乎相当于重新计算 SVD,只是 $A$ 用 $A_k$ 代替.

2) 更新文件

假设有 $r$ 个新的文件. 用 $R$ 表示 $t\times r$ 新文件向量,将 $R$ 添加到 $A_k$ 中,得到 $B$:

$$B = (A_k, R)$$

$B$ 是 $t\times (d+r)$ 矩阵,秩至少是 $k$,再求 $B$ 的 $k$ 秩近似 $B_k = U_k \mathrm{E}_k V_k^T$.

## 6 小 结

LSI模型是对数据库的矩阵表示进行降秩估计的向量空间模型,这种方法在线性代数、统计、图象处理、数据压缩、密码研究中有广泛的应用. 与传统的信息检索方法相比, LSI有许多优点,它是一种自动检索系统,人们已经研究出了现成的软件,现在被广泛应用. 例如, SIAM 99′年会的组织者就利用该软件对提交的 128篇论文进行分组,取得了令人满意的结果[4]. 下面简单总结其特点.

人们通过实验[5]将 LSI与各种检索系统比较,表明 LSI有更高的准确度和检索率. 在

一个测试[6]中,在对标准的信息录入装置(Message Entry Device)文件集进行检索时,LSI给出了比传统的关键词匹配检索方法更多的相关文件. LSI基于含义检索相关的信息,即使与查询没有共同的检索词. 为此,引用[5]中的例子说明这一点.

例 **2** 下面九个文件中 $D_1$— $D_5$ 是有关人与计算机交互影响的,$D_6$— $D_9$ 是有关图论的,下划线单词不止一次出现,作为检索词.

$D_1$: human machine interface for Lab ABC computer applications;
$D_2$: A survey of user opinion of computer system response time;
$D_3$: The EPS user interface management system;
$D_4$: System and human system engineering testing of EPS;
$D_5$: Relation of user-perceived response time to error measurement;
$D_6$: The generation of random, binary, unordered trees;
$D_7$: The intersection graph of paths in trees;
$D_8$: Graph minors IV: Widths of trees and well-quasi-ordering;
$D_9$: Graph minors: A survey.

现在查询与"人与计算机交互影响(human machine interface)"有关的文件,使用简单的关键词匹配检索方法,得到文件: $D_1$, $D_2$, $D_4$,但另两个相关文件 $D_3$ 和 $D_5$ 却丢失了,因为它们没有与查询共同的检索词. 然而运用 LSI检索方法,建立 12× 9检索词—— 文件矩阵 $A$,使用降秩的 SVD分解,本例 $A_2$ 是合适的近似,结果文件 $D_1$— $D_5$ 被检索到(余弦截断值为 0.9).

一般说,多义词和同义词是困扰自动信息检索系统的障碍,多义词使检索的准确度受到影响,而同义词的存在又会降低检索系统的检索率,LSI通过降秩,消除词的用法上的差别,而这正是困扰基于词检索方法的大问题. 此外,LSI中的关联性反馈(Relevance Feedback)[7]功能可以更好的提高总体检索性能,其思想是查询向量用反馈的相关文件的向量和代替,以提高检索的精确性. 以下给出其数学描述.

假设 $q$ 是查询向量,因为只关心 $A_k$ 列空间中包含的信息,所以用其对 $A_k$ 列空间的投影 $U_k U_k^T q$ 代替 $q$,如果检索的结果只有一个相关文件 $a_j$,则新的查询向量 $q_{new}$ 为:

$$q_{new} = U_k U_k^T q + a_j = U_k U_k^T q + U_k E_k V_k^T e_j = U_k (U_k^T q + E_k V_k^T e_j)$$

(如果 $a_j$ 不在 $A_k$ 的列空间中,则用其投影 $U_k U_k^T a_j$ 代替 $a_j$)如果相关的文件不止一个,则新的查询向量 $q_{new}$ 可写成:

$$q_{new} = U_k U_k^T q + \sum_{j=1}^{d} w_j a_j = U_k (U_k^T a_j + E_k V_k^T w)$$

其中如果 $a_j$ 是相关文件,$w_j$ 取 1,否则取 0. 有时新的查询向量只取相关文件的和,即上面和式中的第一项取零. 在用(6)式计算余弦值时,用 $q_{new}$ 代替 $q$. 经验表明,这样计算的结果在一定程度上提高了检索的性能.

有关 *LSI* 模型的研究成果还在不断出现,本文例子的计算使用 Matlab软件.

参考文献:

[1] Michael W Berry, Zlatko Drmac, Elizabeth R Jessup. Matrices, vector spaces, and information retrieval[J].

SIAM Rev, 1999, 41: 335- 362.

[2] Golub G, Loan V Van. Matrix Computations[M]. 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[3] Mirsky L. Symmetric gage functions and unitarilly invariant norm[J]. Q J Math, 1960, 11: 50- 59.

[4] Michael Berry, Jack Dongarra. Atlanta organizers put mathematics to work for the math sciences community[J]. SIAM News, 1999, 32: 10- 11.

[5] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, Richard harshman. Indexing by latent semantic analysis[J]. J of the Amer Soc for Inform Sci, 1990, 41: 391- 407.

[6] Dumais S T. Improving the retrieval of information from external sources[J]. Behavior Res Meth& Comp, 1991, 23: 229- 236.

[7] Salton G, Buckley C. Improving retrieval performance by relevance feedback[J]. J Amer Soc for Inform Sci, 1990, 41: 288- 297.

# LSI Latent Semantic Indexing Model

HE Wei

(Department of Information and Computing Science, Central University for Nationalities, Beijing 100081, China)

**Abstract** In the paper, a information retrieval method based on concept of vector space is introduced. A matrix represents the relationships between terms and documents, a query information is written as vector of weighting of terms, A document is returned as relevant according to the cosine of the angle between the query and document vectros in the database. QR factorization and singular value decomposition provide mechanisms for handling uncertainty in the database itself. The purpose of this paper is to show how fundamental mathematical concepts from linear algebra can be used to manage in information retrieval.

**Keywords** vector spaces; infoirmation retrieval (IR); QR factorizationi; singular value decomposition (SVD)