第35卷 第15期 计 算 机 工 程 2009年8月
Vol.35 No.15 Computer Engineering August 2009
·软件技术与数据库· 文章编号：1000—3428(2009)15—0035—03 文献标识码：A 中图分类号：TP311

一种大规模数据的快速潜在语义索引

卫 威[1]，王建民[2]
(1. 清华大学计算机科学与技术系，北京 100084；2. 清华大学软件学院，北京 100084)

摘 要： 潜在语义索引(LSI)已应用到现代信息检索的多个领域，但矩阵奇异值分解的高复杂度阻碍了该技术在大规模数据上的应用。提出一种大规模数据的快速LSI方法。给出一个降维问题的统一框架，LSI作为一种特征提取算法，可以在这个框架下转化为一个特征选择问题。利用该技术在最大程度保持LSI降维效果的同时，简化LSI的计算，使其能够应用于大规模数据。

关键词： 潜在语义索引；降维；特征选择；特征提取

Fast Latent Semantic Indexing on Large-scale Dataset

WEI Wei[1], WANG Jian-min[2]
(1. Department of Computer Science and Technology, Tsinghua University, Beijing 100084;
2. School of Software, Tsinghua University, Beijing 100084)

【Abstract】 Latent Semantic Indexing(LSI) has been successfully applied to various fields in modern information retrieval. However, the high computational complexity of Singular Value Decomposition(SVD) makes it improbable on the application of large-scale dataset. This paper proposes a fast LSI approach to solve this problem. It gives a unified framework of dimension reduction problem. As a feature extraction method, LSI can be transformed into a feature selection method within this framework. This new strategy can simplify significantly the computation of LSI.

【Key words】 Latent Semantic Indexing(LSI); dimension reduction; feature selection; feature extraction

1 概述

在现代信息信息检索系统中，通过关键词(keywords)进行检索是最为常见的做法。其基本原理是用户提出若干个反映文档(document)主题的词语(term)组成查询(query)，然后将查询与系统中的每篇文档进行比较，检出相关度最高的文档。

但大量研究表明，这种基于关键词的检索系统存在所谓“同义词”(synonymy)和“反义词”(polysemy)的固有缺陷。前者是指表达同一概念的词语可以有多个，因此，用户查询中所用的词语很可能在相关文档中不存在，从而造成检出率(recall)下降；后者是指同一个词语可以表达多个概念，造成检出的文档中虽然包含该词语，但在上下文语境中的意思却非用户所期望，从而导致准确率(precision)下降。

潜在语义索引(Latent Semantic Indexing, LSI)方法正是为了解决上述问题而提出的。这种方法的基本原理是认为文档和词语之间存在着某种潜在的相关关系，这种关系可以通过适当的线性运算映射到一个特定的语义空间中，从而更好地加以利用。在一些文献中，它也被称作潜在语义分析(Latent Semantic Analysis, LSA)。

LSI方法提出后被成功地应用到了文本信息检索的多个领域，包括经典的向量空间模型(Vector Space Model, VSM)、信息过滤(information filtering)、多语言检索(cross-language information retrieval)等，相对传统方法在效果上取得了显著的改善。

事实上，LSI方法从本质上可以看成是一种文本数据上的降维方法。降维方法通常可以分为2大类：特征选择和特征提取，LSI属于后者。特征提取的经典方法包括主成分分析(Principle Component Analysis, PCA)和线性判别分析(Linear Discriminate Analysis, LDA)，而特征选择则包括信息增益(Information Gain, IG)和χ^2统计等算法。前者的优势在于由于采取了比较复杂的矩阵变换，因此可以取得更好的降维效果，但劣势在于计算复杂度通常比较高，不适合应用于大规模数据。

互联网的兴起给传统信息检索的各个领域，包括LSI方法在内，带来了巨大的机遇与挑战。这是因为LSI方法中必须进行的一个步骤——矩阵奇异值分解(Singular Value Decomposition, SVD)的计算复杂度非常高，无法有效地处理海量数据，从而无法在互联网时代的大规模信息检索中得到应用。

本文提出了一种快速LSI方法来解决上述问题，把特征提取和特征选择方法统一到同一个优化框架下，发现前者和后者的差别只是在于转换矩阵的解空间是连续的还是离散的。将LSI方法在这个优化框架下进行了形式化。最后提出一种将LSI方法的解空间由连续空间映射到离散空间的方法，从而大大降低了其计算复杂度，同时通过实验证明新的算法保持了降维的效果。

2 相关工作

LSI方法是由美国科学家Bellcore等人于20世纪80年代后期提出并申请专利[1]。之后的一系列实验证明了这种方法在信息检索的诸多应用领域的效果，包括提高检索精度、对论文和审稿人进行匹配、信息过滤、多语言检索、文档分类以及P2P网络上的分布式检索等[2-3]。

LSI在应用过程中一直未能解决的一个问题便是SVD运算的高复杂度，以至这种方法一直不能在大规模数据上得到

作者简介： 卫 威(1983－)，男，硕士研究生，主研方向：数据挖掘，机器学习；王建民，教授、博士生导师
收稿日期： 2009-02-20 **E-mail：** doublewei@gmail.com

应用。针对这个问题，人们首先想到的是通过各种迭代算法改进 SVD 运算本身的效率。然而，即使是当前公认最快的 LAS2 算法，也不可能在有限时间内处理百万量级的文档。

之后人们陆续提出了一些方案来逼近 LSI 的结果。Kolda 等人提出了一种名为半离散矩阵分解(Semi-Discrete matrix Decomposition, SDD)的方法来替代 LSI 计算中的 SVD 过程[4]。这种方法的优势在于需要的存储空间比较小，但 SDD 的计算时间甚至比 SVD 还要长，因此，该方法依然不适合应用于大规模数据集上。Karypis 等人提出了一种名为概念索引(Concept Indexing, CI)的方法，利用聚类中心向量进行降维[5]；Bingham 等人则使用随机映射(Random Projection, RP)对原始数据进行降维[6]。他们都取得了与 LSI 相近的检索效果。Tang 等人提出了一种名为 eLSI 的方法来降低 LSI 计算中 SVD 过程的复杂度[3]。这种方法首先将数据集中的所有文档进行聚类，然后对每个词语算出其聚合权重(Aggregate Weight, AW)，根据这个权重选出一小部分词语，从而达到降维的效果。试验结果证明，eLSI 算法在 TREC 数据集上的效果明显超过了 CI 和 RP 算法。然而，作者忽视了在大规模数据集上进行聚类本身就是一个计算复杂度非常高的任务，因此，eLSI 方法在实际中的应用会受到很大的限制。本文提出的方法避免了上述方法的缺陷，以很小的计算复杂度有效地逼近了 LSI 的效果。

3 快速潜在语义索引

本文简单地介绍了 LSI 方法的形式化定义以及特征提取和特征选择这 2 种降维方法的同意框架。在此基础上提出了一种将 LSI 方法的解空间由连续空间映射到离散空间的方法，从而降低了计算 LSI 的复杂度，同时有效地逼近了 LSI 的效果。

3.1 潜在语义索引

在本文中，一个由 n 篇文档和 m 个词语组成的数据集可以表示成一个 $m\times n$ 的词语-文档矩阵 $\boldsymbol{X}\in\boldsymbol{R}^{m\times n}$。这个矩阵中的每一列都是一篇文档，都由一个 m 维的向量 $\boldsymbol{x}_i\in\boldsymbol{R}^m$ 表示，向量的每个分量 x_i^j 代表一个词语在这篇文档中的权重，本文是用经典的 TF-IDF 算法计算出权重。

LSI 的核心思想是通过矩阵变换将词语和文档映射到一个低维的向量空间，称之为潜在语义空间(Latent Semantic Space, LSS)，通常这种映射是通过截断的奇异值分解(Truncated Singular Value Decomposition, TSVD)完成的。在线性代数中，矩阵的奇异值分解是将一个秩为 r 的矩阵 $\boldsymbol{X}$ 分解为 2 个正交矩阵和一个对角矩阵的乘积

$$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}$$

其中，

$$\boldsymbol{U}=\{u_1,u_2,\cdots,u_r\}\in\boldsymbol{R}^{m\times r};\boldsymbol{V}=\{v_1,v_2,\cdots,v_r\}\in\boldsymbol{R}^{n\times r};$$

$$\boldsymbol{S}=diag(\sigma_1,\sigma_2,\cdots,\sigma_r)\in\boldsymbol{R}^{r\times r}。$$

这里 $\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{I},\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}=\boldsymbol{I},\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_r$ 是矩阵 $\boldsymbol{X}$ 的 r 个特征值。而 LSI 的做法是在 r 个特征值中选出 k 个最大的值，然后选出正交矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 中对应的 k 个行向量，最后得到原矩阵的一个逼近。

$$\boldsymbol{X}_k=\boldsymbol{U}_k\boldsymbol{S}_k\boldsymbol{V}_k^{\mathrm{T}}$$

可以证明 $\boldsymbol{X}_k$ 是原矩阵 $\boldsymbol{X}$ 在最小二乘意义上的最优逼近。

完成 SVD 分解之后，就可以将原词语-文档矩阵映射到低维语义空间中

$$\boldsymbol{Y}=\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{X}\in\boldsymbol{R}^{k\times n}$$

3.2 降维问题的统一框架

降维问题可以看成寻找从高维空间到低维空间的某种映射(在本文中只限于研究线性映射的情况)，具体来说是寻找这样一个满足某个目标函数 $J(\boldsymbol{W})$的转换矩阵 $\boldsymbol{W}\in\boldsymbol{R}^{m\times k}$，使得原矩阵 $\boldsymbol{X}\in\boldsymbol{R}^{m\times n}$ 可以被映射成 $\boldsymbol{Y}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\in\boldsymbol{R}^{k\times n}$。降维方法一般分为特征提取和特征选择 2 大类，两者之间主要的区别在于前者的解空间是连续的，而后者的解空间必须是离散的。

特征提取算法包括主成分分析、线性判别分析和潜在语义索引等经典算法，它们都可以表达成如下的形式：

$$\boldsymbol{W}^*=\arg\max_{\boldsymbol{W}\in\boldsymbol{H}_{fe}}J_{fe}(\boldsymbol{W})$$

$$\boldsymbol{H}_{fe}=\{\boldsymbol{W}\in\boldsymbol{R}^{m\times k},\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}\}$$

也就是说，特征提取算法的解空间中包括了所有 $m\times k$ 的正交实矩阵。

特征选择算法则包括信息增益、χ^2 准则等经典算法，它们可以形式化成

$$\boldsymbol{W}^*=\arg\max_{\boldsymbol{W}\in\boldsymbol{H}_{fs}}J_{fs}(\boldsymbol{W})$$

其中，解空间 H_{fs} 包括了所有满足以下条件的 $m\times k$ 的正交实矩阵 $\boldsymbol{W}$：

(1)$\boldsymbol{W}$ 中每个元素非 0 即 1。

(2)每一列包含且仅包含一个非零元素。

(3)每一行至多包含一个非零元素。

相对特征提取算法的解空间，称特征选择算法的解空间是离散的。

有了上述的定义，就可以给出降维问题的一个统一框架如下：给定数据矩阵 $\boldsymbol{X}$，针对某个目标函数 $J(\boldsymbol{W})$在解空间 H 中找到某个转换矩阵 $\boldsymbol{W}$ 使得 $J(\boldsymbol{W})$取值最优。

3.3 统一框架下的LSI

有了上述的框架，便可以将 LSI 表达成一个优化问题：给定矩阵 $\boldsymbol{C}=\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$，LSI 解得的转换矩阵 $\boldsymbol{U}$ 正是优化问题 $\boldsymbol{W}^*=\arg\max_{\boldsymbol{W}\in H_{fe}}tr\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{W}\}$ 的最优解。证明如下：

首先有 $\boldsymbol{C}=\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}=\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\cdot\boldsymbol{V}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{U}\boldsymbol{S}^2\boldsymbol{U}^{\mathrm{T}}$，也就是说，对矩阵 $\boldsymbol{C}$ 进行 SVD 得到的转换矩阵正是对 $\boldsymbol{X}$ 计算 LSI 的解。

另一方面，对于转换矩阵 $\boldsymbol{W}=\{w_1,w_2,\cdots,w_k\}$，满足 $\boldsymbol{w}_i^{\mathrm{T}}w_j=\begin{cases}1 & i=j\\ 0\end{cases}$。

对于优化目标函数 $\boldsymbol{W}^*=\arg\max_{\boldsymbol{W}\in H_{fe}}tr\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{W}\}$ 而言，其拉格朗日算符可以写作

$$L(\boldsymbol{w}_p,\lambda_p)=\sum_{q=1}^{k}\boldsymbol{w}_q^{\mathrm{T}}\boldsymbol{C}\boldsymbol{w}_q-\lambda_p(\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{w}_p-1)$$

L 取最值的条件是 $\dfrac{\partial L(\boldsymbol{w}_p,\lambda_p)}{\partial\boldsymbol{w}_p}=(\boldsymbol{C}-\lambda_p\boldsymbol{I})\boldsymbol{w}_p=0$，从而解得 $\boldsymbol{C}\boldsymbol{w}_p=\lambda_p\boldsymbol{w}_p$。也就是说，当且仅当矩阵 $\boldsymbol{W}$ 由矩阵 $\boldsymbol{C}$ 的前 k 个特征向量构成时，目标函数 $tr\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{W}\}$ 取得最大值

$$tr\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{W}\}=\sum_{p=1}^{k}\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{C}\boldsymbol{w}_p=\sum_{p=1}^{k}\lambda_p\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{w}_p=\sum_{p=1}^{k}\lambda_p$$

因为 LSI 解得的转换矩阵 $\boldsymbol{U}$ 是由矩阵 $\boldsymbol{X}$ 的前 k 个特征向量构成，而矩阵 $\boldsymbol{X}$ 的前 k 个特征向量与矩阵 $\boldsymbol{C}$ 相同，所以 LSI 的解正是上述优化问题的解。

3.4 连续空间离散化

得到了 LSI 在降维问题统一框架下的表达形式后，就可以试着将其解空间离散化，将一个特征提取问题转化成一个特征选择问题，从而得到一种快速的潜在语义索引算法。其

形式化定义如下：

$$\boldsymbol{W}^{*}=\arg\max_{W\in H_{fs}}\boldsymbol{J}_{\mathrm{LSI}}(\boldsymbol{W})=\arg\max_{W\in H_{fs}}tr\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{W}\}$$

这里假设从原矩阵 $\boldsymbol{X}$ 中选出的 k 个特征的下标从小到大依次是 $p_1,p_2,\cdots,p_k$，根据前面的定义，有：

$$\boldsymbol{W}=(w_1,w_2,\cdots,w_k)=\{w_l^p\}\in H_{fs}$$

$$w_l^p=\begin{cases}1 & p=p_l\\ 0\end{cases},\quad l=1,2,\cdots,k$$

$$\boldsymbol{J}_{\mathrm{LSI}}(\boldsymbol{W})=tr(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{W})=\sum_{l=1}^{k}\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}_l=\sum_{l=1}^{k}\sum_{i=1}^{n}(x_i^{p_l})^2=\sum_{l=1}^{k}Score(p_l)$$

其中，把 $Score(p)=\sum_{i=1}^{n}(x_i^p)^2$ 定义为第 p 个特征的打分函数。很显然 $Score(p)$的值始终非负，因此，要对目标函数 $\boldsymbol{J}_{\mathrm{LSI}}(\boldsymbol{W})$ 取最大值，只需要所有 m 个特征的打分函数值中选出 k 个最大的即可。

4 实验结果

为了验证本文提出的快速潜在语义索引方法(FLSI)的效果，下面在几个经典的文本数据集上设计了一系列实验。通过这组实验，发现快速潜在语义方法不仅相对同类方法在效果上有明显的提高，而且有效降低了潜在语义索引所需的计算时间。

4.1 实验设置

笔者设计了 2 组实验来验证本文方法的有效性：第 1 组是相似性实验，用于验证经过特征选择之后得到的 LSI 近似跟最初的 LSI 结果的相似程度；第 2 组是检索实验，在经过 LSI 处理的数据集上使用一组标准查询进行测试，从另一个方面验证各种方法逼近 LSI 的效果。

4.1.1 数据集

选择 Reuters-21578 和 MED 这 2 个常用的文本数据集来进行实验，对原始数据进行预处理采用的是著名的 Lemur 工具包。

Reaters-21578 是在文本分类问题中最常使用的标准数据集之一，它包含了 2 066 个文档和 3 856 个词语(经 Lemur 工具包处理后)。而 MED 是著名的 SMART 检索系统中自带的一个数据集，它包含了 1 033 个文档和 5 735 个词语。

4.1.2 基准算法

将快速潜在语义索引方法与另外 2 种常用的无监督的特征选择算法进行了比较，它们分别是文档频率(Document Frequency, DF)算法和聚合权重(Aggregate Weight, AW)算法。DF 算法的基本思想是根据包含一个词的文档数目来决定这个词的权重。该算法的特点是实现简单，计算高效，可以很容易地扩展到大规模的数据集上。其计算公式可以写成：

$$DF(t)=\#(D_i\mid t\in D_i)$$

AW 算法是 Tang 等人提出用在 eLSI 中的，其基本思想是先对所有的文档做一遍 k-means 聚类，然后在得到的聚类中心矩阵 $\boldsymbol{M}=[m_1,m_2,\cdots,m_s]\in R^{m\times s}$ 上为每个词计算权重：

$$AW(t)=\sum_{i=1}^{s}m_i^t$$

其中，s 是指聚类中心的个数，而 m_i 是指第 i 个聚类中心。

4.1.3 关键步骤

在第 1 组实验中，按照以下的步骤来比较 LSI 的近似结果跟真实值的相似程度：

(1)在原数据 $\boldsymbol{X}$ 上计算 LSI，得到转换矩阵 $\boldsymbol{W}$。

(2)将 FLSI/DF/AW 等特征选择算法应用到数据 $\boldsymbol{X}$ 上，得到近似的转换矩阵 $\boldsymbol{W}_1$ 和降维后的矩阵 $\boldsymbol{Y}$。

(3)在已降维的矩阵 $\boldsymbol{Y}$ 上计算 LSI，得到转换矩阵 $\boldsymbol{W}_2$。

(4)采用 Cosine 距离计算 $\boldsymbol{W}_2\boldsymbol{W}_1$ 跟 $\boldsymbol{W}$ 的相似程度。

在第 2 组实验中，按照以下的步骤来测试 LSI 的近似结果在检索应用上的效果：

(1)在原数据 $\boldsymbol{X}$ 上计算 LSI，得到降维后的矩阵 $\boldsymbol{Y}_1$。

(2)先用 FLSI/DF/AW 等特征选择算法对 X 进行降维得到矩阵 $\boldsymbol{Y}_{21}$，然后在 $\boldsymbol{Y}_{21}$ 上计算 LSI，得到降维后矩阵 $\boldsymbol{Y}_{22}$。

(3)在 $\boldsymbol{X}$，$\boldsymbol{Y}_1$ 和 $\boldsymbol{Y}_{22}$ 上进行查询，通过准确率(precision)和检出率(recall)对查询的结果进行度量。

4.2 实验结果

下面分别讨论快速潜在语义索引方法在相似性实验和检索实验中的效果。

4.2.1 相似性实验

在 Reuters-21578 和 MED 数据集上首先选出 1 000 个词语以降低原始数据的维度，再运行 LSI 得到前 200 个特征向量。接下来这些特征向量分别与直接计算 LSI 得到的前 200 个特征向量一一计算 Cosine 距离，用来度量每种特征选择算法逼近 LSI 的程度。图 1 和图 2 分别是 FLSI、DF 和 AW 3 种特征选择算法在 2 个数据集上的结果，横坐标表示的是取前多少个特征向量计算相似度，纵坐标表示的是用 Cosine 距离计算的相似度。从图中很容易看到，FLSI 的逼近效果明显好于 DF 和 AW 算法，特别是在前 50 个特征向量上，相似度均超过了 0.9，平均相似度要高出 50%。

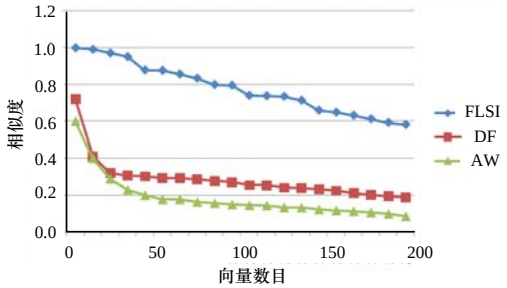

图 1　3 种特征选择算法在 Reuters-21578 数据集上的近似结果

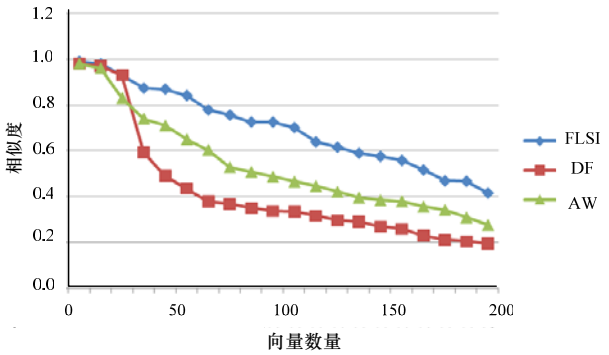

图 2　3 种特征选择算法在 MED 数据集上的近似结果

4.2.2 检索实验

在 MED 数据集上验证了降维后数据对检索效果的影响，实验的设置依然是用特征选择算法选出 1 000 个词语，然后用 LSI 降维到 200 维。图 3 中“NAIVE”的曲线代表的是直接在 MED 原始数据集上查询的结果，“LSI”的曲线是在直接运行 LSI 后的数据集上的查询结果，可以看到后者的效果明显好于前者，这与文献中的结果一致。而 FLSI/DF/AW3 条曲线分别代表用 3 种特征选择算法降维后的数据上查询的结

(下转第 40 页)