一种数据挖掘中的 W-PAM 限制聚类算法

张 松 张 琳
(南京邮电大学计算机学院 南京 210003)

摘 要 在数据挖掘中由于每个数据对象对于知识发现的作用是不同的,为了区分这些相异之处,给每个对象赋予一定量的值,因此在 PAM 聚类算法的基础上提出一种 W-PAM(Weight Partitioning Around Medoids)聚类算法,它为簇中数据对象加入权重来提高算法的准确率,此外利用数据对象间的关联限制能够提高聚类算法的效果。探讨了一种 W-PAM 算法与关联限制相结合的限制聚类算法,该算法同时拥有 W-PAM 算法和关联限制的优点。实验结果证明,W-PAM 的限制聚类算法可以更有效地利用所给的关联限制来改善聚类效果,提高算法的准确率。

关键词 数据挖掘,W-PAM,关联限制,限制聚类
中图法分类号 TP393 **文献标识码** A

W-PAM Restricted Clustering Algorithm in Data Mining

ZHANG Song ZHANG Lin
(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract In data mining, the effect of each data object on knowledge discovery is different. In order to distinguish these differences, this paper gave a certain amount of value to each object, and put forward a W-PAM (Weight Partitioning Around Medoids) clustering algorithm which is based on the PAM algorithm. It can improve the accuracy of the algorithm by adding weight to the data object in the cluster. Moreover, the effect of clustering algorithm can be improved by using the association among the data objects. In this paper, a W-PAM restricted clustering algorithm was proposed, which combines the W-PAM algorithm with the constraint clustering algorithm. The algorithm has advantages of the W-PAM restricted clustering algorithm and relevance constraints. The experimental results show that the W-PAM restricted clustering algorithm can effectively improve the clustering result and improve the accuracy of the algorithm.

Keywords Data mining, W-PAM, Association restriction, Restricted clustering

1 引言

聚类[1]是按照数据的某些属性,将数据对象分组成多个类或簇。聚类的任务是使得类内的对象尽可能地相似,类间的对象尽可能地相异。聚类分析已经广泛地应用于许多应用领域,包括商务智能、图像模式识别、Web 搜索、生物学和安全。随着信息技术、互联网技术的迅猛发展,各种事物之间的属性联系越来越深,人们也越来越认识到了主观因素在聚类过程中的重要性,如何将用户倾向加入聚类过程已经成为一个研究热点。

作为统计学[2]的一个分支,聚类分析已经被广泛研究,主要集中在基于距离的聚类分析。基于 k-均值(k-means)[3]、k-中心点(k-medoids)[4]和其他一些方法[5-9]的聚类分析工具已经被加入到许多统计分析软件包或系统中,例如 S-Plus, SPSS 以及 SAS。通常,在聚类分析时用户根据自己的倾向对数据进行限制,如两个数据对象应该分在一类或不应该分为一类,为了描述这种限制 Wagstaff,提出了必须关联限制(must-link constraint)和不能关联限制(cannot-link constraint)两种关联限制[10]。如果数据集上的两个对象 x 和 y 被指定属于同一类,那么 $Constraint(x,y):must\text{-}link(x,y)=$ True;如果两个对象 x 和 y 被指定不属于一类,那么 $Constraint(x,y):cannot\text{-}link(x,y)=$ True。可以根据数据之间的这种关联限制将它与聚类算法相结合[11],来提高算法的能力。近些年来,数据对象间的关联限制与传统聚类算法的结合已有了一些研究成果。其与 k-means、Complete-link、EM[12]、COBWE 都有多种结合方式,其中 Wagstaff 等人提出的 CKM 算法[13]和 Klein 等人提出的 CCL 算法[14]比较经典。它们都会在不同程度上改善原先算法。根据传统的 PAM 算法[15]本文提出了 W-PAM 限制聚类算法,它是在 PAM 算法的基础上对簇内数据对象加入权重[16]重新分类之后与关联限制相结合,在降低了 PAM 迭代次数的同时提高算法的准确率。

2 数据挖掘中的 W-PAM 限制聚类算法

2.1 理论分析

PAM 算法是一种基于中心点或中心对象进行划分的中

本文受国家自然科学基金(61402241,61572260,61373017,61572261,61472192),江苏省科技支撑计划(BE2015702)资助。
张 松(1990-),男,硕士生,主要研究方向为数据挖掘、信息安全、隐私保护等;张 琳(1980-),女,博士后,副教授,硕士生导师,主要研究方向为云计算、网络安全、信任、可信计算等,E-mail:zhangl@njupt.edu.cn。

心点算法。它用迭代、贪心方法处理该问题。与 k-均值算法一样，初始代表对象任意选取。算法考虑用一个非代表对象替代一个代表对象提高聚类质量。对所有可能进行替换，继续用其他对象替换代表对象的迭代过程，直到结果聚类的质量不可能被任何替换提高。质量用对象与其簇中代表对象的平均相异度的代价函数度量。

准则函数定义如下：

$$E=\sum_{i=1}^{k}\sum_{P_i\in C_j} dist(P_i,o_i)^2 \tag{1}$$

其中，E 是数据集中所有对象 P_i 与 C_i 的代表对象 o_i 的平方差之和。

2.2 代价计算

代价在聚类算法中起着重要的作用，它决定一个非代表对象是否代替一个代表对象。具体地说，设 $o_1,\cdots,o_k$ 是当前代表对象（即中心点）的集合，为了决定一个非代表对象 o_r 是否是一个当前中心点 $o_j(1\leqslant j\leqslant k)$ 的好的替代，计算每个对象 P 到这些代表对象和这个非代表对象中最近对象的距离，并使用该距离更新代价函数，然后对对象重新进行分配。假设一个代表对象 o_j 和一个非代表对象 o_r，通过每一个非代表对象 o_h 计算代价 C_{hjr} 来判断 o_r 是否可以替代 o_j。C_{hjr} 有以下 4 种结果。

(1)如果 o_h 最初属于 o_j 代表的那类，当 o_r 代替 o_j 之后，使得 o_h 距离 o_r 最近，那么进入 o_r 表示的类，代价 $C_{hjr}=d(o_h,o_r)-d(o_h,o_j)$。

(2)如果 o_h 最初属于 o_j 代表的那类，当 o_r 代替 o_j 之后，因为距离另一个代表对象 o_i 最近，那么进入 o_i 表示的类，代价 $C_{hjr}=d(o_h,o_i)-d(o_h,o_j)$。

(3)如果 o_h 不属于 o_j 代表的那类，假设 o_h 属于 o_i 代表的那类，当 o_r 代替 o_j 之后，仍然距离 o_i 最近，仍然在 o_i 那类中，代价 $C_{hjr}=0$。

(4)如果 o_h 最初属于 o_j 代表的那类，假设 o_h 属于 o_i 代表的那类，当 o_r 代替 o_j 之后，使得 o_h 距离 o_r 最近，那么进入 o_r 表示的类，代价 $C_{hjr}=d(o_h,o_r)-d(o_h,o_i)$。

根据上面的代价计算可以计算出 o_r 代替 o_j 的总代价 S，S 为各个非代表对象 o_h 的相应代价 C_{hjr} 之和。当 S 为正值时之前的分类不变；当 S 为负值时，说明 o_r 代替 o_j 是好的替代，然后进行重新分类。

2.3 W-PAM 算法及算法中权重的确定

W-PAM 算法的基本思想就是在传统 PAM 的基础上，对簇及簇中的每个对象计算加权平均值，将数据集中的每个对象重新赋给最类似的簇，进行反复迭代计算，直到准则函数收敛即质量不可能被任何替换提高。

首先为了区分簇中的各个对象可以对对象加入一个参数 W_i 即权重。参数 W_i 为：

$$W_i=\frac{w_i}{\sum_{i=1}^{n} w_i} \tag{2}$$

其中，$w_i=\frac{1}{n}\sum_{j=1}^{n} d(y_i,y_j)$，$d(y_i,y_j)$ 为 y_i 与 y_j 之间的距离。某簇中对象的权重越大，说明该对象与其它对象之间的距离越远或差异越大；相反某对象的权重越小，说明该对象与其它对象之间的距离越近或相似性越大。在比较密集的对象区域，由于距离中心点比较近，它们的权重也相差不大，在聚类时比较容易聚到一类。由于“噪声”和孤立点离中心比较远，计算权重时会比较大，为了减少“噪声”和孤立点对算法精确度的影响，将簇中的对象采用加权平均的方法来解决这一问题。

对簇进行加权平均计算方法为：$Q_j=\frac{1}{m}\sum_{i=1}^{m} W_iP_i$。其中，$Q_j(1\leqslant j\leqslant k)$ 代表簇 C_j 的加权平均值；P_i 代表 C_j 中的某个对象的数值；W_i 代表簇 C_j 中某个对象的权重；m 代表簇中对象的个数，不同的簇中含有的对象也不同。通过以上对簇中的对象进行处理后，可以更好地改善算法的结果，W-PAM 算法的实现过程如表 1 所列。

表 1 W-PAM 的具体算法和流程

输入：k 为结果簇的个数；D 为包含 n 个对象的数据集合

输出：加权平均后 k 个簇的集合

1. 从 D 中随机选择 k 个对象作为初始的代表对象或种子；
2. repeat；
3. 计算簇中每个对象的加权平均值，然后将处理过的对象分配到最近的代表对象所代表的簇；
4. 随机地选择一个非代表对象 o_r；
5. 通过加权平均计算用 o_r 代替代表对象 o_j 的总代价 S；
6. if $S<0$，then o_r 替换 o_j，形成新的 k 个代表对象的集合；
7. until 不发生变化；

准则函数定义如下：

$$E=\sum_{j=1}^{k}\sum_{P_i\in C_j} |P_i-Q_j| \tag{3}$$

其中，P_i 是簇中 C_i 的数据对象。通过准则函数结合代价的计算来对数据集进行簇的分配。

3 W-PAM 限制聚类的具体流程

可以将上面改进过的 W-PAM 算法与关联限制相结合，得到一种新的限制聚类算法。

这种限制聚类算法是一种半监督的聚类问题，处理限制数据和一般数据采用不同的方法，这种处理可以描述为：给出数据集 $Y=\{y_1,y_2,\cdots,y_n\}$，其中 y_i 是数据集中的所有数据对象，$Y=M\cup N$，M 表示限制数据，N 表示一般数据，假设将数据集分成 K 类如 $C_1,C_2,\cdots,C_K$，$M=\bigcup_{h=1}^{K} M_h$，且 $|M_h|>0$，$1\leqslant h\leqslant K$。其中 $M_h=\{y|Class(y)=C_h\}$，其中 $Class(y)$ 表示数据点的类别，目的是用少量的标记数据辅助聚类算法的实现，以提高聚类算法的精度。

下文用 $ML(x,y)=\text{True}$ 和 $CL(x,y)=\text{True}$ 来表示引言中提到的 $Constraint(x,y)$：$must\text{-}link(x,y)=\text{True}$，$Constraint(x,y)$：$cannot\text{-}link(x,y)=\text{True}$ 这两个限制函数，在下面的限制聚类算法中假设数据对象集为 Y，数据集中的限制数据集为 Y_h，若一个数据 $G\in Y_h$ 和数据集 Y 中另一个数据 H 分为一类，则 $ML(G,H)=\text{True}$；若数据对象 G 和数据对象 Y 不分为一类，则 $CL(G,H)=\text{True}$。

因为 W-PAM 限制聚类算法与 COP-Kmeans(CKM)算法有相同的缺陷，在聚类期间有时会产生 *Cannot-Link* 约束违反，此时不会产生一个有效的聚类划分，这里采用最常用的解决办法，即使用不同的随机初始化聚类中心，重新运算算法，在找到满足的限制分隔后，然后再进行迭代直至收敛或者

达到迭代次数为止，同时在迭代过程还要注意限制数据集中的数据对聚类的影响。该限制聚类算法中的数据对象和簇都是在 W-PAM 算法处理的前提下进行限制聚类的。W-PAM 限制聚类算法的实现过程如表 2 所列。

表 2　W-PAM 限制聚类的具体算法和流程

输入：k 为结果簇的个数；D 为包含 n 个对象的数据集合；Must-Link 为限制集 ML；Cannot-Link 为限制集 CL；规定迭代次数 T

输出：限制聚类后的 k 个簇的集合

1. 根据 ML，CL 限制集，选择构成限制数据集 Y_h，设置选择次数上限 MAX_s；
2. 从 D 中随机选择 k 个对象作为初始的代表对象或种子；
3. Repeat；
4. 随机地从 D 中选择任一个非代表对象 o_r，计算 o_r 到各个簇 C_j 的距离即为 o_r 与簇中代表数据对象间的距离。如果某个簇中数据对象与 o_r 存在 CL 关系，则 o_r 与该簇距离为无穷大，在保证函数 Judge-Constraints(o_r，C_j，ML，CL) 成立的情况下，将 o_r 加入与之距离最小的簇；
5. 计算数据对象替换过后的总代价 S；
6. 如果 S<0 将数据进行替换，形成新的 k 个代表对象的集合；
7. until 算法收敛或达到迭代次数 T；

注：Judge-Constraints(Object o，Cluster C，Must-Link ML，Cannot-Link CL)

1)如果 o_i 是簇 C 中的对象，并且 ML(o_r，o_i)=True，则将 o_r 分入簇 C 中，返回步骤 4。

2)如果 o_r 与所有簇中的对象都存在 CL 关系，那么 o_r 不能分入这些簇中，如果没有达到选择上限次数 MAX_s，返回步骤 2，否则退出算法。

3)如果上述两种情况都没出现，那么函数成立，继续执行。

由于数据集中有一个限制数据集 Y_h，因此该限制聚类在获得总代价后的分类还有一种情况要考虑到，就是对于新的代表对象集合，所有非代表数据对象 o_r，如果 $o_r \in Y_h$，则各分簇不变；否则，计算 o_r 与各代表对象之间的距离，加入距离最近的簇。

该算法对代价计算上因为 Y_h 的存在也有一定的要求，假设用任一个非代表 o_r 代替某一个代表对象 o_j，计算总代价 S 与 W-PAM 算法计算总代价一样，S 为各个非代表对象的代价之和。但对于每一个非代表对象 o_h，若 $o_j \in Y_h$，则 C_{hjr} 的计算为：1)如果 o_h 属于 o_j 代表的簇，代价 $C_{hjr}=d(o_h,o_r)-d(o_h,o_j)$；2)如果 o_h 不属于 o_j 代表的簇，代价 $C_{hjr}=0$。

4　实验及结果分析

由于本文提出了两个部分的改进，第一部分是对 PAM 算法进行加权取权重计算，得到 W-PAM 算法；第二部分是得到 W-PAM 算法后再与限制聚类算法相结合。因此本次实验包括两个部分，首先对 PAM [15] 和 W-PAM 的正确率和迭代次数进行分析；其次以传统算法 CKM[12]、CCL [13] 作为参照和 W-PAM 下的限制聚类进行准确率分析。

由于各数据集的属性特点有所不同，为了便于算法的比较和分析，本文从 UCI 机器学习数据库中选取了 Iris，Glass，Wine，Sonar 4 个典型数据集进行实验。其中将 Iris，Glass 和 Wine 3 个数据集用于第一部分实验，将 Iris，Wine 和 Sonar 3 个数据集用于第二部分实验。Iris 数据集由 3 类 150 个实例组成，每个实例包含 4 个属性；Glass 数据集由 6 类 214 个实例组成，每个实例包含 9 个属性；Wine 数据集由 3 类 178 个实例组成，每个实例包含 13 个属性；Sonar 数据集由 2 类 208 个实例组成，每个实例包含 60 个属性。4 个数据集的详细信息如表 3 所列。

表 3　数据集的构成描述

数据集	属性个数	实例数	类别数
Iris	4	150	3
Glass	9	214	6
Wine	13	178	3
Sonar	60	208	2

使用 PAM 和 W-PAM 聚类算法在 Iris，Wine 和 Sonar 数据集上运行，为提高准确性和排除偶然性，将算法在每个数据集上运行 10 次记录平均值，所得结果如表 4－表 6 所列。

表 4　在 Iris 数据集上的聚类结果

数据集	聚类算法	正确聚类数	错误聚类数	正确率	迭代数
Iris	PAM	114.2	35.8	0.761	21
	W-PAM	126.4	23.6	0.843	16

表 5　在 Glass 数据集上的聚类结果

数据集	聚类算法	正确聚类数	错误聚类数	正确率	迭代数
Glass	PAM	148.7	65.3	0.695	33
	W-PAM	157.6	56.4	0.736	24

表 6　在 Wine 数据集上的聚类结果

数据集	聚类算法	正确聚类数	错误聚类数	正确率	迭代数
Wine	PAM	126.7	51.3	0.712	26
	W-PAM	132.2	45.8	0.743	19

由以上数据可以得到，引入权重后 W-PAM 算法比 PAM 算法有更高的正确率，W-PAM 迭代次数也比 PAM 低，通过结果可知 W-PAM 有效地降低了聚类结果对噪声和孤立点数据的敏感性，提高了聚类的正确率。

第二部分实验对 *Must-Link* 和 *Cannot-Link* 约束进行设置，限制集设置方法为：首先假设限制集为空，对要实验的数据集中的数据实例进行编号，然后限制集的发生采用随机数发生器，每次产生一对数字，每位数字对应数据集中编号相同的数据实例，之后分别检查数据实例所在的类别特征，如果两者类别相同，则设定它们为 *Must-Link* 约束，否则设定为 *Cannot-Link* 约束。如果限制集中没有包括此约束，则将它添加到限制集中去，直到限制集中的约束数量达到规定的数目。在设定限制前，初始化随机发生器，每次采用不同的数据进行初始化，因此可以得到不同的限制集。在实验中对于每个特定的约束数进行 20 次实验，多次实验准确率的均值为最后的值，若迭代时有无法分配的情况，那么认为算法失败，然后重新选择约束，在实验时将最大迭代次数记为 1000，如果超过了最大次数也记为失败，重新选择约束。

实验时 W-PAM 限制聚类算法的聚类质量用文献[10]中提出的 Rand 系数作为评价标准，原始数据和聚类结果都可以看作是对数据集的一个划分，因此，若是两两来观察，每个划分可以看作为 $n\cdot(n-1)$ 个分配情况，其中 n 代表数据集的大小。每个分配情况指是否将两个数据实例分配到同一类中，将聚类结果与原始数据集的真实情况进行比较，对每一个实例对，会出现 4 种可能情况：1)原始数据集中的类别相同，聚类后分配到同一类中；2)原始数据集中的类别相同，聚类后分配到不同的类中；3)原始数据集中的类别不同，聚类后分配到同一类中；4)原始数据集中的类别不同，聚类后分配到不同的类中。

上述出现的 4 种情况分别用 a,b,c,d 来表示，n 为数据集的大小。Rand 系数定义形式如下：

$$RI=\frac{a+d}{n\cdot(n-1)} \tag{4}$$

然后用 Iris，Wine 和 Sonar 数据集进行 W-PAM 限制聚类实验，并与 CKM，CCL 算法进行对比，如图 1－图 3 所示。

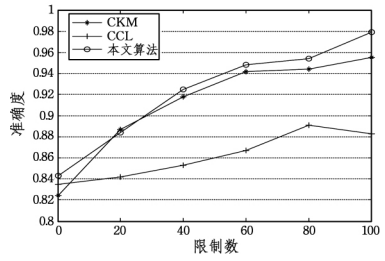

图 1　Iris 数据集

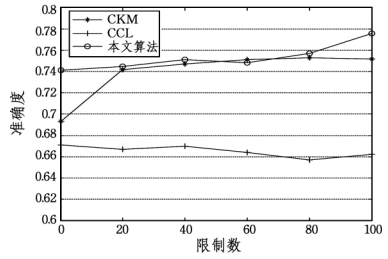

图 2　Wine 数据集

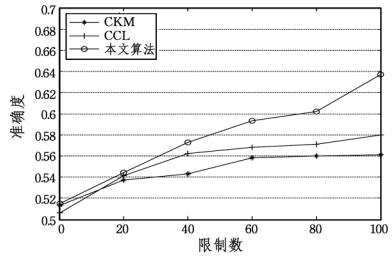

图 3　Sonar 数据集

从图 1－图 3 可以看出，W-PAM 限制聚类算法在聚类质量上有一定的改进，从总体上看，在限制数较小时聚类性能比较接近，随着限制数的增大本文算法更有优势。从 Iirs 数据集和 Sonar 数据集上看比较明显，Wine 数据集上变化不大但也有所提高。

结束语　本文通过结合传统的 PAM 聚类算法和经典的限制聚类算法，提出了一种 W-PAM 算法与关联限制相结合的限制聚类算法，实验结果表明：该算法可以初步处理一定规模的数据，在降低了 PAM 算法迭代次数的同时提高了算法的准确率。由于算法在寻找有效分隔时还存在不稳定性，在今后的学习中还需深入研究。

参 考 文 献

[1] 孙富贵，刘杰，赵连宁. 聚类算法研究[J]. 软件学报，2008，19(1)：48-61

[2] Vapnik V. Statistical Learning Theory [M]. New York：John Wiley，1998

[3] MacQueen J. Some methods for classification and analysis of multivariate observations [C]// Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley：University of California Press，1967：281-297

[4] Park H S，Jun C H. A simple and fast algorithm for K-medoids clustering [J]. Expert Systems with Applications，2009，36(2)：3336-3341

[5] 何萍，徐晓华，陆林，等. 双层随机游走半监督聚类[J]. 软件学报，2014，25(5)：997-1013

[6] 马儒宁，王秀丽，丁军娣. 多层核心集凝聚算法[J]. 软件学报，2013，24(3)：490-506

[7] Zhou Y，Wang X，Wang T，et al. Fault-tolerant multi-path routing protocol for WSN based on HEED[J]. International Journal of Sensor Networks，2016，20(1)：37

[8] 陈克寒，韩盼盼，吴健. 基于用户聚类的异构社交网络推荐算法[J]. 计算机学报，2013，36(2)：349-359

[9] 刘卓，杨悦，张健沛，等. 不确定度模型下数据流自适应网格密度聚类算法[J]. 计算机研究与发展，2014，51(11)：2518-2527

[10] Wagstaff K，Cardie C. Clustering with instance-level constraints [C]// Proc of the 17th International Conference on Machine Learning (ICML-2000). 2000：1103-1110

[11] Sun Jun，Zhao Wen-bo. Xue Jiangwei et al. Clustering with feature order preferences [J]. Intelligent Data Analysis，2010，14(4)：479-495

[12] M Law，A Topchy，A Jain. Clustering with Soft and Group Constraints [C]// Proc of Joint IAPR Int'l Workshop on Structural Syntactic and Statistical Pattern Recognition. 2004：662-670

[13] Wagstaff K，Cardie C，Rogers S，et al. Constrained K-means Clustering with Background Knowledge[C]// Proc of the 18th International Conference on Machine Learning (ICML-2001). 2001：577-584

[14] Klein D，Kamvar S，Manning C. Form Instance-level Constraints to Space-Level Constraints：Making the Most of Prior Knowledge in Data Clustering[C]// Proc of the 19 International Conference on Machine Learning (ICML-2002). 2002：307-314

[15] 韩家炜，堪博著. 数据挖掘：概念与技术[M]. 范明，孟小峰，译. 北京：机械工业出版社，2007：251-267

[16] 刘正，张国印，陈志远. 基于特征加权和非负矩阵分解的多视角聚类算法[J]. 电子学报，2016，44(3)：536-540

(上接第 446 页)

[5] Jing Li-ping，Ng M K，Huang J Z. Knowledge-based vector space model for text clustering [J]. Knowledge and Information Systems，2010，25(1)：35-55

[6] 刘海峰，刘守生，张学仁. 聚类模式下一种优化的 K-means 文本特征选择[J]. 计算机科学，2011，38(1)：195-197

[7] 雷军程，黄同成，柳小文. 一种基于权重的文本特征选择方法[J]. 计算机科学，2012，39(7)：250-252，275

[8] 张保富，施化吉，马素琴. 基于 TFIDF 文本特征加权方法的改进研究[J]. 计算机应用与软件，2011，28(2)：17-20

[9] 朱征宇，孙俊华. 改进的基于知网的词汇语义相似度计算[J]. 计算机应用，2013，33(8)：2276-2279，2288

[10] 王荣波，谌志群，周建政，等. 基于 Wikipedia 的短文本语义相关度计算方法[J]. 计算机应用与软件，2015，32(1)：82-85，92

[11] 宁亚辉，樊兴华，吴渝. 基于领域词语本体的短文本分类[J]. 计算机科学，2009，36(3)：142-145

[12] Batet M. Ontology-based semantic clustering[J]. Ai Communications，2011，24(3)：291-292

[13] 强保华，李巍，邹显春，等. 基于潜在语义分析的 Deep Web 查询接口聚类研究[J]. 计算机科学，2013，40(11)：228-230，247

[14] Xia Yan，Hua Zhao. Chinese Microblog Topic Detection Based on the Latent Semantic Analysis and Structural Property [J]. Journal of Networks，2013，8(4)：917-923

[15] Dumais S T. Latent semantic analysis [J]. Annual Review of Information Science & Technology，2008，3(11)：188-230