第四章

文本聚类算法的调查

查鲁-C。Aggarwal
IBM T. J. Watson研究中心 纽约州约克敦高地
恰ru@us.ibm.com

翟成祥
伊利诺伊大学厄巴纳-香槟分校 伊利诺伊大学厄巴纳分校
czhai@cs.uiuc.edu

摘要 聚类是文本领域中广泛研究的数据挖掘问题。该问题在客户细分、分类、协同过滤、可视化、文档组织和索引方面有许多应用。在本章中，我们将对文本聚类的问题进行详细的调查。我们将研究聚类问题的关键挑战，因为它适用于文本领域。我们将讨论用于文本聚类的关键方法，以及它们的相对优势。我们还将讨论该领域在社会网络和链接数据方面的一些最新进展。

关键字。文本聚类

1. 简介

聚类问题在数据库和统计学文献中被广泛研究，其背景是各种各样的数据挖掘任务[50, 54]。聚类问题被定义为寻找数据中类似对象的群体。这些对象之间的相似性

C.C. Aggarwal和C.X. Zhai（编辑），《*文本数据挖掘*》，DOI 10.1007/978-1-4614-3223-4_4。

用一个相似性函数来衡量这些对象。聚类问题在文本领域是非常有用的，其中要聚类的对象可以有不同的颗粒度，如文档、副图、句子或术语。聚类对于组织文档以改善检索和支持浏览特别有用[11, 26]。

对聚类问题的研究先于其对文本领域的适用性。传统的聚类方法通常集中在定量数据的情况下[44, 71, 50, 54, 108]，其中数据的属性是数字的。对于分类数据[10, 41, 43]，这个问题也被研究过，其中的属性可能是名义值。对聚类的广泛概述（因为它与一般的数字和分类数据有关）可以在[50, 54]中找到。一些适用于文本数据的常见文本聚类算法的实现可以在一些工具包中找到，比如*Lemur*[114]和*BOW*工具包[64]。聚类的问题在许多任务中都有适用性。

- **文件组织和浏览。**将文件分层组织成连贯的类别，对于系统浏览文件集非常有用。一个经典的例子是*Scatter/Gather*方法[25]，它提供了一个系统化的浏览技术，使用了文件集的聚类组织。

- **语料库总结。**聚类技术*以聚类摘要*[83]或*词组*[17, 18]的形式提供了一个连贯的摘要，可以用来对基础语料库的整体内容进行总结性的洞察。这类方法的变种，尤其是句子聚类，也可以用于文档总结，这个话题在第三章详细讨论。聚类的问题也与降维和主题建模的问题密切相关。这类降维方法都是对文档语料库进行总结的不同方式，在第五章中有所涉及。

- **文件分类。**虽然聚类本身是一种无监督的学习方法，但它可以被利用来提高其监督变体的结果质量。特别是，词组[17, 18]和协同训练方法[72]可以用来提高使用聚类技术的监督应用的分类精度。

我们注意到，许多类别的算法，如*k*-means算法，或分层算法，都是通用的方法，这些方法

可以扩展到任何类型的数据，包括文本数据。文本文件可以用二进制数据的形式来表示，当我们用文件中是否存在一个词来创建一个二进制向量。在这种情况下，可以直接使用各种分类数据聚类算法[10, 41, 43]来表示二进制。更加强化的表示方法包括基于文档中单个词的频率以及整个集合中词的频率的精炼加权方法（例如，TF-IDF加权[82]）。定量数据聚类算法[44, 71, 108]可以与这些频率结合使用，以确定数据中最相关的对象组。

然而，这种天真的技术通常不能很好地用于文本数据的分类。这是因为文本数据有一些独特的属性，需要为这个任务设计专门的算法。文本表示法的突出特点如下。

- 文本表示的维度非常大，但基础数据是稀疏的。换句话说，提取文档的词库可能有10个^5，但一个特定的文档可能只包含几百个词。当需要聚类的文档非常短时，这个问题就更严重了（例如，在聚类句子或推文时）。

- 虽然某个文档语料库的词库可能很大，但这些词通常是相互关联的。这意味着数据中的概念（或主成分）的数量远远小于特征空间。这就需要仔细设计算法，以便在聚类过程中考虑到词的相关性。

- 不同文件中的字数（或非零条目）可能有很大差异。因此，在聚类任务中，适当地规范文档的表示是很重要的。

不同文档的稀疏和高维表示使得设计针对文本的文档表示和处理算法成为必要，这也是信息检索文献中大量研究的主题，在这些文献中已经提出了许多技术来优化文档表示，以提高文档与查询匹配的准确性[82, 13]。这些技术中的大多数也可以用来改善文档的表示方法，以便进行聚类。

为了实现有效的聚类过程，需要根据词频在文档和整个集合中的相对出现频率对其进行归一化。一般来说，文本处理中常用的表示方法是*基于向量空间的*TF-IDF表示[81]。在TF-IDF表示法中，每个词的词频都被*反文档频率*（IDF）归一化。反文档频率的归一化降低了集合中更频繁出现的术语的权重。这降低了集合中常见术语的重要性，确保文档的匹配更多地受到集合中频率相对较低的、更具辨别力的词的影响。此外，一个亚线性转换函数经常被应用于术语频率，以避免任何可能在文档中非常频繁的单一术语的不良支配作用。关于文档规范化的工作本身就是一个巨大的研究领域，讨论不同规范化方法的各种其他技术可以在[86, 82]中找到。

文本聚类算法分为多种不同类型，如聚类算法、分区算法和基于标准参数建模的方法，如EM算法。 此外，文本表示法也可以被视为字符串（而不是词包）。这些不同的表示方法使得设计不同类别的聚类算法成为必要。不同的聚类算法在有效性和效率方面有不同的权衡。对不同聚类算法的实验比较可以在[90，111]中找到。在本章中，我们将讨论各种通常用于文本聚类的算法。我们还将讨论相关场景下的文本聚类算法，如动态数据、基于网络的文本数据和半监督场景。

本章的组织结构如下。在第2节中，我们将介绍用于文本聚类的特征选择和转换方法。第3节描述了一些常用的算法，这些算法被用于基于距离的文本文档聚类。第4节描述了使用单词模式和短语进行聚类的方法。第5节描述了对文本流进行聚类的方法。 第6节描述了对文本数据进行概率聚类的方法。 第7节描述了在社会或基于网络的背景下自然发生的文本聚类的方法。第8节讨论了半监督聚类的方法。第9节介绍了结论和总结。

2. 文本聚类的特征选择和转换方法

任何数据挖掘方法，如分类和聚类，其质量在很大程度上取决于用于聚类过程的特征的噪声。 例如，常用的词，如 *"the"*，在提高聚类质量方面可能不是很有用。因此，有效地选择特征是非常关键的，这样在聚类之前就可以把语料库中的噪声词去除。除了*特征选择*，还有一些特征转换方法，如潜在语义索引（LSI）、概率潜在语义分析（PLSA）和非负矩阵分解（NMF），可以提高文档表示的质量，使其更适合于聚类。在这些技术中（通常称为维度重构），词库中的词之间的相关性被利用来创建特征，这些特征对应于数据中的概念或原理成分。在本节中，我们将讨论这两类方法。关于降维的更深入的讨论可以在第五章找到。

2.1 特征选择方法

在文本分类问题中，特征选择是比较常见和容易应用的[99]，在这种情况下，特征选择过程有监督。然而，一些简单的无监督方法也可用于文本聚类中的特征选择。下面将讨论这类方法的一些例子。

2.1.1 基于文档频率的选择。 在文档聚类中，最简单的可能的特征选择方法是使用*文档频率*来过滤掉不相关的特征。虽然使用逆向文档频率可以降低这类词语的重要性，但仅凭这一点可能还不足以降低非常频繁的词语的噪音影响。换句话说，在语料库中过于频繁的词可以被删除，因为它们通常是常见的词，如 "a"、"an"、"the "或 "of"，从聚类的角度来看，这些词是没有辨别力的。这样的词也被称为*停止词*。在文献[76]中，通常有各种方法可以用来去除停止词。通常情况下，常用的约300到400个词的停止词列表被用于检索过程。此外，那些出现频率极低的词也可以从集合中删除。这是因为这些词对大多数聚类方法中使用的相似度计算没有任何帮助。在

在某些情况下，这些词可能是文件中的拼写错误或印刷错误。来自网络、博客或社交网络的噪音文本集更有可能包含这样的术语。我们注意到，一些研究路线纯粹是根据非常不常见的词汇来定义基于文档频率的选择，因为这些词汇对相似度计算的贡献最小。然而，应该强调的是，非常频繁的词也应该被删除，特别是如果它们在集群之间没有区别性。请注意，TF-IDF加权方法也可以自然地以 "软 "方式过滤掉非常常见的词。显然，标准的停顿词集提供了一个有效的词集来修剪。然而，我们希望有一种方法可以直接量化一个词对聚类过程的重要性，这对更积极的修剪是至关重要的。我们将在下面讨论一些这样的方法。

2.1.2 术语强度。 在[94]中提出了一种更为积极的删除停顿词的技术。这种方法的核心思想是将监督学习中使用的技术扩展到无监督情况下。术语强度基本上是用来衡量一个词对于识别两个相关文件的信息量有多大。例如，对于两个相关的文档x和y，术语t的强度$s(t)$是用以下概率来定义的。

$$s(t) = P(t \in y \mid t \in x) \tag{4.1}$$

很明显，主要问题是如何定义文件x和y
作为相关的。一种可能性是使用人工（或用户）反馈来定义
当一对文件是相关的。 这基本上等同于在特征选择过程中利用监督，并且在有预定义的文件类别的情况下可能是实用的。另一方面，在大型文集中以全面的方式手工创建相关对并不实际。因此，最好是使用一种自动化的、纯粹无监督的方式来定义一对文档何时相关的概念。在[94]中已经表明，可以使用自动化的相似性函数，如余弦函数[81]来定义文档对的关联性。如果一对文档的余弦相似度高于用户定义的阈值，则定义为相关。在这种情况下，术语强度$s(t)$可以通过随机抽取一些这样的相关文档对来定义，具体方法如下。

$$s(t) = \frac{t\text{同时出现在两种情况下的配对数量}}{t\text{出现在一对中的第一个的对子数量}} \tag{4.2}$$

在这里，一对文件中的第一个文件可以简单地被随机挑选。为了删减特征，可以将术语强度与

一个随机分布在训练文档中的具有相同频率的术语的预期强度。如果t的词语强度不比随机词语的强度至少大两个标准差，那么它就被从集合中删除。

这种方法的一个优点是，它不需要为特征选择提供初始的超视距或训练数据，这是在无监督情况下的一个关键要求。当然，这种方法也可以用于有监督的聚类[4]或分 类 [100]中的特征选择，当这种训练数据确实可用时。关于这种特征选择方法的一个观察结果是，它特别适合于基于相似性的聚类，因为未推导出的特征的判别性质是基于文档本身的相似性来定义的。

2.1.3 基于熵的排名。基于熵的排名方法是在[27]中提出的。在这种情况下，术语的质量是由删除它时的熵值减少来衡量的。这里，术语t在n个文档集合中的熵$E(t)$定义如下。

$$E(t) = -\sum_{i=1}^{n}\sum_{j=1}^{n}(S_{ij}\cdot\log(S_{ij}) + (1 - S_{ij})\cdot\log(1 - S_{ij})) \tag{4.3}$$

这里$S_{ij} \in (0, 1)$是去除术语t后，集合中第i个 和第j个文档的相似度，定义如下。

$$S_{ij} = 2^{-\frac{\text{远离}(i,j)}{\text{疏远}}} \tag{4.4}$$

这里$dist(i, j)$是删除术语t后i和j之间的距离，$dist$是删除术语t后文档之间的平均距离。我们注意到，每个术语t的$E(t)$的计算需要$O(n^2)$操作。这对于一个包含许多术语的非常大的语料库来说是不切实际的。在[27]中已经表明，通过使用抽样方法，这种方法可以变得更有效率。

2.1.4 定期缴款。 定期缴款的概念
[62]是基于这样一个事实：文本聚类的结果高度依赖于文档的相似度。因此，一个术语的贡献可以被看作是它对文档相似性的贡献。例如，在基于点积的相似性的情况下，两个文档之间的相似性被定义为其归一化频率的点积。因此，一个术语对两个文档的相似性的贡献是它们在两个文档中的正常化频率的乘积。这

需要对所有的文件对进行求和，以确定术语的贡献。和前面的情况一样，这种方法对每个词需要$O(n^2)$的时间，因此可能需要抽样方法来加快贡献率。 这种方法的一个主要批评是，它倾向于倾向于高频率的词，而不考虑聚类过程中的具体判别能力。

在大多数这些方法中，术语选择的优化是基于一些预先假定的相似度函数（如余弦）。虽然这种策略使这些方法成为无监督的，但由于假定的相似性函数的潜在偏差，人们担心术语选择可能会有偏差。也就是说，如果假设一个不同的相似性函数，我们最终可能会得到不同的术语选择结果。因此，选择一个合适的相似性函数对这些方法来说可能很重要。

2.2 基于LSI的方法

在特征选择中，我们试图从原始数据集中明确地选择出特征。特征转换是一种不同的方法，其中新的特征被定义为原始数据集中特征的功能表示。最常见的一类方法是降维方法[53]，其中文档被转换到一个维度较小的新特征空间，其中的特征通常是原始数据中特征的线性组合。诸如潜在语义索引（LSI）[28]等方法都是基于这一共同原理。总的效果是去除数据中大量的二维空间，这些空间对于基于相似性的应用（如聚类）来说是有噪音的。去除这些维度也有助于放大基础数据中的语义效应。

由于LSI与*主成分分析（PCA）*或*单值分解（SVD）*问题密切相关，我们将首先讨论这种方法，以及它与LSI的关系。对于一个d维数据集，PCA构建了数据的对称d d协方差矩阵C，其中(i, j)第1个条目是维度i和j之间的协方差。这个矩阵是正半无限的，可以对角化如下。

$$c = p - d - p^T \tag{4.5}$$

这里P是一个矩阵，其列包含C的正态特征向量，D是一个对角矩阵，包含相应的特征值-ues。我们注意到，特征向量代表了一个新的正态基础系统，数据可以沿着这个系统表示。在这种情况下，当数据沿着这个基系投影时，特征值对应于方差。这个基数系统也是一个第二

数据的阶次协方差被去除，数据中的大部分方差被保留最大的特征值的特征向量所捕获。 因此，为了降低数据的维度，一个常见的方法是用这个新的基数系统来表示数据，通过忽略那些相应的特征值很小的特征向量来进一步截断数据。这是因为沿着这些维度的方差很小，而且数据点的相对行为不会因为把它们从考虑中删除而受到很大影响。事实上，可以证明数据点之间的欧几里得距离不会受到这种转换和相应截断的影响。PCA的方法通常用于数据库检索应用中的相似性搜索。

LSI与PCA非常相似，只是我们使用了协方差矩阵C的近似值，这对文本数据的稀疏和高维性质非常合适。具体来说，让A是n d术语-文档矩阵，其中(i, j)项是文档i中术语j的归一化频率。然后，$A^T A$是一个d d矩阵，它是协方差矩阵的近似（按比例），其中的平均值没有被减去。换句话说，如果数据是以平均数为中心的，那么$A^T A$的值将与协方差矩阵的缩放版本（系数n）相同。虽然文本表述不是以平均数为中心的，但文本的稀疏性确保了使用$A^T A$是对（按比例）协方差的一个相当好的近似。与数字数据的情况一样，我们使用$A^T A$的特征向量，以表示文本的最大变量。在典型的集合中，只需要大约300到400个特征向量来表示。LSI[28]的一个突出特点是，维度的截断消除了同义词和多义词的噪音影响，相似性计算更接近于数据中的语义概念。这对于像文本聚类这样的语义应用特别有用。然而，如果需要更精细的聚类，这种文本的低维空间表示可能没有足够的辨别力；在信息检索中，这个问题通常通过将低维表示与原始的高维基于词的表示混合来解决（例如，见[105]）。

与LSI类似，但基于概率建模的技术是概率潜在语义分析（PLSA）[49]。PLSA和LSI的相似性和等价性在[49]中讨论。

2.2.1 使用聚类的概念分解。

一个有趣的观察是，虽然特征转换经常被用作聚类的预处理技术，但聚类本身可以是

用于一种新的降维技术，即*概念分解*[2, 29]。当然，这导致了在使用这种技术进行聚类时的循环问题，特别是在需要聚类以进行降维的情况下。尽管如此，通过使用两个独立的聚类阶段，仍然可以有效地使用这种技术进行预处理。

概念分解技术在文档的原始表示上使用任何标准的聚类技术[2, 29]。这些聚类中心的频繁术语被用作*基向量*，它们几乎是相互正交的。然后，文件可以用这些基向量更简洁的方式来表示。我们注意到，这种浓缩的概念表示法允许加强聚类和分类。因此，第二阶段的聚类可以应用在这个缩小的表示上，以便更有效地对文件进行分类。这种方法在[87]中也被测试过，它使用词组来表示文档。我们将在本章后面更详细地描述这种方法。

2.3 非负矩阵分解

非负矩阵分解（NMF）技术是一种潜在的空间方法，特别适用于聚类[97]。与LSI的情况一样，NMF方案在一个新的轴系中表示文档，该轴系是基于对术语-文档矩阵的分析。然而，从概念的角度来看，NMF方法与LSI方案有一些关键的区别。特别是，NMF方案是一种特别适合于聚类的特征转换方法。NMF方案的主要概念特征，与LSI方案有很大的不同，具体如下。

- 在LSI中，新的基础系统由一组正交矢 量 组成。而NMF则不是这种情况。

- 在NMF中，基础系统中的向量直接对应于集群主题。因此，一个文件的集群成员资格可以通过检查该文件沿任何一个向量的最大分量来确定。任何文档沿向量的坐标总是非负的。将每个文档表达为基础语义的加法组合，从直观的角度来看有很大的意义。因此，NMF变换特别适合于聚类，而且它还提供了对基础系统在聚类方面的直观理解。

让A是n d术语文档矩阵。让我们假设我们希望从基础文档语料库中创建k个聚类。然后，非负矩阵分解方法试图确定使以下目标函数最小的矩阵U和V。

$$j = (1/2) \cdot ||a - u \cdot v^T|| \tag{4.6}$$

这里||表矩阵中所有元素的平方之和，U是一个n k非负矩阵，V是一个m k非负矩阵。我们注意到，V的列提供了对应于k个不同聚类的k个基向量。

上述优化问题的意义是什么？请注意，通过最小化J，我们试图将A的因式分解近似为。

$$a \approx u \cdot v^T \tag{4.7}$$

对于A的每一行a（文件向量），我们可以将上述等 式 改写为。

$$a \approx u \cdot V^T \tag{4.8}$$

这里u是U的对应行。因此，文档向量a可以被改写为基向量的近似线性（非负）组合，它对应于V^T 的k列。如果k的值与语料库相比相对较小，那么只有当V的列向量^T ，才能发现数据中的潜在结构。此外，矩阵U和V的非负性确保文档被表达为基于术语的特征空间中关键概念（或聚类）区域的非负性组合。

接下来，我们将讨论上述J的优化问题究竟是如何解决的。任何矩阵Q的平方规范都可以表示为矩阵Q的迹线 Q Q^T 。因此，我们可以将上面的目标函数表达为：。 -

$$J = (1/2) \cdot tr((A - U \cdot V^T) \cdot (A - U \cdot V^T)^T)$$
$$= (1/2) \cdot tr(A \cdot A^T) - tr(A \cdot U \cdot V^T) + (1/2) \cdot tr(U \cdot V^T \cdot V \cdot U^T)$$

因此，我们有一个关于矩阵U=[u_{ij}]和V=[v_{ij}]的优化问题，其中的条目u_{ij} 和v_{ij} 是我们需要优化这个问题的变量。此外，由于矩阵是非负的，我们有这样的约束：$u_{ij} \geq 0$和
$v_{ij} \geq 0$。这是一个典型的受限非线性优化问题。
并可以用拉格朗日方法求解。让α = [α_{ij}]和β =
[β_{ij}]分别是与U和V具有相同维度的矩阵。矩阵α和β的元素是相应的 Lagrange

分别为U和V的不同元素的非负性条件的乘数。我们注意到，$tr(\alpha U^T)$简单地等于$-\sum_{ij}\alpha_{ij}u_{ij}$，$tr(\beta V^T)$简单地等于$-\sum_{ij}\beta_{ij}v_{ij}$。这些对应于的非负性约束的拉格朗日表达式。然后，我们可以将拉格朗日优化问题表达如下。

$$L = J + tr(\alpha - U^T) + tr(\beta - V^T) \tag{4.9}$$

然后，我们可以表示L相对于U的偏导和V如下，并将它们设为0。

$$\frac{\delta L}{\delta U} = -A - V + U - V^T - V + \alpha = 0$$
$$\frac{\delta L}{\delta V} = -A^T - U + V - U^T - U + \beta = 0$$

然后我们可以将上述（两个矩阵的）条件的第（i，j）项分别与u_{ij}和v_{ij}相乘。利用库恩-塔克条件$\alpha_{ij} - u_{ij} = 0$和$\beta_{ij} - v_{ij} = 0$，我们得到以下结果。

$$(A - V)_{ij} - u_{ij} - (U - V^T - V)_{ij} - u_{ij} = 0 \; (A^T - U)_{ij} - v_{ij} - (V - U^T - U)_{ij} - v_{ij} = 0$$

我们注意到，这些条件与α和β无关。这导致了以下u_{ij}和v_{ij}的迭代更新规则。

$$u_{ij} = \frac{(A - V)_{ij} - u_{ij}}{(U - V^T - V)_{ij}}$$
$$v_{ij} = \frac{(A^T - U)_{ij} - v_{ij}}{(V - U^T - U)_{ij}}$$

在[58]中已经表明，在这些更新规则下，目标函数不断提高，并收敛到一个最优解。

关于矩阵分解技术的一个有趣的观察是，它也可以用来确定单词集群而不是文档集群。就像V的列提供了一个可以用来发现文档集群的基础，我们可以用U的列来发现一个对应于词集群的基础。正如我们将在后面看到的，文档集群和词集群是密切相关的，同时发现这两个集群往往是有用的，就像在*co-clustering*[30, 31, 75]这样的框架中。矩阵因子化提供了一种实现这一目标的自然方式。理论上和实验上也表明[33, 93]，矩阵因素化技术等同于另一种基于图结构的文档聚类技术，即

作为*光谱聚类*。在[98]中提出了一种类似的技术，称为*概念因子法*，它也可以应用于其中有负值的数据点。

3. 基于距离的聚类算法

基于距离的聚类算法是通过使用相似度函数来衡量文本对象之间的接近程度。在文本领域中，最著名的相似性函数是余弦相似性函数。让$U = (f(u_1) \ldots f(u_k))$和$V = (f(v_1) \ldots f(v_k))$是两个不同文档$U$和$V$中的阻尼和归一化的频率词向量。$u_1 \ldots u_k$和$v_1 \ldots v_k$的值代表（归一化）术语频率，而函数$f()$代表阻尼函数。 典型的阻尼函数$f()$可以代表平方根或对数[25]。然后，两个文档之间的余弦相似度定义如下。

$$\text{余弦}(U, V) = \frac{\sum_{i=1}^{k} f(u_i) \cdot f(v)_i}{\sqrt{\sum_{i=1}^{k} f(u_i)^2} \cdot \sqrt{\sum_{i=1}^{k} f(v_i)^2}} \qquad (4.10)$$

文本相似性的计算是信息检索的一个基本问题。尽管信息检索的大部分工作都集中在如何评估关键词查询和文本文件的相似性，而不是两个文件之间的相似性，但许多加权启发式方法和相似性函数也可以用于优化聚类的相似性函数。有效的信息检索模式一般会抓住三种启发式方法，即TF加权、IDF加权和文档长度标准化[36]。当把文档表示为一个加权术语向量时，给术语分配权重的一个有效方法是BM25术语加权法[78]，其中归一化的TF不仅解决了长度归一化的问题，而且还有一个上限，这提高了鲁棒性，因为它避免了对任何特定术语的匹配进行过度的奖励。一个文档也可以用单词的概率分布来表示（即单片语言模型），然后可以根据信息论的测量方法，如交叉熵或Kullback-Leibler分歧来衡量相似性[105]。对于聚类，这种相似性函数的对称变体可能更合适。

对短小的文本片段（如推文或句子）进行聚类的一个挑战是，准确的关键词匹配可能效果不佳。解决这个问题的一个通用策略是通过利用相关的文本文档来扩展文本表示，这与信息检索中的文档语言模型的平滑化有关[105]。一个具体的

在[79]中提出了一种技术，利用搜索引擎来扩展文本的代表性。对计算短文段的相似性的几种简单措施的比较可以在[66]中找到。

这些相似性函数可以与各种传统聚类算法结合使用[50, 54]。在接下来的小节中，我们将讨论这些技术中的一些。

3.1 聚合和分层聚类算法

层次聚类算法在聚类文献中被广泛研究[50, 54]，用于不同类型的记录，包括多维数字数据、分类数据和文本数据。在文本数据方面，对传统的聚类和分层聚类算法的概述见[69，70，92，96]。不同层次聚类算法的实验比较可以在[110]中找到。聚合分层的方法对于支持各种搜索方法特别有用，因为它自然地创建了一个树状的层次结构，可以在搜索过程中加以利用。特别是，这种方法在提高搜索效率方面的有效性已经在[51,77]中得到证实。

聚合式聚类的一般概念是根据文档之间的相似性将其连续地合并到聚类中。几乎所有的分层聚类算法都是根据这些文件组之间的最佳配对相似度来连续合并分组的。这些类别的方法之间的主要区别在于如何计算不同组的文件之间的成对相似性。例如，一对组之间的相似性可以被计算为最佳情况下的相似性，平均情况下的相似性，或从这对组中抽取的文件之间的最差情况下的相似性。从概念上讲，将文档聚集成更高层次的聚类的过程创造了一个聚类层次结构（或树状图），其中叶子节点对应于单个文档，而内部节点对应于合并后的聚类组。当两个群组合并时，在这棵树上会创建一个新的节点，对应于这个较大的合并群组。这个节点的两个子节点对应于被合并到它的两个文档组。

对于不同的聚类方法，合并文件组的不同方法如下。

- **单一联系聚类。**在单链路聚类中，两组文件之间的相似度是这两组中任何一对文件之间的最大相似度。在单链路聚类中，我们合并了两组，这两组中最接近的一对文件与其他任何一对组相比具有最高的相似性。单链路聚类的主要优点是它在实践中的实施效率极高。这是因为我们可以首先计算出所有的相似度对，并按照降低相似度的顺序进行排序。这些配对按照这个预设的顺序进行处理，如果这些配对属于不同的组，则依次进行合并。可以很容易地证明，这种方法等同于单链路方法。这实质上等同于在完整的对偶距离图上，通过按一定顺序处理图中的边来进行生成树算法。在[92]中已经说明了Prim的最小生成树算法如何适应于单链路聚类。[24]中的另一种方法是将单链路方法与倒指数方法结合起来设计，以避免计算零相似度。

  这种方法的主要缺点是，它可能导致*连锁*现象，即一连串的类似文件导致不同的文件被归入相同的集群。换句话说，如果A与B相似，B与C相似，并不总是意味着A与C相似，因为在相似性计算中缺乏反证。单一联系聚类法鼓励通过这种转折性链对文件进行分组。这往往会导致糟糕的聚类，尤其是在聚类的较高层次。在文档数据的情况下，实现单链接聚类的有效方法可以在[24，92]中找到。

- **组平均联系聚类。**在群体平均联系聚类中，两个聚类之间的相似度是两个聚类中的文件对之间的*平均*相似度。显然，平均联结聚类的过程比单联结聚类要慢一些，因为我们需要确定大量文件对之间的平均相似度，以便确定组间的相似度。另一方面，它在聚类质量方面更为稳健，因为它不表现出单链式聚类的连锁行为。通过计算C_1和C_2的平均文档之间的相似度，可以加快平均联系聚类算法的速度，从而使两个聚类的平均联系相似度接近。[1]

和C的平均文档$_2$。虽然这种方法对所有的数据域都不一样好，但它对文本数据的情况特别好。在这种情况下，运行时间可以减少到$O(n^2)$，其中n是节点的总数。该方法在文档数据的情况下可以相当有效地实现，因为一个集群的中心点只是该集群中的文档的连接。

- **完全联结聚类法。**在这种技术中，两个聚类之间的相似度是两个聚类中任何一对文件的*最坏情况下的*相似度。完全联系聚类也可以避免连锁，因为它避免了将任何一对非常不相干的点放在同一个簇中。然而，和群组平均聚类一样，它在计算上比单链路方法更昂贵。完全联结聚类方法需要$O(n^2)$空间和$O(n^3)$时间。然而，在文本数据领域的情况下，空间要求可以大大降低，因为大量的成对相似性是零。

在文本数据流的背景下，分级聚类算法也被设计出来。[80]中提出了一种用于流式文档分层聚类的分布式建模方法。其主要思想是通过使用多泊松分布对文档中的单词出现频率进行建模。这个模型的参数被学习，以便将文档分配到集群中。 该方法将COBWEB和CLASSIT算法[37, 40]扩展到文本数据的情况。[80]中的工作研究了文档中单词的不同种类的分布假设。 我们注意到，要使这些算法适应文本数据的情况，需要这些分布假设。该方法实质上是改变了分布假设，使该方法能够有效地用于文本数据。

3.2 基于距离的分区算法

分区算法在数据库文献中被广泛使用，以便有效地创建对象的集群。两个最广泛使用的基于距离的分区算法[50, 54]如下。

- ***k-medoid*聚类算法。**在k-*medoid*聚类算法中，我们使用原始数据中的一组点作为锚（或medoids），围绕这些点建立聚类。该算法的关键目的是要*从原始语料库中*确定一个最佳的代表文件集，围绕该文件集建立集群。每个文档都被分配到其最接近的代表，这些代表来自于

的集合。这就从珊瑚中创建了一个运行中的集群，这些集群通过一个随机的过程被连续地改进。

该算法采用迭代的方法，在此过程中，*k*个代表的集合被连续地改进，并使用运行中的相互变化。具体来说，我们使用语料库中每个文档与其最接近的代表的平均相似度作为目标函数，在这个变化过程中需要改进。在每次迭代中，如果能提高聚类的目标函数，我们就用从集合中随机挑选的代表来替换当前集合中的一个代表。这种方法一直应用到达到收敛。

使用基于*k-medoids*的聚类算法有两个主要缺点，其中一个是针对文本数据的情况。*k*-medoids聚类算法的一个普遍缺点是，它们需要大量的迭代来实现收敛，因此相当缓慢。这是因为每次迭代都需要计算一个目标函数，其时间要求与基础语料库的大小成正比。

第二个关键的缺点是，*k*-medoid算法对文本等稀疏数据的效果并不好。这是因为很大一部分文档对没有很多共同的词，而且这些文档对之间的相似性是很小的（和嘈杂的）值。因此，一个单一的文档medoid往往不包含所有的概念，以便有效地围绕它建立一个聚类。这个特点是信息检索领域所特有的，因为基础文本数据的稀疏性。

- ***k-means*聚类算法。***k-means*聚类算法也使用一组*k*代表，围绕这些代表建立聚类。然而，这些代表不一定是从原始数据中获取的，其精炼方式与*k-medoids*方法有些不同。*k-means方法*的最简单形式是以一组来自原始语料库的*k*个种子开始，并根据最接近的相似度 将文档分配给这些种子。在下一次迭代中，分配给每个种子的中心点被用来替换上一次迭代中的种子。换句话说，新的种子被定义，以便它是这个集群的一个更好的中心点。这种方法一直持续到收敛为止。与*k*-medoids方法相比，*k-means方法*的一个优点是它只需要极小数量的

迭代的次数，以达到收敛的目的。来自[25, 83]的观察似乎表明，对于许多大型数据集，使用5次或更少的迭代就足以实现有效的聚类。*k*-means方法的主要缺点是，它对聚类过程中挑选的初始种子集仍然相当敏感。其次，一个给定的文档簇的中心点可能包含大量的词。这将减慢下一次迭代中的相似性计算。有一些方法被用来减少这些影响，这些方法将在本章的后面讨论。

种子的初始选择会影响*k-means*聚类的质量，特别是在文档聚类的情况下。因此，为了提高聚类过程中选择的初始种子的质量，我们使用了一些技术。例如，另一种轻量级的聚类方法，如聚类技术，可以用来决定种子的初始集合。这是在[25]中讨论的有效文档聚类方法的核心。我们将在下一小节中详细讨论这个方法。

改进初始种子集的第二个方法是在初始种子创建过程中使用某种形式的部分监督。这种形式的部分监督也可以帮助创建针对特定应用标准的集群。这种方法的一个例子是在[4]中讨论的，在这个例子中，我们把初始种子集作为从*Y ahoo*！分类法的某个特定类别中抓取的文档的中心点。 这也有一个效果，那就是最终的集群是按照不同的*Y ahoo*！类别中内容的一致性来分组的。这种方法已被证明在一些应用中是相当有效的，如文本归类。这种半监督技术对于信息组织特别有用，在这种情况下，开始的类别集有些嘈杂，但包含足够的信息，以创建满足预先定义的组织类型的集群。

3.3 一种混合方法。散点收集法

虽然分层聚类方法由于倾向于比较所有的文档对，所以往往更加稳健，但它们通常不是很有效，因为它们倾向于至少需要$O(n^2)$时间。另一方面，*K*-means类型的算法比分层算法更有效率，但有时可能不是很有效，因为它们倾向于依赖少量的种子。

[25]中的方法同时使用了分层和部分聚类算法，效果不错。具体来说，它在语料库的一个样本上使用层次聚类算法，以找到一个稳健的初始种子集。这个健壮的种子集与标准的*k-means*聚类算法结合使用，以确定良好的聚类。初始阶段的样本大小是经过精心设计的，以便提供尽可能好的效果，而不使这个阶段成为算法执行的瓶颈。

有两种可能的方法来创建初始种子集，它们分别被称为*buckshot*和*fractionation*。这是两种可供选择的方法，描述如下。

- **Buckshot。**设k是要找到的集群的数量，n是语料库中的文档数量。而不是选取从集合中随机选取k个种子。高估了$k - n$的种子，然后将这些种子聚集在一起。到k个种子。标准的聚类分层聚类算法rith√ms(需要四倍的时间)被应用于这个初始样本的$k - n$个种子。由于我们使用的是四级可扩展的算法，在这个阶段，这个方法需要$O(k\ n)$时间。- 我们注意到，这个种子集比简单地对k个种子进行采样的种子集要稳健得多，因为把一个大的文档样本总结成一个稳健的k个种子集。

- **分割。**分馏算法最初将语料库分成n/m个大小为$m>k$的桶。一个聚类算法被应用于这些桶中的每一个，以减少它们的系数v。因此，在这个阶段结束时，我们总共有$v\ n$个聚集点。这个过程是通过将这些聚集点中的每一个作为一个单独的记录来重复的。这是通过将一个聚集点内的不同文件合并成一个单一的文件来实现的。 当总共剩下k个种子时，该方法就终止了。我们注意到，在分化算法的第一次迭代中，每组m个文档的聚类需要$O(m^2)$的时间，在n/m个不同的组中，其总和为$O(n\ m)$。由于在每次迭代中，个体的数量以几何级数减少v，所有迭代的总运行时间为$O(n\ m\ (1 + \mu + v^2 + \ldots))$。对于常数$v<1$，所有迭代的运行时间仍然是$O$（$n\ m$）。通过选择$m=O(k)$，我们仍然可以确保初始化程序的运行时间为$O(n\ k)$。

*Buckshot*和*Fractionation*程序需要$O(k\ n)$时间，这也相当于k means算法的一个迭代的运行时间。

K-means算法的每一次迭代也需要$O(k\ n)$时间，因为我们需要计算n个文档与k个不同种子的相似度。

我们进一步注意到，分化程序可以应用于将文件随机分组为n/m个不同的桶。当然，我们也可以用一个更精心设计的程序来代替随机分组的方法，以获得更有效的结果。一个这样的程序是通过文件中*第j个*最常见的词的索引来对文件进行排序。这里*j*被选为一个小数字，如3，它对应于数据中的中等频率的词。然后，根据这个排序顺序，通过分割出连续的m个文档组，将文档划分为不同的组。这种方法确保了所创建的组中至少有几个常见的词，因此不是完全随机的。这有时可以提供更好的中心质量，这些中心是由分化算法决定的。

一旦使用*Buckshot*或*Fractionation*算法确定了初始聚类中心，我们就可以应用标准的*k-means*分区算法。具体来说，我们将每个文档分配到最接近的k个聚类中心。每个这样的聚类的中心点被确定为一个聚类中不同文档的串联。这些中心点取代上一次迭代中的种子集。这个过程可以用迭代的方法重复进行，以不断完善聚类的中心。通常情况下，只需要较少的迭代次数，因为最大的改进只发生在前几次迭代中。

也可以使用一些程序来进一步提高基础集群的质量。这些程序如下。

- **分割操作。**分割的过程可以用来进一步将聚类细化为颗粒度更好的组。这可以通过使用k=2对聚类中的单 个 文件应用buckshot程序来实现，然后围绕这些中心重新聚类。对于一个包含n个$_i$ 数据点的聚类，这整个过程需要$O(k - n_i)$时间，因此分割所有组需要$O(k\ n)$时间。然而，没有必要分割*所有*的组。相反，只有一个组的子集可以被分割。这些是不太连贯的组，包含了性质不同的文件。为了衡量一个组的一致性，我们计算一个群组的自相似度。这种自相似性为我们提供了对不相干性的理解。这个数量可以从集群中的文件与中心点的相似度或从集群中的文件与中心点的相似度两个方面来进行计算。

以集群文件之间的相似度为标准。然后，分割标准可以有选择地只应用于那些自我相似度低的群组。这有助于创建更加连贯的聚类。

- **联接操作。**连接操作试图将相似的聚类合并成一个单一的聚类。为了进行合并，我们通过检查中心点的最频繁的词来计算每个集群的*主题词*。如果两个聚类的主题词之间有明显的重叠，则认为这两个聚类是相似的。

我们注意到，该方法经常被称为*Scatter-Gather*聚类方法，但这更多是因为该聚类方法在原始论文[25]中被介绍为用于浏览大型集合的方式。Scatter-Gather方法可以用于有组织地浏览大型文档集，因为它创建了一个类似文档的自然层次。 特别是，用户可能希望以互动的方式浏览集群的层次结构，以了解集合中不同粒度级别的主题。一种可能性是事先进行分层聚类；但是这种方法的缺点是，当用户可能需要的时候，它不能即时合并和重新聚类树形层次的相关分支。 在[26]中提出了一种使用散点收集方法的恒定交互时间浏览方法。这种方法向用户展示了与不同的关键字相关的关键词。用户可以从这些关键词中挑选一个或多个，这也对应着一个或多个集群。这些聚类中的文件被合并并重新聚类到一个更细的颗粒度上。这个更精细的聚类将呈现给用户，供其进一步探索。被用户挑选出来进行探索的文件集被称为*焦点集*。接下来，我们将解释这个焦点集是如何在不断的时间内进一步探索和重新聚类的。

实现这一方法的关键假设是*集群细化假说*。这个假设指出，在一个明显的细粒度分区中属于同一集群的文件也会在一个更粗粒度的分区中一起出现。第一步是在聚类中创建一个文件的层次结构。各种聚类算法，如buckshot方法，都可以用于这个目的。我们注意到，这个树的每个（内部）节点可以被看作是一个元文件，对应于这个子树的叶子中所有文件的连接。集群精简的综合方法使我们能够使用较小的元文件集，而不是用一个小的元文件集来工作。

而不是某个子树上的全部文档。 我们的想法是选择一个常数*M*，它代表我们愿意使用交互式方法重新聚类的最大数量的元文献。然后，焦点集合中的树节点被扩展（优先考虑具有最大程度的分支），最多为*M*个节点。然后用散点收集的方法对这些*M*个节点进行重新聚类。这需要恒定的时间，因为在聚类过程中要使用恒定数量的*M*个元文件。因此，通过对*M*的元文件的工作，我们假定了下层子树的所有节点的聚类精简假设。显然，一个较大的*M*值并不能很好地假设聚类-精化假设，但也要付出较高的代价。该算法的细节在[26]中描述。这种方法的一些扩展也在文献[85]中预发，其中已经显示了这种方法如何被用来在恒定时间内对任意的语料库子集进行聚类。另一个最近的在线聚类算法称为*LAIR2*[55]，为Scatter/Gather浏览提供了恒定的交互时间。该算法的并行化明显快于Buckshot算法的相应版本。还有人认为，*LAIR2*算法在数据中导致了更高质量的聚类。

3.3.1 高效文档聚类的预测。

散点收集算法的挑战之一是，尽管该算法被设计成能很好地平衡聚类和分割阶段的运行时间，但由于一个给定的聚类中心点可能包含大量的不同术语，它有时会在大的文档集合中遭受减速。回顾一下，散点收集算法中的聚类中心点被定义为该集合中所有文档的连接。当集群中的文件数量很大时，这也会导致中心点中有大量的不同术语。这也将导致一些关键的计算速度减慢，如文档和集群中心点之间的相似度计算。

[83]对这个问题提出了一个有趣的解决方案。这个想法是使用*投影*的概念来降低文档表示的维度。这种维度的减少将导致显著的速度提升，因为相似性的计算将变得更加有效。[83]中的工作提出了三种投影。

- **全局投影。**在全局投影中，原始数据集的维度被降低，以便从数据中去除最不重要的（加权的）术语。一个术语的权重是

定义为文件中术语的（归一化和阻尼）频率的总量。

- **局部投影。**在局部投影中，每个聚类中的文档的维度都是以该聚类的*局部特定方法来*降低的。因此，每个聚类中心点的术语被分别截断。具体来说，不同聚类中心点中权重最小的术语被删除。因此，从每个文件中删除的术语可能是不同的，这取决于它们的本地重要性。

- **潜在语义索引。**在这种情况下，文档空间被LSI技术转换，聚类被应用于转换后的文档空间。我们注意到，LSI技术也可以在全局范围内应用于整个文档集合，或者根据需要在本地应用于每个聚类。

在[83]中已经表明，投影方法在有效性方面提供了相近的结果，同时相对于所有竞争的方法来说，保留了极高的效率水平。在这个意义上，聚类方法与相似性搜索不同，因为在进行投影时，它们在质量上几乎没有下降。其中一个原因是，与相似性搜索相比，聚类是一个不那么精细的应用，因此，即使我们在截断的特征空间中工作，也没有可察觉的质量差异。

4. 基于词和短语的聚类

由于文本文档来自于一个固有的高维领域，因此以一种双重方式来看待这个问题是很有用的，在这个问题上，可以找到重要的词簇，并利用它们来寻找文档的簇。在一个包含d个术语和n个文档的语料库中，我们可以把术语-文档矩阵看作是一个n d矩阵，其中（i，j）项是第j个术语在第i 个 文档中的频率。我们注意到，这个矩阵是非常稀疏的，因为一个给定的文档只包含宇宙中极小的一部分词汇。我们注意到，对这个矩阵中的*行*进行聚类的问题是对文档进行聚类，而对这个矩阵中的*列*进行聚类的问题是对词进行聚类。实际上，这两个问题是密切相关的，因为好的词簇可以被利用来寻找好的文档簇，反之亦然。例如，[16]中的工作确定了文档集合中的频繁词项集，并使用它们来确定紧凑的文档集群。这在某种程度上类似于

到使用词的集群[87]来确定文档的集群。最普遍的同时对词和文档进行 聚类的技术被称为*共同聚类*[30, 31]。这种方法同时对词-文档矩阵的行和列进行聚类，以创建这种聚类。这也可以被认为等同于对术语-文档矩阵的行和列进行重新排序的问题，以便在这个矩阵中创建密集的非零条目的矩形块。在某些情况下，为了确定好的集群，可以使用词之间的排序信息。[103]中的工作确定了集合中的频繁短语，并利用它们来确定文档集群。

重要的是要明白，词簇和文档簇的问题本质上是双重问题，它们彼此之间有密切的关系。前者与降维有关，而后者则与传统的聚类有关。这两个问题之间的界限是相当不稳定的，因为好的词簇为寻找好的文档簇提供提示，反之亦然。例如，一个更普遍的概率框架，它同时确定了词簇和文档簇，被称为*主题建模*[49]。主题建模是一个比聚类或降维更通用的框架。我们将在本章后面的章节中介绍主题建模的方法。 本书的下一章也有更详细的论述，这一章是关于降维的，第八章对文本挖掘的概率模型进行了更一般的讨论。

4.1 用频繁的词汇模式进行聚类

频繁模式挖掘[8]是一种在数据挖掘文献中被广泛使用的技术，以确定交易数据中最相关的特征。[16]中的聚类方法就是在这种频繁模式挖掘算法的基础上设计的。文本数据中的频繁项集也被称为*频繁术语集*，因为我们处理的是文档而不是交易。该方法的主要思想是不对高维的文档数据集进行聚类，而是将低维的频繁术语集作为聚类候选。 这实质上意味着一个频繁术语集是对一个集群的描述，它对应于所有包含该频繁术语集的文档。由于一个频繁术语集可以被视为一个集群的描述，所以一组精心选择的频繁术语集可以被视为一个集群。这个集合的适当选择

频繁术语集的定义是基于不同频繁术语集的支持文件之间的重叠。

在[16]中定义的聚类概念不一定使用严格的分区来定义文档的聚类，但它允许一定程度的重叠。这是许多基于术语和短语的聚类算法的一个自然属性，因为在算法的执行过程中，人们并不直接控制文档在聚类中的分配。允许集群之间有一定程度的重叠有时可能更合适，因为它认识到这样一个事实：文档是复杂的对象，不可能干净地将文档划分到特定的集群中，特别是当一些集群彼此之间有部分联系时。[16]的聚类定义假定每个文档至少被一个频繁术语集所覆盖。

让R是选定的频繁术语集的集合，它定义了聚类。让f_i是R中包含在第1个文档中的频繁术语集的数量。f_i的值至少为1，以确保完全覆盖，但我们希望它尽可能低，以尽量减少重叠。因此，我们希望在一个给定的集群中的文档的平均值（$f_i - 1$）低至
可能的。我们可以计算出(f_i 1)的平均值，为docu-
簇中的内容，并试图挑选频繁的术语集，使这个值尽可能的低。然而，这种方法往往有利于包含极少术语的频繁术语集。这是因为如果一个术语集包含m个术语，那么它的所有子集也会被文档所覆盖，结果是标准重叠度会增加。一个给定术语的熵重合度基本上是以下两项之和
的值$-(1/f_i)$-log($1/f_i$)在集群中的所有文件。当每个文件的$f_i = 1$时，该值为0，并单调地增加
随着f_i值的增加。

然后是描述如何从集合中选择频繁术语集。[16]中描述了两种算法，其中一种对应的是平面聚类，另一种对应的是层次聚类。我们将首先描述平面聚类的方法。显然，频繁术语的搜索空间是指数级的，因此一个合理的解决方案是利用一个贪婪算法来选择频繁术语集。在贪婪算法的每一次迭代中，我们选择与其他聚类候选者重叠度最小的频繁术语集。被选中的频繁术语所覆盖的文档将从数据库中删除，并在下一次迭代中计算与剩余文档的重叠度。

分层版本的算法与平面聚类中的大体思路相似，主要区别是每一级聚类的

被应用于包含固定数量k的术语集。换句话说，我们在选择过程中只处理长度为k*的*频繁模式。然后，通过对(k+1)术语集的应用，对所产生的聚类进行进一步划分。为了进一步划分一个给定的聚类，我们只使用那些包含定义该聚类的频繁K*术语*集的（k+1）-术语集。该方法的更多细节可以在[16]中找到。

4.2 利用Word集群进行文档集群

[87]中讨论了一个两阶段的聚类程序，它使用以下步骤来进行文档聚类。

- 在第一阶段，我们从文件中确定词组，这样，当我们用词组而不是词来表示文件时，词和文件之间的大部分相互信息都被保留下来。
- 在第二阶段，我们使用浓缩后的文档的词组表示，以进行最终的文档聚类。具体来说，我们用词组的出现来替换文档中的词的出现，以便进行文档聚类。 这个两阶段程序的一个优点是大大减少了表示中的噪音。

让$X = x_1 \ldots x_n$是对应于行（文件）的随机变量，让$Y = y_1 \ldots y_d$是对应于列（词）的随机变量。我们希望将X划分为k*个*集群，将Y划分为l*个*集群。让这些集群用$\hat{X} = \hat{x}_1 \ldots \hat{x}_k$和$\hat{Y} = \hat{y}_1 \ldots \hat{y}_l$。 换句话说，我们希望找到$C_X$和$C_Y$的映射。其中定义了聚类。

$$C_X : x_1 \ldots x_n \Rightarrow \hat{x}_1 \ldots \hat{x}_k$$
$$C_Y : y_1 \ldots y_d \Rightarrow \hat{y}_1 \ldots \hat{y}_l$$

在程序的第一阶段，我们将Y分组为$\hat{Y}$，因此，大多数$I(X, Y)$中的信息在$I(X, \hat{Y})$中被保留。在第二种情况下阶段，我们再次从X到$\hat{X}$进行聚类。使用的正是同样的程序，以便从$I(X, \hat{Y})$中获得尽可能多的信息在$I(\hat{X}, \hat{Y})$中被保留下来。关于如何进行聚类的每个阶段的细节在[87]中提供。

如何发现有趣的词组（可用于文档聚类）本身就已经在自然语言中引起了关注。

语料处理研究界，对发现能够表征词义[34]或语义概念[21]的词簇特别感兴趣。例如，在[34]中，马尔科夫聚类算法被应用于以无监督的方式发现语料库的特定词义。具体来说，首先构建一个词的关联图，其中相关的词会用一条边连接。对于一个可能具有多种意义的词，我们可以分离出代表其邻居的子图。这些邻居有望根据目标词的不同意义形成集群，因此，通过将相互之间有良好联系的邻居组合在一起，我们可以发现描述目标词不同意义的词集群。在[21]中，提出了一个n-gram类语言模型，基于最小化相邻词之间的相互信息损失来对词进行聚类，可以达到将自然语言文本中具有相似语境的词聚在一起的效果。

4.3 词语和文件的联合聚类

在许多情况下，最好是同时对或然表的行和列进行聚类，并在聚类过程中探索词簇和文档簇之间的相互作用。由于词和文档之间的聚类明显相关，当希望沿着两个维度中的一个找到聚类时，往往希望同时对两者进行聚类。这种方法被称为*共聚类*[30, 31]。共聚类被定义为一对从行到行聚类指数和从列到列聚类指数的地图。这些地图是由算法*同时*确定的，以优化相应的聚类表示。

我们进一步注意到，本章前面讨论的矩阵分解方法[58]可以自然地用于协同聚类，因为它同时发现了词簇和文档簇。在那一节中，我们也讨论了如何将矩阵因数分解看作是一种共聚类技术。虽然矩阵分解没有被广泛地用作共聚类的技术，但我们指出了这种自然的联系，作为未来与其他共聚类方法比较的可能探索。最近的一些工作[60]显示了在文档聚类技术的背景下，如何使用矩阵分解来将知识从单词空间转化为文档空间。

共同聚类的问题也与数据库文献中定量数据中的*子空间聚类*[7]或*投影聚类*[5]问题密切相关。在这个问题中，数据的聚类是通过同时将其与一组点和子空间在多

维空间。协同聚类的概念是这一广泛思想在数据领域的自然应用，这些数据可以表示为**稀疏的**高维矩阵，其中大部分条目为0。因此，传统的子空间聚类方法也可以扩展到协同聚类问题。例如，[59]中提出了一种自适应迭代的文档亚空间聚类方法。

我们注意到，子空间聚类或共同聚类可以被认为是一种*局部特征选择的*形式，其中选择的特征是针对每个聚类的。一个自然的问题是，是否可以像传统的降维技术（如PCA[53]）那样，将特征作为维度的线性组合来选择。这在传统的数据库文献中也被称为*局部降维*[22]或*广义投影聚类*[6]。在这种方法中，基于PCA的技术被用来生成*针对每个聚类的*子空间代表，并被利用来实现更好的聚类过程。特别是，最近设计了这样一种方法[32]，它已被证明对文档数据有很好的效果。

在这一节中，我们将研究两种众所周知的文档共同聚类方法，它们在文档聚类的文献中被普遍使用。其中一个方法使用基于图的术语-文档代表[30]，另一个方法使用信息理论[31]。我们将在下面讨论这两种方法。

4.3.1 带图分区的协同聚类。 这种方法的核心思想[30]是将术语-文档矩阵表示为一个双位图$G = (V_1 \ V_2, E)$，其中V_1和V_2代表这个图的两个双位部分的顶点集，E代表边集。V_1中的每个节点对应于n个文档中的一个，V_2中的每个节点对应于d个术语中的一个。如果文档i包含术语j，则节点i V_1和节点j V_2之间存在一条无方向的边。我们注意到，E中没有直接在术语之间或直接在文档之间的边。因此，该图是二方的。每条边的权重是相应的归一化术语频率。

我们注意到，在这个二方图中，一个词的分区会引起一个文档的分区，反之亦然。给定该图中的文档分区，我们可以将每个词与它所连接的边的权重最大的文档集群联系起来。请注意，这个标准也使各分区的边的权重最小。同样地，给定一个词的分区，我们可以将每个文档与它以最大的边的权重连接的词的分区联系起来。因此，这个问题的一个自然解决方案是

*同时*对该图进行*k-way*分区，使各分区的边的总权重最小。这当然是图划分文献中的一个经典问题。 在[30]中，已经显示了如何有效地使用光谱分区算法来实现这一目的。在[75]中讨论的另一种方法是使用等距双点图分区的方法进行聚类。

4.3.2 在[31]中，最优聚类被定义为使被聚类的随机变量之间的实际信息最大化的**聚类** 。正态化的非负或然率表被视为两个离散随机变量之间的联合概率分布，这些随机变量在行和列上取值。让$X = x_1 \ldots x_n$是对应于行的随机变量，让$Y = y_1 \ldots y_d$是对应于列的随机变量。我们希望将X划分为k个群组，将Y划分为l个群组。让我们用$\hat{X} = \hat{x}_1 \ldots \hat{x}_k$和$\hat{Y} = \hat{y}_1 \ldots \hat{y}_l$来表示这些群组。换句话说，我们希望找到定义聚类的地图C_X和C_Y。

$$C_X : x_1 \ldots x_n \Rightarrow \hat{x}_1 \ldots \hat{x}_k$$
$$C_Y : y_1 \ldots y_d \Rightarrow \hat{y}_1 \ldots \hat{y}_l$$

分区函数C_X和C_Y被允许依赖于联合概率分布$p(X, Y)$。我们注意到，由于$\hat{X}$和$\hat{Y}$是X和Y的更高层次的聚类，所以在更高层次的表示中存在相互信息的损失。换句话说，分布$p(\hat{X}, \hat{Y})$含有比$p(X, Y)$的信息量少，而且相互信息$I(\hat{X}, \hat{Y})$比相互信息$I(X, Y)$低。因此，最佳共聚类问题是确定使相互信息损失最小的映射。换句话说，我们希望找到一个$I(X, Y) - I(\hat{X}, \hat{Y})$尽可能小的共同聚类。[29]中提出了一种迭代算法，用于寻找相互信息损失最小的共同聚类。

4.4 用频繁的短语进行聚类

这种方法与其他文本聚类方法的主要区别之一是，它把一个文档当作一个字符串，而不是一个词包。具体来说，每个文档都被当作一个*词*的字符串，而不是字符。字符串表示法和词包表示法的主要区别在于，前者还为聚类过程保留了口令信息。正如许多

聚类方法，它使用一种索引方法来组织文档集合中的短语，然后使用这种组织来创建聚类[103, 104]。为了创建聚类，有几个步骤被使用。

(1) 第一步是对代表文件的字符串进行清理。通过删除单词的前缀和后缀以及将复数减为单数，使用了一种轻量级的词根算法。句子的边界被标记，非词的标记被剥离。

(2) 第二步是识别基础集群。这些是由集合中的频繁阶段定义的，*以后缀树的*形式表示。后缀树[45]本质上是一个三角形，包含了整个集合的所有后缀。后缀树的每个节点代表一组文档，以及所有这些文档中共同的一个短语。由于后缀树的每个节点也对应于一组文档，所以它也对应于一个基础聚类。每个基础聚类都有一个分数，这个分数基本上是该聚类中的文档数量与基础短语长度的一个非递减函数的乘积。因此，包含大量文件的聚类，以及由相对较长的短语定义的聚类是比较理想的。

(3) 由后缀树创建的基础集群的一个重要特点是，它们没有定义一个严格的分区，而且彼此之间有重叠。例如，同一个文档可能在后缀树的不同部分包含多个短语，因此会被包含在相应的文档组中。该算法的第三步是根据其基础文档集的相似性来合并集群。让P和Q是对应于两个聚类的文档集。基础相似度$BS(P, Q)$定义如下。

$$BS(P, Q) = \frac{|P \cap Q|}{\text{最大}\{|P|, |Q|\}} + 0.5 \qquad (4.11)$$

这个基础相似度要么是0，要么是1，取决于这两个组是否有至少50%的共同文件。然后，我们构建一个图结构，其中的节点代表基础集群，如果这对群组之间相应的基础相似度为1，则两个集群节点之间存在一条边。具体来说，每个连接组件中的文件组的联盟被用作最终的集群集。我们注意到，最终的聚类集彼此之间的重叠程度要小得多，但它们仍然没有定义一个严格的分区。这有时是聚类算法的情况，其中允许适度的重叠，以使聚类质量更好。

5. 概率文档聚类和主题模型

一个流行的概率性文档聚类方法是*主题建模法*。主题建模的理念是为语料库中的文本文档创建一个*概率生成模型*。主要的方法是将语料库表示为一个隐藏的随机变量的函数，其参数是用一个特定的文档集来估计的。任何主题建模方法的主要假设（与相应的随机变量一起）如下。

- 语料库中的n个文档被假定为有可能属于k个主题中的一个。因此，一个给定的文件可能有属于多个主题的概率，这反映了一个事实，即同一个文件可能包含众多的主题。对于一个给定的文档D_i，和一组主题$T_1 \ldots T_k$，文档D_i属于主题T_j的概率是由$P(T_j|D_i)$给出。我们注意到，主题本质上类似于聚类，而$P(T_j D_i)$的值提供了第i个文档属于*第*j个聚类的概率。在非概率聚类方法中，文档在聚类中的成员资格在本质上是确定的，因此，聚类通常是对文档集合的干净划分。然而，当多个聚类中的文档主题存在重叠时，这往往会带来挑战。使用*概率方面的软集群成员资格*是对这一困境的一个优雅的解决方案。在这种情况下，确定文档在集群中的成员资格是一个次要的目标，而不是在基础文本集合中寻找*潜在的主题集群*。因此，这一领域的研究被称为*主题建模*，虽然它与聚类问题有关，但它通常被作为一个与聚类不同的研究领域来研究。

  $P(T_j D_i)$的值是用主题建模的方法估计的，是算法的主要输出之一。k的值是算法的输入之一，类似于聚类的数量。

- 每个主题都与一个概率向量相关联，该概率向量量化了该主题在词库中的不同术语的概率。让$t_1 \ldots t_d$是词库中的d个术语。然后，对于完全属于主题T_j的文档，术语t_l出现在其中的概率由$P(t_l|T_j)$给出。$P(t_l|T_j)$的值是另一个

重要的参数，需要通过主题建模的方法来估计。

注意，文档的数量用n表示，主题用k表示，词库的大小（术语）用d表示。大多数主题建模方法试图使用最大似然法学习上述参数，以便与给定的文档语料库的概率拟合尽可能大。有两种基本的方法被用于主题建模，分别是*概率潜在语义索引（PLSI）*[49]和潜在*狄里奇分配（LDA）*[20]。

在这一节中，我们将重点讨论概率性潜在语义in- dexing方法。 请注意，上述一组随机变量$P(T_j \mid D_i)$和$P(t_l \mid T_j)$允许我们对术语t_l出现在任何文档D_i的概率进行建模。具体而言，术语t_l出现在文档D_i的概率$P(t_l \mid D_i)$可以用上述参数表示如下。

$$P(T_l \mid D_i) = \sum_{j=1}^{k} P(t_l \mid T_j) - P(T_j \mid D_i) \tag{4.12}$$

因此，对于每个术语t_l和文档D_i，我们可以根据这些参数生成一个$n \times d$的概率矩阵，其中n是文档的数量，d是术语的数量。对于一个给定的语料库，我们也有$n \times d$术语-文档发生矩阵X，它告诉我们哪个术语*实际*出现在每个文档中，以及该术语在文档中出现了多少次。换句话说，$X(i, l)$是术语t_l在文档D_i中出现的次数。因此，我们可以使用最大似然估计算法，使整个语料库中每个文档中观察到的术语的概率乘积最大化。其对数可以表示为方程4.12中术语的对数的加权和，其中第（i，l）个术语的权重是其频率计数X（i，l）。这是一个受限的优化问题，它优化了对数似然概率$_{i,l}$的值$X(i, l)\log(P(t_l \mid D_i))$，但受制于每个主题-文档和术语-主题空间的概率值必须和为1。

$$\sum_{l} P(t_l \mid T_j) = 1 \quad \forall T_j \tag{4.13}$$

$$\sum_{j} P(T_j \mid D_i) = 1 \quad \forall D_i \tag{4.14}$$

目标函数中的$P(t_i | D_i)$的值被展开，并使用公式4.12以模型参数表示。我们注意到，可以使用拉格朗日方法来解决这个受限问题。这与我们在本章讨论的非负矩阵分解问题的方法很相似。拉格朗日法的解决方法本质上导致了一组需要估计的相应参数的迭代更新方程。可以证明，这些参数可以通过迭代更新方程来估计[49]。

日期的两个矩阵$[P]_{1k\times n}$和$[P]_{2d\times k}$，分别包含主题-文档概率和术语-主题概率。我们首先要做的是

随机地初始化这些矩阵，并对其进行归一化，使其列中的概率值之和为1。然后，我们分别对P_1和P_2中的每一个迭代执行以下步骤。

对于P_1中的每个条目(j, i)进行更新

$$P_1(j,i) \leftarrow P_1(j,i) - \sum_{r=1}^{d} P_2(r,j) \frac{X(i,r)}{\sum_{v=1}^{k} P_1(v,i) \cdot P_2(r,v)}$$

将P_1的每一列，使之归一为1。

对于P_2中的每个条目(l, j)进行更新

$$P_2(l,j) \leftarrow P_2(l,j) - \sum_{q=1}^{n} P_1(j,q) \frac{X(q,l)}{\sum_{v=1}^{k} P_1(v,q) \cdot P_2(l,v)}$$

将P_2的每一列，使之归一为1。

这个过程反复进行，直到收敛。这种方法的输出是两个矩阵P_1和P_2，其条目分别提供主题-文档和术语-主题的概率。

第二个著名的主题建模方法是*Latent Dirichlet Allocation*。在这种方法中，术语-主题概率和主题-文档概率是以迪里切特分布作为先验来建模的。因此，LDA方法是PLSI技术的贝叶斯版本。也可以证明PLSI方法与LDA技术是等价的，当应用统一的Dirichlet先验时[42]。

LDA的方法是在[20]中首次提出的。随后，与PLSI方法相比，它通常被更广泛地使用。与PLSI方法相比，它的主要优势在于它不太容易出现过拟合的情况。这通常是贝叶斯方法的真实情况，它减少了需要估计的模型参数的数量，因此对较小的数据集来说效果更好。即使对于较大的数据集，PLSI也有一个缺点，即模型参数的数量会随着数据集的大小而线性增长。有人认为[20]，PLSI模型并不是一个完全的生成模型，因为没有办法对不包括在当前数据集中的文档的主题分布进行建模。例如，我们可以使用当前的数据集

在对新文件进行建模时，使用主题分布，但由于PLSI中固有的过度拟合，它可能会更不准确。贝叶斯模型，使用少量的参数，以精心选择的先验分布的形式，如*Dirichlet*，在对新文档进行建模时可能会更加稳健。因此，LDA方法也可以用来更稳健地模拟新文档的主题分布，即使它不存在于原始数据集中。尽管LDA比PLSA有理论上的优势，但最近的一项研究表明，它们在聚类、分类和检索方面的任务表现趋于相似[63]。主题模型的领域相当广泛，本书第五章和第八章将对其进行更深入的处理；本节的目的只是让读者了解这一领域的基本情况以及它与聚类的自然联系。

我们注意到，用于主题建模的EM概念是非常普遍的，可以用于文本聚类任务的不同变化，如文本分类[72]或将用户反馈纳入聚类[46]。例如，[72]中的工作使用了一个EM方法，以便在有标记和无标记数据的混合情况下，对文档进行监督聚类（和分类）。第6章关于文本分类有更详细的讨论。

6. 用文本流进行在线聚类

在文本数据的背景下，流式文本聚类的问题尤其具有挑战性，因为聚类需要不断地实时维护。最早的流式文本聚类方法之一是在[112]中提出的。这种技术被称为*在线球形k-Means算法（OSKM）*，这反映了该方法所使用的广泛方法。这种技术将传入的数据流分成小段，每段都可以在主内存中有效处理。一组*k*-means迭代被应用于每个这样的数据段，以便对它们进行聚类。使用分段法进行聚类的好处是，由于每个分段都可以保存在主内存中，只要每个数据点保存在主内存中，我们就可以多次处理。此外，前一个区段的中心点会在下一次迭代中用于聚类目的。为了使旧文件老化，我们引入了一个衰减因子，以便从聚类的角度来看，新文件被认为更重要。在[112]中，这种方法已经被证明在对大量文本流进行聚类时非常有效。

在[3]中讨论了一种不同的对大量文本和分类数据流进行聚类的方法。在[3]中讨论的方法使用了一种方法

检验基础数据中的异常值、新趋势和集群之间的关系。旧的聚类可能变得不活跃，最终被新的聚类所取代。同样，当新来的数据点不自然地适合于任何特定的聚类时，这些数据最初需要被归类为离群值。然而，随着时间的推移，这些新点可能会创造出一种独特的活动模式，可以被识别为一个新的集群。数据流的时间定位是由这些新的聚类来体现的。例如，在抓取的过程中，属于某个特定类别的第一个网页可能会被识别为一个离群点，但后来可能会形成一个属于自己的文档集群。另一方面，新的异常值不一定会导致新集群的形成。这种异常值是数据中真正的短期异常，因为它们不会导致可持续模式的出现。在[3]中讨论的方法是通过首先将其识别为离群值来识别新的聚类。这种方法通过使用总结方法来工作，在这种方法中，我们使用*浓缩液滴的*概念[3]，以创建基础集群的简明表示。

与OSKM算法的情况一样，我们确保最近的数据点比老的数据点更受重视。这是通过为每个数据点创建一个对时间敏感的权重来实现的。假设每个数据点都有一个由函数$f(t)$定义的随时间变化的权重。该函数$f(t)$也被称为*消逝函数*。衰减函数$f(t)$是一个非单调的递减函数，随时间t均匀衰减。定义半衰期的目的是量化每个数据点在数据流聚类过程中的重要性的衰减率。*衰减率*被定义为半衰期的倒数。

数据流中每个点的权重函数为$f(t) = 2^{-\lambda \cdot t}$，数据流的寿命。我们用$\lambda = 1/t$表示衰减率。从聚类过程的角度来看，每个数据的权重为点是$f(t)$。很容易看出，这个衰减函数创造了一个$1/\lambda$的半衰期。同样明显的是，通过改变λ的值，可以改变数据流中历史信息的重要性衰减的速度。

当一个聚类在流媒体过程中由一个新到达的数据点创建时，它被允许作为一个趋势设定的离群点保持至少一个半衰期。在此期间，如果至少有一个更多的数据点到达，那么这个集群就会成为一个活跃和成熟的集群。另一方面，如果在半衰期内没有新的数据点到来，那么这个趋势设定异常点就被认为是数据流中的一个真正的异常点。在这一点上，这个异常点被从当前集群的列表中删除。我们把这个删除的过程称为*集群死亡*。因此，当一个包含一个数据点的新聚类的（加权）点的数量达到一定程度时，这个聚类就会死亡。

在集群中的数量为0.5。同样的标准被用来定义成熟集群的死亡。满足这个标准的一个必要条件是，集群中的不活动期已经超过了半衰期$1/\lambda$。集群中的点的数量越多，不活动期需要超过其半衰期以满足标准的水平就越大。这是一个自然的解决方案，因为从直觉上讲，对包含较多点数的集群的死亡有更强的要求（更长的不活动期）是可取的。

数据点的统计数据被捕捉到汇总统计中，这些数据被称为*凝结的液滴*。这些数据代表了一个集群内的单词分布，可以用来计算一个传入的数据点与集群的相似度。整个算法的过程如下。在算法执行之初，我们从一个空的聚类组开始。随着新数据点的到来，包含单个数据点的单元集群被创建。一旦创建了最多的k个这样的集群，我们就可以开始在线集群维护的过程。因此，我们最初从一个由k*个集群*组成的微不足道的集合开始。随着新数据点的到来，这些聚类会逐渐更新。

当一个新的数据点X到达时，它与每个簇滴的相似度被计算出来。在文本数据集的情况下，*DF1*和X之间的余弦相似度被使用。相似度值$S(X, C_j)$是从传入的文档X到每个聚类C_j计算出来的。$S(X, C_j)$值最大的聚类被选为数据插入的相关聚类。让我们假设这个集群是$_{mindex}$。我们使用一个阈值，用*thresh*表示，以确定传入的数据点是否是一个离群点。如果$S(X, C_{mindex})$的值大于阈值*thresh*，那么点X就被分配到$_{mindex}$集群中。否则，我们检查在当前的簇滴集合中是否存在一些不活跃的簇。 如果没有这样的非活动集群存在，那么数据点X就被添加到C_{mindex}中。另一方面，当一个不活跃的集群确实存在时，就会创建一个包含单独数据点X的新集群，这个新创建的集群将取代不活跃的集群。我们注意到，这个新的聚类是一个潜在的真正的离群点或一个新的数据点趋势的开始。只有随着数据流的进展，才能进一步了解这个新集群。

如果X被插入到群集$_{mindex}$中，我们更新群集的统计数据，以反映数据点的插入和时间上的衰减统计。否则，我们用一个包含单独数据点X的新簇来替换最不活跃的簇。特别是，被替换的簇是所有不活跃的簇中最近更新最少的簇。这个过程在下列情况下持续进行

在数据流的生命周期中，随着新文件的到来。[3]中的工作还介绍了流聚类技术的各种其他应用，如演变和相关分析。

[48]中描述了在聚类过程中利用文本文档的时间演变的不同方式。具体来说，该工作在
[48]使用*突发性特征*作为数据流中新话题出现的标记。这是因为一个新出现的话题的语义往往反映在文本流中频繁出现的几个独特的词上。在某一特定时期，相关话题的性质可能会导致数据流中特定特征的突发。显然，从聚类的角度来看，这种特征是非常重要的。因此，[48]中讨论的方法使用了一种新的表示方法，它被称为用于挖掘文本流的*突发性特征表示*。在这种表示法中，时间变化的权重与特征相关联，以待其爆发性。这也反映了该特征在聚类过程中的不同重要性。因此，重要的是要记住，一个特定的文件表示法是取决于它被构建的特定瞬间。

在这种方法中，另一个被有效处理的问题是对基础集合的维度的隐性降低。文本本身就是一个高维数据域，根据它们的突发性对一些特征进行预选是降低文档表示维度的一种自然方式。这可以帮助提高基础算法的有效性和效率。

该过程的第一步是识别数据流中的突发性特征。为了实现这一目标，该方法使用Klein- berg的2状态有限自动机模型[57]。一旦这些特征被识别出来，突发特征就会与取决于其突发程度的权重相关联。随后，为了反映特征的基本权重，定义了一个突发性特征的表示。爆发性特征的识别和权重都取决于其基本频率。一个标准的*K*-means方法被应用于新的表示，以构建聚类。在[48]中显示，使用突发性的方法提高了聚类的质量。对[48]中工作的批评是，它主要集中在利用数据流的节奏特征来提高有效性的问题上，而没有解决基础数据流的有效聚类问题。

一般来说，特征提取对所有聚类算法都很重要。虽然[48]的工作重点是利用流的时间特征进行特征提取，但[61]的工作重点是利用*短语提取*进行有效的特征选择。这项工作

也与主题建模的概念有关，这将在下一节详细讨论。这是因为一个集合中的不同主题可以与集合中的群组相关。[61]中的工作使用了话题建模技术进行聚类。[61]的工作的核心思想是，单个单词对于聚类算法来说不是很有效，因为它们错过了单词的使用环境。例如，"明星 "这个词既可能是指一个天体，也可能是指一个艺人。另一方面，当使用 "恒星 "这个短语时，"恒星 "这个词显然是指一个天体。从集合中提取的短语也被称为*主题签名*。

使用这种短语澄清来提高聚类的质量被称为*语义平滑*，因为它减少了与语义模糊性有关的噪音。因此，该方法的一个关键部分是从基础数据流中提取短语。短语提取后，训练过程确定了短语与词汇中的术语的翻译概率。例如，"行星 "这个词可能与 "恒星 "这个短语有很高的关联概率，因为两者都是指天体。因此，对于一个给定的文件，也可以给所有的术语分配一个合理的概率计数。对于每个文档，假定其中的所有术语都是由主题特征模型，或背景收集模型产生的。

[61]中的方法是通过对单词w和集群C_j的软概率$p(\mathrm{w}\,|\,C_j)$进行建模。概率$p(w\ C_j)$被建模为两个因素的线性组合；(a)最大似然模型，该模型对每个集群产生特定词的概率进行了计算；(b)间接（翻译）词成员概率，该概率首先确定了每个主题符号的最大似然概率，然后与每个词的条件概率相乘，给定主题符号。我们注意到，我们可以使用$p(w\,|\,C_j)$来估计$p(d\ C_j)$，通过使用文档中的组成词的乘积。为此，我们使用w在文档d中的频率f（w，d）。

$$p(d\,|\,C_j)=\prod_{w\in d} p(w\,|\,C_j)^{f(w,d)} \tag{4.15}$$

我们注意到，在静态情况下，α为了提高估计过程的稳健性，也可以增加一个背景模型。然而，这在数据流中是不可能的，因为背景收集模型可能需要多次传递才能有效建立。[61]中的工作通过使用*群集概况*以在线方式维护这些概率，该群集*概况*通过使 用 渐变函数对概率进行加权。我们注意到，集群的概念

概况类似于[3]中介绍的浓缩液滴的概念。关键的算法（用OCS表示）是保持一个动态的集群集合，利用相似性计算将文件逐步分配到其中。在[61]中已经显示了如何使用集群概况来有效地计算每个传入文档的$p(d\ C_j)$。这个值然后被用来确定文件与不同聚类的相似度。这被用来将文件分配到它们最接近的集群中。我们注意到，[3, 61]中的方法在以下方面有许多相似之处：(a) 维护聚类概况，(b) 使用聚类概况（或浓缩液滴）来计算相似度并将文档分配到最相似的聚类中，以及(c) 用来决定何时应该创建一个新的单子聚类，或者应该替换一个旧的聚类的规则。

两种算法的主要区别在于计算聚类相似度的技术。OCTS算法使用概率计算$p(d\ C_j)$来计算聚类相似度，在计算过程中考虑到了短语信息。关于OCTS的一个观察结果是，它可能允许非常相似的聚类在当前集合中共存。这就减少了可用于不同集群概况的空间。[61]中还提出了第二种叫做OCTSM的算法，它允许合并非常相似的聚类。在每次分配之前，它都会检查一对类似的聚类是否可以根据相似性进行合并。如果是这样，那么我们就允许合并相似的聚类和它们相应的聚类简介。关于不同聚类算法的详细实验结果和它们的有效性在[61]中介绍。

与聚类密切相关的领域是话题建模，我们在前面的章节中讨论过这个问题。最近，话题建模方法也被扩展到*动态*情况，这对文本流的话题建模很有帮助[107]。

7. 网络中的文本聚类

许多社交网络既包含节点中的文本内容，也包含不同节点之间的链接。显然，链接在理解网络中的相关节点方面提供了有用的线索。不同的链接类型对聚类质量的影响已经在[109]中进行了研究，结果表明，许多形式的隐性和显性链接可以提高聚类质量，因为它们编码了人类知识。因此，一个自然的选择是在对不同节点进行聚类的过程中结合这两个因素。在本节中，我们将讨论一些这样的技术。

一般来说，链接可以被认为是一种侧面信息，可以用属性的形式表示。一个将侧面属性纳入聚类过程的一般方法已经在[1]中提出。这种算法在文本属性上使用*K*-means方法，在聚类过程中使用贝叶斯概率估计的组合。这个想法是为了识别那些对聚类过程有帮助的属性，并利用它们来提高聚类的质量。然而，这种方法实际上是为任何种类的一般属性而设计的，而不是基于链接的属性，在这种情况下，文档与文档之间的链接隐含着一个基本的图结构。尽管如此，在[1]中已经表明，通过将链接信息作为副属性来处理，有可能大大增强聚类的质量。本节将讨论许多其他技术，这些技术是专门针对文本文档的情况提出的，这些文档以网络结构链接在一起。

最早提出结合文本和链接信息进行聚类的方法是在[12]中。本文为聚类过程提出了两种不同的方法。 第一种方法是利用一个节点的邻居的链接信息，以偏重于文档中的术语权重。在聚类过程中，一个文档和它的邻居之间共同的术语权重被赋予更多的重要性。这种方法的一个优点是，我们可以使用任何一种现有的聚类算法来达到这个目的，因为链接信息是隐含在修改后的术语权重中的编码。[12]中提出的第二种方法是一种基于图的方法，在聚类过程中直接使用链接。在这种情况下，该方法试图对一个特定的链接和内容集的特定文件属于一个特定聚类的概率进行建模。这本质上是一种*软聚类*，为每个聚类确定一个分配概率。具有最大分配概率的集群被认为是最相关的集群。为了进行聚类，采用了马尔科夫随机场（MRF）技术。为了计算这个MRF的最大似然参数，使用了一种叫做放松标记的迭代技术。这个方法的更多细节可以在[12]中找到。

[113]中提出了一个最新的方法，以结构和at-tribute相似性来进行聚类。本文的技术可以应用于关系属性和文本属性。本文通过在原始结构顶点之外向网络添加属性顶点，整合了结构和基于属性的聚类。在文本数据的背景下，这意味着每个词都有一个顶点存在于词库中。

con。因此，除了图$G=(V, E)$中的原始顶点集V之外，我们现在还有一个增强的顶点集$V \cup V_1$，这样V_1，每个节点包含一个顶点。我们还对边缘集进行了增强，以便在原来的结构性边缘集E中添加。我们在结构性顶点$i \in V$和属性顶点$j \in V_1$之间添加一条边缘，如果词j包含在节点i中。这个新增加的边的集合表示为E_1。因此，我们现在有一个增强的图$G_1 = (V \cup V_1, E \cup E_1)$，它是半双边的。为了确定顶点的接近程度，我们使用了一个邻域随机行走模型。这种亲近度的测量被用来进行聚类。该算法的主要挑战是在聚类过程中确定结构和属性成分的相对重要性。在随机行走模型的背景下，这意味着在随机行走过程中确定不同边缘的适当权重。[113]中提出了一个学习模型，以学习这些权重，并利用它们进行有效的聚类过程。

在社会网络的社区检测中，经常会遇到网络内容聚类的问题。社交网络图中的文本内容可以附着在网络的节点上[101]，也可以附着在边上[74]。基于节点的方法一般比较常见，本文中提到的大多数技术都可以用附着在节点上的内容来建模。在[101]提出的方法中，为了进行有效的社区检测，结合了以下基于链接和基于内容的步骤。

- 为链接分析提出了一个条件模型，其中给定链接的目的地的条件概率被建模。为了捕捉一个节点被其他节点引用的可能性，引入了一个隐藏变量，以捕捉该节点的受欢迎程度。

- 为了减少嘈杂的内容属性的影响，引入了一个判别性的内容模型。在这个模型中，属性的权重是根据它们区分不同社区的能力来决定的。

- 通过使用最大似然推断的两阶段优化算法，这两个模型被组合成一个统一的框架。这个广泛的框架的一个有趣的特点是，它也可以在其他补充性方法的背景下使用。

该算法的细节在[101]中讨论。

对于基于边缘的社区检测的情况，假定网络中的文本内容是附着在边缘上的[74]。这在涉及不同节点之间广泛交流的应用中很常见。例如，在电子邮件网络或在线聊天网络中，网络中的文本与不同实体之间的通信有关。在这种情况下，文本与底层网络中的一个边缘相关联。与边相关的内容的存在允许在社区检测中采取更细微的方法，因为一个给定的节点可能参与不同类型的社区。与边相关的内容的存在有助于分离出同一个人与不同社区的不同联系。[74]中的工作使用了一种矩阵因子化方法，以便为社区检测过程中的内容和结构共同建模。矩阵因子化方法被用来将表征转化为多维表征，这可以很容易地通过简单的算法（如*k*-means算法）进行聚类。在[74]中显示，使用这种方法可以提供比纯粹的基于内容或链接的聚类方法有效得多的结果。

与聚类密切相关的一个领域是*主题建模*，在这个领域中，我们试图对一个文档属于某个特定聚类的概率进行建模。基于网络的主题建模的一个自然方法是为传统的主题模型（如NetPLSA[65]）添加一个基于网络的正则化约束。在[23]中提出的关系主题模型（RTM）试图按顺序对文档和链接的生成进行建模。生成文档的第一步与LDA相同。其次，该模型根据两个文档中使用的主题混合物的相似性来预测链接。因此，这种方法既可用于主题建模，也可用于预测缺失链接。*iTopicModel*[91]框架中提出了一个更统一的模型，它创建了一个马尔科夫随机场模型，以创建一个生成模型，同时捕捉文本和链接。实验结果表明，这种方法更加普遍，并且优于以前的方法。下一章将讨论将网络信息纳入主题建模的其他一些方法，即降维。

8. 半监督聚类法

在一些应用中，关于基础数据中可用的集群种类的先验知识可能是可用的。这种先验知识可以采取与文件相连的标签形式。

这表明它的基本主题。例如，如果我们希望利用*Yahoo!*分类法中广泛的主题分布来监督一个新的网络集合的聚类过程，一种执行监督的方法是将*Yahoo!*分类法中一些有标签的页面添加到集合中。一般来说，这些页面会包含@*Science*@Astronomy或@*Arts*@Painting这样的标签，表明所添加页面的主题领域。这样的知识对于创建明显更连贯的集群是非常有用的，尤其是当集群的总数很大时。使用这种标签来指导聚类过程的过程被称为*半监督聚类*。这种学习形式是聚类和分类问题之间的桥梁，因为它使用底层的类结构，但它并不完全被具体的结构所束缚。因此，这种方法既适用于聚类，也适用于分类的情况。

将监督纳入聚类过程的最自然的方法是在部分条件聚类方法中这样做，如*K*-手段。这是因为监督可以很容易地通过改变聚类过程中的种子而被纳入。例如，[4]中的工作将*k*-means聚类过程中的初始种子作为基础数据中原始类的中心点。在[15]中也使用了类似的方法，只是对如何选择种子进行了更广泛的探索。

一些概率框架也被设计用于半监督聚类[72, 14]。[72]中的工作使用了一个迭代的EM方法，在这个方法中，未标记的文档被分配了标签，使用的是当前标记的文档的天真贝叶斯方法。然后，这些新标记的文档再次被用于重新训练贝叶斯分类器。这个过程反复进行直到收敛。[72]中的迭代标记方法可以被认为是对未标记的文档进行聚类的部分监督方法。[14]中的工作使用异质马尔科夫随机场（HMRF）模型进行聚类过程。[52]中提出了一种基于图的方法，将先验知识纳入聚类过程。在这种方法中，文档被建模为一个图，其中节点代表文档，边代表它们之间的相似性。 新的边也可以被添加到这个图中，这些边对应于先前的知识。具体来说，当根据先验知识知道这两个文件是相似的时候，就会在图中添加一条边。然后，一个归一化的切割算法[84]被应用到这个图上，以建立最终的聚类。这种方法隐含地使用了先验知识，因为增强的图表示法被用于聚类。

由于半监督聚类在聚类和分类问题之间形成了一个自然的桥梁，所以半监督方法自然也可以用于分类[68]。这也被称为*联合训练*，因为它涉及到使用无监督的文档聚类，以协助训练过程。由于半监督方法同时使用了特征空间中的聚类结构和类别信息，它们在分类场景中有时更加稳健，特别是在可用的标记数据量较少的情况下。在文献[72]中显示，当可用的训练数据量较少时，混合了监督和非监督数据的部分监督协同训练方法可以产生更有效的分类结果。[72]中的工作使用了一个部分监督的EM算法，该算法迭代地将标签分配给未标记的文档，并随着时间的推移在达到收敛时对其进行完善。在[4, 14, 35, 47, 89]中也提出了一些类似的方法，在聚类过程中具有不同程度的监督。部分监督的聚类方法也在分类中使用特征转换的方法，如[17, 18, 88]中所讨论的。其想法是，聚类结构提供了一个压缩的特征空间，它能很好地捕捉到相关的分类结构，因此可以对分类有所帮助。

部分监督的方法也可以与预先存在的分类层次（或原型层次）结合使用[4, 56, 67]。原型层次结构的一个典型例子是*雅虎的*分类法、*开放目录项目*或*路透社的集合*。我们的想法是，这样的层次结构提供了一个很好的聚类结构的总体概念，但由于其典型的人工来源，其中也有相当大的噪音和重叠。部分监督能够纠正噪音和重叠，这导致了一个相对干净和一致的聚类结构。

对文件聚类的一种不寻常的监督方法是使用已知不属于一个聚类的文件的*普遍性*[106]。这主要是指那些不能自然地被归入任何特定组别的背景分布。其直觉是，例子的普遍性提供了一种有效的方法来避免聚类过程中的错误，因为它提供了一个例子的背景来比较一个聚类。

9. 结论和总结

在这一章中，我们介绍了对文本数据的聚类算法的调查。一个好的文本聚类需要有效的特征选择和

为手头的任务正确选择算法。在不同类别的算法中，基于距离的方法在广泛的应用中是最受欢迎的。

近年来，该领域的主要研究趋势是在两种文本数据的背景下进行的。

- **动态应用。**由动态应用（如社交网络或在线聊天应用）产生的大量文本数据，为流式文本聚类应用创造了巨大的需求。这种流式应用需要适用于不太干净的文本，如社交网络等应用中经常出现的情况。
- **异质性应用。**文本应用越来越多地出现在异质应用中，其中文本在链接和其他异质多媒体数据的背景下可用。例如，在*Flickr*等社交网络中，聚类经常需要在这种情况下应用。因此，有效地使基于文本的算法适应异质性多媒体场景是至关重要的。

我们注意到，文本聚类的领域太过广泛，无法在一个章节中全面涵盖。一些方法，如*基于委员会的聚类*[73]，甚至不能被整齐地纳入任何一类方法中，因为它们使用不同聚类方法的组合，以创造一个最终的聚类结果。 本章的主要目的是对该领域经常使用的主要算法进行全面概述，作为进一步研究的出发点。

参考文献

[1] C.C. Aggarwal, Y. Zhao, P. S. Yu.论带有侧面信息的文本聚类, *ICDE会议*, 2012.

[2] C.C. Aggarwal, P. S. Yu.论文本中有效的概念索引和相似性搜索, *ICDM会议*, 2001.

[3] C.C. Aggarwal, P. S. Yu.海量文本和分类数据流的聚类框架, *SIAM数据挖掘会议*, 2006.

[4] C.C. Aggarwal, S. C. Gates, P. S. Yu.On Using Partial Supervi- sion for Text Categorization, *IEEE Transactions on Knowledge and Data Engineering*, 16（2）, 245-255, 2004.

[5] C.C. Aggarwal, C. Procopiuc, J. Wolf, P. S. Yu, J.-S. Park.Park.Fast Algorithms for Projected Clustering, *ACM SIGMOD Conference*, 1999.

[6] C.C. Aggarwal, P. S. Yu.寻找高维空间中的广义投影集群, *ACM SIGMOD会议*, 2000.

[7] R.Agrawal, J. Gehrke, P. Raghavan.D. Gunopulos.用于数据挖掘应用的高维数据的自动子空间聚类，*ACM SIGMOD会议*，1999。

[8] R.Agrawal, R. Srikant.Fast Algorithms for Mining Association Rules in Large Databases, *VLDB Conference*, 1994.

[9] J.Allan, R. Papka, V. Lavrenko.Online new event detection and tracking.*ACM SIGIR会议*, 1998.

[10] P.Andritsos, P. Tsaparas, R. Miller, K. Sevcik.LIMBO：分类数据的可扩展聚类。*EDBT会议*, 2004.

[11] P.Anick, S. Vaithyanathan.Exploiting Clustering and Phrases for Context-Based Information Retrieval.*ACM SIGIR Conference*, 1997.

[12] R.Angelova, S. Siersdorfer.A neighborhood-based approach for clustering of linked document collections.*CIKM会议*, 2006.

[13] R.A. Baeza-Yates, B. A. Ribeiro-Neto, *Modern Information Retrieval - the concepts and technology behind search, Second* ed. Pearson Education Ltd., Harlow, England, 2011.

[14] S.Basu, M. Bilenko, R. J. Mooney.A probabilistic framework for semi-supervised clustering.*ACM KDD Conference*, 2004.

[15] S.Basu, A. Banerjee, R. J. Mooney.Semi-supervised Clustering by Seeding.*ICML会议*, 2002.

[16] F.Beil, M. Ester, X. Xu.基于频繁术语的文本聚类，*ACM KDD会议*，2002。

[17] L.Baker, A. McCallum.用于文本分类的词的分布式聚类，*ACM SIGIR会议*，1998。

[18] R.Bekkerman, R. El-Yaniv, Y. Winter, N. Tishby.On Feature Distributional Clustering for Text Categorization.*ACM SIGIR Conference*, 2001.

[19] D.Blei, J. Lafferty.动态主题模型.*ICML会议*, 2006.

[20] D.Blei, A. Ng, M. Jordan.Latent Dirichlet allocation, *Journal of Machine Learning Research*, 3: pp.993-1022, 2003.

[21] P.F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J/ C. Lai。自然语言的基于类的n-gram模型，*Compu- tational Linguistics*，18，4（1992年12月），467-479。

[22] K.Chakrabarti, S. Mehrotra.Local Dimension reduction:A new Approach to Indexing High Dimensional Spaces, *VLDB Conference*, 2000.

[23] J.Chang, D. Blei.文档网络的主题模型.*aista- sis*, 2009.

[24] W.B. Croft.使用单链路方法对大文件进行分类。*Journal of the American Society of Information Science*, 28: pp.341-344, 1977.

[25] D.Cutting, D. Karger, J. Pedersen, J. Tukey.Scatter/Gather:基于群集的方法来浏览大型文件集。*ACM SIGIR会议*, 1992.

[26] D.Cutting, D. Karger, J. Pederson.大型文件集的恒定交互时间扫描/收集浏览，*ACM SIGIR会议*，1993。

[27] M.Dash, H. Liu.Feature Selection for Clustering, *PAKDD Conference*, pp.110-121, 1997.

[28] S.Deerwester, S. Dumais, T. Landauer, G. Furnas, R. Harshman.Indexing by Latent Semantic Analysis.*JASIS*, 41(6), pp. 391-407, 1990.

[29] I.Dhillon, D. Modha.Concept Decompositions for Large Sparse Data using Clustering, 42(1), pp.143-175, 2001.

[30] I.Dhillon.使用双点谱图分区对文档和单词进行联合聚类, *ACM KDD会议*, 2001.

[31] I.Dhillon, S. Mallela, D. Modha.Information-theoretic Co-Clustering, *ACM KDD Conference*, 2003.

[32] C.Ding, X. He, H. Zha, H. D. Simon.Adaptive Dimension Re-duction for Clustering High Dimensional Data, *ICDM Conference*, 2002.

[33] C.Ding, X. He, H. Simon.论非负矩阵分解和谱系聚类的等价性.*SDM会议*, 2005.

[34] B.Dorow, D. Widdows.发现语料库特定的词义，*第十届计算语言学协会欧洲分会会议论文集-第二卷（EACL'03）*，第79-82页，2003年。

[35] R.El-Yaniv, O. Souroujon.Iterative Double Clustering for Unsupervised and Semi-supervised Learning.*NIPS会议*, 2002.

[36] H.Fang, T. Tao, C. Zhai, A formal study of information retrieval heuristics, *Proceedings of ACM SIGIR 2004*, 2004.

[37] D.Fisher.通过增量概念分类的知识获取。*机器学习*，2：139-172页，1987。

[38] M.Franz, T. Ward, J. McCarley, W.-J. Zhu.主题跟踪的无监督和有监督的聚类。*ACM SIGIR会议*, 2001.

[39] G. P. C. Fung, J. X. Yu, P. Yu, H. Lu.文本流中的无参数突发事件检测, *VLDB会议*, 2005.

[40] J. H. Gennari, P. Langley, D. Fisher.增量概念形成的模型。*Journal of Artificial Intelligence*, 40 pp.11-61, 1989.

[41] D.Gibson, J. Kleinberg, P. Raghavan.Clustering Categorical Data:An Approach Based on Dynamical Systems, *VLDB Conference*, 1998.

[42] M.Girolami, A Kaban.On the Equivalance between PLSI and LDA, *SIGIR会议*，第433-434页，2003年。

[43] S.Guha, R. Rastogi, K. Shim.ROCK: a robust clustering algo- rithm for categorical attributes, *International Conference on Data Engineering*, 1999.

[44] S.Guha, R. Rastogi, K. Shim.CURE:用于大型数据库的高效聚类算法.*ACM SIGMOD会议*, 1998.

[45] D.Gusfield.字符串、树和序列的算法》，*剑桥大学出版社*，1997年。

[46] Y.Huang, T. Mitchell.带有扩展用户反馈的文本聚类。*ACM SIGIR会议*, 2006.

[47] H.Li, K. Yamanishi.使用有限混合模型的文档分类.*计算语言学协会年度会议*，1997。

[48] Q.He, K. Chang, E.-P.Lim, J. Zhang.用于文本流聚类的突发性特征表示.*SDM会议*, 2007.

[49] T.Hofmann.Probabilistic Latent Semantic Indexing.*ACM SIGIR会议*, 1999.

[50] A.Jain, R. C. Dubes.Algorithms for Clustering Data, *Prentice Hall*, Englewood Cliffs, NJ, 1998.

[51] N.Jardine, C. J.van Rijsbergen.分层聚类在信息检索中的应用，*信息存储和检索*，7：217-240页，1971年。

[52] X.Ji, W. Xu.带有先验知识的文档聚类.*ACM SIGIR会议*, 2006.

[53] I.T. Jolliffee.Principal Component Analysis.*Springer*, 2002.

[54] L.Kaufman, P. J. Rousseeuw.Finding Groups in Data:An Intro-duction to Cluster Analysis, *Wiley Interscience*, 1990.

[55] W.Ke, C. Sugimoto, J. Mostafa.Dynamicity vs. effectiveness: studying online clustering for scatter/gather.*ACM SIGIR Confer- ence*, 2009.

[56] H.Kim, S. Lee.A Semi-supervised document clustering technique for information organization, *CIKM Conference*, 2000.

[57] J.Kleinberg, Bursty and hierarchical structure in streams, *ACM KDD Conference*, pp.91-101, 2002.

[58] D.D. Lee, H. S. Seung.通过非负矩阵分解学习物体的部分，*自然*，401：第788-791页，1999。

[59] T.Li, S. Ma, M. Ogihara, Document Clustering via Adaptive Sub-space Iteration, *ACM SIGIR Conference*, 2004.

[60] T.Li, C. Ding, Y. Zhang, B. Shao.从单词空间到文档空间的知识转换.*ACM SIGIR会议*, 2008.

[61] Y.-B.Liu, J.-R.Cai, J. Yin, A. W.-C.Fu.Clustering Text Data Streams, *Journal of Computer Science and Technology*, Vol. 23(1), pp.112-128, 2008.

[62] T.Liu, S. Lin, Z. Chen, W.-Y. Ma.Ma.对文本聚类的特征选择的评估, *ICML会议*, 2003.

[63] Y.吕晓明，梅晓明，翟晓明.调查概率主题模型的任务表现：PLSA和LDA的实证研究, *信息检索*, 14(2):178-203 (2011).

[64] A.McCallum.Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow, 1996.

[65] Q.Mei, D. Cai, D. Zhang, C.-X.Zhai.网络正则化的主题建模.*WWW会议*, 2008.

[66] D.Metzler, S. T. Dumais, C. Meek, Similarity Measures for Short Segments of Text, *Proceedings of ECIR 2007*, 2007.

[67] Z.Ming, K. Wang, T.-S.Chua.基于层次结构的原型聚类，用于网络集合的分类和导航。*ACM SIGIR会议*, 2010.

[68] T.M. Mitchell.无标签数据在监督学习中的作用。*第六届认知科学国际学术会议论文集*，1999年。

[69] F.Murtagh.A Survey of Recent Advances in Hierarchical Cluster- ing Algorithms, *The Computer Journal*, 26(4), pp.354-359, 1983.

[70] F.Murtagh.Hierarchical Clustering Algorithms的复杂性。State of the Art, *Computational Statistics Quarterly*, 1(2), pp.101- 113, 1984.

[71] R.Ng, J. Han.Efficient and Effective Clustering Methods for Spatial Data Mining.*VLDB会议*, 1994.

[72] K.Nigam, A. McCallum, S. Thrun, T. Mitchell.Learning to clas- sify text from labeled and unlabeled documents.*AAAI会议*，1998年。

[73] P.Pantel, D. Lin.Document Clustering with Committees, *ACM SIGIR Conference*, 2002.

[74] G. Qi, C. Aggarwal, T. Huang.社会媒体网络中边缘内容的社区检测, *ICDE会议*, 2012.

[75] M.Rege, M. Dong, F. Fotouhi.Co-clustering Documents and Words Using Bipartite Isoperimetric Graph Partitioning.*ICDM Confer- ence*, pp.532-541, 2006.

[76] C.J. van Rijsbergen.*信息检索*》，Butterworths，1975。

[77] C.J.van Rijsbergen, W. B. Croft.Document Clustering:An Eval- uation of some experiments with the Cranfield 1400 collection, *In- formation Processing and Management*, 11, pp.171-182, 1975.

[78] S.E. Robertson and S. Walker.对概率加权检索的2-泊松模型的一些简单有效的近似值。In *SIGIR*, pages 232-241, 1994.

[79] M.Sahami, T. D. Heilman, A web-based kernel function for mea- suring the similarity of short text snippets, *Proceedings of WWW 2006*, pages 377-386, 2006.

[80] N.Sahoo, J. Callan, R. Krishnan, G. Duncan, R. Padman.Incre- mental Hierarchical Clustering of Text Documents, *ACM CIKM Conference*, 2006.

[81] G. Salton.现代信息检索简介》，*Mc Graw Hill*, 1983.

[82] G. Salton, C. Buckley.Term Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management*, 24(5), pp.513-523, 1988.

[83] H.Schutze, C. Silverstein.Projections for Efficient Document Clustering, *ACM SIGIR Conference*, 1997.

[84] J.Shi, J. Malik.归一化切割和图像分割.*IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2000.

[85] C.Silverstein, J. Pedersen.Almost-constant time clustering of ar- bitrary corpus subsets.*ACM SIGIR Conference*, pp. 60-66, 1997.

[86] A.Singhal, C. Buckley, M. Mitra.Pivoted Document Length Nor- malization.*ACM SIGIR Conference*, pp. 21-29, 1996.

[87] N.Slonim, N. Tishby.通过信息瓶颈法使用词组进行文档聚类，*ACM SIGIR会议*，2000。

[88] N.Slonim, N. Tishby.The power of word clusters for text classification.*European Colloquium on Information Retrieval Research (ECIR)*, 2001.

[89] N.Slonim, N. Friedman, N. Tishby.使用顺序信息最大化的无监督文档分类。*ACM SIGIR会议*, 2002.

[90] M.Steinbach, G. Karypis, V. Kumar.文档聚类技术的比较，*KDD文本挖掘研讨会*，2000。

[91] Y.Sun, J. Han, J. Gao, Y. Yu. iTopicModel:信息网络综合主题建模，*ICDM会议*, 2009.

[92] E.M. Voorhees.Implementing Agglomerative Hierarchical Cluster- ing for use in Information Retrieval, *技术报告 TR86-765, Cornell University, Ithaca, NY,* July 1986.

[93] F.Wang, C. Zhang, T. Li.文件的规则化聚类.
ACM SIGIR会议, 2007.

[94] J.Wilbur, K. Sirotkin.停顿词的自动识别。
J.Inf.Sci., 18: pp.45-55, 1992.

[95] P.Willett.使用倒置文件的方法进行文档聚类。
信息科学杂志》，2：第223-231页，1980年。

[96] P.Willett.分层文档聚类的最新趋势:A Critical Review.*信息处理和管理*，24⑤：第577-597页，1988年。

[97] W.Xu, X. Liu, Y. Gong.基于非负矩阵分解的文档聚类, *ACM SIGIR会议*, 2003.

[98] W.Xu, Y. Gong.概念因子化的文档聚类.
ACM SIGIR会议, 2004.

[99] Y.Yang, J. O. Pederson.文本分类中特征选择的比较研究，*ACM SIGIR会议*，1995。

[100] Y.Yang.在文本分类的统计方法中减少噪音, *ACM SIGIR会议*, 1995.

[101] T.Yang, R. Jin, Y. Chi, S. Zhu.Combining link and content for community detection: a discriminative approach.*ACM KDD Conference*, 2009.

[102] L.Yao, D. Mimno, A. McCallum.Efficient methods for topic model inference on streaming document collections, *ACM KDD Conference*, 2009.

[103] O.Zamir, O. Etzioni.网络文档聚类。A Feasibility Demonstration, *ACM SIGIR Conference*, 1998.

[104] O.Zamir, O. Etzioni, O. Madani, R. M. Karp.快速和直观的网络文档聚类，*ACM KDD会议*，1997。

[105] C.Zhai，《*信息检索的统计语言模型*》（*人类语言技术综合讲座*），Morgan & Claypool出版公司，2008。

[106] D.Zhang, J. Wang, L. Si.用universum进行文档聚类. *ACM SIGIR会议*, 2011.

[107] J.Zhang, Z. Ghahramani, Y. Yang.在线文档聚类的概率模型，适用于新颖性检测。In *Saul L., Weiss Y., Bottou L. (eds) Advances in Neural Information Processing Letters*, 17, 2005.

[108] T.Zhang, R. Ramakrishnan, M. Livny.BIRCH:一个用于非常大的数据库的高效数据聚类方法.*ACM SIGMOD Con- ference,* 1996.

[109] X.Zhang, X. Hu, X. Zhou.A comparative evaluation of different link types on enhancing document clustering.*ACM SIGIR Confer- ence*, 2008.

[110] Y.Zhao, G. Karypis.评估文档数据集的分层聚类算 法 , *CIKM会议*, 2002.

[111] Y.Zhao, G. Karypis.Empirical and Theoretical comparisons of selected criterion functions for document clustering, *Machine Learning*, 55(3), pp.311-331, 2004.

[112] S.Zhong.高效的流媒体文本聚类.*Neural Networks*, Volume 18, Issue 5-6, 2005.

[113] Y.Zhou, H. Cheng, J. X. Yu.基于结构/属性相似性的图聚类, *VLDB会议*, 2009.

[114] http://www.lemurproject.org/

SOM和K-means在文本聚类中的比较

陈一恒 (通讯作者)
哈尔滨工业大学计算机科学与技术学院 邮政信箱：321，哈尔滨，
150001，中国
电话：86-451-8641-3683E-mail ：cyh@ir.hit.edu.cn

秦兵
哈尔滨工业大学计算机科学与技术学院 邮政信箱：321，哈尔滨，
150001，中国
电话：86-451-8641-3683E-mail ：qinb@ir.hit.edu.cn

刘婷
哈尔滨工业大学计算机科学与技术学院 邮政信箱：321，哈尔滨，
150001，中国
电话：86-451-8641-3683E-mail ：tliu@ir.hit.edu.cn

刘远超
哈尔滨工业大学计算机科学与技术学院 哈尔滨，150001，中国
电话：86-451-8641-6460E-mail ：lyc@hit.edu.cn

李胜
哈尔滨工业大学计算机科学与技术学院 邮政信箱：321，哈尔滨，
150001，中国
电话：86-451-8641-3683E-mail ：lis@ir.hit.edu.cn

摘要

SOM和K-means是两种经典的文本聚类方法。在本文中，我们做了一些实验来比较它们的性能。所用的样本数据是来自不同主题的420篇文章。K-means方法简单，容易实现；SOM的结构相对复杂，但聚类结果更直观，容易理解。比较结果还表明，K-means对主动性分布比较敏感，而SOM的整体聚类性能优于K-means，并且在检测噪声文档和拓扑结构保存方面表现良好，从而使其更适合于一些应用，如文档收集的导航、多文档总结等，而SOM的聚类结果对输出层拓扑结构比较敏感。

关键词文本聚类，自组织地图，K-均值，聚类算法

1. 简介

随着网络新闻、电纸书、E-mail文档等众多文本文档在互联网上的爆炸性增长，如何组织和浏览这些文档的任务变得越来越重要和迫切。作为一种无监督的机器学习方法，文本聚类由于能够有效地组织文本文档而吸引了许多研究者（马帅，王腾蛟，003）（王爱华，张明，2001）（吴斌，傅伟鹏，史忠志，2002）。文本聚类的应用包括。文本聚类可以作为其他技术的预处理步骤，如多文档总结（Vasileios Hatzivassiloglou, Judith L. Klavans, 2001）；②对搜索引擎返回的结果进行聚类，从而使用户能够快速找到他需要的东西（Cutting, D., Karger, D., Pedersen, J. and Tukey, J. W. 1992）；③自动组织文本

采集，如Cutting开发的文本导航系统Scatter/Gather(Cutting, D., Karger,D.,Pedersen, J. and Tukey, J. W. 1992)；文本聚类还可以通过处理用户感兴趣的文档或网页来挖掘出特定用户的兴趣模型，从而将缩小的但更相关的文档推荐并推送给用户。此外，微软公司的文继荣（JR Wen, JY Nie, HJ Zhang, 2001）利用文本聚类技术对许多用户的查询记录进行聚类，以更新网站的FAQ。

与其他类型的数据相比，文本聚类中的文档具有很多语义特征，通常在高维特征空间中以向量形式表示。通过对输入文档的聚类过程，可以发现隐藏在许多文档中的主题结构，具有相同主题或非常接近主题的文档会被放入同一个聚类中，而具有不同主题的文档会被分离到不同的聚类中。文本聚类的证明是著名的聚类假设（N. Jardine , C. J. van Rijsbergen, 1971）：密切相关的文档属于一个类别，与同一个查询相关。

在众多的文本聚类方法中，AHC(P. H. Sneath , R. R. Sokal, 1973)、K-means、SOM似乎是三种方法，围绕这些技术出现了许多变体方法。AHC的聚类结果通常比较精细，但计算成本也比其他技术大，而k-means、SOM的效率则比AHC聚类技术要高。网络信息的快速增长要求计算效率必须更高，聚类结果必须易于理解。所以k-means和SOM比AHC方法更受欢迎。

由于文本聚类有很多应用背景，而不同的应用对聚类质量、计算效率和导航能力有不同的要求。因此，有必要根据实际应用背景选择合适的文本聚类方法。

在本文中，我们首先对两种文本聚类方法的基本原理做了分析。SOM和K-means。然后对这两种技术的实际性能进行了一些实验比较和相应的分析，如对初始分布的敏感性和不同情况下的F度量。我们期望通过我们的工作，这两种流行的文本聚类方法的实际性能可以被清楚地显示出来，并可以为相关的研究提供一些参考。

2. 用于文本聚类的K-means

K-means是基于分区的聚类方法。当k-means被用于文本聚类时，所有的文档将被随机放入k个聚类中，然后根据一些原则调整聚类的分区，直到聚类结果稳定。k-means用于文本聚类的基本原理可以描述如下。

输入：N个待聚类的文件，聚类数k

输出：K个集群，每个文件将被分配到一个集群中。

1) 随机选择k个文档作为初始聚类文档的种子。

2) 重复以下两个步骤，如果分区稳定，则转到步骤5）。

3) 根据每个聚类中所有文档的平均向量，将每个文档分配到最相似的聚类中。

4) 根据每个聚类中的文档向量，更新每个聚类的平均向量。

5) 输出生成的群集和分区

3. 用于文本聚类的SOM

SOM(Self-Organizing feature Maps)是由T.Kohonen教授提出的(T.Kohonen, 1982)，当经过充分的训练后，SOM网络的输出层会被分成不同的区域。而不同的神经元对不同的输入样本会有不同的反应。由于这个过程是自动的，所有的输入文件都会被聚类。由自然语言编写的文本文件是高维度的，具有很强的语义特征。在高维空间中很难浏览许多文件。而SOM可以将所有这些高维度的文档映射到2维或1维的空间中，并且它们在原始空间中的关系也可以被保留。此外，SOM对一些嘈杂的文档不是很敏感，聚类的质量也可以得到保证。由于这些优点，SOM技术适合于文本聚类，并已被用于一些领域，如数字图书馆（K.Lagus,T.Honkela,S.Kaski,and T.Kohonen, 1996）。许多SOM的变体也出现了（T.Kohonen, 1998）（D.Merkl, 1993）。

SOM用于文本聚类的原理可以归纳为以下几点。

1) 初始化。为输出层的所有神经元分配一些随机数。并进行归一化处理。神经元的维数与所有文件的维数相同。

2) 输入样本。从文档集合中随机选择一个文档，并将其发送到SOM网络。

3) 找出获胜的神经元。计算输入文件向量和神经元向量之间的相似度， 相似度最高的神经元将成为赢家。

4) 适应赢家和其邻居的向量。适应可以使用以下公式。

$$_{mi}(t \quad 1) = _{mi}(t) + \quad (t) \bullet _{hi}(t) \quad [x(t) - _{mi}(t)] \tag{1}$$

其中$x(t)$是文件向量或时间t。 $_{mi}(t)$ 是神经元的原始向量。 $_{mi}(t \quad 1)$ 是适应后的神经元向量。 (t)和 $_{嗨}(t)$ 分别是学习率和邻居率。

$| x(t) - _{mi}(t) |$ 代表神经元向量和文档向量之间的距离。

适应之后，获胜者及其邻居更接近输入的文档向量，因此，当再次输入类似的文档时，这些神经元将更具竞争力。通过输入足够的样本对SOM网络进行充分的训练，输出层的神经元将只对某个主题的文档敏感，它们的向量将成为这些主题的平均向量。

4. SOM和K-means的比较

K-means很容易实现，而且它的计算成本通常很低，所以它已经成为一种著名的文本聚类方法，并被许多领域所采用（D.R. Cutting, J.O. Pedersen, D.R. Karger, and J.W. Tukey, 1992）（Daniel Boley, 1998）。k-means的缺点是，K值必须事先确定，而且初始文档种子需要随机选择。而这些初始设置会对聚类结果产生影响。K-means本质上是一种贪婪的算法，它很难达到全局最优的聚类结果（R. Ng, J. Han, 1992）。

当使用k-means对文件进行聚类时，有一些规则。

(1) 在初始设置（k值和文档种子）确定后，聚类结果也将被确定。但如果初始设置不同，聚类结果也会不同。

(2) 当初始设置（k值和文档种子）确定后，假设迭代n+1的聚类结果与迭代n相同，那么迭代n+m的聚类结果将与迭代n（m>1）相同。因此，分区是否有变化可以作为聚类迭代的停止标准。

在SOM网络中，输出层的神经元数量与输入文件集中的类别数量有密切关系。如果神经元的数量小于类的数量，将不足以分离所有的类，一些密切相关的类的文件可能被合并到一个类中。如果神经元数量多于类的数量，聚类结果可能过于精细。而聚类效率和聚类质量也可能受到不利影响。

k-means的计算复杂度是O(KlN),其中l是迭代次数，N是文档数量。k-means和SOM的计算复杂度是O(kmN)，其中k是神经元数量，m是训练次数。这两种方法的计算复杂度非常接近。它们都比ACH方法低。

5. 实验结果和劝阻

通过实验比较了这两种方法在文本聚类中的实际表现。实验中的文件集有645个关于不同主题的文件。它们的基本属性列于表1。

F测量法（Michael Steinbach, 2002）被用作系统性能的评价函数。对于一个生成的集群r和一个预定义的类s。

$$recall(r, s) = n(r, s) / _{ns} \tag{2}$$

$$precision(r, s) = n(r, s) / _{nr} \tag{3}$$ (3)

*n(r,s)*是r和s之间的交集的文档编号。n_r 是群集r的文档编号，n_s是类s的文档编号。群集r和类s之间的F测量可以计算为

$$F(r,s) = (2 * recall(r,s) * precision(r,s))/((Precison(r,s) + recall(r,s)) \quad (4) \qquad (4)$$

总体的F衡量标准可以计算为

$$F = \sum_{i} \frac{ni}{n} max\{\mathrm{F}(i,j)\} \quad (5)$$

其中n是所有测试文档的数量。n_i 是第i类的文档数量。一般来说，F值越大意味着聚类结果越好。

首先，我们研究了训练次数对SOM性能的影响。这里我们把训练次数定义为C乘以输入文档的大小。例如，如果有100个文档需要聚类，C=10，我们应该把训练次数设置为100*10=1000。我们的实验表明，当C较低时，SOM文本聚类的性能会随着C值的增加而快速增长。当C值较高时，SOM的性能对C值不是很敏感。

表2显示了SOM的一个聚类结果，输入是4个类别的文件。1,2,3,4.SOM的拓扑结构是矩形的，输出层包括2*2=4个神经元，C=20，在表2中，每一行表示当C值不同时，这4个神经元TL（左上）、TR（右上）、DL（左下）、DR（右下）所映射的文档数量。例如，"TL=29:0:0:5 "表示神经元TL映射了1类的29个文档，4类的5个文档，2、3类的没有。在我们的实验中，我们还发现，当训练次数足够多时，文本聚类的纯度也会提高，这将有助于提高一些自然语言处理的质量，如多文档总结、TDT（话题检测和跟踪）等。

如上所述，SOM和K-means都需要一个初始化的过程。我们比较了它们是否对初始设置敏感。我们将SOM和K-means的k值设置为相等，并比较它们在k=4和k=9时的性能。对于SOM，其输出层的拓扑结构为2*2=4和3*3=9，C=20。当训练结束时，SOM输出层的每个神经元表示一个类别的文档。

在表3和表4中，显示了两种方法运行20次的平均F值。SOM对初始设置不敏感。而k-means的聚类结果并不稳定，每次运行的迭代次数也不同。事实上，如果能够选择合适的初始文档种子，k-means将快速收敛，并能获得更好的聚类质量。由于标准的k-means通常随机选择种子，聚类质量会受到不利影响。因此，当k-means被用于文本聚类时，有必要使用一些方法来选择合适的种子（如最小-最大原则、基于密度的方法等）。

我们的实验结果也证明，当神经元数量多于输入文档的类数时，由于SOM的训练倾向于充分利用每个神经元，一些类可能由2个以上的神经元代表。在这种情况下，这些类的文件通常会被映射到一些邻近的神经元上，如表5和表6所示。在表5中，输出层有3*3=9个神经元，输入文档中有6个类。神经元N11、N33可以分别代表一个类，而N21、N8实际上代表一个普通类。在表6中，有2*4=8个神经元，而输入文件有5个类。神经元N21本身可以代表一个类。而神经元N11、N12、N22实际上代表一个共同的类。

所有这些实验结果表明，SOM的拓扑结构对聚类质量有明显影响。然而，SOM的聚类结果可以提供良好的导航能力，从而使聚类变得有意义和容易理解。在SOM的输出层，相邻的神经元通常会映射相似的文档。因此，用户可以非常迅速地找到他们需要的文件，信息获取的效率得到了极大的提高。在许多应用中，可以设置更多的神经元（超过可能的集群数量）来对文档进行集群。相比之下，k-means需要用户提供k值来开始聚类。文本聚类的无监督特性将受到影响，因为在大多数情况下，用户对输入文档的主题结构知之甚少。

在某些情况下直接比较SOM和k-means的聚类质量（SOM输出层的神经元数量与k-means的k值相同）。当SOM的输出层为2*2、2*3、2*4和3*3时，两种方法的F测量结果如表7所示。每次选择4个文档类别的组合，利用10个聚类结果的平均值作为总体F值。可以看出，SOM的整体聚类质量明显优于K-means。这表明，当SOM的输出层设置合理时，即输出层的神经元能够得到充分的利用，SOM能够取得更好的

聚类质量。k-means的聚类性能对初始设置非常敏感，因此使其聚类质量不稳定，其F值小于SOM。

6. 总结

在本文中，两个文本聚类方法的性能。本文通过实验分析和比较了两种文本聚类方法：SOM和K-means。实验结果表明，k-means对k值和文档种子等初始设置非常敏感。而当输出层的神经元能够被充分利用时，SOM可以实现更好的文本聚类质量。SOM在噪声容忍度和拓扑结构保持方面也有较好的表现，使其成为一种有待进一步研究的文本聚类方法。

参考文献

Cutting, D., Karger, D., Pedersen, J. and Tukey, J. W. (1992).Scatter/Gather:一个基于集群的方法来浏览大型文件集。*第15届国际ACM/SIGIR年会论文集*，哥本哈根，1992：318-329。

D.Merkl.(1993).为重用而构建软件：自组织地图的案例。In International Joint Conference on Neural Networks.名古屋会议中心，日本，1993，(3):2468-2471

D.R. Cutting, J.O. Pedersen, D.R. Karger, and J.W. Tukey。(1992).Scatter/gather:基于集群的方法来浏览大型文档集。在ACM SIGIR会议记录中，哥本哈根，1992：318-329

Daniel Boley.(1998).Principal direction divisive partitioning.Data Mining and Knowledge Discovery.1998, 2(4):325-344

JR Wen, JY Nie, HJ Zhang.(2001).对搜索引擎的用户查询进行分类。第十次国际万维网会议.香港，2001：162-168.

K.Lagus,T.Honkela, S.Kaski, and T.Kohonen.(1996).文件集的自组织地图。互动探索的新方法。第二届知识发现和数据挖掘国际会议论文集》，AAAI出版社，加利福尼亚州门洛帕克。1996:238-243.

M.Steinbach, G. Karypis, and V. Kumar.(2000).文档聚类技术的比较。In KDD Workshop on Text Mining, Boston, MA, USA.

Ma, Shuai, Wang, TengJiao.(2003).一种基于参考值和密度的快速聚类算法，*Journal of Software* .2003,14(6):1089-1095.

Michael Steinbach, George Karypis, Vipin Kumar.(2002).文件聚类技术的比较.明尼苏达大学计算机科学与工程系。技术报告#00-034。

N.Jardine, C. J. van Rijsbergen.(1971).在信息检索中使用分层聚类。*Information Storage and Retrieval.*1971(7):217-240.

P.H. Sneath, R. R. Sokal.(1973).*数学分类学*》。Freeman, London, UK, 1973.

R.Ng, J. Han.(1994).用于空间数据挖掘的高效和有效的聚类方法.In Proc. of the 20th VLDB Conference, Santiago, Chile, 1994:144-155

T.Kohonen.(1982).拓扑学上正确的特征图的自组织形成。Biological Cybernetics.1982(43):59-69

T.Kohonen.(1998).非常大的文件集的自我组织。State of the art.ICANN98会议记录，第八届国际人工神经网络会议，Springer，伦敦。1998(1):65-74.

Vasileios Hatzivassiloglou, Judith L. Klavans.(2001).SIMFINDER:A Flexible Clustering Tool for Summarization.NAACL自动总结研讨会（宾夕法尼亚州匹兹堡），2001：41-49。

王爱华, 张明.(2001).PCCS: A Fast Clustering And Classification Method for Web Document. *Journal of Computer Research and Development*.2001,38(4):415-421.

吴斌, 傅伟本, 史忠志.(2002).一种基于群集智能的网络文档聚类算法.*Journal of Computer Research and Development.*2002,39(11):1429-1435.

表1.数据集的基本信息

分类标识	类别说明	文件编号	分类标识	类别说明	文件编号
1	甲壳类动物	50	9	MBA	40
2	苹果	50	10	MP3	40
3	密封	50	11	游戏	40
4	璐 永祥（中文名）	40	12	约旦 (英文名)	40
5	李国杰 (中国名字)	45	13	清华大学 大学	50
6	数码相机	50	14	旅游业	50
7	笑话	50	15	联想	50
8	音乐	50	16	健康	50

表2.C值对SOM性能的影响（输入文件为1-4类）。

C	F措施	TL	ÃÃÃ	DL	医师
1	0.79	29:0:0:5	0:0:0:25	0:16:28:0	1:14:2:0
2--4	0.79	0:0:0:25	29:0:0:5	0:16:28:0	1:14:2:0
5--9	0.79	0:0:0:25	29:0:0:5	0:16:28:0	1:14:2:0
12	0.88	28:0:0:6	0:0:0:24	2:30:2:0	0:0:28:0
13	0.88	0:0:0:24	29:0:0:6	0:0:28:0	1:30:2:0
14	0.88	30:0:0:6	0:30:2:0	0:0:0:24	0:0:28:0
15	0.90	0:0:28:0	1:30:2:0	0:0:0:26	29:0:0:4
16	0.90	0:0:28:0	28:0:0:3	2:30:2:0	0:0:0:27
17--19	0.92	3:0:0:29	27:0:0:1	0:0:29:0	0:30:1:0
20	0.93	29:0:0:4	1:30:2:0	0:0:0:26	0:0:28:0
50	0.93	29:0:0:5	0:0:30:0	0:0:0:25	1:30:0:0
100	0.93	30:0:0:6	0:30:2:0	0:0:0:24	0:0:28:0

表3.初始设置对SOM和K-means的F值的影响（输入：1-4类的文件；维数：645；SOM输出层：2*2）

没有。	SOM (*C* =20)	k-means (迭代次数)	没有。	SOM (*C* =20)	k-means (迭代次数)
1	0.93	0.87(7)	11	0.93	0.87(6)
2	0.93	0.65(10)	12	0.93	0.92(7)
3	0.93	0.92(7)	13	0.93	0.65(7)
4	0.93	0.67(7)	14	0.93	0.66(6)
5	0.93	0.85(10)	15	0.93	0.72(7)
6	0.93	0.92(7)	16	0.93	0.85(9)
7	0.93	0.65(2)	17	0.93	0.65(9)
8	0.93	0.65(2)	18	0.93	0.90(6)
9	0.93	0.94(6)	19	0.93	0.92(7
10	0.93	0.65(7)	20	0.93	0.87(6)

表4.初始设置对SOM和K-means的F测量的影响

(输入：A、B、C、D、E、F、G、H、I类文件；维数：912；SOM输出层：3*3)

没有。	SOM (C =20)	k-means (迭代次数：7-15)	没有。	SOM (C =20)	k-means (迭代次数：7-15)
1	0.91	0.92	11	0.91	0.88
2	0.91	0.75	12	0.91	0.75
3	0.91	0.90	13	0.91	0.75
4	0.91	0.89	14	0.91	0.98
5	0.91	0.91	15	0.91	0.88
6	0.91	0.75	16	0.91	0.75
7	0.91	0.89	17	0.91	0.75
8	0.91	0.91	18	0.91	0.93
9	0.91	0.88	19	0.91	0.88
10	0.91	0.90	20	0.91	0.75

表5.节点数大于班级数时的分班现象（输入：包括A,B,C,D,E,F的六个班级；SOM输出层：3*3)

	第1栏	第2栏	第3栏
第1行	0:0:30:0:0:0(N11)	15:1:0:4:2:0(N21)	14:0:0:0:0:0(N31)
2号线	0:0:0:0:0:13(N12)	1:1:0:8:0:0(N22)	0:0:0:18:0:0(N32)
三号线	0:12:0:0:0:17(N13)	0:16:0:0:0:0(N23)	0:0:0:0:28:0(N33)

表6.节点数大于班级数时的分班现象（输入：包括1-5的五个班级；SOM输出层：2*4）。

	第1栏	第2栏	第3栏	第4栏
1号线	0:0:0:5:0(N11)	0:0:0:0:30(N21)	0:0:4:0:0(N31)	30:0:0:0:0(N41)
2号线	0:0:0:19:0(N12)	0:0:0:6:0(N22)	0:0:26:0:0(N32)	0:30:0:0:0(N42)

表7.SOM和K-means的性能比较（K值已知）。

班级	F措施		班级	F措施	
	SOM(输出层)	K-means		SOM(输出层)	K-means
1-4	0.93(2*2)	0.76	1-8	0.92(2*4)	0.78
5-8	0.91(2*2)	0.81	3-10	0.87(2*4)	0.78
9-12	0.84(2*2)	0.78	5-12	0.86(2*4)	0.81
13-16	0.93(2*2)	0.80	7-14	0.92(2*4)	0.90
1-6	0.86(2*3)	0.73	9-16	0.91(2*4)	0.83
3-9	0.92(2*3)	0.81	1-8	0.81(3*3)	0.70
6-12	0.86(2*3)	0.71	5-12	0.85(3*3)	0.80
12-16	0.87(2*3)	0.79	9-16	0.89(3*3)	0.84

利用统计数据进行特征选择的文本聚类

李彦君、罗聪南和钟顺明，*IEEE会员*

摘要-- 特征选择是一种重要的方法，通过从语料库中去除多余的和不相关的术语来提高文本分类算法的效率和准确性。在本文中，我们提出了一种新的有监督的特征选择方法，即CHIR，该方法基于χ^2 统计量和新的统计数据，可以测量正向术语
类别的依赖性。我们还提出了一种新的文本聚类算法TCFS，它代表了带有特征选择的文本聚类。TCFS可以结合CHIR来反复识别相关特征（即术语），聚类成为一个学习过程。我们对TCFS和k-means聚类算法与不同的特征选择方法相结合，对各种真实数据集进行了比较。我们的实验结果表明，TCFS与CHIR在F-measure和纯度方面具有更好的聚类精度。

***索引词*--文本聚类，文本挖掘，χ^2 统计，特征选择，性能分析。**

I. 简介

如何探索和利用海量的文本文档是信息检索和文本挖掘领域的一个重要问题。文档聚类是最重要的文本挖掘方法之一，它的开发是为了帮助用户有效地浏览、总结和组织文本文档。通过将大量的文档组织成一些有意义的聚类，文档聚类可以用来浏览文档集合或组织搜索引擎响应用户查询而返回的结果。它可以显著提高信息检索系统的精度和召回率[18]，而且是寻找文档最近邻居的一种有效方法[3]。文档聚类的问题一般定义如下：给定一组文档，我们希望将其划分为预先确定的或自动得出的若干个聚类，使分配到每个聚类的文档比分配到不同聚类的文档更相似。换句话说，一个聚类中的文件共享同一个主题，而不同聚类中的文件代表不同的主题。

在大多数现有的文档聚类算法中，文档是用向量空间模型[18]来表示的，它把一个文档当作一个词包。这种表示方法的一个主要特点是特征空间的高维度，这给聚类算法的性能带来了很大的挑战。他们无法在高维度的由于数据的固有稀疏性，特征空间[1]。另一个问题是，并非所有的特征对于文档聚类都是重要的。有些特征可能是多余的或不相关的。有些甚至会误导聚类的结果，特别是当不相关的特征比相关的特征多的时候。在这种情况下，选择一个原始特征的子集往往能带来更好的聚类性能[12]。特征选择不仅可以降低特征空间的高维度，还可以提供更好的数据理解，从而改善聚类结果。所选的特征集应该包含关于原始数据集的足够或更可靠的信息。对于文档聚类来说，这将被表述为在一组文档中找出信息量最大的词进行聚类的问题。

特征选择已被广泛用于监督学习中，如文本分类。据报道，特征选择可以通过去除语料库中的冗余和不相关词汇来提高文本分类算法的效率和准确性[23]。传统的分类特征选择方法是有监督的还是无监督的，取决于每个文档是否需要类标签信息。那些无监督的特征选择方法，如使用文档频率和术语强度的方法，可以很容易地应用到聚类中[22]。但文献[12]表明，当文档的类标签可用于特征选择时，使用信息增益[16]和χ^2 统计的监督特征选择方法可以比非监督方法更好地提高聚类性能。然而，有监督的特征选择方法不能直接应用于文档聚类，因为通常所需的类标签信息是不可用的。在[12]中，提出了一种迭代特征选择（IF）方法，它利用监督特征选择来迭代选择特征并进行文本聚类。

在以往许多文本挖掘和信息检索的再搜索中，χ^2 术语-类别独立性检验被广泛用于文本分类前的单独预处理步骤中的特征选择[23]。通过对其χ^2 统计值进行排序，可以选择对类别有较强依赖性的特征[12]，[13]；本文将这种方法表示为CHI。最近提出了两种χ^2 统计量的变体。在[14]中，提出了*相关系数*，它可以被看作是 "单边 "的χ^2 统计。Galavotti等人[9]在这个方向上更进一步，提出了χ^2 统计量的简化变体，在[19]中被称为*GSS系数*。在[9]和[14]中，基于χ^2 统计量的这两个变体的特征选择方法被测试用于提高文本分类的性能。

在这项研究中，我们通过引入新的统计数据来扩展χ^2 术语-类别独立性检验，该数据可以衡量一个术语和一个类别之间的依赖关系是正还是负。这种新的统计数据可以更准确地描述术语-类别依存关系，而不是两个变体的

2006年9月26日收到稿件；2007年9月30日修订。李延军在计算机和信息科学系工作。
福特汉姆大学，Bronx, NY 10458。Email:yli@fordham.edu。
罗聪南在Teradata公司工作，地址：San Diego, CA 92127。Email:congnanluo@yahoo.com。
Soon M. Chung在莱特州立大学计算机科学与工程系，Dayton, OH 45435。Email:soon.chung@wright.edu。

数字对象标识符 10.1109/TKDE.2007.1907401041-4347/$25

x^2 统计量--相关系数和GSS系数。我们还开发了一种新的有监督的特征选择方法，名为CHIR，它是基于x^2 统计量和新的术语-类别依赖性测量。与CHI不同的是，CHIR选择的特征与类别有很强的正相关关系。换句话说，CHIR只保留与类别相关的特征。

此外，我们探索了CHIR在文本聚类中的应用，并开发了一种新的文本聚类算法，命名为TCFS，它代表了带有特征选择的文本聚类。与IF方法[12]不同的是，TCFS将有监督的特征选择方法，如CHIR，整合到文本聚类过程中，单独进行文本聚类和特征选择。因此，TCFS基本上是作为一个学习过程工作的。在利用聚类信息寻找更多相关特征（即术语）的同时，通过降低不相关特征的权重来提高聚类结果的质量。随着TCFS算法的收敛，可以得到一个好的聚类结果和一个信息丰富的特征子集。

我们对各种真实数据集的实验结果表明
结果表明，使用CHIR特征选择方法的TCFS算法比K-means、K-means与Term Strength(TS)特征选择方法[12]、IF方法、以及K-means算法的性能更好。
在聚类结果的准确性方面，TCFS与其他有监督的特征选择方法。

本文的其余部分组织如下。在第2节中，我们描述了x^2 术语-类别独立性检验和特征选择方法CHI。然后，我们引入一个新的术语类别我们还比较了新的术语-类别依存度测量方法和新的特征选择方法CHIR。我们还比较了新的术语-类别依赖性测量方法与相关系数和GSS系数的关系。在第3节中，我们提出了一种高性能的文本聚类算法TCFS，它可以在不事先知道文档的类信息的情况下采用特征节方法CHIR。在第4节中，我们将CHIR与三种特征选择方法进行了比较，这三种方法都是基于x^2 统计量及其两个变体，在聚类凝聚力方面。并将带有CHIR的TCFS的聚类精度与其他聚类和特征选择算法进行了比较。第五部分是结论。

II. 基于χ^2的特征选择 统计学

A. x^2 术语-类别独立性测试

在文本挖掘和信息检索中，我们经常使用x^2统计量来衡量一个术语和一个特定类别之间的依赖程度。这可以通过比较2-way contingency table中观察到的共现频率和假定它们是独立的时预期的频率来实现。假设一个语料库包含n个标记的文档，它们属于m个类别。在去除停顿词和词干后，从语料库中提取了不同的术语。我们用一个例子来解释x^2 术语-类别独立性测试。

表一
一个2×2术语类别的或然率表

	c	$\neg c$	
w	40	80	1Σ20
$\neg w$ Σ	60	320	380

例1：为了分析术语w和类别c之间的关系，我们创建了一个双向的或然表，如表I所示。行变量，术语，有两个可能的值：w，$\{\neg w\}$。列变量，类别，可以取c，c中的任何一个。在位置(i, j)的每个单元格，其中$i \in \{w, \neg w\}$，w和$j \in \{c, \neg c\}$包含观察到的频率，用$O(i, j)$表示。例如，$O(w, c)$是属于c类别并包含术语w的文件数量，$O(\neg w, c)$是既不属于c也不包含w的文件数量。

对于x^2 术语-类别独立性检验，我们考虑*无效假设*和*备选假设*。无效假设是两个变量，术语和类别，是相互独立的。另一方面，替代假设是这两个变量之间存在着某种依赖关系。为了检验无效假设，我们将观察到的频率与假设无效假设为真的情况下计算的预期频率进行比较。预期频率$E(i, j)$可以计算为。

$$E(i, j) = \frac{\Sigma a \in \{w, \neg w\}. O(a, j) \; \Sigma b \in \{c, \neg c\} O(i, b)}{n} \quad (1)$$

x^2 统计量的定义为。

$$x^2_{w,c} = \sum_{i \in \{w, \neg w\}} \sum_{j \in \{c, \neg c\}} \frac{(O(i, j) - E(i, j))^2}{E(i, j)} \quad (2)$$

方程2可以用概率来解释如下。

$$x^2_{w,c} = \frac{n(p(w, c)p(\neg w, \neg c) - p(w, \neg c)p(\neg w, c))^2}{P(W)P(\neg W)P(\neg C)P(C)} \quad (3)$$

其中，$p(w, c)$表示类别c中的文档包含术语w的概率，$p(w)$表示语料库中的文档包含术语w的概率，$p(c)$表示语料库中的文档属于类别c的概率，以此类推。这些概率是通过计算语料库中术语和类别的出现次数来估计的。

在例1中，我们得到$E(w, c)$=24，$E(w, \neg c)$=96，$E(\neg w, c)$=76，$E(\neg w, \neg c)$=304，以及x^2 =17.61。对于我们的情况，自由度是（2 1）（2 1）=1。查阅x^2 分布的表格，我们得到临界值
$x^2_{0.001}$ = 10.83，置信度为0.1%。由于x^2 0.001 是远小于17.61，我们拒绝无效假设。这可以解释为观察到的频率和预期频率之间的分歧在统计学上是显著的。这意味着，这种分歧非常不可能仅仅是由随机抽样过程引起的。因此，我们认为w和c之间存在某种依赖关系；也就是说，w这个词的分布与c这个类别有关。

如公式2所示，如果观察到的频率和预期频率之间的差异越大，那么x^2 统计量就越大，该词对该类别的信息量就越大。这也是以往大多数关于文本分类特征选择的研究的基本思路。特征选择方法CHI可以描述如下。对于一个有m个类别的语料库，术语w的好坏通常被定义为以下两者之一。

$$x^2_{平均数}(w) = \sum_{j=1}^{m} p(c_j)\chi^2_{w,c_j} \quad (4)$$

$$x^2_{最大}(w) = \max_j \{\chi^2_{w,c_j}\} \quad (5)$$

其中，$p(c_j)$是文件在
类别c_j。那么，术语良好性测量值为
低于某个阈值的词将被从特征空间中删除。换句话说，CHI选择了对类别有强烈依赖性的术语。

B. 新的术语-类别依赖性测量$R_{w,c}$

在我们的研究中，我们发现特征选择方法CHI并没有充分发掘x^2 术语-类别独立性检验所提供的所有信息。我们将用一个例子来指出问题所在，并提出一个新的术语-类别依赖性度量，用$R_{w,c}$表示，以解决这个问题。

表二

另一个2×2术语类别或然率表

	c	$\neg c$	
w'	60	320	380
$\neg w$	40	80	120
$\sum$			
	100	400	500

例2:我们来比较一下表一和表二。使用公式1和2，我们可以发现两个表都产生相同的x^2 统计量
与$\chi^2_{w,c} = x^2_{w',c} = 17.61$.这很有意思，因为这两个项，$w$和$w'$，实际上在$c$中的分布有很大的不同。
和$\neg c$。

从表一中我们可以看到，存在着正的依赖关系
因为40/100=2/5的文件在w和c之间。

c包含w，40/120=1/3的文件包含w
这意味着，w是c类别中的一个典型术语，而
另一方面，如表二所示，不清楚w'和c之间是否存在正相关关系。
因为即使有60/100=3/5的文件在
与此相反，c中的大多数文件都含有w'。因此，我们很难相信w'与c有关。事实上，我们可以说w'与$\neg c$之间存在着负相关关系。

第二个例子表明，只使用x^2 统计量可能会在估计一个词与一个类别的相关程度时产生很多错误。为了解决这个问题，我们将术语w与类别c的相关度的标准定义为：.
术语应该与类别有很强的正相关关系

c.为了评估一个术语和一个类别之间的依赖关系是积极的还是消极的，我们引入了一个新的衡量标准，$R_{w,c}$，定义为。

$$R_{w,c} = \frac{O(w, c)}{E(w, c)} \tag{6}$$

方程6可以用概率来解释如下。

$$R_{w,c} = \frac{p(w, c)p(\neg w, \neg c) - p(w, \neg c)p(\neg w, c)}{P(W)P(C)} + 1 \tag{7}$$

由于$R_{w,c}$是$O(w, c)$和$E(w, c)$之间的比率，如果不存在术语w和类别c之间的依赖关系（即，$x^2_{w,c}$ 是
如果存在正的依赖性，那么观察到的频率应该大于预期的频率，因此$R_{w,c}$应该大于1；如果存在负的依赖性，$R_{w,c}$应该小于1。

从公式2和6，我们可以看到以下关系
$x^2_{w,c}$ 和 $R_{w,c}$之间：$R_{w,c}$离 1 越远，要么是
当时的概率，它对x^2 的贡献越大。
$x^2_{w,c}$ 是整个或然率表的总结，只是告诉大家
在分布中，一个术语和一个类别之间是否存在依赖关系。但它不能说明这种依存关系是正还是负。另一方面，$R_{w,c}$告诉我们依赖性更多的是
准确。然而，我们仍然需要使用$x^2_{w,c}$ 来评估
因为我们的假设检验是基于理论上的χ^2分布。通过结合χ^2和$R_{w,c}$，我们可以更好地提供关于一个术语和一个类别之间的依赖性的信息。

我们估计，只有当χ^2具有统计学意义且$R_{w,c}$大于1时，w项才与c类相关。使用公式7，我们得到表I的$R_{w,c}$=1.67，$R_{w',c}$=0.79的表二。根据我们的标准，术语w对类别c有很强的正向依赖性，与c相关，而术语w'则不相关，这是一个合理的估计。

有人提出了x^2 统计量的两个变体，以不同的方法解决同一问题。Ng等人[14]提出，特征选择方法应该选择属于某一类别的相关文档的术语，并对该类别的成员资格具有指示性。在[14]中提出了一个x^2 统计量的变体，名为*相关系数*，它可以被看作是"单边"的x^2 统计量。一个术语的*相关系数*
w和一个类别c的定义为：。

$$Cw_{,c} = \frac{\sqrt{n(p(w, c)p(\neg w, \neg c) - p(w, \neg c)p(\neg w, c))}}{P(W)P(\neg W)P(\neg C)P(C)} \tag{8}$$

$Cw_{,c}$和χ^2之间的关系 是$C^2_{w,c} = \chi^2_{w,c}$.
的一个简化变体。
x统计量是在[9]中提出的。它以*相关系数为基础*，在[19]中称为*GSS系数*。一个词w和一个类别c的*GSS系数*sx^2 ，定义为。

$$sx^2_{w,c} = p(w, c)p(\neg w, \neg c) - p(w, \neg c)p(\neg w, c) \tag{9}$$

为了强调术语和类别之间的正相关关系，x^2 统计量的这两个变体在方程 3 中保留了x^2 分子中的第二项，而没有保留 2 的幂。本文将基于*相关系数*和 *GSS 系数的*特征选择方法分别用 CC 和 SCHI 来表示。与CHI一样，对于CC和SCHI方法，一个有m个类的语料库中的术语的好坏定义为
作为$Cw_{,c}$和$sx2$的最大值或平均值。
与 $R_{w,c}$，χ^2 统计数字及其两个变体
当一个词在多个类别中均匀分布时，Rw,c对规模小的类别有偏见。由于$R_{w,c}$
是$O_{w,c}$和$E_{w,c}$之间的比率，对于有
不同的尺寸，$R_{w,c}$ 的值是相同的，如果该术语具有
不同类别中的相同分布。下面的例子
详细解释了$R_{w,c}$, χ^2统计量及其两个变体之间的区别。

表三

3个类别的7个文件，5个术语

	c_1	c_2	c_3
w_1	d_7	D_1，D_2，D_3，D_5	
w_2	d_6，d_7	D_1，D_2，D_3，D_5	
w_3	d_6，d_7	D_1，D_2，D_5	
w_4		d_2，d_3	

表四

术语w_2的统计值与类别c_1和c_2的统计值。

	$\chi^2_{w,cj}$	$Cw_{2,cj}$	$sx^2_{w,cj}$	$Rw_{2,cj}$
c_1	0.467	0.683	0.041	1.167
c_2	1.556	1.247	0.082	1.167

例3：让我们考虑一组有七个标签的文件。{d1，d2,...，d7}，分为三类，{c1，c2，c3}，因为：c1 = {d6，d7}，c2 = {d1，d2，d3，d5}，c3 = {d4}。总共有五个不同的条款，{w1，w2,. ..语料库中共有五个不同的词，{w1, w2, ..., w5}，而详情见表三。我们来看看w2这个词在c1和c2类别中的分布情况：c1和c2类别中的所有文件都包含w2这个词。基于此观察，我们可以估计出w_2这个词同样相关。到类别c1和c2。对于术语w2，χ^2的值

统计数字、*相关系数*、*GSS系数*和$R_{w_2,c}$是计算并列于表IV。由于$x^2_{w_2,c_2}$大于$x^2_{w_2,c_1}$，你可能会得到这样的结论：w_2与c_2更相关。比c_1，这一点没有得到我们观察的支持。*相关系数*和*GSS系数*的统计值也是显示相同的趋势。造成这个问题的原因是，这些数值的x^2，C和sx^2，受到类别大小的影响，而c_1的规模是c_2规模的一半。这三个衡量标准对规模较大的类别给予更多的权重，这是不合适的。另一方面，我们的Rw,c不受类别大小的影响，因此R_{w_2,c_1}和R_{w_2,c_2}是一样的（1.167）。基于这个结果，我们可以估计w_2与c_1和c_2同样相关，这一点被观察所证实。这个例子表明，Rw,c比x^2统计量及其两个变体更准确地描述了术语-类别依赖关系。

C. 新的特征选择方法 CHIR

正如我们在第一节中所讨论的，一个合适的特征选择方法可以通过选择有助于将文档区分为不同聚类的词来提高文本聚类的性能。首先，让我们研究一下特征选择方法CHI是否是文本聚类的一个好的候选方法。

回顾一下，特征选择方法CHI使用最大或平均的x^2统计值作为术语好坏的衡量标准，以从特征空间中选择术语。对于章节中的例3 II-B，五个条款的最大和平均x^2统计值通过对语料库中的术语进行排序，见表五。x^2的降序排列$^2_{最大}$值，我们可以得到一个列表为(w5, w2, w1, w3, w4)。如果我们从这个列表中选择前三个词，{w5, w2, w1}将被选中。然而，这种选择对于文本聚类来说并不理想。首先，w_2排名较高，并选择了因为它显示了对c_3的强烈依赖性，但实际上w并没有$^2_3$。根据x^2统计量，w_2和c_3之间的强依赖关系是负依赖关系，这意味着w2与c3不相关。其次，w_4没有被选中，尽管它是区分c_2的一个很好的特征，因为我们在表三中可以看到，w_4出现在c_2的所有文档中。

如果使用每个词的平均x^2统计值进行排名，结果是一样的。由于w_2出现在c_1和c_2两个类别的所有文件中，其贡献率为的大$x^2_{w_2,c_3}$值，w_2有相对较大的平均x^2统计量价值，并且仍然排名靠前。但是，像w_2这样的术语，它是统一分布在许多类别中，并没有携带多少有用的信息来区分这些类别。这种多余的特征不应该被保留在特征空间中。否则，其他特征（如w_4）的区分能力就会受到抑制。这个例子表明，CHI方法可以删除与某一类别相当相关的术语，而保留不相关的和多余的术语。CHI方法没有提供足够详细的关于所选术语和相应类别之间关系的信息。为了解决这个弱点，我们提出了一种新的特征选择方法，名为CHIR。

基于χ^2统计量和Rw,c，我们提出了一个新的定义的术语w在有m个类的语料库中的好坏程度为。

$$rx^2(w) = \sum_{j=1}^{m} p(R_{w,cj}) x^2_{w,cj} \quad 与R_{w,cj} > 1 \qquad (10)$$

其中$p(Rw,c_j)$是χ^2的权重。$_{w,cj}$语料库中，以Rw,cj并定义为。

$$p(Rw,c_j) = \frac{Rw,cj}{\sum_{j=1}^{m} Rw,cj} \quad 与R_{w,cj} > 1 \qquad (11)$$

这个新的术语良好性度量，$rx^2(w)$，是加权和的$x^2_{w,cj}$统计数据，当这些数据之间存在着正的依赖关系时术语w和类别cj的关系，rx（w）值越大，说明该术语越相关。当术语w对类别cj的依赖性为负时，其x^2对rx^2（w）的计算没有贡献。术语与类别的依赖关系^{2}是正还是负，由Rw,c决定。根据Rw,c的定义，当Rw,cj大于1时，w和cj之间的依赖关系是正的；否则依赖关系是负的。

当一个术语w对多个类别有正向依赖关系时，我们认为w与某一类别c之间较强的正向依赖关系应该对w的术语良好性贡献更大，其权重可以用Rw,c来计算。原因是Rw,c能够准确地衡量术语与类别的依赖关系，并且不受类别大小的影响。

例如，在例3中，w_3出现在c_1的所有文件中，以及c_2的四个文件中的三个，这表明w3和c1之间的正相关关系由χ^2所示比w_3和c_2之间的强。因此，在计算$_{w3,c1}$

$_{w3,c1}$

w3的项好性，χ^2应该比$x^2_{w3,c2}$.如果用c_1和c_2的尺寸来计算重量。x的权重$^2_{w3,c1}$将小于$x^2_{w3,c2}$因为c1比c2小。另一方面，当我们用Rw,c来计算权重，$p(_{Rw3,c1})$比$p(_{Rw3,c2})$大，因为$_{Rw3,c1}$比$_{Rw3,c2}$大。另一个例子是例3中的术语w2。正如我们之前讨论的，w_2具有相同的正对c1和c2的依赖性。在计算χ^2的权重时。

$_{w2,cj}$

$R_{w2,c1}$和$R_{w2,c2}$是比类别的大小更好的候选者，因为它们有相同的值。这两个例子表明，通过使用Rw,c来计算权重，$^r x2(w)$更倾向于强正词-类别依赖性。

我们的特征选择方法CHIR使用rx^2（w）来衡量术语的好坏，并确保每个术语的rx^2统计量只代表正的术语-类别依赖性。这种特征选择方法的目标是找到那些具有强烈的对语料库中某些类别的积极依赖性。在其他词，CHIR选择与类别相关的术语。

表五

5个术语的χ^2统计量、Rw_i、c和$r^{x}2$值

	c1		c2		c3				
	$x^2_{w_i,c1}$	$Rw_i,c1$	$x^2_{w_i,c2}$	$Rw_i,c2$	$x^2_{w_i,c3}$	$Rw_i,c3$	$x^2_{最大}$	$x^2_{平均数}$	$r^{x}2$
w1	0.630	0.700	3.733	1.400	2.917	0	3.733	2.730	3.733
w2	0.467	1.167	1.556	1.167	7.000	0	7.000	2.022	1.012
w3	1.120	1.400	0.058	1.050	2.917	0	2.917	0.770	0.665
w4	1.120	0	2.100	1.750	0.467	0	2.100	1.587	2.100
w5	0.467	0	1.556	0	7.000	7.000	7.000	2.022	7.000

并删除不相关的和多余的术语。CHIR选择q术语的步骤如下。

1) 对于语料库中的每个不同的术语，计算其rx^2 通过使用公式10进行统计。
2) 按照术语的rx^2 统计量的降序排列。
3) 从列表中选择*前十名的*术语。

对于例3中的术语w_2，如表五所示，尽管$x^2 = 7$是其x^2 统计量中最大的，但响应的$_{Rw_2,c3} = 0$表明，$_{w2}$对$_{c3}$有负的依赖性。这被$_{w2}$从未出现在$_{c3}$中的事实所证实（见表三）。因此，在我们的CHIR方法中，rx^2 (w_2)（表五中的1.012）是在没有x^2 的贡献下得到的。同样，对于$_{w3}$这个词，$rx^2(_{w3})$为0.665，而其$^{x}2$为2.917。根据术语的rx^2 统计数据进行排序的结果是一新的术语列表变成$(w_5, w_1, w_4, w_3, w_2)$，如果我们选择前三个词，$_{w5}$、$_{w1}$、$_{w4}$将被选中。在表三中，我们可以看到，这三个词与相应的类别，它们比CHI选择的三个术语$_{w5}$、$_{w2}$、$_{w1}$对语料库的信息量更大。这个例子表明，CHIR选择的特征子集比CHI更好。

III. 带有特征选择的文本聚类（TCFS）算法

A. 文本聚类的概述

在大多数现有的文本聚类算法中，文本文档是通过使用向量空间模型来表示的[18]。在这个模型中，每个文档d*被视为*术语空间中的一个向量，由*术语频率*（TF）向量表示。

$$dtf = [tf1, tf2, \ldots, tfh]。 \quad (12)$$

其中tfi是文档中第 i个 术语的频率，h是文本数据库的维度，即唯一术语的总数。通常有几个预处理步骤，包括去除停顿词和词干，对文档进行预处理。这个模型的一个广泛使用的改进是根据每个术语在语料库中的*反文档频率*（IDF）[18]进行加权。为了考虑到不同长度的文档，每个文档向量的长度被归一化为单位长度。在本文的其余部分，我们假设这个由TF-IDF加权的归一化向量空间模型在聚类过程中被用来表示文档。

对于文本文档的聚类问题，有不同的标准函数可用。最常用的是*余弦*函数[18]。*余弦*函数测量两个文档之间的相似性，作为文档之间的关联性代表它们的向量。对于两个文件d_i和d_j，它们之间的相似度可以计算为：。

$$余弦(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| \|d_j\|} \quad (13)$$

其中 • 表示矢量点积和 ‖ ‖ 表示的是向量d_i的长度。当两个文件完全相同时，*余弦*值为1，如果它们之间没有任何共同之处，余弦值为0。*余弦*值越大，说明这两个文件共享的术语越多，越相似。

k-means算法在解决将数据集聚成k个聚类的问题上非常流行。如果数据集包含n个文档，$_{d1}$，$_{d2}$，…，d_n，*那么*聚类就是将它们归入k个聚类的优化过程，从而使全局的标准函数

$$\sum_{j=1}^{k} \sum_{i=1}^{n} f(d_i, cen_j) \quad (14)$$

cen_j代表集群c_j的中心点，j=1，..k，$f(d_i, cen_j)$是一个文档d_i和一个中心点cen_j*的*聚类标准函数。当使用*余弦*函数时，每个文档被分配到具有最相似中心点的聚类中，结果是全局标准函数最大化。这个优化过程被称为一个NP-complete问题[10]，k-means算法被提出来提供一个近似的解决方案[11]。k-means的步骤如下。

1) 选择k个初始集群中心点。
2) 对于整个数据集中的每个文档，计算具有每个聚类中心点的聚类标准函数。将每个文档分配给其最佳选择。(*聚类步骤*)
3) 根据分配给他们的文件，重新计算k个中心点。(*更新步骤*)
4) 重复步骤2和3，直到收敛。

B. 将有监督的特征选择方法应用于文本排序

由于缺乏文档的类标签信息，将有监督的特征选择方法直接应用于文本聚类是具有挑战性的。然而，在文本聚类中采用有监督的特征选择并非不可能，因为在聚类过程中得到的聚类可以为特征选择提供有价值的信息。著名的期望最大化（EM）算法[7]为我们提供了一个将文本聚类和监督特征选择方法相结合的框架。在EM算法的基础上，我们提出了一种新的文本聚类算法，名为文本聚类与

特征选择（TCFS），它交替进行聚类和监督特征选择，直到收敛。

回顾一下，我们将文本聚类的问题定义为将具有相似主题的文档归入一个群组。由于我们使用的是EM算法，我们假设每个文档聚类都有一个高斯分布的术语。也就是说，一个有k个聚类的语料库被认为是k个高斯分布的混合物。考虑到参数和我们的高斯模型，我们对我们的数据集的可能性进行最大化。最大似然表示我们的高斯模型对数据集的拟合程度。在这种情况下，TCFS算法的聚类标准是最大似然，而特征选择的自然标准也是最大似然。

对于文本聚类，似然函数$p(S|\theta)$代表了当高斯模型的参数向量θ给定时，由n个文档组成的一组S被分成k个聚类的概率，它可以写成。

$$p(S|\theta)=\prod_{i=1}^{n}\sum_{j=1}^{k}p(c_j|\theta)p(d_i|c_j,\theta) \quad (15)$$

其中c_j是第j个集群，$p(c_j|\theta)$是第j个集群的先验概率。

在文本聚类的EM框架中，文档中的术语被假定为有条件地相互独立，似然函数可以被重写为。

$$p(S|\theta)=\prod_{i=1}^{n}\sum_{j=1}^{k}\left(p(c_j|\theta)\prod_{w\in d_i}p(w|c_j,\theta)\right) \quad (16)$$

其中$p(w|c_j,\theta)$是术语w在集群c_j中的条件概率。

正如我们在第一节中所讨论的，并不是所有的术语都与集群同样相关，所以$p(w|c_j,\theta)$可以表示为。

$$p(w|c_j,\theta)=p_r(w|\theta)p_r(w|c_j,\theta)+(1-p_r(w|\theta))p_{ir}(w|c_j,\theta) \quad (17)$$

其中，$p_r(w|\theta)$是指在给定的θ条件下，术语w与语料库相关的概率；$p_r(w|c_j,\theta)$是指在给定的θ条件下，当w相关时，术语w在聚类c_j中的概率；$p_{ir}(w|c_j,\theta)$是指当w不相关时，术语w在聚类c_j中的概率。$p_r(w|\theta)$是通过执行特征选择方法确定。当术语w被选中时，w被估计为与语料库相关，$p_r(w|\theta)$被设置为1；否则，$p_r(w|\theta)$被设置为f，其中f是$[0,1]$范围内的一个预定因素。

EM产生一个估计序列$\{\hat{\theta}(i)$和$\hat{p}_r(i), i=0, 1, 2, \ldots\}$通过使用以下两个步骤。

1) 预期步骤（E-步骤）：$\hat{p}_r(i+1)=E(p_r|S,\hat{\theta}(i))$
2) 最大化步骤（M-step）。

$$\hat{\theta}(i+1)=\arg\max_{\theta}p(S|\theta,\hat{p}_r(i))$$

事实上，E步是通过计算当前给出的聚类结果的预期特征相关性来进行监督下的特征选择，M步是在新的特征空间中对数据集进行重新聚类。

由于k-means聚类算法被认为是EM框架对硬阈值情况的扩展[2]，我们可以使用k-means作为我们TCFS算法的基础。在TCFS中，有监督的特征选择方法，如CHIR，被整合到k-means的*更新步骤*中，新的*更新步骤*被认为是TCFS的E-步骤。在每个E步中，当前集群标签被视为类标签，并进行CHIR以估计每个术语与语料库的相关性，然后将术语相关性的概率$p_r(w|\theta)$设置为1或f，其中f的范围在TCFS中为(0, 1)。这意味着，如果一个术语是根据E步骤中获得的信息选择的，那么该术语被估计为与语料库相关，并保留在特征空间中。否则，该词将被估计为不相关，其权重将以f的系数减少；也就是说，其新的权重是通过将以前的权重与f相乘来计算的。在TCFS的M步骤中，如同k-means的*聚类步骤*，语料库中的文档在新的特征空间中被重新聚类。我们的TCFS算法的详细步骤如下。

1) 在数据集上执行聚类算法，如k-means，并获得初始聚类。
2) 执行监督下的特征选择方法，如CHIR，对数据集采用当前的聚类方式结果作为文档的类别标签信息。被选中的特征（即术语）在特征空间中保持不动，但每个未被选中的特征的权重是减去f，其中f是范围内的一个预先确定的系数的（0,1）。在新的特征空间中计算出k个中心点。(E-步骤)
3) 对于语料库中的每个文档，用新特征空间中的每个聚类中心点计算聚类标准函数。将每个文档分配给其最佳选择。(M-步骤)
4) 重复步骤2和3，直到收敛。

在[12]中提出的IF方法解决了不可用的问题。迭代选择特征并进行聚类，从而获得类标签信息。IF也采用了EM框架和k-means算法，但TCFS和IF之间有两个主要区别。首先，它们的M步是不同的。在IF的M步骤中，整个k-means算法被执行，这与特征选择方法无关。在整个k-means算法完成之前，特征空间不会改变。另一方面，在TCFS的M步骤中，只有k-means算法的*聚类步骤*是在新的特征空间中进行的，而这个新的特征空间是在每个E步骤中得到的，它有助于产生准确的最终聚类结果。其次，IF只是根据每个迭代的E-步骤中计算出的相关度分数来删除未选择的特征。另一方面，在TCFS的E步中，我们降低了未选择的特征在特征空间中的权重。监督下的特征选择方法所利用的类标签信息并不是文档的真实（即最终）类标签信息。如果在一个迭代中类标签不正确，一些特征可能会被错误地未被选中。一旦这些未选择的特征在早期阶段被从特征空间中删除，以后就不能再恢复了。由于这个原因，TCFS在每次迭代时不会简单地删除未选择的特征。随着EM迭代的收敛，我们越来越接近真实的类标签信息，最终我们可以选择具有高相关性分数作为所需的特征子集。

IV. 实验结果

在本节中，首先将新的特征选择方法CHIR与其他特征选择方法CHI、CC（基于*相关系数*）和SCHI（基于*GSS系数*）在聚类凝聚力方面进行比较。然后，对

采用不同特征选择方法的文本聚类算法TCFS与k-means、带有术语强度（TS）的k-means和IF方法进行了比较。K-means被用作实验中所有算法的基础。由于k-means的性能对初始中心点的选择很敏感，对于每个测试数据集，我们随机选择15组k初始中心点。
所有的算法都用这些初始中心点集进行了测试。
并使用结果的平均值进行比较。实验结果表明，带有CHIR的TCFS具有最好的聚类精度。

A. 数据集

我们使用了从两种不同类型的文本数据库中提取的五个测试数据集，这些数据库已被研究人员广泛使用
在信息检索领域。两个数据集，用CACM和MED表示，从CACM和MEDLINE中提取。
摘要，分别收录在经典数据库中[4]。另外三个数据集，用EXC、PEO和TOP表示。
来自Reuters-21578 Distribution 1.0的EXCHANGES、PEOPLE和TOPICS类别集[17]。

测试数据集的每个文档都被预先分类为一个独特的类别。但是，这些信息在聚类过程中是隐藏的，只是用来评估每种聚类算法的聚类精度。在实验之前，删除停顿词和干系词的工作是这样进行的
对数据集进行预处理的步骤。表六总结了
我们的实验中使用的所有数据集的特点。

B. 评价方法

*1）特征选择的评估方法。*我们用
聚类的凝聚力来衡量特征的性能。
选择方法。一个聚类的凝聚力值可以通过使用聚类中的文档之间的相似性的加权和来计算，如下所示[20]。

$$凝聚力(c) = \frac{1}{|c|^2} \sum_{d\in c, d'\in c} 余弦(d', d)$$

$$= \frac{1}{|c|}\sum_{d\in c} d - \frac{1}{|c|}\sum_{d'\in c} d', \quad = cen - cen = cen^2 \quad (18)$$

其中c代表集群，cen是集群的中心点。
d和d'是集群中的文件，*余弦*函数
用来衡量文件之间的成对相似性。
从公式18可以看出，中心点向量长度的平方是集群中两个文档的平均配对相似度。这也包括每个文档与自身的相似度，即为1。当一个特征选择方法应用于文本文档时，每个聚类的内聚度值可能会发生变化。一个好的特征选择方法应该消除不相关的特征，同时获得大的聚类内聚度值。

*2）文本聚类的评价方法。*我们使用*F-测量*和*纯度*来评价聚类算法的准确性。

F-measure是信息检索中使用的*精度*和*召回*值的谐波组合[18]。由于我们的数据集是按照第IV-A节描述的方式准备的，因此得到的每个聚类可以被视为查询的结果，而每个预先分类的文件集可以被视为所需的集合
该查询的文件。因此，我们可以计算出精度
$P(i, j)$和召回率$R(i, j)$的每个群组j的每个类别i。

如果n_i是i类成员的数量，n_j是j群组成员的数量，n_{ij}是j群组中i类成员的数量，那么P（i，j）和R（i，j）可以定义为。

$$P(i, j) = \frac{n_{ij}}{n_j} \quad (19)$$

$$R(i, j) = \frac{n_{ij}}{倪} \quad (20)$$

相应的F-measure $F(i, j)$被定义为。

$$F(i, j) = \frac{2 * P(i, j) * R(i, j)}{P(i, j) + R(i, j)} \quad (21)$$

然后，整个聚类结果的F值被定义为。

$$F = \sum_i \frac{n_i}{n} \max_j (F(i, j)) \quad (22)$$

其中n是数据集中的文档总数。一般来说，F值越大，聚类结果就越好[20]。

一个群组的纯度表示该群组中对应于分配给该群组的最大类文件的部分，因此群组j的纯度定义为：。

$$纯度(j) = \frac{1}{n_j} \max_i (n_{ij}) \quad (23)$$

聚类结果的总体纯度是各簇的纯度值的加权和。

$$纯度 = \sum_j \frac{n_j}{n} Purity(j) \quad (24)$$

一般来说，纯度值越大，聚类结果就越好[24]。

C. 特征选择方法的比较

在我们的实验中，特征选择方法CHIR、CHI、CC
和SCHI进行了评估。为了消除以下因素的影响
文本聚类算法的实验，我们运行了四个
对已标记的文本文件进行特征选择的方法。每一类标记的文档都被当作一个聚类，比较特征选择后每一类的内聚度值。

对于CHI、CC和SCHI，我们使用$x^2_{最大}(\omega)$、$Cmax(\omega)$和$sx^2_{最大}(\omega)$分别作为术语良好性的衡量标准，因为
据报道，x^2 和 sx^2 统计量的最大值比其平均值要好[9], [23].

在我们的实验中，所选特征（即术语）的百分比从5%到90%不等。在每一轮的特征选择中，未被选中的术语被简单地从特征空间中删除，然后文档向量被重新规范化。为了进行比较，计算并记录每个类别的内聚性值。

我们在CACM数据集的43个类上评估了四种特征选择方法（CHIR、CHI、CC和SCHI），其特征选择的比例不同。比较了各类的内聚力值，部分结果见图1和图2。随着特征空间中剩下的术语越来越少，聚类的内聚力值就会增加，因为稀疏的特征被移除，文档之间变得更加相似。当特征选择方法选择了一个合适的特征

表六 数据集摘要

数据集	数值。的文件。	数值。班级的	最小。班级规模	最大。班级规模	数值。独特术语的比例	平均。文件。长度	国家认可。成对相似度按余弦值计算
CACM	842	43	11	51	3,225	59	0.03
医学	287	9	26	39	4,255	77	0.02
预算外资金	334	7	28	97	3,258	67	0.03
PEO	694	15	11	143	5,046	102	0.04
顶端	2279	7	23	750	10,719	113	0.03

子集，该子集比其他子集更能代表聚类，凝聚度值更大。例如，对于CACM数据集的c_1 类，当用20%的特征进行CHIR时，该类的内聚度值为0.24。当进行CHI、CC和SCHI时，c_1 的内聚度值分别为0.201、0.225和0.172。这一结果表明，当选择20%的特征时，CHIR去除不相关的特征比CHI、CC和SCHI更好。

我们的实验结果表明，CHIR在提高集群的内聚性方面一直优于其他三种方法。CC和CHI的性能

在某些情况下非常接近，这告诉我们，仅仅识别正的术语-类别依赖关系是不够的。术语好坏的衡量标准也应该准确描述依赖关系。SCHI在大多数情况下表现不佳，因为sx^2 的定义（见等式9）没有包含x^2 术语-类别独立性检验的足够信息。

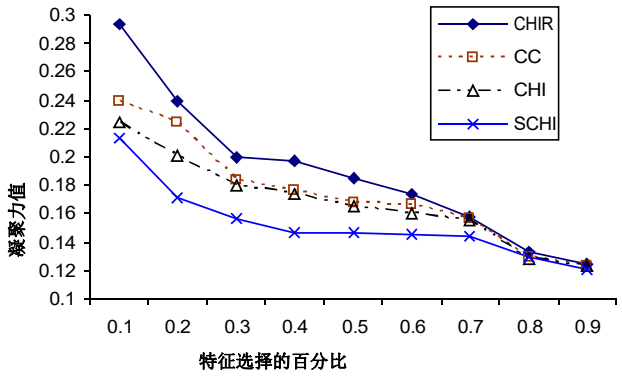


图1. CACM数据集的c类的凝聚力值 $_1$

D. 文本聚类算法的比较

在文献[12]中，通过进行实验，对有监督和无监督的特征选择方法在提高聚类性能方面进行了评估，在这种情况下，文档的类标签可用于特征选择。作为文本聚类的预处理步骤，术语强度（TS）特征选择方法被报告为[12]中评估的无监督特征选择方法中最好的。

TS最初是针对词汇提出和评估的。在文本检索中的减少[21]，后来只在极其积极的特征选择时才应用于文本 分类。观念[22]。它是根据以下的条件概率来计算的，即

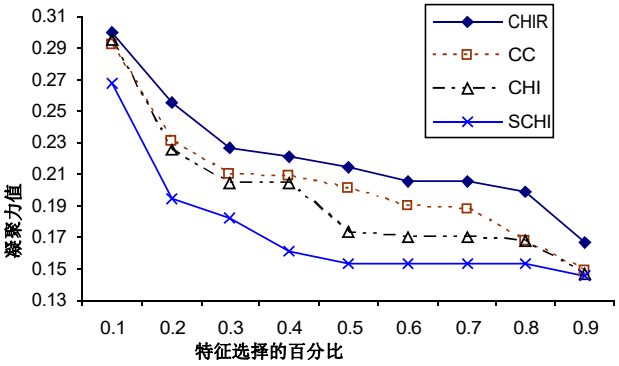


图2. CACM数据集的c类的凝聚力值 $_2$

一个术语出现在一对相关文件的第二个文件中，因为它出现在第一个文件中。

$$TS(w) = \mathrm{p}\,(w \in_{dj} \mid w \in_{di}) \text{，其中}$$

$$similarity(d_i, d_j) \geq \delta \tag{25}$$

$$\approx \frac{\#\text{ 两个文件中都出现}w\text{的对子}}{W\text{在第一个文件中出现的对子数量}} \tag{26}$$

其中δ是用于确定相关文档对的参数。由于我们需要计算每个文档对的相似度，计算TS的时间复杂性是文档数量的二次方。由于不需要类标签信息，TS可以用于文本聚类中的术语减少。在这种情况下，术语按其TS值的降序排列，然后从上面选择一定比例的术语用于聚类。

我们在本节的实验有两个部分。首先，我们比较了k-means、带TS的k-means和TCFS与三种不同特征选择方法的聚类精度。当k-means与TS相结合时，δ被设置为0.1%，首先进行特征选择作为预处理步骤，然后将k-means应用于新特征空间的数据集。所有特征选择方法的特征选择比例都在[5%，90%]范围内变化。

当我们对数据集进行TCFS时，在每个迭代中，根据所选择的监督特征选择方法--CHIR、CHI或CC，选择一定比例的特征。我们没有用SCHI测试TCFS，因为据报道

在[9]中，$sx^2_{最大}(w)$ 提高文本的性能

适用。对于CHI和CC，$x^2_{最大}(w)$和$C_{max}(w)$被选为

分别是术语的好坏度。如第三节所述，每个术语与聚类的相关度是根据每次迭代获得的信息来估计的。术语相关性的概率被设定为1或*f*，其中*f*是一个在（0,1）范围内的预定因子。在每次迭代中，每个不相关的术语在特征空间中的权重都被*f*减少。在我们的实验中，我们将*f*设置为0.5。

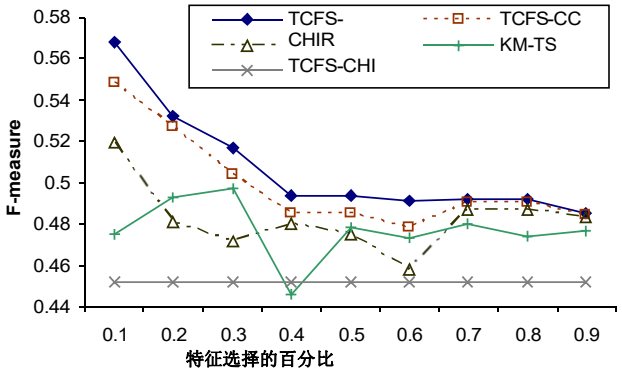


图3. EXC数据集的聚类的F-measure

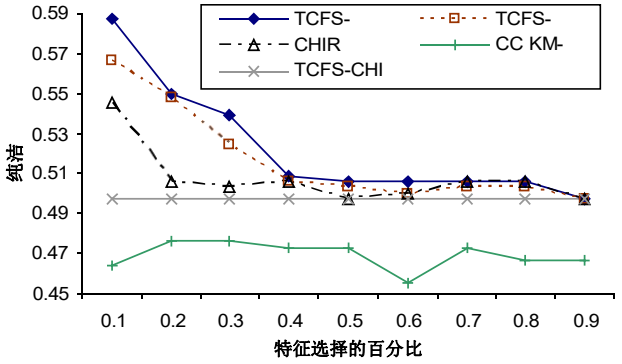


图4. EXC数据集的簇的纯度值

图3和图4显示了在EXC数据集上运行五种聚类算法的结果。表七和表八显示了在选择25%的术语时，在不同的数据集上运行五种聚类算法得到的聚类的F值和纯度值。从实验结果中可以得出一些结论。

- 特征选择方法可以提高文本聚类的性能，因为更多的不相关或多余的术语被删除。
- TCFS与监督下的特征选择方法，如CHIR、CHI或CC，可以达到比K-手段与TS更好的F值。结果表明，根据聚类过程中获得的聚类信息进行超视距的特征选择方法可以提高聚类的准确性。
- 随着特征选择比例的变化，带TS的k-means的表现并没有持续优于k-means。例如，当EXC数据集中40%的术语被TS选择时，F值为0.446，甚至低于单纯执行k-means的情况。在EXC数据集上用TS执行k-means得到的聚类结果的纯度值一直低于的那些执行K-means。这一结果表明，TS并不总是选择一个合适的特征子集。在某些情况下，TS删除了一些相关的词，而保留了一些不相关的词。
- 对于不同的测试数据集，带有CHIR的TCFS在F-measure和纯度值方面优于其他所有聚类算法。

其次，我们比较了TCFS和IF方法。对于这两种算法，我们应用CHIR和CHI作为监督下的特征选择方法。图5和图6显示了通过运行TCFS和IF方法得到的EXC和PEO数据集的聚类的F-measure。由于图幅有限，我们只显示了[50%，90%]范围内的特征选择百分比。TCFS和IF方法的整个过程都涉及到k-means算法的两轮完整的迭代，所以它们之间具有可比性。实验结果表明，在大多数情况下，TCFS的性能要优于IF（k-means的一次迭代）。原因是IF与k-means分别进行了特征选择。

IF方法的主要弱点是：它并不总是随着更多的迭代而提高聚类性能。例如，当PEO数据集的70%的术语在每次迭代中被选中时，聚类结果的F值会从
0.597到0.575，再进行两次迭代（见图6）。由于监督下的特征选择所使用的聚类标签并不是真正的类标签，简单地从特征空间中删除未选择的特征是不可取的。在IF中，一旦一个特征未被选中，它就被删除了，所以以后就没有机会恢复它。在我们的TCFS算法中，我们保留未选择的特征，但减少它们在特征空间的权重。这种方法比较安全，因为在早期阶段的错误可以在以后得到纠正，因为未被选中的特征仍然保留在特征空间中。这有助于在算法收敛的过程中获得更好的聚类结果。

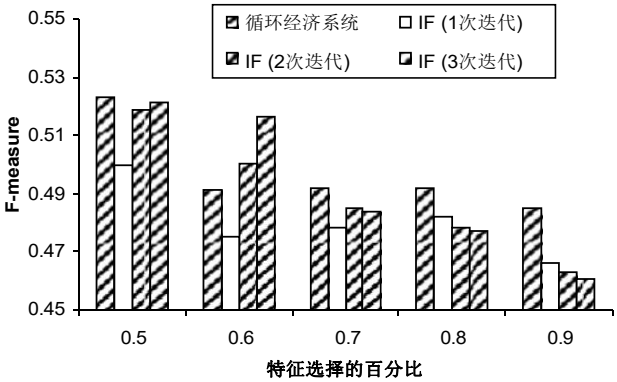


图5.EXC数据集的聚类的F值（TCFS和IF采用CHIR）。

V. 结论

在本文中，我们引入了一个新的术语类别依赖度量，用$Rw_{,c}$表示，它可以告诉人们依赖性是正还是负，从而更准确地描述依赖性。基于χ^2统计量和$Rw_{,c}$，我们提出了一种新的有监督的特征选择方法CHIR。CHIR选择的是
通过利用已知的类别标签信息，对与类别相关的术语进行分类。CHIR可用于文本分类。

表七

集群的f-measure（25%的特征选择）。

数据设置	KM	KM与TS	设有TCFS智	设有TCFS CC	设有TCFS CHIR
CACM	0.429	0.428	0.478	0.478	0.481
医学	0.569	0.702	0.737	0.747	0.753
预算外资金	0.452	0.495	0.477	0.506	0.522
PEO	0.582	0.604	0.607	0.608	0.613
顶端	0.621	0.561	0.640	0.642	0.667

表八

集群的纯度值（25%的特征选择）。

数据设置	KM	KM与TS	设有TCFS智	设有TCFS CC	设有TCFS CHIR
CACM	0.532	0.524	0.597	0.601	0.607
医学	0.606	0.672	0.746	0.750	0.756
预算外资金	0.497	0.476	0.506	0.529	0.542
PEO	0.586	0.608	0.600	0.600	0.611
顶端	0.775	0.709	0.786	0.787	0.790

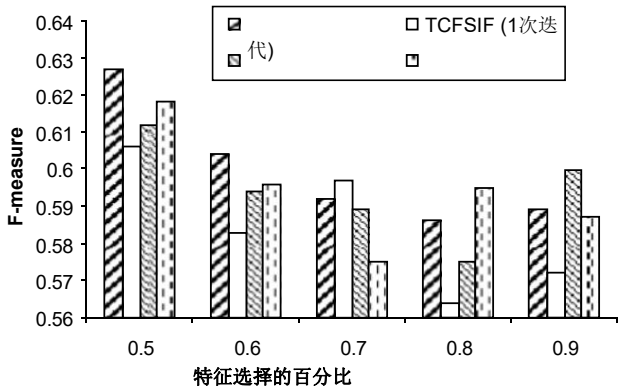


图6.PEO数据集的聚类的F值（TCFS和IF采用CHI）。

文本总结和本体创建。我们使用真实数据集的实验结果表明，与现有的基于x^2 统计量、*相关系数*和*GSS系数的*特征选择方法相比，CHIR产生的聚类更具凝聚力。

我们还提出了一种新的文本聚类算法TCFS，该算法在聚类过程中进行了监督下的特征选择。在聚类过程中获得的聚类标签信息被用作特征选择的已知类标签信息。被选择的特征可以反复提高聚类的质量，随着聚类过程的收敛，聚类结果具有更高的准确性。

带CHIR的TCFS与其他聚类和特征选择算法进行了比较，如k-means、带术语强度（TS）特征选择方法的k-means、IF方法，以及带其他特征选择方法的TCFS。我们的实验结果表明，对于不同的测试数据集，采用CHIR的TCFS在聚类精度方面比其他算法有更好的表现。

鸣谢

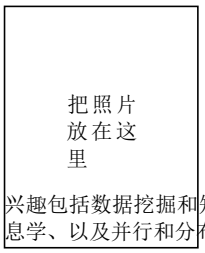
这项研究得到了AFRL/莱特兄弟研究所（WBI）的部分支持。

参考文献

[1] C.C. Aggrawal和P. S. Yu, "Finding Generalized Projected Clusters in High Dimensional Spaces," Proc. of ACM SIGMOD Int'l Conf. on Management of Data, pp.70-81, 2000.
[2] L.Bottou和Y. Bengio, "K-means算法的收敛特性"，《神经信息处理系统进展》7，第585-592页，1994。
[3] C.Buckley和A. F. Lewit, "Optimizations of Inverted Vector Searches," Proc. of Annual ACM SIGIR Conf. on Research and De-velopment in Information Retrieval, pp.
[4] 经典数据集，可在ftp://ftp.cs.cornell.edu/pub/smart/。
[5] M.Dash and H. Liu, "Feature Selection for Classification," Intelligent Data Analysis, vol. 1, no.3, pp. 131-156, 1997.
[6] M.Dash and H. Liu, "Feature Selection for Clustering," Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD), pp
[7] A.P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," Journal of the Royal Statistical Society, vol. 39, no. 1, pp.1-38, 1977.
[8] G. Forman, "特征选择。我们几乎没有触及表面"，IEEE智能系统，2005年11月。
[9] L.Galavotti, F. Sebastiani, and M. Simi, "Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization," Proc. of the 4th European Conf. on Research and Advanced Technology for Digital Libraries, pp.59-68, 2000.
[10] M.R. Garey, D. S. Johnson, and H. S. Witsenhausen, "Complexity of the Generalized Lloyd-Max Problem," IEEE Trans. on Information Theory, vol. 28, no. 2, pp.
[11] J.A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, 1975.
[12] T.Liu, S. Liu, Z. Chen, and W. Ma, "An Evaluation on Feature Selection for Text Clustering, " Proc. of Int'l Conf. on Machine Learning, 2003.
[13] C.Manning和H. Schutze，《*统计自然语言处理的基础*》，麻省理工学院出版社，1999。
[14] H.T. Ng, W. B. Goh, and K. L. Low, "Feature Selection, Perception Learning, and a Usability Case Study for Text Categorization," Proc. of the 20th ACM Int'l Conf. on Research and Development in Information Retrieval, pp.67-73, 1997.
[15] G. H. John, R. Kohavi, and K. Pfleger, "Irelevant Features and the Subset Selection Problem," Proc. of Int'l Conf. on Machine Learning, pp.

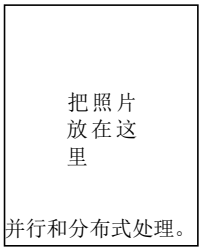
121-129, 1994.

[16] J.R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, pp. 81-106, 1986.
[17] Reuters-21578 Distribution 1.0，可在 http://www.daviddlewis.com/resources/testcollections/reuters21578。
[18] C.J. van Rijsbergen, *Information Retrieval*, 2nd edition, Butterworth, London, 1979.
[19] F.Sebastiani，"自动文本分类中的机器学习"，ACM计算调查，第34卷，第1期，第1-47页，2002年。
[20] M.Steinbach, G. Karypis, and V. Kumar, "A Comparison of Document Clustering Techniques," Proc.KDD文本挖掘研讨会，2000。
[21] W.W. J. Wilbur和K. Sirotkin，"自动识别停顿词"，《信息科学杂志》，第18卷，第1期，第45-55页，1992。
[22] Y.Yang, "Noise Reduction in a Statistical Approach to Text Categorization," Proc. of Annual ACM SIGIR Conf. on Research and Development in Information Retrieval, pp.256-263, 1995.
[23] Y.Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," Proc. of Int'l Conf. on Machine Learning, pp.412-420, 1997.
[24] Y.Zhao和G. Karypis，"选定的文档聚类标准函数的经验和理论比较"，*机器学习*，第55卷。3, pp. 311-331, 2004.



李彦君于1993年在中国北京对外经济贸易大学获得经济学学士学位，2001年在俄亥俄州哥伦布市富兰克林大 学 获得计算机科学学士学位，2003年和2007年分 别 在俄亥俄州代顿市赖特州立大学获得计算机科学硕士学位和计算机科学与工程博士学位。她目前是纽约布朗克斯的福特汉姆大学计算机和信息科学系的助理教授。她的研究兴趣包括数据挖掘和知识发现、文本挖掘、本体论、信息检索、生物信息学、以及并行和分布式计算。

罗聪南1997年获得清华大学计算机科学学士学位，2000年获得中国科学院北京软件研究所计算机科学硕士学位，2006年获得俄亥俄州代顿市莱特州立大学计算机科学与工程博士学位。目前，他是位于加州圣地亚哥的Teradata公司的技术人员，其研究兴趣包括数据挖掘、机器学习和数据库。



Soon M. Chung于1979年在韩国首尔国立大学获得电子工程学士学位，1981年在韩国科学和技术高级研究所获得电子工程硕士学位，1990年在纽约雪城大学获得计算机工程博士学位。他目前是 俄亥俄州代顿市赖特州立大学计算机科学和工程系的教授。他的研究兴趣包括数据库、数据挖掘、网格计算、文本挖掘、XML、以及其他方面。

并行和分布式处理。

模式识别 117 (2021) 107996

内容列表可在ScienceDirect上找到

模式识别

杂志主页: www.elsevier.com/locate/patcog

健全的深度*K-means*。一种有效而简单的数据聚类方法[6]

黄树东[a] , 赵康[b] , 徐增林[c,b,d] , 刘全辉[a,*]

[a]*四川大学计算机学院，成都610065，中国*
[b]*中国电子科技大学计算机科学与工程学院，成都 611731*
[c]*哈尔滨工业大学计算机科学与技术学院，深圳518055，中国*
[d]*人工智能中心，鹏程实验室，深圳518055，中国*

ARTICLE INFO

文章的历史。
2020年10月13日收到的
2021年3月10日修订
2021年4月18日接受
可于2021年4月28日上网

关键词。
k-means算法 鲁棒性
聚类 深度学习

ABSTRACT

聚类的目的是根据一些距离或相似度的测量，将输入数据集划分为不同的组。现在最广泛使用的聚类方法之一是*k-means*算法，因为它的简单性和效率。在过去的几十年里，*k-means*及其各种扩展已经被制定用来解决实际的聚类问题。然而，现有的聚类方法都是以单层公式（即浅层公式）提出的。因此，所获得的低层表示和原始输入数据之间的映射可能包含相当复杂的层次信息。为了克服低层次特征的缺点，我们采用了深度学习技术来提取深层表征并提高聚类性能。在本文中，我们提出了一个深度*K*-means模型来学习与不同的隐性低层属性相关的隐性表征。通过使用深层结构来分层执行*k-means*，数据的层次语义可以以分层的方式被利用。来自同一类别的数据样本被迫逐层靠近，这对聚类任务是有利的。我们的模型的目标函数被导出到一个更可追踪的形式，这样可以更容易地解决优化问题，并获得最终的稳健的再结果。在12个基准数据集上的实验结果证明，与传统的和最先进的方法相比，所提出的模型在聚类性能上取得了突破性进展。

1. 简介

聚类的目标是将数据集划分为同质群体，这无疑是统计学和机器学习中最基本的技术之一[1,2]。Cus- tering已经被发现有惊人的表现，特别是在无监督的情况下[3]。无数的应用可以被模拟成一个聚类问题，文本挖掘[4]，语音识别[5]，图像分割[6]，仅此而已。近年来，基于不同的方法论和统计理论[7]研究了许多聚类方法，如*k-means*聚类[8]、谱聚类[9]、基于非负矩阵因子的聚类[10]、信息论聚类[11,12]、多视图聚类[13,14]等。其中，K-自1967年提出以来，由于其有效性和简单性，成功地吸引了广泛的关注[8]。更重要的是，由于在各种实际问题中对使用和聚类性能的贡献，它已被广泛认可为十大数据挖掘算法之一[15]。

机器学习的最新发展表明，人们可以用深度学习技术更有效地处理爆炸性增长的数据，特别是在无监督的情况下。例如，深度神经网络已被普遍用于聚类任务[16,17]。Ji等人[16]通过在编码器和解码器之间引入一个自我表达层，提出了一个用于子空间聚类的深度神经网络架构。Zhou等人[17]通过利用对抗性学习，进一步将这一架构扩展到深度对抗性子空间聚类模型。 也就是说，一个子空间聚类发生器被用来学习采样表征，而一个质量验证判别器被用来通过估计重新采样的数据是否具有一致的属性来评估当前的聚类性能。Guo等人[18]提出了联合优化聚类标签分配和学习表征的建议。

[6]本文是[1]的扩展版本，已被接受在国际高级应用神经计算会议（NCAA-2020）上发表。

* 通讯作者。
*电子邮件地址：*quanhuiliu@scu.edu.cn (Q. Liu).

https://doi.org/10.1016/j.patcog.2021.107996
0031-3203/

通过利用局部结构保护和应用不完全自动编码器来实现集群。Dizaji等人[19]定义了一个使用KL发散最小化的聚类模型，它可以将原始数据映射到一个判别性的子空间并预测聚类的分配。Peng等人[20]介绍了一种用于子空间聚类的结构化自动编码器，通过最小化重建误差和纳入先验结构化信息，可以保留局部和全局子空间结构。Peng等人[21]在假设不同的距离度量会导致流形上类似的聚类分配的基础上，提出了一个共同不变性。基于这种共同不变性，通过最小化每个数据点的成对样本分配之间的差异来设计一种深度聚类方法。Zhang等人[22]提出了一个端到端的自监督卷积聚类网络，它可以将卷积网络模块、自表达模块和光谱聚类模块完成一个联合优化框架。然而，这些方法的训练过程通常是耗时的，而且不稳定。此外，有无数的参数需要调整，这使得它们在无监督的任务中不实用。更重要的是，这些方法需要大量的数据来训练，这又需要大量的计算能力。将深度学习架构和经典的聚类模型结合到一个统一的框架中，为聚类任务提供了一个更好的潜在解决方案。

在过去的几十年里，经典*K*-means的众多变种都是以其为基础的。已经研究了如何提高聚类性能的方法。Ding和He[23]证明了主成分本质上超越了连续解，这可以被看作是*k*-means聚类的离散指标。Buchta等人[24]采用共正弦相似性来进行基于原型的分区，在此基础上提出了一种用于文本聚类的球形k-*means*算法。Khanmohammadi等人[25]试图通过结合重叠的k-手段和k-*谐波*手段算法来克服模型对初始聚类中心点的敏感性。它使用k-谐波手段方法的输出来初始化重叠*k*-means方法的聚类中心。通过将种子点定位在数据集的密集区域，Kumar和Reddy[26]提高了*k*-means过滤方法的性能，并很好地分离了数据点。与其他方法不同的是，密集区是通过在*kd树中*表示数据点来确定的。考虑到当许多特征不相关时，排除法缺乏统计学上的保证，Chakraborty等人[27]通过应用熵正则化来学习特征相关性，同时进行an- nealing来解决这个问题。该模型的收敛性和一致性得到了保证，并提出了一个可扩展的mainization-minimization算法来优化该模型。这个模型比*k-means*有明显的改进，但保持了相同的计算复杂性。Capó等人[28]打算通过引入k-means问题的有效近似来解决海量数据的瓶颈问题。该方法将数据集划分为若干个子集，每个子集一般都有代表和权重的特征。然后，*K-means*算法在以下情况下执行

这样的局部表示，减少了计算距离的数量。

尽管前述各项工作取得了显著进展

k-means方法，这些方法通常是以单层的形式设计的。因此，所得到的低维表征和原始输入数据之间的映射可能包含相当复杂的层次信息。考虑到深度学习的发展迫使人们采用多个处理层来提取数据的层次信息[29]，本文提出了一个新的稳健的深度*k*-means模型，以利用多个层次属性 的层次信息。我们模型的整体框架如图1所示。如

我们可以看到，通过使用深层结构来分层次地执行

k-means，数据的分层语义可以以分层的方式被利用。也就是说，来自同一类别的数据样本被逐层靠近收集，这对聚类任务非常有利。

这项工作的主要贡献在于三个方面。

- 我们提出了一个新的稳健的深度模型来执行*k-means*的hierarchically，因此数据的分层语义可以以分层的方式进行探索。因此，来自同一类别的数据样本被有效地逐层聚集，从而提供了一个清晰的聚类结构。
- 为了解决我们模型的优化问题，相关的目标函数被推导到一个更可跟踪的形式，并提出了一个替代的更新算法来解决优化问题。
- 在12个基准数据集上进行的实验显示，与经典和先进的方法相比，结果很有希望。

本文的基础工作构思如下。我们在第2节中简要介绍了密切相关的工作。第3节给出了我们模型的细节。实验结果将在第4节中描述。最后，我们在第5节中给出本文的结论。

这项工作与我们早期的论文[1]的区别有三点。首先，我们提出了一个稳健的深度*K*-Kmeans模型的一般形式。具体来说，我们采用了一系列发散函数来测量重建误差（第3 节），而不是仅仅采用对噪声数据和异常值敏感的Frobenius准则。因此，我们先前的工作[1]只是本文的一个特例。第二，对12个基准数据集进行了更全面的实验，证实了我们模型的稳健性和有效性：（1）记录了更多数据集和高级基线的聚类结果（第4.3节）；（2）增加了不同参数设置的实验结果（第4.4节）；（3）展示了收敛分析的实验（第4.5节）；（4）研究了不同分歧函数对聚类性能的影响（第4.6节）。第三，我们介绍了更多密切相关的文献（在第1节和第2节），澄清了与先进技术的联系和区别。这有助于在社区中更好地定位拟议的工作。

2. 预备工作

非负矩阵分解（NMF）由于其直观的基于部分的解释，在数据聚类中受到了很多关注[30,31]。以前的研究表明，NMF在本质上等同于具有宽松条件的*k*-means[30]。这里我们先介绍一下NMF[32]。假设$\mathbf{X}=[x_1,x_2,\ldots,x_n]\in R^{m\times n}$是一个非负数据矩阵，有$n$个数据样本和$m$个特征。NMF的目的是找到两个非负矩阵$\mathbf{U}\in R^{m\times c}$和$\mathbf{V}\in R^{n\times c}$，使得$\mathbf{X}\approx\mathbf{U}\mathbf{V}^T$，NMF的一般形式为

$$D_\beta(\mathbf{X}|\mathbf{U}\mathbf{V}^T)=\sum_{ij}d_\beta(\mathbf{X}_{ij}|(\mathbf{U}\mathbf{V}^T)_{ij}) \quad (1)$$

$$\text{s.t. } \mathbf{U}\geq 0,\mathbf{V}\geq 0。$$

其中，$D_\beta(\mathbf{X}|\mathbf{X})$表示标量成本函数（即衡量$\mathbf{X}$与其重构的$\mathbf{X}$之间的分歧），$x_{ij}$是$\mathbf{X}$的第$ij$个元素。 在公式（1）中，分歧函数家族可以采用称为β- 发 散 [33]。有三种发散函数在NMF中使用最为广泛。

* β=2（欧氏距离）：$d_2(a|b)=(a-b)^2$
* β = 1（库尔贝克-莱布勒分歧）：$d_1(a|b)=a\log\frac{a}{b}-a+b$
* β = 0（板仓-斋藤分歧）：$d(a|b)=\frac{a}{b}-\log\frac{a}{b}-1$

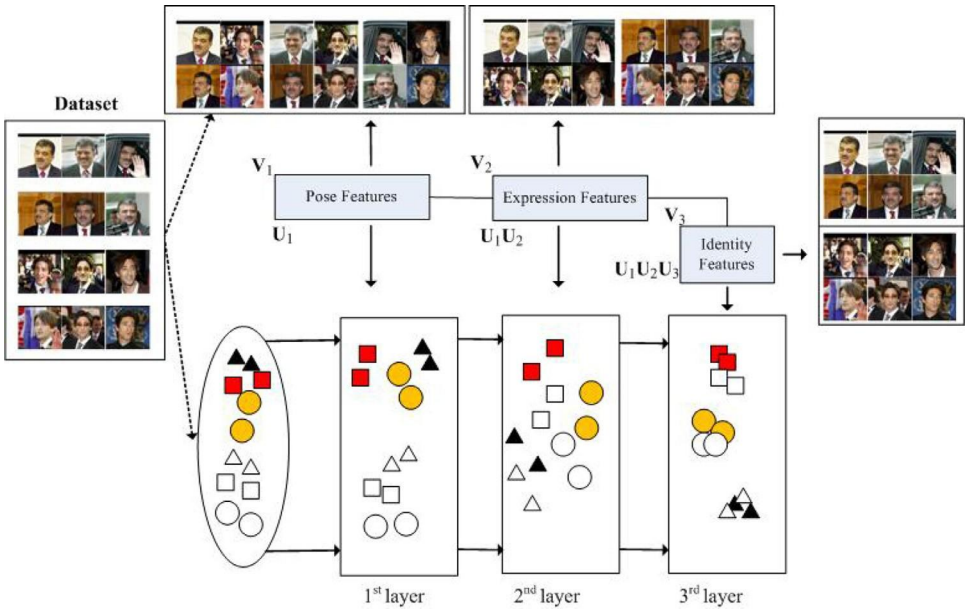


图1. 一个三层的深度*K*-means结构被用来描述我们模型的框架。相同的形状表示数据样本属于同一类别。很明显，人脸数据的变异性可以根据主体的面部表情（有无微笑）或姿势（眼睛在左边、右边或前面）等属性来区分。拟议的深度模型着重于逐层利用层次信息。因此，可以预期会有一个更具判别力的表述。

[32]在公式（1）中采用了欧氏距离，即：。

$$JNMF = \sum_{i=1}^{m} \sum_{j=1}^{n} \left\| X_{ij} - (UV^T)_{ij} \right\|_2^2 = \|X - UV\|_F^2 \quad (2)$$

$$\text{s.t. } U \geq 0, V \geq 0。$$

其中 $\|\cdot\|_F$ 表示Frobenius规范。[32]进一步指出，公式（2）是一个双凸公式（仅在**U**或**V**中凸），并通过应用以下更新规则寻找局部最小值。

$$U_{ij} \leftarrow U_{ij} \frac{(XV)_{ij}}{(UV^TV)_{ij}}, \qquad V_{ij} \leftarrow V_{ij} \frac{(X^TU)_{ij}}{(VU^TU)_{ij}}$$

其中，**V**可以被看作是聚类指标矩阵[30]，**U**代表中心点矩阵，c表示聚类的数量。通常，我们有$c) n$和$c) m$，这意味着公式（2）实际上是在寻找**X**的低维表示**V**。

但在现实中，数据集往往是复杂的，而且总是包含多种层次化的模态（即因素）。以人脸数据集为例，它通常由一些常见的模态组成，如表情、姿势、场景等等。因此，很明显，基于单层的NMF不能完全利用不同因素的隐藏信息。为了填补这一空白，[34]研究了一个多层深度模型，该模型通过分层进行半NMF，创新性地挖掘了数据的层次信息。深度半NMF的基本表述是
NMF模型被定义为

$$\begin{aligned} \mathbf{X} &\approx \mathbf{U}\,\mathbf{V}_1^T, \\ \mathbf{X} &\approx \mathbf{U}\,\mathbf{U}\,\mathbf{V}_{12}^T, \\ &\cdots \\ \mathbf{X} &\approx \mathbf{u}\,\mathbf{u}_{12}\ldots\mathbf{u}\,\mathbf{v}_r^T, \end{aligned} \quad (3)$$

其中，r是层数，U_i和V_i分别是第 i层的基础矩阵和重现矩阵。很明显，深层半NMF的目标也是搜索低维的嵌入。
鼎立表示，即最后一层$\mathbf{V}_r$。通过分层解读
假设每层$V_i(i < r)$，公式(3)能够发现潜在的
层次。与现有的单层NMF模型相比，深度
半NMF可以更好地揭示数据的层次性，因为不同的模式可以被识别出来。
通过不同层的低维表征。因此，我们的模型可以完全实现适合于不同模态的次序聚类的表征。例如，由于

如图1所示，$\mathbf{U}_3$ 对应于表 达 的特征，$\mathbf{U}\,\mathbf{U}_{23}$ 对应于姿势的特征，最后，$\mathbf{U} = \mathbf{U}_{1}\mathbf{U}_{2}\mathbf{U}_{3}$对应于人脸图像的身份映射。这样，一个更好的高可识别性的最终层表示，根据具有最低变异性的特征来进行聚类。
它可以获得。

3.拟议的模型

在这一节中，我们提出了一个新的深度K-*means*模型，称为鲁棒深度*K-means*（RDKM）。我们介绍了一种有效的更新算法来解决相应的优化问题。我们还分析了拟议算法的收敛性。

3.1. 健全的深度K-means

为了探索不同模态的低维表征，本文通过利用深层结构进行k-*means*分层，研究了一种新型的鲁棒性深层k-*means*模型。在本文中，为了扩大我们模型的适用范围（即同时处理负数和非负数数据），省略了U_i上的非负数约束。考虑到非负的
V_i的重力约束使优化问题变得困难。
为了解决这个问题，我们通过引入新的变量V_i^+，将目标函数转化为更能跟踪的形式。通过这种方式，非负引力约束被分离出来，并以等价的方式被采用，其中的
拘束$V_i = V_i^+$。因此，我们不仅扩展了应用，而且还保留了我们模型的强大可解释性。这里我们使用
交替乘法（ADMM）[35]来解决优化问题。在数学上，拟议的RDMK模型被表述为

$$\begin{aligned} &J = D_\beta(\mathbf{X}|\mathbf{Y}) \\ &\text{s.t. } \mathbf{Y} = \mathbf{U}\,\mathbf{U}_{12}\ldots\mathbf{U}\,\mathbf{V}_r^T, \quad (V)_r = \{0,1\}^{r\times c}, \quad (V_r)_{c=1} = 1_c, \\ &\mathbf{V}_i = V_i^+ - V_i^+, V_i^+ \geq 0, i \in [1,\ldots,r-1]。 \end{aligned} \quad (4)$$

在公式（4）中，我们可以看到，在$\mathbf{V}_r$的每一行都采用了1-of-C编码方案。1-of-C编码方案的主要目的是保证$\mathbf{V}_r$的唯一性。此外，基于$\mathbf{V}_r$，我们可以直接获得最终的离散分区结果，而不需要任何后处理。

与公式（2）类似，如果欧氏距离（即β=2）为在公式（4）中采用，我们有

$$J = \|\mathbf{X} - \mathbf{Y}\|_F^2$$

$$\text{s.t.}\ \mathbf{Y} = \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T,\ \mathbf{V}_r \in \{0,1\}^{n\times c},\ \sum_{c=1}^{c}(\mathbf{V}_r)_{ic} = 1, \tag{5}$$

$$\mathbf{V}_i = [\mathbf{V}_i, \mathbf{V}_i] \geq 0, i \in [1,\ldots,r-1].$$

然而，事实证明，Frobenius规范对噪声数据和异常值很敏感[36,37]。为了提高所提出的模型的稳健性，引入了稀疏性规范（即$l_{2,1}$-norm），即

在我们的模型中采用，根据[36]，$l_{2,1}$-norm能够重新[illegible]，在数据点之间进行l_1-norm，因此可以减少异常值的影响。最后，我们的鲁棒深层k-means（RDKM）模型被写为

$$\text{s.t.}\ \mathbf{Y} = \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T,\ \mathbf{V}_r \in \{0,1\}^{n\times c},\ \sum_{c=1}^{c}(\mathbf{V}_r)_{ic} = 1, \tag{6}$$

$$\mathbf{V}_i = [\mathbf{V}_i, \mathbf{V}_i] \geq 0, i \in [1,\ldots,r-1].$$

正如我们所看到的，$\|\mathbf{X} - \mathbf{Y}\|_{2,1}$对于$\mathbf{Y}$来说，是简单的最小化。而$\|\mathbf{X} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T\|_{2,1}$并不是那么简单的，要用最小化的方式关于$U_i$或$V_i$。乘法更新规则隐含地解决了的问题，使$U_i$和$V_i$脱钩。在ADMM背景下，一个自然的提法是优化$\|\mathbf{X} - \mathbf{Y}\|_{2,1}$，约束条件为$\mathbf{y} = \mathbf{u}_1\mathbf{u}_2\cdots\mathbf{u}_r\mathbf{v}_r^T$。**这就是为什么我们考虑解出问题，如**公式（6）。

为了验证的稳健性和有效性。$l_{2,1}$-规范使用于我们的模型，不同发散函数（即β=2，β=1和β=0）的情况将在后面的章节中讨论。关于不同发散函数的优化算法也将在附录A中描述。

对于公式（6），我们引入了一种基于ADMM的有效优化算法[35]。公式（6）的拉格朗日函数为

$$\mathcal{L}(\mathbf{Y},\mathbf{U}_i,\mathbf{V}_i,\mathbf{V}_i^+,\boldsymbol{\mu},\boldsymbol{\lambda}_i) = \|\mathbf{X} - \mathbf{Y}\|_{2,1} + \langle\boldsymbol{\mu},\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T\rangle + \frac{\rho}{2}\|\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T\|_F^2 + \sum_{i=1}^{r-1}\langle\boldsymbol{\lambda}_i,\mathbf{V}_i - \mathbf{V}_i^+\rangle + \frac{\rho}{2}\sum_{i=1}^{r-1}\|\mathbf{V}_i - \mathbf{V}_i^+\|_F^2 \tag{7}$$

其中，ρ表示一个惩罚参数，$\boldsymbol{\mu}$和$\boldsymbol{\lambda}_i$，都表示拉格朗日乘法器，$\langle\cdot,\cdot\rangle$表示内积运算。

式（7）的交替算法是通过最小化得出的$\mathcal{L}$相对于$\mathbf{Y},\mathbf{U},\mathbf{V},\mathbf{V}^+$，一次一个，$_i$同时固定其他。即通过重复以下步骤可以得出解决方案。

$$\begin{aligned}
\mathbf{U}_i^{t+1} &= \arg\min_{\mathbf{U}_i}\mathcal{L}(\mathbf{Y}^t,\mathbf{U}_i,\mathbf{V}_i^t,\mathbf{V}_i^{+t},\boldsymbol{\mu}^t,\boldsymbol{\lambda}_i^t),\\
\mathbf{V}_i^{t+1} &= \arg\min_{\mathbf{V}_i}\mathcal{L}(\mathbf{Y}^t,\mathbf{U}_i^{t+1},\mathbf{V}_i,\mathbf{V}_i^{+t},\boldsymbol{\mu}^t,\boldsymbol{\lambda}_i^t),\\
\mathbf{V}_i^{+t+1} &= \arg\min_{\mathbf{V}_i^+}\mathcal{L}(\mathbf{Y}^t,\mathbf{U}_i^{t+1},\mathbf{V}_i^{t+1},\mathbf{V}_i^+,\boldsymbol{\mu}^t,\boldsymbol{\lambda}_i^t),\\
\mathbf{Y}^{t+1} &= \arg\min_{\mathbf{Y}}\mathcal{L}(\mathbf{Y},\mathbf{U}_i^{t+1},\mathbf{V}_i^{t+1},\mathbf{V}_i^{+t+1},\boldsymbol{\mu}^t,\boldsymbol{\lambda}_i^t),\\
\boldsymbol{\mu}^{t+1} &= \boldsymbol{\mu}^t + \rho(\mathbf{Y}^{t+1} - \mathbf{U}_1^{t+1}\mathbf{U}_2^{t+1}\cdots\mathbf{U}_r^{t+1}\mathbf{V}_r^{T,t+1}),\\
\boldsymbol{\lambda}_i^{t+1} &= \boldsymbol{\lambda}_i^t + \rho(\mathbf{V}_i^{t+1} - \mathbf{V}_i^{+t+1}).
\end{aligned} \tag{8}$$

然后用以下公式更新乘数$\boldsymbol{\mu}$和$\boldsymbol{\lambda}_i$ [35]。

$$\begin{aligned}
\boldsymbol{\mu}^{t+1} &= \boldsymbol{\mu}^t + \rho(\mathbf{Y}^{t+1} - \mathbf{U}_1^{t+1}\mathbf{U}_2^{t+1}\cdots\mathbf{U}_r^{t+1}\mathbf{V}_r^{T,t+1}),\\
\boldsymbol{\lambda}_i^{t+1} &= \boldsymbol{\lambda}_i^t + \rho(\mathbf{V}_i^{t+1} - \mathbf{V}_i^{+t+1}).
\end{aligned} \tag{9}$$

值得注意的是，我们的模型与深层半封闭式模型不同。NMF模型[34]。首先，深度半NMF是基于Frobe- nius norm的，众所周知，Frobe- nius norm对噪声数据和异常值很敏感。另一方面，深层半NMF中的目标数据表示不能直接分配离散聚类结果，因此需要进行后处理，为每个数据样本分配类别标签。因此，它很难达到一个稳定的结果。此外，我们模型的优化算法与[34]完全不同，下面将介绍。

3.2. 优化

本文介绍了一种迭代更新算法来解决优化问题。具体来说，我们更新在保留每个不同变量的情况下，对公式(7)进行分析。其他变量固定。

在优化之前，我们通过分解输入数据矩阵$\mathbf{X}\approx\mathbf{U}_1\mathbf{V}^T$，其中$\mathbf{V}_1\in\mathbb{R}^{n\times c_1}$，$\mathbf{U}_1\in\mathbb{R}^{m\times c_1}$，进行预训练。然后，得到的$_1$表示矩阵$\mathbf{V}_1$被进一步分解为$\mathbf{V}\approx\mathbf{U}_2\mathbf{V}^T$，其中$\mathbf{V}\in\mathbb{R}^{n\times c_2}$，$\mathbf{U}\in\mathbb{R}^{c_1\times c_2}$。我们分别表示$c_1$和$c_2$为第一层和第二层的维度。[1]继续这样做，$_2$所有的层都将被预训练，这将极大地提高我们模型的有效性并减少训练时间。这一招已经很成功了在深度自动编码器网络中完全采用[35]。

通过简单的代数，公式（7）可以被改写为

$$\begin{aligned}
\mathcal{L}(\mathbf{Y},\mathbf{U}_i,\mathbf{V}_i,\mathbf{V}_i^+,\boldsymbol{\mu},\boldsymbol{\lambda}_i) &= \mathrm{Tr}\left[(\mathbf{X}-\mathbf{Y})\mathbf{D}(\mathbf{X}-\mathbf{Y})^T\right]\\
&+ \frac{\rho}{2}\mathrm{Tr}\left[\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r\right]^T\left[\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r\right]\\
&+ \langle\boldsymbol{\mu},\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T\rangle + \sum_{i=1}^{r}\langle\boldsymbol{\lambda}_i,\mathbf{V}_i - \mathbf{V}_i^+\rangle\\
&+ \frac{\rho}{2}\sum_{i=1}^{r}\left[\mathbf{V}_i - \mathbf{V}_i^+\right]^T\left[\mathbf{V}_i - \mathbf{V}_i^+\right].
\end{aligned} \tag{10}$$

其中$\mathbf{D}$表示一个对角线矩阵，第j个对角线元素是

$$d_j = \frac{1}{2\|\mathbf{e}_j\|_2} \tag{11}$$

而$_{ej}$是$\mathbf{E}=\mathbf{X}-\mathbf{Y}$的第$j$列。

3.2.1. 更新U_i

关于U_i的优化问题是

$$\mathcal{L}_{U_i} = \langle\boldsymbol{\mu},\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T\rangle + \frac{\rho}{2}\mathrm{Tr}\left[\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r\right]^T\left[\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r\right]. \tag{12}$$

设定$\frac{\partial\mathcal{L}_{U_i}}{\partial U_i} = 0$，我们有

$$\mathbf{U}_i = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\left(\boldsymbol{\Phi}^T\mathbf{Y}\tilde{\mathbf{V}}_i + \frac{\boldsymbol{\Phi}^T\boldsymbol{\mu}_{\tilde{V}_i}}{\rho}\right)\left(\tilde{\mathbf{V}}_i^T\tilde{\mathbf{V}}_i\right)^{-1}, \tag{13}$$

其中$\boldsymbol{\Phi}=\mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_{i-1}$，$_{Vi}$是第 i个 $\tilde{}$的重建。层的中心点矩阵。

3.2.2. 更新$V_i\ (i < r)$

关于$\mathbf{V}$的 优化问题是

$$\mathcal{L}_V = \langle\boldsymbol{\mu},\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r^T\rangle + \frac{\rho}{2}\mathrm{Tr}\left[\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r\right]^T\left[\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r\right] + \sum_{i=1}^{r}\left[\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2\cdots\mathbf{U}_r\mathbf{V}_r\right]^T\left[\mathbf{V}_i - \mathbf{V}_i^+\right] + \frac{\rho}{2}\sum_{i=1}^{r}\langle\boldsymbol{\lambda}_i,\mathbf{V}_i - \mathbf{V}_i^+\rangle + \frac{\rho}{2}\sum_{i=1}^{r}\left[\mathbf{V}_i - \mathbf{V}_i^+\right]^T\left[\mathbf{V}_i - \mathbf{V}_i^+\right]. \tag{14}$$

同样地，设置$\frac{\partial\mathcal{L}_V}{\partial\mathbf{V}} = 0$，我们得到

$$\mathbf{V}_i = \left(\mathbf{Y}^T\boldsymbol{\Phi}\mathbf{U}_i + \mathbf{V}_i^+ + \frac{\boldsymbol{\mu}^T\boldsymbol{\Phi}\mathbf{U}_i}{\rho} - \frac{\boldsymbol{\lambda}_i}{\rho}\right)\left(\mathbf{I} + \mathbf{U}_i^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{U}_i\right)^{-1}, \tag{15}$$

其中，$\mathbf{I}$表示一个身份矩阵。

[1]为简单起见，本文将第1层至第r层的层大小（维度）写为$[c_1, \ldots, c_r]$。

3.2.3. 更新$\mathbf{V}_r$（即v_i，（$i=r$））。

关于$\mathbf{V}$的优化问题，可以表述为

$$\min_{\mathbf{V}} \|\mathbf{Y}-\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T\|_{2,1} = \min_{\mathbf{v}} \sum_{j=1}^{n} d\left(\|\mathbf{U}_{ij}-\mathbf{v}_2\mathbf{v}\cdots r_j\|_2\right) \quad (16)$$

$$\text{s.t.}\ (\mathbf{V}_r)_{.c}=\{0,1\},\ \sum_{c=1}^{C}(v_r)_c=1,$$

其中x_j是$\mathbf{X}$的第j个数据点，v_j表示$\mathbf{V}$的第j列^T。显然，公式（16）对每个j都是独立的。因此，对于特定的j，可以通过以下方式独立求解

$$\min_{v} d\left(\|x-\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}_r v\|_2^2\right) \quad \text{s.t.}\ v\in\{0,1\}^C,\ \sum_{c=1} v_c=1. \quad (17)$$

由于v满足1-*of*-C编码方案，所以公式（17）显然有C种可能的解。我们可以发现，每一个单独的 解决方案正是身份矩阵$I_C=[f_1,f_2,\ldots,f_C]$的第c列。因此，我们可以通过进行穷举搜索得到最优解，v^*，即。

$$v^*=\mathbf{f}_c \quad (18)$$

其中

$$c=\arg\min_{c} d\left(x\|\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{f}_r\|_2^2\right). \quad (19)$$

3.2.4. 更新$\mathbf{Y}$

关于$\mathbf{Y}$的优化问题是

$$L_Y=\mathrm{Tr}(\mathbf{X}-\mathbf{Y})\mathbf{D}(\mathbf{X}-\mathbf{Y})^T + \frac{\rho}{2}\mathrm{Tr}\left(\mathbf{Y}-\mathbf{U}\mathbf{U}_1\ldots\mathbf{U}\mathbf{V}_r^T\right)\left(\mathbf{Y}-\mathbf{U}\mathbf{U}_1\ldots\mathbf{U}\mathbf{V}_r^T\right)^T + (\boldsymbol{\mu},\mathbf{Y}-\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T). \quad (20)$$

设定$\frac{\partial L_Y}{\partial Y}=0$，我们得到

$$\mathbf{Y}=2\mathbf{X}\mathbf{D}+\rho\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T+\boldsymbol{\mu}(2\mathbf{D}+\rho\mathbf{I})^{-1}. \quad (21)$$

公式(21)给我们提供了当采用$l_{2,1}$-norm来衡量重建误差时的$\mathbf{Y}$的更新规则。人们可能会问，如果采用其他发散函数呢？这里我们给出了不同发散函数下的$\mathbf{Y}$的更新规则。因此，我们需要解决如下的一般优化问题

$$L_Y=D_\beta(\mathbf{X}|\mathbf{Y})+(\boldsymbol{\mu},\mathbf{Y}-\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T)+\frac{\rho}{2}\mathrm{Tr}\left(\mathbf{Y}-\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T\right)\left(\mathbf{Y}-\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T\right)^T. \quad (22)$$

根据公式（22），$\mathbf{Y}$的更新规则为

$$\mathbf{Y}_{ij}=\begin{cases}\dfrac{(2\mathbf{X}+\rho\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T-\mu_{ij})}{2+\rho} & \text{当}\beta=2\text{时。}\\ \dfrac{z+\sqrt{z^2+4\rho x_{ij}}}{2\rho} & \text{当}\beta=1\text{时。}\\ \mathbf{S}_{ij}\mathbf{A}_{ij} & \text{当}\beta=0\text{时。}\end{cases} \quad (23)$$

z、$\mathbf{S}$、$\mathbf{A}$的定义以及关于公式（23）的具体细节在附录A中给出。

3.2.5. 更新$\mathbf{V}^+$

关于$\mathbf{V}^+$的优化问题是

$$L_{V^+}=\sum_{i=1}^{r}\mathrm{Tr}\left(\mathbf{V}_i^+-\mathbf{V}_i\right)\left(\mathbf{V}_i^+-\mathbf{V}_i\right)^T + (\boldsymbol{\lambda},\mathbf{V}^+-\mathbf{V}). \quad (24)$$

同样地，设定$\frac{\partial L_{V^+}}{\partial V_i^+}=0$，我们得到

$$\mathbf{V}_i^+=\mathbf{V}_i+\frac{\lambda_i}{\rho}。 \quad (25)$$

为了清楚起见，所提出的算法在算法1中被一步步地总结出来。

算法1 用于数据聚类的鲁棒性深度k-means（RDKM）。

输入。 输入数据矩阵$\mathbf{X}$，层大小p。

输出。 每层的U_i和V_i。

迭代。

对所有层**做**

$U_i, V_i \leftarrow$ k-*means* V_{i-1},层(i)

结束

按照公式~(11)中的定义，初始化$\mathbf{D}$。

重复

对所有层**做**

1. 更新$\tilde{V}_i=\begin{cases}\mathbf{V} & \text{如果}i=r。\\ \mathbf{U}^r\tilde{\mathbf{V}}^T_{i+1} & \text{否则的话。}\end{cases}$
2. 计算$\boldsymbol{\Phi}=\mathbf{U}_{i-1}\cdots\mathbf{U}_{j=1}$
3. 根据公式~(13)更新U_i。
4. 根据公式~(15)，更新V_i。
5. 根据公式~（21）或公式~（23）更新$\mathbf{Y}$。
6. 根据公式~(25)，更新$\mathbf{V}^+$。
7. 根据公式~(11)计算$\mathbf{D}$。
8. 更新$\boldsymbol{\mu}\leftarrow\boldsymbol{\mu}+\rho(\mathbf{Y}-\mathbf{U}_{1U2}\ldots\mathbf{U}\mathbf{V}_r^T)$。
9. 更新$\boldsymbol{\lambda}_i\leftarrow\boldsymbol{\lambda}_i+\rho(V_i-\mathbf{V}^+_i)$。

结束，直到

收敛

3.3. 收敛分析

我们证明算法1的收敛性如下：方程（6）中的目标函数可以分为四个子问题，每个子问题都是一个凸问题，相对于相应的

变量。通过迭代解决这些子问题，可以保证
我们可以搜索到每个子问题的最优解。最后，算法1将收敛到一个局部最小值。

对于不同的发散函数，相应的优化算法的主要区别在于$\mathbf{Y}$的更新规则。正如我们在附录A中所介绍的，关于不同发散的$\mathbf{Y}$的优化子问题都是凸概率问题，我们可以得到相应的闭式解，如公式（A.2）、（A.5）和（A.14）中所示。因此，关于不同发散的优化算法也将收敛于局部最小值。

4. 实验

在本文中，我们实验性地评估了以下效果
我们的方法。我们在12个基准数据集上对提议的RDKM与六个基准进行了比较：标准K-means[8]。
NMF[32]，正交NMF（ONMF）[30]，半NMF（SNMF）[38]。
$l_{2,1}$-冗余模型[36]和深度半冗余模型（DeepSNMF）[34]。

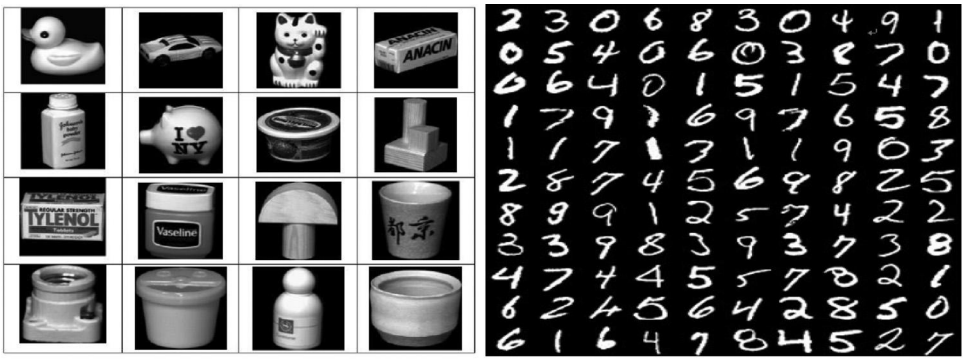
4.1. 数据集

在我们的实验中，我们采用了12个基准数据集，包括两个基因表达数据集，四个文本数据集和六个图像数据集。作为说明，图2显示了数据集COIL和MNIST的样本图像。表1总结了所有数据集的具体尾数，从中我们可以看出实例的数量是
从102到7094不等，特征的数量从256到7511，涵盖了广泛的属性。

4.2. 参数设置

对于k-*means*算法，在所有的数据集上，我们进行k-*means*，直到它收敛为止。为了进行公平的比较，k-*means*的结果也被用来作为其他比较方法的初始化。对于被比较的方法，我们设置的参数与每一个报告

中的参数相同。



(a) COIL (b) MNIST

图2.(a)COIL和(b)MNIST的样本图像。

表1
我们的实验数据集的细节。

身份证	数据集	# 样本	# #特征	#类	#类型
1	睾丸激素	102	5966	2	基因
2	耶鲁大学32	165	1024	15	形象
3	龙门	203	3312	5	基因
4	ORL32	400	1024	40	形象
5	绕组	1440	1024	20	形象
6	塞米昂	1593	256	10	形象
7	MSRA	1799	256	12	形象
8	文本	1946	7511	2	案文
9	鹤岗市	2431	462	2	文本
10	MNIST05	3495	784	10	形象
11	Cacmcisi	4663	348	2	文本
12	经典	7094	462	4	文本

文件。如果没有建议的值，我们就详尽地搜索这些参数，并使用产生最佳性 能 的参数。对于我们的RDKM，根据[14,39]，层大小（如3.2所述）被设定为[50 *C*]、[100 *C*]和[100 50 *C*]。至于
参数ρ，我们从{1e5, 1e4, 1e3, 1e2, 0.1, 1, 10, 100}中搜索。

为了减少初始化的影响，我们将实验重复了20次，并报告了这20次重复的平均性能。

4.3. 结果和分析

表2显示了所有方法在12个数据集上的聚类精度（ACC）、归一化互信息（NMI）和纯度方面的性能。可以看出，所提出的方法在大多数情况下优于其他算法。在去尾，对于ACC，我们的模型在12个数据集中有11次取得了最好的结果。对于NMI，我们的模型取得了10次最佳结果。对于纯净度，这个数字也是10。总而言之，聚类性能足以验证所提模型的有效性。RDKM的优势表明，通过探索数据的层次语义来发现更好的聚类结构是有益的。原因是，通过应用深度框架来分层执行*K*-means，数据的层次信息可以被分层利用，最终为聚类任务获得更好的高可识别性、最终层的表示。通过巧妙地结合深度框架和*k-means*模型，我们的模型能够提高一般情况下的聚类性能。

根据本文的理论分析和实证结果，可以得出结论，将深度结构学习和经典的机器学习模型结合到一个统一的框架中是一个有趣的研究趋势。

4.4. 参数讨论

在我们的模型中，有两个参数，即层大小和惩罚参数ρ，需要进行调整。这里我们研究了不同参数设置下的聚类性能。如图3所示，我们可以发现，聚类性能在不同的层大小设置下是稳定的，而性能对惩罚参数ρ有点敏感。对于图像数据集（如Yale32，ORL32，COIL和MSRA），当ρ在[1e3,1e1]范围内时，可以得到更好的结果。对于基因表达数据集（如Lungml），当ρ在[1e2,1]范围内时，可以获得更好的结果。而对于文本数据集（如Cran- med），在[1e5,1e3]范围内搜索ρ将是一个更好的选择。一般来说，我们可以在[1e3,1]的范围内搜索参数ρ，以获得相对好的性能。

4.5. 收敛分析

在这一小节中，我们用经验说明我们的方法收敛的速度。图4显示了我们的RDKM的收敛曲线，其中X*轴*表示迭代次数，而Y*轴*表示目标值。可以看出，我们的RDKM的更新规则收敛得非常快，通常在100次 迭代之内。对于数据集MSRA，它甚至在10次迭代内收敛，这进一步证明了所提出的操作定时算法的有效性。

4.6. 发散函数选择

如第2节所述，可以采用几种广泛使用的发散函数进行残差计算。我们在之前的实验中使用了$l_{2,1}$-norm。在本节中，我们实证研 究 了不同发散函数对聚类性能的影响。

从公式（6）和（7）中可以看出，当发散函数改变时，只有更新**Y**的步骤会改变。也就是说，在不同的发散函数下，除**Y**外，所有变量的更新都是一样的。我们给出了以下的更新规则
当β=2（欧几里得距离），β=1（Kullback-Leibler Diver-）时，**Y**就会出现。
gence）和β=0（Itakura-Saito Divergence），见附录A。

图5显示了不同发散函数下的聚类性能。为了比较，我们的模型（即基于$l_{2,1\text{-norm}}$的发散函数）的结果也被记录下来。很明显，$l_{2,1\text{-norm}}$在所有数据集上都优于其他发散函数，这再次验证了我们模型的稳健性。对于β=2、β=1和β=0这三种情况，我们可以看到β=1在数据集Yale32上获得了良好的结果，β=0在数据集Lungml上获得了良好的结果，而β=2在数据集上获得了更好的结果。

表2

通过比较方法的准确度/NMI/纯度（平均值±标准差）来衡量聚类性能。每个数据集上的最佳性能被加粗。-/∘表示我们的方法明显优于/劣于比较的方法（95%显著性水平下的配对*t*检验）。最后一行总结了我们方法的赢/输（w/t/l）数。

数据集	度量衡	Kmeans	NMF	ONMF	L21NMF	SNMF	DeepSNMF	RDKM
淫媒	ACC	58.82 ± 0.5-	58.63 ± 0.6 -	57.65 ± 0.9 -	58.14 ± 0.5 -	58.82 ± 0.5-	59.98 ± 6.9 -	**62.78 ± 2.1**
	NMI	12.39 ± 0.3 -	12.33 ± 0.3 -	11.78 ± 0.5 -	11.98 ± 0.3 -	12.39 ± 0.2 -	11.93 ± 1.0 -	**13.36 ± 1.8**
	纯洁性	58.82 ± 0.5 -	58.63 ± 0.6 -	57.65 ± 0.9 -	58.14 ± 0.5 -	58.82 ± 0.5 -	59.98 ± 6.9 -	**62.78 ± 2.1**
Yael32	ACC	37.39 ± 3.1 -	35.88 ± 3.3 -	35.58 ± 3.3 -	38.67 ± 2.7 -	37.64 ± 2.9 -	29.09 ± 1.6 -	**40.06 ± 2.8**
	NMI	43.05 ± 3.1 -	42.42 ± 3.1 -	41.04 ± 3.0	45.02 ± 1.6 -	43.44 ± 2.3 -	28.92 ± 1.1 -	**48.22 ± 1.9**
	纯洁性	39.76 ± 3.1 -	37.88 ± 2.7 -	37.82 ± 3.1 -	40.48 ± 1.7 -	39.82 ± 2.6 -	32.12 ± 1.8 -	**43.52 ± 1.7**
龙门	ACC	68.92 ± 11.1	62.27 ± 7.2 -	68.92 ± 11.1	62.17 ± 5.9 -	68.92 ± 11.1	58.62 ± 7.0 -	**69.01 ± 1.5**
	NMI	52.10 ± 8.1	47.22 ± 4.4 -	52.10 ± 8.1	47.07 ± 3.3 -	52.10 ± 8.1	16.17 ± 1.8 -	**52.72 ± 5.2**
	纯洁性	87.04 ± 3.0 -	85.32 ± 4.0 -	87.04 ± 3.0 -	84.63 ± 3.2 -	87.04 ± 3.0 -	72.17 ± 2.4 -	**89.41 ± 2.9**
ORL32	ACC	50.30 ± 2.2 -	51.97 ± 2.8 -	49.90 ± 3.1 -	53.40 ± 4.1	51.78 ± 3.5 -	49.86 ± 2.0 -	**54.50 ± 1.2**
	NMI	71.06 ± 1.3 -	72.10 ± 1.3 -	70.11 ± 1.7 -	72.70 ± 1.8 -	71.76 ± 1.9 -	68.83 ± 1.3 -	**72.94 ± 1.5**
	纯洁性	56.06 ± 2.4 -	56.37 ± 2.2 -	55.15 ± 2.9 -	58.07 ± 3.5 -	56.07 ± 3.1 -	57.18 ± 2.1 -	**63.55 ± 2.0**
绕组	ACC	59.43 ± 6.8 -	62.24 ± 3.1 -	58.35 ± 6.0 -	63.49 ± 4.4 -	63.78 ± 5.9 -	66.36 ± 6.2 -	**68.03 ± 3.8**
	NMI	74.53 ± 2.8 -	73.12 ± 1.7 -	72.84 ± 2.6 -	74.04 ± 2.3 -	74.91 ± 3.0 -	77.52 ± 7.4 -	**78.99 ± 1.6**
	纯洁性	64.62 ± 5.1 -	66.24 ± 2.7 -	62.95 ± 4.7 -	67.03 ± 3.5 -	67.10 ± 4.8 -	66.94 ± 6.7 -	**71.01 ± 3.6**
塞米昂	ACC	57.34 ± 4.7 -	49.64 ± 5.6 -	53.87 ± 5.8 -	47.88 ± 2.7 -	49.72 ± 5.4 -	56.07 ± 2.0 -	**62.07 ± 5.9**
	NMI	51.93 ± 2.8 -	42.46 ± 3.5 -	48.50 ± 2.8 -	42.47 ± 2.0 -	43.95 ± 2.8 -	44.77 ± 1.2 -	**53.97 ± 3.3**
	纯洁性	59.69 ± 3.8 -	51.65 ± 5.0 -	56.69 ± 4.4 -	50.89 ± 2.0 -	52.32 ± 4.7 -	58.27 ± 2.6 -	**67.93 ± 4.6**
MSRA	ACC	48.73 ± 4.4 -	48.93 ± 2.7 -	48.73 ± 5.8 -	51.63 ± 5.2	49.24 ± 4.4 -	49.19 ± 1.5 -	**52.24 ± 2.2**
	NMI	55.85 ± 5.1 -	55.86 ± 5.0 -	55.64 ± 2.8 -	59.06 ± 4.2	55.44 ± 4.9 -	**60.10 ± 0.2**	59.83 ± 2.6
	纯洁性	52.42 ± 3.8 -	51.86 ± 3.3 -	52.30 ± 4.4 -	55.14 ± 4.2	51.77 ± 3.8 -	54.75 ± 0.2 -	**55.79 ± 2.4**
文本	ACC	91.84 ± 2.1 -	93.85 ± 3.9	92.47 ± 2.9 -	90.21 ± 4.0 -	90.99 ± 2.0 -	90.67 ± 4.5 -	**93.88 ± 5.7**
	NMI	61.31 ± 6.5	61.21 ± 1.5 -	60.88 ± 1.9 -	60.81 ± 3.0 -	57.85 ± 5.2 -	60.01 ± 4.2 -	**61.55 ± 5.4**
	纯洁性	91.84 ± 2.1 -	**94.08 ± 3.6** ∘	92.71 ± 2.7 -	90.62 ± 3.2 -	90.99 ± 2.0 -	90.67 ± 4.5 -	93.88 ± 5.7
鹤岗市	ACC	74.58 ± 0.1 -	80.13 ± 8.8 -	77.31 ± 1.2 -	77.39 ± 2.5 -	76.49 ± 4.9 -	80.23 ± 3.1 -	**82.31 ± 3.9**
	NMI	18.79 ± 0.3 -	31.67 ± 5.6	20.74 ± 0.3 -	20.84 ± 1.2 -	24.66 ± 5.4 -	25.05 ± 2.4 -	**32.89 ± 2.3**
	纯洁性	74.58 ± 0.1 -	80.38 ± 8.0 -	77.78 ± 3.4 -	77.67 ± 3.2 -	79.10 ± 6.1 -	**82.51 ± 3.0** ∘	82.31 ± 3.9
牧师	ACC	54.84 ± 4.1 -	52.30 ± 3.3 -	**58.84 ± 4.7** ∘	54.00 ± 5.1 -	48.65 ± 2.2 -	56.18 ± 1.2∘	55.46 ± 3.7
	NMI	49.44 ± 1.6 -	43.21 ± 2.1 -	49.92 ± 1.4∘	45.26 ± 3.0 -	41.26 ± 1.8 -	**49.81 ± 2.0** ∘	49.53 ± 2.4
	纯洁性	58.58 ± 2.1 -	54.22 ± 2.4 -	59.84 ± 2.3 -	57.18 ± 4.3 -	52.89 ± 2.7 -	60.73 ± 2.4 -	**62.10 ± 2.8**
Cacmcisi	ACC	91.99 ± 0.2 -	89.75 ± 5.4 -	94.96 ± 0.6 -	95.37 ± 0.8 -	92.22 ± 0.3 -	92.80 ± 3.5 -	**97.12 ± 7.7**
	NMI	58.47 ± 0.1 -	60.01 ± 2.5 -	70.42 ± 0.2 -	72.05 ± 0.4 -	70.52 ± 0.2 -	70.07 ± 3.3 -	**73.06 ± 2.4**
	纯洁性	91.99 ± 0.2 -	91.70 ± 3.2 -	94.96 ± 0.6 -	95.37 ± 0.8 -	93.69 ± 0.7 -	94.86 ± 4.1 -	**97.12 ± 7.7**
经典	ACC	67.45 ± 0.3 -	70.31 ± 3.2 -	76.52 ± 6.6 -	76.19 ± 6.4 -	74.44 ± 2.0 -	72.40 ± 1.3 -	**76.98 ± 5.9**
	NMI	46.76 ± 1.3 -	50.12 ± 2.2 -	47.91 ± 6.4 -	57.45 ± 7.9 -	55.30 ± 2.6 -	56.27 ± 2.5 -	**59.61 ± 4.1**
	纯洁性	70.48 ± 1.0 -	74.09 ± 3.3 -	77.74 ± 5.2 -	79.73 ± 4.2 -	76.42 ± 1.2 -	78.41 ± 1.9 -	**81.07 ± 4.1**
RDKM。	ACC	11/1/0	11/1/0	10/1/1	10/2/0	11/1/0	11/0/1	
w/t/l	NMI	10/2/0	11/1/0	9/2/1	11/1/0	11/1/0	10/1/1	
	纯洁性	12/0/0	11/0/1	12/0/0	11/1/0	12/0/0	11/0/1	

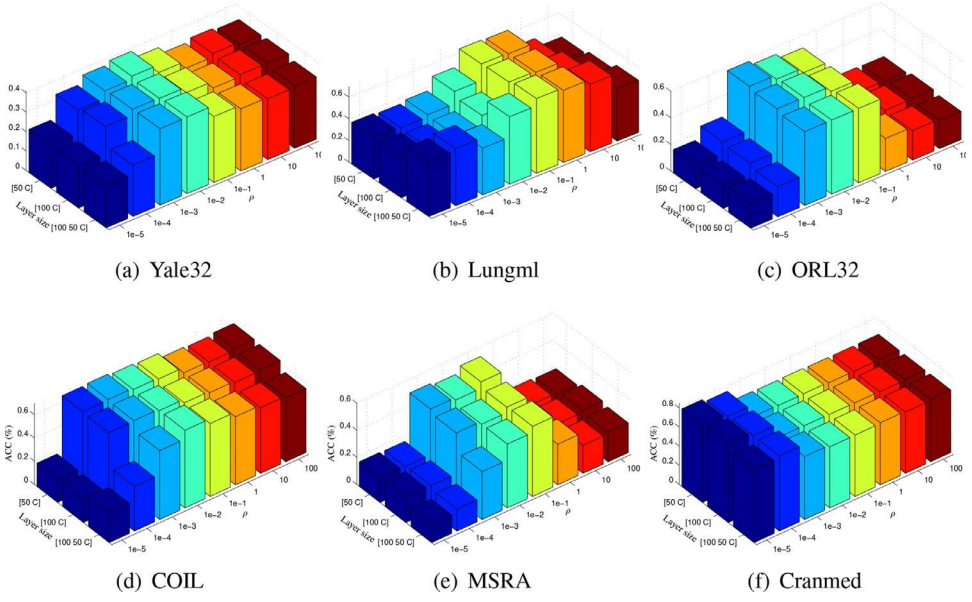


图3.RDKM的聚类结果与层的大小和参数ρ有关。

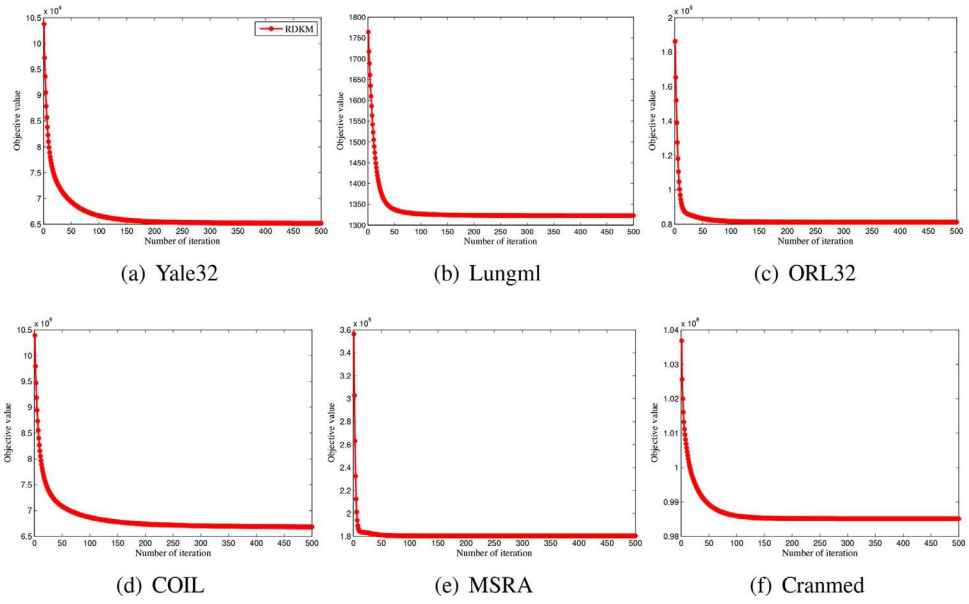


图4.RDKM的收敛速度。

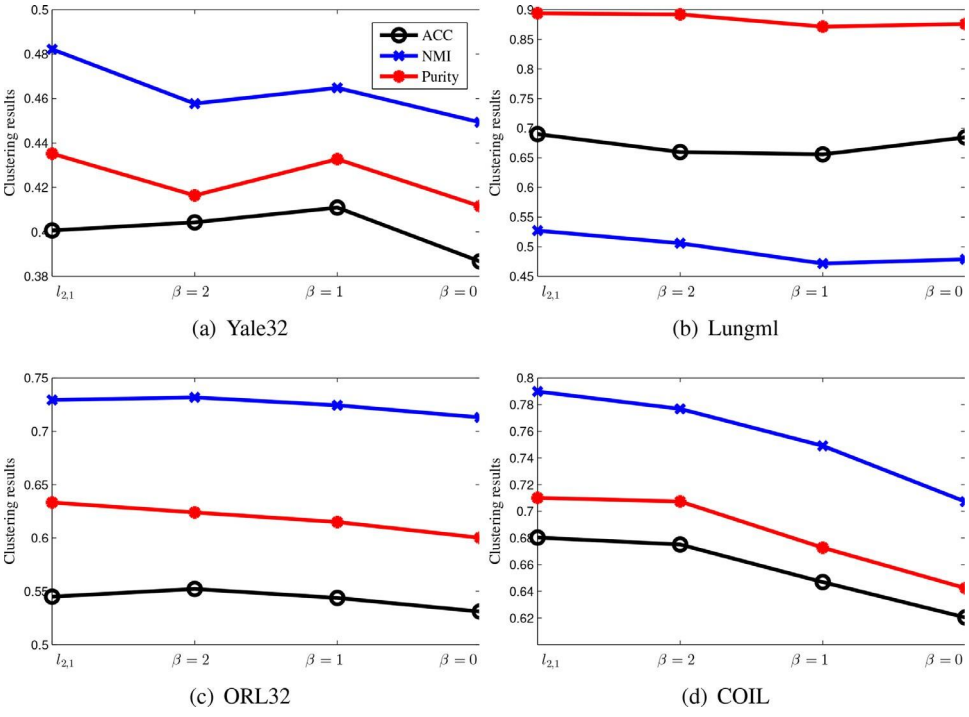


图5.RDKM的聚类性能与分歧函数的关系。

数据集ORL32和COIL。也就是说，不同的发散函数在不同的数据集上得到更好的结果。一般来说，$l_{2,1}$-norm可能是一个更好的选择，因为它始终能取得良好的性能。

5. 总结

在本文中，我们引入了一个稳健的深度*k-means*模型来学习与不同的隐性低级属性相关的隐性表征。通过使用深层结构来分层执行*k-means*，数据的层次语义可以以分层的方式被利用。来自同一类别的数据样本被迫逐层靠近，这有利于聚类---------。

摄取任务。我们的模型的目标函数被推导到一个更可追踪的形式，这样可以更容易地解决优化问题，并获得最终的稳健结果。在12个基准数据集上的经验-心理结果证实了这一点。(i) 与传统的和最先进的方法相比，所提出的模型在聚类性 能 上取得了突破性进展。
(ii) 聚类性能对于不同的层大小以及发散函数的设置是稳健的；（iii）提出的优化算法是有效的，并且收敛速度非常快。在我们未来的工作中，将我们的深度模型和其他机器学习模型（如核学习和分类方法）结合到一个统一的框架中是非常有趣的。

竞争性利益声明

作者声明，他们没有已知的竞争性财务利益或个人关系，可能会影响本文的工作。

鸣谢

这项工作得到了国家科学基金国家重点项目（61836006）、国家自然科学基金杰出青年学者项目（61625204）、四川省科技计划项目（2020YFG0323）和中央高校基础研究基金（1082204112364）的部分支持。

附录A.不同发散的Y的 更新规则

职能

对于不同的发散函数，其主要区别在于
在这里，我们以不同的方式介绍**Y**的更新规则。

发散函数，特别是当β=2、β=1和β=0时。

当β=2（欧氏距离）时，目标函数w.r.t **Y**是

$$L_{Y\beta=2} = D_2(\mathbf{X}|\mathbf{Y}) + (\boldsymbol{\mu}, \mathbf{Y} - \mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}_{Vr})^T + \frac{\rho}{2}\mathrm{Tr}\left(\mathbf{Y} - \mathbf{U}\mathbf{U}_2\ldots\mathbf{u}\mathbf{v}_r^T{}_r\right)^T\left(\mathbf{y} - \mathbf{u}\mathbf{u}_1{}_2\ldots\mathbf{U}\mathbf{V}_r^T{}_r\right). \quad (A.1)$$

计算$L_{Y_{\beta=2}}$对 **Y**的导数并将其设为0，我们有

$$\mathbf{Y} = 2\mathbf{X} + \rho\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T - \boldsymbol{\mu}/(2+\rho)。 \quad (A.2)$$

当β=1（Kullback-Leibler Divergence）时，对 **Y**的目标函数为

$$L_{Y\beta=1} = D_1(\mathbf{X}|\mathbf{Y}) + (\boldsymbol{\mu}, \mathbf{Y} - \mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T) + \frac{\rho}{2}\mathrm{Tr}\left(\mathbf{Y} - \mathbf{U}\mathbf{U}_2\ldots\mathbf{U}\mathbf{V}_r^T{}_r\right)^T\left(\mathbf{Y} - \mathbf{U}\mathbf{U}_2\ldots\mathbf{U}\mathbf{V}_r^T{}_r\right)^T. \quad (A.3)$$

计算L的导数$Y_{\beta=1}$以元素的方式对**Y进行处理**。我们有

$$-\frac{X_{ij}}{Y_{ij}} + 1 + \boldsymbol{\mu}_{ij} + \rho\left(\mathbf{Y} - \mathbf{U}\mathbf{U}\mathbf{U}_{12}\ldots\mathbf{u}\mathbf{v}^T{}_{r\,r\,ij}\right) = 0。 \quad (A.4)$$

因此，**Y**的解决方案是

$$\mathbf{Y}_{ij} = \frac{\rho\mathbf{U}1\mathbf{U}2\ldots\mathbf{U}\mathbf{V}_r^T{}_{r\,ij} - \boldsymbol{\mu}_{ij} - 1}{2\rho}。 \quad (A.5)$$

其中$z = \rho\,\mathrm{U}1\mathrm{U}2\ldots\mathrm{U}\mathbf{V}_r^T{}_{r\,ij} - \boldsymbol{\mu}_{ij} - 1$。

当β=0（板仓-斋藤分歧）时，目标函数**对Y的**影响是

$$L_{Y\beta=0} = D_0(\mathbf{X}|\mathbf{Y}) + (\boldsymbol{\mu}, \mathbf{Y} - \mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T) + \frac{\rho}{2}\mathrm{Tr}\left(\mathbf{Y} - \mathbf{U}\mathbf{U}_2\ldots\mathbf{u}\mathbf{v}_r^T{}_r\right)^T\left(\mathbf{y} - \mathbf{u}\mathbf{u}_1{}_2\ldots\mathbf{U}\mathbf{Y}_r^T{}_r\right)^T. \quad (A.6)$$

很明显，当且仅当梯度在$\mathbf{Y}^*$处消失且Hessian矩阵为正定时，$\mathbf{Y}^*$将是一个最小化器。

$$g_{ij}(\mathbf{Y}^*_{ij}) = 0, \forall i, j, \quad (A.7)$$

其中g_{ij}代表对Y_{ij}的导数，可定义为

$$g_{ij}(y) = \frac{x_{ij}}{y^2} - \frac{1}{y} + \boldsymbol{\mu}_{ij} + \rho\left(y - \mathbf{U}\mathbf{U}_{12}\ldots\mathbf{U}\mathbf{V}_r^T{}_{ij}\right)。 \quad (A.8)$$

记为$p_{ij}(y) = \frac{y^2}{\rho}g_{ij}(y)$，使得$p_{ij}$是一个立方多项式。它是显然，$p_{ij}$与$g_{ij}$有相同的根，$pr_{ij}$与$gr_{ij}$有相同的符号。$p_{ij}$可以明确表示为

$$p_{ij}(y) = y^3 + \mathbf{A}\,y_{ij}^2 + \frac{1}{p}y - \frac{1}{p}\mathbf{X}_{ij}。 \quad (A.9)$$

其中$\mathbf{A} = \frac{1}{\rho}\boldsymbol{\mu} - \mathbf{U}1\mathbf{U}2\ldots\mathbf{U}\mathbf{V}_r^T$，代入$y = s - \frac{1}{3}$Aijin公式(A.9)，可以得到一个压低的立方体

$$q(s) = s^3 + 3\mathbf{B}_{ij}s - 2\mathbf{R}_{ij}。 \quad (A.10)$$

其中$B_{ij} = \frac{1}{\rho} - \frac{1}{3}\mathbf{A}^2$，$R_{ij} = -\frac{1}{2}\mathbf{A}^3 + \frac{1}{6}A_{ij} + \frac{1}{6}X_{ij}$.我们想寻找$q(s)$的正根$s_0 > 0$，使$q_r(s_0) < 0$。当处理非负数据（即$X_{ij} \geq 0$），我们至少可以搜索到一个这样的根，因为$p$（0）$\leq 0$，$p$（$\infty$）$\to\infty$。$q_{ij}$的根是重新$y_t = s_t - \frac{A_{ij}}{3}(t = 0,1,2,$即公式(A.10)中最多有三个根），与$p_{ij}$的根相关。$q_{ij}$的判别式**H**可以定义为

$$\mathbf{H}_{ij} = \mathbf{B}^3_{ij} + \mathbf{R}^2_{ij} \quad (A.11)$$

并有三种情况。

1) $\mathbf{H}_{ij} > 0$, one real root:

$$s_0 = \sqrt[3]{\mathbf{R}_{ij} + \sqrt{\mathbf{H}_{ij}}} + \sqrt[3]{\mathbf{R}_{ij} - \sqrt{\mathbf{H}_{ij}}} \quad (A.12)$$

相应地，p_{ij}的根y_0必须是正的，而且是公式（A.6）的最小化。

2) H_{ij}=0，两个不同的实数根。

一个双根$s_1 = s_2 = -\frac{1}{2}s_0$可以得出。然而，斗漏根对应于$q_{ij}$的拐点，即，$g_{ij}(y_1) = 0$.因此，$y_1$不是公式（A.6）的最小化。因此，相关的根仍然是$S_0$。

3) $\mathbf{H}_{ij} < 0$，三个不同的实数根。

$$s_t = 2\sqrt{-\frac{B_{ij}}{3}}\cos\left(\frac{1}{3}\cos^{-1}\left(\frac{R_{ij}}{\sqrt{-B_{ij}^3}}\right) - t\frac{2\pi}{3}\right)。 \quad (A.13)$$

对于t=0,1,2。当有三个不同的根时，只能是在最小和最大的根上都是正数。由于$s_0 \geq s_1 \geq s_2$，我们只需要检查y_0和y_2（后者仅当y_2>0时）。为了简单起见，我们总是取y_0，这至少可以保证是一个局部的式（A.6）的最小值。我们省略了解决方案的推导，对于更多的详情请参考[40]。

综上所述，我们通过以下方式更新**Y**

$$\mathbf{y}_{ij} = \mathbf{s}_{ij} - \frac{1}{3}\mathbf{a}_{ij}, \quad (A.14)$$

其中

$$s_{ij} = \begin{cases} \sqrt[3]{\mathbf{R}_{ij} + \sqrt{\mathbf{H}_{ij}}} + \sqrt[3]{\mathbf{R}_{ij} - \sqrt{\mathbf{H}_{ij}}}, & \mathbf{H}_{ij} \geq 0, \\ 2\sqrt{-B_{ij}}\cos\left(\frac{1}{3}\cos^{-1}\left(\frac{R_{ij}}{\sqrt{B3_{ij}}}\right) - t\frac{2\pi}{3}\right), & \mathbf{H} < 0. \end{cases} \quad (A.15)$$

参考文献

[1] S. Huang, Z. Kang, Z. Xu, Deep *k-means*: a simple and effective method for data clustering, in:国际神经计算会议论文集
ing for Advanced Applications, Springer, 2020, pp.272-283.

[2] A.K. Jain, Data clustering:50 years beyond *k-means*, Pattern Recognit.Lett.31 (8) (2010) 651-666.

[3] A.Banerjee, S. Merugu, I.S. Dhillon, J. Ghosh, Clustering with Bregman diver- gences, J. Mach.Learn.Res. 6 (2005) 1705-1749.

[4] V.Tunali, T. Bilgin, A. Camurcu, An improved clustering algorithm for text min- ing: multi-cluster spherical *k-means*, Int.Arab J. Inf.Technol.13 (1) (2016).

[5] S.V. Ault, R.J. Perez, C.A. Kimble, J. Wang, 论语音识别算法。Int.J. Mach.学习。Comput.8 (6) (2018) 518-523.

[6] L. Wang, C. Pan, Robust level set image segmentation via a local correntropy--基于*k*-means的聚类, Pattern Recognit.47 (5) (2014) 1917-1925.

[7] Y.Ren, C. Domeniconi, G. Zhang, G. Yu, 加权物体集合聚类。方法和分析, Knowl.Inf.Syst.51 (2) (2017) 661-689.

[8] J.Macqueen,一些分类和分析多变量物体的方法。服务，在。伯克利数学统计学研讨会论文集》(Proceedings of Berkeley Symposium on Mathematical Statistics)

[9] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in:神经信息处理系统的进展，2002年，第849-856页。

[10] D.Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factor- ization for data representation, IEEE Trans.Pattern Anal.Mach.Intell.33 (8) (2010) 1548-1560.

[11] E.Gokcay, J.C. Principe, Information theoretic clustering, IEEE Trans.Pattern Anal.Mach.Intell.24 (2) (2002) 158-171.
[12] F.Gullo, G. Ponti, A. Tagarelli, S. Greco, An information-theoretic approach to hierarchical clustering of uncertain data, Inf.Sci. 402 (2017) 199-215.
[13] S.Huang, I. Tsang, Z. Xu, J.C. Lv, Measuring diversity in graph learning: a uni-fied framework for structured multiview clustering, IEEE Trans.Knowl.Data Eng. (2021) 1-15.
[14] H.Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, in:第21届AAAI人工智能会议论文集，2017，第2921-2927页。
[15] X.Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A.Ng, B. Liu, S.Y. Philip, et al., Top 10 algorithms in data mining, Knowl.Inf.Syst.14 (1) (2008) 1-37.
[16] P.Ji, T. Zhang, H. Li, M. Salzmann, I. Reid, Deep subspace clustering networks, in:Advances in Neural Information Processing Systems, 2017, pp.24-33.
[17] P.Zhou, Y. Hou, J. Feng, Deep adversarial subspace clustering, in:Proceed- ings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp.1596-1604.
[18] X.Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation, in:Proceedings of the 26th International Joint Confer- ence on Artificial Intelligence, 2017, pp.1753-1759.
[19] K.Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in:IEEE国际计算机视觉会议论文集，2017，第5736-5745页。
[20] X.Peng, J. Feng, S. Xiao, W.-Y.Yau, J.T. Zhou, S. Yang, Structured autoencoders for subspace clustering, IEEE Trans.Image Process.27 (10) (2018) 5076-5086.
[21] X.Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, J.T. Zhou, Deep clustering with sam- ple-assignment invariance prior, IEEE Trans.Neural Netw.Learn.Syst.(2019) 1-12.
[22] J.Zhang, C.-G.Li, C. You, X. Qi, H. Zhang, J. Guo, Z. Lin, Self-supervised convo- lutional subspace clustering network, in:IEEE计算机视觉和模式识别会议论文集，2019，第5473-5482页。
[23] C.Ding, X. He, K-means clustering via principal component analysis, in:第21届国际机器学习会议论文集，2004，第29-37页。
[24] C.Buchta, M. Kober, I. Feinerer, K. Hornik, Spherical *k*-means clustering, J. Stat.Softw.50 (10) (2012) 1-22.
[25] S.Khanmohammadi, N. Adibeig, S. Shanehbandy, An improved overlapping *k*-means clustering method for medical applications, Expert Syst.应用 67 (2017) 12-18.
[26] K.M. Kumar, A.R.M. Reddy, An efficient *k*-means clustering filtering algorithm using density based initial cluster centers, Inf.Sci. 418 (2017) 286-301.
[27] S.Chakraborty, D. Paul, S. Das, J. Xu, Entropy weighted power *k*-means cluster- ing, in:International Conference on Artificial Intelligence and Statistics, 2020, pp.691-701.
[28] M.Capó, A. Pérez, J.A. Lozano, An efficient approximation to the *k*-means clus- tering for massive data, Knowledge-Based Syst.117 (2017) 56-69.
[29] Y.Bengio, Learning deep architectures for AI, Found.Trends® Mach.Learn.2 (1) (2009) 1-127.
[30] C.Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix *t*-factorizations for clustering, in:第12届ACM SIGKDD知识发现和数据挖掘国际会议论文集，2006年，第126-135页。
[31] S.Huang, H. Wang, T. Li, T. Li, Z. Xu, Robust graph regularized nonnegative matrix factorization for clustering, Data Min.Knowl.Discov.32 (2) (2018) 483-503.
[32] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in:Ad- vances in Neural Information Processing Systems, 2001, pp.556-562.
[33] C.Févotte, J. Idier, 非负矩阵分解的算法与 *β-分歧*, Neural Comput.23 (9) (2011) 2421-2456.
[34] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, B.W. Schuller, A deep matrix factor-ization method for learning attribute representations, IEEE Trans.Pattern Anal.Mach.Intell.39 (3) (2017) 417-429.
[35] S.Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimiza- tion and statistical learning via the alternating direction method of multipliers, Found.Trends® Mach. learn.3 (1) (2011) 1-122.
[36] D.Kong, C. Ding, H. Huang, Robust nonnegative matrix factorization using l21-norm, in:第20届ACM国际知识管理会议论文集，2011年，第673-682页。
[37] H.Gao, F. Nie, W. Cai, H. Huang, Robust capped norm nonnegative ma- trix factorization: capped norm NMF, in:第24届ACM In- ternational on Conference on Information and Knowledge Management, 2015, pp.871-880.
[38] C.Ding, T. Li, M.I. Jordan, Convex and semi-nonegative matrix factorizations, IEEE Trans.Pattern Anal.Mach.Intell.32 (1) (2010) 45-55.
[39] S.Huang, Z. Kang, Z. Xu, Auto-weighted multi-view clustering via deep matrix decomposition, Pattern Recognit.97 (2020) 1-11.
[40] D.L. Sun, R. Mazumder, Non-negative matrix completion for bandwidth exten- sion: a convex optimization approach, in:IEEE信号处理机器学习国际研讨会论文集，2013年，第1-6页。

黄树东：在中国成都电子科技大学获得计算机科学与工程博士学位。他目前是四川大学计算机科学学院的副研究教授，成都，中国。他的研究兴趣包括机器学习、深度学习、模式识别和数据挖掘。在IEEE TCYB、模式识别、数据挖掘和知识发现、信息科学、基于知识的系统、神经计算、IJCAI、IJCNN等相关会议和期刊上发表研究论文近20篇。他一直是多个顶级会议和期刊的PC会员或审稿人，如AAAI、IJCAI、IEEE TNNLS、IEEE TFS、Neurocomputing等。

赵康：赵康博士于2017年在美国南伊利诺伊大学卡本代尔分校获得计算机科学博士学位。目前，他是中国电子科技大学计算机科学与工程学院的助理教授。他的研究兴趣是机器学习，数据挖掘，模式识别，和深度学习。他在顶级会议和期刊上发表了40多篇研究论文，包括AAAI, IJCAI, ICDE, CVPR, SIGKDD, ICDM, CIKM, SDM, ACML, IEEE Transactions on Cybernetics, ACM TIST, ACM TKDD, Pattern Recognition, Neurocomputing, Knowledge-Based Systems。他是许多顶级会议的PC会员或审稿人，如AAAI, IJCAI, CVPR, ICCV, MM, ICDM, CIKM等。他经常担任IEEE TNNLS、IEEE Transactions on Cybernetics、IEEE TKDE、Neurocomputing等的评审员。

徐增林：在香港中文大学获得计算机科学与工程博士学位。他目前是中国电子科技大学的全职教授。他曾在密歇根州立大学、萨尔州大学卓越中心和马克斯-普朗克信息学研究所，以及后来的普渡大学工作。徐博士的研究兴趣包括机器学习及其在信息检索、健康信息学和社会网络分析中的应用。他曾入选2013年中国青年千人计划。他是2015年AAAI优秀学生论文荣誉奖、2016年ACML最佳学生论文亚军和2016年APNNS青年研究员奖的获得者。

刘全辉：在中国成都电子科技大学获得计算机科学与工程博士学位。他目前是中国成都四川大学计算机科学学院的副研究教授。他的研究兴趣包括社会网络、深度学习和复杂网络。

对文本聚类的特征选择的评估

刘涛 LTMAILBOX@263.SINA.COM
南开大学信息科学系，天津300071，中国

刘胜平 LSP@IS.PKU.EDU.CN
北京大学信息科学系，北京100871，中国

陈正 ZHENGC@MICROSOFT.COM
马维英 WYMA@MICROSOFT.COM
中国北京知春路49号微软亚洲研究院，100080。

摘要

特征选择方法已经成功地应用于文本分类，但由于类标签信息的不可用，很少应用于文本聚类。在本文中，我们首先给出了经验证据，证明特征选择方法可以提高文本聚类算法的效率和性能。然后，我们提出了一种新的特征选择方法，称为"术语贡献（TC）"，并对多种文本聚类的特征选择方法进行了比较研究，包括文档频率（DF）、术语强度（TS）、基于熵（En）、信息增益（IG）和χ^2统计（CHI）。最后，我们提出了一种"迭代特征选择（IF）"方法，通过利用有效的监督特征选择方法来迭代选择特征并进行聚类，从而解决标签不可用的问题。文中提供了关于网络目录数据的详细实验结果。

1. 简介

文本聚类是文本挖掘和信息检索领域的核心问题之一。文本聚类的任务是将相似的文件归为一类。它已被应用于多个领域，包括提高信息检索系统的检索效率（Kowalski，1997），组织搜索引擎响应用户查询的结果（Zamir等人，1997），浏览大型文档集（Cutting等人，1992），以及生成网络文档的分类法（Koller & Sahami，1997）等等。

在文本聚类中，一个文本或文档总是被表示为一个词包。这种表示方法引起了一个严重的问题：特征空间的高维度和固有的数据稀疏性。很明显，单个文档在所有术语的集合上有一个稀疏的向量。由于高维度和数据稀疏性的问题，聚类算法的性能将急剧下降（Aggrawal &Yu, 2000）。因此，降低特征空间的维度是非常可取的。有两种常用的技术来处理这个问题：特征提取和特征选择。特征提取是通过一些功能映射从原始特征中提取一组新的特征（Wyse等人，1980），如主成分分析（PCA）（Jolliffe，1986）和词语聚类（Slonim & Tishby，2000）。这些特征提取方法有一个缺点，即生成的新特征可能没有明确的物理意义，所以聚类结果很难解释（Dash & Liu, 2000）。

特征选择是一个根据某些标准从原始特征集中选择一个子集的过程。所选的特征保留了原有的物理意义，并为数据和学习过程提供了更好的理解。根据是否需要类标签信息，特征选择可以是无监督的，也可以是有监督的。对于有监督的方法，每个特征与类标签的相关性是通过距离、信息依赖性或一致性措施来计算的（Dash & Liu, 1997）。基于信息理论的进一步理论研究可以在（Koller & Sahami, 1996）找到，完整的评论可以在（Blum & Langley, 1997; Jain et al., 2000; Yang & Pedersen, 1997）找到。

关于聚类的特征选择，目前已有一些研究。首先，任何不需要类信息的传统特征选择方法，如文档频率（DF）和术语强度（TS）（Yang，1995），都可以很容易地应用到聚类中。其次，还有一些新提出的方法，例如Dash和Liu（2000）提出了基于熵的特征排序方法（En），其中特征的重要性是通过

第20届国际机器学习会议（ICML-2003）论文集，华盛顿特区，2003年。

对基于数据相似性的熵指数的贡献；个别的 "特征突出性 "被估计出来，使用最小信息长度标准的期望-最大化（EM）算法被推导出来，以选择特征子集和集群的数量（Martin等人，2002）。

虽然上面提到的方法并不直接针对文本文档的聚类，但在本文中我们
介绍两种新的文本特征选择方法
聚类。一个是术语贡献（Term Contribution，TC），它通过对数据集中的文档相似性的总体贡献对特征进行排序。另一种是迭代特征选择（IF），它利用一些成功的特征选择方法（Yang & Pedersen, 1997），如信息增益（IG）和χ^2 统计（CHI），来迭代选择

同时进行文本聚类。

本文的另一个贡献是进行了比较研究
关于文本聚类的特征选择。我们研究了(a)特征选择能在多大程度上提高聚类质量，(b)有多少文档词汇可以被用于聚类。
在文本聚类中，在不损失有用信息的情况下减少了。

(c) 现有功能的优点和缺点是什么？
选择方法在应用于文本聚类时，以及
(d) 不同数据集的结果之间的差异是什么。在本文中，我们试图通过经验性的证据来解决这些问题。我们首先表明，在理想情况下，即每个文档的类别标签已经知道的情况下，特征选择方法可以提高文本聚类的效率和性能。然后，我们对文本聚类的各种特征选择方法进行了比较研究。最后，我们用熵和精度指标评估了基于K-means的迭代特征选择方法的性能。

本文的其余部分组织如下。在第2节中，我们简要介绍了几种特征选择方法，并提出了一种新的特征选择方法，即Term Contribution。在第3节中，我们提出了一种新的迭代特征选择方法，该方法利用有监督的特征选择算法，不需要事先知道类信息。在第4节中，我们进行了几个实验来比较不同的特征选择方法在理想和真实情况下的有效性。最后，我们在第5节中总结了我们的主要贡献。

2. 特征选择方法

在本节中，我们简要介绍了几种有效的特征选择方法，包括两种有监督的方法：IG和CHI，以及四种无监督的方法：DF、TS、En和TC。所有这些方法都为每个单独的特征分配一个分数，然后选择大于预先定义的阈值的特征。

在下文中，让D表示文档集，M表示特征的维度，N表示数据集中文档的数量。

2.1 信息获取(IG)

术语的信息增益（Yang & Pedersen, 1997）衡量的是该术语在文档中的存在与否为类别预测所带来的信息位数。设m为类的数量。术语t的信息增益被定义为

$$IG(t) = -\sum_{i=1}^{m} p(c_i)\log p(c_i) + p(t)\sum_{i=1}^{m} p(c_i \mid t)\log p(c_i \mid t) + p(\bar{t})\sum_{i=1}^{m} p(c_i \mid \bar{t})\log p(c_i \mid \bar{t}) \quad (1)$$

2.2 χ^2 统计数字(CHI)

χ^2统计数字衡量术语之间的关联性
和类别（Galavotti等人，2000）。它被定义为

$$\chi^2(T, C) = \frac{N(p(t,c)\times\bar{p}(\bar{t},c) - p(t,\bar{c})\times p(\bar{t},\bar{c}))^2}{p(t)\times p(\bar{t})\times p(c)\times p(\bar{c})} \quad (2)$$

$$\chi^2(t) \equiv \max_{i=1}^{m}\{\chi^2(t,c_i)\} \quad (3)$$

2.3 文件频率(DF)

文档频率是指一个术语在数据集中出现的文档数量。它是最简单的术语选择标准，很容易以线性计算的复杂性扩展到大型数据集。它是一种简单而有效的文本分类的特征选择方法（Yang & Pedersen, 1997）。

2.4 期限强度(TS)

术语强度最初是为文本检索中的词汇减少而提出和评估的（Wilbur & Sirotkin, 1992），后来又应用于文本分类（Yang, 1995）。它是根据一个术语在一对相关文档的后半部分出现的条件概率来计算的，因为它在前半部分出现过。

$$TS(t) = p(t \in d_j \mid t \in d_i), d_i, d_j \in D \cap sim(d_i, d_j) > \beta \quad (4)$$

其中β是确定相关配对的参数。由于我们需要计算每个文档对的相似度，TS的时间复杂度与文档数量成二次方。由于不需要类标签信息，这种方法也适用于文本聚类中的术语减少。

2.5 基于熵的排名 (En)

基于熵的排名是由Dash和Liu（2000）提出的。在这种方法中，术语被删除后，以熵值的减少来衡量。熵的定义为公式（5）。

$$E(t) = -\sum_{i}^{NN}\sum_{1j\ 1}(S_{i,j} \times \lambda o\gamma(\ _{i,j}) + (1- S_{i,j}) \times \lambda o\gamma(1-\ _{i,j})), \quad (5)$$

其中$_{Si,\ j}$是文件d之间的相似度值$_i$
和$d_{j\cdot Si,j}$被定义为方程（6）。

$$_{Si,j} = \frac{^{e}}{\ln(0.5)} \quad disti^{\cdot j}, \alpha = - \quad (6)$$

其中$_{disti,j}$是文件$_{di}$和j之间的距离。

d_j被删除后，$dist$是术语t被删除后文档之间的平均距离。

这种方法最严重的问题是它的高计算复杂性$O(MN^2)$。当有大量的文件和术语时，它是不切实际的，因此，在实际实验中使用了抽样技术（Dash & Liu, 2000）。

2.6 定期捐款（TC）

我们引入了一种新的特征选择方法，称为 "术语贡献"，它考虑到了术语的权重。因为像DF这样的简单方法假定每个术语在不同的文档中具有相同的重要性，所以它很容易被那些具有高文档频率但在不同类别中分布均匀的普通术语所偏离。TC被提出来处理这个问题。

文本聚类的结果高度依赖于文档的相似度。因此，一个术语的贡献可以被看作是它对文档相似性的贡献。这
文件之间$_{的}$相似性　　和d_j　的计算方法是点积。

$$sim(_{di}, d\ _{j}) = \sum_{t} f(t,\ _{di}) \times f(t, d_j) \quad (7)$$

其中f（t，d）代表tf*idf（Salton，1989）的权重。
文件d中的术语t。

因此，我们把一个术语在数据集中的贡献定义为它对文档相似性的总体贡献。公式为

$$TC(t) = \sum_{i,\ \phi\cap t} f(t,\ _{di}) \times f(t, d_j) \quad (8)$$

"ltc "方案用于计算每个术语的tf*idf值，该方案采用术语在文档中的频率对数，然后乘以该术语的IDF权重，最后将文档长度标准化。

如果所有术语的权重相等，我们只需在术语t出现在文档d中时将$f(t, d)$　1 。那么值$TC(t)$就可以写成方程(9)。

$$TC(t) = DF(t)(DF(t)\quad 1) \quad (9)$$

由于转换是单调增加的，而$DF(t)$（术语t的文档频率）是一个
正整数，DF只是TC的特例。

使用逆向文档索引技术（Salton, 1989），TC的时间复杂度为$O(MN^2)$，其中N是每个术语出现的平均文档。

3. 迭代特征选择（IF）方法

长期以来，特征选择方法已经成功地应用于文本分类。特征选择可以
通过去除高达98%的独特术语，极大地提高了文本分类算法的效率，甚至在一定程度上提高了分类的准确性（Yang & Pedersen, 1997）。因此，将特征选择方法用于文本聚类任务以提高聚类性能是一个有趣的想法。为了测试我们的想法，我们进行了几个实验。从本节可以看出
4.3.1和4.3.2节，如果知道类的标签信息，有监督的特征选择方法，如CHI和IG，比无监督的特征选择方法在文本聚类方面要有效得多，不仅可以去除更多的术语，而且可以产生更好的性能。

有监督的特征选择方法不能直接应用于文本聚类，因为无法获得所需的类标签信息。幸运的是，我们发现聚类性能和特征选择可以相互加强。一方面，好的聚类结果会提供好的类标签，为每个类别选择更好的特征；另一方面，更好的特征会帮助聚类提高性能，提供更好的类标签。在EM算法的启发下，我们
提出了一种新的迭代特征选择方法，该方法
利用监督下的特征选择方法进行文本
聚类方法。

EM是一类迭代算法，用于不完整数据问题中的最大似然估计。
(Dempster et al., 1977)。假设我们有一个由带参数的模型产生的数据集，而且数据有缺失值或不可观察（隐藏）的变量。为了找到能够使似然函数最大化的参数，EM算法首先计算隐藏变量的期望值。
基于E步骤中的当前参数估计的变量。然后在M步骤中，用E步骤中计算出的预期值替换缺失值，以找到一个新的
估算能使完整的数据似然函数最大化的参数。这两个步骤被迭代到收敛。

在聚类应用中，通常假设一个文档是由一个有限的混合模型生成的，并且混合成分和集群之间存在一对一的对应关系。因此，似然函数$p(D \mid\ )$，即：

鉴于模型参数，所有文件D的概率
θ，可以写成方程（10）。

$$p(D \mid\ ) = {}^{N}\ {}^{|C|}\ p(c\ \ |p(d)\ \ \ \ | c, \theta) \quad (10)$$
$$\mathbf{1}\ \mathbf{1}\ \mathbf{1}_{i\ \ 1\ j\ 1}$$

其中，c_j是第j个簇，$|C|$是簇的数量，$p(c_j|\theta)$是簇j 的先验分布，$p(_{di}|c_j,\theta)$是簇j中文档i的分布。进一步假设，给定类标签的术语是条件独立的，那么似然函数可以改写为公式(11)。

$$p(D/\ ) = \prod_{i\ 1}^{N|C|} \sum_{j\ 1} (p(c_j|\theta) \prod_{t\ _{di}} p(t|c_j,\theta \quad (11)$$

))

其中$p(t|c_j,\ )$是术语t的术语分布，在

并非所有的术语都与文档等价相关，所以$p(t|c_j,\ )$可以被视为加权的

相关分布和不相关分布之和为公式（12）。

$$p(t|c_j,\ ) = z(t)\ p(t\ is\ relevant|c_j,\ ) + (1-z(t))\ p(t是不相关的|\ ) \quad (12)$$

其中$z(t) = p(t是相关的)$被定义为概率

术语t是相关的。此外，如果该术语不相关，则假定该术语的分布在不同的群组中是相同的，并表示为$p(t是不相关的|\ )$。因此，似然函数可以

可写成方程（13）。

$$p(D|\theta) = \prod_{i\ 1}^{N|C|} \sum_{1} (p(_{cj}|\ ) \quad (z(t)\ p(t是相关的|_{cj},\theta) + (1-z(t))\ p(t是不相关的|\ ))$$

$_{i\ _{di}}$

(13)

为了使这个似然函数最大化，EM算法可以通过迭代以下两个步骤找到一个局部最大值。

(1) E-步骤: $z^{(k\ 1)} = E(z|D,\ )$ (14)

$^{(k)})$

(2) M步骤： $^{(k\ 1)} = \arg\max_{\theta} p(D|\ ,z^{)}$ (15)

$^{(k)})$

E步对应于计算给定聚类结果的预期特征相关性，M步对应于计算一个新的最大似然。

在新的特征空间中对聚类结果进行估计。

所提出的用于文本聚类和特征的EM算法。

选择是一个一般的框架，完整的实现已经超出了本文的范围。我们的迭代特征选择方法可以装入这个框架。在E步骤中，为了近似于特征相关性的期望值，我们使用监督的特征选择算法来计算每个术语的相关性分数，然后根据术语的相关性是否被简化为$z(t) = \{0,1\}$的概率。

分数大于预定的阈值。因此，在每个迭代中，我们将根据每个术语的计算出的相关性删除一些不相关的术语。在M步骤中。

因为K-means算法可以通过稍微扩展EM算法的数学原理来描述硬阈值的情况（Bottou等人，1995），我们使用K-means聚类算法来获得基所选条款的聚类结果。

4. 实验

我们首先进行了一个理想案例实验，以证明

监管的特征选择方法，包括IG和

CHI，可以显著提高聚类性能。然后，我们评估了无监督特征选择方法的性能，包括实际案例中的DF、TS、TC和En。最后，我们评估了迭代特征选择算法。我们选择了K-means作为我们的基本聚类算法，熵值和精确值分别是

用来评估聚类性能。由于K-means聚类算法很容易受到初始中心点选择的影响，我们随机产生了10套初始中心点

每个数据集的中心点，并取10次表演的平均值作为最终的聚类性能。在进行聚类之前，使用tf*idf（采用 "ltc " 方案）来计算每个词的权重。

4.1 绩效措施

两种流行的测量方法，熵和精度。

用来评估聚类性能。

4.1.1 ENTROPY

熵衡量一个集群的均匀性或纯粹性。让

G', G分别表示获得的聚类的数量和原始类的数量。让A表示所获得的聚类中的文件集，而A的类标签是

每个文档 $_{di} \in A, i = 1,...,|A|$被表示为 $label(_{di})$。

其取值为$c_j (j = 1..., G)$。所有的熵都是

群集的熵的加权和来定义。

所有集群，如公式（16）所示。

$$熵 = \sum_{k} \frac{G'|A|}{N} \sum_{1} p_{ik} \times \log(p_{ik}) \quad (16)$$

其中

$$p_{jk} = \frac{1}{|A_k|} \| _{i} di | label(_{di} = c_j) \| \quad . \quad (17)$$

4.1.2 精确性

由于熵并不直观，我们选择另一个衡量标准来评估聚类性能，即精度。对于每个聚类，它总是由几个不同类别的文件组成。因此，我们简单地选择与该聚类中大多数文档共享的类标签作为最终的类标签。然后，每个聚类的精度被定义为：。

$$精度(A) \quad \frac{1}{|A|}\max(|\{d_i \mid label(d_i)=c_j\}|) \qquad (18)$$

为了避免精度很高的小聚类可能产生的偏差，最终的精度由所有聚类的精度加权和定义，如公式（19）所示。

$$精度=\sum_{k=1}^{G}\frac{|A_k|}{N}Precision(A_k) \qquad (19)$$

4.2 数据集

正如过去的研究工作所报告的那样（Yang & Pederen, 1997），文本分类性能在不同的数据集上有很大差异。因此，我们选择了三个不同的文本数据集来评估文本聚类的性能，包括两个标准的标记数据集。Reuters-21578[1] (Reuters), 20 Newsgroups[2] (20NG), 和一个从Open Directory Project收集的网络目录数据集（Web）[3]。路透社总共有21578个文档，但我们只选择了至少有一个主题和Lewis分割分配的文档，并在评估前给每个文档分配了第一个主题。关于这些数据集的信息如下在表1中。

*表1.*三个数据集的属性

数据集	CLASSES数。	DOCS NUM.	条款编号	平均数。条款每份文件	AVG.平均值术语
REUTERS	80	10733	18484	40.7	23.6
20NG	20	18828	91652	85.3	17.5
网络	35	5035	56399	131.9	11.8

4.3 结果和分析

4.3.1 监督下的特征选择

首先，我们进行了一个理想情况下的实验，看看好的术语是否能帮助文本聚类。也就是说，我们应用监督下的特征选择方法来选择基于类标签信息的最佳术语。然后，我们在这些选定的术语上执行了文本聚类任务，并将聚类结果与基线系统进行了比较，基线系统是在全特征空间上对文档进行聚类。

不同数据集的熵比较见图1，精度比较见图2。可以看出，在任何数据集上，至少有90%的术语可以被移除，而聚类性能要么得到改善，要么没有损失。随着更多术语的删除，路透社和20NG的聚类性能会有一些变化，但在网络目录数据集，有一个显著的性能改善。例如，在Web Directory数据集上使用CHI方法，当98%的术语被移除后，熵从2.305降低到1.870（熵相对降低18.9%），精度从52.9%提高到63.0%（精度相对提高19.1%）。当然，这些结果仅仅是在实际情况下的聚类性能，因为它是很难在没有事先的类别标签信息的情况下，选择有鉴别力的术语。

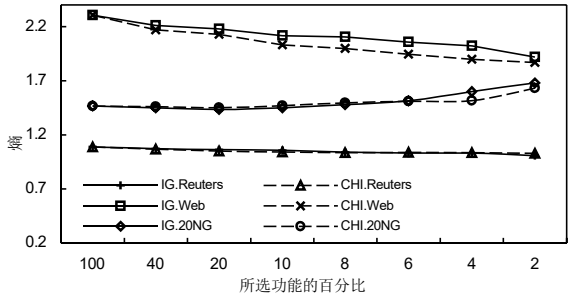


*图1.*3个数据集的熵比较（有监督的）。

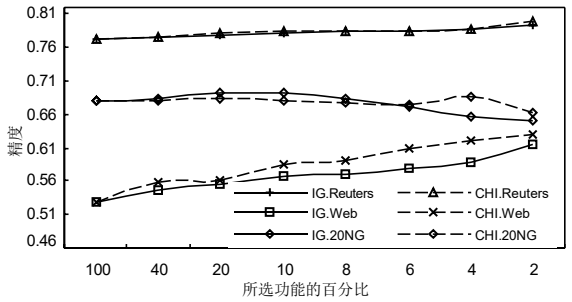


*图2.*3个数据集的精度比较（监督）。

特征选择在路透社和20NG上进展不大，而在网络目录数据集上却取得了很大的改进。这促使我们去寻找原因。在过去关于文本分类的特征选择的研究工作中，从Reuters和20NG数据集中删除一些术语后，大多数分类器，包括Naïve Bayesian分类器和KNN分类器的分类精度几乎相同（Yang & Pedersen, 1997）。我们的结论是，这两个数据集中的大多数术语仍然具有分类的鉴别价值，尽管每个类别的少数术语足以达到可接受的分类精度。换句话说，在这两个数据集中，很少有嘈杂的术语。同样，尽管特征选择可以减少特征的维度，但聚类性能却不能得到明显的改善，因为从这两个数据集中删除一些术语后，一些有用的术语也被忽略了。然而，网络目录数据则不同，其中有更多的

[1]http://www.daviddlewis.com/resources/testcollections/
[2]http://www.ai.mit.edu/people/jrennie/20Newsgroups/
[3]http://dmoz.org/

噪声术语。为了证明这一点，我们在Web Directory数据集上进行了Naïve Bayesian分类实验，发现去除98%的术语后，分类准确率从49.6%提高到57.6%。因此，当这些嘈杂的术语，如 "站点、工具、类别"，在聚类中被移除后，聚类性能也可以得到明显的改善。

4.3.2 无监督的特征选择

我们进行的第二个实验是比较无监督的特征选择方法（DF、TS、TC和En）和有监督的特征选择方法（IG和CHI）。

对路透社的熵值和精度结果见图3和图4。

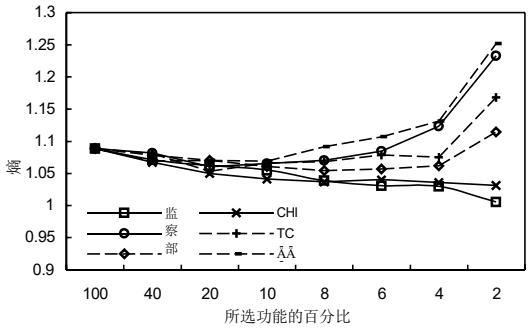


*图3.*路透社上的熵比较（无监督的）。

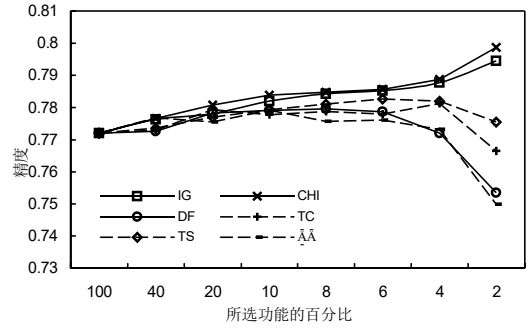


*图4.*路透社的精确度比较（无监督）。

从这些数据中，我们发现了以下几点。首先，无监督的特征选择也可以提高聚类性能，当某些术语被删除后。例如，任何无监督的特征选择方法都可以实现约2%的熵值减少和1%的精度提高，而90%的术语被删除。其次，无监督的特征选择可以与有监督的特征选择相媲美，最多能去除90%的术语。当更多的术语被移除时，有监督方法的性能仍然可以得到改善，但无监督方法的性能却迅速下降。为了找出原因，我们比较了IG和TC在不同删除阈值下选择的术语。经过分析，我们发现，在开始阶段，低

两种方法都删除了文档频率高的术语，但在下一阶段，随着更多的术语被删除，在选择术语时，术语和类之间的区分值比文档频率要重要得多。由于区分值取决于类的信息，监督方法IG可以保留那些不太常见但具有区分价值的术语，如 "OPEC, copper, drill"。然而，如果没有类别信息，TC就不能决定一个词是否具有鉴别力，仍然保留那些常见的词，如 "April, six, expect"。因此，很明显，当更多的术语被删除时，监督方法要比无监督方法好得多。

最后，与其他无监督的特征选择方法相比，TC比DF和En好，但比TS差一点。例如，如图3所示，当96%的术语被删除后，DF和En的熵值远远高于基线值（在全特征空间上），但TC和TS的熵值仍低于基线值。En与DF非常相似，因为去除常见术语比去除罕见术语更容易导致熵值的降低。TS对文档相似性阈值很敏感，β。当β恰好合适时，TS的性能就会下降。

TS比TC好。然而，阈值很难调控。此外，TS的时间复杂度（至少是$O(N^2)$）。

总比TC高（$O(MN^2)$），因为N总是比N小很多（见表1）。因此，TC是首选的方法，具有有效的性能和低计算成本。

在其余两个数据集（20NG和Web）上也可以找到类似的点（图5和图6）。由于论文长度的限制，我们只画了这两个数据集的熵值。从这些数字可以看出，在网络目录数据上也产生了最好的性能。在用TS方法去除96%的特征时，取得了大约8.5%的熵值减少和8.1%的精度提高。

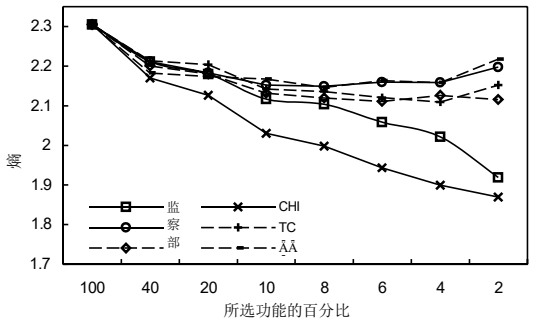


*图5.*网络目录（无监督）的熵比较

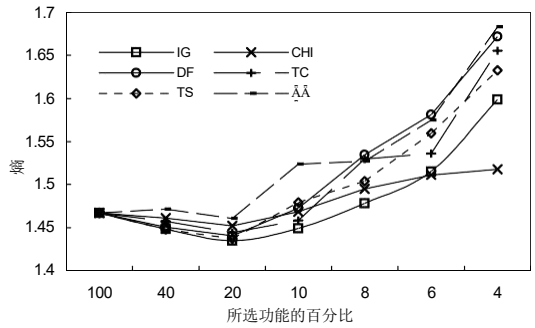


*图6.*20NG上的熵比较（无监督的）。

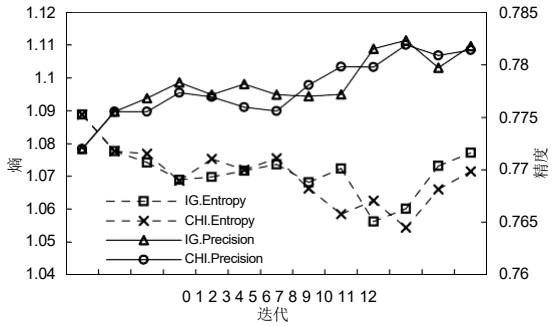


*图7.*路透社上使用IF选择的熵和精度

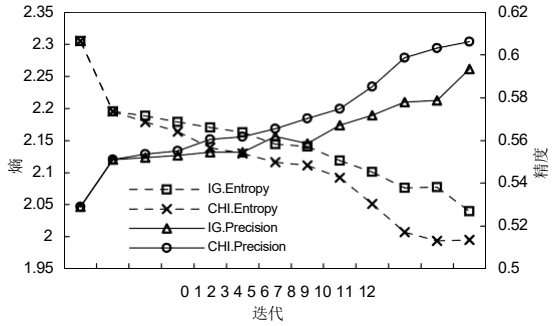


*图8.*熵和精度与网络目录上的IF选择

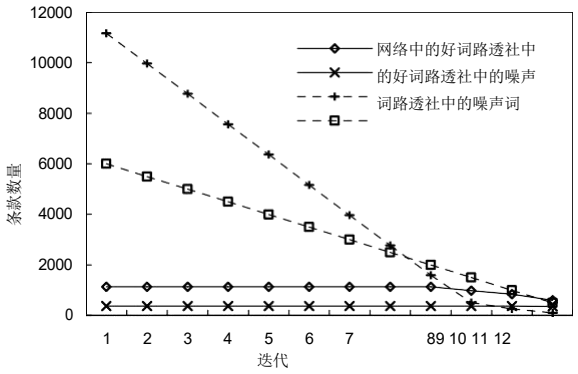


*图9.*路透社和网络目录的特征跟踪

4.3.3 迭代特征选择

我们进行的第三个实验是衡量第三节中提出的迭代特征选择算法。由于IG和CHI在理想情况下有很好的性能，所以在迭代特征选择实验中选择了它们。为了加快迭代过程，我们在聚类前删除了文档频率低于3的术语。然后，在每次迭代中，将IG或CHI输出的排名最低的10%的术语（当剩下的术语少于10%时，则为3%的术语）剔除。图7

显示了路透社的熵（虚线）和精度（实线）结果。图8显示的是网络目录数据集的结果。由于页面的限制，我们没有显示20NG的结果。

从图7和图8可以看出，迭代特征选择的性能相当好。它非常接近于理想情况，比任何无监督的特征选择方法都要好。例如，在Web Directory数据集上用CHI选择法进行了11次迭代后（去除近98%的术语），熵从2.305降到1.994（熵相对减少13.5%），精度从52.9%提高到60.6%（精度相对提高14.6%）。这接近于理想情况下的上限（熵减少18.9%，精度提高19.1%，见4.3.1节）。

为了验证迭代特征选择的有效性，我们追踪了每次迭代中被过滤掉的术语。首先，我们假设在理想情况下，CHI排名前2%的术语是好的术语，而其他的是噪声术语。然后，我们计算了每次迭代中被保留的"好"词的数量。从图9可以看出，在10次迭代后，"好"词（两条底线）几乎被保留下来，而大多数噪声词（两条顶线）被过滤掉了。

5. 结论

在本文中，我们首先证明了在理想情况下，特征选择可以提高文本聚类的效率和性能，即根据类的信息来选择特征。但在现实情况下，类别信息是未知的，所以只能利用无监督的特征选择。在很多情况下，无监督的特征选择比有监督的特征选择要差得多，不仅它们能去除的术语少，而且它们产生的聚类性能也差得多。为了利用有效的监督方法，我们提出了一种迭代特征选择方法，在一个统一的框架内迭代地进行聚类和特征选择。研究发现，其

其性能接近理想情况，比任何无监督的方法都要好。我们做的另一项工作是对几种无监督的特征选择方法进行了比较研究，包括DF、TS、En和一种新提出的方法TC。结果发现，TS和TC比DF和En更好。由于TS的计算复杂度很高，而且很难调整其参数，所以TC是文本聚类的首选无监督特征选择方法。

参考文献

Aggrawal, C.C., & Yu, P.S. (2000).寻找高维空间中的广义投影集群.*Proc. of SIGMOD'00* (pp. 70-81).

Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2001).On Feature Distributional Clustering for Text Categorization.*Proc. of SIGIR'01* (pp. 146-153).

Blum, A. L., & Langley, P. (1997).机器学习中相关特征和例子的选择。*人工智能，1（2）*，245-271。

Bottou L., & Bengio Y. (1995).k-means算法的收敛特性.*Advances in Neural Information Processing Systems,* 7, 585-592.

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey.W. (1992).Scatter/Gather:A Cluster-based Approach to Browsing Large Document Collections.*Proc. of SIGIR'92* (pp. 318-329).

Dash, M., & Liu, H. (1997). 分类的特征选择。*International Journal of Intelligent Data Analysis, 1(3),* 131-156.

Dash, M., & Liu, H. (2000). 聚类的特征选择.*Proc. of PAKDD-00* (pp. 110-121).

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977).通过EM算法的不完全数据的最大似然。*Journal of the Royal Stat.Society,39,* 1-38.

Friedman, J.H. (1987).探索性投射追求。*Journal of American Stat.协会，82*，249-266。

Galavotti, L., Sebastiani, F., & Simi, M. (2000).自动文本分类中的特征选择和负面证据。*Proc. of KDD-00*.

Jain, A.K., Duin P.W., & Jianchang, M. (2000).统计模式识别：一个回顾。*IEEE Transactions on Pattern Analysis and Machine Intelligence, 22,* 4-37.

Jolliffe, I.T. (1986).*Principal Component Analysis*.Springer Series in Statistics.

Koller, D., & Sahami, M. (1996).Toward Optimal Feature Selection.*Proc. of ICML'96* (pp.284-292).

Koller, D., & Sahami, M. (1997).使用极少的词对文件进行分层分类。*Proc. of ICML-97* (pp. 170-178).

Kowalski, G. (1997).*Information Retrieval Systems Theory and Implementation*.Kluwer Academic Publishers.

Martin, H. C. L., Mario, A. T. F., & Jain, A. K (2002).*无监督学习中的特征显著性*（技术报告2002）。Michigan State University.

Salton, G. (1989).*自动文本处理。通过计算机进行信息的转换、分析和检索*。Addison-wesley, Reading, Pennsylvania.

Slonim, N., & Tishby, N. (2000).通过信息瓶颈法使用词簇进行文档聚类。*Proc. of SIGIR'00* (pp. 208-215).

Wilbur, J.W., & Sirotkin, K. (1992).停顿词的自动识别。*信息科学杂志，18*，45-55。

Wyse, N., Dubes, R., & Jain, A.K. (1980).A critical evaluation of intrinsic dimensionality algorithms.*Pattern Recognition in Practice* (pp. 415-425).North-Holland.

Yang, Y. (1995).文本分类的统计方法中的降噪。*Proc. of SIGIR'95* (pp. 256-263).

Yang, Y., & Pedersen, J. O. (1997).A comparative study on feature selection in text categorization.*Proc. of ICML- 97* (pp. 412-420).

Zamir, O., Etzioni, O., Madani, O., & Karp, R. M. (1997). 快速和直观的网络文档聚类。*Proc. of KDD-97* (pp. 287-290).