一种改进K-means算法的聚类算法CARDBK

朱烨行[1] 李艳玲[2] 崔梦天[3,4] 杨献文[5]

(西安邮电大学经济与管理学院 西安 710121)[1] (第二炮兵工程大学电子工程系 西安 710025)[2]

(西南民族大学计算机科学与技术学院 成都 610041)[3]

(电子科技大学计算机科学与工程学院 成都 610000)[4] (西安财经学院信息与教育技术中心 西安 710061)[5]

摘 要 CARDBK聚类算法与批K-means算法的不同之处在于,每个点不是只归属于一个簇,而是同时影响多个簇的质心值,一个点影响某一个簇的质心值的程度取决于该点与其它离该点更近的簇的质心之间的距离值。从聚类结果的熵、纯度、F1值、Rand Index和NMI等5个性能指标值来看,与多个不同算法在多个不同数据集上分别聚类相比,该算法具有较好的聚类结果;与多个不同算法在同一数据集上很多不同的初始化条件下分别聚类相比,该算法具有较好且稳定的聚类结果;该算法在不同大小数据集上聚类时具有线性伸缩性且速度较快。

关键词 聚类,文档聚类,文本聚类,K-means,算法

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.3.041

Clustering Algorithm CARDBK Improved from K-means Algorithm

ZHU Ye-hang[1] LI Yan-ling[2] CUI Meng-tian[3,4] YANG Xian-wen[5]

(School of Economics and Management, Xi'an University of Posts and Telecommunications, Xi'an 710121, China)[1]

(Department of Electronic Engineering, The Second Artillery Engineering University, Xi'an 710025, China)[2]

(School of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610041, China)[3]

(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610000, China)[4]

(Information and Educational Technology Center, Xi'an University of Finance and Economics, Xi'an 710061, China)[5]

Abstract The difference between our clustering algorithm and batch K-means algorithm is that in our algorithm each point is not only attributable to one cluster, instead affects multiple cluster centroid values, and the degree of influence of a point on a cluster centroid depends on the distance values between this point and the other more near cluster centroids. Our algorithm and a number of different algorithms on a number of different data sets were clustered respectively from the point of view of their clustering result's five performance index values such as entropy, purity, F1 value, Rand Index and normalized mutual information, and the results show our algorithm has a better clustering results. Our algorithm and a number of different algorithms were clustered respectively on one same data set but under many different initialization conditions, and clustering results of our algorithm are preferably more stable and better. Cluster on different size data sets by our algorithm has a linear scalability and is faster.

Keywords Clustering, Text clustering, Document clustering, K-means, Algorithm

1 引言

本文提出一种新的聚类算法——CARDBK算法,它是结合CARD算法和批K-means(Batch K-means)算法而形成的聚类算法。其中CARD算法是本文作者提出的一种通过改进Neural Gas算法形成的聚类算法[1],而批K-means算法是K-means算法中的一种计算方法[1]。

硬竞争学习只对赢的一个单元进行修改,而软竞争学习不仅修改赢的一个单元,还要修改靠近赢的多个单元[2]。“硬竞争学习,即赢者全取学习(winner-take-all learning)”[2]的特点是每个输入点只改变一个质心,如K-means算法就是如此。这类算法可能常出现“死单元”(dead units)[3],即除了初始值外没有任何一个点属于这些被称为“死单元”的质心,这通常是由不合适的初始化引起的,是有害的,因为它们对数据分析没有帮助却占用资源[4];还有一个问题是不同的初始化会产生非常不同的结果,局部调整不能使聚类结果走出不良的局部最优点[5]。解决这些问题的一个办法是“改变赢者全取(winner-take-all)为赢者多取(winner-take-most),即采用软

到稿日期:2014-04-09 返修日期:2014-06-20 本文受国家自然科学基金(61379019,71102149),中国博士后科学基金(2013M540704),四川省学术和技术带头人培养资金,四川省博士后科研基金资助。

朱烨行(1969—),男,博士,讲师,主要研究领域为数据挖掘,E-mail:zhuyehang@126.com;李艳玲(1971—),女,博士,副教授,主要研究领域为数据挖掘;崔梦天(1972—),女,博士后,教授,主要研究领域为可信软件、算法,E-mail:happyzg3@163.com(通信作者);杨献文(1967—),男,工程师,主要研究领域为数据挖掘。

竞争学习方法"[6]，不仅赢的一个质心要改变，别的未赢的但靠近赢的质心也要相应地改变，如神经气(Neural Gas)算法通过邻域协作，减少不良的局部最优发生的可能性[6]。

2 CARDBK 聚类算法原理

与批 K-means 算法一样，本文算法也包括两个步骤，这两步反复迭代直到满足停止的条件，但与批 K-means 算法不同，每个点 X 不是只归属于一个簇，只影响一个簇的质心，而是像 Neural Gas 算法一样，每个点 X 同时影响多个簇的质心[7]，与 Neural Gas 算法不同的是，影响某个簇质心的大小程度取决于该点 X 与其它离该点更近的簇的质心的距离值的大小。这可以从文献[8]中找到理论依据：一个点与赢得它的最佳质心匹配程度越差，该点对其它非赢的质心的影响程度越大；一个点与赢得它的最佳质心匹配程度越好，该点对其它非赢的质心的影响程度越小。

本文算法运行过程中反复用到了一个数据结构：数组 Xishu，我们称之为系数矩阵，它的行代表质心，列代表各个点，如图 1 所示。

	点 D_1	点 D_2	…	点 D_m	…	点 D_M
质心 C_1	Xishu[1][1],	Xishu[1][2],	…,	Xishu[1][m],	…,	Xishu[1][M]
质心 C_2	Xishu[2][1],	Xishu[2][2],	…,	Xishu[2][m],	…,	Xishu[2][M]
…	…	…	…	…	…	
质心 C_n	Xishu[n][1],	Xishu[n][2],	…,	Xishu[n][m],	…,	Xishu[n][M]
…	…	…	…	…	…	
质心 C_N	Xishu[N][1],	Xishu[N][2],	…,	Xishu[N][m],	…,	Xishu[N][M]

图 1　系数矩阵 Xishu[N][M]结构示意图

本文算法总的聚类过程可以描述如下[9](改编自：An Introduction to Cluster Analysis for Data Mining)：

输入：点集合 $D=\{D_1, D_2, \cdots, D_M\}$，其中 M 为点集合中点的总个数；最大迭代次数：$NUM_{MAX}$；簇个数：N。

1. 初始化质心集合 $C=\{C_1, C_2, \cdots, C_N\}$；

2. for i=1 to NUM_{MAX}(NUM_{MAX}是最大迭代次数)；

2.0 将系数矩阵 Xishu[N][M]全部置为零，这里的 N 为所有的质心总数，M 为所有的点的总数；

2.1 把每个点分配给离其最近的几个质心(在本文中取 5 个质心)；

2.1.1 取一个点 m，计算点 m 与每个质心的距离；

2.1.2 对步骤 2.1.1 获得的 N 个距离由小到大进行排序；C_{d1} 为离点 m 最近的质心，点 m 与其之间距离为 d_1；C_{d2} 为离点 m 次近的质心，点 m 与其之间距离为 d_2；质心 C_{d3}、C_{d4}、C_{d5} 依次为离点 m 距离由小到大排名 3、4 和 5 的质心，这里 $d_1 \leqslant d_2 \leqslant d_3 \leqslant d_4 \leqslant d_5$。

2.1.3　按如下公式计算系数矩阵 Xishu[][m]中 m 列的值：

$$\begin{aligned}
&Xishu[C_{d1}][m]=1.0; && (1a)\\
&Xishu[C_{d2}][m]=(d_1)^P; && (1b)\\
&Xishu[C_{d3}][m]=(d_1 * d_2)^P; && (1c)\\
&Xishu[C_{d4}][m]=(d_1 * d_2 * d_3)^P; && (1d)\\
&Xishu[C_{d5}][m]=(d_1 * d_2 * d_3 * d_4)^P; && (1e)
\end{aligned} \quad (1)$$

即在系数矩阵 Xishu[][m]的 m 列中，质心 C_{d1} 所对应的行的值为 1.0，质心 C_{d2} 所对应的行的值为 $(d_1)^P$，质心 C_{d3} 所对应的行的值为 $(d_1 * d_2)^P$，质心 C_{d4} 所对应的行的值为 $(d_1 * d_2 * d_3)^P$，质心 C_{d5} 所对应的行的值为 $(d_1 * d_2 * d_3 * d_4)^P$；这里指数 P 常取值为 3，若结果不理想，可尝试其它值：1、2 或 4 等。系数矩阵 Xishu[][m]中 m 列的其它行的元素的值保持为零。

2.2 重新计算每个簇的质心。

2.2.1 对质心 Cn，$n \in \{1, 2, \cdots, N\}$，取步骤 2.1 计算所得的当前系数矩阵 Xishu[N][M]中质心 Cn 对应的行的元素值，分别为 Xishu[Cn][1]，Xishu[Cn][2]，…，Xishu[Cn][M]；M 为所有的点的总数。

2.2.2 设置当前质心 Cn 值对应的向量 Value[]为零；然后按下式计算：

```
Value[]=0;
for i=1 to M
    Value[] += Xishu[Cn][i] * Di;                    (2)
end for i
```

即把点 D_i 的值乘以步骤 2.2.1 中相对应的系数值 Xishu[Cn][i]，加到向量 Value[]上。也就是说，当前质心 Cn 值等于每个点 i 的值 D_i 分别乘以步骤 2.2.1 中相对应的系数值 Xishu[Cn][i]的 M 个乘积的总和。

2.2.3 将步骤 2.2.1 中系数矩阵 Xishu[Cn][]的 Cn 行的所有元素求和，值为 Sum，用它去除步骤 2.2.2 所得的向量值 Value[]，(Value[] / Sum)即为当前质心 Cn 的值。

如果是进行文档聚类，本步骤可更改为对步骤2.2.2中的向量 Value[]值进行归一化处理。

2.3 如果满足式(3)所示的停止条件，则跳出循环，结束运行；否则继续迭代，直到一开始设置的最大迭代次数 NUM_{MAX}为止。

在步骤 2.1 中求出每个点与所有质心的距离的最小值，这样的最小值总共有 M 个(因为总共有 M 个点)，把这 M 个值相加，记为 Quant；上一次迭代循环时记录的该值为 FirstQuant。

$$\frac{|FirstQuant-Quant|}{Quant}<\varepsilon \quad (3)$$

即连续两次循环所得的 Quant 值之差的绝对值，与 Quant 值之比若小于一个很小的数 ε，则认为可以停止运行了。这里 ε 为一个很小的数，例如可以取值为 0.00000001。

end for i(对应于 for 循环，最大迭代次数是 NUM_{MAX})。

3. 输出：最终的质心值 $C'=\{C_1', C_2', \cdots, C_N'\}$。

通过实验发现，本算法步骤 1 即初始化质心集合采用如下方法能取得更好结果：随机选定 N 个点作为最初的 N 个质心，把所有 M 个点按距离远近划归相应的质心，形成相应的簇，计算这样形成的簇的新的质心，把这些新形成的质心作为步骤 1 的输出，用于以后的计算[9]。

当处理文档集时，我们实际上先求的是质心和点(这时是文档)之间的余弦相似度 Sim，然后用值 1.0(余弦相似度的最大值)减去 Sim 作为二者之间的距离来处理。即[9]：

$$d=1.0-Sim+0.01 \quad (4)$$

之所以要加值 0.01，是因为要避免距离值 d 为零，实验证明这样处理效果好[9]。

本算法运行过程中只有一个参数需要进行调整，即系数矩阵中的系数计算过程中用到的指数 P，根据不同的数据集设置不同的 P 值，如对 NewsGroup 数据集选 $P=3$ 最好，而对 Reuter 数据集选 $P=1$ 最好。在应用中可以尝试选 $P=1$、2、3、4 这 4 个值，选择结果最好的那个 P 值，而 P 一经确定，在该类型数据集以后的应用中就不再改变。以后如果不加说明，本算法默认为 $P=3$，这时的 CARDBK 算法在本文中被称为 MY3 算法或 CARDBK-3 算法；当 $P=1$ 时，CARDBK 算法在本文中被称为 MY1 算法或 CARDBK-1 算法。

3 实验结果

我们首先在广泛的不同来源的数据集(如 Reuters-

21578、NewsGroup、WAP、New3、Re0、Hitech、Ohscal、K1a、Tr12 等数据集)上比较 CARDBK 算法和其它聚类算法,然后在单一的数据集 WAP 上采用各种不同的初始化方法比较 CARDBK 算法和其它聚类算法。前者用以证实 CARDBK 算法适应面广,在各种不同来源的数据集上都表现优异;后者用以证实 CARDBK 算法稳定,在不同初始化条件下都性能优良。在不同大小的数据集上聚类,实验证实本文聚类算法速度较快,具有线性可伸缩性。

3.1 参与比较的算法

所有实验在微机上运行,CPU 为 2.8G 的英特奔腾芯片,内存 512M,操作系统为 Windows XP。参与比较的算法有以下 6 个算法。

1) MATLAB:大型仿真软件 MATLAB 中提供的 K-means 算法;

2)CLUTO:软件包 CLUTO 中提供的 K-means 算法;

3)K-means-SB:球形批 K-means 算法,根据算法原理自己编程实现;

4)K-means-SO:球形即时 K-means 算法,根据算法原理自己编程实现;

5)NG only:NG 是 Neural Gas 算法的首字母缩写词,而"only"一词表示只是一个点中的非零值所在的那些维,而不是全部维上的值参与了更新质心的值的运算,根据算法原理我们自己编程实现。

6)NG all:NG 同 5),NG all 算法与 NG only 算法的唯一区别是,在修改与点 X 最相似的若干个质心的向量的值时,用当前点 X 的向量中所有维(包括值为零的维)上的值减去质心向量相应维上的值,它们的差用来对该质心的向量值进行修改[6],根据算法原理我们自己编程实现。

之所以自己编程实现这些算法,是因为其中有些算法对初始化敏感,不同的初始化方法聚类结果质量相差很大,因此在用它们与本算法进行比较时要使用同样的初始化,这样才合理,另外应在相同的编程和运行环境下比较各算法的速度。当然本文也用了一些现成的程序和算法如大型仿真软件 MATLAB 中的 K-means 算法、CLUTO 软件包中的聚类算法。

3.2 评价指标

本文用于评价聚类算法性能的指标分别是:熵(Entropy)、纯度(Purity)、F1 值(F1 measure)、Rand index 值(Rand)和标准化的互信息(NMI)等共 5 个指标。

3.3 数据集

要比较算法,就要在相同的数据集上运行各种算法,为此就需要一些数据集。为了防止某些算法在某类数据集上效果好而在其它类型数据集上很差,就需要多一些不同来源的数据集。为此,给出了 3 种不同来源的数据集,第一种是原始的文本文件数据集,我们自己对其加工处理,然后使用;第二种是别人已加工处理过的稀疏矩阵形式保存的数据集;第三种是对别人加工处理过的数据集 WAP 进行扩展形成不同大小的数据集。

从原始文本数据集 Reuters-21578 和 20_Newsgroups 中分别产生了 5 个稀疏矩阵形式保存的数据集 ReuterOne1、ReuterOne3、ReuTitle5、GroupOne、GroupSub,如表 1 所列。同时也采用了别人加工处理形成的 7 个稀疏矩阵形式保存的数据集:WAP、Tr12、Re0、Hitech、Re1、Ohscal、K1a 等[10],如表 2 所列。

表 1 用于评价各算法性能的数据集(自己编程生成)汇总

数据集	来源	文档个数	包含词数	实际类数
ReuterOne1	Reuters-21578	9494	20163	66
ReuterOne3	Reuters-21578	9494	7613	66
ReuTitle5	Reuters-21578	2334	9433	47
GroupOne	20_newsgroups	2000	24461	20
GroupSub	20_newsgroups	2000	24563	20

表 2 用于评价各算法性能的数据集(别人已生成)汇总

数据集	来源	文档个数	包含词数	实际类数
WAP	WebACE	1560	8460	20
Tr12	TREC	313	5804	8
Re0	Reuters-21578	1504	2886	13
Hitech	San Jose Mercury	2301	126373	6
Re1	Reuters-21578	1657	3758	25
Ohscal	OHSUMED	11162	11465	10
K1a	WebACE	2340	21839	20

为了证实我们的算法具有线性可伸缩性,同时也为了比较在不同大小的数据集上各算法聚类所用时间,需要构造不同大小的数据集。数据集 WAP 所含文档数为 1560[11],我们把数据集 WAP 中的每篇文档再重复一遍,得到一个大小为 3120 的文档集 d2WAP;依此方法扩大形成了 8 个不同大小的数据集[12]。

把我们的算法与软件 MATLAB 中的 K-means 算法、CLUTO 软件中的 K-means 聚类算法、球形批 K-means 算法、球形即时 K-means 算法、两种神经气(Neural Gas)算法等共 7 种聚类算法在 10 个数据集(包括 ReuterOne3、ReuTitle5、GroupOne、GroupSub、WAP、Re0、Hitech、Re1、Ohscal、K1a 等)上分别聚类,计算聚类结果的熵(Entropy)、纯度(Purity)、F1 值、Rand Index 和 NMI 等 5 个性能指标值来对这些聚类算法作比较,然后进行统计分析,进一步证实了本文算法质量优良[13]。

本文算法在不同的初始化条件下聚类结果有起伏。为此在数据集 WAP 上设计了 3 种共计 42 个不同的初始化方式,把本文算法和软件 MATLAB 中的 K-means 算法、球形批 K-means 算法、球形即时 K-means 算法、两种神经气(Neural Gas)算法等共 6 种聚类算法分别以 42 种初始化方式在数据集 WAP 上分别聚类,计算聚类结果的熵(Entropy)、纯度(Purity)、F1 值、Rand Index 和 NMI 等 5 个性能指标值,对这些聚类算法作比较。实验数据表明,本文算法在多数情况下的性能指标优于其它各算法,说明本文算法在不同初始化情况下性能优良稳定[14]。

我们的文档聚类算法在 8 个不同大小(数据集的大小从一千多个文档依次增加到一万多个文档)的数据集上聚类,所得到的实验数据表明本文算法聚类速度较快,具有线性可伸缩性[15]。

3.4 各算法在不同来源数据集上聚类结果的质量分析比较

我们在不同来源的文档数据集上聚类时采用最小最大(MaxMin)[16]初始化方法。

在下列表中,性能处于前 3 名的用黑体字表示,最好的加下划线。

表 3　各聚类算法在数据集 GroupSub 上聚类为 20 个簇时的性能指标值比较

GroupSub (3/55)20	熵	纯度	F1	Rand	NMI
CARDBK-3	0.38955	0.5885	0.57819	0.944058	0.61044
CARDBK-1	0.445145	0.491	0.512009	0.934148	0.554855
MATLAB	0.39483	0.5755	0.596595	0.942521	0.60517
CLUTO	0.409714	0.5805	0.57	0.94193	0.590286
K-means-SB	0.495413	0.505	0.524786	0.930599	0.504587
K-means-SO	0.39584	0.579	0.594029	0.94213	0.60415
NG only	0.42137	0.5455	0.557956	0.938656	0.578623
NG all	0.42022	0.5595	0.562429	0.937795	0.579773

表 4　各聚类算法在 10 个数据集上聚类结果在 5 个评价指标上的均值的比较

算法	熵	纯度	F1	Rand	NMI
CARDBK-3	0.354166	0.66822	0.495851	0.856563	0.468739
CARDBK-1	0.380822	0.629394	0.456094	0.84893	0.442083
MATLAB	0.368682	0.651837	0.509405	0.855081	0.454224
CLUTO	0.36441	0.664955	0.485223	0.853803	0.458495
K-means-SB	0.399099	0.624138	0.487314	0.852141	0.423807
K-means-SO	0.367814	0.650516	0.510572	0.85509	0.455092
NG only	0.37681	0.645511	0.462292	0.848429	0.446095
NG all	0.383	0.636204	0.455354	0.845647	0.439905

图 2 为各聚类算法在 10 个数据集上聚类结果的熵值的平均值的比较，可以看出算法 CARDBK-3 的熵值最小，而熵值越小越好，说明本文算法好。

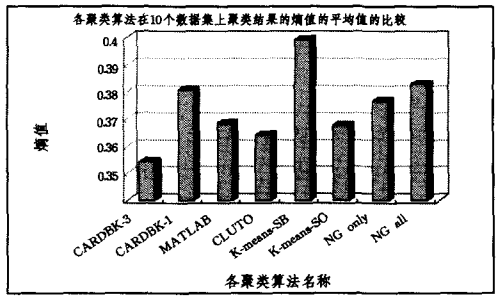

图 2　各聚类算法在 10 个数据集上聚类结果的熵值的平均值的比较

在广泛的不同来源的数据集（如 Reuters-21578、News-Group、WAP、Re0、Hitech、Ohscal、K1a、Tr12 等数据集）上比较本文算法和其它算法得出：从聚类结果的熵值、纯度值、F1 值、Rand Index 值和 NMI 值这 5 个衡量指标来看，本文算法是一种性能优良的聚类算法，在大多数情况下优于文中所述其它几个聚类算法，即本文算法优于下列算法：K-means-SO 算法、MATLAB 中的 K-means 算法、CLUTO 软件包中的参数为 Direct I_2 的聚类算法、NG only 算法、K-means-SB 算法、NG all 算法等[17]。

因此本文算法可以应用于各种不同环境下的文档聚类。

3.5　各算法在 WAP 数据集上不同初始化时聚类结果质量的比较

下面在单一的数据集 WAP 上采用各种不同的初始化方法比较本文算法和其它算法，用以证实本文算法的稳定性，即在不同初始化条件下都性能优良。分 3 种类型的初始化分别考虑：一种是从数据集中进行取样的初始化方式，取 20 个这种初始化方法；另一种是把数据集进行划分的初始化方式，取 20 个这种初始化方法；还有一种是把每个文档按随机产生的权重大小赋给每个不同的簇的初始化方式[13]，取 2 个这种初始化方法。

表 5 列出各算法在 WAP 数据集上采用上述 42 种初始化时聚类结果在 5 个衡量评价标准（分别为熵、纯度、F1、Rand Index 和 NMI）上的值的平均。

表 5　各算法在 WAP 数据集上采用 42 种初始化方法时的聚类结果在 5 个评价指标上的均值的比较

算法	熵	纯度	F1	Rand	NMI
CARDBK-3	0.329558	0.668925	0.513938	0.904287	0.531607
CARDBK-1	0.335018	0.655876	0.501877	0.900901	0.526147
MATLAB	0.343382	0.660241	0.522587	0.901644	0.517783
K-means-SB	0.417905	0.587271	0.472511	0.892899	0.443261
K-means-SO	0.343193	0.663767	0.523857	0.90268	0.517972
NG only	0.33423	0.650962	0.491507	0.899323	0.526935
NG all	0.3331	0.65522	0.492449	0.899696	0.528066

由于我们无法控制 CLUTO 软件包中的聚类算法[11]的初始化方式，这里将本文聚类算法与其它算法做比较时就不作考虑。

图 3 展示 6 个算法在 WAP 数据集上 42 种初始化方法下聚类结果的熵值的分布情况。可以看出，CARDBK-3 算法的熵值的中位数值最小，与其它算法相比，整体分布值更靠下方，下部的 hinge 即盒子的下沿值也是最小的，而熵值越小说明算法越好，从而说明 CARDBK-3 算法在多数情况下好于其它各算法[18]。

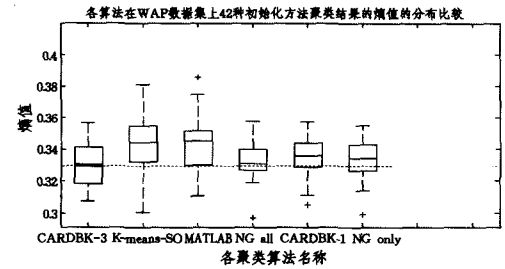

图 3　各算法在 WAP 数据集上采用 42 种初始化方法聚类结果的熵值的分布比较

通过在各种不同初始化方式下在 WAP 数据集上比较本文算法和其它各算法，得出本文算法是一种性能稳定优良的聚类算法，在多数情况下本文所提算法的性能指标优于其它各种算法，特别是在熵值、纯度值、Rand Index 值和 NMI 值这 4 个衡量评价指标上的表现更好[19]。

通过实验验证本文算法具有很好的聚类效果，它不仅在各种不同来源的数据集上表现出色，而且在多种不同初始化条件下在数据集 WAP 上聚类的性能稳定。

3.6　CARDBK 聚类算法所用时间实验分析

下面将通过实验证实本文所提聚类算法具有线性可伸缩性，同时将其与其它几个算法的聚类时间进行比较。

由于 MATLAB 中的 K-means 算法的程序的编程语言是 MATLAB 语言，MATLAB 是解释性语言，执行速度较慢[18]；CLUTO 软件中的参数为 Direct I_2，聚类算法本身是该软件包提供的可执行程序[11]；而其它各算法都是我们采用 Visual C++6.0 环境下的 C 语言编写的。C 语言是编绎型语言，运行速度很快，因为实现方式不同，所以下面在比较各算法的速度的时候就不比较 MATLAB 中的 K-means 算法，因为运行环境不同，比较它们是不公平的[18]。

NG all 算法和 NG only 算法在循环次数为 50 时，在数据集 WAP 上聚类结果的熵值、纯度值和 NMI 值要好于本文算法。为公平起见，让它们在与本文算法循环迭代次数相同的情况下进行比较，在比较时间时，我们把与本文算法循环迭代

次数相同的 NG all 算法和 NG only 算法分别表示为 NG all 20 和 NG only 20，因为当初始化方式用的是最小最大初始化方法时，本文算法在 WAP 数据集上聚类时的循环迭代次数为 20，在 d2WAP 到 d8WAP 这 7 个数据集上循环迭代次数同样也为 20，"20"这个后缀就来缘于此。循环迭代次数为 50 次的 NG all 算法和 NG only 算法被表示为 NG all 50 算法和 NG only 50 算法，以示区别[18]。

表 6 记录了各种聚类算法在 8 个不同大小(从 1560 到 12480)数据集上聚类时所用的时间，它们的初始化方式用的都是最小最大初始化方法(即 MaxMin 方法)[19]。

表 6 各算法在不同大小数据集上聚类所用时间对比(单位：秒)

算法名称	WAP	d2WAP	d3WAP	d4WAP	d5WAP	d6WAP	d7WAP	d8WAP
CARDBK-3	3.87	7.44	10.8	14.9	17.8	21.1	25.9	29.4
CARDBK-1	4.21	7.63	11.5	15	18.5	22.3	26.2	30.9
CLUTO	2.78	4.52	5.12	6	8.38	9.13	11.8	12.4
K-means-SB	2.64	4.52	5.81	7.67	9.44	11.3	13.6	15.6
K-means-SO	2.03	4.42	5.03	7.44	8.69	9.59	11.4	15.5
NG only 50	74.9	150.	226.	319.	430	574.	720.	871.
NG all 50	119.	240.	366.	502.	715.	976.	1188	1444
NG only 20	35.0	69.5	103.	138.	176.	217.	261.	312.
NG all 20	55.2	110.	165.	220.	280.	350.	420.	511.

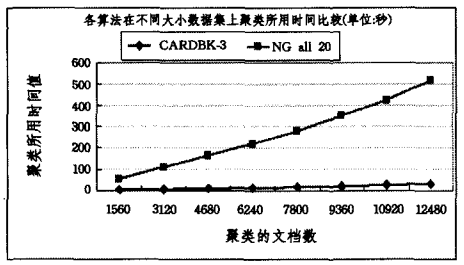

图 4 CARDBK-3 算法和 NG all 20 算法在不同大小数据集上聚类所用时间比较

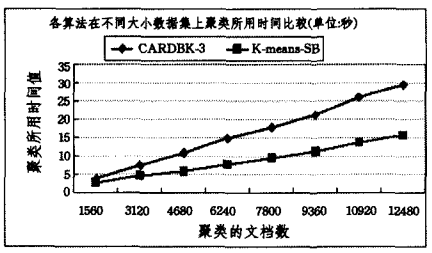

图 5 算法 CARDBK-3 和 K-means-SB 算法在不同大小数据集上聚类所用时间比较

由图 4、图 5 可见，本文算法具有随着聚类文档数增加聚类时间也线性增加的特点，因而具有线性可伸缩性[20]。同时发现本文算法速度较快。

结束语 本文提出了一种聚类算法，并通过实验验证了其具有较好的聚类效果和效率。它不仅在各种不同来源的数据集上表现出色，而且在不同的初始化条件下在数据集 WAP 上聚类的性能稳定，并且聚类所用时间较少，具有线性可伸缩性[19]。

参 考 文 献

[1] 朱烨行. 文档聚类算法研究[D]. 西安：西北工业大学，2009

[2] Zhao Ying, Karypis G. Criterion functions for document clustering: Experiments and analysis[R/OL]. 2003-04-23[2008-10-29]. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

[3] Anon. an Introduction to Cluster Analysis for Data Mining[EB/OL]. 2000-02-10 [2008-12-2]. http://www.dol88.com/p-567183494975.html

[4] 刘泉凤，陆蓓，王小华. 文本挖掘中聚类算法的比较研究[J]. 计算机时代，2005(6)：7-8，22

[5] 谷波，张永奎. 文本聚类算法的分析与比较[J]. 电脑开发与应用，2003，16(11)：4-6

[6] Bernd F. Some Competitive Learning Methods [R/OL]. 1997-04-05[2008-10-22]. http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/

[7] Ridella S, Rovetta S, Zunino R. Plastic Algorithm for Adaptive Vector Quantisation[J]. Neural Computing & Applications, 1998, 7(1): 37-51

[8] Pal N R, Bezdek J C, Tsao E C K. Generalized Clustering Networks and Kohonen's Self-Organizing Scheme[J]. IEEE Transaction on Neural Networks, 1993, 4(4): 549-557

[9] Hansen P, Mladenovic N. J-Means: A New Local Search Heuristic for Minimum Sum-of-Squares Clustering[J]. Pattern Recognition, 2001, 34(2): 405-413

[10] 唐春生，张磊，潘东，等. 文本分类研究进展[EB/OL]. [2008-10-24]. http://c.xml.org.cn/blog/uploadfile/20076211443809.PDF

[11] Karypis G. CLUTO- Software for Clustering High-Dimensional Datasets[CP/OL]. 2008[2008-10-25]. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

[12] Han E H, Boley D, Gini M, et al. WebACE: A webagent for document categorization and exploration[C]// Proceedings of the Second International Conference on Autonomous Agents. Minneapolis, Minnesota, United States: ACM, 1998: 408-415

[13] Beil F, Ester M, Xu Xiao-wei. Frequent term-based text clustering[C]// Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2002: 436-442

[14] Karypis G, Han Eui-hong. Fast supervised dimensionality reduction algorithm with applications to document categorization & retrieval[C] // Proceedings of the Ninth International Conference on Information and Knowledge Management. New York: ACM, 2000: 12-19

[15] Hersh W, Buckley C, Leone T, et al. OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research[C]// Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland: ACM, 1994: 192-201

[16] Juan A, Vidal E. Comparison of Four Initialization Techniques for the K-Medians Clustering Algorithm[C]// Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition. London, UK: Springer-Verlag, 2000: 842-852

[17] 梁冯珍，宋占杰，张玉环. 应用概率统计[M]. 天津：天津大学出版社，2004

[18] Hanselman D, Littlefield B. 精通 MATLAB：综合辅导与揿[M]. 李人厚，张平安，译. 西安：西安交通大学出版社，1998

[19] Velleman P F, Hoaglin D C. Applications, Basics, and Computing of Exploratory Data Analysis[M]. Boston: Duxbury Press, c2004

[20] Macqueen J. Some methods of classification and analysis of multivariate observations[M]// Le Cam L M, Neyman J, ed.. Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. Los Angeles, USA: University of California Press, 1967: 281-297