文章编号:1001 - 9081(2005)09 - 2037 - 04

一种优化初始中心点的 K 平均文本聚类算法

赵万磊[1,2],王永吉[2],张学杰[1],李 娟[2]
(1. 云南大学 信息学院,云南 昆明 650091; 2. 中国科学院 软件研究所,北京 100080)
(zhaowanl@yahoo.com.cn)

摘 要:文本聚类在信息过滤,网页分类中有着很好的应用。但它面临数据量大,特征维度高的难点。由于 K 平均算法易于实现,对数据依赖度底,在文本聚类中得到应用。然而,传统 K 平均以及它的变种会产生有较大波动的聚类结果。因此对 K 平均算法进行了改进,通过优化聚类初始中心的选择,得到一种适合对文本数据聚类分析的改进算法。大量实验显示,该算法可以生成质量较高而且聚类质量波动性较小的结果。

关键词:优化;文本聚类; K 平均
中图分类号: TP391 **文献标识码**:A

Variant of K-means algorithm for document clustering: optimization initial centers

ZHAO Wan-lei[1,2], WANG Yong-ji[2], ZHANG Xue-jie[1], LI Juan[2]
(1. *Institute of Information, Yunnan University, Kunming* 650091, *China*;
2. *Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*)

Abstract: Document clustering had been employed in information filtering, web page classification and so on. K-means is one of the widely used clustering techniques because of its simplicity and high scalability. Owing to its random selection of initial centers, unstable results were often got when using traditional K-means and its variants. Here a technique of optimization initial centers of clustering was proposed. Combined with incremental iteration, it can produce clustering results with high purity, low entropy as well as good stableness.

Key words: optimize; document clustering; K-means

0 引言

聚类分析是知识发现的重要工具,其中的文本聚类是模式识别、机器学习、统计学和信息检索技术相互结合和发展的结果。由于电子邮件、WWW 应用的普及,文本聚类在信息检索,邮件过滤和网页分类等领域有很好的应用。自 20 世纪 50 年代以来,人们提出了多种聚类算法,大致可以将它们分为基于划分和基于层次两种,同时还出现了综合运用两种思想的混合型(hybrid)聚类算法。基于层次的聚类算法又可以细分为凝聚的和分裂的两种[1]。按照度量两簇临近度的不同方式,基于层次的凝聚型聚类算法分为单链接、全链接和平均链接(UPGMA)[2,3]三种。

基于划分的聚类算法主要是 K 平均及其变种。它们聚类速度快、易于实现,而且还适用于文本、图像特征等多种数据的聚类分析。而且,最近的实验结果[4,5]改变了层次型聚类算法优于基于划分方法的传统观点,原因在于层次型的聚类算法过于依赖于最近邻(Nearest Neighbor)来寻找类簇[3]。然而,K 平均算法的缺点是其迭代过程可能会很快终止,得到一个局部最优的结果。而且由于迭代初始时的中心点选择是随机的,聚类结果会有所波动。由于聚类往往应用于最终使用者也无法评判聚类质量的数据,这种波动性在应用中将难以接受。因此,提高聚类结果质量和稳定性具有重要价值。

基于上面的分析,改进了 K 平均算法的初始点选择方法,而且采用了渐变中心的方式迭代优化,得到最终的聚类结果。我们用该算法在多个数据集上进行了聚类实验,验证了该算法的有效性。

1 相关工作

通常 K 平均算法可以分为两个步骤,步骤一是随机选取 K 个特征向量作为中心点,并把其余的特征向量点赋给离它最近的那个中心点,得到 K 个簇;第二步为每一个簇重新寻找它们的中心(仍是一个特征向量),然后再将数据集中的每个特征向量点赋给离它最近的中心,重复第二步直到这 K 中心不再变化为止。

对于步骤二的优化有多种方式:中心点的变更,传统的做法是每趟迭代结束时取簇内点集的平均向量;最近提出的优化方案是,一旦将某一个点赋给某个中心,立即就根据这个点的向量改变中心点向量,这样的优化方法已经证实优于原来的方法,采用这种新的优化方法的 K 平均也即是渐变中心[6,7](incremental)的 K 平均算法。在我们的聚类算法中,也采用了这种渐增迭代的优化方法。有的算法则在算法的第一个步骤就采取这种渐变中心点的方法[7]。因为迭代过程可能终止于一个局部最优点,Lloyd 算法[8]在迭代完毕之后,再探测每个中心点的邻近点,尝试用这些邻近点替换中心点是否可以获得更好的聚类结果。最近,提出了 Repeat Bisecting K 平均算法[4,5,9],对 K 平均算法进行了有效的改进,严格地说,它属于分裂型的层次聚类算法,算法不断使用 K 平均方法将簇集中最大的簇剖分为 2(初始只有一个簇),直到得到

收稿日期:2005 - 03 - 14;**修订日期**:2005 - 07 - 04 **基金项目**:国家 863 计划资助项目(2001AA113180;2002AA116080)

作者简介:赵万磊(1979 -),男,云南楚雄人,硕士研究生,主要研究方向:文本聚类、遗传算法; 王永吉(1962 -),男,吉林人,研究员,博士生导师,主要研究方向:实时系统、先进调度算法、机器人实时控制理论方法、非线性优化理论、计算机实时混合控制理论; 张学杰(1965 -),男,云南昆明人,教授,主要研究方向:高性能计算; 李娟(1977 -),女,山东德州人,博士研究生,主要研究方向:智能软件工程.

所需要的簇。该算法在多个文本数据集上测试取得了很好的结果[2],但由于算法以 K 平均为基础,聚类过程仍然可能终止于一个局部最优点。

为了减小 K 平均聚类算法的波动性,Karypis 等人[4,5]采用将一个聚类重复进行多次,然后用一个评价标准选出其中最好的聚类结果。这是一种直观的方法,但没有从导致波动的实际原因来改进。

最近提出了一种用来建立词义网(Word net)的文本聚类算法 CBC[10,11]。它先通过对每一个文档点的前几个近邻采用 UMPGA(凝聚型的平均链接)进行聚类分析,从而找到整个数据集中的密集区域,这样的密集区域被称为委员会(Committee),算法将生成尽可能多的委员会。但算法采用了两个固定的相似度阈值来断定一个数据区域能否成为一个委员会,因而算法对参数过于敏感,而且在数据集上反复寻找委员会很耗时。本文吸取了 CBC 算法探测数据集密集区域的方法;所不同的是,我们的聚类算法不再生成委员会集,而只是探测到数据集上的密集区域,因为这样的区域很有可能包含类簇的中心。

2 优化初始点的 K 平均

如前所述,K 平均算法聚类结果有波动,造成这一结果的原因是 K 平均法初始时是随机选取中心点,迭代过程可能会终止于一个局部极值点。对此我们的优化方案是优化聚类初始中心点的思路:先用一种方法探测数据集内的密集区域,再采用 K 平均法来迭代优化。

2.1 文档表示方法及相关度量公式

在聚类算法中文档采用信息检索领域使用最广泛的向量模型来表示文档,文档 d_i 表示为 $d_i = (w_{i1}, w_{i2}, \cdots, w_{ij}, \cdots, w_{im})$ 其中 m 为文档集的关键字总数,w_{ij} 是文档 d_i 中编号为 j 的关键词在 d_i 向量中的权重,它由以下公式得到[4]:

$$w_{ij} = freq_{ij} \times \log\left(\frac{N}{dfreq_j}\right) \tag{1}$$

其中 $freq_{ij}$ 是关键词 j 在文档 i 中的出现次数,$dfreq_j$ 是在文档集范围内包含关键词 j 的文档数。N 为文档集大小。

文档间的邻近度采用相似度来度量:

$$sim(d_l, d_m) = \frac{d_l^t \times d_m}{\| d_l \| \times \| d_m \|} \tag{2}$$

文档越相似,该值越大,当两个完全相同时相似度为 1。

算法的主要过程是,先采用凝聚的聚类算法对整个文档集的各个局部进行分析,得到文档分布的一个概要(局部文档中心点集);然后,根据用户输入所要生成的聚类数目,得到初步分类。最后对这个初步分类进行优化获得最终聚类结果。算法介绍如下:

步骤 1:采用平均链接(UMPGA)对每个文档的前 Nb 个近邻(包括文档本身)进行聚类,这样每个文档的邻近区域形成了一棵聚类树(如图 1 所示),算法从这棵聚类层次树上选取 Score = 平均相似度 × 文档数量,Score 最高的结点(实际上是一个密集的文档集合),被加入到一个链表中。图中结点 6 依据 Score 将被选中,它包括了 $\{e, d, f, g, c, a\}$。

步骤 2:按照这些密集小区域的得分(Score)为这个链表进行排序。

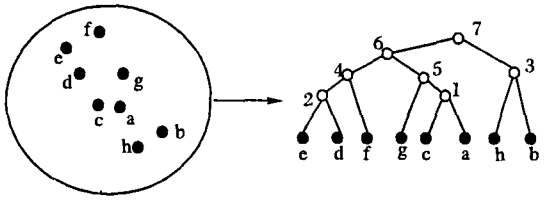

图 1 密集区域探测聚类示意图

步骤 3:为这些密集小区域生成中心点向量。中心向量是取属于这个密集小区域的文档向量各个维权重的平均值。

步骤 4:在每次聚类时,算法接受用户输入的所需聚类数目的参数 K,对于这些中心点,可以采用二分查找法,找到一个合适的相似度阈值,使得在这个相似度阈值下,有 K + REAR 个中心点他们之间的相似度小于这个阈值。

至此,获得了 K + REAR(其中 REAR 是一个正整数常数,在所有的实验中 REAR 取值均为 7)个中心,然后将每一个的文档点赋给与它最相似的中心,这样得到了一个初步分类。

这里需要对不是直接得到 K 个中心点,而是 K + REAR 个中心点做出解释:在实现算法的过程中,由于各个密集小块的疏密情况各不相同,经过步骤 2 的排序,这些小块的疏密程度是递减的,若直接获得 k 个簇,有的簇中心会显得过于松散。

步骤 5:在 K + REAR 个簇中,选出簇内平均相似度较小的前 REAR 个簇,将属于它们的文档点重新分配给其他簇,获得 K 个簇的聚类集。具体方法如下:

①计算每个簇的簇内平均相似度,根据这一相似度,对簇进行排序;

② 取出当前簇内平均相似度最小的簇 CA,将该簇内的文档成员赋给其他离该文档点最近的簇 CB,并根据该文档点向量修改那个簇的中心,即在各维平均分配该点权值。同时修正簇 CB 的簇内平均相似度;

③ 重复①,②直到簇数为 K。

步骤 6:对所得的 k 个簇采用渐变中心的优化方法进行优化[7,8],优化目标是使整个聚类结果的簇内平均相似度尽量大,目标函数如下[4]:

$$f = \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{d_i^t \times C_r}{\| C_r \|} \tag{3}$$

其中 C_r 是簇 r 的中心,S_r 是一个簇,D_i 是簇内的一个文档。

结束条件是不再发生文档从一个簇转移到另一个簇的操作,具体步骤如下:

① 计算每个簇的簇内平均相似度;

② 从文档集中,随机抽取一点,计算如果将该点从当前所属簇移动到其他簇是否会增加 f 的值,若可以则将该点移动到使 f 增加最多的那个簇,那个簇的中心相应改变;

③ 重复②直到从文档集中任意选取一点,不能将其移动到另一个簇。

2.2 算法参数敏感度分析

在聚类算法中,一共使用了 3 个参数,下面对它们在聚类过程中所起的作用及敏感性做一个说明。聚类数目参数 K,每次聚类都要求用户输入该参数,这是许多聚类算法所采取的做法,一般地,K 越大聚类质量越好;算法步骤 1 中的近邻数 Nb,我们设置为 20,它的取值是综合算法时间耗费和探测到的小块是否具有一定代表性来考虑的;REAR 的取值为 7,是出于为聚类选取比较紧凑的中心点考虑。在我们的实验中后两个参数值一直保持不变,可见其敏感度较低。

2.3 算法时间复杂度分析

聚类算法的步骤 1 中,为了对每一个文档点的前 20 个近邻(包括它本身)进行聚类,必须算出整个数据集的相似矩阵,这一步的时间复杂度是 $O(N^2)$,其中 N 为文档集大小,对每个文档点的前 20 个近邻的聚类时间复杂度为 aN,其中 a 为用平均链接算法对 Nh 个近邻的时间复杂度,因为 Nh 为常数,所以 $a = Nh^2\log(Nh)$ 也为一常数。步骤 4 的时间复杂度 $O(N^2)$,算法其余步骤的时间复杂度皆不超过 $N\log(N)$,因此整个算法的时间复杂度为 $O(N^2)$,但是如果采用 R-Tree[12],

或 X-Tree[13]可以使该算法的时间复杂度降为 $N\log(N)$。在算法运行过程中也发现,计算文档集的相似矩阵是最耗时的。

3 实验结果

3.1 实验数据集

在实验中,采用明尼苏达大学计算机科学系提供的测试数据(http://www.cs.umn.edu/~karypis/cluto/files/datasets.tar.gz),其中 k1a,k1b 和来源于项目;其中 k1a,k1b 其实是同一文档集但采用了不同的分类标准,tr31,tr41 是通过对 TREC-6[14]和 TREC-7[14]数据集处理得到。这些文档集在使用之前都已经用(http://www.tartarus.org/~martin/PorterStemmer/)提供的程序去除了词缀,并且去掉了文档中的停止词(如表 1 所示)。

表 1　TREC 测试数据集

数据集	文档总数	分类总数	关键词总数	来源
k1a	2340	20	13879	WebACE
k1b	2340	6	13879	WebACE
tr31	927	7	10128	TREC
tr41	878	10	7454	TREC
wap	1560	20	8460	WebACE

3.2 评价标准

为了评价聚类结果,采用了比较常用的纯度和熵值来衡量。纯度标准(Purity):设簇 c_i 的大小为 n_i,则该簇的纯度定义为[4]:

$$S(c_i) = \frac{1}{n_i}\max(n_i^j) \tag{4}$$

n_i^j 表示簇 c_i 与第 j 类的交集大小,于是整个聚类的纯度定义为[4]:

$$Purity = \sum_{i=1}^{k}\frac{n_i}{n}S(c_i) \tag{5}$$

其中 k 为聚类最终形成的簇的数目。纯度刻画了聚类算法分类的准确性,一般地纯度越高聚类算法越有效。

熵值标准:设簇设簇 c_i 的大小为 n_i,则该簇的熵值定义为[4]:

$$E(c_i) = -\frac{1}{\log q}\sum_{i=1}^{q}\frac{n_i^j}{n_i}\log\frac{n_i^j}{n_i} \tag{6}$$

其中 n_i^j 表示簇 c_i 与第 j 类的交集大小。于是整个簇集的熵值定义为[4]:

$$Entropy = \sum_{i=1}^{k}\frac{n_i}{n}E(c_i) \tag{7}$$

熵值刻画了同一类文档在结果簇集中的分散度,同一类文档在结果集中越分散,熵值越高,聚类结果越差。理想的情况是同一类文档同属于一个簇,此时熵值为 0。

我们采用了几种典型的聚类算法作比较,IK(ikmeans)是采用渐变中心进行优化的 K 平均法,TK(tkmeans)是传统的 K 平均方法;

RB(bisectingkmeans)是二分的 K 平均法,大量实验表明,它可以获得比较优秀的聚类结果;UMPGA 是基于平均链接度量的凝聚的层次型聚类算法,它产生的聚类结果是层次型算法中最好的。我们在几个数据集上分别使用上述聚类算法将它们聚为 15、20 和 25 个簇,为了减少聚类算法本身所造成的聚类质量的波动,其中 TK、IK、RB 和 UMPGA 是取 10 次聚类中的最好结果,聚类算法 HB() 只是取随机 1 次生成的结果,然后比较各种聚类算法生成结果集的纯度(簇数正下方左列)和熵值(簇数正下方右列),以此来对比各算法的性能(如表 2~表 6 所示)。

3.3 结果评价和分析

从表中可以看出,聚类算法和采用 RB 算法得到的聚类结果在熵值和纯度标准下较为接近,甚至在几个数据集(tr41, tr31, wap)上我们的聚类算法明显优于其他聚类算法。算法与渐变中心的 K 平均法(ikmeans)使用了相同的优化过程而结果明显好于 ikmeans,这是因为,ikmeans 开始时的聚类中心是随机选择的,而聚类中心是算法根据整个数据集上文档分布得到,这样更有可能在优化过程中使聚类算法避开局部极值点,从而取得比较好的聚类结果。

表 2　数据集 tr31

算法	簇数					
	15		20		25	
IK	0.82	0.25	0.82	0.23	0.85	0.20
TK	0.88	0.20	0.87	0.18	0.87	0.17
RB	0.85	0.22	0.85	0.19	0.85	0.18
UMPGA	0.82	0.25	0.85	0.22	0.85	0.21
HB	0.89	0.17	0.90	0.15	0.89	0.15

表 3　数据集 tr41

算法	簇数					
	15		20		25	
IK	0.82	0.21	0.80	0.21	0.86	0.16
TK	0.74	0.28	0.82	0.20	0.89	0.14
RB	0.79	0.23	0.83	0.19	0.86	0.16
UMPGA	0.72	0.31	0.77	0.26	0.78	0.25
HB	0.87	0.18	0.88	0.14	0.90	0.13

表 4　数据集 k1a

算法	簇数					
	15		20		25	
IK	0.63	0.37	0.62	0.37	0.69	0.32
TK	0.64	0.36	0.65	0.37	0.66	0.36
RB	0.64	0.36	0.69	0.33	0.69	0.32
UMPGA	0.52	0.48	0.54	0.46	0.55	0.45
HB	0.65	0.35	0.67	0.34	0.70	0.31

表 5　数据集 k1b

算法	簇数					
	15		20		25	
IK	0.92	0.13	0.92	0.12	0.92	0.12
TK	0.86	0.18	0.89	0.15	0.88	0.19
RB	0.94	0.09	0.94	0.08	0.94	0.07
UMPGA	0.84	0.22	0.86	0.18	0.87	0.18
HB	0.91	0.12	0.92	0.12	0.93	0.10

表 6　数据集 wap

算法	簇数					
	15		20		25	
IK	0.62	0.39	0.65	0.36	0.64	0.35
TK	0.62	0.40	0.60	0.38	0.63	0.38
RB	0.66	0.35	0.67	0.34	0.69	0.32
UMPGA	0.49	0.50	0.53	0.47	0.57	0.42
HB	0.69	0.33	0.71	0.30	0.71	0.30

由于采用步骤 6(2.1 节)的优化过程,我们的聚类算法得到的结果也会有波动。为了进一步验证算法对于克服 K 平均优化方法所带来的波动性的优劣,图 2,3 是在 tr41 数

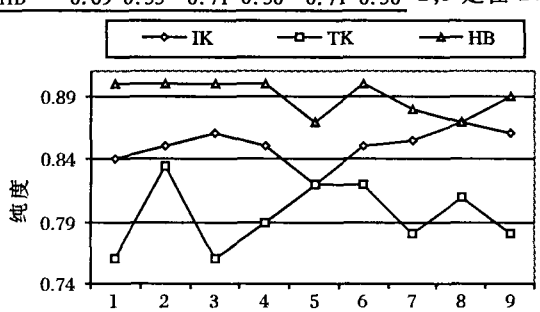

图 2　三种聚类算法纯度波动情况

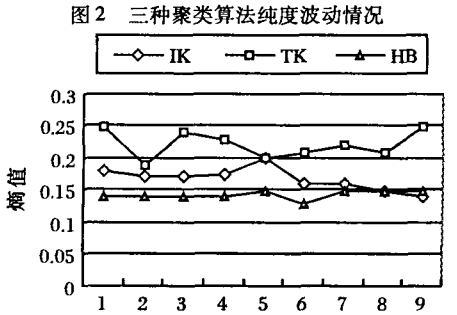

图 3　三种聚类算法熵值波动情况

据集上,采用不同的聚类算法重复 9 次生成 20 个簇的聚类结果的熵值和纯度波动图。由图可见我们的聚类结果波动明显小于其他两种 K 平均。

4 结语

本文对 K 平均方法做了改进,得到一种以优化聚类初始中心为目标的文本聚类算法,在熵值和纯度标准度量下,聚类算法在多个数据集上显示出了很好的性能。聚类算法之所以有这样好的性能是因为我们在使用 K 平均的优化过程之前,对整个文档集有一个分析的过程(初始聚类),这个过程对文档进行一遍扫描,获得文档集内文档分布的概况。这样就为聚类过程提供了一个好的聚类起点。又由于聚类算法 RB (Repeated Bisecting)采用优化目标函数(3),总是可以达到局部最优。因此可以认为,算法有可能跳出一些局部极值点,从而使聚类结果优于 RB。

算法的不足在于,为了实现初始聚类,每对一个数据集聚类,都要事先计算相似矩阵,使得聚类算法的可伸缩性不是很好。在将来的工作中,我们将探索采用其他的方法获得一个对数据集的大概分类,增强该聚类算法的可伸缩性。

参考文献:

[1] HAN JW, KAMBER M. Data Mining Concepts and Techniques [M]. BeiJing: China Machine Press. 2001. 223 - 259.

[2] JAIN AK, MURTY MN, FLYNN PJ. Data Clustering: A Review [J]. ACM Computing Surveys, 1999, 31(3): 265 - 281.

[3] QIAN WN, ZHOU AY. Analyzing Popular Clustering Algorithms from Different View Points [J]. 软件学报, 2002, 13(8): 1382 - 1394.

[4] STEINBACH M, KARYPIS G, KUMAR V. A comparison of Document Clustering Techniques [R]. Department of Comp. Sci. & Eng University of Minnesota, 2000. 1 - 20.

[5] ZHAO Y, KARYPIS G. Criterion Functions for Document Clustering Experiments and Analysis [R]. Department of Comp. Sci. & Eng University of Minnesota, 2001. 01 - 40.

[6] FABER V. Clustering and the Continuous k-Means Algorithm[EB/OL]. http://library.lanl.gov/cgi-bin/getfile?00412967.pdf, 1994.

[7] LARSEN B, AONE C. Fast and Effective Text Mining Using Linear-time Document Clustering[A]. Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining [C]. San Diego CA USA, 1999. 16 - 22.

[8] KANUNGO T, MOUNT DM, NETANYAHU NS, *et al*. An Efficient K-Means Clustering Algorithm: Analysis and Implementation[J]. Pattern Analysis and Machine Intelligence, 2002, 24(7): 881 - 892.

[9] ZHAO Y, KARYPIS G. Evaluation of Hierarchical Clustering Algorithms for Document Dataset [A]. Proceedings of the eleventh international conference on Information and knowledge management[C]. McLean, VA. USA, 2002. 515 - 524.

[10] PANTEL P, LIN D. Efficiently Clustering Document with Committees[A]. Proceedings of SIGIR02[C]. Tampere, Finland, 2002. 199 - 206.

[11] PANTEL P. Clustering by Committee [D]. Edmonton, Alberta: University of Alberta Spring, 2003. 51 - 62.

[12] GUTTMAN A. R-Trees: A Dynamic Index Structure for Spatial Searching[A]. Proceedings of ACM SIGMOD[C]. Boston, USA, 1984. 47 - 57.

[13] BERCHTOLD S, KEIM DA, KRIEGEL HP. The X-tree: An Index Structrue for High-Dimensional Data[A]. Proceedings of the 22nd VLDB Conference[C]. Mumbai, India, 1996. 28 - 39.

[14] TREC. Text REtrieval conference[EB/OL]. http://trec.nist.gov, 1999.

(上接第 2036 页)

练开销和准确度上升率的指标上均优于神经网络,即使初期的推理开销稍稍高于神经网络,但是随着反馈次数的增多,这一开销也有明显的下降并趋于稳定。

表 3 规则匹配推理、训练参数随反馈次数的变化情况

$\delta = 100$, $N = 99$, $n = 98$

反馈次数	平均预测命中率	平均反馈训练时间/ms	平均推理时间/ms
5	60.0%	2.13	20.25
10	72.4%	4.01	19.22
20	86.8%	6.91	17.10
50	94.0%	10.50	10.23
100	97.0%	16.33	3.33
200	98.0%	25.90	2.33
400	98.1%	40.00	2.36
800	98.3%	58.40	2.37
1000	98.3%	90.41	2.38

表 4 神经网络推理、训练参数随反馈次数的变化情况

双隐层, $N = 99$, $n = 98$

反馈次数	平均预测命中率	平均反馈训练时间/ms	平均推理时间/ms
5	47.0%	120.26	5.89
10	47.3%	169.21	5.98
20	48.9%	280.35	6.02
50	52.3%	456.54	6.23
100	56.9%	646.69	6.23
200	63.2%	900.89	6.24
400	72.0%	1170.45	6.26
800	82.8%	1540.11	6.55
1000	86.8%	1817.36	6.61

6 结语

从实际应用的角度讲,特别是对于一些精密设备的故障诊断,在条件十分严格的情况下,我们的系统可以给予用户快速和可区分的量化结果和可成长的自学习推理机制,与传统的一些推理方法相比,也具有较大的性能优势。至于在本系统中,由于数量级过大而导致溢出的问题,可以采用分割专家知识库和运用支持超大整数的程序环境的方法来解决。目前,本系统已经在上海同济大学软件学院的合作项目某故障诊断专家系统中予以实现,在VC++ 6.0 + MS SQL Server 2000 的环境下开发测试,并以较高的评价通过相关部门的评审。对本系统未来的完善,将着重进行以下两个方面的工作:进一步完善故障树的知识表达方式,使其能做到多二叉树的组合故障树诊断,从而使专家知识进一步可视化;完善推理和训练机制,能把不同的应用要求加以参数化,进一步增强通用性。总之,本系统对于一些知识不是很复杂的专家系统来说,有着极好的效能,对于一些规模比较大,知识结构比较复杂的专家系统,在本系统基础上加以改进和完善也是完全可以适用的,对于这种情况下的一些相关问题也正在研究中。

参考文献:

[1] 王道平,冯振声. 不确定性系统理论在故障诊断专家系统中的应用[A]. 系统工程与可持续发展战略[C]. 北京:科学技术文献出版社, 1998.

[2] 王道平,冯振声,郭建校. 基于模糊和规则推理的故障诊断专家系统[A]. '98 人工智能进展论文集[C]. 西安:西安交通大学出版社, 1998.

[3] 胡柏青,李安,高启孝. 基于故障树的通用故障诊断专家系统[A]. 全国船舶仪器仪表 2001 年学术会议论文集[C]. 北京:中国造船工程学会, 2001.

[4] GIARRANTANO J, RILEY G. Expert Systems Principles and Programming[M]. PWS Publishing Company, 1988.