一种应用于中文文本聚类的适应值函数

朱征宇 李力沛 罗 颖 周 智 朱庆生
(重庆大学计算机学院 重庆 400044)

摘 要 文本聚类中的文本对象一般都是高维的,类的大小、密度各不相同,给聚类带来了很大难度。目前国内针对这些问题而提出的应用于遗传算法的适应值函数却很少,国外的通用目标函数比较复杂,而且在文本聚类上的效果一般。针对文本对象的特征提出了一种应用于遗传算法的适应值函数,它具有结构简单、易于计算、适用于高维对象的特点,并且能够帮助遗传算法更好避免陷入局部最优,达到比较准确地描述聚类结果的目的。通过实验与 CS Measure 相比,聚类结果更优。

关键词 适应值函数,遗传算法,文本聚类,数据挖掘
中图法分类号 TP181

Fitness Function Applied to Chinese Text Clustering

ZHU Zheng-yu LI Li-pei LUO Ying ZHOU Zhi ZHU Qing-sheng
(Department of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract Generally, the object of document clustering is high dimension, and the sizes and/or densities of clusters are different. These bring much difficulties to the document clustering. But there are few proposed fitness functions aiming to these problems at home, which are appllied to the genetic algorithm, the foreign general purpose validity measures are comparatively complex, not very effective when they are appllied to document clustering. A fitness function aiming to the characteristic of the document object, which aplly to the genetic algorithm, was proposed. It is simple-structure, easy to calculate, and suitable for high dimension object. It can help genactic algorithm to avoid falling into local optimization, achieve the aim of describing the cluster result exactly. Comparing with the CS Measure through some experiments, the result is better when using our fitness function in document clustering.

Keywords Fitness function, Genetic algorithm, Text clustering, Data mining

1 引言

文本聚类是一个有着广泛应用领域的研究问题。目前流行的文本聚类方法主要有基于划分的方法(K-Means 为代表)及层次化聚类方法等。同时,针对 K-Means 算法对初始解敏感、易于陷入局部最优的缺点,能够在全局进行搜索的遗传算法也被应用到了文本聚类问题中。

遗传算法(Genetic Algorithm,简记为 GA)[1]是一种借鉴生物界自然选择和进化机制发展起来的高度并行、随机、自适应搜索算法,广泛用于求解复杂的优化问题。它模仿生物界"适者生存"原理,根据适应值的大小,从初始种群中选择若干个较好的个体参与交叉和变异操作。选择、交叉和变异操作迭代执行若干次或执行到满足特定的终止规则,最后得到的种群中适应值最高的个体即为优化问题的近似最优解。因此,适应值函数直接影响到进化过程和最后的解。一种好的适应值函数对遗传算法的进化会有很大的帮助,也有助于避开局部最优解,尽可能找到全局最优解。

2 国内外相关研究

以 K-means 为代表的基于划分的聚类算法以目标函数评价聚类结果的好坏。长期以来,这类算法都是以数据点到中心的距离之和作为目标函数。由于 K-means 的目标函数定义与遗传算法的适应值函数十分接近,大部分人在使用遗传算法进行聚类分析的时候,也采用了类似的函数式作为适应值函数,只是在距离函数的选择上稍有不同,例如余弦相似度、欧式距离或者其他距离[2-4]。

但是 K-means 算法要求在聚类前输入准确的聚类簇数目,这不符合情理。于是人们希望通过聚类结果评价函数来选择聚类簇数据的最优值,因此提出了聚类有效性函数。由于有效性函数的引入使得很多聚类算法的非监督性得到提升,因此近年来关于有效性函数的研究很多,关于目标函数的研究很少。虽然遗传算法的适应值函数通常是一种目标函数,即需要在算法执行前输入确定的聚类簇数目,但遗传算法的动态聚类和全局搜索功能在聚类分析中的作用是显而易见

到稿日期:2008-06-24 本课题得到国家科技支撑计划课题:重庆"便民 E 站"服务平台(编号 2007BAH08B04)的资助。
朱征宇 博士,教授,主要研究领域为电子商务、数据挖掘,E-mail:zhu_zhengyu@cqu.edu.cn;李力沛 研究生,主要研究领域为电子商务、数据挖掘;罗 颖 研究生,主要研究领域为电子商务、数据挖掘;周 智 研究生,主要研究领域为电子商务、数据挖掘;朱庆生 教授,博士生导师,主要研究领域为电子商务、图像处理等。

的，具有很多的优势[5]。因此，关于影响遗传算法效果的适应值函数也有很大的研究价值。可是国内外对适应值函数的研究通常都是伴随特定的遗传算法形式或者特定的距离函数，不能适用于普通的遗传算法或者其他变种。例如李宝林等人在文献[6]中提出了一种扩展的适应值函数，但是由于函数式中的惩罚函数、调解参数是其算法特有的，因此只能适用于文献[6]提出的算法，通用性不好；Bandyopadhyay 在文献[4]中提出了一种新的距离定义，而其适应值函数也只能适用于这种距离定义。针对这些情况，本文将提出一种通用性比较好的适应值函数，适用于大部分形式的遗传算法和距离函数。

由于聚类有效性函数不仅判断当前聚类簇数目是否有效，还要评价在特定聚类簇数目下聚类结果的好坏，因此大部分有效性函数同样能作为目标函数及适应值函数，例如文献[7]提出的 CS Measure。本文以 CS Measure 作为遗传算法的适应值函数与本文所提出的适应值函数进行对比。关于 CS Measure 的定义如下：

$$CS(c)=\frac{\sum_{i=1}^{c}\{\frac{1}{|A_i|}\sum_{\vec{x}_j\in A_i}\max\{d(\vec{x}_j,\vec{x}_k)\}}{\sum_{i=1}^{c}\{\min_{j\in c,j\neq i}\{d(\vec{v}_j,\vec{v}_i)\}\}} \tag{1}$$

c 是聚类数；A_i 是一个聚类集合，其中包含被分到此集合中的所有对象，而 $|A_i|$ 是此集合的元素个数；$\vec{v}_i$ 是聚类中心；d 是一个距离函数，可以是点到中心的隶属度[7]，也可以是欧式距离或者余弦相似度。使用 CS Measure 处理类的密度或者形状大小不同数据集时效果很好。

3 本文提出的适应值函数

目标函数包括适应值函数，从最初的

$$J=\sum_{i=1}^{c}\sum_{\vec{x}_j\in A_i}\|\vec{x}_j-\vec{v}_i\| \tag{2}$$

变到现在各式各样的函数式，最主要的原因是因为式(2)仅仅适用于类的形状都是超球体的时候。然而现实中的各种数据集通常都不是标准的超球体或者根本不是超球体，这时使用式(2)肯定不合适。更多的时候，各个类的大小和密度是不相同的，这时候使用式(2)作为目标函数，极容易陷入局部最优，因为式(2)只考虑了类内聚散度，没有考虑类间分离度。按照聚类的定义，聚类就是将数据对象分组成为多个类或簇，在同一个类中的对象之间具有较高的相似度，而不同类中的对象差别较大。因此，单纯考虑类内聚散度是不全面的，应该把类间分离度也考虑进去。在提出我们的适应值函数前，先看下面这种情况。

虽然实际的数据集大多数是多维的，但是在聚类分析里，为了更直观地描述问题，通常都用二维空间里面的数据点作为例子。如图 1 所示的例子，如果假设这一组数据集应该分成两个类，那么从直观上看，图 1 所示的分类应该是比较恰当的。我们将这两个类命名为 A 类和 B 类，如图 2 所示。为了叙述简便，我们分别对 A 类和 B 类的数据点编号，A 类的数据点从最顶端的左边一个点开始，从左到右，从上到下，编号为 $A1,A2,\cdots,A22$。B 类的 4 个数据点分别编号为 $B1,B2,B3,B4$，如图 3 所示。

假设 B 类数据点的坐标分别是 B1(3,9)，B2(4,9)，B3(3,1)，B4(4,1)，A 类数据点的坐标分别是 A1(9,10)，A2(10,10)，A3(7,9)，A4(8,9)，A5(9,9)，A6(10,9)，A7(9,8)，A8(10,8)，A9(9,7)，A10(10,7)，A11(9,6)，A12(9,5)，A13(9,4)，A14(10,4)，A15(9,3)，A16(10,3)，A17(7,2)，A18(8,2)，A19(9,2)，A20(10,2)，A21(9,1)，A22(10,1)。如果利用遗传算法进行聚类分析，由于遗传算法是以染色体表示聚类结果，我们可以假设图 3 的聚类结果为染色体 A，采用式(2)，我们可以计算其最后的适应值，得到的结果是 92.545。然而，这并不一定是适应值最大的染色体。如果我们假设图 4 所示的聚类结果为染色体 B，在这个聚类结果里 A7 被分配到了 B 类里面。此时，同样使用式(2)得到的适应值却是 92.582，大于染色体 A 的适应值。事实上染色体 B 很容易得到，这时候根据遗传算法规则，我们就无法选择染色体 A 作为最终结果，错过了全局最优解。因此，虽然遗传算法是进行全局搜索，但如果没有合适的适应值函数，遗传算法的优势也难以有效发挥，所以我们需要提出新的适应值函数。

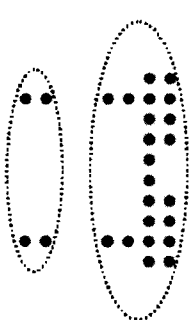

图 1 此数据集的两个聚类类簇虚线划分两个类。

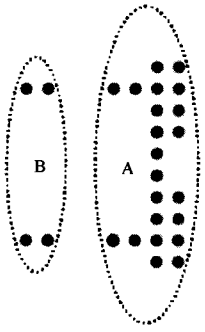

图 2 将两个类分别命名为 A，B 类

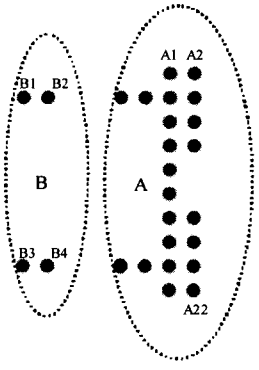

图 3 全局最优的聚类结果图

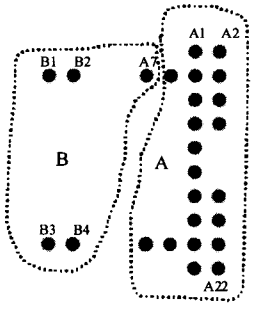

图 4 一种陷入局部最优后的聚类结果图

我们提出的新的适应值函数可以适用于绝大部分包含非超球体的不同密度或者不同大小的类的数据集，能在一定程度上避免陷入局部最优。辅助遗传算法的动态聚类过程，尽可能找到全局最优解。我们提出的适应值函数描述如下。设某数据集 $X=\{x_j;j=1,2,\cdots,N\}$，通过遗传算法得到某组聚类结果 $A_i,i=1,2\cdots c$。首先，我们计算每个类的中心，用以下公式：

$$\vec{v}_i=\frac{1}{|A_i|}\sum_{\vec{x}_j\in A_i}\vec{x}_j \tag{3}$$

这里的 A_i 就是对数据集 X 进行聚类得到的其中某个类。$|A_i|$ 即是此类的数据点数目。我们提出的适应度函数即是：

$$F=\frac{\sum_{i=1}^{c}\sum_{\vec{x}_j\in A_i}d(\vec{x}_j,\vec{v}_i)}{\sum_{i=1}^{c}\sum_{j=1,j\neq i}^{c}d(\vec{v}_i,\vec{v}_j)} \tag{4}$$

其中 $d(x,y)$ 是距离函数，可以是欧氏距离，也可以是余弦相似度。如果距离函数是欧式距离，分母要求越大越好，分子要求越小越好，因此聚类过程即是寻找使此有效性函数值最小的聚类解；如果使用余弦相似度作为距离函数，分母要求越小越好，分子要求越大越好，所以聚类过程是寻找使有效性函数值最大的聚类解。与 CS Measure 一样，我们也使用类内聚散度和类间分离度的比值来评价聚类结果的好坏。式(4)的分子事实上就是式(2)，是以前常用的适应值函数。而分母是每个类的中心之间的距离之和，表示类与类之间的分离情况。这种形式符合聚类的定义描述，而且判断聚类结果好坏标准与 CS Measure 以及式(2)一样。

然后再把我们的适应值公式应用到前面的例子中，同样利用遗传算法进行聚类分析。我们假设图 3 所示的聚类结果为染色体 A，而图 4 所示的聚类结果为染色体 B，用式(4)计算可得染色体 A 的适应值是 13.135，染色体 B 的适应值是 11.717，显然染色体 A 优于染色体 B。如果在退出时，种群里存在染色体 B 和染色体 A，那么根据遗传算法规则，就要选择染色体 A 作为最后结果。事实上用式(4)作为适应值函数的时候，染色体 A 的适应值是最大值，即是全局最优结果。这跟我们直观的感觉是一致的。如果采用式(2)作为适应值函数，那么染色体 A 的适应值就低于染色体 B 的适应值。这样就很有可能最后选择不到染色体 A 作为最后结果，而陷入某种局部最优。

通过大量实验可知，对中文文本这种比较特殊的对象进行聚类分析的时候，使用式(4)可以比较准确地描述聚类结果的好坏。当然也可以使用 CS Measure 作为适应值函数，但相比之下，我们的适应值函数有以下几个方面的优点：

①公式结构简单，运行不复杂，时间消耗少。从时间复杂度上分析，设某数据集有 n 个数据点、c 个类，假设两个公式的输入相同，所采用的算法相同，在大多数情况下，$c<<n$，而且 c 可以看作一个常数，所以可以认为分母计算需花固定时间 t，而分子的复杂度则有所区别。式(4)的分子复杂度很明显是 $O(n)$，CS Measure 的分子用其渐进最优算法计算的话复杂度是 $O(n\log\frac{n}{c})$。当 $c<<n$ 的时候，CS Measure 的分子复杂度大于式(4)的分子复杂度，所以在大多数情况下，式(4)的计算复杂度要低于 CS Measure 的计算复杂度。

②更适用于以余弦相似度作为距离函数的余弦相似度。虽然在大多数情况下使用欧氏距离作为距离函数或者相似性度量，但 A. Strehl 等人的实验表明使用欧式距离作为文档相似的衡量尺度会导致文档聚类效果较差。因此文本聚类分析中常使用余弦相似度代替欧式距离。余弦相似度的取值范围一般是[0,1]。当余弦相似度为 0 时，表示两点间的相似度极小或者不相似，为 1 时表明两点重合。使用 CS Measure 作为适应值函数时，在聚类迭代刚开始的阶段，聚类结果还不准确，在聚得的任意一个类里面都有很多本身互不相关的文本。此时距离最大的两点的余弦相似度值就很可能为 0，因此容易出现分子为 0 的情况，这就给处理带来了一定的难度。如果选择一个折衷的方法来处理分子为 0 的情况，又可能使得适应值计算不准确。而式(4)则一般不会出现这种情况。

在后面一个小节里，我们会用一些具体的实验来对两个适应值函数进行对比分析，说明式(4)的使用效果。

4 对比实验及结果分析

为了更好地比较说明式(4)的优劣，我们分别在两组数据集上做了多次实验来分析。实验数据来自于搜狗网的搜狗实验室的文本分类语料库。从中选择了 4 个类，分别是旅游、财经、教育、招聘。第一组数据集包含这 4 个类，每个类 20 个文本，参与对比实验的适应值函数是式(4)和式(2)。第二组数据集也包含这 4 个类，其中旅游类有 50 个文本、财经有 20 个文本、教育有 15 个文本、招聘有 10 个文本。参与对比实验的适应值函数是式(2)、式(4)和 CS Measure。表 1 即是在式(2)与式(4)在第一组数据集上的对比实验结果。表 2 是式(2)、式(4)和 CS Measure 在第二组数据集上的对比实验结果。

我们采用平均查准率来对实验结果进行评价。平均查准率是基于文献分类结果和分类文献之间的一致性的，平均查准率的计算式是：

$$\text{平均查准率}=\frac{1}{C}\sum\frac{c}{t} \tag{5}$$

C 是类目个数，c 是主要类目中的文献数，t 是属于目标类目中的文献数。

我们对每组数据集进行了 5 次实验，实验结果如所表 1、表 2 所列。

1 到 5 列表示这 5 次实验的每一次的平均查准率，Total 列表示的是 5 次实验的平均查准率的平均值。由表 1 可以看出，使用式(4)的聚类结果的平均查准率比使用式(2)和 CS Measure 高，说明使用式(4)使聚类更精确。由此可见式(4)同样适用于各个类的大小、密度相差不大的数据集。

从表 2 可以看出，使用式(4)作为适应值函数后，聚类结果的平均查准率明显要优于使用式(2)的平均查准率，大约提高了 13%左右。与使用 CS Measure 作为适应值函数的聚类结果的平均查准率相比，也要高大约 7%。可见式(4)比 CS Measure 更适合于使用余弦相似度作为距离函数的遗传算法聚类。

表 1 第一组实验结果

	1	2	3	4	5	Total
公式(2)	82.61%	81.54%	80.60%	74.63%	79.41%	79.76%
CS Measure	85.39%	74.74%	85.93%	80.2%	84.39%	82.13%
公式(4)	89.23%	85.48%	81.54%	90.16%	88.89%	87.06%

表 2 第二组实验结果

	1	2	3	4	5	Total
公式(2)	80.56%	63.75%	40.87%	48.79%	58.85%	58.564%
CS Measure	86.31%	58.95%	61.05%	57.11%	57.63%	64.21%
公式(4)	93.56%	60.73%	78.81%	63.45%	61.8%	71.67%

结束语 本文提出了一种新的遗传算法适应值函数——式(4)。通过实验证明，这个适应值函数具有以下优点：①相比以前普遍使用的欧氏距离函数(式(2))，式(4)更适用于大小、密度不同的非超球体类所构成的数据集，能够有效地避免陷入因为非超球体和大小、密度不同等因素引起的局部最优，配合遗传算法等动态聚类算法效果更好。②比 CS Measure 更适合于使用余弦相似度作为对象距离描述的遗传算法，并且更简单、省时。

事实上，式(4)不仅可以作为遗传算法的适应值函数，更

(下转第 272 页)

建精度。对于给定的 δ 值，当曲面变化较为平缓，即 $N_1 \cdot N_2 + N_1 \cdot N_3 + N_2 \cdot N_3$ 较大时，密贴三角形较大；反之，密贴三角形较小。可见，用 δ 密贴三角形构建的网格不仅可反映曲面的特征，而且能得到优化的结果。

根据 δ 密贴三角形的性质和生成特点，曲面网格的构建适合采用区域增长法，即：先建立种子三角形，以其三条边为基础，分别在点集中搜索满足拓扑关系和重建精度的新顶点，生成新三角形。再以新三角形的生长边为基础，继续生长，直至所有的三角形的边成为非生长边，得到整个曲面网格。这里，利用密贴三角形来保证重建的精度，同时对网格进行优化，即在生长每个三角形时都要保证拓扑关系正确和足够的精度，是区别于一般区域增长法的关键之处。下面具体介绍如何产生种子三角形和如何构造曲面网格。

3.3.1 *种子三角形的产生*

给定重建精度 δ 后，可按如下方法构建种子三角形：1)在点云中任意找一点 P，对 P 的 k 个邻近点按到 P 的距离从大到小排成序列。2)在该序列中取点 Q，以 PQ 为边，在其邻近点中任取一点 S 构成三角形 PQS。3)对 PQS 进行密贴判定和形态选择：若符合条件，则将 PQS 作为种子三角形；否则，在该序列中选择下一点作为点 Q，重新产生一个三角形进行判定。

3.3.2 *曲面网格的构造*

曲面网格的构造是一个反复从生长边生成密贴三角形的过程，其中主要是对当前生长边进行搜索并确定一个新顶点来构造一个新三角面片。一般来说，新顶点应在点云的所有孤立点中搜索，但这样做的效率太低，还可能产生形状欠佳的三角形。为了提高搜索效率，避免欠佳三角形，可将搜索范围限定在当前生长边附近的一个不大的空间。在此空间中依据以下条件来选定一个顶点：1)三角形重叠检测条件，要求三角形与已有面片有正确的拓扑关系；2)密贴条件，要求具有重建精度 δ；3)形态选择条件，减少生成狭长三角形面片的生成。如果在搜索空间中找不到满足上述条件的顶点，则表明当前生长边是曲面的边界。

结束语 基于散乱点集的曲面重建是当前计算机图形学、虚拟现实等领域的热门研究课题。自上世纪初以来，人们对基于散乱点集的曲面重建提出了多种算法，这些经典算法对相关技术的发展起了推动作用，但都存在某些不足之处。近年来，基于成长型神经网络的曲面重建方法和基于特征的曲面重建方法引起了人们的重视。这些新出现的方法对于提高重建曲面的精度、减少计算量比较有效。本文在综述各种经典曲面重建算法的基础上，较详细介绍了后两种方法。

参 考 文 献

[1] Zwicker M, Pfister H, Van Baar J, et al. Surface Splatting[A]// Proc. of ACM SIGGRAPH'01[C]. 2001:371-378

[2] Alexa M, Dachsbacher C, Gross M, et al. Point-based computer graphics[A]// Eurographics2003 Tutorial Notes[C]. 2003

[3] Hoppe H, DeRose T, Duchamp T, et al. Surface Reconstruction from Unorganized Points[C]// SIGGRAPH '92 Proceedings. July 1992:71-78

[4] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm[C]// SIGGRAPH'98. 1998:415-421

[5] Edelsbrunner H, Mücke E. 3D alpha shapes[J]. ACM Transactions on Graphics, 1994, 13(1):43-72

[6] Bradle C. Rapid prototyping models generated from machine vision data[J]. Computer in industry, 2001, 4:159-173

[7] Lin Hongwei, Tai C L, Wang Guojin. A Mesh Reconstruction Algorithm Driven by Intrinsic Property of Point Cloud[J]. Computer-Aided Design, 2004, 36(1):1-9

[8] Fritzkey B. Growing Cell Structures-A Self-organizing Network for Unsupervised and Supervised Learning [J]. International Computer Science Institute, 1993:57-67

[9] Ivrissimtzis I P, Jeong W-K, Seidel H P. Using growing cell structures for surface reconstruction[C]// Shape Modeling International 03, Conference Proceedings. 2003:78-86

[10] 王世东，张佑生. 一种基于成长型神经网络的曲面重建快速算法[J]. 合肥工业大学学报，2006(8):984-987

[11] 偶春生. 复杂场景建模和绘制中若干关键问题的研究[D]. 合肥：合肥工业大学，2007(11):55-73

(上接第 246 页)

可以作为一种通用的聚类目标函数应用到其他算法中，例如 ISODATA 算法、K-Means 算法。只是描述对象间距离或者相似程度的函数 $d(x,y)$ 要随不同的算法而改变。

不过，式(4)仍有很多不足，例如还不能有效地处理类与类之间重叠交叉比较严重的数据集；不能很好地消除孤立点所带来的影响。当孤立点的影响比较严重时，式(4)计算出来的适应值也会有所偏差。

在以后的研究中，我们会针对上述缺点对式(4)进行改进，努力使式(4)的应用范围更广泛，更好地排除孤立点带来的影响。

参 考 文 献

[1] Al-Sultan K S, Khan M M. Computational experience on four algorithms for the hard clustering problem[J]. Pattern Recognition Letters, 1996, 17(3):295-308

[2] 刘远超，王晓龙，刘秉权，等. 基于聚类分析策略的用户偏好挖掘[J]. 计算机应用研究，2005，22(12):27-29

[3] 乐兵，王明文. 基于遗传算法的动态文本聚类[J]. 江西师范大学学报：自然科学版，2006，30(3):78-81

[4] Bandyopadhyay S, Saha S. GAPS: A clustering method using a new point symmetry-based distance measure[J]. Pattern Recognition, 2007, 40(12):3430-3451

[5] Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique[J]. Pattern Recognition, 2000, 33(9):1455-1465

[6] 李宝林，兰芸，张翼英. 基于动态遗传算法的用户模型进化研究[J]. 计算机工程与应用，2006，42(14):204-207

[7] Chou C-H, Su M-C, Lai E. A new cluster validity measure and its application to image compression[J]. Pattern Analysis & Applications (Springer London), 2004, 7(2):205-220

[8] 邓健爽，郑启伦，彭宏，等. 基于搜索引擎的关键词自动聚类法[J]. 计算机科学，2007，34(3):166-168

[9] 刘丽珍，宋瀚涛，陆玉昌. 无标记训练样本的 web 文本分类方法[J]. 计算机科学，2006，33(3):204-205，215