·人工智能及识别技术· 文章编号:1000-3428(2014)11-0172-06 文献标识码:A 中图分类号:TP301.6

# 基于改进粒子群优化的文本聚类算法研究

王永贵,林 琳,刘宪国

(辽宁工程技术大学软件学院,辽宁 葫芦岛 125105)

**摘 要**:针对 k-means 算法的聚类结果高度依赖初始聚类中心选取的问题,提出一种基于改进粒子群优化的文本聚类算法。分析粒子群算法和 k-means 算法的特点,针对粒子群算法搜索精度不高、易陷入局部最优且早熟收敛的缺点,设计自调节惯性权重机制及云变异算子以改进粒子群算法。自调节惯性权重机制根据种群进化程度,动态地调节惯性权重,云变异算子基于云模型的随机性和稳定性,采用全局最优值实现粒子的变异。该算法结合了粒子群算法较强的全局搜索能力与 k-means 算法较强的局部搜索能力。每个粒子是一组聚类中心,类内离散度之和的倒数是适应度函数。实验结果表明,该算法是一种精确而又稳定的文本聚类算法。

**关键词**:粒子群优化;自调节惯性权重机制;进化程度;云变异算子;k-means 算法;文本聚类

**中文引用格式**:王永贵,林 琳,刘宪国. 基于改进粒子群优化的文本聚类算法研究[J]. 计算机工程,2014,40(11):172-177.
**英文引用格式**:Wang Yonggui, Lin Lin, Liu Xianguo. Research on Text Clustering Algorithm Based on Improved Particle Swarm Optimization[J]. Computer Engineering,2014,40(11):172-177.

# Research on Text Clustering Algorithm Based on Improved Particle Swarm Optimization

WANG Yonggui, LIN Lin, LIU Xianguo

(College of Software, Liaoning Technical University, Huludao 125105, China)

**【Abstract】** Clustering result of k-means clustering algorithm is highly dependent on the choice of the initial cluster center. With regards to this, a text clustering algorithm based on improved Particle Swarm Optimization (PSO) is presented. Features of particle swarm algorithm and k-means algorithm are analysed. Considering the disadvantages of PSO including low solving precisions, high possibilities of being trapped in local optimization and premature convergence, self-regulating mechanism of inertia weight and cloud mutation operator are designed to improve PSO. Self-regulating mechanism of inertia weight adjusts the inertia weight dynamically according to the degree of the population evolution. Cloud mutation operator is based on stable tendency and randomness property of cloud model. The global best individual is used to complete mutation on particles. Those two algorithms are combined by taking advantages of power global search ability of PSO and strong capacity of local search of k-means. A particle is a group of clustering centers, and a sum of scatter within class is fitness function. Experimental results show that this algorithm is an accurate, efficient and stable text clustering algorithm.

**【Key words】** Particle Swarm Optimization (PSO); self-regulating mechanism of inertia weight; degree of evolution; cloud mutation operator; k-means algorithm; text clustering
**DOI**:10.3969/j.issn.1000-3428.2014.11.034

## 1 概述

文本信息量随着互联网不断的发展日益膨胀,人们亟需对这些庞大而又复杂的文本信息进行高效的组织和整理,以便能有效地定位满足用户需求的信息,文本挖掘是解决这类问题的主要技术。聚类是数据挖掘、模式识别研究的重要内容之一[1]。文本聚类的目标是将文本划分成若干个簇,使相同簇之间的内容相似度尽可能大,而不同簇之间的内容相似度尽可能小。

文本聚类的方法多种多样,其中,k-means 算法简单、高效,是一种重要的文本聚类算法。但其对初

**基金项目**:国家自然科学基金资助项目(60903082);辽宁省教育厅基金资助项目(L2012113)。
**作者简介**:王永贵(1967-),男,教授,主研方向:智能计算,云计算,数据挖掘;林 琳,硕士;刘宪国,讲师。
**收稿日期**:2013-10-29 **修回日期**:2013-12-19 **E-mail**:lidypli@126.com

始聚类中心选择敏感,许多学者对 k-means 算法进行改进。文献[2]通过对数据集进行多次采样和 k-means 预聚类以产生多组不同的聚类结果,利用不同聚类结果的子簇之间存在的交集构造出关于子簇的加权连通图,并根据连通性合并子簇,提高聚类结果的质量。文献[3]利用混合 Hausdorff 距离作为相似测度实现数据聚类。文献[4]运用谱方法估计特征中心来初始化聚类中心。

粒子群优化(Particle Swarm Optimization, PSO)[5]是一种源于对鸟类捕食行为模拟的重要群体智能算法。PSO 初始化一群随机粒子,即随机解,然后通过迭代找到最优解。在每一次迭代中,粒子通过跟踪个体极值和全局极值来更新自己的速度与位置。该算法具有较强的全局搜索能力且概念简单,易于实现,作为一种有效的优化工具已被成功地应用到诸多工程领域[6]。但它自身也存在缺陷,在遇到局部极值时,粒子的速度迅速降低直到停滞,且很难跳出局部极值点,出现早熟现象,而惯性权重是粒子群算法一个重要参数,用以调节粒子群的搜索能力。

通过对 k-means 算法和粒子群优化的分析,本文提出一种基于改进粒子群优化的文本聚类算法。改进粒子群优化算法,增强其全局搜索能力、避免早熟收敛,并将改进后的粒子群算法与局部搜索能力较强的 k-means 算法相结合,以解决 k-means 算法对初始聚类中心选择过分依赖的问题。

## 2　文本表示

在聚类之前通常将文本转化成易被计算机识别的形式,然后计算文本间的相似性,根据相似性将文本划分成各个簇。

文本聚类问题中常采用向量空间模型(Vector Space Model,VSM)[7]进行文本表示。该模型将每个文本表示成空间向量,特征词作为文本的表示单位,向量的每一维是对应特征词在该文本中的权值。即把文本集 $x$ 表示成$(x_1, x_2,\cdots, x_n)$,$x_i$ 的向量空间表示为$(\omega_1(x_i),\omega_2(x_i),\cdots,\omega_m(x_i))$。其中,$m$ 表示特征项的数目;$\omega_j(x_i)$表示第 $j$ 个特征项在文本 $x_i$ 中的权值。计算特征项权值的方法有很多,一般选用 TF * IDF 算法[8]。

$$\omega_i(x_j) = \frac{tf_{ij} \times \mathrm{lb}(N/N_i + a)}{\sqrt{\sum_{i=1}^{n} [\, tf_{ij} \times \mathrm{lb}(N/N_i + a)\,]^2}} \tag{1}$$

其中,$tf_{ij}$表示第 $i$ 个文本特征在文本 $x_j$ 出现的次数;$N$ 为文档总数;$N_i$ 为文本集合中出现第 $i$ 个特征词的文本数;分母为归一化因子。文本相似度是衡量文本间相似程度大小的一个统计量,是文本聚类的一个主要依据,计算方法有余弦法、内积法、距离函数法等。本文选用距离函数法中的欧氏距离法:

$$D(x_i,x_j) = \sqrt{\sum_{k=1}^{n} (\omega_{ki} - \omega_{kj})^2} \tag{2}$$

其中,$\omega_{ki}$与 $\omega_{kj}$分别为第 $k$ 个文本特征在文本 $x_i$ 与 $x_j$ 的特征权值。

## 3　云理论

设 $U$ 是一个用精确数值表示的论域,$A$ 为 $U$ 上对应着的定性概念,对于 $U$ 中的任意一个元素 $x$,都存在一个有稳定倾向的随机数 $\mu_A(x)$,记做 $y$,$y \in [0,1]$,$y$ 就叫作 $x$ 对概念 $A$ 的确定度,$x$ 在 $U$ 上的分布称为云,记为 $A(x,\mu)$。每一个 $x$ 称为一个云滴[9]。

云的数字特征——期望 $Ex$,熵 $En$ 和超熵 $He$,用于反映云要表达的定性概念 $A$ 的整体特性。其中,期望 $Ex$ 表示论域空间 $U$ 中最能够代表定性概念 $A$ 的点,即这个概念量化的最典型样本点;熵 $En$ 表示熵是定性概念随机性的度量,既反映了代表定性概念 $A$ 的云滴出现的随机程度,又反映了在论域空间 $U$ 中可以被语言值 $A$ 接受的云滴值范围。超熵 $He$ 表示超熵是熵的不确定性的度量,即熵的熵[10]。

生成云滴的算法或硬件称为云发生器[11]。下面对本文用到的基本云发生器的算法[12]进行介绍。

基本云发生器算法的具体步骤:

**Step 1**　生成期望为 $En$,标准差为 $He$ 的正态随机数 $En'$。

**Step 2**　生成期望为 $Ex$,标准差为 $En'$的正态随机数 $x$。$x$ 为论域空间中的一个云滴。

**Step 3**　计算 $y = \mathrm{e}^{\frac{-(x_i - Ex)^2}{2(En')^2}}$。$y$ 为定性概念 $A$ 的确定度。

**Step 4**　重复 Step1 ~ Step3,产生 $n$ 个云滴。

## 4　粒子群算法与 k-means 算法

PSO 算法最初是受到鸟类捕食行为的启发,进而利用群体智能建立的一个简化模型。该算法将每个个体抽象成解空间中的点,具有位置和速度属性。所有的粒子都有一个适应度值,该值由目标函数决定,以评价粒子位置的优劣性。粒子通过跟踪个体极值和全局极值来更新自己的速度和位置,其更新公式如式(3)和式(4)所示。

$$\boldsymbol{v}_{id}^{k+1} = \omega \boldsymbol{v}_{id}^{k} + c_1 r_1 (\boldsymbol{p}_{id} - \boldsymbol{z}_{id}^{k}) + c_2 r_2 (\boldsymbol{p}_{gd} - \boldsymbol{z}_{id}^{k}) \tag{3}$$

$$\boldsymbol{z}_{id}^{k+1} = \boldsymbol{z}_{id}^{k} + \boldsymbol{v}_{id}^{k+1} \tag{4}$$

其中,$\boldsymbol{z}_{id}$为 $i$ 个粒子的 $d$ 维位置矢量;$\boldsymbol{v}_{id}$为粒子的飞行速度;$\boldsymbol{p}_{id}$为粒子迄今为止搜索的最优位置;$\boldsymbol{p}_{gd}$为

整个粒子群迄今为止搜索的最优位置;$\omega$ 为惯性权重,表示先前粒子的速度对当前速度的影响程度;$r_1$ 和 $r_2$ 为[0,1]之间的随机数;$c_1$ 和 $c_2$ 为学习因子,也称加速因子。粒子群算法具有较强的全局搜索能力,但它在优化过程初期收敛速度较快,易陷入局部极值,过早收敛。

k-means 算法是一种典型的聚类算法,具有较强的局部搜索能力。假设将文本集合 $X$ 中的 $n$ 个文本分为 $m$ 类。计算步骤为:选取 $m$ 个文本作为初始聚类中心,计算每个文本到聚类中心的距离,将其划分到离其最近的类中,计算每个类中文本的均值作为新的聚类中心,重复该过程直到聚类中心不发生变化。从上述步骤可以看出聚类中心的选取对结果有很大的影响。因此,合理选择初始聚类中心是 k-means 算法的关键步骤。

## 5 改进的文本聚类算法设计

### 5.1 改进的粒子群算法

粒子群算法虽然简单、易于实现,但它自身也存在缺陷,其局部搜索能力较差,搜索精度不高,后期容易陷入局部最优,失去粒子的多样性,从而难以保证搜索到全局最优解。为优化粒子群算法,设计了自调节惯性权重机制和云变异算子,提高了算法的精度和收敛速度,保证能够有效地搜索到全局最优值。

#### 5.1.1 自调节惯性权重机制

惯性权重 $\omega$ 是粒子群算法中一个重要的参数。从粒子速度更新公式中可以看出,当 $\omega$ 较小时,$v_{id}$ 接近全局极值与局部极值的可能性较大,有利于全局极值的收敛。这在后期是非常有利的,能够加快算法的收敛性,快速向全局极值靠拢。但是在初期却不然,因为初期应大力探索新的解空间,寻找更好的全局极值。$\omega$ 较大时情况与之相反。因此,较大的 $\omega$ 值有利于全局搜索,较小的 $\omega$ 值有利于局部搜索。现在最常用的方法是线性递减 $\omega$ 值,即先设置较大的 $\omega$ 值,增强粒子的全局搜索能力,随着迭代次数的增加 $\omega$ 值线性递减,最后局部搜索,得到精确极值。这样做从一定程度上可以改善粒子的搜索能力,但还存在一定的缺陷:首先,$\omega$ 减小的速度较快,可能极值点所在的解空间还没搜索,就进行局部搜索了。其次,寻优初期可能找到极值点了,由于 $\omega$ 较大而跳过极值点。因此对 $\omega$ 值的调节不仅要依赖于迭代次数,还应依据整个群体的进化程度。为此,本文提出一种自调节惯性权重机制。

$$\begin{cases} A = \{p_{id} \mid p_{gd} - kp_{gd} < p_{id} < p_{gd} + kp_{gd}\} \\ n = cardA \\ \rho = i \times \dfrac{n}{N} \\ \omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times e^{\left(-\frac{1}{\rho t}\right)} \end{cases} \tag{5}$$

其中,$p_{id}$,$p_{gd}$ 分别为个体最优值与全局最优值;$k$ 为常数;$A$ 为个体极值接近全局极值的粒子集合;$i$ 为控制系数;$n$ 为集合 $A$ 内元素的个数;$N$ 为粒子总数;$\rho$ 为个体极值接近全局极值的粒子密度;$\omega_{\max}$,$\omega_{\min}$ 分别为最大惯性权重与最小惯性权重;$t$ 为迭代次数。观察 $\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times e^{\left(-\frac{1}{\rho t}\right)}$,根据极限定理,$0 < e^{\left(-\frac{1}{\rho t}\right)} < 1$,保证了 $\omega$ 在 $[\omega_{\min}, \omega_{\max}]$ 范围内,$\dfrac{1}{\rho t}$ 随着迭代次数的增加而减小,因而 $\omega$ 随着迭代次数的增大而减小,达到快速收敛的目的。此外,粒子群还依据自身情况适当调节 $\omega$ 值。$\rho$ 越大,说明越多的粒子靠近最优值,$\rho t$ 值的增加幅度会变大,从而 $\omega$ 值步长减小,有助于找到更精确的解。当 $\rho$ 较小时,说明粒子离最优值较远,原理与上述过程相反,有利于开发新空间,避免早熟收敛。在计算 $\rho$ 的公式中,控制参数 $i$ 较大时,$\omega$ 减小的速度加快,$i$ 较小时,$\omega$ 减小的速度会较为缓慢。这种基于粒子密度的自调节惯性权重机制能灵活而又有效地调节 $\omega$ 的值,进而提高算法的性能。

#### 5.1.2 云变异算子

随着迭代次数的增加,种群多样性会逐渐减小,有可能发生早熟现象。因此,当种群进化到一定程度时对种群进行变异操作以提高种群多样性,从而避免陷入局部最优。基于云的模糊性和随机性,利用云的 3 个数字特征,即期望 $Ex$,熵 $En$,超熵 $He$ 实现变异过程。

当全局极值较长时间没有变化或者变化幅度较小时,则认为粒子陷入局部最优,利用基本云发生器对全部粒子进行变异。其中,$Ex = P_{gd}$,$En = \dfrac{P_{gd}}{3}$,$He = \dfrac{En}{10}$,$P_{gd}$ 为全局最优值。

$Ex$ 是最能代表定性概念的点,具有很强的代表性,反映了云的重心,虽然此时整个种群已经陷入局部最优,但优秀个体周围可能会存在更优秀的个体,因此,选择全局极值 $P_{gd}$ 作为期望 $Ex$。$En$ 反映了论域空间可被接受的云滴范围,$En$ 越大,可被接受的云滴范围就越大,反之,$En$ 越小,可被接受的云滴范围就越小。进化后期,应缩小云滴水平覆盖范围,从而缩小变异操作时搜索的范围,本文取 $En = \dfrac{P_{gd}}{3}$,同时也是对 $En$ 的动态调节。$He$ 反映的是云滴凝聚度,$He$ 过大,一定程度上会丧失稳定倾向性,$He$ 过小,一定程度上会丧失随机性,本文取 $He = \dfrac{En}{10}$。

#### 5.1.3 改进的粒子群算法描述

下面给出改进的粒子群算法的基本步骤。

**Step 1**　初始化种群。粒子的速度、位置、个体极值、全局极值等。

**Step 2**　计算粒子适应度值,比较各粒子适应度值,更新个体极值与全局极值。

**Step 3**　判断全局极值是否在规定代数内没有发生变化或达到变异阈值。如果是,则产生云变异算子,利用基本云发生器完成对所有粒子的云变异操作。否则,跳转到 Step4。

**Step 4**　采用自调节惯性权重机制,即按式(5)计算惯性权重 $\omega$。

**Step 5**　按式(3)、式(4)更新粒子的速度和位置。

**Step 6**　若达到最大进化代数,输出全局最优值,算法结束。否则,跳转到 Step2。

### 5.2　改进的文本聚类算法原理

5.2.1　粒子编码

现在大多数的粒子群聚类算法都采用实数或者浮点数的编码方式,即维数为 $d$ 的若干个文本组成的文本集合聚成 $k$ 个类,每个粒子的位置是 $k \times d$ 维的向量,速度跟位置具有同样的数据结构,所以粒子的速度也是 $k \times d$ 维的向量。其编码方式如图 1 所示。这种编码方式相对复杂,个体结构较长,必然会增加算法搜索的时间。

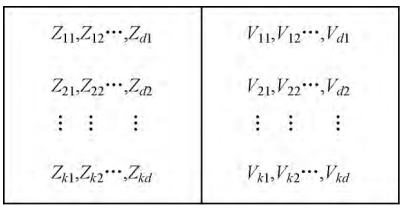


**图 1　多数粒子群聚类算法的个体编码结构**

本文提出了一种新的个体编码方式,将文本集中 $N$ 个文本编号,即 1 ~ $N$,用 $K$ 个聚类中心在文本集中的编号代替聚类中心,其编码结构如图 2 所示。其中,$Z_i(i=1,2,\cdots,K)$ 是第 $i$ 个聚类中心在文本集中对应的编号;$V_i(i=1,2,\cdots,K)$ 是粒子的飞行速度。这种编码方式不仅使粒子长度大大减小,还能保证聚类中心的搜索空间不会随着粒子迭代次数的增加而增大,提高了算法的效率,是一种简单、有效且易于理解的编码方式。

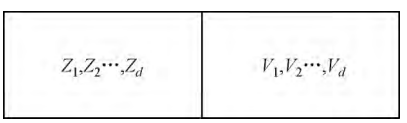


**图 2　本文算法的编码结构**

5.2.2　适应值函数

文本聚类的目标是使各类内文本距离之和的总值最小,本文使用欧氏距离进行文本间相似性度量,将适应度函数定义如下:

$$fit(ind) = \frac{1}{1+\sum_{j=1}^{K}\sum_{X_i\in C_j} D(X_i,B_j)} = \frac{1}{1+\sum_{j=1}^{K}\sum_{X_i\in C_j} \| X_i - B_j \|} \tag{6}$$

其中,$K$ 为聚类数目;$X_i$ 为类 $C_j$ 中的文本;$B_j$ 为聚类中心,其实际意义是各类文本到其聚类中心距离的总和(即离散度之和)加 1 后求倒数,即为粒子的适应度值。这样,粒子的适应度与离散度之和成负相关,离散度之和越小,粒子的适应度值越大。

5.2.3　算法描述

基于改进粒子群算法的文本聚类算法可以描述为:

**Step 1**　种群初始化。将文本集合中 $N$ 个文本编码为 1 ~ $N$,随机选择 $K$ 个文本作为初始聚类中心,用文本编号作为粒子的位置编码,并初始化粒子的速度。重复 $M$ 次,生成种群数量为 $M$ 的粒子群。

**Step 2**　$M$ 个粒子采用 k-means 算法分别对 $N$ 个文本进行聚类划分,并按式(6)计算粒子的适应度值。更新个体极值、全局极值。

**Step 3**　观察全局极值是否在规定代数内没有发生变化,如果是,则生成云变异算子,对所有粒子进行变异操作。否则,跳转到 Step4。

**Step 4**　按式(5)更新惯性权重。按式(3)、式(4)更新粒子的速度和位置。

**Step 5**　判断算法是否达到停止标准,如果是,则跳转到 Step6,否则跳转到 Step2。

**Step 6**　输出适应度最大的粒子作为初始聚类中心,其对应的 k-means 聚类结果为最终聚类结果。

该算法的基本原理是:将文本聚类中心按文本编号编码成粒子个体,利用改进的粒子群算法产生新的粒子,即新的聚类中心,再用 k-means 算法进行优化,如果种群陷入停滞状态,则生成云变异算子对种群进行变异,重复此过程,直至结束,最终得到的最优个体所对应的 k-means 聚类结果为最终结果。该算法将改进的粒子群算法的全局优化能力与 k-means 算法的高效性与局部搜索能力充分结合,从而快速准确地找到初始聚类中心。

## 6　仿真实验与结果分析

### 6.1　实验参数设置

在自调节惯性权重机制中,$k=0.3$,控制参数 $i=1$。文献[13]发现当 $\omega_{\max}=0.95$,$\omega_{\min}=0.4$ 时,算法的性能会显著提高,因此,取 $\omega_{\max}=0.95$,$\omega_{\min}=0.4$。$c_1=c_2=2.0$。改进的粒子群算法对比实验中,算法的终止条件是进化代数达到 50 次。聚类实验中,算法的终止条件是整个种群平均适应度值连续多代无

明显变化。

### 6.2 改进的粒子群算法对比实验

本次实验采用以下 3 个经典函数优化问题(求解最小值)来测试算法性能。其中,Sphere 函数是一个简单的单峰函数,用以测试函数的精度;Griewank 函数与 Schaffer 函数的全局极值点周围包围着很多局部极值点,容易陷入局部最优,是难度较大的复杂优化问题。

(1)Sphere 函数:

$$f_1(x)=\sum_{i=1}^{n}x_i^2,x\in[-10,10],f_{\min}=0$$

(2)Griewank 函数:

$$f_2(x)=\frac{1}{4\ 000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$$

$$x\in[-600,600],f_{\min}=0$$

(3)Schaffer 函数:

$$f_3(x)=0.5+\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{[1+0.001(x_1^2+x_2^2)]^2}$$

$$x\in[-100,100],f_{\min}=0$$

对于上述测试函数分别采用标准的粒子群算法(PSO)、混沌惯性权值调整策略的粒子群优化算法(CIWPSO)[14]及本文改进的粒子群算法进行求解,种群规模为 50。其中,PSO 算法与本文改进的粒子群算法参数按 6.1 节设置,CIWPSO 采用文献[14]的设置,$c_1=c_2=1.5$。实验结果如表 1 所示。

**表 1 3 种算法的实验结果对比**

| 函数 | 算法 | 平均值 | 标准方差 |
|---|---|---|---|
| $f_1$ | PSO 算法 | 2.530e-23 | 1.002e-20 |
| | CIWPSO 算法 | 1.492e-15 | 2.831e-18 |
| | 本文算法 | 0 | 0 |
| $f_2$ | PSO 算法 | 3.411e-04 | 5.228e-05 |
| | CIWPSO 算法 | 4.522e-14 | 2.624e-22 |
| | 本文算法 | 1.352e-30 | 4.002e-43 |
| $f_3$ | PSO 算法 | 1.385e-00 | 9.223e-01 |
| | CIWPSO 算法 | 5.242e-03 | 1.457e-05 |
| | 本文算法 | 5.671e-28 | 4.285e-40 |

从表 1 可以看出,对于所有测试函数,改进的粒子群算法均表现出良好的性能。在 Sphere 函数上,本文改进的粒子群算法的平均值为 0,即为精确的最优解,且方差也为 0,表示本文改进的粒子群算法每次都能搜索到精确最优解。而在 Griewank 函数与 Schaffer 函数上,本文改进的粒子群算法则表现出更高的精确性和稳定性。以上数据表明,该算法具有良好的精确性和鲁棒性。

### 6.3 实验与分析

**实验 1** 算法稳定性测试

在本次实验中,分别采用 k-means 算法、基本的 PSO 聚类算法、优化的 PSO 聚类算法、PSO + K-means 算法与本文算法对源自腾讯网的 150 篇文档(共 5 类,每类 30 篇)进行聚类测试,每种运行 50 次。测试结果如表 2 所示。

**表 2 算法稳定性测试结果**

| 算法 | 平均迭代次数 | 迭代次数的方差 | 得到最优解的次数 | 收敛概率 |
|---|---|---|---|---|
| k-means 算法 | 29 | 0.58 | 38 | 0.76 |
| 基本的 PSO 聚类算法 | 27 | 0.33 | 42 | 0.84 |
| 优化的 PSO 聚类算法 | 26 | 0.31 | 44 | 0.88 |
| PSO + k-means 算法 | 22 | 0.26 | 45 | 0.90 |
| 本文算法 | 13 | 0.20 | 50 | 1.00 |

从上述实验结果可以看出,k-means 算法的稳定性较差,原因在于其对初始聚类中心的依赖性较强。基本的 PSO 聚类算法和优化的 PSO 聚类算法在稳定性上有较大的提高且平均迭代次数也相对减小。PSO + k-means 算法将两者结合,虽然在稳定性和迭代次数上有较大的提高,但因其只是单纯地将两者结合,而没有优化,其性能仍不如本文算法。本文将全局搜索能力较强的粒子群算法进行优化,防止其早熟收敛,再与局部搜索能力较强的 k-means 算法结合,无论是在迭代次数还是稳定性上都有着显著的提高。在收敛概率上,本文算法达到了 100% ,即 50 次实验中,每次都能得到最优解。

**实验 2** 算法精确度测试

本实验的数据集都来自于腾讯网,每组数据集分别是关于娱乐、体育、时尚、新闻方面,其具体情况如表 3 所示。

**表 3 测试数据集**

| 数据集 | 文档数 | 类别 |
|---|---|---|
| $D_1$ | 113 | 电影,音乐,综艺,明星 |
| $D_2$ | 152 | 足球,赛车,体彩 |
| $D_3$ | 80 | 美容,服饰,瘦身,化妆 |
| $D_4$ | 85 | 教育,体育,法律,经济 |

目前大多数聚类结果评价通常用 *F-measure* 来衡量,它是信息搜索领域中测试系统性能的常用指标,综合了 2 个重要的概念——查准率(*precision*)与查全率(*recall*),分别考察的是精确性与完备性。对于一个聚类 $i$ 和主题类别 $j$:

$$precision(i,j)=\frac{N_1}{N_2}$$

$$recall(i,j)=\frac{N_1}{N_3}$$

其中,$N_1$ 为在聚类 $i$ 中但属于主题类别 $j$ 的文本数

量;$N_2$ 为聚类 $i$ 中的文本数量;$N_3$ 为主题类别 $j$ 中的文本数量。主题类别 $j$ 的 *F-measure* 定义为:

$$F-measure(j)=\frac{2\times precision(i,j)\times recall(i,j)}{precision(i,j)+recall(i,j)}$$

对于聚类结果来说,总的 *F-measure* 则由主题类别 $j$ 的 *F-measure* 加权平均值得到:

$$F\text{-}measure=\frac{\sum_j(|j|\times F\text{-}measure(j))}{\sum_j|j|}$$

其中,$|j|$表示主题类 $j$ 中所有文本的数量。

基于实验 1 的实验结果,本次实验选择各性能仅次于本文算法的 PSO + k-means 算法作比较。通过对数据集测试 10 次的总 *F-measure* 值的平均值评价聚类效果,实验结果如图 3 所示。

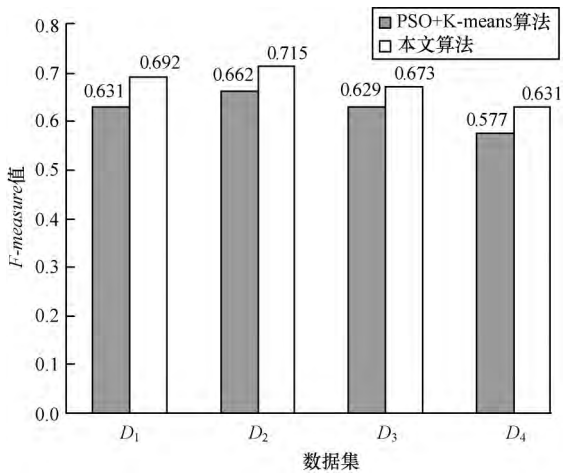


**图 3　2 种算法的 *F-measure* 值**

观察图 3,纵向来看,对于各数据集,本文算法的 *F-measure* 值都要高于 PSO + k-means 算法,约高出 5 个百分点 ~ 6 个百分点。例如对数据集 $D_4$,用 PSO + k-means 算法聚类的 *F-measure* 值是 0.577,但采用本文算法聚类后,*F-measure* 值提高到了 0.631。横向来看,除了数据集 $D_4$,本文算法在其他 3 个数据集上的 *F-measure* 值都超过了 0.67,而 PSO + k-means 算法在数据集 $D_2$ 上的 *F-measure* 值最高,为 0.662。对于在数据集 $D_4$ 上 *F-measure* 值较低可能是因为数据集的选取引起的。整体来看,本文算法在聚类效果上有着显著的优势。

## 7　结束语

本文改进了粒子群优化算法,并将改进后的粒子群算法与 k-means 算法结合,充分发挥粒子群算法的全局优化能力与 k-means 算法的高效性和局部寻优能力,提出了基于改进粒子群算法的文本聚类算法。实验结果证明,该算法具有较强的稳定性和较高的准确性,能够产生较好的聚类效果。在改进的粒子群算法中,$i$ 的选取能有效地控制惯性权重 $\omega$ 的增加或减小的速度,合理的 $i$ 值既能加快收敛速度,又能防止算法早熟。在基于改进粒子群算法的文本聚类算法中聚类算法是在聚类数确定的前提下执行的,而实际聚类问题中,聚类数是未知的。因此,对于控制参数 $i$ 的选取以及如何优化聚类数目是下一步研究的重点。

## 参考文献

[1] 孙吉贵,刘　杰,赵连宁. 聚类算法研究[J]. 软件学报,2008,19(1):48-61.
[2] 雷小锋,谢昆青,林　帆,等. 一种基于 K-means 局部最优性的高效聚类算法[J]. 软件学报,2008,19(7):1683-1692.
[3] 谢红薇,李晓亮. 基于多示例的 k-means 聚类学习算法[J]. 计算机工程,2010,36(17):179-181.
[4] 钱　线,黄萱菁,吴立德. 初始化 K-means 的谱方法[J]. 自动化学报,2007,33(4):342-346.
[5] Kennedy J, Eberhart R. Particle Swarm Optimization[C]//Proceedings of IEEE International Conference on Neural Networks. Piscataway, USA: IEEE Press, 1995: 1942-1948.
[6] 倪庆剑,长志政,王蓁蓁. 一种基于可变多簇结构的动态概率粒子群优化算法[J]. 软件学报,2009,20(2):339-349.
[7] Salton G, Wong A, Yang C S. A Vector Space Model for Automatic Indexing[J]. Communications of the ACM, 1975,18(11):613-620.
[8] Salton G, Buckley B. Term-weighting Approaches in Automatic Text Retrieval[J]. Information Processing and Management, 1988,24(5):513-523.
[9] 王守信,张　莉,李鹤松. 一种基于云模型的主观信任评价方法[J]. 软件学报,2010,21(6):1343-1344.
[10] 李海林,郭崇慧,邱望仁. 正态云模型相似度计算方法[J]. 电子学报,2011,39(11):2561-2567.
[11] Hu Changhua, Si Xiaosheng, Yang Jianbo. System Reliability Prediction Model Based on Evidential Reasoning Algorithm with Nonlinear Optimization[J]. Expert Systems with Applications, 2010, 37(3): 2550-2562.
[12] 戴朝华,朱云芳,陈维荣. 云遗传算法及其应用[J]. 电子学报,2007,35(7):1419-1424.
[13] Shi Y, Eberhart R C. Empirical Study of Particle Swarm Optimization [C]//Proceedings of Congress on Computational Intelligence. Washington D. C., USA: [s. n.], 1999: 1945-1950.
[14] 吴秋波,王允诚,赵秋亮,等. 混沌惯性权值调整策略的粒子群优化算法[J]. 计算机工程与应用,2009,45(7):49-51.

编辑　顾逸斐