

蚁群算法在文本聚类中的应用研究

张海涛, 周爱武

(安徽大学 计算机科学与技术学院, 安徽 合肥 230601)

摘 要: 为提高文本聚类结果的精度和文本聚类的收敛速度, 对蚁群文本聚类算法进行了改进, 改进的措施主要包括修改迭代终止条件、动态调整蚂蚁观察半径、改变蚂蚁移动策略, 并且在复旦大学中文文本分类语料库上进行了仿真实验。实验结果表明, 改进后的蚁群文本聚类算法不仅加快了文本聚类的收敛速度, 而且提高了文本聚类结果的精度。

关键词: 文本聚类; 移动策略; 观察半径; 蚁群算法

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2016)01-0081-04

Research of Text Clustering Based on Ant Colony Algorithm

ZHANG Hai-tao, ZHOU Ai-wu

(School of Computer Science and Technology, Anhui University, Hefei 230601, China)

Abstract: In order to improve the accuracy of text clustering and clustering of convergence, Ant Colony clustering algorithm was improved. Measures for improvement consists of 3 main aspects, namely modified linear iteration stopping conditions, dynamic adjustment of ants observed radius, change ants moving strategy. An experiment was finished in Fudan University on the Chinese text corpus. Experimental results show that the improved Ant Colony clustering algorithm can speed up the convergence of the text clustering and improved the accuracy of text clustering.

Key words: Text clustering; moving strategy; observed radius; Ant Colony algorithm

1 引言

文本聚类就是按照某种算法和相似性策略将文本聚集成一个簇, 簇之间的文本差异尽可能大^[1], 簇内文本差异尽可能小, 在整个聚类过程中不需要任何的监督机制。目前的文本聚类算法中, 常见的有划分聚类、密度聚类、生物模型聚类、层次聚类、网格聚类。各种聚类方法采用不同的模型产生聚类结果, 因此其时间和空间的复杂度是不同的。蚁群文本聚类算法是生物模型聚类中的一种, 该算法是基于 LF 算法模型提出的, LF 算法是 LumerE 和 FaibaB 将 Deneubourg JL 在 1991 年提出的蚁堆聚类分析运用到了数据分析的范畴。分析基本蚁群文本聚类算法可知其存在不少问题, 如算法的收敛速度过慢、蚂蚁移动太过随机、迭代次数过多, 这些问题影响了文

本聚类的效率和文本聚类的效果^[2]。针对这些问题学者们提出了使用信息素来引导蚂蚁的移动, 对聚类过程中产生的簇进行合并, 与遗传算法融合等等方法, 取得了良好的聚类效果。

本文采用修改迭代终止条件、动态调整蚂蚁观察半径、改变蚂蚁移动策略这 3 方面的措施来改进基本蚁群文本聚类算法过程中存在的不足, 通过相应文本分类语料库的仿真实验证明了改进算法的可行性。

2 文本聚类技术概述

2.1 文本表示

在文本聚类之前, 需要对文本进行预处理, 文本的数据格式是非结构的, 所以应该将非结构的文本数据转换成计算机能够识别的格式, 本文使用向量

空间模型 (Vector Space Model, VSM) 来表示文本^[3]. 文本 d 可以表示为

$$V(d) = \{t_1, w_1(d); \dots; t_i, w_i(d); \dots; t_n, w_n(d)\} \quad (1)$$

式中, t_i 表示特征词项, $w_i(d)$ 表示 t_i 在文本中的权重值.

一个特征词在文本向量中的权重可以使用多种方式决定, 本文中采用式(2) 计算 t_i 权重, 其中 tf_i (term frequency) 表示词频即 t_i 在文档 d 中出现的次数, idf_i (inverse document frequency) 表示逆向文档词频, 即 t_i 区分文本 d 与文本集中其他文本的能力, 计算公式为式(3); 为适应本文算法, 在本文中对 $tf * idf$ 的值做了规范化的处理, 处理的公式为式(4).

$$tf_i idf_i = tf_i(d) * idf_i(d) \quad (2)$$

$$idf_i(d) = \log(N/n_i) \quad (3)$$

$$TFIDF_i(d) = \frac{tf_i(d) * idf_i(d)}{\sqrt{\sum_{k=1}^n (tf_i(d) * idf_i(d))^2}} \quad (4)$$

式中, N 表示文本集包含的文本数目, n_i 表示文本集中出现 t_i 的次数, n 为文本中特征词数目.

2.2 文本相似性度量

文本表示成向量以后, 文本之间的相似度可以通过空间中向量间的几何关系来度量^[3]. 结合蚁群文本聚类算法中蚂蚁在观察半径内抬起和放下文本的特点, 本文采用 VSM 模型中的夹角余弦值来度量观察半径内文本之间的相似度, 夹角余弦值越大, 则说明空间中两个向量的夹角越小, 两个文本越相似, 具体的计算公式如下:

$$\text{sim}(d_i, d_j) = \frac{\sum_{k=1}^m (w_{ik} w_{jk})}{\sum_{k=1}^m w_{ik}^2 \sum_{k=1}^m w_{jk}^2} \quad (5)$$

式中, d_i 表示第 i 个文本向量, w_{ik} 表示第 i 个文本向量的第 k 维值, m 表示维度

3 基本蚁群文本聚类算法

3.1 算法的基本思想

首先构建二维网格空间, 对需要的参数初始化, 将文本随机放置于网格中, 每个网格的文本数目不能超过一个^[3], 然后把一定数目蚂蚁置于网格中, 每只蚂蚁随机选择一个文本, 计算该文本与局部区域内所有文本相似度得到概率, 决定蚂蚁是否“抬起”、“放下”该文本, 经过多次迭代, 所有文本按照其相似

性聚集在一起^[4-6].

局部区域相似度计算公式为

$$f(d_i) = \frac{1}{s^2} \sum_{d_j \in \text{Neighs} \times s(r)} \left| 1 - \frac{1 - \text{sim}(d_i, d_j)}{\alpha} \right| \quad (6)$$

如果 $f(d_i) < 0$, 则 $f(d_i) = 0$

式中, $f(d_i)$ 表示群体相似度函数, α 表示群体相似系数, $s \times s$ 表示单元 r 的领域面积.

蚂蚁抬起和放下文本的概率计算公式分别为

$$P_p(d_i) = \left(\frac{k_1}{k_1 + f(d_i)} \right)^2 \quad (7)$$

式中, k_1 是阈值常量.

$$P_d(d_i) = \begin{cases} 2f(d_i), & \text{if } f(d_i) < k_2 \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

式中, k_2 是阈值常量.

3.2 算法的描述

蚁群文本聚类算法描述伪代码如下所示.

输入: 参数 s , 最大迭代次数 $t_{\max}$, k_1 , k_2 , α 、蚂蚁数目 k , 网格的区域大小 $Z \times Z$

输出: 显示聚类结果

```

1 初始化: 创建  $Z \times Z$  网格, 随机生成文本和蚂蚁的位置
2 For  $i$  to  $t_{\max}$  do
3   For  $j$  to  $k$  do
4     If(蚂蚁为空载 and 当前网格单元存在文本数据点  $d_i$ ) Then
5       根据式(6)计算  $f(d_i)$ , 根据式(7)计算  $P_p(d_i)$ 
       随机产生  $P_1 \in [0, 1]$ 
6       If( $P_p(d_i) \geq P_1$ ) Then 抬起该文本并为蚂蚁
       随机分配一新位置
7       Else 不抬起并为蚂蚁随机分配另一个新位置
8       End If
9     End If
10    Else If(蚂蚁为负载 and 当前网格单元不存在文本数据点) Then
11      根据式(6)计算  $f(d_i)$ , 根据式(8)计算  $P_d(d_i)$ 
      随机产生  $P_2 \in [0, 1]$ 
12      If( $P_p(d_i) \geq P_2$ ) Then 放下该文本并为蚂蚁
      随机分配新文本
13      Else 不放下并为蚂蚁随机分配一个新位置
14      End If
15    End If
16    蚂蚁随机移动到不被其他蚂蚁占据的邻近网格中
17  End For
18 End For
```

4 改进蚁群文本聚类算法

4.1 算法改进

(1) 修改算法终止条件. 蚁群文本聚类算法的结束条件是算法到达最大迭代次数, 也就是说算法

没有考虑在某次迭代中聚类的效果已经符合要求,而过多的迭代将不再具有意义.为了解决这个问题,在每一次迭代之后判断聚类 i 的 chsv_i ,聚类 i 和聚类 j 的 $\text{dissim}(i,j)$,分别设置它们的最大阈值为 CHSV 和 DISSIM,当 $\max\{\text{chsv}_i\} \geq \text{CHSV}$ 或者 $\min\{\text{dissim}(i,j)\} \geq \text{DISSIM}$ 时算法将终止.

$$\text{chsv}_i = \frac{1}{d_{\text{num}}} \sum_{j=1}^{d_{\text{num}}} (d_{ij} - d_{\text{center}})^2 \quad (9)$$

式中, d_{num} 为聚类 i 的文本总数, d_{ij} 聚类 i 的第 j 个文本向量, d_{center} 表示聚类 i 中心.

$$\text{dissim}(i,j) = (\mathbf{c}_i - \mathbf{c}_j)^2 \quad (10)$$

式中, $\mathbf{c}_i$ 和 $\mathbf{c}_j$ 为聚类 i 和聚类 j 的中心向量.

(2) 动态调整蚂蚁观察半径. 从式(6)可以知道,观察半径 s 对 $f(d_i)$ 有着直接的影响, s 越大,算法得到的聚类效果越粗糙,反之 s 过小,算法中蚂蚁的移动效率变低.为了平衡聚类精度和蚂蚁移动效率,对蚂蚁设置负载记录标记为 loadcount ,如果当前蚂蚁为负载,使该值每次递增 1,当 $\text{loadcount} \geq \text{maxloadcount}$ 时,则使当前蚂蚁的观察半径设置为 $1.5 \times s$. 其次在算法每次到达 $t_{\min}$ 次数时,使用式(11)所示的方法进行线性递减:

$$s = \text{Int}(-N * s + M) \quad (11)$$

式中, $\text{Int}()$ 是取整函数, M, N 为根据经验取值.

(3) 改变蚂蚁移动策略

基本算法中蚂蚁是随机移动的,本文使用三个规则控制蚂蚁的移动位置,具体规则如下所示.

规则 1: 如果当前蚂蚁将文本 d_i 成功放下时,蚂蚁的移动范围为 $x: [x_{\text{low}}, x_{\text{high}}]; y: [y_{\text{low}}, y_{\text{high}}]$, 假设蚂蚁当前的位置为 (x_1, y_1) , 设此时随机生成的放下阈值为 P_1 , 放下概率为 P_d , 则按如下公式计算 x, y 范围, 网格区域 Z .

$$x_{\text{low}} = \max\{(x_1 - \frac{Z}{2} * \frac{P_1 - P_d}{P_1}), 0\} \quad (12)$$

$$x_{\text{high}} = \min\{(x_1 + \frac{Z}{2} * \frac{P_1 - P_d}{P_1}), Z\} \quad (13)$$

$$y_{\text{low}} = \max\{(y_1 - \frac{Z}{2} * \frac{P_1 - P_d}{P_1}), 0\} \quad (14)$$

$$y_{\text{high}} = \min\{(y_1 + \frac{Z}{2} * \frac{P_1 - P_d}{P_1}), Z\} \quad (15)$$

规则 2: 如果当前的蚂蚁已经拾起文本 d_i , 则搜索所有的文本向量找到与 d_i 相似度最大的文本向量 d_j , 以 d_j 为中心, 以半径 $R = 1$ 递增的方式寻找网格内没有被其他文本向量占据的位置, 如果找到该位置, 则放下 d_i , 如果没有找到, 则以 d_j 为中心, 在 $s \times s$ 的区域内找到与 d_j 最不相似的文本 d_z , 将 d_z

随机分配位置并将 d_i 放在 d_z 处.

规则 3: 在其他的条件下, 蚂蚁的移动范围为 $x: [x_{\text{low}}, x_{\text{high}}]; y: [y_{\text{low}}, y_{\text{high}}]$, 将蚂蚁移动到该区域内没有蚂蚁占据的位置, 假设蚂蚁当前的位置为 (x_0, y_0) , 蚂蚁的观察半径为 s , 网格区域 Z , 则按如下公式计算 x, y 范围:

$$x_{\text{low}} = x_0 - s/2 \quad (16)$$

$$\text{如果 } x_{\text{low}} < 0 \text{ 则 } x_{\text{low}} = 0$$

$$x_{\text{high}} = x_0 + s/2 \quad (17)$$

$$\text{如果 } x_{\text{high}} > Z \text{ 则 } x_{\text{high}} = Z$$

$$y_{\text{low}} = y_0 - s/2 \quad (18)$$

$$\text{如果 } y_{\text{low}} < 0 \text{ 则 } y_{\text{low}} = 0$$

$$y_{\text{high}} = y_0 + s/2 \quad (19)$$

$$\text{如果 } y_{\text{high}} > Z \text{ 则 } y_{\text{high}} = Z$$

4.2 改进算法描述

改进算法描述如下所示.

输入: 参数 s , 最大迭代次数 $t_{\max}$, k_1, k_2, α , 蚂蚁数目 k , 网格的区域大小 $Z \times Z$, CHSVDISSIM, M, N , maxloadcount , $t_{\min}$

输出: 显示聚类结果

```

1 初始化: 创建  $Z \times Z$  网格, 随机生成文本和蚂蚁的位置
2 For  $i$  to  $t_{\max}$  do
3   For  $j$  to  $k$  do
4     If (蚂蚁为空载 and 当前网格单元存在文本数据点  $d_i$ ) Then
5       根据式(6)计算  $f(d_i)$  和式(7)计算  $P_p(d_i)$  随机产生  $P_1 \in [0, 1]$ 
6       If ( $P_p(d_i) \geq P_1$ ) Then 拾起该文本并按移动规则 2 为蚂蚁分配一新位置
7       Else 不拾起并按移动规则 3 为蚂蚁分配一个新位置
8     End If
9   End If
10  Else If (蚂蚁为负载 and 当前网格单元不存在文本数据点) Then
11     根据式(6)计算  $f(d_i)$ , 根据式(8)计算  $P_d(d_i)$ , 随机产生  $P_2 \in [0, 1]$ 
12     If ( $P_p(d_i) \geq P_2$ ) Then 放下该文本并按规则 1 为蚂蚁分配新位置
13     Else 不放下并按移动规则 3 为蚂蚁分配一个新位置
14   End If
15 End For
16 If (蚂蚁负载 and  $\text{loadcount} \geq \text{maxloadcount}$ )
17   暂时扩大当前蚂蚁的观察半径  $1.5 \times s$ 
18 End If
19 按移动规则 3 为蚂蚁分配一个新位置

```

```

20   End For
21   阶段性更新  $s$  的值
22   IF ( $i \% t_{\min} = 0$ )
23      $s = \text{Int}(-N * s + M)$ 
24   End IF
25   IF ( $\max\{\text{chsv}_i\} \geq \text{CHSV}$  or  $\min\{\text{dissim}(i, j)\} \geq$ 
DISSIM
26     break;
27   End If
28   End For

```

5 仿真实验

本文仿真实验采用的数据集为复旦大学中文文本语料库,实验随机获取教育、医学、能源、经济、计算机各类别共 500 篇文档,对各类别进行测试,测试样本为 100 篇,实验的运行环境为 CPU: Pentium4 GHz, RAM: 2 GB, 操作系统为 Windows XP, 使用 F-Measure 作为算法结果的评价标准。

改进后算法的初始参数设定为 $k_1 = 0.1$, $k_2 = 0.15$, $\alpha = 0.35$, $t_{\min} = 1\ 000$, $s = 9$, $t_{\max} = 20\ 000$, $Z = 42$, $k = 45$, $\text{maxloadcount} = 200$, $M = 9$, $N = 0.25$ 。

改进前算法的初始参数设定为 $k_1 = 0.1$, $k_2 = 0.15$, $\alpha = 0.35$, $s = 9$, $t_{\max} = 20\ 000$, $Z = 42$, $k = 45$ 。

从图 1 可以看出改进后的算法 F-Measure 值明显高于改进前的值和 k -means, 改进后的算法提高了文本聚类的精度。这是因为 k -means 算法过于依赖于 k 值, 然而 k 值不容易确定, 如果 k 值偏大或者偏小, 都会影响聚类的效果。在基本算法中蚂蚁的移动是随机的、无目的性的, 所以文本聚类的精度也是较低的。改进后的算法在蚂蚁拾起和放下文本时, 对蚂蚁的移动方向进行了引导, 通过更改蚂蚁的移动策略, 使蚂蚁的移动具有了自适应的能力, 同一簇的文本能更好地聚集在一起, 提高了文本聚类的精度。

从图 2 可以看出在不同文本数目下, 改进后的算法聚类时间明显高于改进前。这是因为蚁群文本聚类算法的终止条件是由最大迭代次数决定且蚂蚁观察半径为静态值, 如果聚类的效果已经达到, 过多的迭代, 只会空耗时间。改进后的算法对迭代后的聚类效果进行评价, 如果达到预定的聚类效果, 则迭代终止。其次动态地改变蚂蚁的观察半径, 平衡了聚类的效果和蚂蚁的移动效率。实验结果表明, 改进蚁群文本聚类算法的文本聚类收敛速度明显提高, 而且文本聚类结果的精度也有了明显的改善。

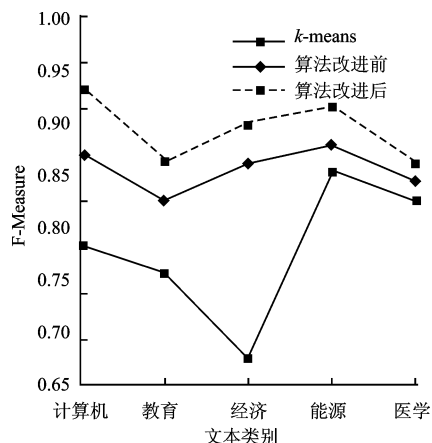


图1 各种算法的聚类结果比较

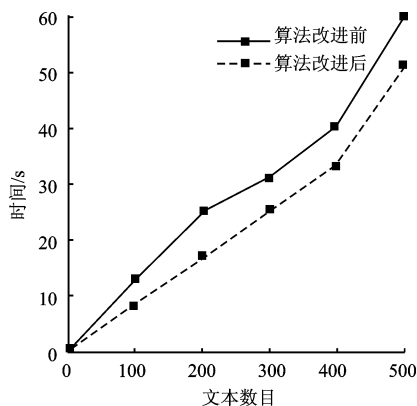


图2 算法改进前后的时间比较

6 结束语

本文对蚁群文本聚类算法进行了 3 个方面的改进: 修改终止条件, 使终止条件不再依赖单独的最大迭代次数; 动态调整蚂蚁观察半径, 平衡了聚类结果和蚂蚁移动效率; 改变蚂蚁移动策略, 使蚂蚁移动具有了目的性。仿真实验表明, 改进后的算法提高了文本聚类的效率, 相比 k -means、蚁群文本聚类算法, 获得了更好的文本聚类效果。

参考文献:

- [1] 唐东明. 聚类分析及其应用研究[D]. 四川: 电子科技大学, 2010.
- [2] 段海滨. 蚁群算法原理及应用[M]. 北京: 科学出版社, 2005
- [3] 吴凤慧, 成颖, 郑彦宁, 等. 文本聚类中文本表示和相似度计算研究综述[J]. 情报科学, 2012, 30(4): 622-627
- [4] Dorigo M, Bonabeau E, Theraulaz G. Ant algorithms and stigmergy[J]. Future Generation Computer System, 2000, 16(8): 851-871.

5 结束语

综上所述,凸包裁剪法只需在初始化阶段计算物体的投影凸包,然后用少量的额外存储空间来存储扇形边界点,在其后的系统运行过程中,只需要根据扇形边界线和视线的夹角就可以判断物体的可见性,一旦摄像机和物体发生变化,也只需要两次角度同侧异侧判断即可更新扇形边界点,消耗的处理时间很短。

从时间复杂度上看,凸包裁剪法比圆柱体裁剪法更复杂,但是在判断结果的精度上比圆柱体判断要高,这就意味着在同等条件下,凸包裁剪法能够过滤掉更多的不可见物体,阻止更多的无效三角形进入后续渲染步骤,提升渲染的整体效率。

参考文献:

[1] 王竹,束搏,邱显杰,等. 基于场景结合的大规模动态群体可见性计算方法[J]. 计算机辅助设计与图形学学报,2009,21(3):331-338.

[2] Andre LaMothe. Tricks of the 3D game programming

gurus:advanced 3D graphics and rasterization[M]. Indiana,USA;Sams Publishing,2003;531-538.

[3] 王超一,丁莹,张鑫,等. 一种基于投影的视锥体裁剪算法[J]. 系统仿真学报,2014,26(10):2412-2417.

[4] 吴思,郭立,郭安泰. 3D图形的裁剪算法及硬件实现技术研究[J]. 电子技术,2008,46(2):44-48.

[5] 黄建国,黄穗. 一种新的视截体裁剪算法[J]. 微计算机应用,2009,30(12):18-22.

[6] 吴文周,李利番,王结臣. 平面点集凸包 Graham 算法的改进[J]. 测绘科学,2010,35(6):123-125.

[7] 王杰臣. 2 维空间数据最小凸包生成算法优化[J]. 测绘学报,2002,46(1):82-86.

[8] 刘光惠,陈传波. 求解简单多边形和平面点集凸包的新算法[J]. 计算机科学,2007,34(12):222-226.

作者简介:

曾磊夫 男,(1981-),硕士研究生. 研究方向为计算机图形、光电通信. E-mail:278984134@qq. com.

刘 爽 女,(1970-),博士生导师. 研究方向为光通信与集成光学、光电探测、光电能源技术

(上接第 84 页)

[5] Hong Jiang, YuQing-song, Gong Yu. An improved ant colony clustering algorithm[C] // Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on pages. Yantai;IEEE;2010; 2368-2372 .

[6] 王小华,沈杰,王荣波. 一种新的基于蚁群和凝聚的混合聚类算法[J]. 杭州电子科技大学学报,2012, 30

(1):26-29.

作者简介:

张海涛 男,(1990-),硕士生. 研究方向为数据挖掘. E-mail:937885321@qq. com

周爱武 女,(1965-),副教授,硕士生导师. 研究方向为智能计算、信息系统安全与数据库与 WEB 技术.