# 基于相似度的蚁群聚类算法*

沈兴鑫[1]　杨余旺[1]　肖高权[2]　徐益民[1]　陈响洲[1]
（1. 南京理工大学计算机科学与工程学院　南京　210094）
（2. 湖南云箭集团有限公司　怀化　419500）

**摘　要**　针对于蚁群聚类算法在搬运数据项过程中随机选择移动位置时，由于无效移动导致的算法收敛速度缓慢等缺陷，论文提出了一种基于相似度的蚁群聚类算法。通过设计相似度矩阵，基于相似移动机制将蚂蚁随机移动方式优化为按照相似度矩阵规则实施目的性的关联。实验选取 Iis、Wine、Haberman 和 Balance-scale 四种经典数据集，相较于现有的 LF 算法及 GACC 算法，结果表明在蚂蚁空载率都为 90%的条件下，论文提出的 SMACC 算法的迭代次数明显降低，均体现出较优的聚类速率。

**关键词**　蚁群聚类；相似度矩阵；相似移动；高速率
**中图分类号**　TP301.6　　**DOI**：10. 3969/j. issn. 1672-9722. 2021. 06. 004

# Ant Colony Clustering Algorithm Based on Similarity

**SHEN Xingxin[1]　YANG Yuwang[1]　XIAO Gaoquan[2]　XU Yimin[1]　CHEN Xiangzhou[1]**
（1. School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing　210094）
（2. Hunan Yunjian Group Co., Ltd., Huaihua　419500）

**Abstract**　This paper proposes an ant colony clustering algorithm based on similarity when the ant colony clustering algorithm randomly selects the moving position during the process of moving data items and the slow convergence speed of the algorithm due to invalid movement. By designing the similarity matrix, the ant random movement method is optimized based on the similarity movement mechanism to implement the purpose association according to the similarity matrix rule. Four classic data sets of Iis, Wine, Haberman and Balance-scale are selected in the experiment. Compared with the existing LF algorithm and GACC algorithm, the results show that the SMACC algorithm proposed in this paper is under the condition that the ant no-load rate is 90%. The number of iterations is significantly reduced, and it both shows better clustering rates.

**Key Words**　ant colony clustering, similarity matrix, similarity movement, high rate
**Class Number**　TP301.6

## 1　引言

聚类[1]（Clustering）是一种重要的数据挖掘分析方法，主要目标是在海量数据中找出相似的数据并按照相似度分为不同的类，从而使得一个集群内的数据项彼此相似，不同集合中的数据尽量不同。从生物学到社会学再到计算机科学等众多领域都存在相同的需求，因此聚类算法得到了广泛的应用。

蚁群聚类（Ant Colony Clustering）是一种受蚁群启发的聚类算法。意大利学者 Dorigo M[2]在模仿蚂蚁群体行为的基础上提出了蚁群算法，根据蚁群的信息素机制寻找蚁穴和食物间的最短路径，并有效地解决 TSP 问题。Deneubourg J[3]基于人工蚁群模型结合蚂蚁的堆尸行为提出了聚类算法（Basic ant colony clustering model，BM）。Lumer E D 和 Faieta B[4]在 BM 模型的基础上改变蚂蚁移动速度提出了 LF 模型，很好地解决了大数据量时的聚类问题。

相比于其他的聚类算法，蚁群聚类有以下几个优点，如灵活性、鲁棒性、分布性和自组织性。因此该算法被更多的人研究并改进。Bin W 和 Zhongzhi S[5]研究了相似系数，并提出了一种更简单的概率转换函数。王慧和甘泉[6]加入了参数自适应调整策略，提高了蚁群聚类的准确性。乔少杰[7]研究

* **收稿日期**：2020 年 11 月 10 日，**修回日期**：2020 年 12 月 22 日
**作者简介**：沈兴鑫，男，硕士研究生，研究方向：数据挖掘。

了蚁群的分布式模型在文本聚类中的应用。Tan S C 等[8]提出了一种简化的基于蚂蚁的聚类方法，该方法基于现有的基于蚂蚁的聚类系统的研究。林金灼[9]结合了主成分分析方法（Principal Component Analysis，PCA），提高了蚁群的聚类质量。Tao 等[10]重新定义了两个数据对象之间的距离，并改进了蚂蚁丢弃和拾取数据对象的策略，从而提出了一种改进的蚁群聚类算法。Xu X 等[11]提出了一种约束蚂蚁聚类算法，该算法嵌入了基于随机游走的启发式步行机制，以解决约束聚类问题。张梦佳[12]采用实时信息素更新规则，提高了蚁群聚类的收敛速度，同时提高了准确度。周峰[13]提出一种结合蚁群聚类和模糊均值聚类思想的聚类算法算法，该算法具有全局优化能力，优化了蚁群聚类易陷入局部最优的缺点。赵宝江[14]利用蚁群聚类算法来进行结构辨识，确定系统的模糊空间和模糊规则数，实现非线性系统的辨识，辨识精度高，可当作复杂系统建模的一种有效手段。

虽然以上方法优化了蚁群聚类算法，但是蚁群聚类前期收敛速度慢的问题未有效的解决。由于蚂蚁随机移动数据项的位置存在多次无效的移动，导致算法计算效率和准确性较低，对于复杂的工程，该问题更为突出。为克服这些缺点，本文提出了一种新的蚁群聚类算法，通过添加相似度矩阵将原算法中蚂蚁随机移动修改成按照相似度矩阵有目的地移动，蚂蚁随机关联修改成有目的地关联。通过实际数据集验证了新算法的性能，并与先前研究中提出的蚁群聚类算法和其他算法进行了比较，验证了新算法的有效性。

## 2　蚁群聚类算法

蚁群聚类算法是一种群体智能方法，受到蚁群堆积尸体和排序幼虫行为的启发而形成的一种聚类方法。因其灵活性、鲁棒性、分散性和自组织性等特点被广泛研究。

### 2.1　LF 算法基本原理

在 LF 模型中，将蚂蚁的尸体建模为需要聚类的数据项，蚂蚁建模为在环境中随机移动的代理，蚂蚁移动的平面建模为一个具有边界条件的二维网格。分散在平面中的数据项通过代理拾取、搬运和放下从而将相似的数据项放在一起。数据项的取放概率受到该数据项与邻域内其他数据项的相似性和密度的影响，在拾取时，相似度越低越有可能拾取，相反，在放下时，相似度越高越可能放下。因此在网格上对数据项进行排列，使得相似度越高的数据项在平面上越靠近。

### 2.2　蚁群聚类算法模型

首先，将 $N$ 个数据项随机地映射到一个 $M\times M$ 的二维平面中，并将 $E$ 只蚂蚁分散到数据平面中并为每只蚂蚁 $e_i$ 关联一个数据项，即形成蚂蚁到数据项映射：

$$F:\ E\to N \tag{1}$$

其中 $E$ 是蚂蚁集合，$N$ 是数据项集合，$F$ 是蚂蚁到数据项的随机映射函数。

再计算出当前数据项与邻域内其他数据项的相似度 $f(e_i)$，相似度计算式（2）所示。

$$f(e_i)=\begin{cases}\dfrac{1}{s^2}\times\displaystyle\sum_{e_j\in Neigh_{s\times s}(r)}\left[1-\dfrac{d(e_i,e_j)}{\alpha\left(1+(v-1)/v_{\max}\right)}\right] & f(e_i)>0\\ 0 & \text{otherwise}\end{cases} \tag{2}$$

其中 $\alpha$ 是自定义的参数，用来调节蚂蚁间的相似度。$V$ 定义了蚂蚁的移动速度，$v_{\max}$ 代表了最大速度，$V$ 随机分布在 $[1,v_{\max}]$ 中。$s$ 是自定义的蚂蚁的搜索长度。$Neigh_{s\times s}(r)$ 代表位置 $r$ 的周围 $s\times s$ 面积。$d(e_i,e_j)$ 是 $e_i$ 和 $e_j$ 之间的距离。$d(e_i,e_j)$ 采用欧式距离，公式如下：

$$d(e_i,e_j)=\sqrt{\sum_{k=1}^{m}\left(e_{ik}-e_{jk}\right)^2} \tag{3}$$

其中 $m$ 是数据项的维度。

当蚂蚁处于空载状态需要拾起数据项时，按照如下公式计算拾起概率 $p_p$：

$$p_p=1-\tanh\left(f(e_i)\right) \tag{4}$$

其中 $f(e_i)$ 是 $e_i$ 的相似函数，$\tanh(x)$ 函数的定义如式（4）所示。

$$\tanh(x)=\frac{1-e^{-cx}}{1+e^{-cx}} \tag{5}$$

其中 $c$ 为自定义的常量，用来调节算法汇聚的速度。

当蚂蚁处于负载状态需要放下数据项时，按照如下公式计算放下概率 $p_d$：

$$p_d=\tanh\left(f(e_i)\right) \tag{6}$$

tanh 函数和 $p_p$ 概率曲线图如图 1 所示。

在本模型中，蚂蚁在拾起或者放下数据项后随机移动，数据项与领域内的数据项相似度可能不高，因此下次被移动的概率又会增加，从而降低了聚类的速率。同时，当负载的蚂蚁放下数据项或者空载的蚂蚁没有拾起数据项而需要重新关联数据项时，数据项的关联也是随机的，增大了再次关联的概率，导致聚类速率降低。

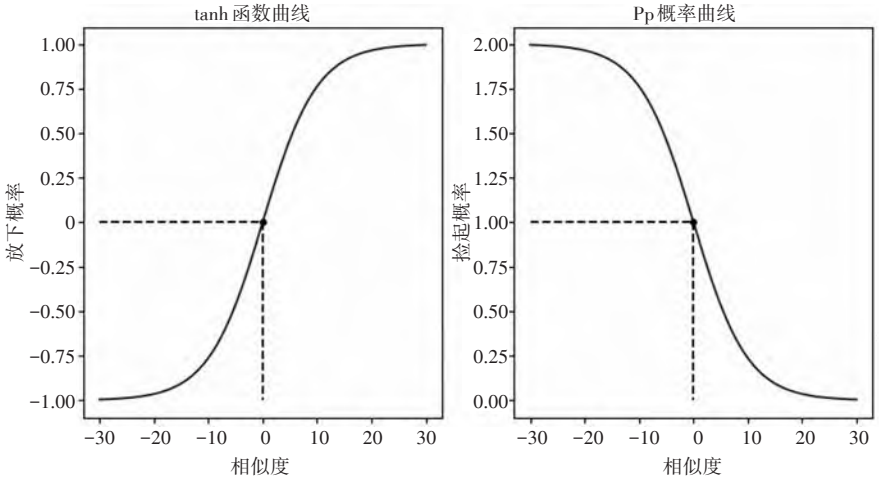



图 1 tanh 函数图和 $p_p$ 概率曲线图

针对以上两个缺点,本文在原有算法基础上设计了相似度矩阵,让蚂蚁按照相似度有目的地移动,同时在蚂蚁需要映射数据项时,根据相似度矩阵有选择的映射,从而加快聚类的速度。

## 3 改进的蚁群聚类算法

传统的蚁群算法移动蚁卵或者关联蚁卵时都是随机,存在无效移动与映射,导致聚类效率降低。改进的蚁群聚类算法在原有算法基础上设计了相似度矩阵,蚂蚁按照相似度有目标地移动,同时有选择地映射到数据项,从而加快聚类速度。

### 3.1 算法原理

蚁群聚类通过蚂蚁移动数据项,使相似度高的数据项在平面上尽可能靠近,从而形成聚类簇。在此过程中,将数据项移动到相似度高的区域附近可以大幅提高聚类速度。蚂蚁因放下或者未能拾起数据项时需要重新关联数据项。此时,当前数据项和邻域的相似度较高,需要移动的概率较低,因此,关联相似度低的数据项,增加操作次数可以提高聚类效率。

### 3.2 算法实现过程

#### 3.2.1 建立相似度序列矩阵

首先计算每个数据项和其他数据项之间的距离。由于余弦相似度计算简单、直接,对于单独判断两个数据对象的相似度准确率高,因此本算法采用余弦相似度来计算数据项间的距离,其定义如下:

$$Sim(e_i, e_j) = \frac{\sum_{k=1}^{n}(e_{ik} \cdot e_{jk})}{\sqrt{\sum_{k=1}^{n}(e_{ik})^2 \cdot \sum_{k=1}^{n}(e_{jk})^2}} \tag{7}$$

其中 $e_i$、$e_j$ 表示两个不同的数据项,$e_{ik}$、$e_{jk}$ 表示不同数据项所代表的矢量坐标值,$n$ 表示数据项的维度。

然后对余弦距离标准化,形成相似矩阵(Similarity Matrix)。该矩阵是一个 $N \times N$ 的对称矩阵,表示 $N$ 个数据项两两之间的相似度,形如:

$$Simmatrix = \begin{pmatrix} 0 & d_{21} & \cdots & d_{n1} \\ d_{12} & 0 & & d_{n2} \\ \vdots & & \ddots & \vdots \\ d_{1n} & d_{2n} & \cdots & 0 \end{pmatrix} \tag{8}$$

其中,$d_{ij}$ 是数据项 $i$ 和对象 $j$ 之间余弦距离的标准化表示。当数据项 $i$ 和数据项 $j$ 越相似,其值越接近 0;相反,两个数据项越不同,其值越接近 1。

再以相似矩阵为基础,以列为单位,按照相似度从高到地的顺序进行排序,形成相似度序列矩阵,形如以下公式:

$$Indexmatix = f(Simmatrix) = f\left(\begin{bmatrix} 0 & d_{21} & \cdots & d_{n1} \\ d_{12} & 0 & & d_{n2} \\ \vdots & & \ddots & \vdots \\ d_{1n} & d_{2n} & \cdots & 0 \end{bmatrix}\right) = \begin{pmatrix} s_{11} & s_{21} & \cdots & s_{n1} \\ s_{12} & s_{22} & & s_{n2} \\ \vdots & & \ddots & \vdots \\ s_{1n} & s_{2n} & \cdots & s_{nn} \end{pmatrix} \tag{9}$$

其中函数 $f(x)$ 是以矩阵 $x$ 为参数的快速排序函数,$s_{ij}$ 表示与第 $i$ 个数据项相似度排序第 $j$ 个的数据项的序号,$s_{ij} \in (1 \ \cdots \ n)$。数据项与该数据项的相似度最高,因此 $[s_{11} \ s_{12} \ \cdots \ s_{n1}] = [1 \ 2 \ \cdots \ n]$。

#### 3.2.2 提高蚂蚁的移动目的性

在蚂蚁移动数据项时,首先根据以下公式确定当前数据项的邻域。

$$N(ant_i) = N(x_i, y_i) =$$

$$\left\{x \bmod w(n), y \bmod h(n) \middle\| x-x_i \middle| \le S_x, \middle| y-y_i \middle| \le S_y \right\} \tag{10}$$

其中 $ant_i$ 表示第 $i$ 只蚂蚁，位于 $(x_i, y_i)$，$S_x$ 和 $S_y$ 分别代表该蚂蚁水平方向和竖直方向的视野。数据项领域如图2所示。

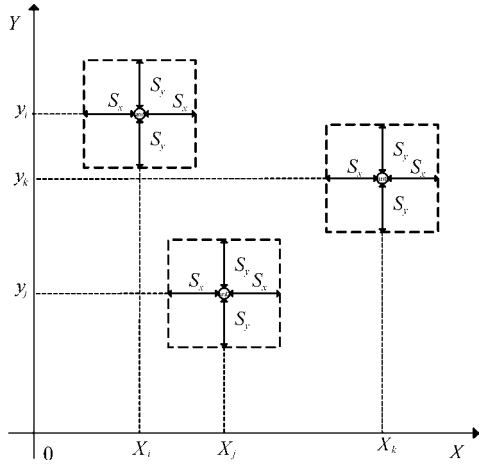



图2　数据项领域范围

然后确定邻域中的数据项，即 $e_j \in N(ant_i)$，再按照相似度序列矩阵找出邻域中相似度最高的数据项，将当前数据项移动到该数据项附近。

Algorithm 1：Similar Move
1）Input：Index of current spawn
2）Ouput：Index of most similar spawn
3）Calculate Neighbor Area（ $N(o_i)$ ）
4）Get column $i$ of Indexmatix
5）For $j$=1；$j$ < Spawn number；$j$++
6）　$k \leftarrow Indexmatix[i][j]$
7）If $o_k \in N(o_i)$
8）　Move $o_i$ nearby $o_k$
9）　Return $k$
10）　Else
11）　Continue
12）　End if
13）End for
14）Return Spawn number−1

3.2.3　增加关联的目的性

按照相似度序列矩阵找到和当前数据项最不相似的数据项并映射到该蚂蚁，伪代码如下：

Algorithm 2：Dissimilar Mapping
1）Input：Index of current ant $i$
2）Output：Index of most dissimilar spawn $k$
3）Get column $i$ of Indexmatix
4）For $j$=Spawn number−1；$j$>=1；$j$−−
5）　$k \leftarrow Indexmatix[i][j]$
6）If $o_k$ not occupied
7）　Map current $ant_i$ to $o_k$
8）　Return $k$
9）Else
10）　Continue
11）End if
12）End for

## 4　实验结果与分析

为验证改进算法的有效性，本文设计了仿真实验，通过不同数据集和不同算法的实验结果对比，验证了本算法的有效性。

### 4.1　实验环境与实验数据集

实验所用的数据集：本实验采用了UCI数据集中最常用的Iris，Wine，Haberman，Balance-scale数据集，数据集的简介如表1所示。

表1　样本参数

| 数据集 | 样本个数 | 属性个数 | 类别个数 |
|---|---|---|---|
| Iris | 150 | 4 | 3 |
| Wine | 178 | 13 | 3 |
| Haberman | 306 | 3 | 2 |
| Balance-scale | 625 | 4 | 3 |

### 4.2　实验结果与分析

本文的算法采用了相似度矩阵（Similarity Matrix）提高算法的聚类效率，记改进的蚁群聚类算法为SMACC（Similarity Matrix Ant Colony Clustering）。

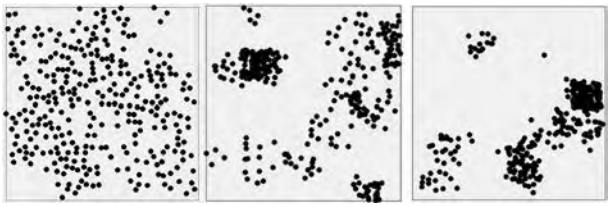
用LF算法，文献[15]中的GACC（G Ant Colony Clustering）算法和SMACC算法对Iris，Wine和Haberman数据集进行聚类，在同样迭代4000次的情况下，运行结果分别如图3、图4、图5所示。由结果可以看出，对Iris，Wine和Haberman数据集迭代4000次后，SMACC算法聚类结果最明显，GACC算法有明显的聚类趋势，而LF算法仅有聚类趋势，这表明本算法相对其他两种算法有较高的聚类效率。Balance-scale数据集的样本个数远大于其他数据集，在迭代4000次后很难比较聚类的效果，图6是三种算法迭代10000次后Balance-scale数据集的聚类结果。结果可以看出，SMACC在大数据集的聚类效果明显优于其他两种算法。

（a）LF算法　（b）GACC算法　（c）SMACC算法

图3　Iris数据集

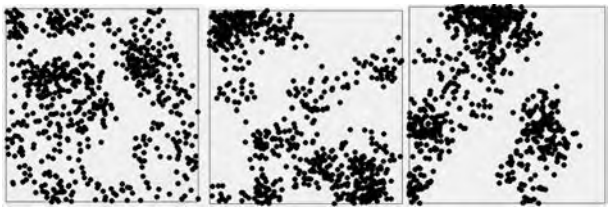
(a)LF算法 (b) GACC算法 (c) SMACC算法

图4 Wine数据集



(a)LF算法 (b) GACC算法 (c) SMACC算法

图5 Haberman数据集



(a)LF算法 (b) GACC算法 (c) SMACC算法

图6 Balance-scale数据集

Iris数据集和Wine数据集有相似的样本个数,但Wine数据集的属性个数远大于Iris数据集的属性个数。比较图1和图2可知,属性个数增加,聚类效果有所降低。但是,由于SMACC算法采用余弦距离计算数据项间的相似距离,余弦距离更有利于区分高维度的数据项,因此,SMACC在高维度数据项的聚类效果远优于其他两个算法。在相同的迭代次数下,Iris数据集的聚类效果比Wine数据集的聚类效果好。

在一次迭代过程中,如果放下或者未捡起数据项的蚂蚁数量占总量的90%时,表明聚类基本完成。三种算法分别在四种数据集上收敛时的六次平均迭代次数如表2所示。

如表2所示,使用相同数据集时,SMACC的平均迭代次数明显小于其他两种算法,该结果表明本算法有效地提高了聚类速度。Balance-scala数据集中,SMACC的迭代次数远小于其他两种算法,说明该算法更适用于大数据集的聚类。

表2 平均迭代次数比较

| 数据集 | LF | GACC | SMACC |
|---|---|---|---|
| Iris | 8000 | 6000 | 4500 |
| Wine | 12000 | 7800 | 6000 |
| Haberman | 17500 | 13000 | 9700 |
| Balance-scale | 38000 | 28000 | 21000 |

F-measure从查准率和查全率综合的角度衡量聚类算法的结果。其一般形式如式(2)所示:

$$F_\beta=\frac{(1+\beta^2)\times P\times R}{\left(\beta^2\times P\right)+R} \tag{11}$$

其中,$\beta$是查准率和查全率的权重比,$P$是查准率,$R$是查全率。

本文采用$F_1$值比较四种数据集上三种算法的聚类效果,结果如图7~10所示。

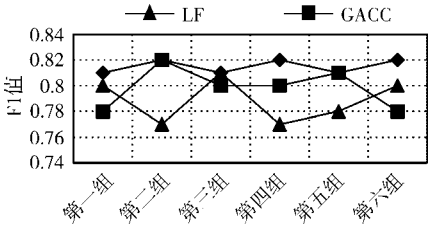



图7 Iris数据集

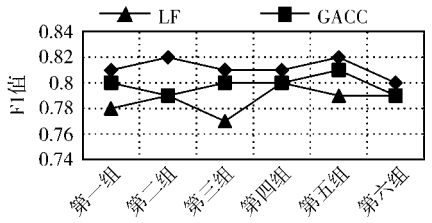



图8 Wine数据集

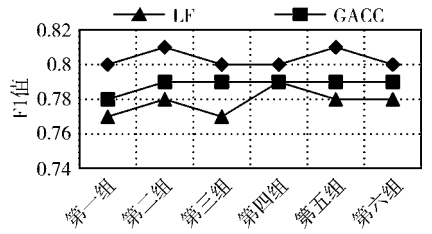



图9 Haberman数据集

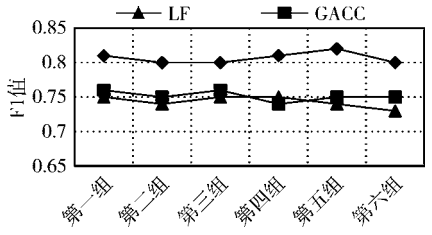



图10 Balance-scale数据集

比较三种聚类算法对四种不同数据集的$F_1$值,GACC和SMACC算法与LF算法的聚类效果总体相似,但GACC和SMACC略好于LF算法。随着数据集中样本数量的增加,三种算法的$F_1$值总体有所下降,由于SMACC算法在蚂蚁移动的过程中有目的地移动,所以数据簇中相似度总体比较高,所以在Balance-scale数据集中$F_1$值明显高于LF和GACC算法。同时,由于蚁群聚类是基于群体行为的聚类方法,随着数据量的增加,聚类结果的波动也减小。图8中,由于数据的维度增加,样本的

区分度有所降低，但SMACC采用余弦距离作为相似距离，所以 $F_1$ 值略高于其他两个算法。

## 5 结语

蚁群聚类是一种利用群体智能的聚类算法，其灵感来自蚁群聚集其尸体的行为。该算法具有鲁棒性强，适合分布式等优点，但是因为蚂蚁随机移动数据项而造成多次无效移动以及资源浪费。针对该问题，本文设计了相似度矩阵，使蚂蚁在相似度矩阵的指导下有目的地移动和关联，从而加快聚类速度。

仿真实验中，对比了本算法和其他两种算法对UCI数据集中四个实际数据集(Iris，Wine，Haberman，Balance-scale)的聚类性能。结果表明，新的蚁群聚类算法在保证聚类精度的基础上，能够以较高的速度解决聚类问题，同时具有很好的计算稳定性。但是SMACC依然存在局部优化的问题，在接下来的工作中希望通过参数的动态调整来解决该问题。

### 参考文献


[1] 孙吉贵，刘杰，赵连宇. 聚类算法研究[J]. 软件学报，2008(01)：48-61.
SUN Jigui，LIU Jie，ZHAO Lianyu. Clustering Algorithms Research[J]. Journal of Software，2008(01)：48-61.

[2] Dorigo M，Maniezzo V，Colorni A. Ant system：optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society，1996，26(1)：29-41.

[3] Deneubourg J，Goss S，Franks N，et al. The dynamics of collective sorting：robot-like ants and ant-like，robots[C]//International Conference on Simulation of Adaptive Behavior：The MIT Press，1991：102-108.

[4] Lumber E，Faieta B. Diversity and adaption in populations of clustering ants[C]//Proc. of the 3rd International Conference on Simulation of Adaptive Behavior：From Animals to Antimates，1994：501-508.

[5] Bin W，Zhongzhi S. A Clustering Algorithm Based on Swarm Intelligence[C]//2001 International Conferences on Info-tech and Info-net：Proceedings，2001：59-63.

[6] 王慧，甘泉. 一种基于信息素的蚁群聚类算法[J]. 太赫兹科学与电子信息学报，2016，14(03)：426-431.
WANG Hui，GAN Quan. An Ant Colony Clustering Algorithm Based on Pheromones[J]. Journal of Terahertz Science and Electronic Information Technology，2016，14(03)：426-431.

[7] 乔少杰，韩楠，金澈清，等. 基于Multi-Agent的分布式文本聚类模型[J]. 计算机学报，2018，41(08)：1709-1721.
QIAO Shaojie，HAN Nan，JIN Ceqing，et al. A Distributed Text Clustering Model Based on Multi-Agent[J]. Chinese Journal Of Computers，2016，14(03)：426-431.

[8] Tan S C，Ting K M，Teng S W. Simplifying and improving ant-based clustering[J]. Procedia Computer Science，2011，4：46-55.

[9] 林金灼，叶东毅. 基于蚁群聚类算法的优化与改进[J]. 计算机系统应用，2013，22(12)：93-99.
LIN Jinzhuo，YE Dongyi. Optimization and Improvement Based on Ant Colony Clustering Algorithm[J]. Computer Systems & Applications，2013，22(12)：93-99.

[10] Tao W，Ma Y，Tian J，et al. An Improved Ant Colony Clustering Algorithm[J]. Computer Simulation，2009，26(8)：179-183.

[11] Xu X，Pan Z，He P，et al. Constrained Clustering via Swarm Intelligence[M]. Bio-Inspired Computing and Applications：Springer Berlin Heidelberg，2011：92-98.

[12] 张梦佳，李秦，王菲菲. 基于蚁群觅食原理的聚类算法的研究及改进[J]. 洛阳理工学院学报(自然科学版)，2017，27(04)：60-64.
ZHANG Mengjia，LI Qin，WANG Feifei. Research and Improvement of Clustering Algorithm Based on Ant Colony Foraging Principle[J]. Journal of Luoyang Institute of Science and Technology (Natural Science Edition)，2017，27(04)：60-64.

[13] 周峰. 融合蚁堆聚类与模糊C-均值聚类的算法研究和分析[D]. 合肥：安徽大学，2012：5-8.
ZHOU Feng. Fusion ant heap of Clustering and Fuzzy C-means clustering algorithm Research and Analysis[D]. Hefei：Anhui University，2012：5-8.

[14] 赵宝江. 蚁群聚类算法的T-S模糊模型辨识[J]. 计算机工程与应用，2011，47(21)：153-156.
ZHAO Baojiang. Identification of T-S fuzzy models based on ant colony clustering algorithm[J]. Computer Engineering and Applications，2011，47(21)：153-156.

[15] 武书舟，闫丽娜，张秋艳. 基于改进蚁群算法的聚类分析方法研究[J]. 计算机与数字工程，2018，46(09)：1721-1725，1849.
WU Shuzhou，YAN Lina，ZHANG Qiuyan. Research on Clustering Algorithm Based on Improved Ant Colony Algorithm[J]. Computer & Digital Engineering，2018，46(09)：1721-1725，1849.