# Information Entropy-based Ant Clustering Algorithm

**Zhang Yi[1], Zhao Weili[2], Zhang Zhiguo[3]**

**[1] Shenyang University, Shenyang, China**

**[2] Shenyang Ligong University, Shenyang, China**

**[3] Neusoft Ltd., Shenyang, China**

**Abstract**

Ant-based clustering is a heuristic clustering method that draws inspiration from the behavior of ants in nature. We revisit these methods in the context of a concrete application and introduce some modifications that yield significant improvements in terms of both quality and efficiency. In this paper, we propose Information Entropy-based Ant Clustering (IEAC) and New Information Entropy-based Ant Clustering (NIEAC) algorithm. Firstly, we apply information entropy and new information entropy to model behaviors of agents, such as picking up and dropping objects. The new entropy function led to better quality clusters than non-entropy functions. Secondly, we introduce a number of modifications that improve the quality of the clustering solutions generated by the algorithm. We have made some experiments on real data sets and synthetic data sets. The results demonstrate that our algorithm has superiority in misclassification error rate and runtime over the classical algorithm.

***Keywords:*** *Ant Clustering, Entropy, Information Entropy, New Information Entropy*

## 1. Introduction

Sorting and clustering methods are inspired by the behavior of real ants which is among the earliest methods in ant-based meta-heuristics. It has the benefit of discovering clusters without any initial partitioning, and without knowing ahead of time how many clusters will be necessary. One of the first studies of ant-based clustering is found in [1], where populations of ant-like agents randomly moving onto a 2D grid are allowed to pick up and drop basic objects in such a way as to cluster them. Lumer and Faieta (LF) [2] have developed the basic model and applied it into exploratory data analysis. Bo Liu [3] applies information entropy (EAC) to modify the picking up and dropping regulations, fewer parameters are needed to set, and clustering speed is fast.

We present in this paper Information Entropy-based Ant Clustering (IEAC) and New Information Entropy-based Ant Clustering (NIEAC) algorithm which follows the basic ideas of LF and EAC. Our method is also based on ants' natural behaviors, but uses new information entropy to guide agents moving and picking up or dropping an item, and we introduce a number of modifications that improve the quality of the clustering solutions generated by the algorithm.

## 2. Ant-Based Clustering

Algorithm name: Ant-based Clustering (LF)

Initialize parameters: $t_{max}$, $N_a$
For every object $o_i$ do
  Place $o_i$ randomly on the grid
End For
For $n$=1 to $N_a$ do
  // $N_a$ is the number of agents
  Place agent at randomly selected site
End For
For $t$=1 to $t_{max}$ do
  // $t_{max}$ is the maximal times that an agent moves
  For $n$=1 to $N_a$ do
    If ((agent unladen) and (site occupied by object $o_i$)) then
      Compute $f(o_i)$ and $p_p(o_i)$
      Draw random real number $R$ between 0 and 1
      If ($R< p_p(o_i)$) then pick up $o_i$
        //picking up rule
    Else If ((agent carrying object $o_i$) and (site is empty)) then
      Compute $f(o_i)$ and $p_d(o_i)$
      Draw random real number $R$ between 0 and 1
      If ($R< p_d(o_i)$) then drop $o_i$
        //dropping rule
    End if
    Move to randomly selected neighbor site not occupied by other agent
  End For

End For

In EAC algorithm, entropy is taken as the criterion for an agent to pick up or drop items.

In the initial state (t=t0), select N0 objects from the current database, and distribute the N0 objects uniformly randomly on the Z×Z grid. Initialize all agents to be unladen.

A subspace with a cluster has lower entropy than a subspace without a cluster [5]. Inspired by this, introduce new information entropy into the clustering algorithm. Denote s×s as the region in which an agent lies. Assuming independence of attributes, the entropy of the s×s area including a set of objects is defined by equation (1), where p(x) is defined by equation (2), obj_num is the total number of objects in s×s; x_num is the number of objects whose attribute Xi has value x.

$$E(s^2) = -\sum_{i=1}^{n}\sum_{x\in X_i} p(x)\log p(x) \qquad (1)$$

$$p(x) = \frac{x\_num}{obj\_num} \qquad (2)$$

Algorithm name: Entropy Ant Clustering (EAC)

```
Initialize parameters: Z, s, tmax, Na
For every object oi do
   Place oi randomly on the plane of Z×Z
End For
For n=1 to Na do
                                   // Na is the number of agents
   Place agent at randomly selected site in Z×Z
End For
For t=1 to tmax do
            // tmax is the maximal times that an agent moves
   For n=1 to Na do
      If ((agent unladen) and (site occupied by object oi)) then
            Compute Information Entropy E1, E2
      If (E1> E2) then pick up oi
                                                //picking up rule
      Else If ((agent carrying object oi) and (site is empty)) then
            Compute Information Entropy E1, E2
      If (E1> E2) then drop oi
                                                //dropping rule
      End if
      Move to randomly selected neighbor site not occupied by other agent
   End For
End For
```

For each site (x,y) in Z×Z do
  Compute Information Entropy of the surrounding area s×s area
End For

## 3. Information Entropy Ant Clustering and New Information Entropy Ant Clustering

In LF and EAC algorithm, objects are selected and moved randomly, information cannot be utilized fully and two clusters cannot be merged effectively. In this section, we build upon the work of LF and EAC to develop an improved algorithm of ant-based clustering. We use new information entropy to guide agents moving and picking up or dropping an item, and we introduce a number of modifications that improve the quality of the clustering solutions generated by the algorithm. We now discuss our modifications that improve both performance and run-time.

**Information Entropy.** We use a formula for information entropy defined by equation (1).

**New Information Entropy.** In order to avoid the Logarithmic, we use a new formula for information entropy [4] defined by equation (3). It can increase the computing speed.

$$E_N(s^2) = \prod_{i\text{-}1}^{n}(1 + p(x)) \qquad x \in X_i \qquad (3)$$

**Increasing Radius of Perception.** By experiments, we find initial clusters form easily with short radius. But if keep radius unchanged, the speed of clustering will slow obviously, it also inhibits the quick formation of clusters during the initial sorting phase. We therefore use a radius of perception that gradually increases over time. This saves computations in the first stage of the clustering process and prevents difficulties with the initial cluster formation. At the same time it accelerates the dissolution of preliminary small clusters. In the current implementation, we start with an initial perceptive radius of 1 and linearly increase it to be 5 in the end. This results in an improved spatial separation between clusters.

**Short-Term Memory.** The "short-term memory" is introduced by Lumer and Faieta in [2]. In their approach, each agent remembers the last few carried data items and their respective dropping positions. When a new data item is picked up, the position of the "best matching" memorized data item is used to bias the direction of the agent's random walk. We have extended this idea as follows.

We permit each ant to exploit its memory: An ant situated at grid cell $p$, and carrying a data item $i$, uses its memory to proceed to all remembered positions, one after the other. Each of them is evaluated by using the information entropy, that is, the suitability of each of them as a dropping site for the currently carried data item $i$ is examined. Subsequently, the ant returns to its starting point $p$.

Out of all evaluated positions, the one of "best match" is the grid cell for which the information entropy yields the lowest value. For the following step of the ant on the grid, we replace the use of a biased random walk with an agent "jump" directly to the position of "best match". If the jump is not made, the agent's memory is de-activated, and in future iterations it reverts to trying random dropping positions until it successfully drops the item.

**Stagnation control.** With complex data, early stagnation of the whole clustering process can be a problem. This is caused by outliers in the data sets. Due to their high dissimilarity to all other data elements, agents do not manage to dispose of these items once they had been picked. This results in blocked ants performing random walks on the grid without contributing to the sorting process. We therefore use a failure counter for each ant. After 100 unsuccessful dropping attempts an ant drops its load regardless of the similarity.

**Parameter Settings**. Ant-based clustering requires a number of different parameters to be set, some of which have been experimentally observed to be independent of the data. These include the number of agents $N_{ant}$ =10, the size of the agents' short-term memory $N_{memory}$ =10, the square grid $N_{table} = \sqrt{10N_{data}} \times \sqrt{10N_{data}}$, and $N_{iteration} = \frac{2000 N_{data}}{N_{ant}}$.

**Information Entropy-based Ant Clustering (IEAC) and New Information Entropy-based Ant Clustering (NIEAC).**

Initialize parameters: $N_{ant}$, $N_{interation}$, $R_{radius}$, $N_{memory}$, $F_{fail}$ and $N_{table}$.
For every object $o_i$ do
Place $o_i$ randomly on the plane of $N_{table}$
End For
For $i$=1 to $N_{ant}$ do
  Pick up object randomly
End For
For $t$=1 to $N_{interation}$ do
If ($t$ is times of $N_{interation}$ /5) then
  $R_{radius} \leftarrow R_{radius}$ +1
  For $j$=1 to $N_{ant}$ do
    If ($F_{fail(j)}$ =100) then
      While (ant$_j$ remembers every positions)
        If ((agent carry object $o_i$) and (agent drop object $o_i$)) then
          Compute Information Entropy $E1$, $E2$
        End If
      End While
      Search the lowest $E2$ satisfied with $E1>E2$
      Drop object $o_i$
      Update ant$_j$ $N_{memory(j)}$
      Pick up other object randomly
    Else
      Search empty grid in $N_{table}$
        Compute Information Entropy $E1$, $E2$
      If ($E1>E2$) then
        Drop object $o_i$
      Update ant$_j$ $N_{memory(j)}$
      Pick up other object randomly
      Else
        $F_{fail(i)} \leftarrow F_{fail(i)}$+1
      End If
    End If
End For
End If
End For

## 4. Experimental results and conclusion

In EAC algorithm, To assess the sensitivity of the algorithm to the proposed variation, we performed an experiment with a modified version of the well-known four classes data set proposed by Lumer and Faieta [2] to study IEAC and NIEAC, which corresponds to four distributions of 25 data points each, defined by Gaussian probability density functions with various means $\mu$ and fixed standard deviation $\sigma$=1.5, $G(\mu,\sigma)$, as follows:

$A = [x \infty G(0,1.5), y \infty G(0,1.5)]$;
$B = [x \infty G(0,1.5), y \infty G(8,1.5)]$;
$C = [x \infty G(8,1.5), y \infty G(0,1.5)]$;
$D = [x \infty G(8,1.5), y \infty G(8,1.5)]$.

We apply the four algorithms (LF, EAC, IEAC and NIEAC) to the data, after 160000 iterations; obtain the following results of clustering and the error rate.

The average qualities of the clusters produced by the three algorithms are shown in Figure 1 to Figure 3. Clearly,

NIEAC performs significantly better than the other two. Furthermore, we evaluate the obtained partitioning using the F-Measure [7], which combines information on the purity and the completeness of the generated clusters. LF is 0.2344, EAC is 0.5146 and NIEAC is 0.9892. The results demonstrate that our algorithm is feasible.

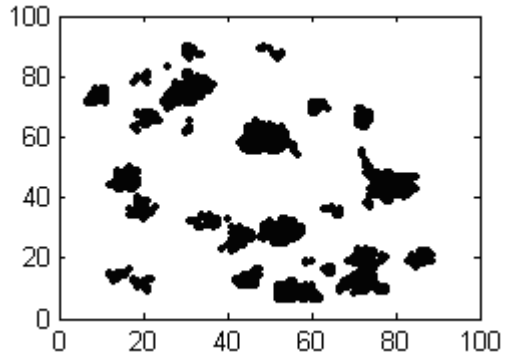

Figure 1. Cluster of LF

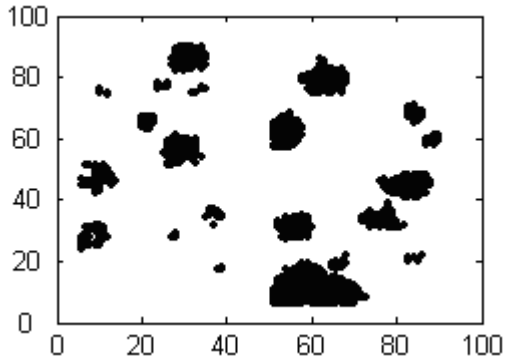

Figure 2. Clusters of EAC

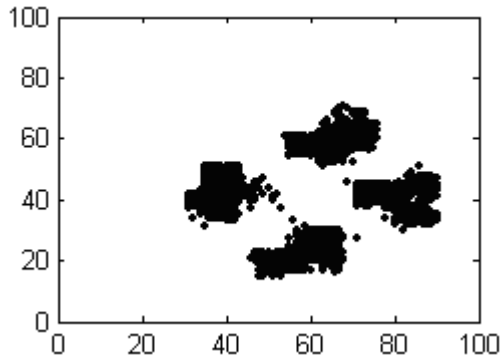

Figure 3. Clusters of IEAC

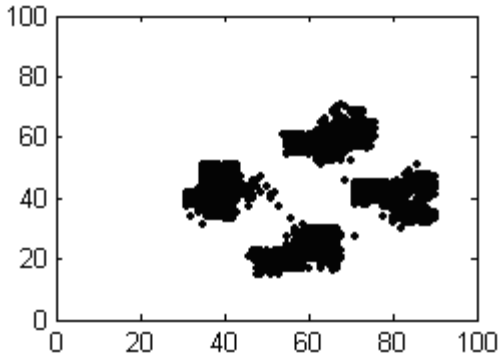

Figure 4. Clusters of NIEAC

In order to show our algorithm's effectiveness, we conducted experiments on four data sets from the UCI repository [6] as follows.

Table 1. Databases

| Database | Data number | Attribute number | Cluster number |
|---|---|---|---|
| tic-tac-toe | 958 | 9 | 2 |
| hayes-roth | 132 | 4 | 3 |
| balance | 625 | 4 | 3 |
| liver-disorders | 345 | 6 | 2 |

We applied the four algorithms (LF, EAC, NIEAC and K-means) to the databases; evaluate the obtained partitioning using the F-Measure. Table 2 showed that our method resulted in slightly high quality clusters.

Table 2. F-Measure of the four algorithms

| algorithm | tic-tac-toe | hayes-roth | balance | liver-disorders |
|---|---|---|---|---|
| LF | 0.22326 | 0.50715 | 0.23019 | 0.31337 |
| EAC | 0.26004 | 0.42906 | 0.13704 | 0.33084 |
| IEAC | 0.55993 | 0.49022 | 0.29904 | 0.49906 |
| NIEAC | 0.56699 | 0.48987 | 0.30045 | 0.5069 |
| K-means | 0.40074 | 0.4326 | 0.597 | 0.64239 |

## 5. Conclusion

In this paper we have introduced two new algorithms for Information Entropy-based ant clustering (IEAC) and New Information Entropy-based ant clustering (NIEAC) permits its direction application to numerical data sets. The results demonstrate that our algorithms have superiority in misclassification error rate and runtime over the classical algorithm.

## Acknowledgement

Project was supported by National Natural Science Foundation of China (No. 31000665).

## References

[1] J.L. Deneubourg, S. Goss, N. Franks, C. Detrain, and L. Chretien: The Dynamics of Collective Sorting: Robot-Like Ant and Ant-Like Robot, Proceedings First Conference on Simulation of Adaptive Behavior: From Animals to Animats, edited by J.A. Meyer and S.W. Wilson, Cambridge, MA:MIT Press, 1991, 356-365.

[2] E. Lumer, and B. Faieta: Diversity and Adaptation in Populations of Clustering Ants, Proceedings Third International Conference on Simulation of Adaptive Behavior: From Animal to Animats 3, Cambridge, MA: MIT Press, 1994, 499-508.

[3] Bo Liu, and Jiuhui Pan: Incremental Clustering Based on Swarm Intelligence, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, 189-196.

[4] Chengmao Wu: A new information entropy definition and its application in image segmentation, JOURNAL OF XI'AN UNIVERSITY OF POST AND TELECOMMUNICATIONS, Vol.14 No 1, Jan. 2009.

[5] Daniel Barbará, Julia Couto, Yi Li: COOLCAT: An Entropy-based Algorithm for Categorical Clustering, Proceedings of the Eleventh International Conference on Information and Knowledge management, 2002, 582-589.

[6] P.M. Murpy, and D.W. Aha: UCI repository of machine learning databases [EB/OL].

[7] http://www.ics.uci.deu/mlearn/ML-Repository.html, Irvine, CA: University of California, 1998.

[8] C. van Rijsbergen. Information Retrieval, 2nd edition. Butterworths, London, UK, 1979.

**Zhang Yi** has over two decades of experience in the field of Information Technology. He specializes in setting up Global R&D and innovation. He is an expert in the area of an Electronics and Telecommunications. He is an Electronics and Telecommunications Engineer from Shen Yang University. His research interest is in the area of Electronics and Telecommunications.