A Multiple Pheromone Algorithm for Cluster Analysis

Jan Chircop¹, Christopher D. Buckingham²

¹Computer Science Department
Aston University Aston Triangle,
Birmingham B4 7 ET, UK
janchircop@gmail.com

²Computer Science Department,
Aston University Aston Triangle,
Birmingham B4 7 ET, UK
C.D.Buckingham@aston.ac.uk

Abstract

Ant colony optimisation algorithms (ACO) work via a process called stigmergy in which ants deposit pheromone traces in order to influence foraging patterns. Pheromone traces are picked up and followed by other ants but they evaporate over time. Paths with more pheromone will survive longer and have a higher chance of getting followed and reinforced whilst weaker traces simply fade away. The premise behind the proposed Multiple Pheromone Algorithm for Cluster Analysis (MPACA) is that ants detect individual features of objects in space and deposit pheromone traces that guide towards these features. Each ant starts off by looking for a particular feature but they can combine with ants looking for other features if the match of their paths is above a given threshold. This enables ants to detect and deposit pheromone corresponding to feature combinations and provides the colony with more powerful cluster analysis and classification tools.

The basic elements of MPACA are that: (i) at the start of the learning process, every object has at least one ant assigned to it for each feature; (ii) each ant searches for other objects with a matching feature value; (iii) a pheromone is laid down whenever an ant has found an object with a matching feature; (iv) if ants detecting different features find their paths are matching above a certain level, they will combine and start looking for the conjunction of features; and (v) ants become members of the same colony when the population density of ants in the area is above a threshold value. This paper explains the algorithm and explores its potential effectiveness for cluster analysis.

Key words

1 Background

Cluster analysis is the task of partitioning data sets into clusters with common likeness so that a data element shares more similarities with its member group rather than any other group. Clustering is often a complex problem because of ambiguous boundaries between classes. When does a plant become a tree rather than a shrub? This classic fuzzy problem combined with uncertainty about what classes one is even expecting from a large set of data has led to a variety of approaches for optimising clusters, including ones that are modelled on insect behaviour, which is the approach used here. Ant Colony Optimisation (ACO), as introduced by Marco Dorigo in his Ph.D. thesis [1], is a probabilistic technique for solving computational problems that can be reduced to finding the best paths through graphs, in which Dorigo uses artificial ants laying pheromone trails traversing these nodes. The ants choose their paths based on probabilities depending on the strengths of pheromone trails that have been previously laid, and a distance function. Path optimisation is based on the fact that paths gradually weaken and disappear due to pheromone evaporation; the smaller the distance the shorter is the time for evaporation. Shorter paths tend to accumulate more pheromone, which in turn attracts more ants in a positive feedback loop. Through this indirect pheromone-driven communication, ant colonies are capable of solving a multitude of optimisation problems. Progress in this field is summarised by Wu et al. [2] and some representative studies will be referenced here. Manfrin et al. used parallel implementations of ACO [3], [4]. ACO has been applied to classification problems, as shown by Parpinelli et al. in the AntMiner algorithm [5], [6]. The AntMiner algorithm was used by a number of other researchers, such as Ji et al. in a sequential covering approach to discover a list of...
classification rules covering all training cases used [7]. Roselin and Thangavel apply it for classifying mammograms [8] and Moayed for web page classification [9]. In AntMiner+, Martens et al. show how ACO is applied to data mining in order to extract rule-based classifiers [10]. The cAnt-Miner2 is used by Michelakos et al. to extend the approach to coping with continuous attributes, a superior process to the cAnt-Miner method [11]. Other data mining applications include an approach by Zhu et al. where they apply a discretization algorithm based on an information-distance criterion and ACO for knowledge extraction on an industrial database [12]. The next section looks at how ACO has been adapted for clustering problems.

1.1 Clustering techniques

Bustos and Sellier's classification of clustering techniques [13].

1.1.1 Ant colony clustering techniques

Bustos and Sellier categorise a number of clustering techniques (Figure 1) and show the K-means and the C-means to be amongst the most popular [13]. Li et al. propose four types of ant colony clustering (ACC) algorithms, where the clustering method makes use of: (1) ants’ self-aggregation; (2) the principle of the formation of an ant heap; (3) the ant chemical identification system according to the classification model of ants’ nests; and (4) use of pheromone trails for modelling ant foraging theory [14].

1.1.2 The basic model and its derivatives

The Basic Model (BM) proposed in the early 1990s by Deneubourg [15], and later extended by Lumer and Faie-ta with their eponymous LF model [16], has been shown to be a set of effective clustering algorithms. Algorithms based on the basic model include an application to customer clustering analysis by Dai et al. [17] and the Improved Ant Colony Clustering (IACC) algorithm by Jiang et al. [18]. The LF method was applied by Zhang et al. to text based clustering [19] and further extensions were devised by Xu et al. with their Ants Sleeping Model (ASM) and Ants Clustering Algorithm (A4C) to solve clustering problems in data mining [20]. Qu and Liu introduce the Quick Ant Clustering algorithm QAC based on the same model [21]. However, these models focus on ant paradigm activities that are unrelated to the foraging process of ants.

1.1.3 Clustering based on the ant foraging model

Traditional ant colony clustering algorithms have been applied to automatic classification of image segmentation. The image data is split with separate ants searching for different attributes of cluster centres defined as the “food sources”. The study conducted by Hooshyar and Khayati presents retinal vessel detection using a fuzzy ant colony algorithm [22]. In their approach the image processing field is the digital image itself, such that ants occupy pixels within the image, and move around the image. The aim of the ants is to locate and map out boundaries within the image. Heuristic information is used to weigh the probability of an ant moving from one point to the next, together with the pheromone quotient.

Guo et al. use an application of ant clustering for edge feature extraction in city aerial images for building recognition [23]. Sun et al. present a hyperspectral image clustering algorithm that was improved by the k-means algorithm [24]. In general, ACC models assume that ants can roam the environment moving to objects from one grid point to the next depending on a similarity function. Alternatively, the ant can use information exchange to modify itself. Many of the approaches, though, have ant movements controlled by an overly mechanical similar.
This paper proposes a new type of ACC called the Multiple Pheromone Algorithm for Cluster Analysis (MPACA). It falls under the ant foraging category but it is important to highlight two major differences between MPACA and other ant algorithms. Firstly, MPACA uses a multiple pheromone system and secondly, it allows ants to detect feature combinations at the individual ant level.

MPACA also handles continuous dimensions by giving ants a parameterised detection range per dimension, where they respond to any object with values within the range. The core ant model for the MPACA algorithm is a food gathering ant, in which we replace the idea of a nest with the starting data point for the ant. This simply serves to define the particular feature value to which it responds and it then tries to find other data points with matching values so that it can link them with pheromone trails. To this extent, the nest and food become interchangeable.

2 Introducing MPACA

MPACA derives its name from the fact that rather than using a single pheromone for intra-ant communication, as is the case with typical ACO algorithms, multiple pheromones are used. Pheromones encourage other ants to follow them via a scent. In this particular model, each pheromone type indicates paths towards a specific feature value in the given search space. The domain is set up by defining a set of features that describe the objects, which are distributed in multidimensional space according to the specific values they have for each feature. Ants are then located at each object, with at least one ant for every feature value of the object. At this stage, there are as many colonies as there are objects in the hyperspace, with the number of ants in each colony being some multiple of the number of dimensions in space. The default setting will be one ant per feature of an object at each object, so a three-dimensional space will have three ants located at each object. Further experiments might suggest reasons for requiring a greater population density than this.

When the clustering process begins, each ant leaves a pheromone trail from its origin object as it searches for other objects in space. The pheromone will be a distinctive one for the specific value of the particular feature and the idea is that objects with similar feature values will have connecting trails. Furthermore, ants searching for different features but that follow similar paths will link up and combine their features so that they subsequently only respond to objects with the conjunctive set. This enables the ants to pick up feature combinations and provides colonies with the ability to detect non-linear relationships. More meaningfully, if an object of colour blue is always found next to a large object, then features colour=blue, size=large can be merged into one. At which point ants no longer are looking for just blue or large, but for the combined features blue AND large. In theory, this means the algorithms can solve the XOR problem, which depends on knowing how values of two separate features occur together. The XOR problem is important to machine learning and has been extensively discussed (e.g. Elizondo [25]).

A second merging operator governs the generation of ant colonies. It works in the same way as the process for merging features of ants except that it combines ants into the same colony, using a different merge threshold parameter, of course. The idea is that ants which frequently meet other ants will eventually become part of the same colony so that colonies build up in areas of high population density. Initially different sets or colonies of ants search for different pheromone scents, and colonies form varying pheromone patterns which in turn lead to clusters of ant densities in particular areas of search space. Consequently, multiple colonies utilising multiple pheromones can independently and asynchronously work in parallel.

As a result of pheromone distribution and ant swarming, paths connect objects and determine the densities of ants moving amongst those objects. This, in turn, determines the colonies and provides the emergent information about clusters. The two main forces at work in this algorithm are: (i) ant movement with pheromone deposition that increases the likelihood of ants searching for similar objects meeting up; and (ii) the process of merging ants based on the frequency of meeting other ants, which has parameters governing both when ants combine features so that they search for conjunctive ones and when ants join the same colony.

2.1 Ant states and pheromone deposition

An ant can be in four states as shown in Figure 2 and explained next:

- **State S1.** Search and deposit mode – The ant just left a feature which is interesting (i.e. matches its feature detectors) and is now depositing a pheromone trace whilst searching for other objects with matching features;
- **State S2.** Search only mode – The ant just left a feature which does not match its feature detectors and so is searching for other objects but without leaving a trail;
- **State S3.** Path following mode – An ant in state S2 found a pheromone trace and starts to follow it;
- **State S4.** Path following and deposition mode – An ant in state S1 found a pheromone trace and follows it at the same time as continuing to deposit its own pheromone trace, thereby reinforcing the path.

These states enable ants to move around and deposit pheromone traces that indicate potential paths for other
ants to follow. At the same time, the paths evaporate with the effect that only those of interest to a number of ants will be maintained. Pheromone paths stochastically influence further ant movement by inducing other ants to follow these paths that lead towards the feature values in which the ants are interested in.

Figure 2: FSM ant state changes whilst roaming.

Once a feature is found, the ant must determine if it matches the value or range of values (for continuous distributions) of its feature detection algorithm, where the values are those of the object where it began and any additional features acquired through merging. If in the affirmative, the ant recommences or continues to deposit a pheromone on its passage away from the object with the matching feature. If in the negative, the ant ceases to deposit the pheromone until a new interesting feature is found. Objects linked with shorter paths will tend to have stronger pheromone trails due to less evaporation and objects with many linking features will attract higher numbers of ants. Together, the trails will be reinforced as a positive feedback loop emerges.

2.2 Creating roads in multidimensional hyperspace

In a pure model of ant movement, ants would be allowed to travel in any direction in the multidimensional space in a random walk. However, this process is computationally heavy and not scalable with increasing dimensions. In order to compensate for such evident limitations, all the objects or nodes in hyperspace are joined by edges which thus constrain the travel options for ants. In effect, they simply prevent ants spending unnecessary time walking in directions that do not lead to any objects. At the same time, because every object is connected to every other object, it does not prejudice the links an ant may make. All it does is indicate that a path leads to another object but gives no information about that object, its features, or its distance away, other than what may already be laid down in a pheromone trail. Hence the search process remains bottom up, even if the linkage of objects is a top-down instantiation of the domain prior to commencement of the ACC process.

The reason why a constrained movement model is selected over the more realistic ant paradigm is best appreciated after a computational complexity analysis of both. In an unconstrained random-walk model, each ant can go forwards, backwards, or remain at the same value for any dimension which is directly linked with the current point $p$, (where $p$ is the ant’s current location). At any point the ant has an exponential number of choices or directions which it can go to, represented by $3^d - 1$. Whilst the reduced model has a linear selection complexity, as ants select an edge to traverse.

Another reason for constraining paths is that it removes the complication of determining when and whether two ants are meeting in the same place and on the same path. They must follow each other along edges so this will greatly simplify the merge operators for both feature combinations and colonies.

2.3 Parameters for the model

The general philosophy, phenomena, and rationale of the model have been introduced but operationalising it is crucially dependent on the parameters. These control, for example, the speed of evaporation, the point when ants Cergy, France, June 14-15, 2011
should merge features or colonies, the choice of paths dependent on pheromones, and so on. They link to the individual phenomena of the model, which will be described in more detail along with their parameters.

### 2.3.1 Evaporation

The evaporation value is the amount of pheromone that is reduced at each time increment. Rather than using a percentage reduction, which would never reduce to zero and unrealistically slows down as the pheromone scent diminishes, a set step reduction is used. The step reduction is related to the unit amount an ant deposits on each step. If this is taken to be one, then the evaporation parameter is some decremented multiple of it.

### 2.3.2 Detection range for continuous dimensions

Continuous dimensions need to be broken up into categorical elements and this is achieved by taking a range of values above and below the particular one for the object at the ant’s origin. The exact method for doing this is open to experimentation but the idea is that ants respond to a range of values with a categorical response: either the value fits the range or not and a trail is either laid or not accordingly.

### 2.3.3 Ant complement

The ant complement is the number of ants present in the system per data element presented. Runtime analysis indicates that one ant per feature per object might not provide the best pheromone learning environment. The model is able to make more sensitive merging decisions with a higher potential population density but the trade-off is greater in computational load for each step. Experiments still have to determine the optimal balance, which undoubtedly depends on both the density of objects within the hyperspace as well as the dimensionality of that space.

### 2.4 The merging operator and the learning process

Initially, each ant follows a single feature value but eventually ants join up and merge into a multi-feature detection mechanism. This process is governed by the merge operator. Each time an ant sees another ant this is registered in the ‘ant-seen’ list, which is maintained within a specified time frame. The ant also keeps a reference of the particular features being detected for the ants seen. Together, these pieces of information enable the ant to determine whether to merge its feature set and/or merge the colonies. There will be two merge thresholds, one for features and one for the colony (population). With respect to feature merging, each time it occurs, the number of objects matching the ant’s detectors will reduce and this will concomitantly reduce the number of paths laid by those ants. So there is a natural check on the combination process by the fact that specialising the detection reduces the number of matching objects. The result is a set of ants that are looking for a variety of features and feature combinations. If these are located in a particular area of the multidimensional space that corresponds to a cluster, this collection of individual and combined features represents the clustering rule. How that is then interpreted as a classification rule will depend on the colony’s ant population and its constituent feature detectors. What constitutes learning are the paths laid down, the merging of features the ants detect to lay down the paths, the reinforcement of the paths, and the collation of ants into a particular location as a result.

### 3 Experimental Setup and Results

In order to demonstrate the operations of MPACA, a non-linearly separable synthetic data set based on the two dimensional Body Mass Index (BMI) is used. Figure 4 shows the data points plotted with weight on the x-axis and height on the y-axis. Three clusters of points have been created: the top left and bottom right sections represent BMIs that are in the potentially unhealthy range, where the people are either too light or too heavy for their height; the diagonal cluster in the middle shows a “healthy” height/weight balance where the heights and weights increase in proportion.
To demonstrate how MPACA learns the clusters within data sets, the domain will be set up with X ants per feature per object. An object at data point (70, 185) will have 2X ants, half all tuned into values matching the x-dimension and half matching values of the y-dimension. Every point in the multidimensional space is joined to every other point as described earlier, which means an ant can travel from one object to any of the others. When the learning process begins, there will be no pheromones on any of the paths and ants will choose one randomly. However, they will lay down a pheromone as they move and this will begin to inform ants about paths that match their own features.

Figure 3: Graphical representation of the BMI.

Figure 4: Ants start leaving the data point they have been placed on.
Figure 5: Ants roam the environment via movements from one point to the next. During this exploration phase an increased number of ants leave their original ‘nest’ and head towards an unknown unpheromone biased destination. Red traces showing pheromone deposited for height connections and green pheromone showing weight connections.

3.1 Stochastic movement in selecting a new edge
When the ant is about to leave an object, it uses a stochastic mechanism to determine which edge it should take next. This is done by using a Monte Carlo method where each edge is assigned a range of numbers where the range depends on pheromone strength. The ant follows the path associated with a randomly-drawn number covering the range across all paths. The more pheromone exists on a path, the more numbers it has and consequently the higher is the chance of getting selected. The following pseudo-code explains this further:

Let \( m = \) possible edges from the current node minus 1 (can’t choose path just followed)
Let \( N = \) Sum of all pheromone units matching the ant’s feature(s) on all edges;
Assign an order from 1 to \( m \) for the paths;
Provide each edge with a range of numbers, \( x + p \), where \( x = 1 + \) the upper-range number of the previous edge and \( p = \) the number of pheromone units for the path;
Generate a random number, \( R \), between 1 and \( N \);
Choose the edge with the range containing \( R \).
Figure 6: After a series of iterations, ants merge and form clusters. As depicted, three clusters are formed.

The changing levels of pheromones on the paths draw ants towards features and thus objects that are of most interest to them and, after a while, clusters of ants will emerge as shown in Figure 6. As the densities of ants in particular locations increase, the thresholds for both merging features and merging colonies will be passed, which means some ants will now be looking for conjunctive features and ants will start belonging to the same colony. The idea is that equilibrium will result where the general populations of ants are clustered in three different areas of the multidimensional space and have ants belonging to three colonies equating to these spaces, as depicted in Figure 6. The three colonies are shown as different colours, with some ants that can still search in areas outside of their colony.

The clustering rules and, thus, a classification algorithm for assigning new objects to one of the clusters, can be determined by analysing the ant populations within each colony. Table 1 depicts the abnormal BMI colonies have a high proportion of ants with conjunctive feature detectors, either looking for both high weight and low height or low weight and high height, and the value ranges which are applicable at colony level to each dimension.

If it was known that both clusters were actually in the same class, then these feature detection densities in the colonies will result in the required disjunctive rule for assigning a new ant to the class (if either of the feature combinations is present). The “normal” BMI colony has ants with single feature detectors or combined feature detectors where the values are in direct proportion across the population.

<table>
<thead>
<tr>
<th>Ant colony 1: Tall and Thin</th>
<th>Ant Colony 2: Average (linear)</th>
<th>Ant Colony 3: Short and Heavy</th>
</tr>
</thead>
<tbody>
<tr>
<td>((Dimension: 0; LL=185; UL=195)</td>
<td>((Dimension: 0; LL=140; UL=165)</td>
<td>((Dimension: 0; LL=145; UL=165)</td>
</tr>
<tr>
<td>OR</td>
<td>AND</td>
<td>AND</td>
</tr>
<tr>
<td>(Dimension:0; LL=180; UL=190))</td>
<td>(Dimension: 1; LL=55; UL=80))</td>
<td>(Dimension: 1; LL=85; UL=100))</td>
</tr>
<tr>
<td>AND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Dimension: 1 LL=55; UL=65)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>((Dimension:0;LL=170; UL=200)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Dimension: 1; LL=75; UL=100)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: In the above table, LL=Lower Limit, and UL = Upper Limit. Dimension 0=height, Dimension 1 = weight. For the corresponding three colonies formed, a representative ant is selected, defining the values matching an ant’s feature detector.
4 Conclusion

This paper has presented a new multi-pheromone ant-colony clustering algorithm where ants learn to detect individual features or combinations of features that link similar objects together in multidimensional space. The key novel features of the algorithm include ants depositing pheromones equating to individual feature values, the ability of ants to merge pheromones and thus features so that feature combinations can be detected, and having a second merge process for enabling ants to conglomerate within colonies. Relatively standard processes were used for depositing pheromone trails and learning the strengths of connections of trails between similar objects by the use of evaporation and reinforcement. Appropriate parameterisation of the set of phenomena controls the sensitivity of ants for detecting individual clusters; where higher merge thresholds means smaller and potentially more diverse clusters. This is rather like determining whether two peaks are part of the same mountain or can be considered separate mountains; the distance apart and the drop in height are both determinants. Likewise for the ants because a higher merge threshold means ants will not join the same colony between the peaks, only at the peaks, thereby creating two separate colonies.

The paper has explained the rationale for MPACA and introduced the main properties of the model. The basic learning mechanisms were then demonstrated for an artificial data set that was deliberately chosen to display the ability to classify non-linearly separable classes, once the clusters have been detected. Of course, the data were designed to facilitate showing how the algorithm works and there will be many more issues to resolve for real-world data of high dimensionality, not least ensuring MPACA is tractable. However, the results illustrate the potential to learn clusters and to convert them into operational classification rules. The next phase of the research will be to test MPACA on larger, more realistic data sets. This has already started for the Iris data set and will be continued with work on much larger real-world databases in health and logistics domains.

Acknowledgements

Work for this paper was supported by the European Commission through the 7th FP project ADVANCE (http://www.advance-logistics.eu) under grant No. 257398.

5 References


