EFFICIENT CLUSTERING WITH FUZZY ANTS

S. SCHOCKAERT, M. DE COCK, C. CORNELIS AND E. E. KERRE

Fuzziness and Uncertainty Modelling Research Unit,
Department of Applied Mathematics and Computer Science,
Ghent University,
Krijgslaan 281 (S9), 9000 Ghent, Belgium
E-mail: Steven.Schockaert@UGent.be
http://fuzzy.UGent.be

In the past decade, various clustering algorithms based on the behaviour of real ants were proposed. The main advantage of these algorithms lies in the fact that no additional information, such as an initial partitioning of the data or the number of clusters, is needed. In this paper we show how the combination of the ant-based approach with fuzzy rules leads to an algorithm which is conceptually simpler, more efficient and more robust than previous approaches.

1. Introduction

While the behaviour of individual ants is very primitive, the resulting behaviour on the colony-level can be quite complex. A particularly interesting example is the clustering of dead nestmates, as observed with several ant species under laboratory conditions[1]. By exhibiting only simple basic actions and without negotiating about where to gather the corpses, ants manage to cluster all corpses into 1 or 2 piles. The conceptual simplicity of this phenomenon, together with the lack of centralized control and *a priori* information, are the main motivations for designing a clustering algorithm inspired by this behaviour.

Real ants are, because of their very limited brain capacity, often assumed to reason only by means of rules of thumb[2]. Therefore in this paper we propose a clustering approach in which the behaviour of the artificial ants (and more precisely, their stimuli for picking up and dropping items) is governed by fuzzy IF-THEN rules. The resulting algorithm is efficient, robust and easy to use thanks to observed dataset independence of the parameter values involved.

2. Related work

Deneubourg *et al.*[1] proposed an agent-based model to explain how ants manage to cluster the corpses of their dead nestmates. Artificial ants (or agents) are moving randomly on a square grid of cells on which some items are scattered. Each cell can only contain a single item. Whenever an unloaded ant encounters an item, this item is picked up with a probability which depends on an estimation of the density of items of the same type in the neighbourhood. If this density is high, the probability of picking up the item will be low. When a loaded ant encounters a free cell on the grid, the probability that this item is dropped also depends on an estimation of the local density of items of the same type. However, when this density is high, the probability of dropping the load will be high. Simulations show that eventually all objects of the same type are clustered together.

Lumer and Faieta[3] extended the model of Deneubourg *et al.*, using a dissimilarity-based evaluation of the local density, in order to make it suitable for data clustering. Unfortunately, the resulting number of clusters is often too high and convergence is slow. Therefore, a number of modifications were proposed, by Lumer and Faieta themselves as well as by others (*e.g.* [4,5]). Since two different clusters can be adjacent on the grid, heuristics are necessary to determine which items belong to the same cluster.

Monmarché[6] proposed an algorithm in which several items are allowed to be on the same cell. Each cell with a nonzero number of items corresponds to a cluster. Each ant a is endowed with a certain capacity $c(a)$. Instead of carrying one item at a time, an ant a can carry a heap of $c(a)$ items. Probabilities for picking up, at most $c(a)$ items from a heap and for dropping the load on a heap are based on characteristics of the heap, such as the average dissimilarity between items of the heap. When an ant decides to pick up items, the $c(a)$ items whose dissimilarity to the centre of the heap under consideration is highest, are chosen. Two particularly interesting values for the capacity of an ant a are $c(a) = 1$ and $c(a) = \infty$. Monmarché proposes to apply this algorithm twice. The first time, the capacity of all ants is 1, which results in a high number of tight clusters. Subsequently the algorithm is repeated with the clusters of the first pass as atomic objects and ants with infinite capacity. After each pass k-means clustering is applied for handling small classification errors.

In a similar way, in [7] an ant-based clustering algorithm is combined with the fuzzy c-means algorithm. Although some work has been done on combining fuzzy rules with ant-based algorithms for optimization problems[8], to our knowledge until now fuzzy rules have not yet been used to control the behaviour of artificial ants in a clustering algorithm.

3. Fuzzy Ants

Our algorithm is in many ways inspired by the algorithm of Monmarché. We will consider however only one ant, since the use of multiple ants on a non-parallel implementation has no advantages. Instead of introducing several passes, our ant can pick up one item from a heap or an entire heap. Which case applies is governed by a model of division of labour in social insects by Bonabeau *et al.*[9]. In this model, a certain stimulus and a response threshold value are associated with each task a (real) ant can perform. The response threshold value is fixed, but the stimulus can change and represents the need for someone to perform the task. The probability that an ant starts performing a task with stimulus s and response threshold value θ is given by

$$T_n(s;\theta) = \frac{s^n}{s^n + \theta^n}$$

where n is a positive integer.

Let us now apply this model to the problem at hand. A loaded ant can only perform one task: dropping its load. Let s_{drop} be the stimulus associated with this task and θ_{drop} the response threshold value. The probability of dropping the load is then given by

$$P_{drop} = T_{n_i}(s_{drop};\theta_{drop}) \tag{1}$$

where $i \in \{1,2\}$ and n_1, n_2 positive integers. When the ant is only carrying one item n_1 is used, otherwise n_2 is used. An unloaded ant can perform two tasks: picking up one item and picking up all the items. Let s_{one} and s_{all} be the respective stimuli and θ_{one} and θ_{all} the respective response threshold values. The probabilities for picking up one item and picking up all the items are given by

$$P_{pickup\_one} = \frac{s_{one}}{s_{one} + s_{all}} \cdot T_{m_1}(s_{one};\theta_{one}) \tag{2}$$

$$P_{pickup\_all} = \frac{s_{all}}{s_{one} + s_{all}} \cdot T_{m_2}(s_{all};\theta_{all}) \tag{3}$$

where m_1 and m_2 are positive integers.

The values of the stimuli are calculated by evaluating fuzzy if-then rules as explained below. First we introduce some notations. Let E be a fuzzy relation in X, *i.e.* a fuzzy set in X^2, which is reflexive and T_W-transitive (*i.e.* $T_W(E(x,y), E(y,z)) \leq E(x,z)$, for all x, y and z in X) where X is the set of items to be clustered and T_W the Łukasiewicz triangular norm defined by $T_W(x,y) = max(0, x+y-1)$, for all x and y in $[0,1]$. For x and y in X, $E(x,y)$ denotes the degree of similarity between the items x and y. For a heap $H \subseteq X$ with centre c in X, we define $avg(H) = \frac{1}{|H|}\sum_{h \in H} E(h,c)$ and $min(H) = \min_{h \in H} E(h,c)$. Let $E^*(H_1, H_2)$ be the similarity between the centres of the heap H_1 and the heap H_2. Because of the limited space we do not go into detail about how to define and/or compute the centre of a heap, as this can be dependent on the kind of the data that needs to be clustered.

Dropping items The stimulus for a loaded ant to drop its load L on a cell which already contains a heap H is based on the average similarity $A = avg(H)$ and an estimation of the average similarity between the centre of H and items of L. This estimation is calculated as $B = T_W(E^*(L,H), avg(L))$ which is a lower bound due to our assumption about the T_W-transitivity of E and can be implemented much more efficiently than the exact value. If B is smaller than A, the stimulus for dropping the load should be low; if B is greater than A, the stimulus should be high. Since heaps should be able to grow, we should also allow the load to be dropped when A is approximately equal to B. Our ant will perceive the values of A and B to be Very High, High, Medium, Low or Very Low. The stimulus will be perceived as Very Very High, Very High, High, Rather High, Medium, Rather Low, Low, Very Low or Very Very Low. These linguistic terms can be represented by triangular fuzzy sets. The rules for dropping the load L onto an existing heap H are summarized in table 1.

Picking up items An unloaded ant should pick up the most dissimilar item from a heap if the similarity between this item and the centre of the heap is far less than the average similarity of the heap. This means that by taking the item away, the heap will become more homogeneous. An unloaded ant should only pick up an entire heap, if the heap is already homogeneous. Thus, the stimulus for an unloaded ant to pick up a single item from a heap H and the stimulus to pick up all items from that heap are based on the average similarity $A = avg(H)$ and the minimal similarity $M = min(H)$ and can be inferred using fuzzy rules. Because of the limited space, we omit the corresponding rule bases. For evaluating the fuzzy rules, we used a Mamdani inference system with COG as defuzzification method.

Table 1. Stimulus for dropping the load.

	A is V. High	A is High	A is Medium	A is Low	A is V. Low
B is V. High	RH	H	VH	VVH	VVH
B is High	L	RH	H	VH	VVH
B is Medium	VVL	L	RH	H	VH
B is Low	VVL	VVL	L	RH	H
B is V. Low	VVL	VVL	VVL	L	RH

The algorithm During the execution of the algorithm, we maintain a list of all heaps. Initially there is a heap, consisting of a single element, for every item in the dataset. Picking up an entire heap corresponds to removing a heap from the list. At each iteration our ant randomly chooses one heap H from the list and acts as follows.

If the ant is unloaded and if H consists of a single element, the element is picked up with a fixed probability. Depending on the definition of the centre of a heap, comparing the minimal and average similarity of a heap consisting of two elements may not be meaningful. If H consists of two elements a and b, one of them is picked up, with a probability $(1 - E(a,b))^{k_1}$, where k_1 is a small positive integer (*e.g.* 2). Otherwise both elements are picked up, with a fixed probability. If H consists of more than two elements, the stimuli for picking up a single element and for picking up all elements are inferred using the fuzzy rule bases and the corresponding probabilities are given by Eqn. (2)-(3).

If the ant is loaded with a heap L, a new heap containing the load L is added to the list of heaps with a fixed probability. Else, if H consists of a single element a, and L consists of a single element b, L is merged with H with a probability $E(a,b)^{k_2}$, were k_2 is a small positive integer (*e.g.* 2). Else, if H consists of more than one element, the stimulus for dropping the load is calculated and the probability that H and L are merged is given by Eq. (1).

The most important parameters of the algorithm are n_1, n_2, m_1, m_2 in Eqn. (1),(2) and (3). Good results were found within a wide range of values, satisfying $m_1 = m_2 < n_1 < n_2$. Moreover, the values of the parameters seem to be independent of the dataset, but are dependent on the definition of the similarity measure E that is used. All response threshold values were set to the modal value of the fuzzy set representing the linguistic term "medium" for the stimulus.

4. Concluding remarks

We have presented a clustering algorithm, inspired by the behaviour of real ants simulated by means of fuzzy IF-THEN rules. Like all ant-based clustering algorithms, no initial partitioning of the data is needed, nor should the number of clusters be known in advance. Initial experimental results indicate good scalability to large datasets. Outliers in noisy data are left apart and hence do not influence the result, and the parameter values appear to be dataset-independent which makes the algorithm robust.

Acknowledgments

Martine De Cock and Chris Cornelis would like to thank the Fund for Scientific Research – Flanders for funding their research.

References

1. J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, L. Chrétien. The Dynamics of Collective Sorting Robot-Like Ants and Ant-Like Robots. *From Animals to Animats: Proc. of the 1st Int. Conf. on Simulation of Adaptive Behaviour.* 356-363 (1990).
2. B. Hölldobler, E. O. Wilson. *The ants.* Springer-Verslag Heidelberg (1990).
3. E. D. Lumer, B. Faieta. Diversity and Adaptation in Populations of Clustering Ants. *From Animals to Animats 3: Proc. of the 3th Int. Conf. on the Simulation of Adaptive Behaviour.* 501-508 (1994).
4. J. Handl, B. Meyer. Improved Ant-Based Clustering and Sorting in a Document Retrieval Interface. *Proc. of the 7th Int. Conf. on Parallel Problem Solving from Nature.* 913-923 (2002).
5. V. Ramos, F. Muge, P. Pina. Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies. *Soft Computing Systems: Design, Management and Applications.* **87**, 500-509 (2002).
6. N. Monmarché. *Algorithmes de Fourmis Artificielles: Applications à la Classification et à l'Optimisation.* PhD thesis, Université François Rabelais (2000).
7. P. M. Kanade, L. O. Hall. Fuzzy Ants as a Clustering Concept. *Proc. of the 22nd Int. Conf. of the North American Fuzzy Information Processing Society.* 227-232 (2003).
8. P. Lučić. *Modelling Transportation Systems using Concepts of Swarm Intelligence and Soft Computing.* PhD thesis, Virginia Tech (2002).
9. E. Bonabeau, A. Sobkowski, G. Theraulaz, J. L. Deneubourg. Adaptive Task Allocation Inspired by a Model of Division of Labor in Social Insects. *Working Paper* **98-01-004**, available at http://ideas.repec.org/p/wop/safiwp/98-01-004.html (1998).