Classification-Rule Discovery with an Ant Colony Algorithm

Rafael S. Parpinelli
CEFET-PR, Brazil

Heitor S. Lopes
CEFET-PR, Brazil

Alex A. Freitas
University of Kent, UK

INTRODUCTION

Ant colony optimization (ACO) is a relatively new computational intelligence paradigm inspired by the behaviour of natural ants (Bonabeau, Dorigo & Theraulaz, 1999). The natural behaviour of ants that we are interested in is the following. Ants often find the shortest path between a food source and the nest of the colony without using visual information. In order to exchange information about which path should be followed, ants communicate with each other by means of a chemical substance called pheromone. As ants move, a certain amount of pheromone is dropped on the ground, creating a pheromone trail. The more ants follow a given trail, the more attractive that trail becomes to be followed by other ants. This process involves a loop of positive feedback, in which the probability that an ant chooses a path is proportional to the number of ants that have already passed by that path.

Hence, individual ants, following very simple rules, interact to produce an intelligent behaviour – a solution to a complex problem – at the higher level of the ant colony. In other words, intelligence is an emergent phenomenon; that is, "the whole is more than the sum of the parts".

In this article we present an overview of Ant-Miner, an ACO algorithm for discovering classification rules in data mining (Parpinelli, Lopes & Freitas, 2002a, 2002b). In essence, in the classification task each case (record) of the data being mined consists of two parts: a goal attribute, whose value is to be predicted, and a set of predictor attributes. The aim is to predict the value of the goal attribute for a case, given the values of the predictor attributes for that case.

To the best of our knowledge, the use of ACO algorithms (Bonabeau, Dorigo & Theraulaz, 1999; Dorigo et al., 2002) for discovering classification rules is a very under-explored research area. There are other ant algorithms developed for the data mining task of clustering – see for example Monmarché (1999) – but that task is very different from the classification task addressed in this article. Note that Ant-Miner was designed specifically for discovering classification rules, rather than for solving other kinds of data mining tasks.

In other research areas ACO algorithms have been shown to produce effective solutions to difficult real-world problems. A detailed review about many other ACO algorithms (designed to solve many other different kinds of problems) and a discussion about their performance can be found in Bonabeau, Dorigo and Theraulaz (1999) and Dorigo et al. (2002).

A typical example of application of ACO is network traffic routing, where artificial ants deposit "virtual pheromone" (information) at the network nodes. In essence, the amount of pheromone deposited at each node is inversely proportional to the congestion of traffic in that node. This reinforces paths through uncongested areas. Both British Telecom and France Telecom have explored this application of ACO in telephone networks.

ANT COLONY OPTIMIZATION

An ACO algorithm is essentially a system based on agents that simulate the natural behavior of ants, including mechanisms of cooperation and adaptation.

ACO algorithms are based on the following ideas:

- Each path followed by an ant is associated with a candidate solution for a given problem;
- When an ant follows a path, the amount of pheromone deposited on that path is proportional to the quality of the corresponding candidate solution for the target problem;

- When an ant has to choose between two or more paths, the path(s) with a larger amount of pheromone have a greater probability of being chosen by the ant.

As a result, the ants eventually converge to a short path, hopefully the optimum or a near-optimum solution for the target problem.

In essence, the design of an ACO algorithm involves the specification of (Bonabeau, Dorigo & Theraulaz, 1999):

- An appropriate representation of the problem, which allows the ants to incrementally construct/modify solutions through the use of a probabilistic transition rule, based on the amount of pheromone in the trail and on a local, problem-dependent heuristic;
- A method to enforce the construction of valid solutions;
- A problem-dependent heuristic function (h) that measures the quality of items that can be added to the current partial solution;
- A rule for pheromone updating, which specifies how to modify the pheromone trail (t);
- A probabilistic transition rule based on the value of the heuristic function (h) and on the contents of the pheromone trail (t) that is used to iteratively construct a solution.

Artificial ants have several characteristics similar to real ants, namely:

- Artificial ants have a probabilistic preference for paths with a larger amount of pheromone;
- Shorter paths tend to have larger rates of growth in their amount of pheromone;
- The ants use an indirect communication system based on the amount of pheromone deposited on each path.

MOTIVATIONS FOR USING ACO

ACO possesses a number of features that are important to computational problem solving (Freitas & Johnson, 2003):

- The algorithms are relatively simple to understand and implement, whilst also offering emergent complexity to deal effectively with challenging problems;
- They can be readily hybridized with other techniques and/or problem-dependent heuristics in a synergistic fashion;
- They are compatible with the current trend towards greater decentralization in computing;
- The algorithms are highly adaptive and robust, enabling them to cope well with noisy data.

Two more features of ACO are particularly useful in data mining applications:

- Many projects in the field of data mining were developed using deterministic decision trees or rule induction algorithms. These algorithms are hill climbing like and are susceptible to finding only locally optimal solutions instead of the global optimum. The utilization of ACO to induce classification rules tries to mitigate this problem of premature convergence to local optima, since ACO algorithms have a stochastic component that favors a global search in the problem's search space;
- Unlike classical methods for rule induction, the ACO heuristic is a population-based one. This characteristic has advantages over other methods because it allows the system to search many different points in the search space concurrently and to use the positive feedback between the ants as a search mechanism.

REPRESENTING A CANDIDATE CLASSIFICATION RULE

In Ant-Miner each artificial ant represents a candidate classification rule of the form:

- IF <*term1* AND *term2* AND … > THEN < *class* >.

Each term is a triple <*attribute, operator, value*>, where *value* is one of the values belonging to the domain of *attribute*. An example of a term is: <*Sex = female*>. *Class* is the value of the goal attribute predicted by the rule for any case that satisfies all the terms of the rule antecedent. An example of a rule is:

- IF <*Salary = high*> AND <*Mortgage = No*> THEN <*Credit = good*>.

In the current version of Ant-Miner the *operator* is always "=", so that Ant-Miner can cope only with categorical (discrete) attributes. Continuous attributes would have to be discretized in a preprocessing step.

DESCRIPTION OF ANT-MINER

The pseudocode of Ant-Miner is described, at a very high level of abstraction, in Algorithm 1. Ant-Miner starts by