Applied Soft Computing 30 (2015) 166–178

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Enhancing the effectiveness of Ant Colony Decision Tree algorithms by co-learning

Urszula Boryczka*, Jan Kozak**

Institute of Computer Science, University of Silesia, Będzińska 39, 41-200 Sosnowiec, Poland

ARTICLE INFO

Article history:
Received 28 July 2013
Received in revised form
11 September 2014
Accepted 24 December 2014
Available online 7 February 2015

Keywords:
Ant Colony Optimization
Ant Colony Decision Trees
Decision trees
Pareto front
Quality of decision trees
Ant-Miner

ABSTRACT

Data mining and visualization techniques for high-dimensional data provide helpful information to substantially augment decision-making. Optimization techniques provide a way to efficiently search for these solutions. ACO applied to data mining tasks – a decision tree construction – is one of these methods and the focus of this paper. The Ant Colony Decision Tree (ACDT) approach generates solutions efficiently and effectively but scales poorly to large problems. This article merges the methods that have been developed for better construction of decision trees by ants. The ACDT approach is tested in the context of the bi-criteria evaluation function by focusing on two problems: the size of the decision trees and the accuracy of classification obtained during ACDT performance. This approach is tested in co-learning mechanism, it means agents–ants can interact during the construction decision trees via pheromone values. This cooperation is a chance of getting better results. The proposed methodology of analysis of ACDT is tested in a number of well-known benchmark data sets from the UCI Machine Learning Repository. The empirical results clearly show that the ACDT algorithm creates good solutions which are located in the Pareto front. The software that implements the ACDT algorithm used to generate the results of this study can be downloaded freely from http://www.acdtalgorithm.com.

1. Introduction

Our goal is to propose new approach for constructing more effective decision trees, concerning classification accuracy and decision tree growth. This solution is possible due to application of Ant Colony Optimization. Using Ant Colony algorithms in decision tree construction allows to construct variety of alternative decision trees presenting different local optima. In case of deterministic algorithms, each decision tree is constructed in the same way. Due to the pheromone updating rules, representing reinforcement learning schema, agent–ants construct better quality decision trees.

This collaborative view of learning can occur without interaction between the learning agents, known as ensemble learning, or with interaction during the learning stage, known as co-learning [1]. This mechanism is a good way to obtain better treatment results, and should always be taken into consideration in comprehensive point of view.

Results obtained during experimental study motivated us to propose new optimization criteria for decision tress evaluation. It is important in case of bi-criterion decision tree evaluation, where as heuristic function, as well as quality function we create via decision tree growth and classification accuracy. In this article we also discuss the decision forest as a more effective classification ensemble.

The additional aim of this article is to arrange information about ACDT algorithm as well as its performance the Ant Colony Decision Tree approach is firstly proposed by Boryczka and Kozak [2]. In the beginning this algorithms created binary decision trees for discrete attributes occurred in data sets. The comparative study with classical approach has shown that ACDT offered better results in context of accuracy of the classification. In the following papers Boryczka and Kozak [3] adjusted this approach to continuous values of attributes. They proposed the inequality test.

Boryczka et al. [4] have also undertaken construction the heterarchical algorithm of the ACDT approach with parallel implementation. Authors also tested the performance of the ACDT in practical, real-life data sets: H-Bond Data Set [5]. Otero et al. [6] proposed a modification of the ACDT approach by using a new splitting criterion derived from C4.5 approach in contrary to our proposition originated from CART algorithm.

This article is organized as follows: Section 1 comprises an introduction to the subject matter of this article. In Sections 2 and 3, the Swarm Intelligence and Ant Colony Optimization is introduced. Section 4 reviews Ant Colony Optimization in Data Mining. Decision Trees and Ant Colony Decision Trees are presented in Sections 5 and 6. Section 7 focuses on the evaluation criteria of the constructed decision tree. Section 8 presents the experimental study that was conducted to evaluate the performance of the ACDT algorithm. Finally, we conclude with general remarks on this work, and some directions for future research are pointed out.

2. Swarm Intelligence

There has recently been a number of research studies regarding the application of Swarm Intelligence to various data-mining problems. Swarm Intelligence describes the ability of groups of decentralized and self-organizing agents to exhibit highly

* Corresponding author. Tel.: +48 32 3689866; fax: +48 32 3689866.
** Corresponding author.
E-mail addresses: urszula.boryczka@us.edu.pl (U. Boryczka), jan.kozak@us.edu.pl (J. Kozak).

http://dx.doi.org/10.1016/j.asoc.2014.12.036

organized behaviours. These global behaviours of swarms often allow the swarm as a whole to accomplish tasks which are beyond the capabilities of any one of the constituent individuals. Following the appearance of the most recent books, ("Swarm Intelligence" [7] and "Swarm Intelligence: From natural to artificial systems" [8]), the area of Swarm Intelligence became a much discussed topic in the fields of computer science, collective intelligence, and robotics. Today, the number of successful applications of Swarm Intelligence continues to grow. This natural phenomenon is an inspiration for Swarm Intelligence systems – a class of algorithms that utilizes the emergent patterns of swarms to solve hard computational problems.

Swarm Intelligence is interesting because it provides a basis with which it is possible to explore collective (or distributed) problem solving without centralized control or the provision of a global model. The term "Swarm Intelligence" was coined in 1989 by Gerardo Beni and Jing Wang in the context of cellular robotic systems [9]. Beni and Wang proposed that groups of simple virtual agents could be programmed to collaboratively solve difficult tasks and described how collectively intelligent behaviours can be exhibited in systems of non-intelligent agents. An artificial system organized in this way would present three distinct advantages:

1. Agents are simple and they will work together to solve problems.
2. The overall system is very reliable due to high levels of redundancy.
3. Problems are solved in a distributed way since each agent handles a portion of the problem.

These three features are present in nearly all Swarm Intelligence systems and are what makes Swarm Intelligence approaches to problem-solving attractive for certain problems. In addition to these general characteristics, nearly all Swarm Intelligence systems exhibit the following features:

Homogeneity – every member of the swarm follows the same rules and decision making processes.
Locality – the actions and decisions of individuals are made on the basis of purely local information and from what agents learn *via* (direct or indirect) communication with others.
Randomness – swarm members introduce randomness into their decision-making processes in order to explore new solutions.
Positive feedback – as with Darwinian evolution, "good" solutions that emerge from the actions of swarm individuals are identified as having good "fitness" and are reinforced over time.

To sum up, Swarm Intelligence [7,8] is the property of a system whereby the collective behaviours of unsophisticated agents that interact locally with their environment cause coherent functional global patterns to emerge. The characteristics of a swarm are distributed, with no central control or data source, no (explicit) model of the environment, perception of the environment, limited time to act and strong emphasis on reaction and adaptation. Perhaps the most well-explored Swarm Intelligence algorithms are the Ant Colony Optimization (ACO) methods that are applied to different optimization problems.

3. Ant Colony Optimization

In Ant Colony Optimization (ACO) a stochastic optimization strategy inspired by the collaborative path-finding behaviours that are exhibited by colonies of real ants is observed. In nature, ants are simple organisms that each possess very limited capabilities and, individually, are only able to accomplish the most simple of tasks. Amazingly, colonies of ants are able to collectively solve difficult problems that are far beyond the abilities of any single member of the group (manifested by emergency in global behaviour). Real ant colonies are a type of distributed and self-organizing system where the complex "global" behaviours exhibited by the colony as a whole are coordinated by indirect communication between the ants. The term "stigmergy" should be incorporated here to explain these interactions [10]. Specifically, ants communicate with one another by depositing pheromone (a chemical substance produced by each ant) on the ground as they move about. As ants make their random explorations of the environment, they are more likely to follow these pheromone trails. The pheromone on a given trail will intensify as more ants follow it, and decrease in intensity over time by the process of evaporation when ants fail to travel on it. The process of pheromone trail laying/following in real ant colonies is mimicked by virtual ant–agents in ACO systems.

Generally, in ACO a population of independent agent–ants moves through an environment (most commonly a graph) that represents the solution space of some target problem. This graph representation of the solution space is referred to as a "construction graph". By generating a path through the construction graph for a target problem, each ant generates a candidate solution to that problem. By repeatedly generating solutions and reinforcing paths which represent good solutions with pheromone, the optimal solution (or path) will eventually be found. The movement of ants in the construction graph is dictated by a stochastic transition rule based on two pieces of local information: pheromone level and heuristic value. Edges in the construction graph have a level of pheromone associated with them, and each node has a certain heuristic value. The amount of pheromone on an edge is a measure of how many ants have recently traversed the edge and as attempts to attract ants to edges which have been identified as components of good solutions; thus, it can be treated as a learning mechanism. The heuristic value of a node is derived from *a priori* knowledge about the target problem and attempts to capture the relative importance of a node in candidate solutions.

An agent–ant located on some vertex in the construction graph will select an edge to traverse with a probability given by a transition rule that takes into account the pheromone level on each connected edge and the heuristic value of the vertices to which the edges connect. The evaporation of a pheromone value is often compared to the learning factor in the Reinforcement Learning algorithm. The decision will favour an edge with a relatively high pheromone level and one that connects to a vertex with a relatively high heuristic value. It should be noted that the transition rule only gives the probability of selecting a given "next move". This probability is used to weight a random selection process where the next move is actually selected. This causes random decisions to be introduced into the system.

Algorithms inspired by ant behaviour were first proposed by Dorigo et al. [11] and Dorigo [12] as a multi-agent approach to difficult combinatorial optimization problems [13], such as that of the Travelling Salesman Problem (TSP), the Quadratic Assignment Problem (QAP) and Scheduling. There is currently much ongoing activity in the scientific community to extend/apply ant-based algorithms to many different discrete optimization problems as described by Dorigo et al. [13,14]. Recent applications cover problems like Vehicle Routing, Sequential Ordering, Graph Coloring, Routing in Communications Networks, and so on [15].

An essential step made in this direction was the development of the Ant System by Dorigo and Stützle [15], a new type of heuristic inspired by analogies to the foraging behaviour of real ant colonies which has proven to work successfully in a series of experimental studies. Diverse modifications of AS have been applied to many different types of discrete optimization problems and have produced very satisfactory results [14]. Recently, the approach has