2012 2nd International Conference on Management and Artificial Intelligence
IPEDR Vol.35 (2012) © (2012) IACSIT Press, Singapore

Diversification with an Ant Colony System for the Feature Selection Problem

Nadia Abd-Alsabour [1*] and Atef Moneim [1]

[1] Cairo University, Egypt

Abstract. Every Metaheuristic should be designed with the aim of effectively and efficiently exploring the search space. The search process should be intelligent in order to both intensively explore areas of the search space with high quality solutions and to move to unexplored areas of the search space. This paper tries different local pheromone update rule with an ant colony system that leads to good results when solving the feature selection problem in four datasets out of used five datasets.

Keywords: Ant colony optimization, feature selection, diversification.

1. Introduction

The behaviour of ants has inspired the development of artificial distributed problem solving systems. Each ant has its own agenda and follows very simple rules; more complex global-level patterns emerge solely through the ants' interactions with each other and their environment (without supervision or central control). There are two types of interactions between ants in a colony: *direct interactions* (involves tactile, visual, or chemical contact.) and *indirect interactions* (are initiated by individuals that exhibit some behavior that modifies the environment which in turn stimulates a change in the behavior of other individuals). Indirect interactions can solve difficult problems although they may be simple [1].

Ants are able to find the shortest path between their nest and food sources because of the chemical substance, called pheromone that they deposit on their way. The pheromone evaporates over time so the shortest paths will contain more pheromone and will subsequently attract a greater number of ants.

ACO algorithms simulate the foraging behavior of some ant species [2]. ACO algorithms use two factors for guiding the search process. These are: 1) *the pheromone values* (numerical values as a simulation for the pheromone that real ants deposit on their way to and from their nest), and 2) *heuristic information*. There are two types of heuristic information used by ACO algorithms; static heuristic information (that is computed at the time of initialization time and then remain unchanged throughout the whole algorithm's run, such as the distances between cities in the traveling salesman problem (TSP)) and dynamic heuristic information (that depends on the partial solution constructed so far and therefore it is computed at each step of an ant's walk) [3].

Many datasets contain too many features [4]. Many of them are not likely to be necessary for accurate classification and including them may lead to a worse model than if they were removed. For example, in building a system to discriminate between images of male and female faces, the colours of a person's eyes, hair, or skin are hardly likely to be useful in this discriminative context. These are features that are easy to measure and are general characteristics of a person's appearance but they carry little information in this particular task [5]. Moreover, some features may be redundant i.e., two or more features contain essentially the same predictive information. An example is that the income feature before tax and the income feature

[*] Corresponding author. Tel.: + 20201120174374
E-mail: nadia_issr@hotmail.com, nadia.abdalsabour@ieee.org

after tax are likely to be highly correlated. Besides, in many applications it is not obvious which features are relevant (having an impact on classifier accuracy) and which ones are irrelevant (will never contribute to the classifier accuracy, and can thus be removed). For example, data that records the day of the week on which a bank loan application was completed is unlikely to be relevant to the success of the application. [5]-[7]. Such features may slow down and mislead the learning step and do not contribute to the classification process. Hence, feature selection (FS) from the original set of features is highly desirable in many situations in order to reduce the number of features, improve the classification accuracy, and simplify the learned representation [8].

Feature selection is defined as a process of finding a subset of features from the original set of features according to the criterion (specifies the details of measuring feature subsets) of feature selection [6], [8]. In general, feature selection is a search problem according to some evaluation criterion [8] and it has three main components:

- *Generating subsets of features:* One way is to generate subsets of features sequentially. If we start with an empty subset and gradually add one feature at a time, it is called sequential forward selection; if we start with a full set and remove one feature at a time, it is called sequential backward selection. Alternatively, a random subset of features could be generated [8].
- *Evaluation criteria:* We need some measure to decide which feature subset should be kept. Each subset needs to be evaluated by a certain evaluation criterion and compared with the previous best one with respect to this criterion. If it is found to be better, it replaces the previous best subset [9]. Evaluation criteria can be categorized broadly into two groups – *wrappe*r and *filter* approaches – based on their dependence on the learning algorithm applied on the selected feature subset. The first approach is the filter approach (also called open-loop, reset bias, front end methods, or independent criterion). These methods do not consider the effect of the selected features on the classifier's performance [6], [9] i.e., they do not rely on a predictor to determine whether a subset is good or not (but the wrapper does) [8]. Ignoring the effect of the selected feature subset on the performance of the classifier (lack of feedback on the classifier's performance) is a weak side of the filter methods. The second are the wrapper approaches (also called performance bias, closed loop, classifier feedback methods, or dependent criterion) [6], [9]. They are based on using the classifier as a "black box" and to have an external loop (or "wrapper") that systematically adds and subtracts features to the current subset and evaluates each subset on the basis of how well the classifier performs applied to this feature subsets (and thus forming processing feedback) [5]-[6]. The wrapper approaches will generally provide a better selection of feature subset since they are based on the ultimate goal (and the criterion) of optimal feature selection, which provides the best prediction [6].
- *Stopping criteria:* We need to force the loop of generating and evaluating feature subsets to terminate. For exhaustive or sequential feature subset generation, the loop will naturally stop when a full feature subset becomes empty or an empty subset becomes full [8].

Finding the optimal feature selection is an NP-hard optimization problem that involves searching the space of possible feature subsets to identify the optimal one. There are 2^n states in the search space (n is the number of features in the dataset). For large n values, evaluating all the states is computationally infeasible. Therefore, many heuristics such as genetic algorithms (GAs) [10] tabu search (TS), simulated annealing (SA) and ant colony optimization algorithms (ACO) have been used for solving feature selection.

The rest of this paper is organized as follows. Section 2 introduces intensification and diversification. Section 3 details the experiments carried out and presents the obtained results. Section 4 concludes this paper and Section 5 highlights future work in this area.

2. Intensification and Diversification

Every Metaheuristic should be designed with the aim of effectively and efficiently exploring the search space. The search process should be intelligent in order to both intensively explore areas of the search space with high quality solutions and to move to unexplored areas of the search space. This is called intensification and diversification. These terms stem from tabu search where intensification strategies are based on modifying choice rules in order to encourage move combination and solution features historically found good. They may also initiate a return to attractive regions to search them more thoroughly. The diversification stage encourages the search process to examine unvisited regions and to generate solutions that differ in various significant ways from those seen before. It should be noticed that in tabu search, intensification and diversification can not be characterized as opposing forces [11].

A metaheuristic will be successful on a given optimization problem if it can provide a balance between the exploitation of the accumulated search experience and the exploration of the search space to identify regions with high quality solutions in a problem specific near optimal way [11].

Metaheuristics are intelligent strategies for exploring a search space. Crucial for the success of a metaheuristic is a well-adjusted balance between diversification and intensification. The balance between diversification and intensification is important because of the following two reasons:

- To quickly identify areas of the search space with high quality solutions, and
- To avoid spending too much time in areas of the search space that are already well explored or that only consist of poor-quality solutions [12].

Intensification and diversification of the search process are quite unexplored topics in ant colony optimization research. There are just a few papers explicitly dealing with this topic. The mechanisms that are already existed can be divided into the following two categories:

- Mechanisms that change the pheromone values, and
- Algorithms that apply multiple colonies and exchange information between them [13].

Intensification is the exploitation of the information gathered by the system at a given time. Diversification is the exploration of search space areas imperfectly taken into account. There are several ways in which these two facets can be achieved:

- The most obvious method is by adjusting the parameters α and β which determine the relative influence of the trails of the pheromone and the heuristic information. The higher value of α, the more significant will be the intensification. This is because the pheromone will have more influence on the choice of the ants. Conversely, the lower the value of α, the stronger diversification will take place. This is because the ants will avoid the trails. Similarly the parameter β acts in a similar manner.
- A viable alternative is the management of the trails of the pheromone. For example, allowing the best solutions to contribute more to the trails this supports intensification [14].

The aim of adding explicit intensification / diversification to ant colony optimization is to better sample the search space [15]. The use of intensification and diversification can lead to achieving good quality solutions. Intensification aims to identify solution components that are common to good solutions and to encourage the search process to seek solutions with those components. Diversification is complementary to this as it allows the search process to enter unexplored regions of the search space [16].

Intensification and diversification techniques are present to an extent in ant colony system algorithm. Diversification is achieved by the local pheromone update rule as pheromone evaporates from the visited edges thus encouraging the system to incorporate less used edges in subsequent iterations [16].

In general there are two types of diversification in ant colony optimization in finding tours and in depositing pheromone [17].

3. Computational Experiments

In order to investigate the effect of using different local pheromone update (diversification) on the performance of ACO for feature selection, we perform experiments on an ACO algorithm that was developed in [18]. This algorithm uses a SVM learning algorithm. Since the primary goal of classification is to maximize the predicative accuracy, classifier accuracy is generally accepted and widely used as the primary measure by researchers and practitioners [8]. Classifier accuracy indicates how accurately a given classifier will label future data on which the classifier has not been trained. One of the ways used to estimate classifier accuracy is *k*-fold cross validation in which the initial data are randomly partitioned into *k* mutually exclusive folds s1, s2, …, sk each of which is of approximately equal size. Training and testing is performed *k* times. The classifier of the first iteration is trained on folds s2, s3, …., sk and the first fold is kept for testing. In iteration *i*, the fold si is the test fold and the remaining folds are collectively used to train the classifier [7]. In this case, the classifier accuracy is computed by considering the average classification accuracy of the k-fold cross-validation experiments as shown in Equation 1:

$$\text{classifier accuracy} = a / b \qquad (1)$$

where a is the overall number of correct classifications from k iterations and b is the total number of samples in the initial data.

The classifier accuracy is used as a fitness function in the algorithm used here where each ant evaluates its solution based on its ratio of correct classifications.

3.1. Datasets

Five experiments using five datasets [19] were carried out. The details of the used datasets are shown in Table 1.

Table1 The details of the used datasets

Dataset Name	No. of features	No. of classes	Class label type	No. of instances
Statistical datasets				
Backache	32	2	Numeric	180
Prnn_virus3	17	4	Numeric	38
Prnn_viruses	17	6	Numeric	61
analcatdata_authorship	70	4	Numeric	841
analcatdata_marketing	32	5	Nominal	364

3.2. Method

Two types of experiments are performed in this paper (on a previously developed ACO algorithm [18]). These are:

- Without diversification [18], and
- With diversification.

In these experiments, 5-fold cross validation was used. The number of ants was set to the number of the features in the given dataset. The initial pheromone was set to 1 so that it does not have effect on the probability at the beginning. The number of iterations is 10 iterations. ρ was set to 0.4. β was set to 0.2. φ was set to 0.2 and α to 1. After many experiments, these values were used since they give the best performance of the used algorithm.

3.3. Results

Table 2 shows the results of the used algorithm with diversification using the above mentioned datasets. The results for the algorithm represent the average of ten independent runs. All the experiments were run on a PC with a 2 GHz CPU and 3 GB RAM.

Table 2: The accuracy of SVM with and without diversification

Dataset Name	**ACS-SVM (with diversification)**		**ACS-SVM (without diversification)**	
	Avg. no. of selected features	Accuracy	Avg. no. of selected features	Accuracy
Statistical datasets				
Backache	12.5	0.9283	18.8	0.9222
Prnn_virus3	8.7	0.9868	9.6	0.9842
Prnn_viruses	9.3	0.9148	10.2	0.9082
Analcatdata_authorship	35.4	1	36.8	1
Analcatdata_marketing	24.7	0.6602	24.6	0.6692

The previous results showed that using diversification worked well in the first four datasets i.e., the classifier accuracy (the third column) is better than without diversification (the last column) and it gives a smaller number of the selected features (the second column).

4. Conclusions

This paper uses different local pheromone update rule than the traditional one in the ant colony system and that leads to better results in four datasets out of used five datasets.

5. Future Work

As a direction for future work in this area, more datasets are required in order to show the effect of the change of the local pheromone update rule on the performance of feature selection algorithms.

6. References

[1] T. Mirzayans, N. Parimi, P. Pilarski, C. Backhouse, L. Wyard-Scott, and P. Musilek: A Swarm-Based System for Object Recognition, neural network world, 2005, **15**: 243-255.

[2] M. Dorigo, E. Bonabeou, and G. Theraulaz: Inspiration for optimization from social insect behavior, Nature, 2000, **406:** 39-42.

[3] M. Dorigo, and T. Stutzle: Ant Colony Optimization, MIT Press, 2004.

[4] M. Kantardzic: Data mining: Concepts, Models, Methods, and Algorithms, IEEE press, 2003.

[5] D. Hand, H. Mannila, and P. Smyth: Principles of Data Mining, Massachusetts Institute of Technology, 2001.

[6] K. J. Cios, W. Pedrycz, and R. Wiswiniarski: Data mining methods for knowledge discovery, Kluwer Academic Publishers, 1998.

[7] J. Han, and M. Kamber: Data mining concepts and techniques, Academic Press, 2001.

[8] W., Klosgen and J. M., Zytkow: Handbook of data mining and knowledge discovery, Oxford University Press, 2002.

[9] N. Ye: The Handbook of Data Mining, Lawrence Erlbaum Associates, Inc, 2003.

[10] J. Yang, and V. Honavar: Feature subset selection using a genetic algorithm, IEEE Intelligent Systems, 1998, **13** **(**2): 44–49.

[11] C. Blum, and A. Roli: Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, ACM Computing Survey, 2003, **35** (3): 268-308.

[12] P. Méndez: Development of a Hybrid Metaheuristic For The Efficient Solution of Strategic Supply Chain Management Problems: Application to The Energy Sector, M.Sc Thesis, 2011.

[13] C. Blum: ACO Applied to Group Shop Scheduling: A Case Study on Intensification and Diversification in Dorigo, M., et al. (Eds.) Ants 2002 Lecture Notes in Computer Science, **2463:** 14-27, Springer, 2002.

[14] J. Dr´eo, A. P´etrowski, P. Siarry and E. Taillard: Metaheuristics for Hard Optimization, Springer, 2006.

[15] M. Randall: Search Space Reduction as a Tool for Achieving Intensification and Diversification in Ant Colony Optimization, IEA/AIE 2006, pp. 254-261, 2006.

[16] M. Randall, and A. Tonkes: Intensification and diversification strategies in ant colony search, Technical report TR00-02, School of information technology, Bond University, Australia.

[17] Y. Nakamichi, and T. Arita: Diversity control in ant colony optimization, Artificial Life Robotics, 2004, **7:** 198-204.

[18] N. Abd-Alsabour and M. Randall: Feature Selection for Classification Using an Ant Colony System, Proceedings of IEEE International Conference on e-Science and Grid Computing Workshops, 2010: 86-91.

[19] http://www.cs.waikato.ac.nz/ml, viewed 20th January 2012.