Applied Soft Computing 35 (2015) 513–524

Contents lists available at ScienceDirect

Applied Soft Computing

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

Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems

Mohammed Alweshah[a], Salwani Abdullah[b,*]

[a] Prince Abdullah Bin Ghazi Faculty of Information Technology, Al-Balqa Applied University, Salt, Jordan
[b] Data Mining and Optimization Research Group (DMO), Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 5 March 2013
Received in revised form 15 June 2015
Accepted 16 June 2015
Available online 25 June 2015

Keywords:
Firefly algorithm
Lévy flight
Simulated annealing
Probabilistic neural networks
Classification problems

ABSTRACT

Classification is one of the important tasks in data mining. The probabilistic neural network (PNN) is a well-known and efficient approach for classification. The objective of the work presented in this paper is to build on this approach to develop an effective method for classification problems that can find high-quality solutions (with respect to classification accuracy) at a high convergence speed. To achieve this objective, we propose a method that hybridizes the firefly algorithm with simulated annealing (denoted as SFA), where simulated annealing is applied to control the randomness step inside the firefly algorithm while optimizing the weights of the standard PNN model. We also extend our work by investigating the effectiveness of using Lévy flight within the firefly algorithm (denoted as LFA) to better explore the search space and by integrating SFA with Lévy flight (denoted as LSFA) in order to improve the performance of the PNN. The algorithms were tested on 11 standard benchmark datasets. Experimental results indicate that the LSFA shows better performance than the SFA and LFA. Moreover, when compared with other algorithms in the literature, the LSFA is able to obtain better results in terms of classification accuracy.

1. Introduction

Classification is one of the key data mining tasks. Classification maps data into predefined groups or families. It is a form of supervised learning because the classes are learned before the data is examined. The goal of a classification method is to create a model that correctly maps the input to the output by using historical data, so that the model can be used to develop output when the desired output is unknown.

Several techniques have been successfully used for classification problems, including the neural network (NN) [1], support vector machine (SVM) [2], naive Bayes (NB) [3], radial basis function (RBF) [4], logistic regression (LR) [5], K-nearest neighbours (KNN), and the iterative dichotomiser 3 (ID3) [6].

The Neural Network is one of the most well-known and widely used techniques for classification. The NN model was first proposed by Rosenblatt in the late 1950s [7]. Since that time, a lot of NN models have been developed, including feed-forward networks, RBF networks, the multi-layer perceptron, modular networks, and the probabilistic neural network (PNN). These models differ from each other in terms of architecture, behaviour and learning approaches, hence they are suitable for solving different problems such as series forecasting [5], stock market prediction [8], weather prediction [9], and pattern recognition [10].

The PNN is one of the appropriate approaches for solving classification problems. It is a general NN model that is based on the notion of 'the gradient steepest descent method', which enables a reduction of errors between the actual and predicted output functions by permitting the network to correct the network weights [1,6,11–14]. Recently, the hybridization of metaheuristics with different kinds of classifiers has been investigated and the developed models, some of which are described below, show better performance than the above-mentioned standard classification approaches.

Single-based and population-based metaheuristics can be used to train a NN. Single-solution-based approaches include the tabu search [15] and the simulated annealing (SA) approach [14], the latter of which is based on a Monte Carlo model that was applied by Metropolis et al. to replicate energy levels in cooling solids. The population-based approach in combination with a NN has attracted great interest because NNs combined with evolutionary algorithms (EAs) result in better intelligent systems than when relying on NNs or EAs alone [16,17]. Among these population-based approaches, a particle swarm optimization (PSO) algorithm in isolation and

* Corresponding author. Tel.: +60 389216667.
E-mail addresses: weshah@bau.edu.jo (M. Alweshah), salwani@ukm.edu.my (S. Abdullah).

http://dx.doi.org/10.1016/j.asoc.2015.06.018
1568-4946/© 2015 Published by Elsevier B.V.

a hybridization PSO algorithm with a local search operator have been employed to train a NN [18,19]. Other swarm intelligence methods such as ant colony optimization (ACO) have also been employed [20,21]. In addition, Chen [22] proposed a novel hybrid algorithm based on the artificial fish swarm algorithm. The genetic algorithm (GA) [23], differential evaluation [24], improved bacterial chemotaxis optimization (IBCO) [25], electromagnetism-like mechanism-based algorithm (EMA) [26], and harmony search algorithm (HSA) [27–30] are some of the other important methods that have been proposed in recent years. The efficiency of metaheuristic algorithms can be attributed to their ability to imitate the best features in nature and the 'selection of the fittest' biological systems. The two most important characteristics of metaheuristic algorithms are diversification and intensification [31]. The aim of intensification, which is also called exploitation, is to search locally and more intensively, while diversification, which is also called exploration, aims to ensure that the algorithm globally explores the search space. The two terms might appear to be contradictory, but their balanced combination is crucial to the success of any metaheuristic algorithm [7,31–33].

For firefly algorithm (FA) optimization, the diversification component is represented by the random movement component, while the intensification component is implicitly controlled by the attraction of different fireflies and the attractiveness strength. Unlike the other metaheuristics, the interactions between exploration and exploitation in the FA are intermingled in some ways, which might be an important factor in its successful solving of classification problems.

In this work, we investigate combining the FA with a SA algorithm and hybridizing the FA with Lévy flight in order to attempt to improve the performance of the PNN by creating an effective balance between exploration and exploitation during the optimization process, which we attempt to achieve by controlling the randomness steps and exploring the search space efficiently in order to find the optimal weights of the PNN classification technique. Such hybridization requires a fine balance between diversification and intensification to ensure faster and more efficient convergence and to ensure the quality of the solutions in order to find the optimal weight of the PNN classification technique. Therefore we start from the first iteration to calculate the accuracy by modifying the PNN using the FA and SA to improve the quality of the best solution. To our knowledge, this is the first attempt to hybridize the FA with SA for classification problems.

The rest of the paper is organized as follows: Section 2 presents the background and literature on FAs and Section 3 describes the proposed method. Section 4 presents a discussion of the experimental results and Section 5 provides details of the computational complexity of the proposed method. Section 6 concludes the work presented in this paper.

2. Firefly algorithm: background and literature

The FA was initially developed by Yang [34] as a population-based technique for solving optimization problems. It was motivated by the short and rhythmic flashing light produced by fireflies. Theses flashing lights enable fireflies to attract each other and assist them to find a mate, attack their prey, and also protect themselves by creating a sense of fear in the minds of predators [35]. The less bright fireflies are easily attracted by the brighter fireflies and the brightness of the light of a firefly is affected by the landscape [7,36]. This process can be formulated as an optimization algorithm because the flashing lights (solutions) can be formulated to match with the fitness function to be optimized. The FA follows three rules: (1) fireflies must be unisex, (2) the less bright firefly is attracted to the randomly moving brighter fireflies, and (3) the brightness of every firefly symbolizes the quality of the solution.

Łukasik and Żak employed a FA for continuous constrained optimization tasks and it was found to consistently outperform PSO [37]. Yang employed and compared a FA with PSO for various test functions and found that the FA obtains better results than PSO and also a GA in terms of efficiency and success rate. It has also been found that the broadcasting ability of the FA gives better and quicker convergence towards optimality [34]. In a similar work by Yang and Deb, experimental results revealed that the FA outperforms other approaches such as PSO [38]. Sayadi introduced a FA with local search for minimizing the makespan in permutation flow scheduling problems and the initial results indicated that the proposed method performs better than an ACO algorithm [39]. Gandomi applied a FA to mixed variable structural optimization problems and the empirical results showed that the FA is better than PSO, GA, SA, and HSA [40]. Another work on the FA can be found in [35], where it was successfully applied to solve the economic emissions load dispatch problem. In light of the foregoing, it can be seen that the FA is more effective than some other methods, which motivated us to further investigate its performance with respect to the classification problem.

In the FA, the form of the attractiveness function of a firefly is denoted by the following:

$$\beta(r) = \beta_0 \exp(-\gamma r^2) \tag{1}$$

where r is the distance between any two fireflies, β_0 is the initial attractiveness at $r = 0$ (set to 1 in this work), and γ is an absorption coefficient that controls the decrease of the light intensity (also set to 1 in this work).

The distance between any two fireflies i and j at positions x_i and x_j, respectively, can be defined as a Cartesian or Euclidean distance as follows:

$$r_{ij} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{2}$$

where d is the dimensionality of the given problem.

The movement of a firefly i, which is attracted by a brighter firefly j, is represented by the following equation:

$$x_i = x_i = \beta_0 * \exp(-\gamma r_{ij}^2) * (x_j - x_i) + \alpha * \left(rand - \frac{1}{2}\right). \tag{3}$$

where the first term is the current position of a firefly, the second term is used to consider the attraction of a firefly towards the intensity of the light of neighbouring fireflies, and the third term is used for the random movement of a firefly when it cannot 'see' any brighter ones. The coefficient α is a randomization parameter determined by the problem of interest, while *rand* is a random number generator consistently distributed in the space [0, 1] [41].

3. Proposed method: hybridized FAs

According to Specht (1991) and Paliwal and Kumar (2009), the PNN is an effective approach for solving classification problems. A PNN has a relatively faster training process than the back-propagation NN and has an intrinsically analogous structure that ensures convergence with an optimal classifier because the size of the representative training set is maximized and training samples can be added or removed without extensive retraining [11,42]. A PNN consists of four layers: input, pattern, summation, and output, as shown in Fig. 1.

The input layer is the first layer of neurons. Each input neuron represents a separate attribute in the training/test datasets (for example, from x_1 to x_n). The number of inputs is equal to the number of attributes in the dataset. The values from the input data are