Neurocomputing 157 (2015) 256–263

Contents lists available at ScienceDirect

Neurocomputing

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

Using gravitational search algorithm in prototype generation for nearest neighbor classification

Mohadese Rezaei [1], Hossein Nezamabadi-pour [*,1]

Department of Electrical Engineering, Shahid Bahonar University of Kerman, PO Box 76619-133, Kerman, Iran

ARTICLE INFO

Article history:
Received 23 May 2014
Received in revised form
8 December 2014
Accepted 6 January 2015
Communicated by Yukun Bao
Available online 19 January 2015

Keywords:
Classification
K-nearest neighbor
Prototype generation
Gravitational search algorithm

ABSTRACT

In recent years, metaheuristic algorithms have emerged as a promising approach to solve clustering and classification problems. In this paper, gravitational search algorithm (GSA) which is one of the newest swarm based metaheuristic search techniques, is adapted to generate prototypes for nearest neighbor classification. The proposed method has been tested on several problems and the results are compared with those obtained by several state-of-the-art techniques. The comparison shows that our proposed method can achieve higher classification accuracy than the competing methods and has good performance in the field of prototype generation.

1. Introduction

Classification is a well-known task in machine learning, which is assigning new samples to one of the existing classes on the basis of extracted knowledge from the data [1]. This knowledge can facilitate decision making. One of the most widely used classification method is the *K*-nearest neighbor (*K-NN*) rule [2], which simply uses the raw training data to predict class label of test samples. To determine the class label of a test sample, the *K-NN* classifier finds *k* closest training samples to that test sample and assigns the most frequently represented class to it. Also, if the number of training samples approaches infinity, the performance of *K-NN* classifier asymptotically approximates the Bayesian classifier. Although the *K-NN* algorithm is simple and effective, it has the disadvantage that the whole training set must be preserved, thus it requires large memory. Moreover, to classify a new input sample, the *K-NN* classifier must search through all the training samples, which leads to high computational cost. In addition, the *K-NN* classification scheme is exactly sensitive to noise objects and outlier samples [3].

In order to overcome these limitations, data reduction techniques have been suggested [4,5]. In data reduction the main goal is to decide which features and instances from the training set should be retained for further use during the learning process. Therefore, from the point of view of samples, an initial given set of samples is modified by replacing a group of samples by a representative called prototype while approximately keeping its original classification accuracy. According to the way in which prototypes are obtained, two different existing approaches are prototype selection (PS) and prototype generation (PG). Prototype selection methods choose a subset of the original training set while the objective of prototype generation methods is to build a new set of prototypes that may be different from any sample in the original training set. PS methods assume that the best prototypes can be find among the original samples and this assumption is their principal drawback, but PG methods add artificial prototypes if needed. From another perspective, data reduction can be viewed as reducing the number of features. These methods are divided into feature selection (FS) [6] and feature extraction [7]. In FS methods a representative subset of features is selected from the initial data set and in feature extraction methods new features are created as a function of old features to represent the data.

Feature weighting [8] is another scheme to enhance *K-NN* classification which consist of assigning a weight to each feature to refine the computation of distances between samples. These weights can be regarded as degree of feature relevance. By hybridizing prototype generation or prototype selection methods with feature selection and feature weighting methods, the results can be improved [9–11]. This paper focuses on prototype generation.

Many efforts have been done to design a proper set of prototypes and various PG schemes have been reported in the literature. Two useful surveys are found in [12] and [13] and experimental study on PG methods has been done in [14]. One of the first PG algorithms was presented by Chang [15]. Chang's algorithm begins by assuming that every sample in the training set is its own prototype. Then these prototypes are merged progressively so that the classification accuracy is preserved. An improved version of Chang's method was proposed

* Corresponding author.
E-mail addresses: mo.rezaei@eng.uk.ac.ir (M. Rezaei), nezam@uk.ac.ir (H. Nezamabadi-pour).
[1] Tel./fax: +98 34 33235900.

http://dx.doi.org/10.1016/j.neucom.2015.01.008
0925-2312/

by Bezdek et al. [16]. Learning vector quantization (LVQ) is a very popular algorithm of generation scheme, introduced by Kohonen [17]. In LVQ algorithm the new position of prototypes are determined by a "reward-punishment" rule. To modify the original LVQ scheme, a number of methods such as decision surface mapping (DSM) [18], learning vector quantization with training counter (LVQTC) [19] and hybrid LVQ3 algorithm (HYB) [20] have been proposed.

Following these initial efforts, some works have been done concerning metaheuristic algorithms [21–23] in data reduction problems [24,25]. Metaheuristic search algorithms contain evolutionary and swarm intelligence algorithms. The focus of these algorithms for prototype generation is on adjusting the position of prototypes. In [26] ENPC algorithm is introduced, which is a genetic-based technique. This algorithm finds the optimal number of prototypes and their locations during an evolutionary process. In [27,28] differential evolution (DE) [29] is used to optimize the position of prototypes. In addition, particle swarm optimization (PSO) [30,31] and artificial immune algorithm [32] have been applied to generate an optimal set of prototypes. The DE and PSO approaches proposed in [27,30] encode a complete solution of PG problem in each particle; i.e. a set of prototypes has to be encoded in each particle. However, in IPADE [28] and AMPSO [31] just one prototype is encoded in each particle and the whole swarm is the potential solution to the PG problem. Experiments demonstrate that PSO [30] and IPADE [28] have high accuracy rates in the final set of prototypes.

Some approaches based on the hybridization of the PG and other data reduction techniques such as feature selection and feature weighting can be found in the literature. In [33] a hybrid approach which combines PG with FW is introduced, which incorporates a FW method within two different PG methods.

Some PG methods, which are based on metaheuristic algorithms such as PSO, DE and IPADE, have achieved better generalization accuracy than the classic methods such as PNN and LVQ3. But, in view of the fact that metaheuristics are approximate algorithms, the optimal solution for PG problem have not been obtained yet. On the other hand, the no-free-lunch theorem states that there is no specific algorithm to achieve the best solution for all optimization problems. Some algorithms give a better solution for some particular problems than others. This theory leads us to examine new metaheuristics in solving PG problem. Gravitational search algorithm (GSA) is one of the recent created swarm based metaheuristic search techniques, which has a flexible and well-balanced mechanism to enhance exploration and exploitation abilities. It seems that, since the construction of new solutions in Gravitational search algorithm is performed based on interactions among all individual in a swarm, the exploration ability of this algorithm is more than the others.

GSA has been inspired by Newtonian laws of gravity and motion [22]. In this algorithm, mass interactions are simulated and objects (agents) move through a search space under the influence of gravitation. Some works have already been done concerning GSA and its binary version [34] in different problems. For instance, in [35] and [36] GSA is used to solve data clustering problem and in [37–41] this optimization algorithm is applied to simultaneously optimize the input feature subset selection and the SVM parameter setting, solve combinatorial optimization problems, model linear and nonlinear filters, identify parameters of hydraulic turbine governing system and present a piecewise function that is applied in parameter identification of automatic voltage regulator (AVR) system. In [42] a prototype based classification approach that uses GSA has been recommended by our team. This approach uses just one prototype per class (1-PC) and finds the appropriate position of each prototype. In other words, the authors in [42] proposed a PG method only with one prototype for each class without considering its characteristics such as the shape of class in the feature space, multi/uni-modality of the class, the number of samples in the class etc. This may reduce the classification accuracy of the classifier with respect to a multi prototypes system whereas increases the memory efficiency and the speed of the classifier in the performance phase.

Thus the success of GSA in various optimization problems motivated us to use it in determining the prototypes in PG problem. Also, similar to [30], we examine the "vote rule" to achieve higher classification accuracy. The prototype generation phase is replicated S times and these S sets of prototype are used to classify each test sample. The "vote rule" is used to combine these S results. According to the vote rule, a test sample is assigned to the class with an absolute majority of agrees between S obtained results. In our study, we evaluate the performance of our GSA based method without vote rule (named as gravitational prototype generation (GPG)) and with vote rule (named as VGPG), using 20 benchmark datasets and the results are compared with eleven existing algorithms. The competing algorithms are the *K-NN*, 1-PC [42], LVQ3 of Kohonen [17], the MGauss [43], which is an adaptive PG method considered in the framework of mixture modeling by Gaussian distributions, the HYB algorithm [20], which is a hybridization of several prototype reduction techniques, the RSP3 [44], which splits the feature space and defines new prototypes in each partition, the ENPC [26], PSO [30], AMPSO [31], IPADE [28] and DE [27].

The rest of this paper is organized as follows. In Section 2 a background on the prototype generation problem and a prologue to gravitational search algorithm are given. In Section 3 the proposed method is introduced, and in Section 4 experimental results are presented. Finally, in Section 5, we summarize our conclusions.

2. Background

2.1. Prototype generation problem

The problem of prototype generation is defined as follows: let $Y = (y_1, y_2, \ldots, y_m, y_c)$ be a sample, which means each sample is represented by an m-dimensional vector and a real class label y_c. Then, assume that there is a training set (called TR) which consists of n samples ($TR = \{Y_i \mid i = 1, \ldots, n\}$) and a test set (called TS) containing t samples ($TS = \{Y_i \mid i = n+1, \ldots, n+t\}$). If a prototype generation algorithm is executed on training set, a set of r prototypes (called $GS = \{P_i \mid i = 1, \ldots, r\}$) is obtained that can be used to classify the samples of the test set by *K-NN* classifier.

According to [14], we can specify four main groups in PG methods: positioning adjustment [45], class relabeling [46], centroid-based [47] and space-splitting [48]. In positioning adjustment methods the position of a subset of prototypes from the initial set, is corrected using an optimization procedure. The generation mechanism of class relabeling methods consists of changing the class labels of samples from training set, which could be suspicious of having errors, and belonging to other different classes. Centroid-based techniques are based on generating artificial prototypes by merging a set of similar examples. The space-splitting set includes the techniques based on different heuristics to partition the feature space and to define new prototypes in each partition.

2.2. Gravitational search algorithm

In GSA, agents are considered as objects, and their masses are measured by their performances. All these objects attract each other by a gravity force, and this force causes the movement of all objects globally toward objects with heavier masses, which correspond to good solutions of the problem [22]. GSA has N objects in which the