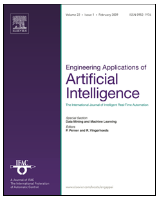
Engineering Applications of Artificial Intelligence 26 (2013) 2424–2430

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Facing the classification of binary problems with a hybrid system based on quantum-inspired binary gravitational search algorithm and K-NN method

XiaoHong Han*, Long Quan, XiaoYan Xiong, Bing Wu

Key Laboratory of Advanced Transducers and Intelligent Control Systems, Ministry of Education China, Taiyuan University of Technology, No. 79, West Yingze Street, Taiyuan 030024, China

ARTICLE INFO

Article history:
Received 8 January 2013
Received in revised form
9 April 2013
Accepted 21 May 2013
Available online 1 July 2013

Keywords:
Feature selection
Binary classification
Gravitational search algorithm
Binary gravitational search algorithm
Quantum computing
K-nearest neighbor

ABSTRACT

Since given classification data often contains redundant, useless or misleading features, feature selection is an important pre-processing step for solving classification problems. This problem is often solved by applying evolutionary algorithms to decrease the dimensional number of features involved. Removing irrelevant features in the feature space and identifying relevant features correctly is the primary objective, which can increase classification accuracy. In this paper, a novel QBGSA–K-NN hybrid system which hybridizes the quantum-inspired binary gravitational search algorithm (QBGSA) with the K-nearest neighbor (K-NN) method with leave-one-out cross-validation (LOOCV) is proposed. The main aim of this system is to improve classification accuracy with an appropriate feature subset in binary problems. We evaluate the proposed hybrid system on several UCI machine learning benchmark examples. The experimental results show that the proposed method is able to select the discriminating input features correctly and achieve high classification accuracy which is comparable to or better than well-known similar classifier systems.

1. Introduction

In the last decade, feature selection has become an important technique in many practical applications of classification in large datasets. Since redundant or irrelevant features increase the size of search space and make generalization more difficult, dimensionality reduction of feature vector has become an imperative task in various areas such as pattern recognition (Mitra et al., 2002), data mining (Dash et al., 2002), gene selection from microarray data (Guyon and Elisseeff, 2003), text categorization (Pourhabibi et al., 2011) and multimedia information retrieval (Lew, 2001; Liu and Dellaert, 1998). The potential benefits of feature selection include reducing the risk of over-fitting (Guyon and Elisseeff, 2003; Maldonado and Weber, 2009), decreasing the model's complexity which improves its generalization ability, less computational effort and reducing the dimensionality to improve prediction performance.

Feature selection involves defining the most informative and discriminative features in the original data for classification. This can be performed by eliminating the redundant, uninformative, and noisy features. In general, there are two different feature selection algorithms: filter methods and wrapper methods (Dash and Liu, 1997; Guyon and Elisseeff, 2003; Liu and Yu, 2005). Filter methods make use of intrinsic properties of databases, being independent of the learning machines. Given that filter methods select the relevant attributes without using a learning machine, they are generally faster and more computationally efficient than wrapper methods for high-dimensional databases. By contrast, wrapper methods select the subset of features based on the results of learning algorithms. In terms of classification accuracy, wrapper methods generally promise better results than filter methods because the search for the best feature subset is based on prediction accuracy (Guyon and Elisseeff, 2003; Kohavi and John, 1997). In this paper, we adopted a wrapper method by using an optimization algorithm for feature selection.

Basically, the problem of feature selection could be treated as a problem of optimization in a search space. Feature selection method based on stochastic search algorithms has attracted great attention. Several methods have been proposed to perform feature selection using evolutionary techniques. Raymer et al. (2000) suggested using genetic algorithms (GA) to tackle the problem. Authors in (Bello et al., 2007; Tanaka et al., 2007; Wang et al., 2007) proposed to use binary particle swarm optimization (PSO) for feature selection. Zhang and Sun applied tabu search in this problem (Zhang and Sun, 2002).

Gravitational search algorithm (GSA) is one of the newest evolutionary optimization algorithms inspired by the Newtonian laws of gravity and motion, which first was proposed by Rashedi et al. (2009). In GSA, an object in the search space attracts every other one with a force that is directly proportional to the product

* Corresponding author. Tel.: +86 13485325973.
E-mail address: jmqchs@sohu.com (X. Han).

0952-1976/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2013.05.011

of their objects and inversely proportional to the square of the distance between them. Further, Rashedi et al. (2010) introduced the binary gravitational search algorithm (BGSA), which is a slightly modified algorithm of the original GSA designed to solve various nonlinear benchmark functions.

Recently, Ibrahim et al. (2012) developed a quantum-inspired BGSA (QBGSA) for solving the optimal PQM placement problem in power systems. The QBGSA integrates the concept and principles of quantum computing into BGSA in order to avoid premature convergence and improve efficiency of the original BGSA (Han and Kim, 2002; Vlachogiannis and Lee, 2008; Jeong et al., 2010). The performance of the QBGSA was compared with the BGSA, the quantum-inspired binary particle swarm optimization (QBPSO) and BPSO. The comparison results indicated that the QBGSA is the most effective and precise among the aforementioned optimization algorithms. Motivated by these studies, this paper has made an attempt to propose a new method for feature selection by means of the QBGSA and the K-nearest neighbor (K-NN) method, in which the K-NN method based on Euclidean distance calculations serves as a classifier for evaluating classification accuracies (Oh et al., 2004). We apply the proposed method to several UCI machine learning benchmark examples. As far as we know, we are the first to develop a method based on QBGSA and K-NN for feature selection. The remainder of this paper is organized as follows. Section 2 introduces the related work used in this study, namely, K-nearest neighbor, binary GSA and Quantum-inspired binary GSA. Section 3 presents the proposed hybrid system for feature selection. In Section 4, we present the experimental results. Finally, a brief conclusion is offered in Section 5.

2. Related work

2.1. K-nearest neighbor method

The K-nearest neighbor (K-NN) method is one of the most popular nonparametric methods introduced by Fix and Hodges in 1951 (Cover and Hart, 1967; Fix and Hodges, 1989; Tan, 2006). Since there is only one parameter K (the number of nearest neighbors) which needs to be determined, it is easy to implement the K-NN method. The number of nearest neighbors is key to the performance of the classification process. K-NN classifies a new object from the testing samples to the training samples based on the minimum distance which is calculated according to Euclidean distance. If an object is close to the K nearest neighbors, the object is categorized into the K-object category. For the purpose of enhancing the classification accuracies, the parameter K must be altered according to the different data set characteristics. In this paper, we use the leave-one-out cross-validation (LOOCV) method to choose the parameter K. When n samples need to be classified, they are divided into one testing sample and $n-1$ training samples at each iteration in the process of evaluating, and a classifier is constructed by training the $n-1$ training samples. The testing sample class can be evaluated by the classifier.

2.2. Binary gravitational search algorithm

GSA is a newly developed stochastic search algorithm based on the law of gravity and mass interactions (Rashedi et al., 2009; Rashedi, 2007; Rashedi et al., 2007). This approach simulates mass interactions, and moves through a multi-dimensional search space under the influence of gravitation. In GSA, agents are considered as objects and their performances are measured by their masses; these objects attract each other by gravitational force which causes a global movement of all objects towards objects with heavier masses (Rashedi et al., 2009; Rashedi, 2007; Rashedi et al., 2007).

Assumed there are k objects (masses), the position of the ith object is defined as follows:

$$X_i=(x_i^1,\ldots,x_i^d,\ldots,x_i^n),\quad i=1,2,\ldots,k \tag{1}$$

where $x_i{}^d$ denotes the position of ith object in the dth direction. The force exerting on the object i from the object j is defined as follows:

$$F_{ij}^d(t)=G(t)\frac{M_i(t)\times M_j(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t)-x_i^d(t)) \tag{2}$$

where M_j is the mass related to object j, M_i is the mass related to object i, G is gravitational constant at time t, ε is a small constant, and $R_{ij}(t)$ is the Euclidian distance between two objects i and j. $G(t)$ is a decreasing function of time, which is set to G_0 at the beginning and decreases exponentially towards zero with lapse of time. The total force $F_i^d(t)$ that exerts on object i in the dth direction is a randomly weighted sum of dth components of the forces from other agents:

$$F_i^d(t)=\sum_{j=1,j\neq i}^{k} rand_j F_{ij}^d(t) \tag{3}$$

where $rand_j$ is a uniform random variable in the interval [0,1].

The acceleration of the object i, $a_i^d(t)$, at time t and in the dth direction, is given as follows:

$$a_i^d(t)=\frac{F_i^d(t)}{M_{ii}(t)} \tag{4}$$

where M_{ii} is the inertial mass of the object i. Its next velocity $v_i^d(t+1)$ and its next position $x_i^d(t+1)$ are calculated as follows:

$$v_i^d(t+1)=rand_i\times v_i^d(t)+a_i^d(t) \tag{5}$$

$$x_i^d(t+1)=x_i^d(t)+v_i^d(t+1) \tag{6}$$

where $rand_i$ is a uniform random variable in the interval [0,1]. This random number is applied to give a randomized characteristic to the search, $v_i^d(t)$ and $x_i^d(t)$ are its current velocity and position, respectively.

The masses of objects are evaluated by the fitness function. Assuming the equality of the gravitational and inertia mass, the mass $M_i(t)$ is updated by the following equations:

$$M_i=M_{ii}, i=1,2,\ldots,k, \tag{7}$$

$$m_i(t)=\frac{fit_i(t)-worst(t)}{best(t)-worst(t)}, \tag{8}$$

$$M_i(t)=\frac{m_i(t)}{\sum_{j=1}^{k} m_j(t)}, \tag{9}$$

$$best(t)=\min_{j\in\{1,\ldots,k\}} fit_j(t), \tag{10}$$

$$worst(t)=\max_{j\in\{1,\ldots,k\}} fit_j(t), \tag{11}$$

where $fit_i(t)$ represents the fitness value of the object i at time t. A larger mass indicates a more efficient object. This means that more efficient objects possess greater attractions and move more slowly.

However, many optimization problems occur in a space featuring discrete, qualitative distinctions between variables and between levels of variables. To apply the GSA to binary space, the binary version of GSA was introduced by Rashedi et al. (2010), which has the same formulation introduced above, but with a different equation for updating the position of each mass.

As already mentioned, feature selection can be seen as an optimization problem in a binary space, where the goal is to select