Information Sciences 197 (2012) 65–76

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Novel swarm optimization for mining classification rules on thyroid gland data

Wei-Chang Yeh

Integration & Collaboration Laboratory, Advanced Analytics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, P.O. Box 123, Broadway, NSW 2007, Australia
Department of Industrial Engineering and Engineering Management, National Tsing Hua University, P.O. Box 24-60, Hsinchu 300, Taiwan, ROC

ARTICLE INFO

Article history:
Received 24 June 2010
Received in revised form 31 October 2011
Accepted 10 February 2012
Available online 18 February 2012

Keywords:
Data mining
Simplified swarm optimization (SSO)
Classification rules
Thyroid gland data
Orthogonal array test (OAT)

ABSTRACT

This work uses a novel rule-based classifier design method, constructed by using improved simplified swarm optimization (SSO), to mine a thyroid gland dataset from UCI databases. An elite concept is added to the proposed method to improve solution quality, close interval encoding (CIE) is added to efficiently represent the rule structure, and the orthogonal array test (OAT) is added to powerfully prune rules to avoid over-fitting the training dataset. To evaluate the classification performance of the proposed improved SSO, computer simulations are performed on well-known thyroid gland data. Computational results compare favorably with those obtained using existing algorithms such as conventional classifiers, including Bayes classifier, k-NN, k-Means, and 2D-SOM, and soft computing based methods such as the simple SSO, immune-estimation of distribution algorithms (IEDA), and genetic algorithm (GA).

1. Introduction

Data mining is an efficient approach for analyzing and discovering knowledge from a large complex dataset of heterogeneous quality, for which a variety of data mining tools have been developed. The rule-based classifier is one such important tool [1–7,12,13,17,20–22,26–28,30–40] which mines a small set of IF-THEN rules (e.g., IF condition THEN conclusion) from training data for classification with predicted classes, and then uses this rule set to predict new data instances [25]. The rule-based classifier has the advantage of generating high-level symbolic-knowledge representation, which increases the comprehensibility of discovered knowledge [1–7,12,13,17,20–22,26–28,30–40], and it has been extensively applied to many real-world problems in medicine, social sciences, management, and engineering [1–7,12,13,17,20–22,26–28,30–40].

Many conventional algorithms have been proposed for classification such as the Bayes classifier [18,19], k-NN, k-Means, and 2D-SOM (self-organizing map) [6,11,15,23]. Soft computing methods (SCs) have been utilized to find optimal or good quality solutions to complex optimization problems in a number of fields [4,5,8–10,13–22,24,30–40]. Consequently, many new data mining techniques have been based on SC, such as the genetic algorithm (GA, a biology-inspired SC) [5,14,31] and particle swarm optimization (PSO, a swarm-intelligence SC) [1,2,8,9,13,17,21,22,26,28,30,32–40].

Swarm-intelligence is an artificial intelligence, primarily inspired by the social behavior patterns of self-organized systems, that considers the interactions among large groups of individuals [1,2,8,9,13,17,21,22,26,28,30,32–40]. The simplified swarm optimization (SSO) proposed by Yeh is a population-based stochastic optimization technique that belongs to the swarm-intelligence category [37–39] and is also an evolutionary computational method inspired by PSO [37]. Also known as discrete PSO (DPSO), SSO was originally proposed to overcome the drawbacks of PSO for discrete-type optimization

E-mail address: yeh@ieee.org

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2012.02.009

problems (e.g., the multi-level redundancy allocation in series systems [38]). Although simple SSO has been successfully applied to classify breast-cancer data [39], it is only valid for discrete data; hence, there is a need to extend simple SSO to efficiently and effectively solve this problem with more complex data.

This work has two principal goals. First, a modification is introduced to simple SSO to deal with non-discrete data by integrating close interval encoding (CIE), the elite concept, and the orthogonal array test (OAT) to establish the classification rules. To demonstrate its efficiency, the proposed algorithm is tested on thyroid gland data, a famous dataset in the UCI database [3,4,7,12,31].

This paper is organized as follows. Section 2 provides a description of simple SSO. The proposed CIE, the elite concept, and OAT are described in Sections 3–5, respectively. The proposed improved SSO combining CIE, the elite concept, and OAT are discussed in Section 6. Three comparisons based on two experiments on the UCI thyroid gland dataset demonstrate the effectiveness of the proposed improved SSO in Section 7. Finally, the conclusion is presented in Section 8.

2. Introduction to SSO

The advantages of SSO are its simplicity, efficiency, and flexibility [37–39]. In its early development, SSO was introduced by the author to solve the redundancy allocation problem by simplifying the traditional PSO procedure to overcome the drawbacks of PSO for discrete variables [37]. Since SSO was only valid for discrete data, it was initially called discrete particle swarm optimization (DPSO) but was later renamed SSO due to its simplicity of use and the introduction of a different schematic to update the solutions (called 'particles' in PSO). The basics of SSO are introduced in this section before discussing the proposed improved SSO.

Like most SCs such as GA, SA, ACO, PSO, or ABC, SSO is also initialized with a population of random solutions inside the problem space and it then searches for optimal solutions by updating generations. Let DIM, POP, GEN, and REP represent the number of attributes, populations, generations, and independent replications, while parameters c_w, c_p, c_g, and c_r represent the probabilities of the new variable value generated from the current solution, *pBest*, *gBest* and a random number in SSO, respectively, where $c_w + c_p + c_g + c_r = 1$. In the update mechanism of SSO, the ith solution at generation $t+1$ $X_i^{t+1} = \left(x_{i1}^{t+1}, x_{i2}^{t+1}, \ldots, x_{i,\mathrm{DIM}}^{t+1}\right)$ is a compromise of the current ith solution at generation t $X_i^t = \left(x_{i1}^t, x_{i2}^t, \ldots, x_{i,\mathrm{DIM}}^t\right)$; the personal best of the current ith solution (*pBest*) $P_i = (p_{i1}, p_{i2}, \ldots p_{i,\mathrm{DIM}}) \in \left\{X_i^1, X_i^2, \ldots, X_i^t\right\}$; the global best value of the whole swarm (*gBest*) $G = (g_1, g_2, \ldots, g_{\mathrm{DIM}}) \in \{P_1, P_2, \ldots, P_{\mathrm{POP}}\}$; and a random movement after $C_w = c_w$, $C_p = C_w + c_p$ and $C_g = C_p + c_g$ is given [37–39] as follows:

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t & \text{if } \rho \in [0, C_w) \\ p_{ij} & \text{if } \rho \in [C_w, C_p) \\ g_i & \text{if } \rho \in [C_p, C_g) \\ x & \text{if } \rho \in [C_g, 1) \end{cases} \tag{1}$$

where x_{ij}^t is the jth variable of the ith solution at generation t; ρ represents a random numbers uniformly distributed in [0, 1]; x represents random numbers uniformly distributed in $[l_j, u_j]$; and l_i, u_i are the lowest and greatest values of the ith attribute, respectively.

In Eq. (1), both concepts of *pBest* and *gBest* are adopted directly from the traditional PSO. Let $F(\bullet)$ be the fitness (function) value for $\bullet$. Then, P_i is a local best such that $F(P_i) \geqslant F\left(X_i^j\right)$ for $j = 1, 2, \ldots, t$, to push X_i^t to climb the local optimum (i.e., local exploitation), and G is a global best such that $F(G) \geqslant F(P_i)$ for $i = 1, 2, \ldots$, POP to guide the search towards unexplored regions to find the global optimizer (i.e., global exploration). To maintain population diversity and enhance the capacity of escaping from a local optimum, a random movement is added to the update mechanism of SSO [37–39].

The above update mechanism considers the interaction between individuals and maintains the diversity between them. It is much simpler than other major SC techniques such as PSO, which needs to calculate both velocity and position functions; GA, which requires genetic operations such as cross-over and mutation; estimation of distribution algorithm (EDA), which proves difficult and complicated for building an appropriate probability model; and IMA, which does not consider the interaction of variables [4]. The overall steps of SSO are listed as follows:

PROCEDURE SSO

STEP S0. Generate X_i randomly, let $t = 1$, $P_i = X_i$, and $G = P_j$, where $F(P_j) = \max_i \{F(P_i)\}$ for $i = 1, 2, \ldots$, POP.
STEP S1. Let $i = 1$.
STEP S2. Update X_i based on Eq. (1) and calculate $F(X_i)$.
STEP S3. If $F(X_i) > F(P_i)$, let $P_i = X_i$; else go to STEP S5.
STEP S4. If $F(P_i) > F(G)$, then let $G = P_i$.
STEP S5. If $i <$ POP, then let $i = i + 1$ and go to STEP S2.
STEP S6. If $t =$ GEN, then G is the final solution and stop; otherwise let $t = t + 1$ and go to STEP S1.

A simple example of the procedure for updating a solution using SSO is illustrated below according to [37]. Let = (2.3,3.5,5.6,4.2,7.8), $P_4 = (6.6, 7.7, 4.7, 2.5, 8)$, $G = (5.4, 3.1, 5.2, 4.5, 8)$, $(l_1, l_2, l_3, l_4, l_5) = (2, 2, 4, 2.5, 4)$, $(u_1, u_2, u_3, u_4, u_5) = (7.5, 8, 7, 5, 8)$, $\rho = (.94, .58, .37, .11, .79)$, and $(C_w, C_p, C_g) =$ (.15,.40,.75). Since