Neural Networks 80 (2016) 19–33

Contents lists available at ScienceDirect

Neural Networks

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

A swarm-trained k-nearest prototypes adaptive classifier with automatic feature selection for interval data

Telmo M. Silva Filho *, Renata M.C.R. Souza *, Ricardo B.C. Prudêncio

Universidade Federal de Pernambuco, Centro de Informática, Av. Jornalista Aníbal Fernandes, s/n, 50.740-560 Recife (PE), Brazil

ARTICLE INFO

Article history:
Received 30 July 2015
Received in revised form 11 April 2016
Accepted 12 April 2016
Available online 19 April 2016

Keywords:
Swarm optimization
Weighted distance
Prototype learning
Feature selection
Symbolic data analysis
Interval data

ABSTRACT

Some complex data types are capable of modeling data variability and imprecision. These data types are studied in the symbolic data analysis field. One such data type is interval data, which represents ranges of values and is more versatile than classic point data for many domains. This paper proposes a new prototype-based classifier for interval data, trained by a swarm optimization method. Our work has two main contributions: a swarm method which is capable of performing both automatic selection of features and pruning of unused prototypes and a generalized weighted squared Euclidean distance for interval data. By discarding unnecessary features and prototypes, the proposed algorithm deals with typical limitations of prototype-based methods, such as the problem of prototype initialization. The proposed distance is useful for learning classes in interval datasets with different shapes, sizes and structures. When compared to other prototype-based methods, the proposed method achieves lower error rates in both synthetic and real interval datasets.

1. Introduction

Classic data are usually defined as vectors of quantitative or qualitative variables. Due to this representation, classic data analysis is not able to satisfactorily deal with variability or uncertainty of complex data. Based on this motivation, symbolic data analysis (SDA) introduced a number of representations for data types that better represent data variability, such as intervals, histograms, lists of values, among others (Diday & Noirhomme-Fraiture, 2008). The most common of these data types is the interval data. Intervals naturally arise from the description of ranges of values, such as daily temperature variation, daily stock prices, among others (Bock & Diday, 2000).

A significant amount of research has been devoted to interval data in the last decade, resulting in both unsupervised (Costa, Pimentel, & Souza, 2013; de Almeida, Souza, & Candeias, 2012; Hajjar & Hamdan, 2011; Hamdan & Hajjar, 2011; Pimentel, da Costa, & Souza, 2011; Pimentel & Souza, 2012) and supervised methods (D'Oliveira, De Carvalho, & Souza, 2004; Mali & Mitra, 2005; Roque, Maté, Arroyo, & Sarabia, 2007; Rossi & Conan-guez, 2002; Silva & Brito, 2006; Silva Filho & Souza, 2012, 2013; Souza, Queiroz, & Cysneiros, 2011). In this work, we focus on investigating prototype and distance-based methods for classification of interval data. These methods have many advantages, such as metric versatility, diversity of training procedures and domain modeling ability.

In our work, we address three different issues concerning prototype-based classification methods for interval data. The first issue is related to the adopted distance function. In De Carvalho, Brito, and Bock (2006), the authors proposed the first distance function for intervals, which was based on the squared Euclidean distance. Despite its novelty at that time, this distance could only model spherical subregions of data, that is, it assumed that features have similar scales and dispersions. In order to overcome this limitation, Silva Filho and Souza (2012) extended the learning vector quantization (LVQ) method to interval data (the WILVQ method) and proposed a weighted distance that considers different feature dispersions in subregions of data modeled by a prototype. In addition, Silva Filho and Souza (2013) proposed the weighted fuzzy learning vector quantization (WFLVQ), in which the weighted distance is extended to model classes with varying sizes, shapes and structures, resulting in better classification performance compared to previous work which adopted non-adaptive distances.

The second issue of prototype-based methods addressed in our work is the local minima problem due to non-optimal prototype initialization, i.e., the quality of these methods depends on a well-placed set of prototypes. Weighted algorithms (Silva Filho & Souza, 2012, 2013) have obtained better prototype placement, but

* Corresponding authors.
E-mail addresses: tmsf@cin.ufpe.br (T.M. Silva Filho), rmcrs@cin.ufpe.br (R.M.C.R. Souza), rbcp@cin.ufpe.br (R.B.C. Prudêncio).

http://dx.doi.org/10.1016/j.neunet.2016.04.006

they still can benefit from optimization of prototype initialization. One way to tackle this issue is to employ swarm optimization methods to prototype selection, which has been successfully done for classic point data in different prototype-based classifiers (Cervantes, Galvan, & Isasi, 2009; De Falco, Della Cioppa, & Tarantino, 2007; Dilmac & Korurek, 2013; Szabo & de Castro, 2010). Swarm techniques can also be useful to automatically find the number of prototypes for a dataset (Cervantes et al., 2009; Szabo & de Castro, 2010) and prune unnecessary prototypes. Research has yet to be done on prototype-based classifiers for interval data, which can also benefit from the advantages of swarm-based algorithms.

Finally, the third issue addressed in our work is the selection of relevant features in prototype-based methods. Feature selection is commonly a useful mechanism to achieve better classification performance. As with prototype selection, feature selection can also be treated as an optimization task. Many swarm-based algorithms for point data have performed automatic feature selection in classification (Ghamisi & Benediktsson, 2015; Ghamisi, Couceiro, & Benediktsson, 2015; Lin, Zhang, & Hung, 2014; Nakamura et al., 2012), although none of these works deal with prototype-based classifiers. Therefore, this is an open subject. For interval data, in turn, existing prototype-based classifiers (Silva Filho & Souza, 2012, 2013) use all available features in a dataset in order to build a model. It is expected that these classifiers can also benefit from automatic feature selection.

Motivated by the three issues above, a new prototype-based classifier was proposed to: (i) model classes in a flexible way, by employing a new adaptive distance; (ii) be robust against poor prototype initialization and unused prototypes; and (iii) perform automatic feature selection. As explained before, the adaptive distance for interval data adopted by the WFLVQ method (Silva Filho & Souza, 2013) has shown to be more effective compared to non-adaptive distances. Despite its advantages, this distance still has some limitations. It suffers from information loss because its weights are calculated separately. Additionally, it makes no distinction between prototypes that affect many instances and prototypes that are far away from their affected instances, which are two very different scenarios. Therefore, the first contribution of the current work is a new weighted Euclidean distance for interval data that overcomes these limitations.

In order to deal with prototype and feature selection, the prototype-based classifier is trained by a recently proposed hybrid swarm optimization method: the velocity-based artificial bee colony algorithm (VABC) (Imanian, Shiri, & Moradi, 2014). This method combines the advantages of both particle swarm optimization (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995) and artificial bee colony (Karaboga & Basturk, 2007) to avoid local minima and to reach better solutions. The VABC was adapted in the proposed prototype-based classifier to adequately deal with interval data. Additionally, we investigated new mechanisms to remove unused prototypes, which can affect the learning performance of prototype-based classifiers as well.

In our experimental analysis, we compared the proposed method to WFLVQ and to three prototype-based classifiers trained by different versions of swarm-based algorithms (De Falco et al., 2007; Imanian et al., 2014; Zhang, Xiong, & Zhang, 2013). Experiments confirmed the usefulness of the proposed method: it shows the best overall classification error rate, with small standard deviations (which relates to avoiding local minima), removes unnecessary prototypes and is able to select features in a meaningful way.

The remainder of this work is organized as follows. Sections 2 and 3 briefly review the algorithms the current work is based on. Section 4 describes the new classifier. Experiments using synthetic and real interval datasets are presented in Section 5. Finally, Section 6 presents final comments and suggestions for further research.

2. Velocity-based artificial bee colony

Swarm optimization algorithms are known for their ability to escape from local minima, which makes them very versatile (Izakian & Abraham, 2011). There are many different types of swarm-based methods, but the current work is based on a combination of particle swarm optimization (PSO) (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995) and artificial bee colony (ABC) (Karaboga & Basturk, 2007), called velocity-based artificial bee colony (Imanian et al., 2014).

Both PSO and ABC have shown good optimization results and have been adopted in many applications, but they have their own limitations. In the case of PSO, if the global best particle falls into a local minimum, the algorithm might not be able to avoid moving the other particles in that direction, reaching a badly optimized solution. By abandoning stagnated food sources, ABC may be able to avoid this scenario. On the other hand, ABC does many small adjustments to a food source before abandoning it, which may lead to bad exploitation (Imanian et al., 2014). VABC was proposed to overcome these limitations, by combining the strengths of both ABC and PSO.

As the ABC algorithm, VABC consists of food sources, which encode solutions, and three bee types: employed, onlooker and scout bees. All food sources are initialized by scout bees. Let SN be the number of food sources and let the vector $\vec{X}_l$, $(l = 1 \dots \mathrm{SN})$, be a solution to the optimization problem. Each $\vec{X}_l$ contains p variables $(X_{lj},\ j = 1 \dots p)$, which must be tuned to optimize (minimize or maximize) an objective function $f(\vec{X}_l)$. After the solution vectors are initialized, employed bees will feed near existing food sources, which is performed according to Eq. (1).

$$X'_{lh} = X_{lh} + \phi * (X_{lh} - X_{kh}), \tag{1}$$

where $\vec{X}_l$ is the memorized food source, $\vec{X}_k$ is a randomly selected food source, which is not the same as $\vec{X}_l$, h is a randomly selected variable and ϕ is a random number within the range $[-1, 1]$. Then, the fitness value of the new food source $\vec{X'}_l$ is calculated. The bee will memorize either $\vec{X}_l$ or $\vec{X'}_l$, based on a greedy selection.

After the employed bees finish their search, they pass the fitness information of the food sources to the onlooker bees. Based on this information, each food source has a probability p_l of being chosen by an onlooker bee. This probability is calculated by Eq. (2), based on the fitness value $f(\vec{X}_l)$.

$$p_l = \frac{f(\vec{X}_l)}{\sum_{k=1}^{\mathrm{SN}} f(\vec{X}_k)}. \tag{2}$$

In VABC, each onlooker bee will choose a food source $\vec{X}_k$, based on these probabilities, and will try to find a better food source $\vec{X'}_k$ by employing the velocity-based particle update equations of PSO (see Eqs. (3) and (4)). The fitness value of the new food source $\vec{X'}_k$ is calculated and the onlooker bee will memorize either $\vec{X}_k$ or $\vec{X'}_k$, based on a greedy selection.

$$\begin{aligned} \vec{V}_k(t+1) \ &= \ \omega\vec{V}_k(t) + (c_1 r_1) * (\mathrm{pbest}_k(t) - \vec{X}_k(t)) \\ &\quad + (c_2 r_2) * (\mathrm{gbest}(t) - \vec{X}_k(t)), \end{aligned} \tag{3}$$

$$\vec{X}_k(t+1) = \vec{X}_k(t) + \vec{V}_k(t), \tag{4}$$

where $\vec{X}_k(t)$ and $\vec{V}_k(t)$ are, respectively, the position and the velocity of the kth particle/food source at instant t, ω is the inertia value, c_1 and c_2 are acceleration coefficients, $\mathrm{pbest}_k(t)$ is the best position that the kth food source achieved until instant t, $\mathrm{gbest}(t)$ is the best position achieved by the swarm until instant t and r_1 and r_2 are random numbers chosen from the interval [0, 1].