Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 41 (2008) 2693–2709

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/pr

A memetic algorithm for evolutionary prototype selection: A scaling up approach[☆]

Salvador García[a,*], José Ramón Cano[b], Francisco Herrera[a]

[a]Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
[b]Department of Computer Science, University of Jaén, 23700 Linares, Jaén, Spain

Received 8 September 2007; received in revised form 18 January 2008; accepted 14 February 2008

Abstract

Prototype selection problem consists of reducing the size of databases by removing samples that are considered noisy or not influential on nearest neighbour classification tasks. Evolutionary algorithms have been used recently for prototype selection showing good results. However, due to the complexity of this problem when the size of the databases increases, the behaviour of evolutionary algorithms could deteriorate considerably because of a lack of convergence. This additional problem is known as the scaling up problem.

Memetic algorithms are approaches for heuristic searches in optimization problems that combine a population-based algorithm with a local search. In this paper, we propose a model of memetic algorithm that incorporates an ad hoc local search specifically designed for optimizing the properties of prototype selection problem with the aim of tackling the scaling up problem. In order to check its performance, we have carried out an empirical study including a comparison between our proposal and previous evolutionary and non-evolutionary approaches studied in the literature.

The results have been contrasted with the use of non-parametric statistical procedures and show that our approach outperforms previously studied methods, especially when the database scales up.

Keywords: Data reduction; Evolutionary algorithms; Memetic algorithms; Prototype selection; Scaling up; Nearest neighbour rule; Data mining

1. Introduction

Considering supervised classification problems, we usually have a training set of samples in which each example is labelled according to a given class. Inside the family of supervised classifiers, we can find the nearest neighbour (NN) rule method [1,2] that predicts the class of a new prototype by computing a similarity [3,4] measure between it and all prototypes from the training set, called the k-nearest neighbours (k-NN) classifier. Recent studies show that k-NN classifier could be improved by employing numerous procedures. Among them, we could cite proposals on instance reduction [5,6], for incorporating weights for improving classification [7], and for accelerating classification task [8], etc.

Prototype selection (PS) is an instance reduction process consisting of maintaining those instances that are more relevant in the classification task of the k-NN algorithm and removing the redundant ones. This attempts to reduce the number of rows in data set with no loss of classification accuracy and obtain an improvement in the classifier. Various approaches of PS algorithms were proposed in the literature, see Refs. [6,9] for review. Another process used for reducing the number of instances in training data is the prototype generation, which consists of building new examples by combining or computing several metrics among original data and including them into the subset of training data [10].

Evolutionary algorithms (EAs) have been successfully used in different data mining problems (see Refs. [11–13]). Given that PS problem could be seen as a combinatorial problem, EAs [14] have been used to solve it with promising results [15], which we have termed evolutionary prototype selection (EPS).

[☆] This work was supported by Projects TIN2005-08386-C05-01 and TIN2005-08386-C05-03. S. García holds a FPU scholarship from Spanish Ministry of Education and Science.

[*] Corresponding author. Tel.: +34 958 240598; fax: +34 958 243317.
E-mail addresses: salvagl@decsai.ugr.es (S. García), jrcano@ujaen.es (J.R. Cano), herrera@decsai.ugr.es (F. Herrera).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.02.006

The increase of the database's size is a staple problem in PS (which is known as scaling up problem). This problem produces excessive storage requirement, increases time complexity and affects generalization accuracy. These drawbacks are presented in EPS because they result in an increment in chromosome size and time execution and also involve a decrease in convergence capabilities of the EA. Traditional EPS approaches generally suffer from excessively slow convergence between solutions because of their failure to exploit local information. This often limits the practicality of EAs on many large-scale problems where computational time is a crucial consideration. A first rapprochement about the use of EAs when this problem scales up can be found in Ref. [16].

The combination of EAs with local search (LS) was named "memetic algorithm" (MA) in Ref. [17]. Formally, a MA is defined as an EA that includes one or more LS phases within its evolutionary cycle [18]. The choice of name is inspired by concept of a *meme*, which represents a unit of cultural evolution that can show local refinement [19]. MAs have been shown to be more efficient (i.e., needing fewer evaluations to find optima) and more effective (identifying higher quality solutions) than traditional EAs for some problem domains. In the literature, we can find a lot of applications of MAs for different problems; see Ref. [20] for an understanding of MA issues and examples of MAs applied to different domain problems.

The aim of this paper is to present a proposal of MA for EPS for dealing with the scaling up problem. The process of designing effective and efficient MAs currently remains fairly ad hoc. It is frequently hidden behind problem-specific details. In our case, the meme used is ad hoc designed for the PS problem, taking advantage of its divisible nature and simplicity of hybridization within the EA itself, and allowing us good convergence with increase of the problem size. We will compare it with other EPS and non-EPS algorithms already studied in the literature, paying special attention to the scaling up problem, analysing its behaviour when we increase the problem size.

This paper is organized in the following manner. Section 2 presents the PS problem formally and enumerates some PS methods. A review of EPS is given in Section 3. In Section 4 we explain our MA approach and meme procedure. Details of empirical experiments and results obtained are reported in Section 5. Section 6 contains a brief summary of the work and the conclusions reached.

2. Preliminaries: PS

PS methods are instance selection methods [5] which expect to find training sets offering best classification accuracy by using the nearest neighbour rule (1-NN).

A formal specification of the problem is the following: Let $\overrightarrow{x_p}$ an example where $\overrightarrow{x_p} = (x_{p1}, x_{p2}, \ldots, x_{pm}, x_{pl})$, with $\overrightarrow{x_p}$ belonging to a class c given by x_{pl} and a m-dimensional space in which x_{pi} is the value of the ith feature of the pth sample. Then, let us assume that there is a training set TR which consists of n instances $\overrightarrow{x_p}$ and a test set TS composed by t instances $\overrightarrow{x_p}$. Let $S \subseteq TR$ be the subset of selected samples resulted for the execution of a PS algorithm, then we classify a new pattern from TS by the 1-NN rule acting over S.

Wilson and Martinez in Ref. [6] suggest that the determination of the k value in the k-NN classifier may depend according to the proposal of the PS algorithm. In k-NN, setting k greater than 1, decreases the sensitivity of the algorithm to noise and tends to smooth the decision boundaries. In some PS algorithms, a value $k > 1$ may be convenient, when its interest lies in protecting the classification task of noisy instances. In any case, Wilson and Martinez state that it may be appropriate to find a value of k to use during the reduction process, and then redetermine the best value of k in the classification task. In EPS we have used the value $k = 1$, given that EAs need to have the greatest possible sensitivity to noise during the reduction process. In this manner, an EPS algorithm could better detect the noisy instances and the redundant ones in order to find a good subset of instances perfectly adapted to the simplest method of NNs. By considering only an instance during the evolutionary process, the reduction–accuracy trade-off is more balanced and the efficiency is improved. The implication of this fact is the use of $k = 1$ in the classification, as Wilson and Martinez point out.

In the next subsection, we will describe the algorithms used in this study but not the EAs (which will be described in Section 3).

2.1. PS methods

Algorithms for PS may be classified according to the heuristic followed in the selection. We have selected the most representative and well-known methods belonging to the non-evolutionary family and the algorithms that offer the best performance for the PS problem.

- Enn [21]. Edited NN edits out noisy instances, as well as close border cases, leaving smoother decision boundaries. It also retains internal points. It works by editing out those instances in which class does not agree with the majority of classes of its k NNs.
- Allknn [22]. Allknn is an extension of Enn. The algorithm, for $i = 1$ to k flags as bad any instance not correctly classified by its i NNs. When the loop is completed k times, it removes the instances flagged as bad.
- Pop [23]. This algorithm consists of eliminating the samples that are not within the limits of the decision boundaries. This means that its behaviour is in opposite direction from that of Enn and Allknn.
- Rnn [24]. The reduced NN rule searches a minimal and consistent subset which correctly classifies all the learning instances.
- Drop3 [6]. An associate of $\overrightarrow{x_p}$ is that sample $\overrightarrow{x_i}$ which has $\overrightarrow{x_p}$ as NN. This method removes $\overrightarrow{x_p}$ if at least as many of its associates in TR would be classified correctly without $\overrightarrow{x_p}$. Prior to this process, it applies a noise reduction filter (Enn).
- Ib3 [25]. It introduces the *acceptable* concept, based on the statistical confidence of inserting a certain instance in the subset, to carry out the selection.