Neurocomputing 125 (2014) 102–118

Contents lists available at ScienceDirect

Neurocomputing

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

Multiobjective genetic programming for maximizing ROC performance

Pu Wang [a], Ke Tang [a,*], Thomas Weise [a], E.P.K. Tsang [b], Xin Yao [c]

[a] Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, Anhui 230027, China
[b] Department of Computer Science, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK
[c] Center of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, The University of Birmingham Edgbaston, Birmingham B15 2TT, UK

ARTICLE INFO

Available online 1 March 2013

Keywords:
Classification
ROC analysis
AUC
ROCCH
Genetic programming
Evolutionary multiobjective algorithm
Memetic algorithm
Decision tree

ABSTRACT

In binary classification problems, receiver operating characteristic (ROC) graphs are commonly used for visualizing, organizing and selecting classifiers based on their performances. An important issue in the ROC literature is to obtain the ROC convex hull (ROCCH) that covers potentially optima for a given set of classifiers [1]. Maximizing the ROCCH means to maximize the true positive rate (*tpr*) and minimize the false positive rate (*fpr*) for every classifier in ROC space, while *tpr* and *fpr* are conflicting with each other. In this paper, we propose multiobjective genetic programming (MOGP) to obtain a group of nondominated classifiers, with which the maximum ROCCH can be achieved. Four different multi-objective frameworks, including Nondominated Sorting Genetic Algorithm II (NSGA-II), Multiobjective Evolutionary Algorithms Based on Decomposition (MOEA/D), Multiobjective selection based on dominated hypervolume (SMS-EMOA), and Approximation-Guided Evolutionary Multi-Objective (AG-EMOA) are adopted into GP, because all of them are successfully applied into many problems and have their own characters. To improve the performance of each individual in GP, we further propose a memetic approach into GP by defining two local search strategies specifically designed for classification problems. Experimental results based on 27 well-known UCI data sets show that MOGP performs significantly better than single objective algorithms such as FGP, GGP, EGP, and MGP, and other traditional machine learning algorithms such as C4.5, Naive Bayes, and PRIE. The experiments also demonstrate the efficacy of the local search operator in the MOGP framework.

1. Introduction

Classification [2] is one of the most important areas in machine learning. Here, the goal is to find assignments of classes to un-classified and unseen instances (data samples) based on information previously learned. In the most common case, referred to as binary classification, there are two classes or categories and all instances in a data set belong to one of them. Solving classification problems basically means to design good classifier(s) which make right assignments as often as possible.

One open question is how to measure the performance of a classifier. If classifiers are synthesized with optimization algorithms, the choice of the performance measure will have tremendous impact on the results that we will obtain. Simple classification accuracy, though being used as the performance metric for a long time, is actually not a good choice [3]. The receiver operating characteristics, or ROC for short, has been claimed as a generally useful performance visualizing method because its properties are not sensitive to skewed class distributions or unequal misclassification costs, two characteristics which are known to have a negative impact on the utility of the accuracy measure.

The ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance [1]. It has been widely used in signal detection [4], medical decision making [5], and other fields over the course of the past 40 years. In recent years, because of the ever-increasing use of ROC graphs in the machine learning community, the ROC analysis became a central technique for tackling classification problems. The ROC curve, an important topic in ROC analysis, is obtained by varying discriminative thresholds over the output of a classifier [1]. The area under the ROC curve (AUC) is accepted as a fair indicator to measure the classifier performance for binary classification, since it is invariant to operating conditions such as different misclassification costs and skewed class distributions [6]. ROCCH, another important topic in classification problems, represents the convex hull of a set of points (hard classifiers) obtained from several

* Corresponding author.
E-mail addresses: wuyou308@mail.ustc.edu.cn (P. Wang), ketang@ustc.edu.cn (K. Tang).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.06.054

curves (i.e., soft classifiers) [7]. A classifier is potentially optimal if and only if it touches the ROCCH. Otherwise, we can always find a better classifier. It is possible to get a potentially optimal classifier in ROCCH even if the data sets have skewed class distributions or misclassification costs. Actually, we can consider the ROC curve as a special ROCCH when there is only a single soft classifier. This means that ROCCH could work more powerfully than a plain ROC curve. Consequently, we mainly consider the ROCCH in this paper and we will focus on searching a group of classifiers not only to maximize the ROCCH performance but also try to maximize the AUC of a single soft classifier in binary classification problems.

In this paper, we utilize GP combined with multiobjective techniques to approximate the optimal ROCCH. This work empirically investigates multiobjective genetic programming (MOGP) with four different frameworks on binary classification problems. We show that local search strategies can play a key role in GP for classification problems and that special local search operators can improve the performance.

This paper is organized as follows: Section 2 outlines the related work and in Section 3, we introduce the background and basic algorithms used in our research. Section 4 will describe our framework to classification problems and presents local search operators working in GP. Experiments are studied in Section 5 where four research questions are answered. Section 6 provides the conclusion and a discussions on the important aspects and future perspectives of this work.

2. Related work

2.1. ROCCH in classification

The roots of ROCCH maximization problems can be traced back to [7]. In that work, *iso-performance lines*[1] are translated by operating conditions of classifiers and used to identify a portion of the ROCCH, by which we can choose suitable classifiers for data sets with different skewed class distribution or misclassification costs. In [8], a combination of rule sets to produce instance scores indicating the likelihood that an instance belongs to a given class is described.

Flach et al. [9] investigated a method to detect and repair concavities in ROC curves. The basic idea here is that if a point lies below the line spanned by two other points in ROC, then it can be mirrored to a better point which could perform well beyond the original ROC curve. This can be used to expand the ROCCH. Prati [10] introduced a rule selection algorithm based on ROC analysis to find minimal rule sets with compatible AUC values. Here, selection is based on whether a rule can improve the current ROCCH.

In [11], a method which takes Neyman–Pearson lemma [12] as the theoretical basis for finding the optimal combination of classifiers to maximize the ROCCH is given. Fawcett [13] presents a method for learning rules directly from ROC space. This method utilizes the geometrical properties of the ROC to generate new rules to maximize the ROC performance. Essentially, all above works are searching a rule sets to maximize ROCCH.

2.2. Genetic programming for classification

Genetic programming (GP) [14] is a branch of evolutionary algorithms (EAs). Standard GP has a tree-like representation which can be generated by modular, grammatical, and developmental methods [15]. Tree-based classifiers have a long tradition in machine learning. They are considered to be more explicit, intuitive, and interpretable than, e.g., neural networks. GP therefore has widely been used for solving classification problems [16,17].

An example of using GP to evolve regression rules for a data set with intertwined spirals pattern is already given in Koza's 1992 book [14]. Another early work [18] used in image recognition dates 20 years back. In the area of data mining, GP has been applied most successful in two particular fields: one is classification for data sets with different misclassification costs, as GP is suitable for cost-sensitive learning. [19], e.g., focused on financial forecasting problems by consolidating two types of misclassification errors into a single objective function. GP involving cost-sensitive learning has furthermore been adopted in filtering junk E-mail [20].

The second field is classification of imbalanced data sets, i.e., data sets where one class occurs much more often than the other—one of the areas where the accuracy metric may become useless. Ref. [21] adopts GP to bankruptcy prediction, a prime example for this issue as there are significantly more solvent firms than defaulting ones. Patterson [22] gave a new fitness function for GP applied on highly imbalanced database. Moreover, Bhowan et al. [23] proposed a multiobjective genetic programming approach to evolving accurate and diverse ensembles of genetic program classifiers with good performance on both the minority and majority classes.

Many technologies have been combined with GP to improve the classification performance in these two fields, ranging from ensemble learning over multiobjective methods to local search strategies. As both imbalanced problems and different misclassification costs can be included in the ROCCH [7], this work will focus on GP for maximizing the ROCCH. It should further be mentioned that there is a strong analogy of ROCCH and the Pareto front in multiobjective optimization [24].

In this paper, we use multiobjective GP (MOGP) to approximate the optimal ROCCH. We empirically investigate MOGP with four different frameworks on binary classification problems. Additionally, we show that local search strategies can play a key role in GP for classification problems as special local search operators can carefully be designed to improve the performance.

3. ROCCH, classification, and multiobjective optimization

3.1. Overview of ROCCH in classification problems

3.1.1. ROC Graph and ROCCH

In binary classification problems, each instance I in the data set is marked a certain label from the set $\{\boldsymbol{p}, \boldsymbol{n}\}$ of positive and negative class labels. A classifier is a mapping from instances to predicted classes, and *accuracy* is the most commonly used evaluation measure. However, its disadvantage are known for a long time [25]. Generally, accuracy is not a suitable metric for cost sensitive and skewed class distribution classification problems. To overcome the weakness of accuracy, ROC analysis has been introduced in machine learning. Ref. [26] demonstrated the value of ROC curves in evaluating and comparing algorithms. An important tool of ROC analysis is the ROC graph which is used to visualize the performance of classifiers. The X-axis and Y-axis of ROC graphs display the true positive rate(tpr) and false positive rate (fpr). The performance of a *hard* or *discrete* classifier on a data set can be mapped in a single point in this graph. The upper left point (0,1) represents a perfect classifier which predicts positive (or Yes) to all positive instances and negative (or No) to all negative instances. The points in lower right area are conservative classifiers which produce more negative labels than positive

[1] All classifiers corresponding to the points on one line have the same expected costs.