Genetic Programming with Boosting for Ambiguities in Regression Problems

Grégory Paris, Denis Robilliard, and Cyril Fonlupt
Université du Littoral-Côte d’Opale, LIL
BP 719, 62228 Calais Cedex, France
paris@lil.univ-littoral.fr
phone: +33-321 465 667

Abstract. Facing ambiguities in regression problems is a challenge. There exists many powerful evolutionary schemes to deal with regression, however, these techniques do not usually take into account ambiguities (i.e. the existence of 2 or more solutions for some or all points in the domain). Nonetheless ambiguities are present in some real world inverse problems, and it is interesting in such cases to provide the user with a choice of possible solutions. We propose in this article an approach based on boosted genetic programming in order to propose several solutions when ambiguities are detected.

1 Introduction

In a regression problem, the goal is to find a function that closely matches a finite set of points taken from an unknown signal or an unknown function.

Genetic Programming (GP) has been successfully applied to tackle difficult regression problems, but most of these studies make the assumption that only one function is needed to fit the sample points.

Many real world inverse problems can be seen as regression problems. Some of these are of a very important scientific or economic interest, ranging from physics, combinatorics, mathematical analysis, medicine, ecology... However these problems are often mathematically “ill-posed”, meaning that the existence and the uniqueness of a solution cannot be guaranteed. For example, seeking to retrieve phytoplankton chlorophyll-a concentration in coastal waters from remote sensing spectrometer data, the so-called ocean color problem, is known to be an ambiguous problem (see [1]).

It is also possible that we are given a sample set made of points taken from several signals. In such case, we do not know how many signals there are and it is not easy to determine from which signal a given point comes.

Let us look at a simple example to illustrate the notion of ambiguity: suppose we try to invert a process modeled by \( f(x) = |x| \), then what is the inverse of 3? Two values are possible, 3 and \(-3\). So, one solution is not sufficient. Our aim in this article is to propose a way to overcome ambiguities in such cases and to be able to propose one, two or more solutions to fit the data. We will
use the boosting scheme in conjunction with GP to search and propose multiple solutions on a set of ambiguous test regression problems.

Next section defines the boosting scheme and introduces our GP method targeted for regression problem: GPboost. We show then how we use it to approximate several models. In Sect. 3, we explain the dendrograms principles, hierarchical classification trees which aim at clustering some set of values. In Sect. 4, we show how to deal with the multiple hypotheses provided by the GP-boost scheme, and then we provide results on an ambiguous regression problems in Sect. 5. Finally, we draw some conclusions and we provide ideas for further research.

2 Genetic Programming and Boosting

2.1 Principles

Boosting appeared at the beginning of the 90’s, proposed by Schapire [2] and Freund [3]. It aims at improving already known methods from the machine learning field (cf [4]), methods which need a learning set or fitness set to provide a function which will best fit training points.

The first principle of boosting is to include a distribution on the learning set. Each point is weighted and this weight is taken into account in the learning algorithm. This distribution is changed according to the difficulty the algorithm has to learn points of the learning set.

The second principle is that the boosting algorithm offers several hypothesis. These hypothesis will finally be combined to give a final result.

To sum up, we first have an uniform distribution on the learning set. The boosted algorithm gives a function denoted $f_1$. Weights of the badly learned points are increased while the others are decreased. The algorithm is run a second time and provides $f_2$. This scheme is looped over, until we get $T$ functions $f_t$ ($T$ is a parameter of the boosting method).

To determine the output value of a given point, values proposed by each function $f_t$ are combined by a vote or another method.

Part 2.2 introduces an application of boosting to genetic programming [5] (for more details about boosting, see [6]).

There is a theoretical proof showing that error on the learning set of the final hypothesis is better than an “un-boosted” version of the algorithm, when the base algorithm is a weak learner, according to the PAC model [7]. Boosting may also be seen as a particular case within the family of adaptive re-weighting and combining algorithms, also called arcing algorithms, see [8], [9] and also [10].

2.2 The GPboost Algorithm

Based on the work of Drucker [11], Iba proposed in [12] a boosting version of GP. Our method GPboost [5] is dedicated to regression problems, and it differs from Iba’s scheme by retaining the precision of weights for every training cases,