Chapter IX

Extracting Knowledge from Databases and ANNs with Genetic Programming: Iris Flower Classification Problem

Daniel Rivero, University of A Coruña, Spain
Juan R. Rabuñal, University of A Coruña, Spain
Julián Dorado, University of A Coruña, Spain
Alejandro Pazos, University of A Coruña, Spain
Nieves Pedreira, University of A Coruña, Spain

ABSTRACT

In this chapter, we present an application of Genetic Programming (GP) in the field of data mining and extraction of Artificial Neural Networks (ANN) rules. To do this, we will use its syntactic properties to obtain high level expressions that represent knowledge. These expressions will have different types as there is the need at each moment: we will obtain different expressions like IF-THEN-ELSE rules, mathematical relations between variables or boolean expressions. In this chapter, we will not only apply GP to solve the problem, but we will try different modifications and
different ways to apply it to solve the problem. We will show how making a data pre-processing we can obtain better results than using the original values. That is, by adding a little knowledge from the problem we can improve the performance of GP.

INTRODUCTION

In the world of Artificial Intelligence (AI), the extraction of knowledge has been a very useful tool for many different purposes, and it has been tried with many different techniques. Here, we will show how we can use Genetic Programming (GP) to solve a classification problem from a database, and we will show how we can adapt this tool in two different ways: to improve its performance and to make it possible to detect errors. Results show that the technique developed in this chapter opens a new area for research in the field, extracting knowledge from more complicated structures such as Artificial Neural Networks (ANNs).

BACKGROUND

Genetic Programming and Artificial Neural Networks

Genetic Programming (GP) (Koza, 1992) is an evolutionary method used to create computer programs that represent approximate or exact solutions to a problem. This technique allows for the finding of programs with the shape of a tree, and, in its most common application, those programs will be mathematical expressions combining mathematical operators, input variables, constants, decision rules, relational operators, etc.

All of these possible operators must be specified before starting the search. So, with them, GP must be able to build trees with the objective of finding the desired expression which models the relation between the input variables and the desired output. This set of operators is divided into two groups: the terminal set contains the operators which cannot accept parameters, like variables or constants; and the function set, which contains the operators, such as add or subtract, which need parameters. Once the terminal and non-terminal operators are specified, it is possible to establish types. Each node will have a type, and the construction of child expressions needs to follow the rules of the nodal type (Montana, 1995).

GP creates automatic program generation by means of a process based on the evolution theory of (Darwin, 1864), in which, after subsequent generations,
Latent Topic Model for Indexing Arabic Documents
[www.igi-global.com/article/latent-topic-model-for-indexing-arabic-documents/113331?camid=4v1a](www.igi-global.com/article/latent-topic-model-for-indexing-arabic-documents/113331?camid=4v1a)