Chapter VII
Genetic Learning:
Initialization and Representation Issues

Ivan Bruha
McMaster University, Canada

ABSTRACT

This chapter discusses the incorporation of genetic algorithms into machine learning. It does not present the principles of genetic algorithms (because it has been already done by many more or less large monographs) but rather focuses particularly on some important issues and enhancements of genetic algorithms design is faced by, namely: (a) the ways of initialization of a population for a genetic algorithm, (b) representation of chromosomes (individuals) in genetic algorithms (because it plays an influential role in the entire processing), and (c) discretization and fuzzification of numerical attributes for genetic algorithms (since they are not in its genuine form able to process these attributes). Furthermore, this chapter surveys new trends of dealing with the variable-length chromosomes and other issues related to the genetic learners. It concludes by discussing some directions of further improvement of the genetic learners, namely two topologies with the ‘meta’ level.

INTRODUCTION

A machine learning algorithm induces descriptions of concepts (classes) of a given problem. The induction of concepts from databases consists of searching usually a huge space of possible concept descriptions. There exist several paradigms how to control this search, e.g., various statistical methods, logical/symbolic algorithms, neural nets, and the like. This chapter presents the paradigm of genetic algorithms for such a search. There are several projects of incorporating genetic algorithms into machine learning (ML); we would like to present our contribution to this field, called genetic learning.

Genetic algorithms (GAs) represent a relatively new efficient optimization technique for inducing new concepts. They emulate biological evolution and are utilized in optimization processes, similarly to simulated annealing, rough sets, and
neural nets. The optimization is performed by processing a population (a set) of individuals. Each individual (chromosome) is usually divided into several substructures, called genes. An individual as a unit is in most applications represented by a binary string. For representation purposes, we distinguish fixed- and variable-length strings.

A designer of a GA has to provide an evaluation function, called fitness; it evaluates any individual (chromosome). Also the formulas/procedures for so-called genetic operators must be provided by the user (see their definitions below).

GAs have become an attractive alternative to classical search algorithms for exploring large spaces of hypotheses (models, concept descriptions). They are two important characteristics of the search performed by a GA: the search is usually global and is parallel in nature since a GA processes not just a single individual (chromosome) but a large set (population) of individuals.

There exist both theoretical and experimental results showing that GAs can rapidly locate individuals with high fitness rating (even for very huge search spaces) by using a relatively small population of 50-100 individuals (chromosomes). This chapter, of course, does not present the principles of genetic algorithms; detailed and exhausted discussion can be found, e.g. in (Holland, 1975; De Jong, 1980; Grefenstette, 1986).

**BACKGROUND**

Research projects of incorporating (merging) genetic algorithms into the field of learning algorithms inducing concept descriptions can be divided into two categories.

In the first category, called external one, the actual concept description (model) is derived (induced) by a traditional machine learning algorithm and a GA serves as an evaluation or parameter-optimizing device. Behaviour of any performance system (not only a ML algorithm) exhibits several parameters that affect its behaviour; a suitable strategy for changing these parameters’ values improves performance of the system. How can a GA serve in this issue?

In one application, a GA plays a role of an evaluation device, and the performance system itself modifies its internal structure by GA’s evaluation characteristics.

In a more sophisticated application, a GA itself searches for a combination of parameters that improve the performance of a machine learning (performance) system. Here, the parameters are viewed as genes, and the genetic material of individuals (chromosomes) as a fixed-length string of genes.

Another application of GAs (De Jong, 1988): in some tasks, a simple vector of parameters is not adequate, but rather a more complex data structure of parameters is involved in controlling the behaviour of the performance system. Such a data structure can be ‘linearized’ into a binary string representation; however, the traditional genetic operators (namely, crossover and mutation) could produce illegal data structures (not having the sense in a given task). Therefore, representation-sensitive operators are to be designed and applied.

To illustrate this fashion, we introduce a few existing systems. For instance, (Kelly & Davis, 1991) describes a hybrid learning system that utilizes the k-nearest-neighbour classification algorithm in a standard way and a genetic algorithm that learns weights associated with the attributes. Similar hybrid system may be found in (Turney, 1995): C4.5 generates decision trees and the genetic algorithm evolves a population of bias parameters that control C4.5’s performance.

In the other fashion (called genuine one), a genetic algorithm itself generates concept descriptions (models), usually in a form of decision trees or sets of decision rules. Regal (Giordana & Saitta, 1993; Neri & Saitta, 1996) is a large system that utilizes genetic search for deriving first-order-logic concept descriptions. Samuel (Grefenstette et al., 1990), another genetic algorithm generating
Related Content

Creation and Integration of Reference Ontologies for Efficient LOD Management
[www.igi-global.com/chapter/creation-integration-reference-ontologies-efficient/63902?camid=4v1a](www.igi-global.com/chapter/creation-integration-reference-ontologies-efficient/63902?camid=4v1a)

Incorporating Correlations among Gene Ontology Terms into Predicting Protein Functions
Pingzhao Hu, Hui Jiang and Andrew Emili (2011). *Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances* (pp. 154-173).
[www.igi-global.com/chapter/incorporating-correlations-among-gene-ontology/53885?camid=4v1a](www.igi-global.com/chapter/incorporating-correlations-among-gene-ontology/53885?camid=4v1a)

How Does the Hierarchical Management System Influence the Climate of Creativity in Chinese University Laboratories?

Automated Learning of Social Ontologies
Konstantinos Kotis and Andreas Papasalouros (2011). *Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances* (pp. 227-246).
[www.igi-global.com/chapter/automated-learning-social-ontologies/53889?camid=4v1a](www.igi-global.com/chapter/automated-learning-social-ontologies/53889?camid=4v1a)