Knowledge Discovery Using Heuristics

Alina Lazar
Youngstown State University, USA

INTRODUCTION

Uninformed or blind search, which processes and evaluates all nodes of a search space in the worst case, is not realistic for extracting knowledge from large data sets because of time constraints that are closely related to the dimension of the data. Generally, the search space increases exponentially with problem size, thereby limiting the size of problems that can realistically be solved using exact techniques such as exhaustive search. An alternative solution is represented by heuristic techniques, which can provide much help in areas where classical search methods failed.

The word "heuristic" comes from Greek and means "to know," "to find," "to discover" or "to guide an investigation". Specifically, "Heuristics are techniques which seek good (near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution is" (Russell & Norvig, 1995).

Heuristic refers to any techniques that improve the average-case performance on a problem-solving task but do not necessarily improve the worst-case performance. Heuristic techniques search the problem space "intelligently" using knowledge of previously tried solutions to guide the search into fruitful areas of the search space. Often, search spaces are so large that only heuristic search can produce a solution in reasonable time. These techniques improve the efficiency of a search process, sometimes by sacrificing the completeness or the optimality of the solution. Heuristics are estimates of the distance remaining to the goal, estimates computed based on the domain knowledge.

The domain knowledge provides help to heuristics in guiding the search and can be represented in a variety of knowledge formats. These formats include patterns, networks, trees, graphs, version spaces, rule sets, equations, and contingency tables. With regard to heuristics, there are a number of generic approaches such as greedy, A* search, tabu search, simulating annealing, and population-based heuristics. The heuristic methods can be applied to a wide class of problems in optimization, classification, statistics, recognition, planning and design.

Of special interest is the integration of heuristic search principles with the dynamic processes in which data become available in successive stages, or where data and inputs are subject to uncertainties or with large-scale data sets. The integration is a vehicle to generate data driven hypotheses.

The kind of knowledge produced, and the heuristic search algorithm selected, will reflect the nature of the data analysis task. The hypotheses are being represented as sets of decision rules and the extracted rules will be represented in terms of rough sets. Rough sets were selected because of the nature of our data sets.

From a mathematical point of view, the problems can be formulated in terms of the well-known, minimal set cover problem, which is a combinatorial optimization problem.

Traditional methods for combinatorial optimization problems are not appropriate here for several reasons. These methods are NP-hard in the worst case and would be costly to use given the size of the data sets. Also, since large data sets are dynamical in nature, adding new data would require running the traditional combinatorial approach again.

The techniques used to solve these difficult optimization problems have slowly evolved from constructive methods, like uniformed search, to local search techniques and to population-based algorithms. Our research goal was to use blend population-based algorithms with methods dealing with uncertainty in order to induce rules from large data sets.

BACKGROUND

Population-based heuristic methods are iterative solution techniques that handle a population of individuals who are evolving according to a given search strategy. At each iteration, periods of self-adaptation (mutations) alternate with periods of cooperation (crossover), and periods of competition (selection). The population-based heuristic search (Conrad, 1978) is dependent on the following components: the knowledge representation for the specific problem to solve and the search strategy or the evolution process. The adaptability of an individual represents its ability to survive in an uncertain environment. Artificial intelligence researchers have explored different ways to represent uncertainty (Russell & Norvig, 1995): belief networks, default reasoning, Dempster-Shafer theory, fuzzy sets theory, rough sets theory.

The learning task will require a representation that explicitly deals with uncertainty. The evolutionary learning methods that are employed must be able to work with such a representation. In this article we look first at basic ways to represent uncertainty in developing rules. And, then we will investigate how that uncertain knowledge can be used to direct evolutionary search and learning.

Uncertainty, as well as evolution, is a part of nature. When humans describe complex environments, they use linguistic descriptors of real-world circumstances, which are often not precise, but rather “fuzzy”. The theory of fuzzy sets (Zadeh, 1965) provides an effective method of describing the behavior of a system that is too complex to be handled with the classical precise mathematical analysis.

The theory of rough sets (Pawlak, 1991) emerged as another mathematical approach for dealing with uncertainty that arises from inexact, noisy or incomplete information. Fuzzy sets theory assumes that the membership of the objects in some set is defined as a degree ranging over the interval [0,1]. Rough sets theory focuses on the ambiguity caused by the limited distinction between objects in a given domain.

Fuzzy sets have been employed to represent rules generated by evolutionary learning systems. Using fuzzy concepts, Hu and Tzeng (2003) tried to overcome the limitations of the conventional rule-based classifier system (Holland, 1975) when representing continuous variables.

Likewise, fuzzy functions have been used to describe and update knowledge in cultural algorithms. First, Reynolds (1994) employed a fuzzy acceptance and influence function in the solution of real-valued constrained optimization problems. Following the same idea, Zhu designed a fully fuzzy cultural algorithm (Reynolds & Zhu, 2001) that included a fuzzy knowledge representation scheme in order to deal with the continuous variables in the belief space, as well as a fuzzy acceptance and influence function. All these approaches were tested on real-values function optimization problems. Jin (2000) used a fuzzy knowledge representation for normative knowledge in the belief space of cultural algorithms to solve the real-valued constrained function optimization.

The design of a fuzzy representation system is not an easy job, because the membership functions should be carefully chosen, and the procedures that use these functions should be specified precisely. The problem is to optimize the fuzzy membership functions for a problem and to find optimum plans related to the fuzzy performance measures. It is a natural approach to use heuristics (i.e., evolutionary algorithms) to solve this task.

Another approach to represent uncertainty is with rough sets. Rough sets are based on equivalence relations and set approximations, and the algorithms for computing rough set properties are combinatorial in nature. Wroblewski (1995) implemented a genetic algorithm for computing reducts, based on permutation code as well as a “greedy” algorithm. Another approach for building reducts is described by Vinterbo (2000) and it is based on the set cover problem, in particular on finding minimal hitting sets using a classical genetic algorithm. Finding a minimal set of decision rules or a satisfactory set is an NP-complete problem. Agotnes (1999) used a genetic algorithm to build an optimal set of decision rules, where the fitness function was based on the quality of each rule. Decision rules extracted via rough set theory could be used to represent hard coded information from neural networks (Lazar & Sethi, 1999).

Evolution can be defined in one word, “adaptation” in an uncertain environment. Nature has a robust way of dealing with the adaptation of organisms to all kinds of changes and to evolve successful organisms. According to the principles of natural selection, the organisms that have a good performance in a given environment survive and reproduce, whereas the others die off. After reproduction, a new generation of offspring derived from the members of the previous generation is formed. The selection of parents from these offspring is often based upon fitness. Changes in the environment will affect the population of organisms through the random mutations. Mayr said, “Evolution is a dynamic, two-step process of random variation and selection” (Fogel, 1995). Using examples from natural systems and theories of adaptive behavior, researchers have been trying to build heuristic evolutionary learning systems.

Evolutionary algorithms are heuristic optimization methods inspired from natural evolution processes. Currently there are three basic population-only mechanisms that model evolution: genetic algorithms, evolution strategies and evolutionary programming. Each of the methods models the evolution of a population of individuals at a different scale and applies election and reproduction operators to find an individual that is fit with regard of the fitness function. The genetic algorithm models evolution at the gene scale, but evolutionary strategies and evolutionary programming model evolution at the species level.

The cultural algorithms (Reynolds, 1994) approach adds another level to the evolutionary process inspired from the human societies and cultural evolution. It adds to the population space, belief space. The belief space will be a collection of symbolic knowledge that will be used to guide the evolution of the population.