A Hybrid Heuristic Model for Classification Rule Discovery

by
Gokbora Uran

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing

at

School of Computer Science and Information Systems

Pace University

May 2005

UMI Number: 3172518

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

A Hybrid Heuristic Model for Classification Rule Discovery

by
Gokbora Uran

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing

May 2005

This dissertation studies hybrid heuristic models in the context of classification rule discovery. Nature inspired search algorithms such as Genetic Algorithms, Ant Colonies and Particle Swarm Optimization have been previously studied on data mining tasks, in particular, classification rule discovery. We extended this work by applying a hybrid model which combines GA, PSO and hill climbers, in same type of classification tasks. Such models have been tested and proved to be performing better than individual search algorithms, in various combinatorial optimization problems. Our research focused on studying the same kind of performance enhancements in classification rule discovery tasks.

As a part of this dissertation, we developed a model for a hybrid heuristic based classifier and implemented different variations of it in Java. These algorithms have been benchmarked against the well known decision tree induction algorithm C4.5. Results have been compared in terms of prediction accuracy, speed and comprehensibility. Our results showed that, heuristic based classifiers can compete with C4.5 in terms of prediction accuracy on certain data sets and outperform C4.5 in general in terms of comprehensibility. C4.5 always outperformed heuristic based classifiers in terms of speed due to relative inefficiency inherent in heuristic based classification models. We also showed that hybridization of heuristics could bring improvements in terms of execution speed in comparison to plain heuristic based classifiers.

Acknowledgements

I would like to thank my dissertation advisor, Dr. Michael Gargano, for his support and guidance.

And I would like to thank my family, for their patience and tolerance.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Classification is a classical learning problem extensively studied by the researchers of different areas such as statistics and machine learning. Algorithms based on inductive learning, Bayesian probability, neural networks and other methods from the fields of artificial intelligence and statistics, have been developed for this learning task. The type of learning classification rule discovery usually employed is known as Supervised Concept Learning due to the fact that learning process uses examples of the already known concepts.

The objective of Supervised Concept Learning, an inductive learning method, is to construct a function $f(X)=Y$, where X is a vector of features and Y is the concept to be learned. Learning is achieved via induction from a given set of (X, Y) pairs. Translated into general terms of classification rule discovery (as well as the terms used in this dissertation), $f(X)=Y$ is a model for prediction and can be represented in many different forms such as decision trees or classification rules. We'll call X an instance, and features in X, attributes. Y is the class to be predicted. If an attribute has a finite number of possible values, it's a discrete or nominal attribute, otherwise it is a numeric or continuous attribute. The set of (X,Y) pairs that are used for the induction process is called a training data set and consists of instances with known classes. Each instance can be considered as a point in an n-dimensional space of hypotheses, where n is the

number of attributes. The hypotheses are possible classification rules, and the learning process tries to find those hypotheses that are consistent with the highest number of instances within the training data set. So, rule learning is a process of search through the space of possible hypotheses, and the training data set is used to direct this search.

Classification Rule Discovery has been the most studied task of Data Mining. Data Mining has emerged as an interdisciplinary field of machine learning, statistics and database. Data mining is the core of Knowledge Discovery in Databases [5], which is a collection of all the processes that are used to extract useful information from data. As the volume of data that is accumulated by organizations has grown exponentially in recent years, the desire to use this data for extraction of useful patterns and prediction made the Knowledge Discovery in Databases a field with a lot of practical applications in such areas as medical diagnosis, credit rating, marketing and fraud detection. This wide range of applicability requires robust and scalable data mining algorithms that can process data sets from very large databases consisting of different data types at reasonable speeds. These algorithms require generality and flexibility to be used in variety of domains. In order to find better methods that can satisfy these requirements, researchers have been trying different approaches. For example, evolutionary algorithms are a set of biology inspired heuristic methods that have been successfully used to solve different types of search problems. They are also used in this thesis, as one of the heuristics constructing our classification model.

Evolutionary Algorithms are a subset of a larger set of heuristic methods. Heuristics are approximate methods attempting to find “very good” solutions in very large search spaces in reasonable time. These are in contrast to exact methods that find optimal

solution. The development of new heuristics has been an extensively studied area due to the practical importance of the types of problems they can solve, especially in the area of Combinatorial Optimization. While a lot of problem specific methods have been developed over the years, researchers have also developed general purpose, problem independent search strategies for more efficient and effective exploration of the solution search spaces. These *metaheuristics*, as they are generally called, provide different strategies to escape local optimums in the search space by employing different intensification and diversification techniques. Diversification refers to exploration of the different regions of the search space, whereas intensification limits the search area based on accumulated search experience. The balance between these two is the key to the success of a metaheuristic. Over the years, researchers have recognized the relative strengths and weaknesses of these different methods, and developed hybrids of them to be able utilize their strengths in different regions of the search space and/or in different phases of the search process.

This dissertation studies these types of hybrid heuristic models in the domain of the classification rule discovery problem. The knowledge that has been produced by previous research in heuristic based classifiers and as well as previous research in hybridization of different heuristics in the domain of combinatorial optimization, have both been applied to create new models for heuristic based classifiers. The primary objective of this work has been to investigate the contribution of hybridization in the development of classifiers that can perform better in terms of prediction accuracy, speed and comprehensibility. This investigation has concentrated on development of hybrid heuristic based classifiers and testing them against different well known classifiers.

The inspiring source of our hybrid model has been introduced by Løvbjerg [15] as the "Life Cycle Model". It is a hybrid of Genetic Algorithms, Particle Swarm Optimization and Stochastic Hill Climbers. Genetic Algorithms, based on Darwinian concepts, have been studied by many different researchers in the context of classification rule discovery. Freitas, in [8], provides a comprehensive survey of this work. Particle Swarm optimization, based on colonies of social agents, a relatively more recent population based stochastic method, has been successfully applied in many different problem areas. [26], by Sousa, Neves and Silva, is the only study of Particle Swarm Optimization based classifier that we are aware of. Hill climbers are among the oldest heuristic methods. They have been studied extensively, but we haven't found any in the context of classification rule discovery. Hybrid heuristic models have been studied by many researchers. Talbi provides a taxonomy and survey in [28], but we are not aware of any application of this to classification rule discovery.

The rest of this dissertation is organized as follows: Chapter 2 provides a background on metaheuristics, particularly on Hill Climbers, Particle Swarm Optimization and Genetic Algorithms, as they are a significant part of the model developed in this thesis. Bringing performance improvements to the heuristic based classifiers by hybridization of heuristics is one of the critical objectives of the model we developed. Therefore, in this chapter we focus on major metaheuristic methods and their hybrid models.

In Chapter 3, Classification Rule Discovery is introduced. The first section includes details of C4.5, a landmark decision tree induction algorithm developed by Quinlan [22]. C4.5 has been used for the benchmarking of algorithms developed as a part of this dissertation. Heuristic based classifiers are discussed in the second section of Chapter 3.

Details of the experimentation and our heuristic model are provided in Chapter 4. In this chapter we first provide details of our experiment environment, then, based on the materials provided in second and third chapters, the details of our classifier model. Finally a summary of results from our test runs of the model is provided.

In Chapter 5 we conclude with a discussion of our results and our work and suggested future work.

Chapter 2

Metaheuristics

Many real-world problems are complex problems with a very high number of possible solutions. The size of such a search space prohibits an exhaustive search for the best solution. Computational Complexity, the area of theoretical computer science which analyzes resources needed to solve computational problems, classifies problems with varying computational complexity [7]. Intuitively, a problem is considered intractable if the amount of time required to solve it increases at a faster-than-polynomial rate as the size of the problem increases. These types of problems are called "NP-Complete" in computational complexity theory. This notion is introduced here only to pave the way for a discussion of the need for heuristics. A well known example of an NP-complete problem, the traveling salesperson problem, will be mentioned below. There is a great deal of theory behind the concept of "NP-completeness". It will not be discussed in this thesis. More information on the theory can be found, for example, in [7].

Traveling salesperson problem requires finding the shortest closed path (or tour) through a set of nodes, or cities, without passing through any node more than once and finally ending up at the original starting point. With each additional city the number of additional tours to consider grows more than exponentially; with N nodes there are *N!* tours possible. In a symmetrical traveling salesperson problem, in which distance between two cities is same both directions, the search space is *(N-1)! / 2* [18]. So when

the number of cities is 10, there are 181440 possible solutions. When the number of cities increases to 20, the size of the search space increases to approximately 10,000,000,000,000,000 (6.0823 x 10^{16} to be exact) possible tours. Traveling salesperson problem is a NP-complete problem with no known efficient solution that reduces the search time needed for a brute force search.

Since searching for an exact solution using brute force for these types of problems will lead to computation times too high for practical purposes, approximate methods or heuristics are used. When these methods are used, there is a practical trade off made between the guarantee of a best solution and finding a good solution in a significantly reduced time.

2.1 Local Search

Local Search is one of the basic heuristic methods. It is the simple iterative process of evaluating a randomly chosen potential solution in the search space, moving to another solution within the neighborhood of the original solution, and comparing the new solution to the original solution. If the new solution is better than the original solution, the search continues with neighbors of the new solution, otherwise the next potential solution in the neighborhood of the original solution is tested. This iteration continues until no better solution is found in the neighborhood. As opposed to an exhaustive search which searches entire search space, local search is regional. Local Search searches only a small region of the search space and this region is determined based on the starting point and neighborhood structure. Neighborhood structure is the key to effectiveness and efficiency of the local search algorithms. As the neighborhood size gets larger, the

efficiency of the algorithm is reduced, however, the chance of finding a better solution is increased.

Hill Climbing, also known as greedy search, is one of the simplest search methods that attempts to find a local maximum by moving in an "uphill" direction. The basic Hill-climbing technique, given in Figure 2.1, illustrates the steps of this algorithm. Like any other local search method, its success is mainly dependent on the starting point and neighborhood structure.


```
procedure iterated hill-climber
begin
        t ← 0
        initialize best to zero
        repeat
                local ← false
                select current point v_c at random
                evaluate v_c
                repeat
                        select all in the neighborhood of v_c
                        select the point v_n from the set of new points
                                with the best value of evaluation function eval
                        if eval(v_n) is better than eval(v_c)
                                then v_c ← v_n
                        else local ← true
                until local
                t ← t + 1
                if v_c is better than the best
                        then best←v_c
        until t = MAX
end
```

Figure 2.1 Simple iterated Hill Climber [18]

A neighborhood is the set of solutions that can be derived from the original solution using a metric function such as distance. Depending on the type of problem, the search space, and the representation used, neighborhood solutions can be derived from the original

solutions in many different ways. Euclidian distance, for example, is commonly used in search spaces with continuous variables [18].

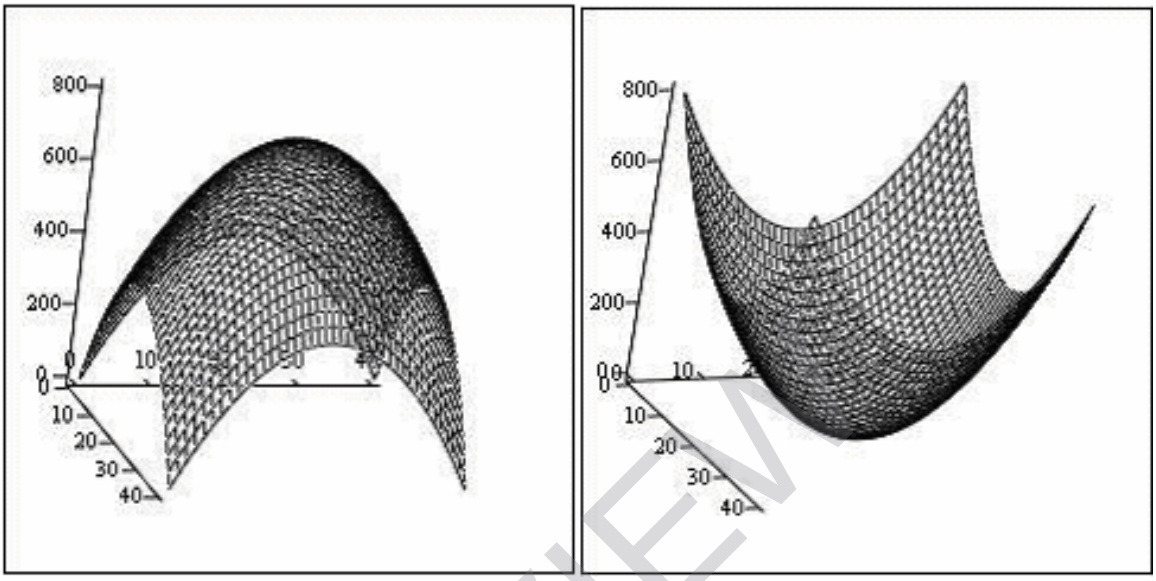

Figure 2.2 Hill Climbers: Gradient Ascent/Descent

Hill climbers can be very effective. They move to better solutions very fast and in search spaces like that in Figure 2.2, they can find the global optima. But real-world search spaces have many local optimums, plateaus and ridges (Figure 2.3). A local maxima, is a peak higher than or equal to any other point in its neighborhood, but possibly not as high as a global maxima. A hill climbing algorithm which gets closer to a local maxima will be drawn upwards to that peak and will get stuck there, thus ignoring other potentially better local optima as well as global optima [24].

Getting stuck in a local optima is a major problem that has been worked on by many researchers for a long time. Different search algorithms have been designed whose heuristics escape local optima, allow the search to be independent of starting point, and

thus attain solutions close to global optima. The rest of this chapter will provide an overview of these algorithms which are generally called *metaheuristics*.

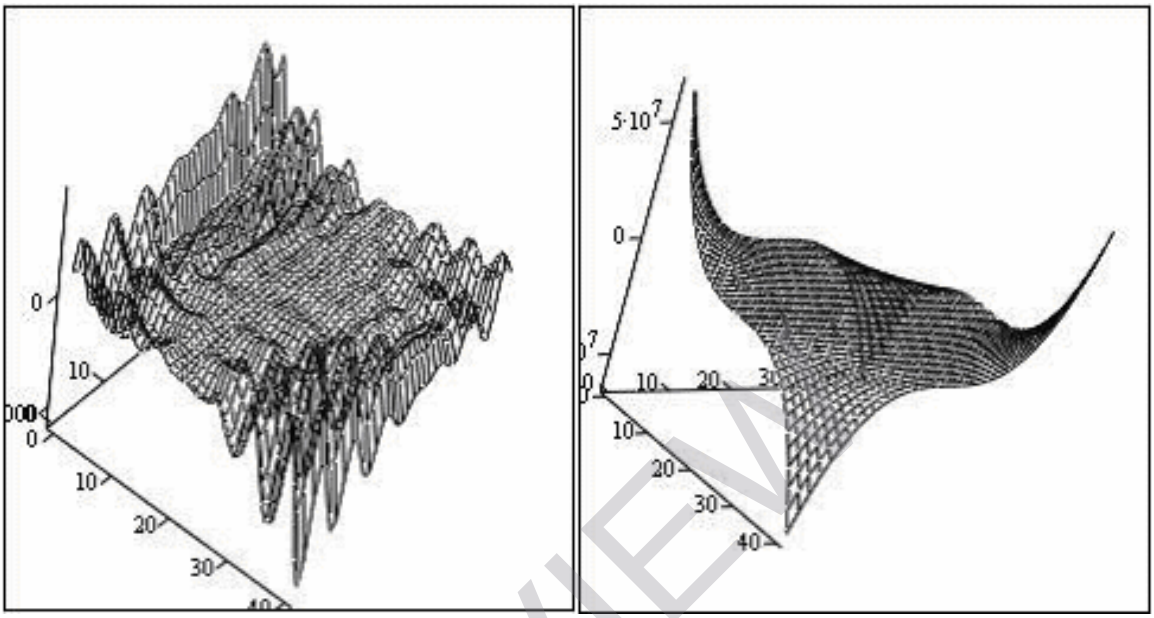

Figure 2.3 Real world search spaces

2.2 Escaping Local Optima – Metaheuristics

A metaheuristic is a general-purpose heuristic method designed to guide the search for a good solution towards more promising regions of the search space [10]. Effective and efficient exploration of the search space are the primary objectives of metaheuristics. Roli and Blum surveyed the literature for metaheuristics in [2] and outlined the following properties:

- Metaheuristics are strategies that "guide" the search process.
- The goal is to efficiently explore the search space in order to find optimal or near optimal solutions.

- Techniques which constitute metaheuristic algorithms range from simple local search procedures to more complex learning processes.
- Metaheuristic algorithms are approximate and usually non-deterministic.
- They may incorporate mechanisms to avoid getting trapped in confined areas of the search space.
- Metaheuristics are not problem-specific.
- Metaheuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy.
- Today's more advanced metaheuristics use search experience (embodied in some form of memory) to guide the search.

The antipodal strategies of diversification and intensification are used in metaheuristic search. Diversification refers to exploration of the different regions of the search space, whereas intensification limits the search area based on accumulated search experience. The balance between these two is the key to the success of a metaheuristic.

Metaheuristics are classified based on different properties of diversification and intensification strategies they used. There are different taxonomies in the literature but discussion of these is beyond the scope of this work. In this dissertation we study the following two classes: trajectory methods and population based methods [2].

2.3 Trajectory Methods

Local search-based algorithms working on a single solution are called trajectory methods. They all share the property of describing a trajectory of this singleton in the search space during the search process. Like the hill climbing algorithm illustrated in Figure 2.1 above, they start with an initial solution and follow a path based on the strategy defined in the algorithm, problem representation, and neighborhood structure. In the iterative local search algorithm, a simple strategy of moving towards another potential solution in the neighborhood, provided that it's better than current solution, is used. As previously mentioned, this approach is simple but ineffective in avoiding the local optimums which may be far worse than other local optima in the search space. Simulated Annealing and Tabu Search are two of the approaches developed by researchers to address this deficiency in simple local search.

2.3.1 Simulated Annealing

The main reason for getting stuck in local optima in a simple trajectory method is that the strategy always moves toward a better solution. So one simple technique to escape local optima would be occasionally moving to a solution, even though it's not better than the current solution. This will lead to exploration of new regions of the search space that would otherwise not be visited at all. This is exactly what the Stochastic Hill Climber algorithm in Figure 2.4 does. Here, the decision to accept a potential solution from a neighborhood is based on a probability function. This makes moving to a solution which is worse than the current solution possible. One of the factors this probability depends on is the difference between the fitness values of the current solution and new solution. Thus, if the new solution is worse than the current solution, then there is a lower, albeit

nonzero, probability of accepting the new solution. Parameter T is used to control this dependency. With very high values of T, such as 10^{10}, the search becomes completely stochastic. This means that the relative fitness values of the competing solutions do not affect the acceptance decision at all. On the other hand, if T is set to 1, the search becomes purely deterministic and acts like the simple iterative search.

procedure iterated hill-climber
begin
    $t \leftarrow 0$
    select current point v_c at random
    evaluate v_c
    repeat
        select the point v_n from the neighborhood randomly
        select v_n as the new solution with probability $\frac{1}{1+e^{\frac{eval(v_c)-eval(v_n)}{T}}}$
        $t \leftarrow t + 1$
    until $t = MAX$
end

Figure 2.4 Stochastic Hill Climber [18]

In the algorithm above, the value of T stays constant. Therefore, the probability of accepting a solution that is worse than the current solution is same throughout the run. Allowing the acceptance of worse solutions in the beginning of the search facilitates diversification and prevents premature convergence to a suboptimal global solution. However at the later stages of search, intensification might be preferable. Such a strategy, that adjusts the degree of diversification and intensification during the search, requires changing values of T. This is the main difference between Simulated Annealing (Figure 2.5) and stochastic hill climbing described above.