Presented at "AAAI: Knowledge Discovery and Data Mining 1996", Portland, Oregon.

A Genetic Algorithm-Based Approach to Data Mining

Ian W. Flockhart[a] **and Nicholas J. Radcliffe**[a,b]
{iwf,njr}@quadstone.co.uk
[a]Quadstone Ltd, 16 Chester Street, Edinburgh EH3 7RA, UK
[b]Department of Mathematics and Statistics, University of Edinburgh,
King's Buildings, EH9 3JZ, UK

Abstract

Most data mining systems to date have used variants of traditional machine-learning algorithms to tackle the task of *directed* knowledge discovery. This paper presents an approach which, as well as being useful for such directed data mining, can also be applied to the further tasks of *undirected* data mining and *hypothesis refinement*. This approach exploits parallel genetic algorithms as the search mechanism and seeks to evolve explicit "rules" for maximum comprehensibility. Example rules found in real commercial datasets are presented.

Introduction

Genetic algorithms (Holland, 1975) have been used successfully in a variety of search and optimisation problems. Work on genetic algorithm-based learning has traditionally been grouped into one of two general approaches. The *Pitt* approach (Smith, 1980) uses a traditional genetic algorithm in which each entity in the population is a set of rules representing a complete solution to the learning problem. The *Michigan* approach (Holland, 1986) has generally used a distinctly different evolutionary mechanism in which the population consists of individual rules, each of which represents a partial solution to the overall learning task.

The system we present here — GA-MINER — is a general pattern search tool supporting several pattern forms and capable of functioning with varying levels of user supervision. Although the system may be used for traditional classification tasks, the emphasis has been placed more on pattern discovery. The authors have sought to divide data mining into three types or levels as follows:

- *Undirected or Pure Data Mining.* Here the concept is that the user asks of the data miner: "Tell me something interesting about my data". The key point is that the user is *not* specifying what kind of rule is desired. The system is left relatively unconstrained and is therefore given the greatest "freedom" to discover patterns in the data free of prejudices from the user. It seems likely that in these circumstances there is the greatest scope for finding completely unexpected patterns in the data, which has been one of the "promises" of data mining.

- *Directed data mining.* The user asks something much more specific, such as: "Characterise my high spending customers". Here a much stronger "steer" is being given to the system and the problem usually changes from a general pattern-detection problem to a rather better defined *induction* problem.

- *Hypothesis testing and refinement.* The user conceptually says: "I think that there is a positive correlation between sales of peaches and sales of cream: am I right?". Now the idea is that the system first evaluates the hypothesis but then—if the evidence for it is not strong—seeks to refine it. Depending on what scope for variation the system is allowed, this may make the task even more directed than "directed data mining", or almost as open as "undirected data mining".

GA-MINER is unusual in being applicable to all three kinds of data mining. It is undirected data mining that has been our defining goal, but we have deliberately built a system which allows directed data mining and hypothesis refinement also to be tackled. Directed data mining is achieved by fixing certain parts of the pattern over the course of the run, and hypothesis refinement is achieved by "seeding" the system with the hypothesis but then allowing some or all parts of it to vary.

Related Work on Genetic Algorithm-Based Learning

An early example of a genetic algorithm-based machine learning system is LS-1 (Smith, 1980, 1984), which introduced a structured representation based on the semantics of the problem domain with genetic operators

working at each level. GABIL (DeJong *et al.,* 1993) uses the Pitt approach for evolving concept descriptions, which are defined as a collection of possibly overlapping classification rules. COGIN (Greene & Smith, 1993, 1994) addresses multi-class problem domains by introducing competition for coverage of training examples, encouraging the population to work together to solve the concept learning task. Another recent example is REGAL (Neri & Giordana, 1995; Giordana *et al.,* 1994) which also uses a coverage-based approach for multi-concept learning and introduces a new *Universal Suffrage* selection operator to encourage cooperation between population members. Augier *et al.* (1995) present an algorithm, SIAO1, for learning first order logic rules with a genetic algorithm. Domain knowledge may be introduced in the form of domain hierarchies and the algorithm uses a covering technique to ensure that all examples are covered by some rule.

Related Work on Data Mining

GA-MINER has also drawn ideas from a number of non-genetic data mining tools, particularly regarding pattern forms. EXPLORA (Kloesgen, 1994) is an interactive statistical analysis tool for discovery in databases. A number of *statement types* (or patterns) are defined, and users may select the most appropriate for their particular analysis purposes. Forty-Niner (Zytkow & Baker, 1991; Zytkow & Zembowicz, 1993) searches for regularities in databases, that is, a pattern and the range within which it holds. The representation of a range is as a conjunction of attribute/value-sets, while a pattern is either a function (an equation relating the attributes) or a contingency table.

The Reproductive Plan Language

Fast and flexible development of GA-MINER was made possible by its implementation in the *Reproductive Plan Language* RPL2 (Surry & Radcliffe, 1994b, 1994a). RPL2 is an extensible language, interpreter and run-time system for the implementation of stochastic search algorithms, with a special emphasis on evolutionary algorithms such as genetic algorithms. The main features of RPL2 pertinent to GA-MINER are automatic parallelism, support for arbitrary representations (important in the current context as the rule forms are structured and not string-like), and its large library of functions which has allowed almost the entire project to be devoted to an exploration of data mining itself, rather than merely coding up ideas.

Pattern Representation

GA-MINER includes a variety of pattern forms, drawing on ideas from EXPLORA and Forty-Niner, including explicit rule patterns, distribution shift patterns and correlation patterns. The basis for all supported patterns is *subset description*.

Subset descriptions are clauses which are used to select subsets of the database, and form the main heritable units which are manipulated by the genetic algorithm. A subset description consists of a disjunction of conjunctions of attribute-value or attribute-range constraints, shown in *Backus-Naur* form below:

Subset Description	::=	*Clause* [**or** *Clause*]
Clause	::=	*Term* [**and** *Term*]
Term	::=	*Attribute* **in** *Value Set*
	|	*Attribute* **in** *Range*

Patterns are then constructed as higher level interpretations of a number of these subsets. For example, an explicit rule pattern may use two subset descriptions, C and P to represent the *condition* and *prediction* respectively of a rule: "**if** C **then** P". Note that the interpretation of a collection of subset descriptions as a particular form of pattern is defined entirely by the chosen evaluation function. For example, a rule "**when** S, **if** C **then** P" can be constructed from three subset descriptions, S, C and P respectively, combined with an appropriate evaluation function.

In a similar manner, a distribution shift pattern may be formed from two subset descriptions together with a *hypothesis variable*, which is simply a field from the database to which the pattern refers. In this case the subset descriptions C and P and the hypothesis variable A are interpreted as a pattern of the form:

> The distribution of A **when** C **and** P
> is significantly different from the distribution of A
> when C.

Finally, correlation patterns express a relationship which holds between two hypothesis variables within a particular subset of the database. i.e. patterns of the form:

> **when** C, the variables A and B are correlated.

Since the same underlying representation is used for all pattern forms, the same genetic algorithm may be used to manipulate all these patterns.

Pattern Templates

Pattern templates are used to constrain the system to particular forms of patterns, and allow extensive control over a number of features including the high-level pattern form as described above, the database fields permitted to appear in each of the subset descriptions, the maximum numbers of disjunctions and conjunctions permitted in each subset description and

any clauses, terms or fields which must be included as part of a subset description.

The component parts of the pattern template are marked as either *initialised* or *fixed*. Fixed parts of the template are inherited by every pattern in the population and are never modified by crossover or mutation. Initialised parts of the template appear in all newly generated patterns but may be modified during the search.

Undirected data mining may be performed by using a minimal template, and directed data mining by restricting the pattern form more tightly. Hypothesis refinement is achieved by seeding the initial population of the genetic algorithm with patterns based on the template but with additional randomly generated components, and the search is permitted to modify these patterns subject to the constraints specified by the template.

Pattern Evaluation

A number of evaluation functions for estimating pattern interest have been used during the course of this work, mainly based on statistical measures. For example, for rule patterns we have used, among others, information gain (see Frawley, 1991), the J-measure (Smyth & Goodman, 1991) and the form suggested in Piatetsky-Shapiro (1991). Our experience has shown that many of these evaluation mechanisms give qualitatively similar results (Flockhart & Radcliffe, 1995).

While falling short of providing fully satisfactory definitions of interesting patterns, the evaluation functions have been sufficiently successful for the system to discover several useful patterns in the databases provided by our industrial collaborators, GMAP Ltd and Barclays Bank plc. Although the system still produces some non-interesting rules, tautologies are generally discarded early in the search and the pattern templates may be used to steer the system away from obvious domain knowledge.

The Genetic Algorithm

The genetic algorithm in GA-MINER uses a structured population model in which each genome's reproductive partner is selected from within its local neighbourhood (using tournament selection). This helps prolong diversity within the population and encourages local niching, which tends to result in exploration of several areas of the search space and matches well with the goal of finding several patterns in a single run.

The crossover operator is defined at a variety of levels, reflecting the structure of the representation. Disjunct clauses, clauses, terms and attribute values each comprise a single gene at the appropriate level. Within subset descriptions, crossover at the disjunct clause level is based on uniform crossover and enforces positional alignment of component clauses. Both uniform and single-point crossover are used at the clause level, while crossover at the term level is again based on uniform crossover. Mutation is also defined at a variety of levels, with separate probabilities specified for mutating each of the component parts. Clauses, terms and values are added or deleted with specified probabilities and can be regarded as distinct specialisation and generalisation operators.

GA-MINER collects sets of patterns during the run of the algorithm for presentation to the user. A simple heuristic is used for updating the set, based on a strategy of continually replacing either the lowest fitness rule or the most similar rule (if the similarity is over a given threshold) by a higher fitness rule.

Examples of Discovered Patterns

Many patterns of varying strength were discovered within the data provided by our industrial collaborators, however, we restrict ourselves to just two representative examples of the kinds of patterns discovered. Both were found by using the system in "undirected" mode with a minimal pattern template.

Explicit Rule Pattern

if Proportion of households with 1 child ≥ 0.12
(Approximate percentiles 36% - 100%)
(true: 1618 false: 939 unique false: 272)

and

Number of Ford Dealers > 0
(Approximate percentiles 62% - 100%)
(true: 936 false: 1621 unique false: 809)

and

Proportion of households with 3+ cars **in** 0.01 .. 0.07
(Approximate percentiles 4% - 86%)
(true: 2038 false: 519 unique false: 108)

then

Ford market share segment F (Sierra) **in** 0.06 .. 0.75
(Approximate percentiles 30% - 100%)
(true: 1770 false: 787 unique false: 787)

Left hand side matches 19% of the database
Right hand side matches 69% of the database

	Expected	Actual
Accuracy:	69%	93%
Coverage:	20%	27%

The rule above states that there is a 93% probability that Ford market share of segment F (Sierra) is between 0.06 and 0.75 in postal districts where there is at least one Ford dealer, the proportion of households with 1 child is relatively high (in the top 64% of the distribution) and the proportion of households with 3 or more cars is neither very high nor very low. This compares to an expected probability of 69%, under the assumption of no relationship between the left and right hand sides of the rule. The true and false count for each clause show the number of times that clause is true and false respectively, while the unique

false count shows the number of times the clause is false when all the others are true. This gives some indication of the relative importance of the terms.

Distribution Shift Pattern

The distribution of "Ford market share of segment D (Escort)"
when Proportion of households — Council ≥ 0.20
(Approximate percentiles 62% - 10%)
(true: 929 false: 1628 unique false: 123)
and
Proportion of households with 2 children ≥ 0.11
(Approximate percentiles 26% - 100%)
(true: 1854 false: 703 unique false: 233)
and
Proportion of unemployed in population ≥ 0.04
(Approximate percentiles 52% - 100%)
(true: 1183 false: 1374 unique false: 48)
and
Proportion of households with 0 cars ≥ 0.31
(Approximate percentiles 60% - 100%)
(true: 1015 false: 1542 unique false: 89)

has median 0.28 and is significantly shifted
from the overall distribution which has median value 0.20.

The distribution shift rule above says that Ford market share in segment D is 8% higher in postal districts with a high proportion of council houses, relatively high unemployment, a high proportion of households with no car and relatively few households with 2 children.

Conclusions and Future Work

GA-MINER has demonstrated that genetic algorithms may be used successfully for a variety of pattern discovery tasks in addition to their traditional use in classification and concept learning. Genetic algorithms appear well suited to undirected data mining, given their limited need for user direction and user interaction, however we have also demonstrated that they may be used for more directed forms of data mining through the use of pattern templates. In particular, the authors believe that the system's use for hypothesis refinement holds much promise. The scrutibility of the patterns generated by the system also make results more understandable than those produced by many other unsupervised methods such as neural networks, an essential component for any system which is to be widely used by non-experts.

The system would undoubtedly benefit from increased used of domain knowledge, perhaps in the form of domain hierarchies. Further work to allow some degree of fuzziness in the form of pattern templates would also be useful.

Finally, although GA-MINER has been successfully parallelised and is scalable on main memory databases, it is becoming increasingly apparent that commercial data mining systems will require to access volumes of data far in excess of available main memory. This is likely to mean that effective data mining systems will be dependent more on scalable data warehouse systems than on explicitly parallel algorithms.

Acknowledgements

This work was funded by a research grant from the UK Engineering and Physical Science Research Council's AIKMS programme, and was carried out while the authors were at Edinburgh Parallel Computing Centre.

References

S. Augier, G. Venturini, and Y. Kodratoff, 1995. Learning first order logic rules with a genetic algorithm. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*. AAAI Press.

Kenneth A. DeJong, William M Spears, and Diana F Gordon, 1993. Using genetic algorithms for concept learning. *Machine Learning*, 13:161–188.

Ian W. Flockhart and Nicholas J. Radcliffe, 1995. GA-MINER: Parallel data mining with hierarchical genetic algorithms. Technical Report EPCC-AIKMS-GA-MINER-REPORT, Edinburgh Parallel Computing Centre.

William J. Frawley, 1991. Using functions to encode domain and contextual knowledge in statistical induction. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, *Knowledge Discovery in Databases*, pages 261–275. MIT Press.

Attilio Giordana, Filippo Neri, and Lorenza Saiat, 1994. Search-intensive concept induction. Technical report, Univerità di Torino, Dipartimento di Informatica, Corso Svizzera 185, 10149 Torino, Italy.

David Perry Green and Stephen F. Smith, 1993. Competition-based induction of decision models from examples. *Machine Learning*, 13:229–257.

David Perry Greene and Stephen F. Smith, 1994. Using coverage as a model building constraint in learning classifier systems. *Evolutionary Computation*, 2(1).

John H. Holland, 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press (Ann Arbor).

John H. Holland, 1986. Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems. *Machine Learning, an artificial intelligence approach*, 2.

W. Klösgen, 1994. Exploration of simulation experiments by discovery. In *Proceedings of KDD-94 Workshop*. AAAI.

Filippo Neri and Attilio Giordana, 1995. A parallel genetic algorithm for concept learning. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*. Morgan Kaufman.

Gregory Piatetsky-Shapiro, 1991. Discovery, analysis and presentation of strong rules. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, *Knowledge Discovery in Databases*. MIT Press.

Stephen F. Smith, 1980. *A Learning System Based on Genetic Adaptive Algorithms*. PhD thesis, University of Pittsburgh.

Stephen F. Smith, 1984. Adaptive learning systems. In Richard Forsyth, editor, *Expert Systems, Principles and case studies*. Chapman and Hall Ltd.

Padhraic Smyth and Rodney M. Goodman, 1991. Rule induction using information theory. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, *Knowledge Discovery in Databases*, pages 159–176. MIT Press.

Patrick D. Surry and Nicholas J. Radcliffe, 1994a. *The Reproductive Plan Language RPL2*. Edinburgh Parallel Computing Centre.

Patrick D. Surry and Nicholas J. Radcliffe, 1994b. RPL2: A language and parallel framework for evolutionary computing. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature III*, pages 628–637. Springer-Verlag, Lecture Notes in Computer Science 866.

Jan M. Zytkow and John Baker, 1991. Interactive mining of regularities in databases. In Gregory Piatetsky-Shapiro and William J. Frawley, editors, *Knowledge Discovery in Databases*, pages 31–53. MIT Press.

Jan M. Zytkow and Robert Zembowicz, 1993. Database exploration in search of regularities. *Journal of Intelligent Information Systems*, 2:39–81.