Genetic Algorithms for Selection and Partitioning of Attributes in Large-Scale Data Mining Problems

William H. Hsu, Michael Welge, Jie Wu, and Ting-Hao Yang
Automated Learning Group, National Center for Supercomputing Applications
601 East Springfield Avenue, Champaign IL 61820
{bhsu | billp | welge | jiewu | tingy}@ncsa.uiuc.edu http://www.ncsa.uiuc.edu/STI/ALG

Abstract

This paper proposes and surveys genetic implementations of algorithms for selection and partitioning of attributes in large-scale concept learning problems. Algorithms of this type apply *relevance determination* criteria to attributes from those specified for the original data set. The selected attributes are used to define new data clusters that are used as intermediate training targets. The purpose of this *change of representation* step is to improve the accuracy of supervised learning using the reformulated data. Domain knowledge about these operators has been shown to reduce the number of fitness evaluations for candidate attributes. This paper examines the genetic encoding of attribute selection and partitioning specifications, and the encoding of domain knowledge about operators in a fitness function. The purpose of this approach is to improve upon existing search-based algorithms (or *wrappers*) in terms of training sample efficiency. Several GA implementations of alternative (search-based and knowledge-based) attribute synthesis algorithms are surveyed, and their application to large-scale concept learning problems is addressed.

Keywords: **genetic algorithms, constructive induction, multi-strategy (hybrid) learning, wrappers, large-scale data mining**

Introduction

This paper presents the problems of reducing and decomposing large-scale concept learning problems in knowledge discovery in databases (KDD). The approach described here adapts the methodology of *wrappers* for performance enhancement and attribute subset selection [JKP94, KJ97] to a genetic optimization problem. The fitness functions for this problem are defined in terms of classification accuracy given a particular supervised learning technique (or *inducer*) [KS96]. More precisely, the quality of a subset of attributes is measured in terms of empirical generalization quality (accuracy on cross-validation data, or a continuation of the data in the case of time series prediction).

The paper first presents a brief introduction to constructive induction, discusses the role of attribute synthesis (feature construction) in the framework of constructive induction for KDD applications. Next, it summarizes existing search-based approaches to knowledge-based constructive induction, their tradeoffs, and their benefits and shortcomings. The primary focus of this paper is on recent and new research in applying genetic algorithms to attribute synthesis [RPG+97], especially adaptation of algorithms based on constraint knowledge [Do96]. The paper then describes critical scalability issues for industrial applications (in the context of KDD problem solving environments) that have arisen in preliminary applied work by the author. Finally, the paper documents work in progress on designing genetic encodings for attribute selection and partitioning problems, the selection criteria, incorporation and refinement of constraint knowledge about operators, and scalability issues (such as task-parallel configurations of GA wrappers).

Attribute Selection, Partitioning, and Synthesis

The synthesis of a new group of attributes (also known as the *feature construction* problem) in inductive concept learning is an optimization problem. Its control parameters include the attributes used (i.e., which of the original inputs are relevant to distinguishing a particular target concept) [KR92, KJ97, Hs98], how they are grouped (with respect to multiple targets), and how new attributes are defined in terms of *ground* (original) attributes. This synthesis and selection problem is a key initial step in *constructive induction* – the reformulation of a learning problem in terms of its inputs (attributes) and outputs (concept class descriptors).

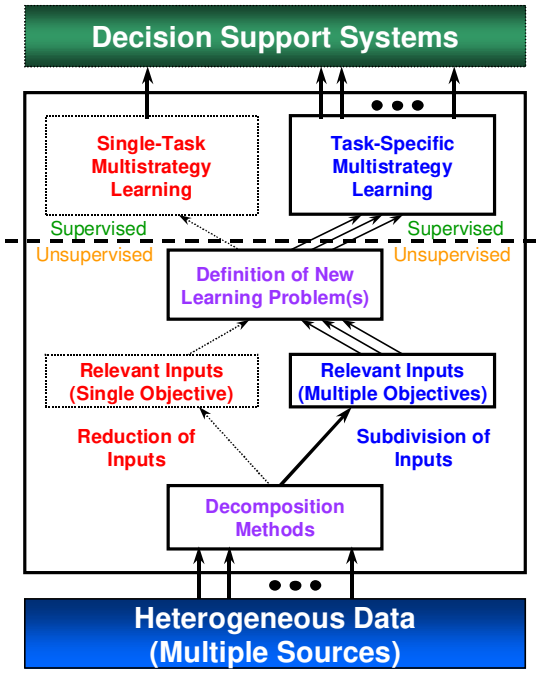

Figure 1. Attribute-based transformations in KDD

Figure 1 illustrates the role of attribute selection (reduction of inputs) and partitioning (subdivision of inputs) in constructive induction (the "unsupervised" component of this generic KDD process). In this framework, the input consists of *heterogeneous* data (that originating from multiple sources). The performance element includes time series classification [Hs98, HR98a, HR98b] and other forms of pattern recognition that are important for decision support.

Attribute Partitioning in Constructive Induction

Attribute subset selection is the task of focusing a learning algorithm's attention on some subset of the given input attributes, while ignoring the rest [KR92, Ko95, KJ97]. In this research, subset selection is adapted to the systematic decomposition of concept learning problems in heterogeneous KDD. Instead of focusing a single algorithm on a single subset, the set of all input attributes is partitioned, and a specialized algorithm is focused on *each* subset. While subset selection is used to refinement of attribute sets in single-model learning, attribute partitioning is designed for multiple-model learning.

This new approach adopts the role of feature construction in constructive induction: to formulate a new input specification from the original one [Do96]. It uses subset partitioning to *decompose* a learning task into parts that are individually useful, rather than to *reduce* attributes to a single useful group. This permits new intermediate concepts to be formed by unsupervised learning (e.g., conceptual clustering [Mi83] or cluster formation using self-organizing algorithms [HW99]). The newly defined problem or problems can then be mapped to one or more appropriate hypothesis languages (model specifications) as illustrated in Figure 1. In the new system, the subproblem definitions obtained by partitioning of attributes also specify a mixture estimation problem. A data fusion step, shown in Figure 2, occurs after training of the models for all subproblems) [HR99].

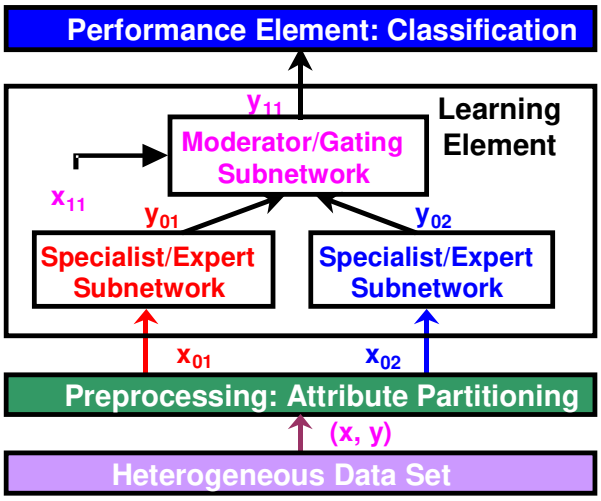

Figure 2. The attribute partitioning approach

Together with attribute subset selection, attribute partitioning permits a concept learning problem to be refined for both increased classification accuracy and comprehensibility. The latter increases the utility of the model in systems that combine multiple models, such as hierarchical data fusion systems [HR98a, HR98b, RH98, Hs98] and large-scale multi-strategy data mining systems [HW99]. Note that these systems may incorporate different type of concept learning algorithms, such as artificial neural networks. In our application, the multi-strategy (hybrid) learning system is a GA wrapper that selects and configures probabilistic networks (especially temporal ANNs) and decision trees for KDD applications.

The primary novel issue addressed by this position paper is the practical application of attribute synthesis methods to very large databases. These methods been extensively studied as solutions to a high-level parameter optimization problem – especially in the wrapper research of Kohavi *et al* [KJ97]) – but have (to date) not yet been investigated in depth as an application of genetic algorithms. This paper presents the case that scaling up of large-scale data mining methods, especially using GAs, depends on first understanding how to systematically control the *definition* of inductive concept learning problems. The experimental framework for adaptation of constraint-based attribute selection and synthesis methods that is described here proposes that this type of knowledge can be readily incorporated into a genetic algorithm for constructive induction.

Background

The Constructive Induction Problem and Supervised Concept Learning

In current practice, optimization problems in constructive induction are treated as a state space search [Do96, KJ97, Hs98]. The primary difficulty encountered in applying search-based algorithms to synthesize attributes [Do96], select subsets of relevant attributes [KJ97], or partition attributes into useful categories [Hs98] is the combinatorial complexity of uninformed search. The ability to constrain and control the search for useful attributes (or groups of them) is critical to making constructive induction viable. Toward this end, both domain knowledge and evaluation metrics have been applied in informed search algorithms (gradient and A*) for attribute subset selection [KJ97] and partitioning [Hs98]. The dissertations of Gunsch [Gu91] and Donoho [Do96], which respectively address constructive induction based on *partial* domain knowledge ("opportunistic") and based on constraint knowledge ("theory-guided"), contain excellent and comprehensive bibliographies and surveys relating to constructive induction. The interested reader is therefore referred to these publications for a general introduction to previous related work.

The definition of a concept learning problem consists of input *attributes* and *concept classes*. Each attribute is a function that maps an example, **x**, into a value.

Conversely, a *classified* example can be defined as an object (a tuple of attribute values) whose type is the range of all combinations of these attributes together with a concept class, y. The task of an *inductive concept learning* algorithm is to produce a *concept description*, $y = g(\mathbf{x})$, that maps a newly observed example $\mathbf{x}$ to its class y [Do96]. In inductive concept learning, therefore, the input (a training data set) consists of classified examples, and the output is a concept descriptor (a representation of the concept description such as a decision tree, classification rule base, linear separator, or classifier system) [BGH89, Mi97]. This classifier can then be applied to each new (unclassified) example to obtain a prediction (hypothesis) of its class.

Constructive induction is the problem of producing new descriptors of training examples (instances) and target classes in concept learning [Mi83, RS90]. It can be regarded as an *unsupervised learning* process that refines, or *filters*, the attributes (also referred to as *features* or *instance variables*) of some concept learning problem [KJ97, Hs98]. The objective function of this process, called an attribute *filter* in the attribute subset selection and extraction problem [KJ97, Hs98], is the *expected performance* of a given supervised learning algorithm on the data set, restricted to the selected attributes. This expected performance measure can be based on any quantitative or qualitative analysis of the data set (including heuristic figures of merit), but the common trait of all attribute filters is that they operate independently of the induction algorithm (i.e., they ignore credit assignment based on actual supervised learning quality). The filter method can be used not only to *select* attributes, but to *compose* them using operators, such as the arithmetic operators {**+, -, *, /**}. The objective criterion is still based strictly on factors other than direct observation of supervised learning quality.

A more sophisticated variant, suitable for attribute selection [Ko95, KJ97], partitioning [Hs98], or synthesis [Do96], casts the selection problem (for 0-1 subset membership, i.e., inclusion-exclusion; for subset membership; or for operator application order) as a multi-criterion optimization function. This function is defined subject to constraints of supervised learning performance: cross-validated classification accuracy and convergence time are most prevalent. This type of optimization is based on multiple runs of the supervised learning algorithm (concurrent across any population of candidate configurations, i.e., subsets, partitions, or synthetic attribute sets; serial among generations of candidates). Because it takes the supervised learning algorithm into account and invokes it as a subroutine, this approach is referred to as the *wrapper* methodology. Wrappers can be used for both attribute reformulation (part of constructive induction) and other forms of parameter tuning in inductive learning [Ko95]. It is important to note that to date, attribute selection, partitioning, and synthesis wrappers have not been studied as genetic algorithms, although stochastic and heuristic search and optimization methods have been applied [Ko95, KJ97, Hs98].

Composition of new attributes by such methods has been shown to increase accuracy of the classifiers produced by applying supervised learning algorithms to the reformulated data. The rationale is that concept learnability can be improved relative to given supervised learning algorithm through alternative representation [Ha89]. The step of transforming low-level attributes into useful attributes for supervised learning is known as *attribute synthesis* or, as is more common in the computational intelligence literature, *feature construction*. The complementary step to feature construction is *cluster definition*, the transformation of a given class definition into a more useful one [Do96, HW99].

Attribute Partitioning as Search

Both filters and wrappers for attribute selection and partitioning can be purely search-based or can incorporate constraint knowledge about operators, especially *which groups of attributes are coupled* (i.e., should be taken together for purposes of computing joint relevance measures [KJ97]). [DR95] and [Do96] discuss the use of such constraint knowledge in constructive induction. For example, in the automobile insurance KDD problem surveyed below, formulae are computed for *loss ratio* in automobile insurance customer evaluation. Only the number of exposures (units of customer membership) should be allowed as a denominator. Only certain attributes denoting loss paid (on accidents, for example) should be permitted as numerators, and these should always be summed. Similarly, *duration* attributes are a type of attribute that is always produced by taking the difference of two dates. This type of domain knowledge-guided constructive induction drastically reduces the search space of candidate attributes from which the filter or wrapper algorithm must select.

For the rest of this paper, the discussion shall focus on wrappers that apply supervised learning algorithms (such as decision tree induction, simple GAs, and multi-layer perceptrons). The objective criterion for reformulation of a large-scale inductive learning problem in KDD is defined in terms of classification accuracy, and this leads naturally to the family of fitness functions and the scalability issues described below.

In search-based algorithms for attribute synthesis, constraint knowledge about operators has been shown to reduce the number of fitness evaluations for candidate attributes [Do96]. This paper shows how constraint knowledge about operators can be encoded in a fitness function. The purpose of this approach is to improve upon the non-genetic, search-based algorithm in terms of training sample efficiency. Several GA implementations of alternative (search-based and knowledge-based) attribute synthesis algorithms are surveyed, and their application to large-scale concept learning problems is addressed.

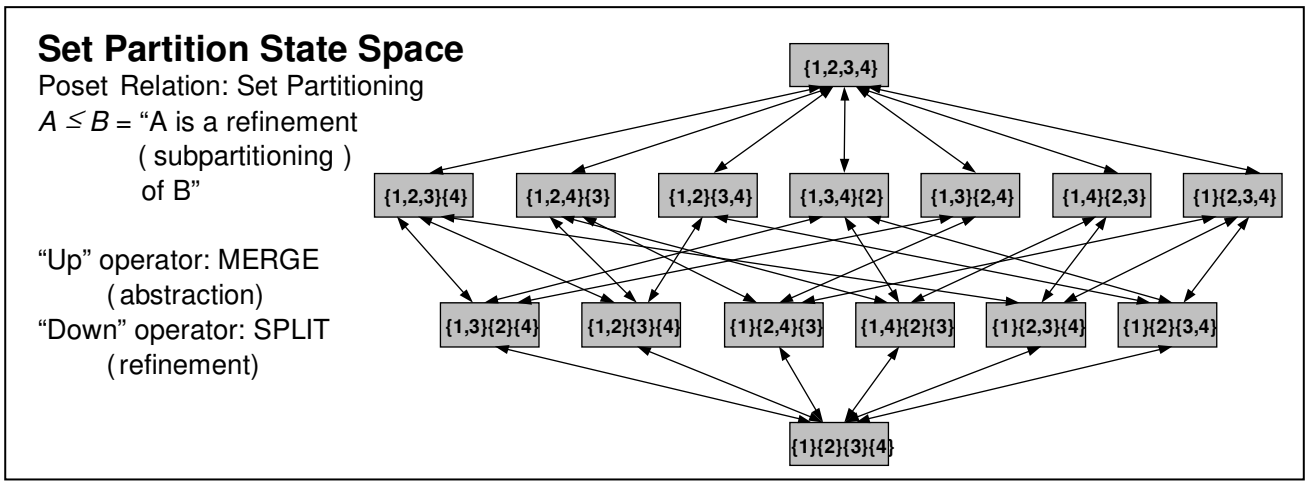

Figure 3. A numeric encoding of individuals for attribute partitioning

Methodology of Applying GAs to Constructive Induction

Extending the Traditional Algorithm

This section briefly describes an encoding for attribute synthesis specifications for a simple GA with single-point crossover and a family of fitness functions that captures the objective criteria for wrapper systems.

Raymer *et al* [RPG+97] use a masking GA, containing indicator bits for attributes to simultaneously extract and select attributes for a *k-nearest neighbor* (knn) supervised learning component. This masking GA is very similar to the state space encoding used by Kohavi *et al* for attribute subset selection [Ko95, KJ97], and is quite standard (e.g., forward selection and backward elimination algorithms in linear regression are described in similar fashion). Furthermore, the bit mask (inclusion-exclusion) encoding has an analogue in attribute partitioning [Hs98] that can be applied to encode pairwise sequential operations on attributes. Some related work on genetic search for feature selection permits replication of attributes by using a membership coding [CS96].

The bit-mask coding is natural for attribute selection, but must be adapted for attribute partitioning. In the genetic wrapper for partitioning, two codings can be used. The first is a sparse n-by-n bit matrix encoding, where 1 in column j of row i denotes membership of the ith attribute in subset j. Empty subsets are permitted, but there can be no more than n. Also, in this design, membership is mutually exclusive (in a true partition, there is no overlap among subsets). The second coding uses numeric membership as in the state space representation, and is shown in Figure 3; this is a more compact encoding but requires specialized crossover operators (corresponding to subset exchange) as well as mutation operators (corresponding to abstraction and refinement).

For an attribute selection, partitioning, or synthesis wrapper, the fitness function must always reflect the figure(s) of merit specified for the performance element of the KDD system. If this is a basic supervised concept learner that generates predictions, the fitness function should be based upon classification error (0-1, mean-squared error, or whatever loss function is actually used to evaluate the learner). This is not *necessarily* the same loss function as is used in the supervised learning algorithm (which may, for example, be based on gradient descent), but it frequently is. If the performance element is a classifier system [BGH97], the fitness function for this wrapper should express the same criteria.

Finally (and most important), the constraint knowledge for operator *preference* can be encoded as a penalty function and summed with the performance measure (or applied as a quick-rejection criterion). That is, if some operator is not permitted or not preferred, a penalty can be assessed that is either continuous or 0-1 loss.

Functional (Task-Level) Parallelism in Change-of-Representation Search

As do simple GAs for most concept learning problems (supervised and unsupervised), genetic wrappers exhibit a high degree of functional (task-level) parallelism, as opposed to data parallelism (*aka* array or vector parallelism). This is doubly true for genetic attribute synthesis wrappers. With replication of the data across cluster nodes, the inter-task communication is limited to a specification string and the fitness value, with all of the computation for *one run* of the supervised learning algorithm being performed on a separate processor. The evaluation of each component of the specification (i.e., each synthetic attribute) can be also be functionally decomposed and parallelized. This approach, however,

has a high internal data access overhead. Possible solutions include use of distributed shared memory and parallel I/O. Nevertheless, the break-even point for communications overhead is favorable, because the fitness function computations (for applications surveyed below) range from 5 minutes (for data sets on the order of 100 attributes and 25000 exemplars) to 75 minutes (for data sets on the order of 400 attributes and 100000 exemplars).

Applications of Genetic Constructive Induction in Large-Scale Data Mining

Record and Document Clustering (Information Retrieval)

The simple GA for attribute partitioning can be applied to knowledge discovery in very large databases. The purpose of constructive induction in these problems is to perform *change of representation* for supervised learning, thereby reducing the computational complexity of the learning problem given the transformed problem. For example self-organizing maps can be used [HW99] to produce multiple, *intermediate* training targets (new, constructed attributes) that are used to define a new supervised learning problem. This technique has been used at NCSA (using manual and non-genetic methods such as Kohonen's self-organizing maps, or SOM) to cluster sales transaction records, insurance policy records, and claims data, as well as technical natural language reports (repair documents, warranty documents, and patent literature). In current research, the simple GA and more sophisticated genetic methods for attribute synthesis in record clustering (especially for repair documents and patent literature) are being evaluated in a Java-based infrastructure for large-scale KDD.

Supervised Learning in Precision Agriculture

A family of real-world applications that involves highly heterogeneous time series data is that of monitoring problems in precision agriculture. Experiments using hierarchical mixtures have shown the feasibility of isolating multiple stochastic process types using a model selection wrapper [HR98b, Hs98]. These experiments were conducted using (subjective) weekly *crop condition* estimates from corn fields in Illinois (years 1985-1995). They show that the data is heterogeneous because it contains both an autoregressive pattern (linear increments in autocorrelation for the first 10 weeks of the growing season) and a moving average pattern (larger, unevenly spaced increments in autocorrelation). The autoregressive process, which can be represented by a time-delay model, expresses weather "memory" (correlating early and late drought); the moving average process, which can be represented by an exponential trace model, physiological damage from drought. Task decomposition can improve performance here, by isolating the AR and MA components for identification and application of the correct specialized architecture (a time delay neural network or simple recurrent network, respectively). For details of this experiment, the interested reader is referred to [Hs98].

Supervised Learning for Insurance Policy Classification

Finally, another real-world application is multi-attribute risk assessment (prediction of expected financial loss) using insurance policy data. The input data is partitioned using a state space search over subdivisions of attributes (this approach is an extension of existing work on attribute subset selection The supervised learning task is represented as a discrete classification (concept learning) problem over continuous-valued input. It can be systematically decomposed by partitioning the input attributes (or fields) based on prior information such as *typing* of attributes (e.g., geographical, automobile-specific demographics, driver-specific demographics, etc.). Preliminary experiments indicate that synthesis of *intra-type* attributes (such as paid loss, the sum of losses from different subcategories, and duration or membership, the difference between termination date and effective date of an insurance policy) and *inter-type* attributes (such as loss ratio) can be highly useful in supervised learning. This includes definition of new input attributes as well as intermediate target concepts [HW99].

Current and Future Work

Relevance Determination: Evaluating Attribute Quality

The credit assignment problem for synthetic attribute evaluation is complicated by the simultaneous question of which attributes are relevant. Current research addresses the problem of automatic relevance determination in series with attribute synthesis; future work will address how a classifier system can be built to optimize concurrently on both criteria.

Wrappers for Constraint Selection: Evaluating Quality of Domain Knowledge

The constraints themselves can be evaluated and the penalty weights adjusted, as is done in an advanced version of Raymer *et al*'s attribute extraction and selection system [RPG+97]. A future project that is part of the KDD infrastructure initiative at NCSA is the construction of a genetic wrapper for evaluating and refining domain knowledge in the form of constraints on attribute synthesis operators.

Scalable Computing Issues: Exploiting Functional (Task-Level) Parallelism

Finally, the current research and development for the KDD infrastructure is focused on taking advantage of functional parallelism in genetic attribute selection and partitioning. This is accomplished by dividing fitness evaluation tasks (application of supervised learning) among multiple high-performance processors.

Acknowledgements

This research was made possible by the Strategic Technologies Initiative and by the Strategic Partners Program at NCSA. The authors thank Loretta Auvil, John Martirano, Tom Redman, and David Tcheng, the other architects of the NCSA KDD infrastructure.

References

[BGH89] L. B. Booker, D. E. Goldberg, and J. H. Holland. Classifier Systems and Genetic Algorithms. *Artificial Intelligence*, 40:235-282, 1989.

[Ca99] E. Cantu-Paz. Personal communication.

[CS96] K. J. Cherkauer and J. W. Shavlik. Growing Simpler Decision Trees to Facilitiate Knowledge Discovery. In *Proceedings of the Second International Conference of Knowledge Discovery and Data Mining (KDD-96)*, Portland, OR, August, 1996.

[Do96] S. K. Donoho. *Knowledge-Guided Constructive Induction*. Ph.D. thesis, University of Illinois at Urbana-Champaign (Technical Report UIUC-DCS-R1970), July, 1996.

[DR95] S. K. Donoho and L. A. Rendell. Rerepresenting and restructuring domain theories: A constructive induction approach. *Journal of Artificial Intelligence Research*, 2:411-446, 1995.

[Gu91] G. H. Gunsch. *Opportunistic Constructive Induction: Using Fragments of Domain Knowledge to Guide Construction.* Ph.D. thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1991.

[Ha89] D. Haussler. Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. *Artificial Intelligence*, 36:177-221, 1989.

[Hs98] W. H. Hsu. *Time Series Learning With Probabilistic Network Composites*. Ph.D. thesis, University of Illinois at Urbana-Champaign (Technical Report UIUC-DCS-R2063). URL: http://www.ncsa.uiuc.edu/People/bhsu/thesis.html, August, 1998.

[HR98a] W. H. Hsu and S. R. Ray. A New Mixture Model for Concept Learning From Time Series (Extended Abstract). In *Proceedings of the Joint AAAI-ICML Workshop on AI Approaches to Time Series Problems*, pp. 42-43. Madison, WI, July, 1998.

[HR98b] W. H. Hsu and S. R. Ray. Quantitative Model Selection for Heterogeneous Time Series. In *Proceedings of the Joint AAAI-ICML Workshop on the Methodology of Applying Machine Learning*, pp. 8-12. Madison, WI, July, 1998.

[HR99] W. H. Hsu and S. R. Ray. Construction of Recurrent Mixture Models for Time Series Classification. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-99)*, Washington, DC.

[HW99] W. H. Hsu and M. Welge. Self-Organizing Systems for Knowledge Discovery in Databases. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-99)*, Washington, DC, to appear.

[JKP94] G. John, R. Kohavi, and K. Pfleger. Irrelevant Features and the Subset Selection Problem. In *Proceedings of the 11th International Conference on Machine Learning*, p. 121-129, New Brunswick, NJ. Morgan-Kaufmann, Los Altos, CA, 1994.

[Ko95] R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs.* Ph.D. thesis, Department of Computer Science, Stanford University, 1995.

[Ko98] R. Kohavi. *MineSet v2.6*, Silicon Graphics Incorporated, CA, 1998.

[KS96] R. Kohavi and D. Sommerfield. *MLC++: Machine Learning Library in C++, Utilities v2.0.* URL: http://www.sgi.com/Technology/mlc.

[KJ97] R. Kohavi and G. H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence, Special Issue on Relevance, 97*(1-2):273-324, 1997.

[KR92] K. Kira and L. A. Rendell. The Feature Selection Problem: Traditional Methods and a New Algorithm. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-92)*, p. 129-134, San Jose, CA. MIT Press, Cambridge, MA, 1992.

[Mi83] R. S. Michalski. A Theory and Methodology of Inductive Learning. *Artificial Intelligence*, 20(2):111-161, 1983.

[Mi97] T. M. Mitchell. *Machine Learning.* McGraw-Hill, New York, NY, 1997.

[RH98] S. R. Ray and W. H. Hsu. Self-Organized-Expert Modular Network for Classification of Spatiotemporal Sequences. *Journal of Intelligent Data Analysis,* October, 1998.

[RPG+97] M. Raymer, W. Punch, E. Goodman, P. Sanschagrin, and L. Kuhn, Simultaneous Feature Extraction and Selection using a Masking Genetic Algorithm, In *Proceedings of the 7th International Conference on Genetic Algorithms*, pp. 561-567, San Francisco, CA, July, 1997.

[RS90] L. A. Rendell and R. Seshu. Learning Hard Concepts through Constructive Induction: Framework and Rationale. *Computational Intelligence*, 6(4):247-270, 1990.