

Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey

SREERAMA K. MURTHY

murthy@scr.siemens.com

Siemens Corporate Research, Princeton, NJ 08540, USA

Abstract. Decision trees have proved to be valuable tools for the description, classification and generalization of data. Work on constructing decision trees from data exists in multiple disciplines such as statistics, pattern recognition, decision theory, signal processing, machine learning and artificial neural networks. Researchers in these disciplines, sometimes working on quite different problems, identified similar issues and heuristics for decision tree construction. This paper surveys existing work on decision tree construction, attempting to identify the important issues involved, directions the work has taken and the current state of the art.

Keywords: classification, tree-structured classifiers, data compaction

1. Introduction

Advances in data collection methods, storage and processing technology are providing a unique challenge and opportunity for automated data exploration techniques. Enormous amounts of data are being collected daily from major scientific projects (e.g., Human Genome Project, the Hubble Space Telescope, Geographical Information Systems), from stocks trading, from hospital information systems, from computerized sales records and other sources. In addition, researchers and practitioners from more diverse disciplines than ever before are attempting to use automated methods to analyze their data. As the quantity and variety of data available to data exploration methods increases, there is a commensurate need for robust, efficient and versatile data exploration methods.

Decision trees are a way to represent rules underlying data with hierarchical, sequential structures that recursively partition the data. A decision tree can be used for data exploration in one or more of the following ways: ¹

- **Description:** To reduce a volume of data by transforming it into a more compact form which preserves the essential characteristics and provides an accurate summary.
- **Classification:** Discovering whether the data contains well-separated classes of objects, such that the classes can be interpreted meaningfully in the context of a substantive theory.
- **Generalization:** Uncovering a mapping from independent to dependent variables that is useful for predicting the value of the dependent variable in the future.

Automatic construction of rules in the form of decision trees has been attempted virtually in all disciplines in which data exploration methods have been developed.

It has been traditionally developed in the fields of statistics, engineering (pattern recognition) and decision theory (decision table programming). Recently renewed interest has been generated by research in artificial intelligence (machine learning) and the neurosciences (neural networks). Though the terminology and emphases differ from discipline to discipline, there are many similarities in the methodology.

Decision trees automatically constructed from data have been used successfully in many real-world situations. Their effectiveness has been compared widely to other automated data exploration methods and to human experts. Several advantages of decision tree-based classification have been pointed out.

- Knowledge acquisition from pre-classified examples circumvents the bottleneck of acquiring knowledge from a domain expert.
- Tree methods are exploratory as opposed to inferential. They are also non-parametric. As only a few assumptions are made about the model and the data distribution, trees can model a wide range of data distributions.
- Hierarchical decomposition implies better use of available features and computational efficiency in classification.
- As opposed to some statistical methods, tree classifiers can treat uni-modal as well as multi-modal data in the same fashion.
- Trees can be used with the same ease in deterministic as well as incomplete problems. (In deterministic domains, the dependent variable can be determined perfectly from the independent variables, whereas in incomplete problems, it cannot be.)
- Trees perform classification by a sequence of simple, easy-to-understand tests whose semantics are intuitively clear to domain experts. The decision tree formalism itself is intuitively appealing.

For these and other reasons, decision tree methodology can provide an important tool in every data mining researcher/practitioner's tool box. In fact, many existing data mining products are based on constructing decision trees from data.²

In order to gain optimal benefit from the existing methods, or to develop improved algorithms, it is crucial to have an understanding of the existing work on this subject. Some existing decision tree work lacks step-by-step progress. Researchers and system developers often tried *ad hoc* variations of the basic methodology until they found something that "worked" or was "interesting." Due to this practice, one often encounters instances of redundant effort. Although it is not the intent of the current paper to point out specific instances of redundant work, a careful reader may notice several such examples. (The *ad hoc* nature is obviously not true of all work on decision trees. A good counter-example is Ross Quinlan's work over the years. It progresses in a series of carefully chosen steps that advance our understanding of decision trees.)

In spite of a large body of existing work and substantial practical success of this technique, there exist no comprehensive, multi-disciplinary surveys of results on decision tree construction from data. (See Section 2 for a discussion of existing surveys.) The current paper attempts to fill this gap. We summarize significant

results related to automatically constructing decision trees from data, from fields such as pattern recognition, statistics, decision theory, machine learning, mathematical programming and neural networks. We maintain the conciseness of this survey using the following guidelines and limitations.

- We do not attempt a tutorial overview of any specific topics. Our main emphasis is to trace the directions that decision tree work has taken. For this reason, readers with a basic knowledge of automatic decision tree construction methodology may benefit more from this survey than readers who are completely new to trees.
- We avoid repeating many of the references from three existing surveys [292, 259, 320]. This is partly because the above surveys had different emphases than ours, as outlined in Section 2.
- We limit our references to refereed journals, published books and recent conferences.
- Our coverage of decision tree applications falls far short of being comprehensive; it is merely illustrative. Same is true of our coverage of comparisons between trees and other techniques.

1.1. Outline and survey overview

We briefly outline and motivate below several issues involved in constructing decision trees and using them. Along with each issue, we mention the corresponding section in the survey. This section aims to establish a structural organization for the large body of existing literature on trees. We use below terminology from machine learning and statistics. Alternative terminology may be found in Section 1.3.

- Greedy top-down construction is the most commonly used method for tree growing today (see Section 5.10 for exceptions). A hierarchical model can be constructed top-down, starting from the entire data, somehow partitioning it into subsets, and recursing the partitioning procedure. A description of tree growing then reduces to a description of techniques for splitting data into meaningful subsets. Section 3 reviews dozens of “splitting rules” that have been proposed in the literature, their classification and comparative evaluations. This section also covers in detail multivariate splitting rules.
- Whether a model is intended for description, classification or generalization, we would like it to be “better” than the data, capturing only the true characteristics of the data but not the noise and randomness. In the context of trees, this concern translates into the problem of finding the *right sized* trees. Techniques to find right sized trees, including pruning, their evaluations and comparisons are the topic of Section 4. When more than one tree can describe a data set perfectly, we need metrics to quantify the “goodness” of trees. Tree quality measures proposed in the literature are summarized in Section 5.9.

- Sample size versus dimensionality of a data set greatly influences the quality of trees constructed from it. Work analysing this influence is reviewed in Section 5.1. This section also covers methods that preprocess the data before inducing trees, such as feature subset selection (removing redundant or correlated features), composite feature construction and data subsampling.
- Most real-world data is complex and imperfect. Variable costs are associated with different features and classes, and missing feature values are the rule not the exception. We review the work dealing with these two issues in Sections 5.2 and 5.3 respectively.
- The shortcomings of decision tree models, as well as solutions to alleviate them, have been extensively reported in the literature. Greedy splitting heuristics are efficient and adequate for most applications, but are essentially suboptimal. In situations where processing resources are not as important as the optimality of the result, several ways to improving upon greedy induction exist (Section 5.4). Crisp decisions that decision trees usually output may not be adequate or useful in some settings. Techniques to use tree models as probability estimators have been suggested (Section 5.5). Individual decision trees have high variance in terms of generalization accuracy, so many authors have suggested combining the results from multiple decision trees (Section 5.6). Trees cause data fragmentation, which reduces the probabilistic significance of near-leaf nodes. A solution to this is the use of *soft* splits (Section 5.8).
- We discuss many other miscellaneous aspects of tree construction (Section 5.10), including incremental tree construction (Section 5.7).
- Some natural questions to ask in the context of tree construction are “is it possible to build optimal trees?”, “exactly how good is a specific algorithm?”, etc. Researchers have theoretically and empirically analyzed the tree construction methodology. Section 6 reviews this work in detail, covering NP-completeness results and analyses of biases in tree induction.
- Section 7 is devoted to the practical promise of decision trees. We discuss recent “real world” applications, available software packages and comparisons with alternative data analysis techniques, all of which establish decision trees as versatile and effective data analysis tools.

The above binning interestingly brings out a paucity of the work on decision trees. By dividing model construction into individual subtasks, we risk losing track of the overall purpose of this exercise. Apparent improvements in individual steps are *not* guaranteed to lead to better algorithms overall. Splitting rules are a good example. Splitting rules have to be defined, evaluated and improved in the broader context of the tree construction method. Otherwise, they are reduced to mere *ad hoc* greedy heuristics. It is not surprising that most existing splitting rules are functionally equivalent.

The author acknowledges a shortcoming of this organization. Papers dealing with more than one topic are either listed multiple times or their mention is omitted from

some places. A good example is [44] which has relevance to many of the issues we address, and is referenced repeatedly under Sections 5.4, 5.5, 5.6 and 5.10.

The next section (1.2) introduces briefly the basic concepts involved in decision tree construction. Section 1.3 discusses alternative terminology. Section 2 summarizes high level pointers, mentioning existing surveys, text books and historical origins. Sections 3, 4, 5, 6 and 7 together comprise the survey whose organization is described in detail above. Section 8 concludes the paper with some general comments.

1.2. Basics of decision trees

Readers completely unfamiliar with decision trees should refer to [320], Section II for a good summary of basic definitions. A decision tree is constructed from a *training set*, which consists of *objects*. Each object is completely described by a set of *attributes* and a *class* label. Attributes can have *ordered* (e.g., real) or *unordered* (e.g., Boolean) values. The *concept* underlying a data set is the true mapping between the attributes and the class. A *noise-free* training set is one in which all the objects are “generated” using the underlying concept.

A decision tree contains zero or more *internal* nodes and one or more *leaf* nodes. All internal nodes have two or more *child* nodes.³ All internal nodes contain *splits*, which test the value of an expression of the attributes. *Arcs* from an internal node t to its children are labeled with distinct outcomes of the test at t . Each leaf node has a class label associated with it.⁴

The task of constructing a tree from the training set has been called *tree induction*, *tree building* and *tree growing*. Most existing tree induction systems proceed in a greedy top-down fashion. (Section 5.10 lists exceptions). Starting with an empty tree and the entire training set, some variant of the following algorithm is applied until no more splits are possible.

1. If all the training examples at the current node t belong to category c , create a leaf node with the class c .
2. Otherwise, score each one of the set of possible splits \mathcal{S} , using a *goodness measure*.
3. Choose the best split s^* as the test at the current node.
4. Create as many child nodes as there are distinct outcomes of s^* . Label edges between the parent and child nodes with outcomes of s^* , and partition the training data using s^* into the child nodes.
5. A child node t is said to be *pure* if all the training samples at t belong to the same class. Repeat the previous steps on all impure child nodes.

Discrimination is the process of deriving classification rules from samples of classified objects, and *classification* is applying the rules to new objects of unknown class [138]⁵. Decision trees have been used for discrimination as well as classification.

An object X is classified by passing it through the tree starting at the root node. The test at each internal node along the path is applied to the attributes of X , to

determine the next arc along which X should go down. The label at the leaf node at which X ends up is output as its classification. An object is *misclassified* by a tree if the classification output by the tree is not the same as the object's correct class label. The proportion of objects correctly classified by a decision tree is known as its *accuracy*, whereas the proportion of misclassified objects is the *error*.

1.3. Terminology

Structures similar to decision trees have been called classification trees, branched testing sequences, discriminant trees and identification keys. Training sets consist of objects, also known as samples, observations, examples or instances. Attributes have been referred to as features, predictors or independent variables. In an ordered attribute space, a decision tree imposes a partitioning that can be geometrically represented as a collection of hyper-surfaces and regions. Much of the work on decision trees uses only a specific type of surface, namely hyper-planes. (For exceptions, see the Neural Trees and Other Methods paragraphs in Section 3.2.) For this reason, splits are often referred to as hyper-planes, attributes as dimensions and objects as points.

Category or dependent variable is the same as class label. Ordered domains are equivalent to or comprise continuous, integer, real-valued and monotonous domains. Unordered domains have categorical, discrete or free variables. Internal nodes are the same as non-terminals or test nodes. Leaf nodes are referred to as the terminal nodes or decision nodes. Goodness measures are also known as feature evaluation criteria, feature selection criteria, impurity measures or splitting rules.

2. High level pointers

A decision tree performs multistage hierarchical decision making. For a general rationale for multistage classification schemes and a categorization of such schemes, see [174].

2.1. Origins

Work on decision tree induction in statistics began due to the need for exploring survey data [103]. Statistical programs such as AID [346], MAID [124], THAID [260] and CHAID [176] built binary segmentation trees aimed towards unearthing the interactions between predictor and dependent variables.

Pattern recognition work on decision trees was motivated by the need to interpret images from remote sensing satellites such as LANDSAT in the 1970s [350].

Decision trees in particular, and induction methods in general, arose in machine learning to avoid the knowledge acquisition bottleneck [101] for expert systems.

In sequential fault diagnosis, the inputs are a set of possible tests with associated costs and a set of system states with associated prior probabilities. One of the states is a "fault-free" state and the other states represent distinct faults. The aim

is to build a test algorithm that unambiguously identifies the occurrence of any system state using the given tests, while minimizing the total cost. The testing algorithms normally take the form of decision trees or AND/OR trees [368, 291]. Many heuristics used to construct decision trees are used for test sequencing also.

2.2. *Treatises and surveys*

An overview of work on decision trees in the pattern recognition literature can be found in [76]. A high level comparative perspective on the classification literature in pattern recognition and artificial intelligence can be found in [53]. Tree induction from a statistical perspective, as it is popularly used today, is reviewed in Breiman *et al.*'s excellent book *Classification and Regression Trees* [31]. For a review of earlier statistical work on hierarchical classification, see [103]. A majority of work on decision trees in machine learning is an offshoot of Breiman *et al.*'s work and Quinlan's ID3 algorithm [301]. Quinlan's book on C4.5 [306], although specific to his tree building program, provides an outline of tree induction methodology from a machine learning perspective.

Payne and Preece [292] surveyed results on constructing *taxonomic identification keys*, in a paper that attempted "a synthesis of a large and widely-dispersed literature" from fields such as biology, pattern recognition, decision table programming, machine fault location, coding theory and questionnaire design. Taxonomic identification keys are tree structures that have one object per leaf and for which the set of available tests (splits) is pre-specified. The problem of constructing identification keys is not the same as the problem of constructing decision trees from data, but many common concerns exist, such as optimal key construction and choosing good tests at tree nodes.

Moret [259] provided a tutorial overview of the work on representing Boolean functions as decision trees and diagrams. He summarized results on constructing decision trees in discrete variable domains. Although Moret mentions some pattern recognition work on constructing decision trees from data, this was not his primary emphasis.

Safavin and Landgrebe [320] surveyed the literature on decision tree classifiers, almost entirely from a pattern recognition perspective. This survey had the aim of bringing the disparate issues in decision tree classifiers together, providing a more unified view, and cautioning the "casual" users about the pitfalls of each method.

The current paper differs from the above surveys in the following ways.

- A substantial body of work that has been done after the existing surveys were written (e.g., almost all the machine learning work on tree construction) is covered. Some topics that were not discussed in the existing surveys (e.g., multivariate trees, NP-completeness) are covered.
- This paper brings into a common organization decision tree work in multiple disciplines.
- Our main emphasis is on automatically constructing decision trees for parsimonious descriptions of, and generalization from, data. (In contrast, for example,

the main emphasis of [259] was on representing Boolean functions as decision trees.)

2.3. *What is not covered*

In recent years, there has been a growing amount of work in Computational Learning Theory (COLT), on matters related to decision tree induction. We cover very little of this work in the survey, primarily due to the author's ignorance. Proceedings of the annual COLT conferences and International Conferences on Machine Learning (ICML) are good starting points to explore this work. A few good papers, to get a flavor for this work, are [169, 285, 177, 178, 148].

Work on learning Bayesian or inference networks from data is closely related to automatic decision tree construction. There are an increasing number of papers on the former topic, although the similarities with tree induction are usually not pointed out. For a good discussion of decision tree induction from a Bayesian networks point of view, see [42]. For a good introduction to the literature on learning Bayesian networks, see [45].

Work on automatic construction of hierarchical structures from data in which the dependent variable is unknown (*unsupervised* learning), present in fields such as cluster analysis [93], machine learning (e.g., [105, 121]) and vector quantization [122] is not covered. Work on hand-constructed decision trees (common in medicine) is also not considered. We do not discuss regression trees. There is a rich body of literature on this topic which shares many issues with the decision tree literature. For an introduction, see [31, 55]. We do not discuss binary decision diagrams and decision graphs [188]. We do not discuss patents.⁶

3. Finding splits

To build a decision tree, it is necessary to find at each internal node a test for splitting the data into subsets. In case of univariate trees, finding a split amounts to finding the attribute that is the most “useful” in discriminating the input data, and finding a decision rule using the attribute. In case of multivariate trees, finding a split can be seen as finding a “composite” feature, a combination of existing attributes that has good discriminatory power. In either case, a basic task in tree building is to rank features (single or composite) according to their usefulness in discriminating the classes in the data.

3.1. *Feature evaluation rules*

In pattern recognition and statistics literature, features are typically ranked using *feature evaluation rules*, and the single best feature *or* a good feature subset are chosen from the ranked list. In machine learning, however, feature evaluation rules are used mainly for picking the single best feature at every node of the decision tree.

Methods used for selecting a good subset of features are typically quite different. We will postpone the discussion of feature subset selection methods to Section 5.1.1.

Ben Bassat [19] divides feature evaluation rules into three categories: rules derived from information theory, rules derived from distance measures and rules derived from dependence measures. These categories are sometimes arbitrary and not distinct. Some measures belonging to different categories can be shown to be equivalent. Many can be shown to be approximations of each other.

Rules derived from information theory: Examples of this variety are rules based on Shannon’s entropy.⁷ Tree construction by maximizing global *mutual information*, i.e., by expanding tree nodes that contribute to the largest gain in average mutual information of the whole tree, is explored in pattern recognition [126, 333, 351].⁸ Tree construction by locally optimizing *information gain*, the reduction in entropy due to splitting each individual node, is explored in pattern recognition [142, 372, 49, 139], in sequential fault diagnosis [368] and in machine learning [301]. Mingers [246] suggested the G-statistic, an information theoretic measure that is a close approximation to χ^2 distribution, for tree construction as well as for deciding when to stop. De Merckt [367] suggested an attribute selection measure that combined geometric distance with information gain, and argued that such measures are more appropriate for numeric attribute spaces.

Rules derived from distance measures: “Distance” here refers to the distance between class probability distributions. The feature evaluation criteria in this class measure separability, divergence or discrimination between classes. A popular distance measure is the Gini index of diversity⁹, which has been used for tree construction in statistics [31], pattern recognition [119] and sequential fault diagnosis [291]. Breiman *et al.* pointed out that the Gini index has difficulty when there are a relatively large number of classes, and suggested the *twoing rule* [31] as a remedy. Taylor and Silverman [355] pointed out that the Gini index emphasizes equal sized offspring and purity of both children. They suggested a splitting criterion, called mean posterior improvement (MPI), that emphasizes exclusivity between offspring class subsets instead.

Bhattacharya distance [218], Kolmogorov-Smirnoff distance [113, 316, 143] and the χ^2 statistic [17, 141, 246, 389, 380] are some other distance-based measures that have been used for tree induction. Though the Kolmogorov-Smirnoff distance was originally proposed for tree induction in two-class problems [113, 316], it was subsequently extended to multiclass domains [143]. Class separation-based metrics developed in the machine learning literature [98, 388] are also distance measures. A relatively simplistic method for estimating class separation, which assumes that the values of each feature follow a Gaussian distribution in each class, was used for tree construction in [227].

Rules derived from dependence measures: These measure the statistical dependence between two random variables. All dependence-based measures can be interpreted as belonging to one of the above two categories [19].

There exist many attribute selection criteria that do not clearly belong to any category in Ben Bassat’s taxonomy. Gleser and Collen [126] and Talmon [351] used a combination of mutual information and χ^2 measures. They first measured the

gain in average mutual information $I(T_i)$ due to a new split T_i , and then quantified the probability $P(I(T_i))$ that this gain is due to chance, using χ^2 tables. The split that minimized $P(I(T_i))$ was chosen by these methods. A permutation statistic was used for univariate tree construction for 2-class problems in [214]. The main advantage of this statistic is that, unlike most of the other measures, its distribution is independent of the number of training instances. As will be seen in Section 4, this property provides a natural measure of when to stop tree growth.

Measures that use the *activity* of an attribute have been explored for tree construction [258, 252]. The activity of a variable is equal to the testing cost of the variable times the *a priori* probability that it will be tested. The computational requirements for computing activity are the same as those for the information-based measures. Quinlan and Rivest [309] suggested the use of Risannen’s minimum description length [314] for deciding which splits to prefer over others and also for pruning. Kalkanis [172] pointed out that measures like information gain and Gini index are all concave (i.e., they never report a worse goodness value after trying a split than before splitting), so there is no natural way of assessing where to stop further expansion of a node. As a remedy, Kalkanis suggested the use of the upper bounds in the confidence intervals for the misclassification error as an attribute selection criterion.¹⁰

The total number of misclassified points has been explored as a selection criterion by many authors. Two examples are Heath’s *sum minority* [147] and Lubinsky’s *inaccuracy* [223, 224]. The CART book [31], among others, discuss why this is not a good measure for tree induction. Additional tricks are needed to make this measure useful [223, 269]. Heath [147] also used *max minority* (maximum of the number of misclassified points on two sides of a binary split) and *sum of impurities* (which assigns an integer to each class and measures the variance between class numbers in each partition) [147, 269]. An almost identical measure to sum of impurities was used earlier in the Automatic Interaction Detection (AID) program [103].

Most of the above feature evaluation criteria assume no knowledge of the probability distribution of the training objects. The optimal decision rule at each tree node, a rule that minimizes the overall error probability, is considered in [204, 205, 206] assuming that complete probabilistic information about the data is known. Shang and Breiman [335] argue that trees built from probability distributions (which in turn are inferred from attribute values) are more accurate than trees built directly from attribute values. Grewe and Kak [133] proposed a method for building multi-attribute hash tables using decision trees for object localization and detection in 3D. Their decision trees are also built from probability distributions of attributes rather than the attribute values themselves. Pal *et al.* [286] recently proposed a variant of the ID3 algorithm for real data, in which tests at an internal node are found using genetic algorithms.

3.1.1. Evaluations, Comparisons Given the large number of feature evaluation rules, a natural concern is to measure their relative effectiveness for constructing “good” trees. Evaluations in this direction, in statistics, pattern recognition and machine learning, have been predominantly empirical in nature, though there have

been a few theoretical evaluations. (We defer the discussion of the latter to Section 6.)

In spite of a large number of comparative studies, very few so far have concluded that a particular feature evaluation rule is significantly better than others. A majority of studies have concluded that there is not much difference between different measures. This is to be expected as induction *per se* can not rigorously justify performance on unseen instances.¹¹ A lot of splitting rules are similar from a functional perspective. Splitting rules are essentially *ad hoc* heuristics for evaluating the strength of dependence between attributes and the class. Comparisons of individual methods may still be interesting if they enlighten the reader about which metric should be used in what situations.

Baker and Jain [15] reported experiments comparing eleven feature evaluation criteria and concluded that the feature rankings induced by various rules are very similar. Several feature evaluation criteria, including Shannon's entropy and divergence measures, are compared using simulated data in [18], on a sequential, multi-class classification problem. The conclusions are that no feature selection rule is consistently superior to the others, *and* that no specific strategy for alternating different rules seems to be significantly more effective. Breiman *et al.* [31] conjectured that decision tree design is rather insensitive to any one from a large class of splitting rules, and it is the stopping rule that is crucial. Mingers [248] compared several attribute selection criteria, and concluded that tree quality doesn't seem to depend on the specific criterion used. He even claimed that random attribute selection criteria are as good as measures like information gain [301]. This later claim was refuted in [41, 219], where the authors argued that random attribute selection criteria are prone to overfitting, and also fail when there are several noisy attributes.

Miyakawa [252] compared three activity-based measures, Q , O and *loss*, both analytically and empirically. He showed that Q and O do not chose non-essential variables at tree nodes, and that they produce trees that are 1/4th the size of the trees produced by *loss*. Fayyad and Irani [98] showed that their measure C-SEP, performs better than Gini index [31] and information gain [301] for specific types of problems.

Several researchers [141, 301] pointed out that information gain is biased towards attributes with a large number of possible values. Mingers [246] compared information gain and the χ^2 statistic for growing the tree as well as for stop-splitting. He concluded that χ^2 corrected information gain's bias towards multivalued attributes, however to such an extent that they were never chosen, and the latter produced trees that were extremely deep and hard to interpret. Quinlan [306] suggested *gain ratio* as a remedy for the bias of information gain. Mantaras [233] argued that gain ratio had its own set of problems, and suggested using information theory-based *distance* between partitions for tree construction. He formally proved that his measure is not biased towards multiple-valued attributes. However, White and Liu [380] present experiments to conclude that information gain, gain ratio *and* Mantaras' measure are worse than a χ^2 based statistical measure, in terms of their bias towards multiple-valued attributes. A hyper-geometric function is proposed as a

means to avoid the biases of information gain, gain ratio and χ^2 metrics by Martin [235]. Martin proposed and examined several alternatives in Quinlan's measures (including distance, orthogonality, a Beta function and two chi-squared tests). In a different paper [236], Martin proved that the time complexity of induction and post-processing is exponential in tree height in the worst case and, under fairly general conditions, in the average case. This puts a premium on designs which tend to produce shallower trees (e.g., multi-way rather than binary splits and selection criteria which prefer more balanced splits). Kononenko [193] pointed out that Minimum Description Length-based feature evaluation criteria have the least bias towards multi-valued attributes.

3.2. Multivariate splits

Decision trees have been popularly univariate, i.e., they use splits based on a single attribute at each internal node. Even though several methods have been developed in the literature for constructing multivariate trees, this body of work is not as well-known.

Most of the work on multivariate splits considers linear (oblique) trees. These are trees which have tests based on a linear combination of the attributes at some internal nodes. The problem of finding an optimal linear split (optimal with respect to any of the feature evaluation measures in Section 3.1) is more difficult than that of finding the optimal univariate split. In fact, finding optimal linear splits is known to be intractable for some feature evaluation rules (see Section 6.1), so heuristic methods are required for finding good, albeit suboptimal, linear splits. Methods used in the literature for finding good linear tests include linear discriminant analysis, hill climbing search, linear programming, perceptron training and others.

Linear Discriminant Trees: Several authors have considered the problem of constructing tree-structured classifiers that have linear discriminants [85] at each node. You and Fu [386] used a linear discriminant at each node in the decision tree, computing the hyper-plane coefficients using the Fletcher-Powell descent method [107]. Their method requires that the best set of features at each node be pre-specified by a human. Friedman [113] reported that applying Fisher's linear discriminants, instead of atomic features, at some internal nodes was useful in building better trees. Qing-Yun and Fu [298] also describe a method to build linear discriminant trees. Their method uses multivariate stepwise regression to optimize the structure of the decision tree as well as to choose subsets of features to be used in the linear discriminants. More recently, use of linear discriminants at each node is considered by Loh and Vanichsetakul [220]. Unlike in [386], the variables at each stage are appropriately chosen in [220] according to the data and the type of splits desired. Other features of the tree building algorithm in [220] are: (1) it yields trees with univariate, linear combination or linear combination of polar coordinate splits, and (2) allows both ordered and unordered variables in the same linear split. Use of linear discriminants in a decision tree is considered in the remote sensing literature in [158]. A method for building linear discriminant classification trees, in

which the user can decide at each node what classes need to be split, is described in [357]. John [167] recently considered linear discriminant trees in the machine learning literature. An extension of linear discriminants are linear machines [276], which are linear structures that can discriminate between multiple classes. In the machine learning literature, Utgoff *et al.* explored decision trees that used linear machines at internal nodes [35, 83].

Locally Opposed Clusters of Objects: Sklansky and his students developed several piecewise linear discriminants based on the principle of locally opposed clusters of objects. Wassel and Sklansky [374, 344] suggested a procedure to train a linear split to minimize the error probability. Using this procedure, Sklansky and Michelotti [343] developed a system to induce a piece-wise linear classifier. Their method identifies the closest-opposed pairs of clusters in the data, and trains each linear discriminant locally. The final classifier produced by this method is a piecewise linear decision surface, not a tree. Foroutan [110] discovered that the re-substitution error rate of optimized piece-wise linear classifiers is nearly monotonic with respect to the number of features. Based on this result, Foroutan and Sklansky [111] suggest an effective feature selection procedure for linear splits that uses zero-one integer programming. Park and Sklansky [290, 289] describe methods to induce linear tree classifiers and piece-wise linear discriminants. The main idea in these methods is to find hyper-planes that cut a maximal number of *Tomek* links. Tomek links of a data set connect opposed pairs of data points for which the circle of influence between the points doesn't contain any other points.

Hill Climbing Methods: CART's use of linear combinations of attributes ([31], Chapter 5) is well-known. This algorithm uses heuristic hill climbing and backward feature elimination to find good linear combinations at each node. Murthy *et al.* [268, 269] described significant extensions to CART's linear combinations algorithm, using randomized techniques.

Perceptron Learning: A perceptron is a linear function neuron [249, 137] which can be trained to optimize the sum of distances of the misclassified objects to it, using a convergent procedure for adjusting its coefficients. *Perceptron trees*, which are decision trees with perceptrons just above the leaf nodes, were discussed in [362]. Decision trees with perceptrons at all internal nodes were described in [365, 334].

Mathematical Programming: Linear programming has been used for building adaptive classifiers since late 1960s [156]. Given two possibly intersecting sets of points, Duda and Hart [85] proposed a linear programming formulation for finding the split whose distance from the misclassified points is minimized. More recently, Mangasarian and Bennett used linear and quadratic programming techniques to build machine learning systems in general and decision trees in particular [232, 22, 20, 230, 21]. Use of zero-one integer programming for designing vector quantizers can be found in [217]. Brown and Pittard [37] also employed linear programming for finding optimal multivariate splits at classification tree nodes. Almost all the above papers attempt to minimize the distance of the misclassified points from the decision boundary. In that sense, these methods are more similar to perceptron training methods [249], than to decision tree splitting criteria. Mangasarian [231]

described a linear programming formulation to minimize the number of misclassified points instead of the geometric distance.

Neural Trees: In the neural networks community, many researchers have considered hybrid structures between decision trees and neural nets. Though these techniques were developed as neural networks whose structure could be automatically determined, their outcome can be interpreted as decision trees with non-linear splits. Techniques very similar to those used in tree construction, such as information theoretic splitting criteria and pruning, can be found in neural tree construction also. Examples of this work include [127, 342, 32, 59, 150, 324, 72]. Sethi [331] described a method for converting a univariate decision tree into a neural net and then retraining it, resulting in tree structured *entropy nets* with sigmoidal splits. An extension of entropy nets, that converts linear decision trees into neural nets was described in [288]. Decision trees with small multi-layer networks at each node, implementing nonlinear, multivariate splits, were described in [134]. Jordan and Jacobs [170] described hierarchical parametric classifiers with small “experts” at internal nodes. Training methods for tree structured Boltzmann machines are described in [325].

Other Methods: Use of polynomial splits at tree nodes is explored in decision theory [330]. In Machine Learning, recently a method has been suggested [165] for “manufacturing” second or higher degree features and then inducing linear splits on these complex features to get non-linear decision trees. In information theory, Gelfand and Ravishanker [118] describe a method to build a tree structured filter that has linear processing elements at internal nodes. Heath *et al.* [147, 145] used simulated annealing to find the best oblique split at each tree node. Chai *et al.* [52] recently suggested using genetic algorithms to search for linear splits at non-terminal nodes in a tree. Lubinsky [225, 224] attempted bivariate trees, trees in which some functions of two variables can be used as tests at internal nodes. Lubinsky considered the use of linear cuts, corner cuts and rectangular cuts, using ordered and unordered variables.

3.3. Ordered vs. unordered attributes

The fields of pattern recognition and statistics historically have considered ordered or numeric attributes as the default. This seems natural considering application domains such as spectral analysis and remote sensing [350]. In these fields, special techniques [332] were developed to accommodate discrete attributes into what were primarily algorithms for ordered attributes. Fast methods for splitting multiple valued categorical variables are described in [57].

In machine learning, a subfield of Artificial Intelligence, which in turn has been dominated by symbolic processing, many tree induction methods (e.g., [299] were originally developed for categorical attributes. The problem of incorporating continuous attributes into these algorithms is considered subsequently. The problem of meaningfully discretizing a continuous dimension is considered in [99, 181, 367, 263]. Fast methods for splitting a continuous dimension into more than two ranges is considered in the machine learning literature [100, 115].¹² An extension to ID3 [301]

that distinguishes between attributes with unordered domains and attributes with linearly ordered domains is suggested in [60]. Quinlan [308] recently discussed improved ways of using continuous attributes with C4.5.

4. Obtaining the right sized trees

See Breslow and Aha's recent survey [33] on simplifying decision trees for a detailed account of the motivation for tree simplification and existing solution approaches.

One of the main difficulties of inducing a recursive partitioning structure is knowing when to stop. Obtaining the "right" sized trees is important for several reasons, which depend on the size of the classification problem [119]. For moderate sized problems, the critical issues are generalization accuracy, honest error rate estimation and gaining insight into the predictive and generalization structure of the data. For very large tree classifiers, the critical issue is optimizing structural properties such as height and balance [372, 50].

Breiman *et al.* [31] pointed out that tree quality depends more on good stopping rules than on splitting rules. Effects of noise on generalization are discussed in [275, 186]. Overfitting avoidance as a specific bias is studied in [383, 326]. Effect of noise on classification tree construction methods is studied in the pattern recognition literature in [353].

Several techniques have been suggested for obtaining the right sized trees. The most popular of these is *pruning*, whose discussion we will defer to Section 4.1. The following are some alternatives to pruning that have been attempted.

- Restrictions on minimum node size: A node is not split if it has smaller than k objects, where k is a parameter to the tree induction algorithm. This strategy, which is known to be not robust, is used in some early methods [113].
- Two stage search: In this variant, tree induction is divided into two subtasks: first, a good structure for the tree is determined; then splits are found at all the nodes.¹³ The optimization method in the first stage may or may not be related to that used in the second stage. Lin and Fu [218] use K -means clustering for both stages, whereas Qing-Yun and Fu [298] use multi-variate stepwise regression for the first stage and linear discriminant analysis for the second stage.
- Thresholds on Impurity: In this method, a threshold is imposed on the value of the splitting criterion, such that if the splitting criterion falls below (above) the threshold, tree growth is aborted. Thresholds can be imposed on local (i.e., individual node) goodness measures or on global (i.e., entire tree) goodness. The former alternative is used in [126, 316, 300, 235] and the latter in [333]. A problem with the former method is that the value of most splitting criteria (Section 3.1) varies with the size of the training sample. Imposing a single threshold that is meaningful at all nodes in the tree is not easy and may not even be possible. Some feature evaluation rules, whose distribution does *not* depend on the number of training samples (i.e., a goodness value of k would have the same significance anywhere in the tree) have been suggested in the

literature [214, 389, 172]. Martin and Hirschberg [236] argue that pre-pruning or simple pruning is linear in tree height, contrasted to the exponential growth of more complex operations. The key factor that influences whether simple pruning will suffice is whether the split selection and pruning heuristics are the same and unbiased.

- Trees to rules conversion: Quinlan [302, 306] gave efficient procedures for converting a decision tree into a set of production rules. Simple heuristics to generalize and combine the rules generated from trees can act as a substitute for pruning for Quinlan’s univariate trees.
- Tree reduction: Cockett and Herrera [61] suggested a method to reduce an arbitrary binary decision tree to an “irreducible” form, using discrete decision theory principles. Every irreducible tree is optimal with respect to some expected testing cost criterion, and the tree reduction algorithm has the same worst-case complexity as most greedy tree induction methods.

4.1. Pruning

Pruning, the method most widely used for obtaining right sized trees, was proposed by Breiman *et al.* ([31], Chapter 3). They suggested the following procedure: build the complete tree (a tree in which splitting no leaf node further will improve the accuracy on the training data) and then remove subtrees that are not contributing significantly towards generalization accuracy. It is argued that this method is better than stop-splitting rules, because it can compensate, to some extent, for the sub-optimality of greedy tree induction. For instance, if there is very good node T_2 a few levels below a not-so-good node T_1 , a stop-splitting rule will stop tree growth at T_1 , whereas pruning may give a high rating for, and retain, the whole subtree at T_1 . Kim and Koehler [183] analytically investigate the conditions under which pruning is beneficial for accuracy. Their main result states that pruning is more beneficial with increasing skewness in class distribution and/or increasing sample size.

Breiman *et al.*’s pruning method [31] *cost complexity* pruning (a.k.a. weakest link pruning or error complexity pruning) proceeds in two stages. In the first stage, a sequence of increasingly smaller trees are built on the training data. In the second stage, one of these trees is chosen as the pruned tree, based on its classification accuracy on a *pruning set*. Pruning set is a portion of the training data that is set aside exclusively for pruning alone. Use of a separate pruning set is a fairly common practice. Another pruning method that needs a separate data set is Quinlan’s [302] reduced error pruning. This method, unlike cost complexity pruning, does not build a sequence of trees and hence is claimed to be faster.

The requirement for an independent pruning set might be problematic especially when small training samples are involved. Several solutions have been suggested to get around this problem. Breiman *et al.* [31] describe a cross validation procedure that avoids reserving part of training data for pruning, but has a large computa-

tional complexity. Quinlan's pessimistic pruning [302, 306] does away with the need for a separate pruning set by using a statistical correlation test.

Crawford [69] analyzed Breiman *et al.*'s cross validation procedure, and pointed out that it has a large variance, especially for small training samples. He suggested a *.632 bootstrap* method¹⁴ as an effective alternative. Gelfand *et al.* [119] claimed that the cross validation method is both inefficient and possibly ineffective in finding the optimally pruned tree. They suggested an efficient iterative tree growing and pruning algorithm that is guaranteed to converge. This algorithm divides the training sample into two halves and iteratively grows the tree using one half and prunes using the other half, exchanging the roles of the halves in each iteration.

Quinlan and Rivest [309] used minimum description length [314] for tree construction as well as for pruning. An error in their coding method (which did not have an effect on their main conclusions) was pointed out in [371]. Another pruning method that is based on viewing the decision tree as an encoding for the training data was suggested by Forsyth *et al.* [112]. Use of dynamic programming to prune trees optimally and efficiently has been explored in [25].

A few studies have been done to study the relative effectiveness of pruning methods [247, 62, 91]. Just as in the case of splitting criteria, no single *ad hoc* pruning method has been adjudged to be superior to the others. The choice of a pruning method depends on factors such as the size of the training set and availability of additional data for pruning.

5. Other issues

Tree construction involves many issues other than finding good splits and knowing when to stop recursive splitting. This section bundles together several such issues.

5.1. Sample size versus dimensionality

The relationship between the size of the training set and the dimensionality of the problem is studied extensively in the pattern recognition literature [153, 175, 108, 54, 173, 202, 166, 114]. Researchers considered the problem of how sample size should vary according to dimensionality and *vice versa*. Intuitively, an imbalance between the number of samples and the number of features (i.e., too many samples with too few attributes, or too few samples with too many attributes) can make induction more difficult. Some conclusions from the above papers can be summarized, informally, as follows:

- For a finite sized data with little or no *a priori* information, the ratio of the sample size to dimensionality must be as large as possible to suppress optimistically biased evaluations of the performance of the classifier.
- For a given sample size used in training a classifier, there exists an optimum feature size and quantization complexity. (Optimality here is in terms of tree size, not predictive accuracy. Quantization complexity refers to the number of

ranges a dimension is split into.) This result is true for both two-class problems and multi-class problems.¹⁵

- The ratio of the sample size to dimensionality should vary inversely proportional to the amount of available knowledge about the class conditional densities.

In tasks where more features than the “optimal” are available, decision tree quality is known to be affected by the redundant and irrelevant attributes [10, 323]. To avoid this problem, either a feature subset selection method (Section 5.1.1) or a method to form a small set of composite features (Section 5.1.2) can be used as a preprocessing step to tree induction. An orthogonal step to feature selection is instance selection. If the training sample is too large to allow for efficient classifier induction, a subsample selection method (Section 5.1.3) can be employed.

5.1.1. Feature subset selection There is a large body of work on choosing relevant subsets of features (see the texts [84, 27, 245]). Much of this work was not developed in the context of tree induction, but a lot of it has direct applicability. There are two components to any method that attempts to choose the best subset of features. The first is a metric using which two feature subsets can be compared to determine which is better. Feature subsets have been compared in the literature using direct error estimation [111, 168] or using any of the feature evaluation criteria discussed in Section 3.1 (e.g. Bhattacharya distance was used for comparing subsets of features in [272]). Direct error estimation is similar to the wrapper approach [191], which advocates that the induction algorithm be used as a “black box” by the feature subset selection method.

The second component of feature subset selection methods is a search algorithm through the space of possible feature subsets. Most existing search procedures are heuristic in nature, as exhaustive search for the best feature subset is typically prohibitively expensive. (An exception is the optimal feature subset selection method using zero-one integer programming, suggested by Ichino and Sklansky [157].) A heuristic commonly used is the greedy heuristic. In *stepwise forward selection*, we start with an empty feature set, and add, at each stage, the best feature according to some criterion. In *stepwise backward elimination*, we start with the full feature set and remove, at each step, the worst feature. When more than one feature is greedily added or removed, *beam search* is said to have been performed [341, 48]. A combination of forward selection and backward elimination, a bidirectional search, was attempted in [341].

Comparisons of heuristic feature subset selection methods resound the conclusions of studies comparing feature evaluation criteria and studies comparing pruning methods — no feature subset selection heuristic is far superior to the others. [64, 366] showed that heuristic sequential feature selection methods can do arbitrarily worse than the optimal strategy. Mucciardi and Gose [262] compared seven feature subset selection techniques empirically and concluded that no technique was uniformly superior to the others. There has been a recent surge of interest in feature subset selection methods in the machine learning community, resulting in several empirical evaluations. These studies provide interesting insights on how

to increase the efficiency and effectiveness of the heuristic search for good feature subsets [185, 210, 48, 81, 257, 5].

5.1.2. Composite features Sometimes the aim is not to choose a good subset of features, but instead to find a few good “composite” features, which are arithmetic or logical combinations of the atomic features. In the decision tree literature, Henrichon and Fu [149] were probably the first to discuss “transgenerated” features, features generated from the original attributes. Friedman’s [113] tree induction method could consider with equal ease atomic and composite features. Techniques to search for multivariate splits (Section 3.2) can be seen as ways for constructing composite features. Use of linear regression to find good feature combinations has been explored recently in [28].

Discovery of good combinations of Boolean features to be used as tests at tree nodes is explored in the machine learning literature in [284] as well as in signal processing [14]. Ragavan and Rendell [310] describe a method that constructs Boolean features using lookahead, and uses the constructed feature combinations as tests at tree nodes. Lookahead for construction of Boolean feature combinations is also considered in [389]. Linear threshold unit trees for Boolean functions are described in [321]. Decision trees having first order predicate calculus representations, with Horn clauses as tests at internal nodes, are considered in [375].

5.1.3. Subsample selection Feature subset selection attempts to choose useful features. Similarly, subsample selection attempts to choose appropriate training samples for induction. Quinlan suggested “windowing”, a random training set sampling method, for his programs ID3 and C4.5 [306, 382]. A initially randomly chosen window can be iteratively expanded to include only the “important” training samples. Several ways of choosing representative samples for Nearest Neighbor learning methods exist (see [74, 75], for examples). Some of these techniques may be helpful for inducing decision trees on large samples, provided they are efficient. Oates and Jensen recently analyzed the effect of training set size on decision tree complexity [280].

5.2. Incorporating costs

In most real-world domains, attributes can have costs of measurement, and objects can have misclassification costs. If the measurement (misclassification) costs are not identical between different attributes (classes), decision tree algorithms may need to explicitly prefer cheaper trees. Several attempts have been made to make tree construction cost-sensitive. These involve incorporating attribute measurement costs (machine learning: [278, 279, 354, 360], pattern recognition: [77, 261], statistics: [184]) and incorporating misclassification costs [31, 66, 83, 51, 360]. Methods to incorporate attribute measurement costs typically include a cost term into the feature evaluation criterion, whereas variable misclassification costs are accounted for by using prior probabilities or cost matrices.

5.3. Missing attribute values

In real world data sets, it is often the case that some attribute values are missing from the data. Several researchers have addressed the problem of dealing with missing attribute values in the training as well as testing sets. For training data, Friedman [113] suggested that all objects with missing attribute values can be ignored while forming the split at each node. If it is feared that too much discrimination information will be lost due to ignoring, missing values may be substituted by the mean value of the particular feature in the training *subsample* in question. Once a split is formed, all objects with missing values can be passed down to all child nodes, both in the training and testing stages. The classification of an object with missing attribute values will be the largest represented class in the union of all the leaf nodes at which the object ends up. Breiman *et al.*'s CART system [31] more or less implemented Friedman's suggestions. Quinlan [304] also considered the problem of missing attribute values.

5.4. Improving upon greedy induction

Most tree induction systems use a greedy approach — trees are induced top-down, a node at a time. Several authors (e.g., [117, 311]) pointed out the inadequacy of greedy induction for difficult concepts. The problem of inducing globally optimal decision trees has been addressed time and again. For early work using dynamic programming and branch-and-bound techniques to convert decision tables to optimal trees, see [259].

Tree construction using partial or exhaustive lookahead has been considered in statistics [103, 57, 88], in pattern recognition [142], for tree structured vector quantizers [315], for Bayesian class probability trees [44], for neural trees [72] and in machine learning [278, 310, 271]. Most of these studies indicate that lookahead does not cause considerable improvements over greedy induction. Murthy and Salzberg [271] demonstrate that one-level lookahead does not help build significantly better trees and can actually *worsen* the quality of trees, causing *pathology* [273]. This seemingly unintuitive behavior is caused because of the way feature selection heuristics are defined and used within the greedy framework.

Constructing optimal or near-optimal decision trees using a two-stage approach has been attempted by many authors. In the first stage, a sufficient partitioning is induced using any reasonably good (greedy) method. In the second stage, the tree is *refined* to be as close to optimal as possible. Refinement techniques attempted include dynamic programming [241], fuzzy logic search [373] and multi-linear programming [23].

The build-and-refine strategy can be seen as a search through the space of all possible decision trees, starting at the greedily built suboptimal tree. In order to escape local minima in the search space, randomized search techniques, such as genetic programming [197] and simulated annealing [38, 228], have been attempted. These methods search the space of all decision trees using random perturbations, additions and deletions of the splits. A deterministic hill-climbing search procedure

has also been suggested for searching for optimal trees, in the context of sequential fault diagnosis [349]. Kroger [200] discusses the strategies and algorithm improvements needed to generate “optimal” classification trees.

Inducing topologically minimal trees, trees in which the number of occurrences of each attribute along each path are minimized, is the topic of [369]. Suen and Wang [348] described an algorithm that attempted to minimize the entropy of the whole tree and the class overlap simultaneously. (Class overlap is measured by the number of terminal nodes that represent the same class.)

5.5. *Estimating probabilities*

Decision trees have crisp decisions at leaf nodes. On the contrary, class probability trees assign a probability distribution for all classes at the terminal nodes. Breiman *et al.* ([31], Chapter 4) proposed a method for building class probability trees. Quinlan [305] discussed methods of extracting probabilities from decision trees. Buntine [44] described Bayesian methods for building, smoothing and averaging class probability trees. (Smoothing is the process of adjusting probabilities at a node in the tree based on the probabilities at other nodes on the same path. Averaging improves probability estimates by considering multiple trees.) Smoothing in the context of tree structured vector quantizers is described in [14]. An approach, which refines the class probability estimates in a greedily induced decision tree using local kernel density estimates has been suggested in [345].

Assignment of probabilistic goodness to splits in a decision tree is described in [136]. A unified methodology for combining uncertainties associated with attributes into that of a given test, which can then be systematically propagated down the decision tree, is given in [256].

5.6. *Multiple trees*

A known peril of decision tree construction is its variance, especially when the samples are small and the features are many [79]. Variance can be caused by random choice of training and pruning samples, by many equally good attributes only one of which can be chosen at a node, due to cross validation or because of other reasons. Many authors have suggested using a collection of decision trees, instead of just one, to reduce the variance in classification performance [207, 339, 340, 44, 146, 30]. The idea is to build a set of (correlated or uncorrelated) trees for the same training sample, and then combine their results.¹⁶ Multiple trees have been built using randomness [146] or using different subsets of attributes for each tree [339, 340]. Classification results of the trees have been combined using either simplistic voting methods [146] or using statistical methods for combining evidence [339]. The relationship between the correlation of errors of individual classifiers and the error of the combined classifier has been explored [9].

An alternative to multiple trees is a hybrid classifier that uses several small classifiers as parts of a larger classifier. Brodley [34] describes a system that automatically

selects the most suitable among a univariate decision tree, a linear discriminant and an instance based classifier at each node of a hierarchical, recursive classifier.

5.7. *Incremental tree induction*

Most tree induction algorithms use batch training — the entire tree needs to be recomputed to accommodate a new training example. A crucial property of neural network training methods is that they are incremental — network weights can be continually adjusted to accommodate training examples. Incremental induction of decision trees is considered by several authors. Friedman’s [113] binary tree induction method could use “adaptive” features for some splits. An adaptive split depends on the training subsample it is splitting. (An overly simple example of an adaptive split is a test on the mean value of a feature.) Utgoff *et al.* proposed incremental tree induction methods in the context of univariate decision trees [361, 363, 364] as well as multivariate trees [365]. Crawford [69] argues that approaches which attempt to update the tree so that the “best” split according to the updated sample is taken at each node, suffer from repeated restructuring. This occurs because the best split at a node vacillates widely while the sample at the node is still small. An incremental version of CART that uses significance thresholds to avoid the above problem is described in [69].

5.8. *Soft splits*

Two common criticisms of decision trees are the following: (1) As decisions in the lower levels of a tree are based on increasingly smaller fragments of the data, some of them may not have much probabilistic significance (data fragmentation). (2) As several leaf nodes can represent the same class, unnecessarily large trees may result, especially when the number of classes is large (high class overlap).

Several researchers have considered using *soft* splits of data for decision trees. A hard split divides the data into mutually exclusive partitions. A soft split, on the other hand, assigns a probability that each point belongs to a partition, thus allowing points to belong to multiple partitions. C4.5 [306] uses a simple form of soft splitting. Use of soft splits in pattern recognition literature can be found in [330, 373]. Jordan and Jacobs [170] describe a parametric, hierarchical classifier with soft splits. Multivariate regression trees using soft splitting criteria are considered [109]. Induction of fuzzy decision trees has been considered in [211, 387].

5.9. *Tree quality measures*

The fact that several trees can correctly represent the same data raises the question of how to decide that one tree is better than another. Several measures have been suggested to quantify tree quality. Moret [259] summarizes work on measures such as tree size, expected testing cost and worst-case testing cost. He shows that these three measures are pairwise incompatible, which implies that an algorithm mini-

mizing one measure is guaranteed *not* to minimize the others, for some tree. Fayyad and Irani [97] argue that, by concentrating on optimizing one measure, number of leaf nodes, one can achieve performance improvement along other measures.

Generalization accuracy is a popular measure for quantifying the goodness of learning systems. The accuracy of the tree is computed using a testing set that is independent of the training set or using estimation techniques like cross-validation or bootstrap. 10-fold cross-validation is generally believed to be a good “honest” assessment of tree predictive quality. Kononenko and Bratko [194] pointed out that comparisons on the basis of classification accuracy are unreliable, because different classifiers produce different types of estimates (e.g., some produce yes-or-no classifications, some output class probabilities) and accuracy values can vary with prior probabilities of the classes. They suggested an information based metric to evaluate a classifier, as a remedy to the above problems. Martin [234] argued that information theoretic measures of classifier complexity are not practically computable except within severely restricted families of classifiers, and suggested a generalized version of CART’s [31] 1-standard error rule as a means of achieving a tradeoff between classifier complexity and accuracy.

Description length, the number of bits required to “code” the tree and the data using some compact encoding, has been suggested as a means to combine the accuracy and complexity of a classifier [309, 112] .

5.10. Miscellaneous

Most existing tree induction systems proceed in a greedy top-down fashion. Bottom up induction of trees is considered in [209]. Bottom up tree induction is also common [291] in problems such as building identification keys and optimal test sequences. A hybrid approach to tree construction, that combined top-down and bottom-up induction can be found in [182].

We concentrate in this paper on decision trees that are constructed from labeled examples. The problem of learning trees from decision rules instead of examples is addressed in [162]. The problem of learning trees solely from prior probability distributions is considered in [11]. Learning decision trees from qualitative causal models acquired from domain experts is the topic of [295]. Given a trained network or any other learned model, Craven’s algorithm TREPAN [68] uses queries to induce a decision tree that approximates the function represented by the model.

Several attempts at generalizing the decision tree representation exist. Chou [56] considered decision *trellises*, where trellises are directed acyclic graphs with class probability vectors at the leaves and tests at internal nodes. Option trees, in which every internal node holds several optional tests along with their respective subtrees, are discussed in [43, 44]. Oliver [281] suggested a method to build decision graphs, which are similar to Chou’s decision trellises, using minimum length encoding principles [370]. Rymon [318] suggested SE-trees, set enumeration structures each of which can embed several decision trees.

Cox [65] argues that classification tree technology, as implemented in commercially available systems, is often more useful for pattern recognition than it is for

decision support. He suggests several ways of modifying existing methods to be *prescriptive* rather than descriptive.

An interesting method for displaying decision trees on multidimensional data, using *block diagrams*, is proposed in [355]. Block diagrams can point out features of the data as well as the deficiencies in the classification method. Parallelization of tree induction algorithms is discussed in detail in [293]. Hardware architectures to implement decision trees are described in [164].

6. Analyses

Researchers have tried to evaluate the tree induction method itself, to precisely answer questions such as “is it possible to build optimal trees?” and “how good is a specific feature evaluation rule?”. Most such investigations are theoretical, though there have been a few recent empirical ones.

6.1. NP-completeness

Several aspects of optimal tree construction are shown to be intractable. Hyafil and Rivest [155] proved that the problem of building optimal decision trees from decision tables, optimal in the sense of minimizing the expected number of tests required to classify an unknown sample, is NP-Complete. For sequential fault diagnosis, Cox *et al.* [67] showed that, for an arbitrary distribution of attribute costs and for an arbitrary distribution of input vectors, the problem of constructing a minimum expected cost classification tree to represent a simple function, the linear threshold function, is NP-complete. They show that even the problem of identifying the root node in an optimal strategy is NP-hard. The problem of building optimal trees from decision tables is considered by Murphy and McCraw [264], who proved that for most cases, construction of storage optimal trees is NP-complete. Naumov [274] proved that optimal decision tree construction from decision tables is NP-complete under a variety of measures. All the measures considered by the earlier papers on NP-completeness appear to be a subset of Naumov’s measures. The problem of constructing the smallest decision tree which best distinguishes characteristics of multiple distinct groups is shown to be NP-complete in [358].

Comer and Sethi [63] studied the asymptotic complexity of trie index construction in the document retrieval literature. Megiddo [240] investigated the problem of polyhedral separability (separating two sets of points using k hyper-planes), and proved that several variants of this problem are NP-complete. Results in the above three papers throw light on the complexity of decision tree induction. Lin *et al.* [216, 215] discussed NP-hardness of the problem of designing optimal pruned tree structured vector quantizers (TSVQ).

Most of the above results consider only univariate decision tree construction. Intuitively, linear or multivariate tree construction should be more difficult than univariate tree construction, as there is a much larger space of splits to be searched. Heath [145] proved that the problem of finding the split that minimizes the number of misclassified points, given two sets of mutually exclusive points, is NP-complete.

Hoeffgen *et al.* [151] proved that a more general problem is NP-hard — they proved that, for any $C \geq 1$, the problem of finding a hyper-plane that misclassifies no more than $C * \text{opt}$ examples, where opt is the minimum number of misclassifications possible using a hyper-plane, is also NP-hard.

As the problem of finding a single linear split is NP-hard, it is no surprise that the problem of building the optimal linear decision trees is NP-hard. However, one might hope that, by reducing the size of the decision tree, or the dimensionality of the data, it might be possible to make the problem tractable. This does not seem to be the case either. Blum and Rivest [24] showed that the problem of constructing an optimal 3-node neural network is NP-complete. Goodrich [130] proved that optimal (smallest) linear decision tree construction is NP-complete even in three dimensions.

6.2. Theoretical Insights

Goodman and Smyth [128] showed that greedy top-down induction of decision trees is directly equivalent to a form of Shannon-Fano prefix coding [96]. A consequence of this result is that top-down tree induction (using mutual information) is necessarily suboptimal in terms of average tree depth. Trees of maximal size generated by the CART algorithm [31] have been shown to have an error rate bounded by twice the Bayes error rate, and to be asymptotically Bayes optimal [131]. Miyakawa [251] considered the problem of converting decision tables to optimal trees, and studied the properties of *optimal* variables, the class of attributes only members of which can be used at the root of an optimal tree. Eades and Staples [86] showed that the optimality in search trees, in terms of worst-case depth, is very closely related to *regularity*.¹⁷ As irregular trees are not likely to be optimal, splitting rules (Section 3.1) that tend to slice off small corners of the attribute space building highly unbalanced trees are less likely to find optimal trees.

Some authors pointed out the similarity or equivalence between the problem of constructing decision trees and existing, seemingly unrelated, problems. Such insights provide valuable tools for analyzing decision trees. Wang and Suen [372] show that entropy-reduction point of view is powerful in theoretically bounding search depth and classification error. Chou and Gray [58] view decision trees as variable-length encoder-decoder pairs, and show that rate is equivalent to tree depth while distortion is the probability of misclassification. Brandman *et al.* [29] suggested a universal technique to lower bound the size and other characteristics of decision trees for arbitrary Boolean functions. This technique is based on the power spectrum coefficients of the n -dimensional Fourier transform of the function. Turksen and Zhao [359] proved the equivalence between a pseudo-Boolean analysis and the ID3 algorithm [301].

6.3. Assumptions and biases

Most tree induction methods are heuristic in nature. They use several assumptions and biases, hoping that together the heuristics produce good trees. Some authors

have attempted to evaluate the validity and relevance of the assumptions and biases in tree induction.¹⁸

Assumption: Multi-stage classifiers may be more accurate than single stage classifiers. Analysis: However, the data fragmentation caused by multi-stage hierarchical classifiers may compensate for the gain in accuracy. Michie [243] argues that top-down induction algorithms may provide overly complex classifiers that have no real conceptual structure in encoding relevant knowledge. As a solution to this problem, Gray [132] suggested an induction method that generates a single disjuncts of conjuncts rule, using the same time complexity as tree induction. The efficacy of multi-level decision trees is compared by Holte [152] to simple, one-level classification rules. He concluded that, on most real world data sets commonly used by the machine learning community [266], decision trees do not perform significantly better than one level rules. These conclusions, however, were refuted by Elomaa [89] on several grounds. Elomaa argued that Holte’s observations may have been the peculiarities of the data he used, and that the slight differences in accuracy that Holte observed were still significant.

Bias: Smaller consistent decision trees have higher generalization accuracy than larger consistent trees (Occam’s Razor). Analysis: Murphy and Pazzani [267] empirically investigated the truth of this bias. Their experiments indicate that this conjecture seems to be true. However, their experiments indicate that the smallest decision trees typically have lesser generalization accuracy than trees that are slightly larger. In an extension of this study, Murphy [265] evaluated the size bias as a function of concept size. He concluded that (1) bias for smaller trees is generally beneficial in terms of accuracy and that (2) though larger trees perform better than smaller ones for high-complexity concepts, it is better to guess the correct size randomly than to have a pre-specified size bias.

Assumption: Locally optimizing information or distance based splitting criteria, (Section 3.1) tends to produce small, shallow, accurate trees. Analysis: A class of binary splits \mathcal{S} for a data set is said to be complete if, informally, for every partition of the data, there exists a member of \mathcal{S} that induces the partition. Zimmerman [390] considered the problem of building identification keys for complete classes of splits, given arbitrary class distributions. Garey and Graham [117] analyze the properties of recursive greedy splitting on the quality of trees induced from decision tables, and showed that greedy algorithms using information theoretic splitting criteria can be made to perform arbitrarily worse than the optimal. Kurzynski [204] showed that, for globally optimum performance, decisions made at each node should “emphasize the decision that leads to a greater joint probability of correct classification at the next level”, i.e., decisions made at different nodes in the tree should *not* be independent. Loveland [222] analyzed the performance of variants of Gini index in the context of sequential fault diagnosis.

Goodman and Smyth [128, 129] analyzed greedy tree induction from an information theoretic view point. They proved that mutual information-based induction is equivalent to a form of Shannon-Fano prefix coding, and through this insight argued that greedily induced trees are nearly optimal in terms of depth. This conjecture is substantiated empirically in [270], where it is shown that the expected depth of

trees greedily induced using information gain [301] and Gini index [31] is very close to that of the optimal, under a variety of experimental conditions. Relationship between feature evaluation by Shannon's entropy and the probability of error is investigated in [196, 312].

7. The practical promise

The discussion so far in the paper has concentrated on techniques for and analysis of decision tree construction. All these are in vain unless this technique is practically useful and perhaps outperforms some competitive techniques. In this section, we address these two issues. We argue that decision trees are practically a very useful technique, by tabulating examples of their use in diverse real-world applications. We briefly discuss existing software packages for building decision trees from data. We also summarize work comparing decision trees to alternative techniques for data analysis, such as neural networks, nearest neighbor methods and regression analysis.

7.1. *Selected real-world applications*

This section lists a few recent real-world applications of decision trees. The aim is to give the reader a “feel” for the versatility and usefulness of decision tree methods for data exploration, and not to be useful for readers interested in finding the potential of tree classifiers in specific domains. Our coverage of applications is, by necessity, very limited. All the application papers cited below were published in refereed journals or as Ph.D theses, after 1993. We restrict to application domains where the domain scientists tried to use decision trees, rather than where decision tree researchers tested their algorithm(s) on several application domains. The application areas are listed below in alphabetical order.

- **Agriculture:** Application of a range of machine learning methods including decision trees to problems in agriculture and horticulture is described in [239].
- **Astronomy:** Astronomy has been an active domain for using automated classification techniques.¹⁹ Use of decision trees has been reported for filtering noise from Hubble Space Telescope images [323], in star-galaxy classification [378], for determining galaxy counts [377] and discovering quasars [180] in the Second Palomar Sky Survey.
- **Biomedical Engineering:** For identifying features to be used in implantable devices [123].
- **Control Systems:** For control of nonlinear dynamical systems [154] and control of flotation plants [8].
- **Financial analysis:** For asserting the attractiveness of buy-writes [242], among many other data mining applications.

- **Image processing:** For the interpretation of digital images in radiology [294], for recognizing 3-D objects [39], for high level vision [187] and outdoor image segmentation [40].
- **Language processing:** For medical text classification [212], for acquiring a statistical parser from a set of parsed sentences [229].
- **Law:** For discovering knowledge in international conflict and conflict management databases, for the possible avoidance and termination of crises and wars [116].
- **Manufacturing and Production:** To non-destructively test welding quality [90], for semiconductor manufacturing [163], for increasing productivity [179], for material procurement method selection [73], to accelerate rotogravure printing [92], for process optimization in electro-chemical machining [95], to schedule printed circuit board assembly lines [296], to uncover flaws in a Boeing manufacturing process [313] and for quality control [135]. For a recent review of the use of machine learning (decision trees and other techniques) in scheduling, see [13].
- **Medicine:** Medical research and practice have long been important areas of application for decision tree techniques. Recent uses of automatic induction of decision trees can be found in cardiology [221, 94, 192], study of tooth enamel [277], psychiatry [238], gastroenterology [171], for detecting microcalcifications in mammography [385], to analyze Sudden Infant Death (SID) syndrome [381] and for diagnosing thyroid disorders [104].
- **Molecular biology:** Initiatives such as the Human Genome Project and the GenBank database offer fascinating opportunities for machine learning and other data exploration methods in molecular biology. Recent use of decision trees for analyzing amino acid sequences can be found in [338] and [322].
- **Pharmacology:** Use of tree based classification for drug analysis can be found in [71].
- **Physics:** For the detection of physical particles [26].
- **Plant diseases:** To assess the hazard of mortality to pine trees [16].
- **Power systems:** For power system security assessment [144] and power stability prediction [317].
- **Remote Sensing:** Remote sensing has been a strong application area for pattern recognition work on decision trees (see [350, 182]). Recent uses of tree-based classification in remote sensing can be found in [319, 82, 208].
- **Software development:** To estimate the development effort of a given software module in [199].
- **Other:** Decision trees have also been used recently for building personal learning assistants [250] and for classifying sleep signals [201].

7.2. *Software packages*

Today, there are many research codes and commercial products whose purpose is constructing decision trees from data. In addition, decision tree construction is a primary function provided in many general-purpose data mining tool suites. In the interest of brevity we will not survey decision tree software tools here. A good list of current software can be found in the “Siftware” section in the Knowledge Discovery Nuggets web page <http://www.kdnuggets.com/siftware.html>.²⁰ In addition to the decision-tree entries, many entries listed under “software suites” and “classification using multiple approaches” are also relevant.

Available decision tree software varies in terms of the specific algorithms implemented, sophistication of auxiliary functions such as visualization, data formats supported and speed. The web page above just lists decision tree (and other) software packages. It does not evaluate them. Objective comparative evaluation of decision tree software, in terms of available functionality, programmability, efficiency, user-friendliness, visualization support, database interface and price would be a very interesting, relevant but not necessarily an easy or straightforward exercise. The author is unaware of any existing comparisons.

It is perhaps important to point out that no single available software program implements all that is known about decision trees. Each package chooses its favorite algorithms and heuristics to implement. These choices should not be seen as shortcomings of the packages, because implementing everything known is a very significant task which may have primarily research value.

7.3. *Trees versus other data analysis methods*

This section, like Section 7.1 above, is not comprehensive but merely illustrative. We briskly provide pointers to work that has compared decision trees against competing techniques for data analysis in statistics and machine learning.

Brown *et al.* [36] compared back-propagation neural networks with decision trees on three problems that are known to be multi-modal. Their analysis indicated that there was not much difference between both methods, and that neither method performed very well in its “vanilla” state. The performance of decision trees improved in their study when multivariate splits were used, and back-propagation networks did better with feature selection. Comparisons of symbolic and connectionist methods can also be found in [379, 337]. Multi-layer perceptrons and CART [31] with and without linear combinations are compared in [12] to find that there is not much difference in accuracy. Similar conclusions were reached in [106] when ID3 [301] and back-propagation were compared. Talmon *et al.* [352] compared classification trees and neural networks for analyzing electrocardiograms (ECG) and concluded that no technique is superior to the other. In contrast, ID3 is adjudged to be slightly better than connectionist and Bayesian methods in [347].

Giplin *et al.* [125] compared stepwise linear discriminant analysis, stepwise logistic regression and CART [31] to three senior cardiologists, for predicting whether a patient would die within a year of being discharged after an acute myocardial in-

farction. Their results showed that there was no difference between the physicians and the computers, in terms of the prediction accuracy. Kors and Van Bemmelen [195] compared statistical multivariate methods with heuristic decision tree methods, in the domain of electrocardiogram (ECG) analysis. Their comparisons show that decision tree classifiers are more comprehensible and flexible to incorporate or change existing categories. Comparisons of CART to multiple linear regression and discriminant analysis can be found in [46] where it is argued that CART is more suitable than the other methods for very noisy domains with lots of missing values. Comparisons between decision trees and statistical methods like linear discriminant function analysis and automatic interaction detection (AID) are given in [237], where it is argued that machine learning methods sometimes outperform the statistical methods and so should not be ignored.

Feng *et al.* [102] present a comparison of several machine learning methods (including decision trees, neural networks and statistical classifiers) as a part of the European Statlog project. The Statlog project [244] was initiated by the European Commission for “The Comparative Testing of Statistical and Logical Learning Algorithms on Large-Scale Applications to Classification, Prediction and Control”. Feng *et al.*’s main conclusions were that (1) no method seems uniformly superior to others, (2) machine learning methods seem to be superior for multimodal distributions, and (3) statistical methods are computationally the most efficient. Thrun *et al.* [356] compared several learning algorithms on simulated Monk’s problems.

Long *et al.* [221] compared Quinlan’s C4 [306] to logistic regression on the problem of diagnosing acute cardiac ischemia, and concluded that both methods came fairly close to the expertise of the physicians. In their experiments, logistic regression outperformed C4. Curram and Mingers [70] compare decision trees, neural networks and discriminant analysis on several real world data sets. Their comparisons reveal that linear discriminant analysis is the fastest of the methods, when the underlying assumptions are met, and that decision trees methods overfit in the presence of noise. Dietterich *et al.* [78] argue that the inadequacy of trees for certain domains may be due to the fact that trees are unable to take into account some statistical information that is available to other methods like neural networks. They show that decision trees perform significantly better on the text-to-speech conversion problem when extra statistical knowledge is provided.

Pizzi and Jackson [297] compare an expert system developed using traditional knowledge engineering methods to Quinlan’s ID3 [301] in the domain of tonsillectomy. Quinlan empirically compared decision trees to genetic classifiers [303] and to neural networks [307]. Palvia and Gordon [287] compared decision tables, decision trees and decision rules, to determine which formalism is best for decision analysis. Many methods for learning from examples are compared in an early study by Dietterich and Michalski [80].

8. Conclusions

This paper attempted a multi-disciplinary survey of work in automatically constructing decision trees from data. We gave pointers to work in fields such as pattern

recognition, statistics, decision theory, machine learning, mathematical programming and neural networks. We attempted to provide a concise description of the directions which decision tree work has taken over the years. Our goal is to provide an overview of existing work in decision trees, and a taste of their usefulness, to the newcomers as well as practitioners in the field of data mining and knowledge discovery. We also hope that overviews like these can help avoid some redundant, *ad hoc* effort, both from researchers and from system developers.

The hierarchical, recursive tree construction methodology is very powerful and has repeatedly been shown to be useful for diverse real-world problems. It is also simple and intuitively appealing. However, the simplicity of the methodology should not lead a practitioner to take a slack attitude towards using decision trees. Just as in the case of statistical methods or neural networks, building a successful tree classifier for an application requires a thorough understanding of the problem itself, and a deep knowledge of tree methodology.

Acknowledgments

Simon Kasif first pointed out to me that a multi-disciplinary survey on decision trees is a worthwhile exercise to undertake. I thank Simon, Steven Salzberg and Lewis Stiller for reading and commenting on the manuscript. I am grateful to Wray Buntine for writing a great review which helped improve the paper.

Notes

1. This is adapted from [282], where a similar taxonomy was suggested in the general framework of searching for structure in data.
2. Several earlier data mining products are old machine learning methods just repackaged under new titles.
3. Lubinsky [226] considered trees that can have internal nodes with just one child. At these nodes, the data are not split, but residuals are taken from a single variable regression.
4. While converting decision tables to trees, it is common to have leaf nodes that have a “no decision” label. (A good recent paper on the use of decision tables in classification is [189].)
5. A decision tree is said to perform classification if the class labels are discrete values, and *regression* if the class labels are continuous. We restrict almost entirely to classification trees in this paper.
6. One interesting early patent on decision tree growing was assigned to IBM (US Patent 4,719,571).
7. The desirable properties of a measure of entropy include symmetry, expandability, decisivity, additivity and recursivity. Shannon’s entropy [336] possesses all of these properties [4]. For an insightful treatment of entropy reduction as a common theme underlying several pattern recognition problems, see [376].
8. Goodman and Smyth [128] report that the idea of using the mutual information between features and classes to select the best feature was originally put forward by Lewis [213].
9. named after the Italian economist Corrado Gini (1884–1965)
10. Quinlan’s C4.5 [306] uses a naive version of the confidence intervals for doing pessimistic pruning.

11. Schaffer [327] stated and proved a conservation theorem that states, essentially, that positive performance in some learning situations must be offset by an equal degree of negative performance in others. To clarify the, sometimes non-intuitive, consequences of the conservation theorem, Schaffer [328] gave an example of a concept for which information *loss* gives better generalization accuracy than information gain. Schaffer's work draws heavily upon Wolpert's earlier results [384, 383].
12. Trees in which an internal node can have more than 2 children, have also been considered in the vector quantization literature [329].
13. Techniques that start with a sufficient partitioning and then optimize the structure (e.g., [241]) can be thought of as being a converse to this approach.
14. In bootstrapping, B independent learning samples, each of size N are created by random sampling with replacement from the original learning sample L . In cross validation, L is divided randomly into B mutually exclusive, equal sized partitions. Efron [87] showed that, although cross validation closely approximates the true result, bootstrap has much less variance, especially for small samples. However, there exist arguments that cross validation is clearly preferable to bootstrap in practice [190].
15. Van Campenhout [47] argues that increasing the amount of information in a measurement subset through enlarging its size or complexity never worsens the error probability of a truly Bayesian classifier. Even after this guarantee, the cost and complexity due to additional measurements may not be worth the slight (if any) improvement in accuracy. Moreover, most real world classifiers are not truly Bayesian.
16. A lot of work exists in the neural networks literature on using committees or ensembles of networks to improve classification performance. See [140] for example.
17. A c -regular tree is a tree in which all nodes have c children, and if one child of an internal node is a leaf, then so are all other children. A tree is regular if it is c -regular for any c .
18. It is argued empirically [79] that the variance in decision tree methods is more a reason than bias for their poor performance on some domains.
19. For a general description of modern classification problems in astronomy, which prompt the use of pattern recognition and machine learning techniques, see [203].
20. Considerable ongoing discussion exists about the appropriateness of Internet references in scholarly publications. Critics argue that such references assume the availability of the Internet/WWW to the readership as well as the relative permanence and continued correctness of the referenced articles. While acknowledging the merits of such criticism, we nevertheless resort to referencing the KDNuggets web site here. This is partly because any reasonable survey of decision tree software tools would be involved and long, and has a relatively brief life span because of the ever-evolving nature of the market.

References

1. AAAI. *AAAI-92: Proc. of the Tenth National Conf. on Artificial Intelligence*, San Jose, CA, 12-16th, July 1992. AAAI Press / The MIT Press.
2. AAAI. *AAAI-93: Proc. of the Eleventh National Conf. on Artificial Intelligence*, Washington, DC, 11-15th, July 1993. AAAI Press / The MIT Press.
3. AAAI. *AAAI-94: Proc. of the Twelfth National Conf. on Artificial Intelligence*, volume 1, Seattle, WA, 31st July - 4th August 1994. AAAI Press / The MIT Press.
4. J. Aczel and J. Daroczy. *On measures of information and their characterizations*. Academic Pub., New York, 1975.
5. David W. Aha and Richard L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *AI&Statistics-95* [7], pages 1-7.
6. *AI&Stats-93: Preliminary Papers of the Fourth Int. Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL, 3rd-6th, January 1993. Society for AI and Statistics.
7. *AI&Stats-95: Preliminary Papers of the Fifth Int. Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL, 4-7th, January 1995. Society for AI and Statistics.
8. C. Aldrich, D. W. Moolman, F. S. Gouws, and G. P. J. Schmitz. Machine learning strategies for control of flotation plants. *Control Eng. Practice*, 5(2):263-269, February 1997.

9. Kamal M. Ali and Michael J. Pazzani. On the link between error correlation and error reduction in decision tree ensembles. Technical Report ICS-TR-95-38, University of California, Irvine, Department of Information and Computer Science, September 1995.
10. Hussein Almuallim and Thomas G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69:279–305, 1994.
11. Peter Argentiero, Roland Chin, and Paul Beaudet. An automated approach to the design of decision tree classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-4(1):51–57, January 1982.
12. Les Atlas, Ronald Cole, Yeshwant Muthuswamy, Alan Lipman, Jerome Connor, Dong Park, Muhammed El-Sharkawi, and Robert J. Marks II. A performance comparison of trained multilayer perceptrons and trained classification trees. *Proc. of the IEEE*, 78(10):1614–1619, 1990.
13. Haldun Aytug, Siddhartha Bhattacharya, Gary J. Koehler, and Jane L. Snowdon. A review of machine learning in scheduling. *IEEE Trans. on Eng. Management*, 41(2):165–171, May 1994.
14. L. Bahl, P.F. Brown, P.V. de Souza, and R. L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37(7):1001–1008, 1989.
15. Eard Baker and A. K. Jain. On feature ordering in practice and some finite sample effects. In *Proc. of the Third Int. Joint Conf. on Pattern Recognition*, pages 45–49, San Diego, CA, 1976.
16. F. A. Baker, David L. Verbyla, C. S. Hodges Jr., and E. W. Ross. Classification and regression tree analysis for assessing hazard of pine mortality caused by hetero basidion annosum. *Plant Disease*, 77(2):136, February 1993.
17. W. A. Belson. Matching and prediction on the principle of biological classification. *Applied Statistics*, 8:65–75, 1959.
18. Moshe Ben-Bassat. Myopic policies in sequential classification. *IEEE Trans. on Computing*, 27(2):170–174, February 1978.
19. Moshe Ben-Bassat. Use of distance measures, information measures and error bounds on feature evaluation. In Krishnaiah and Kanal [198], pages 773–791.
20. K.P. Bennett and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.
21. K.P. Bennett and O.L. Mangasarian. Multicategory discrimination via linear programming. *Optimization Methods and Software*, 3:29–39, 1994.
22. Kristin P. Bennett. Decision tree construction via linear programming. In *Proc. of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conf.*, pages 97–101, 1992.
23. Kristin P. Bennett. Global tree optimization: A non-greedy decision tree algorithm. In *Proc. of Interface 94: The 26th Symposium on the Interface*, Research Triangle, North Carolina, 1994.
24. A. Blum and R. Rivest. Training a 3-node neural network is NP-complete. In *Proc. of the 1988 Workshop on Computational Learning Theory*, pages 9–18, Boston, MA, 1988. Morgan Kaufmann.
25. Marko Bohanec and Ivan Bratko. Trading accuracy for simplicity in decision trees. *Machine Learning*, 15:223–250, 1994.
26. David Bowser-Chao and Debra L. Dzialo. Comparison of the use of binary decision trees and neural networks in top quark detection. *Physical Review D: Particles and Fields*, 47(5):1900, March 1993.
27. D. Boyce, A. Farhi, and R. Weishedel. *Optimal Subset Selection*. Springer-Verlag, 1974.
28. Anna Bramanti-Gregor and Henry W. Davis. The statistical learning of accurate heuristics. In IJCAI-93 [160], pages 1079–1085. Editor: Ruzena Bajcsy.
29. Y. Brandman, A. Orlitsky, and J. Hennessy. A spectral lower bound technique for the size of decision trees and two-level AND/OR circuits. *IEEE Trans. on Comp.*, 39(2):282–286, February 1990.
30. Leo Breiman. Bagging predictors. Technical report, Department of Statistics, Univ. of California, Berkeley, CA, 1994.
31. Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and Regression Trees*. Wadsworth Int. Group, 1984.

32. Richard P. Brent. Fast training algorithms for multilayer neural nets. *IEEE Trans. on Neural Networks*, 2(3):346–354, May 1991.
33. Leonard A. Breslow and David W. Aha. Simplifying decision trees: A survey. Technical Report AIC-96-014, Navy Center for Applied Research in Artificial Intelligence, Naval Research Lab., Washington DC 20375, 1996. breslow, aha.aic.nrl.navy.mil.
34. Carla E. Brodley. *Recursive Automatic Algorithm Selection for Inductive Learning*. PhD thesis, Univ. of Massachusetts, Amherst, MA, 1994.
35. Carla E. Brodley and Paul E. Utgoff. Multivariate decision trees. *Machine Learning*, 19:45–77, 1995.
36. Donald E. Brown, Vincent Corruble, and Clarence Louis Pittard. A comparison of decision tree classifiers with backpropagation neural networks for multimodal classification problems. *Pattern Recognition*, 26(6):953–961, 1993.
37. Donald E. Brown and Clarence Louis Pittard. Classification trees with optimal multivariate splits. In *Proc. of the Int. Conf. on Systems, Man and Cybernetics*, volume 3, pages 475–477, Le Touquet, France, 17–20th, October 1993. IEEE, New York.
38. R.S. Bucy and R.S. Diesposti. Decision tree design by simulated annealing. *Mathematical Modeling and Numerical Analysis*, 27(5):515–534, 1993. A RAIRO J.
39. M. E. Bullock, D. L. Wang, Fairchild S. R., and T. J. Patterson. Automated training of 3-D morphology algorithm for object recognition. *Proc. of SPIE – The Int. Society for Optical Eng.*, 2234:238–251, 1994. Issue title: Automatic Object Recognition IV.
40. Shashi D. Buluswer and Bruce A. Draper. Non-parametric classification of pixels under varying illumination. *SPIE: The Int. Society for Optical Eng.*, 2353:529–536, November 1994.
41. W. Buntine and T. Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8:75–85, 1992.
42. W. L. Buntine. Decision tree induction systems: a Bayesian analysis. In L. N. Kanal, T. S. Levitt, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 3*. Elsevier Science Publishers, Amsterdam, 1989.
43. Wray Buntine. *A theory of learning classification rules*. PhD thesis, Univ. of Technology, Sydney, Australia, 1991.
44. Wray Buntine. Learning classification trees. *Statistics and Computing*, 2:63–73, 1992.
45. Wray Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Trans. on Knowledge and Data Engineering*, 1996.
46. Janice D. Callahan and Stephen W. Sorensen. Rule induction for group decisions with statistical data - an example. *J. of the Operational Research Society*, 42(3):227–234, March 1991.
47. Jan M. Van Campenhout. Topics in measurement selection. In Krishnaiah and Kanal [198], pages 793–803.
48. Rich Caruana and Dayne Freitag. Greedy attribute selection. In ML-94 [254], pages 28–36. Editors: William W. Cohen and Haym Hirsh.
49. Richard G. Casey and George Nagy. Decision tree design using a probabilistic model. *IEEE Trans. on Information Theory*, IT-30(1):93–99, January 1984.
50. Jason Catlett. *Megainduction*. PhD thesis, Basser Department of Computer Science, Univ. of Sydney, Australia, 1991.
51. Jason Catlett. Tailoring rulesets to misclassification costs. In AI&Statistics-95 [7], pages 88–94.
52. Bing-Bing Chai, Xinhua Zhuang, Yunxin Zhao, and Jack Sklansky. Binary linear decision tree with genetic algorithm. In *Proc. of the 13th Int. Conf. on Pattern Recognition 4*. IEEE Computer Society Press, Los Alamitos, CA, 1996.
53. B. Chandrasekaran. From numbers to symbols to knowledge structures: Pattern Recognition and Artificial Intelligence perspectives on the classification task. volume 2, pages 547–559. Elsevier Science, Amsterdam, The Netherlands, 1986.
54. B. Chandrasekaran and A. K. Jain. Quantization complexity and independent measurements. *IEEE Trans. on Comp.*, C-23(1):102–106, January 1974.
55. P. Chaudhuri, W. D. Lo, W. Y. Loh, and C. C. Yang. Generalized regression trees. *Statistica Sinica*, 5(2):641–666, 1995.

56. Philip A. Chou. *Applications of Information Theory to Pattern Recognition and the Design of Decision Trees and Trellises*. PhD thesis, Stanford Univ., 1988.
57. Philip A. Chou. Optimal partitioning for classification and regression trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(4):340–354, April 1991.
58. Philip A. Chou and Robert M. Gray. On decision trees for pattern recognition. In *Proc. of the IEEE Symposium on Information Theory*, page 69, Ann Arbor, MI, 1986.
59. Krzysztof J. Cios and Ning Liu. A machine learning method for generation of a neural network architecture: A continuous ID3 algorithm. *IEEE Trans. on Neural Networks*, 3(2):280–291, March 1992.
60. I. Cleote and H. Theron. CID3: An extension of ID3 for attributes with ordered domains. *South African Computer J.*, 4:10–16, March 1991.
61. J.R.B. Cockett and J.A. Herrera. Decision tree reduction. *J. of the ACM*, 37(4):815–842, October 1990.
62. W.W. Cohen. Efficient pruning methods for separate-and-conquer rule learning systems. In *IJCAI-93 [160]*, pages 988–994. Editor: Ruzena Bajcsy.
63. Douglas Comer and Ravi Sethi. The complexity of trie index construction. *J. of the ACM*, 24(3):428–440, July 1977.
64. T.M. Cover and J.M. Van Campenhout. On the possible orderings in the measurement selection problems. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-7(9), 1977.
65. Louis Anthony Cox. Using causal knowledge to learn more useful decision rules from data. In *AI&Statistics-95 [7]*, pages 151–160.
66. Louis Anthony Cox and Yuping Qiu. Minimizing the expected costs of classifying patterns by sequential costly inspections. In *AI&Statistics-93 [6]*.
67. Louis Anthony Cox, Yuping Qiu, and Warren Kuehner. Heuristic least-cost computation of discrete classification functions with uncertain argument values. *Annals of Operations Research*, 21(1):1–30, 1989.
68. Mark W. Craven. Extracting comprehensible models from trained neural networks. Technical Report CS-TR-96-1326, University of Wisconsin, Madison, September 1996.
69. Stuart L. Crawford. Extensions to the CART algorithm. *Int. J. of Man-Machine Studies*, 31(2):197–217, August 1989.
70. Stephen P. Curram and John Mingers. Neural networks, decision tree induction and discriminant analysis: An empirical comparison. *J. of the Operational Research Society*, 45(4):440–450, April 1994.
71. K.T. Dago, R. Luthringer, R. Lengelle, G. Rinaudo, and J. P. Matcher. Statistical decision tree: A tool for studying pharmacoe-EEG effects of CNS-active drugs. *Neuropsychobiology*, 29(2):91–96, 1994.
72. Florence D'Alché-Buc, Didier Zwierski, and Jean-Pierre Nadal. Trio learning: A new strategy for building hybrid neural trees. *Int. J. of Neural Systems*, 5(4):259–274, December 1994.
73. S.K. Das and S. Bhambri. A decision tree approach for selecting between demand based, reorder and JIT/kanban methods for material procurement. *Production Planning and Control*, 5(4):342, 1994.
74. Belur V. Dasarthy, editor. *Nearest neighbor (NN) norms: NN pattern classification techniques*. IEEE Computer Society Press, Los Alamitos, CA, 1991.
75. Belur V. Dasarthy. Minimal consistent set (MCS) identification for optimal nearest neighbor systems design. *IEEE Trans. on systems, man and cybernetics*, 24(3):511–517, 1994.
76. G. R. Dattatreya and Laveen N. Kanal. Decision trees in pattern recognition. In Kanal and Rosenfeld, editors, *Progress in Pattern Recognition*, volume 2, pages 189–239. Elsevier Science, 1985.
77. G. R. Dattatreya and V. V. S. Sarma. Bayesian and decision tree approaches to pattern recognition including feature measurement costs. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-3(3):293–298, 1981.
78. Thomas G. Dietterich, Hermann Hild, and Ghulam Bakiri. A comparison of ID3 and back-propagation for english text-to-speech mapping. *Machine Learning*, 18:51–80, 1995.
79. Thomas G. Dietterich and Eun Bae Kong. Machine learning bias, statistical bias and statistical variance of decision tree algorithms. In *ML-95 [255]*. to appear.

80. Thomas G. Dietterich and Ryszard S. Michalski. A comparative view of selected methods for learning from examples. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning, an Artificial Intelligence Approach*, volume 1, pages 41–81. Morgan Kaufmann, San Mateo, CA, 1983.
81. Justin Doak. An evaluation of search algorithms for feature selection. Technical report, Graduate Group in Computer Science, Univ. of California at Davis; and Safeguards Systems Group, Los Alamos National Lab., January 1994.
82. D. L. Dowe and N. Krusel. Decision tree models of bushfire activity. *AI Applications*, 8(3):71–72, 1994.
83. B. A. Draper, Carla E. Brodley, and Paul E. Utgoff. Goal-directed classification using linear machine decision trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(9):888, 1994.
84. N. R. Draper and H. Smith. *Applied Regression Analysis*. Wiley, New York, 1966. 2nd edition in 1981.
85. R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
86. Eades and Staples. On optimal trees. *Journal of Algorithms*, 2(4):369–384, 1981.
87. Bradley Efron. Estimating the error rate of a prediction rule: improvements on cross-validation. *J. of American Statistical Association*, 78(382):316–331, June 1983.
88. John F. Elder, IV. Heuristic search for model structure. In AI&Statistics-95 [7], pages 199–210.
89. Tapio Elomaa. In defence of C4.5: Notes on learning one-level decision trees. In ML-94 [254], pages 62–69. Editors: William W. Cohen and Haym Hirsh.
90. A. Ercil. Classification trees prove useful in nondestructive testing of spotweld quality. *Welding J.*, 72(9):59, September 1993. Issue Title: Special emphasis: Rebuilding America's roads, railways and bridges.
91. Floriana Esposito, Donato Malerba, and Giovanni Semeraro. A further study of pruning methods in decision tree induction. In AI&Statistics-95 [7], pages 211–218.
92. Bob Evans and Doug Fisher. Overcoming process delays with decision tree induction. *IEEE Expert*, pages 60–66, February 1994.
93. Brian Everitt. *Cluster Analysis - 3rd Edition*. E. Arnold Press, London., 1993.
94. Judith A. Falconer, Bruce J. Naughton, Dorothy D. Dunlop, Elliot J. Roth, and Dale C. Strasser. Predicting stroke inpatient rehabilitation outcome using a classification tree approach. *Archives of Physical Medicine and Rehabilitation*, 75(6):619, June 1994.
95. A. Famili. Use of decision tree induction for process optimization and knowledge refinement of an industrial process. *Artificial Intelligence for Eng. Design, Analysis and Manufacturing (AI EDAM)*, 8(1):63–75, Winter 1994.
96. R. M. Fano. *Transmission of Information*. MIT Press, Cambridge, MA, 1961.
97. Usama M. Fayyad and Keki B. Irani. What should be minimized in a decision tree? In AAAI-90: *Proc. of the National Conf. on Artificial Intelligence*, volume 2, pages 749–754. AAAI, 1990.
98. Usama M. Fayyad and Keki B. Irani. The attribute specification problem in decision tree generation. In AAAI-92 [1], pages 104–110.
99. Usama M. Fayyad and Keki B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8(2):87–102, 1992.
100. Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous valued attributes for classification learning. In IJCAI-93 [160], pages 1022–1027. Editor: Ruzena Bajcsy.
101. Edward A. Feigenbaum. Expert systems in the 1980s. In A. Bond, editor, *State of the Art in Machine Intelligence*. Pergamon-Infotech, Maidenhead, 1981.
102. C. Feng, A. Sutherland, R. King, S. Muggleton, and R. Henery. Comparison of machine learning classifiers to statistics and neural networks. In AI&Statistics-93 [6], pages 41–52.
103. A. Fielding. Binary segmentation: the automatic interaction detector and related techniques for exploring data structure. In O'Muircheartaigh and Payne [283], pages 221–257.
104. P. E. File, P. I. Dugard, and A. S. Houston. Evaluation of the use of induction in the development of a medical expert system. *Comp. and Biomedical Research*, 27(5):383–395, October 1994.

105. Douglas Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:130–172, 1987.
106. Douglas Fisher and Kathleen McKusick. An empirical comparison of ID3 and back propagation. In IJCAI-89 [159]. Editor: N. S. Sridharan.
107. R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *Computer J.*, 6(ISS.2):163–168, 1963.
108. D. H. Foley. Considerations of sample and feature size. *IEEE Trans. on Information Theory*, IT-18:618–626, 1972.
109. F. Forouraghi, L. W. Schmerr, and G. M. Prabhu. Induction of multivariate regression trees for design optimization. In AAAI-94 [3], pages 607–612.
110. Iman Foroutan. *Feature Selection for Piecewise Linear Classifiers*. PhD thesis, Univ. of California, Irvine, CA, 1985.
111. Iman Foroutan and Jack Sklansky. Feature selection for automatic classification of non-Gaussian data. *IEEE Trans. on Systems, Man and Cybernetics*, 17(2):187–198, March/April 1987.
112. Richard S. Forsyth, David D. Clarke, and Richard L. Wright. Overfitting revisited: an information-theoretic approach to simplifying discrimination trees. *J. of Experimental and Theoretical Artificial Intelligence*, 6(3):289–302, July–September 1994.
113. Jerome H. Friedman. A recursive partitioning decision rule for nonparametric classifiers. *IEEE Trans. on Comp.*, C-26:404–408, April 1977.
114. Keinosuke Fukunaga and R. A. Hayes. Effect of sample size in classifier design. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:873–885, 1989.
115. Truxton K. Fulton, Simon Kasif, and Steven Salzberg. An efficient algorithm for finding multi-way splits for decision trees. In ML-95 [255]. to appear.
116. J. Furnkranz, J. Petrak, and R. Trappl. Knowledge discovery in international conflict databases. *Applied Artificial Intelligence*, 11:91–118, 1997.
117. Michael R. Garey and Ronald L. Graham. Performance bounds on the splitting algorithm for binary testing. *Acta Informatica*, 3(Fasc. 4):347–355, 1974.
118. S. B. Gelfand and C. S. Ravishankar. A tree-structured piecewise-linear adaptive filter. *IEEE Trans. on Information Theory*, 39(6):1907–1922, November 1993.
119. Saul B. Gelfand, C. S. Ravishankar, and Edward J. Delp. An iterative growing and pruning algorithm for classification tree design. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13(2):163–174, February 1991.
120. Edzard S. Gelsema and Laveen S. Kanal, editors. *Pattern Recognition in Practice IV: Multiple paradigms, Comparative studies and hybrid systems*, volume 16 of *Machine Intelligence and Pattern Recognition*. Series editors: Kanal, L. S. and Rozenfeld, A. Elsevier, 1994.
121. G. H. Gennari, Pat Langley, and Douglas Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40(1–3):11–62, September 1989.
122. Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Pub., 1991.
123. W. J. Gibb, D. M. Auslander, and J. C. Griffin. Selection of myocardial electrogram features for use by implantable devices. *IEEE Trans. on Biomedical Eng.*, 40(8):727–735, August 1993.
124. M. W. Gillo. MAID: A Honeywell 600 program for an automatised survey analysis. *Behavioral Science*, 17:251–252, 1972.
125. Elizabeth A. Giplin, Richard A. Olshen, Kanu Chatterjee, John Kjekshus, Arthur J. Moss, Harmut Henning, Robert Engler, A. Robert Blacky, Howard Ditttrich, and John Ross Jr. Predicting 1-year outcome following acute myocardial infarction. *Comp. and biomedical research*, 23(1):46–63, February 1990.
126. Malcolm A. Gleser and Morris F. Collen. Towards automated medical decisions. *Comp. and Biomedical Research*, 5(2):180–189, April 1972.
127. M. Golea and M. Marchand. A growth algorithm for neural network decision trees. *Euro-Physics Letters*, 12(3):205–210, June 1990.
128. Rodney M. Goodman and Padhraic J. Smyth. Decision tree design from a communication theory standpoint. *IEEE Trans. on Information Theory*, 34(5):979–994, September 1988.
129. Rodney M. Goodman and Padhriac J. Smyth. Decision tree design using information theory. *Knowledge Acquisition*, 2:1–19, 1990.

130. Michael T. Goodrich, Vincent Mirelli, Mark Orletsky, and Jeffery Salowe. Decision tree construction in fixed dimensions: Being global is hard but local greed is good. Technical Report TR-95-1, Johns Hopkins Univ., Department of Computer Science, Baltimore, MD 21218, May 1995.
131. L. Gordon and R. A. Olshen. Asymptotically efficient solutions to the classification problem. *Annals of Statistics*, 6(3):515–533, 1978.
132. N. A. B. Gray. Capturing knowledge through top-down induction of decision trees. *IEEE Expert*, 5(3):41–50, June 1990.
133. L. Grewe and A.C. Kak. Interactive learning of a multi-attribute hash table classifier for fast object recognition. *Computer Vision and Image Understanding*, 61(3):387–416, May 1995.
134. Heng Guo and Saul B. Gelfand. Classification trees with neural network feature extraction. *IEEE Trans. on Neural Networks.*, 3(6):923–933, November 1992.
135. Y. Guo and K.J. Dooley. Distinguishing between mean, variance and autocorrelation changes in statistical quality control. *Int. J. of Production Research*, 33(2):497–510, February 1995.
136. Ouzden Guur-Ali and William A. Wallace. Induction of rules subject to a quality constraint: Probabilistic inductive learning. *IEEE Trans. on Knowledge and Data Eng.*, 5(6):979–984, December 1993. Special Issue on Learning and Discovery in Knowledge-based databases.
137. S.E. Hampson and D.J. Volper. Linear function neurons: Structure and training. *Biological Cybernetics*, 53(4):203–217, 1986.
138. D. J. Hand. *Discrimination and Classification*. Wiley, Chichester, UK, 1981.
139. W. Hanisch. Design and optimization of a hierarchical classifier. *J. of new Generation Computer Systems*, 3(2):159–173, 1990.
140. L. K. Hansen and P. Salomon. Neural network ensembles. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
141. A. Hart. Experience in the use of an inductive system in knowledge eng. In M. Bramer, editor, *Research and Development in Expert Systems*. Cambridge Univ. Press, Cambridge, MA, 1984.
142. Carlos R. P. Hartmann, Pramod K. Varshney, Kishan G. Mehrotra, and Carl L. Gerberich. Application of information theory to the construction of efficient decision trees. *IEEE Trans. on Information Theory*, IT-28(4):565–577, July 1982.
143. R. E. Haskell and A. Noui-Mehidi. Design of hierarchical classifiers. In N. A. Sherwani, E. de Doncker, and J. A. Kapenga, editors, *Computing in the 90's: The First Great Lakes Computer Science Conf. Proc.*, pages 118–124, Berlin, 1991. Springer-Verlag. Conf. held in Kalamazoo, MI on 18th-20th, October 1989.
144. N.D. Hatziargyriou, G.C. Contaxis, and N.C. Sideris. A decision tree method for on-line steady state security assessment. *IEEE Trans. on Power Systems*, 9(2):1052, 1994.
145. D. Heath. *A Geometric Framework for Machine Learning*. PhD thesis, Johns Hopkins Univ., Baltimore, MD, 1992.
146. D. Heath, S. Kasif, and S. Salzberg. k-DT: A multi-tree learning method. In *Proc. of the Second Int. Workshop on Multistrategy Learning*, pages 138–149, Harpers Ferry, WV, 1993. George Mason Univ.
147. D. Heath, S. Kasif, and S. Salzberg. Learning oblique decision trees. In IJCAI-93 [160], pages 1002–1007. Editor: Ruzena Bajcsy.
148. D. P. Helmbold and R. E. Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, pages 51–68, 1997. Earlier version in COLT95.
149. Ernest G. Henrichon Jr. and King-Sun Fu. A nonparametric partitioning procedure for pattern classification. *IEEE Trans. on Comp.*, C-18(7):614–624, July 1969.
150. Gabor T. Herman and K.T. Daniel Yeung. On piecewise-linear classification. *IEEE Trans. on PAMI*, 14(7):782–786, July 1992.
151. Klaus-U Hoeffgen, Hans-U Simon, and Kevin S. Van Horn. Robust trainability of single neurons. *J. of Computer System Sciences*, 50(1):114–125, 1995.
152. R. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90, 1993.
153. G. E. Hughes. On the mean accuracy of statistical pattern recognition. *IEEE Trans. on Information Theory*, IT-14(1):55–63, January 1968.
154. K. J. Hunt. Classification by induction: Applications to modelling and control of non-linear dynamic systems. *Intelligent Systems Eng.*, 2(4):231–245, Winter 1993.

155. Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15–17, 1976.
156. Toshihide Ibaraki and Saburo Muroga. Adaptive linear classifiers by linear programming. Technical Report 284, Department of Computer Science, Univ. of Illinois, Urbana-Champaign, 1968.
157. M. Ichino and Jack Sklansky. Optimum feature selection by zero-one integer programming. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-14:737–746, September/October 1984.
158. Y. Iikura and Y. Yasuoka. Utilization of a best linear discriminant function for designing the binary decision tree. *Int. Journal of Remote Sensing*, 12(1):55–67, January 1991.
159. *IJCAI-89: Proc. of the Eleventh Int. Joint Conf. on Artificial Intelligence*. Morgan Kaufmann Pub. Inc., San Mateo, CA, 1989. Editor: N. S. Sridharan.
160. *IJCAI-93: Proc. of the Thirteenth Int. Joint Conf. on Artificial Intelligence*, volume 2, Chambery, France, 28th August–3rd September 1993. Morgan Kaufmann Pub. Inc., San Mateo, CA. Editor: Ruzena Bajcsy.
161. *IJCAI-95: Proc. of the Fourteenth Int. Joint Conf. on Artificial Intelligence*, Montreal, Canada, 16th–21st, August 1995. Morgan Kaufmann Pub. Inc., San Mateo, CA. Editor: Chris Mellish.
162. I. F. Imam and Ryszard S. Michalski. Should decision trees be learned from examples or from decision rules? In *Methodologies for Intelligent Systems: 7th Int. Symposium. ISMIS '93*, volume 689 of *LNCS*, pages 395–404. Springer-Verlag, Trondheim, Norway, June 1993.
163. Keki B. Irani, Cheng Jie, Usama M. Fayyad, and Qian Zhaogang. Applying machine learning to semiconductor manufacturing. *IEEE Expert*, 8(1):41–47, February 1993.
164. P. Israel and C. Koutsougeras. A hybrid electro-optical architecture for classification trees and associative memory mechanisms. *Int. J. on Artificial Intelligence Tools (Architectures, Languages, Algorithms)*, 2(3):373–393, September 1993.
165. Andreas Ittner and Michael Schlosser. Non-linear decision trees - NDT. In *Int. Conf. on Machine Learning*. 1996.
166. A. K. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition. In Krishnaiah and Kanal [198], pages 835–855.
167. George H. John. Robust linear discriminant trees. In *AI&Statistics-95* [7], pages 285–291.
168. George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *ML-94* [254], pages 121–129. Editors: William W. Cohen and Haym Hirsh.
169. Michael I. Jordan. A statistical approach to decision tree modeling. In *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory*, pages 13–20, New Brunswick, New Jersey, 1994. ACM Press.
170. Michael I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.
171. J. Judmaier, P. Meyersbach, G. Weiss, H. Wachter, and G. Reibnegger. The role of Neopterin in assessing disease activity in Crohn's disease: Classification and regression trees. *The American J. of Gastroenterology*, 88(5):706, May 1993.
172. G. Kalkanis. The application of confidence interval error analysis to the design of decision tree classifiers. *Pattern Recognition Letters*, 14(5):355–361, May 1993.
173. Laveen N. Kanal. Patterns in pattern recognition: 1968–1974. *IEEE Trans. in Information Theory*, 20:697–722, 1974.
174. Laveen N. Kanal. Problem solving methods and search strategies for pattern recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-1:193–201, 1979.
175. Laveen N. Kanal and B. Chandrasekaran. On dimensionality and sample size in statistical pattern classification. *Pattern Recognition*, 3:225–234, 1971.
176. G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.
177. Michael Kearns. Boosting theory towards practice: Recent developments in decision tree induction and the weak learning framework. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 1337–1339, Menlo Park, 1996. AAAI Press / MIT Press.
178. Michael Kearns and Yishay Mansour. On the boosting ability of top-down decision tree learning algorithms. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 459–468, Philadelphia, Pennsylvania, 1996.

179. Davis M. Kennedy. Decision tree bears fruit. *Products Finishing*, 57(10):66, July 1993.
180. J. D. Kennefick, R. R. Carvalho, S. G. Djorgovski, M. M. Wilber, E. S. Dickson, N. Weir, U. Fayyad, and J. Roden. The discovery of five quasars at $z > 4$ using the second Palomar Sky Survey. *The Astronomical J.*, 110(1):78, 1995.
181. Randy Kerber. Chimerge: Discretization of numeric attributes. In AAAI-92 [1], pages 123–128.
182. Byungyong Kim and David Landgrebe. Hierarchical decision tree classifiers in high-dimensional and large class data. *IEEE Trans. on Geoscience and Remote Sensing*, 29(4):518–528, July 1991.
183. Hyunsoo Kim and G. J. Koehler. An investigation on the conditions of pruning an induced decision tree. *European J. of Operational Research*, 77(1):82, August 1994.
184. Sung-Ho Kim. A general property among nested, pruned subtrees of a decision support tree. *Communications in Statistics—Theory and Methods*, 23(4):1227–1238, April 1994.
185. Kenji Kira and Larry A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In AAAI-92 [1], pages 129–134.
186. Y. Kodratoff and M. Manago. Generalization and noise. *Int. J. of Man-Machine Studies*, 27:181–204, 1987.
187. Y. Kodratoff and S. Moscatelli. Machine learning for object recognition and scene analysis. *International J. of Pattern recognition and AI*, 8(1):259–304, 1994.
188. Ron Kohavi. Bottom-up induction of oblivious, read-once decision graphs: Strengths and limitations. In AAAI-94 [3].
189. Ron Kohavi. The power of decision tables. In *The European Conference on Machine Learning*, 1995.
190. Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In IJCAI-95 [161], pages 1137–1143. Editor: Chris Mellish.
191. Ron Kohavi. Wrappers for performance enhancements and oblivious decision graphs. Ph.D. Thesis CS-TR-95-1560, Stanford University, Department of Computer Science, September 1995.
192. P. Kokol, M. Mernik, J. Zavrsnik, and K. Kancler. Decision trees based on automatic learning and their use in cardiology. *Journal of Medical Systems*, 18(4):201, 1994.
193. Igor Kononenko. On biases in estimating multi-valued attributes. In IJCAI-95 [161], pages 1034–1040. Editor: Chris Mellish.
194. Igor Kononenko and Ivan Bratko. Information based evaluation criterion for classifier's performance. *Machine Learning*, 6(1):67–80, January 1991.
195. J. A. Kors and J. H. Van Bommel. Classification methods for computerized interpretation of the electrocardiogram. *Methods of Information in Medicine*, 29(4):330–336, September 1990.
196. V. A. Kovalevsky. The problem of character recognition from the point of view of mathematical statistics. In V. A. Kovalevsky, editor, *Character Readers and Pattern Recognition*. Spartan, New York, 1968.
197. J. R. Koza. Concept formation and decision tree induction using the genetic programming paradigm. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature - Proc. of 1st Workshop, PPSN 1*, volume 496 of *LNCs*, pages 124–128, Dortmund, Germany, October 1991. Springer-Verlag, Berlin, Germany.
198. Paruchuri Rama Krishnaiah and Laveen N. Kanal, editors. *Classification, Pattern Recognition and Reduction of Dimensionality*, volume 2 of *Handbook of Statistics*. North-Holland Publishing Company, Amsterdam, 1987.
199. Srinivasan Krishnamoorthy and Douglas Fisher. Machine learning approaches to estimating software development effort. *IEEE Trans. on Software Eng.*, 21(2):126–137, February 1995.
200. M. Kroger. Optimization of classification trees: strategy and algorithm improvement. *Computer Physics Communications*, 99(1):81–93, December 1996.
201. M. Kubat, G. Pfurtscheller, and D. Flotzinger. AI-based approach to automatic sleep classification. *Biological Cybernetics*, 70(5):443–448, 1994.
202. Ashok K. Kulkarni. On the mean accuracy of hierarchical classifiers. *IEEE Trans. on Comp.*, C-27(8):771–776, August 1978.

203. Michael J. Kurtz. Astronomical object classification. In E. S. Gelsema and Laveen N. Kanal, editors, *Pattern Recognition and Artificial Intelligence*, pages 317–328. Elsevier Science Pub., Amsterdam, 1988.
204. M. W. Kurzynski. The optimal strategy of a tree classifier. *Pattern Recognition*, 16:81–87, 1983.
205. M. W. Kurzynski. On the multi-stage Bayes classifier. *Pattern Recognition*, 21(4):355–365, 1988.
206. M. W. Kurzynski. On the identity of optimal strategies for multi-stage classifiers. *Pattern Recognition Letters*, 10(1):39–46, July 1989.
207. S.W. Kwok and Carter. C. Multiple decision trees. In R.D. Schachter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, volume 4, pages 327–335. Elsevier Science, Amsterdam, 1990.
208. P. Lagacherie and S. Holmes. Addressing geographical data errors in classification tree for soil unit prediction. *Int. J. of Geographical Information Science*, 11(2):183–198, March 1997.
209. G. Landeweerd, T. Timmers, E. Gersema, M. Bins, and M. Halic. Binary tree versus single level tree classification of white blood cells. *Pattern Recognition*, 16:571–577, 1983.
210. Pat Langley and Stephanie Sage. Scaling to domains with irrelevant features. In Thomas Petsche and Stephen Jose Hanson Russell Greiner, editor, *Computational Learning Theory and Natural Learning Systems*, volume vol-IV. MIT Press, 1997.
211. Seong-Whan Lee. Noisy Hangul character recognition with fuzzy tree classifier. *Proc. of SPIE*, 1661:127–136, 1992. Volume title: Machine vision applications in character recognition and industrial inspection. Conf. location: San Jose, CA. 10th–12th February, 1992.
212. Wendy Lehnert, Stephen Soderland, David Aronow, Fangfang Feng, and Avinoam Shmueli. Inductive text classification for medical applications. *Journal of Experimental and Theoretical Artificial Intelligence*, 7(1):49–80, January-March 1995.
213. P.M. Lewis. The characteristic selection problem in recognition systems. *IRE Trans. on Information Theory*, IT-18:171–178, 1962.
214. Xiaobo Li and Richard C. Dubes. Tree classifier design with a permutation statistic. *Pattern Recognition*, 19(3):229–235, 1986.
215. Jianhia Lin and L.A. Storer. Design and performance of tree structured vector quantizers. *Information Processing and Management*, 30(6):851–862, 1994.
216. Jianhua Lin, J. A. Storer, and M. Cohn. Optimal pruning for tree-structured vector quantizers. *Information Processing and Management*, 28(6):723–733, 1992.
217. Jyh-Han Lin and J. S. Vitter. Nearly optimal vector quantization via linear programming. In J. A. Storer and M. Cohn, editors, *DCC 92. Data Compression Conf.*, pages 22–31, Los Alamitos, CA, March 24th–27th 1992. IEEE Computer Society Press.
218. Y. K. Lin and King-Sun Fu. Automatic classification of cervical cells using a binary tree classifier. *Pattern Recognition*, 16(1):69–80, 1983.
219. W. Z. Liu and A. P. White. The importance of attribute selection measures in decision tree induction. *Machine Learning*, 15:25–41, 1994.
220. Wei-Yin Loh and Nunta Vanichsetakul. Tree-structured classification via generalized discriminant analysis. *J. of the American Statistical Association*, 83(403):715–728, September 1988.
221. William J. Long, John L. Griffith, Harry P. Selker, and Ralph B. D’Agostino. A comparison of logistic regression to decision tree induction in a medical domain. *Comp. and Biomedical Research*, 26(1):74–97, February 1993.
222. D.W. Loveland. Performance bounds for binary testing with arbitrary weights. *Acta Informatica*, 22:101–114, 1985.
223. David Lubinsky. Algorithmic speedups in growing classification trees by using an additive split criterion. In AI&Statistics-93 [6], pages 435–444.
224. David Lubinsky. *Bivariate splits and consistent split criteria in dichotomous classification trees*. PhD thesis, Department of Computer Science, Rutgers Univ., New Brunswick, NJ, 1994.
225. David Lubinsky. Classification trees with bivariate splits. *Applied Intelligence: The Int. J. of Artificial Intelligence, Neural Networks and Complex Problem-Solving Technologies*, 4(3):283–296, July 1994.

226. David Lubinsky. Tree structured interpretable regression. In *AI&Statistics-95* [7], pages 331–340.
227. Ren C. Luo, Ralph S. Scherp, and Mark Lanzo. Object identification using automated decision tree construction approach for robotics applications. *J. of Robotic Systems*, 4(3):423–433, June 1987.
228. J. F. Lutsko and B. Kuijpers. Simulated annealing in the construction of near-optimal decision trees. In *AI&Statistics-93* [6].
229. David M. Magerman. Natural language parsing as statistical pattern recognition. Thesis CS-TR-94-1502, Stanford University, Department of Computer Science, February 1994.
230. Olvi Mangasarian. Mathematical programming in neural networks. *ORSA J. on Computing*, 5(4):349–360, Fall 1993.
231. Olvi L. Mangasarian. Misclassification minimization, 1994. Unpublished manuscript.
232. Olvi L. Mangasarian, R. Setiono, and W. Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. In *SIAM Workshop on Optimization*, 1990.
233. López de Màntaras. Technical note: A distance-based attribute selection measure for decision tree induction. *Machine Learning*, 6(1):81–92, 1991.
234. J. Kent Martin. Evaluating and comparing classifiers: complexity measures. In *AI&Statistics-95* [7], pages 372–378.
235. J. Kent Martin. An exact probability metric for decision tree splitting and stopping. *Machine Learning*, 28:257–291, 1997.
236. J. Kent Martin and Daniel S. Hirschberg. The time complexity of decision tree induction. Technical Report ICS-TR-95-27, University of California, Irvine, Department of Information and Computer Science, August 1995.
237. Dean P. McKenzie and Lee Hun Low. The construction of computerized classification systems using machine learning algorithms: An overview. *Comp. in Human Behaviour*, 8(2/3):155–167, 1992.
238. Dean P. McKenzie, P. D. McGorry, C. S. Wallace, Lee Hun Low, D. L. Copolov, and B. S. Singh. Constructing a minimal diagnostic decision tree. *Methods of Information in Medicine*, 32(2):161–166, April 1993.
239. R.J. McQueen, S. R. Garner, C.G. Nevill-Manning, and I.H. Witten. Applying machine learning to agricultural data. *Comp. and Electronics in Agriculture*, 12(4):275–293, June 1995.
240. Nimrod Megiddo. On the complexity of polyhedral separability. *Discrete and Computational Geometry*, 3:325–337, 1988.
241. William S. Meisel and Demetrios A. Michalopoulos. A partitioning algorithm with application in pattern classification and the optimization of decision trees. *IEEE Trans. on Comp.*, C-22(1):93–103, January 1973.
242. Joseph J. Mezrich. When is a tree a hedge? *Financial Analysts J.*, pages 75–81, November–December 1994.
243. Donald Michie. The superarticulatory phenomenon in the context of software manufacture. *Proc. of the Royal Society of London*, 405A:185–212, 1986.
244. Spiegelhalter Michie and Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994. The Statlog Project.
245. A. J. Miller. *Subset Selection in Regression*. Chapman and Hall, 1990.
246. John Mingers. Expert systems — rule induction with statistical data. *J. of the Operational Research Society*, 38(1):39–47, 1987.
247. John Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4(2):227–243, 1989.
248. John Mingers. An empirical comparison of selection measures for decision tree induction. *Machine Learning*, 3:319–342, 1989.
249. M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.
250. Tom Mitchell, Rich Caruana, Dayne Freitag, John McDermott, and David Zabowski. Experience with a learning personal assistant. *Communications of the ACM*, July 1994.
251. Masahiro Miyakawa. Optimum decision trees – an optimal variable theorem and its related applications. *Acta Informatica*, 22(5):475–498, 1985.

252. Masahiro Miyakawa. Criteria for selecting a variable in the construction of efficient decision trees. *IEEE Trans. on Comp.*, 38(1):130–141, January 1989.
253. *Machine Learning: Proc. of the Tenth Int. Conf.*, Univ. of Massachusetts, Amherst, MA, 27–29th, June 1993. Morgan Kaufmann Pub. Inc. Editor: Paul E. Utgoff.
254. *Machine Learning: Proc. of the Eleventh Int. Conf.*, Rutgers Univ., New Brunswick, NJ, 10–13th, July 1994. Morgan Kaufmann Pub. Inc. Editors: William W. Cohen and Haym Hirsh.
255. *Machine Learning: Proc. of the Twelfth Int. Conf.*, Tahoe City, CA, 10–13th, July 1995. Morgan Kaufmann Pub. Inc., San Mateo, CA. Editor: Jeffrey Schlimmer.
256. Advait Mogre, Robert McLaren, James Keller, and Raghuram Krishnapuram. Uncertainty management for rule-based systems with application to image analysis. *IEEE Trans. on Systems, Man and Cybernetics*, 24(3):470–481, March 1994.
257. Andrew W. Moore and Mary S. Lee. Efficient algorithms for minimizing cross validation error. In ML-94 [254], pages 190–198. Editors: William W. Cohen and Haym Hirsh.
258. Bernard M. E. Moret, M. G. Thomason, and R. C. Gonzalez. The activity of a variable and its relation to decision trees. *ACM Trans. on Programming Language Systems*, 2(4):580–595, October 1980.
259. Bernard M.E. Moret. Decision trees and diagrams. *Computing Surveys*, 14(4):593–623, December 1982.
260. J. N. Morgan and R. C. Messenger. THAID: a sequential search program for the analysis of nominal scale dependent variables. Technical report, Institute for Social Research, Univ. of Michigan, Ann Arbor, MI, 1973.
261. D. T. Morris and D. Kalles. Decision trees and domain knowledge in pattern recognition. In Gelsema and Kanal [120], pages 25–36.
262. A. N. Mucciardi and E. E. Gose. A comparison of seven techniques for choosing subsets of pattern recognition properties. *IEEE Trans. on Comp.*, C-20(9):1023–1031, September 1971.
263. W. Muller and F. Wysotzki. Automatic construction of decision trees for classification. *Annals of Operations Research*, 52:231, 1994.
264. O. J. Murphy and R. L. McCraw. Designing storage efficient decision trees. *IEEE Trans. on Comp.*, 40(3):315–319, March 1991.
265. Patrick M. Murphy. An empirical analysis of the benefit of decision tree size biases as a function of concept distribution. Submitted to the Machine Learning journal, July 1994.
266. Patrick M. Murphy and David Aha. UCI repository of machine learning databases – a machine-readable data repository. Maintained at the Department of Information and Computer Science, Univ. of California, Irvine. Anonymous FTP from ics.uci.edu in the directory pub/machine-learning-databases, 1994.
267. Patrick M. Murphy and Michael J. Pazzani. Exploring the decision forest: An empirical investigation of Occam's Razor in decision tree induction. *J. of Artificial Intelligence Research*, 1:257–275, 1994.
268. Sreerama K. Murthy, S. Kasif, S. Salzberg, and R. Beigel. OC1: Randomized induction of oblique decision trees. In AAAI-93 [2], pages 322–327.
269. Sreerama K. Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *J. of Artificial Intelligence Research*, 2:1–33, August 1994.
270. Sreerama K. Murthy and Steven Salzberg. Decision tree induction: How effective is the greedy heuristic? In *Proc. of the First Int. Conf. on Knowledge Discovery in Databases*, Montreal, Canada, August 1995.
271. Sreerama K. Murthy and Steven Salzberg. Lookahead and pathology in decision tree induction. In IJCAI-95 [161]. to appear.
272. P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. on Comp.*, C-26(9):917–922, 1977.
273. Dana S. Nau. Decision quality as a function of search depth on game trees. *J. of the Association of Computing Machinery*, 30(4):687–708, October 1983.
274. G. E. Naumov. NP-completeness of problems of construction of optimal decision trees. *Soviet Physics, Doklady*, 36(4):270–271, April 1991.
275. T. Niblett. Constructing decision trees in noisy domains. In I. Bratko and N. Lavrac, editors, *Progress in Machine Learning*. Sigma Press, England, 1986.

276. N.J. Nilsson. *Learning Machines*. Morgan Kaufmann, 1990.
277. T. Nilsson, T. Lundgren, H. Odelius, R. Sillen, and J.G. Noren. A computerized induction analysis of possible co-variations among different elements in human tooth enamel. *Artificial Intelligence in Medicine*, 8(6):515–526, November 1996.
278. Steven W. Norton. Generating better decision trees. In IJCAI-89 [159], pages 800–805. Editor: N. S. Sridharan.
279. M. Núñez. The use of background knowledge in decision tree induction. *Machine Learning*, 6:231–250, 1991.
280. Tim Oates and David Jensen. The effects of training set size on decision tree complexity. In *Proceedings of the 14th International Conference on Machine Learning*, pages 254–262. Morgan Kaufmann, 1997.
281. J. Oliver. Decision graphs—an extension of decision trees. In AI&Statistics-93 [6].
282. Colm A. O’Muircheartaigh. Statistical analysis in the context of survey research. In O’Muircheartaigh and Payne [283], pages 1–40.
283. Colm A. O’Muircheartaigh and Clive Payne, editors. *The analysis of survey data*, volume I. John Wiley & Sons, Chichester, UK, 1977.
284. Giulia M. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 5(1):71–99, March 1990.
285. C. D. Page and S. Muggleton. How U-learnability fits machine learning practice: a learnability result for the decision tree learner CART. In *Proceedings of the Conference on Applied Decision Technologies (ADT’95). Volume 1: Computational Learning and Probabilistic Reasoning*, pages 325–342, Uxbridge, UK, April 1995. Unicom Seminars.
286. N.R. Pal, S. Chakraborty, and A. Bagchi. RID3: An id3-like algorithm for real data. *Information Sciences*, 96(3-4):271–290, February 1997.
287. Shailendra C. Palvia and Steven R. Gordon. Tables, trees and formulas in decision analysis. *Communications of the ACM*, 35(10):104–113, October 1992.
288. Youngtae Park. A comparison of neural net classifiers and linear tree classifiers: Their similarities and differences. *Pattern Recognition*, 27(11):1493–1503, 1994.
289. Youngtae Park and Jack Sklansky. Automated design of linear tree classifiers. *Pattern Recognition*, 23(12):1393–1412, 1990.
290. Yountae Park and Jack Sklansky. Automated design of multiple-class piecewise linear classifiers. *J. of Classification*, 6:195–222, 1989.
291. Krishna R. Pattipati and Mark G. Alexandridis. Application of heuristic search and information theory to sequential fault diagnosis. *IEEE Trans. on Systems, Man and Cybernetics*, 20(4):872–887, July/August 1990.
292. R. W. Payne and D. A. Preece. Identification keys and diagnostic tables: A review. *J. of the Royal Statistical Society: series A*, 143:253, 1980.
293. R. A. Pearson and P. E. Stokes. Vector evaluation in induction algorithms. *Int. J. of High Speed Computing*, 2(1):25–100, March 1990.
294. P. Perner, T. B. Belikova, and N. I. Yashunskaya. Knowledge acquisition by symbolic decision tree induction for interpretation of digital images in radiology. *Lecture Notes in Computer Science*, 1121:208, 1996.
295. F. Pipitone, K. A. De Jong, and W. M. Spears. An artificial intelligence approach to analog systems diagnosis. In Ruey-wen Liu, editor, *Testing and Diagnosis of Analog Circuits and Systems*. Van Nostrand-Reinhold, New York, 1991.
296. Selwyn Piramuthu, Narayan Raman, and Michael J. Shaw. Learning-based scheduling in a flexible manufacturing flow line. *IEEE Trans. on Eng. Management*, 41(2):172–182, May 1994.
297. N. J. Pizzi and D. Jackson. Comparative review of knowledge eng. and inductive learning using data in a medical domain. *Proc. of the SPIE: The Int. Society for Optical Eng.*, 1293(2):671–679, April 1990.
298. Shi Qing-Yun and King-Sun Fu. A method for the design of binary tree classifiers. *Pattern Recognition*, 16:593–603, 1983.
299. John Ross Quinlan. Discovering rules by induction from large collections of examples. In Donald Michie, editor, *Expert Systems in the Micro Electronic Age*. Edinburgh Univ. Press, Edinburgh, UK, 1979.

300. John Ross Quinlan. The effect of noise on concept learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, volume 2. Morgan Kaufmann, San Mateo, CA, 1986.
301. John Ross Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
302. John Ross Quinlan. Simplifying decision trees. *Int. J. of Man-Machine Studies*, 27:221–234, 1987.
303. John Ross Quinlan. An empirical comparison of genetic and decision tree classifiers. In *Fifth Int. Conf. on Machine Learning*, pages 135–141, Ann Arbor, Michigan, 1988. Morgan Kaufmann.
304. John Ross Quinlan. Unknown attribute values in induction. In *Proc. of the Sixth Int. Workshop on Machine Learning*, pages 164–168, San Mateo, CA, 1989. Morgan Kaufmann.
305. John Ross Quinlan. Probabilistic decision trees. In R.S. Michalski and Y. Kodratoff, editors, *Machine Learning: An Artificial Intelligence Approach - Volume 3*. Morgan Kaufmann, San Mateo, CA, 1990.
306. John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Pub., San Mateo, CA, 1993.
307. John Ross Quinlan. Comparing connectionist and symbolic learning methods. In S. Hanson, G. Drastal, and R. Rivest, editors, *Computational Learning Theory and Natural Learning Systems: Constraints and Prospects*. MIT Press, 1993.
308. John Ross Quinlan. Improved use of continuous attributes in C4.5. *J. of Artificial Intelligence Research*, 4:77–90, March 1996.
309. John Ross Quinlan and Ronald L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80(3):227–248, March 1989.
310. Harish Ragavan and Larry Rendell. Lookahead feature construction for learning hard concepts. In ML-93 [253], pages 252–259. Editor: Paul E. Utgoff.
311. Larry Rendell and Harish Ragavan. Improving the design of induction methods by analyzing algorithm functionality and data-based concept complexity. In IJCAI-93 [160], pages 952–958. Editor: Ruzena Bajcsy.
312. Alfred Renyi and Laszlo Vekerdi. *Probability Theory*. North-Holland Publishing Company, Amsterdam, 1970.
313. P. Riddle, R. Segal, and O. Etzioni. Representation design and brute-force induction in a Boeing manufacturing domain. *Applied Artificial Intelligence*, 8(1):125–147, January-March 1994.
314. Jorma Risannen. *Stochastic Complexity in Statistical Enquiry*. World Scientific, 1989.
315. Eve A. Riskin and Robert M. Gray. Lookahead in growing tree-structured vector quantizers. In *ICASSP 91: Int. Conf. on Acoustics, Speech and Signal Processing*, volume 4, pages 2289–2292, Toronto, Ontario, May 14th–17th 1991. IEEE.
316. E. Rounds. A combined non-parametric approach to feature selection and binary decision tree design. *Pattern Recognition*, 12:313–317, 1980.
317. Steven Rovnyak, Stein Kretsinger, James Thorp, and Donald Brown. Decision trees for real time transient stability prediction. *IEEE Trans. on Power Systems*, 9(3):1417–1426, August 1994.
318. Ron Rymon. An SE-tree based characterization of the induction problem. In ML-93 [253], pages 268–275. Editor: Paul E. Utgoff.
319. Ron Rymon and N. M. Short, Jr. Automatic cataloging and characterization of earth science data using set enumeration trees. *Telematics and Informatics*, 11(4):309–318, Fall 1994.
320. S. Rasoul Safavin and David Landgrebe. A survey of decision tree classifier methodology. *IEEE Trans. on Systems, Man and Cybernetics*, 21(3):660–674, May/June 1991.
321. M. Sahami. Learning non-linearly separable boolean functions with linear threshold unit trees and madaline-style networks. In AAAI-93 [2], pages 335–341.
322. Steven Salzberg. Locating protein coding regions in human DNA using a decision tree algorithm. *J. of Computational Biology*, 1995. To appear in Fall.
323. Steven Salzberg, Rupali Chandar, Holland Ford, Sreerama Murthy, and Rick White. Decision trees for automated identification of cosmic-ray hits in Hubble Space Telescope images. *Publications of the Astronomical Society of the Pacific*, 107:1–10, March 1995.
324. Anant Sankar and Richard J. Mammone. Growing and pruning neural tree networks. *IEEE Trans. on Comp.*, 42(3):291–299, March 1993.

325. Lawrence Saul and Michael I. Jordan. Learning in Boltzmann trees. *Neural Computation*, 6(6):1174–1184, November 1994.
326. Cullen Schaffer. Overfitting avoidance as bias. *Machine Learning*, 10:153–178, 1993.
327. Cullen Schaffer. A conservation law for generalization performance. In ML-94 [254], pages 259–265. Editors: William W. Cohen and Haym Hirsh.
328. Cullen Schaffer. Conservation of generalization: A case study. Technical report, Department of Computer Science, CUNY/Hunter College, February 1995.
329. T. M. Schmidl, P. C. Cosman, and Robert M. Gray. Unbalanced non-binary tree-structured vector quantizers. In A. Singh, editor, *Conf. Record of the Twenty-Seventh Asilomar Conf. on Signals, Systems and Comp.*, volume 2, pages 1519–1523, Los Alamitos, CA, November 1st–3rd 1993. IEEE Computer Society Press. Conf. held at Pacific Grove, CA.
330. J. Schuermann and W. Doster. A decision-theoretic approach in hierarchical classifier design. *Pattern Recognition*, 17:359–369, 1984.
331. Ishwar Krishnan Sethi. Entropy nets: From decision trees to neural networks. *Proc. of the IEEE*, 78(10), October 1990.
332. Ishwar Krishnan Sethi and B. Chatterjee. Efficient decision tree design for discrete variable pattern recognition problems. *Pattern Recognition*, 9:197–206, 1977.
333. Ishwar Krishnan Sethi and G.P.R. Sarvarayudu. Hierarchical classifier design using mutual information. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-4(4):441–445, July 1982.
334. Ishwar Krishnan Sethi and J. H. Yoo. Design of multicategory, multifeature split decision trees using perceptron learning. *Pattern Recognition*, 27(7):939–947, 1994.
335. Nong Shang and Leo Breiman. Distribution based trees are more accurate. In *Proc. of the Int. Conf. on Neural Information Processing*, pages 133–138, 1996.
336. C. E. Shannon. A mathematical theory of communication. *Bell System Technical J.*, 27:379–423, 623–656, 1948.
337. Jude W. Shavlik, R. J. Mooney, and G. G. Towell. Symbolic and neural learning algorithms: An empirical comparison. *Machine Learning*, 6(2):111–144, 1991.
338. S. Shimozone, A. Shinohara, T. Shinohara, S. Miyano, S. Kuhara, and S. Arikawa. Knowledge acquisition from amino acid sequences by machine learning system BONSAI. *Trans. of the Information Processing Society of Japan*, 35(10):2009–2018, October 1994.
339. Seymour Shlien. Multiple binary decision tree classifiers. *Pattern Recognition*, 23(7):757–763, 1990.
340. Seymour Shlien. Nonparametric classification using matched binary decision trees. *Pattern Recognition Letters*, 13(2):83–88, February 1992.
341. W. Siedlecki and J. Skalsky. On automatic feature selection. *Int. J. of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.
342. J.A. Sirat and J.-P. Nadal. Neural trees: A new tool for classification. *Network: Computation in Neural Systems*, 1(4):423–438, October 1990.
343. Jack Sklansky and Leo Michelotti. Locally trained piecewise linear classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-2(2):101–111, March 1980.
344. Jack Sklansky and Gustav Nicholas Wassel. *Pattern classifiers and trainable machines*. Springer-Verlag, New York, 1981.
345. Padhraic Smyth, Alexander Gray, and Usama M. Fayyad. Retrofitting decision tree classifiers using kernel density estimation. In *Proc. 12th International Conference on Machine Learning*, pages 506–514. Morgan Kaufmann, 1995.
346. J. A. Sonquist, E. L. Baker, and J. N. Morgan. *Searching for Structure*. Institute for Social Research, Univ. of Michigan, Ann Arbor, MI, 1971.
347. S. Schwartz, J. Wiles, I. Gough, and S. Philips. Connectionist, rule-based and bayesian decision aids: An empirical comparison. pages 264–278. Chapman & Hall, London, 1993.
348. C. Y. Suen and Qing Ren Wang. ISOETRP – an interactive clustering algorithm with new objectives. *Pattern Recognition*, 17:211–219, 1984.
349. Xiaorong Sun, Yuping Qiu, and Louis Anthony Cox. A hill-climbing approach to construct near-optimal decision trees. In *AI&Statistics-95* [7], pages 513–519.
350. P. Swain and H. Hauska. The decision tree classifier design and potential. *IEEE Trans. on Geoscience and Electronics*, GE-15:142–147, 1977.

351. Jan L. Talmon. A multiclass nonparametric partitioning algorithm. *Pattern Recognition Letters*, 4:31–38, 1986.
352. Jan L. Talmon, Willem R. M. Dassen, and Vincent Karthaus. Neural nets and classification trees: A comparison in the domain of ECG analysis. In Gelsema and Kanal [120], pages 415–423.
353. Jan L. Talmon and P. McNair. The effect of noise and biases on the performance of machine learning algorithms. *Int. J. of Bio-Medical Computing*, 31(1):45–57, July 1992.
354. Ming Tan. Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning*, 13:7–33, 1993.
355. Paul C. Taylor and Bernard W. Silverman. Block diagrams and splitting criteria for classification trees. *Statistics and Computing*, 3(4):147–161, December 1993.
356. Sebastian Thrun and et al. The monk's problems: A performance comparison of different learning algorithms. Technical Report CMU-CS-91-197, School of Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA, 1991.
357. R. Todeshini and E. Marengo. Linear discriminant classification tree: a user-driven multi-criteria classification method. *Chemometrics and Intelligent Lab. Systems*, 16:25–35, 1992.
358. Pei-Lei Tu and Jen-Yao Chung. A new decision-tree classification algorithm for machine learning. In *Proc. of the IEEE Int. Conf. on Tools with AI*, pages 370–377, Arlington, Virginia, November 1992.
359. I. B. Turksen and H. Zhao. An equivalence between inductive learning and pseudo-Boolean logic simplification: a rule generation and reduction scheme. *IEEE Trans. on Systems, Man and Cybernetics*, 23(3):907–917, May-June 1993.
360. Peter D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409, March 1995.
361. Paul E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.
362. Paul E. Utgoff. Perceptron trees: A case study in hybrid concept representations. *Connection Science*, 1(4):377–391, 1989.
363. Paul E. Utgoff. An improved algorithm for incremental induction of decision trees. In ML-94 [254], pages 318–325. Editors: William W. Cohen and Haym Hirsh.
364. Paul E. Utgoff, Neil C. Berkman, and Jeffery A. Clouse. Decision tree induction based on efficient tree restructuring. *Machine Learning*, 29:5–44, 1997.
365. Paul E. Utgoff and Carla E. Brodley. An incremental method for finding multivariate splits for decision trees. In *Proc. of the Seventh Int. Conf. on Machine Learning*, pages 58–65, Los Altos, CA, 1990. Morgan Kaufmann.
366. J.M. Van Campenhout. *On the Problem of Measurement Selection*. PhD thesis, Stanford Univ., Dept. of Electrical Eng., 1978.
367. Thierry Van de Merckt. Decision trees in numerical attribute spaces. In IJCAI-93 [160], pages 1016–1021. Editor: Ruzena Bajcsy.
368. P.K. Varshney, C.R.P. Hartmann, and J.M. De Faria Jr. Applications of information theory to sequential fault diagnosis. *IEEE Trans. on Comp.*, C-31(2):164–170, 1982.
369. Walter Van de Velde. Incremental induction of topologically minimal trees. In Bruce W. Porter and Ray J. Mooney, editors, *Proc. of the Seventh Int. Conf. on Machine Learning*, pages 66–74, Austin, Texas, 1990.
370. C. S. Wallace and D. M. Boulton. An information measure for classification. *Computer J.*, 11:185–194, 1968.
371. C. S. Wallace and J. D. Patrick. Coding decision trees. *Machine Learning*, 11(1):7–22, April 1993.
372. Qing Ren Wang and C. Y. Suen. Analysis and design of a decision tree based on entropy reduction and its application to large character set recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:406–417, 1984.
373. Qing Ren Wang and Ching Y. Suen. Large tree classifier with heuristic search and global training. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(1):91–102, January 1987.
374. Gustav Nicholas Wassel and Jack Sklansky. Training a one-dimensional classifier to minimize the probability of error. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-2:533–541, September 1972.

375. Larry Watanabe and Larry Rendell. Learning structural decision trees from examples. volume 2, pages 770–776, Darling Harbour, Sydney, Australia, 24–30th, August 1991. Morgan Kaufmann Pub. Inc., San Mateo, CA. Editors: John Mylopoulos and Ray Reiter.
376. S. Watanabe. Pattern recognition as a quest for minimum entropy. *Pattern Recognition*, 13:381–387, 1981.
377. Nicholas Weir, S. Djorgovski, and Usama M. Fayyad. Initial galaxy counts from digitized POSS-II. *The Astronomical J.*, 110(1):1, 1995.
378. Nicholas Weir, Usama M. Fayyad, and S. Djorgovski. Automated star/galaxy classification for digitized POSS-II. *The Astronomical J.*, 109(6):2401, 1995.
379. S. Weiss and I. Kapouleas. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In IJCAI-89 [159], pages 781–787. Editor: N. S. Sridharan.
380. Allan P. White and Wei Zhang Liu. Technical note: Bias in information-based measures in decision tree induction. *Machine Learning*, 15(3):321–329, June 1994.
381. P.A.D. Wilks and M.J. English. Accurate segmentation of respiration waveforms from infants enabling identification and classification of irregular breathing patterns. *Medical Eng. and Physics*, 16(1):19–23, January 1994.
382. J. Wirth and J. Catlett. Experiments on the costs and benefits of windowing in ID3. In *Fifth Int. Conf. on Machine Learning*, pages 87–99, Ann Arbor, Michigan, 1988. Morgan Kaufmann.
383. David H. Wolpert. On overfitting avoidance as bias. Technical Report SFI TR 92-03-5001, The Santa Fe Institute, 1992.
384. David H. Wolpert. On the connection between in-sample testing and generalization error. *Complex Systems*, 6:47–94, 1992.
385. K. S. Woods, C. C. Doss, K. W. Vowyer, J. L. Solka, C. E. Prieve, and W. P. Jr. Kegelmeyer. Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *Int. J. of Pattern Recognition and Artificial Intelligence*, 7(6):1417–1436, December 1993.
386. K. C. You and King-Sun Fu. An approach to the design of a linear binary tree classifier. In *Proc. of the Third Symposium on Machine Processing of Remotely Sensed Data*, West Lafayette, IN, 1976. Purdue Univ.
387. Y. Yuan and M. J. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69(2):125, 1995.
388. Wang Zhengou and Lin Yan. A new inductive learning algorithm: Separability-Based Inductive learning algorithm. *Acta Automatica Sinica*, 5(3):267–270, 1993. Translated into *Chinese J. of Automation*.
389. Xiao Jia Zhou and Tharam S. Dillon. A statistical-heuristic feature selection criterion for decision tree induction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-13(8):834–841, August 1991.
390. Seth Zimmerman. An optimal search procedure. *The American Mathematical Monthly*, 66(8):690–693, March 1959.

Contributing Authors

Sreerama K. Murthy received a Ph.D. in Computer Science from the Johns Hopkins University, Baltimore, MD in 1995. Prior to that, he studied at the Indian Institute of Technology, Madras, India and the Motilal Nehru Regional Engineering College, Allahabad, India. Since 1995, Dr. Murthy has been working in the Imaging & Visualization department at Siemens Corporate Research, Princeton, NJ. Dr. Murthy is interested in unifying decision tree work from multiple disciplines, and in finding new applications of decision trees, particularly in image analysis and computer aided diagnosis.