A Comprehensive Case Study: An Examination of Machine Learning and Connectionist Algorithms

A Thesis

Presented to the

Department of Computer Science

Brigham Young University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Frederick Zarndt

fzarndt@novell.com

June 1995

Contents

1. Introduction 1

2. Databases 6
   - 2.1 Database Names 7
   - 2.2 Database Statistics 11

3. Learning Models 20
   - 3.1 Decision Trees 20
   - 3.2 Nearest Neighbor 21
   - 3.3 Statistical 21
   - 3.4 Rule Based 22
   - 3.6 Neural Networks 22

4. Method 24
   - 4.1 Cross-Validation 24
   - 4.2 Consistency of Training/Testing Sets and Presentation Order 26
   - 4.3 Attribute Discretization 26
   - 4.4 Unknown Attribute Values 28
   - 4.5 Learning Model Parameters 30

5. Results 31

6. Limitations 52

7. Conclusion 53

Appendix A: Standard Database Format (SDF) 55

Appendix B: Tools 58
   - B.1 Database Tools 58
     - B.1.1 *verify* 58
     - B.1.2 *verify.ksh* 58
     - B.1.3 *dsstats* 58
     - B.1.4 *dsstats.ksh* 59
     - B.1.5 *xlate* 59
   - B.2 Test Tools 61
     - B.2.1 *xval.<algorithm>* 61
     - B.2.2 *test-ml* 66
     - B.2.3 *test-conn* 67
   - B.3 Result Extraction Tools 68
     - B.3.1 *average* 68
     - B.3.2 *summarize* 68

Appendix C: Database Generators 69

Appendix D. Learning Model Parameters 70
D.1 Decision Trees 70
D.2 Nearest Neighbor 70
D.3 Statistical 71
D.4 Rule Based 71
D.5 Neural Networks 72

Appendix E. Database Descriptions 80

Bibliography 85

A Comprehensive Case Study: An Examination

of Machine Learning and Connectionist Algorithms

Frederick Zarndt

Department of Computer Science

M.S. Degree, June 1995

ABSTRACT

This case study examines the performance of 16 well-known and widely-used inductive learning algorithms on a comprehensive collection (102) of commonly available learning problems. It is distinguished from other, similar studies by the number of learning models used, the number of problems examined, and the rigor with which it has been conducted. The results of future case studies, which are done with the method and tools presented here, with learning problems used in this study, and with the same rigor, should be readily reproducible and directly comparable to the results of this study. An extensive set of tools is offered.

COMMITTEE APPROVAL:

Tony Martinez, Committee Chairman

Evan Ivie, Committee Member

Douglas Campbell, Committee Member

Tony Martinez, Graduate Coordinator

1. Introduction

The performance of machine learning and connectionist algorithms is frequently evaluated by testing the algorithms against one or more databases. Such an evaluation is called a case study. Implicitly or explicitly, case studies hypothesize that their results are not limited by the data examined but also apply to a general class of learning problems and databases [8]. Likewise, implicitly or explicitly, case studies assume that their results are reproducible in other case studies. There are three factors which often complicate the comparison of the results of one case study with another: (1) A different set of learning models is used in each study; (2) A different set of databases is used in each study, and (3) The results of a study are not reproducible.

While the machine learning/connectionist research community has a good "feel" for the relative performances of the learning models on a set of databases, there is no firm theoretical or empirical basis for extending the results of case studies to include other "similar" learning models or databases. Furthermore, since the characteristics of the databases are often not well understood (except to say that a database is "simple" or "complex"), the anticipated performance of the learning models may differ substantially from the actual performance [8, 108].

When the performance of one or more learning algorithms is reported in a case study, it should be expected, since one of the basic tenets of science is reproducibility of results, that the results of the study are reproducible by another researcher in another case study. The case study should be described in sufficient detail so that its results can be reproduced by someone else with only the paper in hand. However, this is seldom the case.

The focus of this thesis is not primarily on machine learning/connectionist learning models but on the data used to evaluate their performance and on the way in which results are reported. The intention is not to find the best learning model for a particular database but rather to establish a set of databases to serve as a standard or a benchmark against which current and future learning models can be measured, and a methodology by which results can be reported accurately and reproducibly. Others have also sought to define a benchmark for machine learning [58, 62, 63, 66, 138] or connectionist [97] models and have remarked on the lack of scientific rigor in case studies [83, 96, 97]. Many others have conducted case studies most of which suffer the limitations noted above [14, 43, 44, 53, 68, 87, 108, 119, 121, 122, 132, 137]. One exception to this rule is the Statlog project [88] conducted by many machine learning researchers from October 1990 to June 1993 under the ESPRIT program of the European community. With this notable exception, it is the scope and rigor (reproducibility) of this case study that distinguishes it from other studies.

When the machine learning/connectionist research community refers to a database, it is not speaking of a database in the usual sense of the term. In this context database simply refers to a collection of records with non-repeating fields, each record of which contains data about a specific instance of a particular problem. For example, one such database commonly used is the Voting Records (*votes)* database. Each record in this database is composed of the voting record for each of the U.S. House of Representatives Congressmen on 16 key votes drawn from the Congressional Quarterly Almanac, 98th Congress, 2nd Session 1984 (Figure 1.1).

Figure 1.1

1. Handicapped Infants	Yes|No|Unknown
2. Water Project Cost Sharing	Yes|No|Unknown
3. Adoption of the Budget Resolution	Yes|No|Unknown
4. Physician Fee Freeze	Yes|No|Unknown
5. El Salvador Aid	Yes|No|Unknown
6. Religious Groups in Schools	Yes|No|Unknown
7. Anti-Satellite Test Ban	Yes|No|Unknown
8. Aid to Nicaraguan Contras	Yes|No|Unknown
9. MX Missile	Yes|No|Unknown
10. Immigration	Yes|No|Unknown
11. Synfuels Corporation Cutback	Yes|No|Unknown
12. Education Spending	Yes|No|Unknown
13. Superfund Right to Sue	Yes|No|Unknown
14. Crime	Yes|No|Unknown
15. Duty Free Exports	Yes|No|Unknown
16. Export Administration Act South Africa	Yes|No|Unknown
17. Party	Democrat|Republican

An inductive learner uses the attribute/value pairs in such a database to construct a predictive model for the data. In the *votes* database an inductive learner uses the values of the first sixteen attribute value pairs to predict the value of the seventeenth attribute value pair. The database is usually arbitrarily partitioned into a test set and a training set. The inductive learner then uses the training set to discover which attributes characterize each of the classes and uses the test set to determine the accuracy of its characterization.

These databases are chosen because, using some learning model, we wish to discover relationships between the attributes that allow us to correctly classify any current or future unseen data from the problem domain. They come from a variety of problem domains ranging from mushroom taxonomy to thyroid disorders to soybean diseases to voting records. The data is always contained in records with non-repeating fields, that is, flat files. As mentioned above, these are not databases in the usual sense of the term since the relationships between the data they contain are unknown, and the attributes frequently do not have a well-defined

range. The databases are built from data which describe various features of the problem domain. Some of the features may be irrelevant to the classification of the data, and frequently the data is also "noisy" in that values are missing or inaccurate.

Even though focus is given principally to the data rather than the learning models, these cannot be altogether ignored. We intend to use only supervised learning models, models, which, when given examples of the form (x_i,y_i), learn a function f such that $f(x_i) = y_i$. This is called supervised learning because, during the training phase for the model, the y_i are provided for the learning model by a supervisor. There are generally only a few y_i values, and the function f must map each x_i, which is typically a composite description of some object, situation, or event (see *votes* database above), to one y_i. The function f captures patterns in the databases so that it can predict the value of y_i for previously unseen values of x_i.

In general we speak of databases D consisting of a finite number n of instances or examples. Each instance I is composed of an input vector $\boldsymbol{x}_i$ and an output class y_i. The input vector consists of m input attributes $\boldsymbol{x}_i = (x_1,...,x_m)$ and the output class, y_i, is one of c classes where $c > 1$. Thus $D = ((\boldsymbol{x}_1,y_1),...,(\boldsymbol{x}_n,y_n))$. It is the job of a learning algorithm to discover a function f that maps an input vector $\boldsymbol{x}_i$ onto the output class y_i, that is, $f(\boldsymbol{x}_i) = y_i$.

For the purposes of this thesis, learning models which are widely used by the machine learning/connectionist research community are chosen (see section 4). With the exception of backpropagation and the perceptron, all learning models are used in a generic configuration, that is, no effort is made to tune learning model parameters for optimal performance on any given database.

Section 2 describes the databases used in this study. In section 3, the learning models used are listed and briefly characterized and in section 4 the experimental method is explained. Section 5 presents several views of learning model performance as well as a brief discussion of the results. Section 6 characterizes the study's limitations and section 7 offers conclusions. In addition, the appendices describe a standard database format, software tools used in the study, and learning model parameters used in the study.

2. Databases

As explained above, databases for supervised learning consist of instances of attribute/value pairs. Attributes are either input attributes or output attributes. While databases may have more than one output attribute in general, we shall limit ourselves to the case of a single output attribute with 2 or more values (classes). We denote the i^{th} input attribute of a database with n input attributes as x_i and the output attribute or class as y.

Databases readily available to the machine learning/connectionist community were chosen for this study. The databases can be found at a variety of ftp sites, but most frequently the databases located in the machine learning database repository at ics.uci.edu [92] or the connectionist database repository at ftp.cs.cmu.edu [38] are used in this study and by other researchers in similar studies. These databases are in various formats, formats which generally reflect the particular syntax chosen for the implementation of a learning model by the researcher who donated the database.

To simplify using the databases with the different software packages used in this study, each of which requires a different syntax for data input, all databases were translated into a common standard data format (SDF). (See Appendix A.) Except for the ordering of the attribute values and instances, no information was added to or removed from the database in the translation. However, if a particular database consisted of separate training and test sets, the training and test sets were combined into a single database. The translated SDF databases are available at [ftp://synapse.cs.byu.edu/pub/db].

2.1 Database Names

Database names are sometimes a source of confusion in case studies. For example, when a case study such as [121] refers to the soybean database, which of the four soybean databases located in ftp://ics.uci.edu/pub/machine-learning-databases/soybean is meant? Or when [37] refers to the thyroid database, which of the eleven thyroid disease databases in ftp://ics.uci.edu/pub/machine-learning-databases/soybean/thyroid-disease is meant? These two case studies provide just two examples of imprecise or incomplete specification of data -- a cursory search of the machine learning/connectionist literature will yield many other examples. Prechelt [97] has also remarked on the frequent confusion of database names.

Table 2.1a lists the 102 databases used in this study and gives a short, distinct name for each. (The table of names is also deposited in the aforementioned database repositories as well as at [ftp://synapse.cs.byu.edu:/pub/db]). The names are distinct so that no confusion results when, for example, reference is made to a soybean database. (*Soyl*, *soyf*, *soys*, and *soyso* all name a particular soybean database in ftp://ics.uci.edu/pub/machine-learning-database/soybean). The names are short so that formatting tables of results is convenient. Thus, for example, instead of labeling the row or column of a table with *shuttle-landing-control*, it is labeled *slc* with no loss of precision. The names of the SDF files are also the names of the translated files themselves, and they are 8 characters or fewer in length for the convenience of those who work in the 8.3 world of file names.

Database Names
Table 2.1a

Name	Location of Original Database	Location of SDF Database
adult!	[UCI]/balloons/adult+stretch.data	[BYU]/balloons
adult-	[UCI]/balloons/adult-stretch.data	[BYU]/balloons
agaric	[UCI]/mushroom/agaricus-lepiota.data	[BYU]/mushroom
ahyper	[UCI]/thyroid-disease/allhyper.{data,test}	[BYU]/thyroid-disease
ahypo	[UCI]/thyroid-disease/allhypo.{data,test}	[BYU]/thyroid-disease
allbp	[UCI]/thyroid-disease/allbp.{data,test}	[BYU]/thyroid-disease
allrep	[UCI]/thyroid-disease/allrep.{data,test}	[BYU]/thyroid-disease
ann	[UCI]/thyroid-disease/ann-{train,test}.data!	[BYU]/thyroid-disease
anneal	[UCI]/annealing/anneal.{data,test}	[BYU]/annealing
audio	[UCI]/audiology/audiology.standardized.{data,test}	[BYU]/audiology
auto	[UCI]/auto-mpg/auto-mpg.data	[BYU]/auto-mpg
balanc	[UCI]/balance-scale/balance.data	[BYU]/balance-scale
breast	[UCI]/breast-cancer/breast-cancer.data	[BYU]/breast-cancer
breastw	[UCI]/breast-cancer-wisconsin/breast-cancer-wisconsin.data	[BYU]/breast-cancer-wisconsin
bridg1	[UCI]/bridges/bridges.data.version1	[BYU]/bridges
bridg2	[UCI]/bridges/bridges.data.version2	[BYU]/bridges
bupa	[UCI]/liver-disorders/bupa.data	[BYU]/liver-disorders
cmupa7	[BYU]/cmu/cmupa7	[BYU]/cmu
cmupa8	[BYU]/cmu/cmupa8	[BYU]/cmu
cmupro	[CMU]/protein.data	[BYU]/cmu
cmuson	[CMU]/sonar.data	[BYU]/cmu
cmusp	[BYU]/cmu/cmusp	[BYU]/cmu
cmusp2	[BYU]/cmu/cmusp2	[BYU]/cmu
cmuvow	[CMU]/vowel.{tst,trn}	[BYU]/cmu
crx	[UCI]/credit-screening/crx.data	[BYU]/credit-screening
dis	[UCI]/thyroid-disease/dis.{data,test}	[BYU]/thyroid-disease
echoc	[UCI]/echocardiogram/echocardiogram.data	[BYU]/echocardiogram
flag	[UCI]/flags/flag.data	[BYU]/flags
glass	[UCI]/glass/glass.data	[BYU]/glass
hayes	[UCI]/hayes-roth/hayes-roth.{data,test}	[BYU]/hayes-roth
heart	[UCI]/heart-disease/processed.{cleveland,hungarian,switzerland,va},data	[BYU]/heart-disease
heartc	[UCI]/heart-disease/processed.cleveland.data	[BYU]/heart-disease
hearth	[UCI]/heart-disease/processed.hungarian.data	[BYU]/heart-disease
hearts	[UCI]/heart-disease/processed.switzerland.data	[BYU]/heart-disease
heartv	[UCI]/heart-disease/processed.va.data	[BYU]/heart-disease
hepat	[UCI]/hepatitis/hepatitis.data	[BYU]/hepatitis
horse	[UCI]/horse-colic/horse-colic.{data,test}	[BYU]/horse-colic
house	[UCI]/housing/housing.data	[BYU]/housing
hypoth	[UCI]/thyroid-disease/hypothyroid.data	[BYU]/thyroid-disease
import	[UCI]/autos/imports-85.data	[BYU]/autos
iono	[UCI]/ionosphere/ionosphere.data	[BYU]/ionosphere
iris	[UCI]/iris/iris.data	[BYU]/iris
isol5	[UCI]/isolet/isolet5.data	[BYU]/isolet
kinsh	[UCI]/kinship/kinship.data	[BYU]/kinship
krvskp	[UCI]/chess/king-rook-vs-king-pawn/kr-vs-kp.data	[BYU]/chess
laborn	[UCI]/labor-negotiations/labor-negotiations.{data,test}	[BYU]/labor-negotiations
led17	[BYU]/led-display-creator/led17	[BYU]/led-display-creator
led7	[BYU]/led-display-creator/led7	[BYU]/led-display-creator
lense	[UCI]/lenses/lenses.data	[BYU]/lenses
letter	[UCI]/letter-recognition/letter-recognition.data	[BYU]/letter-recognition
lrs	[UCI]/spectrometer/lrs.data	[BYU]/spectrometer

[BYU] = ftp://synapse.cs.byu.edu/pub/db
[CMU] = ftp://ftp.cs.cmu.edu/afs/cs/project/connect/bench [UCI] = ftp://ics.uci.edu/pub/machine-learning-databases

Database Names
Table 2.1a

Name	Location of Original Database	• Location of SDF Database
lungc	[UCI]/lung-cancer/lung-cancer.data	[BYU]/lung-cancer
lymph	[UCI]/lymphography/lymphography.data	[BYU]/lymphography
mache	[UCI]/cpu-performance/machine.data	[BYU]/cpu-performance
machp	[UCI]/cpu-performance/machine.data	[BYU]/cpu-performance
monk1	[UCI]/monks-problems/monks-1.{train,test}	[BYU]/monks-problems
monk2	[UCI]/monks-problems/monks-2.{train,test}	[BYU]/monks-problems
monk3	[UCI]/monks-problems/monks-3.{train,test}	[BYU]/monks-problems
musk1	[UCI]/musk/clean1.data	[BYU]/musk
musk2	[UCI]/musk/clean2.data	[BYU]/musk
netpho	[UCI]/undocumented/connectionist-bench/nettalk/nettalk.data	[BYU]/nettalk
netstr	[UCI]/undocumented/connectionist-bench/nettalk/nettalk.data	[BYU]/nettalk
nettyp	[UCI]/undocumented/connectionist-bench/nettalk/nettalk.data	[BYU]/nettalk
newthy	[UCI]/thyroid-disease/new-thyroid.data	[BYU]/thyroid-disease
pima	[UCI]/pima-indians-diabetes/pima-indians-diabetes.data	[BYU]/pima-indians-diabetes
postop	[UCI]/postoperative-patient-data/post-operative.data	[BYU]/postoperative-patient-data
promot	[UCI]/molecular-biology/promoter-gene-sequences/promoters.data	[BYU]/promoter-gene-sequence
protei	[UCI]/molecular-biology/protein-secondary-structure/protein-secondary-structure.{train,test}	[BYU]/protein-secondary-structures
ptumor	[UCI]/primary-tumor/primary-tumor.data	[BYU]/primary-tumor
segm	[UCI]/image/segmentation.{data,test}	[BYU]/image
servo	[UCI]/servo/servo.data	[BYU]/servo
sick	[UCI]/thyroid-disease/sick.{data,test}	[BYU]/thyroid-disease
sickeu	[UCI]/thyroid-disease/sick-euthyroid.data	[BYU]/thyroid-disease
slc	[UCI]/shuttle-landing-control/shuttle-landing-control.data	[BYU]/shuttle-landing-
sonar	[UCI]/undocumented/connectionist-bench/sonar/sonar.all-data	[BYU]/sonar
soyf	[UCI]/soybean/fisher-order	[BYU]/soybean
soyl	[UCI]/soybean/soybean-large.{data,test}	[BYU]/soybean
soys	[UCI]/soybean/soybean-small.data	[BYU]/soybean
soyso	[UCI]/soybean/stepp-order	[BYU]/soybean
splice	[UCI]/molecular-biology/splice-junction-gene-sequences/splice.data	[BYU]/splice-junction-gene-
ssblow	[UCI]/space-shuttle/o-ring-erosion-or-blowby.data	[BYU]/space-shuttle
ssonly	[UCI]/space-shuttle/o-ring-erosion-only.data	[BYU]/space-shuttle
staust	[UCI]/statlog/australian/australian.dat	[BYU]/australian
stgern	[UCI]/statlog/german/german.data-numeric	[BYU]/german
stgers	[UCI]/statlog/german/german.data	[BYU]/german
sthear	[UCI]/statlog/heart/heart.dat	[BYU]/heart
stsati	[UCI]/statlog/satimage/sat.{trn,tst}	[BYU]/satimage
stsegm	[UCI]/statlog/segment/segment.dat	[BYU]/segment
stshut	[UCI]/statlog/shuttle/shuttle.{trn,tst}	[BYU]/shuttle
stvehi	[UCI]/statlog/vehicle/{xaa,xab,xac,xad,xae,xaf,xag,xah,xai}.dat	[BYU]/vehicle
thy387	[UCI]/thyroid-disease/thyroid0387.data	[BYU]/thyroid-disease
tictac	[UCI]/tic-tac-toe/tic-tac-toe.data	[BYU]/tic-tac-toe
trains	[UCI]/trains/trains-transformed.data	[BYU]/trains
votes	[UCI]/voting-records/house-votes-84.data	[BYU]/voting-records
vowel	[UCI]/undocumented/connectionist-bench/vowel/vowel.tr-orig-order	[BYU]/vowel
water	[UCI]/water-treatment/water-treatment.data	[BYU]/water-treatment
wave21	[BYU]/waveform/wave21	[BYU]/waveform
wave40	[BYU]/waveform/wave40	[BYU]/waveform
wine	[UCI]/wine/wine.data	[BYU]/wine
yellow	[UCI]/balloons/yellow-small.data	[BYU]/balloons
ys!as	[UCI]/balloons/yellow-small+adult-stretch.data	[BYU]/balloons
zoo	[UCI]/zoo/zoo.data	[BYU]/zoo

[BYU] = ftp://synapse.cs.byu.edu/pub/db
[CMU] = ftp://ftp.cs.cmu.edu/afs/cs/project/connect/bench [UCI] = ftp://ics.uci.edu/pub/machine-learning-databases

Database Characteristics
Table 2.1b

Name	C	D	N	U	50	Name	C	D	N	U	50
adult!		x			x	lymph		x			
adult-		x			x	mache	x	x	x		
agaric		x		x		machp	x	x	x		
ahyper	x	x		x		monk1		x			
ahypo	x	x		x		monk2		x			
allbp	x	x		x		monk3		x	x		
allrep	x	x		x		musk1	x				
ann	x	x				musk2	x				
anneal	x	x		x		netpho		x	x		
audio		x		x		netstr		x	x		
auto	x	x		x		nettyp		x	x		
balanc		x				newthy	x				
breast		x	x	x		pima	x				
breastw		x		x		postop	x	x	x	x	
bridg1	x	x		x		promot		x			
bridg2	x	x		x		protei		x	x		
bupa	x					ptumor	x	x	x	x	
cmupa7		x				segm	x				
cmupa8		x				servo		x			
cmupro		x	x			sick	x x		x		
cmuson	x					sickeu	x	x		x	
cmusp	x					slc		x		x	x
cmusp2	x					sonar	x				
cmuvow x						soyf		x			x
crx	x	x		x		soyl	x	x	x		
dis	x	x		x		soys		x			x
echoc	x	x		x		soyso		x			x
flag	x	x				splice		x	x		
glass	x					ssblow	x				x
hayes		x	x			ssonly	x				x
heart	x	x		x		staust	x	x			
heartc	x	x		x		stgern	x				
hearth	x	x		x		stgers	x	x			
hearts	x	x		x		sthear	x	x			
heartv	x	x		x		stsati	x				
hepat	x	x		x		stsegm	x				
horse	x	x		x		stshut	x				
house	x	x				stvehi	x				
hypoth	x	x	x	x		thy387	x	x		x	
import	x	x		x		tictac		x			
iono	x					trains		x			x
iris	x					votes		x		x	
isol5	x					vowel	x				
kinsh		x				water	x				
krvskp		x				wave21	x				
laborn	x	x		x	x	wave40	x				
led17		x				wine	x				
led7		x	x			yellow		x			x
lense		x			x	ys!as		x			x
letter		x				zoo		x			
lrs	x										
lungc		x		x	x	Totals	61	74	16	35	14

C = Continuous D = Discrete N = Noise U = Unknowns 50 = Fewer than 50 Instances

Table 2.1b categorizes the databases into those with at least one discrete attribute, those with at least one continuous attribute, those with noise (only instances whose attribute values are identical but class values are different are classified as noisy in this table), those with at least one unknown attribute value, and those with 50 or fewer instances. For a brief description of the databases and their owners or creators, see Appendix E.

2.2 Database Statistics

Table 2.2 presents statistics about the databases used in this study. They have also been presented in part elsewhere [92, 138]. New to this table of statistics is volume, density, and noise.

The volume V of a database is defined as

$$V = \prod_i k_i,$$

where

k_i is the number of different values for each attribute a, and

$i = 1, ..., m$ is the number of input attributes in the database.

For discrete attributes, the value of k_i is obvious. For continuous attributes, k_i is

$$k_i = (\max(x_i) - \min(x_i)) \cdot 10^p,$$

where $\max(x_i)$ is the maximum value of the i^{th} attribute and $\min(x_i)$ is the minimum value of the i^{th} attribute. The precision of the i^{th} attribute, p, is the maximum number of significant digits to the right of the decimal point for that attribute. Given a value for the precision of a continuous attribute, k_i is simply the number of different values that the attribute could possibly have over its range. For example, if $\max(x_i)$ is 10.11, $\min(x_i)$ is 1.11, and the number

of significant digits to the right of the decimal is 2 ($p = 2$), then $k_i = (10.11 - 1.11) \cdot 10^2 = 900$. This calculation assumes that the number of significant digits for a continuous attribute is statistically meaningful, that is, that a continuous attribute was actually measured with the indicated precision and that the precision was not arbitrarily extended or truncated. It also assumes that, for a given domain, the attribute can actually assume any value in the range $\max(x_i)$ to $\min(x_i)$, that is, the probability that the i^{th} attribute, a_i, has value x_i is not zero ($P(a_i=v_i) \neq 0$) regardless of the value of x_i.

Zheng [138] computes database volume in the same way, however, for continuous attributes, k_i is simply the number of different values that actually occur in the database for the i^{th} attribute.

The density D of a database is

$$D = V/n, \tag{1}$$

where n is the number of distinct instances in the database.

The values for noise are integers the first of which is simply the number of pairs of instances in a database for which all attribute values are the same but the class value is different. This represents a minimum value, at least for databases with 1 or more continuous-valued attributes, since values of continuous attributes are only compared for equality and no allowance is made for overlapping distributions. Using the *ptumor* database, the two records

```
class input vector
14    2,1,2,1,1,2,2,2,2,2,2,2,2,2,2,2,2
4     2,1,2,1,1,2,2,2,2,2,2,2,2,2,2,2,2
```

provide an example of this type of noise.

The second integer indicates the number of pairs of instances with different class attributes, but which have at least one or more unknown values and for which all attribute

values would be the same if the unknown attribute values in one of the instances were replaced by the values of the corresponding attribute in the second instance. Using the *votes* database, the two records

class	input vector
democrat	?,y,y,?,y,y,n,n,n,n,y,n,y,y,n,n
republican	n,?,?,?,?,?,?,?,?,?,?,?,?,y,?,?

provide an example of this type of noise.

The third integer is the number of instances in the database for which the class value is unknown. These instances are removed from the database before the cross-validation trials.

Default accuracy (DA) is the accuracy achieved by an algorithm which simply classifies every instance in database *D* as belonging to the class with the highest probability.

The entropy is a measure of the randomness or uncertainty about the class of an instance in database *D* and is computed as

$$Entropy = -\sum_i p_i \log_2(p_i)$$

where

$i = 1, ...,$ number of classes in the database

p_i = probability of the i^{th} class.

Database Statistics
Table 2.2

Name	NI/NUI/NUV	B/C/D/I	NA	NC	Volume	Density	Noise	DA	Entropy
adult!	20 0 0	5/0/0/0	5	2	1.6000e+01 1.6000e+01	1.2500e+00 1.2500e+00	0 0 0	60.0	0.9710
adult-	20 0 0	5/0/0/0	5	2	1.6000e+01 1.6000e+01	1.2500e+00 1.2500e+00	0 0 0	60.0	0.9710
agaric	8124 2480 2480	5/0/18/0	23	2	1.4043e+15 1.4043e+15	5.7851e-12 5.7851e-12	0 0 0	51.8	0.9991
ahyper	3772 3772 6064	21/7/2/0	30	5	6.0000e+00 6.0000e+00	6.2867e+02 6.2867e+02	0 0 0	97.3	0.2075
ahypo	3772 3772 6064	21/7/2/0	30	5	6.0000e+00 6.0000e+00	6.2867e+02 6.2867e+02	0 0 0	92.3	0.4666
allbp	3772 3772 6064	21/7/2/0	30	3	6.0000e+00 6.0000e+00	6.2867e+02 6.2867e+02	0 0 0	95.7	0.2751
allrep	3772 3772 6064	21/7/2/0	30	4	6.0000e+00 6.0000e+00	6.2867e+02 6.2867e+02	0 0 0	96.7	0.2599
ann	7200 0 0	15/6/1/0	22	3	2.4740e+28 4.5650e+27	2.9103e-25 1.5772e-24	0 0 0	92.6	0.4476
anneal	898 898 22175	22/6/11/0	39	6	1.5204e+35 6.8412e+30	5.9062e-33 1.3126e-28	0 7 0	76.2	1.1898
audio	226 222 317	61/0/9/1	71	24	1.1206e+23 1.1206e+23	2.0167e-21 2.0167e-21	0 0 0	25.2	3.4219
auto	398 6 6	0/5/3/0	8		8.2277e+18 4.8191e+12	4.8373e-17 8.2587e-11	0 0 0		
balanc	625 0 0	0/0/5/0	5	3	6.2500e+02 6.2500e+02	1.0000e+00 1.0000e+00	0 0 0	46.1	1.3181
breast	286 9 9	4/0/6/0	10	2	5.0544e+05 5.0544e+05	5.6584e-04 5.6584e-04	7 0 0	70.3	0.8778
breastw	699 16 16	1/0/9/1	11	2	1.0000e+09 1.0000e+09	6.9900e-07 6.9900e-07	0 0 0	65.5	0.9293
bridg1	108 38 75	2/3/7/1	13	7	2.2232e+14 6.4699e+09	4.8579e-13 1.6693e-08	0 0 2	40.7	2.3553
bridg2	108 38 75	2/1/9/1	13	7	4.2301e+07 8.8750e+06	2.5531e-06 1.2169e-05	0 0 2	40.7	2.3553
bupa	345 0 0	1/6/0/0	7	2	2.9673e+17 1.6862e+15	1.1627e-15 2.0460e-13	0 0 0	58.0	0.9816
cmupa7	128 0 0	8/0/0/0	8	2	1.2800e+02 1.2800e+02	1.0000e+00 1.0000e+00	0 0 0	50.0	1.0000

NI=Number of Instances NUI=Number of Instances with Unknown Values NUV=Number of Unknown Values
B=Number of Boolean Attributes C=Number of Continuous Attributes D=Number of Discrete Attributes
I=Number of Ignored Attributes NA = Number Of Attributes (B+C+D+I) NC=Number of Classes
Volume=Database Volume/Zheng Database Volume Density = Database Density/Zheng Database Density
Noise=Class Noise/Unknowns Noise/Unknown Classes DA=Default Accuracy Entropy = Entropy of Class Attribute

Database Statistics
Table 2.2

Name	NI/NUI/NUV	B/C/D/I	NA	NC	Volume	Density	Noise	DA	Entropy
cmupa8	256 0 0	9/0/0/0	9	2	2.5600e+02 2.5600e+02	1.0000e+00 1.0000e+00	0 0 0	50.0	1.0000
cmupro	21627 0 0	0/0/14/0	14	3	1.5447e+17 1.5447e+17	1.4001e-13 1.4001e-13	61 0 0	54.5	1.4449
cmuson	208 0 0	1/60/0/0	61	2	1.1012e+218 1.2128e+139	1.8889e-216 1.7150e-137	0 0 0	53.4	0.9967
cmusp	194 0 0	1/2/0/0	3	2	1.5600e+12 3.7636e+04	1.2436e-10 5.1546e-03	0 0 0	50.0	1.0000
cmusp2	386 0 0	1/2/0/0	3	2	1.5606e+12 1.4900e+05	2.4734e-10 2.5907e-03	0 0 0	50.0	1.0000
cmuvow	990 0 0	0/10/1/0	11	11	4.4320e+34 9.0438e+29	2.2337e-32 1.0947e-27	0 0 0	9.1	3.4594
crx	690 37 67	5/6/5/0	16	2	5.1609e+30 7.5511e+21	1.3370e-28 9.1378e-20	0 0 0	55.5	0.9912
dis	3772 3772 6064	22/7/1/0	30	2	6.0000e+00 6.0000e+00	6.2867e+02 6.2867e+02	0 0 0	98.5	0.1146
echoc	131 70 98	3/7/0/0	10	2	4.6277e+26 1.9139e+15	2.8308e-25 6.8446e-14	0 0 0	67.2	0.9131
flag	194 0 0	13/3/13/1	30	8	2.0049e+25 1.2965e+20	9.6764e-24 1.4963e-18	0 0 0	35.6	2.3217
glass	214 0 0	0/9/1/1	11	7	1.3125e+24 9.4129e+20	1.6305e-22 2.2735e-19	0 0 0	35.5	2.1765
hayes	93 0 0	0/0/5/0	5	3	1.9200e+02 1.9200e+02	4.8438e-01 4.8438e-01	9 0 0	33.3	1.5850
heart	929 334 983	3/5/6/0	14	5	7.2746e+18 6.0853e+18	1.2770e-16 1.5266e-16	0 5 0	41.7	2.0453
heartc	303 6 6	3/5/6/0	14	5	1.9461e+18 2.7460e+16	1.5569e-16 1.1034e-14	0 0 0	54.1	1.8459
hearth	294 293 782	3/5/6/0	14	5	1.2343e+18 2.1622e+16	2.3819e-16 1.3597e-14	0 0 0	63.9	0.9431
hearts	123 123 273	3/5/6/0	14	5	1.0330e+08 1.9184e+07	1.1907e-06 6.4115e-06	0 0 0	39.0	1.9759
heartv	200 199 698	3/5/6/0	14	5	1.9629e+18 1.2651e+15	1.0189e-16 1.5809e-13	0 3 0	28.0	2.1745
hepat	155 75 167	14/6/0/0	20	2	3.2843e+21 4.4030e+16	4.7194e-20 3.5203e-15	0 0 0	54.8	0.9932

NI=Number of Instances NUI=Number of Instances with Unknown Values NUV=Number of Unknown Values
B=Number of Boolean Attributes C=Number of Continuous Attributes D=Number of Discrete Attributes
I=Number of Ignored Attributes NA = Number Of Attributes (B+C+D+I) NC=Number of Classes
Volume=Database Volume/Zheng Database Volume Density = Database Density/Zheng Database Density
Noise=Class Noise/Unknowns Noise/Unknown Classes DA=Default Accuracy Entropy = Entropy of Class Attribute

Database Statistics
Table 2.2

Name	NI/NUI/NUV	B/C/D/I	NA	NC	Volume	Density	Noise	DA	Entropy
horse	368 361 1925	3/12/12/1	28	3	1.3738e+48 3.3385e+36	2.6787e-46 1.1023e-34	0 0 2	61.1	1.3282
house	506 0 0	1/13/0/0	14		2.1406e+50 5.6343e+32	2.3638e-48 8.9808e-31	0 0 0		
hypoth	3163 3161 5329	19/7/0/0	26	2	1.4628e+32 3.0509e+28	2.1622e-29 1.0368e-25	1 0 0	95.2	0.2767
import	201 42 51	4/15/7/0	26		2.0580e+52 1.9555e+39	9.7667e-51 1.0279e-37	0 0 0		
iono	351 0 0	1/34/0/0	35	2	4.2950e+169 2.8173e+81	8.1724e-168 1.2459e-79	0 0 0	64.1	0.9418
iris	150 0 0	0/4/1/0	5	3	1.2234e+06 5.0625e+08	1.2261e-04 2.9630e-07	0 0 0	33.3	1.5850
isol5	1559 0 0	0/617/1/0	618	26	1.0004e+2631 9.6840e+1969	1.5584e-2628 1.6099e-1967	0 0 0	3.8	4.7004
kinsh	112 0 0	0/0/3/0	3	12	5.7600e+02 5.7600e+02	1.9444e-01 1.9444e-01	0 0 0	10.7	3.5436
krvskp	3196 0 0	36/0/1/0	37	2	1.0308e+11 1.0308e+11	3.1005e-08 3.1005e-08	0 0 0	52.2	0.9986
laborn	27 27 176	4/7/6/0	17	2	6.9717e+16 1.6005e+12	3.8728e-16 1.6870e-11	0 0 0	66.7	0.9183
led17	200 0 0	24/0/1/0	25	10	1.6777e+07 1.6777e+07	1.1921e-05 1.1921e-05	0 0 0	12.0	3.3080
led7	200 0 0	7/0/1/0	8	10	1.2800e+02 1.2800e+02	1.5625e+00 1.5625e+00	204 0 0	14.5	3.2921
lense	24 0 0	3/0/2/1	6	3	2.4000e+01 2.4000e+01	1.0000e+00 1.0000e+00	0 0 0	62.5	1.3261
letter	20000 0 0	0/0/17/0	17	26	1.8447e+19 1.8447e+19	1.0842e-15 1.0842e-15	0 0 0	4.1	4.6998
lrs	531 0 0	0/102/0/0	102		4.6265e+806 2.1586e+264	1.1477e-804 2.4599e-262	0 0 0		
lungc	32 5 5	0/0/57/0	57	3	5.1923e+33 5.1923e+33	6.1630e-33 6.1630e-33	0 0 0	40.6	1.5671
lymph	148 0 0	9/0/10/0	19	4	3.0199e+08 3.0199e+08	4.9008e-07 4.9008e-07	0 0 0	54.7	1.2277
mache	209 0 0	0/8/1/0	9		2.6030e+30 5.2257e+17	8.0293e-29 3.9995e-16	4 0 0		

NI=Number of Instances NUI=Number of Instances with Unknown Values NUV=Number of Unknown Values
B=Number of Boolean Attributes C=Number of Continuous Attributes D=Number of Discrete Attributes
I=Number of Ignored Attributes NA = Number Of Attributes (B+C+D+I) NC=Number of Classes
Volume=Database Volume/Zheng Database Volume Density = Database Density/Zheng Database Density
Noise=Class Noise/Unknowns Noise/Unknown Classes DA=Default Accuracy Entropy = Entropy of Class Attribute

Database Statistics
Table 2.2

Name	NI/NUI/NUV	B/C/D/I	NA	NC	Volume	Density	Noise	DA	Entropy
machp	209 0 0	0/8/1/0	9		2.6030e+30 5.2257e+17	8.0293e-29 3.9995e-16	4 0 0		
monk1	556 0 0	3/0/4/1	8	2	4.3200e+02 4.3200e+02	1.2870e+00 1.2870e+00	0 0 0	50.0	1.0000
monk2	601 0 0	3/0/4/1	8	2	4.3200e+02 4.3200e+02	1.3912e+00 1.3912e+00	0 0 0	65.7	0.9274
monk3	554 0 0	3/0/4/1	8	2	4.3200e+02 4.3200e+02	1.2824e+00 1.2824e+00	6 0 0	52.0	0.9989
musk1	476 0 0	1/166/0/1	168	2	4.7893e+589 3.0392e+444	9.9387e-588 1.5662e-442	0 0 0	56.5	0.9877
musk2	6598 0 0	1/166/0/1	168	2	2.2630e+598 1.0530e+634	2.9157e-595 6.2657e-631	0 0 0	84.6	0.6201
netpho	20005 0 0	0/0/8/0	8	52	1.0460e+10 1.0460e+10	1.9125e-06 1.9125e-06	169 0 0	14.8	4.7461
netstr	20005 0 0	0/0/8/0	8	6	1.0460e+10 1.0460e+10	1.9125e-06 1.9125e-06	499 0 0	35.3	2.0692
nettyp	20005 0 0	0/0/8/0	8	3	1.0460e+10 1.0460e+10	1.9125e-06 1.9125e-06	23 0 0	99.3	0.0695
newthy	215 0 0	0/5/1/0	6	3	6.1615e+12 4.5940e+11	3.4894e-11 4.6800e-10	0 0 0	69.8	1.1851
pima	768 0 0	1/8/0/0	9	2	3.2593e+24 1.2103e+23	2.3563e-22 6.3456e-21	0 0 0	65.1	0.9331
postop	90 3 3	0/1/8/0	9	3	2.9160e+05 2.5369e+05	3.0864e-04 3.5476e-04	8 0 0	71.1	0.9803
promot	106 0 0	1/0/57/1	59	2	2.0769e+34 2.0769e+34	5.1037e-33 5.1037e-33	0 0 0	50.0	1.0000
protei	21625 0 0	0/0/14/0	14	3	1.5447e+17 1.5447e+17	1.3999e-13 1.3999e-13	61 0 0	54.5	1.4450
ptumor	339 207 225	14/0/4/0	18	22	4.4237e+05 4.4237e+05	7.6633e-04 7.6633e-04	53 121 0	24.8	3.6437
segm	2310 0 0	0/19/1/0	20	7	1.6668e+160 6.5734e+53	1.3859e-157 3.5142e-51	0 0 0	14.3	2.8074
servo	167 0 0	0/0/5/0	5	20	5.0000e+02 5.0000e+02	3.3400e-01 3.3400e-01	0 0 0	31.7	2.8142
sick	3772 3772 6064	22/7/1/0	30	2	6.0000e+00 6.0000e+00	6.2867e+02 6.2867e+02	0 0 0	93.9	0.3324

NI=Number of Instances NUI=Number of Instances with Unknown Values NUV=Number of Unknown Values
B=Number of Boolean Attributes C=Number of Continuous Attributes D=Number of Discrete Attributes
I=Number of Ignored Attributes NA = Number Of Attributes (B+C+D+I) NC=Number of Classes
Volume=Database Volume/Zheng Database Volume Density = Database Density/Zheng Database Density
Noise=Class Noise/Unknowns Noise/Unknown Classes DA=Default Accuracy Entropy = Entropy of Class Attribute

Database Statistics
Table 2.2

Name	NI/NUI/NUV	B/C/D/I	NA	NC	Volume	Density	Noise	DA	Entropy
sickeu	3163 3161 5329	19/7/0/0	26	2	1.4628e+32 3.0509e+28	2.1622e-29 1.0368e-25	0 0 0	90.7	0.4452
slc	15 9 26	5/0/2/0	7	2	2.5600e+02 2.5600e+02	5.8594e-02 5.8594e-02	0 0 0	60.0	0.9710
sonar	208 0 0	1/60/0/0	61	2	1.1012e+218 1.2128e+139	1.8889e-216 1.7150e-137	0 0 0	53.4	0.9967
soyf	47 0 0	16/0/20/0	36	4	1.2483e+15 1.2483e+15	3.7653e-14 3.7653e-14	0 0 0	36.2	1.9558
soyl	683 121 2337	16/0/20/0	36	19	1.2483e+15 1.2483e+15	5.4716e-13 5.4716e-13	1 112 0	13.5	3.8355
soys	47 0 0	16/0/20/0	36	4	1.2483e+15 1.2483e+15	3.7653e-14 3.7653e-14	0 0 0	36.2	1.9558
soyso	47 0 0	16/0/20/0	36	4	1.2483e+15 1.2483e+15	3.7653e-14 3.7653e-14	0 0 0	36.2	1.9558
splice	3190 0 0	0/0/61/1	62	3	1.5325e+54 1.5325e+54	2.0816e-51 2.0816e-51	1 0 0	51.9	1.4802
ssblow	23 0 0	0/4/1/0	5	3	9.2400e+07 1.2167e+04	2.4892e-07 1.8904e-03	0 0 0	69.6	1.1492
ssonly	23 0 0	0/4/1/0	5	3	9.2400e+07 1.2167e+04	2.4892e-07 1.8904e-03	0 0 0	73.9	0.9976
staust	690 0 0	5/6/4/0	15	2	1.2902e+34 1.9581e+21	5.3479e-32 3.5239e-19	0 0 0	55.5	0.9912
stgern	1000 0 0	1/24/0/0	25	2	2.1557e+34 1.0000e+72	4.6389e-32 1.0000e-69	0 0 0	70.0	0.8813
stgers	1000 0 0	3/7/11/0	21	2	8.8795e+23 4.7520e+28	1.1262e-21 2.1044e-26	0 0 0	70.0	0.8813
sthear	270 0 0	4/5/5/0	14	2	6.2554e+17 4.9590e+15	4.3162e-16 5.4447e-14	0 0 0	55.6	0.9911
stsati	6435 0 0	0/36/1/0	37	6	1.4650e+107 1.2816e+137	4.3924e-104 5.0211e-134	0 0 0	23.8	2.4833
stsegm	2310 0 0	0/19/1/0	20	7	1.6668e+160 6.5734e+53	1.3859e-157 3.5142e-51	0 0 0	14.3	2.8074
stshut	58000 0 0	0/9/1/0	10	7	1.5103e+36 7.4277e+42	3.8402e-32 7.8087e-39	0 0 0	78.6	0.9603
stvehi	846 0 0	0/18/1/0	19	4	6.6502e+50 4.9280e+52	1.2721e-48 1.7167e-50	0 0 0	25.8	1.9991

NI=Number of Instances NUI=Number of Instances with Unknown Values NUV=Number of Unknown Values
B=Number of Boolean Attributes C=Number of Continuous Attributes D=Number of Discrete Attributes
I=Number of Ignored Attributes NA = Number Of Attributes (B+C+D+I) NC=Number of Classes
Volume=Database Volume/Zheng Database Volume Density = Database Density/Zheng Database Density
Noise=Class Noise/Unknowns Noise/Unknown Classes DA=Default Accuracy Entropy = Entropy of Class Attribute

Database Statistics
Table 2.2

Name	NI/NUI/NUV	B/C/D/I	NA	NC	Volume	Density	Noise	DA	Entropy
thy387	9172 9152 14629	21/7/2/0	30	34	1.7815e+54 1.3466e+33	5.1486e-51 6.8110e-30	0 0 0	73.8	1.7901
tictac	958 0 0	1/0/9/0	10	2	1.9683e+04 1.9683e+04	4.8671e-02 4.8671e-02	0 0 0	65.3	0.9310
trains	10 0 0	15/0/18/0	33	2	3.0831e+16 3.0831e+16	3.2435e-16 3.2435e-16	0 0 0	50.0	1.0000
votes	435 203 392	17/0/0/0	17	2	6.5536e+04 6.5536e+04	6.6376e-03 6.6376e-03	0 346 0	61.4	0.9623
vowel	528 0 0	0/10/1/0	11	11	6.1513e+35 1.6840e+27	8.5836e-34 3.1354e-25	0 0 0	9.1	3.4594
water	527 147 591	0/38/1/0	39	13	3.5279e+119 8.4629e+102	1.4938e-117 6.2272e-101	0 0 0	52.2	1.9036
wave21	300 0 0	0/21/1/0	22	3	2.2871e+60 1.0460e+52	1.3117e-58 2.8680e-50	0 0 0	35.3	1.5832
wave40	300 0 0	0/40/1/0	41	3	4.7508e+112 1.2158e+99	6.3147e-111 2.4676e-97	0 0 0	35.3	1.5832
wine	178 0 0	0/13/1/0	14	3	8.1775e+35 1.8008e+29	2.1767e-34 9.8846e-28	0 0 0	39.9	1.5668
yellow	20 0 0	5/0/0/0	5	2	1.6000e+01 1.6000e+01	1.2500e+00 1.2500e+00	0 0 0	60.0	0.9710
ys!as	16 0 0	5/0/0/0	5	2	1.6000e+01 1.6000e+01	1.0000e+00 1.0000e+00	0 0 0	56.2	0.9887
zoo	90 0 0	15/0/2/1	18	7	1.9661e+05 1.9661e+05	4.5776e-04 4.5776e-04	0 0 0	41.1	2.3611

NI=Number of Instances NUI=Number of Instances with Unknown Values NUV=Number of Unknown Values
B=Number of Boolean Attributes C=Number of Continuous Attributes D=Number of Discrete Attributes
I=Number of Ignored Attributes NA = Number Of Attributes (B+C+D+I) NC=Number of Classes
Volume=Database Volume/Zheng Database Volume Density = Database Density/Zheng Database Density
Noise=Class Noise/Unknowns Noise/Unknown Classes DA=Default Accuracy Entropy = Entropy of Class Attribute

3. Learning Models

The selection criteria for including a learning model in this study was the general acceptance by the machine learning/connectionist community of the model and, to a lesser degree, the availability of software for the model. Acceptance was determined by an informal literature survey: If a model or some variation on the model was studied by many researchers, was cited by many researchers, or was used in many case studies similar to this one, it met the acceptance criteria. Availability of software was important only insofar as it would not have been possible, given reasonable time constraints, to implement all of the learning models which met the acceptance criteria. Thus, for example, while AQ11 and its successors are often cited in machine learning literature, software which implemented it was not available for the computers at the author's disposal.

The learning models included in this study may be classified into 5 broad categories as follows: (1) Decision trees, (2) Nearest neighbor, (3) Statistical, (4) Rule-based, and (5) Feed-forward neural networks. Which of these models belong to which category is perhaps open to debate, however, these classifications are not intended to be anything more than indicators of the general type of model. The learning models in each of these categories and the software implementing them listed below.

3.1 Decision Trees

Decision trees are perhaps the most widely studied inductive learning models in the machine learning community. The literature abounds with papers proposing new models or variations of existing models and case studies using decision trees ([14, 21, 22, 25, 30, 34,

40, 43, 49, 50, 51, 53, 89, 93, 98, 99, 100, 101, 102, 104, 105, 106, 107, 109, 110, 111, 112, 113, 114, 118, 120, 123, 126, 129, 130, 131, 133, 134, 136]). For this case study, we use decision tree software from Quinlan and Buntine.

Quinlan introduces decision trees and illustrates the use of his C4.5 software for decision trees (c4.5tree) and production rules derived therefrom (c4.5rule) in [105].

Several decision tree algorithms (cart, id3, c4, minimum message length) are described by Buntine in [26, 27, 29, 30]. The use of the Buntine/Caruana IND v2.1 decision tree software available from NASA's Cosmic facility is explained in [31].

3.2 Nearest Neighbor

Nearest-neighbor models or instance-based learners are described in [1, 2, 3, 4, 5, 7, 9, 10, 13, 16, 18, 41, 54, 70, 71, 72, 81]. David Aha provided software for the models labeled ib1, ib2, ib3, and ib4 [6] based on the algorithms described in [7].

3.3 Statistical

Statistical classification methods may be divided into two main categories: parametric and non-parametric [88]. Parametric methods often assume that data has a Gaussian distribution while non-parametric methods make no assumptions about the distribution of data. Even though the goal of statistical classifiers is the same as the goal of machine learning and connectionist learning models, we are primarily interested in the latter. We include the naive Bayes classifier (bayes) for contrast. It is described in [33, 77, 88].

3.4 Rule Based

Rule-based algorithms are thought by some researchers [36, 88, 105] to be especially desirable since rules are easily understood by humans. The CN2 induction algorithm incorporates ideas from Quinlan's ID3 decision tree algorithm and from Michalski's AQ. CN2 generates ordered (ocn2) or unordered (ucn2) lists of classification rules and is described in [35, 36, 37]. The CN2 software used in this study was obtained from David Aha [23]. Software for CN2 is also available from the Turing Institute [24].

The c45 rule algorithm uses the decision tree constructed by c45 tree to build production rules. Quinlan describes the method used to transform a decision tree into rules in [100, 105]. The c45 rule software is available with Quinlan's book [105].

3.6 Neural Networks

The perceptron is the oldest of all inductive learners having first been described by Rosenblatt [116, 117] and Hebb [60] in the 60s and in 1949 respectively. Minsky and Papert discuss the perceptron and its limitations extensively in [90]. Other learning models employ concepts derived from the perceptron [15, 20, 59, 94, 115], and it is frequently included as a model against which other models are compared [64]. As Minsky and Papert note, the principal drawback to the perceptron is its inability to solve non-linearly separable problems.

The backpropagation algorithm is described in the frequently referenced [84]. Back-propagation, the best known and most widely used connectionist model, and many variations of it have since been described in many places, among them [11, 12, 17, 28, 39, 46, 47, 48, 52, 53, 56, 59, 61, 67, 85, 91, 125]. Two frequent difficulties with back-propagation are the

number of parameters that must be adjusted for each problem and the amount of time necessary to train the network. Learning rate, momentum, initial weights, number of hidden nodes, number of connections between nodes, and convergence or stopping criteria are all parameters that must be adjusted for each problem. No one has yet discovered an algorithm which can predict parameter values for a particular problem although rules of thumb do exist. And even though similar problems may be solved with similar values for each of these parameters, this is by no means guaranteed [59, 73].

For problems with many attributes or networks with many connections, a network may be trained for several days before the convergence or stopping criteria are reached and there is no guarantee that this network is even close to optimal. In general several different combinations of parameters must be tried before a "good" network is discovered. For this study, a back-propagation network was judged to be "good" if its average accuracy on the test instances for all cross-validations closely agreed with results from previous studies, or, when such results were not available, if its accuracy was close to the accuracies of the other learning models.

The perceptron and back-propagation software was written by the author specially for this project.

4. Method

Each of the learning models in section 3 was trained and tested using each of the databases in section 2. Except as mentioned in section 3 above, no effort was made to optimize any model's parameters for any database. Each model was cross-validated using either 10-fold or 5-fold cross-validation (depending on the size of the database as explained below) with different training/testing sets.

4.1 Cross-Validation

To estimate the accuracy of the learning models included in this study, cross-validation, which "provides a nearly unbiased estimate" [45] of the accuracy, is used. Cross-validation in its simplest form consists of the division of a data sample into two sub-samples, then training the learning model with one of the sub-samples and testing it with the other sub-sample [45, 78, 124]. As noted in [45], cross-validation estimates of accuracy can have a high variability especially with small sample sizes.

Early in the study, it was discovered that dividing small databases into 10 sub-samples produced anomalous results that were attributed to small sample sizes and to unfavorable distributions of instances. This can be illustrated using the *ssblow* database: *ssblow* contains 23 instances and 3 classes, of which one class has but two instances in the database. Dividing it into 10 sub-samples results in 7 sub-samples of 2 instances and 3 sub-samples of 3 instances. Obviously 7 of the sub-samples must be missing any instance of at least one class. If a learning model is trained using the 7 sub-samples of 2 instances and 2 sub-samples of 3 instances and tested using the remaining sub-sample of 3 instances, there is a potential

imbalance in the class distributions between the original database and the training and testing samples.

Given a database D with n instances and n_j instances of each class j, a distribution of classes in the cross-validation sub-samples D_v for v-fold cross validation different from the distribution in D cannot be avoided unless the following conditions are fulfilled: (1) $n_j \geq v$ for each class j and (2) for each class j, the number of instances in each sub-sample n_v, and n_j must have an integral least common multiple. These conditions are seldom fulfilled for any choice of v. Fortunately fulfilling these conditions becomes less important as n increases. However for small n, the anomalies introduced by not meeting these conditions may be quite pronounced.

In order to reduce the effect of these anomalies, each database is classified as small or large. A small database is defined as one for which $n \leq 50$ while a large database is any database for which $n > 50$. If the database is small, then 5-fold cross-validation was used; otherwise, 10-fold cross-validation was used. Of the 102 databases used in this case study, 14 of them are small.

Each database D was randomly divided into v sub-samples. In order to prevent wide variance in predictive accuracy of the learning models, the class distribution of each sub-sample D_v was matched as closely as possible to the class distribution of each database D. The following algorithm was used to select instances for each sub-sample (see the description of *xlate* in Appendix B):

```
for each sub-sample v
    for each class j
        Randomly select ((n_j / n) · n_v) instances from the j^th class;
for each sub-sample v
    Randomly re-arrange the previously selected instances
```

(If the aforementioned conditions for avoiding class distributions in the sub-samples D_v which are different from D are not realized, there are obviously some round-off errors that must be accounted for, but, in the interest of simplicity, these details have been omitted.)

Each learning model is trained and tested by selecting one sub-sample as the test set and using the remaining sub-samples as the training set. This is repeated v times, each time selecting a different sub-sample for the test set.

4.2 Consistency of Training/Testing Sets and Presentation Order

In order to eliminate any effects that may result from the composition of the training and testing sets, the instances selected in each of the sets for each learning model were identical. Thus, while each software package requires a different format for its training and testing files, the instances in each of the training and testing sets for each learning algorithm were identical (see Appendix B).

Likewise, in order to eliminate any effects that may result from the order in which instances are presented to the learning model for those models which are sensitive to presentation order, the training and testing sets were presented to each learning model in exactly the same order, even if the learning model is not sensitive to presentation order.

4.3 Attribute Discretization

Connectionist models are capable of learning even if the output attributes, y_i, are not discrete. However, all machine learning models in this study require that the y_i be discrete. Therefore the output classes of the 6 databases (*auto*, *house*, *import*, *lrs*, *mache*, and *machp*)

in this study with continuous output attributes are discretized using an equal width method [32, 69, 128]. The discretized output attributes are used for both the machine learning and connectionist learning models even though the connectionist models could have used continuous values for output classes. This was done to eliminate any performance differences which may have resulted from using discrete attributes for some learning models and continuous attributes for others. The statistics for the databases resulting from this discretization are summarized in Table 4.3.

Similarly, some learning models are unable to handle continuous input attributes. In this study, CN2 is unable, in its current incarnation, to handle continuous attributes. Thus, before presentation to CN2, all continuous-valued input attributes in all database are discretized using the Kerber's Chi-Merge method [69]. Note that even though continuous-valued input attributes are discretized before presentation to the CN2 software, the comments made above regarding the composition of the training and testing sets and presentation order still apply, that is, the composition and order of the training and testing sets with discretized attributes is identical to the undiscretized training and testing sets. While it could be argued that all learning models should also use the discretized data, we felt that this would unfairly penalize those models which are able to handle continuous attributes.

Database Statistics
Table 4.3

Name	NI/NUI/NUV	B/C/D/I	NA	NC	Volume	Density	Noise	DA	Entropy
auto	398 6 6	0/5/3/0	8	5	8.2277e+18 4.8191e+12	4.8373e-17 8.2587e-11	0 0 0	33.7	2.0652
house	506 0 0	1/13/0/0	14	5	2.1406e+50 5.6343e+32	2.3638e-48 8.9808e-31	0 0 0	47.2	1.9386
import	201 42 51	4/15/7/0	26	5	2.0580e+52 1.9555e+39	9.7667e-51 1.0279e-37	0 0 0	62.7	1.4476
lrs	531 0 0	0/102/0/0	102	10	4.6265e+806 2.1586e+264	1.1477e-804 2.4599e-262	0 0 0	51.4	2.0309
mache	209 0 0	0/8/1/0	9	10	2.6030e+30 5.2257e+17	8.0293e-29 3.9995e-16	4 0 0	76.6	1.2670
machp	209 0 0	0/8/1/0	9	10	2.6030e+30 5.2257e+17	8.0293e-29 3.9995e-16	4 0 0	76.6	1.2670

4.4 Unknown Attribute Values

Real-world data frequently contains instances with missing input attribute values. Consequently, most learning models handle missing values in some way. However, exceptions include all connectionist models used in this study. Since 35 of the databases used in this study contain instances with missing input attribute values, it was necessary to use some method of handling missing input attribute values for the connectionist models. Even though connectionist models such as back-propagation have been used in the academic and commercial communities for many years, there is still no commonly accepted method for handling unknown values. Many methods have been proposed [42, 57, 79, 80, 95, 103, 127] with varying degrees of success.

Note that 4 of the databases (*import*, *bridg1*, *bridg2*, and *horse*) used in this study also contain instances with missing output attribute values. Rather than guess at the values of instances with missing class values, these instances were simply discarded.

For connectionist learning models, methods of handling missing attribute values can be

categorized in one of two ways: Pre-processing methods and real-time methods. Pre-processing transforms the unknown values before they are given to the learning model and include methods such as (1) Discarding instances with unknown values; (2) Replacing unknown values with some value such as the average value for the attribute [95, 127]; (3) Replacing unknown values with a special value [95, 127], for example zero; or (4) Replacing unknown values with an estimated value calculated by statistical means such as the Expectation-Maximization (EM) algorithm [11, 57, 95, 127]. Since these methods transform the data before it is given to the connectionist learning model, each of the pre-processing methods works either during training or testing.

Real-time methods transform the unknown values as they are given to the learning model and include methods such as (1) Varying the unknown input value over its range to find a global minimum for the classification error [127]; (2) Training a subnet for each combination of unknown values and known values so the subnet would yield the most likely value for the unknown input value [127]; (3) Associating with each attribute that has unknown values a special input node which, when turned "on", indicates that the value for that attribute is unknown [127];. Since the first two methods require excessive processing time, they may be rejected out of hand for a study using 102 databases.

In this study, unknown values for the connectionist models were handled by the third real-time method, that is, by adding a special input node to every attribute (including those attributes with no unknown values) for any database with missing attribute values. When the input value for the attribute was known, the input node(s) of the attribute were given values according to the representation selected (see Appendix D) and the special input node was

"off", that is, its input value was such that the activation function would yield its minimum with this value as its argument. When the input value for the attribute was unknown, the input node(s) of the attribute were "off", and the special node was "on", that is, its input value was such that the activation function would yield its maximum with this value as its argument.

4.5 Learning Model Parameters

In order to ensure reproducibility of results, the parameters used for each learning model must be recorded [97]. Unless otherwise indicated, default parameters are used for each machine learning model, and the same parameters are used for each database. Because there is no single set of parameters that gives "reasonable results" for connectionist learning models for all databases, parameters for connectionist models are varied until "reasonable results" are obtained. The actual parameter values for all databases for the machine learning and connectionist models are listed in Appendix D.

5. Results

Results in case studies are typically presented as an *accuracy*, that is, the percent of the test instances that are correctly classified by the algorithm after training on the training instances. Some researchers [97, 105] prefer to report the *error rate* or the percent of test instances that are incorrectly classified. The *error rate* may give a representation of algorithm performance that is less misleading as Prechelt claims [97], however, since the *error rate* is simply 100% - *accuracy* and since reporting *accuracy* makes comparison with default accuracy easier, we report algorithm performance using *accuracy*.

In addition, we also report *average information score* and *relative information score* using Kononenko and Bratko's information-based evaluation criterion [74]. An algorithm can achieve an accuracy equal to the default accuracy of a problem simply by choosing the majority class, or the class with the highest probability, for every instance. Thus, for problems with high default accuracies, such as the thyroid databases, an algorithm would seem to perform well by using the straightforward expedient of always choosing the class with highest probability. However, the *average information score* gives an indication of how much information an algorithm is actually adding to the information already contained in the problem.

The information-based evaluation criterion computes an algorithm's *average information score* by rewarding correct classification of an instance more if it belongs to a minority class and by penalizing the incorrect classification of an instance more if it belongs to a majority class. Using $P(C)$ as the probability of class C and $P'(C)$ as the probability computed by an algorithm for class C, the information score in bits for classifying an instance belonging to

class C correctly (the algorithm reports that $P'(C) \geq P(C)$) is

$$I = V_c = -\log_2 P(C) + \log_2 P'(C)$$

while for misleadingly or incorrectly classifying an instance belonging to class C (the algorithm reports that $P'(C) < P(C)$) it is

$$V_c = -\log_2 P(1-C) + \log_2 P'(1-C)$$

$$I = -V_c$$

The *average information score*, I_a, is the average of information scores of all test instances or

$$I_a = 1/n_{test} \sum_1 I_i$$

where n_{test} is the number of test instances. The *relative information score*, I_r, is defined as

$$I_r = I_a / E,$$

where E is the entropy of the database (see Table 2.2). The definition of I_r assumes that the class distribution in both the training and test sets is equal. This is not always achievable as previously explained (cf. section 4.1). (For a thorough discussion of the information-based evaluation criterion see [74].)

The *average information score* is also a means by which the performance of an algorithm can be compared in different problem domains. For example, simply comparing the classification accuracy of an algorithm on *allhyper* which has a default accuracy of 97.3% to that on *audio* which has a default accuracy of 25.2% may lead one to believe that the algorithm is performing better in the *allhyper* problem domain than it is on the *audio* domain. However, this may not be the case since if the algorithm correctly classifies 50% of the test instances in *audio* and 99% in *allhyper*, it is actually adding more information to *audio* than

it is to *allhyper*. The *average information score* provides a way to compare performance of a classifier in different domains.

Except as explained in section 4.5 and in appendix D, no effort was made to tune algorithm parameters to yield optimal performance. However, by default, c4.5 tree gives accuracies for both the "raw" and "pruned" decision tree. The result reported in Table 5.1 is the better of the two. Whether the "raw" or "pruned" result was chosen is designated with subscript "$_r$" or "$_p$" respectively. No subscript indicates the performance of both is equal. For back-propagation, several, in some cases as many as several hundred, cross-validation trials were done using a different set of parameters for each trial. Likewise some experimentation was done with perceptron parameters, typically those affecting data representation. The results reported in Table 5.1 are the best of these trials.

The results of the cross-validation trials are presented in Table 5.1. The column labeled DA contains the default accuracy and the number of instances for each database (this information is also contained in Table 2.2 and is repeated here for ease of comparison). The first four numbers for each entry in the table represent respectively the average test accuracy over all cross-validations, the standard deviation of the test accuracy over all cross-validations, the average of *average information scores*, and the average of *relative information scores* over all cross-validations. If an entry does not have a fifth number, it indicates that all instances (the second number in the DA column) in the database were used in this trial. If an entry does have a fifth number, it is the number of instances actually used.

If fewer than all the instances in a database were used in a trial, it was invariably because

the software could not cope with a large database[*]. When this occurred, approximately half of the instances in the databases were selected and the cross-validation trial was re-run. This was repeated until the trial completed successfully. Note that in selecting instances, the class distribution of the original database was maintained as closely as possible (cf. section 4.1). Furthermore, because some software was unable to contend with databases having a large number of attributes, some entries in the table are empty[**]. Databases for which 5-fold rather than 10-fold cross-validation was done are marked with [5] (superscript 5).

The bolded entries in Table 5.1 indicate the algorithm which performs best on that database. For each database the best algorithm was chosen as follows:

1. Examine each *accuracy* in the row and pick the biggest ones. If there are no ties, mark the entry and stop; otherwise continue.

2. Examine the *average information scores* of the entries picked in step 1and pick the biggest ones. If there are no ties, mark the entry and stop; otherwise continue.

3. Examine the *relative information scores* of the entries picked in step 2 and pick the biggest ones. If there are no ties, mark the entry and stop; otherwise continue.

4. Examine the *standard deviations* of the entries picked in step 3 and pick the smallest ones. Mark all entries that have been picked.

Note that the averages column does not include the default accuracy (DA). The second number for each database in the averages column is the standard deviation of the accuracies

---

[*] It is not surprising that software, especially non-commercial software, should not be able to contend with large databases since the largest database, *stshut*, contains 58,000 instances and several databases contain more than 20,000 instances. The instance-based learning, CN2, and c4.5 rules software especially suffered from a lack of robustness when learning large databases.

[**] One could expect that non-commercial software not be able to contend with databases having many attributes. Three databases, *isol5*, *musk1*, and *musk2* have 618, 168, and 168 attributes respectively.

in that row. The second number for each algorithm in the averages row is the average of the standard deviations in the column.

The performance of the *c4.5 tree* algorithm exceeded that of the *c4.5 rule* alogrithm by 10.0% or more on the following problems: *auto, balanc, cmusp, cmuvow, crx, heart, hearth, kinsh, krvskp, musk2, netstr, slc, splice, ssblow, ssonly, staust, sthear, stsati, stvehi, tictac, trains*. Since *c4.5 rule* uses the decision tree created by *c4.5 tree* to produce rules, one can speculate that in the aforementioned cases, either that *c4.5 rule* is discarding useful information from the decision tree or that production rules cannot represent the problem as well as a decision tree. In only one case (*hayes*) did the performance of *c4.5 rule* exceed that of c4.5 tree by 10.0% or more.

The performance of the *ib3* and *ib4* algorithms was very poor on databases with 32 or fewer instances but improved on *soyf, soys,* and *soyso* which each have 47 instances. This behavior could be anticipated since instance-based learners require sample sizes sufficiently large to learn effectively [3]. For different reasons *ib1* and *ib2* did not perform well on *cmupa7* or *cmupa8* [7].

The performance of the *ocn2* and *ucn2* algorithms was likewise poor on the parity problems *cmupa7* and *cmupa8* , again as a result of the nature of the algorithm [35, 36, 37]. That these algorithms do not perform well on the parity problems is not a serious deficiency since these are purely artificial problems and have been included in this study only because they are included in many other studies.

Test Results Table 5.1

Name	DA	c45 rule	ocn2	ucn2	c45 tree	c4	id3	mml	smml
adult![5]	60.0 20	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***
adult-[5]	60.0 20	85.0 33.5 0.5623 0.5535	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	85.0 33.5 0.5623 0.5535	85.0 33.5 0.5623 0.5535	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	85.0 33.5 0.5623 0.5535
agaric	51.8 8124	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***
ahyper	97.3 3772	98.6 0.4 0.1253 0.6078	97.3 0.3 0.0368 0.1789	97.2 0.3 0.0352 0.1720	***98.9*** r ***0.3*** ***0.1387*** ***0.6745***	98.4 0.5 0.1072 0.5195	98.9 0.6 0.1379 0.6704	98.8 0.5 0.1301 0.6317	98.7 0.5 0.1273 0.6210
ahypo	92.3 3772	99.4 0.3 0.4377 0.9411	93.8 0.6 0.1928 0.4146	92.2 0.5 0.1030 0.2210	99.5 0.3 0.4411 0.9486	99.5 0.3 0.4407 0.9478	99.5 0.3 0.4409 0.9482	99.5 0.4 0.4392 0.9444	99.5 0.4 0.4419 0.9503
allbp	95.7 3772	95.9 0.7 0.0750 0.2727	95.5 0.5 0.0509 0.1842	96.1 0.5 0.0766 0.2794	97.3 p 0.7 0.1436 0.5253	97.3 0.8 0.1390 0.5074	97.0 0.6 0.1315 0.4788	***97.7*** ***0.8*** ***0.1568*** ***0.5727***	97.0 0.4 0.1210 0.4415
allrep	96.7 3772	98.9 0.3 0.1913 0.7359	96.5 0.8 0.0477 0.1843	96.7 0.1 0.0461 0.1775	***99.2*** p ***0.3*** ***0.2103*** ***0.8100***	***99.2*** ***0.3*** ***0.2103*** ***0.8105***	98.8 0.4 0.1850 0.7120	99.1 0.5 0.2023 0.7797	99.1 0.4 0.2072 0.7984
ann	92.6 7200	***99.8*** ***0.2*** ***0.4375*** ***0.9775***	93.5 0.4 0.1627 0.3636	93.5 0.3 0.1554 0.3471	99.7 p 0.2 0.4350 0.9720	99.6 0.2 0.4292 0.9589	99.6 0.2 0.4298 0.9603	99.7 0.1 0.4337 0.9690	99.5 0.2 0.4291 0.9587
anneal	76.2 898	89.5 2.8 0.8214 0.6967	98.6 1.3 1.1327 0.9589	97.6 1.5 1.0947 0.9266	94.5 r 1.4 1.0064 0.8516	82.6 1.7 0.5440 0.4591	95.5 1.9 1.0304 0.8728	88.8 5.1 0.8309 0.7047	86.6 3.8 0.7657 0.6494
audio	25.2 226	77.5 9.1 2.1622 0.7211	81.1 10.6 2.2796 0.7601	78.0 7.8 2.1747 0.7255	77.5 p 8.4 2.1663 0.7231	77.1 9.4 2.1357 0.7125	78.8 8.5 2.2408 0.7460	81.9 9.5 2.3305 0.7779	73.9 9.7 2.0296 0.6779
auto	33.7 398	56.5 10.9 0.9131 0.4482	68.1 10.0 1.1964 0.5843	69.6 8.6 1.2384 0.6042	71.3 r 9.1 1.2974 0.6375	73.9 7.4 1.3586 0.6660	71.3 7.7 1.3037 0.6404	67.8 6.3 1.2116 0.5946	66.3 6.9 1.1617 0.5699
balanc	46.1 625	48.5 2.1 0.1426 0.1079	88.7 5.0 0.9884 0.7547	79.5 4.8 0.7652 0.5842	68.2 r 3.9 0.5367 0.4090	78.9 4.9 0.7524 0.5744	68.2 4.6 0.5367 0.4094	63.4 5.4 0.4401 0.3361	62.4 5.1 0.4206 0.3212
breast	70.3 286	69.7 7.2 0.1909 0.2190	66.1 6.0 0.1075 0.1202	70.7 7.8 0.2144 0.2452	73.9 p 7.1 0.2860 0.3277	71.4 5.4 0.2294 0.2622	66.2 8.5 0.1114 0.1275	***75.3*** ***7.8*** ***0.3178*** ***0.3641***	71.7 9.9 0.2341 0.2654

Test Results Table 5.1

Name	ib1	ib2	ib3	ib4	bayes	cart	per	bp	Average
adult![5]	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	85.0 13.7 0.5830 0.6023	55.0 20.9 -0.0585 -0.0954	50.0 17.7 -0.1585 -0.1954	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	90.0 22.4 0.6830 0.7023	***100.0*** ***0.0***	***100.0*** ***0.0***	92.5 15.7 0.7353 0.7867
adult-[5]	85.0 22.4 0.5623 0.5535	90.0 13.7 0.7038 0.7512	40.0 13.7 -0.3792 -0.4442	50.0 25.0 -0.1170 -0.0977	100.0 0.0 0.9245 1.0000	85.0 33.5 0.5623 0.5535	***100.0*** ***0.0***	***100.0*** ***0.0***	86.9 17.3 0.5860 0.6093
agaric	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.1*** ***0.9981*** ***0.9990***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.1*** ***0.9983*** ***0.9993***	***100.0*** ***0.0*** ***0.9991*** ***1.0000***	***100.0*** ***0.0*** ***0.9986*** ***0.9995***	99.7 0.6	***100.0*** ***0.0***	100.0 0.1 0.9989 0.9998
ahyper	97.5 0.4 0.0548 0.2663	95.5 1.0 -0.0483 -0.2352	93.9 1.3 -0.1268 -0.6133	94.1 1.8 -0.1136 -0.5532	98.8 0.4 0.1339 0.6503	98.5 0.5 0.1090 0.5296	97.9 0.5	98.1 0.5	97.6 1.6 0.0605 0.2943
ahypo	91.4 1.1 0.0993 0.2140	84.7 1.3 -0.1534 -0.3303	79.9 3.5 -0.3362 -0.7245	79.4 3.3 -0.3529 -0.7586	***99.5*** ***0.4*** ***0.4413*** ***0.9491***	99.4 0.6 0.4378 0.9415	93.7 1.7	93.7 1.7	94.0 6.8 0.2192 0.4714
allbp	96.2 0.5 0.0909 0.3316	93.8 1.3 -0.0196 -0.0712	91.3 2.4 -0.1370 -0.4988	90.0 2.8 -0.1960 -0.7099	97.4 0.8 0.1517 0.5542	97.0 0.8 0.1220 0.4453	96.3 1.1	96.7 0.9	95.8 2.2 0.0647 0.2367
allrep	96.5 0.6 0.0480 0.1832	95.0 0.7 -0.0204 -0.0801	90.7 1.7 -0.2213 -0.8577	90.2 2.4 -0.2499 -0.9679	99.2 0.4 0.2090 0.8052	99.1 0.5 0.2051 0.7904	96.6 0.5	97.5 0.4	97.0 2.8 0.0908 0.3487
ann	92.6 0.6 0.1286 0.2874	86.2 1.3 -0.1175 -0.2626	83.7 2.0 -0.2096 -0.4686	83.4 2.9 -0.2227 -0.4979	99.7 0.1 0.4338 0.9693	99.6 0.2 0.4291 0.9588	95.0 0.5	*94.3* *2.4*	88.1 22.9 0.1789 0.3995
anneal	95.1 1.0 1.0090 0.8534	96.9 2.3 1.0836 0.9165	93.4 3.4 0.9831 0.8325	84.0 3.3 0.7285 0.6163	92.1 5.0 0.9130 0.7753	92.3 3.3 0.9225 0.7821	96.3 3.4	***99.3*** ***0.6***	92.7 4.9 0.9190 0.7783
audio	77.5 9.8 2.1537 0.7192	72.2 9.9 1.9707 0.6563	63.3 9.4 1.6742 0.5567	68.2 10.0 1.7877 0.5945	***85.4*** ***7.7*** ***2.4845*** ***0.8274***	71.7 9.0 1.9125 0.6387	76.7 11.2	81.9 8.5	76.4 5.4 2.1073 0.7026
auto	70.3 4.9 1.2723 0.6239	66.1 4.5 1.1809 0.5777	66.5 9.4 1.1824 0.5796	65.8 9.2 1.1652 0.5722	71.3 6.7 1.2905 0.6334	67.8 8.6 1.1953 0.5868	68.3 8.1	***74.3*** ***8.0***	68.4 4.0 1.2120 0.5942
balanc	84.5 3.4 0.8648 0.6596	66.7 6.0 0.5075 0.3879	70.7 7.9 0.5885 0.4499	62.7 6.7 0.4386 0.3359	67.5 5.7 0.5238 0.3998	79.4 5.4 0.7618 0.5816	***89.8*** ***3.5***	*80.4* *23.1*	72.5 10.8 0.5906 0.4508
breast	71.8 7.5 0.2386 0.2742	64.0 9.6 0.0604 0.0664	67.9 7.7 0.1503 0.1717	67.6 13.0 0.1428 0.1629	69.3 10.0 0.1807 0.2053	71.4 5.0 0.2287 0.2617	66.5 9.8	73.5 9.4	69.8 3.1 0.1924 0.2195

Test Results Table 5.1

Name	DA	c45 rule	ocn2	ucn2	c45 tree	c4	id3	mml	smml
breastw	65.5 699	86.7 5.9 0.6434 0.6926	95.2 2.1 0.8250 0.8881	93.0 2.9 0.7788 0.8381	94.7 p 2.0 p 0.8158 0.8781	95.1 3.8 0.8248 0.8876	94.3 2.6 0.8062 0.8676	94.8 1.8 0.8187 0.8812	94.7 1.6 0.8157 0.8779
bridg1	40.7 108	52.6 17.0 0.7291 0.3530 106	58.2 13.6 1.0071 0.4646 106	61.1 13.8 1.0093 0.4759 106	56.5 p 11.8 p 0.8383 0.4003 106	53.7 10.2 0.7735 0.3625 106	61.3 13.6 1.0273 0.4896 106	60.4 12.0 0.9055 0.4269 106	50.0 7.1 0.6482 0.3051 106
bridg2	40.7 108	60.3 9.9 0.9643 0.4511 106	60.4 11.8 1.0660 0.4929 106	66.1 8.7 1.1586 0.5380 106	***68.2*** r ***15.2*** r ***1.1898*** ***0.5411*** ***106***	64.1 6.3 1.0304 0.4831 106	61.3 13.2 1.0080 0.4635 106	60.4 10.4 0.8990 0.4193 106	50.9 5.6 0.6616 0.3107 106
bupa	58.0 345	59.1 4.8 0.1442 0.1482	58.0 3.4 0.1205 0.1237	58.0 3.4 0.1205 0.1237	62.6 r 8.2 r 0.2134 0.2166	64.4 8.8 0.2501 0.2552	65.2 9.5 0.2678 0.2732	67.5 8.2 0.3156 0.3226	66.4 8.9 0.2925 0.3004
cmupa7	50.0 128	45.4 4.0 -0.1151 -0.1179	3.1 4.1 -0.9682 -0.9797	0.0 0.0 -1.0312 -1.0429	36.0 p 11.3 p -0.3041 -0.3075	29.7 7.9 -0.4313 -0.4362	3.9 4.1 -0.9522 -0.9630	29.7 9.4 -0.4302 -0.4340	45.4 4.0 -0.1151 -0.1179
cmupa8	50.0 256	44.5 4.6 -0.1420 -0.1465	2.3 3.3 -0.9964 -1.0122	0.4 1.3 -1.0352 -1.0513	35.2 p 6.4 p -0.3297 -0.3353	35.2 6.4 -0.3297 -0.3353	2.0 2.1 -1.0032 -1.0188	39.5 5.9 -0.2429 -0.2475	44.5 4.6 -0.1420 -0.1465
cmupro	54.5 21627	54.5 0.1 0.3054 0.2114 10811	53.5 1.7 0.3642 0.2521 10818	55.2 0.9 0.3531 0.2444 5406	56.4 p 0.8 p 0.3864 0.2674	52.7 1.3 0.3559 0.2463	53.4 0.7 0.3732 0.2583	55.6 0.8 0.3534 0.2446	54.8 0.7 0.3261 0.2257
cmuson	53.4 208	65.8 5.0 0.3052 0.3071	48.0 9.4 -0.0546 -0.0554	51.4 6.9 0.0138 0.0137	63.9 5.9 0.2665 0.2683	69.6 13.2 0.3814 0.3837	73.5 13.0 0.4602 0.4629	74.5 10.8 0.4802 0.4833	70.6 12.6 0.4022 0.4053
cmusp	50.0 194	52.1 10.0 0.0038 0.0026	42.8 3.8 -0.1868 -0.1920	42.8 3.8 -0.1868 -0.1920	63.9 r 5.9 r 0.2438 0.2470	61.5 12.8 0.1967 0.2019	63.4 9.8 0.2348 0.2390	43.3 4.1 -0.1761 -0.1810	42.8 3.8 -0.1868 -0.1920
cmusp2	50.0 386	50.8 5.7 0.0043 0.0034	47.7 3.7 -0.0577 -0.0586	47.1 3.3 -0.0682 -0.0692	78.9 p 8.6 p 0.5719 0.5749	79.8 9.7 0.5890 0.5926	88.0 5.6 0.7543 0.7579	53.9 10.1 0.0694 0.0698	48.2 4.0 -0.0471 -0.0481
cmuvow	9.1 990	61.5 5.7 2.0752 0.5998	58.9 4.5 1.9807 0.5726	59.7 5.5 2.0098 0.5810	79.9 r 4.2 r 2.7364 0.7910	79.8 4.2 2.7328 0.7899	80.8 2.9 2.7691 0.8005	77.9 4.4 2.6637 0.7700	71.6 5.1 2.4385 0.7049
crx	55.5 690	59.6 4.7 0.1752 0.1768	83.0 3.9 0.6489 0.6548	82.6 4.3 0.6402 0.6460	83.5 p 3.9 p 0.6577 0.6635	83.2 3.8 0.6518 0.6577	80.0 3.8 0.5875 0.5928	84.5 3.3 0.6781 0.6842	83.2 4.1 0.6518 0.6577
dis	98.5 3772	98.5 0.4 0.0266 0.2351	98.3 0.2 0.0121 0.1052	98.5 0.1 0.0217 0.1890	98.9 p 0.5 p 0.0473 0.4134	98.4 0.4 0.0152 0.1322	98.3 0.7 0.0105 0.0917	***99.1*** ***0.3*** ***0.0600*** ***0.5214***	98.1 0.6 0.0023 0.0129

Test Results Table 5.1

Name	ib1	ib2	ib3	ib4	bayes	cart	per	bp	Average
breastw	***96.3*** ***1.9*** ***0.8493*** ***0.9140***	91.0 2.8 0.7357 0.7917	95.0 4.0 0.8218 0.8845	94.1 3.1 0.8036 0.8651	93.6 3.8 0.7911 0.8515	94.4 2.4 0.8096 0.8715	93.0 2.9	***96.3*** ***2.0***	93.9 2.2 0.7957 0.8564
bridg1	60.6 17.1 1.0042 0.4688 106	51.1 17.8 0.7609 0.3565 106	56.8 15.3 0.8900 0.4187 106	55.8 17.4 0.8666 0.4113 106	66.1 10.6 1.0949 0.5151 106	56.5 8.9 0.8101 0.3818 106	64.0 17.8	67.6 16.5 106	58.3 5.0 0.8832 0.4164
bridg2	59.5 17.8 0.9540 0.4532 106	52.7 14.8 0.8094 0.3793 106	54.0 17.6 0.8170 0.3859 106	57.5 17.7 0.9552 0.4568 106	66.0 14.5 1.1170 0.5135 106	64.3 12.9 1.0217 0.4757 106	59.4 11.4	***62.4*** ***11.1*** ***106***	60.5 4.7 0.9751 0.4546
bupa	62.3 11.3 0.2077 0.2112	61.7 9.9 0.1962 0.2009	53.6 10.5 0.0312 0.0317	61.4 7.6 0.1910 0.1961	64.6 7.7 0.2562 0.2620	67.3 9.6 0.3112 0.3193	66.4 9.2	***69.0*** ***9.1***	63.0 4.1 0.2084 0.2132
cmupa7	0.0 0.0 -1.0312 -1.0429	0.0 0.0 -1.0312 -1.0429	48.4 6.6 -0.0524 -0.0525	53.1 7.4 0.0427 0.0440	3.9 4.1 -0.9522 -0.9630	45.4 4.0 -0.1151 -0.1179	31.2 9.5	***91.3*** ***5.9***	29.2 25.2 -0.5348 -0.5410
cmupa8	0.0 0.0 -0.4241 -0.4336	0.0 0.0 -0.4241 -0.4336	22.4 29.3 0.0328 0.0329	18.0 23.3 -0.0574 -0.0596	2.3 2.0 -0.9955 -1.0111	44.5 4.6 -0.1420 -0.1465	41.5 10.4	***77.4*** ***27.2***	25.6 22.4 -0.4451 -0.4532
cmupro	***64.1*** ***0.9*** ***0.6127*** ***0.4240***	52.5 1.7 0.3631 0.2513 10813	52.1 1.5 0.3551 0.2457	43.2 6.8 0.1726 0.1195	56.8 0.8 0.4210 0.2913	57.5 0.6 0.4140 0.2865	44.5 2.7	54.5 0.1	53.8 4.7 0.3683 0.2549
cmuson	***86.5*** ***6.2*** ***0.7224*** ***0.7272***	82.7 8.1 0.6453 0.6497	72.6 6.2 0.4410 0.4439	73.1 6.5 0.4514 0.4541	75.0 10.5 0.4895 0.4924	71.6 9.4 0.4209 0.4236	74.4 9.6	*71.2* *18.6*	70.3 9.4 0.3875 0.3900
cmusp	***80.3*** ***8.9*** ***0.5802*** ***0.5918***	78.3 11.2 0.5382 0.5477	52.2 11.4 0.0049 0.0039	51.8 15.0 -0.0027 -0.0035	50.4 10.8 -0.0318 -0.0346	57.9 14.1 0.1232 0.1263	49.0 10.1	57.2 3.8	66.5 17.8 0.3571 0.3587
cmusp2	***99.7*** ***0.8*** ***0.9896*** ***0.9946***	93.3 4.3 0.8600 0.8645	56.4 8.5 0.1181 0.1183	53.9 11.6 0.0667 0.0665	85.2 6.1 0.6977 0.7017	73.0 8.1 0.4520 0.4541	51.3 10.1	56.8 9.5	55.6 11.3 0.0825 0.0832
cmuvow	***98.8*** ***0.8*** ***3.4158*** ***0.9874***	94.9 1.9 3.2778 0.9475	91.9 1.8 3.1688 0.9160	92.1 1.9 3.1760 0.9181	80.7 4.3 2.7655 0.7994	77.2 4.5 2.6383 0.7626	51.3 8.3	80.1 7.1	77.3 13.4 2.7035 0.7815
crx	81.3 2.9 0.6138 0.6193	74.1 5.6 0.4675 0.4715	82.5 3.6 0.6372 0.6430	84.2 2.9 0.6723 0.6784	82.2 4.0 0.6313 0.6369	85.5 4.7 0.6986 0.7048	83.6 4.5	***85.1*** ***3.6***	81.1 6.1 0.6009 0.6062
dis	98.1 0.5 0.0022 0.0101	96.0 1.0 -0.1299 -1.1699	90.6 5.9 -0.4585 -4.1551	92.9 2.3 -0.3125 -2.7622	98.8 0.5 0.0441 0.3864	98.5 0.3 0.0248 0.2132	97.9 0.7	98.6 0.3	97.5 2.3 -0.0453 -0.4126

Test Results Table 5.1

Name	DA	c45 rule	ocn2	ucn2	c45 tree	c4	id3	mml	smml
echoc	67.2 131	90.0 7.3 0.6897 0.7553	83.2 10.2 0.5384 0.5876	71.7 9.1 0.2888 0.3148	90.1 p 6.2 p 0.6917 0.7584	90.1 6.2 0.6917 0.7584	86.2 6.1 0.6071 0.6647	***92.4*** ***5.1*** ***0.7412*** ***0.8131***	***92.4*** ***5.1*** ***0.7412*** ***0.8131***
flag	35.6 194	50.5 13.7 0.7998 0.3726	51.6 7.2 0.8262 0.3825	55.6 12.0 0.9319 0.4259	56.2 p 12.0 p 0.9383 0.4329	50.5 15.2 0.8002 0.3685	51.9 9.6 0.8873 0.4033	56.6 11.4 0.9674 0.4437	52.6 9.4 0.8448 0.3887
glass	35.5 214	61.2 10.1 1.1310 0.5335	59.8 5.6 1.0298 0.4854	64.9 8.7 1.1321 0.5339	65.8 r 9.6 r 1.2391 0.5828	63.5 9.1 1.1959 0.5635	69.1 6.5 1.3153 0.6201	63.5 8.4 1.1555 0.5440	62.7 11.5 1.1163 0.5265
hayes	33.3 93	74.6 13.6 0.9907 0.6466	74.6 11.6 0.9814 0.6414	72.1 13.8 0.9326 0.6100	59.7 r 17.3 r 0.6524 0.4284	73.3 9.8 0.9667 0.6279	57.6 20.0 0.6094 0.4008	61.8 17.9 0.6973 0.4594	55.2 12.4 0.5593 0.3673
heart	41.7 929	42.5 1.2 0.4016 0.1964	68.8 4.0 1.1862 0.5802	61.1 3.2 0.9359 0.4577	63.7 r 5.0 r 1.0554 0.5163	58.8 4.1 0.8954 0.4379	***74.0*** ***4.3*** ***1.3268*** ***0.6490***	54.9 3.9 0.7725 0.3780	47.9 4.0 0.5723 0.2800
heartc	54.1 303	53.8 5.9 0.3871 0.2129	53.1 6.3 0.4802 0.2594	53.8 5.6 0.4129 0.2251	51.4 p 4.4 p 0.4022 0.2180	51.5 8.7 0.3746 0.2056	49.4 8.2 0.3852 0.2097	54.1 5.9 0.4563 0.2487	50.5 5.4 0.3491 0.1918
hearth	63.9 294	65.3 8.7 0.1954 0.2061 294	79.2 6.0 0.4937 0.5270	78.6 9.5 0.4795 0.5118	79.3 p 9.6 p 0.4929 0.5239	80.3 9.8 0.5169 0.5525	76.5 7.7 0.4344 0.4628	81.0 8.0 0.5306 0.5668	81.3 8.1 0.5376 0.5738
hearts	39.0 123	29.1 14.2 0.0216 0.0189	34.8 16.8 0.1725 0.0920	38.1 20.0 0.2439 0.1396	27.6 p 12.2 p 0.0623 0.0334	33.6 13.2 0.1508 0.0800	35.0 10.0 0.2088 0.1218	38.2 13.8 0.2695 0.1534	40.5 13.2 0.2658 0.1512
heartv	28.0 200	29.5 4.4 0.3034 0.1399	26.0 11.7 0.2472 0.1138	32.5 8.6 0.3945 0.1818	31.5 r 11.8 r 0.3929 0.1809	29.0 5.2 0.3271 0.1506	***34.5*** ***7.6*** ***0.4650*** ***0.2142***	27.0 4.2 0.2370 0.1092	28.0 2.6 0.2596 0.1196
hepat	54.8 155	57.4 11.2 0.1280 0.1307	63.3 11.5 0.2459 0.2488	60.8 13.1 0.1968 0.1999	55.7 p 15.1 p 0.0945 0.0977	62.0 12.2 0.2200 0.2237	56.3 13.2 0.1052 0.1076	63.4 11.1 0.2483 0.2521	58.2 8.8 0.1440 0.1467
horse	61.1 368	61.5 4.1 0.2998 0.2277 366	65.1 9.0 0.4653 0.3545 366	67.8 5.5 0.4798 0.3614 366	***70.0*** p ***5.0*** ***0.5835*** ***0.4421*** ***366***	68.9 9.0 0.5495 0.4167 366	67.8 5.9 0.5018 0.3786 366	66.7 7.5 0.4861 0.3673 366	63.7 8.0 0.4019 0.3052 366
house	47.2 506	66.8 7.4 1.0636 0.5501	67.2 6.4 1.0629 0.5501	72.6 5.4 1.1771 0.6099	***74.9*** p ***7.9*** ***1.2672*** ***0.6556***	75.1 4.9 1.2771 0.6612	69.8 6.7 1.1661 0.6038	73.9 8.0 1.2442 0.6440	73.5 6.9 1.2233 0.6334
hypoth	95.2 3163	99.2 0.3 0.2400 0.8670	95.2 0.4 0.0625 0.2262	95.3 0.2 0.0652 0.2358	99.2 r 0.4 r 0.2415 0.8724	99.2 0.4 0.2372 0.8567	98.7 0.5 0.2175 0.7855	99.2 0.3 0.2400 0.8673	99.2 0.4 0.2414 0.8721

Test Results Table 5.1

Name	ib1	ib2	ib3	ib4	bayes	cart	per	bp	Average
echoc	84.0 11.4 0.5596 0.6132	80.2 11.9 0.4749 0.5194	83.9 10.6 0.5544 0.6042	80.2 12.1 0.4741 0.5189	90.0 6.4 0.6909 0.7579	***92.4*** ***5.1*** ***0.7412*** ***0.8131***	87.0 5.2	89.3 7.5	86.4 5.5 0.6061 0.6637
flag	56.6 11.2 0.9848 0.4475	52.5 9.2 0.9055 0.4165	53.1 11.3 0.9197 0.4243	53.7 9.9 0.8701 0.3988	52.5 10.0 0.8358 0.3796	***62.4*** ***9.8*** ***1.0290*** ***0.4728***	45.3 5.0	58.2 6.9	53.7 3.8 0.8958 0.4113
glass	71.1 5.3 1.3365 0.6324	67.7 7.8 1.2598 0.5960	61.8 9.1 1.0937 0.5150	64.0 9.8 1.1613 0.5466	***71.8*** ***9.5*** ***1.3761*** ***0.6486***	63.6 10.5 1.1344 0.5344	56.4 9.9	68.7 8.6	64.7 4.0 1.1912 0.5616
hayes	61.7 15.0 0.7078 0.4587	57.3 17.2 0.5921 0.3843	50.8 15.4 0.4459 0.2882	53.0 14.2 0.5201 0.3425	57.6 20.0 0.6094 0.4008	72.2 11.1 0.9442 0.6138	***86.3*** ***11.9***	*83.0* *12.3*	65.7 10.6 0.7292 0.4764
heart	73.0 4.2 1.3107 0.6412	68.9 5.1 1.2193 0.5964	50.4 4.8 0.7050 0.3448	50.1 4.1 0.6941 0.3396	70.5 3.6 1.2250 0.5992	56.7 3.5 0.8416 0.4116	48.7 3.1	65.9 3.0	59.7 9.6 0.9387 0.4592
heartc	57.0 12.2 0.5772 0.3186	52.1 9.2 0.4698 0.2584	49.2 11.0 0.3752 0.2079	48.5 9.9 0.3702 0.2039	49.5 10.4 0.3877 0.2142	56.4 5.9 0.4863 0.2674	54.1 7.1	***58.1*** ***5.8***	52.7 2.8 0.4224 0.2315
hearth	77.6 9.5 0.4605 0.4942	73.5 12.4 0.3683 0.3902	80.6 8.5 0.5225 0.5569	77.3 8.9 0.4520 0.4829	75.8 9.7 0.4208 0.4484	80.6 4.9 0.5220 0.5561	81.3 10.0	***82.7*** ***7.3***	78.2 4.1 0.4591 0.4895
hearts	30.8 12.0 0.1214 0.0535	28.3 11.7 0.0716 0.0244	29.2 19.0 0.0373 0.0287	24.3 16.5 -0.0297 -0.0127	25.8 17.7 0.0086 0.0085	33.5 13.0 0.1700 0.0830	38.1 8.9	***44.7*** ***10.4***	33.2 5.5 0.1267 0.0697
heartv	32.0 10.6 0.4116 0.1897	28.0 10.3 0.3008 0.1388	26.5 7.5 0.2524 0.1164	26.0 8.4 0.2456 0.1132	30.0 9.7 0.3443 0.1588	28.0 8.9 0.2906 0.1339	29.5 9.8	33.0 13.2	29.4 2.5 0.3194 0.1472
hepat	66.6 16.7 0.3131 0.3176	65.8 11.7 0.2970 0.3003	63.5 14.1 0.2501 0.2541	54.1 11.7 0.0601 0.0605	57.5 15.9 0.1304 0.1329	59.4 7.5 0.1676 0.1696	67.3 11.9	***68.5*** ***13.7***	61.2 4.3 0.1858 0.1887
horse	64.8 6.1 0.4428 0.3325 366	59.9 7.4 0.3520 0.2658 366	56.1 8.6 0.2597 0.1963 366	62.0 10.8 0.4199 0.3158 366	68.6 7.7 0.5291 0.3971 366	67.6 9.6 0.4929 0.3742 366	60.1 5.4	66.9 6.0 366	64.8 3.8 0.4474 0.3382
house	70.3 6.2 1.1532 0.5969	65.6 5.7 1.0263 0.5309	56.5 5.2 0.8304 0.4299	59.5 6.0 0.8857 0.4590	73.3 6.7 1.2569 0.6510	72.1 6.0 1.1958 0.6193	68.6 2.6	73.3 5.1	69.6 5.2 1.1307 0.5854
hypoth	97.0 1.2 0.1414 0.5119	92.0 2.6 -0.0787 -0.2869	89.9 3.1 -0.1726 -0.6205	90.3 3.1 -0.1572 -0.5671	***99.3*** ***0.3*** ***0.2429*** ***0.8776***	99.1 0.4 0.2358 0.8516	98.0 1.0	*97.5* *1.4*	96.8 3.2 0.1255 0.4535

Test Results Table 5.1

Name	DA	c45 rule	ocn2	ucn2	c45 tree	c4	id3	mml	smml
import	62.7 201	83.6 5.8 0.9416 0.6717	76.6 7.5 0.7260 0.5167	79.6 5.9 0.7946 0.5666	83.6 r 4.7 0.9380 0.6721	82.1 6.8 0.9086 0.6451	84.6 7.3 0.9321 0.6650	83.1 4.9 0.9170 0.6539	81.6 4.7 0.8562 0.6120
iono	64.1 351	91.7 5.5 0.7641 0.8134	82.6 5.3 0.5712 0.6082	82.6 4.5 0.5707 0.6073	90.9 p 4.8 0.7463 0.7945	90.6 6.9 0.7407 0.7889	88.3 5.6 0.6906 0.7336	88.3 6.4 0.6906 0.7331	90.0 3.8 0.7272 0.7728
iris	33.3 150	94.0 4.9 1.4548 0.9178	92.7 6.6 1.4258 0.8996	93.3 5.4 1.4403 0.9087	94.0 4.9 1.4548 0.9178	94.0 4.9 1.4548 0.9178	93.3 7.0 1.4403 0.9087	92.7 7.3 1.4258 0.8996	92.7 7.3 1.4258 0.8996
isol5	3.8 1559	74.2 3.8 3.4556 0.7385			77.0 p 3.1 3.5865 0.7664	***77.8*** ***3.0*** ***3.6263*** ***0.7749***	76.3 3.3 3.5557 0.7598	76.6 3.6 3.5681 0.7625	75.2 2.5 3.5046 0.7489
kinsh	10.7 112	8.9 8.6 0.0746 0.0210	19.6 8.2 0.4394 0.1485	15.2 7.2 0.3036 0.1009	27.5 13.9 0.6739 0.2285	***40.1*** ***10.0*** ***1.0940*** ***0.3623***	27.5 13.9 0.6739 0.2285	27.5 13.9 0.6739 0.2285	27.5 13.9 0.6739 0.2285
krvskp	52.2 3196	65.4 17.5 0.3051 0.3056	99.5 0.4 0.9873 0.9887	98.4 0.8 0.9672 0.9686	99.2 r 0.5 0.9829 0.9843	99.2 0.6 0.9822 0.9837	***99.6*** ***0.3*** ***0.9910*** ***0.9925***	99.5 0.4 0.9873 0.9887	99.1 0.6 0.9804 0.9818
laborn[5]	66.7 27	50.7 20.3 -0.5395 -0.7680	***92.0*** ***11.0*** ***0.5353*** ***0.7070***	77.3 25.2 0.1791 0.2565	50.0 r 24.9 -0.5503 -0.7685	42.7 23.4 -0.7510 -1.0610	61.3 28.4 -0.2439 -0.3295	43.3 17.0 -0.7402 -1.0606	58.0 23.3 -0.3388 -0.4755
led17	12.0 200	65.5 6.9 2.0974 0.6399	61.0 5.2 1.9557 0.5969	64.0 10.7 2.0524 0.6260	66.5 p 8.2 2.1271 0.6492	69.5 7.6 2.2334 0.6815	60.0 11.8 1.9046 0.5812	68.5 10.0 2.2062 0.6731	69.5 7.6 2.2125 0.6750
led7	14.5 200	69.5 8.0 2.2271 0.6834	68.5 5.8 2.1852 0.6708	69.0 7.0 2.1976 0.6748	***70.0*** r ***7.1*** ***2.2420*** ***0.6880***	70.0 7.5 2.2394 0.6873	68.0 5.9 2.1561 0.6617	68.5 7.1 2.1898 0.6721	66.0 6.1 2.0984 0.6442
lense[5]	62.5 24	84.0 16.7 0.6300 0.5064	67.0 17.2 0.2388 0.1722	71.0 17.5 0.3445 0.3187	84.0 16.7 0.6300 0.5064	84.0 16.7 0.6300 0.5064	76.0 21.9 0.4419 0.3058	84.0 16.7 0.6300 0.5064	80.0 20.0 0.5477 0.4523
letter	4.1 20000	6.3 0.6 0.2419 0.0515 9990	74.8 1.1 3.4990 0.7445	65.6 1.6 3.0626 0.6517 9992	78.1 r 0.7 3.6582 0.7784	83.2 1.2 3.8991 0.8296	77.1 1.0 3.6125 0.7687	76.0 1.2 3.5592 0.7573	73.8 0.9 3.4530 0.7347
lrs	51.4 531	83.2 5.5 1.3963 0.7011	74.2 5.5 1.1709 0.5874	73.8 4.6 1.0948 0.5500	82.8 p 4.1 1.3873 0.6966	83.0 5.0 1.4000 0.7032	83.6 5.0 1.4266 0.7153	84.0 3.9 1.4489 0.7275	82.5 4.1 1.3790 0.6917
lungc[5]	40.6 32	50.5 25.3 0.4231 0.2721	36.7 14.8 0.1387 0.0917	38.1 20.0 0.1324 0.0809	47.6 r 23.1 0.3880 0.2507	47.6 18.1 0.3312 0.2144	50.0 12.8 0.4355 0.2841	50.5 25.3 0.4231 0.2721	53.3 9.0 0.4340 0.2829

Test Results Table 5.1

Name	ib1	ib2	ib3	ib4	bayes	cart	per	bp	Average
import	81.1 5.7 0.8355 0.5973	76.1 5.7 0.7289 0.5218	78.1 8.6 0.7927 0.5651	78.6 10.0 0.7922 0.5634	***86.0*** ***5.7*** ***1.0007*** ***0.7132***	81.6 4.7 0.8428 0.6019	72.1 10.3	83.6 6.2	80.8 3.6 0.8576 0.6118
iono	86.3 6.3 0.6504 0.6935	84.9 3.6 0.6194 0.6597	85.8 5.9 0.6381 0.6803	89.5 3.8 0.7163 0.7629	85.5 4.9 0.6309 0.6707	89.5 4.4 0.7150 0.7596	82.0 6.7	***92.0*** ***4.4***	87.5 3.2 0.6765 0.7199
iris	95.3 5.5 1.4837 0.9361	92.7 7.3 1.4259 0.8996	95.3 4.5 1.4837 0.9361	***96.6*** ***3.5*** ***1.5127*** ***0.9544***	94.7 6.9 1.4692 0.9270	92.0 6.1 1.4114 0.8905	95.3 5.5	96.0 4.7	94.0 1.3 1.4506 0.9152
isol5					75.4 2.5 3.5139 0.7509	75.3 4.9 3.5077 0.7496	64.7 4.9		74.7 3.7 3.5398 0.7564
kinsh	23.3 7.7 0.5697 0.1877	16.1 10.4 0.3149 0.1066	6.2 7.2 -0.0188 -0.0070	7.1 5.7 0.0206 0.0048	28.3 13.7 0.7091 0.2376	36.5 17.8 1.0337 0.3444	24.2 12.0	34.1 9.7	23.1 9.9 0.5169 0.1729
krvskp	96.1 0.8 0.9208 0.9222	91.7 1.4 0.8318 0.8330	89.3 1.5 0.7848 0.7860	91.0 1.5 0.8180 0.8192	99.6 0.3 0.9891 0.9906	98.9 0.6 0.9772 0.9786	95.6 2.8	96.8 3.3	94.9 8.3 0.8932 0.8945
laborn[5]	78.0 22.8 0.2126 0.3242	74.0 27.9 0.1068 0.1778	50.7 30.8 -0.5169 -0.7007	50.7 28.5 -0.5395 -0.7680	61.3 24.7 -0.2439 -0.3294	49.3 34.2 -0.5612 -0.7689	88.7 17.6	88.7 10.4	63.5 16.5 -0.2851 -0.3975
led17	43.5 8.8 1.3436 0.4104	39.5 5.0 1.2126 0.3701	39.5 10.9 1.2076 0.3683	64.0 8.4 2.0264 0.6185	64.5 9.3 2.0698 0.6316	***72.5*** ***11.6*** ***2.3213*** ***0.7081***	60.5 11.9	62.0 9.5	60.7 10.1 1.9265 0.5878
led7	68.5 8.5 2.1756 0.6678	63.5 8.2 2.0044 0.6155	68.5 8.8 2.1923 0.6730	68.0 10.6 2.1704 0.6661	68.5 6.7 2.1827 0.6699	69.5 8.6 2.2150 0.6797	70.0 7.1	*69.0* *9.1*	68.4 1.6 2.1769 0.6682
lense[5]	63.0 25.4 0.1564 0.1181	63.0 25.4 0.0862 -0.1514	35.0 31.2 -0.5839 -0.8380	46.0 26.1 -0.2348 -0.2280	76.0 21.9 0.4419 0.3058	72.0 33.5 0.3829 0.3441	84.0 16.7	92.0 11.0	72.6 14.7 0.3101 0.2018
letter	***85.5*** ***1.2*** ***4.0104*** ***0.8533*** ***9991***	83.4 1.0 3.9113 0.8322	81.6 1.2 3.8250 0.8139	79.5 1.1 3.7260 0.7928	78.1 0.6 3.6574 0.7782	83.5 0.8 3.9146 0.8329	26.7 2.8 987	*7.1* *3.8*	66.3 26.1 3.4307 0.7300
lrs	80.2 6.9 1.3323 0.6682	76.1 7.4 1.2370 0.6214	77.2 5.6 1.2482 0.6268	78.7 4.2 1.2868 0.6460	84.5 4.5 1.4673 0.7371	85.0 5.0 1.4248 0.7147	78.5 4.9	***88.5*** ***4.1***	81.0 4.1 1.3357 0.6705
lungc[5]	47.6 23.1 0.4163 0.2701	47.6 13.8 0.3880 0.2535	34.3 12.9 0.0400 0.0297	43.8 6.9 0.2634 0.1728	47.6 23.1 0.3651 0.2349	***60.5*** ***26.7*** ***0.6288*** ***0.4091***	41.0 9.7	55.9 22.9	47.0 6.8 0.3434 0.2228

Test Results Table 5.1

Name	DA	c45 rule	ocn2	ucn2	c45 tree	c4	id3	mml	smml
lymph	54.7 148	79.0 5.8 0.6398 0.5634	***84.4*** ***12.0*** ***0.7607*** ***0.6814***	78.9 10.8 0.6223 0.5585	79.1 p 7.9 0.6550 0.5753	73.6 13.3 0.5399 0.4875	76.3 9.3 0.6253 0.5377	77.6 9.8 0.6135 0.5490	73.0 8.9 0.5152 0.4607
mache	76.6 209	82.8 6.1 0.5204 0.4623	82.3 5.7 0.4809 0.4317	81.8 5.1 0.4485 0.3974	***84.2*** p ***4.5*** ***0.5619*** ***0.4926***	82.3 5.6 0.4916 0.4384	79.9 5.5 0.4098 0.3610	83.7 4.3 0.5354 0.4702	83.7 5.3 0.5374 0.4823
machp	76.7 209	82.8 6.1 0.5204 0.4623	82.3 5.7 0.4809 0.4317	81.8 5.1 0.4485 0.3974	84.2 p 4.5 0.5619 0.4926	81.8 5.9 0.4748 0.4256	79.9 5.5 0.4098 0.3610	83.2 4.3 0.5186 0.4574	83.7 5.3 0.5374 0.4823
monk1	50.0 556	***100.0*** ***0.0*** ***0.9987*** ***1.0000***	***100.0*** ***0.0*** ***0.9987*** ***1.0000***	***100.0*** ***0.0*** ***0.9987*** ***1.0000***	98.9 p 2.3 0.9769 0.9782	96.4 3.7 0.9266 0.9278	96.8 4.1 0.9340 0.9353	87.8 4.9 0.7542 0.7553	75.7 5.6 0.5126 0.5133
monk2	65.7 601	65.4 2.3 0.1825 0.1973	86.5 4.5 0.6361 0.6854	76.7 4.4 0.4259 0.4599	64.5 r 7.9 0.1651 0.1789	97.0 3.4 0.8626 0.9311	71.7 8.8 0.3195 0.3461	66.4 6.1 0.2043 0.2210	64.7 2.8 0.1682 0.1818
monk3	52.0 554	***98.9*** ***1.5*** ***0.9764*** ***0.9782***	97.7 1.9 0.9510 0.9528	***98.9*** ***1.5*** ***0.9764*** ***0.9782***	***98.9*** p ***1.5*** ***0.9764*** ***0.9782***	***98.9*** ***1.5*** ***0.9764*** ***0.9782***	97.5 1.9 0.9475 0.9492	***98.9*** ***1.5*** ***0.9764*** ***0.9782***	***98.9*** ***1.5*** ***0.9764*** ***0.9782***
musk1	56.5 476	73.5 12.8 0.4471 0.4553			82.4 6.7 0.6263 0.6371	82.2 4.8 0.6210 0.6317	83.2 7.7 0.6411 0.6513	80.7 8.8 0.5910 0.6012	76.2 7.1 0.5012 0.5097
musk2	84.6 6598	86.0 0.5 0.2076 0.3349			97.1 p 0.6 0.5337 0.8608	96.1 0.7 0.5043 0.8133	97.1 0.7 0.5346 0.8623	***97.3*** ***0.6*** ***0.5394*** ***0.8699***	96.1 0.6 0.5052 0.8147
netpho	14.8 20005	33.3 1.0 1.4492 0.3057 9991	72.4 1.6 3.2402 0.6835 9991	71.4 1.9 3.2147 0.6781 9991	85.0 r 0.5 3.9779 0.8383	***85.5*** ***0.7*** ***3.9914*** ***0.8412***	84.7 0.5 3.9627 0.8351	70.5 0.8 3.2469 0.6842	70.4 0.7 3.2430 0.6834
netstr	35.3 20005	36.1 1.2 0.3250 0.1571	***82.2*** ***1.4*** ***1.5512*** ***0.7497***	74.0 1.2 1.3109 0.6336 10001	80.2 p 0.4 1.4760 0.7133	82.2 1.0 1.5443 0.7463	80.2 1.1 1.4806 0.7156	77.6 0.7 1.4038 0.6784	77.0 0.7 1.3881 0.6708
nettyp	99.3 20005	99.3 0.0 0.0101 0.1454	99.2 0.1 0.0044 0.0648	99.3 0.0 0.0101 0.1454	***99.3*** p ***0.0*** ***0.0104*** ***0.1504***	***99.3*** ***0.0*** ***0.0104*** ***0.1504***	99.0 0.2 -0.0069 -0.1007	***99.3*** ***0.0*** ***0.0104*** ***0.1504***	***99.3*** ***0.0*** ***0.0104*** ***0.1504***
newthy	69.8 215	95.4 3.8 1.0351 0.8919	93.0 5.8 0.9720 0.8345	94.9 5.0 1.0097 0.8699	93.5 4.5 0.9832 0.8490	94.4 4.3 1.0035 0.8657	93.9 4.4 0.9822 0.8426	94.0 4.4 0.9870 0.8464	93.5 3.3 0.9793 0.8400
pima	65.1 768	67.0 2.9 0.2280 0.2444	65.1 1.1 0.1866 0.2002	65.1 1.1 0.1866 0.2002	72.7 p 6.6 0.3483 0.3734	73.6 7.1 0.3678 0.3943	71.7 6.6 0.3284 0.3515	75.5 6.3 0.4092 0.4383	74.3 5.3 0.3836 0.4107

Test Results Table 5.1

Name	ib1	ib2	ib3	ib4	bayes	cart	per	bp	Average
lymph	83.1 11.0 0.7332 0.6493	75.0 9.8 0.5703 0.4987	75.2 9.8 0.5752 0.4926	71.0 8.4 0.5037 0.4380	74.4 6.0 0.5299 0.4620	77.6 7.4 0.5979 0.5401	75.8 9.4	82.4 11.4	77.3 3.6 0.6059 0.5353
mache	83.3 5.7 0.5101 0.4532	78.5 6.5 0.3874 0.3389	75.2 13.4 0.2910 0.2665	79.8 8.1 0.4641 0.4202	83.3 5.6 0.5314 0.4570	82.3 4.7 0.4873 0.4333	81.3 5.8	82.8 5.2	81.7 2.3 0.4755 0.4218
machp	83.3 5.7 0.5101 0.4532	78.9 6.1 0.4087 0.3555	75.7 13.7 0.3138 0.2814	80.3 7.6 0.4853 0.4368	82.8 5.1 0.5146 0.4442	82.3 4.7 0.4873 0.4333	81.3 5.8	***84.2*** ***3.2***	81.8 2.1 0.4766 0.4225
monk1	99.5 1.2 0.9880 0.9893	87.6 5.1 0.7499 0.7509	71.4 11.1 0.4271 0.4275	76.1 8.0 0.5201 0.5208	90.5 5.7 0.8083 0.8095	93.2 7.0 0.8622 0.8634	72.3 7.5	***100.0*** ***0.0***	90.4 10.4 0.8183 0.8194
monk2	76.2 7.0 0.4157 0.4493	67.4 5.9 0.2259 0.2444	57.9 7.0 0.0217 0.0240	55.6 8.6 -0.0283 -0.0302	79.0 5.1 0.4767 0.5153	93.2 5.3 0.7793 0.8402	61.6 5.7	***100.0*** ***0.0***	74.0 13.4 0.3468 0.3746
monk3	96.6 2.6 0.9293 0.9310	90.6 4.2 0.8100 0.8115	89.9 3.9 0.7957 0.7972	89.7 8.0 0.7924 0.7937	98.7 1.5 0.9727 0.9745	***98.9*** ***1.5*** ***0.9764*** ***0.9782***	98.2 2.8	98.7 1.9	96.9 3.3 0.9310 0.9327
musk1					***83.4*** ***6.8*** ***0.6462*** ***0.6571***	75.0 8.4 0.4764 0.4852	80.5 5.7	*58.0* *9.2*	79.3 8.9 0.6142 0.6248
musk2					97.1 0.7 0.5350 0.8628	96.4 0.5 0.5123 0.8262			95.4 3.6 0.4840 0.7806
netpho	77.2 0.6 3.5612 0.7505	73.6 0.8 3.4047 0.7175	72.8 0.9 3.3693 0.7101	57.2 3.1 2.5822 0.5442	81.1 0.8 3.8131 0.8036	85.5 0.8 3.9861 0.8400	54.9	*11.5* *5.6*	68.8 20.1 3.3602 0.7082
netstr	81.6 1.4 1.5334 0.7411	77.0 1.2 1.4232 0.6878	76.5 1.3 1.4143 0.6835	57.1 4.1 0.9136 0.4415	79.5 0.6 1.4491 0.7003	82.2 0.8 1.5387 0.7436	61.3 3.0	*39.4* *19.7*	72.2 14.8 1.3394 0.6473
nettyp	99.2 0.1 0.0015 0.0185	96.0 0.5 -0.2232 -3.2212	94.0 1.6 -0.3624 -5.2204	70.8 24.7 -2.0170 -29.3683	99.2 0.1 0.0066 0.0940	***99.3*** ***0.0*** ***0.0104*** ***0.1504***	99.1 0.2	***99.3*** ***0.0***	96.9 6.9 -0.1803 -2.6208
newthy	***96.7*** ***3.8*** ***1.0715*** ***0.9269***	94.0 3.7 1.0082 0.8721	91.2 6.6 0.9327 0.8033	92.5 4.6 0.9642 0.8326	92.5 4.5 0.9469 0.8142	89.3 7.1 0.8701 0.7511	95.8 4.7	96.7 4.5	93.8 1.9 0.9818 0.8457
pima	70.4 6.2 0.3014 0.3236	63.9 4.6 0.1613 0.1725	71.7 5.0 0.3281 0.3515	70.6 5.6 0.3035 0.3253	72.2 6.9 0.3394 0.3632	74.7 5.4 0.3929 0.4213	74.6 5.0	***75.8*** ***6.2***	71.2 3.8 0.3047 0.3265

Test Results Table 5.1

Name	DA	c45 rule	ocn2	ucn2	c45 tree	c4	id3	mml	smml
postop	71.1 90	70.0 5.4 0.1885 0.2010	50.0 15.1 -0.2616 -0.2876	60.0 9.4 -0.0405 -0.0677	***71.1*** p ***5.7*** ***0.2167*** ***0.2378***	70.0 7.5 0.1925 0.2116	50.0 16.0 -0.2656 -0.3120	***71.1*** ***5.7*** ***0.2167*** ***0.2378***	***71.1*** ***5.7*** ***0.2167*** ***0.2378***
promot	50.0 106	81.9 21.5 0.5187 0.4970	87.8 4.2 0.6784 0.7208	73.4 12.0 0.3567 0.3486	77.3 r 18.8 0.4259 0.4069	76.3 16.9 0.4043 0.3832	74.5 18.1 0.3710 0.3517	79.1 22.8 0.4569 0.4290	69.5 16.6 0.2699 0.2517
protei	54.5 21625	54.5 0.0 0.3048 0.2110 10810	54.9 1.1 0.3949 0.2733 10812	55.1 1.7 0.3479 0.2408 5404	56.4 p 0.7 0.3869 0.2678	52.7 1.3 0.3552 0.2458	53.8 1.1 0.3829 0.2649	56.0 0.6 0.3660 0.2533	55.4 0.8 0.3414 0.2362
ptumor	24.8 339	43.7 3.3 1.2529 0.3606	40.4 7.8 1.1437 0.3301	42.8 10.2 1.2053 0.3482	***44.0*** p ***6.6*** ***1.2613*** ***0.3640***	41.3 4.7 1.1906 0.3433	36.9 7.6 1.0597 0.3058	39.0 5.9 1.0499 0.3025	35.4 6.6 0.9397 0.2703
segm	14.3 2310	87.9 5.4 2.4401 0.8692	94.1 1.1 2.6290 0.9365	93.4 2.1 2.6080 0.9290	96.7 p 0.8 2.7077 0.9645	96.7 1.3 2.7090 0.9649	96.5 1.2 2.6998 0.9617	96.9 1.2 2.7142 0.9668	95.1 1.6 2.6591 0.9472
servo	31.7 167	49.1 12.0 0.8618 0.3567	46.0 10.9 0.7580 0.3177	42.4 12.3 0.6410 0.2688	51.5 r 9.8 0.8769 0.3645	***56.2*** ***10.3*** ***1.0515*** ***0.4441***	47.9 7.3 0.7826 0.3266	34.5 12.7 0.4707 0.1975	35.8 12.2 0.5202 0.2156
sick	93.9 3772	94.2 0.9 0.0942 0.2837	93.6 0.3 0.0669 0.2014	93.9 0.1 0.0800 0.2408	98.8 p 0.6 0.2832 0.8523	98.4 0.6 0.2679 0.8059	98.8 0.6 0.2832 0.8518	98.9 0.6 0.2876 0.8653	98.6 0.6 0.2734 0.8227
sickeu	90.7 3163	95.0 1.8 0.2656 0.5967	90.5 0.5 0.1041 0.2342	90.7 0.3 0.1131 0.2542	97.8 p 0.7 0.3684 0.8278	97.8 0.7 0.3685 0.8280	97.6 0.7 0.3583 0.8051	97.8 0.7 0.3673 0.8256	***97.9*** ***0.8*** ***0.3696*** ***0.8309***
slc[5]	60.0 15	40.0 14.9 -0.3836 -0.4178	***66.7*** ***23.6*** ***0.1950*** ***0.2123***	53.3 29.8 -0.0943 -0.1027	60.0 p 14.9 0.0503 0.0548	60.0 14.9 0.0503 0.0548	60.0 27.9 0.0503 0.0548	60.0 14.9 0.0503 0.0548	60.0 14.9 0.0503 0.0548
sonar	53.4 208	70.1 12.9 0.3911 0.3932	55.4 8.6 0.0965 0.0978	55.3 5.3 0.0943 0.0947	73.0 p 12.7 0.4495 0.4519	71.6 8.9 0.4207 0.4227	74.0 10.3 0.4711 0.4746	72.6 10.9 0.4432 0.4464	67.8 9.9 0.3450 0.3471
soyf[5]	36.2 47	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	98.0 4.5 1.6300 0.9772	98.0 4.5 1.6300 0.9772	80.7 27.7 1.1991 0.7358	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	83.3 23.6 1.3405 0.7818
soyl	13.5 683	88.7 7.3 3.3564 0.8949	92.0 3.3 3.4735 0.9261	91.1 3.5 3.4195 0.9116	90.1 p 2.5 3.3778 0.9004	90.8 3.2 3.4172 0.9106	88.3 4.5 3.3154 0.8835	90.5 3.1 3.3746 0.8993	86.2 3.8 3.1679 0.8445
soys[5]	36.2 47	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	98.0 4.5 1.6300 0.9772	98.0 4.5 1.6300 0.9772	80.7 27.7 1.1991 0.7358	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	83.3 23.6 1.3405 0.7818

Test Results Table 5.1

Name	ib1	ib2	ib3	ib4	bayes	cart	per	bp	Average
postop	58.9 13.9 -0.0606 -0.0564	56.7 11.1 -0.1089 -0.1162	52.2 21.7 -0.2335 -0.3142	42.2 25.0 -0.4344 -0.4834	58.9 21.0 -0.0647 -0.1151	***71.1*** ***5.7*** ***0.2167*** ***0.2378***	57.8 19.5	***71.1*** ***5.7***	61.4 9.3 -0.0159 -0.0278
promot	82.1 10.2 0.5488 0.5686	72.3 14.7 0.3412 0.3455	77.2 16.2 0.4417 0.4534	79.2 11.8 0.4984 0.5323	78.2 19.9 0.4435 0.4235	65.7 13.7 0.1924 0.1730	75.9 20.2	***87.9*** ***9.8***	77.4 5.7 0.4248 0.4204
protei	***63.7*** ***1.3*** ***0.6035*** ***0.4177***	52.4 1.4 0.3604 0.2494 10812	52.7 1.5 0.3664 0.2536	45.3 6.2 0.2152 0.1490	56.4 0.8 0.4104 0.2840	57.1 1.0 0.4057 0.2808	45.6 3.2	54.5 0.1	54.2 4.2 0.3744 0.2591
ptumor	39.3 5.5 1.1493 0.3312	34.3 9.7 0.9983 0.2886	38.6 6.6 1.1200 0.3230	38.7 11.4 1.1588 0.3341	39.9 7.0 1.1677 0.3367	41.3 4.1 1.1665 0.3356	40.4 6.0	*35.7* *10.1*	39.5 2.8 1.1331 0.3267
segm	97.1 1.2 2.7182 0.9682	95.1 1.9 2.6591 0.9472	94.0 1.5 2.6263 0.9355	94.4 1.5 2.6369 0.9393	***97.1*** ***1.1*** ***2.7208*** ***0.9691***	95.4 1.4 2.6683 0.9505	92.6 1.1	*51.6* *21.1*	92.2 10.7 2.6569 0.9464
servo	50.8 11.3 0.8901 0.3718	46.6 14.3 0.7865 0.3330	39.4 10.6 0.5798 0.2422	39.5 12.9 0.6221 0.2616	45.4 13.0 0.7149 0.3037	52.6 15.5 0.9649 0.4076	52.2 13.4	54.1 11.8	46.5 6.4 0.7515 0.3151
sick	95.8 1.0 0.1608 0.4838	93.2 1.5 0.0506 0.1524	91.8 1.4 -0.0063 -0.0190	90.3 1.6 -0.0685 -0.2062	***99.0*** ***0.6*** ***0.2898*** ***0.8722***	98.8 0.8 0.2810 0.8455	96.6 0.8	*95.3* *1.9*	96.0 2.8 0.1674 0.5038
sickeu	92.8 1.2 0.1879 0.4228	88.4 1.9 0.0297 0.0674	88.9 2.1 0.0475 0.1063	86.6 2.0 -0.0336 -0.0754	97.5 0.6 0.3571 0.8024	97.8 0.8 0.3674 0.8258	93.8 3.3	*93.1* *3.0*	94.0 3.8 0.2336 0.5251
slc[5]	53.3 18.3 -0.0943 -0.1027	***66.7*** ***23.6*** ***0.1950*** ***0.2123***	53.3 18.3 -0.0943 -0.1027	40.0 14.9 -0.3836 -0.4178	53.3 18.3 -0.0943 -0.1027	60.0 14.9 0.0503 0.0548	60.0 14.9	60.0 14.9	56.7 7.5 -0.0323 -0.0352
sonar	***86.5*** ***8.0*** ***0.7206*** ***0.7245***	85.0 8.0 0.6906 0.6940	71.1 9.2 0.4108 0.4125	71.1 6.9 0.4126 0.4152	73.1 11.3 0.4530 0.4564	70.7 7.0 0.4027 0.4050	73.2 6.2	76.4 11.9	71.7 7.8 0.4144 0.4169
soyf[5]	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	97.8 5.0 1.6188 0.9632	97.8 5.0 1.6302 0.9756	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0***	***100.0*** ***0.0***	96.9 5.7 1.5938 0.9530
soyl	90.3 4.1 3.4017 0.9067	88.7 3.0 3.3359 0.8892	88.6 3.3 3.3422 0.8906	89.8 4.2 3.4103 0.9089	***92.1*** ***3.5*** ***3.4683*** ***0.9243***	87.4 4.6 3.2587 0.8689	89.2 6.1	*60.3* *22.9*	87.8 7.3 3.3657 0.8971
soys[5]	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	97.8 5.0 1.6188 0.9632	97.8 5.0 1.6302 0.9756	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0***	***100.0*** ***0.0***	96.9 5.7 1.5938 0.9530

Test Results Table 5.1

Name	DA	c45 rule	ocn2	ucn2	c45 tree	c4	id3	mml	smml
soyso[5]	36.2 47	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	98.0 4.5 1.6300 0.9772	98.0 4.5 1.6300 0.9772	80.7 27.7 1.1991 0.7358	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	83.3 23.6 1.3405 0.7818
splice	51.9 3190	72.3 1.9 0.8073 0.5454	91.1 2.0 1.2779 0.8634	76.7 5.0 0.9473 0.6400	***94.4*** p ***1.2*** ***1.3560*** ***0.9161***	91.7 4.7 1.2857 0.8686	91.5 1.1 1.2879 0.8701	93.8 1.1 1.3417 0.9065	93.0 1.5 1.3240 0.8945
ssblow[5]	69.6 23	68.0 16.4 0.3774 0.3306	61.0 8.9 0.0835 0.0221	65.0 12.2 0.1893 0.1686	***78.0*** p ***17.9*** ***0.3774*** ***0.3306***	***78.0*** ***17.9*** ***0.3774*** ***0.3306***	65.0 18.7 0.1893 0.1933	***78.0*** ***17.9*** ***0.3774*** ***0.3306***	70.0 20.0 0.1893 0.1686
ssonly[5]	73.9 23	71.0 30.1 0.0609 0.0211	65.0 12.2 -0.0541 -0.0813	65.0 12.2 -0.0541 -0.0813	***83.0*** p ***9.7*** ***0.3782*** ***0.4606***	***83.0*** ***9.7*** ***0.3782*** ***0.4606***	70.0 9.4 0.0459 0.0187	***83.0*** ***9.7*** ***0.3782*** ***0.4606***	***83.0*** ***9.7*** ***0.3782*** ***0.4606***
staust	55.5 690	57.0 1.7 0.1226 0.1237	82.0 2.6 0.6284 0.6340	82.9 2.2 0.6459 0.6517	85.4 p 2.9 0.6957 0.7019	82.2 5.5 0.6313 0.6369	80.8 3.9 0.6021 0.6075	84.8 4.2 0.6840 0.6900	83.5 3.8 0.6577 0.6636
stgern	70.0 1000	70.7 1.3 0.2216 0.2515	67.4 3.5 0.1473 0.1671	72.1 1.9 0.2531 0.2872	73.5 p 3.1 0.2846 0.3230	72.4 2.5 0.2599 0.2949	71.5 5.2 0.2396 0.2719	72.3 5.8 0.2576 0.2923	72.3 5.4 0.2576 0.2923
stgers	70.0 1000	69.8 2.6 0.2013 0.2285	66.4 2.5 0.1248 0.1416	72.7 2.8 0.2666 0.3025	71.0 p 5.4 0.2284 0.2591	71.6 4.0 0.2419 0.2744	67.8 4.5 0.1563 0.1774	72.0 4.3 0.2509 0.2847	69.3 3.6 0.1901 0.2157
sthear	55.6 270	60.4 5.2 0.1914 0.1931	78.2 6.4 0.5501 0.5551	***82.6*** ***7.4*** ***0.6398*** ***0.6456***	73.4 p 5.7 0.4530 0.4570	76.7 6.5 0.5202 0.5249	77.1 8.3 0.5277 0.5324	81.9 6.9 0.6249 0.6305	76.3 7.8 0.5127 0.5174
stsati	23.8 6435	39.4 6.6 0.7511 0.3025	83.7 1.4 1.9956 0.8036	80.4 1.2 1.8891 0.7607	86.2 p 1.5 2.0708 0.8339	88.1 4.5 2.1263 0.8562	85.9 1.7 2.0635 0.8310	87.0 1.6 2.0880 0.8408	86.5 1.3 2.0758 0.8359
stsegm	14.3 2310	88.9 3.5 2.4703 0.8799	94.2 1.3 2.6316 0.9374	93.2 1.5 2.6014 0.9267	96.5 r 1.2 2.7011 0.9622	96.1 1.9 2.6893 0.9579	96.8 1.3 2.7116 0.9659	96.3 1.4 2.6959 0.9603	94.4 1.4 2.6395 0.9402
stshut	78.6 58000	100.0 0.0 0.9589 0.9986	98.7 0.2 0.8996 0.9391 14496	93.4 1.0 0.7138 0.7451 14496	***100.0*** ***0.0*** ***0.9591*** ***0.9988***	82.8 9.0 0.3751 0.3907	78.6 0.0 0.2314 0.2410	78.6 0.0 0.2314 0.2410	78.6 0.0 0.2314 0.2410
stvehi	25.8 846	59.5 9.0 1.0214 0.5115	67.0 5.1 1.2033 0.6025	64.2 4.9 1.1341 0.5679	71.2 r 3.7 1.3027 0.6523	69.4 3.6 1.2612 0.6315	74.1 4.2 1.3729 0.6875	72.7 4.0 1.3388 0.6703	70.5 2.7 1.2875 0.6447
thy387	73.8 9172	86.3 2.3 1.0674 0.5980	76.4 1.3 0.4803 0.2777 2281	75.9 0.5 0.4223 0.2394 4578	***95.4*** p ***0.8*** ***1.5571*** ***0.8726***	94.1 0.6 1.4745 0.8263	95.2 0.7 1.5399 0.8630	94.0 0.7 1.4719 0.8249	93.0 0.6 1.4121 0.7913

Test Results Table 5.1

Name	ib1	ib2	ib3	ib4	bayes	cart	per	bp	Average
soyso[5]	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	97.8 5.0 1.6188 0.9632	97.8 5.0 1.6302 0.9756	***100.0*** ***0.0*** ***1.6750*** ***1.0000***	98.0 4.5 1.6300 0.9772	***100.0*** ***0.0***	***100.0*** ***0.0***	96.9 5.7 1.5938 0.9530
splice	76.5 2.6 0.9867 0.6666	73.7 2.6 0.9108 0.6153	73.8 2.5 0.9139 0.6174	85.4 4.6 1.1579 0.7823	92.7 1.2 1.3109 0.8856	94.4 1.5 1.3547 0.9153	84.6 2.8	*80.0* *25.2*	85.3 8.2 1.1616 0.7848
ssblow[5]	61.0 8.9 0.0835 0.0221	49.0 28.8 -0.1635 -0.1581	34.0 24.1 -0.3635 -0.4549	36.0 15.2 -0.3750 -0.4758	73.0 19.2 0.3540 0.3382	70.0 20.0 0.1893 0.1686	77.0 17.9	78.0 22.8	65.1 13.8 0.1347 0.0939
ssonly[5]	74.0 8.2 0.1517 0.1652	65.0 12.2 -0.0748 -0.1301	32.0 28.0 -0.8944 -1.2067	48.0 21.7 -0.4921 -0.6696	66.0 16.4 -0.0599 -0.1278	73.0 13.0 0.1574 0.2117	73.0 19.2	78.0 14.4	69.5 13.2 0.0214 -0.0027
staust	81.0 3.2 0.6080 0.6134	74.2 3.7 0.4706 0.4749	83.2 5.7 0.6517 0.6575	84.5 3.9 0.6781 0.6842	83.1 4.6 0.6489 0.6547	***85.5*** ***4.6*** ***0.6986*** ***0.7049***	84.9 5.8	*84.5* *5.8*	81.2 6.8 0.6017 0.6071
stgern	67.8 2.6 0.1563 0.1774	63.2 4.4 0.0527 0.0598	63.6 7.5 0.0618 0.0700	64.0 7.7 0.0708 0.0803	70.4 4.5 0.2148 0.2438	73.9 4.6 0.2936 0.3332	***74.2*** ***4.7***	*71.6* *4.3*	70.1 3.6 0.1980 0.2246
stgers	72.1 2.7 0.2531 0.2872	62.5 5.7 0.0369 0.0420	65.7 6.3 0.1090 0.1237	64.1 3.8 0.0730 0.0828	69.4 3.9 0.1923 0.2182	***74.2*** ***3.6*** ***0.3004*** ***0.3408***	70.6 7.4	71.9 4.9	69.4 3.2 0.1875 0.2128
sthear	79.6 6.3 0.5800 0.5852	73.7 8.4 0.4604 0.4646	72.6 13.3 0.4380 0.4420	75.9 6.1 0.5053 0.5098	75.6 6.8 0.4978 0.5023	78.9 7.4 0.5651 0.5702	80.8 5.2	82.6 7.6	76.6 5.2 0.5047 0.5093
stsati	***90.5*** ***1.0*** ***2.2056*** ***0.8882***	86.7 0.9 2.0997 0.8456	87.3 1.2 2.1148 0.8516	87.1 1.0 2.1112 0.8502	87.0 1.3 2.0926 0.8427	86.4 1.4 2.0650 0.8316	76.6 5.8	*33.6* *25.8*	79.5 16.6 1.9821 0.7982
stsegm	***97.0*** ***0.8*** ***2.7168*** ***0.9678***	94.3 0.8 2.6329 0.9379	94.1 0.6 2.6303 0.9369	93.7 1.2 2.6172 0.9323	96.6 1.6 2.7051 0.9636	94.9 0.9 2.6539 0.9453	91.3 1.7	*51.9* *27.3*	91.9 10.5 2.6498 0.9439
stshut	99.9 0.0 0.9463 0.9878 14496	99.7 0.1 0.9433 0.9847 14496	99.3 0.7 0.9272 0.9678 14496	99.2 0.7 0.9259 0.9665 14 496	78.6 0.0 0.2314 0.2410	99.9 0.0 0.9517 0.9918 28997	94.2 1.3	*80.7* *3.1* *9997*	91.4 9.3 0.6805 0.7096
stvehi	70.2 4.7 1.2805 0.6412	64.7 3.8 1.1478 0.5747	65.5 3.4 1.1660 0.5838	64.9 5.1 1.1528 0.5772	72.7 4.1 1.3405 0.6712	69.8 5.0 1.2739 0.6379	***76.6*** ***5.5***	*33.1* *10.4*	66.6 9.6 1.2345 0.6182
thy387	81.6 0.9 0.9091 0.5094	76.4 0.9 0.7962 0.4462	72.7 1.3 0.7235 0.4054	70.2 1.4 0.6383 0.3576	94.8 0.7 1.5193 0.8514	93.6 0.6 1.4322 0.8026	79.3 3.9	*59.2* *30.8*	83.6 10.9 1.1032 0.6190

Test Results Table 5.1

Name	DA	c45 rule	ocn2	ucn2	c45 tree	c4	id3	mml	smml
tictac	65.3 958	65.4 0.9 0.1904 0.2046	98.0 2.6 0.8883 0.9544	98.5 1.3 0.8994 0.9662	85.6 r 2.4 0.6220 0.6682	90.1 4.5 0.7188 0.7729	83.9 3.2 0.5864 0.6301	86.5 2.5 0.6423 0.6901	79.7 3.1 0.4968 0.5338
trains[5]	50.0 10	70.0 44.7 0.4000 0.4000	60.0 41.8 0.2000 0.2000	70.0 27.4 0.4000 0.4000	80.0 27.4 0.6000 0.6000	60.0 41.8 0.2000 0.2000	50.0 0.0 0.0000 0.0000	50.0 35.4 0.0000 0.0000	50.0 35.4 0.0000 0.0000
votes	61.4 435	95.9 3.6 0.8746 0.9103	93.8 2.9 0.8320 0.8663	94.7 4.3 0.8507 0.8856	96.8 p 2.7 0.8938 0.9304	96.8 2.7 0.8938 0.9304	94.5 3.9 0.8456 0.8800	***97.3*** ***2.8*** ***0.9034*** ***0.9405***	95.6 2.8 0.8699 0.9055
vowel	9.1 528	75.9 6.1 2.5851 0.7496	58.1 6.3 1.9530 0.5663	60.6 10.3 2.0401 0.5915	78.8 p 6.6 2.6863 0.7790	79.0 5.1 2.6930 0.7809	82.6 4.4 2.8249 0.8191	76.1 4.0 2.5943 0.7523	68.0 4.8 2.3015 0.6674
water	52.2 527	60.7 5.4 0.7129 0.3882	66.6 5.2 0.8611 0.4695	65.7 3.4 0.7887 0.4313	70.2 p 7.5 0.9961 0.5443	68.9 7.1 0.9660 0.5259	66.0 6.8 0.8786 0.4811	67.1 7.8 0.9211 0.5037	64.5 4.1 0.8578 0.4704
wave21	35.3 300	61.7 9.7 0.7515 0.4750	72.0 5.3 0.9751 0.6163	70.7 5.8 0.9422 0.5956	64.3 p 10.4 0.8087 0.5112	62.7 10.9 0.7731 0.4887	66.3 6.6 0.8518 0.5384	66.0 5.8 0.8432 0.5330	67.7 8.7 0.8758 0.5536
wave40	35.3 300	61.7 6.5 0.7501 0.4741	68.3 11.0 0.8967 0.5668	63.7 6.6 0.7898 0.4992	65.0 7.1 0.8211 0.5190	66.3 5.1 0.8499 0.5372	67.0 9.1 0.8663 0.5475	66.3 7.9 0.8521 0.5386	66.3 6.4 0.8492 0.5368
wine	39.9 178	94.4 6.9 1.4286 0.9215	90.9 4.9 1.3615 0.8771	91.0 2.9 1.3590 0.8762	93.3 6.8 1.4037 0.9057	92.7 6.4 1.3908 0.8975	93.8 7.1 1.4177 0.9143	95.0 4.9 1.4401 0.9288	94.4 5.9 1.4281 0.9212
yellow	60.0 20	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	***100.0*** ***0.0*** ***0.9245*** ***1.0000***
ys!as	56.2 16	61.7 16.3 0.0528 0.0635	66.7 33.4 0.1736 0.2123	86.7 29.8 0.6076 0.6849	61.7 16.3 0.0528 0.0635	61.7 16.3 0.0528 0.0635	56.7 43.5 -0.0679 -0.0853	55.0 20.1 -0.0918 -0.0940	55.0 20.1 -0.0918 -0.0940
zoo	41.1 90	92.2 5.4 1.9072 0.8794	96.7 5.4 2.0557 0.9504	95.6 5.7 2.0185 0.9306	93.3 5.7 1.9443 0.8974	93.3 5.7 1.9443 0.8974	***97.8*** ***4.7*** ***2.0928*** ***0.9657***	95.6 5.7 2.0185 0.9308	90.0 6.3 1.8330 0.8480
Average	52.4 2793.5	71.3 6.9 0.7233 0.4553 2385.1	74.7 5.8 0.7865 0.4815 2047.6	74.0 6.0 0.7868 0.4659 1674.1	***77.9*** ***6.5*** ***0.9228*** ***0.5773*** ***2793.5***	77.3 7.4 0.9042 0.5562 2793.5	76.2 6.5 0.8838 0.5321 2793.5	76.7 6.1 0.8839 0.5523 2793.5	74.6 6.7 0.8266 0.5129 2793.5

Test Results Table 5.1

Name	ib1	ib2	ib3	ib4	bayes	cart	per	bp	Average
tictac	***98.9*** ***0.6*** ***0.9062*** ***0.9736***	94.4 2.6 0.8098 0.8699	85.1 5.7 0.6105 0.6557	70.7 4.9 0.3022 0.3244	85.3 2.1 0.6153 0.6611	92.0 4.8 0.7591 0.8160	97.0 2.4	98.9 0.9	88.1 9.7 0.6463 0.6944
trains[5]	60.0 41.8 0.2000 0.2000	50.0 35.4 0.0000 0.0000	60.0 41.8 0.2000 0.2000	60.0 41.8 0.2000 0.2000	50.0 35.4 0.0000 0.0000	***90.0*** ***22.4*** ***0.8000*** ***0.8000***	80.0 27.4	80.0 27.4	63.7 12.7 0.2286 0.2286
votes	92.4 4.2 0.8032 0.8364	91.2 3.8 0.7781 0.8094	90.6 4.0 0.7639 0.7948	92.4 3.1 0.8023 0.8348	95.9 3.9 0.8745 0.9100	96.3 2.9 0.8845 0.9209	94.5 4.3	95.0 4.3	94.6 2.0 0.8479 0.8825
vowel	***99.2*** ***1.3*** ***3.4217*** ***0.9922***	94.5 3.7 3.2511 0.9427	92.4 4.2 3.1779 0.9215	92.4 4.3 3.1783 0.9216	79.3 3.2 2.7091 0.7856	69.7 7.3 2.3641 0.6855	50.8 7.9	*48.3* *23.9*	75.4 14.9 2.6986 0.7825
water	67.2 5.2 0.9122 0.4966	61.5 5.0 0.7923 0.4312	64.5 4.0 0.8874 0.4849	61.9 5.7 0.8341 0.4553	***70.4*** ***6.5*** ***0.9960*** ***0.5427***	68.1 6.7 0.9209 0.5055	65.1 6.1	*52.1* *1.6*	65.0 4.3 0.8804 0.4808
wave21	69.7 9.2 0.9243 0.5843	66.3 9.8 0.8518 0.5384	72.0 9.2 0.9753 0.6164	73.7 4.3 1.0113 0.6392	68.7 7.4 0.9025 0.5705	70.7 8.4 0.9456 0.5977	76.3 7.9	***83.0*** ***7.4***	69.5 5.2 0.8880 0.5613
wave40	69.7 6.4 0.9261 0.5853	64.3 9.0 0.8091 0.5114	67.3 10.9 0.8729 0.5517	68.3 8.2 0.8958 0.5662	66.3 8.1 0.8525 0.5389	70.0 7.0 0.9309 0.5884	70.6 10.9	***79.7*** ***4.3***	67.5 3.9 0.8545 0.5401
wine	94.9 3.2 1.4452 0.9322	93.2 6.4 1.4096 0.9086	91.5 6.7 1.3717 0.8838	92.7 7.4 1.3896 0.8963	94.4 5.9 1.4289 0.9216	87.6 9.1 1.2867 0.8285	***98.3*** ***2.8***	***98.3*** ***2.8***	93.5 2.6 1.3972 0.9010
yellow	95.0 11.2 0.8038 0.8512	85.0 33.5 0.5623 0.5535	45.0 11.2 -0.2792 -0.3442	65.0 33.5 0.1830 0.2023	***100.0*** ***0.0*** ***0.9245*** ***1.0000***	80.0 27.4 0.4415 0.4046	***100.0*** ***0.0***	***100.0*** ***0.0***	91.9 15.5 0.7166 0.7620
ys!as	61.7 28.6 0.0528 0.0635	66.7 33.4 0.1736 0.2123	33.3 23.6 -0.5987 -0.6981	43.3 14.9 -0.3572 -0.4004	56.7 43.5 -0.0679 -0.0853	48.3 20.8 -0.2365 -0.2516	65.0 26.6	***90.0*** ***21.1***	60.6 13.5 -0.0247 -0.0247
zoo	96.7 5.4 2.0557 0.9504	95.6 7.8 2.0185 0.9350	94.5 7.8 1.9814 0.9188	91.1 8.8 1.8848 0.8718	***97.8*** ***4.7*** ***2.0928*** ***0.9657***	82.2 5.7 1.5732 0.7245	96.7 5.4	*95.6* *5.7*	94.0 3.8 1.9586 0.9047
Average	77.3 6.5 0.9285 0.5669 2250	73.1 7.5 0.8401 0.5374 2133	69.2 8.8 0.6878 0.2197 2351.5	69.2/69.2 8.8/8.6 0.6788/0.7063 -.0008/0.2989 2351.5/2171.4	76.5 6.9 0.8930 0.5432 2793.5	77.4 7.4 0.9073 0.5624 2509.1	75.2 6.9 2215.3	***76.1/80.7*** ***8.3/6.8*** ***2287.8/1510.***	

6. Limitations

The accuracy of a learning model used in this study may be lower than its maximal because the parameters of the model were not adjusted for best performance. Time limitations simply did not permit careful tuning of model parameters for each learning domain. In some cases, these parameters may affect a model's performance significantly.

For example, Holte [64] conducted a survey of different learning models in many different case studies. Taking the c4 learning model as an example, his survey shows that its performance varies from 66.9% to 72.1% on *breast*, from 63.2% to 65.5% on *glass*, and from 73.6% to 75.4% on *heartc*. Some of the performance differences may be accounted for by differences in the model parameters. Indeed, Holte's survey indicates, again using c4 as an example, that pruning the decision tree generated by c4 does affect its performance as one might expect. In addition, the survey shows that a model's performance results are often given without also reporting the values of the parameters used to obtain the results. Prechelt also comments on this fact [97].

Results may likewise differ from other studies because of differences in training/test set composition and, for those models which are sensitive to the order of presentation, because of differences in presentation order. Unless the training/test sets used in a case study have been archived or otherwise preserved for use in other case studies, these differences cannot be eliminated. However, see appendix B which describes *xlate*, a tool built expressly to eliminate these differences.

7. Conclusion

We have presented extensive performance results for 16 different machine learning, statistical, and connectionist inductive learners. Because of the complex nature of inductive learning, the current state of knowledge about the learning models, and the inductive learning problem domains, the simplest way to characterize a learning model is to compare its performance in different domains by conducting a case study [8, 76, 108]. Similarly, because inductive learners are too complex and too heterogeneous to lend themselves to analytical comparison, the simplest, and frequently only, way to compare the models is with a case study.

As noted above, it is often difficult to compare the results of one case study with another. This thesis has enlarged the scope of the typical case study so that such comparison is easier. It has used performance measures proposed by Kononenko and Bratko [74] that facilitate the comparison of one algorithm to others in the same problem domain and that enable the comparison of an algorithm across problem domains.

This work can be extended in several directions. Since the tools to manage the tests have already been built, results for other models can easily be added simply by acquiring and running the software with which they have been implemented. Additional databases can also be added. CMU's Statlib has, at last count, more than 45 databases that could be used in studies such as this. Likewise, the National Climatic Data Center, the National Geophysical Data Center, the National Oceanographic Data Center, and others have data that can be used to test supervised inductive learners.

Given the large scope of a study such as this, there is some difficulty in tuning learning

model parameters for best performance. One solution is to use techniques from genetic programming or evolutionary computing to optimize these parameters [55, 75].

Others have made efforts similar to this one. Zheng [138], using different techniques and fewer databases, has constructed a benchmark for machine learning algorithms. Prechelt [97] proposed rules for benchmarking neural networks and has made available 13 datasets in a format suitable for connectionist algorithms. The Statlog project, described in [88], combined the efforts of many researchers in the field of inductive learning to produce a case study similar in nature to this one. This project evaluated 22 machine learning, statistical, and connectionist algorithms on 21 different datasets (11 of which are also used in this study). While the breadth of the Statlog project may be somewhat less than that of the present study, its depth is commendable and to be recommended to anyone interested in inductive learning.

Appendix A: Standard Database Format (SDF)

Databases were collected from several sites, principally ics.uci.edu [92] and ftp.cs.cmu.edu [38]. Machine learning researchers from time to time deposit databases at ics.uci.edu since it seems to be oriented toward the machine learning community. The collection at ftp.cs.cmu.edu, with primarily a connectionist flavor, has not changed for some time. As a consequence of collecting databases from both sites, the databases were in many different formats. Furthermore, each machine learning/connectionist software package requires data to be formatted in a different way. To make use of the databases with a common set of software tools easier and to simplify the task of ensuring that the training and test sets derived from a database were identical for all learning models expect for differences in format, all databases were all translated to a common format. Others, with perhaps somewhat different objectives, have made similar, but more limited, efforts [97].

The standard database format (SDF) was designed with two goals in mind: (1) The format must be similar to the formats used in available software and (2) The format must be easily readable by humans and by machine.

In the SDF format, there is no distinction made between *class* and *attribute*. It was felt that this distinction is not necessary in creating a standard format since, at this level, *class* may simply be considered as another attribute-value pair. For the remainder of this appendix, attribute refers to both *class* and *attribute*. For the purposes of this study, attributes come in two vanities, *discrete* and *continuous*. Sometimes a third type, *ordered*, is added, but since none of the learning models in this study use *ordered* attributes, and also since none of the databases make a distinction *ordered* and *discrete* or between *ordered* and *continuous*, the

type *ordered* was ignored.

Attributes of type *ignore* are, as one might guess, not to be used in training or testing a learning algorithm, but since such attributes may have some value as labels and since they are found in some of the databases, they are included in the SDF format for completeness.

To explain some of the less obvious features about SDF files:

- No comments are allowed in SDF files. (This feature may be added at some later date.)
- Only the characters ';', ',', ':', '\t', '\n', and ' ' are reserved and even these may be used within a quoted STRING.
- Attributes must be defined on a single line.
- All values of an instance must be on a single line.

To simplify the task of translating a database to SDF format, an extended format verifier, *verify*, was built. It is described in Appendix B.

BNF Description of Standard Data Format (SDF) Files

```
SDFFile          ::= fileDescription fileBody
fileDescription  ::= numberOfAttrs numberOfInstances '\n'
numberOfAttrs    ::= UNSIGNED
numberOfInstances ::= UNSIGNED
fileBody         ::= descriptions instances
descriptions     ::= description | descriptions description
description      ::= attrName ':' attrValues ';' '\n'
attrName         ::= STRING
attrValues       ::= attrValue | attrValues ',' attrValue
attrValue        ::= STRING | "continuous" | "ignore"
instances        ::= instance | instances instance
instance         ::= values ';' '\n'
values           ::= value | values ',' value
value            ::= FLOAT | SIGNED | UNSIGNED | STRING | '?'

FLOAT            ::= [-+]?((([0-9]+\.)|([0-9]*\.[0-9]+))([eE][-+]?[0-9]+)?)
SIGNED           ::= [-+]?[0-9]+
STRING           ::= \"[^"\n]*\" |
                     [-+!-)*<=>@^_{-~/A-z]{1} |
                     [-+!-)*<=>@^_{-~/0-9]*[A-z] |
                     [-+!-)*<=>@^_{-~/0-9A-z]*
UNKNOWN          ::= '?'
UNSIGNED         ::= [0-9]+
WHITE_SPACE      ::= [ \t]+
```

Figure A1

Appendix B: Tools

As aids for this study we built an extensive set of tools. They fall into 4 categories: Database tools (*verify*, *verify.ksh*, *dsstats*, *dsstats.ksh*, *xlate*), test tools (*test-ml*, *test-conn*, and *xval.<algorithm>*), result extraction tools (*average* and *summarize*), and a function library.

B.1 Database Tools

B.1.1 *verify*

verify is a C executable that uses flex, yacc, and the dsfunc library to check the correctness of an SDF file.

Usage is: verify <options>

-?	Show usage
-f <file name>	Name of data file; use - for stdin (required)
-h	Show usage

verify reads an SDF file, checks it for syntax errors and also for attribute value errors. If errors are found, the line number and nearest token to the error is written to stderr.

B.1.2 *verify.ksh*

verify.ksh is a ksh shell script that reads a file containing a list of SDF files and passes the SDF files to *verify*.

B.1.3 *dsstats*

dsstats is a C executable that computes statistics about an SDF file (Table 2.1 was built

using *dsstats* and *dsstats.ksh*).

Usage is: dsstats <options>

-?	Show usage
-a <algorithm>	File format for which statistics are to be computed bp | c45 | cn2 | ibl | ind | perceptron | sdf (default)
-[no]boolean	If a discrete attr has only 2 values or is marked BOOLEAN, treat it as a boolean attr; otherwise, treat it as a discrete attr (default boolean)
-f <file>	Name of file for which statistics are to be computed (required)
-help	Show usage
-type	Type of file for which statistics are to be computed (sdf (default) | test | train)
-[v]verbose <0|1>	Level of verboseness where n =0 | 1; -vverbose prints column headers (default -verbose 0)

B.1.4 *dsstats.ksh*

dsstats.ksh is a ksh shell script that reads a file containing a list of SDF files and passes the SDF files to *dsstats*.

Usage is

dsstats.ksh -f <file> -o <options> -s <file>

where

-f <file> is a required parameter that specifies a file containing a list of the databases for which statistics are to be collected

-o <options> is an optional parameter that specifies any options to be passed to dsstats (default <options> = -verbose 0)

-s <file> is an optional parameter that specifies the file in which statistics are collected (default <file> = Stats)

B.1.5 *xlate*

xlate is a C executable that creates test and training sets from SDF files in a format

appropriate for any of the software used in this study. The composition and order of the test and training sets are preserved regardless of the format provided that the sets are generated using the same random number seed.

xlate can optionally do a variety of other pre-processing operations, such as, normalization, discretization, balancing, attribute exclusion, elimination of instances with unknown values, and assigning values to instances of attributes with unknown values using equal width, equal frequency, or chi-merge methods.

```
Usage is: xlate <options>

    -?                          Show usage
    -a <algorithm>              Algorithm for which data are to be transformed
                                Values may be: bp | c45 | cn2 | hut | ibl | ind |pdp[+|-] | perceptron
                                | sdf
    -[no]balance                Try to balance the data sets created so that their class distributions
                                are approximately the same as the original data file/ (default
                                nobalance)
    -class <unsigned>           Designate the class attr if it is not the 1st or zeroth attr in the SDF
                                file (default attr 0)
    -dis <unsigned> <option>    Convert attr <unsigned> from continuous to
                                discrete; option is one of
                                chi [<double> [<unsigned> [<unsigned>]]]
                                Discretize attr using ChiMerge where
                                <double> is the minimum chi-square
                                probability (default  0.9)
                                <unsigned> is max intervals (default is number
                                of classes)
                                <unsigned> is min intervals (default 1)
                                frequency [<unsigned>]
                                Discretize attr using <unsigned> equal
                                frequency intervals (default is number of
                                classes)
                                interval [<unsigned>]
                                Discretize attr using <unsigned> equal
                                intervals (default is number of classes)
                                May be repeated or, to convert all attrs, use
                                ALL in place of <unsigned>
    -f <file name>              Name of data file (required)
    -help                       Show usage
    -[no]ignore                 [Don't] Ignore attrs marked IGNORE in raw data
                                file (default ignore)
    -normalize <unsigned> [<double> <double>]
                                Normalize attr <unsigned> using 1st <double>
                                as its minimum value and 2nd <double> as its
```

```
                        maximum value; may be repeated or to
                        normalize all attrs, use ALL in place of
                        <unsigned> (default min = 0 and max = 1.0)
-o <file name>          Name of output data file (default input
                        <file name>.<ext> where <ext> is the extension
                        appropriate for <algorithm>)
-percent <double>       Percent of instances used as training instances
                        when sets == 2 (default 67%)
-[no]randomize          [Don't] randomly pick instances (default
                        randomize)
-remove <unsigned>      Remove all instances for which attr <unsigned>
                        has unknown values; may be repeated or use
                        ALL to remove all unknown values for all attrs
-seed <unsigned> Seed for random number generator (optional)
-sets <unsigned>        Number of instance sets to be created (1 - 99);
                        if <unsigned> != 2, the instances are written to
                        <file name>.xval (default sets == 2)
-size <unsigned long> Number of instances in the SDF data set to be
                        used 1 - sizeof(data set) (default sizeof(data set))
-unk <unsigned> <option>Assign values to instances of attr <unsigned>
                        with unknown values where <option> is one of
                        assign <string>
                        Assigns value represented by <string> to
                        instances of attr with unknown value; if
                        attr is continuous, <string> is converted
                        to <double>
                        simple
                        For each class, assigns most frequent value of
                        attr <unsigned> for that class to instances with
                        unknown values
                        May be repeated or to assign values to all attrs,
                        use ALL in place of <unsigned>
-xclude <unsigned>      Exclude attr number <unsigned> in SDF data set;
                        may be repeated
```

B.2 Test Tools

B.2.1 *xval.<algorithm>*

xval.<algorithm> is a ksh shell script that does a cross-validation test using a .xval file produced by *xlate* for <algorithm>. There are 6 different *xval.<algorithm>* scripts, one each for back-propagation (*xval.bp*), c4.5 (*xval.c45)*, cn2 (*xval.cn2*), the instance-based learning software (*xval.ibl*), the IND decision tree software (*xval.ind*), the CN2 concept induction software (*xval.cn2*), and perceptron (*xval.per*).

B.2.1.1 *xval.bp*

For *xval.bp*

Usage is

xval.bp <options>

<options> are

-b <boolean> is an optional parameter specifying how to encode boolean attrs (<boolean> = 1|2)

-c <cross-validations> is a required parameter specifying the number of cross-validations to be done

-d <discrete> is an optional parameter specifying how to encode discrete attrs (<discrete> = distribute|encode|linear|log2)

-E <error> is an optional parameter specifying the maximum normalized error

-e <epochs> is an optional parameter specifying the maximum number of epochs to train the network

-f <file> is a required parameter which names the file stem

-h <hidden nodes> is an optional parameter specifying the number of hidden nodes in the network

-m <momentum> is an optional parameter specifying the momentum at which to train the network

-n is an optional parameter, which if given, nices the process used to evaluate the algorithm

-o is an optional parameter specifying other options

-R is an optional parameter which forces training instances to be randomly presented to the network

-r <learning rate> is an optional parameter specifying the learning rate at which to train the network

-S <verbose> is an optional parameter, which if given, calculates the statistics for the .test and .train sets (value 0 | 1)

-s <seed> is an optional parameter specifying a seed for the random number generator

-t is an optional parameter which forces the network to be tested even if the convergence criteria have not been fulfilled when training stops

-u <unknowns> is an optional parameter which specifies what to do with unknown attr values (<unknowns> = discard|node|value)

-V is an optional parameter which, if given, specifies the degree of verboseness

-v is an optional parameter which specifies the size of a validation set if one is to be used

-w <weight range> is an optional parameter specifying the weight range for each node in the network

-x is an optional parameter specifying whether or not to restart a previous cross-validation

B.2.1.2 *xval.c45*

For *xval.c45*

Usage is

xval.c45 <options>

<options> are

-c <cross-validations> is a required parameter specifying the number of cross-validations to be done

-e is an optional parameter, which if given, echoes some commands to stdout

-f <file> is a required parameter which names the file stem

-n is an optional parameter, which if given, nices the process used to evaluate the algorithm

-S <verbose> is an optional parameter, which if given, calculates the statistics for the .test and .train sets (value 0 | 1)

B.2.1.3 *xval.cn2*

For *xval.cn2*

Usage is

xval.cn2 <options>

<options> are

-c <cross-validations> is a required parameter specifying the number of cross-validations to be done

-e is an optional parameter, which if given, echoes some commands to stdout

-f <file> is a required parameter which names the file stem

-n is an optional parameter, which if given, nices the process used to evaluate the algorithm

-S <verbose> is an optional parameter, which if given, calculates the statistics for the .test and .train sets (value 0 | 1)

-s <style> is a required parameter specifying the cn2 style to be used (value ordered | unordered)

B.2.1.4 *xval.ibl*

For *xval.ibl*

Usage is

xval.ibl <options>

<options> are

-c <cross-validations> is a required parameter specifying the number of cross-validations to be done

-e is an optional parameter, which if given, echoes some commands to stdout

-f <file> is a required parameter which names the file stem

-n is an optional parameter, which if given, nices the process used to evaluate the algorithm

-S is an optional parameter, which if given, specifies a seed for ibl's random number generator

-s <style> is a required parameter specifying the learning style (ib1 | ib2 | ib3 | ib4)

-v <verbose> is an optional parameter, which if given, calculates the statistics for the .test and .train sets (value 0 | 1)

B.2.1.5 *xval.ind*

For *xval.ind*

Usage is

xval.ind <options>

<options> are

-c <cross-validations> is a required parameter specifying the number of cross-validations to be done

-e is an optional parameter, which if given, echoes some commands to stdout

-f <file> is a required parameter which names the file stem

-n is an optional parameter, which if given, nices the process used to evaluate the algorithm

-s <style> is a required parameter specifying the learning style (bayes | c4 | cart | dgraph | id3 | lgraph | mml | smml)

-v <verbose> is an optional parameter, which if given, calculates the statistics for the .test and .train sets (value 0 | 1)

B.2.1.6 *xval.per*

For *xval.per*

Usage is

xval.per <options>

<options> are

-b <boolean> is an optional parameter specifying how to encode boolean attrs (<boolean> = 1|2)

-c <cross-validations> is a required parameter specifying the number of cross-validations to be done

-d <discrete> is an optional parameter specifying how to encode discrete attrs (<discrete> = distribute|encode|linear|log2)

-E <error> is an optional parameter specifying the maximum normalized error

-e <epochs> is an optional parameter specifying the maximum number of epochs to train the network

-f <file> is a required parameter which names the file stem

-l <rule> is an optional parameter specifying the learning rule to use (value delta | hebbian)

-n is an optional parameter, which if given, nices the process used to evaluate the algorithm

-o s an optional parameter specifying other options

-R is an optional parameter which forces training instances to be randomly presented to the network

-r <learning rate> is an optional parameter specifying the learning rate at which to train the network

-s <seed> is an optional parameter specifying a seed for the random number generator

-t is an optional parameter which forces the network to be tested even if the convergence criteria have not been fulfilled when training stops

-u <unknowns> is an optional parameter which specifies what to do with unknown attr values (<unknowns> = discard|node|value)

-V is an optional parameter which, if given, specifies the degree of verboseness

-v is an optional parameter which specifies the size of a validation set if one is to be used

-w <weight range> is an optional parameter specifying the weight range for each node in the network

-x is an optional parameter specifying whether or not to restart a previous cross-validation

B.2.2 *test-ml*

test-ml is a ksh shell script that reads a file containing a list of SDF files and passes the files one at a time to the specified machine learning *xval.<algorithm>* script (either *xval.c45*, *xval.cn2*, *xval.ibl*, or *xval.ind*).

Usage is

test-ml -a <algo> -C [squeaky] -c <validations> -e -f <list> -h -n -o <options> -s

where

-a <algo> is a required parameter which specifies the machine-learning/connectionist algorithm to test (value c45 | cn2 | ibl | ind)

-C <clean> is an optional parameter which has the effect of removing intermediate files when the cross-validation is finished, for example, all *.test, *.train, *.attr, ... files (value inter | squeaky | immaculate)

-c <validations> is a required parameter specifying the number of cross-validations to be done (value 1 - 99)

-e is an optional parameter which has the effect of echoing some the of the commands to stdout

-h is an optional parameter which displays this message

-f <list> is a required parameter which specifies a text file containing the names of the SDF data files to be used

-n is an optional parameter which has the effect of 'niceing' the cross-validation

-o <algorithm specific options> is an optional parameter specifying any options to be passed to the algorithm; the options must be enclosed in double quotes ("")

-S is an optional parameter which specifies the seed to be used for ibl

-s is a required parameter for cn2 | ibl | ind which specifies the style of learning

-v <verbose> is an optional parameter, which if given, calculates the statistics for the .test

and .train sets (0 | 1)

B.2.3 *test-conn*

test-conn is a ksh shell script that reads a file containing a list of SDF files and passes the files one at a time to the specified connectionist *xval.<algorithm>* script (either *xval.bp* or *xval.per*).

Usage is

test-conn <options>

<options> are

-a <algo> is a required parameter which specifies the algorithm to be used (value bp | per)

-C <clean> is an optional parameter which has the effect of removing intermediate files when the cross-validation is finished, for example, all *.test, *.train, *.attr, ... files (value immaculate | inter | squeaky)

-c <validations> is an optional parameter which specifies the number of cross-validations to be done (default 10)

-e is an optional parameter which has the effect of echoing some the of the commands to stdout

-f <list> is a required parameter which specifies a file containing a list of sdf data files to be tested

-o <options> is an optional parameter specifying any other parameters to be passed to xval.<algo>. <options> must be enclosed in "".

-r is an optional parameter indicating whether or not the shell script should re-start a previous run

-X is an optional parameter indicating that the .xval file should be created even if it exists

B.3 Result Extraction Tools

B.3.1 *average*

average is C executable that reads *n* similar lines of text from stdin and scans them for numbers. It prints to stdout each of the lines read and 4 lines which (1) average the numbers which occur in the same relative position on the *n* lines of text, (2) show the minimum of the *n* numbers, (3) show the maximum of the *n* numbers, and (4) show the standard deviation of the *n* numbers. *average* is used to summarize the results from each of the cross-validation trials.

B.3.2 *summarize*

summarize is a ksh shell script that reads a file containing a list of train/test files then scans the .result files for each learning model used in this study and builds a table of results. The table is suitable for importing into a spread-sheet for further processing. The shell script will need to be modified depending on the user's specific needs.

Appendix C: Database Generators

The databases *cmupa7*, *cmupa8*, *cmusp*, *cmusp2*, *led17*, *led7*, *wave21*, and *wave40* were artificially generated. The programs which generate these databases are *parity*, *two-spirals*, *led-creator*, *led-creator-+17*, *waveform*, and *waveform-+noise*. The original C code for each of these generators is found at ftp.cs.cmu.edu [61] and ics.uci.edu [173] and was written by Matt White (parity), Matt White and Alexis Wieland (two-spirals), David Aha (led-creator, led-creator-+17, waveform, and waveform-+noise). The code was modified to produce SDF files and made more portable. The modified code is available at ftp://synapse.cs.byu.edu/pub/db/cmu, ftp://synapse.cs.byu.edu/pub/db/led-creator, and ftp://synapse.cs.byu.edu/pub/db/waveform respectively.

Appendix D. Learning Model Parameters

D.1 Decision Trees

The command line used for the c4.5 decision tree induction (*c4.5 tree*) trials was

```
c4.5 -f <file name> -u
```

Default values were used for all other parameters. The parameters are explained in depth in [105].

The command lines used for c4 decision tree induction (*c4*), classification and regression trees (*cart*), id3 decision trees (*id3*), minimum message length decision trees (*mml*), and strict minimum message length decision trees (*smml*) were

```
mktree -e -v -s c4 <file name>
mktree -e -v -s cart <file name>
mktree -e -v -s id3 <file name>
mktree -e -v -s mml <file name>
mktree -e -v -s smml <file name>
```

respectively. Default values were used for all other parameters. The parameters are explained in depth in [31].

D.2 Nearest Neighbor

The command lines used for the nearest neighbor learning models (*ib1*, *ib2*, *ib3*, and *ib4*)

```
ibl <description file> <names file> <train file> <test file> <output file>  97 -ib1
      -reportrate 10000 -testlast
ibl <description file> <names file> <train file> <test file> <output file>  97 -ib2
      -reportrate 10000 -testlast
ibl <description file> <names file> <train file> <test file> <output file>  97 -ib3
      -reportrate 10000 -testlast
ibl <description file> <names file> <train file> <test file> <output file>  97 -ib4
      -reportrate 10000 -testlast
```

respectively (97 is the seed for *ibl*'s random number generator). Default values were used for all other parameters. The parameters are explained in depth in [6].

D.3 Statistical

The command lines used for Bayesian induction (*bayes*) was

```
mktree -e -v -s bayes <file name>
```

Default values were used for all other parameters. The parameters are explained in depth in [31].

D.4 Rule Based

The command line used for the c4.5 rule induction (*c4.5 rule*) trials was

```
c4.5rules -f <file name> -u
```

Default values were used for all other parameters. The parameters are explained in depth in [105].

The following parameter values were used with cn2 ordered rule induction (*ocn2*) and cn2 unordered rule induction (*ucn2*)

```
algorithm ordered
    read atts <attribute file name>
    read examples <train file name>
    induce
    xecute all
    quit
    read examples <test file name>
    xecute all
    quit
    quit

algorithm unordered
    read atts <attribute file name>
    read examples <train file name>
    induce
    xecute all
    quit
    read examples <test file name>
    xecute all
    quit
    quit
```

Default values were used for all other parameters. The parameters are explained in depth in [23, 24].

D.5 Neural Networks

For this study, the perceptron always used the logistic activation function

$$\frac{1}{1-e^{-x}}$$

and the delta learning rule (cf. [56, 84, 93]). During training, weights were adjusted after every instance. The net was trained either until the sum of errors squared or the normalized error reached a specified value. The sum or errors squared is, as usual, defined to be

$$\sum_i \sum_p (o_{ip} - t_{ip})^2$$

while the normalized error is defined to be

$$\frac{o_{\max} - o_{\min}}{N \cdot P} \sum_i \sum_p (o_{ip} - t_{ip})^2$$

where $o_{\max}$ and $o_{\min}$ are the maximum and minimum of the activation function respectively, N is the number of output nodes, P is the number of instances in the training set, o_{ip} is the actual output value for the i^{th} output node and p^{th} instance, and t_{ip} is the target value for the i^{th} output node and p^{th} instance. (See [97]).

Boolean attributes are always encoded using a single input node. Discrete attributes are either encoded using one input node for every possible value of the attribute or are encoded using ceil ($\log_2$(<number of input values>)) input nodes where ceil(x) is the smallest integer

greater than *x*.

Any database with an instance of an attribute with an unknown value is allocated an extra input node for every attribute in the database as described in 4.4. Databases with no instances of unknown values are not allocated extra input nodes.

Perceptron Parameters Table D1

Name	network cfg	err	max epochs	weight range	learning rate	seed	attr encoding
adult!	4-1	0.0001	5000	0.10	0.10	0007	enc
adult-	4-1	0.0001	5000	0.10	0.10	0007	enc
agaric	85-1	0.0001	5000	0.10	0.10	0007	enc
ahyper	60-5	0.0001	5000	0.10	0.10	9973	enc
ahypo	60-5	0.0001	5000	0.10	0.10	9973	enc
allbp	60-5	0.0001	5000	fan	0.10	9973	enc
allrep	60-5	0.0001	5000	fan	0.10	0007	enc
ann	21-3	0.0001	5000	fan	0.10	9973	enc
anneal	95-6	0.0001	5000	0.10	0.10	9973	enc
audio	150-24	0.0001	5000	fan	0.10	9973	enc
auto	20-5	0.0001	5000	fan	0.10	9973	enc
balanc	12-3	0.0100	5000	fan	0.10	0007	enc
breast	31-1	0.0100	5000	0.10	0.10	9973	enc
breastw	45-1	0.0100	5000	fan	0.10	0007	enc
bridg1	31-7	0.0100	5000	fan	0.10	9973	enc
bridg2	31-7	0.0100	5000	fan	0.10	9973	enc
bupa	6-1	0.0001	5000	fan	0.10	9973	
cmupa7	7-1	0.0001	5000	fan	0.10	9973	enc
cmupa8	8-1	0.0001	5000	fan	0.10	9973	enc
cmupro	65-3	0.0100	5000	fan	0.10	0007	enc
cmuson	60-1	0.0001	5000	fan	0.10	9973	
cmusp	2-1	0.0001	5000	fan	0.10	9973	
cmusp2	2-1	0.0001	5000	fan	0.10	9973	
cmuvow	10-11	0.0001	5000	fan	0.10	9973	
crx	40-1	0.0001	5000	fan	0.10	9973	enc
dis	60-1	0.0001	5000	fan	0.10	9973	enc
echoc	18-1	0.0001	5000	fan	0.10	9973	enc
flag	55-8	0.0100	5000	fan	0.10	0007	enc
glass	9-7	0.0100	5000	fan	0.10	0007	
hayes	11-3	0.0001	5000	fan	0.10	9973	enc
heart	35-5	0.0001	5000	fan	0.10	9973	enc
heartc	35-5	0.0001	5000	fan	0.10	9973	enc
hearth	35-5	0.0001	5000	fan	0.10	9973	enc
hearts	35-5	0.0001	5000	fan	0.10	9973	enc

Perceptron Parameters Table D1

Name	network cfg	err	max epochs	weight range	learning rate	seed	attr encoding
heartv	35-5	0.0001	5000	fan	0.10	9973	enc
hepat	38-1	0.0001	5000	fan	0.10	9973	enc
horse	44-1	0.0100	5000	fan	0.10	9973	enc
house	13-5	0.0001	5000	fan	0.10	9973	enc
hypoth	50-1	0.0001	5000	fan	0.10	9973	enc
import	66-5	0.0001	5000	fan	0.10	9973	enc
iono	34-1	0.0001	5000	fan	0.10	9973	
iris	4-3	0.0001	5000	fan	0.10	9973	
isol5	617-26	0.0010	5000	fan	0.10	9973	
kinsh	10-12	0.0001	5000	fan	0.10	9973	enc
krvskp	37-1	0.0001	5000	fan	0.10	9973	enc
laborn	39-1	0.0100	5000	0.50	0.10	0007	enc
led17	24-10	0.0001	5000	fan	0.10	9973	enc
led7	7-10	0.0001	5000	fan	0.10	9973	enc
lense	5-3	0.0001	5000	fan	0.10	9973	enc
letter	80-26	0.0010	10000	fan	0.10	9973	enc
lrs	101-10	0.0001	5000	fan	0.10	9973	
lungc	224-3	0.0100	5000	0.99	0.20	9973	enc
lymph	36-4	0.0001	5000	fan	0.10	9973	enc
mache	12-10	0.0001	5000	fan	0.10	9973	enc
machp	12-10	0.0001	5000	fan	0.10	9973	enc
monk1	11-1	0.0001	5000	fan	0.10	9973	enc
monk2	11-1	0.0001	5000	fan	0.10	9973	enc
monk3	11-1	0.0001	5000	fan	0.10	9973	enc
musk1	166-1	0.0001	5000	fan	0.10	9973	
musk2							
netpho	35-52	0.0001	5000	fan	0.10	9973	enc
netstr	35-6	0.0001	5000	fan	0.10	9973	enc
nettyp	35-3	0.0001	5000	fan	0.10	9973	enc
newthy	5-3	0.0001	5000	fan	0.10	9973	enc
pima	8-1	0.0001	5000	fan	0.10	9973	
postop	16-3	0.0100	5000	fan	0.10	9973	enc
promot	171-1	0.0001	5000	fan	0.10	9973	enc
protei	65-3	0.0001	5000	fan	0.10	9973	enc

Perceptron Parameters Table D1

Name	network cfg	err	max epochs	weight range	learning rate	seed	attr encoding
ptumor	37-22	0.0100	5000	fan	0.10	9973	enc
segm	19-7	0.0001	5000	fan	0.10	9973	
servo	12-20	0.0001	5000	fan	0.10	9973	enc
sick	60-1	0.0001	5000	fan	0.10	9973	enc
sickeu	50-1	0.0001	5000	fan	0.10	9973	enc
slc	16-1	0.0001	5000	0.99	0.50	0007	enc
sonar	60-1	0.0001	5000	fan	0.10	9973	
soyf	61-4	0.0001	5000	fan	0.10	9973	enc
soyl	61-19	0.0100	5000	fan	0.10	9973	unk val
soys	61-4	0.0001	5000	fan	0.10	9973	enc
soyso	61-4	0.0001	5000	fan	0.10	9973	enc
splice	240-3	0.0001	5000	fan	0.10	9973	enc
ssblow	4-3	0.0001	5000	fan	0.10	9973	
ssonly	4-3	0.0001	5000	fan	0.90	9973	
staust	22-1	0.0001	5000	fan	0.10	9973	enc
stgern	24-1	0.0001	5000	fan	0.10	9973	
stgers	40-1	0.0001	5000	fan	0.10	9973	
sthear	20-1	0.0001	5000	fan	0.10	9973	enc
stsati	36-6	0.0001	5000	fan	0.10	9973	
stsegm	19-7	0.0001	5000	fan	0.10	9973	
stshut	9-7	0.0001	5000	fan	0.10	9973	
stvehi	18-4	0.0001	5000	fan	0.10	9973	
thy387	60-34	0.0001	5000	fan	0.10	9973	enc
tictac	18-1	0.0001	5000	fan	0.10	9973	enc
trains	67-1	0.0001	5000	fan	0.10	9973	emc
votes	32-1	0.0100	5000	fan	0.10	9973	enc
vowel	10-11	0.0001	5000	fan	0.10	9973	
water	38-13	0.0001	5000	fan	0.10	9973	
wave21	21-3	0.0001	5000	fan	0.10	9973	
wave40	40-3	0.0001	5000	fan	0.10	9973	
wine	13-3	0.0001	5000	fan	0.10	9973	
yellow	4-1	0.0001	5000	fan	0.10	9973	enc
ys!as	4-1	0.0001	5000	fan	0.10	9973	enc
zoo	16-7	0.0001	5000	fan	0.10	9973	enc

Back-Propagation Parameters Table D2

Name	network cfg	err	max epochs	weight range	learning rate	seed	attr encoding	momentum
adult!	4-2-1	0.0001	05000	fan	0.10	9973	enc	0.90
adult-	4-2-1	0.0001	05000	fan	0.10	9973	enc	0.90
agaric	143-32-1	0.0001	10000	fan	0.10	9973	dis	0.99
ahyper	60-3-5	0.0001	05000	fan	0.10	9973	enc	0.90
ahypo	60-2-5	0.0001	05000	fan	0.10	9973	enc	0.90
allbp	60-4-5	0.0001	05000	fan	0.10	9973	enc	0.90
allrep	60-2-5	0.0001	05000	fan	0.10	9973	enc	0.90
ann	21-3-3	0.0001	05000	fan	0.10	9973	enc	0.90
anneal	95-7-6	0.0001	05000	fan	0.10	9973	enc	0.10
audio	150-17-24	0.0001	05000	fan	0.10	9973	enc	0.50
auto	20-9-5	0.0001	05000	fan	0.10	9973	enc	0.10
balanc	12-2-3	0.0001	05000	fan	0.10	9973	enc	0.90
breast	31-2-1	0.0001	05000	fan	0.10	9973	enc	0.95
breastw	45-5-1	0.0001	05000	fan	0.90	9973	enc	0.95
bridg1	31-12-7	0.0001	05000	fan	0.10	9973	enc	0.50
bridg2	31-3-7	0.0001	05000	fan	0.10	9973	enc	0.50
bupa	6-13-1	0.0001	05000	fan	0.10	9973		0.90
cmupa7	7-8-1	0.0001	05000	fan	0.10	9973	enc	0.90
cmupa8	8-6-1	0.0001	05000	fan	0.10	9973	enc	0.90
cmupro	65-4-3	0.0001	05000	fan	0.90	9973	enc	0.99
cmuson	60-9-1	0.0001	05000	fan	0.10	9973		0.90
cmusp	2-4-2-1	0.0001	20000	fan	0.10	9973		0.99
cmusp2	2-8-5-1	0.0001	05000	fan	0.10	9973		0.50
cmuvow	10-13-4-11	0.0100	05000	00.90	0.20	9973		0.20
crx	40-4-1	0.0001	05000	fan	0.10	9973	enc	0.20
dis	60-3-1	0.0001	05000	fan	0.10	9973	enc	0.90
echoc	18-6-1	0.0001	05000	fan	0.20	9973	enc	0.90
flag	55-11-8	0.0001	05000	fan	0.10	9973	enc	0.90
glass	9-7-3-7	0.0100	05000	00.90	0.20	9973		0.20
hayes	11-10-3	0.0001	05000	fan	0.10	9973	enc	0.95
heart	35-17-5	0.0001	05000	fan	0.10	9973	enc	0.20
heartc	35-2-5	0.0001	05000	fan	0.10	9973	enc	0.10
hearth	35-7-5	0.0001	05000	fan	0.10	9973	enc	0.50
hearts	35-12-5	0.0001	05000	fan	0.90	9973	enc	0.20

Back-Propagation Parameters Table D2

Name	network cfg	err	max epochs	weight range	learning rate	seed	attr encoding	momentum
heartv	35-12-5	0.0001	05000	fan	0.10	9973	enc	0.20
hepat	38-5-1	0.0001	05000	fan	0.10	9973	enc	0.90
horse	44-10-1	0.0001	05000	fan	0.20	9973	enc	0.10
house	13-11-3-5	0.0100	05000	00.90	0.20	9973	dis	0.20
hypoth	50-5-1	0.0001	05000	fan	0.10	9973	enc	0.90
import	66-9-5	0.0001	05000	fan	0.10	9973	enc	0.50
iono	34-5-4-1	0.0100	05000	00.90	0.20	9973		0.20
iris	4-2-3	0.0001	05000	fan	0.10	9973		0.90
isol5								
kinsh	10-11-12	0.0001	05000	fan	0.10	9973	enc	0.95
krvskp	37-10-1	0.0001	05000	fan	0.10	9973	enc	0.95
laborn	48-5-4-1	0.0001	05000	fan	0.10	9973	dis	0.99
led17	24-17-10	0.0001	05000	fan	0.10	9973	enc	0.90
led7	7-11-10	0.0001	05000	fan	0.10	9973	enc	0.90
lense	5-2-3	0.0001	05000	00.99	0.90	9973	enc	0.10
letter	80-13-26	0.0001	05000	fan	0.10	9973	enc	0.90
lrs	101-11-3-10	0.0100	05000	00.90	0.20	9973		0.20
lungc	224-8-3	0.0001	05000	fan	0.10	9973	enc	0.90
lymph	36-7-4	0.0001	05000	fan	0.10	9973	enc	0.90
mache	12-8-10	0.0001	05000	fan	0.10	9973	enc	0.90
machp	12-13-10	0.0001	05000	fan	0.10	9973	enc	0.90
monk1	15-4-1	0.0010	05000	fan	0.10	9973	dis	0.20
monk2	15-3-1	0.0010	05000	fan	0.10	9973	dis	0.50
monk3	15-2-1	0.0001	05000	fan	0.10	9973	dis	0.50
musk1	166-8-1	0.0001	05000	fan	0.10	9973		0.90
musk2								
netpho	35-5-52	0.0001	05000	fan	0.10	9973	enc	0.90
netstr	35-8-6	0.0001	05000	fan	0.10	9973	enc	0.90
nettyp	35-2-3	0.0001	05000	fan	0.10	9973	enc	0.90
newthy	5-4-3	0.0001	05000	fan	0.10	9973	enc	0.90
pima	8-3-3-1	0.0100	05000	00.90	0.20	9973		0.20
postop	16-17-3	0.0001	05000	fan	0.90	9973	enc	0.99
promot	171-3-1	0.0001	05000	fan	0.10	9973	enc	0.90
protei	65-8-3	0.0001	05000	fan	0.90	9973	enc	0.99

Back-Propagation Parameters Table D2

Name	network cfg	err	max epochs	weight range	learning rate	seed	attr encoding	momentum
ptumor	37-8-22	0.0001	05000	fan	0.10	9973	enc	0.90
segm	19-17-7	0.0001	05000	fan	0.90	9973		0.90
servo	12-5-20	0.0001	05000	fan	0.10	9973	enc	0.90
sick	60-4-1	0.0001	05000	fan	0.10	9973	enc	0.90
sickeu	50-5-1	0.0001	05000	fan	0.10	9973	enc	0.90
slc	16-4-1	0.0001	05000	fan	0.10	9973	enc	0.95
sonar	60-5-1	0.0001	05000	fan	0.10	9973		0.90
soyf	61-2-4	0.0001	05000	00.90	0.90	9973	enc	0.10
soyl	61-8-19	0.0001	05000	fan	0.10	9973	enc	0.90
soys	61-2-4	0.0001	05000	fan	0.10	9973	enc	0.90
soyso	61-2-4	0.0001	05000	fan	0.10	9973	enc	0.90
splice	240-5-3	0.0001	05000	fan	0.10	9973	enc	0.90
ssblow	4-3-3	0.0001	10000	00.10	0.20	0007		0.99
ssonly	4-11-3	0.0001	05000	fan	0.10	9973		0.99
staust	39-10-1	0.0001	10000	fan	0.10	9973	dis	0.99
stgern	24-24-1	0.0001	10000	fan	0.10	9973		0.90
stgers	40-3-1	0.0001	10000	fan	0.10	9973		0.90
sthear	20-17-1	0.0001	05000	fan	0.10	9973	enc	0.90
stsati	36-5-6	0.0001	05000	fan	0.10	9973		0.90
stsegm	19-23-7	0.0001	20000	fan	0.10	9973		0.95
stshut	9-11-7	0.0001	20000	fan	0.10	9973		0.99
stvehi	18-2-4	0.0001	05000	fan	0.10	9973		0.90
thy387	60-17-34	0.0001	05000	fan	0.90	9973	enc	0.99
tictac	27-19-1	0.0001	10000	fan	0.10	0007	dis	0.50
trains	67-2-1	0.0001	05000	fan	0.10	9973	enc	0.10
votes	32-3-1	0.0001	05000	fan	0.10	9973	enc	0.90
vowel	10-6-11	0.0001	05000	fan	0.10	9973		0.90
water	38-7-13	0.0001	05000	fan	0.90	9973		0.95
wave21	21-32-3	0.0001	10000	fan	0.10	9973		0.50
wave40	40-17-3	0.0001	10000	fan	0.10	9973		0.50
wine	13-3-3	0.0001	05000	fan	0.10	9973		0.90
yellow	4-2-1	0.0001	05000	fan	0.10	9973	enc	0.90
ys!as	4-2-1	0.0001	05000	fan	0.10	9973	enc	0.90
zoo	16-5-7	0.0001	05000	fan	0.10	9973	enc	0.90

Appendix E. Database Descriptions

The databases used in this study are described with their title from [38] or [92], the short names as used in this study, the names of their creators or owners, and a brief description of the learning problem contained in the database.

Balloons Data — adult-, adult!, yellow, ys!as
Michael Pazzani
Cognitive Psychology: Influence of prior knowledge on concept acquisition

Mushrooms Data — agaric
Jeffery Schlimmer
Classify mushrooms as poisonous or edible using data from Audubon Society Field Guide to North American Mushrooms

Thyroid Data — ahyper, ahypo, allrep,ann, dis, hypoth, newthy, sick, sickeu, thy387
J. Ross Quinlan, Garavan Institute
Thyroid patient records classified into disjoint diseases

Annealing Data — anneal
David Sterling and Wray Buntine
Steel annealing

Standardized Audiology Data — audio
Professor Jergen, Baylor College of Medicine, and J. Ross Quinlan

Auto-Mpg Data — auto
Modified by J. Ross Quinlan from CMU's Statlib library
City-cycle fuel consumption in miles per gallon

Balance Scale Weight and Distance Data — balanc
R.S. Siegler and Tim Hume
Cognitive psychology experimental results

Breast Cancer Data — breast
M. Zwitter and M. Soklic, Institue of Oncology, Ljubljana, Yugoslavia

Wisconsin Breast Cancer Data — breastw
W. Wolberg, University of Wisconsin Hospitals

Pittsburgh Bridges Data — bridg1, bridg2
Yoram Reich and Steven Fenves, CMU
Design knowledge

Liver-disorders Data — bupa
BUPA Medical Research Ltd.
Blood tests thought to be sensitive to liver disorders

Parity Problems — cmupa7, cmupa8
Matt White
7- and 8-Bit parity problems created using M. White's parity benchmark data generator

Secondary Structure of Globular Protein — cmupro
Terrence Sejnowski
Use linear sequence of amino acids to predict which secondary structure it belongs to

Sonar Data — cmuson
Terrence Sejnowski
Predict whether sonar signal is from a rock or from a metal cylinder, ie, a mine -- differs slightly from sonar

Two-Spirals Data — cmusp, cmusp2
Matt White
Points on two logarithmic spirals created using M. White's generator

Vowel Recognition Data — cmuvow
Tony Robinson
Speaker independent recognition of eleven steady state vowels of British English

Credit Card Application Approval Data — crx
J. Ross Quinlan

Echocardiogram Data: Reed Institute — echoc
Evlin Kinney
Predict whether heart attack patient will survive one year after his heart attack

Flags Data — flag
Collins Gem Guide to Flags
Predict things about a country using size and color of its flag.

Glass Identification Data — glass
B. German, Home Office Forensic Science Service
Identify glass samples from their oxide content

Hayes-Roth and Hayes-Roth's Data — hayes
Barbara and Frederick Hayes-Roth
Concept learning and the recognition and classification of exemplars.

Heart Disease Data — heart, heartc, hearth, hearts, heartv
Robert Detrano, Cleveland Clinic Foundation
Andras Janosi, Hungarian Institute of Cardiology
William Steinbrunn and Matthias Pfisterer, University Hospitals, Switzerland
Robert Detrano, VA Medical Center, Long Beach
Predict heart disease.

Hepatitis Data — hepat
Bojan Cestnik, Jozef Sefan Institute, Ljubljana, Yugoslavia
Predict death of hepatitis suffer.

Horse Colic Data — horse
Mary McLeish and Matt Cecile, University of Guelph
Predict death of horse suffering from colic.

Housing Data — house
D. Harrison and D. Rubinfeld
Housing prices in suburbs of Boston.

1985 Auto Imports Data — imports
Jeffery Schlimmer
Predict price of a car.

Ionosphere Data — iono
Vince Sigillito, Johns Hopkins University
Classification of radar returns from ionosphere.

Iris Plant Data — iris
R.A. Fisher
Classify irises from petal and sepal lengths.

Isolated Spoken Letter Recognition Data — isol5
Ron Cole and Mark Fanty, Oregon Graduate School
Predict which letter-name was spoken.

Kinship Data kinsh
Geoff Hinton
Predict familial relationships.

Chess Data krvskp
Alen Shapiro
Predict outcome of king-rook versus king-pawn end game.

Labor Relations Data laborn
Collective Bargaining Review
Final settlements in labor negotiations in Canadian industry.

LED Displays led7, led17
L. Breiman, J. Friedman, R. Olshen, and C. Stone
LED display with 7 noisy attributes and with 17 irrelevant attributes

Lenses Data: Fitting contact lenses lense
J. Cendrowska
Predict if someone should be fitted with contact lenses and the type of lenses

Letter Recognition Data letter
David Slate
Identify pixel image as one of 26 capital letters.

Low Resolution Spectrometer Data lrs
NASA Ames Research Center
Identify class of infra-red spectral data.

Lung Cancer Data lungc
Stefan Aeberhard
Identify type of lung cancer.

Lymphography Data lymph
M. Zwitter and M. Soklic, Institute of Oncology, Ljubljana, Yugoslavia
Predict health of lymph system.

Computer Hardware Data mache, machp
Phillip Ein-Dor and Jacob Feldmesses, Tel Aviv University
Predict relative CPU performance characterized by cycle time, memory size

Monk's Problems monk1, monk2, monk3
Sebastian Thrun, CMU
This artificial domain was basis of first international comparison of learning algorithms

Musk Data musk1, musk2
AI Group at Arris Pharmaceutical Corp.
Determine if a molecule is musk or non-musk

NetTalk Corpus netpho, netstr, nettyp
Terrence Sejnowski
Predict proper phonemes, stress, and word type (irregular, foreign, other) using string of letters as input.

Pima Indians Diabetes Database pima
National Institute of Diabetes and Digestive and Kidney Diseases
Predict onset of diabetes.

Postoperative Patient Data postop
Sharon Summers, University of Kansas and Linda Woolery, University of Missouri
Determine where patients in postoperative recovery area should go to next.

Molecular Biology promot
C. Harley, R. Reynolds, M. Noordewier
E. Coli Promoter gene sequences (DNA)

Molecular Biology protei
Terrence Sejnowski
Secondary Structure of Globular Proteins

Primary Tumor Data ptumor
M. Zwitter and M. Soklic, Institute of Oncology, Ljubljana, Yugoslavia
Predict location of tumor

Image Segmentation Data segm
Vision Group, University of Massachusetts
Classify 3x3 pixel images.

Servo Data servo
Karl Ulrich
Predict rise time of a servomechanism.

Space Shuttle Autolanding Data slc
NASA
Determine whether to autoland or manually land the space shuttle.

Sonar Data sonar
Terrence Sejnowski
Predict whether sonar signal is from a rock or from a metal cylinder, ie, a mine (differs slightly from cmuson)

Soybean Disease Data soyl, soyf, soys, soyso
R.S. Michalski
Soybean diseases, subset of soyl used in Fisher's dissertation, subset of soyl, subset of soyl used in Stepp's dissertation

Molecular Biology splice
Genbank 64.1
Primate splice-junction gene sequences (DNA)

Challenger USA Space Shuttle O-Ring Data ssblow, ssonly
David Draper, University of California, Los Angeles
Analysis of launch temperature vs. O-ring stress

Statlog Project Australian Credit Approval Data staust
Credit card applications

Statlog Project German Credit Data stgern, stgers
Professor Dr. Hans Hofmann, Hamburg University
Classifies people described by a set of numeric (numeric and symbolic) attributes as good or bad credit risks

Statlog Project Heart Disease Data sthear
Predict presence or absence of heart disease

Statlog Project Landsat Satellite Data stsati
Australian Centre for Remote Sensing
Multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood.

Statlog Project Shuttle Data stshut
Jason Catlett, University of Sydney

Statlog Project Image Segmentation Data stsegm
Vision Group, University of Massachusetts
Classify 3x3 pixel images.

Statlog Project Vehicle Silhouettes Data stvehi
Turing Institute, Glasgow, Scotland
Classify a silhouette as one of four types of vehicle.

Tic-Tac-Toe Endgame Data tictac
David Aha
Complete set of possible board configurations for tic-tac-toe endgames.

Trains Data trains
R.S. Michalski and Robert Stepp
Determine whether train is traveling east or west based on composition of train.

1984 Congressional Voting Records — votes
Congressional Quarterly Almanac, 98th Congress
Predict party affiliation based on voting record.

Vowel Recognition — vowel
David Deterding, Mahesan Niranjan, Tony Robinson
Train on one set of speakers and recognize vowels said by test speakers.

Water Treatment Plant Data — water
Manel Pock
Classify operational state of a urban waste water treatment plant.

Waveform Data — wave21, wave40
L. Breiman, J. Friedman, R. Olshen, and C. Stone
Three classes of waveforms with noisy (19 of which are pure noise) attributes.

Wine Recognition Data — wine
M. Forina and Sefan Aeberhard
Use chemical analysis to determine origin of Italian wines.

Zoological Data — zoo
Richard Forsyth
Classify animal given an “artificial” description

Bibliography

1. David Aha and D. Kibler (1989). Noise-tolerant instance-based learning algorithms. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers. San Mateo, CA. 794-799.
2. David W. Aha (1989). Incremental, Instance-Based Learning of Independent and Graded Concept Descriptions. Proceedings of the Sixth International Workshop on Machine Learning. Morgan Kaufmann Publishers. San Mateo, CA. 387-391.
3. David Aha, D. Kibler, and Albert, M. (1989). Instance-based prediction of real-valued attributes. Computational Intelligence, 5. 51-57.
4. David Aha (1991). Incremental constructive induction: An instance-based approach. Proceedings of the Eighth International Workshop on Machine Learning. Morgan Kaufmann Publishers, Evanston, IL. 117-121.
5. David Aha (1992). Tolerating Noisy, Irrelevant, and Novel Attributes in Instance-Based Learning Algorithms. International Journal of Man-Machine Studies, 36. 267-287.
6. David Aha (1993). IBL software documentation.
7. David Aha, Dennis Kibler, and Marc Albert (1991). Instance-Based Learning Algorithms. Machine Learning 6. 37-66.
8. David Aha (1992). Generalizing from Case Studies: A Case Study. Proceeding of the Ninth International Conference on Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.
9. David Aha and Marc Albert (1995). Analyses of Instance-Based Learning Algorithms. To appear in Proceedings of the Ninth National Conference on Artificial Intelligence. AAAI Press.
10. David Aha and Steven Salzberg (1993). Learning to Catch: Applying Nearest Neighbor Algorithms to Dynamic Control Tasks. Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics. Unpublished.
11. Subutai Ahmad, Gerald Tesauro, and Yu He (1990). Asymptotic Convergence of Backpropagation: Numerical Experiments. Advances in Neural Information Processing Systems 2 David S. Touretzky editor. Morgan Kaufmann Publishers, San Mateo CA. 606-613.
12. Subutai Ahmad and Volker Tresp (1993). Some Solutions to the Missing Feature Problem in Vision. To appear in Advances in Neural Information Processing Systems 5 S.J. Hanson, J.D. Cowan, and C.L. Giles editors. Morgan Kaufmann Publishers, San Mateo CA.
13. Albert, M. K., & Aha, D. W. (1991). Analyses of instance-based learning algorithms. Proceedings of the Ninth National Conference on Artificial Intelligence. AAAI Press Anaheim, CA. 553-558.
14. Les Atlas, Ronald Cole, Jerome Connor, Mohammed El-Sharkawi, Robert J. Marks II, Yeshwant Muthusamy, and Etienne Barnard (1990). Performance Comparisons Between Backpropagation Networks and Classification Trees on Three Real-World Applications. Advances in Neural Information Processing

Systems 2 David S. Touretzky editor. Morgan Kaufmann Publishers, San Mateo CA. 622-629.

15. Eric B. Baum (1990). The Perceptron Algorithm Is Fast for Non-Malicious Distributions. Advances in Neural Information Processing Systems 2 David S. Touretzky editor. Morgan Kaufmann Publishers, San Mateo CA. pp 676-685.
16. Eric Baum (1994). When are k-Nearest Neighbor and Backpropagation Accurate for Feasible-Sized Sets of Examples? Computation Learning: Theory and Natural Learning Systems Stephen Hanson, George Drastal, and Ronald Rivest editors. The MIT Press, Cambridge, Massachusetts. pp. 416-442.
17. R. Beale and T. Jackson (1990). Neural Computing: An Introduction. Adam Hilger, Bristol, England.
18. F. Bergandano, S. Matwin, R.S. Michalski, and J. Zhang (1992). Learning two-tiered descriptions of flexible concepts: The Poseidon system. Machine Learning 8. 5-44.
19. Rick Bertelsen and Tony Martinez (1994). Extending ID3 Through Discretization of Continuous Inputs. Proceedings of the 7th Florida Artificial Intelligence Research Symposium. Douglas Dankel II and John Stewman editors. Florida AI Research Society. 122-125.
20. Herbert D. Block (1970). A Review of Perceptrons. Information and Control 17. 501-522.
21. D.M. Boulton and C.S. Wallace (1973). An information measure for hierarchic classification. The Computer Journal 16(3). 254-261.
22. D.M. Boulton and C.S. Wallace (1973). An information measure for single link classification. The Computer Journal 18(3). 236-238.
23. Robin Boswell (1993). Manual for CN2 version 6.1. Technical Report TI/P2154/RAB/4/1.5.
24. Robin Boswell and Tim Niblett (1993). Manual for CN2 version 6.2. Technical Report TI/P2154/RAB/4/1.6.
25. Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone (1984). Classification and Regression Trees. Wadsworth International Group, Belmont CA.
26. Wray Buntine (1989). Learning Classification Rules Using Bayes. Proceedings of the Sixth International Workshop on Machine Learning. Morgan Kaufmann Publishers San Mateo CA. 94-98.
27. Wray Buntine (1990). Myths and Legends in Learning Classification Rules. AAAI-90 Proceedings of the 8th National Conference on Artificial Intelligence. AAAI Press/MIT Press, Cambridge, Massachusetts. 736-742.
28. Wray Buntine and Andr Lopez de Mantaras eas S. Weigend (1992). Bayesian Back-Propagation. Complex Systems 5. 603-643.
29. Wray Buntine (1992). Tree Classification Software. Presented at Technology 2002: The Third National Technology Transfer Conference and Exposition.
30. Wray Buntine (1992). Learning Classification Trees. Statistics and Computing 2. 63-73.

31. Wray Buntine (1993). IND v2.1 user documentation.
32. J. Catlett (1991). On Changing Continuous Attributes into Ordered Discrete Attributes. in Lecture Notes in Artificial Intelligence Vol. 482 J. Siekmann editor. Springer-Verlag, Berlin. 164-178.
33. Peter Cheeseman *et al* (1988). Bayesian Classification. Proceedings of the 7th National Conference on Artificial Intelligence. Morgan Kaufmann Publishers, San Mateo, CA. 607-611.
34. Jie Cheng *et al* (1988). Improved Decision Trees: A Generalized Version of ID3. Proceedings of the 5th International Conference on Machine Learning.Morgan Kaufmann Publishers, San Mateo, CA. 100-106.
35. Peter Clark (1989). Functional Specification of CN and AQ. Technical Report TI/P2154/PC/4/1.2 Turing Institute.
36. Peter Clark and Tim Niblett (1989). The CN2 Induction Algorithm. Machine Learning 3. 261-283.
37. Peter Clark and Robin Boswell (1991). Rule Induction with CN2: Some Recent Improvements. Machine Learning--EWSL-91: European Working Session on Learning in Lecture Notes in Artificial Intelligence Vol. 482, Springer-Verlag. 151-163.
38. CMU Neural Network Learning Benchmark Database.
39. Yann le Cun (1988). A Theoretical Framework for Back-Propagation. Proceedings of the 1988 Connectionist Models Summer School. David Touretzky, Geoffrey Hinton, and Terrence Sejnowski editors. Morgan Kaufmann Publishers, San Mateo, CA. 21-28.
40. Vlad Dabija, Katsuhiko Tsupino, and Shogo Nishida (1992). Learning to Learn Decision Trees. AAAI-92 Proceedings of the 10th National Conference on Artificial Intelligence. AAAI Press/MIT Press, Cambridge, Massachusetts. 88-95.
41. Belur V. Dasarathy editor (1991). Nearest Neighbor(NN) Norms: NN Pattern Classificatiion Techniques. IEEE Computer Society Press, Los Alamitos, California.
42. A.P. Dempster, N.M. Laird, and D.B. Rubin (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society Series B 39(1). 1-22.
43. Thomas G. Dietterich, Hermann Hild, and Ghulum Bakiri (1995). A Comparison of ID3 and Backpropagation for English Text-to-Speech Mapping. Machine Learning 18. 51-80.
44. Thomas Dietterich and Ryszard Michalski. A Comparative Review of Selected Methods for Learning from Examples. Machine Learning: An Artificial Intelligence Approach Vol. 1.
45. Bradley Efron (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. Journal of the American Statistical Association 78(382). 316-331.
46. Scott E. Fahlman (1988). Faster-Learning Variations on Back-Propagation: An Empirical Study. Proceedings of the 1988 Connectionist Models Summer School.

D. Touretzky, G. Hinton, and T. Sejnowski editors. Morgan Kaufmann Publishers, San Mateo CA. 38-51.

47. Scott E. Fahlman and Christian Lebiere (1990). The Cascade-Correlation Learning Architecture. Advances in Neural Information Processing Systems 2 David S. Touretzky editor. Morgan Kaufmann Publishers, San Mateo CA. 524-532.
48. Scott E. Fahlman and Christian Lebiere (1991). The Cascade-Correlation Learning Architecture. Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh PA.
49. Usama Fayyad and Keki Irani (1990). What Should Be Minimized in a Decision Tree? AAAI-90 Proceedings of the 8th National Conference on Artificial Intelligence. AAAI Press/MIT Press, Cambridge, Massachusetts. 749-754.
50. Usama Fayyad and Keki Irani (1992). The Attribute Selection Problem in Decision Tree Generation. AAAI-92 Proceedings of the 10th National Conference on Artificial Intelligence. AAAI Press/MIT Press, Cambridge, Massachusetts. 104-110.
51. Usama Fayyad and Keki Irani (1992). On the Handling of Continuous-Valued Attributes in Decision Tree Generation. Machine Learning 8. 87-102.
52. E. Fiesler (1993). Neural Network Classification and Formalization. Submitted to Computer Standards and Interfaces John Fulcher editor. North-Holland, Elsevier Science Publishers, Amsterdam, The Netherlands.
53. Douglas Fisher and Kathleen McKusick (1989). An Empirical Comparison of ID3 and Back-propagation. Proceedings of the 11th International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Mateo, CA.788-793.
54. Terence Fogarty (1992). First Nearest Neighbor Classification on Frey and Slate's Letter Recognition Problem. Machine Learning 9. 387-388.
55. David Fogel (1995). Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, New York.
56. James A. Freeman and David M. Skapura (1991). Neural Networks: Algorithms, Applications, and Programming Techniques. Addison-Wesley Publishing Company, Reading, Massachusetts.
57. Zoubin Ghahramani and Michael I. Jordan (1994). Supervised learning from incomplete data via an EM approach. Advances in Neural Information Processing Systems 6. J.D. Cowan, G. Tesauro, and J. Alspector editors. Morgan Kaufmann Publishers, San Francisco CA.
58. Leonard G.C. Hamey (1992). Benchmarking Feed-Forward Neural Networks: Models and Measures. Advances in Neural Information Processing Systems 4. John Moody, Steve Hanson, and Richard Lippmann editors. Morgan Kaufmann Publishers, San Mateo, CA. 1167-1174.
59. Simon Haykin (1994). Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, New York.
60. Donald Hebb (1949). The Organization of Behaviour. John Wiley & Sons, Inc, New York.

61. John Hertz, Anders Krogh, and Richard G. Palmer (1991). Introduction to the Theory of Neural Computation. Lecture Notes Volume I. Santa Fe Institute. Studies in the Sciences of Complexity. Addison-Wesley Publishing Company, Redwood City CA.
62. Ray Hickey (1991). A Test Bed for Some Machine Learning Algorithms. AI and Cognitive Science 1990 Micheal McTear and Norman Creaney editors. Springer-Verlag. 70-88.
63. Geoffrey Holmes, Andres Donkin, and Ian H. Witten (1994). WEKA: A Machine Learning Workbench. Department of Computer Science, University of Waikato, Hamilton, New Zealand.
64. Robert C. Holte (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning 11. 63-91.
65. Jiarong Hong, Igor Mozetic, Ryszard S. Michalski (1986). AQ15: Incremental Learning of Attribute-Based Descriptions from Examples The Method and User's Guide. Technical Report ISG 86-5.
66. Qing Hu, Robert Plant, and David Hertz (1994). An Intelligent Trainer for Neural Networks. Proceedings of the 7th Florida Artificial Intelligence Research Symposium. Douglas Dankel II and John Stewman editors. Florida AI Research Society. 11-15.
67. Benedikt Humpert (1994). Improving Back Propagation With a New Error Function. Neural Networks 7(8). 1191-1192.
68. Michael Kattan, Dennis Adams, and Michael Parks (1993). A Comparison of Machine Learning with Human Judgement. Journal of Management Information Systems 9 (4). 37-57.
69. Randy Kerber (1992). ChiMerge: Discretization of Numeric Attributes. AAAI-92 Proceedings of the 10th National Conference on Artificial Intelligence. AAAI Press/MIT Press, Cambridge, Massachusetts. 123-127.
70. D. Kibler and David Aha (1988). Instance-based prediction of real-valued attributes. Proceedings of the Seventh Biennial Canadian Conference on Artificial Intelligence. Morgan Kaufmann Publishers, San Mateo, CA. 110-116.
71. D. Kibler and David Aha (1988). Comparing instance-averaging with instance-filtering learning algorithms. EWSL88: Proceedings of the 3rd European Working Session on Learning. Pitman. 63-69.
72. D. Kibler and David Aha (1989). Comparing instance-saving with instance-averaging learning algorithms. Change of Representation and Inductive Bias D. P. Benjamin editor. Kluwer Academic Publishers, Norwell, MA.
73. John F. Kolen and Jordan B. Pollack (1991). Back Propagation is Sensitive to Initial Conditions. Advances in Neural Information Processing Systems 3 R. Lippmann, J. Moody, and D. Touretzky editors. Morgan Kaufmann Publishers, San Mateo, CA. 860-867.
74. Igor Kononenko and Ivan Bratko (1990). Information-Based Evaluation Criterion for Classifier's Performance. Machine Learning 6. 67-80.
75. John R. Koza (1992). Genetic Programming: On the Programming of Computers

by Means of Natural Selection. The MIT Press, Cambridge, Massachusetts.
76. Pat Langley (1988). Machine Learning as an Experimental Science. Machine Learning 3. 5-8.
77. Pat Langley, Wayne Iba, and Kevin Thompson (1992). An Analysis of Bayesian Classifiers. AAAI-92 Proceedings of the 10th National Conference on Artificial Intelligence. AAAI Press/MIT Press, Cambridge, Massachusetts. 223-228.
78. Raoul Lepage and Lynne Billard editors (1992). Exploring the Limits of Bootstrap. John Wiley & Sons, Inc, New York.
79. Roderick Little and Donald Rubin (1987). Statistical Analysis with Missing Data. John Wiley & Sons, New York.
80. Roderick J.A. Little (1992). Regression With Missing X's: A Review. Journal of the American Statistical Association. 87(420). 1227-1237.
81. R. Lopez de Mantaras (1991). A Distance-Based Attribute Selection Measure for Decision Tree Induction. Machine Learning 6. 81-92.
82. H. Lounis and G. Bisson (1991). Evaluation of Learning Systems: An Artificial Data-Based Approach. Lecture Notes in Artificial Intelligence Vol. 482 J. Siekmann editor. Springer-Verlag, Berlin. 463-481.
83. Paul Lukowicz, Ernst Heinz, Lutz Prechelt, and Walter Tichy (1994). Experimental Evaluation in Computer Science: A Quantitative Study. Technical Report 17/94, Department of Informatics, University of Karlsruhe, Germany.
84. James L. McClelland and David E. Rumelhart (1988). Explorations in Parallel Distributed Processing. The MIT Press, Cambridge, Massachusetts.
85. James L. McClelland and David E. Rumelhart (1988). Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises. The MIT Press, Cambridge, Massachusetts.
86. Warren S. McCulloch (1965). Embodiments of Mind. MIT Press, Cambridge, Massachusetts.
87. Ryszard Michalski *et al* (1986). The Multi-Purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains. Proceedings of the 5th National Conference on Artificial Intelligence. Morgan Kaufmann Publishers, San Mateo, CA. 1041-1045.
88. D. Michie, D. Spiegelhalter, and C. Taylor (1994). Machine Learning, Neural and Statistical Classification. Ellis Horwood, Hertfordshire, England. Book-19
89. John Mingers (1989). An Empirical Comparison of Pruning Methods for Decision Tree Induction. Machine Learning 4. 227-243.
90. Marvin Minsky and Seymour Papert (1988). Perceptrons: An Introduction to Computational Geometry. Expanded Edition. The MIT Press, Cambridge, Massachusetts.
91. B. Mueller and J. Reinhardt (1990). Neural Networks: An Introduction. Springer-Verlag, New York.
92. Patrick M. Murphy and David W. Aha. UCI Repository of Machine Learning Databases. University of California, Department of Information and Computer Science (ics.uci.edu). Irvine, CA.

93. Balas K. Natarajan (1991). Machine Learning: A Theoretical Approach. Morgan Kaufmann Publishers, San Mateo, CA.
94. Nils J. Nilsson (1990). The Mathematical Foundations of Learning Machines. Morgan Kaufmann Publishers, Inc, San Mateo CA.
95. Lutz Prechelt (1994). Summary of Usenet discussion on unknown input values in neural networks.
96. Lutz Prechelt (1994). A Study of Experimental Evaluations of Neural Network Learning Algorithms: Current Research Practice. Technical Report 19/94, Fakultaet fuer Informatik, Universitaet Karlsruhe, Germany.
97. Lutz Prechelt (1994). Proben1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules. Fakultaet fuer Informatik, Universitaet Karlsruhe, Germany.
98. J. Ross Quinlan (1986). Induction of Decision Trees. Machine Learning 1. 81-106.
99. J. Ross Quinlan (1986). The Effect of Noise on Concept Learning. Machine Learning: An Artificial Intelligence Approach R. Michalski, J. Carbonell, and T. Mitchell editors. Morgan Kaufmann Publishers, Los Altos, CA. 149-166.
100. J.R. Quinlan (1987). Generating Production Rules from Decision Trees. Proceedings of the 10th International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers, San Mateo, CA. 304-307.
101. J. R. Quinlan (1987). Simplifying decision trees. International Journal of Man-Machine Studies 4. 221-234.
102. J.R. Quinlan (1988). An Empirical Comparison of Genetic and Decision-Tree Classifiers. Proceedings of the 5th International Conference on Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. 135-141.
103. J.R. Quinlan (1989). Unknown Attribute Values in Induction. Proceedings of the Sixth International Workshop on Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.164-168.
104. J.R. Quinlan and Ronald L. Rivest (1989). Inferring Decision Trees Using the Minimum Length Principle. Information and Computation 80. 227-248.
105. J. Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.
106. J.R. Quinlan and R.M. Cameron-Jones (1993). FOIL: A Midterm Report. ECML-93: Proceedings of the European Conference on Machine Learning Pavel Brazdil editor. Springer Verlag. 3-20.
107. J.R. Quinlan (1994). Comparing Connectionist and Symbolic Learning Methods in Computation Learning: Theory and Natural Learning Systems Stephen Hanson, George Drastal, and Ronald Rivest editors. The MIT Press, Cambridge, Massachusetts. 445-456.
108. Larry Rendell and Howard Cho (1990). Empirical Learning as a Function of Concept Character. Machine Learning 5. 267-298.
109. Jorma Rissanen (1978). Modeling By Shortest Data Description. International Federation of Automatic Control. Pergammon Press Ltd.

110. Jorma Rissanen and Glen G. Langdon Jr (1981). Universal Modeling and Coding. IEEE Transactions on Information Theory IT-27(1). 12-23.
111. Jorma Rissanen (1983). A Universal Prior for Integers and Estimation by Minimum Description Length. The Annal of Statistics 11(2). 416-431.
112. Jorma Rissanen (1984). Universal Coding, Information, Prediction, and Estimation. IEEE Transactions on Information Theory IT-30(4). 629-636.
113. Jorma Rissanen (1987). Stochastic Complexity. Journal of the Royal Statistical Society B 49(3). 223-239.
114. Ronald Rivest (1987). Learning Decision Lists. Machine Learning 2. 229-246.
115. Steve Romaniuk (1993). Evolutionary Growth Perceptrons. Department of Information Systems and Computer Science, National University of Singapore.
116. Frank Rosenblatt (1960). Perceptual generalization over transformation groups. Self-Organizing Systems. Pergamon Press, New York. 63-96.
117. Frank Rosenblatt (1962). Principles of Neurodynamics. Spartan Books, New York.
118. Cullen Schaffer (1991). When Does Overfitting Decrease Prediction Accuracy in Induced Decision Trees and Rule Sets? in Lecture Notes in Artificial Intelligence Vol. 482 edited by J. Siekmann. Springer-Verlag, Berlin. 192-205.
119. Cullen Schaffer (1993). Selecting a Classification Method by Cross-Validation. Dept of Computer Science, Hunter College NY.
120. Jeffery Schlimmer and Douglas Fisher (1986). A Case Study of Incremental Concept Induction. Proceedings of the 5th National Conference on Artificial Intelligence. Morgan Kaufmann Publishers, San Mateo, CA. 496-501.
121. Jude Shavlik, Raymond Mooney, and Geoffrey Towell (August 1990). Symbolic and Neural Learning Algorithms: An Experimental Comparison (Revised). University of Wisconsin Computer Sciences Technical Report 955.
122. M.L. Southcott and R.E. Bogner (1994). Classification of incomplete data using neural networks. Technical report University of Adelaide, Australia.
123. Scott Spangler, Usama M. Fayyad, and Ramasamy Uthurusamy (1989). Induction of Decision Trees from Inconclusive Data. Proceedings of the Sixth International Workshop on Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. 146-150.
124. M. Stone (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society Series B 36(1). 111-147.
125. Brian Telfer and Harold Szu (1994). Energy Functions for Minimizing Misclassification Error With Minimum-Complexity Networks. Neural Networks 7(5). 809-818.
126. A.C. Tsoi and R.A. Pearson (1991). Comparison of three classification techniques, CART, C4.5 and Multi-Layer Perceptrons. Advances in Neural Information Processing Systems 3 R. Lippmann, J. Moody, and D. Touretzky editors. Morgan Kaufmann Publishers, San Mateo, CA. 963-969.
127. Peter Vamplew and Anthony Adams (1993). Missing Values in Backpropagation Neural Net. Technical Report Department of Computer Science, University of Tasmania.

128. Dan Ventura and Tony Martinez (1994). BRACE: A Paradigm For the Discretization of Continuously Valued Data. Proceedings of the 7th Florida Artificial Intelligence Research Symposium. Douglas Dankel II and John Stewman editors. Florida AI Research Society. 118-121.
129. C.S. Wallace and M.P. Georgeff (1985). A General Selection Criterion for Inductive Inference. Advances in Artificial Intelligence T. O'Shea editor. Elsevier Science Publishers B.V. (North-Holland). 219-229.
130. C.S. Wallace and P.R. Freeman (1987). Estimation and Inference by Compact Coding. Journal of the Royal Statistical Society B. 49(3). 240-265.
131. C.S. Wallace and J.D. Patrick (1993). Coding Decision Trees. Machine Learning 11. 7-22.
132. Sam Waugh and Anthony Adams (1993). Comparison of Inductive Learning of Classification Tasks by Neural Networks. Technical Report R93-5 Department of Computer Science, University of Tasmania.
133. Sholom Weiss, Robert Galen, and Prasad Tadepalli (1987). Optimizing the Predictive Value of Diagnostic Decision Rules. Proceedings of the 6th National Conference on Artificial Intelligence. Morgan Kaufmann Publsihers, San Mateo, CA. 521-526.
134. S.M. Weiss and I. Kapouleas (1990). An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence. Morgan Kaufmann, San Mateo, CA.781-787.
135. Allan White and Wei Zhong Liu (1994). Bias in Information-Based Measures in Decision Tree Induction. Machine Learning 15. 321-329.
136. Jarryl Wirth and Jason Catlett (1988). Experiments on the Costs and Benefits of Windowing in ID3. Proceedings of the 5th International Conference on Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. 87-99.
137. Janusz Wnek and Ryszard Michalski (1991). An Experimental Comparison of Symbolic and Subsymbolic Learning Paradigms: Phase I - Learning Logic-style Concepts. First International Multi-Strategy Learning Workshop.
138. Zijian Zheng (1993). A Benchmark for Classifier Learning. Technical Report 474, Basser Department of Computer Science, University of Sydney.
139. Carnegie Mellon University StatLib Dataset Archive (http://lib.stat.cmu.edu/datasets/).
140. Scripps Institution of Oceanography Library (http://orpheus.ucsd.edu/sio/dataserv/)
141. National Climatic Data Center (http://www.ncdc.noaa.gov/ncdc.html)
142. National Geophysical Data Center (http://web.ngdc.noaa.gov/
143. National Oceanographic Data Center (http://www.nodc.gov/NODC-home.html)