References

[1] Blake, C.L. and Merz, C.J. (1998). UCI Repository of Machine Learning Databases. Irvine, CA: University of California, Department of Information and Computer Science. [http://www.ics.uci.edu/∼mlearn/MLRepository.html]

[2] Michie, D. (1990). Machine Executable Skills from 'Silent' Brains. In *Research and Development in Expert Systems VII*, Cambridge University Press.

[3] Quinlan, J.R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.

[4] Quinlan, J.R. (1986). Induction of Decision Trees. *Machine Learning*, vol. 1, pp. 81–106.

[5] Mingers, J. (1989). An Empirical Comparison of Pruning Methods for Decision Tree Induction. *Machine Learning*, vol. 4, pp. 227–243.

[6] Quinlan, J.R. (1979). Discovering Rules by Induction from Large Collections of Examples. In Michie, D. (ed.), *Expert Systems in the Microelectronic Age*. Edinburgh University Press, pp. 168–201.

[7] Kerber, R. (1992). ChiMerge: Discretization of Numeric Attributes. In *Proceedings of the 10*th *National Conference on Artificial Intelligence*. AAAI Press, pp. 123–128.

[8] Esposito, F., Malerba, D. and Semeraro, G. (1997). A Comparative Analysis of Methods for Pruning Decision Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19 (5).

[9] McSherry, D. and Stretch, C. (2003). Information Gain. University of Ulster Technical Note.

[10] Noordewier, M.O., Towell, G.G. and Shavlik, J.W. (1991). Training Knowledge-Based Neural Networks to Recognize Genes in DNA Sequences. *Advances in Neural Information Processing Systems*, vol. 3, Morgan Kaufmann.

[11] Cendrowska, J. (1987). PRISM: An Algorithm for Inducing Modular Rules. *International Journal of Man-Machine Studies*, vol. 27, pp. 349–370.

[12] Cendrowska, J. (1990). Knowledge Acquisition for Expert Systems: Inducing Modular Rules from Examples. PhD Thesis, The Open University.

[13] Bramer, M.A. (2000). Automatic Induction of Classification Rules from Examples Using N-Prism. In *Research and Development in Intelligent Systems XVI*, Springer-Verlag, pp. 99–121.

[14] Piatetsky-Shapiro, G. (1991). Discovery, Analysis and Presentation of Strong Rules. In Piatetsky-Shapiro, G. and Frawley, W.J. (eds.), *Knowledge Discovery in Databases*, AAAI Press, pp. 229–248.

[15] Smyth, P. and Goodman, R.M. (1992). Rule Induction Using Information Theory. In Piatetsky-Shapiro, G. and Frawley, W.J. (eds.), *Knowledge Discovery in Databases*, AAAI Press, pp. 159–176.

[16] Agrawal, R. and Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In Bocca, J.B., Jarke, M. and Zaniolo, C. (eds.), *Proceedings of the 20th International Conference on Very Large Databases (VLDB94)*, Morgan Kaufmann, pp. 487–499. [http://citeseer.nj.nec.com/agrawal94fast.html]

A
Essential Mathematics

This appendix gives a basic description of the main mathematical notation and techniques used in this book. It has four sections, which deal with, in order:

– the subscript notation for variables and the Σ (or 'sigma') notation for summation (these are used throughout the book, particularly in Chapters 3–5)

– tree structures used to represent data items and the processes applied to them (these are used particularly in Chapters 3, 4 and 8)

– the mathematical function $\log_2 X$ (used particularly in Chapters 4, 5 and 9)

– set theory (which is used in Chapter 13).

If you are already familiar with this material, or can follow it fairly easily, you should have no trouble reading this book. Everything else will be explained as we come to it. If you have difficulty following the notation in some parts of the book, you can usually safely ignore it, just concentrating on the results and the detailed examples given.

A.1 Subscript Notation

This section introduces the subscript notation for variables and the Σ (or 'sigma') notation for summation which are used throughout the book, particularly in Chapters 3–5.

It is common practice to use variables to represent numerical values. For example, if we have six values we can represent them by a, b, c, d, e and f, although any other six variables would be equally valid. Their sum is $a+b+c+d+e+f$ and their average is $(a+b+c+d+e+f)/6$.

This is fine as long as there are only a small number of values, but what if there were 1,000 or 10,000 or a number that varied from one occasion to another? In that case we could not realistically use a different variable for each value.

The situation is analogous to the naming of houses. This is reasonable for a small road of 6 houses, but what about a long road with 200 or so? In the latter case, it is greatly more convenient to use a numbering system such as 1 High Street, 2 High Street, 3 High Street etc.

The mathematical equivalent of numbering houses is to use a *subscript notation* for variables. We can call the first value a_1, the second a_2 and so on, with the numbers 1, 2 etc. written slightly 'below the line' as subscripts. (We pronounce a_1 in the obvious way as the letter 'a' followed by the digit 'one'.) Incidentally, there is no need for the first value to be a_1. Subscripts beginning with zero are sometimes used, and in principle the first subscript can be any number, as long as they then increase in steps of one.

If we have 100 variables from a_1 up to a_{100}, we can write them as $a_1, a_2, \ldots, a_{100}$. The three dots, called an *ellipsis*, indicate that the intermediate values a_3 up to a_{99} have been omitted.

In the general case where the number of variables is unknown or can vary from one occasion to another, we often use a letter near the middle of the alphabet (such as n) to represent the number of values and write them as a_1, a_2, $\ldots$, a_n.

A.1.1 Sigma Notation for Summation

If we wish to indicate the sum of the values a_1, a_2, $\ldots$, a_n we can write it as $a_1 + a_2 + \cdots + a_n$. However there is a more compact and often very useful notation which uses the Greek letter Σ ('sigma'). Sigma is the Greek equivalent of the letter 's', which is the first letter of the word 'sum'.

We can write a 'typical' value from the sequence $a_1, a_2, \ldots, a_n$ as a_i. Here i is called a *dummy variable*. We can use other variables instead of i, of course, but traditionally letters such as i, j and k are used. We can now write the sum $a_1 + a_2 + \cdots + a_n$ as

$$\sum_{i=1}^{i=n} a_i$$

(This is read as 'the sum of a_i for i equals 1 to n' or 'sigma a_i for $i = 1$ to n'.)

The notation is often simplified to

$\sum_{i=1}^{n} a_i$

The dummy variable i is called the *index of summation*. The *lower* and *upper bounds of summation* are 1 and n, respectively.

The values summed are not restricted to just a_i. There can be any formula, for example

$\sum_{i=1}^{i=n} a_i^2$ or $\sum_{i=1}^{i=n} (i.a_i)$.

The choice of dummy variable makes no difference of course, so

$\sum_{i=1}^{i=n} a_i = \sum_{j=1}^{j=n} a_j$

Some other useful results are

$\sum_{i=1}^{i=n} k.a_i = k.\sum_{i=1}^{i=n} a_i$ (where k is a constant)

and

$\sum_{i=1}^{i=n} (a_i + b_i) = \sum_{i=1}^{i=n} a_i + \sum_{i=1}^{i=n} b_i$

A.1.2 Double Subscript Notation

In some situations a single subscript is not enough and we find it helpful to use two (or occasionally even more). This is analogous to saying 'the fifth house on the third street' or similar.

We can think of a variable with two subscripts, e.g. a_{11}, a_{46}, or in general a_{ij} as representing the cells of a table. The figure below shows the standard way of referring to the cells of a table with 5 rows and 6 columns. For example, in a_{45} the first subscript refers to the fourth row and the second subscript refers to the fifth column. (By convention tables are labelled with the row numbers increasing from 1 as we move downwards and column numbers increasing from 1 as we move from left to right.) The subscripts can be separated by a comma if it is necessary to avoid ambiguity.

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}
a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}

We can write a typical value as a_{ij}, using two dummy variables i and j.

If we have a table with m rows and n columns, the second row of the table is a_{21}, a_{22}, ..., a_{2n} and the sum of the values in the second row is $a_{21} + a_{22} + \cdots + a_{2n}$, i.e.

$\sum_{j=1}^{j=n} a_{2j}$

In general the sum of the values in the ith row is

$\sum_{j=1}^{j=n} a_{ij}$

To find the total value of all the cells we need to add the sums of all m rows together, which gives

$\sum_{i=1}^{i=m} \sum_{j=1}^{j=n} a_{ij}$

(This formula, with two 'sigma' symbols, is called a 'double summation'.)

Alternatively we can form the sum of the m values in the jth column, which is

$\sum_{i=1}^{i=m} a_{ij}$

and then form the total of the sums for all n columns, giving

$\sum_{j=1}^{j=n} \sum_{i=1}^{i=m} a_{ij}$

It does not matter which of these two ways we use to find the overall total. Whichever way we calculate it, the result must be the same, so we have the useful result

$$\sum_{i=1}^{i=m} \sum_{j=1}^{j=n} a_{ij} = \sum_{j=1}^{j=n} \sum_{i=1}^{i=m} a_{ij}.$$

A.1.3 Other Uses of Subscripts

Finally, we need to point out that subscripts are not always used in the way shown previously in this appendix. In Chapters 4, 5 and 9 we illustrate the calculation of two values of a variable E, essentially the 'before' and 'after' values. We call the original value E_{start} and the second value E_{new}. This is just a convenient way of labelling two values of the same variable. There is no meaningful way of using an index of summation.

A.2 Trees

Computer Scientists and Mathematicians often use a structure called a *tree* to represent data items and the processes applied to them.

Trees are used extensively in the first half of this book, especially in Chapters 3, 4 and 8.

Figure A.1 is an example of a tree. The letters A to M are labels added for ease of reference and are not part of the tree itself.

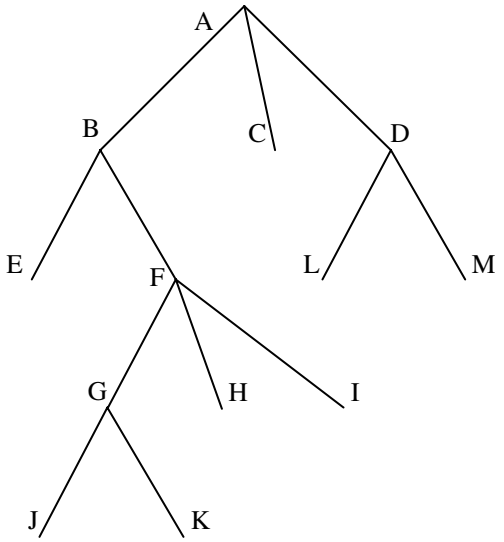


Figure A.1 A Tree with 13 Nodes

A.2.1 Terminology

In general a tree consists of a collection of points, called *nodes*, joined by straight lines, called *links*. Each link has a single node at each end. This is an example of a link joining two nodes G and J.

Figure A.1 comprises 13 nodes, labelled from A to M, joined by a total of 12 links.

The node at the top of the tree is called the *root* of the tree, or the *root node* or just the *root*. (In Computer Science, trees grow downwards from their roots.)

There is an implicit notion of movement down the tree, i.e. it is possible to go from the root node A to node D, or from node F to node H via a link. There is also a *path* from node A to node H via the 'chain' of links A to B, B to F, F to H and a path from node F to node K via links F to G then G to K. There is no way of going from B to A or from G to B, as we cannot go 'backwards' up the tree.

There are a number of conditions that must be satisfied to make a structure such as Figure A.1 a tree:

1. There must be a single node, the root, with no links 'flowing into' it from above.
2. There must be a path from the root node A to every other node in the tree (so the structure is connected).
3. There must be only *one path* from the root to each of the other nodes. If we added a link from F to L to Figure A.1 it would no longer be a tree, as there would be two paths from the root to node L: A to B, B to F, F to L and A to D, D to L.

Nodes such as C, E, H, I, J, K, L and M that have no other nodes below them in the tree are called *leaf nodes* or just *leaves*. Nodes such as B, D, F and G that are neither the root nor a leaf node are called *internal nodes*. Thus Figure A.1 has one root node, eight leaf nodes and four internal nodes.

The path from the root node of a tree to any of its leaf nodes is called a *branch*. Thus for Figure A.1 one of the branches is A to B, B to F, F to G, G to K. A tree has as many branches as it has leaf nodes.

A.2.2 Interpretation

A tree structure is one with which many people are familiar from family trees, flowcharts etc. We might say that the root node A of Figure A.1 represents the most senior person in a family tree, say John. His children are represented by nodes B, C and D, their children are E, F, L and M and so on. Finally John's great-great-grandchildren are represented by nodes J and K.

For the trees used in this book a different kind of interpretation is more helpful.

Figure A.2 is Figure A.1 augmented by numbers placed in parentheses at each of the nodes. We can think of 100 units placed at the root and flowing down to the leaves like water flowing down a mountainside from a single source (the root) to a number of pools (the leaves). There are 100 units at A. They flow down to form 60 at B, 30 at C and 10 at D. The 60 at B flow down to E

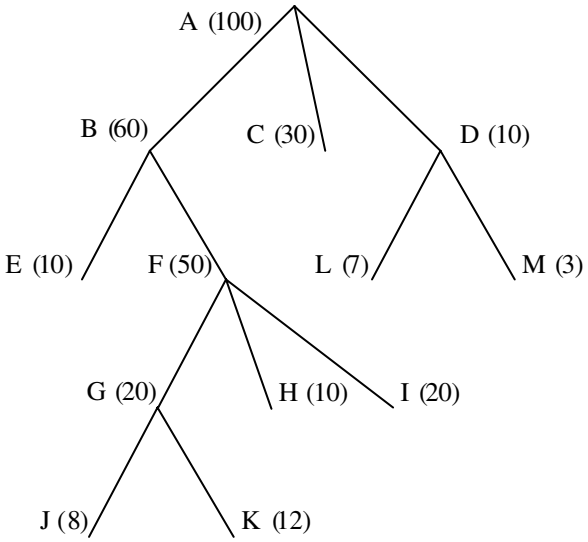


Figure A.2 Figure A.1 (revised)

(10 units) and F (50 units), and so on. We can think of the tree as a means of distributing the original 100 units from the root step-by-step to a number of leaves. The relevance of this to using decision trees for classification will become clear in Chapter 3.

A.2.3 Subtrees

If we consider the part of Figure A.1 that hangs below node F, there are six nodes (including F itself) and five links which form a tree in their own right (see Figure A.3). We call this a *subtree* of the original tree. It is the subtree 'descending from' (or 'hanging from') node F. A subtree has all the characteristics of a tree in its own right, including its own root (node F).

Sometimes we wish to 'prune' a tree by removing the subtree which descends from a node such as F (leaving the node F itself intact), to give a simpler tree, such as Figure A.4. Pruning trees in this way is dealt with in Chapter 8.

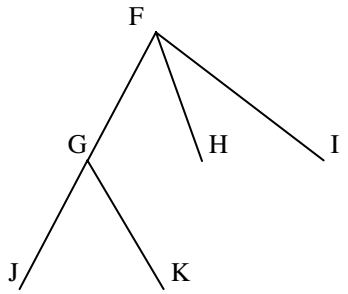


Figure A.3 Subtree Descending From Node F

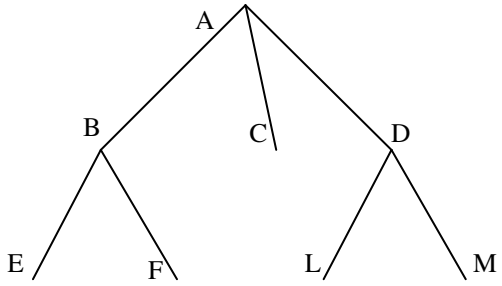


Figure A.4 Pruned Version of Figure A.1

A.3 The Logarithm Function $\log_2 X$

The mathematical function $\log_2 X$, pronounced 'log to base 2 of X', 'log 2 of X' or just 'log X' is widely used in scientific applications. It plays an important part in this book, especially in connection with classification in Chapters 4 and 5 and in Chapter 9.

$\log_2 X = Y$ means that $2^Y = X$.

So for example $\log_2 8 = 3$ because $2^3 = 8$.

The 2 is always written as a subscript. In $\log_2 X$ the value of X is called the 'argument' of the $\log_2$ function. The argument is often written in parentheses, e.g. $\log_2(X)$ but we will usually omit the parentheses in the interests of simplicity when no ambiguity is possible, e.g. $\log_2 4$.

The value of the function is only defined for values of X greater than zero. Its graph is shown in Figure A.5. (The horizontal and vertical axes correspond to values of X and $\log_2 X$, respectively.)

Some important properties of the logarithm function are given in Figure A.6.

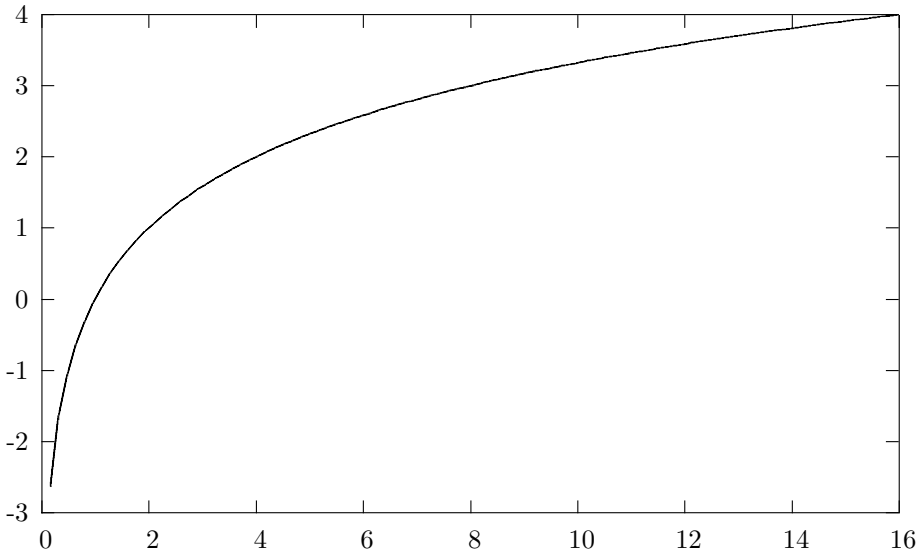


Figure A.5 The $\log_2 X$ Function

The value of $\log_2 X$ is

- negative when $X < 1$
- zero when $X = 1$
- positive when $X > 1$

Figure A.6 Properties of the Logarithm Function

Some useful values of the function are given below.

$$\begin{aligned}
\log_2(1/8) &= -3 \\
\log_2(1/4) &= -2 \\
\log_2(1/2) &= -1 \\
\log_2 1 &= 0 \\
\log_2 2 &= 1 \\
\log_2 4 &= 2 \\
\log_2 8 &= 3 \\
\log_2 16 &= 4 \\
\log_2 32 &= 5
\end{aligned}$$

The $\log_2$ function has some unusual (and very helpful) properties that

greatly assist calculations using it. These are given in Figure A.7.

$\log_2(a \times b) = \log_2 a + \log_2 b$
$\log_2(a/b) = \log_2 a - \log_2 b$
$\log_2(a^n) = n \times \log_2 a$
$\log_2(1/a) = -\log_2 a$

Figure A.7 More Properties of the Logarithm Function

So, for example,

$$\log_2 96 = \log_2(32 \times 3) = \log_2 32 + \log_2 3 = 5 + \log_2 3$$
$$\log_2(q/32) = \log_2 q - \log_2 32 = \log_2 q - 5$$
$$\log_2(6 \times p) = \log_2 6 + \log_2 p$$

The logarithm function can have other bases as well as 2. In fact any positive number can be a base. All the properties given in Figures A.6 and A.7 apply for any base.

Another commonly used base is base 10. $\log_{10} X = Y$ means $10^Y = X$, so $\log_{10} 100 = 2$, $\log_{10} 1000 = 3$ etc.

Perhaps the most widely used base of all is the 'mathematical constant' with the very innocuous name of e. The value of e is approximately 2.71828. Logarithms to base e are of such importance that instead of $\log_e X$ we often write $\ln X$ and speak of the 'natural logarithm', but explaining the importance of this constant is considerably outside the scope of this book.

Few calculators have a $\log_2$ function, but many have a $\log_{10}$, $\log_e$ or ln function. To calculate $\log_2 X$ from the other bases use $\log_2 X = \log_e X/0.6931$ or $\log_{10} X/0.3010$ or $\ln X/0.6931$.

A.3.1 The Function $-X \log_2 X$

The only base of logarithms used in this book is base 2. However the $\log_2$ function also appears in the formula $-X \log_2 X$ in the discussion of entropy in Chapters 4 and 9. The value of this function is also only defined for values of X greater than zero. However the function is only of importance when X is between 0 and 1. The graph of the important part of this function is given in Figure A.8.

The initial minus sign is included to make the value of the function positive (or zero) for all X between 0 and 1.

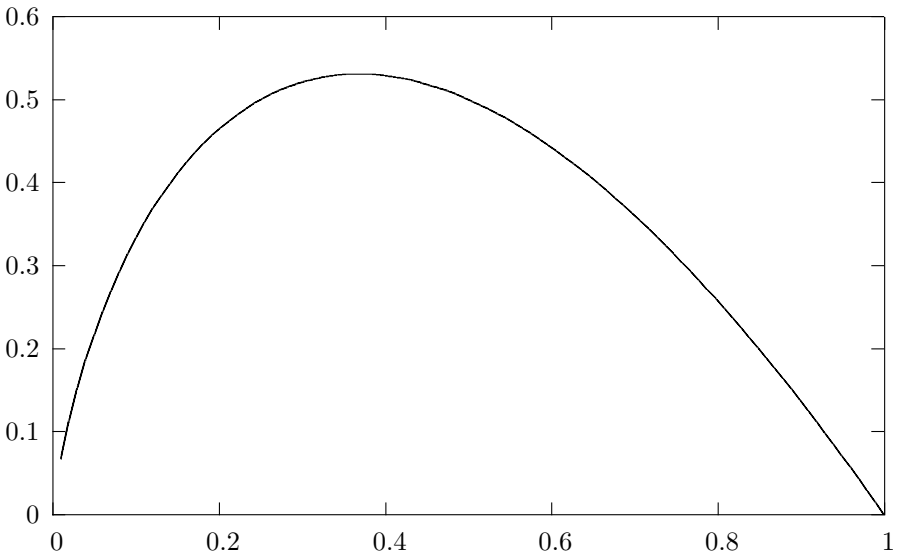


Figure A.8 The function $-X \log_2 X$

It can be proved that the function $-X \log_2 X$ has its maximum value when $X = 1/e = 0.3679$ (e is the 'mathematical constant' mentioned above). When X takes the value $1/e$, the value of the function is approximately 0.5307.

Values of X from 0 to 1 can sometimes usefully be thought of as probabilities (from 0 = impossible to 1 = certain), so we may write the function as $-p \log_2(p)$. The variable used is of course irrelevant as long as we are consistent. Using the fourth property in Figure A.7, the function can equivalently be written as $p \log_2(1/p)$. This is the form in which it mainly appears in Chapters 4 and 9.

A.4 Introduction to Set Theory

Set theory plays an important part in Chapter 13: Association Rule Mining II.

A set is a sequence of items, called *set elements* or *members*, separated by commas and enclosed in braces, i.e. the characters { and }. Two examples of sets are $\{a, 6.4, -2, dog, alpha\}$ and $\{z, y, x, 27\}$. Set elements can be numeric, non-numeric or a combination of the two.

A set can have another set as a member, so $\{a, b, \{a, b, c\}, d, e\}$ is a valid set, with five members. Note that the third element of the set, i.e. $\{a, b, c\}$ is counted as a single member.

No element may appear in a set more than once, so $\{a, b, c, b\}$ is not a valid set. The order in which the elements of a set are listed is not significant, so $\{a, b, c\}$ and $\{c, b, a\}$ are the same set.

The *cardinality* of a set is the number of elements it contains, so {dog, cat, mouse} has cardinality three and $\{a, b, \{a, b, c\}, d, e\}$ has cardinality five. The set with no elements $\{\}$ is called the empty set and is written as $\emptyset$.

We usually think of the members of a set being drawn from some 'universe of discourse', such as all the people who belong to a certain club. Let us assume that set A contains all those who are aged under 25 and set B contains all those who are married.

We call the set containing all the elements that occur in either A or B or both the *union* of the two sets A and B. It is written as $A \cup B$. If A is the set {John, Mary, Henry} and B is the set {Paul, John, Mary, Sarah} then $A \cup B$ is the set {John, Mary, Henry, Paul, Sarah}, the set of people who are either under 25 or married or both. Figure A.9 shows two overlapping sets. The shaded area is their union.

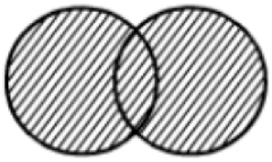


Figure A.9 Union of Two Overlapping Sets

We call the set containing all the elements (if there are any) that occur in both A and B the *intersection* of the two sets A and B. It is written $A \cap B$. If A is the set {John, Mary, Henry} and B is the set {Paul, John, Mary, Sarah} as before, then $A \cap B$ is the set {John, Mary}, the set of people who are both under 25 and married. Figure A.10 shows two overlapping sets. In this case, the shaded area is their intersection.

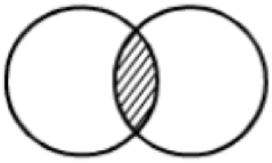


Figure A.10 Intersection of Two Overlapping Sets

Two sets are called *disjoint* if they have no elements in common, for example A ={Max, Dawn} and B={Frances, Bryony, Gavin}. In this case their intersection $A \cap B$ is the set with no elements, which we call the empty set and represent by {} or (more often) by $\emptyset$. Figure A.11 shows this case.

Figure A.11 Intersection of Two Disjoint Sets

If two sets are disjoint their union is the set comprising all the elements in the first set and all those in the second set.

There is no reason to be restricted to two sets. It is meaningful to refer to the union of any number of sets (the set comprising those elements that appear in any one or more of the sets) and the intersection of any number of sets (the set comprising those elements that appear in all the sets). Figure A.12 shows three sets, say A, B and C. The shaded area is their intersection $A \cap B \cap C$.

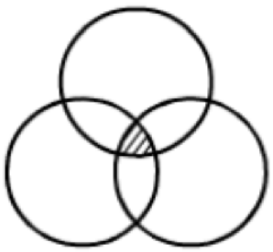


Figure A.12 Intersection of Three Sets

A.4.1 Subsets

A set A is called a *subset* of another set B if every element in A also occurs in B. We can illustrate this by Figure A.13, which shows a set B (the outer circle) with a set A (the inner circle) completely inside it. The implication is that B includes A, i.e. every element in A is also in B and there may also be

one or more other elements in B. For example B and A may be $\{p, q, r, s, t\}$ and $\{q, t\}$ respectively.

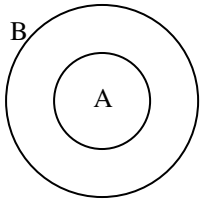


Figure A.13 A is a Subset of B

We indicate that A is a subset of B by the notation $A \subseteq B$. So $\{q, t\} \subseteq \{p, r, s, q, t\}$. The empty set is a subset of every set and every set is a subset of itself.

We sometimes want to specify that a subset A of set B must have fewer elements than B itself, in order to rule out the possibility of treating B as one of its own subsets. In this case we say that A is a *strict subset* of B, written $A \subset B$. So $\{q, t\}$ is a strict subset of $\{p, r, s, q, t\}$ but $\{t, s, r, q, p\}$ is not a strict subset of $\{p, r, s, q, t\}$, as it is the same set. (The order in which the elements are written is irrelevant.)

If A is a subset of B, we say that B is a *superset* of A, written as $B \supseteq A$.

If A is a strict subset of B we say that B is a *strict superset* of A, written as $B \supset A$.

A set with three elements such as $\{a, b, c\}$ has eight subsets, including the empty set and itself. They are $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$ and $\{a, b, c\}$.

In general a set with n elements has 2^n subsets, including the empty set and the set itself. Each member of the set can be included or not included in a subset. The number of possible subsets is therefore the same as the total number of possible include/do not include choices, which is 2 multiplied by itself n times, i.e. 2^n.

The set containing all the subsets of A is called the *power set* of A. Thus the power set of $\{a, b, c\}$ is $\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$.

If set A has n elements its power set contains 2^n elements.

A.4.2 Summary of Set Notation

$\{\}$	The 'brace' characters that enclose the elements of a set, e.g. {apples, oranges, bananas}
$\emptyset$	The empty set. Also written as $\{\}$
$A \cup B$	The union of sets A and B. The set that contains all the elements that occur either in A or B or both.
$A \cap B$	The intersection of two sets A and B. The set that includes all the elements (if there are any) that occur in both A and B.
$A \subseteq B$	A is a subset of B, i.e. every element in A also occurs in B.
$A \subset B$	A is a strict subset of B, i.e. A is a subset of B and A contains fewer elements than B.
$A \supseteq B$	A is a superset of B. True if and only if B is a subset of A.
$A \supset B$	A is a strict superset of B. True if and only if B is a strict subset of A.

B
Datasets

The methods described in this book are illustrated by tests on a number of datasets, with a range of sizes and characteristics. Basic information about each dataset is summarised in Figure B.1.

Dataset	Description	classes*	attributes**		instances	
			categ	cts	training set	test set
anonymous	Football/ Netball Data (anonymised)	2 (58%)	4		12	
bcst96	Text Classification Dataset	2		13430 !	1186	509
chess	Chess Endgame	2 (95%)	7		647	
contact_ lenses	Contact Lenses	3 (88%)	5		108	
crx	Credit Card Applications	2 (56%)	9	6	690 (37)	200 (12)
degrees	Degree Class	2 (77%)	5		26	

football/ netball	Sports Club Preference	2 (58%)	4		12	
genetics	DNA Sequences	3 (52%)	60		3190	
glass	Glass Identification Database	7 (36%)		9 !!	214	
golf	Decision Whether to Play	2 (64%)	2	2	14	
hepatitis	Hepatitis Data	2 (79%)	13	6	155 (75)	
hypo	Hypothyroid Disorders	5 (92%)	22	7	2514 (2514)	1258 (371)
iris	Iris Plant Classification	3 (33.3%)		4	150	
labor-ne	Labor Negotiations	2 (65%)	8	8	40 (39)	17 (17)
lens24	Contact Lenses (reduced version)	3 (63%)	4		24	
monk1	Monk's Problem 1	2 (50%)	6		124	432
monk2	Monk's Problem 2	2 (62%)	6		169	432
monk3	Monk's Problem 3	2 (51%)	6		122	432
pima-indians	Prevalence of Diabetes in Pima Indian Women	2 (65%)		8	768	
sick-euthyroid	Thyroid Disease Data	2 (91%)	18	7	3163	
train	Train Punctuality	4 (70%)	4		20	

vote	Voting in US Congress	2 (61%)	16		300	135
wake_ vortex	Air Traffic Control	2 (50%)	3	1	1714	
wake_ vortex2	Air Traffic Control	2 (50%)	19	32	1714	

* % size of largest class in training set is given in parentheses
** 'categ' and 'cts' stand for Categorical and Continuous, respectively
! Including 1749 with only a single value for instances in the training set
!! Plus one 'ignore' attribute

Figure B.1 Basic Information About Datasets

The *degrees*, *train*, *football/netball* and *anonymous* datasets were created by the author for illustrative purposes only. The *bcst96*, *wake_vortex* and *wake_vortex2* datasets are not generally available. Details of the other datasets are given on the following pages. In each case the class with the largest number of corresponding instances in the training set is shown in **bold**.

Datasets shown as 'Source: UCI Repository' can be downloaded (sometimes with slight differences) from the World Wide Web at
`http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Dataset *chess*

Description
This dataset was used for one of a well-known series of experiments by the Australian researcher Ross Quinlan, taking as an experimental testbed the Chess endgame with White king and rook versus Black king and knight. This endgame served as the basis for several studies of Machine Learning and other Artificial Intelligence techniques in the 1970s and 1980s.

The task is to classify positions (all with Black to move) as either 'safe' or 'lost', using attributes that correspond to configurations of the pieces. The classification 'lost' implies that whatever move Black makes, White will immediately be able to give checkmate or alternatively will be able to capture the Knight without giving stalemate or leaving his Rook vulnerable to immediate capture. Generally this is not possible, in which case the position is 'safe'. This task is trivial for human experts but has proved remarkably difficult to automate in a satisfactory way. In this experiment (Quinlan's 'third problem'), the simplifying assumption is made that the board is of infinite size. Despite this, the classification task remains a hard one. Further information is given in [6].

Source: Reconstructed by the author from description given in [6].

Classes
safe, lost

Attributes and Attribute Values

The first four attributes represent the distance between pairs of pieces (wk and wr: White King and Rook, bk and bn: Black King and Knight). They all have values 1, 2 and 3 (3 denoting any value greater than 2).

dist_bk_bn
dist_bk_wr
dist_wk_bn
dist_wk_wr

The other three attributes all have values 1 (denoting true) and 2 (denoting false).

inline (Black King and Knight and White Rook in line)
wr_bears_bk (White Rook bears on the Black King)
wr_bears_bn (White Rook bears on the Black Knight)

Instances
Training set: 647 instances
No separate test set

Dataset *contact_lenses*

Description
Data from ophthalmic optics relating clinical data about a patient to a decision as to whether he/she should be fitted with hard contact lenses, soft contact lenses or none at all.

Source: Reconstructed by the author from data given in [12].

Classes
hard lenses: The patient should be fitted with hard contact lenses
soft lenses: The patient should be fitted with soft contact lenses
no lenses: The patient should not be fitted with contact lenses

Attributes and Attribute Values
age: 1 (young), 2 (pre-presbyopic), 3 (presbyopic)

specRx (Spectacle Prescription): 1 (myopia), 2 (high hypermetropia), 3 (low hypermetropia)

astig (Whether Astigmatic): 1 (no), 2 (yes)

tears (Tear Production Rate): 1 (reduced), 2 (normal)

tbu (Tear Break-up Time): 1 (less than or equal to 5 seconds), 2 (greater than 5 seconds and less than or equal to 10 seconds), 3 (greater than 10 seconds)

Instances
Training set: 108 instances
No separate test set

Dataset *crx*

Description
This dataset concerns credit card applications. The data is genuine but the attribute names and values have been changed to meaningless symbols to protect confidentiality of the data.

Source: UCI Repository

Classes
+ and − denoting a successful application and an unsuccessful application, respectively (largest class for the training data is −)

Attributes and Attribute Values
A1: b, a
A2: continuous
A3: continuous
A4: u, y, l, t
A5: g, p, gg
A6: c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff
A7: v, h, bb, j, n, z, dd, ff, o
A8: continuous
A9: t, f
A10: t, f
A11: continuous
A12: t, f
A13: g, p, s
A14: continuous
A15: continuous

Instances
Training set: 690 instances (including 37 with missing values)
Test set: 200 instances (including 12 with missing values)

Dataset *genetics*

Description
Each instance comprises the values of a sequence of 60 DNA elements classified into one of three possible categories. For further information see [10].

Source: UCI Repository

Classes
EI, IE and **N**

Attributes and Attribute Values
There are 60 attributes, named A0 to A59, all of which are categorical. Each attribute has eight possible values: A, T, G, C, N, D, S and R.

Instances
Training set: 3190 instances
No separate test set

Dataset *glass*

Description
This dataset is concerned with the classification of glass left at the scene of a crime into one of six types (such as 'tableware', 'headlamp' or 'building_windows_float_processed'), for purposes of criminological investigation. The classification is made on the basis of nine continuous attributes (plus an identification number, which is ignored).

Source: UCI Repository

Classes
1, **2**, 3, 5, 6, 7
Type of glass:
1 building_windows_float_processed
2 building_windows_non_float_processed
3 vehicle_windows_float_processed
4 vehicle_windows_non_float_processed (none in this dataset)
5 container
6 tableware
7 headlamp

Attributes and Attribute Values
Id number: 1 to 214 (an 'ignore' attribute)
plus nine continuous attributes
RI: refractive index
Na: Sodium (unit measurement: weight percent in corresponding oxide, as are the attributes that follow)
Mg: Magnesium
Al: Aluminum
Si: Silicon
K: Potassium
Ca: Calcium
Ba: Barium
Fe: Iron

Instances
Training set: 214 instances
No separate test set

Dataset *golf*

Description
A synthetic dataset relating a decision on whether or not to play golf to weather observations.

Source: UCI Repository

Classes
Play, Don't Play

Attributes and Attribute Values
outlook: sunny, overcast, rain
temperature: continuous
humidity: continuous
windy: true, false

Instances
Training set: 14 instances
No separate test set

Dataset *hepatitis*

Description
The aim is to classify patients into one of two classes, representing 'will live' or 'will die', on the basis of 13 categorical and 9 continuous attributes.

Source: UCI Repository

Classes
1 and **2** representing 'will die' and 'will live' respectively

Attributes and Attribute Values
Age: continuous.
Sex: 1, 2 (representing male, female)
Steroid: 1, 2 (representing no, yes)
Antivirals: 1, 2 (representing no, yes)
Fatigue: 1, 2 (representing no, yes)
Malaise: 1, 2 (representing no, yes)
Anorexia: 1, 2 (representing no, yes)
Liver Big: 1, 2 (representing no, yes)
Liver Firm: 1, 2 (representing no, yes)
Spleen Palpable: 1, 2 (representing no, yes)
Spiders: 1, 2 (representing no, yes)
Ascites: 1, 2 (representing no, yes)
Varices: 1, 2 (representing no, yes)
Bilirubin: continuous
Alk Phosphate: continuous
SGOT: continuous
Albumin: continuous
Protime: continuous
Histology: 1, 2 (representing no, yes)

Instances
Training set: 155 instances (including 75 with missing values)
No separate test set

Dataset *hypo*

Description
This is a dataset on Hypothyroid disorders collected by the Garvan Institute in Australia. Subjects are divided into five classes based on the values of 29 attributes (22 categorical and 7 continuous).

Source: UCI Repository

Classes
hyperthyroid, primary hypothyroid, compensated hypothyroid, secondary hypothyroid, **negative**

Attributes and Attribute Values
age: continuous
sex: M, F
on thyroxine, query on thyroxine, on antithyroid medication, sick, pregnant, thyroid surgery, I131 treatment, query hypothyroid, query hyperthyroid, lithium, goitre, tumor, hypopituitary, psych, TSH measured **ALL** f, t
TSH: continuous
T3 measured: f, t
T3: continuous
TT4 measured: f, t
TT4: continuous
T4U measured: f, t
T4U: continuous
FTI measured: f, t
FTI: continuous
TBG measured: f, t
TBG: continuous
referral source: WEST, STMW, SVHC, SVI, SVHD, other

Instances
Training set: 2514 instances (all with missing values)
Test set: 1258 instances (371 with missing values)

Dataset *iris*

Description
Iris Plant Classification. This is one of the best known classification datasets, which is widely referenced in the technical literature. The aim is to classify iris plants into one of three classes on the basis of the values of four categorical attributes.

Source: UCI Repository

Classes
Iris-setosa, Iris-versicolor, Iris-virginica (there are 50 instances in the dataset for each classification)

Attributes and Attribute Values
Four continuous attributes: sepal length, sepal width, petal length and petal width.

Instances
Training set: 150 instances
No separate test set

Dataset *labor-ne*

Description
This is a small dataset, created by *Collective Bargaining Review* (a monthly publication). It gives details of the final settlements in labor negotiations in Canadian industry in 1987 and the first quarter of 1988. The data includes all collective agreements reached in the business and personal services sector for local organisations with at least 500 members (teachers, nurses, university staff, police, etc).

Source: UCI Repository

Classes
good, bad

Attributes and Attribute Values
duration: continuous [1..7] *
wage increase first year: continuous [2.0..7.0]
wage increase second year: continuous [2.0..7.0]
wage increase third year: continuous [2.0..7.0]
cost of living adjustment: none, tcf, tc
working hours: continuous [35..40]
pension: none, ret_allw, empl_contr (employer contributions to pension plan)
standby pay: continuous [2..25]
shift differential: continuous [1..25] (supplement for work on II and III shift)
education allowance: yes, no
statutory holidays: continuous [9..15] (number of statutory holidays)
vacation: below average, average, generous (number of paid vacation days)
longterm disability assistance: yes, no
contribution to dental plan: none, half, full
bereavement assistance: yes, no (employer's financial contribution towards covering the costs of bereavement)
contribution to health plan: none, half, full

Instances
Training set: 40 instances (39 with missing values)
Test set: 17 instances (all with missing values)

* The notation [1..7] denotes a value in the range from 1 to 7 inclusive

Dataset *lens24*

Description
A reduced and simplified version of *contact_lenses* with only 24 instances.

Source: Reconstructed by the author from data given in [12].

Classes
1, 2, **3**

Attributes and Attribute Values
age: 1, 2, 3
specRx: 1, 2
astig: 1, 2
tears: 1, 2

Instances
Training set: 24 instances
No separate test set

Dataset *monk1*

Description
Monk's Problem 1. The 'Monk's Problems' are a set of three artificial problems with the same set of six categorical attributes. They have been used to test a wide range of classification algorithms, originally at the second European Summer School on Machine Learning, held in Belgium during summer 1991. There are $3 \times 3 \times 2 \times 3 \times 4 \times 2 = 432$ possible instances. All of them are included in the test set for each problem, which therefore includes the training set in each case.
The 'true' concept underlying Monk's Problem 1 is: *if (attribute#1 = attribute#2) or (attribute#5 = 1) then class = 1 else class = 0*

Source: UCI Repository

Classes
0, 1 (62 instances for each classification)

Attributes and Attribute Values
attribute#1: 1, 2, 3
attribute#2: 1, 2, 3
attribute#3: 1, 2
attribute#4: 1, 2, 3
attribute#5: 1, 2, 3, 4
attribute#6: 1, 2

Instances
Training set: 124 instances
Test set: 432 instances

Dataset *monk2*

Description
Monk's Problem 2. See *monk1* for general information about the Monk's Problems. The 'true' concept underlying Monk's problem 2 is: *if (attribute#n = 1) for exactly two choices of n (from 1 to 6) then class = 1 else class = 0*

Source: UCI Repository.

Classes
0, 1

Attributes and Attribute Values
attribute#1: 1, 2, 3
attribute#2: 1, 2, 3
attribute#3: 1, 2
attribute#4: 1, 2, 3
attribute#5: 1, 2, 3, 4
attribute#6: 1, 2

Instances
Training set: 169 instances
Test set: 432 instances

Dataset *monk3*

Description
Monk's Problem 3. See *monk1* for general information about the Monk's Problems. The 'true' concept underlying Monk's Problem 3 is:
if (attribute#5 = 3 and attribute#4 = 1) or (attribute#5 ≠ 4 and attribute#2 ≠ 3) then class = 1 else class = 0
This dataset has 5% noise (misclassifications) in the training set.

Source: UCI Repository

Classes
0, 1

Attributes and Attribute Values
attribute#1: 1, 2, 3
attribute#2: 1, 2, 3
attribute#3: 1, 2
attribute#4: 1, 2, 3
attribute#5: 1, 2, 3, 4
attribute#6: 1, 2

Instances
Training set: 122 instances
Test set: 432 instances

Dataset *pima-indians*

Description
The dataset concerns the prevalence of diabetes in Pima Indian women. It is considered to be a difficult dataset to classify.
The dataset was created by the (United States) National Institute of Diabetes and Digestive and Kidney Diseases and is the result of a study on 768 adult female Pima Indians living near Phoenix. The goal is to predict the presence of diabetes using seven health-related attributes, such as 'Number of times pregnant' and 'Diastolic blood pressure', together with age.

Source: UCI Repository

Classes
0 ('tested negative for diabetes') and 1 ('tested positive for diabetes')

Attributes and Attribute Values
Eight attributes, all continuous: Number of times pregnant, Plasma glucose concentration, Diastolic blood pressure, Triceps skin fold thickness, 2-Hour serum insulin, Body mass index, Diabetes pedigree function, Age (in years).

Instances
Training set: 768 instances
No separate test set

Dataset *sick-euthyroid*

Description Thyroid Disease data.

Source: UCI Repository

Classes
sick-euthyroid and **negative**

Attributes and Attribute Values
age: continuous
sex: M, F
on_thyroxine: f, t
query_on_thyroxine: f, t
on_antithyroid_medication: f, t
thyroid_surgery: f, t
query_hypothyroid: f, t
query_hyperthyroid: f, t
pregnant: f, t
sick: f, t
tumor: f, t
lithium: f, t
goitre: f, t
TSH_measured: y, n
TSH: continuous
T3_measured: y, n
T3: continuous
TT4_measured: y, n
TT4: continuous
T4U_measured: y, n
T4U: continuous.
FTI_measured: y, n
FTI: continuous
TBG_measured: y, n
TBG: continuous

Instances
Training set: 3163 instances
No separate test set

Dataset *vote*

Description
Voting records drawn from the Congressional Quarterly Almanac, 98th Congress, 2nd session 1984, Volume XL: Congressional Quarterly Inc. Washington, DC, 1985.
This dataset includes votes for each of the US House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of vote: voted for, paired for, and announced for (these three simplified to yea), voted against, paired against, and announced against (these three simplified to nay), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known (these three simplified to an unknown disposition).
The instances are classified according to the party to which the voter belonged, either Democrat or Republican. The aim is to predict the voter's party on the basis of 16 categorical attributes recording the votes on topics such as handicapped infants, aid to the Nicaraguan Contras, immigration, a physician fee freeze and aid to El Salvador.

Source: UCI Repository

Classes
democrat, republican

Attributes and Attribute Values
Sixteen categorical attributes, all with values y, n and u (standing for 'yea', 'nay' and 'unknown disposition', respectively): handicapped infants, water project cost sharing, adoption of the budget resolution, physician fee freeze, el salvador aid, religious groups in schools, anti satellite test ban, aid to nicaraguan contras, mx missile, immigration, synfuels corporation cutback, education spending, superfund right to sue, crime, duty free exports, export administration act south africa.

Instances
Training set: 300 instances
Test set: 135 instances

C
Sources of Further Information

Websites

There is a great deal of information about all aspects of data mining available on the World Wide Web. A good place to start looking is the 'Knowledge Discovery Nuggets' site at `http://www.kdnuggets.com`, which has links to information on software, products, companies, datasets, other websites, courses, conferences etc.

Another very useful source of information is The Data Mine at `http://www.the-data-mine.com`.

The KDNet (Knowledge Discovery Network of Excellence) website at `http://www.kdnet.org` has links to journals, conferences, calls for papers and other sources of information.

The Natural Computing Applications Forum (NCAF) is an active British-based group specialising in Neural Nets and related technologies. Their website is at `http://www.ncaf.org.uk`.

Books

There are many books on Data Mining. Some popular ones are listed below.

1. **Data Mining: Concepts and Techniques** by J. Han and M. Kamber. Morgan Kaufmann, 2001. ISBN: 1-55860-489-8.
2. **The Elements of Statistical Learning: Data Mining, Inference,**

and Prediction by T. Hastie, R. Tibshirani and J. Friedman. Springer-Verlag, 2001. ISBN: 0-38795-284-5.

3. **Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations** by I.H. Witten and E. Frank. Morgan Kaufmann, 2000. ISBN: 1-55860-552-5.

   This book is based around *Weka*, a collection of open source machine learning algorithms for data mining tasks that can either be applied directly to a dataset or called from the user's own Java code. Full details are available at `http://www.cs.waikato.ac.nz/ml/weka/index.html`.

4. **C4.5: Programs for Machine Learning** by Ross Quinlan. Morgan Kaufmann, 1993. ISBN: 1-55860-238-0.

   This book gives a detailed account of the author's celebrated tree induction system C4.5, together with a machine-readable version of the software and some sample datasets.

5. **Machine Learning** by Tom Mitchell. McGraw-Hill, 1997. ISBN: 0-07042-807-7.

6. **Survey of Text Mining: Clustering, Classification, and Retrieval** edited by Michael Berry. Springer, 2003. ISBN: 0-38795-563-1.

Books on Neural Nets

Some introductory books on Neural Nets (a topic not covered in this book) are:

1. **Practical Guide to Neural Nets** by Marilyn McCord Nelson and W.T. Illingworth. Addison-Wesley, 1994. ISBN: 0-20163-378-7.

2. **An Introduction to Neural Networks** by K. Gurney. Routledge, 1997. ISBN: 1-85728-503-4.

3. **An Introduction to Neural Networks** by Jeff T. Heaton. CRC Press, 1997. ISBN: 0-97732-060-X.

Conferences

There are many conferences and workshops on Data Mining every year. Two of the most important regular series are:

The annual KDD-20xx series of conferences organised by SIGKDD (the ACM Special Interest Group on Knowledge Discovery and Data Mining) in the United States and Canada. For details see the SIGKDD website at `http://www.acm.org/sigs/sigkdd`.

The annual IEEE ICDM (International Conferences on Data Mining) series. These move around the world, with every second year in the United States. For details see the ICDM website at `http://www.cs.uvm.edu/~icdm`.

Information About Association Rule Mining

A valuable source of information about new methods is the series of international workshops known as *FIMI*, standing for Frequent Itemset Mining Implementations, which are run as part of the annual International Conference on Data Mining organised by the Institution of Electrical and Electronic Engineering. The FIMI website at `http://fimi.cs.helsinki.fi` holds not only a collection of research papers about new techniques but also downloadable implementations of many of them and a collection of standard datasets that researchers can use to test their own algorithms.

D
Glossary and Notation

$a < b$	a is less than b
$a \leq b$	a is less than or equal to b
$a > b$	a is greater than b
$a \geq b$	a is greater than or equal to b
a_i	i is a *subscript*. Subscript notation is explained in Appendix A
$\sum_{i=1}^{N} a_i$	The sum $a_1 + a_2 + a_3 + \cdots + a_N$
$\sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij}$	The sum $a_{11} + a_{12} + \cdots + a_{1M} + a_{21} + a_{22} + \cdots + a_{2M} + \ldots + a_{N1} + a_{N2} + \cdots + a_{NM}$
$\prod_{j=1}^{M} b_j$	The product $b_1 \times b_2 \times b_3 \times \cdots \times b_M$
$P(E)$	The **probability of event** E occurring (a number from 0 to 1 inclusive)
$P(E \mid x = a)$	The **probability of event** E occurring *given that* variable x has value a (a **conditional probability**)
$\log_2 X$	Logarithm to base 2 of X. Logarithms are explained in Appendix A
$dist(X, Y)$	The distance between two points X and Y
Z_{CL}	In Chapter 6, the number of **standard errors** needed for a **confidence level** of CL

$a \pm b$	Generally 'a plus or minus b', e.g. 6 ± 2 denotes a number from 4 to 8 inclusive. In Chapter 6, $a \pm b$ is used to indicate that a **classifier** has a **predictive accuracy** of a with **standard error** b.
N_{LEFT}	The number of **instances** matching the left-hand side of a rule
N_{RIGHT}	The number of **instances** matching the right-hand side of a rule
N_{BOTH}	The number of **instances** matching both sides of a rule
N_{TOTAL}	The total number of **instances** in a dataset
$\{\}$	The 'brace' characters that enclose the elements of a set, e.g. {apples, oranges, bananas}
$\emptyset$	The **empty set**. Also written as $\{\}$
$A \cup B$	The **union** of sets A and B. The set that contains all the elements that occur either in A or B or both.
$A \cap B$	The **intersection** of two sets A and B. The set that includes all the elements (if there are any) that occur in both A and B
$A \subseteq B$	A is a **subset** of B, i.e. every element in A also occurs in B.
$A \subset B$	A is a **strict subset** of B, i.e. A is a **subset** of B and A contains fewer elements than B
$A \supseteq B$	A is a **superset** of B. True if and only if B is a **subset** of A
$A \supset B$	A is a **strict superset** of B. True if and only if B is a **strict subset** of A
$\text{count}(S)$	The **support count** of **itemset** S. See Chapter 13
$\text{support}(S)$	The **support** of **itemset** S. See Chapter 13
$cd \rightarrow e$	In **Association Rule Mining** used to denote the rule 'if we know that items c and d were bought, predict that item e was also bought'. See Chapter 13
L_k	The set containing all **supported itemsets** with **cardinality** k. See Chapter 13
C_k	A **candidate set** containing **itemsets** of **cardinality** k. See Chapter 13
$L \rightarrow R$	Denotes a rule with **antecedent** L and **consequent** R
$\text{confidence}(L \rightarrow R)$	The **confidence** of the rule $L \rightarrow R$
${}_kC_i$	Represents the value $\frac{k!}{(k-i)!i!}$ (The number of ways of selecting i values from k, when the order in which they are selected is unimportant)

'a posteriori' probability Another name for **posterior probability**

'a priori' probability Another name for **prior probability**

Abduction A type of reasoning. See Section 3.3

Adequacy Condition (for **TDIDT** algorithm) The condition that no two **instances** with the same values of all the **attributes** may belong to different **classes**

Agglomerative Hierarchical Clustering A widely used method of **clustering**

Antecedent of a Rule The 'if' part (left-hand side) of an *IF... THEN* rule

Apriori Algorithm An algorithm for **Association Rule Mining**. See Chapter 13

Association Rule A rule representing a relationship amongst the values of **variables**. A general form of rule, where a conjunction of *attribute = value* terms can occur on both the left- and the right-hand side

Association Rule Mining (**ARM**) The process of extracting **association rules** from a given **dataset**

Attribute An alternative name for **variable**, used in some areas of **data mining**.

Attribute Selection In this book, generally used to mean the selection of an **attribute** for splitting on when generating a **decision tree**

Attribute Selection Strategy An algorithm for **attribute selection**

Automatic Rule Induction Another term for **Rule Induction**

Average-link Clustering For **hierarchical clustering**, a method of calculating the distance between two **clusters** using the average distance from any member of one cluster to any member of the other

Backed-up Error Rate Estimate (at a **node** in a **decision tree**) An estimate based on the estimated **error rates** of the nodes below it in the tree

Backward Pruning Another name for **post-pruning**

Bag-of-Words Representation A word-based representation of a text document

Bigram A combination of two consecutive characters in a text document

Binary Variable A type of **variable**. See Section 1.2

Bit (short for 'binary digit') The basic unit of information. It corresponds to a switch being open or closed or an electric current flowing or not flowing

Body of a Rule Another name for rule **antecedent**

Branch (of a **decision tree**) The path from the **root node** of a **tree** to any of its **leaf nodes**

Candidate Set A set containing **itemsets** of **cardinality** k that includes all the **supported itemsets** of that cardinality and possibly also some non-supported ones

Cardinality of a Set The number of members of the **set**

Categorical Attribute An **attribute** that can only take one of a number of distinct values, such as 'red', 'blue', 'green'.

Centroid of a Cluster The 'centre' of a **cluster**

ChiMerge An algorithm for **global discretisation**. See Section 7.4

Chi Square Test A statistical test used as part of the **ChiMerge algorithm**

City Block Distance. Another name for **Manhattan distance**

Clash (in a **training set**) A situation where two or more of the **instances** in a training set have identical **attribute** values but different **classifications**

Clash Set A set of **instances** in a **training set** associated with a **clash**

Clash Threshold A middle approach between the 'delete branch' and the 'majority voting' strategies for dealing with **clashes** when generating a **decision tree**. See Chapter 8

Class One of a number of mutually **exclusive and exhaustive categories** to which **objects** are assigned by a **classification** process or algorithm

Classification

1. A process of dividing up **objects** so that each object is assigned to one of a number of **mutually exclusive and exhaustive categories** known as **classes**
2. For **labelled data** the classification is the value of a specially designated **categorical attribute**. The aim is frequently to predict the classification for one or more **unseen instances**
3. **Supervised learning** where the designated attribute has **categorical** values

Classification Rules A set of **rules** that can be used to predict the **classification** of an **unseen instance**

Classification Tree A way of representing a set of **classification rules**

Classifier Any algorithm that assigns a **classification** to **unseen instances**

Cluster A group of **objects** that are similar to one another and (relatively) dissimilar to those in other clusters

Clustering Grouping together **objects** (e.g. **instances** in a **dataset**) that are similar to each other and (relatively) dissimilar to the objects belonging to other **clusters**

Complete-link Clustering For **hierarchical clustering**, a method of calculating the distance between two **clusters** using the longest distance from any member of one cluster to any member of the other

Completeness A **rule interestingness measure**

Conditional Probability The probability of an event occurring given that we have additional information (as well as its observed frequency in a series of trials).

Confidence Level The probability with which we know (or wish to know) the interval in which the **predictive accuracy** of a **classifier** lies

Confidence of a Rule The **predictive accuracy** of a rule (a **rule interestingness measure**)

Confident Itemset An **itemset** on the right-hand side of an **association rule** for which the value of **confidence** is greater than or equal to a minimum threshold value

Conflict Resolution Strategy A strategy for deciding which **rule** or rules to give priority when two or more **rules fire** for a given **instance**

Confusion Matrix A tabular way of illustrating the performance of a classifier. The table shows the number of times each combination of predicted and actual **classifications** occurred for a given **dataset**

Consequent of a Rule The 'then' part (right-hand side) of an *IF... THEN* rule

Continuous Attribute An **attribute** that takes numerical values

Count of an Itemset Another name for **support count of an itemset**

Cross-entropy An alternative name for ***j*-measure**

Cut Point An end point of one of a number of non-overlapping ranges into which the values of a **continuous attribute** are split

Cut Value Another name for **cut point**

Data Compression Converting the data in a **dataset** to a more compact form such as a **decision tree**

Data Mining The central data processing stage of **Knowledge Discovery**. See Introduction

Dataset The complete set of data available for an application. Datasets are divided into **instances** or **records**. A dataset is often represented by a table, with each row representing an **instance** and each column containing the values of one of the **variables** (**attributes**) for each of the instances

Decision Rule Another term for **classification rule**

Decision Tree Another name for a **classification tree**

Decision Tree Induction Another term for **tree induction**

Deduction A type of reasoning. See Section 3.3

Dendrogram A graphical representation of **agglomerative hierarchical clustering**

Depth Cutoff A possible criterion for **pre-pruning** a **decision tree**

Dictionary (for **text classification**) See **Local Dictionary** and **Global Dictionary**

Dimension The number of **attributes** recorded for each **instance**

Dimension Reduction An alternative term for **feature reduction**

Discretisation The conversion of a continuous attribute to one with a discrete set of values, i.e. a **categorical attribute**

Discriminability A **rule interestingness measure**

Disjoint Sets Sets with no common members

Disjunct One of a set of rules in **disjunctive normal form**

Disjunctive Normal Form (DNF) A rule is in disjunctive normal form if it comprises a number of terms of the form *variable = value* (or *variable* $\neq$ *value*) joined by the logical 'and' operator. For example the rule *IF x = 1 AND y = 'yes' AND z = 'good' THEN class = 6* is in DNF

Distance-based Clustering Algorithm A method of **clustering** that makes use of a measure of the distance between two **instances**

Distance Measure A means of measuring the similarity between two **instances**. The smaller the value, the greater the similarity

Dot Product (of two **unit vectors**) The sum of the products of the corresponding pairs of component values

Downward Closure Property of Itemsets The property that if an **itemset** is **supported**, all its (non-empty) subsets are also supported

Eager Learning For **classification** tasks, a form of learning where the **training data** is generalised into a representation (or model) such as a table of probabilities, a **decision tree** or a neural net without waiting for an **unseen instance** to be presented for classification. See **Lazy Learning**

Empty set A **set** with no elements, written as $\emptyset$ or $\{\}$

Entropy An information-theoretic measure of the 'uncertainty' of a **training set**, due to the presence of more than one **classification**. See Chapters 4 and 9

Entropy Method of Attribute Selection (when constructing a **decision tree**) Choosing to split on the **attribute** that gives the greatest value of **Information Gain**. See Chapter 4

Entropy Reduction Equivalent to **information gain**

Equal Frequency Intervals Method A method of **discretising** a **continuous attribute**

Equal Width Intervals Method A method of **discretising** a **continuous attribute**

Error Rate The 'reverse' of the **predictive accuracy** of a **classifier**. A predictive accuracy of 0.8 (i.e. 80%) implies an error rate of 0.2 (i.e. 20%)

Euclidean Distance Between Two Points A widely used measure of the distance between two points.

Exact Rule One for which the value of **confidence** is 1

Exclusive Clustering Algorithm A **clustering** algorithm that places each **object** in precisely one of a set of **clusters**

F1 Score A performance measure for a **classifier**

False Alarm Rate Another name for **false positive rate**

False Negative Classification The classification of an **unseen instance** as negative, when it is actually positive

False Negative Rate of a Classifier The proportion of positive instances that are classified as negative

False Positive Classification The classification of an **unseen instance** as positive, when it is actually negative

False Positive Rate of a Classifier The proportion of negative instances that are classified as positive

Feature Another name for **attribute**

Feature Reduction The reduction of the number of **features** (i.e. **attributes** or **variables**) for each **instance** in a **dataset**. The discarding of relatively unimportant **attributes**.

Feature Space For **text classification**, the set of words included in the **dictionary**

Forward Pruning Another name for **pre-pruning**

Frequency Table A table used for **attribute selection** for the **TDIDT algorithm**. It gives the number of occurrences of each **classification** for each value of an **attribute**. See Chapter 5. (The term is used in a more general sense in Chapter 10.)

Frequent Itemset Another name for **supported itemset**

Gain Ratio A measure used for **attribute selection** for the **TDIDT algorithm**. See Chapter 5

Generalised Rule Induction (GRI) Another name for **Association Rule Mining**

Generalising a Rule Making a rule apply to more **instances** by deleting one or more of its **terms**

Gini Index of Diversity A measure used for **attribute selection** for the **TDIDT Algorithm**. See Chapter 5

Global Dictionary In **text classification** a dictionary that contains all the words that occur at least once in any of the documents under consideration. See **Local Dictionary**

Global Discretisation A form of **discretisation** where each **continuous attribute** is converted to a **categorical attribute** once and for all before any **data mining** algorithm is applied

Head of a Rule Another name for rule **consequent**

Hierarchical Clustering In this book, another name for **Agglomerative Hierarchical Clustering**

Hit Rate Another name for **true positive rate**

Hypertext Categorisation The automatic classification of web documents into predefined categories

Hypertext Classification Another name for **hypertext categorisation**

'Ignore' Attribute An **attribute** that is of no significance for a given application

Induction A type of reasoning. See Section 3.3

Inductive Bias A preference for one algorithm, formula etc. over another that is not determined by the data itself. Inductive bias is unavoidable in any inductive learning system

Information Gain When constructing a **decision tree** by **splitting on attributes**, information gain is the difference between the **entropy** of a node and the weighted average of the entropies of its immediate descendants. It can be shown that the value of information gain is always positive or zero

Instance One of the stored examples in a **dataset**. Each **instance** comprises the values of a number of **variables**, which in **data mining** are often called **attributes**

Integer Variable A type of **variable**. See Section 1.2

Internal Node (of a **tree**) A **node** of a tree that is neither a **root node** nor a **leaf node**

Intersection (of two **sets**) The intersection of two sets A and B, written as $A \cap B$, is the set that includes all the elements (if there are any) that occur in both of the sets

Interval-scaled Variable A type of **variable**. See Section 1.2

Invalid Value An **attribute** value that is invalid for a given dataset. See **Noise**

Item For **Market Basket Analysis**, each item corresponds to one of the purchases made by a customer, e.g. bread or milk. We are not usually concerned with items that were not purchased

Itemset For **Market Basket Analysis**, a set of **items** purchased by a customer, effectively the same as a **transaction**. Itemsets are generally written in list notation, e.g. {fish, cheese, milk}

***J*-Measure** A **rule interestingness measure** that quantifies the information content of a rule

***j*-Measure** A value used in calculating the ***J*-measure** of a rule

Jack-knifing Another name for ***N*-fold cross-validation**

***k*-fold Cross-validation** A strategy for estimating the performance of a **classifier**

***k*-Means Clustering** A widely used method of **clustering**

***k*-Nearest Neighbour Classification** A method of classifying an **unseen instance** using the **classification** of the **instance** or instances closest to it (see Chapter 2)

Knowledge Discovery The non-trivial extraction of implicit, previously unknown and potentially useful information from data. See Introduction

Labelled Data Data where each **instance** has a specially designated **attribute** which can be either **categorical** or **continuous**. The aim is generally to predict its value. See **Unlabelled Data**

Large Itemset Another name for **Supported Itemset**

Lazy Learning For **classification** tasks, a form of learning where the **training data** is left unchanged until an **unseen instance** is presented for classification. See **Eager Learning**

Leaf Node A **node** of a **tree** which has no other **nodes** descending from it

Leave-one-out Cross-validation Another name for ***N*-fold cross-validation**

Length of a Vector The square root of the sum of the squares of its component values. See **Unit Vector**

Leverage A rule interestingness measure

Lift A rule interestingness measure

Local Dictionary In **text classification** a dictionary that contains only those words that occur in the documents under consideration that are classified as being in a specific category. See **Global Dictionary**

Local Discretisation A form of **discretisation** where each **continuous attribute** is converted to a **categorical attribute** at each stage of the **data mining** process

Logarithm Function See Appendix A

Manhattan Distance A measure of the distance between two points

Market Basket Analysis A special form of **Association Rule Mining**. See Chapter 13

Matches An **itemset** matches a **transaction** if all the items in the former are also in the latter

Maximum Dimension Distance A measure of the distance between two points.

Missing Branches An effect that can occur during the generation of a **decision tree** that makes the tree unable to classify certain **unseen instances**. See Section 5.6

Missing Value An **attribute** value that is not recorded

Model-based Classification Algorithm One that gives an explicit representation of the **training data** (in the form of a **decision tree**, **set of rules** etc.) that can be used to classify **unseen instances** without reference to the training data itself

Mutually Exclusive and Exhaustive Categories A set of categories chosen so that each **object** of interest belongs to precisely one of the categories

Mutually Exclusive and Exhaustive Events A set of events, one and only one of which must always occur

***n*-dimensional Space** A point in n-dimensional space is a graphical way of representing an **instance** with n **attribute** values

***N*-dimensional Vector** In **text classification**, a way of representing a **labelled instance** with N **attributes** by its N attribute values (or other values derived from them), enclosed in parentheses and separated by commas, e.g. (2, yes, 7, 4, no). The **classification** is not generally included

***N*-fold Cross-validation** A strategy for estimating the performance of a **classifier**

Naïve Bayes Algorithm A means of combining **prior and conditional probabilities** to calculate the probability of alternative **classifications**. See Chapter 2

Naïve Bayes Classification A method of classification that uses Mathematical probability theory to find the most likely classification for an **unseen instance**

Nearest Neighbour Classification See ***k*-Nearest Neighbour Classification**

Node (of a **decision tree**) A **tree** consists of a collection of points, called **nodes**, joined by straight lines, called *links*. See Appendix A.2

Noise An **attribute** value that is valid for a given dataset, but is incorrectly recorded. See **Invalid Value**

Nominal Variable A type of **variable**. See Section 1.2

Normalisation (of an **Attribute**) Adjustment of the values of an **attribute**, generally to make them fall in a specified range such as 0 to 1

Normalised Vector Space Model A **vector space model** where the components of a **vector** are adjusted so that the **length** of each vector is 1

Numerical Prediction Supervised learning where the designated **attribute** has a numerical value. Also called *regression*

Object One of a **universe of objects**. It is described by the values of a number of **variables** that correspond to its properties

Objective Function For **clustering**, a measure of the quality of a set of **clusters**

Order of a Rule The number of terms in the **antecedent** of a rule in **disjunctive normal form**

Ordinal Variable A type of **variable**. See Section 1.2

Overfitting A **classification** algorithm is said to overfit to the **training data** if it generates a **decision tree**, set of **classification rules** or any other representation of the data that depends too much on irrelevant features of the training instances, with the result that it performs well on the training data but relatively poorly on **unseen instances**. See Chapter 8

Piatetsky-Shapiro Criteria Criteria that it has been proposed should be met by any **rule interestingness measure**

Positive Predictive Value Another name for **precision**

Post-pruning a Decision Tree Removing parts of a **decision tree** that has already been generated, with the aim of reducing **overfitting**

Posterior Probability The probability of an event occurring given additional information that we have

Pre-pruning a Decision Tree Generating a **decision tree** with fewer **branches** than would otherwise be the case, with the aim of reducing **overfitting**

Precision A performance measure for a **classifier**

Prediction Using the data in a **training set** to predict (as far as this book is concerned) the **classification** for one or more previously **unseen instances**

Predictive Accuracy For **classification** applications, the proportion of a set of **unseen instances** for which the correct **classification** is predicted. A **rule interestingness** measure, also known as **confidence**

Prior Probability The probability of an event occurring based solely on its observed frequency in a series of trials, without any additional information

Prism An algorithm for inducing **classification rules** directly, without using the intermediate representation of a **decision tree**

Probability of an Event The proportion of times we would expect an event to occur over a long series of trials

Pruned Tree A **tree** to which **pre-pruning** or **post-pruning** has been applied

Pruning Set Part of a **dataset** used during **post-pruning** of a **decision tree**

Pseudo-attribute A test on the value of a **continuous attribute**, e.g. $A < 35$. This is effectively the same as a **categorical attribute** that has only two values: true and false

Ratio-scaled Variable A type of **variable**. See Section 1.2

Recall Another name for **true positive rate**

Receiver Operating Characteristics Graph The full name for **ROC Graph**

Record Another term for **instance**

Recursive Partitioning Generating a **decision tree** by repeatedly **splitting on the values of attributes**

Reliability A **rule interestingness** measure. Another name for **confidence**

RI Measure A **rule interestingness measure**

ROC Curve A **ROC Graph** on which related points are joined together to form a curve

ROC Graph A diagrammatic way of representing the **true positive rate** and **false positive rate** of one or more **classifiers**

Root Node The top-most **node** of a **tree**. The starting node for every **branch**

Rule The statement of a relationship between a condition, known as the **antecedent**, and a conclusion, known as the **consequent**. If the condition is satisfied, the conclusion follows

Rule Fires The **antecedent** of the rule is satisfied for a given **instance**

Rule Induction The automatic generation of rules from examples

Rule Interestingness Measure A measure of the importance of a rule

Ruleset A collection of rules

Search Space In Chapter 12, the set of possible rules of interest

Search Strategy A method of examining the contents of a **search space** (usually in an efficient order)

Sensitivity Another name for **true positive rate**

Set An unordered collection of items, known as *elements*. See Appendix A. The elements of a set are often written between 'brace' characters and separated by commas, e.g. {apples, oranges, bananas}

Single-link Clustering For **hierarchical clustering**, a method of calculating the distance between two **clusters** using the shortest distance from any member of one cluster to any member of the other

Size Cutoff A possible criterion for **pre-pruning** a **decision tree**

Specialising a Rule Making a rule apply to fewer **instances** by adding one or more additional **terms**

Specificity Another name for **true negative rate**

Split Information A value used in the calculation of **Gain Ratio**. See Chapter 5

Split Value A value used in connection with **continuous attributes** when **splitting on an attribute** to construct a **decision tree**. The test is normally whether the value is 'less than or equal to' or 'greater than' the split value

Splitting on an Attribute (while constructing a **decision tree**) Testing the value of an **attribute** and then creating a branch for each of its possible values

Standard Error (associated with a value) A statistical estimate of the reliability of the value. See Section 6.2.1

Static Error Rate Estimate (at a **node** in a **decision tree**) An estimate based on the **instances** corresponding to the node, as opposed to a **backed-up estimate**

Stemming Converting a word to its linguistic root (e.g. 'computing', 'computer' and 'computation' to 'comput')

Stop Words Common words that are unlikely to be useful for **text classification**

Strict Subset A **set** A is a strict subset of a set B, written as $A \subset B$, if A is a subset of B and A contains fewer elements than B

Strict Superset A **set** A is a strict superset of a set B, written as $A \supset B$, if and only if B is a **strict subset** of A

Subset A **set** A is a subset of a set B, written as $A \subseteq B$, if every element in A also occurs in B

Subtree The part of a **tree** that descends from (or 'hangs from') one of its **nodes** A (including node A itself). A subtree is a tree in its own right, with its own **root node** (A) etc. See Appendix A.2

Superset A **set** A is a superset of a set B, written as $A \supseteq B$, if and only if B is a **subset** of A

Supervised Learning A form of **Data Mining** using **labelled data**

Support Count of an Itemset For **Market Basket Analysis**, the number of **transactions** in the database matched by the **itemset**

Support of a Rule The proportion of the database to which the rule successfully applies (a **rule interestingness measure)**

Support of an Itemset The proportion of **transactions** in the database that are matched by the **itemset**

Supported Itemset An **itemset** for which the **support** value is greater than or equal to a minimum threshold value

Symmetry condition (for a **distance measure**) The distance from point A to point B is the same as the distance from point B to point A

TDIDT An abbreviation for *Top-Down Induction of Decision Trees*. See Chapter 3

Term In this book, a component of a rule. A term takes the form *variable* = *value*. See **Disjunctive Normal Form**

Term Frequency In **text classification**, the number of occurrences of a term in a given document

Test Set A collection of **unseen instances**

Text Classification A particular type of **classification**, where the **objects** are text documents such as articles in newspapers, scientific papers etc. See also **Hypertext Categorisation**

TFIDF (Term Frequency Inverse Document Frequency) In **text classification**, a measure combining the frequency of a term with its rarity in a set of documents

Top Down Induction of Decision Trees A widely-used algorithm for **classification**. See Chapter 3

Train and Test A strategy for estimating the performance of a **classifier**

Training Data Another name for **training set**

Training Set A **dataset** or part of a dataset that is used for purposes of **classification**

Transaction Another name for **record** or **instance**, generally used when the application is **Market Basket Analysis**. A transaction generally represents a set of **items** bought by a customer

Tree A structure used to represent data items and the processes applied to them. See Appendix A.2

Tree Induction Generating **decision rules** in the implicit form of a **decision tree**

Triangle Inequality (for a **distance measure**) A condition corresponding to the idea that 'the shortest distance between any two points is a straight line'

Trigram A combination of three consecutive characters in a text document

True Negative Classification The correct classification of an **unseen instance** as negative

True Negative Rate of a Classifier The proportion of negative instances that are classified as negative

True Positive Classification The correct classification of an **unseen instance** as positive

True Positive Rate of a Classifier The proportion of positive instances that are classified as positive

Two-dimensional Space See ***n*-dimensional Space**

Type 1 Error Another name for **false positive classification**

Type 2 Error Another name for **false negative classification**

UCI Repository The library of datasets maintained by the University of California at Irvine. See Section 1.6

Unconfident Itemset An **itemset** which is not **confident**

Union of Two Sets The set of items that occur in either or both of the sets

Unit Vector A **vector** of length 1

Universe of Objects See Section 1.1

Unlabelled Data Data where each instance has no specially designated **attribute**. See **Labelled Data**

Unseen Instance An instance that does not occur in a **training set**. We frequently want to predict the **classification** of one or more unseen instances. See also **Test Set**

Unseen Test Set Another term for **test set**

Unsupervised Learning A form of **Data Mining** using **unlabelled data**

Variable One of the properties of an **object** in a **universe of objects**

Vector In **text classification**, another name for N**-dimensional vector**

Vector Space Model (VSM) The complete set of **vectors** corresponding to a set of documents under consideration. See N**-dimensional vector**

E
Solutions to Self-assessment Exercises

Self-assessment Exercise 1

Question 1

Labelled data has a specially designated attribute. The aim is to use the data given to predict the value of that attribute for instances that have not yet been seen. Data that does not have any specially designated attribute is called unlabelled.

Question 2

Name: Nominal
Date of Birth: Ordinal
Sex: Binary
Weight: Ratio-scaled
Height: Ratio-scaled
Marital Status: Nominal (assuming that there are more than two values, e.g. single, married, widowed, divorced)
Number of Children: Integer

Question 3

– Discard all instances where there is at least one missing value and use the remainder.

– Estimate missing values of each categorical attribute by its most frequently occurring value in the training set and estimate missing values of each continuous attribute by the average of its values for the training set.

Self-assessment Exercise 2

Question 1

Using the values in Figure 2.2, the probability of each class for the unseen instance

weekday	summer	high	heavy	????

is as follows.
class = on time
$0.70 \times 0.64 \times 0.43 \times 0.29 \times 0.07 = 0.0039$
class = late
$0.10 \times 0.5 \times 0 \times 0.5 \times 0.5 = 0$
class = very late
$0.15 \times 1 \times 0 \times 0.33 \times 0.67 = 0$
class = cancelled
$0.05 \times 0 \times 0 \times 1 \times 1 = 0$
The largest value is for class = on time
The probability of each class for the unseen instance

sunday	summer	normal	slight	????

is as follows.
class = on time
$0.70 \times 0.07 \times 0.43 \times 0.36 \times 0.57 = 0.0043$
class = late
$0.10 \times 0 \times 0 \times 0.5 \times 0 = 0$
class = very late
$0.15 \times 0 \times 0 \times 0.67 \times 0 = 0$
class = cancelled
$0.05 \times 0 \times 0 \times 0 \times 0 = 0$
The largest value is for class = on time

Question 2

The distance of the first instance in Figure 2.5 from the unseen instance is the square root of $(0.8 - 9.1)^2 + (6.3 - 11.0)^2$, i.e. 9.538.

The distances for the 20 instances are given in the table below.

Attribute 1	Attribute 2	Distance	
0.8	6.3	9.538	
1.4	8.1	8.228	

2.1	7.4	7.871	
2.6	14.3	7.290	
6.8	12.6	2.802	*
8.8	9.8	1.237	*
9.2	11.6	0.608	*
10.8	9.6	2.202	*
11.8	9.9	2.915	*
12.4	6.5	5.580	
12.8	1.1	10.569	
14.0	19.9	10.160	
14.2	18.5	9.070	
15.6	17.4	9.122	
15.8	12.2	6.807	
16.6	6.7	8.645	
17.4	4.5	10.542	
18.2	6.9	9.981	
19.0	3.4	12.481	
19.6	11.1	10.500	

The five nearest neighbours are marked with asterisks in the rightmost column.

Self-assessment Exercise 3

Question 1

No two instances with the same values of all the attributes may belong to different classes.

Question 2

The most likely cause is probably noise or missing values in the training set.

Question 3

Provided the adequacy condition is satisfied the TDIDT algorithm is guaranteed to terminate and give a decision tree corresponding to the training set.

Question 4

A situation will be reached where a branch has been generated to the maximum length possible, i.e. with a term for each of the attributes, but the corresponding subset of the training set still has more than one classification.

Self-assessment Exercise 4

Question 1

(a) The proportions of instances with each of the two classifications are 6/26 and 20/26. So $E_{start} = -(6/26)\log_2(6/26) - (20/26)\log_2(20/26) = 0.7793$.

(b) The following shows the calculations.

Splitting on SoftEng
SoftEng = A
Proportions of each class: FIRST 6/14, SECOND 8/14
Entropy = $-(6/14)\log_2(6/14) - (8/14)\log_2(8/14) = 0.9852$
SoftEng = B
Proportions of each class: FIRST 0/12, SECOND 12/12
Entropy = 0 [all the instances have the same classification]
Weighted average entropy $E_{new} = (14/26) \times 0.9852 + (12/26) \times 0 = 0.5305$
Information Gain = $0.7793 - 0.5305 = 0.2488$

Splitting on ARIN
ARIN = A
Proportions of each class: FIRST 4/12, SECOND 8/12
Entropy = 0.9183
ARIN = B
Proportions of each class: FIRST 2/14, SECOND 12/14
Entropy = 0.5917
Weighted average entropy $E_{new} = (12/26) \times 0.9183 + 14/26 \times 0.5917 = 0.7424$
Information Gain = $0.7793 - 0.7424 = 0.0369$

Splitting on HCI
HCI = A
Proportions of each class: FIRST 1/9, SECOND 8/9
Entropy = 0.5033
HCI = B
Proportions of each class: FIRST 5/17, SECOND 12/17
Entropy = 0.8740
Weighted average entropy $E_{new} = (9/26) \times 0.5033 + (17/26) \times 0.8740 = 0.7457$
Information Gain = $0.7793 - 0.7457 = 0.0337$

Splitting on CSA
CSA = A
Proportions of each class: FIRST 3/7, SECOND 4/7
Entropy = 0.9852
CSA = B

Proportions of each class: FIRST 3/19, SECOND 16/19
Entropy = 0.6292
Weighted average entropy $E_{new} = (7/26) \times 0.9852 + (19/26) \times 0.6292 = 0.7251$
Information Gain = $0.7793 - 0.7251 = 0.0543$

Splitting on Project
Project = A
Proportions of each class: FIRST 5/9, SECOND 4/9
Entropy = 0.9911
Project = B
Proportions of each class: FIRST 1/17, SECOND 16/17
Entropy = 0.3228
Weighted average entropy $E_{new} = (9/26) \times 0.9911 + (17/26) \times 0.3228 = 0.5541$
Information Gain = $0.7793 - 0.5541 = 0.2253$
The maximum value of information gain is for attribute SoftEng.

Question 2

The TDIDT algorithm inevitably leads to a decision tree where all nodes have entropy zero. Reducing the average entropy as much as possible at each step would seem like an efficient way of achieving this in a relatively small number of steps. The use of entropy minimisation (or information gain maximisation) appears generally to lead to a small decision tree compared with other attribute selection criteria. The *Occam's Razor* principle suggests that small trees are most likely to be the best, i.e. to have the greatest predictive power.

Self-assessment Exercise 5

Question 1

The frequency table for splitting on attribute SoftEng is as follows.

	Attribute value	
Class	A	B
FIRST	6	0
SECOND	8	12
Total	14	12

Using the method of calculating entropy given in Chapter 5, the value is:
$-(6/26)\log_2(6/26) - (8/26)\log_2(8/26) - (12/26)\log_2(12/26)$
$+(14/26)\log_2(14/26) + (12/26)\log_2(12/26)$
$= 0.5305$

This is the same value as was obtained using the original method for Self-assessment Exercise 1 for Chapter 4. Similar results apply for the other attributes.

Question 2

It was shown previously that the entropy of the chess dataset is: 0.7793.

The value of Gini Index is $1 - (6/26)^2 - (20/26)^2 = 0.3550$.

Splitting on attribute SoftEng

	Attribute value	
Class	A	B
FIRST	6	0
SECOND	8	12
Total	14	12

The entropy is:
$-(6/26)\log_2(6/26) - (8/26)\log_2(8/26) - (12/26)\log_2(12/26)$
$+(14/26)\log_2(14/26) + (12/26)\log_2(12/26)$
$= 0.5305$
The value of split information is $-(14/26)\log_2(14/26) - (12/26)\log_2(12/26)$
$= 0.9957$
The information gain is $0.7793 - 0.5305 = 0.2488$
Gain ratio is $0.2488/0.9957 = 0.2499$

Gini Index Calculation
Contribution for 'SoftEng = A' is $(6^2 + 8^2)/14 = 7.1429$
Contribution for 'SoftEng = B' is $(0^2 + 12^2)/12 = 12$
New value of Gini Index $= 1 - (7.1429 + 12)/26 = 0.2637$

Splitting on attribute ARIN

	Attribute value	
Class	A	B
FIRST	4	2
SECOND	8	12
Total	12	14

The value of entropy is 0.7424
The value of split information is 0.9957
So the information gain is $0.7793 - 0.7424 = 0.0369$
and the gain ratio is $0.0369/0.9957 = 0.0371$
New value of Gini Index = 0.3370

Splitting on attribute HCI

	Attribute value	
Class	A	B
FIRST	1	5
SECOND	8	12
Total	9	17

The value of entropy is 0.7457
The value is split information is 0.9306
So the information gain is $0.7793 - 0.7457 = 0.0337$
and the gain ratio is $0.0336/0.9306 = 0.0362$
New value of Gini Index = 0.3399

Splitting on attribute CSA

	Attribute value	
Class	A	B
FIRST	3	3
SECOND	4	16
Total	7	19

The value of entropy is 0.7251
The value is split information is 0.8404
So the information gain is $0.7793 - 0.7251 = 0.0543$
and the gain ratio is $0.0542/0.8404 = 0.0646$
New value of Gini Index = 0.3262

Splitting on attribute Project

	Attribute value	
Class	A	B
FIRST	5	1
SECOND	4	16
Total	9	17

The value of entropy is 0.5541
The value of split information is 0.9306
So the information gain is $0.7793 - 0.5541 = 0.2253$
and the gain ratio is $0.2252/0.9306 = 0.2421$
New value of Gini Index = 0.2433
The largest value of Gain Ratio is when the attribute is SoftEng.

The largest value of Gini Index reduction is for attribute Project.
The reduction is $0.3550 - 0.2433 = 0.1117$.

Question 3

Any dataset for which there is an attribute with a large number of values is a possible answer, e.g. one that contains a 'nationality' attribute or a 'job title' attribute. Using Gain Ratio will probably ensure that such attributes are not chosen.

Self-assessment Exercise 6

Question 1

***vote* Dataset, Figure 6.14**

The number of correct predictions is 127 and the total number of instances is 135.

We have $p = 127/135 = 0.9407$, $N = 135$, so the standard error is $\sqrt{p \times (1-p)/N} = \sqrt{0.9407 \times 0.0593/135} = 0.0203$.

The value of the predictive accuracy can be expected to lie in the following ranges:

probability 0.90: from $0.9407 - 1.64 \times 0.0203$ to $0.9407 + 1.64 \times 0.0203$, i.e. from 0.9074 to 0.9741

probability 0.95: from $0.9407 - 1.96 \times 0.0203$ to $0.9407 + 1.96 \times 0.0203$, i.e. from 0.9009 to 0.9806

probability 0.99: from $0.9407 - 2.58 \times 0.0203$ to $0.9407 + 2.58 \times 0.0203$, i.e. from 0.8883 to 0.9932

***glass* Dataset, Figure 6.15**

The number of correct predictions is 149 and the total number of instances is 214.

We have $p = 149/214 = 0.6963$, $N = 214$, so the standard error is $\sqrt{p \times (1-p)/N} = \sqrt{0.6963 \times 0.3037/214} = 0.0314$.

The value of the predictive accuracy can be expected to lie in the following ranges:

probability 0.90: from $0.6963 - 1.64 \times 0.0314$ to $0.6963 + 1.64 \times 0.0314$, i.e. from 0.6447 to 0.7478

probability 0.95: from $0.6963 - 1.96 \times 0.0314$ to $0.6963 + 1.96 \times 0.0314$, i.e. from 0.6346 to 0.7579

probability 0.99: from $0.6963 - 2.58 \times 0.0314$ to $0.6963 + 2.58 \times 0.0314$, i.e. from 0.6152 to 0.7774

Question 2

False positive classifications would be undesirable in applications such as the prediction of equipment that will fail in the near future, which may lead to expensive and unnecessary preventative maintenance. False classifications of individuals as likely criminals or terrorists can have very serious repercussions for the wrongly accused.

False negative classifications would be undesirable in applications such as medical screening, e.g. for patients who may have a major illness requiring treatment, or prediction of catastrophic events such as hurricanes or earthquakes.

Decisions about the proportion of false negative (positive) classifications that would be acceptable to reduce the proportion of false positives (negatives) to zero is a matter of personal taste. There is no general answer.

Self-assessment Exercise 7

Question 1

Sorting the values of *humidity* into ascending numerical order gives the following table.

Humidity (%)	Class
65	play
70	play
70	play
70	don't play
75	play
78	play
80	don't play
80	play
80	play
85	don't play
90	don't play
90	play
95	don't play
96	play

The amended rule for selecting cut points given in Section 7.3.2 is: 'only include attribute values for which the class value is different from that for the previous attribute value, together with any attribute which occurs more than once and the attribute immediately following it'.

This rule gives the cut points for the *humidity* attribute as all the values in the above table except 65 and 78.

Question 2

Figure 7.12(c) is reproduced below.

Value of A	Frequency for class			Total	Value of χ^2
	$c1$	$c2$	$c3$		
1.3	1	0	4	5	3.74
1.4	1	2	1	4	5.14
2.4	6	0	2	8	3.62
6.5	3	2	4	9	4.62
8.7	6	0	1	7	1.89
12.1	7	2	3	12	1.73
29.4	0	0	1	1	3.20
56.2	2	4	0	6	6.67
87.1	0	1	3	4	1.20
89.0	1	1	2	4	
Total	27	12	21	60	

After the 87.1 and 89.0 rows are merged, the figure looks like this.

Value of A	Frequency for class			Total	Value of χ^2
	$c1$	$c2$	$c3$		
1.3	1	0	4	5	3.74
1.4	1	2	1	4	5.14
2.4	6	0	2	8	3.62
6.5	3	2	4	9	4.62
8.7	6	0	1	7	1.89
12.1	7	2	3	12	1.73
29.4	0	0	1	1	3.20
56.2	2	4	0	6	**6.67**
87.1	1	2	5	8	
Total	27	12	21	60	

The previous values of χ^2 are shown in the rightmost column. Only the one given in bold can have been changed by the merging process, so this value needs to be recalculated.

For the adjacent intervals labelled 56.2 and 87.1 the values of O and E are as follows.

Value of *A*	**Frequency for class**						**Total observed**
	***c*1**		***c*2**		***c*3**		
	O	***E***	***O***	***E***	***O***	***E***	
56.2	2	1.29	4	2.57	0	2.14	6
87.1	1	1.71	2	3.43	5	2.86	8
Total	3		6		5		14

The O (observed) values are taken from the previous figure. The E (expected) values are calculated from the row and column sums. Thus for row 56.2 and class $c1$, the expected value E is $3 \times 6/14 = 1.29$.

The next step is to calculate the value of $(O - E)^2/E$ for each of the six combinations. These are shown in the Val columns in the figure below.

Value of A	**Frequency for class**									**Total observed**
	***c*1**			***c*2**			***c*3**			
	O	***E***	**Val**	***O***	***E***	**Val**	***O***	***E***	**Val**	
56.2	2	1.29	0.40	4	2.57	0.79	0	2.14	2.14	6
87.1	1	1.71	0.30	2	3.43	0.60	5	2.86	1.61	8
Total	3			6			5			14

The value of χ^2 is then the sum of the six values of $(O - E)^2/E$. For the pair of rows shown the value of χ^2 is 5.83.

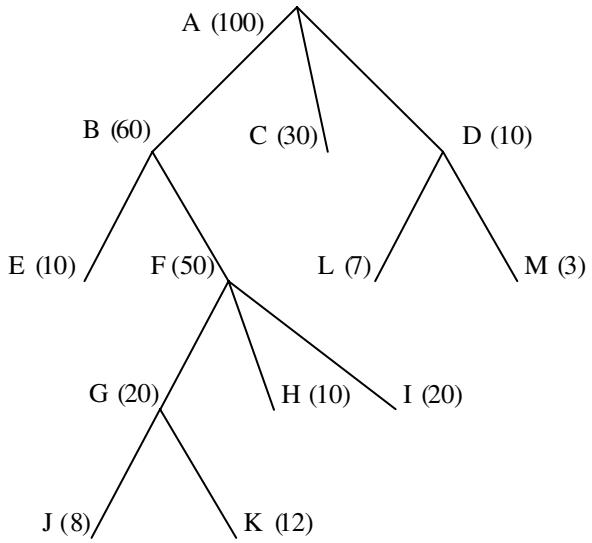
This gives a revised version of the frequency table as follows.

Value of A	Frequency for class			Total	Value of χ^2
	$c1$	$c2$	$c3$		
1.3	1	0	4	5	3.74
1.4	1	2	1	4	5.14
2.4	6	0	2	8	3.62
6.5	3	2	4	9	4.62
8.7	6	0	1	7	1.89
12.1	7	2	3	12	1.73
29.4	0	0	1	1	3.20
56.2	2	4	0	6	5.83
87.1	1	2	5	8	
Total	27	12	21	60	

The smallest value of χ^2 is now 1.73, in the row labelled 12.1. This value is less than the threshold value of 4.61, so the rows (intervals) labelled 12.1 and 29.4 are merged.

Self-assessment Exercise 8

The decision tree shown in Figure 8.8 is reproduced below for ease of reference.

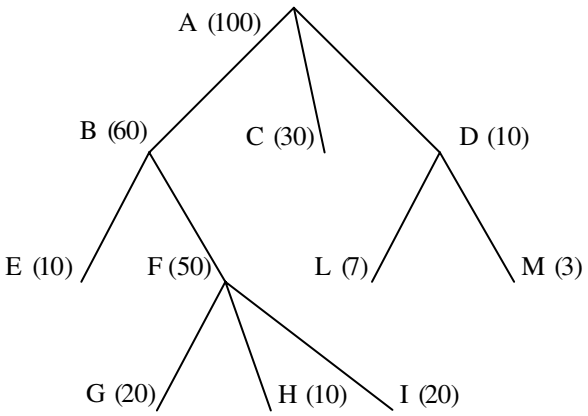


The table of error rates specified in the question is as follows.

Node	Estimated error rate
A	0.2
B	0.35
C	0.1
D	0.2
E	0.01
F	0.25
G	0.05
H	0.1
I	0.2
J	0.15
K	0.2
L	0.1
M	0.1

The post-pruning process starts by considering the possibility of pruning at node G.

The backed-up error rate at that node is $(8/20) \times 0.15 + (12/20) \times 0.2 = 0.18$. This is more than the static error rate, which is only 0.05. This means that splitting at node G increases the error rate at that node so we prune the subtree descending from G, giving the following figure [which is the same as Figure 8.11].



We now consider pruning at node F. The backed-up error rate is $(20/50) \times 0.05 + (10/50) \times 0.1 + (20/50) \times 0.2 = 0.12$. This is less than the static error rate. Splitting at node F reduces the average error rate so we do not prune.

The method given in Chapter 8 specifies that we only consider pruning at nodes that have a descendant subtree of depth one (i.e. all the nodes one level down are leaf nodes).

The only remaining candidate is node D. For this node the backed-up error rate is $(7/10) \times 0.1 + (3/10) \times 0.1 = 0.1$. This is less than the static error rate at the node, so we do not prune.

There are no further candidates for pruning, so the process terminates.

Self-assessment Exercise 9

Question 1

The entropy of a training set depends only on the relative proportions of the classifications, not on the number of instances it contains. Thus for both training sets the answer is the same.

Entropy $= -0.2 \times \log_2 0.2 - 0.3 \times \log_2 0.3 - 0.25 \times \log_2 0.25 - 0.25 \times \log_2 0.25$ $= 1.985$

Question 2

It is best to ask any question that divides the people into two approximately equal halves. An obvious question would be 'Is the person male?'. This might

well be appropriate in a restaurant, a theatre etc. but would not be suitable for a group where there is a large predominance of one sex, e.g. a football match. In such a case a question such as 'Does he or she have brown eyes?' might be better, or even 'Does he or she live in a house or flat with an odd number?'

Self-assessment Exercise 10

The *degrees* dataset given in Figure 3.3 is reproduced below for ease of reference.

SoftEng	ARIN	HCI	CSA	Project	Class
A	B	A	B	B	SECOND
A	B	B	B	A	FIRST
A	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	A	B	B	A	FIRST
B	A	A	B	B	SECOND
A	B	B	B	B	SECOND
A	B	B	B	B	SECOND
A	A	A	A	A	FIRST
B	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	B	B	A	B	SECOND
B	B	B	B	A	SECOND
A	A	B	A	B	FIRST
B	B	B	B	A	SECOND
A	A	B	B	B	SECOND
B	B	B	B	B	SECOND
A	A	B	A	A	FIRST
B	B	B	A	A	SECOND
B	B	A	A	B	SECOND
B	B	B	B	A	SECOND
B	A	B	A	B	SECOND
A	B	B	B	A	FIRST
A	B	A	B	B	SECOND
B	A	B	B	B	SECOND
A	B	B	B	B	SECOND

The Prism algorithm starts by constructing a table showing the probability of class = FIRST occurring for each attribute/value pair over the whole training set of 26 instances.

Attribute/value pair	Frequency for class = FIRST	Total frequency (out of 26 instances)	Probability
SoftEng = A	6	14	0.429
SoftEng = B	0	12	0
ARIN = A	4	12	0.333
ARIN = B	2	14	0.143
HCI = A	1	9	0.111
HCI = B	5	17	0.294
CSA = A	3	7	0.429
CSA = B	3	19	0.158
Project = A	5	9	0.556
Project = B	1	17	0.059

The maximum probability is when Project = A
Incomplete rule induced so far:

IF Project = A THEN class = FIRST

The subset of the training set covered by this incomplete rule is:

SoftEng	ARIN	HCI	CSA	Project	Class
A	B	B	B	A	FIRST
A	A	B	B	A	FIRST
A	A	A	A	A	FIRST
B	B	B	B	A	SECOND
B	B	B	B	A	SECOND
A	A	B	A	A	FIRST
B	B	B	A	A	SECOND
B	B	B	B	A	SECOND
A	B	B	B	A	FIRST

The next table shows the probability of class = FIRST occurring for each attribute/value pair (not involving attribute Project) for this subset.

Attribute/value pair	Frequency for class = FIRST	Total frequency (out of 9 instances)	Probability
SoftEng = A	5	5	1.0
SoftEng = B	0	4	0
ARIN = A	3	3	1.0

ARIN = B	2	6	0.333
HCI = A	1	1	1.0
HCI = B	4	8	0.5
CSA = A	2	3	0.667
CSA = B	3	6	0.5

Three attribute/value combinations give a probability of 1.0. Of these SoftEng = A is based on most instances, so will probably be selected by tie-breaking.

Incomplete rule induced so far:

IF Project = A AND SoftEng = A THEN class = FIRST

The subset of the training set covered by this incomplete rule is:

SoftEng	ARIN	HCI	CSA	Project	Class
A	B	B	B	A	FIRST
A	A	B	B	A	FIRST
A	A	A	A	A	FIRST
A	A	B	A	A	FIRST
A	B	B	B	A	FIRST

This subset contains instances with only one classification, so the rule is complete.

The final induced rule is therefore:

IF Project = A AND SoftEng = A THEN class = FIRST

Self-assessment Exercise 11

The true positive rate is the number of instances that are correctly predicted as positive divided by the number of instances that are actually positive.

The false positive rate is the number of instances that are wrongly predicted as positive divided by the number of instances that are actually negative.

		Predicted class	
		+	−
Actual class	+	50	10
	−	10	30

For the table above the values are:

True positive rate: $50/60 = 0.833$

False positive rate: $10/40 = 0.25$

The Euclidean distance is defined as: $Euc = \sqrt{fprate^2 + (1 - tprate)^2}$

For this table $Euc = \sqrt{(0.25)^2 + (1 - 0.833)^2} = 0.300$.

For the other three tables specified in the Exercise the values are as follows.

Second table

True positive rate: $55/60 = 0.917$
False positive rate: $5/40 = 0.125$
$Euc = 0.150$

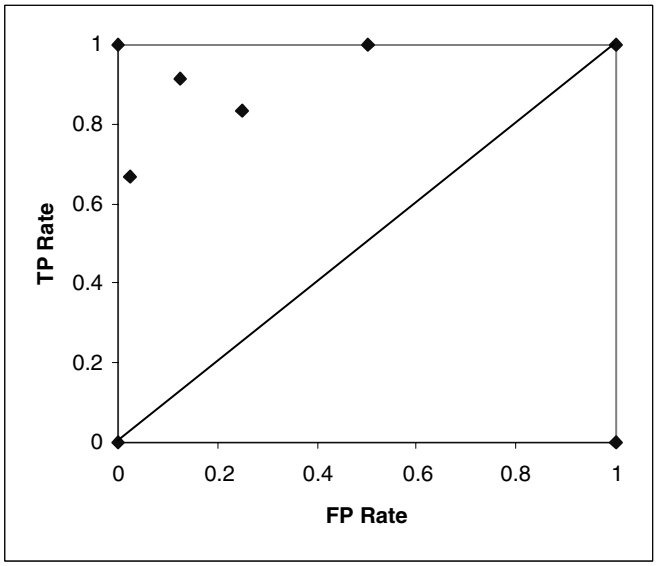
Third table

True positive rate: $40/60 = 0.667$
False positive rate: $1/40 = 0.025$
$Euc = 0.334$

Fourth table

True positive rate: $60/60 = 1.0$
False positive rate: $20/40 = 0.5$
$Euc = 0.500$

The following ROC graph shows the four classifiers as well as the four hypothetical ones at $(0, 0)$, $(1, 0)$, $(0, 1)$ and $(1, 1)$.



If we were equally concerned about avoiding false positive and false negative classifications we should choose the one given in the second table in the Exercise, which has true positive rate 0.917 and false positive rate 0.125. This is the one closest to $(0, 1)$ the perfect classifier in the ROC graph.

Self-assessment Exercise 12

Question 1

Using the formulae for Confidence, Completeness, Support, Discriminability and RI given in Chapter 12, the values for the five rules are as follows.

Rule	Confid.	Complete	Support	Discrim.	RI
1	0.972	0.875	0.7	0.9	124.0
2	0.933	0.215	0.157	0.958	30.4
3	1.0	0.5	0.415	1.0	170.8
4	0.5	0.8	0.289	0.548	55.5
5	0.983	0.421	0.361	0.957	38.0

Question 2

Let us assume that the attribute w has the three values w_1, w_2 and w_3 and similarly for attributes x, y and z.

If we arbitrarily choose attribute w to be on the right-hand side of each rule, there are three possible types of rule:

IF ... THEN $w = w_1$
IF ... THEN $w = w_2$
IF ... THEN $w = w_3$

Let us choose one of these, say the first, and calculate how many possible left-hand sides there are for such rules.

The number of 'attribute = value' terms on the left-hand side can be one, two or three. We consider each case separately.

One term on left-hand side

There are three possible terms: x, y and z. Each has three possible values, so there are $3 \times 3 = 9$ possible left-hand sides, e.g.

IF $x = x_1$

Two terms on left-hand side

There are three ways in which a combination of two attributes may appear on the left-hand side (the order in which they appear is irrelevant): x and y, x and z, and y and z. Each attribute has three values, so for each pair of attributes there are $3 \times 3 = 9$ possible left-hand sides, e.g.

IF $x = x_1$ AND $y = y_1$

There are three possible pairs of attributes, so the total number of possible left-hand sides is $3 \times 9 = 27$.

Three terms on left-hand side

All three attributes x, y and z must be on the left-hand side (the order in which they appear is irrelevant). Each has three values, so there are $3\times3\times3 = 27$ possible left-hand sides, ignoring the order in which the attributes appear, e.g.

IF $x = x_1$ AND $y = y_1$ AND $z = z_1$

So for each of the three possible 'w = value' terms on the right-hand side, the total number of left-hand sides with one, two or three terms is $9+27+27 = 63$. Thus there are $3 \times 63 = 189$ possible rules with attribute w on the right-hand side.

The attribute on the right-hand side could be any of four possibilities (w, x, y and z) not just w. So the total possible number of rules is $4 \times 189 = 756$.

Self-assessment Exercise 13

Question 1

At the join step of the *Apriori-gen* algorithm, each member (set) is compared with every other member. If all the elements of the two members are identical except the right-most ones (i.e. if the first two elements are identical in the case of the sets of three elements specified in the Exercise), the union of the two sets is placed into C_4.

For the members of L_3 given the following sets of four elements are placed into C_4: $\{a, b, c, d\}$, $\{b, c, d, w\}$, $\{b, c, d, x\}$, $\{b, c, w, x\}$, $\{p, q, r, s\}$, $\{p, q, r, t\}$ and $\{p, q, s, t\}$.

At the prune step of the algorithm, each member of C_4 is checked to see whether all its subsets of 3 elements are members of L_3.

The results in this case are as follows.

Itemset in C_4	Subsets all in L_3?
$\{a, b, c, d\}$	Yes
$\{b, c, d, w\}$	No. $\{b, d, w\}$ and $\{c, d, w\}$ are not members of L_3
$\{b, c, d, x\}$	No. $\{b, d, x\}$ and $\{c, d, x\}$ are not members of L_3
$\{b, c, w, x\}$	No. $\{b, w, x\}$ and $\{c, w, x\}$ are not members of L_3
$\{p, q, r, s\}$	Yes
$\{p, q, r, t\}$	No. $\{p, r, t\}$ and $\{q, r, t\}$ are not members of L_3
$\{p, q, s, t\}$	No. $\{p, s, t\}$ and $\{q, s, t\}$ are not members of L_3

So $\{b, c, d, w\}$, $\{b, c, d, x\}$, $\{b, c, w, x\}$, $\{p, q, r, t\}$ and $\{p, q, s, t\}$ are removed by the prune step, leaving C_4 as $\{\{a, b, c, d\}, \{p, q, r, s\}\}$.

Question 2

The relevant formulae for support, confidence, lift and leverage for a database of 5000 transactions are:

support$(L \rightarrow R)$ = support$(L \cup R)$ = count$(L \cup R)/5000 = 3000/5000 = 0.6$

confidence$(L \rightarrow R)$ = count$(L \cup R)$/count$(L) = 3000/3400 = 0.882$

lift$(L \rightarrow R.) = 5000 \times$ confidence$(L \rightarrow R)$/count$(R) = 5000 \times 0.882/4000 = 1.103$

leverage$(L \rightarrow R)$ = support$(L \cup R)$ − support$(L) \times$ support(R)
$=$ count$(L \cup R)/5000 -$ (count$(L)/5000) \times$ (count$(R)/5000) = 0.056$

Self-assessment Exercise 14

Question 1

We begin by choosing three of the instances to form the initial centroids. We can do this in many possible ways, but it seems reasonable to select three instances that are fairly far apart. One possible choice is as follows.

	Initial	
	x	y
Centroid 1	2.3	8.4
Centroid 2	8.4	12.6
Centroid 3	17.1	17.2

In the following table the columns headed $d1$, $d2$ and $d3$ show the Euclidean distance of each of the 16 points from the three centroids. The column headed 'cluster' indicates the centroid closest to each point and thus the cluster to which it should be assigned.

	x	y	$d1$	$d2$	$d3$	cluster
1	10.9	12.6	9.6	2.5	7.7	2
2	2.3	8.4	0.0	7.4	17.2	1
3	8.4	12.6	7.4	0.0	9.8	2
4	12.1	16.2	12.5	5.2	5.1	3
5	7.3	8.9	5.0	3.9	12.8	2
6	23.4	11.3	21.3	15.1	8.6	3
7	19.7	18.5	20.1	12.7	2.9	3
8	17.1	17.2	17.2	9.8	0.0	3
9	3.2	3.4	5.1	10.6	19.6	1
10	1.3	22.8	14.4	12.4	16.8	2
11	2.4	6.9	1.5	8.3	17.9	1

12	2.4	7.1	1.3	8.1	17.8	1
13	3.1	8.3	0.8	6.8	16.6	1
14	2.9	6.9	1.6	7.9	17.5	1
15	11.2	4.4	9.8	8.7	14.1	2
16	8.3	8.7	6.0	3.9	12.2	2

We now reassign all the objects to the cluster to which they are closest and recalculate the centroid of each cluster. The new centroids are shown below.

	After first iteration	
	x	y
Centroid 1	2.717	6.833
Centroid 2	7.9	11.667
Centroid 3	18.075	15.8

We now calculate the distance of each object from the three new centroids. As before the column headed 'cluster' indicates the centroid closest to each point and thus the cluster to which it should be assigned.

x	y	$d1$	$d2$	$d3$	cluster
10.9	12.6	10.0	3.1	7.9	2
2.3	8.4	1.6	6.5	17.4	1
8.4	12.6	8.1	1.1	10.2	2
12.1	16.2	13.3	6.2	6.0	3
7.3	8.9	5.0	2.8	12.8	2
23.4	11.3	21.2	15.5	7.0	3
19.7	18.5	20.6	13.6	3.2	3
17.1	17.2	17.7	10.7	1.7	3
3.2	3.4	3.5	9.5	19.4	1
1.3	22.8	16.0	12.9	18.2	2
2.4	6.9	0.3	7.3	18.0	1
2.4	7.1	0.4	7.1	17.9	1
3.1	8.3	1.5	5.9	16.7	1
2.9	6.9	0.2	6.9	17.6	1
11.2	4.4	8.8	8.0	13.3	2
8.3	8.7	5.9	3.0	12.1	2

We now again reassign all the objects to the cluster to which they are closest and recalculate the centroid of each cluster. The new centroids are shown below.

	After second iteration	
	x	y
Centroid 1	2.717	6.833
Centroid 2	7.9	11.667
Centroid 3	18.075	15.8

These are unchanged from the first iteration, so the process terminates. The objects in the final three clusters are as follows.

Cluster 1: 2, 9, 11, 12, 13, 14

Cluster 2: 1, 3, 5, 10, 15, 16

Cluster 3: 4, 6, 7, 8

Question 2

In Section 14.3.1 the initial distance matrix between the six objects a, b, c, d, e and f is the following.

	a	b	c	d	e	f
a	0	12	6	3	25	4
b	12	0	19	8	14	15
c	6	19	0	12	5	18
d	3	8	12	0	11	9
e	25	14	5	11	0	7
f	4	15	18	9	7	0

The closest objects are those with the smallest non-zero distance value in the table. These are objects a and d which have a distance value of 3. We combine these into a single cluster of two objects which we call ad. We can now rewrite the distance matrix with rows a and d replaced by a single row ad and similarly for the columns.

As in Section 4.3.1, the entries in the matrix for the various distances between b, c, e and f obviously remain the same, but how should we calculate the entries in row and column ad?

	ad	b	c	e	f
ad	0	?	?	?	?
b	?	0	19	14	15
c	?	19	0	5	18
e	?	14	5	0	7
f	?	15	18	7	0

The question specifies that complete link clustering should be used. For this method the distance between two clusters is taken to be the longest distance

from any member of one cluster to any member of the other cluster. On this basis the distance from *ad* to *b* is 12, the longer of the distance from *a* to *b* (12) and the distance from *d* to *b* (8) in the original distance matrix. The distance from *ad* to *c* is also 12, the longer of the distance from *a* to *c* (6) and the distance from *d* to *c* (12) in the original distance matrix. The complete distance matrix after the first merger is now as follows.

	ad	*b*	*c*	*e*	*f*
ad	0	12	12	25	9
b	12	0	19	14	15
c	12	19	0	5	18
e	25	14	5	0	7
f	9	15	18	7	0

The smallest non-zero value in this table is now 5, so we merge *c* and *e* giving *ce*.

The distance matrix now becomes:

	ad	*b*	*ce*	*f*
ad	0	12	25	9
b	12	0	19	15
ce	25	19	0	18
f	9	15	18	0

The distance from *ad* to *ce* is 25, the longer of the distance from *c* to *ad* (12) and the distance from *e* to *ad* (25) in the previous distance matrix. Other values are calculated in the same way.

The smallest non-zero in this distance matrix is now 9, so *ad* and *f* are merged giving *adf*. The distance matrix after this third merger is given below.

	adf	*b*	*ce*
adf	0	15	25
b	15	0	19
ce	25	19	0

Self-assessment Exercise 15

Question 1

The value of TFIDF is the product of two values, t_j and $\log_2(n/n_j)$, where t_j is the frequency of the term in the current document, n_j is the number of documents containing the term and n is the total number of documents.

For term 'dog' the value of TFIDF is $2 \times \log_2(1000/800) = 0.64$
For term 'cat' the value of TFIDF is $10 \times \log_2(1000/700) = 5.15$
For term 'man' the value of TFIDF is $50 \times \log_2(1000/2) = 448.29$
For term 'woman' the value of TFIDF is $6 \times \log_2(1000/30) = 30.35$
The small number of documents containing the term 'man' accounts for the high TFIDF value.

Question 2

To normalise a vector, each element needs to be divided by its length, which is the square root of the sum of the squares of all the elements. For vector $(20, 10, 8, 12, 56)$ the length is the square root of $20^2 + 10^2 + 8^2 + 12^2 + 56^2 = \sqrt{3844} = 62$. So the normalised vector is $(20/62, 10/62, 8/62, 12/62, 56/62)$, i.e. $(0.323, 0.161, 0.129, 0.194, 0.903)$.

For vector $(0, 15, 12, 8, 0)$ the length is $\sqrt{433} = 20.809$. The normalised form is $(0, 0.721, 0.577, 0.384, 0)$.

The distance between the two normalised vectors can be calculated using the dot product formula as the sum of the products of the corresponding pairs of values, i.e. $0.323 \times 0 + 0.161 \times 0.721 + 0.129 \times 0.577 + 0.194 \times 0.384 + 0.903 \times 0 = 0.265$.

Index