Artificial Intelligence & Data Mining

CRN 34123, UMC G400 20077

Dr Bryant

2024/2025

Preface

This booklet was generated using LaTeX 2_ε on 16th September, 2024 .

0.1 Bookmarks

This booklet includes bookmarks for chapters, sections and subsections. If you are using the Adobe Acrobat Reader, then these may be displayed in the Navigation Pane on the left hand side. You can control whether the bookmarks are displayed by selecting the option from the menu at: View → Show/Hide → navigation panes.

0.2 Hypertext Links

This booklet contains hypertext links for web addresses, citations, page references, section references, table references and figure references. The links are displayed in colour. The colour depends on the the type of link.

magenta (deep purplish red) for web addresses.

cyan (light blue) for bibliographical citations in text.

red for all the other types of links.

The magenta hypertext links for web addresses will only work if the security settings of the PDF reader you are using permit it to display a web page using a web browser.

Note that Adobe Acrobat Reader includes a back button which you may find very useful after following one of these links. You can control whether the back button is displayed on the tool bar by selecting the option from the menu at: View → Show/Hide → Toolbar Items → Show Page Navigation Tools → Previous View

0.3 The Inclusive Student Experience Project

The booklet has been made inclusive and accessible as follows.

- Headings are consistently positioned to one side and can be clearly differentiated from the body of the text.
- Paragraph text is consistent in size and style.
- Neither capitals or the size or colour of text has been used to emphasise a point.
- The default font size is 12 points.
- A sans serif font has been used.
- Images have only been used when they are necessary or enhance understanding of written content.
- All images, diagrams and tables have a legend.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Module Handbook

This module will introduce you to Artificial Intelligence (AI) and Data Mining (DM) techniques for problem solving. You will become familiar with AI techniques and terminology for knowledge representation and searching, and gain an understanding of DM algorithms, and learn how these techniques are used in real world applications.

1.1 Programmes Taking the Module

- BSc (Hons) Computer Science
- BSc (Hons) Computer Science with Cyber Security

1.2 Aims of the Module

1. To introduce Artificial Intelligence (AI) and Data Mining (DM) techniques for problem solving.
2. To provide experience with AI techniques and terminology for knowledge representation and searching.
3. To develop an understanding of DM algorithms.
4. To highlight the use of the practical techniques in real world applications.

1.3 Learning Strategy

1.3.1 Semester 1

The lectures will provide students with an understanding of the aims, concepts and processes of data mining and explain data mining algorithms and their application. During the tutorials, students will manually execute algorithms or investigate properties of the algorithms. During the workshop, students will apply Weka's implementation of these algorithms to small datasets.

1.3.2 Semester 2

The classes will comprises lectures and tutorials on Artificial Intelligence (AI).

Intelligent agents need knowledge about the world in order to reach good decisions. Knowledge is contained in agents in the form of sentences in a knowledge representation language that are stored in a knowledge base. A knowledge-based agent is composed of a knowledge base and an inference mechanism. Therefore, the syllabus covers knowledge representation and logical inference.

The syllabus focuses on two knowledge representation languages: propositional logic and first-order predicate logic. Propositional logic illustrates all the basic concepts of logic and provides a pedagogical stepping stone to first-order predicate logic. First-order predicate logic is sufficiently expressive to represent a good deal of our common sense knowledge. It also either subsumes or form the foundation of many other representation languages.

Inference is the process of deriving new sentences from old ones. Inference rules are patterns of sound inference that can be used to find proofs. The syllabus includes various inference rules, including resolution which provides a complete proof system for both propositional logic and first-order predicate logic, using knowledge bases expressed in conjunctive normal form.

1.4 Syllabus

1.4.1 Syllabus for Semester 1

- Introduction to Data Mining
- Data Mining Algorithms
  - OneR
  - Naive Bayes
  - Instance-based Learning (e.g., Nearest Neighbour)
  - Covering algorithms (e.g., PRISM)
  - Decision trees (e.g., ID3)
  - Clustering (e.g., k-means clustering)
- The Process Of Data Mining

  - Evaluating results.
  - Holdout and Cross-validation.
  - Deriving confidence intervals for predictive accuracy estimates.

1.4.2 Syllabus for Semester 2

- Introduction to Artificial Intelligence
  - Foundations of AI laid over last 2500 years.
  - History of AI (1943-2015).
  - Turing Test.
- Knowledge Representation
  - Propositional Logic
    - Syntax, semantics, grammar and connectives.
    - Tautologies, contradictions and contingencies.
    - Equivalences and entailment.
  - Predicate Logic
    - Quantifiers and duality (De Morgan's laws).
    - Scoping rules; bound and free variables; the equality symbol.
- Logical inference
  - Transformational proofs
  - Deductive proofs
    - Inference rules.
    - Conditional proofs (proof by adopting a premise).
    - Indirect proofs (reductio ad absurdum or proofs by contradiction).
    - Resolution.
  - Handling variables.
  - Inductive proofs
    - Inverse resolution
- Logic and Data Mining
- Relational Data Mining

1.5 Learning Outcomes

Upon successful completion of the module, students will be able to:

- apply AI and DM aims, concepts, terminology and processes;
- use techniques for knowledge representation and searching;
- formulate problems in logic and use logical inference to reach sound conclusions;
- differentiate multiple DM algorithms and identity suitable circumstances when they can be applied;
- discuss some real world applications of AI and DM.

1.6 Provisional Schedule

Tables 1.1 and 1.2 show the provisional schedule for the classes.

Week	1^{st} Lecture	2^{nd} Lecture	Workshop	Private Study
T1.1	Introduction	Input/Output		
T1.2	1R	Chapter 3	Chapter 4	
T1.3	Naïve Bayes	Chapter 5	Chapter 6	
T1.4	Nearest Neighbour	Chapter 7	Chapter 8	
T1.5	PRISM	Chapter 9	Chapter 10	
T1.6	ID3	Chapter 11	Chapter 12	
T1.7	Towards C4.5 and J48	Chapter 13	Chapter 14	
T1.8	Evaluating Results	Cross-validation		Chapter 17
T1.9	Confidence Intervals	Chapter 15		Chapter 17
T1.10	Clustering&Applications	Chapter 16		
T1.11	A Deeper Look	Exam Preparation		

Table 1.1: Provisional schedule for Semester One.

Before you come to a lecture, carefully read through the slides of the presentation. During the lecture, make written notes. After the lecture, read your notes and the slides again. Before you come to a tutorial, revise the content of the corresponding lecture. During the tutorial, write down your answers to the questions of the tutorial exercise and do not be surprised if a member of staff asks to see your answers. After the tutorial, compare your answers with Dr Bryant's solutions.

Week	1^{st} Lecture	2^{nd} Lecture	Tutorial
T2.1 (18)	What is AI?	Foundations of AI	Chapter 18
T2.2 (19)	History (1943-1969)	History (1970-2020)	Chapter 19
T2.3 (20)	Propositional Logic	Propositional Logic	Chapter 20
T2.4 (21)	Transformational Proofs	Transformational Proofs	Chapter 21
T2.5 (22)	Validity & Inference	Validity & Inference	Chapter 22
T2.6 (23)	Deductive Proofs	Deductive Proofs	Chapter 23
T2.7 (24)	Predicate Logic	Predicate Logic	Chapter 24
T2.8 (25)	Quantifiers	Quantifiers	Chapter 25
T2.9 (26)	Deductive Proofs	Deductive Proofs	Chapter 26
T2.10 (27)	Resolution	Resolution	Chapter 27
T2.11 (28)	Logic and Data Mining	Relational Data Mining	Chapter 28

Table 1.2: Provisional schedule for Semester Two

1.7 Textbooks and Teaching Materials

Lecture slides, datasets and the exercise booklet containing the tutorials and workshop will be made available on Blackboard. For more information about Blackboard, see Section 2.6 on page 14.

The remainder of this section concerns text books. Specific details of essential (core) reading are given at the end of every lecture. The recommendations are listed on a slide near the end of each presentation in the booklet of lecture slides. Generally speaking, it is not normally necessary for a student to purchase the core text books for this module because they are available in electronic or/an paper format from the library.

1.7.1 Data Mining

The core textbook for Semester 1 is (Witten et al., 2016). To get a broader view of Data Mining, you may wish to begin by reading Chapters 1-3. This may help you to see the big picture. Another good book on Data Mining is (Han et al., 2022).

Any student taking this module who does not know the maths in the book by (Croft & Davidson, 2020) could benefit from reading it.

1.7.2 Artificial Intelligence

The core textbook during Semester 2 will be (Russell & Norvig, 2022).

1.8 Software Used on the Module

The software used on the module is called Weka. Weka stands for Waikato environment for knowledge analysis. Weka is an open source data mining package developed by the University of Waikato in New Zealand. Weka includes:

- sophisticated learning algorithms which you can apply to your data set. These are mainly *classifiers* although it also includes algorithms for learning association rules (where there is no class attribute) and for clustering (grouping similar instances together).
- tools for transforming your data sets, i.e. *filters*.

You may run Weka on:

Computers provided by the University of Salford Weka should be installed on the Windows operating system on the computers in the laboratories on campus.

Your Own Computer Weka is freely available at: http://www.cs.waikato.ac.nz/ml/weka/ You can download either a platform-specific installer or an execuatable Java jar file that you run in the usual way if Java is installed. You should install one of the stable versions, rather than one of the development versions. At the time of writing, the 3.8.* versions are the stable versions. All the versions are available from https://sourceforge.net/projects/weka/files/.

Depending upon which version of Weka you are using then you may have to install a package which includes algorithms taught in the lectures, such as NaiveBayesSimple and Prism. To install it, do the following:

1. Start the package manager from the Tools pull down menu of Weka.
2. Search for the package ”Simpleeducationallearningschemes”.
3. Highlight it and click install.

1.9 Surgeries (Office Hours)

If you want to have a one-to-one, in depth conversation with Dr Bryant then do **not** attempt to do this during the ten minute gap between classes. Instead, please attend one of his surgeries. Please note that this is a drop-in service. In other words, there is no need to make an appointment. The time and location of his next surgery is shown on Blackboard's calendar, which you find at:
Blackboard
⟶ AI & Data Mining
⟶ Calendar tab (at the top on the left-hand-side.)
⟶ Click on the button ”Month”.

1.10 Recording of Teaching Sessions

Students are prohibited from **video** recording classes. In other words, you are not allowed to do this. Students who wish to **audio** record teaching sessions must comply with the rules set out here. These rules apply to all students of the University. These rules apply to any teaching session including lectures, seminars, tutorials (group or one to one), discussion groups, supervision sessions, laboratory work, fieldwork or other learning activity.

1.11 Assessment Methodology

Each level of an undergraduate degree programme consists of 120 credits. Each level is divided into a number of modules. This module is worth 20 credits.

Assessment of your performance in the module will be by the following method:

- The material covered during Semester One is assessed by exam at the end of Semester One. This assessment will contribute 50% of your overall module mark.
- During Semester Two, you will be assessed by coursework on material covered during Semester Two. This will contribute 50% of your overall module mark.

To pass the module you must achieve a mark of 40% or greater overall.

Chapter 2

Where to Get Help

2.1 Aim of the Exercise

Dr Bryant and your laboratory tutor can answer your questions on Artificial Intelligence and Data Mining. The purpose of this exercise is to check that you know where to get help on other matters.

2.2 Exercise

- Verify that you have a calculator which is appropriate (see Section 2.3.1).
- Check that you have a Student ID card and that it grants you access to the laboratory where your workshop is taking place. If not, take remedial action (see Section 2.4).
- Make sure you can log into the computers in the laboratory. If not, take remedial action (see Section 2.5).
- Verify that you have access to the part of Blackboard for this module. If not, take remedial action (see Section 2.6).
- If you are planning to use the electronic copy of a text book then check that you can access it (see Section 2.7.2).
- Set up an email signature that contains the information listed in Section 2.12.1.
- Find previous exam papers for this module (see Section 2.7.1).
- If necessary, spend some time making yourself familiar with the route to the rooms where your classes for this module are taking place (see Section 2.8).

2.3 Help with Maths

2.3.1 Scientific Calculator

Most of you will find it easier to do this module if you bring a scientific calculator to the tutorials and workshops. Other types of calculator may not have all the functions you will probably want to use on this module. You can buy a very good one for less than £15. Calculators are available from the Inspire on-line shop http://www.salford-inspire.co.uk Calculators are allowed in the exam if they are cleared of all pre-stored programmes or information.

Can't I just use the calculator on the computer in the lab? If it is a scientific calculator then this would be useful during the workshop. Of course, you will not have access to a computer during the examination.

Can't I just use the calculator on my mobile phone? You may find that the calculator on your mobile phone is rather basic and does not have all the functions you want on this module. Mobile phones are strictly forbidden in examinations.

Which type of calculator should I buy? If you are selecting a calculator then bear in mind the type of maths you will be doing. The maths used in this module includes: arithmetic and fractions, logarithms, geometry, probability, combinatorics and statistics. Dr Bryant uses a Casio scientific calculator, see https://www.casio.co.uk/calculators/education/scientific

2.3.2 Mathematical Constants and Formulae Sheet

Dr Bryant plans to include a mathematical constants and formulae sheet in the examination question paper. A copy of this sheet is available from Blackboard. So, as part of your preparations for the exam, you should make yourself familiar with it. This plan is subject to a routine part of our quality assurance procedures, namely the verification of the exam question paper by the external examiner. Dr Bryant will endeavour to notify you if there is any change to the plan.

2.3.3 MathScope

MathScope is a support unit for students who may experience difficulties with mathematics in whatever subject they are studying. MathScope is staffed and resourced throughout Semester 1 and 2 and offers a comprehensive service for students who are having difficulty coping with the mathematical demands of their course at whatever level. The contact details of MathScope are given in Section 2.12.6.

2.4 Student ID Cards

You may need a Student ID to enter the laboratory where your workshop is taking place. If your card will not grant you access to your laboratory then please go to the School Office (see Section 2.12.7). If your card is lost or damaged then please purchase a replacement card via the web shop at http://shop.salford.ac.uk/ by clicking on:
Product Catalogue ⟶ Student Administration ⟶ Replacement Student ID Card.
Please note that Dr Bryant has no involvement with the issue and maintenance of these cards. Consequently he cannot rectify any problems with them.

2.5 Computers in the Laboratory

You will need a User name and password to use the computers in the laboratory. If you have registered as a student at this University then your should already have these. Please raise any problems with your account, email, the operating system, the laboratory hardware or printers with Digital IT. Digital IT aim to resolve all cases which are logged and then communicate the resolution to you.

Please note that Dr Bryant is not part of Digital IT. Dr Bryant has no involvement with the set up and maintenance of the laboratory hardware, operating systems, printers or User accounts. Consequently he cannot rectify any problems with them. For obvious security reasons, Dr Bryant does not know your account password.

2.5.1 How do I report problems to Digital IT?

When reporting issues to DigitalIT you need to supply the following information:

- your contact details,
- a description of the problem, and
- the date and time when the problem occurred.

When reporting issues with a computer provided by the university you also need to supply the asset tag, which is usually displayed on the side or back of the computer, rather than the monitor/screen.

It is important to request a case-reference number. You are strongly advised to keep a record of both your communication with Digital IT and your case-reference number. The contact details of Digital IT are given in Section 2.12.3.

2.6 Blackboard

The university's chosen standard virtual learning environment is called Blackboard. Electronic copies of some of the teaching materials will be made available on Blackboard. Dr Bryant may use Blackboard's facilities for communicating with students such as announcements and email.

Where is the best place to seek help with Blackboard? This depends upon what your problem is.

You find Blackboard confusing. If you don't know how to use Blackboard then help is available at https://www.salford.ac.uk/library/know-how/blackboard.

You cannot log into Blackboard Please contact Digital IT (see Section 2.12.3).

You cannot find this module on Blackboard. If you cannot see this module on Blackboard then this may be because there is a problem with your registration. Please ask Digital IT (see Section 2.12.3) whether your module enrolment details are correct.

Blackboard is broken. Please contact Digital IT (see Section 2.12.3).

You cannot find the teaching materials. If you can see this module on Blackboard but you cannot find the teaching materials for this module then please ask your laboratory tutor to help you.

Dr Bryant is not part of the support team responsible for maintaining Blackboard. Dr Bryant cannot fix Blackboard if it is not working properly.

2.7 Library Resources

2.7.1 Past Exam Papers

Past examination papers are available from the library's website (see Section 2.12.5).

2.7.2 Electronic Books

Copies of the books listed in Section 1.7 are available from the library. Some books are available in electronic format. To access an e-Book off campus you should expect to have to login. If you need help then please contact the library (see Section 2.12.5).

Please note that Dr Bryant is not a librarian. Dr Bryant has no involvement with the maintenance of the library and its resources. Consequently he cannot rectify any problems with them. For obvious security reasons, Dr Bryant does not know your password.

If you plan to purchase your own copy of a textbook then please note that some books are available from the Inspire on-line shop at http://www.salford-inspire.co.uk.

2.8 Directions to the Classes

Your timetable should show the rooms where your classes for this module are taking place. If necessary, spend some time making yourself familiar with the route to these rooms. (As there is only a ten minute gap between consecutive classes, you may not have time to discover the direct routes if you leave it until the last minute.) Unfortunately Dr Bryant does not have time to answer emails requesting directions to rooms or buildings on campus. You can find a map at: http://www.salford.ac.uk/about-us/travel
If you find yourself lost in a building then a good place to ask for help is the reception.

It is your responsibility to make sure you get up early enough to allow sufficient time for you to travel to your first class of the day. Alarm clocks are available from the Inspire on-line shop at http://www.salford-inspire.co.uk

2.9 What should I do if I am ill?

Do not enter the classroom if you are ill, especially if you believe that you may have an infection or contagion. Instead, please read the part of your Student Handbook concerning sickness and absence, which you can access by clicking here.

2.10 How can I catch up if I miss a class?

It is your responsibility to keep up-to-date. If you fall behind then you should endeavour to catch up. Tables 1.1 and 1.2 show the provisional schedule for the classes. This tells you what need to do if you miss a class. Unless Dr Bryant tells you otherwise, you may assume that we are adhering to this schedule. You must keep up-to-date with the tutorials and workshop; otherwise you will find it difficult to perform well in the assessments.

2.11 Disabled Students

The Disability and Inclusion Service (see Section 2.12.4) exists to support disabled students throughout their studies at the university. Disabled students are kindly invited to discuss their support requirements for *this particular module* with Dr Bryant should they so wish to do so. The best time to do this is during one of his surgeries (see Section 1.9).

2.12 Contact Details

2.12.1 Guidance on How to Contact Staff

Face-to-face If you decide to visit any of the people mentioned in this section in person then remember to take your ID card with you.

Email If your email message is about a particular module then you should state this in your message because most members of staff work on more than one module. When writing a message to send to any of the email addresses given in this section:

- use your university email account;
- write sentences of English which are lucid, grammatically correct and complete;
- strive to avoid spelling mistakes;
- use an appropriate salutation. (An example of an appropriate salutation would be “Dear Dr Bryant” and an example of an inappropriate one would be “Hey”.)
- Include the following information at the end your message.
  - Your name.
  - Your student roll number. (Student roll numbers usually comprise the @ symbol followed by eight digits.)
  - Your User ID. (User IDs usually comprise three characters followed by three digits.)
  - Your programme (see Section 1.1).
  - The level at which you are studying, namely, Level 5.

MS Teams If you are new to MS Teams then please note that you can find an introduction to MS Teams on the link here to linkedin.com. Please remember to have your microphone switched on if you want to be heard and to have your webcam switched on if you want to be seen. Please check that you can get your webcam and microphone to work with MS Teams before you try to contact a member of staff. A very helpful video that tells you how to do this can be found on the link here to linkedin.com. If you do not have a microphone then please note that headsets are available from the Inspire on-line shop at http://www.salford-inspire.co.uk. A headset includes headphones and a microphone.

2.12.2 Contact Details of Dr Bryant

Face-to-face on campus: Attend his surgery (see Section 1.9 on page 8).

Email: c.h.bryant@salford.ac.uk

Virtual face-to-face: MS Teams

If you just want to send Dr Bryant an written message, then do not use MS Teams or Blackboard to do this. Instead, please send it to his email address.

2.12.3 Contact Details of Digital IT

Face-to-face on campus: click here

Email: Digital-ITServicedesk@salford.ac.uk

Telephone: 0161 295 2444

Web: https://testlivesalfordac.sharepoint.com/sites/Uos_Students/SitePages/Digital-IT.aspx

2.12.4 Contact Details of the Disability and Inclusion Service

Face-to-face on campus: University House, Peel Park Campus.

Email: disability@salford.ac.uk

Telephone: 0161 295 0023 (option 1, option 2)

Web: https://www.salford.ac.uk/askus/topics/disability-inclusion-service

2.12.5 Contact Details of The Library

Face-to-face on campus: Clifford Whitworth Building, Peel Park Campus.

Email: library-SEE@salford.ac.uk

Web: http://www.salford.ac.uk/library

2.12.6 Contact Details of MathScope

Face-to-face on campus: Room 02.05, Second Floor, SoSEE building, Peel Park Campus.

Email: mathscope@salford.ac.uk

Web https://www.salford.ac.uk/skills/maths-and-numeracy-support/mathscope

2.12.7 Contact Details of The School Office

Face-to-face on campus: Room 01.07, First Floor, SoSEE building, Peel Park Campus.

Email: SEESchoolEnquiries@salford.ac.uk

Telephone: 0161 295 5338

Web click here

Part II

Data Mining

Chapter 3

Tutorial: 1R

3.1 Aim of the Exercise

The aim of this exercise is for you to gain experience of explaining how the 1R algorithm can be applied to some small datasets.

3.2 History of 1R

1R is simple algorithm for learning rudimentary sets of classification rules. It was proposed by Robert C. Holte of the University of Ottawa. He published a thorough investigation of 1R (Holte, 1993). His take home message was that applying high-powered inductive inference methods to simple datasets was like using a sledge-hammer to crack a nut.

3.3 Predicting the Desirability of Accommodation

Let's suppose you are an estate agent and that you are interested in predicting whether some accommodation is desirable or not.

3.3.1 Nominal Values

Assume that you have the examples of properties shown in Table 3.1.

	Price	Location	State	Desirability
1	Average	Central	OK	Yes
2	High	Countryside	OK	No
3	Low	Central	Good	Yes
4	High	Central	Good	Yes
5	Average	Countryside	OK	Yes
6	Average	Central	Bad	No
7	Low	Countryside	Bad	Yes

Table 3.1: Accommodation dataset.

1. Manually apply the 1R algorithm to the set of examples listed above.
2. If you have to make any arbitrary decisions then discuss whether or not these have any implications.

3.3.2 Missing Values

Manually apply the 1R algorithm to the set of examples listed in Table 3.2.

	Price	Location	State	Desirability
1	?	Central	OK	Yes
2	High	Countryside	OK	No
3	Low	Central	?	Yes
4	High	Central	?	Yes
5	?	Countryside	OK	Yes
6	Average	Central	Bad	No
7	Low	Countryside	?	Yes

Table 3.2: Accommodation dataset with some values missing.

3.4 Predicting Whether a Container is Suitable

Let's suppose you are interested in containers which are used to transport goods. Consider the data on containers listed in Table 3.3, where the class is suitable.

country	capacity	transport	suitable
cuba	910	air	no
cuba	800	boat	no
ethiopia	830	air	yes
germany	700	air	yes
germany	680	air	yes
germany	650	boat	no
ethiopia	590	boat	yes
cuba	740	air	no
cuba	690	air	yes
germany	750	air	yes
cuba	750	boat	yes
ethiopia	740	boat	yes
ethiopia	810	air	no
germany	710	boat	no

Table 3.3: Transport dataset.

1. Showing all the steps, use the discretisation method and a bucket size of 3 to obtain nominal values for attribute capacity. Rewrite the dataset using discretised capacities.
2. Apply algorithm 1R to the discretised dataset. Show all intermediate steps.
3. Explain how 1R deals with missing nominal and numeric values. Illustrate your explanation using the dataset above if the third instance (ethiopia 830 air yes) changes to (ethiopia ? ? yes), i.e., it has missing values for capacity and transport.

Chapter 4

Workshop: Introduction to Weka

4.1 Aim of the Exercise

The purpose of this exercise is to introduce you to the software which will be used in the all the workshop exercises on data mining.

4.2 Weather Data

For your convenience, the weather data is shown in Table 4.1.

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

Table 4.1: Weather training data with some numeric attributes.

4.3 An Input File for Weka

We have seen in the lecture that the input to a learning algorithm needs to be in *ARFF* format. An *ARFF* file consists of:

- The dataset name, defined using the *@relation* tag.
- Attribute information, described using the *@attribute* tag. It includes the name and the type of data: *numeric (real, integer)* , *nominal (enumerated)*, *date* and *string*.
- A list of instances with all their attribute values separated by commas, defined using the *@data* tag.

For example, assume that we have the data shown in Table 4.1. In CSV format, it will appear as shown in Table 4.2. Table 4.3 shows the same data in ARFF format. Note that we have not specified the *class* because Weka assumes that the class is the last

```
outlook,temperature,humidity,windy,play
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
...
```

Table 4.2: Part of the weather training data (with numeric attributes) in CSV.

```
% This is a comment about the data set.
% This data describes examples of whether to play
% a game or not depending on weather conditions.
@relation letsPlay
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
...
```

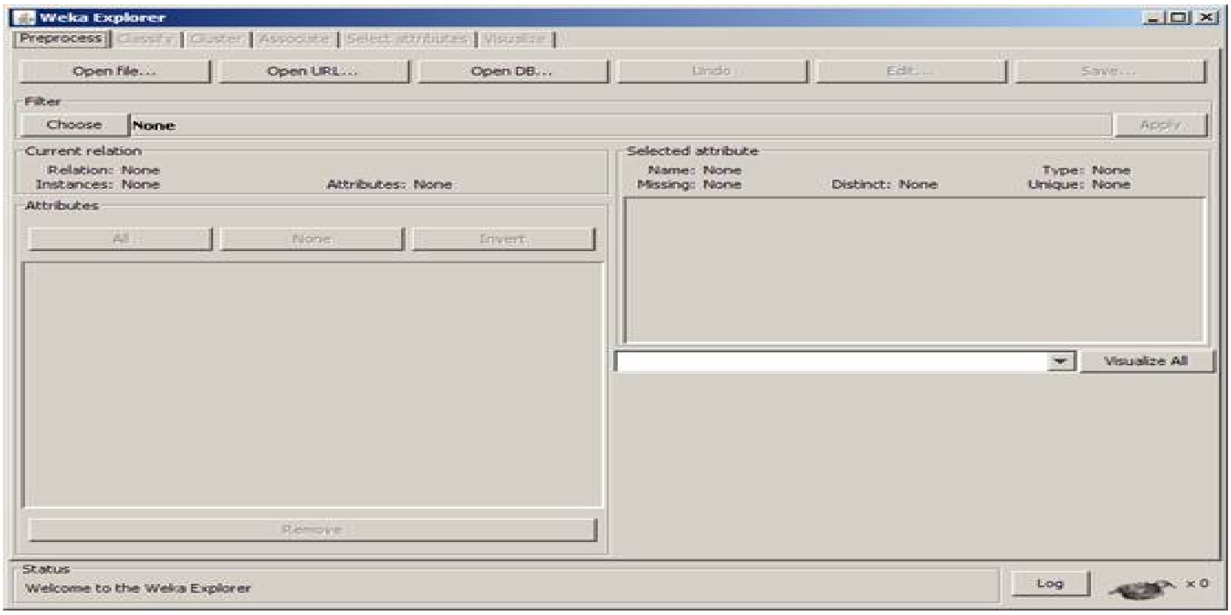
Table 4.3: Part of the weather training data (with numeric attributes) in ARFF.

attribute. Weka input files are plain text files in a *flat* format. So you can create a Weka input file using a text editor. A Weka input file must have the extension *.arff* to indicate that it is in ARFF format.

During the exercises listed in Section 4.4 you will use a data file called weather.arff. This is available from Blackboard. Briefly examine the contents of this file. Take care to not make any changes to the content. The data is in Weka's ARFF format.

4.3.1 Weka's Explorer Graphical User Interface

Weka has three graphical user interfaces (GUIs) and one command line interface. Throughout the workshop, you will use just one of these, namely the *Explorer* GUI. When Weka starts you will see the GUI Chooser panel. Always select Explorer from the four choices on the right hand side of this panel. You will then see something very similar to the screenshot shown in Figure 4.1. The Explorer environment contains six tabs:



Filter: None

Relation: None
Instances: None
Attributes: None

Name: None
Missing: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

Figure 4.1: Preprocess tab of Weka's explorer interface.

Preprocess: to select and modify datasets.

Classify: to train and test learning algorithms for classification (or for regression).

Cluster: to learn clusters from the dataset. This is used in the exercise in Chapter 14.

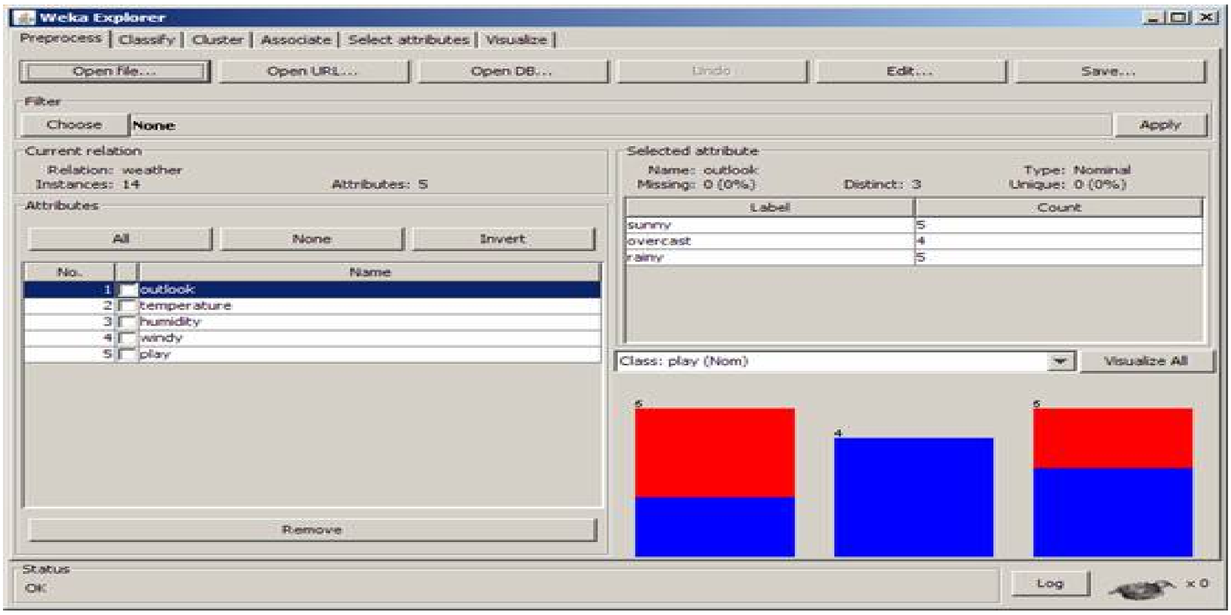
Associate: to learn association rules from the dataset.

Select attributes: to choose appropriate attributes which describe the dataset.

Visualise: to plot the data (interactive, 2D).

4.4 Exercises

Click on the “Open file” button and load in the Weather dataset (see Section 4.2). You should now see something similar to the screenshot shown in Figure 4.2.



Filter: None

Relation: weather
Instances: 14
Attributes: 5

No.		Name
1	□	outlook
2	□	temperature
3	□	humidity
4	□	windy
5	□	play

Name: outlook
Type: Nominal
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)

Label	Count
sunny	5
overcast	4
rainy	5

Class: play (Nom)

Status: OK

Figure 4.2: Preprocess tab after loading weather dataset.

4.4.1 Getting To Know Your Data

One of your the first tasks as a Data Miner is to get to know your data. Weka helps you do this. The screen shows us some information about the data. In the attributes window on the left hand side we can see that there are five attributes in this dataset, outlook, temperature, humidity, windy and play. We can see what all the data looks like by clicking on the edit button. The purpose of this dataset is to predict whether or not we will play (the play attribute) based on the outlook, temperature, humidity and windy attributes.

Can you work out what the window pane entitled ‘Selected attribute’ is telling you? Note that in the screendump shown above, the attribute selected is ‘outlook’ and the class selected is ‘play’. It tells you how often the outlook is sunny, how often it is overcast and how often it is rainy.

Can you see how the columns of the histogram on the right relate to the rows of the window pane entitled ‘Selected attribute’? The histogram shows how often each value of outlook occurs in each class. If the outlook is sunny or rainy sometimes we play and sometimes we do not. If the outlook is overcast we always play. So now can you see what the colours mean? To check that you have understood what the colours mean, click on the play attribute in the window pane entitled ‘Attributes’. (Click on the name of the attribute, rather than the square box just to the left of it.)

Next, select the attribute windy, leaving the class set to ‘play’, and check you understand the histogram. It is important to understand what the Preprocess tab tells you about your data. So please check that you have understood by asking yourself the following questions.

- What is the window pane entitled ‘Selected attribute’ telling me?
- How do the columns of the histogram on the right relate to the rows of the window pane entitled ‘Selected attribute’?

Next, select the attribute temperature, again leaving the class set to ‘play’. Before you try to understand the histogram, recall that temperature is a numeric attribute.

By now you should be getting to know the weather data set.

4.5 Further (Optional) Reading

The Weka workbench is described in Appendix B of (Witten et al., 2016).

Chapter 5

Tutorial: Naïve Bayes

5.1 Aim of the Exercise

The aim of this exercise is for you to gain experience of explaining how to use Naïve Bayes for classification.

5.2 History of Using Naïve Bayes for Classification

Thomas Bayes was an eighteenth century philosopher who was born in England. He published (Bayes, 1763) his theory of probability in 1763. The rule that bears his name has been a cornerstone of probability theory ever since. Bayesian techniques had been used in the field of pattern recognition for twenty years (Duda & Hart, 1973) before they were adopted by Machine Learning researchers at the start of the 1990s.

Pierre Laplace was an eighteenth century French statistician. The first half of the lecture on Naïve Bayes explains how to apply Naïve Bayes to attributes with nominal values and why this sometimes give rise to the need for a Laplace estimator/correction. You will use these ideas during the tutorial.

The second half of the lecture focuses on applying Naïve Bayes to attributes with numeric values. You will apply Naïve Bayes to the weather dataset containing numeric values during the workshop exercise in Chapter 6.

5.3 The Medical Diagnosis Dataset

Consider the training data shown in Table 5.1, where diagnosis is the class. Suppose you decide to apply Naïve Bayes to this data. Unfortunately your computer is broken so you cannot use Weka.

Temperature	Skin	Blood Pressure	Blocked Nose	Diagnosis
Low	Pale	Normal	True	N
Moderate	Pale	Normal	True	B
High	Normal	High	False	N
Moderate	Pale	Normal	False	B
High	Red	High	False	N
High	Red	High	True	N
Moderate	Red	High	False	B
Low	Normal	High	False	B
Low	Pale	Normal	False	B
Low	Normal	Normal	False	B
High	Normal	Normal	True	B
Moderate	Normal	High	True	B
Moderate	Red	Normal	False	B
Low	Normal	High	True	N

Table 5.1: Medical diagnosis training dataset

5.4 Exercise Without Laplace Corrections

- Do not use the remedy proposed by the French statistician Pierre Laplace.
- Note that the dataset does not include any numeric attributes.

1. Draw the table of likelihoods used in Bayesian probability.
2. Manually apply Bayesian probability to solve the following three problems, where ? indicates that the value for that attribute is missing. Show all your calculations.

Problem 1	Problem 2	Problem 3
Temperature = low	Temperature = low	Temperature = moderate
Skin = normal	Skin =?	Skin = normal
Blood pressure = high	Blood pressure = normal	Blood pressure = high
Blocked nose = true	Blocked nose = true	Blocked nose = true

5.5 Exercise With Laplace Corrections

Repeat the exercise above but this time make Laplace corrections.

Chapter 6

Workshop: Naïve Bayes

6.1 Aim of the Exercise

The aim of this exercise is that you will learn how to use Weka to:

1. generate a probabilistic Naïve Bayes classifier from training data;
2. evaluate this classifier on test data.

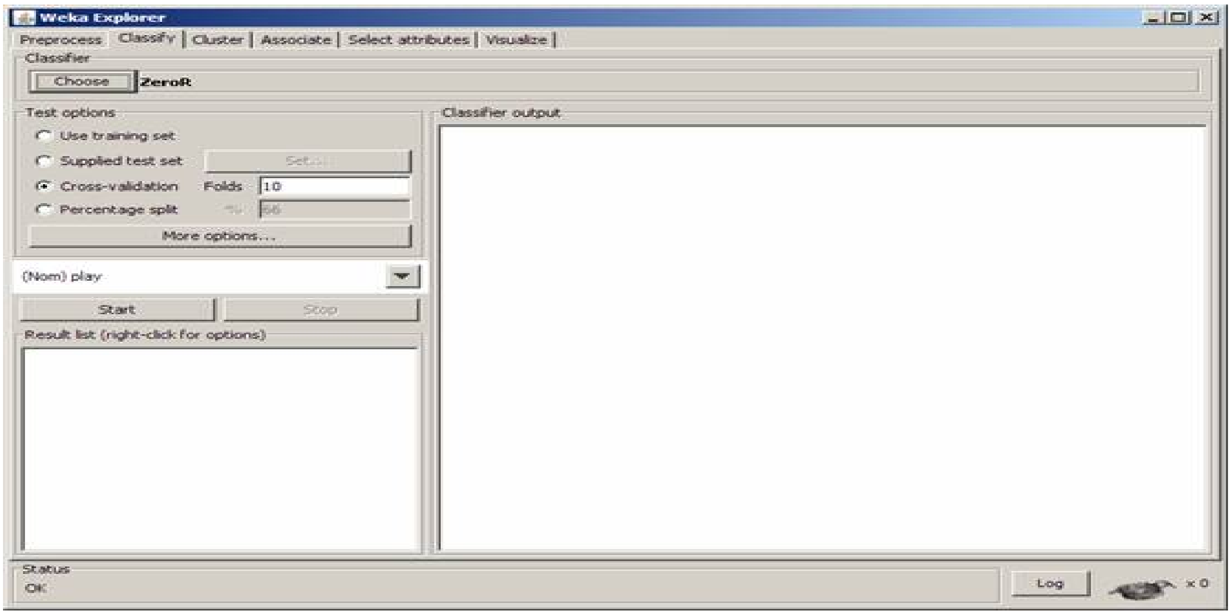
6.2 Applying Naïve Bayes to the Weather Dataset

Depending upon which version of Weka you are using, you may have to install the package ”Simple educational learning schemes” (see Section 1.8 on page 8). Next, click on the ‘Open file’ button on the Preprocess tab of the Explorer interface and load in the weather dataset containing numeric values. (If you cannot remember where to access the dataset then you need to read Section 4.2 on page 25 again.)

To get Weka to generate a probabilistic Naïve Bayes classifier for us we need to click on the Classify tab. You should see something very similar to the screenshot shown in Figure 6.1. The first thing we need to do is decide what type of classifier we want. We want a Bayesian classifier, so we need to select ‘bayes’ from the choose button. Weka includes at least seven Bayesian classifiers. The one called NaïveBayesSimple implements the probabilistic Naïve Bayes classifier that we studied during the lecture. It uses the remedy proposed by the French statistician Pierre Laplace for the zero-frequency problem. It uses the normal distribution to model numeric attributes.

6.2.1 Run Naïve Bayes on the Weather Dataset

To allow you to compare the output with the lecture slides, use the test option ‘Use training set’. Click on the “More Options” button on the Test options pane. To reduce the amount of output from Weka to just what we need, turn off all the options except ‘Output



Classifier: ZeroR

Test options:
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) play

Result list (right-click for options)

Classifier output

Status: OK

Figure 6.1: Classify tab of Weka's explorer interface.

model'. Now run NaïveBayesSimple on the weather dataset by clicking on the 'Start' button. You should see that the means for temperature and humidity have the values that you saw in the lecture. Notice that, on first inspection, the values for the discrete values appear to be different to those in the lecture. Write down an explanation of this difference in the box below.

Your explanation:

6.3 Testing Options

Evaluating the models we generate using data mining is very important. There will be several lectures on this topic later in the semester. The Test options pane on the Classify tab of the Weka Explorer allows you to select one of the following four options:

Use training set The classifier will be evaluated on how well it predicts the class of the instances it was trained on.

Supplied test set The classifier will be evaluated on how well it predicts the class of a set of instances loaded from a file.

Cross-validation The classifier will be evaluated by cross-validation, using the number of folds that are entered in the Folds text field. Cross-validation (Kohavi, 1995) is more complicated. We will study cross-validation in detail later in this module.

Percentage split The classifier will be evaluated on how well it predicts a certain percentage of the data which is held out for testing. The amount of data held out depends on the value entered in the % field. The value in the % field is the proportion of the data that is used during training. The remainder is reserved for testing.

6.4 Run Naïve Bayes on the Medical Diagnosis Dataset

1. Create a Weka input file called medical.arff containing the training data for the medical diagnosis problem shown in Section 5.3 on page 31. (Recall from Section 4.3 on page 26 that Weka requires input to be in *ARFF* format.)

2. Run NaïveBayesSimple on this dataset. Compare the model generated with the table of likelihoods which you obtained during the tutorial. (Ignore the part of the output about the evaluation on training set because in a moment you are going to evaluate the model on independent test data.)

3. Next you are going to apply the resulting classifier to solve the following three problems:

   Problem 1:
   Temperature = low
   Skin = normal
   Blood pressure = high
   Blocked nose = true

   Problem 2:
   Temperature = low
   Skin =?
   Blood pressure = normal
   Blocked nose = true

   Problem 3:
   Temperature = moderate
   Skin = normal
   Blood pressure = high
   Blocked nose = true

   where ? indicates that the value for that attribute is missing. Notice that these are the three problems that you solved during the tutorial.

4. Create a file called medical_test_data.arff which contains the three test examples. This file must be in *ARFF* format too. Table 6.1 shows such a test file. Notice that the classes of the examples are those given in the solutions to the tutorial.

5. Select the option ‘Supplied Test Set’ on the Test options pane on the Classify tab of the Weka Explorer. Click on the ‘Set’ button next to this option. A window called ‘Test Instances’ will pop-up. Click on ‘Open File’ and load in your file called medical_test_data.arff.

6. Click on the “More Options” button on the Test options pane. Click on the button next to ’Output predictions’ and select ”Plain text”.

7. Click ‘Start’.

8. Compare the predictions with those you obtained during the tutorial. You should see that both the normalised probabilities and the predicted classes are the same as those in the tutorial.

9. Compare the execution time of Weka with the time it took you to manually execute the algorithm during the tutorial.

```
@relation medical
@attribute Temperature {Low,Moderate,High}
@attribute Skin {Pale,Normal,Red}
@attribute BloodPressure {Normal,High}
@attribute BlockedNose {True,False}
@attribute Diagnosis {N,B}
@data
Low,Normal,High,True,N
Low,?,Normal,True,B
Moderate,Normal,High,True,B
```

Table 6.1: Medical diagnosis test data in ARFF.

6.5 Accessing Previous Results

Weka keeps track of every set of results you create so you can try lots of different algorithms and parameters and Weka will keep the results so that you can just go back to a previous result set without having to do things twice. These previous results can be accessed from the results list on the left hand side of the Classify tab. Weka keeps these results until you exit from Weka.

Chapter 7

Tutorial: Nearest Neighbour

7.1 Aim of the Exercise

The aim of this exercise is for you to gain experience of explaining how instance-based learning can be used for classification.

7.2 History of the k–Nearest Neighbour Algorithm

In instance-based learning, the training examples are stored verbatim, and a distance function is used to determine which member of the training set is closest to an unknown test instance. As the learner does not bother to generate a model, instance-based learning is sometimes called lazy learning.

The exercise uses a famous, well-respected but simple instance-based learning algorithm called k–Nearest Neighbour. This algorithm originated many decades ago. Statisticians analysed k-nearest neighbour methods in the early 1950s and it was first applied to classification in 1961 (Johns, 1961). Nearest neighbour methods subsequently gained popularity in the machine learning community (Aha, 1991).

The exercise measures proximity using Euclidean distance, which is calculated using the formula below.

$$\sqrt{(a_1 - a'_1)^2 + (a_2 - a'_2)^2 + \ldots + (a_n - a'_n)^2}$$

where a and a' are two examples with n attributes and a_i is the value of attribute i for a. Euclidean distance is named after Eucleides, a Greek mathematician who taught at Alexandria about the year 300BC.

7.3 Exercise

During this tutorial we will apply the k–Nearest Neighbour algorithm to the weather training data with numeric values (see Section 4.2 on page 25). For your convenience, the data is listed again in Table 7.1.

1. (a) Normalise the numeric attributes.

   (b) Using the Euclidean distance and the 3–NN method, find the solution to:

   outlook = sunny
   temperature = 72
   humidity = 76
   windy = true

2. Assume that the value of humidity is missing for the first instance and that k=3. Using the Euclidean distance and the 3-NN method, find the solution to:

   outlook = sunny
   temperature = ?
   humidity = 76
   windy = true

outlook	temperature	humidity	windy	play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
rainy	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rainy	71	91	TRUE	no

Table 7.1: Weather training data with some numeric attributes.

Chapter 8

Workshop: Nearest Neighbour

8.1 Aim of the Exercise

The aim of this exercise is that you will learn how to use Weka to execute the k–Nearest Neighbour algorithm that we studied during the lecture and tutorial.

8.2 Apply k–Nearest Neighbour to the Weather Dataset

1. Click on the ‘Open file’ button on the Preprocess tab of the Explorer interface and load in the weather training data containing numeric values. If you cannot remember where to access the dataset then you need to read Section 4.2 on page 25 again.

2. Recall that instance-based learning is sometimes called lazy learning because the learner does not bother to generate a model. So click on the Classify tab and select ‘lazy’ from the choose button. The k–Nearest Neighbour algorithm that we studied during the lecture and tutorial is called IBk in Weka. So select IBk from the list of lazy learners.

3. Set k to 3 by clicking on the box to the right of the choose button. Enter 3 in the relevant part of the pop-up window and click the ‘OK’ button.

4. Next you are going to to solve the following two problems:

   Problem 1:
   outlook = sunny
   temperature = 72
   humidity = 76
   windy = true

Problem 2:

outlook = sunny
temperature = ?
humidity = 76
windy = true

where ? indicates that the value for that attribute is missing. Notice that these are the two problems that you solved during the tutorial.

5. Create a file called weather_test_data_for_knn.arff which contains the two test examples. This file must be in *ARFF* format too. (If you have forgotten how to create a file in this format then read Section 4.3 on page 26 again.) Such a test file is shown in Table 8.1. Notice that the classes of the examples are those given in the solutions to the tutorial.

6. Select the option 'Supplied Test Set' on the Test options pane on the Classify tab of the Weka Explorer. Click on the 'Set' button next to this option. A window called 'Test Instances' will pop-up. Click on 'Open File' and load in your file called weather_test_data_for_knn.arff.

7. Click on the 'More Options' button on the Test options pane. Click on the button next to 'Output predictions' and select 'Plain text'. Turn off the other options.

8. Click 'Start'.

9. Compare the predictions with those you obtained during the tutorial. You should see that the predicted classes are the same as those in the tutorial.

10. Compare the execution time of Weka with the time it took you to manually execute the algorithm.

```
@relation weather
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,72,76,TRUE,yes
sunny,?,76,TRUE,yes
```

Table 8.1: Weather test data (with numeric values) in ARFF.

8.3 The UCI Machine Learning Repository

The UCI machine learning repository (Kelly et al., 2023) is a well-organised, long-standing repository of data-sets, together with descriptions of them, used by the Machine Learning community. The UCI repository is available at:

https://archive.ics.uci.edu

The interface to this repository can be used to focus the search for a suitable dataset.

- Select a dataset which you find interesting.
- Convert the dataset to ARFF format.
- Load the dataset into Weka.
- Apply one or more Data Mining algorithms to the dataset.
- Analyse the results.

Chapter 9

Tutorial: PRISM

9.1 Aim of the Exercise

The aim of this exercise is for you to gain experience of explaining how a covering algorithm can generate a set of classification rules.

9.2 The Separate-and-conquer Approach

The exercise uses a simple covering algorithm called PRISM (Cendrowska, 1987). PRISM deals with each class separately. For each class, it uses a *separate-and-conquer* approach: a rule is generated that covers many instances in the class, the covered instances are separated out because they are already taken care of by the rule, and the process continues on those examples which are left.

The separate-and-conquer approach contrasts with the divide-and-conquer approach to decision tree induction where the tree is generated from the top down, selecting an attribute for each node which best separates the classes and then recursively processing the subproblems that result from the split.

9.3 Loan Application Training Data

Suppose a bank lends money to customers. The bank wants to avoid lending money to customers who may not pay the money back later. Therefore, before agreeing to lend money to a particular customer, it assesses the risk of doing so. The manager of the bank wants to review how such risk assessments have been conducted in the past. He has therefore decided to hire you as a Data Miner to find out.

The bank has the data shown in Table 9.1 on previous customers who have undergone a risk assessment. Each of these customers was assessed as either high, moderate or low risk.

	Credit History	Debt	Collateral	Income	Risk
1	bad	high	none	\$0 to \$15k	high
2	unknown	high	none	\$15 to \$35k	high
3	unknown	low	none	\$15 to \$35k	moderate
4	unknown	low	none	\$0 to \$15k	high
5	unknown	low	none	over \$35k	low
6	unknown	low	adequate	over \$35k	low
7	bad	low	none	\$0 to \$15k	high
8	bad	low	adequate	over \$35k	moderate
9	good	low	none	over \$35k	low
10	good	high	adequate	over \$35k	low
11	good	high	none	\$0 to \$15k	high
12	good	high	none	\$15 to \$35k	moderate
13	good	high	none	over \$35k	low
14	bad	high	none	\$15 to \$35k	high

Table 9.1: Credit risk training data

9.4 Exercise

Suppose you decide to generate a rule-set from the training data shown in Table 9.1 using the PRISM algorithm. Unfortunately your computer is broken so you cannot use Weka. Manually apply the PRISM algorithm to find rules for predicting when risk is high. Write down in full all of the steps involved.

Chapter 10

Workshop: PRISM

10.1 Aim of the Exercises

The aim of this exercise is that you will learn how to use Weka to generate a set of classification rules.

For the sake of simplicity, throughout these exercises, we will use the same examples for training and testing. So always select the test option “Use training set” on the classifier pane of Weka Explorer.

10.2 Exercise: Contact Lenses

During the lecture we manually applied the PRISM algorithm to data on contact lenses. The PRISM algorithm is available from Weka. Use Weka’s implementation of PRISM to generate a rule-set from all 24 examples in the contact lenses dataset.

1. Download the file contact_lens.arff from Blackboard.

2. Load this dataset into Weka.

3. Go to the Classify tab and select ‘rules’ from the choose button.

4. Select Prism.

5. Click on the ‘More Options’ button on the Test options pane. To reduce the amount of output from Weka to just what we need, turn off the option ‘Output per-class stats’ by removing the tick next to it and click OK.

6. Press ‘Start’ and study the output.

7. Compare the rules for predicting when to prescribe hard contact lenses with the ones we generated manually.

8. Compare the execution time of Weka with the time it took to manually execute the PRISM algorithm during the lecture.

The output from Weka includes the following which you can ignore because they lie beyond the scope of this module.

- the Kappa statistic,
- the mean absolute error,
- the root mean-squared error,
- relative absolute error, and
- the root relative squared error.

10.3 Exercise: Applying for a Loan

During the tutorial in Chapter 9, you manually applied the PRISM algorithm to data on credit risk assessments.

1. Download the file loan.arff from Blackboard.
2. Load this dataset into Weka.
3. Use Weka's implementation of PRISM to generate a rule-set from all 14 examples.
4. Compare the rules for predicting when risk is high with the ones you generated manually.
5. Compare the execution time of Weka with the time it took you to manually execute the PRISM algorithm during the tutorial.

10.4 Exercise: The Weather Dataset

Try to instruct Weka to use its implementation of PRISM on the weather data set by selecting the classifier 'Choose' button. When you look under rules, you will see that the word PRISM is shown in a grey font rather than the usual black. Weka is preventing you from trying to apply PRISM to the weather dataset containing numeric values. Why? Write down your answer in the box below,

Your solution:

10.4.1 Discretisation

Go back to the preprocess screen and click on the choose button under filters. Make the following selections.
filters ⟶ unsupervised ⟶ attribute ⟶ discretize
If we click on the name of the filter once it is loaded we can change some parameters. The one we are most interested in is the "bin" parameter; this tells the algorithm how many groups we want to split our continuous data into. Choose 5 for now. Then click on apply. If you now click on the attributes you will notice that temperature and humidity have been put into ranges.

Go back and try the PRISM rule algorithm again and observe what happens. Weka should output the rules shown in Table 10.1.

```
If outlook = overcast then yes
If temperature = '(72.4-76.6)' then yes
If humidity = '(77.4-83.6)' then yes
If temperature = '(68.2-72.4)' and humidity = '(-inf-71.2)' then yes
If outlook = rainy and windy = FALSE then yes
If temperature = '(76.6-80.8)' then no
If outlook = sunny and temperature = '(80.8-inf)' then no
If humidity = '(89.8-inf)' and outlook = sunny then no
If outlook = rainy and windy = TRUE then no
```

Table 10.1: Rules Generated by PRISM from Weather dataset.

10.5 KD Nuggets

KD Nuggets lists:

- Companies in Analytics, Data Mining, and Data Science.
- Jobs in Data Mining and Analytics.
- Data Repositories.

KD Nuggets is available at:

http://www.kdnuggets.com/

10.5.1 Optional Exercise

- Select a company which you find interesting.
- Find a job you find appealing.
- Start to plan you career in Data Mining.

Chapter 11

Lecture: Decision Trees: ID3

11.1 Loan Application Training Data

Consider again the loan application problem described in Section 9.3 on page 43. For your convenience, the training data is listed again in Table 11.1.

	Credit History	Debt	Collateral	Income	Risk
1	bad	high	none	\$0 to \$15k	high
2	unknown	high	none	\$15 to \$35k	high
3	unknown	low	none	\$15 to \$35k	moderate
4	unknown	low	none	\$0 to \$15k	high
5	unknown	low	none	over \$35k	low
6	unknown	low	adequate	over \$35k	low
7	bad	low	none	\$0 to \$15k	high
8	bad	low	adequate	over \$35k	moderate
9	good	low	none	over \$35k	low
10	good	high	adequate	over \$35k	low
11	good	high	none	\$0 to \$15k	high
12	good	high	none	\$15 to \$35k	moderate
13	good	high	none	over \$35k	low
14	bad	high	none	\$15 to \$35k	high

Table 11.1: Credit risk training data.

Suppose that you decide to generate a decision tree from this training data using the ID3 algorithm. Unfortunately your computer is broken so you cannot use Weka. So you decide to do it manually instead. The remainder of this chapter shows some of the calculations that you would need to do.

11.2 Calculating the Entropy of the Class Attribute

Let's begin by calculating the entropy (information) before splitting. Recall that the formula for calculating entropy is:

$$I(p_1, p_2, \ldots p_n) = -p_1 \times \log_2 p_1 - p_2 \times \log_2 p_2 \ldots - p_n \times \log_2 p_n$$

where n is the number of classes. We are going to apply this formula to calculate the entropy of the class attribute, Risk.

Before splitting, there are 14 examples, so the denominator in all of the fractions is 14. Of these 14 examples, there are:

- 6 examples where Risk = high,
- 3 examples where Risk = moderate and
- 5 examples where Risk = low.

So, the amount of information for the class attribute, Risk, is:

$$I(Risk) = -\frac{6}{14} \times \log_2 \frac{6}{14} - \frac{3}{14} \times \log_2 \frac{3}{14} - \frac{5}{14} \times \log_2 \frac{5}{14} = 1.531\ bits$$

We are going to calculate the Information Gain for each of the other attributes. We start with the attribute Income.

11.3 Calculating the Gain for Income

If Income is the root of the tree, we have three subsets of the set of 14 examples:

- {1, 4, 7, 11}, i.e. the set of examples where Income is \$0 to \$15k;
- {2, 3, 12, 14}, i.e. the set of examples where Income is \$15 to \$35k;
- {5, 6, 8, 9, 10, 13}, i.e. the set of examples where Income is over \$35k.

So, after on splitting Income, four examples go down the branch \$0 to \$15k, four examples go down the branch \$15 to \$35k and six examples go down the branch over \$35k.

We are going to apply the formula for entropy to each of the three values of Income. For each one, the denominator in all of the fractions is the number of examples which go down that branch.

Of the 4 examples which go down the branch \$0 to \$15k, there are:

- 4 examples where Risk = high,

- 0 examples where Risk = moderate and
- 0 examples where Risk = low.

So, the amount of information for the value \$0 to \$15k is:

$$\begin{aligned} I(Income = \$0 to \$15k) &= -\frac{4}{4} \times \log_2 \frac{4}{4} - \frac{0}{4} \times \log_2 \frac{0}{4} - \frac{0}{4} \times \log_2 \frac{0}{4} \\ &= -1 \times 0 - 0 \times \log_2 \frac{0}{4} - 0 \times \log_2 \frac{0}{4} \\ &= 0\ bits \end{aligned}$$

Of the 4 examples which go down the branch \$15 to \$35k, there are:

- 2 examples where Risk = high,
- 2 examples where Risk = moderate and
- 0 examples where Risk = low.

So, the amount of information for the value \$15 to \$35k is:

$$\begin{aligned} I(Income = \$15 to \$35k) &= -\frac{2}{4} \times \log_2 \frac{2}{4} - \frac{2}{4} \times \log_2 \frac{2}{4} - \frac{0}{4} \times \log_2 \frac{0}{4} \\ &= -0.5 \times -1 - 0.5 \times -1 - 0 \times \log_2 \frac{0}{4} \\ &= 0.5 + 0.5 + 0 \\ &= 1\ bits \end{aligned}$$

Of the 6 examples which go down the branch over \$35, there are:

- 0 examples where Risk = high,
- 1 examples where Risk = moderate and
- 5 examples where Risk = low.

So, the amount of information for the value over \$35k is:

$$\begin{aligned} I(Income = over \$35k) &= -\frac{0}{6} \times \log_2 \frac{0}{6} - \frac{1}{6} \times \log_2 \frac{1}{6} - \frac{5}{6} \times \log_2 \frac{5}{6} \\ &= 0 \times \log_2 \frac{0}{6} + 0.43083 + 0.2193 \\ &= 0.6501\ bits \end{aligned}$$

We now calculate E(Income), i.e., the expected information of the attribute Income. E(Income) is the weighted average of the entropies for each value of Income. The weighting for each branch is the fraction of examples which go down that branch. Four go down the branch $0 to $15k, four go down the branch $15 to $35k and six go down the branch over $35k. So the weighted average is:

$$\begin{aligned}E(Income) &= \frac{4}{14}\times I(\$0to\$15k)+\frac{4}{14}\times I(\$15to\$35k)+\frac{6}{14}\times I(over\$35k)\\ &= \frac{4}{14}\times 0+\frac{4}{14}\times 1+\frac{6}{14}\times 0.6501\\ &= 0.564\,bits\end{aligned}$$

Recall that the information gain is equal to the amount of information before splitting minus the amount of information after splitting. Expressing this mathematically gives:

$$\begin{aligned}gain(Income) &= I(Risk)-E(Income)\\ &= 1.531-0.564\\ &= 0.967\,bits\end{aligned}$$

Next we calculate the Information Gain for the attribute Credit History.

11.4 Calculating the Gain for Credit History

If Credit History is the root of the tree, we have three subsets of the set of 14 examples:

- {1, 7, 8, 14}, i.e. the set of examples where Credit History is bad;
- {9, 10, 11, 12, 13}, i.e. the set of examples where Credit History is good;
- {2, 3, 4, 5, 6}, i.e. the set of examples where Credit History is unknown.

So, after splitting on Credit History, four examples go down the branch bad, five examples go down the branch good and five examples go down the branch unknown.

We are going to apply the formula for entropy to each of the three values of Credit History. For each one, the denominator for all of the fractions is the number of examples which go down that branch.

$$\begin{aligned}I(CreditHistory=bad) &= -\frac{3}{4}\times\log_2\frac{3}{4}-\frac{1}{4}\times\log_2\frac{1}{4}-\frac{0}{4}\times\log_2\frac{0}{4}\\ &= -0.75\times -0.4150-0.25\times -2-0\times\log_2\frac{0}{4}\\ &= 0.31125+0.5-0\\ &= 0.81125\,bits\end{aligned}$$

$$\begin{aligned}I(CreditHistory=good) &= -\frac{1}{5}\times\log_2\frac{1}{5}-\frac{1}{5}\times\log_2\frac{1}{5}-\frac{3}{5}\times\log_2\frac{3}{5}\\ &= -0.2\times -2.3220-0.2\times -2.3220-0.6\times -0.7370\\ &= 0.4644+0.4644+0.4422\\ &= 1.371\,bits\end{aligned}$$

$$\begin{aligned} I(CreditHistory = unknown) &= -\frac{2}{5} \times \log_2 \frac{2}{5} - \frac{1}{5} \times \log_2 \frac{1}{5} - \frac{2}{5} \times \log_2 \frac{2}{5} \\ &= -0.4 \times -1.3220 + -0.2 \times -2.3220 + -0.4 \times -1.3220 \\ &= 0.5288 + 0.4644 + 0.5288 \\ &= 1.522\, bits \end{aligned}$$

We now calculate E(Credit History), i.e., the expected information of the attribute Credit History. E(Credit History) is the weighted average of the entropies for each value of Credit History. The weighting for each branch is the fraction of examples which go down that branch. Four go down the branch bad, five go down the branch good and five go down the branch unknown. So the weighted average is:

$$\begin{aligned} E(CreditHistory) &= \frac{4}{14} \times I(bad) + \frac{5}{14} \times I(good) + \frac{5}{14} \times I(unknown) \\ &= \frac{4}{14} \times 0.81125 + \frac{5}{14} \times 1.371 + \frac{5}{14} \times 1.522 \\ &= 1.2643\, bits \end{aligned}$$

Hence the information gain is:

$$\begin{aligned} gain(CreditHistory) &= I(Risk) - E(CreditHistory) \\ &= 1.531 - 1.2643 \\ &= 0.2667\, bits \end{aligned}$$

11.5 Calculating the Gain for Debt

If Debt is the root of the tree, we have two subsets of the set of 14 examples:

- {1, 2, 10, 11, 12, 13, 14}, i.e. the set of examples where Debt is high;
- {3, 4, 5, 6, 7, 8, 9 }, i.e. the set of examples where Debt is low.

So, after splitting on Debt, seven examples go down the branch high and seven examples go down the branch low.

We are going to apply the formula for entropy to both of the values of Debt. For each one, the denominator for all of the fractions is the number of examples which go down that branch.

$$\begin{aligned} I(Debt = high) &= -\frac{4}{7} \times \log_2 \frac{4}{7} - \frac{1}{7} \times \log_2 \frac{1}{7} - \frac{2}{7} \times \log_2 \frac{2}{7} \\ &= 1.379\, bits \end{aligned}$$

$$\begin{aligned} I(Debt = low) &= -\frac{2}{7} \times \log_2 \frac{2}{7} - \frac{2}{7} \times \log_2 \frac{2}{7} - \frac{3}{7} \times \log_2 \frac{3}{7} \\ &= 1.557\, bits \end{aligned}$$

We now calculate E(Debt), i.e., the expected information of the attribute Debt. E(Debt) is the weighted average of the entropies for each value of Debt. The weighting for each branch is the fraction of examples which go down that branch. Seven go down the branch high and seven go down the branch low. So the weighted average is:

$$\begin{aligned} E(Debt) &= \frac{7}{14} \times I(high) + \frac{7}{14} \times I(low) \\ &= \frac{7}{14} \times 1.379 + \frac{7}{14} \times 1.557 \\ &= 1.468\, bits \end{aligned}$$

Hence the information gain is:

$$\begin{aligned} gain(Debt) &= I(Risk) - E(Debt) \\ &= 1.531 - 1.468 \\ &= 0.063\, bits \end{aligned}$$

11.6 Calculating the Gain for Collateral

If Collateral is the root of the tree, we have two subsets of the set of 14 examples:

- {1, 2, 3, 4, 5, 7, 9, 11, 12, 13, 14}, i.e. the set of examples where Collateral is none;
- {6, 8, 10 }, i.e. the set of examples where Collateral is adequate.

So, after splitting on Collateral, eleven examples go down the branch none and three examples go down the branch adequate.

We are going to apply the formula for entropy to both of the values of Collateral. For each one, the denominator for all of the fractions is the number of examples which go down that branch.

$$\begin{aligned} I(Collateral = none) &= -\frac{6}{11} \times \log_2 \frac{6}{11} - \frac{2}{11} \times \log_2 \frac{2}{11} - \frac{3}{11} \times \log_2 \frac{3}{11} \\ &= 1.435\, bits \end{aligned}$$

$$\begin{aligned} I(Collateral = adequate) &= -\frac{0}{3} \times \log_2 \frac{0}{3} - \frac{1}{3} \times \log_2 \frac{1}{3} - \frac{2}{3} \times \log_2 \frac{2}{3} \\ &= 0.918\, bits \end{aligned}$$

We now calculate E(Collateral), i.e., the expected information of the attribute Collateral. E(Collateral) is the weighted average of the entropies for each value of Collateral. The weighting for each branch is the fraction of examples which go down that branch. Eleven go down the branch none and three go down the branch adequate. So the weighted average is:

$$\begin{aligned} E(Collateral) &= \frac{11}{14} \times I(none) + \frac{3}{14} \times I(adequate) \\ &= \frac{11}{14} \times 1.435 + \frac{3}{14} \times 0.918 \\ &= 1.324\, bits \end{aligned}$$

Hence the information gain is:

$$\begin{aligned} gain(Collateral) &= I(Risk) - E(Collateral) \\ &= 1.531 - 1.324 \\ &= 0.207\, bits \end{aligned}$$

11.7 Building the Tree

$$\begin{aligned} gain(Income) &= 0.967\, bits \\ gain(CreditHistory) &= 0.2667\, bits \\ gain(Debt) &= 0.063\, bits \\ gain(Collateral) &= 0.207\, bits \end{aligned}$$

From the above calculations, we can see that making Income the root of the tree provides the highest information gain. Hence, we make Income the root of the decision tree. We then apply the algorithm to each of the subsets shown in Section 11.3. The final result of applying ID3 to the credit risk dataset is shown in Figure 11.1. Notice that the set of examples where income is $0 to $15k all belong to the same class, i.e., high risk. Recall that one of the stopping conditions of the ID3 algorithm is all the examples have the same class. Therefore, this branch does not need to be extended further. Instead a leaf node is created with the class high risk.

However the other two branches are not pure, i.e., the examples which go down their branches do not belong to a single class. So the algorithm is applied in a recursive manner to each of the other two branches.

11.8 Extending the Branch Income = over $35k

This section describes how ID3 is called recursively to build the subtree for the branch Income = over $35k. Let's begin by calculating the entropy (information) before splitting, i.e., the entropy of Income = over $35k. Of the six examples which go down this branch,

- none are high risk,

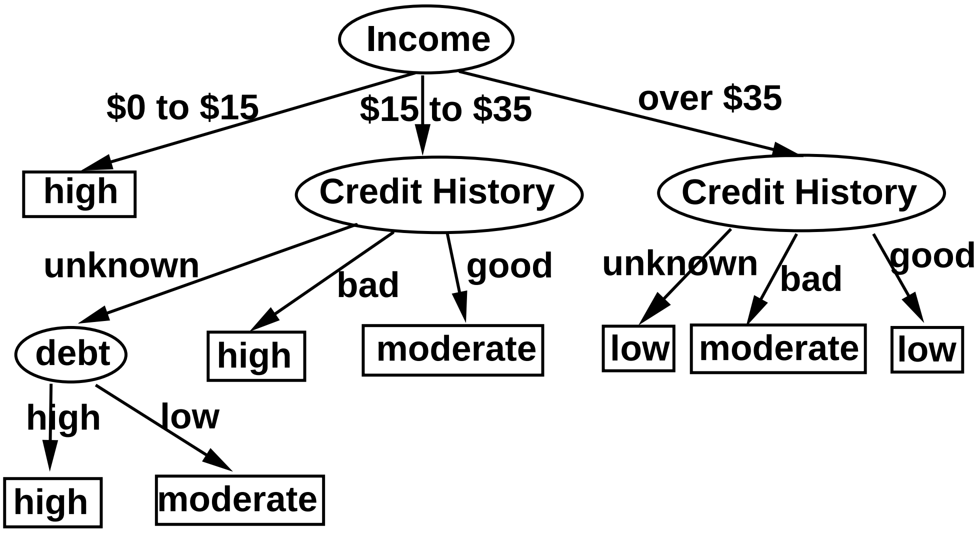


Figure 11.1: Decision tree generated by ID3 from the credit risk dataset.

- one is a moderate risk and
- five are low risk.

$$\begin{aligned} I(Income = over\$35k) &= -\frac{0}{6} \times \log_2 \frac{0}{6} - \frac{1}{6} \times \log_2 \frac{1}{6} - \frac{5}{6} \times \log_2 \frac{5}{6} \\ &= 0 \times \log_2 \frac{0}{6} + 0.43083 + 0.2193 \\ &= 0.6501\ bits \end{aligned}$$

We are going to calculate the Information Gain for each of the other attributes except Income. We no longer take Income into account, since all the examples in a subset have the same value for Income.

If Credit History is selected as the root of the subtree, we have three subsets of the set of examples:

- {8}, i.e. the set of examples where Credit History is bad;
- {9, 10, 13}, i.e. the set of examples where Credit History is good;
- {5, 6}, i.e. the set of examples where Credit History is unknown.

So, after splitting on Credit History, one example goes down the branch bad, three examples go down the branch good and two examples go down the branch unknown.

We apply the formula for entropy to each of the three values of Credit History. For each one, the denominator for all of the fractions is the number of examples which go down that branch.

$$\begin{aligned} I(CreditHistory = bad) &= -\frac{0}{1} \times \log_2 \frac{0}{1} - \frac{0}{1} \times \log_2 \frac{0}{1} - \frac{1}{1} \times \log_2 \frac{1}{1} \\ &= 0 - 0 - 1 \times \log_2 1 \\ &= 0\ bits \end{aligned}$$

$$\begin{aligned} I(CreditHistory = good) &= -\frac{0}{3} \times \log_2 \frac{0}{3} - \frac{0}{3} \times \log_2 \frac{0}{3} - \frac{3}{3} \times \log_2 \frac{3}{3} \\ &= 0 - 0 - 1 \times \log_2 1 \\ &= 0\ bits \end{aligned}$$

$$\begin{aligned} I(CreditHistory = unknown) &= -\frac{2}{2} \times \log_2 \frac{2}{2} - \frac{0}{2} \times \log_2 \frac{0}{2} - \frac{0}{2} \times \log_2 \frac{0}{2} \\ &= -1 \times \log_2 1 + 0 + 0 \\ &= 0\ bits \end{aligned}$$

Next, we calculate E(Credit History), i.e., the expected information of the attribute Credit History. E(Credit History) is the weighted average of the entropies for each value of Credit History.

The weighting for each branch is the fraction of examples which go down that branch. One goes down the branch bad, three go down the branch good and two go down the branch unknown. So the weighted average is:

$$\begin{aligned} E(CreditHistory) &= \frac{1}{6} \times I(bad) + \frac{3}{6} \times I(good) + \frac{2}{6} \times I(unknown) \\ &= \frac{1}{6} \times 0 + \frac{3}{6} \times 0 + \frac{2}{6} \times 0 \\ &= 0\ bits \end{aligned}$$

Hence the information gain is:

$$\begin{aligned} gain(CreditHistory) &= I(Income = over\$35k) - E(CreditHistory) \\ &= 0.6501 - 0 \\ &= 0.6501\ bits \end{aligned}$$

Notice that E(Credit History) = 0 bits, i.e, the node is pure. Zero is the minimal value for entropy. Neither of the other two attributes (Debt and Collateral) can beat this, although they could match it. If Debt or Collateral did match it, then an arbitrary choice would have to be made and we could select Credit History. So there is no need do any further calculations on this branch. Credit History is selected for this node.

This branch of the tree does not need to be grown further for the following reasons.

- The set of examples where $Income = over\$35k$ and $CreditHistory = good$ all belong to the same class, i.e., low risk. Therefore, this branch does not need to be extended further. Instead a leaf node is created with the class low risk.
- The set of examples where $Income = over\$35k$ and $CreditHistory = bad$ all belong to the same class, i.e., moderate risk. Therefore, this branch does not need to be extended further. Instead a leaf node is created with the class moderate risk.
- The set of examples where $Income = over\$35k$ and $CreditHistory = unknown$ all belong to the same class, i.e., low risk. Therefore, this branch does not need to be extended further. Instead a leaf node is created with the class low risk.

Chapter 12

Workshop: ID3 and J48

12.1 Aim of the Exercise

The aim of this exercise is that you will learn how to use Weka to generate both decision trees and an analysis of those trees.

12.2 Top-down Induction of Decision Trees

The divide-and-conquer approach to decision tree induction is sometimes called top-down induction of decision trees.

The first lecture on constructing decision trees described how to maximise information gain. This approach is essentially the ID3 algorithm (Quinlan, 1986) developed by Ross Quinlan, a Machine learning researcher from 1970's. Four statisticians published a similar approach to Quinlan's (Breiman et al., 1984). These statisticians and Quinlan only became aware of one another's work much later one.

The ID3 algorithm is available from Weka. You will use Weka's implementation of ID3 during the workshop exercise in Section 12.3.

A series of improvements to ID3 culminated in a practical and influential system for decision tree induction called C4.5 (Quinlan, 1993). The J4.8 algorithm is Weka's implementation of the C4.5 decision tree learner. You will use J4.8 during the workshop exercise in Section 12.4 on page 60. (In fact, J4.8 actually implements a later and slightly improved version called C4.5 revision 8, which was the last public version of this family of algorithms before the commercial implementation C5.0 was released.)

12.3 Exercise: Applying for a Loan

Let's begin by using Weka's implementation of ID3 to generate a decision tree from all of the 14 examples in the dataset on credit risk assessments (see Section 9.3).

1. Load the Loan dataset into Weka by clicking on the ‘Open file’ button on the Preprocess tab of the Explorer interface.
2. Click on the Classify tab and select ‘tree’ from the choose button.
3. Select ID3 from the list of decision tree algorithms.
4. Select the option ‘Use training set’ on the Test options pane on the Classify tab of the Weka Explorer.
5. Click on the ‘More Options’ button on the Test options pane. To reduce the amount of output from Weka to just what we need, turn off the option ‘Output per-class stats’ by removing the tick next to it and click OK.
6. Press ‘Start’ and study the output.
7. Compare the execution time of Weka with the time it took to manually execute the algorithm during the lecture.

The output should include the textual representation of the tree shown in Table 12.1. Compare the textual representation with Figure 11.1 on page 56. Are they equivalent?

```
Income = $0_to_$15k: high
Income = $15_to_$35k
|       CreditHistory = bad: high
|       CreditHistory = good: moderate
|       CreditHistory = unknown
|       |       Debt = high: high
|       |       Debt = low: moderate
Income = over_$35k
|       CreditHistory = bad: moderate
|       CreditHistory = good: low
|       CreditHistory = unknown: low
```

Table 12.1: Decision tree generated by ID3 from the credit risk dataset.

12.4 Generate a Tree from the Weather Dataset

1. Load the weather dataset containing numeric values.
2. Select the J48 algorithm, as shown in Figure 12.1.
3. Select the test option ‘Use training set’.

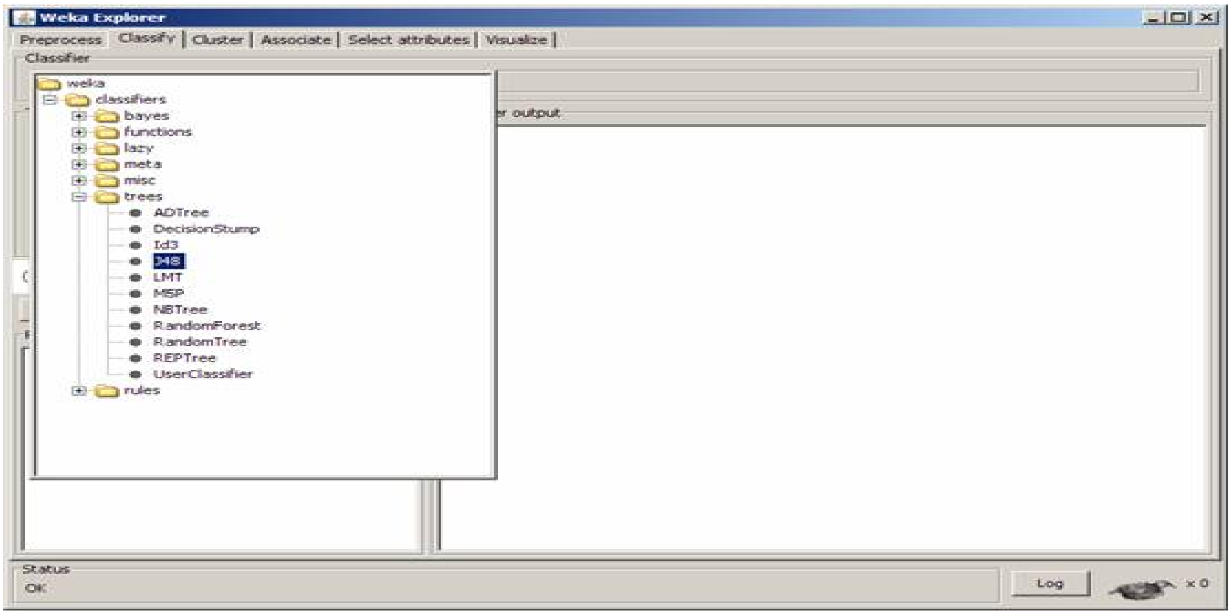


Figure 12.1: Selecting one of Weka's classification algorithms.

4. Press 'Start'.

The output should include the textual representation of the tree shown in Table 12.2.

```
outlook = sunny
|       humidity <= 75: yes (2.0)
|       humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|       windy = TRUE: no (2.0)
|       windy = FALSE: yes (3.0)
```

Table 12.2: Decision tree generated by J48 from the weather dataset.

The textual representation of the tree shown above does not look like a tree. This style of representing trees could be confusing for large trees. Fortunately Weka will generate a clearer view (see Figure 12.2) if we right click on the entry in result list and select 'visualize tree'. The picture shows us that the first split is outlook. If it is sunny then we are interested in humidity; if it is rainy then we are interested in whether it is windy or not; and if the outlook is overcast we always play so there are no more decisions to make.

Underneath the textual representation of the tree, Weka lists some statistics. The first two tell us the tree has five leaf nodes and 8 nodes in total. Check that you understand what these mean by looking at Weka's graphical depiction of the tree. Weka follows the usual convention in Data Mining of drawing trees upside down. So the root node of the tree is shown at the top of the picture and the leaves are shown at the bottom. The leaf nodes are shown inside rectangles in the picture.

Look again at the textual representation of the tree. There are some numbers in round parentheses. What do these numbers mean?

Your answer:

Count how many of the examples in the training set go down each branch of the tree.

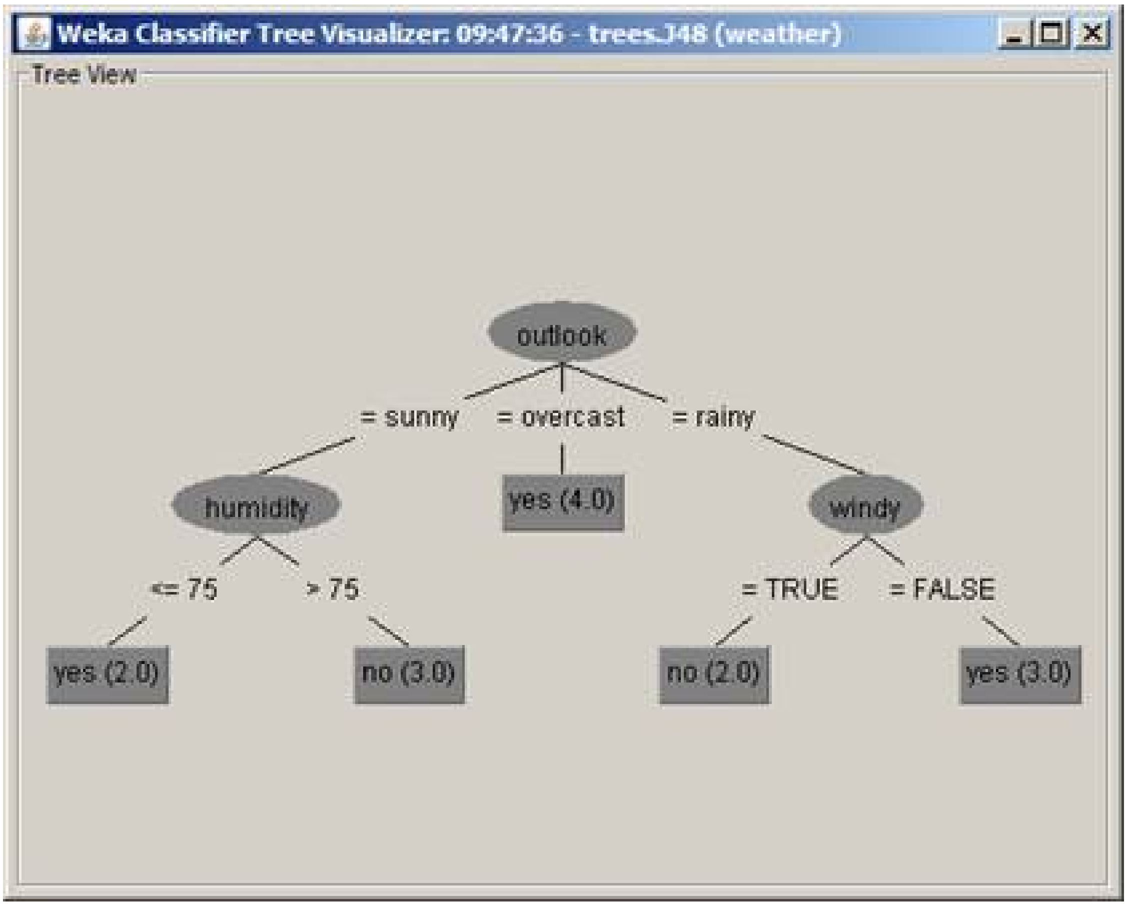


Figure 12.2: Decision tree generated by J48 from the weather set dataset.

12.5 Measuring the Performance of the Decision Tree

12.5.1 Confusion Matrix

A confusion matrix shows how often a classifier is making different types of errors. (Confusion matrices are sometimes called contingency tables.) For a dataset with just two classes, the confusion matrix has two rows and two columns. In general, the cells of a confusion matrix may contain actual counts or percentages. In Weka, they contain actual counts. Table 12.3 shows the general form of a confusion matrix.

TP = Number of positives correctly classified as positive.

FP = Number of negatives falsely classified as positive. These are known as errors of **c**ommission.

TN = Number of negatives correctly classified negative.

FN = Number of positives falsely classified as negative. These are known as errors of omission.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Table 12.3: Confusion matrix or continency table for a dataset with just two classes.

12.5.2 Predictive Accuracy or Success Rate

The performance of a classifier is often measured using predictive accuracy. Predictive accuracy is sometimes referred to as the success rate. We can now give a precise mathematical definition of predictive accuracy in terms of the counts in a confusion matrix.

$$PredictiveAccuracy = 100 \times \frac{TP + TN}{TP + TN + FP + FN} \tag{12.1}$$

12.5.3 Weka's Evaluation

Below the textual representation of the tree, Weka outputs an evaluation of the decision tree. This comprises a list of statistics. The first three lines of Weka's evaluation (see Table 12.4) show that the decision tree correctly predicts the class of all 14 examples.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances       14       100%
```

Table 12.4: First 3 lines of Weka's evaluation.

Next get Weka to generate a confusion matrix by clicking on the 'More Options' button on the Test options pane, ticking the box 'Output confusion matrix' and pressing 'Start' again. If you study the output from Weka then you will see that it includes the confusion matrix shown in Table 12.5. Of nine instances where play = yes in the dataset, the decision tree predicts play=yes for all nine of these. In other words, it got nine right and zero wrong. Of the five instances where play = no in the dataset, the decision tree predicts play = no for five of these. In other words, it got five right and zero wrong. So the tree correctly predicts the class of all the training examples.

```
=== Confusion Matrix ===

a b    <– classified as
9 0 |  a = yes
0 5 |  b = no
```

Table 12.5: Confusion matrix output by Weka.

Recall that we selected the test option "Use training set". This instructed Weka to evaluate the decision tree by recording how well it predicts the class of the examples from which the tree was generated. In other words, to use the same examples for training and testing.

12.5.4 Exercise: Holdout

The purpose of this exercise is to study how holding-out a proportion of the data for testing affects the output of Weka. We are going to run J48 on the weather data-set again but this time with the test option set to Percentage split. So select this option. We will use the default of 66% so leave this unchanged.

Before clicking the start button, you need to take into account a peculiar feature of Weka. The decision tree which is output by Weka Explorer is the one that is generated by J48 given the full training set, regardless of the test option selected. It is not easy to guess this by looking at the Explorer interface. However there is a hint. Click on the "More Options" button on the Test options pane on the Classify tab of the Weka Explorer. Hover the pointer over the first option entitled "Output Model" and read the pop-up.

In this exercise, we are not interested in the tree generated by J48 given the full training set because we are going to hold-out a proportion of the data for testing. So turn off the "Output Model" option.

Now click the start button and study the output. Note that the part of the output entitled “Run information” indicates that there are 14 examples in the dataset. The part of the output entitled “Evaluation on test split” indicates that 5 of these 14 have been used for testing. Make sure you understand why this is. Study the confusion matrix. Calculate the predictive accuracy from it.

12.6 Manually Generating Rules from Trees

The lecture entitled “A Deeper Look at Concept Description” includes a slide entitled “From trees to rules” which explains how a decision tree can be converted to a set of classification rules.

Let’s generate a set of rules from the decision tree that you generated for the weather data set (see Section 12.4 on page 60). In the box below, write down a set of rules to match the tree. Create one rule for each leaf of the tree.

Your solution:

Compare these rules with the ones which PRISM created from the weather dataset (see Section 10.4.1 on page 47). Why are they different?

Your answer:

Chapter 13

Tutorial: Decision Trees: ID3

13.1 Aim of the Exercise

The aim of the exercise in Section 13.3 on page 68 is for you to gain experience of explaining how the ID3 algorithm (Quinlan, 1986) generates decision trees.

13.2 The Father of Information Theory

Claude Shannon (1916-2001) (Wikipedia, 2017) is regarded as the father of Information Theory.

Perhaps more than anyone, he laid the groundwork for today's digital revolution. His exposition of information theory, stating that all information could be represented mathematically as a succession of noughts and ones, facilitated the digital manipulation of data without which today's information society would be unthinkable.

Shannon's master's thesis, obtained in 1940 at MIT, demonstrated that problem solving could be achieved by manipulating the symbols 0 and 1 in a process that could be carried out automatically with electrical circuitry. That dissertation has been hailed as one of the most significant master's theses of the 20th century. Eight years later, Shannon published another landmark paper, A Mathematical Theory of Communication, generally taken as his most important scientific contribution.

Shannon applied the same radical approach to cryptography research, in which he later became a consultant to the US government.

Many of Shannon's pioneering insights were developed before they could be applied in practical form. He was truly a remarkable man, yet unknown to most of the world.

13.3 Exercise: Predicting the Desirability of Accommodation

Assume that you are an estate agent who has the examples of properties shown in Table 13.1 and that you are interested in predicting whether some accommodation is desirable or not.

	Price	Location	State	Desirability
1	Average	Central	OK	Yes
2	High	Countryside	OK	No
3	Low	Central	Good	Yes
4	High	Central	Good	Yes
5	Average	Countryside	OK	Yes
6	Average	Central	Bad	No
7	Low	Countryside	Bad	Yes

Table 13.1: Accommodation dataset.

1. Manually apply the ID3 algorithm to the data in Table 13.1. Write down in full all of the mathematical calculations required. Justify the choice of all the nodes in the decision tree. You may refer to the table of values for $\log_2$ function given in Appendix A on page 109.

2. Draw the decision tree generated by the algorithm.

Chapter 14

Workshop: Clustering

14.1 Aim of the Exercise

To learn how to analyse datasets using Weka's implementation of the k-means clustering algorithm.

14.2 What is Clustering?

Clustering is not used when there is a class to be predicted. Clustering techniques are used when the examples need to be divided into natural groups. By natural, we mean that these clusters reflect some mechanism at work in the domain from which the examples are drawn, a mechanism that causes some examples to bear a stronger resemblance to each other. In this workshop, you will study an algorithm that forms clusters in numeric domains, partitioning the examples into disjoint clusters. It is a simple and straightforward technique that has been used for several decades.

14.3 The SimpleKMeans Algorithm

The k-means clustering algorithm (Hartigan, 1975) involves the following steps:

1. Specify how many clusters, k, are sought.
2. k points are chosen at random as cluster centres.
3. Instances are assigned to their closest cluster centre according to the ordinary Euclidean distance function.
4. Centroid or mean of all instances in each cluster is calculated and these become the new centres.
5. Steps 3 and 4 are repeated until there are no changes.

14.4 Iris Dataset

The iris dataset is one of the most famous datasets in the world. It was created by Sir Ronald Aylmer Fisher in 1936 and has been used in many statistical experiments developing multivariate statistical techniques. It has various attributes for three types of iris (a flower) and the idea is to separate out the three different varieties.

1. Please down-load the file called iris.arff from Blackboard.
2. Load this dataset into Weka.
3. Use the preprocess tab of Weka to help you to get to know the Iris dataset.

14.5 Applying SimpleKMeans to the Iris Dataset

1. Go to the cluster tab and choose SimpleKMeans.
2. Click on the box to the right of the ‘Choose’ button and notice that SimpleKMeans is currently set to create two clusters.
3. Change ‘numClusters’ to a more suitable value. (Recall that the iris dataset has three varieties of iris.)
4. Click OK.
5. Notice that the cluster tab includes the ‘Cluster mode’ pane on the left hand side. Select ‘classes to cluster evaluation’ and select ‘(nom) class’ as the target.
6. Click ‘Start’ and study the output.

14.5.1 Plot the Clusters

To plot the clusters right click in the result list and select ‘visualize cluster assignments’. The clusters are given different colours based on which group they are in. Xs indicate that a point is in the correct cluster whilst a square indicates it is in the wrong cluster. We can look at the clustering around different attributes by altering the attributes in the boxes at the top. Have a look at them and see if you can find the best attribute for creating clusters.

Chapter 15

Tutorial: Evaluation

15.1 Aim of the Exercise

The aim of the exercise in Section 15.2 is for you to gain experience of using statistics to evaluate patterns.

15.2 Exercise

1. Suppose that you use the algorithm PRISM to generate a rule-set for predicting whether a road vehicle is a car, bus, or tram. You then use a test set containing N unseen vehicles to estimate the predictive accuracy. You then use another test set of N unseen vehicles to get another estimate of the predictive accuracy. Finally, you notice that the estimates differ slightly.

   (a) Name the probability distribution that describes this variation.

   (b) Explain how this probability distribution describes this variation.

   (c) Explain why confidence intervals are calculated for estimates of predictive accuracy.

2. Suppose you use a rule-set to classify the examples in a test-set and you obtain the contingency table shown in Table 15.1.

   (a) Estimate the predictive accuracy of the rule-set.

   (b) Find the 80% confidence interval for the predictive accuracy. Confidence limits for the normal distribution for a random variable X which has a mean of 0 and a variance of 1 are shown in Table 15.2.

		Predicted		
		car	bus	tram
Actual	car	150	30	50
	bus	50	180	40
	tram	50	30	420

Table 15.1: Contingency table for road vehicles.

Pr[X≥z]	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
40%	0.25

Table 15.2: Confidence limits for the normal distribution for a random variable X which has a mean of 0 and a variance of 1.

3. In the holdout method of estimating predictive accuracy, a sample of the data is used for training and another independent sample is held over for testing. Describe the following two types of holdout.

   (a) Stratified holdout.

   (b) Repeated holdout.

Chapter 16

Tutorial: Revision

16.1 Aim of the Exercise

The aim of the exercise in Section 16.2 is to help you consolidate your knowledge of data mining.

16.2 Exercise

1. Convert the decision tree shown in Figure 16.1 into a set of rules, where each rule does **not** contain any disjunctions, i.e., logical ORs.

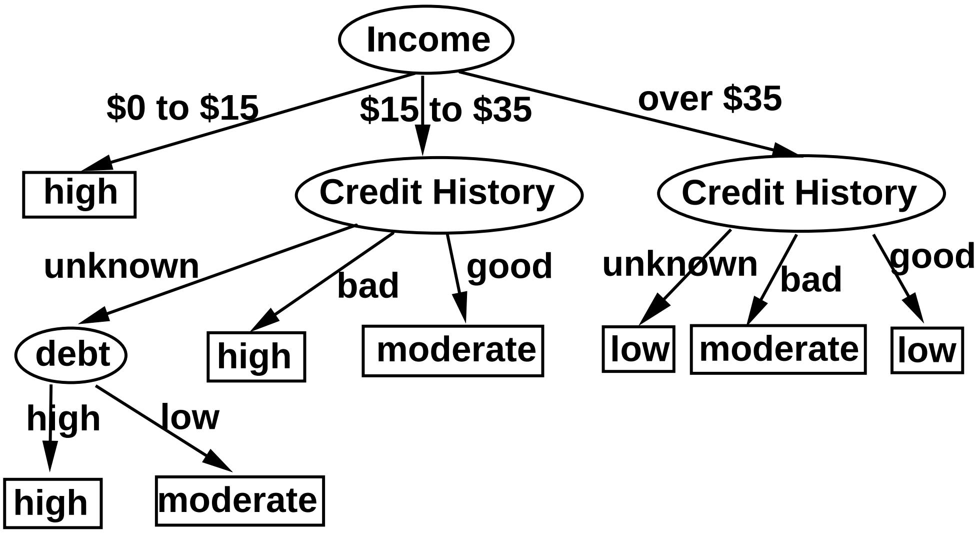


Figure 16.1: Decision tree generated by ID3 from the credit risk dataset.

2. Suppose that a sports club stores data on various characteristics of its members. The club wants to carry out the tasks listed below using the following algorithms

of Weka: ID3, SimpleKMeans and APriori. APriori is an algorithm for learning association rules. For each task, write down which of the above algorithms you would advise the club to use.

(a) Clustering members of the club into groups.

(b) Finding characteristics of members that tend to occur together.

(c) Predicting the favourite sport of a new member.

3. Discuss ethical issues which can arise in practical applications of Data Mining.

Chapter 17

Case Study

17.1 Aim of the Case-Study

The aim of this case-study is that you will gain experience of mining a real-world dataset which is larger than the 'toy' datasets that we used when you were learning about the algorithms. This will involve the following steps.

1. Getting to know the data.
2. Preparing the data.
3. Applying data mining algorithms to the data.
4. Analysing the results.

You should complete this chapter in your own time. Recall that the official documentation for this module states that you should spend more than 135 hours on independent study. So, you should do this case-study as part of your independent study.

17.2 Background and Motivation

Trade unions are organisations that represent people at work. Their purpose is to protect and improve people's pay and conditions of employment. They also campaign for laws and policies which will benefit working people. Trade unions exist because an individual worker has very little power to influence decisions that are made about his or her job. By joining together with other workers, there is more chance of having a voice and influence. For more information, see:
http://www.tuc.org.uk/about-tuc/mainly-students/what-tuc. If you are curious to learn more about trade unions then you may wish to visit the People's History Museum which is not far from the university. It is a two minute walk from Salford Central railway station. Admission is free to all. For more information, see:

http://www.phm.org.uk/. (Please note that the content of the websites mentioned in this section are not part of the specification of the case-study.)

The Trade Union Congress (TUC) is an umbrella organisation of Britain's trade unions. Let's suppose that the TUC wants to find interesting patterns in a large amount of data on labour negotiations. It is looking to fill a graduate position with a person who can do this. Imagine that the TUC is assessing whether applicants are suitable by asking them to complete the following related task.

17.3 Equipment and Facilities to be Used

Throughout this case-study use:

- the Explorer interface to Weka;
- predictive accuracy to measure performance.

17.4 The Task

17.4.1 Origin of the Dataset

The data used in this case-study originally comes from the Industrial Relations Information Service in Ottawa, Canada. The data includes agreements reached between employers and trade unions in the business and personal services sector in Canada during 1987 and the first quarter of 1988. These sectors include teachers, nurses, university staff, police, etc.

17.4.2 The Learning Task

The dataset contains 57 examples. Each example represents a contract of employment using 16 attribute-value pairs. Contracts are classified as either acceptable (good) or unacceptable (bad). The acceptable contracts are ones which were accepted by both trade unions and management. The learning task is to predict whether or not a contract is acceptable.

17.4.3 Downloading and Formatting the Dataset

The dataset for this case-study was taken from the UCI machine learning repository (Kelly et al., 2023), see:

http://archive.ics.uci.edu/ml/datasets/Labor+Relations

The files are available locally. Please download the files called labor-neg.data, labor-neg.names and labor-neg.test from Blackboard. To make absolutely sure that you are using the same dataset as Dr Bryant, you should download the files from one of these local sources rather than the UCI repository: the repository could be updated at any time.

Recall that one of your the first tasks as a Data Miner is to get to know your data. Translate the dataset into ARFF format. The examples in the file called labor-neg.test should only be used for testing. Nominal types should be declared for the following attributes.

- Cost of Living Adjustment
- Pension
- Education Allowance
- Vacation
- Long-term Disability Assistance
- Contribution to Dental Plan
- Bereavement Assistance
- Contribution to Health Plan
- Acceptable

All the other attributes should be of type real.

17.4.4 Rules

Using the default values for their parameters, apply ZeroR and OneR to the data.

ZeroR The algorithm ZeroR does not generate a concept description. Instead (for problems where class attribute is nominal) it always predicts that the class will be the majority class. Explain why ZeroR achieves the predictive accuracy reported by Weka. You explanation should include the working of the mathematical calculation.

OneR Translate the concept description generated by OneR into sentences of plain English which can be understood by someone with no knowledge of computing. Do not include symbols for mathematic operators in your translation. Your translation should be concise, as opposed to verbose.

PRISM Give two reasons why PRISM cannot be applied directly to the data.

17.4.5 k-Nearest Neighbour

Apply k-Nearest Neighbour to the data.

- Use the default values for all of its parameters except the k parameter.
- Investigate the effect of varying the k parameter by setting it to the following values in turn: 1, 2, 3, 4, 5 and 6. For each value, apply k-Nearest Neighbour to the data and note the predictive accuracy.
- Compare the results when $k = 1$ with the results when $k > 1$ and write down what this suggests about the data.

17.4.6 J48

Apply J48 to the data.

- Use the default values for all of its parameters except confidenceFactor.
- The confidenceFactor parameter of J48 controls the level of pruning. Investigate the effect of varying this parameter by setting it to the following values in turn: 0.1, 0.25, 0.5, 0.75 and 1.0. For each value, apply J48 to the data and note the predictive accuracy and the number of nodes in the decision tree.
- Suggest an explanation for the effect of varying the parameter on the predictive accuracy.

17.5 Documentation Required

Algorithm	Predictive Accuracy (%)	Number of Errors	
		Omission	Commission
ZeroR OneR k-NN (when k =3) J48 (when confidenceFactor = 0.25)			

Table 17.1: Case-study: Summary of Results

Prepare a document which includes the following items.

1. A listing of both the training data and test data in ARFF format (see Section 17.4.3).
2. A completed version of the Table 17.1.

3. The English translation of the concept description generated by OneR and your analysis of the output of the rule-learning algorithms (see Section 17.4.4).

4. A completed version of Tables 17.2 and 17.3 and your analysis of them (see Sections 17.4.5 and 17.4.6).

k	Predictive Accuracy of k-NN (%)
1	
2	
3	
4	
5	
6	

Table 17.2: Case-study: k-NN Results

confidenceFactor	Predictive Accuracy of J48 (%)	No of Nodes
0.1		
0.25		
0.5		
0.75		
1.0		

Table 17.3: Case-study: J48 Results

Part III

Artificial Intelligence

Chapter 18

Tutorial: What is Artificial Intelligence?

18.1 Aim of the Exercise

The aim of this tutorial is to find out what Artificial Intelligence is.

18.2 Exercise

1. How could introspection be inaccurate? (Introspection is the examination of one's own thoughts, impressions and feelings, especially over long periods.)
2. Why would evolution tend to result in systems that act rationally? What goals are such systems designed to achieve?
3. Consider the following two statements.
   - "Surely computers cannot be intelligent."
   - "Computers can do only what their programmers tell them."

   Is the latter statement true? Does the latter statement imply the former statement?
4. Consider the following two statements.
   - "Surely animals cannot be intelligent."
   - "Animals can do only what their genes tell them."

   Is the latter statement true? Does the latter statement imply the former statement?
5. Consider the following two statements.
   - "Surely animals, humans, and computers cannot be intelligent."
   - "They can do only what their constituent atoms are told to do by the laws of physics."

Is the latter statement true? Does the latter statement imply the former statement?

6. To what extent are the following computer systems instances of artificial intelligence.

   (a) Supermarket bar code scanners.

   (b) Web search engines.

   (c) Voice-activated telephone **menus**.

   (d) Internet routing algorithms that respond dynamically to the state of the network.

7. Is AI a science or engineering or neither or both? Explain your answer.

   (This question is intended to be about the essential nature of the AI problem and what is required to solve it. Do not interpret it as a sociological question about the current practice of AI researchers.)

Chapter 19

Seminar: State of the Art

19.1 Aim of the Seminar

The aim of this seminar is to find out what is the state of the art in AI.

19.2 Preparation

You should complete this preparation in your own time before you arrive at the class. Recall that the official documentation for this module states that you should spend more than 135 hours on independent study. You should work in groups of, at least, five students and divide the work between the members of the group.

Gather information which is relevant to the discussion proposed in Section 19.3. As a student of the University of Salford, you have access to a wide range of library resources, including books, journals, and databases.

19.3 Discussion

Working in groups, discuss whether the tasks listed below can be currently solved by computers. For the currently infeasible tasks, try to find out what the difficulties are and predict when, if ever, they will be overcome. As time is limited, each group should only discuss a few of the tasks.

- Manufacture a car.
- Drive safely along a motorway.
- Drive safely in Manchester city centre.
- Buy a week's worth of groceries on the web.

- Buy a week's worth of groceries at the supermarket.
- Play a decent game of table tennis.
- Play a decent game of bridge at a competitive level.
- Discover and prove a new mathematical theorem.
- Design and execute a research program in molecular biology.
- Write an intentionally funny story.
- Give competent legal advice in a specialised area of law.
- Translate spoken English into spoken Swedish in real time.
- Converse successfully with another person for an hour.
- Perform a complex surgical operation.
- Unload any dishwasher and put everything away.

19.4 Report

Your group must prepare a single report which includes the following.

1. The group name.
2. The date.
3. The two most and the two least useful sources of information. Justify for your choices. Include citations.
4. A brief summary of your discussion and your conclusions.
5. A list of references.

The report must be sent to Dr Bryant (via email) by 4pm on Friday 24^{th} January 2024.

- Citations and references must conform to the APA 7^{th} (Harvard) style.
- The length of the report must not exceed one side of A4.
- The report should be typed (rather than hand-written) in, at least, font size 12pt.
- The report must be in PDF format; do **not** submit a zip file.
- The name of the PDF file must include the group name.

Please note that, in due course, Dr Bryant will upload all of the reports to the AI&DM module on Blackboard.

Chapter 20

Tutorial: Introduction to Propositional Logic

20.1 Aim of Exercise

The aim of the exercise is for you to gain experience of the following for propositional logic:

- Identifying propositions and connectives.
- Using the precedence order of connectives.
- Classifying a proposition as a tautology, contradiction or contingency.

20.2 Exercise

1. Which of the following English sentences express a proposition?
   (a) I think therefore I am.
   (b) Do as I say, not as I do!
   (c) Whenever the assignment x = y is executed, the value of y remains unaltered.
   (d) Write clearly and legibly.
   (e) How do you know your answers are correct?
2. List the atomic propositions and connectives which appear in the following propositions and write down a well-formed formula for each one.
   (a) If it rains then I am going to get wet.
   (b) Increased spending overheats the economy.

(c) Increased spending coupled with tax cuts overheats the economy.

(d) Overheating economy is a synonym for rise in excess demand.

(e) Inflation either rises or does not.

3. Remove as many brackets as possible from the following propositions without altering their meaning (i.e. the truth table).

   (a) $((q \Leftrightarrow ((\neg r) \vee (s \wedge p))) \Leftrightarrow (q \Rightarrow p))$

   (b) $(((p \wedge (\neg q)) \wedge r) \vee s)$

   (c) $((p \Rightarrow (q \vee r)) \wedge (\neg (r \Rightarrow s)))$

   (d) $((\neg (\neg (\neg (q \vee r)))) \Leftrightarrow (q \Leftrightarrow r))$

   (e) $(p \vee (q \vee r))$

4. Decide using truth tables whether each of the following is a tautology, contradiction or contingency.

   (a) $p \Rightarrow \neg p$

   (b) $p \wedge q \Rightarrow p$

   (c) $(p \Rightarrow \neg p) \wedge (\neg p \Rightarrow p)$

Chapter 21

Tutorial: Transformational Proofs

21.1 Aim of Exercise

The aim of the exercise is for you to gain experience of the following for propositional logic:

- Appreciating its origin.
- Proving the equivalence of formulae using truth tables.
- Using the laws of equivalence.
- Performing a transformational proof.

21.2 Exercise

1. Whose seminal work attempted to apply the formal laws of algebra and arithmetic to the principles of logic and used substitution of logically equivalent expressions as its primary inference method?
2. Using truth tables, demonstrate the following logical equivalences.

   (a) The law of negation, i.e., $\neg (\neg p) \equiv p$.

   (b) The contraposition law, i.e., $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$.

3. Provide the names of the laws used in each step of the proof below.

   $p \land (q \land r)$

   $\equiv p \land \neg\neg (q \land r)$ ....

   $\equiv p \land \neg (\neg q \lor \neg r)$ ....

   $\equiv \neg\neg (p \land (\neg (\neg q \lor \neg r)))$ ....

   $\equiv \neg (\neg p \lor \neg (\neg (\neg q \lor \neg r)))$ ....

   $\equiv \neg (\neg p \lor (\neg q \lor \neg r))$ ....

   $\equiv \neg ((\neg p \lor \neg q) \lor \neg r)$ ....

   $\equiv \neg (\neg (p \land q) \lor \neg r)$ ....

   $\equiv \neg\neg (p \land q) \land \neg\neg r$ ....

   $\equiv (p \land q) \land r$ ....

4. Show that the following pair of sentences are logically equivalent by performing a transformational proof.

   - If it rains then the crops grow.
   - If the crops don't grow then there has been no rain.

5. Show that the following pair of sentences are logically equivalent by performing a transformational proof. Use the symbols L, W and C to represent the atomic propositions.

   - If you are not lazy then you work hard, or you are clever and lazy.
   - You work hard or you are lazy.

6. Using truth tables, demonstrate the following logical equivalence.

   $p \Leftrightarrow (q \Leftrightarrow r) \equiv (p \Leftrightarrow q) \Leftrightarrow r$

Chapter 22

Tutorial: Validity and Inference Rules

22.1 Aim of Exercise

The aim of the exercise is for you to gain experience of the following for propositional logic:

- Appreciating its origin.
- Establish the validity of an argument using truth tables.
- Demonstrate the invalidity of an argument.
- Understand inference rules.

22.2 Exercise

1. Define the word syllogism.
2. Who developed an informal system of syllogisms in ancient Greece?
3. Demonstrate the validity of the following inference rules using truth tables.

   Double Negation $\neg \, Elim$

   $$\frac{\neg\neg \text{p}}{\text{p}}$$

   Hypothetical syllogism This says that if p implies q and q implies r, then it can be logically concluded that p implies r.

   $$\frac{\begin{array}{l}\text{p} \Rightarrow \text{q}\\ \text{q} \Rightarrow \text{r}\end{array}}{\text{p} \Rightarrow \text{r}}$$

4. Suggest why the hypothetical syllogism is sometimes called ‘the chain rule’.

5. Demonstrate the invalidity of the following arguments.

   (a) $p \vee q$
   q
   Therefore, p

   (b) $p \Rightarrow q$
   $q \Rightarrow p$
   Therefore, $p \wedge q$

6. The following inference rule is known as ‘Constructive dilemma’.

$$\begin{array}{l} p \Rightarrow q \\ r \Rightarrow s \\ p \vee r \\ \hline q \vee s \end{array}$$

Explain what it means in English.

Chapter 23

Tutorial: Deductive Proofs

23.1 Aim of Exercise

The aim of the exercise is for you to gain experience of the following for propositional logic:

- Recalling and using inference rules.
- Performing a deductive proof.
- Proving the validity of an argument presented in English.

23.2 Exercise

1. Fill in the missing steps in the proof and add the names of the laws.

1 $A \Rightarrow \neg B$	premise	
2 $(B \vee C) \vee D$	premise	
3 $\neg C \vee D \Rightarrow A$	premise	
4 $\neg C$	premise	
5	....	....
6 A	....	....
7	....	....
8 $\neg B \wedge \neg C$	....	....
9	....	....
10 D	....	....

2. In a previous tutorial, we studied the Hypothetical syllogism which says that if p implies q and q implies r, then it can be logically concluded that p implies r.

$$\begin{array}{l} p \Rightarrow q \\ q \Rightarrow r \\ \hline p \Rightarrow r \end{array}$$

We demonstrated its validity using a truth table. Fill in the missing steps in the following proof and add the names of the laws.

1 $A \Rightarrow B$	premise
2 $B \Rightarrow C$	premise
3 ...	....
4 B	....
5 ...	....
6 $A \Rightarrow C$	....

3. Prove the following argument.

> If you give your order by telephone or fax then we will deal with it promptly and efficiently. Your goods will arrive on the next day if we deal with your order promptly or we use Zippo couriers. Therefore, if you give your order by telephone your goods will arrive on the next day.

Chapter 24

Tutorial: Introduction to Predicate Logic

24.1 Aim of Exercise

The aim of the exercise is for you to gain experience of the following for predicate logic:

- Appreciating its origin.
- Expressing simple arguments in predicate logic.
- Using unary and binary predicates.
- Using quantifiers to make propositions from formulae without replacing variables with specific individuals.
- Using recurrence relations, i.e., recursive definitions.

24.2 Exercise

1. Who extended Boole's logic to include objects and relations, creating predicate logic?
2. The extended family tree shown in Figure 24.1 can be represented in predicate logic by facts such as:

father(Philip, Charles)	father(Philip, Anne)	...
mother(Mum, Margaret)	mother(Mum, Elizabeth)	...
married(Dianna, Charles)	married(Elizabeth, Philip)	...
male(Philip)	male(Charles)	...
female(Beatrice)	female(Margaret)	...

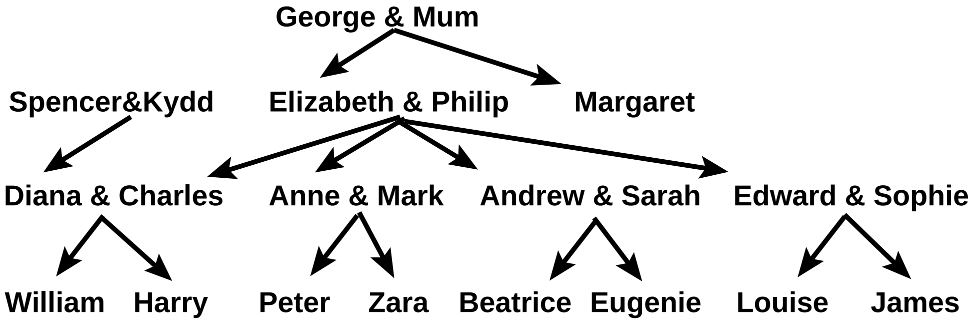


Figure 24.1: Extended Family Tree.

Write down axioms describing the following predicates. (Make sure you write definitions with ⇔. If you use ⇒, you are only imposing constraints, not writing a real definition.)

(a) parent
(b) daughter
(c) son
(d) sibling
(e) brother
(f) sister
(g) brotherinlaw
(h) sisterinlaw
(i) grandparent
(j) greatgrandparent
(k) ancestor

3. Predicate logic can represent axioms which include recurrence relations, i.e., recursive definitions.

   (a) Which characteristic of the definition of ancestor makes it recursive?
   (b) Explain why a recurrence relation is needed to define ancestor concisely.

4. Using appropriate unary predicates, express each of the following sentences in predicate logic.

   (a) All students are clever.
   (b) No one can be clever without being hardworking.
   (c) Clever students work hard.

(d) Those who do not work hard are lazy.

(e) Not being lazy is equivalent to being hardworking.

5. Express the following formulae in colloquial English. Sentences starting with phrases such as “For every x there is. . . ” are not colloquial because of the style and the use of the variable x, which can be avoided.

(a) $\exists x \bullet$ clever(x,John) $\wedge$ short(John) $\wedge$ respects(Jane,x)

(b) $\forall x \bullet$ $\forall y \bullet$ loves(x,y) $\wedge$ tall(y) $\Rightarrow$ respects(x,y)

(c) $\exists x \bullet$ $\forall y \bullet$ loves(x,y)

(d) $\neg$ $\forall x \bullet$ $\forall y \bullet$ loves(x,y)

(e) $\forall x \bullet$ $\forall y \bullet$ loves(x,y) $\Rightarrow$ loves(y,x)

(f) $\exists x \bullet$ loves(x,Jane) $\wedge$ $\neg$ respects(Jane,x)

(g) $\exists x \bullet$ $\forall y \bullet$ $\neg$ respects(y,x)

(h) $\exists x \bullet$ $\neg$ $\exists y \bullet$ loves(y,x) $\vee$ respects(y,x)

Chapter 25

Tutorial: Predicate Logic: A Closer Look at Quantifiers

25.1 Aim of Exercise

The aim of the exercise is for you to consolidate the knowledge you acquired during the previous tutorial and to gain experience of the following for predicate logic:

- Evaluating the truth of formulae;
- Using the equality symbol.

25.2 Exercise

1. Given

    - Just 4 individuals: Ahmed, Khan, Patel and Scott.
    - 3 properties represented by 3 unary predicates: male, tall and short.
    - Just the following propositions hold true:

      male(Ahmed)
      male(Patel)
      male(Scott)

      tall(Ahmed)
      tall(Patel)

      short(Khan)
      short(Scott)

    Evaluate the truth of the following formula.

    $$\forall x \bullet\ \text{tall}(x) \Leftrightarrow \text{male}(x) \land \neg\ \text{short}(x)$$

2. Using appropriate binary predicates, express each of the following sentences in predicate logic.

(a) Salford stores only supply stores outside of Salford.

(b) No store supplies itself.

(c) There are no stores in Eccles but there are some in Trafford.

(d) Stores do not supply stores that are supplied by stores which they supply.

(e) Stores which supply each other are always in the same place.

3. Consider a vocabulary with the following symbols.

   - occupation(p, o): a predicate meaning person p has occupation o.
   - customer(p1, p2): a predicate meaning person p1 is a customer of person p2.
   - boss(p1,p2): a predicate meaning person p1 is a boss of person p2.
   - doctor, surgeon, lawyer, actor: constants denoting occupations.
   - emily, joe: constants denoting people.

   Use these symbols to write the following in predicate logic.

   (a) Emily is either a surgeon or a lawyer.

   (b) Joe is an actor, but he also holds another job.

   (c) All surgeons are doctors.

   (d) Joe does not have a lawyer (i.e., is not a customer of any lawyer).

   (e) Emily has a boss who is a lawyer.

   (f) There exists a lawyer all of whose customers are doctors.

   (g) Every surgeon has a lawyer.

25.3 Challenge Exercise

The universal quantifier usually goes with an implication. But not always. Express each of the following in predicate logic.

1. Tall children are not clever if there are short adults.

2. Some adults are clever exactly when none of the children are clever.

Chapter 26

Tutorial: Predicate Logic: Deductive Proofs

26.1 Aim of Exercise

The aim of the exercise is for you to gain experience of the following for predicate logic:

- Recalling and using inference rules.
- Performing a deductive proof.
- Adhering to the constraints on handling variables and constants when eliminating and introducing quantifiers.
- Proving the validity of an argument presented in English.

26.2 Exercise

1. Consider the following argument

   > All masters and slaves are adults. However, there is at least one person who is neither a master nor a slave. Also, not all persons are young or not slaves. Therefore, there are some adults in this society.

   Annotate the proof listed on the next page.

1 $\forall x \bullet\ master(x) \lor slave(x) \Rightarrow adult(x)$	premise	
2 $\exists y \bullet\ \neg\ master(y) \land \neg\ slave(y)$	premise	
3 $\neg\ \forall z \bullet\ young(z) \lor \neg\ slave(z)$	premise	
4 $\exists z \bullet\ \neg\ (young(z) \lor \neg\ slave(z))$	....	....
5 $\neg\ (young(z) \lor \neg\ slave(z))$	....	....
6 $\neg\ young(z) \land \neg\ \neg\ slave(z)$	....	....
7 $\neg\ \neg\ slave(z)$	....	....
8 $slave(z)$	....	....
9 $slave(z) \lor master(z)$	....	....
10 $master(z) \lor slave(z)$	....	....
11 $master(z) \lor slave(z) \Rightarrow adult(z)$	....	....
12 $adult(z)$	....	....
13 $\exists z \bullet\ adult(z)$	....	....

2. Prove the following argument

    Some visitors attend all events. No visitors attend outdoor events.
    Therefore, there are no outdoor events.

3. Recall that formula p is said to logically imply (entail $\models$) formula q if and only if the implication p $\Rightarrow$ q is a tautology. Proofs of entailments do not involve any premises. Prove the following generalised version of Modus Ponens.

    $(\forall x \bullet\ p(x) \Rightarrow q(x)) \models ((\forall x \bullet\ p(x)) \Rightarrow (\forall x \bullet\ q(x)))$

    Every step of your proof should be numbered and fully annotated.

Chapter 27

Tutorial: Resolution and Propositional Logic

27.1 Aim of Exercise

The aim of the exercise is for you to gain experience of the following for propositional logic:

- Transforming formulae into conjunctive normal form.
- Applying resolution.
- Proving the validity of an argument presented in English.

27.2 Exercise

1. Use the resolution method to prove the following argument.

   > If you give your order by telephone or fax then we will deal with it promptly and efficiently. Your goods will arrive on the next day only if we deal with your order promptly or we use Zippo couriers. Therefore, if you give your order by telephone your goods will arrive on the next day.

   Your answer should include the following:

   (a) A formalisation of the argument in propositional logic. (Do not be confused if you encounter a sense of déjà-vu: you should already have done this when you did Chapter 23. You can use the same formalisation here.)

   (b) A transformation of each premise into conjunctive normal form.

   (c) A transformation of the negated goal into conjunctive normal form.

(d) A proof in which the only inference rule used is resolution.

2. Compare your solution with the proof you wrote for the same argument when you did the exercise in Chapter 23. Consider the advantages and disadvantages of each approach with regard to mechanisation. In other words, ponder how easy it would be to implement them on a computer. Write down an explanation of why resolution is easy to mechanise.

Chapter 28

Tutorial: Resolution and Predicate Logic

28.1 Aim of Exercise

The aim of the exercise is for you to gain experience of the following for predicate logic:

- Transforming formulae into conjunctive normal form.
- Applying resolution.
- Proving the validity of an argument presented in English.

28.2 Exercise

Use the resolution method to prove the following argument. From

> “Sheep are animals”

it follows that

> “The head of a sheep is the head of an animal.”

Your answer should include the following:

1. A formalisation of the argument in predicate logic. Use the following three predicates only: sheep(x), animal(x) and head(z,x), meaning z is the head of x.
2. A transformation of the premise into conjunctive normal form.
3. A transformation of the negated conclusion into conjunctive normal form.
4. A proof in which the only inference rule used is resolution and unifications are shown using the notation $\theta = \{\text{x/y}\}$.

Part IV

Appendices and References

Appendix A

Values for Logarithmic Function

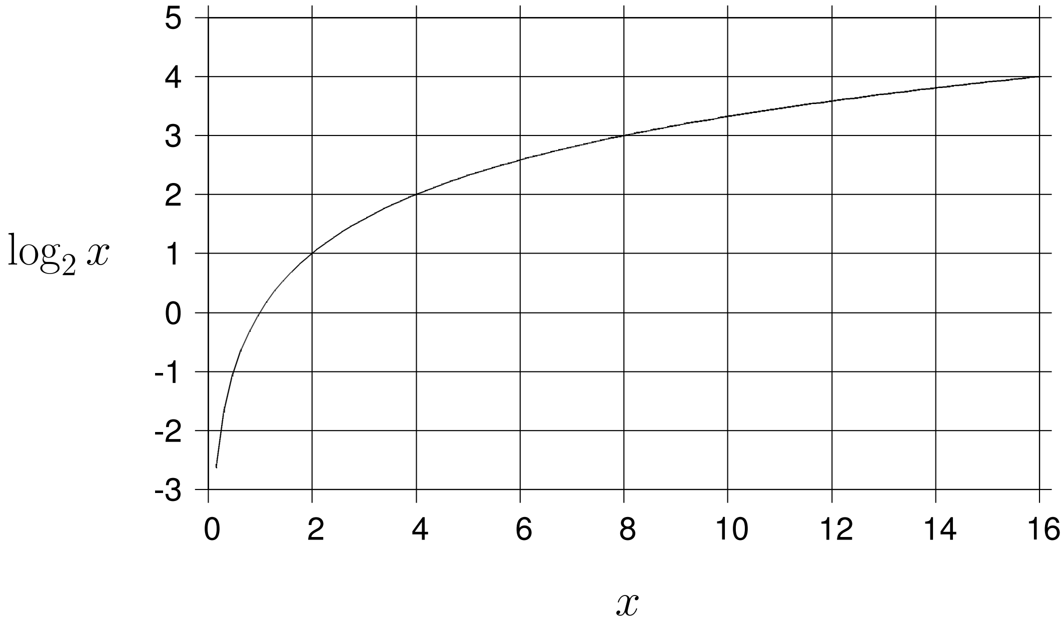


Figure A.1: Plot of $\log_2(x)$ versus x, for the range 0 to 16.

When $\log_2(x)$ is used to generate decision trees, x will be a probability. As probabilities are between zero and one, the graph shown in Figure A.2 focuses on this range. Table A.1 lists the coordinates of points on the plot shown in Figure A.2.

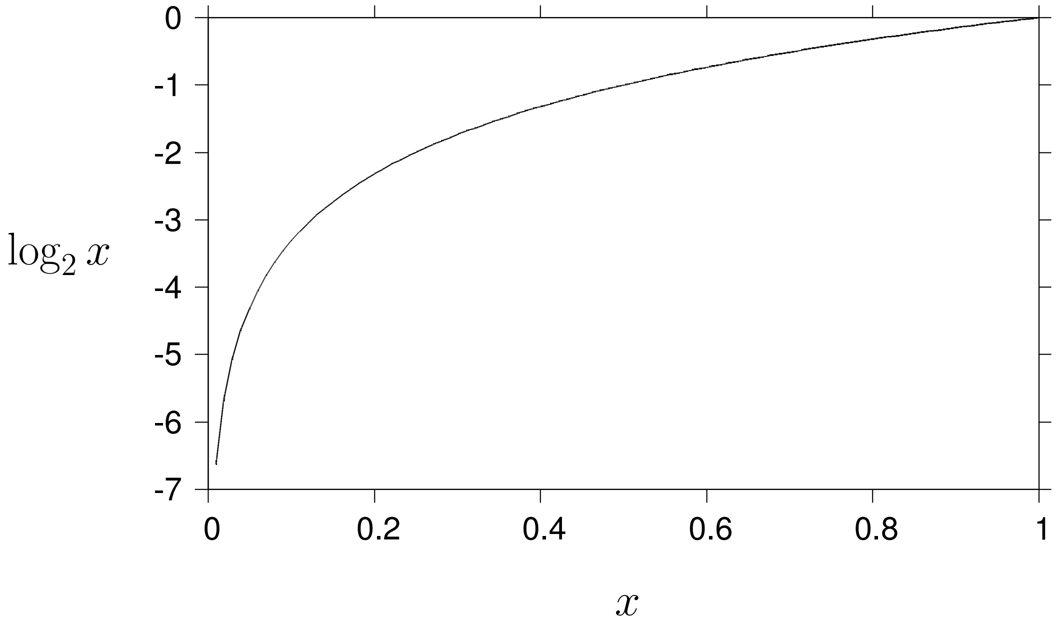


Figure A.2: Plot of $\log_2(x)$ versus x, for the range zero to one.

x	$\log_2 x$	x	$\log_2 x$	x	$\log_2 x$	x	$\log_2 x$	x	$\log_2 x$
0.01	-6.644	0.21	-2.252	0.41	-1.286	0.61	-0.713	0.81	-0.304
0.02	-5.644	0.22	-2.184	0.42	-1.252	0.62	-0.690	0.82	-0.286
0.03	-5.059	0.23	-2.120	0.43	-1.218	0.63	-0.667	0.83	-0.269
0.04	-4.644	0.24	-2.059	0.44	-1.184	0.64	-0.644	0.84	-0.252
0.05	-4.322	0.25	-2.000	0.45	-1.152	0.65	-0.621	0.85	-0.234
0.06	-4.059	0.26	-1.943	0.46	-1.120	0.66	-0.599	0.86	-0.218
0.07	-3.837	0.27	-1.889	0.47	-1.089	0.67	-0.578	0.87	-0.201
0.08	-3.644	0.28	-1.837	0.48	-1.059	0.68	-0.556	0.88	-0.184
0.09	-3.474	0.29	-1.786	0.49	-1.029	0.69	-0.535	0.89	-0.168
0.10	-3.322	0.30	-1.737	0.50	-1.000	0.70	-0.515	0.90	-0.152
0.11	-3.184	0.31	-1.690	0.51	-0.971	0.71	-0.494	0.91	-0.136
0.12	-3.059	0.32	-1.644	0.52	-0.943	0.72	-0.474	0.92	-0.120
0.13	-2.943	0.33	-1.599	0.53	-0.916	0.73	-0.454	0.93	-0.105
0.14	-2.837	0.34	-1.556	0.54	-0.889	0.74	-0.434	0.94	-0.089
0.15	-2.737	0.35	-1.515	0.55	-0.862	0.75	-0.415	0.95	-0.074
0.16	-2.644	0.36	-1.474	0.56	-0.837	0.76	-0.396	0.96	-0.059
0.17	-2.556	0.37	-1.434	0.57	-0.811	0.77	-0.377	0.97	-0.044
0.18	-2.474	0.38	-1.396	0.58	-0.786	0.78	-0.358	0.98	-0.029
0.19	-2.396	0.39	-1.358	0.59	-0.761	0.79	-0.340	0.99	-0.014
0.20	-2.322	0.40	-1.322	0.60	-0.737	0.80	-0.322	1.00	0.000

Table A.1: Values of $\log_2(x)$ for the range zero to one.

Appendix B

Properties of Entropy

B.1 Entropy for Boolean Classes

If there are c values of the class attribute, then entropy can be a large as $c \times -\frac{1}{c}\log_2\frac{1}{c}$ $= -\log_2\frac{1}{c} = \log_2 c$. For Boolean classes, $c = 2$, so entropy can be as large as $\log_2 2 = 1$.

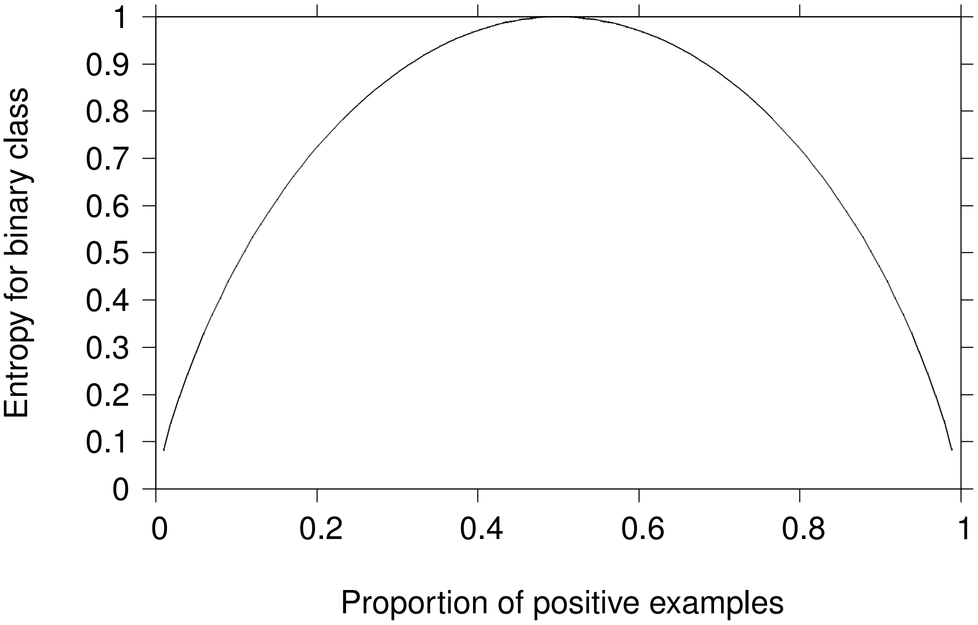


Figure B.1: Plot of entropy for a binary class versus proportion of positive examples.

B.2 The Multistage Property of Entropy

The multistage property expressed generally is

$$I(p,q,r) \quad = \quad I(p,q+r) + (q+r) \times I\left(\frac{q}{q+r}, \frac{r}{q+r}\right)$$

where $p+q+r=1$. Suppose that $p=\frac{2}{9}$, $q=\frac{3}{9}$ and $r=\frac{4}{9}$.

$$\begin{aligned}
I(\frac{2}{9},\frac{3}{9},\frac{4}{9}) &= I\left(\frac{2}{9},\frac{3}{9}+\frac{4}{9}\right) + \left(\frac{3}{9}+\frac{4}{9}\right) \times I\left(\frac{\frac{3}{9}}{\frac{3}{9}+\frac{4}{9}}, \frac{\frac{4}{9}}{\frac{3}{9}+\frac{4}{9}}\right) \\
&= I\left(\frac{2}{9},\frac{7}{9}\right) + \frac{7}{9} \times I\left(\frac{\frac{3}{9}}{\frac{7}{9}}, \frac{\frac{4}{9}}{\frac{7}{9}}\right) \\
&= I\left(\frac{2}{9},\frac{7}{9}\right) + \frac{7}{9} \times I\left(\frac{3}{7}, \frac{4}{7}\right) \\
I([2,3,4]) &= I([2,7]) + \frac{7}{9} \times I([3,4])
\end{aligned}$$

Appendix C

Manipulating Logarithmic Formulae

This appendix explains how the computation of individual fractions for each class can be avoided during the computation of entropy by rearranging the equation. E.g.,

$$\begin{aligned} I([2,3,4]) &= -\frac{2}{9} \times log_2\frac{2}{9} - \frac{3}{9} \times log_2\frac{3}{9} - \frac{4}{9} \times log_2\frac{4}{9} \\ &= \frac{-2 \times log_2 2 - 3 \times log_2 3 - 4 \times log_2 4 + 9 log_2 9}{9} \end{aligned}$$

C.1 Separating a Numerator and Denominator

If the argument to a logarithmic function is a fraction then we can separate the numerator and denominator.

$$\log_2 \frac{x}{y} = \log_2 x - \log_2 y \tag{C.1}$$

Equation C.1 holds for any particular values of x and y.

E.g., suppose $x = 8$ and $y = 2$.

$$\log_2 \frac{8}{2} = \log_2 4 = 2 = 3 - 1 = \log_2 8 - \log_2 2$$

E.g., suppose $x = 64$ and $y = 8$.

$$\log_2 \frac{64}{8} = \log_2 8 = 3 = 6 - 3 = \log_2 64 - \log_2 8$$

C.2 Factorising using a Common Denominator

This section extends the idea of Equation C.1, i.e., separating the numerator and denominator of a fraction appearing in the argument to a logarithmic function. If the terms of a mathematical expression are logarithmic functions whose arguments are factions with a common denominator, z, then this expression can be rearranged as follows:

$$\log_2 \frac{x}{z} + \log_2 \frac{y}{z} = \log_2 x + \log_2 y - 2 \times \log_2 z \tag{C.2}$$

Equation C.2 holds for any particular values of x, y and z.

E.g., suppose $x = 8$, $y = 64$ and $z = 2$.

$$\log_2 \frac{8}{2} + \log_2 \frac{64}{2} = \log_2 4 + \log_2 32 = 2 + 5 = 7$$

$$\log_2 8 + \log_2 64 - 2 \times \log_2 2 = 3 + 6 - 2 \times 1 = 7$$

Consider an example with three fractions on the left-hand side, rather than two.

$$\log_2 \frac{8}{2} + \log_2 \frac{64}{2} + \log_2 \frac{32}{2} = \log_2 4 + \log_2 32 + \log_2 16 = 2 + 5 + 4 = 11$$

$$\log_2 8 + \log_2 64 + \log_2 32 - 3 \times \log_2 2 = 3 + 6 + 5 - 3 \times 1 = 11$$

Consider a similar example where each such term is being multiplied by a negative fraction.

$$-\frac{2}{9} \times \log_2 \frac{2}{9} - \frac{3}{9} \times \log_2 \frac{3}{9} - \frac{4}{9} \times \log_2 \frac{4}{9}$$

Notice that this is the example which appears at the beginning of this appendix. Also, note that each term is being divided by nine, so $\frac{1}{9}$ is a common factor. So we can apply the distributive law of algebra to give:

$$= \frac{-2 \times \log_2 \frac{2}{9} - 3 \times \log_2 \frac{3}{9} - 4 \times \log_2 \frac{4}{9}}{9}$$

Finally, applying the idea from Equation C.2 gives:

$$= \frac{-2 \times \log_2 2 - 3 \times \log_2 3 - 4 \times \log_2 4 + 9 \log_2 9}{9}$$

Appendix D

Laws of Equivalence

Negation law	$\neg (\neg p) \equiv p$
Law of equivalence	$p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$
Law of implication	$p \Rightarrow q \equiv \neg p \vee q$
Contraposition Law	$p \Rightarrow q \equiv \neg q \Rightarrow \neg p$
Law of excluded middle	$p \vee \neg p \equiv \text{true}$
Law of contradiction	$p \wedge \neg p \equiv \text{false}.$
Law of simplification	$p \wedge \text{true} \equiv p$
Law of simplification	$p \vee \text{true} \equiv \text{true}$
Law of simplification	$p \wedge \text{false} \equiv \text{false}$
Law of simplification	$p \vee \text{false} \equiv p$
Law of simplification	$p \vee (p \wedge q) \equiv p$
Law of simplification	$p \wedge (p \vee q) \equiv p$
Idempotence Law	$p \vee p \equiv p$
Idempotence Law	$p \wedge p \equiv p$

Commutative Law	$p \wedge q \equiv q \wedge p$
Commutative Law	$p \vee q \equiv q \vee p$
Commutative Law	$p \Leftrightarrow q \equiv q \Leftrightarrow p$
Associative Law	$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$
Associative Law	$p \vee (q \vee r) \equiv (p \vee q) \vee r$
Associative Law	$p \Leftrightarrow (q \Leftrightarrow r) \equiv (p \Leftrightarrow q) \Leftrightarrow r$
Distributive Law	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
Distributive Law	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
De Morgan’s Law	$\neg (p \wedge q) \equiv \neg p \vee \neg q$
De Morgan’s Law	$\neg (p \vee q) \equiv \neg p \wedge \neg q$

Predicate logic has the following four extra cases of De Morgan’s law.

$$\neg \exists x \bullet p(x) \quad \equiv \quad \forall x \bullet \neg p(x)$$

$$\neg \exists x \bullet \neg p(x) \quad \equiv \quad \forall x \bullet p(x)$$

$$\neg \forall x \bullet p(x) \quad \equiv \quad \exists x \bullet \neg p(x)$$

$$\neg \forall x \bullet \neg p(x) \quad \equiv \quad \exists x \bullet p(x)$$

Appendix E

Inference Rules

Conjunction $\wedge \, Intro$

$$\begin{array}{l} \text{p} \\ \text{q} \\ \hline \text{p} \wedge \text{q} \end{array}$$

Simplification $\wedge \, Elim$

$$\begin{array}{l} \text{p} \wedge \text{q} \\ \hline \text{p} \end{array}$$

Addition $\vee \, Intro$

$$\begin{array}{l} \text{p} \\ \hline \text{p} \vee \text{q} \end{array}$$

Disjunctive syllogism $\vee \, Elim$

$$\begin{array}{l} \text{p} \vee \text{q} \\ \neg \, \text{p} \\ \hline \text{q} \end{array}$$

Modus Ponens $\Rightarrow Elim$

$$\begin{array}{l} \text{p} \Rightarrow \text{q} \\ \text{p} \\ \hline \text{q} \end{array}$$

Modus Tollens $\Rightarrow Elim$

$$\begin{array}{l} \text{p} \Rightarrow \text{q} \\ \neg \, \text{q} \\ \hline \neg \, \text{p} \end{array}$$

Double Negation $\neg \, Elim$

$$\begin{array}{l} \neg \, \neg \, \text{p} \\ \hline \text{p} \end{array}$$

Law of equivalence $\Leftrightarrow Elim$ $$\begin{array}{l} p \Leftrightarrow q \\ \hline p \Rightarrow q \end{array}$$

Law of equivalence $\Leftrightarrow Elim$ $$\begin{array}{l} p \Leftrightarrow q \\ \hline q \Rightarrow p \end{array}$$

Transitivity of equivalence $$\begin{array}{l} p \Leftrightarrow q \\ q \Leftrightarrow r \\ \hline p \Leftrightarrow r \end{array}$$

Hypothetical syllogism (or the chain rule) $$\begin{array}{l} p \Rightarrow q \\ q \Rightarrow r \\ \hline p \Rightarrow r \end{array}$$

Constructive dilemma $$\begin{array}{l} p \Rightarrow q \\ r \Rightarrow s \\ p \vee r \\ \hline q \vee s \end{array}$$

Deductive Theorem $\Rightarrow Intro$ $$\begin{array}{l} p, \ldots, r, \boxed{\mathbf{s}} \vdash t \\ \hline p, \ldots, r \vdash s \Rightarrow t \end{array}$$

where the highlighted $\boxed{\mathbf{s}}$ is a formula which is an **assumption**.

Reductio ad absurdum $\neg\, Intro$ $$\begin{array}{l} p, \ldots, q, \boxed{\mathbf{r}} \vdash s \\ p, \ldots, q, \boxed{\mathbf{r}} \vdash \neg\, s \\ \hline p, \ldots, q \vdash \neg\, r \end{array}$$

where the highlighted $\boxed{\mathbf{r}}$ is a formula which is an **assumption**.

Resolution Inference Rule $$\begin{array}{l} X \vee A \\ Y \vee \neg\, A \\ \hline X \vee Y \end{array}$$

Bibliography

Aha, D. (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37–66.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53, 370–418.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth.

Cendrowska, J. (1987). Prism: an algorithm for inducing module rules. *Journal of Man-Machine Studies*, 27(4), 349–370.

Croft, A. & Davidson, R. (2020). *Foundation Maths*. Number ISBN: 9781292289687. Harlow: Pearson, seventh edition.

Duda, R. & Hart, P. (1973). *Pattern Classification and Scene Analysis.* New York: John Wiley.

Han, J., Pei, J., & Tong, H. (2022). *Data Mining: Concepts and Techniques*. Number ISBN: 9780128117606. Morgan Kaufmann, fourth edition.

Hartigan, J. (1975). *Clustering algorithms*. New York: John Wiley.

Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11, 63–91.

Johns, M. (1961). An empirical bayes approach to nonparametric two-way classification. In H. Solomon (Ed.), *Studies in item analysis and prediction* Palo Alto, CA: Standford University Press.

Kelly, M., Longjohn, R., & Nottingham, K. (2023). The UCI machine learning repository. https://archive.ics.uci.edu.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. Mellish (Ed.), *Proceedings of 14th international joint conference on artificial intelligence* (pp. 1137–1143). San Francisco, CA.: Morgan Kaufmann.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106.

Quinlan, R. (1993). *C4.5: Programs for machine learning*. San Francisco: Morgan Kaufmann.

Russell, S. & Norvig, P. (2022). *Artificial Intelligence: A Modern Approach: Global Edition*. Number ISBN: 9781292401133. Pearson, fourth edition.

Wikipedia (2017). Claude Shannon. Wikipedia, The Free Encyclopedia.

Witten, I., Frank, E., Hall, M., & Pal, C. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Number ISBN: 978-0-12-804291-5. Morgan Kaufmann, fourth edition.