Department of Computer Science

The University of Waikato
Te Whare Wānanga o Waikato

Hamilton, NewZealand

Prediction Intervals for Class Probabilities

Xiaofeng Yu

This thesis is submitted in partial fulfilment of the requirements for the degree of Master of Science at The University of Waikato.

January 2007

Abstract

Prediction intervals for class probabilities are of interest in machine learning because they can quantify the uncertainty about the class probability estimate for a test instance. The idea is that all likely class probability values of the test instance are included, with a pre-specified confidence level, in the calculated prediction interval. This thesis proposes a probabilistic model for calculating such prediction intervals. Given the unobservability of class probabilities, a Bayesian approach is employed to derive a complete distribution of the class probability of a test instance based on a set of class observations of training instances in the neighbourhood of the test instance. A random decision tree ensemble learning algorithm is also proposed, whose prediction output constitutes the neighbourhood that is used by the Bayesian model to produce a PI for the test instance. The Bayesian model, which is used in conjunction with the ensemble learning algorithm and the standard nearest-neighbour classifier, is evaluated on artificial datasets and modified real datasets.

Acknowledgments

First of all, I am deeply indebted to my supervisor Dr. Eibe Frank, for the time he has spent on the thesis, for his invaluable knowledge, great patience, and consistent encouragement throughout my study. Without him, there wouldn't be this thesis! He taught me how to approach a research problem in different ways and find a breakthrough. He showed me by setting an example the need to be persistent and have a never-give-up attitude in order to achieve a success. He was always there to listen and give advice. He would repeat explaining the same question until I finally understood, and I will always be grateful for that.

There are many people in the Machine Learning group, who have been very supportive of my study. I first thank Ashraf Kibriya for helping me with configuring Weka Experimenter. I would like to thank Dr. Bernhard Pfahringer for helping me interpret the Fortune code. I would also like to thank Peter Reutemann, Richard Kirkby, Gabi Schmidberger, Stefan Mutter, Haijian Shi, Grant Anderson, and Lin Dong, and many others in the group, for all the help they have given to me.

I thank Dr. Bill Bolstad from Statistics Department for the copy of WinBUGS manual, and my friend Quan Qiu from the Digital Library group for the LaTeX editor.

Finally, I thank my wife, Shaoqun, who has always been supportive during my study by taking over all the household chores. It'll soon be over :-)

Contents

Abstract **i**

Acknowledgments **iii**

1 Introduction **1**

1.1 A Medical Diagnosis Example 1

1.2 Prediction Intervals 3

1.3 Aims of the Study 4

1.4 Thesis Structure 5

2 Background **7**

2.1 Bayesian Data Analysis 7

2.1.1 The Bayesian Framework 7

2.1.2 Model Specification 9

2.1.3 Numerical Computation 12

2.2 Statistical Intervals 14

2.2.1 Point Versus Interval Estimation 14

2.2.2 Types of Statistical Intervals 16

2.2.3 Why Do We Need PIs 17

2.2.4 Assumptions in Interval Calculation 18

2.2.5 Distribution-free Statistical Intervals 21

2.3 Classification Models 21

2.3.1 Regression Model 22

2.3.2 Nearest-Neighbour Classification 25

2.3.3 Ensemble of Classifiers 26

2.4 Summary 27

3 Literature Review **29**

3.1 Introduction 29

3.2 PI Computation Using Theoretical Formulae 29

3.3 Empirically based PIs 32

3.4 PI Methods by Simulation and Resampling . . . 33
3.5 Distribution-free PIs . . . 34
3.6 PIs based on the Bayesian Approach . . . 36
3.7 Summary . . . 37

4 Methodology **39**
4.1 Introduction . . . 39
4.2 General Setup . . . 39
4.3 Parametric Modelling Framework . . . 41
4.4 Modelling Class Probabilities . . . 42
4.5 PI Computation by Model Simulation . . . 46
4.6 A Random Tree Ensemble Classifier . . . 53
4.7 Summary . . . 56

5 Evaluation **57**
5.1 Classifier Evaluation . . . 57
5.2 Evaluation of the Bayesian PI Model . . . 61
5.2.1 Experimental Setup for the Bayesian PI Model . . . 62
5.2.2 Testing with Beta Random Numbers . . . 63
5.2.3 Testing with Artificial Datasets . . . 69
5.2.4 Testing with Semi-Real Datasets . . . 74
5.3 Summary . . . 80

6 Conclusions and Future Work **81**
6.1 Contributions . . . 81
6.2 Future Work . . . 83

Appendix A **87**

List of Figures

1.1 Illustration showing a prediction interval (PI). 4

2.1 Comparison of widths of statistical intervals for the same example. 17

3.1 The new observation x_{n+1} could occupy any of the $n+1$ positions formed by the n ordered observations. 35

4.1 Examples of beta densities. 44

4.2 (a) p($P_L \leq P_{n+1} \leq P_U$) = the area under the density curve between P_L and P_U. (b) Smooth curve area is simulated by summation of a large number of rectangles. 49

4.3 Different priors come with different sampling spaces in relation to the computation of the posterior distribution. (a) The straight line forms the sampling space for the prior $\alpha + \beta = 10$. (b) The triangular area under the line $\alpha + \beta = 10$ forms the sampling space for the prior U(0, 10). (c) The whole square area forms the sampling space for the prior gamma(1, 1). 51

4.4 A U-shaped symmetric posterior density graph for which the 95% PI is computed in two different ways: (a) a central joint PI; (b) a disjoint PI with the narrowest width. A skewed J-shaped posterior density curve, for which the calculated 95% PI is computed: (c) a central PI; (d) the narrowest PI becomes an one-sided interval bound. 52

4.5 Illustration of the random tree ensemble inducting and prediction process. The letters A to K represent 10 training instances in the training dataset. The superscript denotes the corresponding instance's class. The induced ensemble consists of four single trees, in which splitting nodes are represented by circles and leaves by rectangles. In this example, Leaf-1 to Leaf-4 are used to form the neighbourhood for a test instance. 53

5.1 Prediction accuracy comparison between EnsembleRT and Knear. EnsembleRT is better than Knear on datasets D6, D7, and D9; Knear is more accurate than EnsembleRT on D5 and D8. 60

5.2 Prediction accuracy comparison between KnearEnsembleRT and Knear. Only on one dataset (D6) do the two have significant difference, where KnearEnsembleRT outperforms Knear. 60
5.3 Prediction accuracy comparison between EnsembleRT and KnearEnsembleRT. EnsembleRT and KnearEnsembleRT outperform each other on the datasets D7, D9 and D5, D8, respectively. 60
5.4 Depiction of PIs and their targets (the ■) on eight trials. The number on the PI line is the width of the PI. In 100L% of the trials the value of the target should lie in the computed PI. The numbers marked on the lines are the widths of the PIs. 62
5.5 Illustration of the localised instances (the oval for EnsembleRT, and the small rounded rectangles for the other two), based on which the prediction is made. 63
5.6 Comparison of the PICPs (the four splines in the upper part) and the average widths of the PIs (the three splines in the lower part), computed at various confidence levels. The number of trials at each level is 1000. The red spline with upward triangles represents the theoretical (ideal) PICPs, i.e. the corresponding confidence levels. 66
5.7 Comparison of the highest and lowest widths of the PIs, computed at various confidence levels. The number of trials is also 1000. 66
5.8 Experiment results of 10,000 trials. The PIs were computed with the 95% confidence level. (a) The trend of PICP, corresponding to 1000, 5000, and 10,000 trials. (b) The trend of average width. (c) The comparison of the average value of $(\frac{k}{n} - \frac{\alpha}{\alpha+\beta})$ between the successful trials (marked as 'capture') and those failed (marked as 'miss'). 66
5.9 Results of experiments computing the PICPs and the widths of the PIs when systematically varying the union size with different sampling priors for the beta parameters. The sampled beta parameter values are: (a) $\alpha = 1.93$, $\beta = 0.14$; (b) $\alpha = 4.82$, $\beta = 7.98$; (c) $\alpha = 9.64$, $\beta = 0.36$. 69
5.10 Illustration of the distributions of the two classes of a generated artificial dataset with a single attribute. 73
5.11 Experiment results for classifier EnsembleRT on artificial datasets created with two different class proportions. 75
5.12 Experiment results for classifier Knear on artificial datasets created with two different class proportions. 75

5.13 Experiment results for classifier KnearEnsembleRT on artificial datasets created with two different class proportions. 75
5.14 Comparison between datasets with linear and nonlinear class spaces. (a) – (c) for EnsembleRT; (d) – (f) for Knear; (g) – (i) for KnearEnsembleRT. . . 77
5.15 PICP and average width at various confidence levels. The classifier used was EnsembleRT, and the dataset was ionosphere.arff. 77
5.16 Comparison of PICP, average width, and prediction accuracy when normalising and not normalising the attribute values of the dataset. (a) – (c): classifier Knear on linear dataset, and (d) – (f): Knear on nonlinear dataset; (g) – (i): classifier KnearEnsembleRT on linear dataset, and (j) – (l): KnearEnsembleRT on nonlinear dataset. 78

List of Tables

5.1 Available options for the three classifiers. 58
5.2 Datasets used in the experiments. 58
5.3 Option values tested with the three classifiers, from which the optimum values were selected. 58
5.4 Optimum option values selected for the three classifiers on the nine datasets tested in the experiments. 59
5.5 The class proportions of the artificial datasets generated using equal prior probability (the third column), and based on the class proportions of the real datasets listed in Table 5.2 (the fourth column). 76

Chapter 1

Introduction

From helping to perform critical medical diagnosis to forecasting of destructive natural phenomena, decision makers are frequently faced with the necessity of obtaining not only accurate predictions but also uncertainty estimates associated with the predictions. Machine learning technologies have become important tools desired by decision makers to make accurate and reliable predictions across various fields. However, these techniques generally fail to quantify predictive uncertainty: the prediction is a single number, and it does not provide any information about how likely the number is the desired 'true' value.

When using machine learning techniques to perform categorical data analysis, the output normally takes the form of a classification, often with a probability estimate. The estimated probability indicates, in conjunction with the classification of the training data, the degree of belief that an unknown example belongs to the predicted class. A commonly used approach to obtaining a classification and the corresponding probability estimate is to take a majority vote to obtain the classification and then compute the proportion of interest among all the observations in order to estimate the class probability. Instead of simply providing a probability estimate for the class of an unknown example, this thesis attempts to quantify the uncertainty associated with the class probability estimate, as the estimate is usually based on limited data.

1.1 A Medical Diagnosis Example

Using machine learning algorithms for the analysis of medical data has a long history. Today, technologies derived from these algorithms are well suited for specialised diagnosis problems. Many modern hospitals are equipped with expert systems built upon machine learning techniques, and patients are often diagnosed by these 'smart' programs before being actually admitted by human physicians (Kononenko, 2002, p. 2). Suppose a potential cancer patient comes to a hospital. Based on her or his symptoms, searching a database of records of previous cancer patients gives a result of five similar cases, of

which four were diagnosed with cancer. Taking a majority vote gives the diagnosis that the patient has developed cancer, and the estimated probability that this diagnosis is correct is 0.8 (four out of five).

Before asking the patient to take any further tests, can the physician be sure, based on the patient's specific symptoms, that 0.8 is the 'true' probability that she or he is having cancer? The answer to this question depends on the similarity between this patient's symptoms and the symptoms of the five previous patients, on which the diagnosis was based. It also depends on the amount of data (i.e. the number of patient records) that was involved in the diagnosis and used to come up with the estimated probability, 0.8. In fact, this probability value (0.8) is just an estimate of the likelihood that any patient, who has a set of symptoms *similar* to that of the patient currently being diagnosed, has developed cancer. This is not what the physician really wants. Ideally, the physician would like a probability, *and* an interval estimate, that are estimated based on the records of a group of patients with the *exact* symptoms of the current patient. If such data were available, the task of predicting the desired class probability, along with an interval estimate, would be easily accomplished by many existing techniques. Unfortunately, in practice, we are unlikely to observe different patients with known diagnoses who present exactly the same symptoms. Therefore probabilities and intervals of this sort cannot be directly estimated, and straightforward application of standard methods is not possible.

All we can do with machine learning is use the records from patients with similar symptoms in order to diagnose a new patient. We have noticed that the new patient is still sharing at least some of the symptoms with each of the previous patients with known diagnoses. Thus, we first use these common symptoms to select a group of patients with symptoms similar to that of the new patient, resulting in a set of values that are either 1 or 0, where 1 means cancer and 0 means not cancer. Then we derive a Bayesian model, which takes as input this set of values (i.e. 0s and 1s) to produce an entire range of plausible probabilities that contain, with a specified degree of confidence, the 'true' value of probability of the new patient having cancer. Therefore, we can use the width of the computed probability range to predict what the chances of the patient having cancer really are. If the obtained range is too wide (such as one near 1.0), which indicates the estimate is too imprecise, we may have to ask the patient to take further tests before making the final diagnosis.

A question that may arise at this point is that a patient either has developed cancer or has not, which means the 'true' probability value for the patient is either 1 or 0. How can we say a probability takes on a value somewhere between 0 and 1? The explanation for this is that we do not have complete knowledge about the physical conditions of the patient. Here, by 'complete knowledge', we mean that anything and everything that accompanies the disease and is regarded as an indication of existence of the disease must be known. In this sense, our limited knowledge, in the form of the symptoms we have observed on the patient, leads to an estimated probability value between 0 and 1, rather than just the two possible values 0 or 1.

1.2 Prediction Intervals

In statistics, an interval like the one described above is called *prediction interval* (henceforth abbreviated **PI**), which is computed to contain an unknown quantity with a specified confidence level. In the above discussed medical diagnosis example, the unknown quantity is the probability of a patient having cancer. Applying the medical example in the machine learning context, the unknown quantity is considered to be the class probability of an unknown instance. A PI is usually comprised of an upper and a lower limit, between which a future unknown value is expected to lie with a prescribed confidence level. The future value can be something that is observable, such as a person's height. Or it may be unobservable, such as the probability value in the cancer patient example.

Figure 1.1 illustrates the concept of PI. Note that the calculated vertical PI line forms a joint continuous range. In fact, a PI could comprise two (or may be more) disjoint intervals, in which case, the term prediction regions may be more appropriate. Regardless of this, the desired 'true' probability value must not, with the pre-specified degree of confidence, fall outside the interval(s). The width of the computed PI can thus be used to quantify the uncertainty about the probability prediction of how likely the estimated probability is the 'true' probability.

Despite being rather neglected (Chatfield, 1998), there have been several approaches to computing PI proposed by researchers. Unfortunately, none of the methods derived from these approaches were designed, nor are they suitable for solving the problem in our situation. The reason for this is that those PIs are computed for either an *observable*

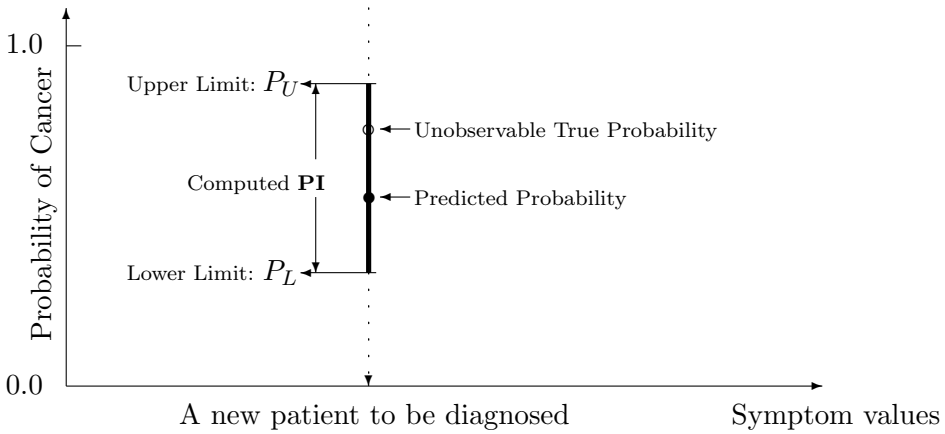

Figure 1.1: Illustration showing a prediction interval (PI).

unknown quantity, or the *observable* properties of an unknown quantity, whereas we want to compute PIs for class probabilities, which cannot be directly observed.

1.3 Aims of the Study

In this study, we restrict our attention to computing PIs for the class probability of a single unknown instance, not a set of unknown instances. We also consider only classification problems with dichotomous outcome, and do not consider the more difficult multi-class problems.

We begin with deriving a model, using the Bayesian approach. The model can be used to compute a PI for the class probability of an unknown instance, based on a set of previous instances with known classes. Then we introduce a random decision tree ensemble learning algorithm, based on which a classifier is built to produce a group of similar training instances (e.g. the instances in a leaf node of a tree). These selected instances are then used to compute an estimated class probability for an unknown instance. The class labels of those selected training instances are used as input in the proposed Bayesian probabilistic model to produce a PI for the class probability of the tested new instance. The calculated PI is supposed to capture the 'true' class probability of the unknown instance.

We conduct experiments with the aim of testing how well the Bayesian PI model performs in terms of the percentage that the PIs capture the 'true' class probabilities, and the narrowness of the widths of the computed PIs. The proposed decision tree learner is

evaluated by the measurement of root-mean-squared-error (RMSE), which is used to measure the accuracy of the generated probability estimates. The results of the experiments show that, (a) the decision tree learner has prediction performance comparable with the k-nearest-neighbour classifier, and (b) the capture percentage of the PI calculation model reaches the specified confidence level while still maintaining reasonably narrow widths.

1.4 Thesis Structure

The rest of the thesis is organised as follows. In Chapter 2, we first introduce some background on Bayesian data analysis. Then we review the basic concepts that are useful for understanding statistical intervals. We also discuss with examples three commonly used types of statistical intervals. Finally, we discuss three classification models.

Chapter 3 conducts a brief survey of literature regarding the construction of PIs. We categorise the methods of calculating PIs according to various approaches, based on which different methods have been derived. We give examples of the methods and point out the advantages and disadvantages.

The proposed Bayesian probabilistic model derived within a parametric modelling framework is developed in Chapter 4. We present the reasoning behind the mathematical form of the proposed model, and the computation of the resulting distribution. Issues such as the assumptions made during the model derivation process are also discussed. A random decision tree ensemble learning algorithm is also proposed in this chapter.

In Chapter 5, experiments are performed to evaluate the proposed ensemble learning algorithm and the Bayesian PI model. In the evaluation, we use hypothetical instances, artificially generated datasets, and datasets selected from the UCI repository of machine learning datasets. We analyse and discuss the results from the experiments on respective datasets.

The thesis is concluded in Chapter 6. We summarise the main findings of the study. We discuss the advantages and limitations of the proposed model and prioritise issues that require further investigation.

Chapter 2
Background

This chapter is devoted to material that is useful for understanding what will be discussed in subsequent chapters. We begin in Section 2.1 with a introduction to the basic concepts of Bayesian data analysis. We describe the core idea behind Bayesian thinking: updating prior knowledge about an unknown quantity with the observed data to arrive at the desired posterior distribution. This is followed by a description of the main tasks involved in the Bayesian learning process such as specifying the prior probability model. We also discuss some of the commonly used numerical methods for posterior computation. Section 2.2 introduces the basic concepts of statistical intervals. Three types of statistical intervals are described. We point out the situations in which each should be used. Section 2.3 discusses several relevant classification models, including the nearest-neighbour classifier, the linear and logistic regression models, and some ensemble learning approaches that utilise a combination of several models. We summarise the chapter in Section 2.4.

2.1 Bayesian Data Analysis

Driven by the availability of modern computing capabilities, as well as the philosophical advantages of Bayesian thinking, applications of Bayesian data analysis have rapidly appeared in many different fields in recent years. In this section, we introduce the basic concepts of Bayesian data analysis.

2.1.1 The Bayesian Framework

The Bayesian approach to data analysis computes conditional probability distributions of unknown quantities, such as future observations, based on the observed data. Let y denote an unknown quantity of interest and D denote the observed data. The goal is to obtain a probabilistic statement about the unknown quantity y given D: $p(y \mid D)$.

From the definition of conditional probability, we can make the following statement about the joint probability, $p(y, D)$, which describes how y and D behave in conjunction

$$p(y, D) = p(y)p(D \mid y) \tag{2.1}$$

The first function on the right-hand side of the equation, $p(y)$, is called the prior distribution of y. It is termed prior because it is specified before incorporating the observed data into the model. The form of $p(y)$ depends on our prior knowledge about y. The second factor, $p(D \mid y)$, is the likelihood function, which represents how likely the data D is, based on y. Yet, the joint probability $p(y \mid D)$ is what we are really interested in – the distribution of the unknown quantity y. This is where Bayes' theorem comes into play.

Bayes' theorem is a result of probability theory. It forms the most fundamental basis of probability calculation. Suppose that there are two events, A and B. The axioms of probability tell us that the probability of A conditional on B is given by:

$$p(A \mid B) = \frac{p(A, B)}{p(B)} \tag{2.2}$$

Likewise

$$p(B \mid A) = \frac{p(B, A)}{p(A)} \tag{2.3}$$

Since $p(A, B) = p(B, A)$, rearranging (2.2) and (2.3) gives

$$\begin{aligned} p(A \mid B)p(B) &= p(B \mid A)p(A) \\ p(A \mid B) &= \frac{p(A)p(B \mid A)}{p(B)} \end{aligned} \tag{2.4}$$

Equation (2.4) is the famous Bayes' theorem.

Following a similar process as above, we can produce the following equation from (2.1):

$$p(y)p(D \mid y) = p(D)p(y \mid D) \tag{2.5}$$

Rearranging it produces

$$p(y \mid D) = \frac{p(y)p(D \mid y)}{p(D)}, \tag{2.6}$$

which gives the desired probability distribution of the unknown quantity y conditioned on the observed data D. The denominator $p(D)$ is the unconditional probability of D.

In Equation (2.6), if y is a continuous random variable, the term $p(D)$ can be computed as

$$p(D) = \int p(D, y)dy = \int p(y)p(D \mid y)dy, \tag{2.7}$$

which is achieved by integrating $p(D)$ over all possible support values of y. In the case of discrete y, the sum over y is used instead, i.e. $p(D) = \sum_y p(y)p(D \mid y)$. $p(D)$ is typically called the normalising constant, or the prior predictive distribution. Its purpose is to ensure that $p(y \mid D)$ integrates to one, which is required by the definition of probability density function. Because the denominator $p(D)$ is independent of y, omitting it from Equation (2.6) yields

$$p(y \mid D) \propto p(y)p(D \mid y)$$

This states that the unnormalised posterior distribution of y is proportional to ($\propto$) the prior distribution times the likelihood function, i.e.

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

We can summarise the preceding discussion as the following Bayesian learning process: specify a probability model that incorporates some prior knowledge about the unknown quantity, then incorporate the information from the observed data into the specified probability distribution through the likelihood function, and finally, derive (analytically or by simulation) the posterior distribution of the unknown quantity.

There are some assumptions implied in the summarised Bayesian learning process. First, the probability model specified for the unknown quantity is in parametric form, which is chosen by the individual modeler. This highlights the main difference between parametric and nonparametric modelling. Second, from the Bayesian perspective, the unknown quantity is probabilistically described and thus assumed to follow a distribution rather than have a fixed value as in the traditional 'frequentist' approach (Gill, 2002, p. 3).

2.1.2 Model Specification

Having placed ourselves in the Bayesian parametric modelling framework, the first step towards making predictive inferences about an unknown quantity is to assign a probability distribution for it. This is actually the process of encoding our prior knowledge about the unknown quantity into a probabilistic parametric model. We now discuss some of

the principles of assigning such a model and review the typology of prior distributions commonly applied in Bayesian work.

The Principles

Developing Bayesian models requires specifying prior distributions for unknown quantities. Gill (2002, p. 114) discusses three different approaches that can be applied in the specification of prior distributions. Classical Bayesians consider prior distribution as an inconvenience and thus tend to specify a noninformative prior so as to inject the least possible amount of prior knowledge. Modern parametric Bayesians prefer conjugate priors because of the benefit of mathematical conveniences. Subjective Bayesians derive prior distributions based on existing scientific knowledge from previous empirical work in the field. In practice, these three categories are not mutually exclusive, and it is common to use a mixed approach that adopts a prior combining various aspects.

In practice, model specifications based on the observed data are much recommended. Gelman (2004, p. 14) states the following principle of how probability models can be specified: "whenever there is replication, in the sense of many exchangeable units observed, there is scope for estimating features of a probability distribution from data and thus making the analysis more 'objective.'" Another general approach, which also specifies models based on the observed data, is summarised by Gregory (2005, p. 185). In that, constraint information is first abstracted from observed data (called *testable information*). Then, if there is more than one probability distribution that agrees with the given constraint information, select the one that "maximises the uncertainty in the probability distribution, while still being maximally constrained by the given testable information", so as to minimise the subjectiveness injected by the modeler.

Common Priors

- Proper or Improper Prior

  Proper priors are distributions that add, in the case of probability mass function (PMF) for discrete variables, or integrate, in the case of probability density function (PDF) for continuous variables, to a finite quantity. The following is an example of

proper prior (Press, 2003, p. 54):

$$p(y) = 1, \quad 0 \leq y \leq 1 \tag{2.8}$$

where y is an unknown quantity. This is a proper prior because it has a bounded value. It is also called a normalised proper prior, because it integrates to one. Improper priors are ones that do not possess bounded values. For example, if there are some reasons not to bound the value to be less than one, then a prior

$$p(y) = 1, \quad 0 \leq y \leq k \tag{2.9}$$

can be specified. This prior is unnormalised because it does not integrate to one; but it is a proper prior because it still yields a finite value.

In most circumstances, it is more common to specify proper priors because they lead to the desired proper posteriors, and thus the check of properness of the resulting posterior can be relaxed. Moreover, since an improper posterior only occurs in situations where an improper prior is specified, improper priors must be used with caution (Gill, 2002, p. 128).

- Noninformative Prior

  A noninformative prior is one that provides the least amount of knowledge about the unknown quantity. Noninformative priors are usually given in situations where there is no previous subjective information about the quantity of interest known to the modeler. Thus, they are sometimes referred to as the 'ignorance' prior. The uniform distribution, among some others, is an obvious choice of such noninformative prior. As discussed above, a uniform prior can be specified as a proper or an improper prior by defining it over a bounded or an unbounded space.

- Conjugate Prior

  In Bayesian probability theory, a conjugate prior is a family of prior probability distributions which has the property that the posterior probability distribution also belongs to that family. In other words, when both the prior and posterior come from the same distribution family, then the prior and likelihood are said to be conjugate.

  Consider the problem of inferring a distribution for the unknown quantity y, which

we discussed in Equation (2.6) and (2.7),

$$\begin{aligned} p(y \mid D) &= \frac{p(y)p(D \mid y)}{p(D)} \\ &= \frac{p(y)p(D \mid y)}{\int p(y)p(D \mid y)dy} \end{aligned}$$

Different distributions of the prior $p(y)$ may cause the product $p(y)p(D \mid y)$ to have different forms. For certain choices of $p(y)$, the posterior has the same algebraic form as the prior. Such a choice is a conjugate prior. A conjugate prior is a mathematical convenience: otherwise a difficult numerical integration may be necessary.

Examples of some sampling distributions along with their corresponding natural conjugate priors include the Poisson distribution with its mean following a gamma distribution, and the exponential distribution with a gamma distributed mean, etc.

After specifying a probability distribution that models the quantity of interest, the next step involves combining the specified probability model with the observed data through the likelihood function. Since we are in the Bayesian parametric modelling context, probability distributions are specified with parameters. These parameters need to be estimated in order to compute the posterior distribution. We shall discuss the estimation of model parameters in the next section.

2.1.3 Numerical Computation

Numerical techniques arise due to the increased use of complex models, especially when the parameter vector of the model is high-dimensional, for which the analytical calculation is often not possible. Moreover, for models that are analytically solvable, numerical computation sometimes is still required. The advent of modern powerful computers has made such kind of computation much easier.

In numerical Bayesian analysis, we are mostly interested in estimating an integral quantity and thus obtaining the posterior distribution, from which the desired inference can be drawn. By utilising simulation, analytically unsolvable integrals can be delivered through approximation. In particular, a set of simulated values that exhibit the

distributional properties as the posterior density are first generated. Then the empirical distribution of these simulated values can be determined and used to describe the posterior.

Simulation techniques form a central part of much applied Bayesian analysis. With simulation techniques involved, it is relatively easy to generate samples from a probability distribution. Over the years, numerous such techniques have been proposed by researchers. What follows is a brief summary of some of the methods used in the numerical computation of integrals.

- Mote Carlo Integration

  This method is based on the idea that an integration can be approximated by summing over a large number of simulated values. If the integration is bounded in a range, for example $[\alpha : \beta]$, then the values are randomly generated only within $[\alpha : \beta]$.

- Rejection Sampling

  This method is used instead of the basic Monte Carlo method when simulating the required samples is not possible. The idea is to obtain an integral quantity through generating random samples, but only accept those that are determined to belong to the correct distribution.

- Importance Sampling

  This is an improved version of the rejection sampling method, and is useful for quantile estimation. The idea is to control the sampling in order to take more samples from the part of the distribution that is important to the problem being estimated. It thus places more emphasis on higher density regions than others, and hence achieves more efficiency through variance reduction.

Numerical methods have progressively become important tools in Bayesian data analysis. However, there have been concerns about "less human consideration of the details," which may cause the simulation results to be over-trusted (Gill, 2002, p. 280). To avoid problems of such kind, analytic calculations should be tried as the first solution whenever possible.

2.2 Statistical Intervals

This section presents the basic concepts of statistical intervals. We discuss the difference between point and interval estimation. We differentiate at an elementary level among three types of statistical intervals. We also discuss the necessary assumptions that are required in the calculation of the intervals, and point out the situations where each type of intervals should be used.

2.2.1 Point Versus Interval Estimation

When an unknown quantity is estimated, the estimate can either be a single number – a point estimate, or an range of values – an interval estimate. Point estimates provide no information about the precision and reliability of estimation, and hence are usually not as informative as interval estimates.

Suppose a sample of hot dogs is randomly selected from a production process, and the average value $\bar{x}$ of the fat content of these hot dogs is calculated. If $\bar{x}$ is used to provide a point estimate of the true average fat content μ of all hot dogs that are produced from the same process, because of sampling variability, it is quite unlikely that $\bar{x} = \mu$. The value $\bar{x}$ is of course eligible for estimating μ, but to what extent is this estimate reliable? An answer to this question is to calculate a range of possible values of the true average fat content, i.e. an interval estimate.

Interval estimation predicts an unknown quantity by not giving exact answers. It admits uncertainty or inability to estimate a single, exact value of the unknown quantity. Through the following properties, interval estimates can provide much more information about the predicted quantity than point estimates:

- Confidence level

  An interval estimate is always calculated by first selecting a *confidence level*. The phrase confidence level is used to describe the likelihood that the resulting interval does contain the unknown quantity. The confidence level is usually expressed as a percentage. In practice, the most commonly used percentage value is 95% (or 0.95).

  Depending on whether a Bayesian or 'frequentist' approach is used, an interval computed with a 95% confidence level can be interpreted in two ways. On one hand, it means that for 95% of the time the intervals computed using the same sampling

procedure will capture the true value of the unknown quantity, but for the other 5% of the time they will not. It is equivalent to say that 95% of all possible random samples from the population can result in an interval that will capture the true value of interest, as long as the calculating procedure is consistent. Despite this, if we select any of the random samples (or any of the calculated intervals), we would not be able to know whether it is in the successful 95% portion or in the failed 5%. This is known as the 'frequentist,' 'classical,' or 'sampling theory' approach to statistical inference. On the other hand, Bayesian theory states that, for any of the calculated intervals, we have 95% confidence that it will contain the true value of the unknown quantity, as long as the same procedure is used; in the meantime there is also 5% chance that a computed interval will not contain the true value of the unknown quantity.

It would be wonderful if we could be 100% confident that a calculated interval contains the target value. Unfortunately, unless the entire population has been sampled or the interval is so wide as to be useless, gaining 100% confidence is not possible. An unusually high confidence level will probably cause the calculated interval not to be informative.

- Interval width

  Information about the precision of an interval estimate is conveyed by the width of the interval. The width of statistical interval varies according to two factors: confidence level and sample size. The confidence level is the degree of assurance that the calculated interval contains the value of the unknown quantity. Thus, when choosing a confidence level, the risk of not capturing the target value must be taken into account because when the level of confidence is raised, the width of the resulting interval becomes wider. Statistical intervals are usually constructed based on limited sample data. In general, for a fixed level of confidence, narrower intervals are expected with larger samples. Thus, when the sample sizes are relatively small,the confidence level has to be decreased in order to obtain a narrower interval.

Putting it all together, if the confidence level is set to a high value and the resulting interval is quite narrow, our knowledge about the value of the unknown quantity is reasonably precise. A wide interval however, might indicates that there is a great deal

of uncertainty concerning the estimated value. In general, how much confidence we need to hold when calculating a statistical interval and how precise we want the calculated interval to be depend on specific applications. For instance, in critical situations, such as diagnosing a patient with a life-threatening disease, or analysing whether a terrorist group is going to launch a nuclear strike, we need to set up a high confidence level and a narrow width, whereas in situations that are less serious, lower confidence levels and wider widths may be acceptable.

2.2.2 Types of Statistical Intervals

Having defined the term statistical interval, we now look at the three frequently used statistical intervals: confidence interval, tolerance interval, and prediction interval (PI) (Hahn & Meeker, 1991, pp. 2–3). They each serve different purposes.

- A confidence interval, the most commonly used type of statistical intervals, is computed to contain an unknown value of a property of a population or process. The properties of a population include its mean and standard deviation, etc. We have discussed an example in Section 2.2.1, in which an interval is calculated, based on a random sample of the products, to contain the average fat content of hot dogs produced from the same production process. This is an example of computing a confidence interval to contain the population mean.

- Tolerance intervals are defined to contain a specified proportion of the population, from which a sample was drawn. For instance, based on the sampled data of hot dog fat content, we might wish to compute an interval to contain the fat content values of at least 95% of the hot dogs produced from the same production process.

- A prediction interval (PI) is computed to contain one or more future observations from a previously sampled population. For example, after having measured 20 hot dogs, if an interval is computed to contain the fat content of the 21^{th} hot dog that is randomly selected from the same process, the computed interval is a PI that contains a single future observation. If the fat content values of five such hot dogs were claimed to be included in the computed interval, then the interval is called a PI for all five future observations. And intuitively, it is also possible to calculate a PI that contains the average fat content of all five future hot dogs.

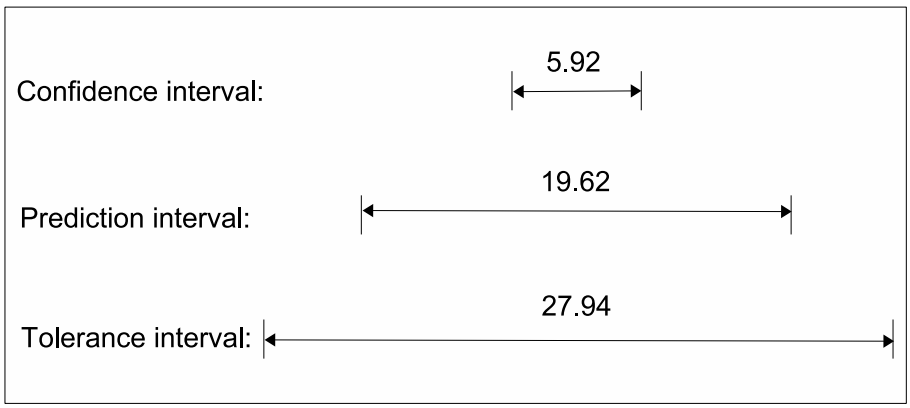

Figure 2.1: Comparison of widths of statistical intervals for the same example.

We now perform a quick comparison between the three types of intervals by their relative widths computed from the same example. The following is a sample of fat content (as a percentage) of 10 hot dogs (Beilken, Eadie, Jones & Harris, 1990, pp. 395–409; Devore, 2000, p. 299):

25.2, 21.3, 22.8, 17.0, 29.8, 21.0, 25.5, 16.0, 20.9, 19.5.

By assumption, these were randomly selected from a normally distributed hot dog population. With the same 95% confidence level, we compute a confidence interval for the mean fat content of the population, a PI for the fat content of the 11^{th} hot dog, and a tolerance interval to contain the fat content percentages of at least 95% of the hot dog population. Figure 2.1 shows the computed intervals [based on Figure 5.1 in (Wadsworth, 1998, Section 5.6)]. Note that, both the PI and the tolerance interval are substantially wider than the confidence interval. This is because the confidence interval is computed to contain the average fat content of the hot dogs, which is a fixed value regarding a particular population. Whereas the PI is calculated to include the fat content of a single future hot dog, which is a random variable. The tolerance interval has the largest width because it is computed to cover, not a single value, but the values of 95% of the population.

2.2.3 Why Do We Need PIs

As described above, various types of statistical intervals exist and users can use them to answer relevant questions from given data. This requires users choose an appropriate type of interval in order to solve a specific problem. The following general guidelines on the choice of statistical intervals are given in (Hahn & Meeker, 1991, Section 2.1). In particular, it involves answering the following two questions:

(1) Is the interval for description or prediction?

In general, statistical intervals are designed for two main purposes: "describing the population or process" from which random samples were selected, or "predicting the results of a future sample from the same population." (p. 27) Regarding the differentiation of the three types of intervals, confidence interval and tolerance interval are more related to the purpose of description, whereas the intention of PI is more of prediction.

(2) What is the interval like?

By examining the characteristics of an unknown quantity, for which statistical intervals are computed, we can classify which type of intervals is appropriate in a specific situation. It is equivalent to answer the following questions: is the unknown quantity about (a) *location* – what the single future value might be; (b) *spread* – how a sample or population deviates (standard deviation); (c) an *enclosure* – how large the proportion of a sample or population could be; (d) *limit* – probabilities that a value will exceed a threshold (Hahn & Meeker, 1991, p. 28).

Here is an example that can help to elaborate the above discussed guidelines. When we are testing the performance of a classifier in a series of experiments, we may be interested in the accuracy of the classifier in the next experiment – a PI; its average accuracy over all the experiments – a confidence interval; or the accuracy values of at least 95% or 99% of total experiments – a tolerance interval.

Consider the example of diagnosing a cancer patient (Section 1.1), what we are really interested in is the uncertainty about the cancer probability of the next patient – a new individual – based on a group of patients with symptoms similar to that of this new patient. Here, we assume that the given group of patients forms a random sample that was drawn from the same population as the new patient. Hence, what we need is a PI that is computed based on a observed sample and contains the desired probability value of a single future observation (the new patient) with a specified confidence level.

2.2.4 Assumptions in Interval Calculation

The calculation of statistical intervals normally requires certain assumptions, either explicitly or implicitly. Two basic assumptions are the random sample assumption and

the normality assumption. These are discussed in this section.

Random Sample Assumption

The random sample assumption is important because the way that how samples are selected from a population may affect the validity of the inferences about the properties of the population computed based on the selected samples. For the resulting inferences to be valid, samples must be drawn randomly in order to be representative of a population.

The following example illustrates how to draw a random sample (Ross, 2004, p. 217). Consider selecting a sample of size 2 from a population consisting of three elements, a, b, and c. To draw a random sample, the first element of the sample must equally likely be any of a, b, or c; the second element must then equally likely be any of the remaining two elements of the population, i.e. sampling without replacement. In other words, the sample is equally likely to be any of three subsets, (a, b), (b, c), and (a, c). Moreover, a sample is said to be random also means that the sample elements are independent of one another. This may not be the case in the cancer patient example. If the patients were from the same town or region where the living conditions, such as air and water, are quite similar, this assumption would be highly questionable.

In the case of computing a PI for a future observation, the random sample assumption implies that the sample that is used to estimate the properties of the underlying population must be selected from the same population as the future observation for which the PI is to be calculated. Going back to the cancer patient example again, since the candidate patients with known diagnoses are from records in the past, whereas the real interest is in diagnosing an upcoming new patient. In this case, and in many others, the sampled population may differ from the population about which inferences are to be drawn, and thus PIs calculated using the 'sampling theory' approach (or 'frequentist' approach) based on such an invalid assumption may not be correct.

Normality Assumption

The normal distribution is the most commonly assumed probability distribution because it can be observed in many natural processes. In situations where the normality assumption is met, the mean and standard deviation of a population can be estimated using the mean and standard deviation of a random sample drawn from the population.

When the normal distribution assumption cannot be justified and the sample size is also fairly small, the sample mean becomes a random variable from a population of an unknown distribution. It would thus create a great deal of uncertainty by using the sample mean to estimate the population mean, which is fixed, but unknown value. This also holds for the sample standard deviation. Hence, the calculation of statistical intervals using the population mean and standard deviation estimated based on the random samples would lead to seriously incorrect intervals (Hahn & Meeker, 1991, Section 4.9).

When the sample size is large, however, the normality assumption may be relaxed. For example, in the process of computing a confidence interval to contain the population mean, an important theorem called the Central Limit Theorem can make the normality assumption less critical than it would be with small sample sizes (Wadsworth, 1998, Section 5.2.3). The Central Limit Theorem states that, when the sample size is large, no matter what the nature of the underlying population distribution is, the sample mean will have an approximately normal distribution. Also, the larger the sample size, the better the approximation. Hence, a normality-assumption-bounded procedure for computing, for example, a confidence interval for the population mean may be used in situations where the normal distribution assumption is not strictly met.

There are also other statistical intervals, of which the calculations can make use of the Central Limit Theorem when the underlying distribution is not normal but the sample size is fairly large. For example, when calculating PIs containing the mean fat content of all hot dogs sold in the whole month next month (i.e. PIs to contain the mean of a future sample), the normality assumption can be relaxed given the size of the future sample is large. Unfortunately, for PIs to contain a single future value (which is of particular interest in our study), the use of the Central Limit Theorem cannot be justified (because the size of the future sample is only one). Therefore, normal-distribution-based methods for computing PIs to contain a single future observation may be appreciably off when sampling from a non-normal population (Hahn & Meeker, 1991, Section 4.9, p. 65).

It is worth pointing out that, in addition to the above two assumptions, statistical intervals are also generally computed by assuming symmetricalness around the mean, or whatever the point predictor being used. This implicitly assumes that the predictor is an unbiased point estimate. Hence, they are often called unbiased mean and standard

deviation predictors.

2.2.5 Distribution-free Statistical Intervals

When the assumption of normality is not met, one may have to find the underlying distribution in order to produce the desired statistical intervals for an unknown quantity. An alternative to this is the distribution-free approach. Statistical intervals computed using distribution-free methods are usually not as informative as those based on an assumption of the underlying model (Wadsworth, 1998, Section 5.2.3).

In particular, distribution-free methods have the following limitations. First, distribution-free intervals will normally be wider than the corresponding intervals based on a particular distribution assumption, given the assumed distribution is correct. One way of looking at this limitation is to image that distribution-free intervals extend the computed widths to compensate for the void of uncertainty, which is eliminated by the distributional assumptions.

The second drawback of distribution-free intervals comes from the way they are computed: they use specifically selected values in the sample as interval endpoints. Because the observed sample values are fixed, it is generally not possible to obtain an interval with the desired confidence level (Hahn & Meeker, 1991, Section 5.1, p. 76). In other words, the intervals are 'observed' rather than calculated. An example of calculation of such distribution-free intervals will be discussed in the next chapter.

However, as the name implies, distribution-free intervals do not require one assume a particular population distribution (although the randomness assumption still pertains). Because of this, statistical intervals computed this way have less inherent uncertainty introduced by distributional assumptions than ones calculated otherwise, and thus are still useful in certain situations.

2.3 Classification Models

Interval estimates are computed as supplements for point estimates that are made by classification models. A classification model stores information that can be used to make predictions about an unknown example; the stored information can also be used to

compute intervals for the unknown example. In this section, we discuss three classification models.

2.3.1 Regression Model

Regression analysis concerns with modelling relationships among variables. It quantifies how a response (or dependent) variable is related to a set of explanatory (independent or predictor) variables. If the true relationships among the variables were known exactly, one would be able to accurately predict the responses corresponding to new explanatory variables. Unfortunately, this is rarely the case and one has to rely on empirical evidence to develop approximations to the relationships.

Moreover, the responses will vary from time to time, because in practice the experiment cannot be repeated under absolutely identical conditions. The variation of the responses is called noise, which occurs from one repetition to the next. The noise can come from many sources, for example, measurement errors. To take this into account, a probabilistic model is needed to approximate the relationship in order to fulfill the prediction tasks.

The most common form of structural assumption about the relationship is that, there is a single response variable Y, which depends on the values of a set of explanatory variables through a mathematical function f and an additive random error component ϵ, such that

$$Y = f + \epsilon \tag{2.10}$$

where the error term ϵ is assumed to be normally distributed with zero mean. This is called the regression model. Since ϵ is a random variable, the model (2.10) is probabilistic.

To use the regression model for prediction, the unknown regression function f must be determined. This can be achieved by first making assumptions about the form of the relationship, then estimating the parameters of the model (called *regression parameters*), based on the observed data, $(x_1, y_1), \ldots, (x_n, y_n)$, which consists of n observed responses at corresponding predictor locations.

The Linear Regression Model

To approximate the form of the relationship between the explanatory and response variables, a simple and historically much used solution is to make direct linear assumption about f, i.e.

$$y_i(x_i) = f(\beta; a^i) + \epsilon = \beta_0 + \beta_1 a_1^i + \cdots + \beta_m a_m^i + \epsilon, \tag{2.11}$$

where $f(\beta; a^i)$ is a mathematical function of the m independent variables $a_1^i, \ldots, a_m^i$ at the predictor location x^i and the unknown parameters $\beta_0, \beta_1, \ldots, \beta_m$.

In the machine learning context, $y_1, \ldots, y_n$ are the classes of the corresponding instances $x_1, \ldots, x_n$. Variables $a_1^i, \ldots, a_m^i$ are the attributes of instance x_i, $i = 1, \ldots, n$; and $\beta_0, \beta_1, \ldots, \beta_m$ are the weights observed together with the attributes. The quantity ϵ is a random variable, which is assumed to be normally distributed. The quantity ϵ plays an important role in computing the value of the response variable in regression models. Without ϵ, any observed pair (x, y) would correspond to a point falling exactly on the line $y(x) = \beta_0 + \beta_1 a_1 + \cdots + \beta_m a_m$ with resolved parameters $\beta_0, \beta_1, \ldots, \beta_m$. This line is called the *true regression line*.

We can use linear regression to predict the numeric class if all other attributes of the instance are also numeric. The unknown parameters, $\beta_0, \beta_1, \cdots, \beta_m$, can be estimated using the well-known principle *least squares*. The least squares method minimises the sum of the squares of the difference between the predicted and the actual class values. It thus measures the goodness of fit of an estimated regression line to the observed instances. The estimated parameters are called the *least squares estimates*.

The method of using linear regression for classification in domains with numeric attributes is called *multiresponse linear regression*. The idea is to estimate a set of parameters for each of the classes. It proceeds by setting the output of the regression function equal to one for instances that belong to the class and zero for those that do not. Thus, each class has a linear regression function formed by the estimated set of parameters. This again makes use of the least squares estimation technique. Given a new instance, its membership values for each class are calculated and the class corresponding the biggest value is chosen as its classification (Witten & Frank 2005, p. 119).

Linear regression is an excellent, simple method for numeric prediction and classifica-

tion. However, several drawbacks exhibit.

(a) If the response has a nonlinear relationship with the predictor variables, the 'best-fitting' regression line may not fit very well.

(b) The least squares line should not be used to make a prediction about responses at input levels that are far from those used to obtain the estimated regression line. This is because the fitted relationship, i.e. the estimated regression line, may not be valid for such values due to the "danger of extrapolation" (Devore, 2000, p. 501).

(c) Multi-response linear regression can not produce proper class probability values. That is, the regression equation

$$Pr(y \mid x) = \beta_0 + \beta_1 a_1 + \cdots + \beta_m a_m$$

computes the probability $Pr(y \mid x)$, which must be a value between 0 and 1, but $\beta_0 + \beta_1 a_1 + \cdots + \beta_m a_m$ need not to be in the range.

(d) The independency and the normality assumption that are required by the least squares principle are violated (Witten & Frank, 2005, p. 121).

The Logistic Regression Model

Logistic regression does not suffer from the problems (c) and (d) discussed in the last section. It allows $Pr(y \mid x)$ to be a function of $(\beta_0 + \beta_1 a_1 + \cdots + \beta_m a_m)$, rather than $(\beta_0 + \beta_1 a_1 + \cdots + \beta_m a_m)$ itself. The function is called *logit function* and, for a binary-class problem, has the form

$$Pr(1 \mid x) = \frac{e^{\beta_0 + \beta_1 a_1 + \cdots + \beta_m a_m}}{1 + e^{\beta_0 + \beta_1 a_1 + \cdots + \beta_m a_m}} \tag{2.12}$$

Straightforward algebra gives

$$\frac{Pr(y \mid x)}{1 - Pr(y \mid x)} = e^{\beta_0 + \beta_1 a_1 + \cdots + \beta_m a_m} \tag{2.13}$$

where the expression on the left-hand side is called the *odds ratio*.

Instead of using the squared error in linear regression, logistic regression uses the $log - likelihood$ to measure the goodness of fit. The log-likelihood function of the model

is given as

$$\sum_{i=1}^{n}(1-y_i)log[1-Pr(1 \mid x_i)] + y_i log[Pr(1 \mid x_i)]$$

Just as in linear regression, fitting the logistic regression to the observed data requires the parameters $\beta_0, \beta_1, \cdots, \beta_m$ be estimated so that the log-likelihood reaches its maximum value. Maximising the log-likelihood gives the parameter values, for which the observed data is most likely to be generated (Witten & Frank, 2005, Section 4.6). The details of the maximum likelihood technique are quite involved. Fortunately most prepackaged software will readily do this on request.

2.3.2 Nearest-Neighbour Classification

The standard nearest-neighbour approach is quite simple. Given the training data, the nearest-neighbour classifier finds the closest (according to some distance metric, which will be discussed further below) training instance to an unknown test instance, and predicts the class of that training instance.

The k-nearest-neighbour algorithm is a variation of standard nearest-neighbour. Instead of searching for only one instance that is closest to the test instance, k such neighbouring instances are computed and the most frequently occurring class in the k neighbours (or the distance-weighted average, in the case of numeric class) is assigned to the test instance. In the event of a tie, a class that is randomly chosen between the tied classes can be used.

In the nearest-neighbour approach, which instance is selected as a neighbour depends on the distance between the instance being examined and the test instance. The usual choice of the distance metric is Euclidean distance ($\sqrt{\sum_{i=1}^{n}(x_{1i}-x_{2i})^2}$). But others are also possible, among which are Absolute Distance ($\frac{1}{n}\sum_{i=1}^{n}|x_{1i}-x_{2i}|$), City Blocks ($\arg\max_i|x_{1i}-x_{2i}|$) and Mahannobolis ($\sqrt{\sum_{i=1}^{n}\omega_i(x_{1i}-x_{2i})^2}$, $\sum_{i=1}^{n}\omega_i = 1$) (Holmes & Adams, 2002).

The nearest-neighbour classifier has been widely used in the field of statistical pattern recognition. It has the advantages of being able to learn quickly from a small dataset, working well for numeric data, and simplicity and ease of implementation. However, one of its limitations is that it can only provide predictions of class label for a new instance;

it cannot be used to derive a suitable model from the training data. Thus, there is no probabilistic interpretation that can be attached to those predicted labels. Another disadvantage of the the algorithm is that some attributes are more associated with the class than others, and deriving suitable attribute weights from training set to reflect this can be difficult (Witten & Frank, 2005, pp. 78–79).

2.3.3 Ensemble of Classifiers

Combining multi-classifiers to make predictions, instead of using a single model, is a common approach to increasing predictive performance. Several techniques based on this approach have emerged in the machine learning context, prominent among which are schemes called *bagging*, *boosting*, and *stacking*. In general, these can be used to make numeric predictions as well as classifications. The first two generate ensemble classifiers using the same learning algorithm (e.g. decision tree or neural network training), while the third approach combines classifiers generated using different learning algorithms (Dzeroski, 2004, p. 255).

A summarisation given in (Dietterich, 1998, pp. 838–839) listed the following four methods that have been previously used to construct ensemble classifiers:

1. Manipulating the training data

   The bagging and boosting methods can be characterised into this category. Bagging works by resampling the original dataset, whereas boosting combines multiple classifiers by explicitly seeking models that complement one another.

2. Manipulating the attributes

   This strategy relies on the fact that some subset of the attributes may be more effectively associated with the class than all the attributes as a whole. By performing attribute selection, prediction accuracy can be improved. Preprocessing methods play an important role in this approach.

3. Manipulating the output targets

   The use of error-correcting output codes is an example of this strategy. In general, the method proceeds by first learning a set of classifiers from the training dataset. Each of the classifiers then have to vote for the classes. Eventually the prediction is assigned to the class with the highest number of votes. It is claimed that better

performance from both C4.5 decision trees and the back propagation neural network method is achieved on many classification problems by using this technique (Dietterich & Bakiri, 1995).

4. Randomising the fitting procedure

   This is an alternative to the method bagging, in which randomness is injected into the learning algorithm's input in order to generate a diverse ensemble of classifiers.

2.4 Summary

In this chapter, we have discussed some background knowledge about Bayesian data analysis, statistical intervals and predictive models, which are useful for the development of PI computation methods. We looked at some general principles of how to encode prior knowledge into a probability model. We discussed how different choices of prior distributions relate to corresponding posterior distributions. We also discussed some numerical techniques that are commonly used to compute Bayesian distributional models. Different types of statistical intervals are designed for different purposes. We have given examples of how to choose the appropriate type of interval in various situations and explained why prediction interval (PI) is the appropriate interval for expressing the uncertainty about the class probability of a single unknown instance when the prediction is made based on a group of similar instances. Finally, three classification models, regression, nearest neighbour classification, and ensemble learning were discussed in Section 2.3. In the next chapter, we will present a literature survey of existing approaches to PI calculation.

Chapter 3

Literature Review

3.1 Introduction

As stated in earlier chapters, measures of precision and reliability in predictions are of paramount importance in many applied fields, such as forecasting and medical diagnosis. One way to express the uncertainty about a point prediction is to provide an interval estimate to supplement the point estimate. Thus, more information about the predicted quantity is given by the computed interval. A PI is computed to contain, with a certain confidence level, a range of possible estimates, in which the 'true' value of the quantity is supposed to lie. The uncertainty is thus quantified by the width of the computed PI.

Whilst pointwise prediction and the well-known confidence interval have been widely studied, the construction of PIs has received fairly little attention. The literature with regard to calculating PIs has been surveyed by Chatfield (1993). Previously proposed approaches include: (a) using theoretical formulae conditional on a best-fitting model; (b) relying on the observed empirical distribution of prediction errors, rather than assuming that the chosen model is true; (c) using simulation and resampling methods to generate possible future trends either based on simulation from the fitted model or by finding the variance of the prediction errors; (d) applying a distribution-free approach; and (e) computing PIs based on the Bayesian approach. This chapter reviews these approaches and presents insight into how PIs are computed using various methods.

3.2 PI Computation Using Theoretical Formulae

The commonest approach to computing PIs is to use existing theoretical formulae conditional on a best-fitting model. When using this approach, the basic steps in the construction of a PI can be summarised as follows:

Step - 1 By assumption, find the probability distribution of the predicted unknown quantity, for which a PI is to be computed.

Step - 2 Compute the point estimate of the unknown quantity.

Step - 3 Compute the probability distribution of the residual – the difference between the unknown quantity and its estimate, and derive a PI based on the resulting distribution and the specified confidence level.

For the purpose of demonstration, we discuss a method of constructing a PI for the value of a new response variable y_{n+1} at the input level x_{n+1}. The prediction uses a simple linear regression model, based on the observed data $(x_1, y_1), \ldots, (x_n, y_n)$. The discussion below closely follows and extends the discussion in Ross (2004, p. 374).

For Step - 1, suppose the response y_{n+1} follows a normal distribution, and has its expected value calculated as $E[y_{n+1}] = \alpha + \beta x_{n+1}$ (the true regression line discussed in the last chapter) and its variance σ^2, i.e. $y_{n+1} \sim \mathcal{N}(\alpha + \beta x_{n+1},\ \sigma^2)$. The notation $\sim \mathcal{N}(\cdot)$ means that the random variable is normally distributed with the necessary parameters inside the parentheses. The parameters, α and β, and the standard deviation σ are yet to be estimated based on the observed data.

The next step (Step - 2) towards computing a PI for the response y_{n+1} is to find a point estimator for it. Let A be the estimator of α and B be the estimator of β, then the estimator of y_{n+1}, corresponding to the input x_{n+1}, is $A + Bx_{n+1}$. By using the method of least squares to minimise the sum of the squared differences between the estimated responses and the actual response values, $\sum_1^n (y_i - A - Bx_i)^2$, it is not difficult to compute the two estimators A and B. This is actually the process of fitting the model by finding the most appropriate parameters and thus making the error term the smallest.

With A and B estimated, the task in Step - 3 is to determine the probability distribution of the residual $\varepsilon = y_{n+1} - (A + Bx_{n+1})$. The residual is a linear combination of two independent normal random variables, so itself is normally distributed. The properties that need to be determined for this distribution are the mean and variance. The residual has the mean value of zero because the two random variables y_{n+1} and $A + Bx_{n+1}$ have the same mean $\alpha + \beta x_{n+1}$. Its variance is the addition of the variances of the two terms, i.e. $Var(\varepsilon) = Var(y_{n+1}) + Var(A + Bx_{n+1})$. Based on the normality assumption of y_{n+1}, and the relationship between the normal distribution and the chi-square distribution, both

$Var(y_{n+1})$ and $Var(A + Bx_{n+1})$ can be computed using the observed data. Hence, with $100L$ percent confidence, the value of the response y_{n+1} at the input level x_{n+1} will be contained in the interval

$$A + Bx_{n+1} \pm t_{(1-L)/2,n-2}\sqrt{Var_n(\varepsilon)} \tag{3.1}$$

where $t_{(1-L)/2,n-2}$ is the appropriate (two-tailed) percentage point on the measurement axis, for which the area under the t curve with $n-2$ degrees of freedom to the right of the percentage point is $(1-L)/2$.

The above PI method can be used to compute a PI to contain the 'true' value of an unknown quantity. From the method development process, we can see that this standard technique relies on the normality assumption of the underlying distribution to calculate the percentile value t. It also requires the values of the existing data be observable. Moreover, a study in (Phillips, 1979) shows that, when the model parameters, α and β, are estimated from the same data that is used to compute the point estimator of the unknown quantity, the distribution of the residual may not in general be normal. Nevertheless, Equation (3.1) forms the basis for many approaches to PI calculation.

There are also other methods that utilise this theoretical formula approach. Olive (1998) proposed a PI method based on a large sample of observations. The author claims that the normality assumption used in some other methods can be relaxed by putting appropriate weights on the model parameters (p. 11). Møller et al. (2005) used the Poisson model to compute PIs for cancer incidence rates for a future period based on past records in different countries. They estimated the variance of the residual by decomposing it into two components: the variance reflecting the uncertainty of the model and the random variation of the future number of cancer cases (for which a PI is to be calculated). They computed the PIs based on that: (a) there is no uncertainty about the Poisson model, hence assuming the adequacy of the chosen model (the 'true' model), and (b) the distribution of the future number of cancer cases is normal. They concluded that, as the sample size increases, uncertainty about the model adequacy is too substantial to be ignored, and the coverage percentages of the calculated PIs fall below the specified confidence level.

3.3 Empirically based PIs

An empirically Based PI is an alternative when theoretical formulas for certain models are not available, or the validity of the chosen model is doubtful. Methods that employ this approach rely on observing the actual distribution of the prediction residual, rather than assuming a distribution for it. Being computationally demanding is one of the disadvantages of the empirically based PI approach.

A recently proposed method (Shrestha & Solomatine, 2006) using this approach computes PIs by estimating the underlying functional relationship between the input and the prediction limits of the PIs. It estimates the variance of the prediction residual by observing the empirical distribution of historical residuals between the model outputs (expressed as functions of the model's input values) and the corresponding actually observed data. Shrestha & Solomatine believe that these residuals are the best available indicators of the discrepancy between the real-world process and the model that is assumed to represent it.

The proposed method begins with partitioning the training dataset into clusters with similar residuals or residuals with similar distributions. Then PIs for each cluster are computed based on the empirical distribution of the residuals of a particular cluster. In the case of crisp clustering, each instance in a cluster has the same PI, whereas if fuzzy clustering is used (which means that a particular instance might be clustered into two or more clusters), membership grades have to be taken into account. After the PIs for the training instances have been established, the instances are now ready to be used to train any algorithm (e.g. neural networks) to construct a mapping function relating a future input instance to its corresponding PI (p. 229).

An earlier empirical PI method proposed by Gardner (1988, p. 546) employs a 'stepwise' strategy. The method fits the model parameters to the training data in k steps and uses the Chebyshev inequality to estimate the error variances in order to calculate the PIs. Based on Gardner's method, Talor & Bunn (1999) proposes an empirically based approach, which is nonparametric and thus avoids the common normality assumption. Another development by Williams & Goodman (1971) also uses a technique similar to Gardner's method and calculates PIs by using the appropriate percentage points of the empirically found distribution, rather than relying on distributional assumptions.

3.4 PI Methods by Simulation and Resampling

Simulation and resampling (or bootstrap) methods provide an alternative to empirically based PI approach. These methods require fewer assumptions than those based on other approaches and are useful when the size of the observed data is small or when the assumption about normally distributed residuals can not be justified. However, these methods can be even more computationally demanding than empirically based ones, particularly when using resampling. This is because in practical problems, the estimation is quite complex and large number of replications is thus often required.

The bootstrap, for example, is a nonparametric resampling approach to estimating the conditional distribution of an unknown quantity, x_{n+1}, by resampling the residuals with replacement. As discussed in Section 3.2, producing PIs requires resolving the α percentile value t_α, which is typically unknown. The bootstrap method does not assume any parametric distribution. Instead, it seeks an approximation to the absolute value of t_α. In the case of calculating one-sided PIs (i.e. *prediction bounds*), the following resampling scheme was discussed in (Mojirsheibani, 1998, p. 491). Let F be the empirical distribution of the observed data $x_i, i = 1, \ldots, n$, i.e. $x_i \sim F$; let $x_i^*, i = 1, \ldots, n$ be an independent and identically-distributed (iid) random sample drawn with replacement from F, with mean $\bar{x}^* = \frac{1}{n}\sum_{i=1}^{n} x_i^*$ and standard deviation s^*; also let x_{n+1}^* be from F, independent of x_i^*. A distribution function T^* is defined as

$$T^* = \frac{x_{n+1}^* - \bar{x}^*}{s^*\sqrt{1+\frac{1}{n}}}.$$

If we use t_α^* to refer to the α percentile value of the distribution T^*, then t_α^* can be found using bootstrap resampling. Using t_α^* to approximate t_α, yields a bootstrap PI for x_{n+1}

$$\left[\bar{x} + t_\alpha^*\sqrt{s^2(1+\frac{1}{n})}\ ,\quad \infty\right)$$

Bai (1990) provides discussions of coverage error rates for such bootstrap PIs. Stine (1985) looks at nonparametric bootstrap PIs for regression and concludes that unconditional PIs, such as the ones obtained using the bootstrap, compare favourably with normal-distribution-based PIs. Thombs & Schucany (1990) compute bootstrap PIs under

the primary assumption that the chosen model does fit the observed data. Finally, Romano (1992) reviews and compared various PI methods.

3.5 Distribution-free PIs

As discussed in Section 2.3.4, some shortcomings may limit the value of distribution-free statistical intervals and thus discourage users from considering using them. However, such intervals still serve as a useful alternative in certain situations. In this section, we use an example to demonstrate the calculation of distribution-free PI in the case of discrete variables. The calculation uses the typical "order statistics" strategy for computing distribution-free PIs (Hahn & Meeker, 1991, Section 5.4, p. 76; Guttman, 1970, pp. 7–8).

The general setup is as follows: let a sample of n independent observations, $x_i, i = 1, \ldots, n$, be taken from a population of an unknown distribution. Suppose a new but unknown observation, x_{n+1}, is also randomly selected from the same population, independent of the previous x_i. We wish to determine an interval, (P_L, P_U), such that x_{n+1} falls in the interval with a specified confidence level L. Arrange the sample observations x_i according to their values, denoting the ordered x_i by x'_i, where $x'_1 < x'_2 < \ldots < x'_n$. Since by assumption $x_1, \ldots, x_n, x_{n+1}$ are invariant under all permutations, the $n + 1^{th}$ observation is equally likely to occupy any of the 'bins' formed by the order statistics of $x'_1 < x'_2 < \ldots < x'_n$, including the two leftmost and the rightmost positions if x_{n+1} happens to be the smallest or the largest. Because there are in total $n + 1$ gaps formed by the n available observations, as illustrated in Figure 3.1, each bin carries a probability of $1/(n+1)$. It then follows that the probability that the $(n+1)^{th}$ observation falls inside the entire range of n observations (excluding the two leftmost and rightmost positions: position 1 and $n + 1$ in the figure) is $1 - \frac{1}{n+1} \times 2$. That is,

$$P(P_L < x_{n+1} < P_U) = 1 - \frac{1}{n+1} \times 2 = \frac{n-1}{n+1}$$

where $P_L = x'_1$ and $P_U = x'_n$. The probability that the interval (P_L, P_U) captures the $(n+1)^{th}$ observation is just the specified confidence level L, i.e.

$$P(P_L < x_{n+1} < P_U) = \frac{n-1}{n+1} = L$$

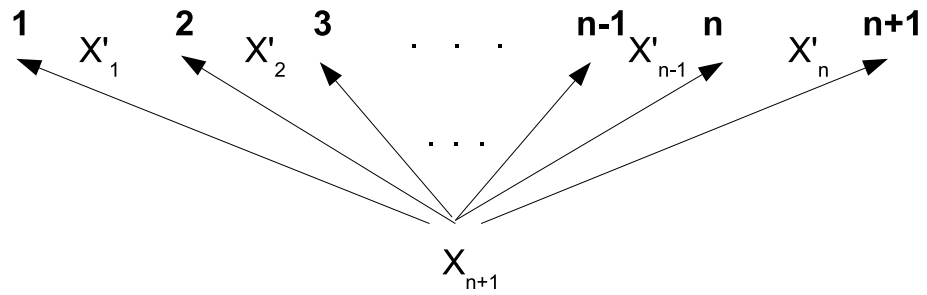

Figure 3.1: The new observation x_{n+1} could occupy any of the $n+1$ positions formed by the n ordered observations.

Straightforward algebra gives the sample size requirement

$$n = \frac{1+L}{1-L}.$$

Therefore, to calculate, for example, a 95% PI, we need to sample at least $n = \frac{1+0.95}{1-0.95} = 39$ observations and find the values of the largest and smallest observations as the interval endpoints (alternative bootstrap-alike techniques may be needed otherwise).

In the case of continuous variables, Saw, Yang & Mo (1984) computed the following distribution-free PI for sample data x_i:

$$\bar{x} \pm \lambda(1 + 1/n)^{1/2} s \tag{3.2}$$

where $\lambda \geq 1$, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean, and $s = [\sum_{i-1}^{n}(x_i - \bar{x})^2/(n-1)]^{1/2}$ is the usual unbiased estimator of the standard deviation. Konijn (1987) elaborated some useful properties of the PI in (3.2) with regard to the parameter λ.

There are also other PI methods based on the distribution-free approach. Butler & Rothman (1980) proposed a method based on a cross-validation or sample reuse methodology that makes use of a "one-at-a-time schema of observational omissions." Aiming to get precise network state information, Yin, Chaw & Yaacob (2005) adopt the nonparametric distribution-free approach to construct two-sided PIs in helping to infer the future available bandwidth and generate quality of service (QoS) metrics.

3.6 PIs based on the Bayesian Approach

The PIs we have discussed up to this point are computed and evaluated from the so-called 'classical' ('frequentist' or 'sampling theory') point of view. Bayesian methods provide a useful alternative that allows the analyst to combine his or her prior beliefs about the quantity of interest with information contained in the observed sample, and then make statistical inferences about the quantity by summarising the computed posterior distribution. In this section, we review some of the methods that use this approach to compute PIs.

Chhikara & Guttman (1982, pp. 321–322) constructed a two-sided PI for the inverse Gaussian distribution. In order to primarily rely on the likelihood based on the observed data, they used the following "diffuse" prior:

$$p(\theta, \lambda) = p(\theta \mid \lambda)p(\lambda) \propto \frac{1}{\lambda} \tag{3.3}$$

where $\theta = 1/\mu$; and μ and λ are the two parameters of the inverse Gaussian distribution. The parameterisations $I(\frac{1}{\mu}, \lambda)$, instead of the usual $I(\mu, \lambda)$, for the inverse Gaussian distribution is also employed to avoid improper prior and posterior distributions for the parameters. A closed form of the predictive distribution, $h(x_{n+1} \mid X)$, of a single future observation x_{n+1}, given the data X, is then found analytically. The determination of the PI limit points, however, still requires numerical integration from computer power. A PI was also constructed by using the 'frequentist' approach in the same paper. A comparison of the two approaches leads to a conclusion that Bayesian inference has a "definite advantage" over the other (p. 323).

A method of computing PIs for the unimodal log-normal distribution, $Y = \log X \sim \mathcal{N}(\mu, \sigma^2)$, was presented in (Dahiya, 1982, p. 279). They used the same "diffuse" prior as in (3.3) for the normal parameter σ^2, and also found a similar form of the predictive density function. They derived a method that can be used to compute PIs with the shortest width. The method was illustrated by using the log-normal distribution to model the "time to failure" models (p. 290).

Jaheen (2003) constructed one-sided prediction bounds for the sth future observation. The proposed method uses "a finite mixture of two-component Gompertz model" to represent the underlying population, from which the observations were sampled. By using Monte Carlo simulations, the proposed method computes the lower and upper prediction

bounds for the minimum and maximum of future observations with a coverage percentage "close to the specified confidence level."

Faulkenberry (1973) also proposed a Bayesian procedure for computing PIs. The method obtains a distribution for an unknown observation for which the PI is to be computed. The obtained model is "conditional on a sufficient statistic" for the model parameter. The observed sample data and the unknown observation are assumed to have different distributional functions indexed by the same parameter, and the joint distribution is derived and used to compute the PI. Two examples of using the procedure were demonstrated, using the negative exponential and the Poisson distribution, respectively.

Other probability distributions, for which consideration of PI computation has been given using the Bayesian approach, include the exponential distribution for different sampling schemes with regard to the observed data, and a distribution whose pdf itself is randomly selected according to a Dirichlet process on the space of distribution functions (Campbell & Hollander, 1982).

3.7 Summary

In this chapter, we have reviewed some previously proposed approaches to computing PIs. The most widely used approach is the theoretical PI formula with the normal distribution assumed for the underlying process. When the probability assumption is not met, or when the theoretical formulas are not available for some complex or nonlinear models, approaches that require fewer distributional assumptions can be used. Distribution-free methods can be useful in certain situations. Empirically based and resampling approaches are normally computationally demanding. Yet modern computer technologies have made it possible to solve problems that could not be solved before. The Bayesian approach has many practical advantages such as easy accommodation of unobservable variables and incorporating information from previous studies through prior distributions, etc. In the next chapter, we will derive a Bayesian model to compute PIs for class probability predictions made by models such as decision trees, i.e. the particular machine learning context considered in this thesis.

Chapter 4
Methodology

4.1 Introduction

As we discussed in the last two chapters, inevitable uncertainties come with point estimates. Interval estimates are known to be able to provide much more information than point estimates, and have been widely used to quantify the uncertainties around point predictions. A prediction interval (PI) is computed to contain, with a specified confidence level, a single unknown quantity, and the width of the PI is used to indicate the uncertainty.

Various PI methods have been proposed with the aim of solving a particular problem in a particular situation. However, the construction of PIs for the class probability of an unknown instance has not been attempted in previous research. In this chapter, a Bayesian approach is employed to derive a theoretical formula that can be used to compute a PI for the class probability of a test instance based on the outcome of a decision tree classifier, or similar techniques that provide a neighbourhood for a test instance.

The chapter is organised as follows. We discuss the proposed Bayesian probabilistic model from Section 4.2 to 4.5, including the setup of the framework, the modelling of the class probability and the computation of the model. A decision tree ensemble algorithm is introduced in Section 4.6. The chapter is summarised in Section 5.3.

4.2 General Setup

In the cancer diagnosis example discussed in Chapter 1, the task of building a diagnostic system is to establish a relationship between a set of symptoms of a patient and a corresponding diagnosis. Let x be a patient; and y be the diagnosis for the patient, which we shall define as taking the value 1 if a patient has developed cancer and 0 otherwise. In the machine learning context, the patients, their symptoms and the diagnoses are

generalised as instances (the xs), the attributes, and the classes (the ys) of the instances, respectively. Since the class y only takes on value 0 and 1, we have a binary classification problem.

Suppose an ensemble of random decision tree classifiers (discussed in Section 4.6) is trained using a set of training instances, and the ensemble defines a group of n instances $(x_1, y_1), \ldots, (x_n, y_n)$ as the neighbourhood of an unknown instance (x_{n+1}, y_{n+1}). The classification for the unknown instance can be obtained by calculating the majority class in the observations $y_1, \ldots, y_n$. We can also calculate k/n, where k is the number of 1s in $y_1, \ldots, y_n$, to express an estimate of the true probability that the unknown instance belongs to the class 1, i.e. $P(y_{n+1} = 1) = k/n$.

The aim of the study, however, is to compute a PI to include the true class probability of $P(y_{n+1} = 1)$, with a specified confidence level L, given only the class observations $D = (y_1, \ldots, y_n)$. We assume the computed PI is a joint interval, which can be expressed as $[P_L,\ P_U]$ with the lower and upper limits P_L and P_U, i.e. $P_L < P(y_{n+1} = 1) < P_U$. (We will discuss the more complicated disjoint PIs later in the chapter.) For simplicity purposes, we shall hereafter use P_{n+1} to implicitly mean $P(y_{n+1} = 1)$.

One of the often praised advantages of Bayesian inference is that it enables us to find a complete distribution of an unknown quantity of interest. Thus, we can compute a PI for P_{n+1} by seeking the probability distribution of P_{n+1} conditional on the observed data D, i.e. $Pr(P_{n+1} \mid D)$. The distribution $Pr(P_{n+1} \mid D)$ defines the relationship between the probability value P_{n+1} and the observed class set D.

Deriving a distribution of the quantity of interest is just the first necessary step towards constructing a PI. To compute PIs of various kinds, such as a PI with the narrowest width, different methods are required. With the yet to be found, applicable density function $Pr(P_{n+1} \mid D)$, they are all attainable. We will focus on deriving the form of $Pr(P_{n+1} \mid D)$ in the next two sections, and discuss the computation of the narrowest PIs in Section 4.5.

4.3 Parametric Modelling Framework

The density function $Pr(P_{n+1} \mid D)$ is the probability distribution of P_{n+1}, conditional on $D(y_1, \ldots, y_n)$, about which the theory of conditional probability tells us that

$$Pr(P_{n+1} \mid D) = \frac{Pr(P_{n+1} \cap D)}{Pr(D)} = \frac{Pr(P_{n+1}, y_1, \ldots, y_n)}{Pr(y_1, \ldots, y_n)} \tag{4.1}$$

whenever the mass function in the denominator is non-zero. In fact, with fixed $(y_1, \ldots, y_n)$, the factor $Pr(D)$ does not depend on P_{n+1} and thus is considered a constant, yielding an equivalent form of (4.1), with $Pr(D)$ omitted, as follows:

$$Pr(P_{n+1} \mid D) \propto Pr(P_{n+1}, y_1, \ldots, y_n) \tag{4.2}$$

The form on the right-hand side of (4.2) is called the *unnormalised posterior density*, because of the omission of the denominator $Pr(D)$.

We need a model for the distribution of P_{n+1}. So we introduce a parameter vector θ that lies in the parameter space Θ, i.e. $\theta \in \Theta$, such that,

$$Pr(P_{n+1} \mid D) \propto Pr(P_{n+1}, y_1, \ldots, y_n) = \int_{\Theta} Pr(P_{n+1}, \mathbf{y} \mid \theta) Pr(\theta) d\theta \tag{4.3}$$

where $\mathbf{y} = (y_1, \ldots, y_n)$.

Now let us look at the composition of the data space D. From the discussion about the cancer patient example in earlier chapters, we know that the probability of the $(n+1)^{th}$ patient having developed cancer depends on not only the class set $y_1, \ldots, y_n$, but also $P_1, \ldots, P_n$, which are the probability values of the corresponding patients with known diagnoses. Hence, we define $D \in (y_1, \ldots, y_n, P_1, \ldots, P_n)$.

Since $P_1, \ldots, P_n$ are unobservable, we take them into account by integrating over all possible values of $P_1, \ldots, P_n$ in the range (0, 1). Thus, by substituting Equation 4.3 into the right-hand side of (4.2), together with $D \in (y_1, \ldots, y_n, P_1, \ldots, P_n)$, the distribution $Pr(P_{n+1} \mid D)$ becomes

$$\begin{aligned} Pr(P_{n+1} \mid D) &\propto \int_{\theta} \int_{P=0}^{1} Pr(P_{n+1}, \mathbf{y}, \mathbf{P} \mid \theta) Pr(\theta) d\mathbf{P} d\theta \\ &\propto \int_{\theta} \int_{P=0}^{1} Pr(P_{n+1} \mid \mathbf{y}, \mathbf{P}, \theta) Pr(\mathbf{y}, \mathbf{P} \mid \theta) Pr(\theta) d\mathbf{P} d\theta \end{aligned} \tag{4.4}$$

where $\mathbf{y} = (y_1, \ldots, y_n)$, and $\mathbf{P} = (P_1, \ldots, P_n)$.

The term $Pr(P_{n+1} \mid \mathbf{y}, \mathbf{P}, \theta)$ in (4.4) can be further simplified to

$$Pr(P_{n+1} \mid \mathbf{y}, \mathbf{P}, \theta) = Pr(P_{n+1} \mid \theta), \tag{4.5}$$

based on the fact that P_{n+1} is independent of $\mathbf{y}$ and $\mathbf{P}$, given the model parameter θ (Gelman, 2004, p. 9).

From the definition of conditional probability, we can also write

$$Pr(\mathbf{y}, \mathbf{P} \mid \theta) Pr(\theta) = Pr(\mathbf{y}, \mathbf{P}, \theta) \tag{4.6}$$

Substituting (4.5) and (4.6) into (4.4) results in

$$\begin{aligned} Pr(P_{n+1} \mid D) &\propto \int_\theta \int_{P=0}^{1} Pr(P_{n+1} \mid \theta) Pr(\mathbf{y}, \mathbf{P}, \theta) d\mathbf{P} d\theta \\ &\propto \int_\theta \int_{P=0}^{1} Pr(P_{n+1} \mid \theta) Pr(\mathbf{y} \mid \mathbf{P}) Pr(\mathbf{P} \mid \theta) Pr(\theta) d\mathbf{P} d\theta \\ &\propto \int_\theta Pr(P_{n+1} \mid \theta) \prod_{i=1}^{n} \left[\int_0^1 Pr(y_i \mid P_i) Pr(P_i \mid \theta) dP_i \right] Pr(\theta) d\theta \end{aligned} \tag{4.7}$$

Here, we have used that $Pr(\mathbf{y} \mid \mathbf{P}, \theta) = Pr(\mathbf{y}, \mathbf{P})$. The last equation follows from the fact that $y_1, \ldots, y_n$ and $P_1, \ldots, P_n$ are exchangeable. We continue the computation of the right-hand side of (4.7) in the next section.

4.4 Modelling Class Probabilities

In this section, we specify a distributional model for the class probabilities. We also discuss the parameter vector θ that is associated with the specified probability distribution.

Ideally, modelling the class probabilities of the instances involves making a specific distributional assumption about $P_1, \ldots, P_n$, based on our prior knowledge of their likely values. Unfortunately, because the values of $P_1, \ldots, P_n$ are unobservable, we do not have the necessary prior information about their empirical distribution. All we know about the values of $P_1, \ldots, P_n$ is that they are in the range [0, 1]. Therefore, to fulfill the principle of maintaining maximum uncertainty in the chosen class of models, as discussed in Chapter

2, we choose the beta distribution to model the distribution of $P_1, \ldots, P_n$. We make our choice based on the following grounds:

(1) The flexibility of the beta distribution, which is represented by a wide variety of density curves, reflects our lack of knowledge about the unobservable probability values.

(2) We can use the beta distribution to approximate any smooth unimodal distribution in the range [0, 1]. Figure 4.1 shows some examples of the beta density graph in the range [0, 1], corresponding to different combinations of its α and β parameter values between 0 and 3 (Lee, 2004, pp. 74–75; Gregory, 2005, p. 117).

(3) The beta distribution is commonly used in Bayesian analysis to describe initial knowledge concerning probability of success or failure, i.e. class 1 or 0 (Hahn & Shapiro, 1994, p. 94).

A beta distribution that is defined in the range [0, 1] is called the standard beta distribution. A random variable X, which follows a standard beta distribution with the two parameters α and β (both positive), has the following probability density function:

$$f(x;\alpha,\beta) = \begin{cases} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} & 0 \le x \le 1, \quad 0 < \alpha, \quad 0 < \beta \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

In Figure 4.1, we can see that different combinations of α and β result in different shapes of the beta distribution. For example, when $\alpha > 1$ and $\beta > 1$, the distribution is single peaked; when $\alpha < 1$ and $\beta < 1$, it is U shaped; with $\alpha < 1$ and $\beta \geq 1$, it is reverse J shaped, and so on. Thus, both parameters affect the distribution shape, i.e., they are both shape parameters. Now we can define the parameter space θ in (4.7) to be $\theta \in (\alpha, \beta)$.

To avoid carrying the whole heavy notation, we for now only deal with the part

$$\int_0^1 Pr(y_i \mid P_i) Pr(P_i \mid \theta) dP_i \tag{4.9}$$

in (4.7). Putting P_i in place of x in Equation (4.8) yields

$$Pr(P_i \mid \theta) = Pr(P_i \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)} P_i^{\alpha-1}(1-P_i)^{\beta-1} \tag{4.10}$$

A property that the class observations $y_1, \ldots, y_n$ have is that, each y_i takes on value 1 with the corresponding probability P_i, or 0 with probability $1 - P_i$. That is, y_i are Bernoulli random variables. Thus, the class observations y_i have the probability mass

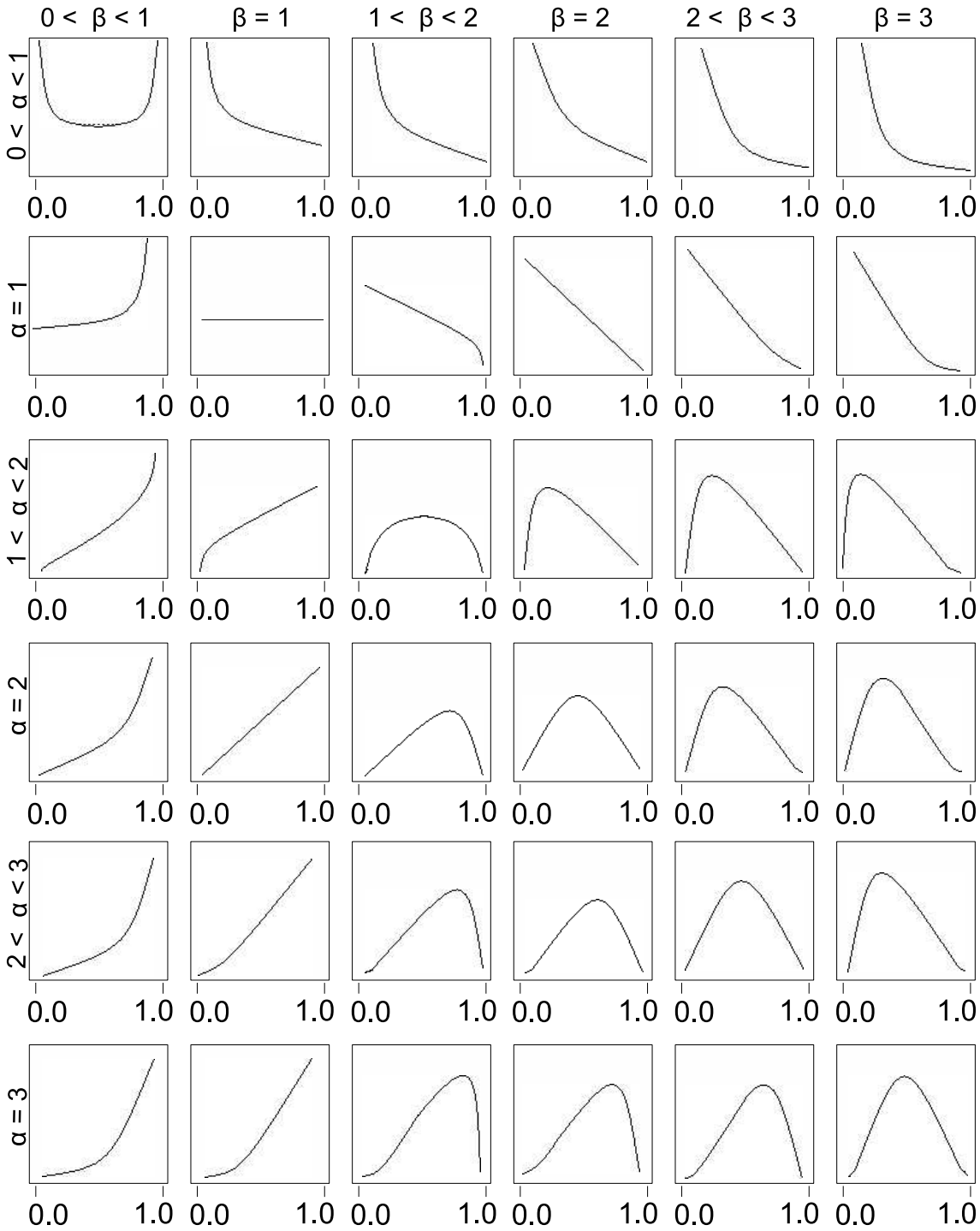

Figure 4.1: Examples of beta densities.

function:

$$Pr(y_i) = \begin{cases} P_i & y_i = 1 \\ 1 - P_i & y_i = 0 \end{cases}$$

Therefore,

$$Pr(y_i \mid P_i) = y_i \cdot P_i + (1 - y_i) \cdot (1 - P_i) \tag{4.11}$$

Substituting Equation (4.10) and (4.11) in (4.9) yields

$$\begin{aligned} \int_0^1 Pr(y_i \mid P_i) Pr(P_i \mid \theta) dP_i &= \int_0^1 [y_i P_i + (1 - y_i)(1 - P_i)] \\ &\quad \cdot \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \cdot \Gamma(\beta)} P_i^{\alpha - 1} (1 - P_i)^{\beta - 1} dP_i \end{aligned} \tag{4.12}$$

A solution to this integral was obtained using the Mathematica software package:

$$\frac{\alpha y_i \ \cdot {}_2F_1(\alpha + 1, 1 - \beta; \alpha + 2; 1) - (\alpha + 1)(y_i - 1) \cdot \ {}_2F_1(\alpha, -\beta; \alpha + 1; 1)}{\alpha(\alpha + 1) B(\alpha, \beta)} \tag{4.13}$$

where $B(\cdot)$ is the beta function, which can be decomposed using the gamma function $\Gamma(\cdot)$ as $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$, and the function ${}_2F_1(\cdot)$ has the generalised form ${}_pF_q(a_1, \ldots, a_p; b_1, \ldots, b_q; x)$, that, with the corresponding parameters $p = 2$, $q = 1$, is called the first hypergeometric function (also known as the Gauss's hypergeometric function).

The reasoning that is needed to simplify (4.13) is quite involved, yet with Kummer's (1836) solution to the first hypergeometric function (also called Kummer's first formula), it is achievable. We omit the lengthy manipulation here and give the details in Appendix A. The following is the final form of the result:

$$\frac{y_i \alpha + (1 - y_i)\beta}{\alpha + \beta} \tag{4.14}$$

Substituting (4.14) back into (4.7) yields

$$\begin{aligned} Pr(P_{n+1} \mid D) &\propto \int Pr(P_{n+1} \mid \theta) \prod_{i=1}^{n} \left[\frac{y_i \alpha + (1 - y_i)\beta}{\alpha + \beta} \right] Pr(\theta) d\theta \\ &\propto \int Pr(P_{n+1} \mid \theta) \frac{\prod_{i=1}^{n} [y_i \alpha + (1 - y_i)\beta]}{(\alpha + \beta)^n} Pr(\theta) d\theta \end{aligned} \tag{4.15}$$

To simplify the product $\prod_{i=1}^{n} [y_i \alpha + (1 - y_i)\beta]$ in (4.15), let k be the number of observations in $y_1, \ldots, y_n$ that take on value 1, and hence the remaining $n - k$ observations all

take on value 0. Since

$$y_i\alpha + (1 - y_i)\beta = \begin{cases} \alpha & y_i = 1 \\ \beta & y_i = 0 \end{cases}$$

the above product can be written as

$$\prod_{i=1}^{n}[y_i\alpha + (1 - y_i)\beta] = \alpha^k\beta^{n-k} \tag{4.16}$$

Again, by the earlier modelling assumption that P_{n+1} is beta distributed with parameters α and β, Equation (4.15) becomes

$$\begin{aligned} Pr(P_{n+1} \mid D) &\propto \int Pr(P_{n+1} \mid \theta)\frac{\alpha^k\beta^{n-k}}{(\alpha+\beta)^n}Pr(\theta)d\theta \\ &\propto \int\int \frac{(P_{n+1})^{\alpha-1}\cdot(1-P_{n+1})^{\beta-1}}{B(\alpha,\beta)}\cdot\frac{\alpha^k\beta^{n-k}}{(\alpha+\beta)^n}Pr(\alpha,\beta)d\alpha d\beta \end{aligned} \tag{4.17}$$

where $0 \leq P_{n+1} \leq 1$, $0 < \alpha < \infty$, $0 < \beta < \infty$, and $Pr(\alpha, \beta)$ is the prior distribution of the parameters and will be discussed in the next section.

4.5 PI Computation by Model Simulation

Now we have established a probability model from which we can make various posterior inferences about P_{n+1}. An example of such inferences is to report the entire posterior distribution, i.e. a graphical display, as we will see later in the section. Other numerical summaries of the distribution, which are more of practical use, include reports on location, such as the mean, median, and mode of the distribution; and the standard deviation and other percentiles to summarise the variation of the distribution. The desired inference in this study is an interval summary (a PI) to report the posterior uncertainty of an unknown quantity.

To compute a PI for P_{n+1}, which is a continuous random variable in the range [0, 1], we begin with the definition of probability density function. If X is a continuous random variable, then the probability density function of X is a function $f(x)$ such that any two numbers, a and b, with $a \leq b$,

$$p(a \leq x \leq b) = \int_a^b f(x)dx \tag{4.18}$$

The left-hand side of Equation (4.18) is the probability that x takes on a value in the interval [a, b], which is also the area under the density $f(x)$. Substituting [a, b] with

$[P_L,\ P_U]$ and $f(x)$ with $Pr(P_{n+1} \mid D)$ in (4.18), we have

$$p(P_L \leq P_{n+1} \leq P_U) = \int_{P_L}^{P_U} Pr(P_{n+1} \mid D) dP_{n+1} \tag{4.19}$$

From the definition of confidence level, we know that if the integral $[P_L,\ P_U]$ claims to contain the quantity P_{n+1} with confidence level L, then L is just the probability that P_{n+1} lies in $[P_L,\ P_U]$. That is,

$$P[P_L < P(y_{n+1} = 1) < P_U] = L \tag{4.20}$$

Combining Equation (4.20) with (4.19) yields

$$\int_{P_L}^{P_U} Pr(P_{n+1} \mid D) dP_{n+1} = L \tag{4.21}$$

That is, to compute a PI for P_{n+1} at the confidence level L, we need to find the two interval endpoints P_L and P_U such that the area under the graph of the density function $Pr(P_{n+1} \mid D)$ between the two percentiles P_L and P_U is L (e.g., 95% or an equivalent value 0.95).

In general, a PI (i.e. $[P_L,\ P_U]$) calculated using Equation (4.21) is not unique unless further conditions are imposed. However, when computing a PI of coverage level L, which is the so-called *central interval* of the distribution, the two endpoints can be uniquely identified in the following calculation. Since $0 \leq P_{n+1} \leq 1$, from the definition of probability density function, we have

$$p(P_{n+1} < P_L) + p(P_L < P_{n+1} < P_U) + p(P_{n+1} > P_U) = 1$$

Also, because it is a central coverage interval that we are computing, it follows that

$$\begin{aligned} p(P_{n+1} < P_L) &= p(P_{n+1} > P_U) \\ &= \frac{1 - p(P_L < P_{n+1} < P_U)}{2} \\ &= \frac{1 - L}{2} \end{aligned}$$

Then the two endpoints of the PI, P_L and P_U, are given by

$$\int_{0}^{P_L} Pr(P_{n+1} \mid D) dP_{n+1} = \frac{1 - L}{2} \tag{4.22}$$

and

$$\int_0^{P_U} Pr(P_{n+1} \mid D) dP_{n+1} = \frac{1+L}{2} \tag{4.23}$$

An example of central coverage interval is shown in Figure 4.2 (a). PIs with other requirements such as the narrowest PIs can also be computed by demanding different strategies. These will be discussed later in the section.

For the right-hand side of (4.17) to be used in the calculations of P_L and P_U, such as the above discussion, it must be a proper density. Recall that the density function in (4.17) is called the unnormalised posterior density, because the denominator was omitted in the beginning of the reasoning (Equation 4.2). To get a proper density, normalisation is required. This can be achieved by dividing the value of each point on the unnormalised density curve by the total area under the curve.

To calculate the area under a density curve within a certain range, we have to compute the value of the integral on the right-hand side of (4.17). Hence, we need to decide on the prior distribution $Pr(\alpha, \beta)$ of α and β. We choose the gamma distribution with both of its shape and scale parameter equal to one. This assumption is an attempt at noninformativeness to reflect our lack of knowledge about the two beta parameters. We also choose a Monte Carlo approach to compute the integral. This is discussed in more detail further below. For the moment, let us assume that we can compute the density curve for P_{n+1} by evaluating the integral over different values of P_{n+1}.

The next problem is the computation of the area under the density curve. We employ a simple approximation technique for this. The idea is to 'discretise' the X-axis in the interval [0, 1] into 10,000 units. Thus, the total area under the density curve is uniformly divided into 10,000 'rectangles' with all of them having a tiny 'curvy' top and the same unit width. Given the small range of [0, 1], the size 10,000 for the discretisation can make the calculation of the area quite accurate. Now the integration of the density curve can be approximated by the summation of the areas of those 'flat' top rectangles with each of them calculated by the unit width (i.e. 1/10,000=0.0001) times the corresponding density values. Note that the height of a particular rectangle is approximated by the density value of a point on the curve that corresponds to the rectangle [see Figure 4.2 (b)].

One thing worth pointing out here is that, when computing the area under the density

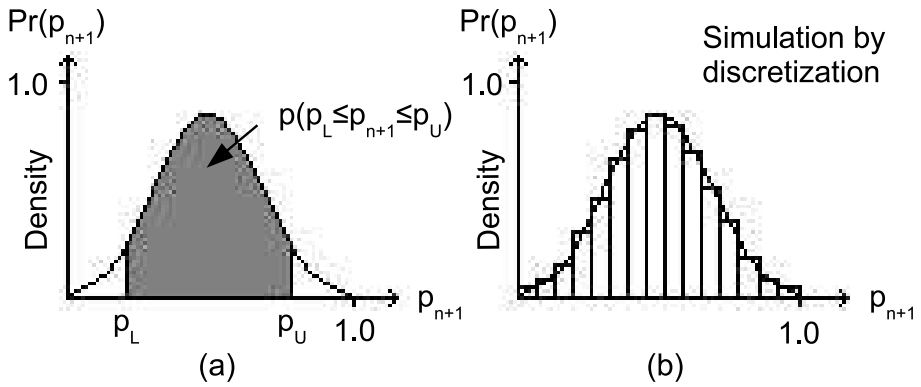

Figure 4.2: (a) p($P_L \leq P_{n+1} \leq P_U$) = the area under the density curve between P_L and P_U. (b) Smooth curve area is simulated by summation of a large number of rectangles.

curve, the two endpoints 0 and 1 (or P_L and P_U) are excluded to prevent a division by zero. This can be justified because

(a) the probability assigned to any particular single value is zero, and

(b) the probability of an interval does not depend on whether either of its endpoints is included.

These properties follow from the fact that the total area that corresponds to a particular single value is zero (the area can be thought of as covering only a vertical line with a zero width), and that the area under the curve above an interval is unaffected by exclusion or inclusion of the endpoints of that interval [see Figure 4.2 (a)].

Now we compute the density curve for P_{n+1}, using (4.17). We achieve this using a Monte Carlo approach based on the gamma prior for the parameters α and β. We proceed as follows:

(1) Generate values of the beta parameters α and β by sampling from the gamma distribution with its two parameters both equal to 1. We shall write this gamma distribution as gamma(1, 1).

(2) For each pair of the generated α and β values, we take each of the 'discretised' values of P_{n+1} in the interval (0,1) with the endpoints excluded, and compute the value of

$$\frac{(P_{n+1})^{\alpha-1} \cdot (1-P_{n+1})^{\beta-1}}{B(\alpha,\beta)} \cdot \frac{\alpha^k \beta^{n-k}}{(\alpha+\beta)^n}$$

(3) For each value of P_{n+1}, we compute 1000 such density values and sum them up. The accumulated density is then normalised, which gives us the simulated posterior distri-

bution (for the adequacy of using the sampling size 1000, see Gelman, 2004, p. 25–26).

To understand how the accumulation of the sampled densities works, consider the following proposition:

Proposition 1. *When the condition*

$$\frac{\alpha}{\alpha+\beta} = \frac{k}{n} \tag{4.24}$$

is met, the likelihood function $\frac{\alpha^k \beta^{n-k}}{(\alpha+\beta)^n}$ *in (4.17) reaches its maximum.*

Thus, the α and β pairs that best 'fit' the observations (represented by the values k and n) dominate the result of the accumulation, so as to present the most influence on the posterior distribution.

Besides using the prior gamma(1, 1) for sampling the two beta parameters, another choice of relatively noninformative prior is to use the uniform distribution. For example, one prior that uses the uniform distribution is $\alpha \sim$ U(0, 10) and $\beta \sim$ U(0, 10) [we use the notation that U(a, b) denotes the uniform distribution in the range (a, b) exclusive]. The third prior that we have tested in the experiments is one that assumes $\alpha + \beta = 10$, where α and β are generated by first sampling $\alpha \sim$ U(0, 10), then subtracting α from 10 to get β. Note that the chosen number 10 could be replaced by any other number of moderate size.

The difference among the above mentioned priors is that they have different sizes of sampling space, from which the parameters α and β are sampled. This is illustrated in Figure 4.3. The figure shows that the sampling space for the priors in (a), (b), and (c) increases as the priors become more and more nonspecific. The trend is also reflected in the widths of the computed PIs using these three priors (as shown in the results of the experiments in the next chapter). A likely explanation for this is that the smaller the sampling space, the less uncertainty we introduce into the model. However, this does not mean that we can use priors as specific as we want, because the more specific the prior, the more optimistic our interval estimates become.

Note that, the computation using the prior (a) in Figure 4.3 is slightly different from the other two, in the way that how α and β are sampled in our experiments. That is, the

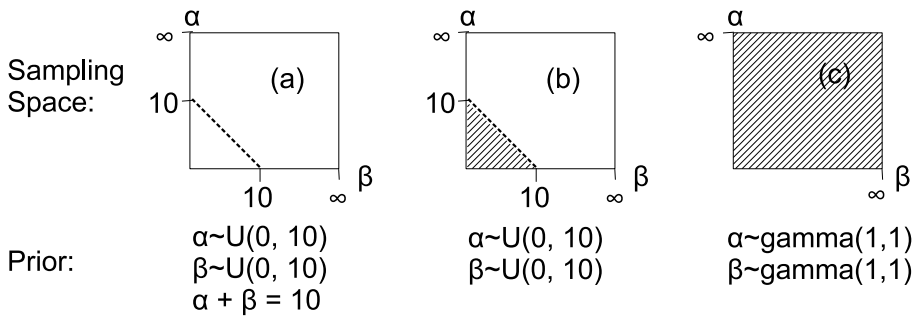

Figure 4.3: Different priors come with different sampling spaces in relation to the computation of the posterior distribution. (a) The straight line forms the sampling space for the prior $\alpha + \beta = 10$. (b) The triangular area under the line $\alpha + \beta = 10$ forms the sampling space for the prior U(0, 10). (c) The whole square area forms the sampling space for the prior gamma(1, 1).

1000 fixed points are uniformly selected on the line $\alpha + \beta = 10$. Thus, the 1000 pairs of α and β values used in the posterior computation are defined as (0.01, 9.99), (0.02, 9.98), $\cdots$, (9.99, 0.01).

The PI that we computed in equations (4.22) and (4.23) is the central coverage PI with a joint range of values. In many situations such as diagnosing life threatening diseases, however, of more practical use are PIs that have the narrowest width given a certain confidence level. The usefulness of the narrowest PIs is that the unknown quantity is confined in the smallest possible range. Thus, the most definite quantification of uncertainty is provided.

Note that, when the resulting probability model is not single peaked, a central coverage PI may not have the narrowest width, given a specified confidence level. For example, for the highly conjectural bimodal posterior density graph pictured in Figure 4.4 (a) and (b), a central coverage joint PI with cuts from both ends in (a) is wider than its counterpart in (b) consisting of two disjoint intervals, with both of them computed at the same confidence level.

A central coverage PI that does not have the narrowest width also occurs when the density curve is highly skewed. For example, in Figure 4.4 (c) and (d), the central PI in (c) is clearly wider than that in (d), in which a large part of the width has been cut off while still maintaining the same size of grey area as that in (c). Obviously, for a central joint PI to also have the narrowest width, the posterior distribution has to be single peaked and symmetric.

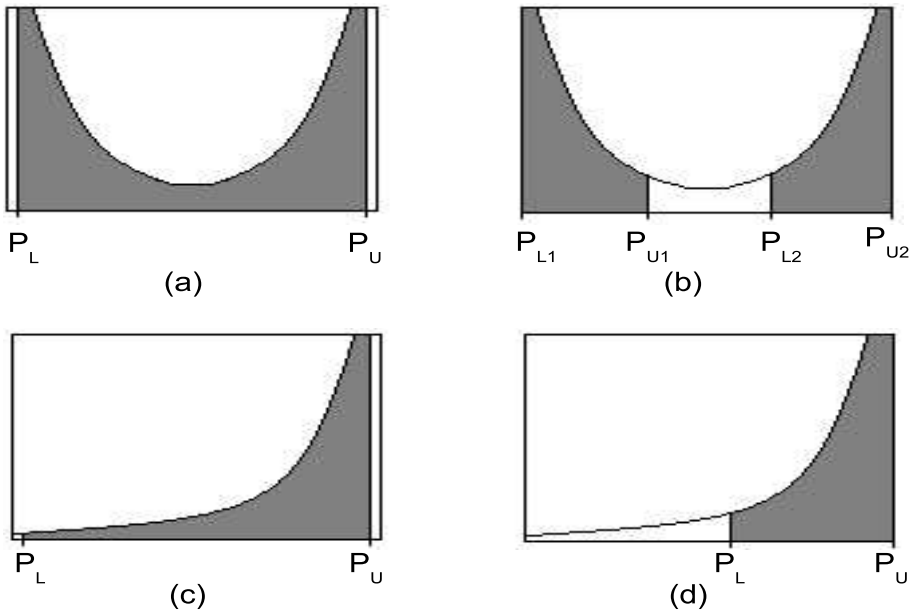

Figure 4.4: A U-shaped symmetric posterior density graph for which the 95% PI is computed in two different ways: (a) a central joint PI; (b) a disjoint PI with the narrowest width. A skewed J-shaped posterior density curve, for which the calculated 95% PI is computed: (c) a central PI; (d) the narrowest PI becomes an one-sided interval bound.

From Figure 4.4, it is also to be noticed that not only do the gray areas in (b) and (d) contain 95% of the posterior probability, they also have the characteristic that the density within the area is never lower than that outside. This is the characteristic that differentiates the types of PIs fundamentally. In Bayesian inference, the type of PIs in (b) and (d), which has the shortest width for a given confidence level, is also referred to as *highest posterior density region* (HPD).

Computing a central PI with a confidence level L by using the method in Equation (4.22) and (4.23) is equivalent to cutting off the area under the curve from both ends until the specified criterion (i.e. $\frac{1-L}{2}$) is reached. The method of calculating a PI with the narrowest width, however, is to gradually read off the 'rectangles' one by one in the order from the shortest to the highest, until the remaining area is L, resulting in an interval corresponding to the region with the highest density.

Different types (joint/disjoint) of PIs demand different interpretations in various situations. For instance, for a disjoint PI such as the one shown in Figure 4.4 (b) that is computed for the cancer patient example, it means that the probability of the patient who has developed cancer is either low (in the region $[P_{L1}, P_{U1}]$), or high (in the region

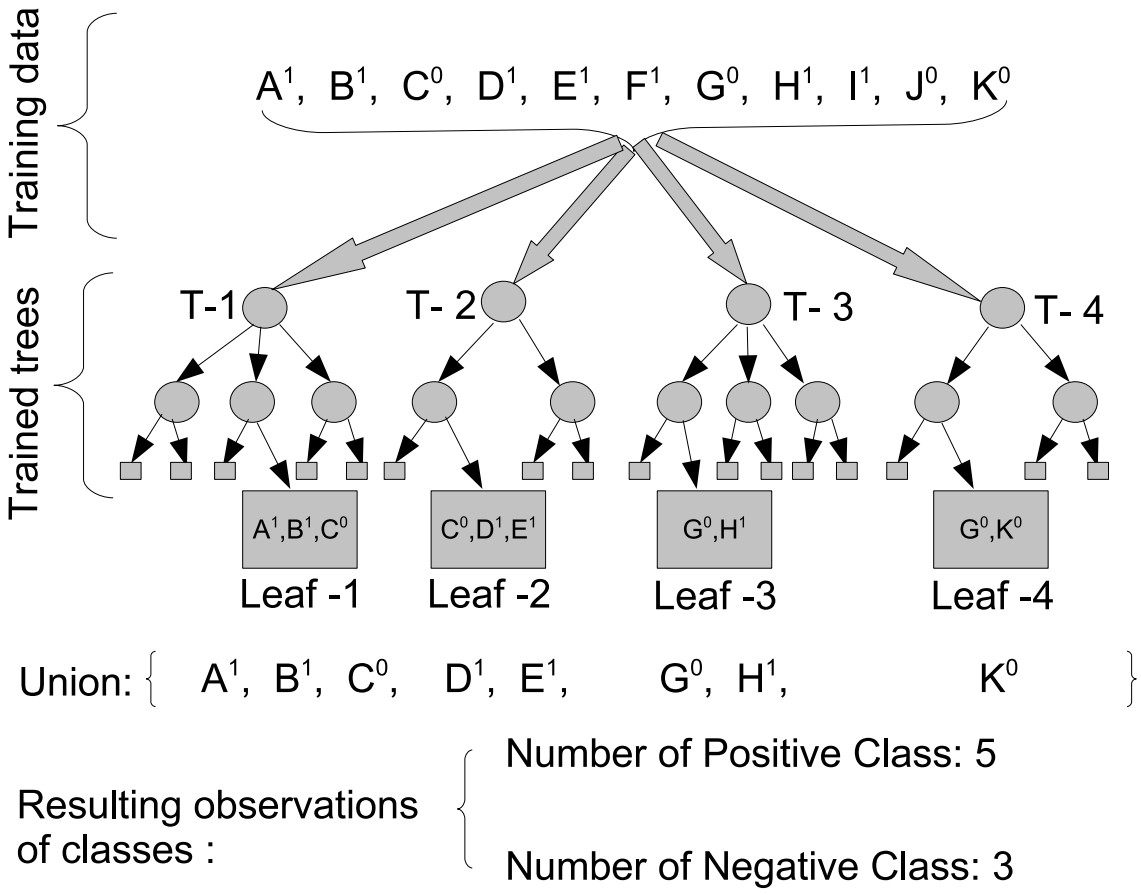

Figure 4.5: Illustration of the random tree ensemble inducting and prediction process. The letters A to K represent 10 training instances in the training dataset. The superscript denotes the corresponding instance's class. The induced ensemble consists of four single trees, in which splitting nodes are represented by circles and leaves by rectangles. In this example, Leaf-1 to Leaf-4 are used to form the neighbourhood for a test instance.

$[P_{L2}, P_{U2}]$). Whereas the PI in Figure 4.4 (a) tells us what the highest and lowest probabilities are (the two endpoints of the interval), and yet contains more uncertainty with a wider width. An interpretation that is slightly more deterministic can also be given when a PI such as one in Figure 4.4 (d) is calculated, in which case the patient definitely has a high probability of being diagnosed with cancer.

4.6 A Random Tree Ensemble Classifier

PI computation methods, such as the Bayesian model discussed above, are used in conjunction with prediction methods when constructing PIs. The prediction method is required to provide a set of class labels for a test instance, which are obtained from the neighbourhood of the test instance. This section presents such an algorithm that is based

on an ensemble of random decision trees. The algorithm does not do any search to fit the ensemble to the data so as to avoid biasing the computed PIs.

Decision trees are powerful and popular tools for classification and prediction. They are simple and easy to understand and interpret, able to handle both nominal and categorical data, and able to analyse a large amount of data in a short time and perform well. One of the useful advantages of decision trees is that we can use them to identify target groups. For the example of diagnosing cancer patients, a decision tree diagnosing system allows the patients to self-diagnose themselves by simply answering a few yes/no questions, or filling in a couple of values in a form, such as "the time period of having been presenting a symptoms". Then an answer of Yes/No (cancer/not cancer) and a numeric value that shows how likely the answer is correct are returned from the system. In this case, the system is actually making use of the records of a group of identified patients to give the diagnosis. It is also easy for the physician to access the records that were used by the system to make the diagnoses, e.g. how many cancer/noncancer records contributed to the resulting probability value, etc. Also, We can combine multiple decision trees and form an overall prediction. In a typical ensemble of trees, the process is repeated for each tree and the predictions are combined.

The general decision tree structure comprises both splitting and leaf nodes. The splitting nodes involve testing a particular attribute. Depending on the attribute value, instances are assigned to the corresponding branches from a particular splitting node. When classifying an unknown instance, it is routed down each of individual trees and one leaf node in each tree is identified. Leaf nodes give a classification that applies to all instances that reach the leaf. In the terminology of trees, the node at the top of the tree, where the splitting starts, is called the root node, and the splitting nodes and leaf nodes are also called branching and terminal nodes, respectively.

We build an ensemble of random trees based on the following algorithm. At each node we select a splitting attribute at random, starting with the root node. If the attribute that is tested at a splitting node is a nominal attribute, the number of split branches is the number of values of the attribute; if the attribute is numeric, a constant value is determined by averaging the attribute values of two randomly selected instances. Then the comparison between the attribute value of a tested instance and the constant determines which branch the instance goes to, giving a two-way split. Note that, a

numeric attribute can be tested several times in any given path down the tree from the root to a leaf with each test involving a different constant.

When training the ensemble of decision trees, the same induction process applies to each tree in the ensemble. However, this does not mean that the resulting trees are all the same, because (a) each time an attribute is picked out at a splitting node, it is randomly selected from all the candidate attributes, (b) the constants used at each splitting node are computed from the values of randomly selected instances. Thus, the trees in the ensemble are different from one another.

A distinctive feature of our ensemble random tree algorithm is that, instead of combining the predictions made by each of the trees in an ensemble to form an overall decision as existing algorithms usually do, it unites all unique instances from the trees' leaves before making a prediction. This is so that we can get a neighbourhood to compute a PI. Figure 4.5 illustrates the induction of an ensemble of decision trees and how the neighbourhood for a test instance is calculated.

In Figure 4.5, four classifiers (T-1 to T-4) are generated from a dataset with 10 instances. Suppose a test instance is routed down from the root nodes of the four trees, and four leaf nodes (Leaf-1 to Leaf-4) are reached by this test instance. Because each individual tree is built from the same training dataset, the ensemble is likely to have duplicate instances contained in the four leaf nodes. To level the impact of each training instance on the prediction for the test instance, and so as to make it possible to compute unbiased PIs, we unify the instances in the four leaves and eliminate all duplicate instances. As a result, two instances C^0 and G^0 are removed because they appear more than once. The resulting observations of the two classes are 5 and 3, respectively. The statistics are used by the Bayesian model to compute a PI for the class probability of the test instance.

To prevent the resulting trees in the ensemble from containing empty leaf nodes where no instance resides, we impose a restriction to allow no fewer than a certain number of instances in a leaf node: this number is taken as the value of a parameter of the tree induction algorithm.

Another situation that may possibly occur during the tree growing process is that all

instances in a node belong to the same class before the total number of instances reaches the imposed limit. In this case, a parameter is set to signify whether to stop splitting such a 'pure' node, or continue until the minimum number limit is reached. Other parameters used in the induction of the classifier include whether to allow numeric attributes to be further split in the subsequent nodes, etc. These are discussed in the next chapter.

There is also another slightly modified version of this random tree ensemble algorithm that we investigate. It combines the strategies of the k-nearest-neighbour algorithm and decision tree learning. That is, instead of using all the instances in the resulting union, a subset of the instances that are the closest to the test instance are used (measured by Euclidean distance). The specified number of the closest instances is also taken as a parameter of the tree model. If the number happens to be smaller than the total number of instances in the union, all instances are used.

4.7 Summary

Most of this chapter has been concerned with deriving a Bayesian probabilistic model. The model defines the density function of the class probability of a test instance, given a set of instances with known classes. When calculating the likelihood function, a beta distribution is assigned to the class probabilities of the known instances. We do this because (a) the class probability values are unobservable, (b) we lack the knowledge about the true underlying distribution. Thus, employing a distribution like beta, which is flexible enough to model any unimodal distribution in a limited range, is the most conservative choice. We computed the model by simulation. Depending on what assumptions we are willing to make about the prior distribution, various types of PIs can be computed.

The final section of the chapter dealt with the construction of an ensemble of random trees (a random forest classifier), of which the prediction result for a test instance is taken as the input of the Bayesian PI model. Both the random tree ensemble classifier and the Bayesian model will be evaluated in the next chapter.

Chapter 5

Evaluation

In this chapter, experiments are run with the aim of evaluating the performance of the proposed Bayesian PI model and the two random tree ensemble classifiers. To avoid confusion, we name one random tree ensemble classifier $EnsembleRT$, and the other $KnearEnsembleRT$ (as it has the features of both EnsembleRT and the k-nearest-neighbour algorithm). We shall also call the k-nearest-neighbour classifier $Knear$.

The chapter is organised as follows. Section 5.1 compares classifiers EnsembleRT, KnearEnsembleRT, and Knear. The Bayesian PI model is evaluated in Section 5.2, in conjunction with the three classifiers. Section 5.3 summarises the chapter.

5.1 Classifier Evaluation

Since the three classifiers are parameterised with different sets of options, the experimental objective of this section is not only to conduct a performance comparison among the classifiers, but also to find the set of option values, with which the classifiers have their best prediction performance. In subsequent sections of the chapter, the discovered option values will be used in the evaluation of the Bayesian PI model. We do this because the coverage percentages of the computed PIs are only useful when the classifiers are making accurate classifications. Table 5.1 lists the options that apply to the classifiers.

In terms of the scope of the evaluation and the dataset types to be covered, learning tasks for the classifiers are restricted to binary class datasets only. Also, the instances with missing values are removed from the datasets before the datasets are used to train the classifiers. The nine datasets used in the experiments are listed in Table 5.2. The datasets contain mixed numeric and nominal attributes with a wide range of difficulties and class imbalances. The testing is performed using 10–fold cross validation.

Table 5.1: Available options for the three classifiers.

Classifier	Option	Function
EnsembleRT	-E	Number of trees in the ensemble
	-M	Minimum number of instances allowed in leaf node
	-U	Stop splitting when all instances in a node have the same class
	-F	Allow a numeric attribute to be further split on in subsequent splitting nodes
	-S	Seed used when randomly selecting an attribute to split on and split points for numeric attributes
Knear	-K	Number of neighbour instances used in the prediction
	-N	Normalise numeric attribute values when computing distances between instances
KnearEnsembleRT	⋮	(It has all options of the above two classifiers.)

Table 5.2: Datasets used in the experiments.

No.	Dataset	Number of Instances	Number of Attributes	Class Proportion	Source Reference
D1	breast-cancer	277	10	81:196	M. Zwitter, 1988
D2	breast-w	683	10	239:444	M. Zwitter, 1988
D3	credit-rating	653	16	357:296	J. R. Quinlan, 1992
D4	pima-diabetes	768	9	268:500	V. Sigillito, 1990
D5	heart-statlog	270	14	120:150	D. W. Aha, 1988
D6	hepatitis	80	20	67:13	G. Gong, 1988
D7	ionosphere	351	35	225:126	V. Sigillito, 1990
D8	sonar	208	61	111:97	T. Sejnowski
D9	vote	232	17	108:124	J. Schlimmer, 1987

Table 5.3: Option values tested with the three classifiers, from which the optimum values were selected.

Classifier	Option	Option values tested in the experiments
EnsembleRT	-E	{5, 10, 20, 30, 40, 50, 60, 80, 100, 200, 500}
	-M	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20}
	-U	on/off
	-F	on
	-S	{1}
Knear	-K	{1, 2, 4, 6, 8, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80}
	-N	on/off

Classifier: KnearEnsembleRT							
D1	-E{10, 20, 40, 60, 80}	-M{7, 8, 9, 10}	-K{40}	-U	-F	-S{1}	-N
D2	-E{20, 30}	-M{1, 2, 4, 6, 8, 10}	-K{20}	-U	-F	-S{1}	-N
D3	-E{20, 30}	-M{1, 2, 4, 6, 8, 10}	-K{10}	-U	-F	-S{1}	-N
D4	-E{10, 20, 30}	-M{4, 5, 6, 8, 10}	-K{20}	-U	-F	-S{1}	-N
D5	-E{30, 40, 60, 80}	-M{1, 2, 8, 10}	-K{20, 30}	-U	-F	-S{1}	-N
D6	-E{30, 40, 50}	-M{1, 2, 4, 6, 8, 10}	-K{10}	-U	-F	-S{1}	-N
D7	-E{20, 30, 40}	-M{1, 2}	-K{2, 4}	-U	-F	-S{1}	-N
D8	-E{20, 30, 40, 50}	-M{1, 2, 4}	-K{4}	-U	-F	-S{1}	-N
D9	-E{10, 20, 30}	-M{1, 2, 4, 6}	-K{10}	-U	-F	-S{1}	-N

Table 5.4: Optimum option values selected for the three classifiers on the nine datasets tested in the experiments.

No.	Dataset	EnsembleRT	Knear	KnearEnsembleRT
D1	breast-cancer	-E10 -M7 -U -F	-K40 -N	-E80 -M10 -U -F -K40 -S1 -N
D2	breast-w	-E30 -M1 -U -F	-K20 -N	-E20 -M8 -U -F -K20 -S1 -N
D3	credit-rating	-E30 -M1 -U -F	-K10 -N	-E30 -M10 -U -F -K10 -S1 -N
D4	pima-diabetes	-E10 -M5 -U -F	-K20 -N	-E30 -M10 -U -F -K20 -S1 -N
D5	heart-statlog	-E80 -M1 -U -F	-K30 -N	-E30 -M10 -U -F -K20 -S1 -N
D6	hepatitis	-E30 -M1 -U -F	-K10 -N	-E40 -M10 -U -F -K10 -S1 -N
D7	ionosphere	-E20 -M1 -U -F	-K4 -N	-E40 -M1 -U -F -K2 -S1 -N
D8	sonar	-E20 -M1 -U -F	-K4 -N	-E50 -M1 -U -F -K4 -S1 -N
D9	vote	-E10 -M2 -U -F	-K10 -N	-E10 -M2 -U -F -K10 -S1 -N

Results and Discussion

Table 5.3 lists the option values that are tested with the three classifiers. The classifiers EnsembleRT and Knear are tested using the combinations of their corresponding option values listed in the upper part of the table. Based on the performance of EnsembleRT and Knear, the option values in the lower part of the table are then selected and used to test the classifier KnearEnsembleRT on the corresponding datasets. Table 5.4 lists the option values that give the classifiers the best performance in terms of prediction accuracy, based on the above performed experiments.

Figure 5.1, 5.2, and 5.3 present the prediction performance of the three classifiers, which is accomplished using the option values shown in Table 5.4. The performance is measured using root-mean-squared-error (RMSE) on the nine datasets listed in Table 5.2. The RMSE for a single test instance is given by:

$$\sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}}$$

where $p_1, \ldots, p_n$ are predicted values on the tested instances, $a_1, \ldots, a_n$ are the actual values. The datasets listed under the classifier names, on the right-hand side of each figure, are the ones on which the corresponding classifier has significantly better performance than the other, according to a corrected resampled t-test (Nadeau & Bengio, 2003). For the datasets unlisted, there is no significant difference between the two particular classifiers in terms of RMSE.

The three figures show that, on comparison, none of the classifiers stands out and significantly surpasses the others. In particular, regarding the number of datasets for

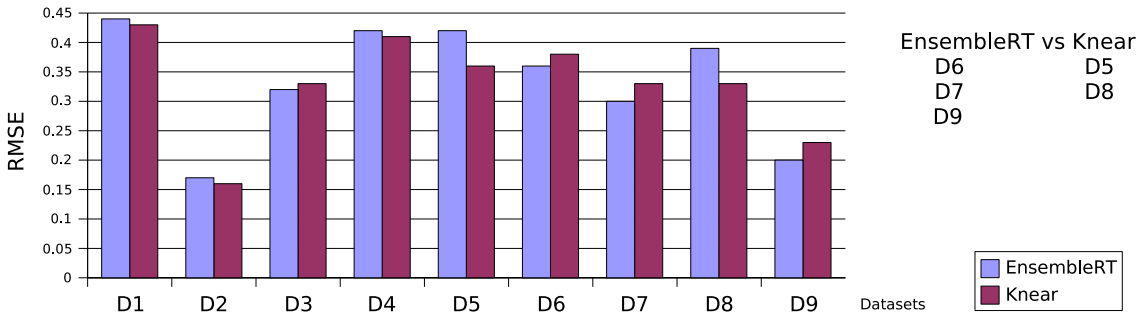

Figure 5.1: Prediction accuracy comparison between EnsembleRT and Knear. EnsembleRT is better than Knear on datasets D6, D7, and D9; Knear is more accurate than EnsembleRT on D5 and D8.

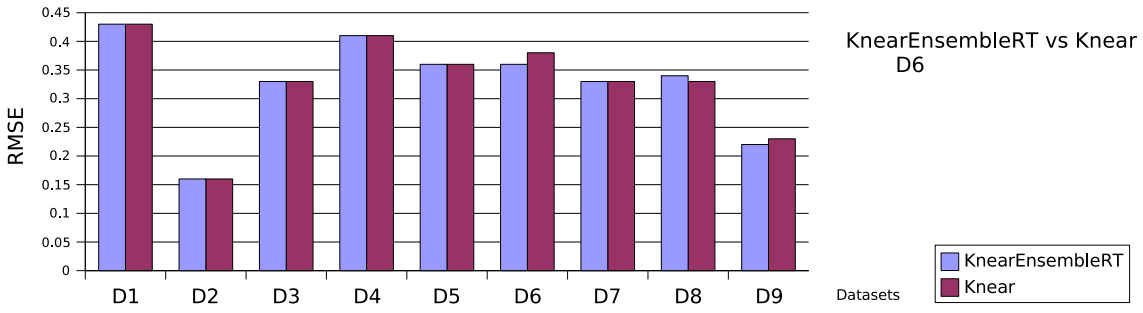

Figure 5.2: Prediction accuracy comparison between KnearEnsembleRT and Knear. Only on one dataset (D6) do the two have significant difference, where KnearEnsembleRT outperforms Knear.

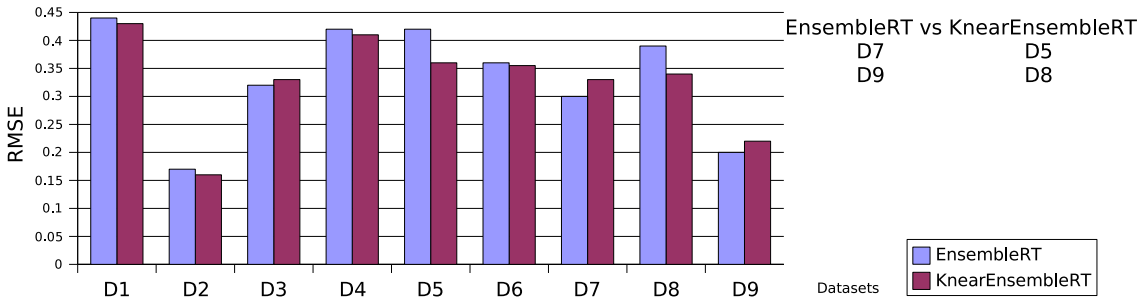

Figure 5.3: Prediction accuracy comparison between EnsembleRT and KnearEnsembleRT. EnsembleRT and KnearEnsembleRT outperform each other on the datasets D7, D9 and D5, D8, respectively.

which better predictions are produced, EnsembleRT and KnearEnsembleRT are all better on only one more dataset than Knear, while in the other comparison EnsembleRT and KnearEnsembleRT are level with each other. Also note that the three classifiers perform analogously on the nine datasets.

5.2 Evaluation of the Bayesian PI Model

In the evaluation of the Bayesian model, we define the term *trial* to denote a leave-one-out testing process, in which an instance is selected from the dataset as the test instance while the rest of the dataset is used to train the classifier and then a PI for the class probability of the test instance is computed. If the true class probability of the test instance is within the range of the calculated PI, it is called a *capture* for the trial; otherwise, it is a *miss*. For each of the nine datasets listed in Table 5.2, 100 trials are performed. Out of 100 trials, $100L\%$ are suppose to result in capture, where L is the specified confidence level that is used in the computation of the PIs. The observed percentage of captures for a particular dataset is called the capture percentage of the PI (PICP) of the Bayesian model for the tested dataset. The idea of this evaluation process is illustrated in Figure 5.4 [based on Fig. 3 in (Willink & Lira, 2005, p. 65)].

Apart from evaluating whether the PICP is equal to (or exceeds) the specified confidence level L, another measure in evaluating the Bayesian model is the width of the computed PI. Since there is no widely accepted standard about a PI's width in the literature, we set the following guidelines for the assessment of the average width of 100 computed PIs at one of the most commonly used confidence levels – 95%:

- If a PI's width is in the range [0, 5.5], it is said to be narrow.
- If the width is in the range [5.5, 6.5], it is good.
- A PI with a width within [6.5, 7.5] is acceptable.
- A width in the range [7.5, 1.0] is wide.

The above criteria are used for the evaluation in this section.

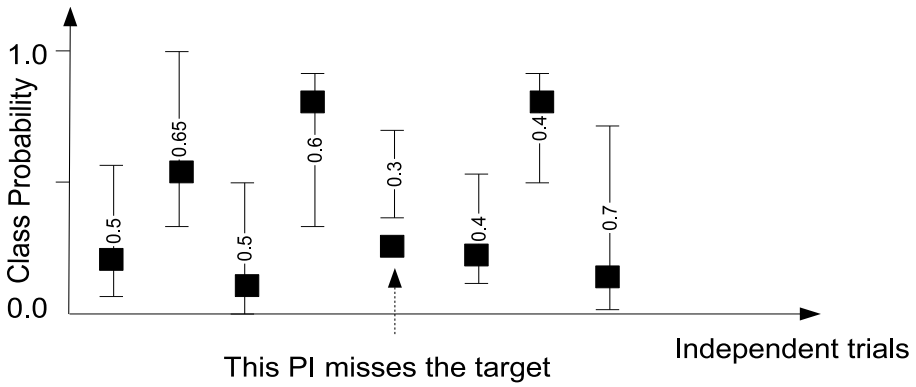

Figure 5.4: Depiction of PIs and their targets (the ■) on eight trials. The number on the PI line is the width of the PI. In 100L% of the trials the value of the target should lie in the computed PI. The numbers marked on the lines are the widths of the PIs.

5.2.1 Experimental Setup for the Bayesian PI Model

The experiments for the Bayesian PI model are run in three stages. In the first stage, we generate random numbers from the beta distribution. The numbers are used to simulate class probability values of hypothetical test instances. We do this because the prior distribution we use in the Bayesian model is the beta distribution, and when the underlying distribution of the class probabilities matches the assumption of the prior distribution, the Bayesian PI model is expected to have satisfactory performance. We take this stage as the validation of the Bayesian model. There is no prediction method (classifier) involved in this first stage.

In the second stage, the model is evaluated on artificial datasets. The generation of the artificial datasets is based on assuming a multivariate normal distribution for the attribute values in each class. The goal in this stage is to test whether the posterior resulting from the beta prior distribution is able to model class probabilities that do not follow the beta distribution.

In the third stage, experiments are run using the real datasets that were used in Section 5.1. A difficulty with the experiments in this stage is the unavailability of the necessary 'true' class probability values of the instances in the dataset. A possible solution is to simulate these class probability values for the instances based on their attribute values. Because the attribute values will not be involved in the computation of the PIs, the probability values calculated in this way are independent of the Bayesian model and can thus be used as the 'true' class probabilities in the evaluation of the Bayesian model.

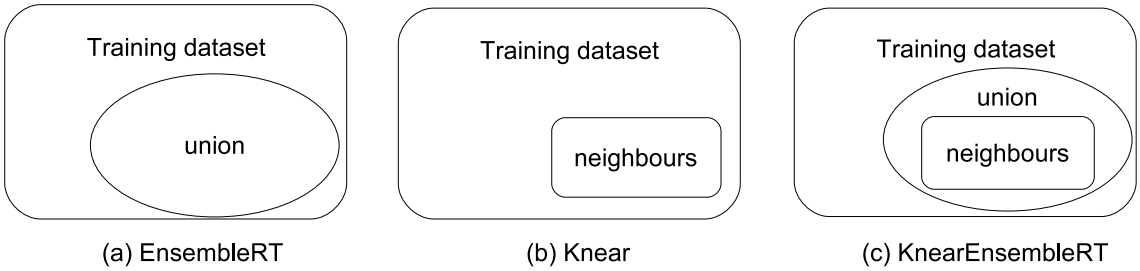

Figure 5.5: Illustration of the localised instances (the oval for EnsembleRT, and the small rounded rectangles for the other two), based on which the prediction is made.

We use the support vector machines combined with logistic regression to compute the probability values.

5.2.2 Testing with Beta Random Numbers

As discussed in the last chapter, the Bayesian model

$$Pr(P_{n+1} \mid D) \propto \int\int \frac{(P_{n+1})^{\alpha-1} \cdot (1 - P_{n+1})^{\beta-1}}{B(\alpha, \beta)} \cdot \frac{\alpha^k \beta^{n-k}}{(\alpha + \beta)^n} d\alpha d\beta \tag{5.1}$$

can be used to compute a PI for P_{n+1}, based on the observed data D. In the model, α and β are the two parameters of the beta distribution; n is the number of class observations in D; and k is the number of observations with the positive class label in n. To compute PI for P_{n+1}, these four parameters need to be determined.

One way to generate the two parameters α and β is to sample from the assumed prior, for example the prior gamma(1, 1), which we discussed in the last chapter. To generate a sample, the inverse CDF method is used. That is, the two parameters of the gamma distribution, $\gamma = 1$ and $\theta = 1$, are substituted into the following gamma probability density function:

$$f(x) = x^{\gamma-1} \frac{e^{-x/\theta}}{\theta^{\gamma}\, \Gamma(\gamma)}, \tag{5.2}$$

where $x \geq 0$. This gives $f(x) = e^{-x}$. Taking the natural logarithm on both sides yields $x = -lnf(x)$, in which $f(x) \sim$ U(0, 1), i.e. it is a randomly generated real value from the uniform distribution in the range (0, 1), and x follows a gamma distribution with the desired parameters.

Without any prediction methods being involved in this stage (because there are no

real instances involved either), we have to generate the proportion (i.e. the two numbers n and k) by simulation. In fact, the value n is just the number of a group of instances that the classifiers use to make predictions for a test instance. The instance group is defined as: for classifier EnsembleRT – the union of instances from the different trees' leaves; for Knear – the group of neighbouring instances; and for KnearEnsembleRT – the group of neighbouring instances within the union. The concepts are illustrated in Figure 5.5. We will hereafter universally call the group of decisive instances the 'union' for all the three classifiers.

To generate n and k, we assume that the size of the union from the leaf nodes of a hypothetical ensemble of trees is less than 40. Thus, the value of n is simulated by randomly selecting an integer number between 1 and 40, based on the uniform distribution. Once the size of the union is determined, the n simulated class probability values are generated as follows. We draw a pair of values of the two beta parameters α and β from the distribution gamma(1, 1). Using the values of α and β, we sample n random numbers from the standard beta distribution. An extra beta random number is also generated to simulate the $(n + 1)^{th}$ observation – the class probability of the predicted test instance.

Depending on different values of the parameters α and β, which are both positive real numbers, there are the following four cases to be considered in the generation of a particular beta random number. All cases use the acceptance/rejection strategy, except for the trivial case when both α and β are equal to one, in which the inverse method is used.

Case 1: When $\alpha = 1$ and $\beta = 1$, the beta distribution becomes the uniform distribution.

Case 2: When both α and β are less than one, the compact algorithm by Johnk (1964, Metrika 8, pp. 5–15) is used.

Case 3: When both α and β are greater than one, the algorithm BB of Cheng (1978, C.A.C.M., 21, pp. 317–322, 10, Crain, I. K.) is used.

Case 4: This case uses the algorithm BA of Cheng (1978) and catches all other combinations of α and β.

Each of the generated beta random numbers represents a hypothetical instance in the union. To assign class values to these instances, the binomial distribution is used. This is

done by comparing each number with a randomly generated real value between 0 and 1 from the uniform distribution. In each comparison, a different random value is generated a corresponding beta number. If the beta random number is greater than the uniform random value, it is labelled positive; otherwise, it is negative.

The value of k is defined as the number of the instances with the positive class label in the union (the n random numbers). Thus, k and n together predict the class of the $(n+1)^{th}$ instance, and the value k/n represents the estimated probability of the instance having the positive class [i.e. the value of the $(n+1)^{th}$ number]. The PI that is computed using the Bayesian model based on n and k is then checked to see whether it contains the probability of the $(n+1)^{th}$ instance having the positive class. Note that, throughout the three-stage experiments, we calculate PIs to contain the probability that a tested instance belongs to the *positive* class so as to maintain consistency.

Having obtained n and k, to compute the PI for the $(n+1)^{th}$ random number [the $(n+1)^{th}$ instance's class probability], we need to sample 1000 pairs of α and β values, using the same prior as used in the generation of the beta random numbers [i.e. the $1^{st}, 2^{nd}, \cdots, n^{th}, (n+1)^{th}$ numbers]. These 1000 pairs are used to compute the posterior distribution by accumulation (as discussed in Section 4.5 in the last chapter).

Apart from the prior gamma(1, 1), the other two priors, U(0, 10) and $\alpha + \beta = 10$, which were also discussed in the last chapter, are tested in the experiments as well. In that case, those priors are used to obtain the two beta parameters to generate the hypothetical data, and also used to sample the 1000 pairs to compute the posterior, respectively.

Results and Discussion

The experimental results are shown in Figure 5.6, 5.7, and 5.8. Figure 5.6 shows the two performance indicators: PICP and average width. We only present PIs at confidence levels ranging from 50% to 99%, as PIs at confidence levels lower than 50% are rarely used in practice.

The capture percentages represented by the four splines in the upper part of Figure 5.6 show that almost all the PICPs are above their corresponding confidence levels. They are close to the desired ideal percentages at confidence levels higher than 80%, especially

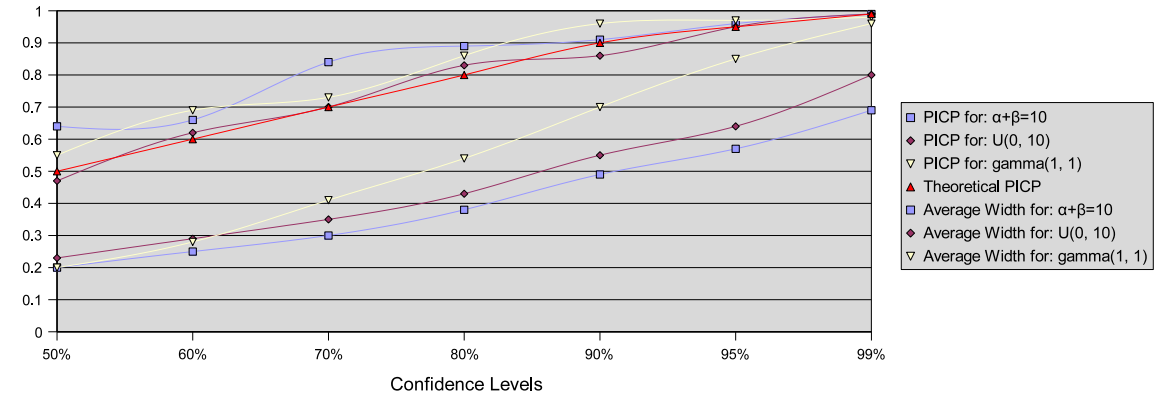

Figure 5.6: Comparison of the PICPs (the four splines in the upper part) and the average widths of the PIs (the three splines in the lower part), computed at various confidence levels. The number of trials at each level is 1000. The red spline with upward triangles represents the theoretical (ideal) PICPs, i.e. the corresponding confidence levels.

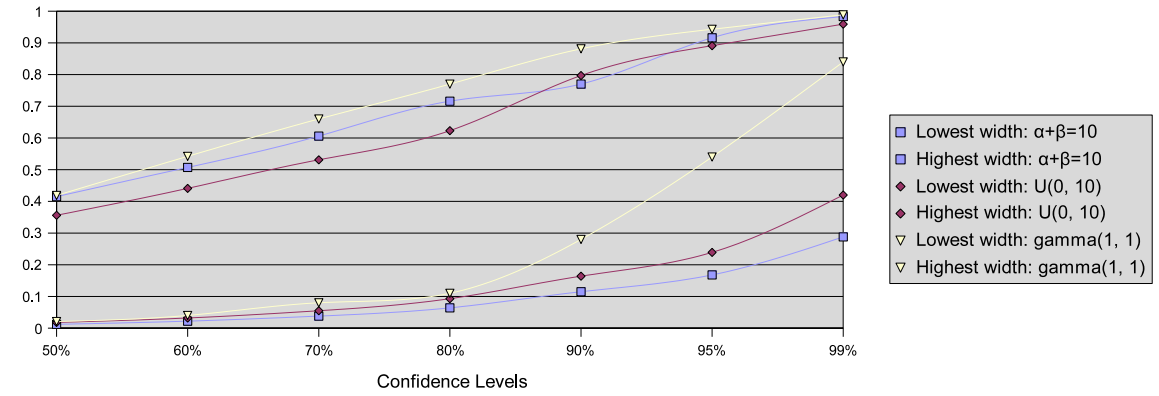

Figure 5.7: Comparison of the highest and lowest widths of the PIs, computed at various confidence levels. The number of trials is also 1000.

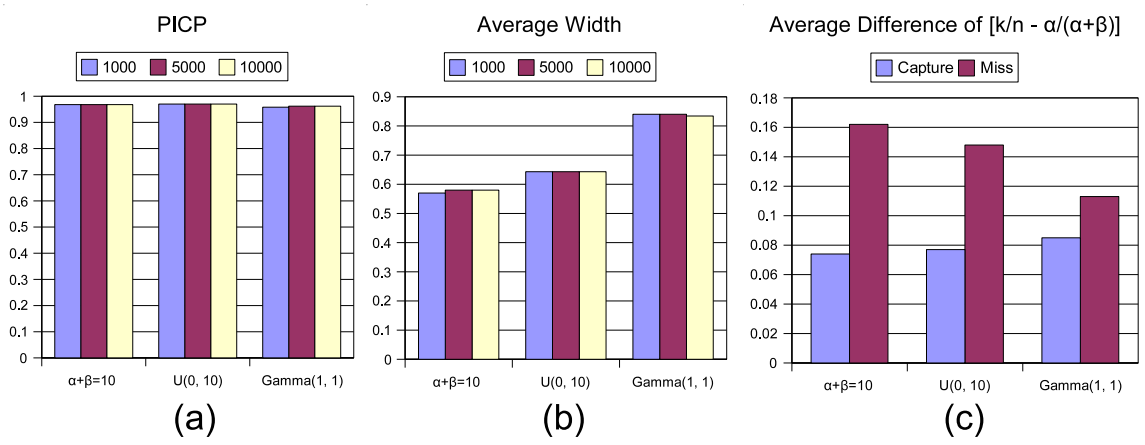

Figure 5.8: Experiment results of 10,000 trials. The PIs were computed with the 95% confidence level. (a) The trend of PICP, corresponding to 1000, 5000, and 10,000 trials. (b) The trend of average width. (c) The comparison of the average value of $(\frac{k}{n} - \frac{\alpha}{\alpha+\beta})$ between the successful trials (marked as 'capture') and those failed (marked as 'miss').

at 95% and 99% (these are the mostly used confidence levels). Because the margins between the PICPs and the theoretical line at confidence levels below 90% are larger than that at confidence levels 95% and 99%, we may say that the PICPs computed at confidence levels below 90% are a little overoptimistic, compared to those at 95% and 99%. In general, the overall performances of the PICPs for the three priors are similar to one another.

The average widths of the PIs are compared in the lower part of Figure 5.6. First, note that the average widths clearly ascend as the confidence levels increase. This follows from the fact that higher confidence levels require wider intervals for those levels to be achieved. Secondly, at most of the confidence levels (except for those below 60%), the widths for the priors gamma(1, 1), U(0, 10) and $\alpha + \beta = 10$ follow the order of becoming narrower. Moreover, according to the guidelines discussed at the beginning of the section, the widths for the three priors fall into three different categories: good, acceptable, and wide, respectively (at the confidence level 95%). This is because they have different sampling spaces, as we discussed in Chapter 4, which introduces different degrees of uncertainty into the posterior distribution.

From Figure 5.6, we can see that different sampling spaces, resulting from using different priors for the model parameters, have much more impact on the width than on the PICP of the PI. As a result, the widths of the computed PIs can be adjusted by choosing different priors for the model parameters, as long as the assumptions implicit in the prior can be reasonably met in situations where the model is applied. Figure 5.6 also verifies that the proposed Bayesian PI model achieves PICPs over the corresponding confidence levels.

Figure 5.7 compares the lowest (the three splines in the lower part) and the highest widths (the three splines in the upper part) of the PIs. Note that, at confidence levels above 90%, the highest widths for the three priors are comparable. It is the lowest widths that bring the differences into their average widths (refer back to the three splines in the lower part of Figure 5.6), which makes the PIs for the prior gamma(1, 1) wider than the other two.

An experiment with more trials was also run with the aim of testing the performance of the computed PIs when the number of trials increases. The results are shown in Figure

5.8. From Figure 5.8 (a), we can see that the PICPs stay almost the same from 1000 trials to 10,000 trials; and they stay much the same across the three different priors as well.

The PIs' average widths are presented in Figure 5.8 (b). The figure shows that the three priors share the same stability of the average widths as the number of trials increases from 1000 to 10,000. Among the three priors, the average widths still maintain the pattern that we discovered in Figure 5.6; that is, the PIs become wider as the sampling space increases from $\alpha + \beta = 10$ to gamma(1, 1).

Recall that, based on Proposition 1 discussed in Section 4.5, the more the value k/n agrees with the fraction $\alpha/(\alpha + \beta)$, the more influence the observed data has on the posterior distribution. The results shown in Figure 5.8 (c) are from a test that calculates the average value of $(\frac{k}{n} - \frac{\alpha}{\alpha+\beta})$ over trials with a capture and trials with a miss, respectively (refer to the illustration in Figure 5.4). Here, α and β are the (α, β) pair that was sampled from a corresponding prior [e.g. gamma(1, 1), etc.], and k and n are based on the data generated from that pair. Obviously, because the data generation process is random, the two fractions differ. The results show that, when a trial fails (i.e. corresponds to a miss), the value of $(\frac{k}{n} - \frac{\alpha}{\alpha+\beta})$ is much larger than when the trial succeeds (i.e. corresponds to a capture). And for the priors $\alpha + \beta = 10$ and U(0, 10), the calculated difference value for the former is almost twice as large as that for the later. This means that the more the observed data agrees with the true parameters, the more likely the PI is to succeed.

In the experiments in this section, we have been assuming the size of the union is between 1 and 40. The following experiment is conducted to see how the PICPs and the average widths react to the variation of the union size. To do this, we let the union size increase in steps of 10 instances from 10 to 100. As before, the three priors, gamma(1, 1), U(0, 10), and $\alpha + \beta = 10$, are used to sample the two beta parameters, α and β, in three different experiments. In each experiment, the same pair of values of α and β is used in the generation of the random beta numbers in the union for all union sizes. At each union size, 100 random beta numbers are tested, and the average value of the widths of the 100 computed PIs is calculated. The PICP is computed as the capture percentage as usual. The value k was also determined using the same method as before. The confidence level used when computing the PIs was 95%. Figure 5.9 shows the results.

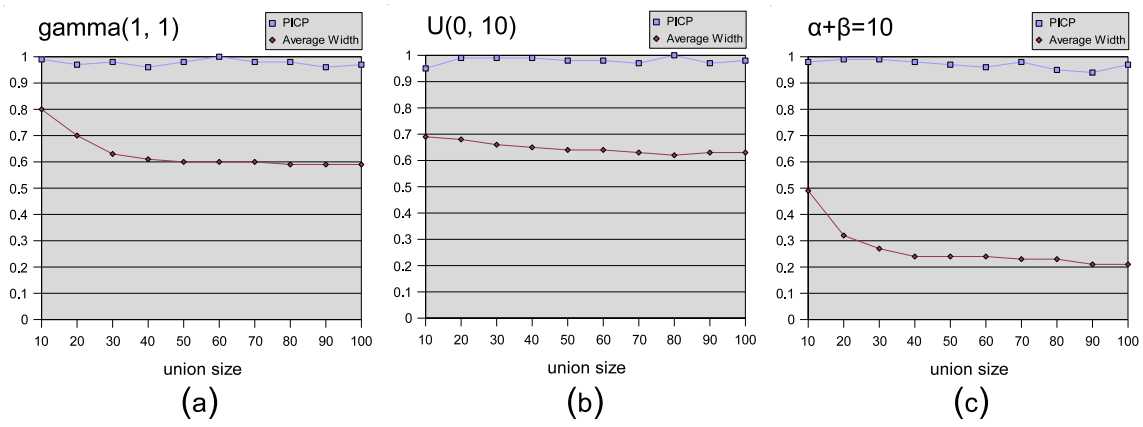

Figure 5.9: Results of experiments computing the PICPs and the widths of the PIs when systematically varying the union size with different sampling priors for the beta parameters. The sampled beta parameter values are: (a) $\alpha = 1.93$, $\beta = 0.14$; (b) $\alpha = 4.82$, $\beta = 7.98$; (c) $\alpha = 9.64$, $\beta = 0.36$.

The results in the figure show that, as the union size increases, the average width of the PI decreases. However, when the class observations in the union increase to a certain size, there exists a point after which the average width remains stable. Specifically, for gamma(1, 1) and U(0, 10), this occurs at the union size 50; for $\alpha + \beta = 10$, it is at the size 40. This is when there is enough data to pick out the appropriate beta distribution(s) based on their likelihood. The PIs are then effectively based on this reduced set. The PICPs exhibit some random fluctuation around 95% as expected because the class proportions in the unions are randomly determined for each union size.

5.2.3 Testing with Artificial Datasets

In the last section, the Bayesian PI model was tested on random numbers generated from beta distributions, and it was validated by obtaining satisfactory results for the PICPs and the average widths of the PIs. In this section, we test the model using artificial datasets, in which the class probabilities of the instances are generated by assuming that the attribute values in each class follow a multivariate normal distribution (or multi-normal distribution).

A dataset is composed of instances; an instance consists of a number of attributes, one of which represents the class of the instance. To create an instance, we need to generate a specified number of attributes, based on which the class probability of the instance is produced. To simplify matters, we make all the attributes numeric, and also store the

class probability in one of the attributes, specifically the second to last attribute (the last attribute is set to be the class attribute by default). we call the second to the last attribute of an instance the *class probability attribute*.

Class Probability Computation

If the true distribution of the attribute values in each class is known, computing the class probabilities for the instances is simple and can be done based on Bayes' rule of conditional probability (discussed in Chapter 2). That is, the probability that an instance x has the class c_j can be computed as

$$Pr(c_j \mid x) = \frac{Pr(x \mid c_j) \cdot Pr(c_j)}{Pr(x)} = \frac{Pr(x \mid c_j) \cdot Pr(c_j)}{\sum_{j=1}^{i} Pr(x \mid c_j) \cdot Pr(c_j)} \tag{5.3}$$

where $Pr(x)$ is called the marginal distribution of the instance x, and expressed in the above equation using the conditional probability $Pr(x \mid c_j)$, and the prior probability $Pr(c_j)$. For a binary class problem, i.e. $i = 2$ in the equation, the probability that an instance x has the class *yes* can be written as

$$Pr(yes \mid x) \quad = \quad \frac{Pr(x \mid yes) \cdot Pr(yes)}{Pr(x \mid yes) \cdot Pr(yes) + Pr(x \mid no) \cdot Pr(no)} \tag{5.4}$$

The probability that the instance x is in class *no* [i.e. $Pr(no \mid x)$] can be computed by substituting *no* for *yes* in the numerator in the equation, or simply subtracting $Pr(yes \mid x)$ from one.

To use the expressions to compute the desired class probabilities of the instances, we first need to assign class labels to the instances using the prior probabilities $Pr(yes)$ and $Pr(no)$. The attribute values of an instance are generated based on the normal distribution with specified mean and standard deviation, depending on what class label has been assigned to the instance. The conditional probabilities $Pr(x \mid yes)$ and $Pr(x \mid no)$ for each instance are generated based on the normal distribution densities of each attribute of the instance. The class probability value of an instance can then be computed using these conditional probability values and the above expressions. Note that, in the class probability attribute of an instance, we only store the probability of the instance having the class *yes*, regardless of the class label of the instance; and it is consistent for all instances in the generated datasets.

The specified prior probabilities, $Pr(yes)$ and $Pr(no)$, determine the class proportions in the generated instances of the dataset, which are in turn determined based on the prior knowledge that we have about the dataset. We specify the values of the two prior probabilities in two ways. The first one is simple: we make $Pr(yes) = Pr(no) = 0.5$. These equal class proportions are a little unrealistic, as there should be a variety of class proportions for real datasets. Another option is to specify the class proportions according to real datasets, i.e. empirical class proportions. We use the datasets that were used in the evaluation of the classifiers in the last section (Table 5.2).

Having specified the prior probabilities, we can use them to label the instances. In particular, for each instance, a different random value between 0 and 1 is generated based on the uniform distribution. This random value is then compared with a prior probability value (whether the prior probability value for the class *yes* or class *no* is used does not matter, as long as the same probability is used consistently for all instances in the dataset). If the former is larger than the later, the instance is assigned to the class *yes*; otherwise the class *no* (this also has to be consistent for all instances).

Now we generate attribute values of the instances, based on their corresponding class labels assigned in the above step. As mentioned at the beginning of this section, in the generation of the artificial datasets, we use a multivariate normal distribution to model the attribute values in each class. In probability theory and statistics, a multivariate normal distribution can be thought of as a generalisation of the one-dimensional normal distribution. In the general case, the multivariate normal density for an instance x for a particular class c can be written in the following notation:

$$Pr(x \mid c) = N_m \ (\mu, \sum), \tag{5.5}$$

where m is the number of attributes of the instance x; μ is the mean; and $\sum$ is the covariance matrix of the attributes. Note the notation N_m in Expression (5.5), which makes it explicit that $Pr(x \mid c)$ is m-dimensional. In the above expression, we assume that $\sum$ is a diagonal matrix. The diagonal entries of the matrix are variances of the normally distributed attributes, while the off-diagonal entries are all zero, which is based on the assumption that the attributes are independent of one another.

Because the attributes in each class are assumed to be multinormally distributed

and the attributes are also assumed to be independent of one another, the values of a particular attribute in a dataset are normally distributed with a specified mean and standard deviation. We also know that, by 'standardising,' any probability involving a normal random variable (denoted by rv) with mean μ and standard deviation δ can be expressed as $(rv - \mu)/\delta$, which is a standard normal random variable. That is, subtracting μ shifts the mean from μ to zero, then dividing by δ scales the variable so that the standard deviation is 1 rather than δ. Therefore, we can simply reverse the process and 'unstandardise' a standard normal random variable with a specified mean and standard deviation to generate the value of an attribute of an instance for each class.

This is achieved specifically as follows. A random value from the standard normal distribution (with mean zero and standard deviation one) is produced. Then the following two values are computed for an attribute a:

$$\begin{aligned} a_{yes} &= rv * \delta_{yes} + \mu_{yes} \\ a_{no} &= rv * \delta_{no} + \mu_{no} \end{aligned} \tag{5.6}$$

where μs and δs are the means and standard deviations, which are further specified below. Depending on the class label of the instance, the attribute is assigned either the value a_{yes} or a_{no}.

Now we need to specify the values of the mean (μ) and the standard deviation (δ) for each of the attributes in the dataset in order to generate the attribute values. We set the mean and standard deviation as follows:

$$\begin{aligned} &\mu_{yes} = 1,\ \mu_{no} = -1; \\ &\delta_{yes} \text{ and } \delta_{no} \text{ are randomly selected from } \{0.05, 0.1, \ldots, 1\}. \end{aligned} \tag{5.7}$$

That is, if an instance is in the class *yes*, each of its attributes is assigned the value 1 to be the attribute's mean; if the class is *no*, the mean is assigned -1. Two values randomly selected from $\{0.05, 0.1, \ldots, 1\}$ are assigned as the attribute's standard deviations for class *yes* and class *no*, respectively. This specification generates datasets such that the distributions of the two classes slightly overlap with each other, which is common in real datasets. The two distributions are illustrated in Figure 5.10, using an example of a single attribute dataset.

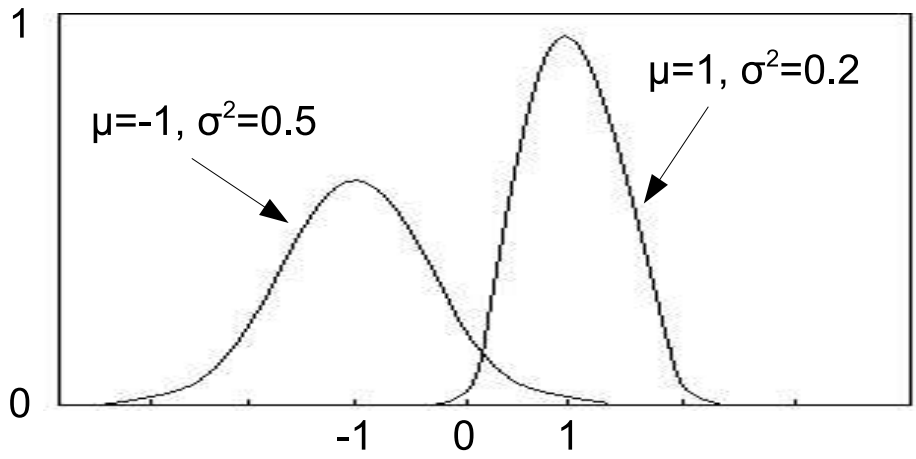

Figure 5.10: Illustration of the distributions of the two classes of a generated artificial dataset with a single attribute.

Having generated the attribute values, the conditional probability of an instance can be computed by calculating the product of individual normal densities of each attribute value with regard to each class. That is,

$$\begin{aligned} Pr(x \mid yes) &= \prod_{i=1}^{m} f(a_{i,yes};\ \mu_{i,yes}, \delta_{i,yes}), \\ Pr(x \mid no) &= \prod_{i=1}^{m} f(a_{i,no};\ \mu_{i,no}, \delta_{i,no}) \end{aligned} \tag{5.8}$$

where x is the instance; m is the number of attributes; $a_{i,yes}$ and $a_{i,no}$ are the attribute values generated using the expressions in (5.6); and f represents the normal density function. Note that the generation of the conditional probability $Pr(x \mid \cdot)$ uses all attributes and thus allows them to make equal contribution to the likelihood (no weighting is used), which is, again, based on the assumption that the attributes are equally important and independent of one another.

By using the conditional probabilities $Pr(x \mid yes)$ and $Pr(x \mid no)$ generated for each instance, the class probabilities that the instances belong to the class *yes* (as discussed above, we always use the probabilities of class *yes*) can be computed based on the expression in (5.4). The generated datasets are tested in the next section.

Results and Discussion

The properties of the nine generated artificial datasets are listed in Table 5.5. Figure 5.11, 5.12, and 5.13 show the experimental results in terms of PICP, average width, and average

union size, using the three classifiers. In the figures, the blue splines with squares represent the datasets generated by assuming the prior distributions $Pr(yes) = Pr(no) = 0.5$, i.e. equal class proportion; the brown splines with diamonds represent those with prior probabilities based on the real datasets listed in Table 5.2, i.e. empirical class proportion. The statistics in these figures are obtained by using the gamma(1, 1) prior distribution to sample the two beta parameters of the Bayesian model, and the confidence level used in the computation of the PIs is 95%.

Figure 5.11 (a) shows that most of the PICPs are close to the desired 95% level. The PICPs for datasets D2 and D3 with both class proportions, however, are around 80%. To see a likely reason for the poor performance on D2 and D3, we look at Figure 5.11 (c), in which the average union sizes for D2 and D3 are the largest ones among the datasets. A large union can cause variation in the underlying class probabilities, thus making it harder for the sampling process to catch the true class proportion (i.e. k/n) and create a reliable interval.

Looking at the average widths in Figure 5.11 (b), most of the widths are in the acceptable range, i.e. below 0.75. By examining both (b) and (c) in Figure 5.11, we note that, the larger the average union size, the lower the average width. This applies to the datasets with both class proportions. Also, when the average union sizes of the two class proportions are close to each other, their average widths are also close. This can also be observed from the experimental results for classifiers Knear and KnearEnsembleRT in Figure 5.12 and 5.13, in which the average widths obtained from the two class proportions are almost always the same because of the same union sizes used in the corresponding tests. In general, the PICPs for the equal class proportion are slightly higher than those for the empirical class proportion. This holds for all three classifiers.

5.2.4 Testing with Semi-Real Datasets

As the section title implies, the datasets that we will be testing on are not the real datasets as it were because we need to simulate the class probabilities for the instances so that the dataset can be used to test the Bayesian PI model. In this section, the nine datasets listed in Table 5.2 are modified and then tested using the same procedure that was used in the last section.

There are two tasks involved in the modification of the datasets. We first have to

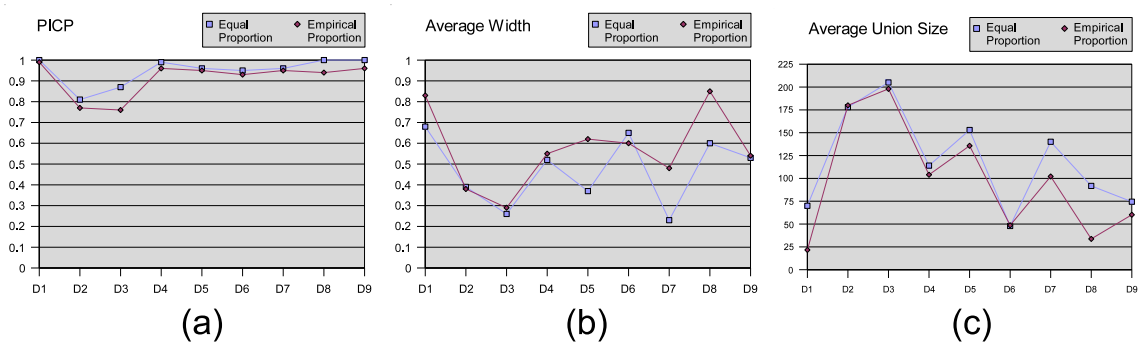

Figure 5.11: Experiment results for classifier EnsembleRT on artificial datasets created with two different class proportions.

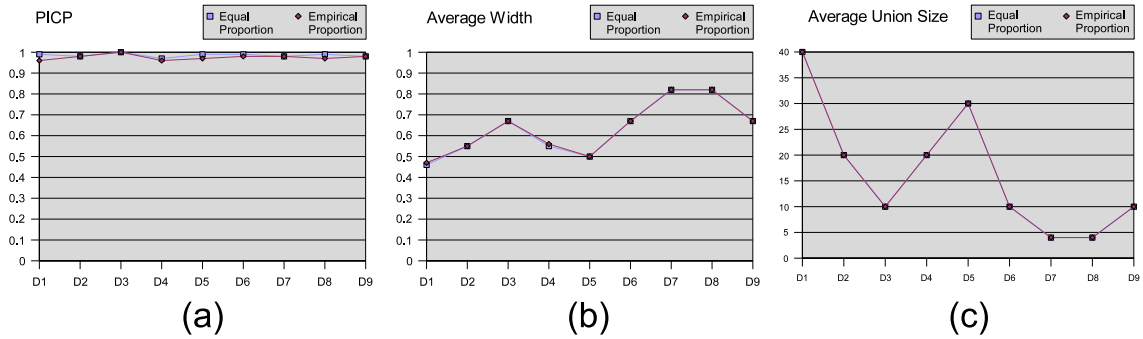

Figure 5.12: Experiment results for classifier Knear on artificial datasets created with two different class proportions.

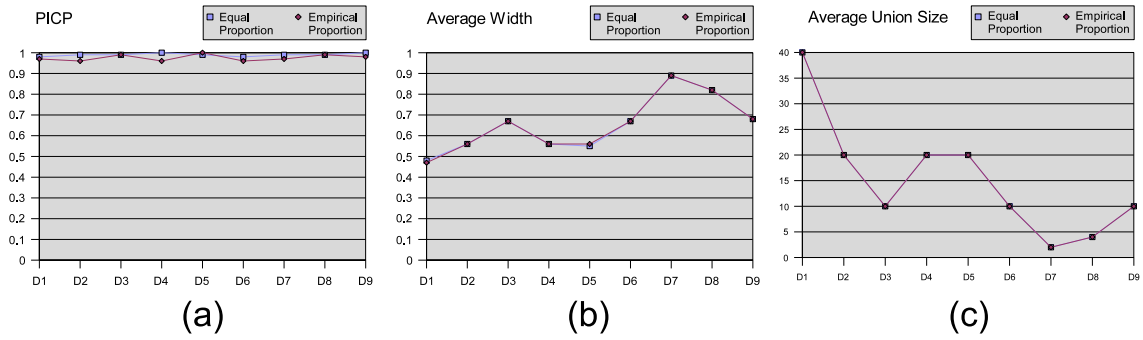

Figure 5.13: Experiment results for classifier KnearEnsembleRT on artificial datasets created with two different class proportions.

Table 5.5: The class proportions of the artificial datasets generated using equal prior probability (the third column), and based on the class proportions of the real datasets listed in Table 5.2 (the fourth column).

Artificial Dataset	Number of Instances	Equal Proportion	Empirical Proportion
D1	277	140:137	78:199
D2	683	339:344	231:452
D3	653	326:327	365:288
D4	768	378:390	267:501
D5	270	137:133	124:146
D6	80	40:40	64:16
D7	351	178:173	225:126
D8	208	103:105	112:96
D9	232	115:117	107:125

add an extra attribute into the dataset, which holds the simulated class probability of a particular instance. Secondly, the original classes of the instances have to be reassigned based on the simulated class probability values. In particular, the class of an instance is assigned based on the result of comparing its class probability value with a real value between 0 and 1, which is randomly generated from the uniform distribution. For each instance, a different value is generated, with which the class probability of the instance is compared. The classes are also assigned with consistency. This means that, for any instance in the dataset, if the class probability value of a particular instance is greater than the correspondingly generated random value, a positive class (i.e. the value 1.0) is assigned to the instance. Otherwise a negative class (i.e. the value 0.0) is assigned.

We use a support vector machine with a logistic regression model fit to its output to generate the simulated class probability for an instance, based on the attribute values of that instance. Two types of kernels are used with the support vector model, of which one kernel (the polynomial kernel with the exponent set to 1) creates a linear boundary between the two classes of the instances, while the other utilises *redial basis functions* (RBF) and nonlinearly divides the data space according to the two classes.

Results and Discussion

Figure 5.14 shows the testing results of the Bayesian PI model on the modified real datasets, in which the model is tested in conjunction with the three classifiers. We use a 95% confidence level in the PI calculation, and gamma(1, 1) as the prior distribution for the two parameters of the beta distribution throughout the experiments in this section, unless otherwise stated. The prediction accuracy shown in the rightmost column in the

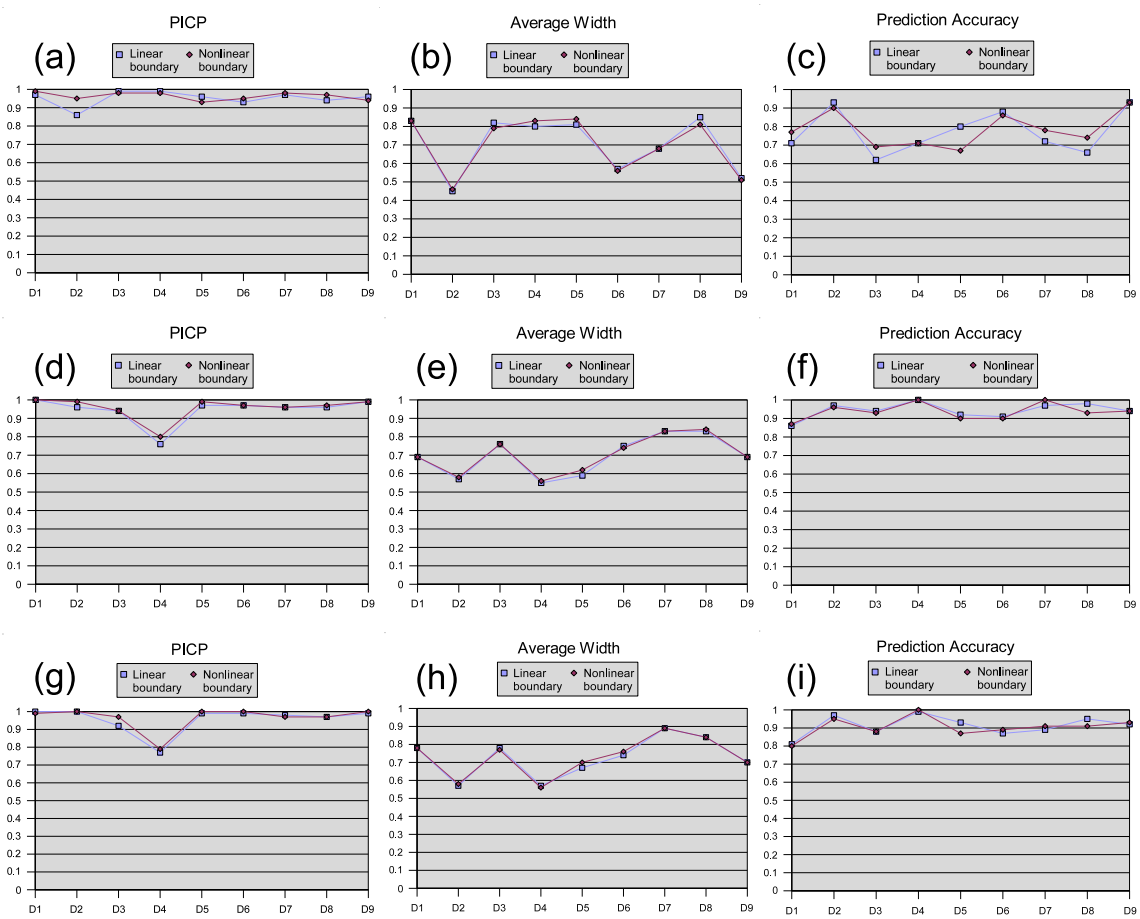

Figure 5.14: Comparison between datasets with linear and nonlinear class spaces. (a) – (c) for EnsembleRT; (d) – (f) for Knear; (g) – (i) for KnearEnsembleRT.

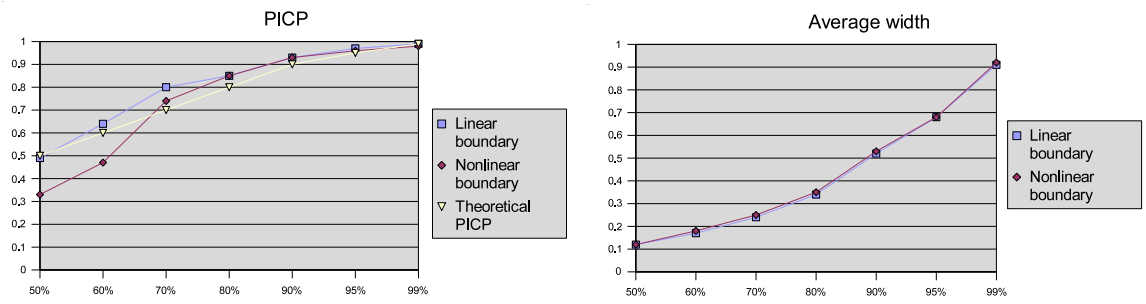

Figure 5.15: PICP and average width at various confidence levels. The classifier used was EnsembleRT, and the dataset was ionosphere.arff.

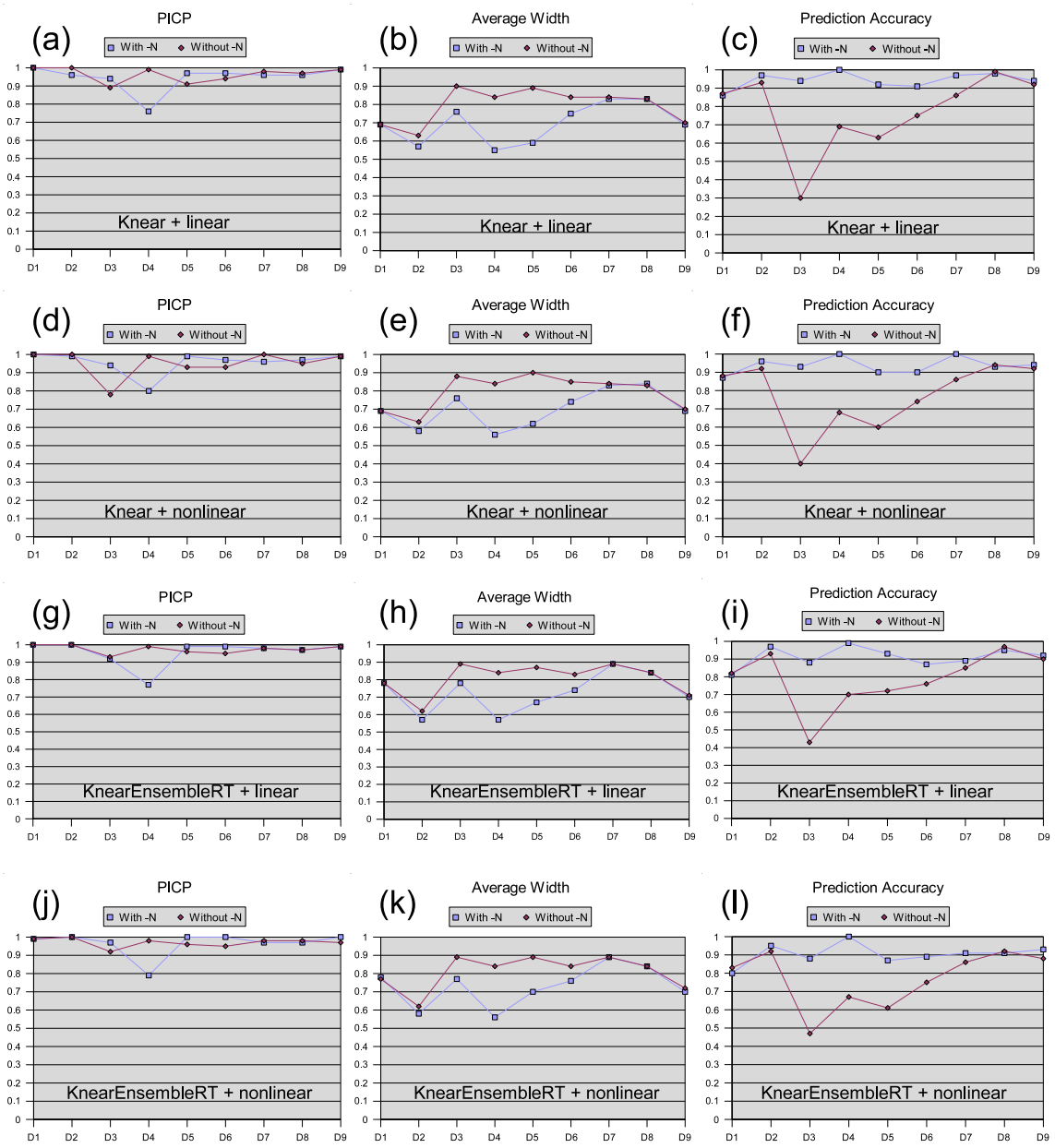

Figure 5.16: Comparison of PICP, average width, and prediction accuracy when normalising and not normalising the attribute values of the dataset. (a) – (c): classifier Knear on linear dataset, and (d) – (f): Knear on nonlinear dataset; (g) – (i): classifier KnearEnsembleRT on linear dataset, and (j) – (l): KnearEnsembleRT on nonlinear dataset.

figure is measured with percentage correct of class predictions.

We can see from the graphs in Figure 5.14 that the model is showing consistent performance across the two types of datasets for all three classifiers. Also note that when the PICP is lower [e.g. for dataset D2 in (a) and dataset D4 in (d) and (g)], the average widths of the PIs are also narrower [in (b), (e), and (h)]. For the classifier EnsembleRT, the best performance occurs on dataset D6, in which the PICP is close to the specified confidence level, the average width is below 0.6, and the prediction accuracy is close to 90%. The other two classifiers, Knear and KnearEnsembleRT, exhibit their best result on dataset D2, with the average width remaining below 0.6 and good PICP and accuracy.

Figure 5.15 shows the trend of the PICPs and the average widths at various confidence levels. In the figure on the left, the PICPs for the dataset with linear boundary in the instance space stay above the theoretical line across all the confidence levels. The PICP line also becomes closer to the theoretical line at the levels above 90%. The same is observed for the datasets with nonlinear boundaries, except at confidence levels lower than 60% when the percentages fall below the theoretical line. In the figure on the right, almost undistinguishable lines of average widths between linear and nonlinear datasets are obtained in the figure, with acceptable widths at 95% confidence level and the levels below, and a wide PI (close to 0.9) at 99%.

Finally, a comparison is made in Figure 5.16 between normalising and not normalising the attribute values when computing the neighbouring distance between instances. The experiments are thus run only with classifiers Knear and KnearEnsembleRT. The results show that, regarding the three measurements (PICP, average width, and prediction accuracy), there is not much difference between the datasets with the linear and nonlinear class boundaries. So are the PICPs for all cases (the four figures in the first column). However, when the normalisation option is turned off, the interval widths become wider and the prediction accuracies decrease for most of the datasets. The experiments on the original real datasets (before they were modified to suit the need of PI testing by using the predictions of the support vector machines to obtain class probabilities and new class labels) also showed that when the normalisation option -N was on, the highest accuracies occur (Section 5.1). Another observation in the figures is that turning off the -N option does not affect the model performance on some of the datasets, for example, on dataset D1. This is because these datasets either only have nominal attributes or only contain a

minority of numeric attributes.

5.3 Summary

There were three classifiers together with the proposed Bayesian PI model involved in the evaluation described in this chapter. We first evaluated the three classifiers by finding the best set of the parameter values for the classifiers (Section 5.1). With the specified parameter values, the classifiers performed to a comparable degree in terms of prediction accuracy across the nine selected binary class datasets. The Bayesian PI computation model was evaluated using three different, successively more realistic scenarios. It was first validated by computing PIs for random numbers generated from the beta distribution, which is the same probability distribution as the prior assumed for the model. Then it was tested on artificial datasets, in which the attribute values of the instances follow a multivariate normal distribution in each class. The results from the tests on the artificial datasets demonstrate that our model based on the beta distribution produces satisfactory PIs in most cases. Finally, promising results were also obtained from the experiments testing the Bayesian PI model on modified real datasets.

Chapter 6
Conclusions and Future Work

As the machine learning approach becomes widely applied in the fields of forecasting and medical diagnosis, and many others, there is a need for expressing the uncertainties that inevitably come with the class probability predictions for new examples. Unfortunately, not only has there been no available method in any of the existing systems that can fulfill the function, but there also appears to have been no such attempt in the research literature in the case where prediction intervals (PIs) are calculated for the class probability predictions made by classifiers based on decision tree or nearest neighbour learning. There is thus a need to investigate methods of constructing PIs for the class probabilities of a predicted instance. This defines the central mission of this study.

Specifically, in the first chapter of this thesis, we set up the following two objectives: (a) derive a Bayesian model for PI computation, and (b) introduce a decision-tree-based learning algorithm suitable for the computation of PIs, with the former being the main goal of this study. This chapter describes the contributions of the thesis, possibilities for extensions and future work.

6.1 Contributions

The strong point of this thesis is that it appears to contribute the first attempt to quantify the uncertainty inherent in the class probability estimates of a decision-tree-based classifier and a nearest neighbour classifier. To this end, it introduces a Bayesian method for computing PIs. The inevitable uncertainty about the class probability estimate for a new instance can now be quantified by the width of the calculated PI. The significance of the study lies in that, when it comes to reporting a probability estimate, as well as the precision of the estimate, users can now directly examine the uncertainty about the estimate by computing a PI for it, rather than having to rely on other means to infer the

reliability of the predicted probability.

The research described in this thesis is also an effort of employing Bayesian inference to derive PIs for class probability estimates of machine learning classifiers. The challenge imposed by the unobservability of the class probabilities is surmounted by taking advantage of the Bayesian approach, in which a prior distribution is assigned to the possible distributions of unobservable class probabilities (which we assumed to be beta densities), and the prior is updated based on the observed data – the instances with known classes in the neighbourhood of the test instance. These known instances are returned from the prediction models (e.g. the decision tree classifier introduced in the thesis) that are built from binary class training datasets. The returned instances provide a set of class labels (i.e. a set of 0s and 1s). These labels are then taken as the input of the Bayesian PI model in the form of class proportions, which represent the number of the instances with positive and negative classes, respectively. A complete predictive distribution for the class probability of a test instance is derived from the combination of the specified prior distribution and the class observations. The desired PI can then be calculated from this posterior distribution. We discussed and evaluated a method for performing this calculation, which ensures that the width of the resulting PI covers the area that exhibits the highest posterior density, and thus constructs the PI with the shortest width.

A new decision-tree-based learning algorithm was also proposed and described in this thesis. The classifier induced by the proposed algorithm is composed of an ensemble of a specified number of random decision tree classifiers. Each classifier in the ensemble is trained upon the same training dataset, but with randomness injected into the training process. The class labels in the training data are only used (optionally) to decide when to stop splitting, so as not to bias the subsequent computation of the PI. Another feature of the proposed algorithm that distinguishes it from existing ones is that, instead of combining the predictions made by the individual classifiers, it unifies the leaf nodes from the different tree classifiers in the ensemble and then comes up with a prediction based on the union of the instances of those leaves. This is so that every instance is only counted once when the PI is computed.

The proposed random tree ensemble classifier (the EnsembleRT classifier) was also modified by applying the well-known k-nearest-neighbour algorithm to the obtained union of instances, which results in the derivation of the k-nearest-neighbour random tree

ensemble classifier (the KnearEnsembleRT classifier).

The Bayesian PI model and the proposed random tree ensemble classifiers were implemented and evaluated, and the two classifiers were compared with the k-nearest-neighbour classifier. The Bayesian model was evaluated, in conjunction with each of the three classifiers, based on hypothetical class probabilities generated from beta distributions, one artificial dataset, and nine selected real-world binary datasets, for which the class probabilities were generated using a support vector machine.

The results of the experiments show that the two random tree ensemble classifiers perform comparably with the k-nearest-neighbour classifier. Performance was measured by the root-mean-squared-error (RMSE) of the predicted probabilities. The tests of the Bayesian PI model also show that, (a) it produces PIs with capture percentages close to the specified confidence levels when the assumption of beta distributed class probabilities is correct, and (b) when applied to the predictions of the learning algorithm, it produces PIs with capture percentage exceeding the specified confidence level in most cases, while still maintaining relatively narrow interval widths.

6.2 Future Work

It is often the case that more questions than answers are produced in academic research. This study is no exception to this rule. This section lists some points that could give rise to further exploration of the proposed Bayesian PI model.

- When the Bayesian model was tested on the modified real datasets, we found that altering the option -N specified for the classifiers did not cause significant changes to the PICPs, but that it influenced the widths of the PIs for most of the datasets. Therefore, more experiments with other prediction methods, combined with the Bayesian model, need to be tested to see whether there is a connection between the configuration of the prediction method used and the properties of the computed PIs. In other words, is it possible in practice to simultaneously achieve an appropriate PICP, the narrowest width of the PI, and the highest accuracy of the prediction? In addition to what has been discovered in this study, a comprehensive investigation of the relationship between the PICP, the width of the PI and the prediction accuracy would be very useful.

- The beta distribution was used to model the unobservable class probabilities pri-

marily because of its simplicity and flexibility. However, we do not want to rule out the possibility that a distribution other than the beta distribution could be used to model the class probabilities of the instances in the union, and could perhaps be more appropriate.

- In the computation of the posterior, the state of convergence was assumed to have occurred when a large, fixed number of density values has been accumulated. Although, initial experiments showed that varying the number of iterations has only a minor effect on the posterior distribution, an accurate determination of the convergence could bring up a more truthful distribution of the unknown quantity.

- There is always uncertainty created when simulation techniques are used in the computation of a quantity. Analytically solving the calculation of the posterior (e.g. by finding an appropriate conjugate prior) would eliminate the uncertainty introduced by the simulation process. Another solution to reducing uncertainty, which is more practical, may be to test the model with priors that are more informative than the ones used in the experiments.

- In this study, only one PI is computed for the unknown class probability. We could compute several PIs for the same estimate with randomness injected into the computation of each PI, and then somehow combine them together. One way to combine these PIs may be to average their upper and lower limits, respectively, but an obvious obstacle is that the PIs could be disjoint. Averaging PIs should make the interval estimates more reliable.

- Another approach to combining multiple PIs is that, a PI could be computed based on the leaf node of each individual tree classifier in the ensemble, and these PIs could then be combined to form one single PI.

- A functionality that is missing in the implementation of the Bayesian model is the ability to handle missing values in a dataset.

- Dealing with multi-class problems is another possible extension of the method. This could be achieved using the multinomial distribution and the Dirichlet distribution.

There are so many situations in which decision makers should have interval estimates available when making critical decisions. For example, in the teleconference meeting on the night before the space shuttle *Challenger* was launched, there was a debate on the

issue whether it was safe to launch the shuttle the next morning because there had been a forecast of a 31°F temperature at launch time and this low temperature could impact the performance of some parts of the shuttle. If the participants in the meeting had available an interval estimate for the probability of failure for the scheduled launch (or even an interval estimate for the probability of damage of a particular part), they might not have spent three hours in the discussion and at the end made the wrong decision to launch the shuttle at the scheduled time, and a catastrophic event could possibly have been avoided (Dalal et al., 1989, p. 945). This example demonstrates a situation similar to diagnosing a patient who possibly has developed cancer (discussed in Chapter 1), and to many others as well. Over the years, many researchers and other interested parties are placing a great emphasis on improving the accuracy of prediction systems. It is hoped that this study will help to draw more attention to interval prediction, because when uncertainty about a prediction is inevitable, producing an interval estimate can provide a basis for the process of making important decisions.

Appendix A

We pick up from Equation (4.13) in Chapter 4.

To solve the formula

$$\frac{\alpha y_i \ \cdot {}_2F_1(\alpha+1, 1-\beta; \alpha+2; 1) - (\alpha+1)(y_i-1) \cdot \ {}_2F_1(\alpha, -\beta; \alpha+1; 1)}{\alpha(\alpha+1)B(\alpha,\beta)} \tag{1}$$

we utilize the following Kummer's first formula:

$${}_2F_1(\frac{1}{2}+m-k; -n; 2m+1; 1) = \frac{\Gamma(2m+1)\Gamma(\frac{1}{2}+m+k+n)}{\Gamma(\frac{1}{2}+m+k)\Gamma(2m+1+n)}$$

The simplification of the formula can be done by letting in

$$a = \frac{1}{2}+m-k, \quad b=-n, \quad \text{and} \quad c = 2m+1,$$

which gives

$${}_2F_1(a; b; c; 1) = \frac{\Gamma(c)\Gamma(c-a-b)}{\Gamma(c-a)\Gamma(c-b)} \tag{2}$$

Applying the above simplified formula to Equation (1) yields

$$\begin{aligned}
& \tfrac{\alpha y_i \ \cdot {}_2F_1(\alpha+1,1-\beta;\alpha+2;1) - (\alpha+1)(y_i-1)\cdot \ {}_2F_1(\alpha,-\beta;\alpha+1;1)}{\alpha(\alpha+1)B(\alpha,\beta)} \\
= \quad & \tfrac{-1}{\alpha(\alpha+1)B(\alpha,\beta)} \left[(\alpha+1)(y_i-1)\tfrac{\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+1)} - \alpha y \tfrac{\Gamma(\alpha+2)\Gamma(\beta)}{\Gamma(\alpha+\beta+1)}\right] \\
= \quad & \tfrac{-1}{\alpha(\alpha+1)B(\alpha,\beta)} \left[(\alpha+1)(y_i-1)\tfrac{\Gamma(\alpha+1)\Gamma(\beta)\beta}{\Gamma(\alpha+\beta+1)} - \alpha y \tfrac{\Gamma(\alpha+1)(\alpha+1)\Gamma(\beta)}{\Gamma(\alpha+\beta+1)}\right]
\end{aligned} \tag{3}$$

The last line follows because $\Gamma(x) = (x-1)\Gamma(x-1)$. Also because

$$\frac{\Gamma(\alpha+1)\Gamma(\beta)}{\Gamma(\alpha+\beta+1)} = B(\alpha+1, \beta)$$

we can simplify (3) as

$$\frac{y\alpha(\alpha+1)B(\alpha+1,\beta)-\beta(\alpha+1)(y-1)B(\alpha+1,\beta)}{\alpha(\alpha+1)B(\alpha,\beta)}$$

$$= \quad \frac{yB(\alpha+1,\beta)-\frac{\beta}{\alpha}B(\alpha+1,\beta)(y-1)}{B(\alpha,\beta)} \tag{4}$$

By letting in the following one of the beta function's identities derived using the Gauss multiplication formula

$$\frac{\beta}{\alpha}B(\alpha+1,\beta)=B(\alpha,\beta+1) \tag{5}$$

formula (4) becomes

$$\frac{yB(\alpha+1,\beta)-(y-1)B(\alpha,\beta+1)}{B(\alpha,\beta)} \tag{6}$$

Substituting $B(\alpha,\beta)$ for $B(\alpha+1,\beta)+B(\alpha,\beta+1)$ gives

$$\frac{yB(\alpha+1,\beta)-(y-1)B(\alpha,\beta+1)}{B(\alpha+1,\beta)B(\alpha,\beta+1)}$$

$$= \quad y\frac{1}{1+\frac{B(\alpha,\beta+1)}{B(\alpha+1,\beta)}}-(y-1)\frac{1}{1+\frac{B(\alpha+1,\beta)}{B(\alpha,\beta+1)}} \tag{7}$$

Simple manipulation of the beta function's identity (5) yields

$$\frac{B(\alpha,\beta+1)}{B(\alpha+1,\beta)}=\frac{\beta}{\alpha}$$

Substituting it into (7) gives

$$y\frac{1}{1+\frac{\beta}{\alpha}}-(y-1)\frac{1}{1+\frac{\alpha}{\beta}}$$

$$= \quad \frac{y\alpha+(1-y)\beta}{\alpha+\beta} \tag{8}$$

The above solves Equation (4.14).

Bibliography

Agresti, A. (2002). *Categorical Data Analysis*. New York: Wiley-Interscience.

Akaike, H. (1980). The interpretation of improper prior distribution as limits of data-dependent proper prior distributions. *Journal of the Roral Statistical Society*, *42(408)*, 945–957.

Ansley, C. F. (1993). Calculating interval forecasts: Comment. *Journal of Business & Economic Statistics*, *11(2)*, 136–137.

Barnes, E. W. (1908). A new development in the theory of the hypergeometric functions. In *London Math. Soc.*, Volume 6 (pp. 141–177).

Barnett, V. (2004). *Comparative Statistical Inference*. New York: Wiley.

Bauwens, L., Lubrano, M. & Richard, J.-F. (1999). *Bayesian Inference in Dynamic Econometric Modeling*. Oxford England: Oxford University Press.

Beilken, S. L., Eadie, L. M., Jones, P. N. & Harris, P. V. (1990). Sensory and mechanical asseessment of the quality of frankfurters. *Journal of Texture Studies*, *21*, 395 – 409.

Bernardo, J. M. & Smith, A. F. M. (1994). *Bayesian Theory*. New York: Wiley.

Box, G. E. P. (1980). Sampling and bayes'inference in scientific modelling and robustness. *Journal of the Royal Statistical Society*, *143(4)*, 384–430.

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

Butler, R. & Rothman, E. D. (1980). Predictive intervals based on reuse of the sample. *Journal of the American Statistical Association*, *75(372)*, 881–889.

Campbell, G. & Hollander, M. (1982). Prediction intervals with a dirichlet-process prior distribution. *The Canadian Journal of Statistics*, *10(2)*, 103–111.

Chatfield, C. (1993). Prediction intervals. *Journal of Business and Economic Statistics*, *11(2)*, 121–135.

Chatfield, C. (1998). Prediction intervals. Technical report, University of Bath.

Chen, M.-H. & Deely, J. J. (1996). Bayesian analysis for a constrained linear multiple regression problem for predicting the new crop of apples. *Journal of Agricultural, Biological, and Environmental Statistics*, *1(4)*, 467–489.

Chhikara, R. S. & Guttman, I. (1982). Prediction limits for the inverse gaussian distribution. *Technometrics*, *24(4)*, 319–324.

Dahiya, R. C. & Guttman, I. (1982). Shortest confidence and prediction intervals for the log-normal. *The Canadian Journal of Statistics*, *10(4)*, 277–291. The Canadian Journal of Statistics is currently published by Statistical society of Canada.

Dalal, S. R., Fowlkes, E. B. & Hoadley, B. (1989). Risk analysis of the space shuttle: pre-challenger prediction of failure. *Journal of the American Statistics Association*, *84(408)*, 945–957.

David, H. A. (1981). *Order Statistics*. New York: John Wiley and Sons Inc.

Deming, E. W. (1975). Probability as a basis for action. *Am. Stat.*, *29*, 146–152.

Denison, D. G. T., Holmes, C. C., Mallick, B. K. & Smith, A. F. M. (2002). *Bayesian Methods for Nonlinear Classification and Regression*. West Succex, England: Wiley.

Devore, J. L. (2000). *Probability and Statistics for Engineering and The Sciences*. USA: Brooks Cole.

Diaconis, P. & Ylvisaker, D. (1985). Quantifying prior opinion. *Bayesian Statistics*, *2*.

Dietterich, T. G. (1998). Discussion: Arcing classifiers. *The Annals of Statistics*, *26(3)*, 838–841.

Draper, D. (1995). Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society, Series B(57)*, 45–97.

Dunsmore, R. (1974). The bayesian predictive distribution in life testing models. *Technometrics*, *16(3)*, 455–460.

Dzeroski, S. & Zenko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, *54(3)*, 255–273.

Faulkenberry, G. D. (1973). A method of obtaining prediction intervals. *Journal of the American Statistical Association*, *64(342)*, 433–435.

Fine, T. L. (2006). *Probability and Probabilistic Reasoning*. New Jersey: Pearson Education, Inc.

de Finetti, B. (1930). Funzione caratteristica di un fenomeno aleatoria. *Men. Acad. Naz. Lincei*, *4*, 86–133.

Fisher, R. A. (1930). Inverse probability. In *Proceedings of the Cambridge Philosophical Society*.

Gardner, E. S. (1988). A simple method of computing prediction intervals for time series forecasts. *Management Science*, *34(4)*, 541–546.

Gardner, M. J. & Altman, D. G. (1989). *Statistics with Confidence*. London: British Medical Journal.

Gauss, C. F. (1812). Disquisitiones generales circa seriem infinitam $\left[\frac{\alpha\beta}{1\cdot\gamma}\right] x + \left[\frac{\alpha(\alpha+1)\beta(\beta+1)}{1\cdot 2\cdot\gamma(\gamma+1)}\right] x^2$+etc. pars prior. *Commentationes Societiones Regiae Scientiarum Gottingensis Recentiores*, *2*. Reprinted in Gesammelte Werke, Bd. 3, pp. 123-163 and 207-229, 1866.

Gelman, A. (2004). *Bayesian Data Analysis*. Wiley.

Gill, J. (2002). *Bayesian Mehtods A Social and Behavioral Science Approach*. Chapmen and Hall.

Gregory, P. (2005). *Bayesian Logical Data Analysis for the Physical Sciences*. Cambridge England: Cambridge University Press.

Hahn, G. J. (1969). Factors for calculating two-sided prediction intervals for samples from a normal distribution. *Journal of the American Statistical Association*, *64(327)*, 878–888.

Hahn, G. J. & Meeker, W. Q. (1991). *Statistical Intercals: A Guide for Practitioners*. New York: Wiley-International Publication.

Hahn, G. J. & Shapiro, S. S. (1994). *Statistical Models in Engineering*. New York: Wiley.

Hamilton, H., Gurak, E., Findlater, L. & Olive, W. (2001). Overview of decision trees. Rudjer Boskovic Institute.

Holmes, C. C. & Adams, N. M. (2002). A probabilistic nearest-neighbour method for statistical patten recognition. *J. Roy. Statist. Soc.*, *B64*, 1–12.

Jaheen, Z. F. (2003). Bayesian prediction under a mixture of two-component gompertz lifetime model. *Text*, *12(2)*, 413–426.

Jaynes, E. T. (1957). How does the brain do plausible reasoning? Technical report, Stanford University Microwave Laboratory Report. Reprinted in Maximum Entropy and Bayesian Methods in Science and Engineering.

Konijn, H. S. (1987). Distribution-free and other prediction intervals. *The American Statistician*, *41(1)*, 11–15.

Kononenko, I. (2002). Machine learning for diagnosis: History, state of the art and perspective.

Kummer, E. E. (1836). über die hypergeometrische reihe. *J. reine angew. Math.*, *15*, 39–83 and 127–172.

Lee, P. M. (2004). *Bayesian Statistics: an introduction*. Hodder Arnold: a member of the headline group.

Leonard, T. & Hsu, J. S. J. (1999). *Bayesian Methods: An Analysis of Statisticians and Interdisciplinary Researchers*. Oxford England: Oxford University Press.

Lindley, D. V. (1965). *Introduction to Probability and Statistics from a Bayesian Viewpoint*. Cambridge England: Cambridge University Press.

Mojirsheibani, M. (1998). Iterated bootstrap prediction intervals. *Statistica Sinica*, *8*, 489–504.

Møller, B., Weedon-Fekjer, H. & Haldorsen, T. (2005). Empirical evaluation of prediction intervals for cancer incidence. *BMC Medical Research Methodology*, *5(21)*.

Nadeau, C. & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, *52*, 239 – 281.

Olive, D. J. (2006). Prediction intervals for regression models. Technical report, Dpartment of Mathematics, Southen Illinois University, Carbondale, IL 62901-4408, USA.

Phillips, P. C. B. (1979). The sampling distribution of forecasts from a first-order autoregression. *Journal of Econometrics*, *9*, 241–261.

Press, S. J. (2003). *Subjective and Objective Bayesian Statistics: principles, models, and applications*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Preston, S. (2004). Course: Introduction to statistics. State university of new york website.

Romano, C. L. D. N. P. J. P. (1992). Bootstrap technology and applications. *Technometrics*, *34(4)*, 378–398.

Ross, S. M. (2004). *Introduction to Probability and Statistics for Engineers and Scientists*. London: Elsevier Academic Press.

Saw, J. G., Yang, M. C. K. & Mo, T. C. (1984). Chebyshev inequality with estimated mean and variance. *The American Statistician*, *38*, 130–132.

Shannon, C. E. (1948). The mathematical theory of communication. *Journal of Bell Systems Tech.*, *(27)*, 379.

Shrestha, D. L. & Solomatine, D. P. (2006). Machine learning approach for estimation of prediction interval for the model output. Technical report, UNESCO-IHE Institue for Water Education.

Smith, A. F. M. (1986). Some bayesian thoughts on modelling and model choice. *The Statistician*, *35(2)*, 97–101.

Stine, R. A. (1985). Bootstrap prediction intervals for regression. *Journal of the American Statistical Association*, *80(392)*, 1026–1031.

Talor, J. W. & Bunn, D. W. (1999). A quantile approach to generating prediction intervals. *Management Science*.

Thombs, L. A. & Schucany, W. R. (1990). Bootstrap prediction intervals for autoregression. *Journal of the American Statistical Association*, *85(410)*, 486–492.

Wadsworth, H. M. (1998). *Handbook of Statistical Methods for Engineers and Scientists*. New York: McGraw-Hill.

Wikipedia (2006). *Prediction Interval*. http://en.wikipedia.org/wiki/Prediction interval.

Williams, W. H. & Goodman, M. L. (1971). A simple method for the construction of empirical confidence limit for economic forecasts. *Journal of Amer. Statist. Assoc.*, *66*, 752–754.

Willink, R. & Lira, I. (2005). A united interpretation of different uncertainty intervals. *Journal of Measurement*, *38*, 61–66.

Winkler, R. L. (1972). A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association* (pp. 187–191).

Wintle, B. A., McCarthy, M. A., Volinsky, C. T. & Kavanagh, R. P. (2003). The use of bayesian model averaging to better represent uncertainty in ecological models. *Conservation Biology*, *17(6)*, 1579–1590.

Witten, I. H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgen Kaufmann.

Yin, C. S., Chaw, L. T. & Yaacob, M. (2005). Hop-by-hop qos routing using statistical distribution-free approach. *Malaysian Journal of Computer Science*, *18(2)*, 28–37. Faculty of Computer Science & Information Technology University of Malaya 50603 Kuala Lumpur Malaysia email: sychan@perdana.um.edu.my tchaw@um.edu.my mashkuri@um.edu.my.