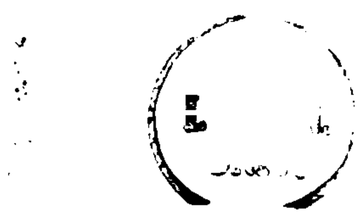
University of London
Imperial College of Science, Technology and Medicine
Department of Computing

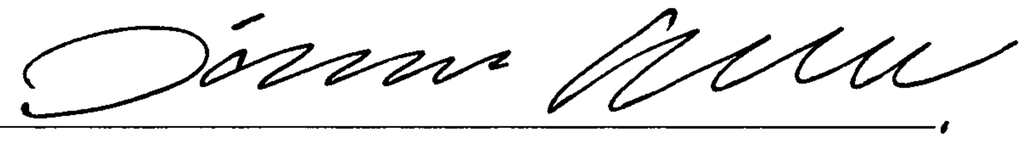
Knowledge Integration in Distributed Data Mining

Janjao Sutiwaraphun

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Engineering of the University of London and the Diploma of Imperial College, December 2000



I declare that the work presented in this thesis is my own except where it is stated otherwise.



Janjao Sutiwaraphun

Abstract

The integration of the results of several learning processes, namely knowledge integration or model integration, is an active research area in machine learning. The concept of model integration in machine learning can be applied to distributed data mining. However, there are two limitations in applying the available model integration approaches to integrate models in distributed data mining. The first limitation is that there are only a few approaches that focus on the comprehensibility of the output of the integration. The second limitation is that none of them addresses the characteristics of the local models that are learnt from data that are already distributed. This thesis addresses these two limitations in the design of Distributed Learning with Knowledge Probing (DLKP), the framework we propose for model integration in distributed data mining.

DLKP integrates models by learning from the data set whose class values are assigned by the combined predictions of the local models. Our study shows that the effectiveness of DLKP depends on the coverage of the local models on the instance space. To optimise the effectiveness of DLKP, we propose the use of a margin distribution graph as a tool for observing different forms of the coverage of the local models. Based on the analysis of the margin distribution graph, different scenarios of model integration with DLKP are presented. In the scenario where the local models are not suitable for integration, an adjustment in the local data distributions can improve the coverage of the local models. This thesis proposes two novel techniques, *data redistribution* and *data contraction*, for adjusting the local data distributions in that scenario.

With the use of the margin distribution graph and the two data distribution adjusting techniques, the accuracy of the results of DLKP is stable across a broad range of local models in distributed data mining. They are, at least, as accurate as the average accuracy of the local models, and are also comprehensible. An empirical study and a case study on real distributed data are presented to justify the design of DLKP, and its associated tools and techniques. The implementation issues of DLKP are also investigated.

Acknowledgements

First of all, I would like to thank Professor John Darlington for introducing me to Data Mining Group, Imperial College and for his kind support over the past few years. I owe my supervisor, Dr. Yike Guo, for his discussion, guidance, and constant support. No matter how busy, Yike always has time for a meeting with me when I request.

I would like to thank my pals, Jaturon Chattratichat, Chatchawan Pittayaratkul, Phuchphop and Rattachat Mongkolnavin, Isarez and Yeowarez Yongsunthon, Chanchana Meenakarn, Sasivimol Kittivoravitkul, and James Roca. I would like to thank them for being great companions in good and bad time, for their encouragement and advice, for their help in emergencies, and above all, for their friendship. I am truly appreciated it and hope that it will always remain the same.

My appreciation also goes to Sue Brookes for her caring especially on my physical and mental health. For me, Sue is a person who is warm and always understanding. I am so glad that we have her around in our section. I also would like to thank my colleagues and friends in the Data Mining Group for making a friendly and very enjoyable working environment. Thanks to Peter Au, Moustafa Ghanem, and Sarah Spence, in particular, for kindly proof-reading and giving many valuable comments on early drafts of this thesis.

I would like to acknowledge the support from friends in Thailand, Sasipa Chankaoropkhun, Sirirat Pavawongsak, Maneechai Chainuvati, and Monticha Komonbut, for their encouragement and for their friendship that has never been changed since we first met. I gratefully acknowledge the Royal Thai Government for the financial support during the course of my postgraduate studies.

Lastly, this thesis could not have been possible without the unconditional love and support from my family. I cannot thank them enough for all they have done for me in my life. I have a special thank to Bhumi, my four-year-old nephew. Thanks for his little wish to be someone like me. That is a great encouragement for creating a good piece of work.

To my parents,

Mongkol and Jurairat Sutiwaraphun

Contents

1 Introduction **18**

1.1 Data mining and machine learning 18

1.1.1 Machine learning: Learning as search 19

1.1.2 Data mining concepts 22

1.2 Distributed data mining: Why and how data are distributed ? . . 24

1.3 Model integration in distributed data mining 26

1.3.1 Why integrate models into one single model ? 26

1.3.2 Motivations of research 29

1.3.3 Objectives of study 30

1.4 Overview of the thesis 30

2 Research Background **32**

2.1 Introduction 32

2.2 Types and roles of inductive biases 33

2.3 Bias and variance decomposition 34

2.4 Model coverage 35

2.5 Margin and margin distribution graph 36

2.6 Active learning 41

2.7 Empirical evaluation approaches 42

2.7.1 Estimating hypothesis accuracy 42

2.7.2 Confidence intervals of the accuracy estimation 44

2.7.3 Comparing learning algorithms 45

2.8 Relations of the concepts in the formation of DLKP 48

3 Distributed Learning with Knowledge Probing: DLKP Framework **50**

3.1 Introduction 50

3.2 Distributed inductive learning concepts 51

3.3 Distributed learning with Knowledge Probing (DLKP) 53

3.4 Key issue of DLKP: Consistency of the local models 58

3.5 Four main scenarios of applying DLKP 62

3.6 Summary 65

4 Analysis of DLKP's Components **67**

4.1 Introduction 67

4.2 Local Model Learning phase 69

4.3 Knowledge Probing phase 70

4.3.1 Prediction scheme: A scheme to combine predictive ability of local models 71

4.3.2 Probing set selection approach 73

4.4 Techniques for adjusting local data distributions 81

4.4.1 Data redistribution technique 81

4.4.2 Data contraction technique 83

4.5 Summary 84

5 Empirical Studies **87**

5.1 Introduction 87

5.2 Framework of the empirical study 88

5.2.1 Data set selection 88

5.2.2 Parameter setting 88

5.2.3 Comparison criteria 89

5.2.4 Control of the experiments 90

5.2.5 Methods of analysis 90

5.3 Study of the effectiveness of DLKP 92

5.3.1 Design of the experiments 92

5.3.2 Margin distribution graph: The indicator of the effectiveness of DLKP 93
5.3.3 Improvement in accuracy by the analysis of model coverage 96
5.3.4 Summary of the effectiveness study 99
5.4 Bias and variance decomposition study 99
5.4.1 Design of the experiments 101
5.4.2 The experimental results 107
5.4.3 Analysis of biases and variances between the two learning phases of DLKP 109
5.4.4 Analysis of biases and variances in different scenarios . . . 112
5.4.5 Summary of the bias-variance decomposition study 113
5.5 Discussion on the results of the empirical studies 114
5.5.1 Model integration by selecting a final model 115
5.5.2 Model integration by Knowledge Probing 115
5.5.3 Model integration by Knowledge Probing after adjusting local data distributions 118
5.5.4 Comparison between DLKP and the centralised learning approach 121
5.6 Summary 123

6 Case Study **126**
6.1 Introduction 126
6.2 The problem domain: Heart disease diagnosis data 126
6.2.1 Data description 127
6.2.2 Why integrate the HEART DISEASE models ? 127
6.3 Approach of the study 130
6.3.1 Data preparation 130
6.3.2 Three classification algorithms used in the case study . . . 130
6.3.3 Design of the experiments 134
6.4 The effectiveness of DLKP on real distributed data 135
6.4.1 Analysis of margin distribution graphs 135

6.4.2 Improvement by the data distribution adjusting techniques 137
6.4.3 Effect of the stability of the learning algorithm on the data distribution adjusting techniques 138
6.5 Integration of models learnt from different learning algorithms . . 140
6.6 Discussion on the results of the case study 141
6.7 Summary 142

7 Implementation of DLKP 143
7.1 Introduction 143
7.2 KENSINGTON architecture 144
7.3 System requirements for implementing DLKP on KENSINGTON . 147
7.4 Implementations of DLKP on KENSINGTON and their associated costs 149
7.4.1 Preliminary evaluation process 150
7.4.2 First scenario: Arbitrarily selecting the final model 151
7.4.3 Second scenario: Direct Knowledge Probing 152
7.4.4 The third scenario: Knowledge Probing after applying the data redistribution technique 154
7.4.5 The fourth scenario: Knowledge Probing after applying the data contraction technique 155
7.5 Performance analysis 157
7.5.1 Effect of varying problem characteristics on DLKP 159
7.5.2 Cost comparison between DLKP and the centralised learning approach 160
7.6 Summary 161

8 Related Works 163
8.1 Introduction 163
8.2 Related works in the aspect of learning from data partitions . . . 164
8.2.1 Comparison to DLKP 167
8.3 Related works in the aspect of extracting the representation form of a model 168

8.3.1 Comparison to DLKP 170
8.4 Related works in the aspect of active learning 172
8.4.1 Comparison to DLKP 173
8.5 Summary 174

9 Summary, Conclusions and Further Work **176**
9.1 Contributions 176
9.2 Conclusions 177
9.3 Further works 179
9.4 Epilogue 181

A Glossary of Terms **183**
A.1 Data descriptions 183
A.2 Learning related terms 184
A.3 Model properties 185
A.4 Learning approaches 185
A.5 Experimental methodology 186
A.6 DLKP related terms 186

B Descriptions of data sets **188**
B.1 ABALONE data set 188
B.2 GERMAN data set 189
B.3 HYPOTHYROID data set 190
B.4 KR-VS-KP data set 191
B.5 LETTER data set 192
B.6 NURSERY data set 193
B.7 SATELLITE data set 194
B.8 SEGMENT data set 196
B.9 SICK data set 197
B.10 THYROID data set 197
B.11 LED7 data set 198
B.12 WAVEFORM data set 199

B.13 HEART data set 200

References **202**

List of Figures

1.1 The illustration of the spaces in machine learning problems and their relationships with inductive biases. 21

2.1 Examples of margin distribution graphs and locations of their slopes. 38
2.2 Two examples of the interpretation of the main slope in margin distribution graph. 40
2.3 A procedure for estimating the difference in error between two learning algorithms. 47

3.1 Abstractions of the distributed inductive learning from homogeneously and heterogeneously distributed data sets. 52
3.2 Abstractions of the process of Knowledge Probing from a black box model (a), and the process of Knowledge Probing from a set of local models and a prediction scheme (b). 55
3.3 Knowledge Probing concept. 56
3.4 The DLKP framework. 57
3.5 Improvement in the consistency of the local models by the data redistribution technique. 59
3.6 Abstractions of five scenarios of coverage of a model committee on the instance space. 61
3.7 Margin distribution graphs of model committees that have different forms of coverage on the instance space. 63

4.1 An illustration of the three main components of DLKP; Local Model Learning phase, Knowledge Probing phase, and techniques for adjusting local data distributions. 68
4.2 The relations between the Local Model Learning phase, the Knowledge Probing phase, and the techniques for adjusting local data distributions. 69
4.3 Abstractions of the areas in the instance space that are included in the probing set under the four probing set selection schemes. . 74
4.4 The algorithm of the data redistribution technique. 82
4.5 The algorithm of the data contraction technique. 85

5.1 Margin distribution graphs of the twelve model committees on the test sets. 95
5.2 Margin distribution graphs of model committees and the corresponding probed models before and after the adjustment in the local data distributions. 100
5.3 Learning curves of the twelve data sets. 103
5.4 Data preparation procedure of the bias-variance decomposition experiments. 105
5.5 The local model learning procedure and the centralised learning procedure for the bias-variance decomposition experiments. 106
5.6 The Knowledge Probing procedure for the bias-variance decomposition experiments. 107
5.7 Biases and variances of the Knowledge Probing phase (KP) plotted against the corresponding biases and variances of the Local Model Learning phase (LML). 110
5.8 Biases and variances resulted from both phases of DLKP plotted against the corresponding biases and variances of the centralised learning approach (CL). 111
5.9 Margin distribution graphs of each model committee in the second scenario and its corresponding probed model. 117

5.10 The changes in the inconsistency level and the relabelling percentages of ABALONE, GERMAN and LED7 data sets during the data distribution adjusting process. 120
5.11 Comparison of the margin distribution graphs of the probed models and the corresponding centrally learnt models. 122

6.1 Margin distribution graphs of the model committees that are learnt by C4.5, NBC and PEBLS. 136
6.2 Margin distribution graphs of the model committees and the probed models before and after the adjustments in local data distributions. 139

7.1 Kensington tree-tier client-server components and their interaction. 145
7.2 A screen shot of learning with a model committee on KENSINGTON. 147
7.3 An example of KENSINGTON configuration for co-operating data mining process between two distributed sites. 148

8.1 An illustration of an arbiter and a combiner with n local models. . 165

List of Tables

1.1 A list of data mining algorithms and their tasks. 23

1.2 The comparison of the corresponding terms between data mining and machine learning. 24

4.1 Comparison of the sizes of the probing sets under the four probing set selection schemes. 76

4.2 Percentages of the sizes of the probing sets under the four probing set selection schemes in relative to the sizes of the available training data. 77

4.3 Comparison of the classification error of the probed models of the four probing set selection schemes. 78

5.1 Details of the data sets used in the experiments. 89

5.2 Comparison between the classification error of each probed model and the average classification error of the corresponding local models. 94

5.3 Analysis of the slope of the twelve model committees. 96

5.4 Comparison of classification error between the average error of the local models, the probed model and the arbitrarily selected model of the data set whose local models are consistently accurate. . . . 98

5.5 Comparison between the classification error of each probed model after the adjustment in local data distributions and the average classification error of the corresponding local models. 99

5.6 The characteristics of the four groups of model committees and the improvement in classification errors by DLKP and its associated tools and techniques over the average errors of the corresponding local models. 101

5.7 Comparison of $bias^2$ and variance between the Local Model Learning phase and the Knowledge Probing phases. 108

5.8 Comparison of $bias^2$ and variance between the centralised learning approach and DLKP. 109

5.9 Summary of the average bias and variance of data sets in different scenarios. 112

5.10 Summary of classification errors in three different stages of DLKP framework. 114

5.11 The percentage of disagreement in the predictions between the local models and the corresponding model committee on the probing set and the test sets. 116

5.12 Percentage of relabelled instances during the Knowledge Probing phase. 118

5.13 Comparison of the average error of the local models, the error of the probed model before and after the adjustment in local data distributions in the scenario where margin distribution graphs of model committees are relatively poor. 119

5.14 Comparison of classification error between the final models of DLKP and the corresponding centrally learnt models. 123

5.15 Comparison of the model sizes between the probed models and the corresponding centrally learnt models. 124

6.1 The locations and names of the principle investigators responsible for the collection of the HEART DISEASE data. 127

6.2 Class distributions of the four data sets of the HEART DISEASE data. 128

6.3 The percentages of the missing values in the four HEART DISEASE data sets. 128

6.4 The classification errors of the local models that are learnt by a decision tree learning algorithm (C4.5), a naive Bayes classifier (NBC) and a k-nearest neighbour algorithm (PEBLS). 129
6.5 Comparison of the classification errors in different stages of DLKP before and after the adjustment in local data distributions. C4.5, NBC and PEBLS are in turn used as the local learning algorithm. C4.5 is used as the probing algorithm. 137
6.6 Comparison of the classification errors in different stages of DLKP. The first two local models are learnt by PEBLS and the other two local models are learnt by NBC. The probed model is learnt by C4.5. 140

7.1 Symbols used in the analysis of computation and communication costs. 150
7.2 The computation and communication cost of the preliminary evaluation process. 151
7.3 The computation and communication cost of the scenario of directly applying the Knowledge Probing phase. 153
7.4 Computation and communication cost of the scenario where the data redistribution technique is applied before the Knowledge Probing phase. 156
7.5 Computation and communication cost of the scenario where the data contraction technique is applied before the Knowledge Probing phase. 158
7.6 Costs of the four main scenarios with respect to the characteristics of the problem domain. 159

Chapter 1

Introduction

This chapter begins with an introduction of the concept of machine learning as search and its relation to data mining concepts, which will be the basis of the discussions in this thesis. Then, we discuss the requirements of distributed data mining and different types of distributed data. Next, the requirements of model integration in distributed data mining, which are the motivations of this research, are discussed, followed by an overview of the thesis.

1.1 Data mining and machine learning

Data mining can be considered as a new interdisciplinary research area. It involves many well-established research fields, such as artificial intelligence, statistics, pattern recognition, databases, knowledge acquisition for expert systems, data visualisation and *machine learning*. Data mining and machine learning are closely related in theories and algorithms for extracting patterns from data. Data mining, however, focuses on the extension of applications of machine learning techniques to large sets of real world data [40].

In the next subsection, the fundamental concept of machine learning, learning as search, is introduced. This is followed by the introduction of the data mining concept. The relations between data mining and machine learning are discussed at the end of this section.

1.1.1 Machine learning: Learning as search

Machine learning addresses the task in building computer programs that improve their performance through experience [72]. Mathematically, a learning process can be viewed as an inductive procedure. Given D as a set of instances (x, y) that contains a target function f, $y = f(x)$, a learning task is defined as a task of approximating the target function f from the data set D. The approximation of f is called a model. The model is expected to aid the understanding of the problem domain of D, or to be used to predict the value y of future instances, or both. D is usually referred to as a training set.

A learning process consists of four main components: a training set, a target function, a representation form of the target function, and a learning algorithm. In general, during a learning process, the learning algorithm repeatedly produces hypotheses that are expressible in the specified representation form, verifies them with the training set, and modifies the hypotheses such that they fit the training set better. The output is the hypothesis that best fits the training set with respect to the learning algorithm. Thus, if we consider all possible training sets as an *instance space*, consider the training set as the *target area*[1] in the instance space, and consider all possible hypotheses as a *hypothesis space*, the learning process can be viewed as a *search* through a large hypothesis space to find the hypotheses that best fits the target area in the instance space. The coverage of these hypotheses over the instance space is referred to as *coverage of the model.* The instance space is defined by number and types of attributes that form an instance, and the hypothesis space is defined by the representation form that is allowed to specify the target function.

In practical applications, however, the entire instance space is often so large that a given training set is viewed as only a small fraction of it [52]. From this small fraction, a model has to be formed such that it can be generalised to the entire instance space. Similarly, the hypothesis space is often too large

[1]The area that contains the target function.

to be thoroughly explored in reasonable time. Thus, if no restriction is placed on the hypothesis space, and no preference criterion is supplied for comparing competing hypotheses, then all possible hypotheses are equally possible, and, in average, no learning method can do better than random guessing [71]. Therefore, *inductive biases*[2] are introduced to cut down the search space. Note that the bias in this context is different from the term "bias" in Statistics. Inductive bias means the preference of exploring one particular area over the others. A good example of inductive biases is the bias used in the well-known C4.5 decision tree learning algorithm [79]. C4.5 uses the minimum description length principle [81] for building a decision tree. According to this principle, the best model is the one that minimises the total number of bits needed to encode the model and the data. C4.5 is, therefore, inherently biased towards the smallest tree (a tree that has a minimum depth).

Figure 1.1 is the illustration of the instance space, the hypothesis space, the coverage of the model, and their relationships with inductive biases. Figure 1.1 shows that the instance space (Layer (a) in Figure 1.1) can be of any shape, and the given training set (Layer (b) in Figure 1.1) is normally only a fraction of it. In a learning process, the type of representation allowed to specify an output model restricts the possible search area in the instance space. For example, the instance space in Figure 1.1 has a curly shape, but the hypothesis space (Layer (c) in Figure 1.1) is only able to represent it in a rectangular shape. Therefore, the hypothesis space cannot exactly represent the instance space. This type of limitation is actually a bias that is inherent in the output form of the learning algorithm and is called a *representation bias*. The space that is representable under this bias is referred to as a hypothesis space. During the search, a *preference bias* is introduced to narrow down the search space by making some hypotheses more preferable than others. For example, the preference bias in C4.5 is the preference for the smallest tree. Thus, given two fractions of hypothesis space, C4.5 will search on the fraction that leads towards a smaller tree. An output

[2]Inductive biases will be further discussed in Section 2.2.

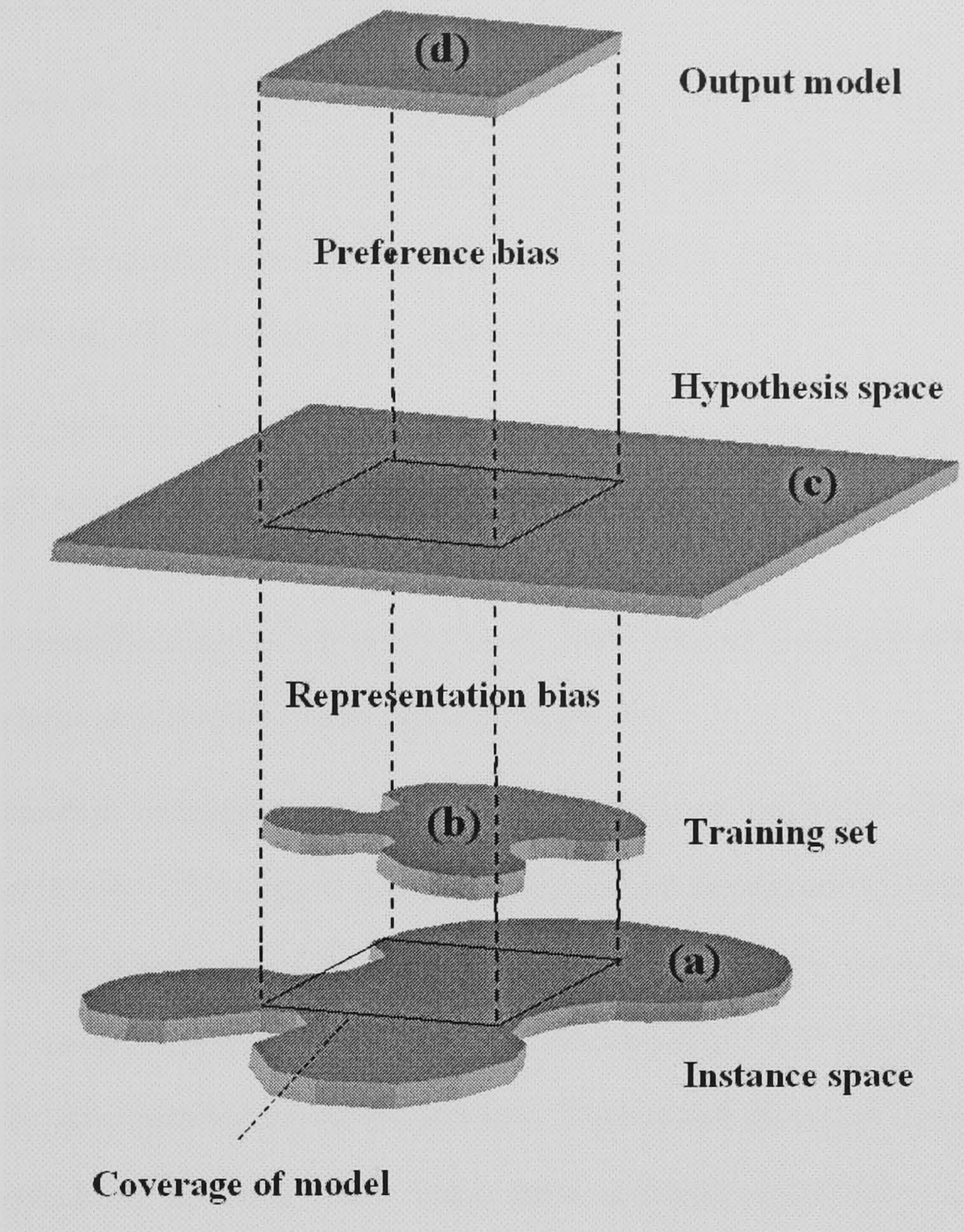


Figure 1.1: The illustration of the spaces in machine learning problems and their relationships with inductive biases.

model (Layer (d) in Figure 1.1) is the result of a search in the hypothesis space. This is a fraction in the hypothesis space that best fits the given fraction of the instance space (the training data). The coverage of the model over the instance space (the projection in Figure 1.1) indicates how well the model represents the target function underlying in the instance space. These concepts will be used as the basis of the discussions in this thesis.

1.1.2 Data mining concepts

Data Mining is defined by Fayyad et al. [40] as

> A step in the KDD (Knowledge Discovery in Databases) process consisting of particular data mining algorithms that, under some acceptable computational efficiency limitations, produces a particular enumeration of patterns over data sets.

The space of patterns is often infinite, and the enumeration of patterns involves some forms of search in this space.

Each data mining algorithm consists of three main components, *model representation*, *model evaluation*, and *search methods*.

- **The model representation** is the language used for describing discovered patterns. The representation should be expressive enough to represent all possible patterns of a particular data set. If the representation is limited to certain types of expressions, its capacity for producing an accurate model is also limited to some extent. One good example is the decision tree representation in C4.5 [79]. C4.5 uses a single attribute as a test in each internal node of the tree. As a result, the class regions (in the hypothesis space) produced by a resulted tree are all hyperrectangles. Therefore, they cannot perfectly represent class regions in the instance space that are not hyperrectangles, no matter how much training data are given. A powerful representation, i.e. the language that can be mapped to most forms of regions in the instance space, however, also has the disadvantage of increasing the chance of overfitting the training data. In addition, the search under such a representation is more complex, and the interpretation of the resulted model is often more difficult.

- **Model evaluation** estimates how well a particular pattern satisfies the criteria of interest. The evaluation mainly involves *efficient properties*, such as predictive accuracy, or *descriptive properties*, such as understandability of the model.

Algorithms	**Tasks**
Classification	To learn a function that maps (classifies) a data item into one of several predefined classes
Regression	To learn a function which maps an instance to a prediction of real-value variables
Clustering	To identify a finite set of categories or clusters to describe the data
Dependency Modelling	To find a model which describes significant dependencies between variables
Change and Deviation Detection	To discover the most significant changes in the data from previously measured or normative values
Summarisation	To find a compact description for a subset of data

Table 1.1: A list of data mining algorithms and their tasks.

- **Search methods** can be categorised into *parameter search* and *model search*. In parameter search, given a fixed model skeleton and observed data, the algorithm has to search for the parameters that optimise model evaluation criteria. An example of the parameter search method is the back propagation algorithm for neural networks where the topology of the networks is specified before the learning process and the weights within the topology are adjusted during the learning process. On the other hand, model search can be viewed as a loop over the parameter search. In each iteration, a model is formed and the parameter search method is used to optimise the model. However, since a number of possible models often prohibit exhaustive search, an heuristic technique is often used in the implementation of model search.

Data mining algorithms can be roughly categorised as in Table 1.1 [40]. It should be noted that this thesis mainly focuses on classification data mining algorithms. Therefore, the approaches that will be described later in this thesis are only suitable for distributed classification data mining environments. Data mining algorithms usually generate the output in two forms, the predictive form and the descriptive form. The predictive form is a model that is able to give predictions on unseen instances. The descriptive form is a set of descriptive

Data Mining	Machine Learning
Hypothesis	Pattern
Hypothesis space	Space of patterns
Representation bias	Model representation
Preference bias	Heuristic technique

Table 1.2: The comparison of the corresponding terms between data mining and machine learning.

patterns that can aid human understanding of the problem domain.

Data mining is closely related to machine learning in that they involve some forms of search in the space of possible answers. Also, to make the search possible with reasonable resources, biases are introduced in the representation form of the output and in the search method. The comparison of the corresponding terms between data mining and machine learning is shown in Table 1.2.

In the next section, distributed data mining is introduced. The requirements of distributed data mining and different types of distributed data are discussed.

1.2 Distributed data mining: Why and how data are distributed ?

Data in data mining applications can be distributed for several reasons:

- **Distributed to speed up the mining process** Data mining applications tend to deal with relatively large data sets, and their execution time is often a function of the size of a training set. Thus, the centralised learning approach[3] where data is stored at one physical location may not provide the result in acceptable time. In this scenario, data is distributed such that subsets of data can be mined concurrently, resulting in speed up of the whole mining process.

[3]For the rest of the thesis, we will refer to a learning process that is performed on a complete set of data and gives a single model as an output as the centralised learning approach. We will refer to its output as the centrally learnt model.

- **Distributed to avoid the memory limitation problem** Most of data mining algorithms require a complete set of data in memory during the mining process. Thus, no matter how large a memory a computer has, there is always a limitation in its physical size. Distributed data mining provides a viable solution to the problem of limited memory space.

- **Distributed by nature** In this scenario, data is distributed because of the physical nature of the problem. For example, one international company may have several branches worldwide, and each branch contains its own data locally. One way of using these distributed data is to combine the data and mine them at one single site. However, the combination of these data sets can be restricted by several difficulties such as insufficient bandwidth, high communication overhead, or limited storage space. For these reasons, data may be kept distributed and made accessible via the networks. Therefore, an efficient way of using these distributed data is to locally mine the data at each site and integrate the mining results.

In summary, we can view the key word *"distributed"* in two different ways:

- **Distributed by partitioning** In this sense, the original data set exists as one large piece. It is divided into smaller subsets by using data partitioning techniques.

- **Distributed by nature** The data is physically distributed at different sites and sometimes data sharing is not permitted due to confidentiality or cost of data migration.

In the scenario where data are distributed by partitioning, distributed data mining constitutes a scaling mechanism for learning from large data sets. Since the execution time of a learning process can be prohibitive when size of data is very large, data partitioning techniques can be applied to break down a large data set into small subsets. The partitioning can be carried out horizontally ([18, 77]) by partitioning the data into several subsets of instances, or vertically ([54]) by

partitioning the data into several sets of different attributes. These subsets are regarded as distributed data sets that can be learnt concurrently with a distributed/parallel computing system. The learnt results can be integrated into a single model that represents the knowledge learnt from the entire data set.

In the second scenario, distributed data mining is a mechanism for distributed information comprehension. That is, we learn a global knowledge from distributed data where the distribution of each partition is unknown in advance and can be in any shape ranging from evenly distributed to highly unbalanced. This learning procedure involves integrating models that are learnt from these local data in a pragmatically and statistically sound fashion.

The difference in understanding distributed data mining results in the diversity of research. However, there are two main characteristics that both scenarios have in common:

1. Learning local models from parts of the data.
2. Approximating a global model by integrating the local models.

Therefore, the major challenge in distributed data mining is to learn a high quality global model from local models that are learnt from different parts of data.

1.3 Model integration in distributed data mining

To give an overview of the problem that motivates this thesis, this section discusses the requirements of model integration in distributed data mining, followed by the motivations and the objectives of this thesis.

1.3.1 Why integrate models into one single model ?

There are at least four reasons why models that are the results of distributed learning processes should be integrated into one single model:

- The requirement of an integrated model to understand global knowledge of the problem domain
- The requirement to learn from larger amount of data for improving the accuracy of the learning
- The requirement of a model that is convenient to use
- The requirement of a model that is not only accurate, but also provides insight into the problem domain.

The first reason stems from physical nature of distributed data mining applications. While a local model is useful for making local decision and aiding the understanding of the local problem, a global model is also useful for making a global decision and aiding the understanding of the problem as a whole. An example of this scenario is data mining applications for studying consumer behaviour in a multinational business. It is convenient for each local site, i.e. each branch of the company, to maintain its own data and produce a model for local use. However, to study consumer behaviour on a larger scale i.e. European region, there is a need for a global model. To generate a global model by integrating existing models is less cumbersome than by gathering all local data and learning them at one single site.

The second reason for the need of model integration is the requirement to learn from larger amount of data. In practical applications, it is unlikely that the available training data will be sufficient to fully determine every required generalisation of the problem domain [70]. Model integration provides an indirect way of learning from larger amount of data.

Intuitively, learning with more data should lead to a more accurate model. However, Oates & Jensen [74] show that increasing the amount of training data often results in a linear increase in model size, even when that additional complexity results in no significant increase in model accuracy. Therefore, the increasing amount of data in the context of model integration should be interpreted in

qualitative rather than in *quantitative* terms. A duplicate set of training data is considered useless for a learning process, no matter how large it is. In qualitative terms, increasing the amount of data should mean the additional data lead to a generation of patterns that are not already existing in the original set of data. The integration of such patterns can be seen as a way of compensating for the imperfect coverage of other sets of patterns in the integration. Nonetheless, there is also no guarantee that the integration will always yield a model of higher accuracy than each local model. The general claim that combining models is always helpful is fallacious [67]. It is only beneficial in some cases. This is because the quality of an integrated model depends on the quality of the contributing models. An integration of a group of similar models will result in no better accuracy than an arbitrarily selected model. Thus, there is a need for a criterion to determine the scenario where an integration of models is worthwhile.

It can be concluded that although model integration does not always result in better accuracy, at least, it increases the probability of finding a better quality model. Model integration can be carried out efficiently and usefully when applied together with a proper integration criterion.

The third reason for the requirement of model integration is the fact that although a global prediction can be derived from a set of local predictions, the prediction process is cumbersome. This is because predictions from every local model are required to make a single combined prediction. Thus, large amounts of computation is required to make each combined prediction. This is especially important in environments where local models are located in different locations.

Another important issue is the comprehensibility of models. When a set of local models is used to make predictions, the prediction process does not provide insight into how the prediction is made. For example, human users can often interpret a single decision tree, but an ensemble of multiple decision trees is much more difficult to understand. Comprehensibility enables and improves the validation of the learning process. In some applications where the cost of misclassification is

crucial, such as medical applications, gaining insight into how the model derives the prediction is highly important. The users of such a system must understand how the system makes its decision in order to be confident in the predictions. In addition, the goals of most learning applications are to gain insight into the problem domain rather than just to make predictions [55]. Integrating local models into one single comprehensible model aids the understanding of how global predictions are made.

1.3.2 Motivations of research

The discussion in Section 1.3.1 shows that there are needs for model integration in distributed data mining. Most of the research in model integration [11, 4, 34, 6] has been conducted in the area of multiple model learning which is an active research field in machine learning. Multiple model learning concerns learning from models that are generated by means of variations in either the distributions of the data or the inductive biases of the learning algorithm. These models are combined to make joint predictions on unseen data. Both multiple model learning and distributed data mining deal with learning from multiple models. However, the semantics of base models in multiple model learning are different from those of local models in distributed data mining. Base models in multiple model learning are generated from the same data source. These base models, therefore, are about one *whole single* source of data. In distributed data mining, local models are about the *partial* data in the case where data are distributed by partitioning, or about the *multiple* sources of data in the case where data are distributed by nature. Thus, the level of control in the distributions of data between the two paradigms are very different. Therefore, the existing model integration approaches in multiple model learning cannot be applied to distributed data mining. New techniques for model integration in distributed data mining should be investigated.

1.3.3 Objectives of study

This thesis focuses on model integration in distributed data mining, especially in the environment where data is distributed by nature. Because data in such an environment can be distributed in a variety of forms and can lead to different types of local models, this thesis aims at developing a framework of model integration for different types of local models in distributed data mining. The integration is aimed at generating a single comprehensible model that can well represent all the local models.

In respect to the goal of the proposed framework, the integrated model should, at least, have the same accuracy as the average accuracy of the local models. Theoretically, the integration should aim at integrating the coverage of each local models together, and result in the most accurate model. Practically, this aim is not always possible to achieve. This is due to the introduction of inductive biases to the learning algorithm. While increasing the efficiency of the search, these biases also pose a limitation on the capability of the searching process, especially if the biases do not fit with the problem domain. Hence, it is not always possible to represent the whole instance space with only one single model. It will be more practical if the integration is aimed at integrating the main area of the coverage of the local models, in particular, the area where their coverage is overlapped. Therefore, it is reasonable to set a preliminary goal that the integrated model should, at least, have the same accuracy as the average accuracy of the local models.

1.4 Overview of the thesis

In Chapter 2, the research background of the study is reviewed. This includes the concepts behind the development of the proposed framework, Distributed Learning with Knowledge Probing (DLKP), the mechanism to differentiate different scenarios of model integration in distributed data mining, and the methodologies used in the empirical studies.

In Chapter 3, DLKP is introduced. Key issues in the formation of the framework are discussed. Based on these issues, four main scenarios of applying DLKP are described.

Chapter 4 is the review of three major components of DLKP. Since DLKP involves many design choices, the discussion also focuses on the design choices chosen in the empirical study.

DLKP is empirically investigated in Chapter 5. Details of two main experiments, study of the effectiveness of DLKP and bias-variance decomposition study, and their results are presented to justify the approach.

In Chapter 6, we study the effectiveness of DLKP on real distributed data. We also investigate the applicability of different classification algorithms in DLKP, and explore the possibility of applying DLKP to local models that are learnt by different classification algorithms.

Chapter 7 provides an example of the application of DLKP on a real distributed data mining system, the KENSINGTON system. System requirements for implementing DLKP on KENSINGTON are discussed. The applications of the four main scenarios of DLKP on KENSINGTON are illustrated. Their associated costs are analysed correspondingly. Performance issues of DLKP are investigated.

The related works in the areas of learning from data partition, extracting the representation form of model, and active learning are reviewed in Chapter 8. These works are compared and contrasted to the corresponding components in DLKP.

Chapter 9 is the summary of the thesis. The major contributions are summarised. The conclusions of the study are presented. Some possible directions of further work are also discussed.

Chapter 2

Research Background

This chapter reviews the technical backgrounds that are involved in the formation and evaluation of Distributed Learning with Knowledge Probing (DLKP), the framework proposed in this thesis for model integration in distributed data mining.

2.1 Introduction

This chapter first introduces types and roles of inductive biases in the learning process. Since inductive bias and variance of a learning algorithm constitute the error of the output of the learning process [44], a technique to decompose the bias and variance from classification error is presented. After that the concept of model coverage is described. It shows how well a model represents the target area in the instance space. Next, the margin of predictions and its distribution graph, which are the tools we use for observing the form of the coverage of the local models in distributed data mining, are introduced. This is followed by a discussion on active learning, which is the concept we used to optimise the coverage of the local models in the case where it is not suitable to be integrated. After that, the approaches that we use for empirically evaluating DLKP in Chapter 5 are described. Then, the relation of these concepts to the design and empirical study of DLKP is presented.

2.2 Types and roles of inductive biases

As mentioned in Section 1.1.1, inductive learning is a search in the large hypothesis space for general hypotheses that fit the training data. An important element in inductive learning is inductive bias, which is a predisposition for some solutions over others. Most learning algorithms inherently have at least some form of inductive bias. Without it, all possible functions must be entertained as hypotheses. Taken together, these functions predict all possible future outcomes as equally likely, so they cannot provide a basis for generalisation or prediction [72]. Generally, there are two forms of inductive bias.

The first bias is a *representation bias.* This is the bias that restricts the hypothesis space to only patterns that can be represented in a specified form of representation. It is inherently introduced in the representation form of the output of the learning algorithm. Consequently, no matter how well an area in the instance space represents a target function, if that area is not representable by the representation form of the learning algorithm, the search path of that algorithm will never reach that area. Therefore, if the representation bias is unsuitable for the problem domain, the accuracy of the result of the learning process will be limited.

The second bias is a *preference bias.* This is the bias that introduces a preference ordering in the hypothesis space. An example of this bias is a preference of simple patterns in the given representation form, or a preference of patterns that have a lower misclassification cost. Preference bias constructs a particular search path in the hypothesis space that is expected to contain the unknown target function. Due to this fact, a preference bias is typically more desirable than a representation bias because the latter introduces the possibility of excluding the unknown target function. Nonetheless, it is often the case that some learning algorithms employ both forms of biases. An example of such an algorithm is C4.5 [79]. C4.5 uses a decision tree as the representation form. The decision tree representation restricts the output model of C4.5 to only patterns that are parallel to the attribute axes. At the same time, its preference of the higher

information gain attribute over the lower one, introduces a preference bias to the search process.

Therefore, it can be concluded that inductive biases are essential to the learning process. However, improper use of the inductive bias, such as the mismatch between inductive bias and problem domain, can increase the error of the output model.

2.3 Bias and variance decomposition

The error of a classification learning process can be decomposed into three quantities; the error that is caused by a bias of the learning algorithm, the error that is caused by a variance of the learning algorithm to changes in the training data, and the error that is caused by noise in the training data [44]. A decomposition of these elements aids the understanding of the behaviour of the learning algorithm. Several techniques for bias-variance decomposition have been proposed in machine learning and data mining publications. It includes Kong & Dietterich [59], Kohavi & Wolpert [58], and Breiman [12]. The definitions of bias suggested in Kong & Dietterich [59] and Breiman [12] are for binary classification. Also, their definitions of variance can give negative value. Thus, the decomposition approach used in this thesis is based on Kohavi & Wolpert's approach [58], which can be described as follows:

Given a fixed target function and training set size, the expected classification error is comprised of three non-negative quantities:

- *Squared bias ($bias_x^2$)*: This quantity measures how much the prediction of a learning algorithm (average over all possible training sets of the given training set size) deviates from the target (the actual class values of the training data).

- *Variance ($variance_x$)*: This quantity measures the sensitivity of a learning algorithm to the difference in the training set of the given size.

- *Intrinsic target noise (σ_x^2)*: This quantity is a natural noise hidden in the training set. The noise is independent of the learning algorithm. This is a lower bound of the expected error of any learning algorithm.

Let X be the input space with cardinality $|X|$, and Y be the output space with cardinality $|Y|$.

The "target" f is a parametric function mapping X to Y, for all $x \in X$ and $y \in Y$, $y = f(x)$, and if f is a noise-free function from X to Y, for any fixed x, we have $P(f(x) = y) = 1$ for one value of Y, and 0 for all others.

The "model" $\hat{f}$ generated by a learning algorithm is an approximation of f. Also, for any fixed x, $P(\hat{f}(x) = y) = 1$, for one value of Y, and 0 for all others.

The expected classification error, $E(C)$, can be defined as:

$$E(C) = \sum_x P(x)(bias_x^2 + variance_x + \sigma_x^2)$$

where

$$\begin{aligned} bias_x^2 &\equiv \frac{1}{2}\sum_{y \in Y}[P(f(x) = y) - P(\hat{f}(x) = y)]^2 \\ variance_x &\equiv \frac{1}{2}\left(1 - \sum_{y \in Y} P(\hat{f}(x) = y)^2\right) \\ \sigma_x^2 &\equiv \frac{1}{2}\left(1 - \sum_{y \in Y} P(f(x) = y)^2\right) \end{aligned} \tag{2.1}$$

Classification error is a quantitative approach to measure the quality of the model. The quality of the model can also be observed via the form of its coverage over the instance space. The concept of model coverage is introduced next.

2.4 Model coverage

As discussed in Section 1.1.1, the generalisation implicit in a learning process can be viewed as a search in the space of possible hypotheses, and an output model can be viewed as an area in the hypothesis space that best represents the training instances. We refer to the coverage of this area over the instance space as model coverage. The characteristics of the coverage, such as the confidence that the

model has in its coverage, show how well the model represents the target area in the instance space.

Model coverage can be used as an additional heuristic to make model integration more effective. One study in employing model coverage as a heuristic in the model integration process is conducted by Brodley & Lane [14]. They approximate the coverage of models in the integration from the sum of two quantities, the percentage of the test data that two or more of the models classify correctly and the percentage in the test data that each individual model classifies correctly. In this thesis, in addition to the percentage of the coverage, we are also interested in other characteristics of the coverage, such as the confidence and the form of the coverage. This is because, by chance, the model may cover some areas that are not the target area, as a result of variation in training data. Thus, no matter how large the model coverage is, it is more important to measure the confidence of the model in its coverage. This is especially important for local models in distributed data mining where local data tend to vary. Thus, in order to make the integration effective, the integration should only include the area where the local models are confident in their coverage. To observe the confidence of the local models in covering the instance space, we propose the use of a margin distribution graph, which is discussed in the next section.

2.5 Margin and margin distribution graph

In classification problems, the concept of weight is often introduced to solve conflicts when making a prediction. Given an instance x with $1, \ldots, k$ possible classes, a model M assigns weight w_i, $i = 1, \ldots, k$ to each class of x such that the higher the weight w_i, the more likely the model M will predict class i as a prediction. Assume that the weights of all classes are normalised so that they add up to one, the classification *margin* is defined as the difference between the weight assigned to the correct class and the maximal weight assigned to any single incorrect class [82]. Given an instance x with $1, \ldots, k$ possible classes, a model M predicts each class of x with weight $w_i, i = 1, \ldots, k$ such that $\sum_{i=1}^{k} w_i = 1$, if

the actual class of x is c, the margin of the prediction with respect to M is:

$$margin_M(x) = w_c - w_x,\ w_x = \overset{k}{\underset{i=1}{MAX}} w_i,\ i \neq c. \tag{2.2}$$

The value of margin, thus, ranges between -1 and 1. Since the class of the highest probability is often chosen as a prediction, the margin is *positive* if the model makes a correct prediction. In this case, the value of margin represents the confidence of the model in its prediction. The higher the margin, the higher the confidence the model has in its prediction. The margin is *negative* when the model makes an incorrect prediction. In this case, the margin represents a penalty of the wrong prediction. The more negative the margin, the more inaccurate the model is in predicting the class value of that particular instance.

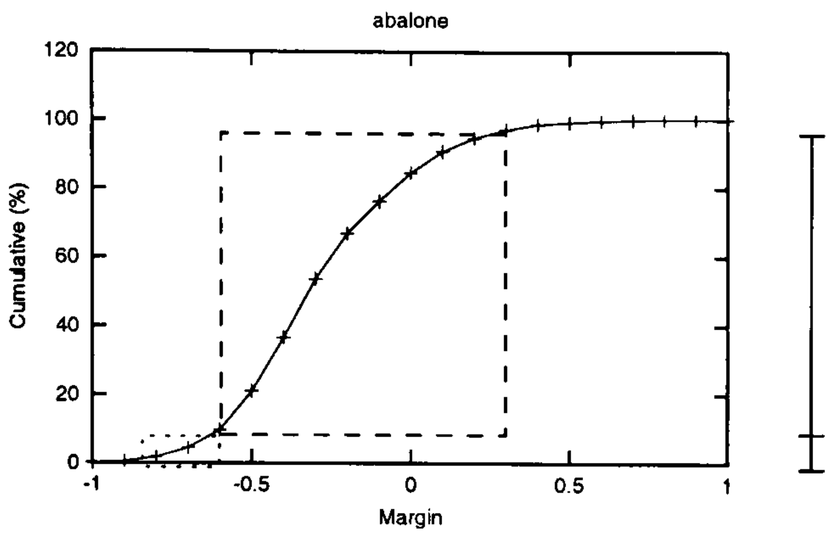
The use of *margin* can be applied to a set of models, where the models are used collaboratively in making predictions. That is, in each prediction, the models first give class predictions of an instance. These class predictions are combined into a single prediction by a prediction scheme, such as uniform voting[1]. After being combined by the prediction scheme, the probability of each class is the sum of the weights that each model gives to that class. These weights are normalised so that they add up to one. Given an instance x with k possible classes, a set of models $\mathcal{M}$ consists of n models, and each model M in $\mathcal{M}$ predicts each class of x with weight $w_{1M}, \ldots, w_{kM}$ such that $\sum_{i=1}^{k} w_{iM} = 1$, if the actual class of x is c, the margin of the prediction with respect to $\mathcal{M}$ is:

$$margin_{\mathcal{M}}(x) = w_c - w_x,\ w_x = \overset{k}{\underset{i=1}{MAX}} w_i,\ i \neq c,\ \text{where } w_i = \frac{1}{n}\sum_{j=1}^{n} w_{ij}. \tag{2.3}$$

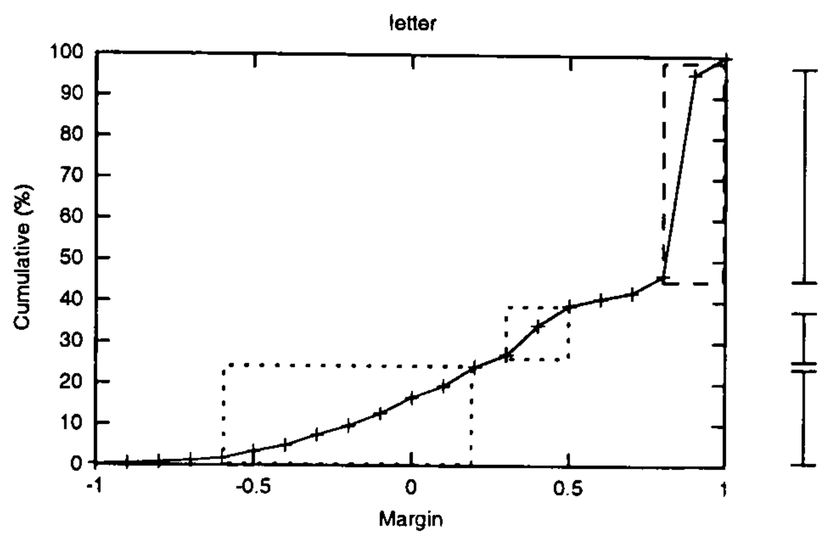
Thus, *margin* can be used to analyse two types of models:

- *Analysing a prediction of a single model* Given a set of weights that a model assigns to all possible classes in a single prediction, the margin of that prediction can be computed. The margin shows the predictive ability

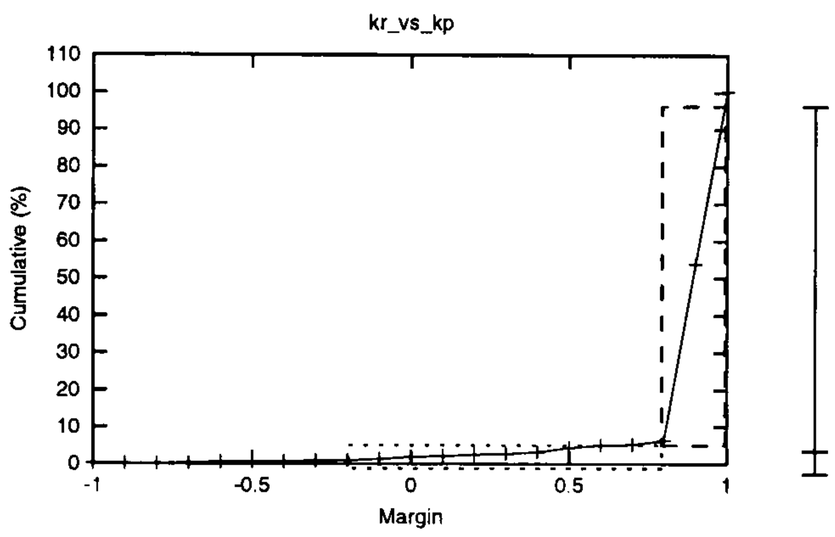
[1]In uniform voting, each model gives only one class prediction. The class that is most frequently predicted becomes the combined prediction of that set of models.



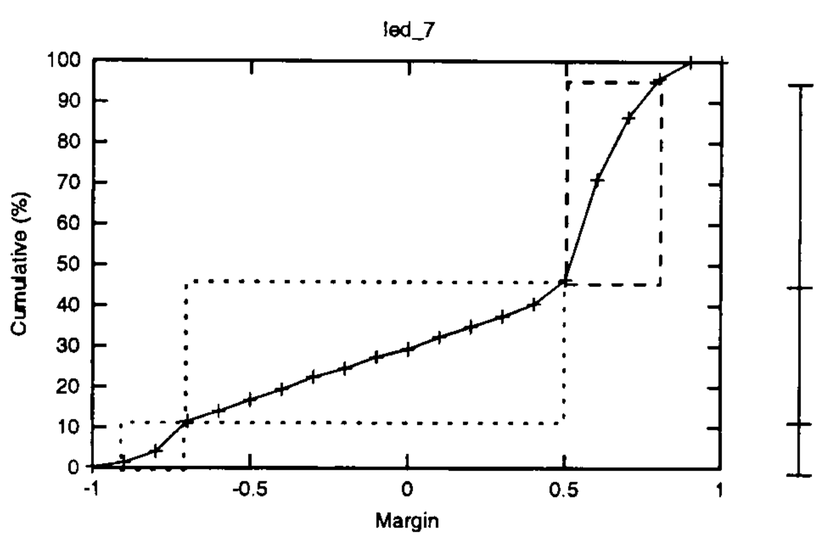
(a) ABALONE



(b) LETTER



(c) CHESS



(d) LED–7

Figure 2.1: Examples of margin distribution graphs and locations of their slopes. The main slopes are shown in dashed rectangles. The other slopes are shown in dotted rectangles. The height of each slope is shown on the right hand side of the graph.

of the model on that instance.

- *Analysing a prediction of a set of models* By using the sum of the weights that the models give to each class as a class weight and normalising the weights so that they add up to one, the margin of the prediction from a set of models can be calculated.

The margin that has been described so far is a margin of the prediction of a single instance. The margin of the predictions on the whole data set can be represented in a *margin distribution graph*. The margin distribution graph indicates how confident a model is in predicting class values of a particular data set. Examples of the margin distribution graphs are shown Figure 2.1. The X-axis represents the margin range. The Y-axis represents the percentage of

instances whose margin is *less than or equal to* the corresponding margin on the X-axis. Since the value of margin ranges between -1 and 1, the left half of the graph ([-1,0)) represents the misclassification behaviour of the model, and the right half ((0,1]) represents its correct classification behaviour.

An important information conveyed in the margin distribution graph is the area where the *main slope*[2] of the graph is located. The main slope of each graph in Figure 2.1 is shown in a dashed rectangle and the other slopes are shown in dotted rectangles. The main slope of the graph is the slope with the highest height. The main slope shows that the main coverage of the model covers the instance space with a particular margin range. The projection of the main slope on the Y-axis represents the percentage of the instances that are under the main coverage of the model. The projection of the main slope on the X-axis represents the margin range of the main coverage. Since a margin represents the accuracy and confidence of a model in making a single prediction, the margin range where the main slope is located, shows how accurate and confident a model is in predicting that data set. For example, the main slope that is located in the positive margin area implies that the model is mainly accurate in predicting the data set of interest and the more positive the margin range, the more confident the model is in its predictions.

Two examples of the interpretation of the main slope in margin distribution graph are shown in Figure 2.2. The slope in Figure 2.2 (a) can be interpreted as the model covers 90% of this data set with a margin between -0.6 and 0.3. Similarly, the slope in Figure 2.2 (b) can be interpreted as the model covers 50% of the data set with a margin between 0.5 to 0.8. In these two examples, the model in Figure 2.2 (b) is considered to have much better coverage than the model in Figure 2.2 (a), since the majority of its coverage (50%) covers the instance space with a positive margin, which means it covers the right target area with moderately high confidence. In Figure 2.2 (a), the majority of the coverage of the model (90%) is somewhere between the right and wrong target area. The

[2]Given $Y = a + bX$, where X is an independent variable and Y is a dependent variable, b, which is the amount of change produced in Y by a unit change in X, is a slope.

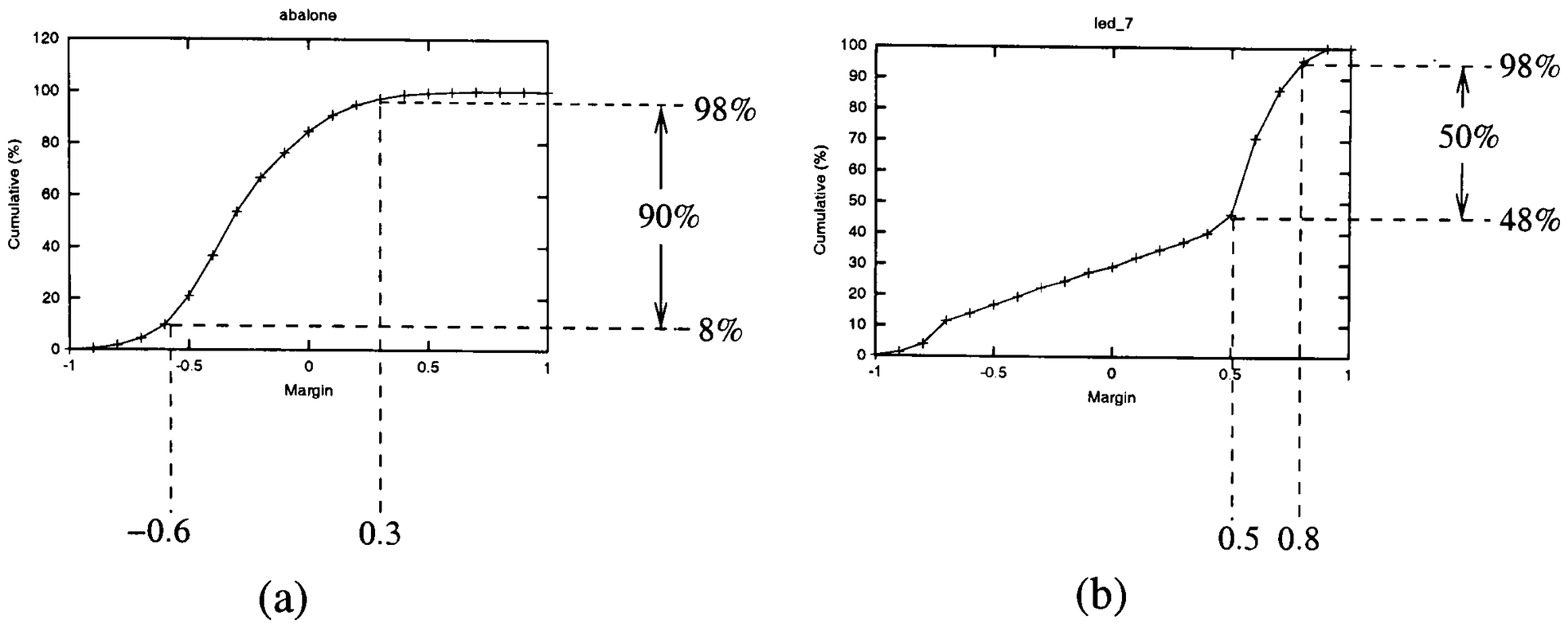


Figure 2.2: Two examples of the interpretation of the main slope in margin distribution graph.

slope starts in the area where the margin is -0.6, which is considered as a poor coverage area. In addition, the slope ends in the area where the margin equals 0.3, which is considered a relatively low-confidence coverage area. In an ideal case, where the model covers the right target area and has high confidence in its coverage, the starting point of the graph will be parallel to the X-axis and sloping up as the value of x (margin) tends towards 1. We will refer to this shape of a margin distribution graph as the ideal shape. An example of the ideal shape is shown in Figure 2.1 (c).

The margin distribution graph that has been described so far is based on a single data set. Since the margin distribution graph shows the coverage of the model on a particular data set, if that data set represents the instance space, then the margin distribution graph shows the coverage of the model over the instance space. Also, since the margin of an instance implies the location of that instance in the model coverage, it can be used to select instances for adjusting the coverage of models via the active learning concept, which will be discussed next.

2.6 Active learning

Active learning is a form of learning, in which the learning procedure has some control over the input it trains on [24, 75, 61]. That is, the learning procedure can *selectively* sample or query examples from the current environment on parts of the problem domain it considers useful[3]. This part of the problem domain is referred to as *a region of uncertainty* [24]. An example of this region, is the part of the problem domain where the model has low confidence in its predictions.

In active learning, the learning process is iterative, thus data regions are viewed differently during the process. In order to perform selective sampling effectively, an *ordering scheme* for differentiating a region of uncertainty from the vast instance space and an approach to *approximate* the region of uncertainty are required. The ordering scheme provides a powerful basis for organising a search through the hypothesis space [70]. An example of such an ordering scheme, and an approximation of the area of uncertainty, can be seen from *SG-network* [24]. *SG-network* is a neural network version-space search algorithm. In *SG-network*, Cohn et al. [24] express the order of region by imposing a *general-specific* concept on all concepts. By maintaining the *most general* and *most specific* concepts, they can use the difference between the coverage of the two concepts to approximate an area of uncertainty. In this thesis, a margin which provides information about the confidence of a model in predicting an instance, is used to impose an order of confidence of the model on covering the instance space. Instances whose margins of assigned classes are centred around zero, can represent the area of uncertainty with respect to the model. By manipulating instances in this area, we can improve the coverage of local models in the case where it is not suitable to be integrated.

Up to this point, the technical concepts involved in the formation of DLKP have been described. In the next section, the approaches that are used to evaluate the results of the empirical studies in Chapter 5, are described.

---

[3]If the data distribution $\mathcal{D}$ is known, the example can be directly selected based on $\mathcal{D}$. Frequently, the data distribution is unknown. Therefore, this information can be observed from the current environment, such as a current model or a current set of models.

2.7 Empirical evaluation approaches

One fundamental problem in experimental studies is to establish how confident one is with the experimental results. This is also the case in machine learning and data mining studies where carrying out an empirical study is the key to understanding the problems, but it is often performed on limited samples of data. To learn a model and estimate its accuracy from limited samples can be misleading. Consequently, statistical methods should be used to bound the difference between the observed accuracy over the available samples of data and the true accuracy over the entire distributions of data.

In the following subsections, statistical methods that are used in the empirical study of DLKP, which will be discussed in Chapter 5, are reviewed. An approach for estimating the accuracy of the hypothesis is presented first. Then, the confidence intervals of the accuracy estimation are described. The last subsection introduces an approach for comparing two learning algorithms. The discussions will be based on the following setting.

There is some space of possible instances X, over which various target functions may be defined. Different instances in X may be encountered with different frequencies. It is assumed that there is some unknown probability distribution $\mathcal{D}$, that defines the probability to encounter each instance in X. The learning task is to learn the target function f, by considering a space H, of possible hypotheses. The output of the learning task is a function $\hat{f}$ that is an approximation of f. Each training instance is independently drawn according to the distribution $\mathcal{D}$ and its correct class value $f(x)$ is passed to the learner.

2.7.1 Estimating hypothesis accuracy

First, the two notions of error need to be distinguished. The first notion is the error rate of the hypothesis over the *available* sample of instances. The other is the error rate of the hypothesis over the *entire* distribution $\mathcal{D}$ of instances. These errors will be referred to as the *sample error* and the *true error*, respectively.

With respect to sample S of instances drawn from X, the *sample error* of a hypothesis is a fraction of S that has been misclassified. The sample error ($error_S(\hat{f})$) of hypothesis $\hat{f}$ with respect to target function f and data sample S is

$$error_S(\hat{f}) \equiv \frac{1}{m} \sum_{x \in S} \delta(f(x), \hat{f}(x)) \tag{2.4}$$

where m is the number of examples in S, and the quantity

$$\delta(f(x), \hat{f}(x)) = \begin{cases} 1 & \text{if } f(x) \neq \hat{f}(x), \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

The *true error* of hypothesis is the probability that the hypothesis will incorrectly classify a single instance, randomly drawn from the distribution $\mathcal{D}$. The true error ($error_{\mathcal{D}}(\hat{f})$) of hypothesis $\hat{f}$, with respect to target function f and distribution $\mathcal{D}$, is the probability that $\hat{f}$ will incorrectly classify an instance drawn at random according to $\mathcal{D}$. That is,

$$error_{\mathcal{D}}(\hat{f}) \equiv P_{x \in \mathcal{D}}[f(x) \neq \hat{f}(x)] \tag{2.6}$$

where $P_{x \in \mathcal{D}}$ denotes that the probability is taken over the instance distribution $\mathcal{D}$.

The true error of the hypothesis ($error_{\mathcal{D}}(\hat{f})$) is the quantity of interest in the empirical study. This is because it is an error expected when the hypothesis is applied to future instances. However, the quantity that is normally measured is the sample error of the hypothesis on the data sample S ($error_S(f)$). This leads to the discussion in the next subsection on how confident we can be in using $error_S(\hat{f})$ as an estimation of $error_{\mathcal{D}}(\hat{f})$.

2.7.2 Confidence intervals of the accuracy estimation

A common way to describe the uncertainty associated with an estimate is to give an interval within which the true value is expected to fall, together with the probability with which it is expected to fall in the interval. Such an estimate is called the *confidence interval* estimate [53]. An $N\%$ confidence interval for some parameters q is an interval that is expected to contain q with probability $N\%$.

The confidence intervals for $error_{\mathcal{D}}(\hat{f})$ can be derived, based on the fact that the estimation of $error_{\mathcal{S}}(\hat{f})$ is governed by the Binomial probability distribution. The mean value of this distribution is $error_{\mathcal{D}}$, and the standard deviation is given by

$$\sigma_{error_{\mathcal{S}}(\hat{f})} \approx \sqrt{\frac{error_{\mathcal{S}}(\hat{f})(1 - error_{\mathcal{S}}(\hat{f}))}{m}} \tag{2.7}$$

For example, a 95% confidence interval for $error_{\mathcal{S}}(\hat{f})$ is the interval which is centred on the mean value $error_{\mathcal{D}}(\hat{f})$ and wide enough to contain 95% of the total probability under this distribution. That is, $error_{\mathcal{S}}(\hat{f})$ must fall into this interval 95% of the time. The calculation of the size of the interval that contains $N\%$ of the probability can be approximated by the Normal distribution. This is based on the fact that for sufficient large example sizes ($m \geq 30$), the Binomial distribution can be very closely approximated by the Normal distribution with the same mean and variance [72]. If a random variable Y obeys a Normal distribution with mean μ and standard deviation σ, then the measured random value y of Y will fall into the following interval $N\%$ of the time.

$$\mu \pm \mathcal{Z}_N \sigma \tag{2.8}$$

The constant $\mathcal{Z}_N$ defines the width of the smallest interval around the mean that includes $N\%$ of the probability mass under the Normal distribution [72]. Substituting μ and σ with $error_{\mathcal{S}}(\hat{f})$ and the standard deviation given in Equation 2.7

yields:

$$error_S(\hat{f}) \pm \mathcal{Z}_N \sqrt{\frac{error_S(\hat{f})(1 - error_S(\hat{f}))}{m}} \tag{2.9}$$

for a $N\%$ confidence interval for discrete-valued hypotheses.

Up to this point, the standard techniques for estimating the accuracy of the hypothesis and its confidence interval are described. The next subsection discusses the technique for comparing two learning algorithms when applied to the same data set.

2.7.3 Comparing learning algorithms

It is known that many variations can occur within a learning process. These variations can be caused by a variation in data samples or the sensitivity of the learning algorithm to different samples of data. Therefore, an appropriate method is required for comparing the performance of two learning algorithms. This is to ensure that the observed difference between the algorithms is statistically significant.

Given L_A and L_B as the two learning algorithms of interest and a training set size m, the relative performance of these two learning algorithms should be considered over all possible training sets of size m. Every training set is expected to be drawn from the underlying data distribution $\mathcal{D}$. In statistical terms, this procedure is the estimation of the expected value of the difference in their errors:

$$E_{S \subset \mathcal{D}}[error_{\mathcal{D}}(L_A(S)) - error_{\mathcal{D}}(L_B(S))] \tag{2.10}$$

where $L(S)$ denotes the output of learning algorithm L, given the sample S from the training data and where $E_{S \subset \mathcal{D}}$ denotes that the expectation of the value is taken over sample S which is drawn according to the underlying data distribution $\mathcal{D}$.

In the scenario where the available data, D_0, is limited, a reasonable approach to estimate this quantity is to divide D_0 into a training set S_0 and a test set T_0. Both L_A and L_B can then learn from S_0 and their output can be compared using T_0. In statistical terms, the comparison procedure measures:

$$error_{T_0}(L_A(S_0)) - error_{T_0}(L_B(S_0)) \tag{2.11}$$

In Equation 2.11, $error_{T_0}(\hat{f})$ is used to approximate $error_{\mathcal{D}}(\hat{f})$, and the measurement is taken over one training set S_0, rather than the over all sample S that might be drawn from the distribution $\mathcal{D}$.

The estimation in Equation 2.11 can be improved by repeatedly partitioning D_0 into disjointed training and test sets, and taking the mean of the difference in the test error of both learning algorithms ($\bar{\delta}$) as a difference between L_A and L_B. That is, the data D_0 is first partitioned into l equal-size subsets. The learning and test procedures are repeated l times, in turn using each of the l subsets as the test set, and using the remaining $l-1$ subsets as the training set. This procedure is called *l-fold cross-validation.* The procedure of l-fold cross-validation for comparing two learning algorithms is shown in Figure 2.3. The advantage of this approach is that the learning algorithm is tested on l test sets that are independent from the training data. The mean difference in error ($\bar{\delta}$) is used as an estimate of the difference between the two learning algorithms. $\bar{\delta}$ can be viewed as an estimation of the quantity:

$$E_{S \subset D_0}[error_{\mathcal{D}}(L_A(S)) - error_{\mathcal{D}}(L_B(S))] \tag{2.12}$$

where S represents a random sample of size $\frac{l-1}{l}|D_0|$ drawn uniformly from D_0. The difference between the expression in Equation 2.10 and 2.12 is that the latter takes the expected value over subsets of the available data D_0, rather than over subsets drawn from the full instance distribution $\mathcal{D}$.

The approximate $N\%$ confidence interval for the estimation in Equation 2.12

Input:

Data D_0, Learning algorithm L_A and L_B

Data partition phase:

Partition data D_0 into l disjointed subsets $T_1, T_2, \ldots, T_l$ of equal size.

Learning phase:

For $i = 1$ to l {
use T_i as a test set, and use $D_0 - T_i$ as a training set S_i
$M_A = L_A(S_i)$
$M_B = L_B(S_i)$
$\delta_i = error_{T_i}(M_A) - error_{T_i}(M_B)$
}

Output:

The average of the difference of the error $\bar{\delta}$, where $\bar{\delta} \equiv \frac{1}{l}\sum_{i=1}^{l}\delta_i$

Figure 2.3: A procedure for estimating the difference in error between two learning algorithms.

is given by:

$$\bar{\delta} \pm t_{N,r-1} S_{\bar{\delta}} \tag{2.13}$$

where $S_{\bar{\delta}}$ is an estimate of the standard deviation of the distribution governing $\bar{\delta}$. $S_{\bar{\delta}}$ is defined as:

$$S_{\bar{\delta}} \equiv \sqrt{\frac{1}{l(l-1)}\sum_{i=1}^{l}(\delta_i - \bar{\delta})^2}. \tag{2.14}$$

The constant $t_{N,r-1}$ characterises the area under the t distribution, and it has two subscripts. The first (N) represents the confidence level. The second ($r-1$) is the number of degrees of freedom. This number is related to the number of independent random events in producing $\bar{\delta}$. In this setting, the number of degrees of freedom is $l - 1$.

2.8 Relations of the concepts in the formation of DLKP

Up to this point, the background of the concepts used in the formation and empirical evaluation of DLKP have been described. The following introduces how these concepts relate to the formation of DLKP.

The design of Distributed Learning with Knowledge Probing (DLKP) is based on the concept of model coverage over the instance space. In distributed data mining, it is not always possible to join the coverage of local models into one piece. This depends on the coverage of each local model. However, in distributed data mining, the coverage of the models that are learnt from distributed data sets is not known in advance. Thus, there is a need for a tool to observe the current form of the coverage of the local models. We propose the use of margin distribution graph as a tool for this purpose. The margin distribution graph indicates whether the current form of coverage of local models is suitable for integration. In a scenario where it is unsuitable to be integrated, we propose an iterative process for adjusting local data distributions so that the coverage of the learnt models becomes more suitable. The proposed adjusting techniques are based on the active learning concept. That is, in each iteration of the adjusting process, we selectively manipulate data that the models have low confidence in predicting their class values, so that the coverage of the re-learnt models becomes more suitable for integration.

The framework proposed is justified by empirical studies. Thus, statistical techniques are required to evaluate the significance of the experimental results. An empirical study in machine learning and data mining is often performed on limited training data, and it is very unlikely that the available data will completely cover the instance space. Thus, the first and second tools used are the techniques for estimating the accuracy of the hypothesis, along with the confidence of the intervals of such an estimation. Since we are also performing a comparative study between DLKP and the centralised learning approach, the

third statistical technique required is the technique for comparing the two learning algorithms.

Specifically, the empirical studies are performed to evaluate and justify the design of DLKP, the use of margin distribution graph, and the techniques used in adjusting data distributions. Apart from these three aspects, the empirical analysis also covers bias and variance decomposition study, which is one of the standard analysis in machine learning and data mining research. This is aimed at improving the understanding of the contribution in bias and variance from each learning phase of DLKP to the final result.

DLKP is introduced in the next chapter, in which the key issue and the scenarios of applying DLKP are discussed.

Chapter 3

Distributed Learning with Knowledge Probing: DLKP Framework

In this chapter, we propose Distributed Learning with Knowledge Probing (DLKP) as a framework for model integration in distributed data mining. The key issue of DLKP is addressed. Four main scenarios of model integration with DLKP are presented.

3.1 Introduction

As mentioned in Section 1.3.2, local models in distributed data mining are different from base models in multiple model learning. We propose DLKP as a new framework for model integration in distributed data mining. However, since the coverage of the local models in distributed data mining can be in a variety of forms, we address the key issue of the effectiveness of DLKP[1], the consistency of the local models in the integration. We propose the use of the margin distribution graph as a tool for measuring the consistency of the local models in terms of their coverage. This is to enable the optimisation of the effectiveness of DLKP

[1]With respect to classification accuracy.

in the scenarios where the local models are not suitable for integration. Based on the analysis of the margin distribution graph, four main scenarios of model integration with DLKP are presented.

3.2 Distributed inductive learning concepts

Generally, a distributed inductive learning system consists of n *local learners* (L-learners) and a *meta learner* (M-learner). Each L-learner independently observes a set of examples generated independently under a local target distribution $P(S_i|\theta_i), i = 1, \ldots, n$, and outputs a model which is able to estimate θ_i. Let $\hat{\theta}_i$ be the estimation computed by the i-th L-learner. The M-learner takes $\hat{\theta}_1, \ldots, \hat{\theta}_n$ as input, and outputs an estimation of θ as a function of $\hat{\theta}_1, \ldots, \hat{\theta}_n$. That is, the M-learner forms a global model from the results of n L-learners.

This formation of distributed inductive learning reveals the basic features of distributed learning by introducing the concept of L-learner and M-learner:

1. L-learner : gaining a local model from a part of data.

2. M-learner : approximating a global model by integrating the local models.

It may be the case that all the local target distributions $P(S_i|\theta_i)$ in a distributed inductive learning system are identical. That is, all the local data are generated under the same target distribution $P(S_i|\theta)$, i.e. $\forall_{i \neq j} \theta_i = \theta_j = \theta$. In this case, each L-learner estimates a common parameter θ, and the M-learner integrates the local models by averaging all the local $\hat{\theta}_i$. This corresponds to the notion of *Plain Learning Model* of distributed Bayesian learning that is proposed by Yamanishi [93]. An abstraction of this scenario is shown in the top diagram of Figure 3.1.

When each L-learner deals with different target distributions, however, a more complicated distributed learning approach is required. For example, we may assume that all the local target distributions $P(S_i|\theta_i)$ in a distributed inductive learning system are not necessarily identical, and each of unknown parameters θ_i is independently generated under a prior distribution $P(\theta|\mu)$, where μ is an

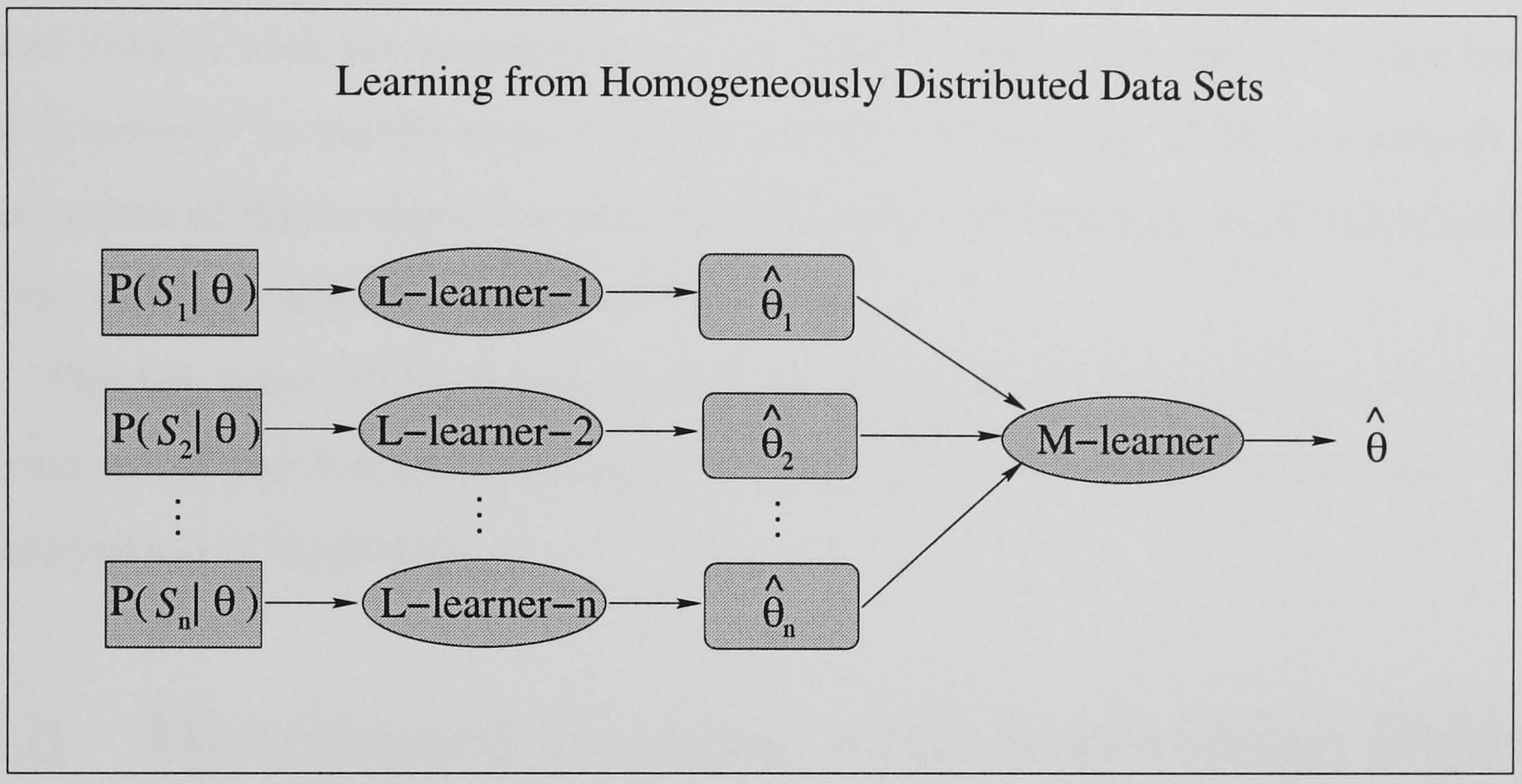


Figure 3.1: Abstractions of the distributed inductive learning from homogeneously and heterogeneously distributed data sets.

unknown meta-parameter. In this case, a distributed learning system should not only involve an induction of all θ_i which model local data distributions, but should also involve an induction of μ which models the interrelationship among local data distributions. An iterative process may be required to obtain optimal μ. That is, at a particular time step t, θ_i^t is learnt by the i-th L-learner from the local data S_i with the parameter μ^{t-1}. μ^t will be induced by the M-learner from all $\hat{\theta}_i^t$ where μ^0 is usually given based on the prior knowledge. This corresponds to the notion of *Hierarchical Learning Model* in [93]. An abstraction of this scenario is shown in the second diagram of Figure 3.1.

The key point of this formation of distributed inductive learning is that the structure of the learning process is determined by the assumptions about the distribution of local data.

3.3 Distributed learning with Knowledge Probing (DLKP)

In this section, we propose a framework for model integration in distributed data mining. The framework is designed for the distributed environments where the local data sets are generated under the same target distribution. Thus, this framework is corresponding to the *Plain Learning Model* described in the previous section.

Given a group of n data sets $S_1, S_2, \ldots, S_n$, the learning algorithms L_1, L_2,..., L_n are applied to the corresponding data sets to produce local models $M_1, M_2, \ldots, M_n$. The local models are integrated by a combining procedure to produce a final model. This architecture is the same as the *learning using data partitioning (LUDP)* model described by Provost & Kolluri [76] as a method for scaling up inductive learning algorithms. By applying this architecture to distributed data mining where local data already exist, the focus of the architecture is changed from scaling up the learning process to integrating local models into a single model that reflects a global view of the local models. This integration

can be achieved by applying the *Knowledge Probing* technique.

Knowledge Probing was proposed by Guo et al. [47] as a technique to extract descriptive knowledge from a black box model[2], such as a neural network. Given a black box model B, Knowledge Probing aims at deriving a descriptive model from B. Knowledge Probing can be divided into two phases, the *prediction phase* and the *learning phase*. In the prediction phase, the black box B is used to give class values of instances in a probing set PD. Since PD is used to probe a global model, it is randomly sampled from the available training data. Ideally, each instance in PD should not have a class value. However, this is unavoidable when PD is formed by sampling from the available training data. Therefore, given an instance x from PD, if B predicts the class value of x differently from its original class value, the class value of x will be replaced by the class prediction from B. We will refer to this process as a *relabelling* process. In the learning phase, given a learning algorithm DL that gives a descriptive output, the probing set PD whose class values are assigned by B is learnt by DL. The output of Knowledge Probing is, therefore, a descriptive model that incorporates the predictive ability of B, and, at the same time, aids the understanding of the problem domain. An abstraction of this process is shown in Figure 3.2 (a). The concept of Knowledge Probing is summarised in Figure 3.3.

The Knowledge Probing concept can be extended to the application of model integration in distributed data mining. In distributed data mining, after local models are learnt, they can be used collaboratively to make a prediction. In each prediction, the local models first give class predictions of an instance. These class predictions are combined into a single combined prediction by a prediction scheme such as uniform voting. We will refer to a set of local models and its prediction scheme as a *model committee*. By viewing a model committee as a

[2]A black box model is a result of the black box learning approach [68]. The black box learning approach, such as neural networks, develops its own representation that is used for approximating the target function. However, its internal representation is difficult for human users to interpret, so it does not provide insight into the problem domain. The black box learning approach typically involves numeric calculations of coefficients, distances or weights.

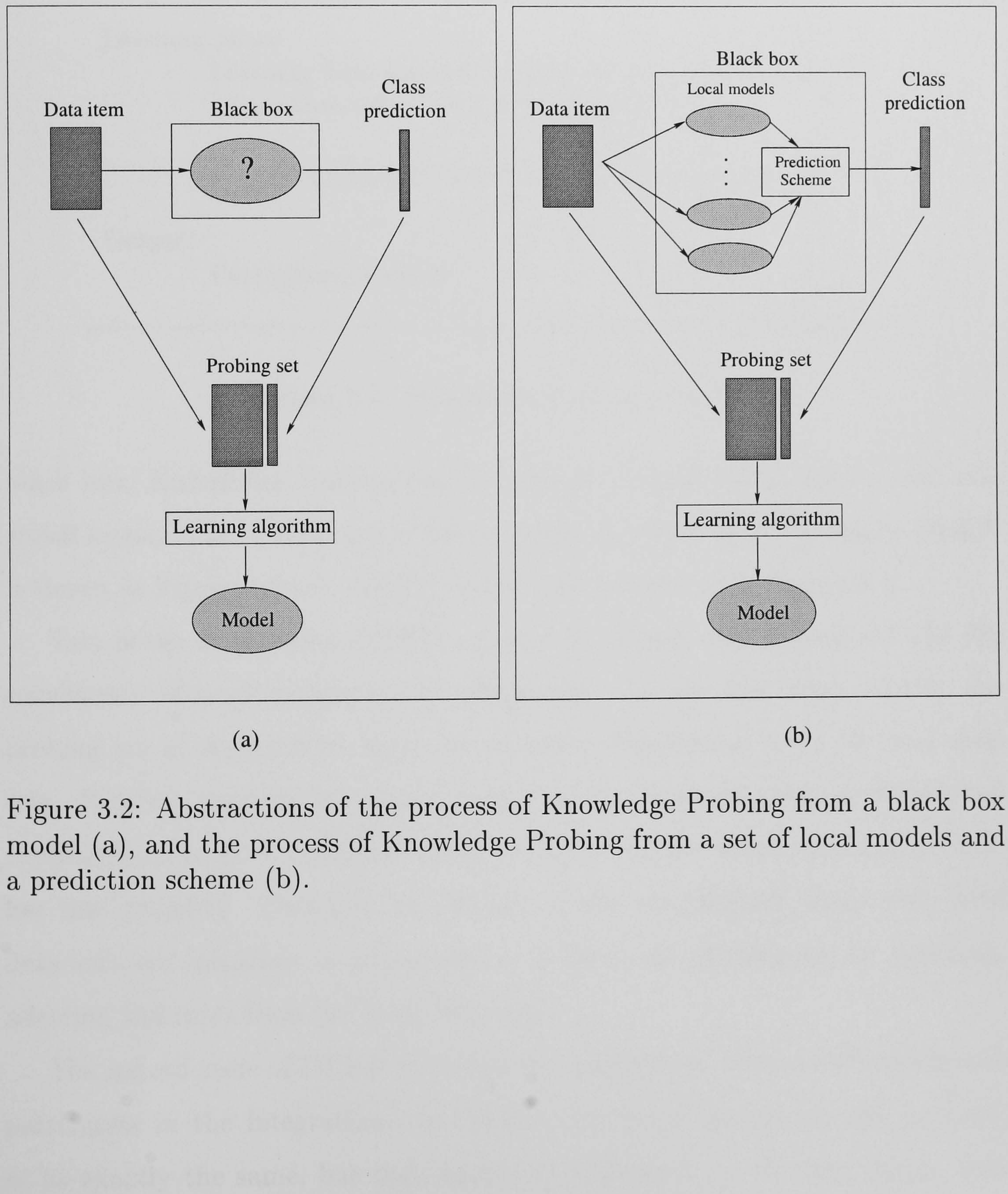


Figure 3.2: Abstractions of the process of Knowledge Probing from a black box model (a), and the process of Knowledge Probing from a set of local models and a prediction scheme (b).

Input:

A probing set PD of m instances,
a black box model B,
a learning algorithm DL which provides descriptive output.

Prediction phase:

Creating the class values of instances in PD using B.

$$\mathcal{C} = B(\ \{x_i \mid x_i \in PD, i = 1, \ldots, m\}\ )$$

Learning phase:

Learning from the new training set constructed from PD whose class values are the predictions in $\mathcal{C}$.

$$\hat{B} = DL(\ \{(x_i, c_i) \mid x_i \in PD, c_i \in \mathcal{C}, i = 1, \ldots, m\}\ )$$

Output:

Descriptive model $\hat{B}$

Figure 3.3: Knowledge Probing concept.

black box, Knowledge Probing can be used to integrate local models into one global model. The abstraction of the extension of Knowledge Probing to DLKP is shown in Figure 3.2 (b). DLKP can be summarised as in Figure 3.4.

Two issues in applying DLKP are how to prepare the probing set and the consistency of local models in the integration. For the first issue, ideally, the probing set is expected to share the common distribution with all local data sets. However, because we cannot expect all the local data sets in distributed environments to have a common distribution, we cannot form a probing set that has that property. Therefore, practically, in the environment where only local data sets are available, it is reasonable to form the probing set by randomly selecting instances from the local data sets.

The second issue of DLKP concerns the consistency of the local models that participate in the integration. In DLKP, the local models are not expected to be exactly the same, but they should be consistent, or in other words, they should cover the main target area in the instance space. This is because the

Input:

A probing set PD, $PD = \{x_1, \ldots, x_m\}$,
A set $\mathcal{S}$ of n local data sets, $S_1, \ldots, S_n$,
a prediction scheme PS,
a local learning algorithm L,
a learning algorithm DL which provides descriptive output.

Local model learning:

Learning n local models.

for i = 1 to n {
$M_i = L(S_i)$
}
$\mathcal{M} = \{M_i \mid i = 1, \ldots, n\}$

Prediction phase:

Creating the class values of instances in PD using
n local models and the prediction scheme PS.

for i = 1 to m {
for j = 1 to n {
$\{c_{ij} = M_j(x_i) \mid M_j \in \mathcal{M}, x_i \in PD\}$
}
$C_i = PS(c_{i1}, \ldots, c_{in})$
}

$\mathcal{C} = \{C_i \mid i = 1, \ldots, m\}$

Learning phase:

Learning from the new training set which consists of instances from PD
whose class values are the corresponding combined prediction in $\mathcal{C}$.

$\hat{B} = DL(\ \{(x_i, C_i) \mid x_i \in PD, C_i \in \mathcal{C}, i = 1, \ldots, m\}\ )$

Output:

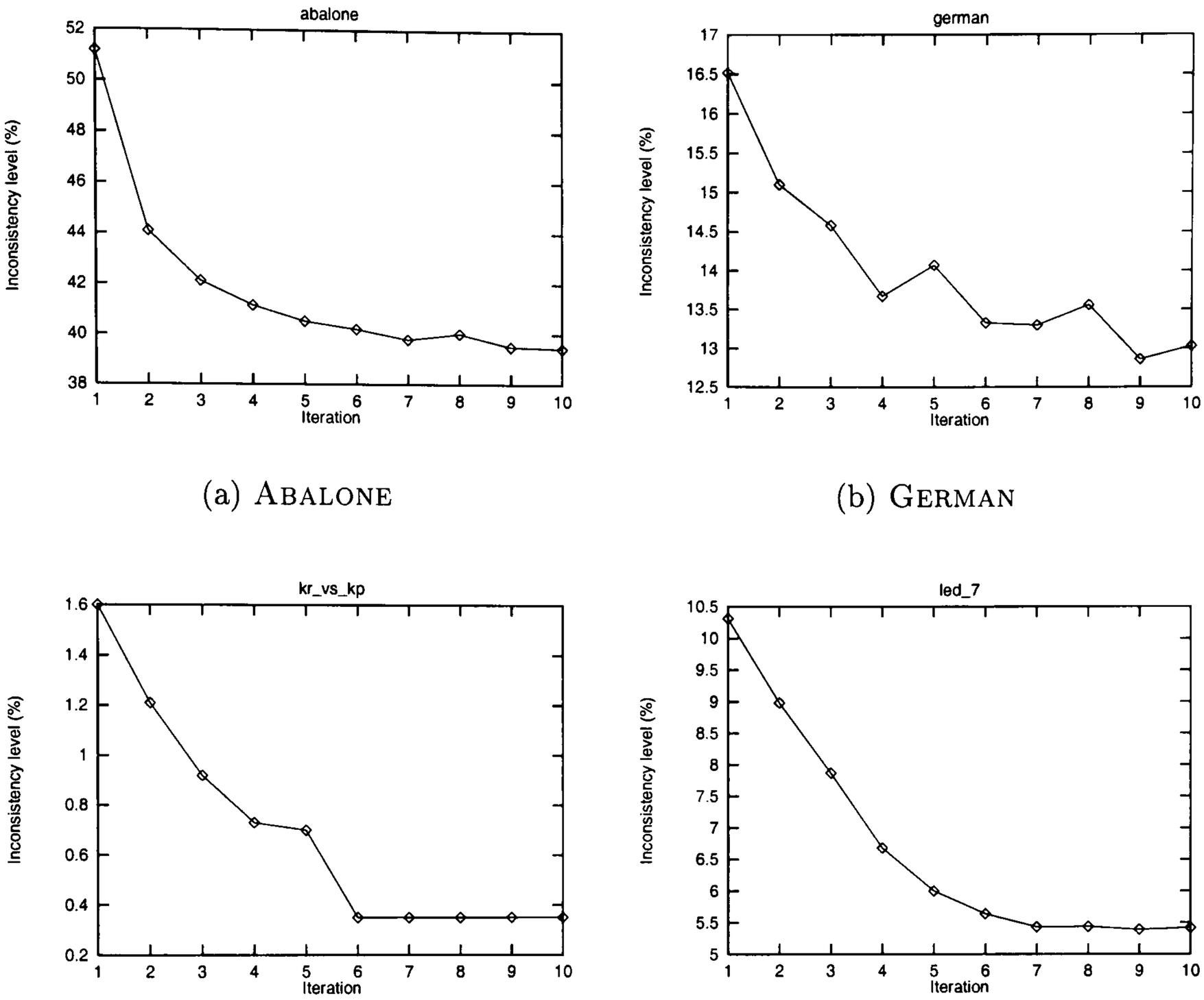
Descriptive model $\hat{B}$.

Figure 3.4: The DLKP framework.

predictive ability of the local models is combined through a prediction scheme, which is normally based on some quantitative measurement, such as majority voting. Thus, if the local models are highly inconsistent, the predictions combined by the prediction scheme cannot be accurate. This leads to a decrease in the accuracy of the model committee as a whole. The poor model committee, or, in other words, the poor black box will result in a poor global model. Therefore, inconsistent local models are not suitable to be integrated with DLKP unless their inconsistency is detected and improved. This issue is further discussed in the next section.

3.4 Key issue of DLKP: Consistency of the local models

As mentioned earlier, in order to integrate local models into one global model, the local models should be consistent to certain levels i.e. they largely agree on their predictions. If local models are highly inconsistent, some procedures should be employed to dynamically adjust local data distributions so that the learnt local models become more suitable to be integrated with DLKP. In this thesis, an iterative learning procedure is used to adjust local data distributions. The details of the techniques used will be discussed in Section 4.4. Briefly, a model committee is used to predict class values of instances in a consistency test set. These predictions are used to plot a margin distribution graph of the model committee. This graph is used to observe the consistency of the local models in the committee. The result of the observation suggests either to increase or to decrease the local data sets at time step t. In time step $t+1$, the newly refined data sets will be re-learnt to induce a new set of local models. By repeatedly adjusting local data distributions, the local models are expected to gradually become consistent. This iteration continues until a stopping criterion, such as the local models largely agreeing with their predictions, is met. After the local models become consistent, DLKP can be applied to integrate them into a global



(a) ABALONE (b) GERMAN

(c) KR-VS-KP (d) LED-7

Figure 3.5: Improvement in the consistency of the local models by the data redistribution technique.

model. Our previous work [48, 49] on a data redistribution technique, which is one of the techniques we use for adjusting data distributions, suggests that the data redistribution technique can be used to improve the consistency of local models. Figure 3.5 shows the improvement in the consistency of the local models by the data redistribution technique[3]. In the graphs, the percentage of instances in the consistency test set that local models disagree with the predictions of their class values is plotted against the number of iterations during the data distribution adjusting process. The graphs show that the technique decreases the number of conflict instances as the iteration proceeds.

[3]The descriptions of the four data sets in Figure 3.5 can be found in Appendix B.

It should be noted that although it is generally known that diversity among local models, or base models[4], is useful in combining multiple models [62, 61, 4, 85], base models used in those works are deliberately generated from a single data source. Different techniques, such as Stochastic Hill-climbing [4], Bagging [11] and Randomisation [31], are used to inject stochastic effects into the base model generation process to increase diversity among base models. In other words, those techniques stochastically expand the coverage of the base models (especially in the border area)[5]. This means that although diverse, those base models are not completely different and still share a common coverage on the instance space (abstraction shown in Figure 3.6 (c)). This *diversity* among the base models should not be confused with the *inconsistency* of local model in distributed data mining. While the diversity that is introduced to the same source of data is beneficial to model integration, the inconsistency that may exists as a result of distributed data is harmful to the generation of a global model with a proper coverage. Thus, in the case of DLKP, which is designed for the environment where data are already distributed[6], there is a requirement of a tool to measure the current form of the coverage of a model committee. Also, there is a requirement of techniques to improve the coverage and the consistency of models in the committee when they are not suitable to be integrated.

In practical applications, the coverage of a model committee is not easily visualised as in Figure 3.6. Thus, we propose an indirect way to observe the coverage of a model committee via the analysis of its *margin distribution graph*[7]. Based on this analysis, the following section describes four main scenarios of applying DLKP to model committees of different coverage.

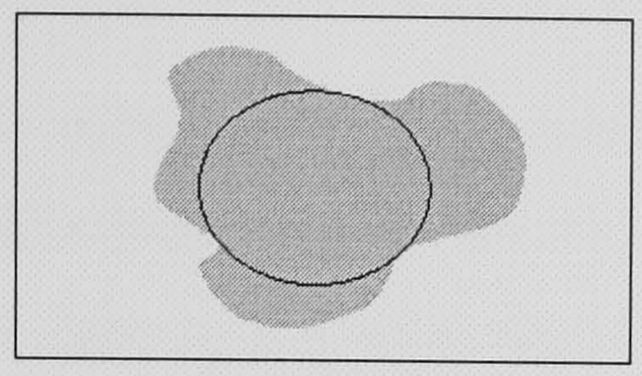
---

[4]We purposely use the term *local models* in the discussion of model integration in distributed environments, and use the term *base models* in the discussion of model integration in multiple model learning where distributed data sets are not the issue of concern.

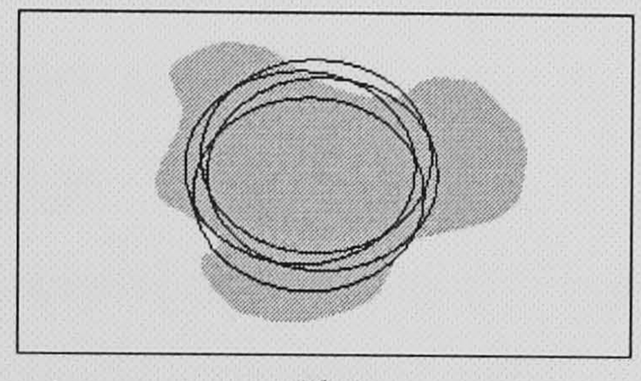
[5]By viewing inductive learning as a search through the hypothesis space as discussed in Section 1.1.1.

[6]The empirical study in Chapter 5 simulates the worst scenario of such an environment by using a data partitioning technique.

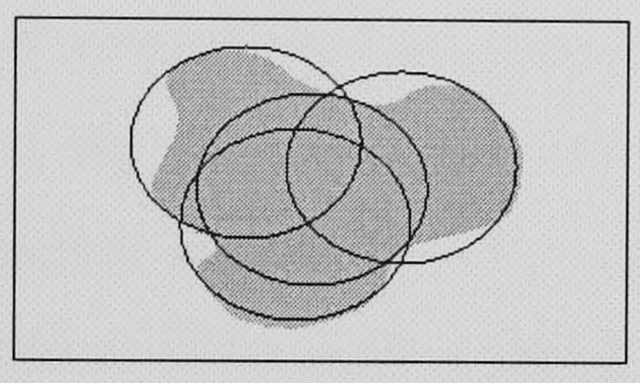
[7]The description and interpretation of margin distribution graphs were given in Section 2.5.



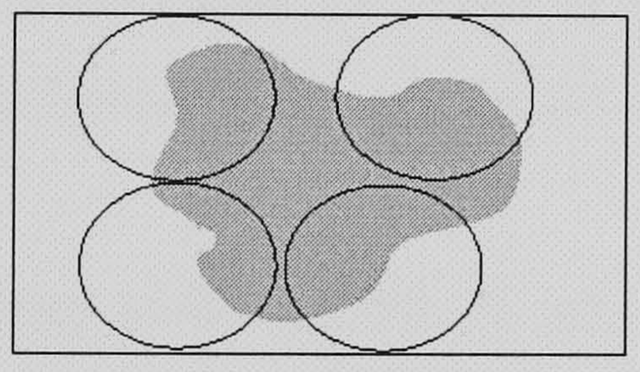
(a)



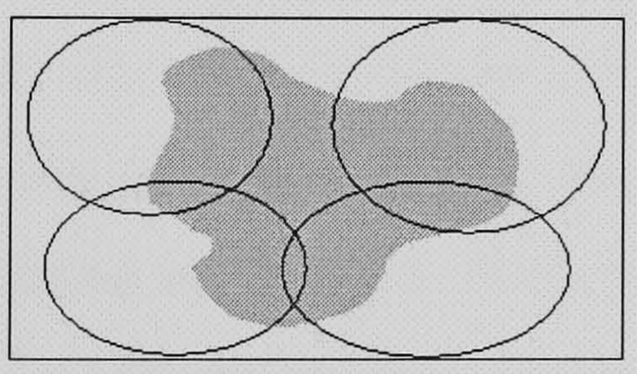
(b)



(c)



(d)



(e)

Figure 3.6: Abstractions of five scenarios of coverage of a model committee on the instance space. (The dark grey area represents the target area in the instance space. A circle represents the coverage of a model.) Diagram (a) represents the coverage of a single model on the instance space. Diagram (b) represents the coverage of a model committee whose local models are highly consistent. Diagram (c) represents the coverage of a model committee whose local models vary, but there is an overlapping in their coverage. Diagram (d) represents the coverage of a model committee whose local models are completely unrelated. Diagram (e) represents the coverage of a model committee whose local models are partly covering the wrong area.

3.5 Four main scenarios of applying DLKP

As described in Section 2.5, margin distribution graphs can be used to measure the main coverage of a model committee and the consistency of its local models. The projection of the main slope of a margin distribution graph on the Y-axis shows the percentage of the instance space that is under the main coverage of the model committee. The project of the main slope on the X-axis shows the margin range of the main coverage. These two pieces of information can be used to analyse the accuracy of the model committee on its main coverage, and the consistency of the local models in the committee. This is because each margin range represents different degree of accuracy and consistency of the model committee.

Concerning the accuracy, the main coverage that is in positive margin area shows that the model committee is accurate. The higher the margin range, the more confident the committee is in its main coverage. The main coverage that is in negative margin range shows that the model committee is inaccurate. The lower the margin range, the further away the model committee is from covering the target area.

Concerning the consistency of the models in the committee, as described in Section 2.5 that the set of probabilities involved in the prediction of the model committee is the sum of the sets of probabilities from each local model. Thus, the consistent local models can be observed from the margin distribution graph whose the main slope is located in the area where the margin is either highly positive or highly negative i.e. the area where margin is close to 1 or -1. The level of consistency among the local models tends to decrease when the absolute value of margin becomes smaller.

By analysing margin distribution graphs of model committees in this way, the coverage of model committees can be categorised into four main scenarios.

In the first scenario, local models are *highly consistent and accurate*, and the coverage of the local models almost completely overlaps. An abstraction of this scenario is shown in Figure 3.6 (b). This scenario can be observed from

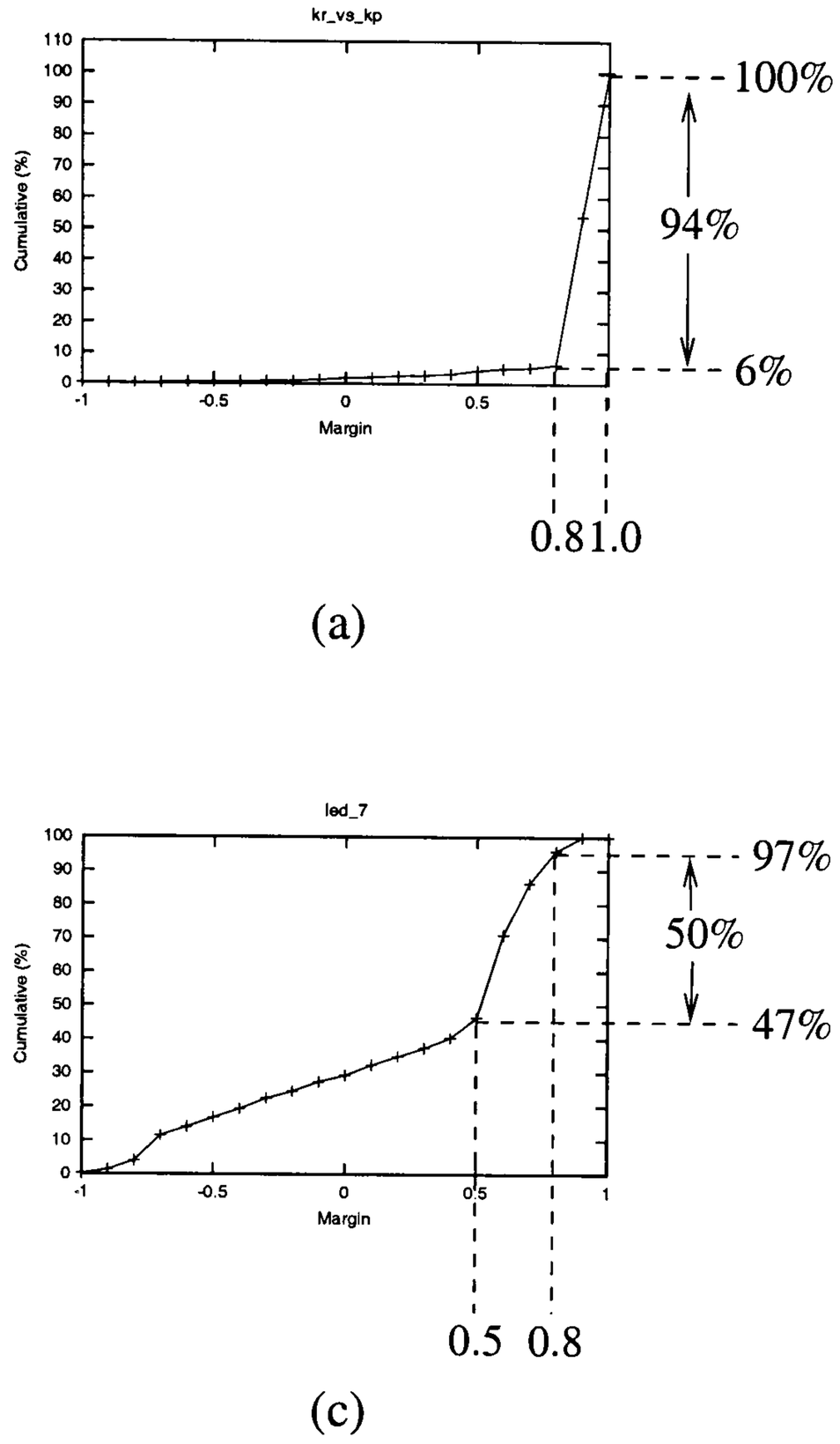


Figure 3.7: Margin distribution graphs of model committees that have different forms of coverage on the instance space.

the margin distribution graph that has ideal shape i.e. more than 90% of the coverage of the model committee has a margin more than 0.8. An example of this shape of margin distribution graph is shown in Figure 3.7 (a)[8]. This can be interpreted as very little or none of the data has negative margin. This means that the prediction of the model committee agrees with the original class of the instance most of the time. As a result, the relabelling process[9] of DLKP will rarely happen. Thus, in this scenario, the learning phase of DLKP is equal to learning from a probing set whose instances are randomly selected from the

[8]The descriptions of the four data sets in Figure 3.7 can be found in Appendix B.

[9]Note that in DLKP, the relabelling process only occurs when there is a probing instance having a negative margin and a probing instance will have a *negative* margin only when the prediction of the model committee on that instance conflicts with its original class value.

local data. In addition, the ideal margin distribution graph implies that the local models are consistently accurate. That is, they are more or less the same. Hence, no new information will be obtained from integrating them. Therefore, a final model can be arbitrarily selected from the local models.

In the second scenario, local models are *moderately accurate* and *consistent* i.e. at least 50% of the coverage of the model committee has margin higher than 0.8. An example of this shape of margin distribution graph is shown Figure 3.7 (b). This shape of graph implies that the coverage of the local models overlaps, but not as much as that in the first scenario. There are some degrees of randomness in the border area of their coverage. The coverage in this area increases the probability of obtaining additional accuracy from the integration. Thus, DLKP can be straightforwardly used to integrate models in this scenario. An abstraction of model coverage in this scenario is shown in of Figure 3.6 (c).

In the third scenario, local models are *fairly accurate*, but also *fairly inconsistent* i.e. the main coverage of the model committee has margin less than 0.8, but it is still in positive margin area. This shape of margin distribution graph is shown in Figure 3.7 (c). Because the majority of the main coverage is still in positive area, this implies that the local models are covering different areas in the instance space, and there is little or no overlap in their coverage. An abstraction of this scenario is shown in Figure 3.6 (d). We proposes a novel *data redistribution* technique as a technique to expand the coverage of local models so that they will increasingly become more accurate and more consistent. The data redistribution technique will be discussed in Section 4.4.1.

In the fourth scenario, local models are *inconsistent* and *inaccurate* i.e. the main coverage has margin less than 0.8 and it falls in the negative margin area (Figure 3.7 (d)). This implies that the local models are covering the wrong area in the instance space either because the given instances do not represent the problem domain well or the incompatibility between the inductive biases and the problem domain. In the first case, no attempt can be made to improve the

situation. In the latter case (Figure 3.6 (e)), we proposes a novel *data contraction* technique as a technique to contract the coverage of the local models so that the models will be gradually shifted to the area that they have higher confidence, and become more accurate and more consistent. The data contraction technique will be discussed in Section 4.4.2.

We have shown that margin provides a means of measuring the consistency and the accuracy of the model committee. However, it should be noted that the margin distribution graph we used provides the information about the coverage of a model committee as a whole. It does not provide the information about the coverage of each local model over the instance space. Therefore, the four main scenarios described may not cover every possible form of coverage of a model committee over the instance space. However, they should be able to cover most of them.

3.6 Summary

Based on distributed learning concepts, this chapter introduced DLKP as a framework for model integration in distributed data mining. The key issue of model integration in distributed data mining, the consistency of the local models, is addressed. That is, if the local models are highly consistent and highly accurate, there is limited benefit in integrating them. On the other hand, the highly inconsistent local models constitute an unstable and inaccurate model committee, which can lead to a poor integrated model. To observe the consistency of the local models, we introduced the use of margin distribution graph of the model committee. By analysing the margin distribution graph, we illustrated four main scenarios of applying DLKP to model committees of different coverage.

Note that for the rest of the thesis, we will refer to the first phase of DLKP where local models are learnt and used collaboratively to make combined predictions as the Local Model Learning phase. The second phase of DLKP where

a global model is probed from the model committee will be referred to as the Knowledge Probing phase. We will refer to the final result as a probed model or a final model as appropriate.

In the next chapter, the three main components of DLKP: the Local Model Learning phase, the Knowledge Probing phase, and the techniques for adjusting local data distributions, are discussed in details.

Chapter 4

Analysis of DLKP's Components

In this chapter, three main components of DLKP, *Local Model Learning phase*, *Knowledge Probing phase*, and *techniques for adjusting local data distributions*, are discussed. Since, the implementation of these components involves many design choices, issues of concerns in the design of each component are discussed. The discussion is also aimed at justifying the design choices used in the empirical study in the next chapter.

4.1 Introduction

In Chapter 3, we introduced DLKP as a framework for model integration in distributed data mining. DLKP consists of three main components, which are illustrated in Figure 4.1. The first two components are the two learning phases, which are the *Local Model Learning phase* (Figure 4.1 (a)) and the *Knowledge Probing phase* (Figure 4.1 (b)). The third component is the *technique for adjusting local data distributions* (Figure 4.1 (c)) in the scenario where the coverage of the model committee is not suitable to be integrated. The relations of the three components are shown in Figure 4.2. Figure 4.2 shows that there are two process branches for model integration under DLKP. The left branch of the flow chart in Figure 4.2 represents the scenario where the coverage of the model committee is suitable for the integration. Thus, from the Local Model Learning phase the process moves directly to the Knowledge Probing phase. The right branch of the

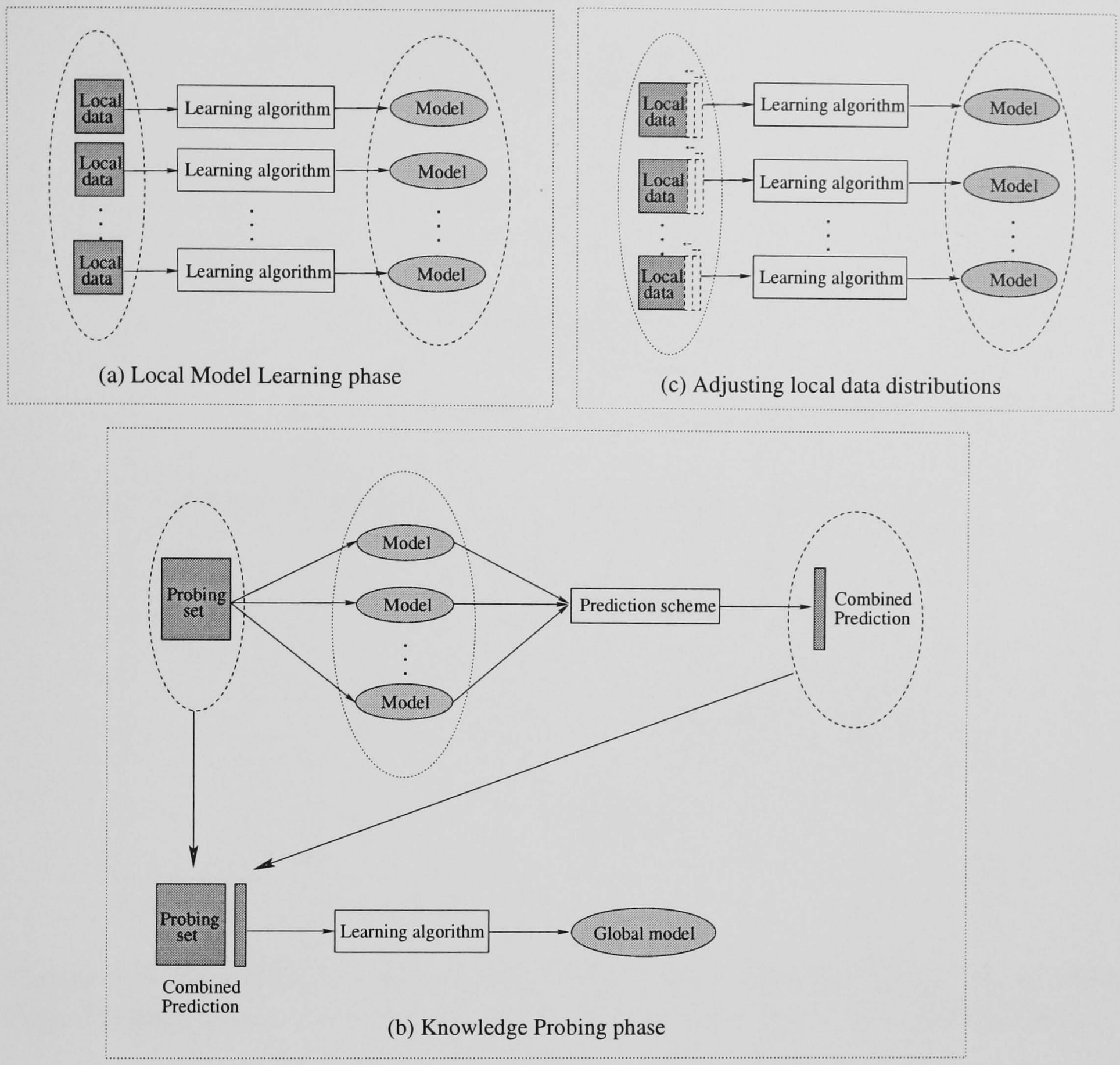


Figure 4.1: An illustration of the three main components of DLKP; Local Model Learning phase, Knowledge Probing phase, and techniques for adjusting local data distributions.

flow chart in Figure 4.2 represents the scenario where the coverage of the local models needs to be adjusted. Therefore, from the Local Model Learning phase, the process moves to the process of adjusting local data distributions before it moves to the Knowledge Probing phase. In this scenario, there is an iteration between the process of adjusting local data distributions and the Knowledge Probing process until the stopping criterion[1] is met.

The implementation of the three components of DLKP involves many design choices. These choices have an influence on the effectiveness of DLKP. In this

[1]The stopping criterion will be discussed in Section 4.4.

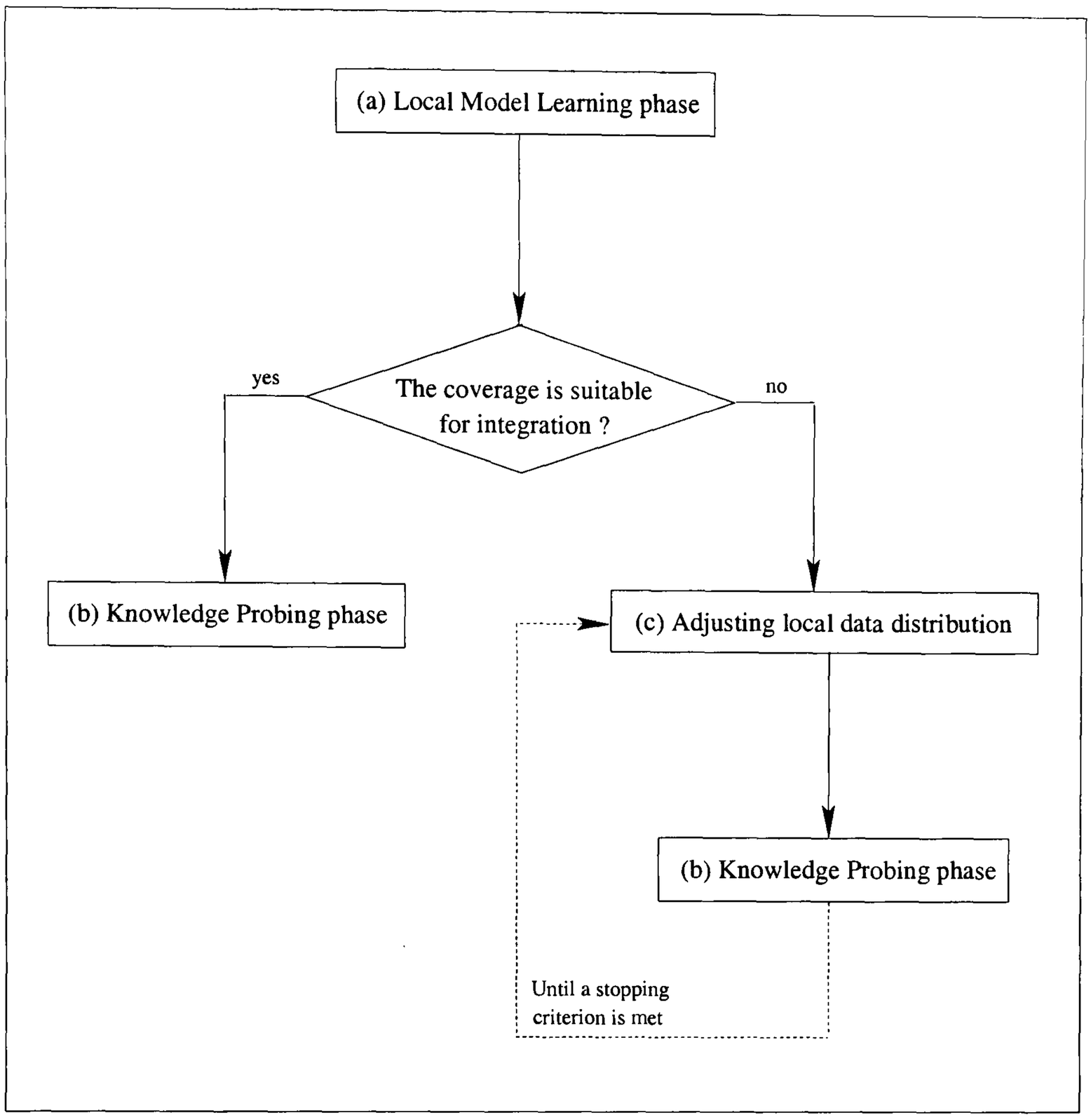


Figure 4.2: The relations between the Local Model Learning phase, the Knowledge Probing phase, and the techniques for adjusting local data distributions.

chapter, the three components are discussed together with some possible design choices, especially the design choices used in the empirical study in Chapter 5.

4.2 Local Model Learning phase

It is shown in many studies [90, 59, 20, 11, 4, 41] that learning with a committee of models can lead to an increase in accuracy over learning with a single model. This is because a set of models has a higher probability to cover different parts of the instance space than a single model has. To integrate their predictions by a prediction scheme, such as uniform voting or weighted voting can be viewed as a process of integrating the coverage of each model into one piece. As discussed in Section 3.5, the benefit of this integration process, however, depends on the

layout of the coverage of the local models in the committee. If their coverage is completely overlapped, there will be no benefit from the integration. On the other hand, if only some parts of their coverage is overlapped, the integration will lead to a wider area of coverage and it will result in an increase in accuracy. Thus, a model committee can increase the prediction accuracy when there is a variance in the coverage of its constituent models.

In multiple model learning, several techniques have been used to increase variance in the coverage of the models in the committee. Those techniques increase the variance by means of variations in the learning algorithm [31], the training data [11], or both [42]. In distributed data mining where each local data is independent from the others, there tends to be variation among the coverage of their local models. However, since the initial distribution of data is not controllable, in the scenario where the coverage of local models is not suitable for model integration, i.e. they are not covering the common target area, there is need for a technique for improving their coverage. The techniques that are used in this thesis will be described in Section 4.4.

4.3 Knowledge Probing phase

It is known that learning with a model committee provides a simple way of model integration. However, a common problem of this type of learning is that the committee itself lacks comprehensibility. This is because the combined predictions are derived from the predictions of the local models. Thus, a model committee does not provide insight into how the combined predictions are made. There is a requirement for an integration procedure that does not only integrate predictions of the models in the committee, but also physically integrates them into a single model that provides insight into the problem domain. In DLKP, such an integration is carried out in the Knowledge Probing phase. The Knowledge Probing phase involves two main design choices; a prediction scheme and a probing set selection.

4.3.1 Prediction scheme: A scheme to combine predictive ability of local models

In distributed learning, a set of local models can be used collaboratively to make combined predictions through the use of a *prediction scheme*. A prediction scheme is an approach to make a joint prediction of the local models. This technique has also been adopted in multiple model learning. Many prediction schemes, such as *Uniform Voting* [62], *Trained Predictor* [18], *Likelihood Combination* [1], and *Bayesian Combination* [15, 2], have been proposed. A comprehensive set of prediction schemes can be found in [4]. Short descriptions of some approaches are as follows.

- *Uniform Voting* In this scheme, each local model first gives a class prediction on an unseen instance. All the predictions are assigned equal reliability. The class that is most frequently predicted becomes the final prediction. In the case of tie[2], an arbitration rule has to be adopted.

- *Trained Predictor* In this approach, a special model is learnt to resolve conflict among local models. This can be trained on confusing instances, i.e. the instances that cause disagreement among local models, or on a set of predictions from local models and the correct predictions. This approach is proposed in [90] as model stacking and used in most of Chan & Stolfo's works [17, 18, 20, 21] for distributed classification.

- *Likelihood Combination* In this method, each rule is associated the *degree of logical sufficiency* (LS) [37]. The LS of a rule of $class_i$ is defined as the following ratio of probabilities:

$$\frac{p(rule(x) = true \mid x \in Class_i)}{p(rule(x) = true \mid x \notin Class_i)}$$

where x is a random example. Each probability is estimated using the Laplace method. The LS of the representative rule (the most reliable one)

[2]More than one class has the highest frequency count.

of a class is multiplied by the prior odds of that class. The final prediction is the class that maximises this product.

- *Bayesian Combination* In this approach, each model is weighted by its posterior probability given the training data [15]. In general, the posterior probability of a model is higher if the model fits the training data better [3]. The combined prediction is the class having the highest expected posterior probability.

- *Distributed Summation* In this method [3], the coverage of satisfied rules on the training set are summed. Each rule is attached with a k-component vector, where k is the number of classes. Each component represents the number of training examples of a particular class that are covered by that rule. A vector summation is formed over all rules that match a test example. The combined prediction is the class of the highest component in the summed vector.

These methods can be adopted as a prediction scheme in distributed learning.

In the empirical study in Chapter 5, the *probabilistic prediction* scheme [6, 82] is employed. In probabilistic prediction, each local model gives a prediction as a set of probabilities for each class. The probabilities from all local models are then summed by class. The final prediction c is the class with the highest probability. Given a model M, an instance x with k possible classes, and $\forall j = 1, \ldots, k$ $p_j(x)$ where $p_j(x)$ is a probability of class j being the class prediction of an instance x, a model M makes a prediction by

$$c_M = \underset{j=1}{\overset{k}{argmax}}\ p_j \text{ where } \sum_{j=1}^{k} p_j = 1.$$

By applying the probabilistic prediction scheme to a model committee $\mathcal{M}$ which consists of n local models, the combined prediction can be derived by

$$c_{\mathcal{M}} = \underset{j=1}{\overset{k}{argmax}} \sum_{i=1}^{n} p_{ij} \text{ where } \forall_i \sum_{j=1}^{k} p_{ij} = 1,\ i = 1, \ldots, n.$$

In general, the probabilistic prediction scheme gives a finer prediction than the uniform voting scheme because the class that has the highest vote under the uniform voting scheme might be assigned a lower probability under the probabilistic prediction scheme. For example, given a set of three local models of a classification problem with three possible classes where (p_{i1}, p_{i2}, p_{i3}) are the probabilities that a model i gives to the first, second, and third class, respectively. If $(0.10, 0.40, 0.50)$, $(0.05, 0.40, 0.55)$ and $(0.10, 0.70, 0.20)$ are the predictions from the first, second and third local models, the summed probabilities of the three classes will be $(0.25, 1.50, 1.25)$. Thus, the second class which is the class that has the highest summed probability will be the combined prediction under the probabilistic prediction scheme. However, since the first and the second model assign the highest probability to the third class, the third class is the combined prediction under the uniform voting scheme. This scenario can be interpreted as the model committee predicts the second class with higher confidence than the third class. It should be noted that the probabilistic prediction scheme can be applied to a model committee that, in each prediction, the local models indicate probabilities that they assign to each class.

In this thesis, we mainly focus on decision tree learning algorithm in particular C4.5 [79]. In the prediction process of a given instance, C4.5 generates a set of probabilities that it assigns to each class before it selects the class with the highest probability as the prediction. Therefore, the probabilistic prediction scheme is suitable for our empirical study.

4.3.2 Probing set selection approach

In addition to the prediction scheme, another important design issue in the Knowledge Probing phase is the probing set selection method. Since a global model is learnt from the probing set, the distribution and the size of the probing set have a significant influence on the quality of the global model.

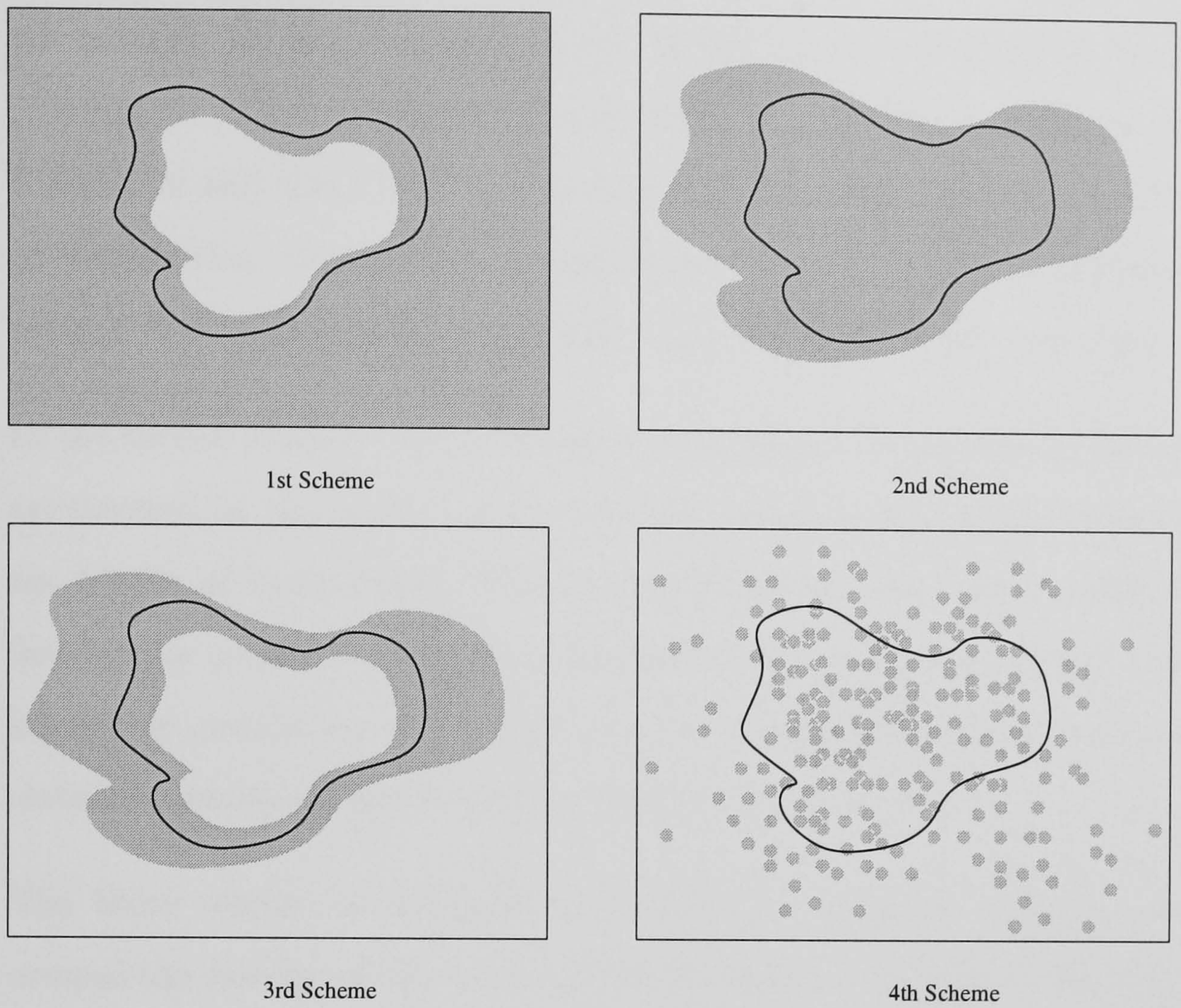


Figure 4.3: Abstractions of the areas in the instance space that are included in the probing set under the four probing set selection schemes. A square represents the instance space. A heart-shape area represents the coverage of the model committee. The shaded area represents the area from which the probing set is selected.

How to select a probing set

The comparison study is carried out to investigate the effect of different probing set selection schemes on the accuracy of the probed models. Because our integration is aimed at joining the coverage of the local models together, we are particularly interested in the area bordering the coverage of the model committee. Thus, in the followings probing set selection schemes, each of them is designed to include different combination of data around the border area in the probing set. The schemes in the study can be separated into two groups. The first three schemes are coverage-based schemes and the fourth scheme is a random selection scheme. The abstractions of the area that will be included in the probing set under each selection scheme are shown in Figure 4.3.

- In the first scheme, the probing set is designed to include instances that

are located outside and around the border of the coverage of the model committee. They are all instances that are *incorrectly* classified by the committee and also the instances that the committee has low confidence in predicting their class values. In margin terms, they are the instances whose margin is *less than* or *equal* 0.2 with respect to the model committee.

- In the second scheme, the probing set is designed to include instances that are covered by the model committee and instances that are located around the border of its coverage. They are the instances that are *correctly* classified by the committee and the instances that the committee has low confidence in predicting their class values. In margin terms, they are the instances whose margin is *greater than* or *equal* to -0.2.

- The third scheme is designed to include the instances that are located *around* the border of the coverage of the model committee. They are the instances that the committee has low confidence in predicting their class values. In margin terms, they are the instances whose margin is between -0.2 and 0.2 inclusive.

- The fourth scheme is designed to include the instances that are randomly selected from all the local data. The size of the probing set is assigned to be the same as the size of each local data.

The experiments were carried out on twelve data sets which were obtained from the UCI Repository of machine learning databases [8][3]. The comparison of the sizes of the probing sets under different schemes is shown in Table 4.1[4]. The percentages of these sizes relative to the sizes of the available training data are shown in Table 4.2. The comparison of the classification errors of the models learnt from the probing sets selected under different probing set selection schemes is shown in Table 4.3.

[3]The descriptions of these data sets can be found in Appendix B.

[4]There is no variation in size of the probing sets selected under the 4th scheme because the size is fixed in advance.

Data Set	Probing set size			
	1st	2nd	3rd	4th
ABALONE	3078.00 ± 91.84	2843.20 ± 30.33	2135.60 ± 50.86	1127.00
GERMAN	286.60 ± 13.26	775.50 ± 8.45	162.10 ± 16.86	270.00
HYPOTHYROID	25.50 ± 4.84	2826.30 ± 1.42	6.10 ± 4.53	853.00
CHESS	54.80 ± 12.98	2857.00 ± 3.37	36.40 ± 12.23	862.00
LED7	549.40 ± 24.66	1438.00 ± 21.86	187.40 ± 43.57	540.00
LETTER	2789.80 ± 53.43	17292.90 ± 45.19	2082.70 ± 78.29	5400.00
NURSERY	628.10 ± 47.40	11361.70 ± 34.92	325.80 ± 62.07	3499.00
SATELLITE	769.80 ± 15.27	5475.70 ± 18.33	455.00 ± 19.57	1737.00
SEGMENT	94.80 ± 9.78	2039.60 ± 6.48	55.40 ± 14.03	623.00
SICK	67.70 ± 5.19	2795.60 ± 4.86	17.60 ± 4.50	853.00
THYROID	30.30 ± 3.71	3379.10 ± 3.84	15.60 ± 6.15	1017.80
WAVEFORM	999.40 ± 22.39	4223.70 ± 19.22	723.10 ± 19.97	1350.00

Table 4.1: Comparison of the sizes of the probing sets under the four probing set selection schemes. The 1st, 2nd, 3rd, and 4th schemes stand for the schemes that select probing sets from training data whose margin is ≤ 0.2, ≥ -0.2, between -0.2 and 0.2 (inclusive), and random, respectively.

From Table 4.3, the probed models from the third and the first scheme, which are the schemes that select probing sets from the area that is outside the main coverage of the model committee, appear to have the highest and the second highest error rate, respectively. Each probing set under the third scheme is a subset of the corresponding probing set under the first scheme. The error of the probed models from both schemes is high because their probing sets do not include the data that are the core of the formation of the model (the main area of the coverage). This behaviour can be caused by the nature of data distribution in a data set and by the nature of the inductive learning algorithm. Generally, data distribution in a data set tends to be dominated in some areas, such as the target area, in the instance space. The learning algorithm aims at detecting this distribution and producing a model whose border of its coverage separates the highly populated area from the lower one. Thus, the border area is likely to contain relatively small amounts of data. Table 4.1 and Table 4.2 shows that in eleven out of twelve data sets, the probing set from the third scheme appears to be smallest in size. The third scheme is clearly not an approach of choice.

In the first scheme where the probing data includes the instances in the border

Data Set	Probing set size (%)			
	1st	2nd	3rd	4th
ABALONE	81.88	75.64	56.81	29.98
GERMAN	31.84	86.17	18.01	30.00
HYPOTHYROID	0.90	99.31	0.21	29.97
CHESS	1.91	99.34	1.27	29.97
LED7	30.52	79.89	10.41	30.00
LETTER	15.50	96.07	11.57	30.00
NURSERY	5.38	97.41	2.79	30.00
SATELLITE	13.29	94.56	7.86	29.99
SEGMENT	4.56	98.10	2.66	29.97
SICK	2.38	98.23	0.62	29.97
THYROID	0.89	99.56	0.46	29.99
WAVEFORM	22.21	93.86	16.07	30.00

Table 4.2: Percentages of the sizes of the probing sets under the four probing set selection schemes in relative to the sizes of the available training data. The 1st, 2nd, 3rd, and 4th schemes stand for the schemes that select probing sets from training data whose margin is ≤ 0.2, ≥ -0.2, between -0.2 and 0.2 (inclusive), and random, respectively.

area and the instances in the area that is not covered by the model committee, the quality of its probed models depends on the distribution of data in that area. This is observed from Table 4.3 that, comparing the error of the first scheme to the corresponding error of the fourth scheme, the probed models of the first scheme seem to have higher error if its probing set is smaller. In the GERMAN and LED7 data sets, where the probing sets of both schemes are comparable in size, their error rates are comparable. This means that to probe a model using data that are located outside the main coverage of the model committee can yield a reasonably good model if the data distribution around such an area is sufficiently significant.

The probed models of the second and the fourth scheme appear to have the lowest and the second lowest error rate, respectively. In the second scheme, the probing set is selected from the area that is under the main coverage of the model committee and from its border area. In the fourth scheme, the probing set is randomly selected from the local data. Between these two schemes, although

Data Set	Classification error of different schemes(%)			
	1st	2nd	3rd	4th
ABALONE	76.23 ± 1.3873	75.76 ± 2.1078	76.23 ± 2.8585	75.78 ± 2.5183
GERMAN	30.40 ± 4.0607	27.60 ± 3.8355	46.70 ± 7.2579	29.50 ± 4.9272
HYPO.	5.86 ± 4.1908	1.15 ± 0.5148	41.63 ± 45.9653	1.15 ± 0.5759
CHESS	52.30 ± 3.4104	1.22 ± 0.7671	48.11 ± 8.1214	2.21 ± 0.9960
LED7	29.50 ± 3.1535	27.60 ± 2.6854	64.40 ± 8.1711	28.00 ± 2.5055
LETTER	60.09 ± 2.9471	14.83 ± 1.0594	66.86 ± 2.3675	20.62 ± 0.8991
NURSERY	58.09 ± 4.0021	4.92 ± 0.4566	66.64 ± 3.3586	6.43 ± 0.7025
SATELLITE	34.50 ± 4.7591	13.60 ± 1.1235	43.69 ± 5.8198	15.13 ± 1.2428
SEGMENT	56.15 ± 11.8776	4.94 ± 0.9204	69.60 ± 10.5883	6.60 ± 1.5253
SICK	14.62 ± 5.8780	2.24 ± 1.0002	40.71 ± 30.5067	2.74 ± 1.4144
THYROID	37.90 ± 29.7912	0.65 ± 0.4327	33.72 ± 27.8095	0.80 ± 0.5735
WAVEFORM	40.86 ± 3.6515	23.14 ± 1.0330	46.14 ± 3.4268	25.08 ± 1.9025

Table 4.3: Comparison of the classification error of the probed models of the four probing set selection schemes. The 1st, 2nd, 3rd, and 4th schemes stand for the schemes that select probing sets from training data whose margin is ≤ 0.2, ≥ -0.2, between -0.2 and 0.2 (inclusive), and random, respectively.

the probed models of the second scheme have lower error rate in every data set, the difference in their error is not very significant. In addition, it is not surprising that the probed models of the second scheme appear to have lowest error rate because its probing set includes data whose distributions are significant enough to be observed by the learning algorithm. It can be seen from Table 4.1 and Table 4.2 that in all but one data sets, the probing sets selected under the second scheme are the largest in size. This is especially the case when the same learning algorithm is used in both phases of DLKP[5]. Since the same inductive bias is used in the learning, the data that are previously filtered by the model committee from the Local Model Learning phase would make the learning in the Knowledge Probing phase easier.

For the random selection scheme (the fourth scheme), the errors of the probed models are not significantly higher than those of the second scheme, but its probing sets are much smaller in size. This is because random examples not only provide the learning algorithm with information about the current coverage

[5]which is the scenario in this study.

of the model, but they also provide the information about the distribution of future test instances [43]. The investigation reveals that, on average, the error of the probed models from the second scheme is 1.37% lower than that of the fourth scheme, but with the cost of 284.66% larger probing sets. In distributed data mining, this means that, between the two schemes, the computation and communication overhead of the second scheme will be at least three time higher for a gain of only 1.37% in accuracy. The fourth scheme, therefore, appears to be the proper approach.

The conclusion drawn from the study of the four probing set selection schemes is that a margin should not be used as a criterion for selecting a probing set. This is because different data sets tend to have different forms of data distributions. To employ a certain margin range as a criterion for selecting a probing set can make the approach unstable and can lead to a variation in the results. Some margin-based schemes, such as the second probing set selection scheme where probing data are selected from the main area of the coverage of the model committee, can yield a low classification error. However, the investigation reveals that the cost of its application is much more significant than the improvement in accuracy gained. Therefore, the random selection technique appears to be the most reasonable approach both in terms of accuracy and the computation and the communication cost.

What is the optimal size of a probing set ?

Since we choose to randomly select the probing set, the next design choice is the size of the probing set. In the design of DLKP, a subset of the training data is used as a probing set. Although to use a subset of the training data to derive a global model is seen as not making full use of the available data, there are at least three reasons for this choice.

- There is evidence [36, 73] that models that are learnt from subsets of available data are often just as accurate as models that are learnt from all the available data. In general, increasing training set size is only useful up to

the point where the accuracy becomes asymptotic [34]. That point is different for each problem domain and an approach to specify this is still an issue of research [51, 73].

- The second reason concerns the applicability of the approach under distributed data mining. To reduce the communication and computation overhead, the size of the probing set should be relatively small. If all the available training data has to be used as a probing set, e.g. as in Domingos' knowledge acquisition approach [34], then the approach will be, at least, equal to collecting all data together and performing a learning procedure at one single site.

- The third reason concerns the fairness of comparison in our empirical study. To set a standard for comparison of the effectiveness of the approach, the size of the probing set should not be much larger than the average size of the local data. If the probing set is assigned to be much larger, it is not clear whether the increased accuracy is a result of the additional training data or as a result of DLKP itself.

Therefore, the size of the probing set is set to be equal to the size of each local data plus 15% randomly selected data. An additional 15% of data is added to the probing set to increase the chance that the instances will be selected from different parts of the coverage of the model committee.

In this section, we have discussed two important design decisions in the Knowledge Probing phase; the prediction scheme and the probing set selection scheme. The Knowledge Probing phase can be directly applied to a model committee whose local models are consistent to certain levels. In the scenarios where local models are not consistent, some techniques are required to increase their consistency so that they become more suitable for model integration. In the next section, we propose two techniques for improving the consistency of local models.

4.4 Techniques for adjusting local data distributions

In this thesis, local data distributions are adjusted on the basis of the coverage of the model committee. This is based on the observation that too broad coverage may cover data points (e.g. noise) that can degrade the fit of the model committee, and too narrow coverage may neglect data points that increase confidence of the fit of the model committee. We propose the use of a margin as a tool for adjusting local data distributions. In our approaches, data whose margins of their class predictions are between some ranges are actively selected for use in the data distribution adjusting process. This margin-driven active sampling provides the means for querying the environment for new information, such as instances that will increase the confidence of the coverage of the model committee. In the next two subsections, two techniques that we use to adjust distributions of local data; *data redistribution* and *data contraction* techniques are described.

4.4.1 Data redistribution technique

In the scenario where the model committee is fairly accurate, but the confidence in its predictions is quite low. It can be the case that models in the committee are covering the target areas in the instance space, but they are not consistent i.e. they do not share a common target area. In coverage terms, this means the majority of the coverage of the model committee falls in the margin interval that is less than 0.8, but most of the interval is still in positive margin area, such as Figure 3.7 (c). In this scenario, we propose *data redistribution* as a technique for adjusting local data distributions. The algorithm of data redistribution technique is shown in Figure 4.4. Given a model committee and a set of local training data, we first randomly select a consistency test set from the local training data. The size of the consistency test set is assigned to be equal to the average size of the local training data. The model committee is used to predict class values of instances in the consistency test set. The instance whose margin of the class pre-

Input:
A model committee $\mathcal{M}_0$ of n local models M_0,
n sets of local data S_0 of size m,
a function fm to calculate margin,
a learning algorithm L that is used in the Local Model Learning phase,
a procedure KP that is used to probe knowledge from a model committee,
a procedure E for evaluating the error rate of a model.

Select a consistency test set:
Randomly select a set of m instances from n local data sets.

$\{CS \mid cs_i, i = 1, \ldots, m \text{ and } cs_i \in x_{jk},\ j = 1, \ldots, n,\ k = 1, \ldots, m\}$

Repeat (at time t):
/*Select data to be distributed i.e. data whose margin is between -0.2 and 0.2*/
$R_t = \emptyset$
for $i = 1$ to m {
$margin_i = fm(\mathcal{M}_{t-1}, cs_i),\ cs_i \in CS$
if $margin_i \geq -0.2$ and $margin_i \leq 0.2$ {
$R_t = R_t + \{cs_i\}$
}
}

/*Redistribute the selected data to every local data sets*/
for $i = 1$ to n {
$S_{t,i} = S_{t-1,i} + R_t$
}

/*Re-learn local models and form a new model committee*/
for $i = 1$ to n {
$M_{t,i} = L(S_{t,i})$
}
$M_{t,i} \in \mathcal{M}_t,\ i = 1, \ldots, n$

/*Probe model from the model committee and find the error rate*/
$PM_t = KP(\mathcal{M}_t)$
$Error_t = E(PM_t)$
Until
$R_t \subseteq \emptyset$ or $Error_t > Error_{t-1}$

Output:
The probed model PM_{t-1}.

Figure 4.4: The algorithm of the data redistribution technique.

diction is between −0.2 and 0.2 (inclusive) is redistributed to each local training data. These expanded local training data are used to re-learn a new set of local models, which is later used to probe a probed model. The process is iterated until there are no more instances to be redistributed or the error of the probed model increases.

This technique is designed to increase the consistency among the local models by expanding the joint coverage around the border area of each local model. The expansion is achieved by adding the instances that lie around the border of the coverage of the model committee to each local data set. In margin terms, those instances are instances whose margins of their class predictions are between −0.2 and 0.2 inclusive. These instances are considered as "near miss" instances that can be used to place stronger constraints on the learning process than randomly selected examples [69].

One may question why using a margin as a criterion for selecting data to be redistributed is better than simply redistributing instances that cause conflict among local models. It should be noted that the aim of the data redistribution technique is to increase the consistency among the local models. The "consistency" in this context does not mean that the local models have to be the same. They can be different, but when working together, there should be confident in their predictions on different parts of the instance space. Redistributing conflicting instances can lead to situations where the local models agree with their predictions on every part of the instance space, i.e. they are more or less the same. In such a scenario there will be no benefit from model integration.

4.4.2 Data contraction technique

In the scenario where the model committee is inaccurate and has low confidence in its predictions, it can be the case that the model committee does not cover the right area in the instance space. In coverage terms, this means that the majority of the coverage of the model committee falls in the negative margin range. Such a coverage can be a result of noise or the mismatch of the problem domain and the

inductive bias in the learning algorithm. This scenario can be observed from the margin distribution graph whose slope is located in the area where the margin is less than 0.8 and most of the slope is in the negative margin area. We propose *data contraction* as a technique for adjusting local data distributions in this scenario. The algorithm of data contraction technique is shown in Figure 4.5. In the data contraction technique, each local training set is used as its local consistency test set. At each local site, the local model predicts class values of its training set. The instance whose margin of the class prediction is between -0.2 and 0.2 (inclusive) is removed from the local training set. The contracted local training set is used to re-learn a new local model. The new local model from each local site forms a new model committee that is later used to probe a probed model. The process is iterated until there are no more instances to be contracted or the error of the probed model increases.

The data contraction technique is opposite to the data redistribution technique. Instead of expanding the coverage of the local models, this is designed to gradually move the coverage of the model committee to different areas by contracting the coverage of each local model from the wrong coverage area i.e. the area that is not the target area. For each local model, the contraction is achieved by removing local instances whose prediction margin is between -0.2 and 0.2 inclusive[6] from the local data. This technique can also be seen as a means of removing examples that are considered ambiguous or difficult to learn[7] from the training data.

4.5 Summary

In this chapter, the three main components of DLKP are discussed. The application of these components involves many design choices and the effectiveness of DLKP depends on the chosen choice of each component.

For the Local Model Learning phase, the local models should cover a common

[6]With respect to the local model.
[7]With respect to the current learning algorithm.

Input:

A model committee $\mathcal{M}_0$ of n local models M_0,
n sets of local data S_0 of size m,
a function fm to calculate margin,
a learning algorithm L that is used in the Local Model Learning phase,
a procedure KP that is used to probe knowledge from a model committee,
a procedure E for evaluating the error rate of a model.

Repeat (at time t):

/*From each local data, select data to be contracted
i.e. data whose margin is between -0.2 and 0.2*/
for $i = 1$ to n {
$C_{t,i} = \emptyset$
for $j = 1$ to m {
$margin_j = fm(M_{t-1,i}, s_{ij}),\ s_{ij} \in S_{t-1,i}$
if $margin_j \geq -0.2$ and $margin_j \leq 0.2$ {
$C_{t,i} = C_{t,i} + s_{ij}$
}
}
}

/*Contract the selected data from every local data sets*/
for $i = 1$ to n {
$S_{t,i} = S_{t-1,i} - \{C_{t,i}\}$
}

/*Re-learn local models and form a new model committee*/
for $i = 1$ to n {
$M_{t,i} = L(S_{t,i})$
}
$M_{t,i} \in \mathcal{M}_t,\ i = 1, \ldots, n$

/*Probe model from the model committee and find the error rate*/
$PM_t = KP(\mathcal{M}_t)$
$Error_t = E(PM_t)$

Until:

$C_{t,i} \subseteq \emptyset,\ i = 1, \ldots, n$ or $Error_t > Error_{t-1}$

Output:

The probed model PM_{t-1}.

Figure 4.5: The algorithm of the data contraction technique.

target area, but to maximise the benefit of model integration, there should be a variance in their coverage.

For the Knowledge Probing phase, two main design choices involved are a prediction scheme and a probing set selection. The prediction scheme provides a way to combine the predictive abilities of the local models. A good prediction scheme will lead to an accurate model committee and also a good black box for the Knowledge Probing phase. We use the probabilistic prediction scheme as a prediction scheme in the empirical study in Chapter 5. This is because, despite its simplicity, it generates fine-scale predictions and it is also suitable for the learning algorithm we will use in our empirical study. For the probing set selection scheme, our study shows that a static criterion, such as a certain margin range, should not be used as a probing set selection scheme. This is because data distributions in different problem domains can be very different. To select a probing set by a static criterion can make the framework unstable. Our study shows that to randomly select a probing set from the local data appears to be a reasonable approach with respect to the accuracy of the global models and the computation costs. In the next chapter, we choose to use a probing set that is not much larger than each local data set. Mainly, it is to set the standard of the measurement of the effectiveness of the framework.

Since each local data can have different distributions, in the scenario where this difference causes an inconsistency among local models, we propose the use of the *data redistribution* and the *data contraction* techniques. These two techniques manipulates data that the model committee has low confidence in predicting their class values. The data redistribution technique is aimed at expanding the joint coverage of the models in the committee by redistributing some data to each local data set. In contrast, the data contraction technique is aimed at increasing the accuracy of the model committee by contracting the coverage of each local model from the area that is considered as difficult with respect to the learning algorithm.

In the next chapter, two sets of the empirical studies and their results are presented to justify the design of DLKPand its associated tools and techniques.

Chapter 5

Empirical Studies

Research in machine learning is inherently empirical because the performances of machine learning algorithms are determined by how well their underlying assumptions match the structure of the problem domains. In this chapter, two main experiments and their results are presented to evaluate and to justify the design of DLKP. The analysis concerns three properties of a model: its accuracy, its coverage, and its comprehensibility.

5.1 Introduction

We begin with the design of the framework of the empirical study. Two sets of experiments are then presented. The first set of experiments is aimed at justifying the design of DLKP in two aspects. The first aspect concerns the use of margin distribution graph to observe the coverage of the model committee. The second aspect concerns the improvement in classification accuracy by analysing the coverage and by using data distribution adjusting techniques. The second set of the experiments is designed to observe a contribution from each of the two learning phases of DLKP to the bias and the variance in the final result. This is to aid the understanding of the behaviour of DLKP. The results of the two sets of experiments are presented and discussed.

5.2 Framework of the empirical study

This section describes the framework that is used in the empirical study. This includes some fixed factors in the experiments, such as the data sets, the learning algorithm, and the number of local models. The framework of the empirical study also includes the factors involved in the analysis of the results, such as the criteria of interest in the comparison, the control of the experiments, and the methods of analysis.

5.2.1 Data set selection

The data sets used in the empirical study were obtained from the UCI Repository of machine learning databases [8]. To investigate the generality and the effectiveness of DLKP, we use data sets that vary in size, in number of classes, and in number and type of attributes. We simulate distributed data by mutually exclusively partitioning each data set into several subsets. Thus, in order to provide a reasonable amount of data for local model learning, the data sets chosen are relatively large in size (at least 1,000 instances). Details of the chosen data sets are shown in Table 5.1. The description of each data set can be found in Appendix B.

5.2.2 Parameter setting

Learning algorithms We choose decision tree learning algorithms as the basis for the study. The advantages of decision tree learning algorithms are the comprehensibility of their results, and their speed of learning and evaluation. In addition, they do not require specifications of the underlying parameters, such as a network topology as in neural network learning algorithms. Thus, the experimental results will be general in that they can be applied to broad range of classification algorithms. Moreover, decision trees are known to provide complete coverage on the instance space [27]. This is because they are able to give a class prediction for every instance in the instance space. This property is preferable

Data Set	Size	Attribute		Missing value(%)	#Class
		Discr.	Cont.		
Real World Data Set					
ABALONE	4,177	1	7	0	29
GERMAN	1,000	13	7	0	2
HYPOTHYROID	3,163	18	7	6.74	2
CHESS	3,196	36	0	0	2
LETTER	20,000	0	16	0	26
NURSERY	12,960	8	0	0	5
SATELLITE	6,435	0	36	0	7
SEGMENT	2,310	0	19	0	7
SICK	3,162	18	7	6.74	2
THYROID	3,772	22	7	5.54	5
Artificial Data Set					
LED7	2,000	7	0	0	10
WAVEFORM	5,000	0	40	0	3

Table 5.1: Details of the data sets used in the experiments.

for a coverage-based framework, such as DLKP, because it enables observation on parts of the instance space that have not yet been correctly covered by the model.

In this study, C4.5 [79], a decision tree learning algorithm is used as a learning algorithm in both phases of learning. C4.5 is one of the most well-known learning algorithms in machine learning study and in practical applications. Since it has been widely used, C4.5 constitutes a good basis for the empirical study.

Number of local models A number of local models is chosen to be four. This is because we use a data partitioning technique to simulate distributed data sets, but the data sets used in the experiments are relatively small compared to real world data sets. To provide a reasonable amount of data for local model learning, the number of local models has to be small.

5.2.3 Comparison criteria

In this chapter, the results of DLKP will be compared against the corresponding results of the Local Model Learning phase (the local models) and against those

of the centralised learning approach.

The comparison between the results of DLKP and the average results of the Local Model Learning phase represents the comparison between the *integrated* models and the *non-integrated* models (the local models). According to the setting of the experiments, these two groups of models are learnt from data that are about the same size.

The comparison between the results of DLKP and the centrally learnt models represents the comparison between the *model integration* approach (DLKP) and the *data integration* approach. According to the setting of the experiments, the data that are used to train the centrally learnt model are four time larger than those used to train the final model of DLKP (the probed model).

Hence, these two standards represent the minimum and the maximum goals for DLKP, respectively. That is, the result of DLKP should be, at least, as good as the average result of the Local Model Learning phase, and, at most, comparable or better than the centrally learnt model.

5.2.4 Control of the experiments

To control the experiments so that the results can be obtained in the statistically sound fashion, the experiments of DLKP and the centralised learning approach are performed under the same environment. That is, for each pair of the experiments comparing the two approaches, two learning procedures are performed on the same training set and test set. We use 10-fold cross validation[1] to approximate the accuracy of the results.

5.2.5 Methods of analysis

The details of some methods of analysis were given in Chapter 2. The following is a brief summary of all the methods that are used in the analysis of this chapter.

[1]The details of l-fold cross validation were given in Section 2.7.3.

Accuracy and model size As described in Chapter 2, the accuracy is defined as the percentage of the instances in a test set that are correctly classified by the model. As this thesis focuses on classification data mining, classification accuracy is the most important criterion in the comparison.

Apart from accuracy, another property of a model is its comprehensibility. Generally, the comprehensibility of a model is difficult to measure because it is highly subjective. Although not perfect, model size can be used to partly represent the comprehensibility of the model because the compactness of the model usually helps comprehensibility [9, 35, 55].

Margin and margin distribution graph The margin of a prediction conveys information of how confident a model or a model committee is in making a prediction. The margin can be used to extend an analysis of the performance of a model and a model committee on the instance space because a model that is highly accurate may have a poor margin distribution. This means although accurate, its confidence in the predictions in general is low. Thus, the model is considered unstable. Therefore, the analysis of margin is used to enhance the analysis of the empirical results. While the margin of a prediction shows the predictive behaviour of the model on a single instance, a margin distribution graph shows the predictive behaviour of the model on a set of data. The details of margin and margin distribution graph can be found in Section 2.5.

Model coverage A model coverage can be interpreted from the main slope in a margin distribution graph. The percentage of the main coverage of the model is the difference of the y value when the main slope is projected on the Y-axis. The margin range of the main coverage of the model is the range on the X-axis where the main slope falls in. This can be obtained by projecting the main slope on the X-axis. The coverage of the model can be used to aid the understanding of different forms of coverage of a model committee in distributed data mining. The discussion of model coverage can be found in Section 2.4.

Pairwise t-test Since the data sets are different in size, number and type of attributes, and number of classes, a statistical approach is needed to provide a standard of comparison between the two sets of results. Although statistical models rarely perfectly fit the practical constraints in testing learning approaches when available data are limited [72], they do provide approximate confidence intervals that can help in interpreting experimental results. In this study, the pairwise t-test is used to statistically compare the two sets of results. In the pairwise t-test, two learning procedures are compared over different samples of data from the same problem domain. The difference between the two sets of results are regarded as a set of univariant data that is used to calculate the mean, the deviation and the t-value. With a specific level of confidence, the t-value is used to decide whether to accept or to reject the null hypothesis that the two learning approaches are not significantly different. In the empirical study, we perform the two-sided t-test using a 95% level of confidence. The procedure of the pairwise t-test for comparing learning algorithms was discussed in Section 2.7.3.

5.3 Study of the effectiveness of DLKP

The effectiveness of DLKP is measured by the improvement in accuracy by the final model of DLKP over the average accuracy of the corresponding local models, which is a minimum requirement of DLKP. We claim that the effectiveness of DLKP depends on the form of the coverage of the models in the committee. This study is aimed at justifying our claim. The study is also designed to justify the use of the analysis of the coverage of the model committee to guide an adjustment in local data distributions.

5.3.1 Design of the experiments

Each experiment is separated into two phases, the Local Model Learning phase and the Knowledge Probing phase. The data is prepared for ten-fold cross validation.

Local Model Learning: Each training set is divided into four roughly equal subsets by mutually exclusively partitioning. Given a learning algorithm L, each subset is used to learn a local model.

Knowledge Probing: The probing set consists of instances that are randomly selected from the training set. The size of the probing set is equal to the average size of the local data plus an additional 15%. The set of four local models from the Local Model Learning phase is used to give four sets of predictions on the probing set. A prediction scheme PS is used to combine those sets of predictions into a single set of combined predictions. The probing set whose class values are reassigned by the combined predictions is used as the training data for L to learn a final model. The final model is evaluated on the test set. For the rest of the chapter this model is referred to as a probed model.

In the experiments, C4.5 is used as a learning algorithm (L). The probabilistic prediction is used as a prediction scheme (PS).

5.3.2 Margin distribution graph: The indicator of the effectiveness of DLKP

The results of this set of experiments are shown in Table 5.2[2]. The results from Table 5.2 can be categorised into two groups. The first group consists of the data set where the error in its probed model is not significantly different from the average error of the local models. The difference is in the range of 0.20%. This group consists of the results from HYPOTHYROID, CHESS, SICK, and THYROID data sets (the top part of Table 5.2). The second group consists of the data set where the error of its probed model is significantly different from the average error of the local models. The difference is in the range of 3.48%. The second group consists of the results from ABALONE, GERMAN, LED7, LETTER,

[2]The higher variance in the classification error of the probed models (the third column of Table 5.2) is the matter of the design of the experiments. That is, each reported error of the local models was averaged over 40 models where that of the corresponding probed model was averaged over 10 models. We are more interested in the significance of the difference in their errors (the fourth column of Table 5.2).

Data Set	Classification error (%)		Confident level
	Average of local models	Probed model	
HYPOTHYROID	1.3240 ± 0.3390	1.1500 ± 0.5759	0.8131
CHESS	2.0830 ± 0.6135	2.2100 ± 0.9960	0.3863
SICK	2.7690 ± 0.9637	2.7400 ± 1.4144	0.1140
THYROID	1.0030 ± 0.4155	0.8000 ± 0.5735	0.9093
ABALONE	79.2580 ± 0.8964	75.7800 ± 2.5183	0.9994
GERMAN	30.3750 ± 1.5195	29.5000 ± 4.9272	0.4237
LED7	29.3390 ± 2.7170	28.0000 ± 2.5055	0.9928
LETTER	21.3170 ± 0.9652	20.6200 ± 0.8991	0.9798
NURSERY	7.1370 ± 0.4117	6.6800 ± 0.6015	0.9966
SATELLITE	16.5320 ± 0.5699	15.1300 ± 1.2428	0.9992
SEGMENT	6.9560 ± 1.0572	7.5500 ± 1.4539	0.7822
WAVEFORM	26.6750 ± 0.8804	25.0800 ± 1.9025	0.9646

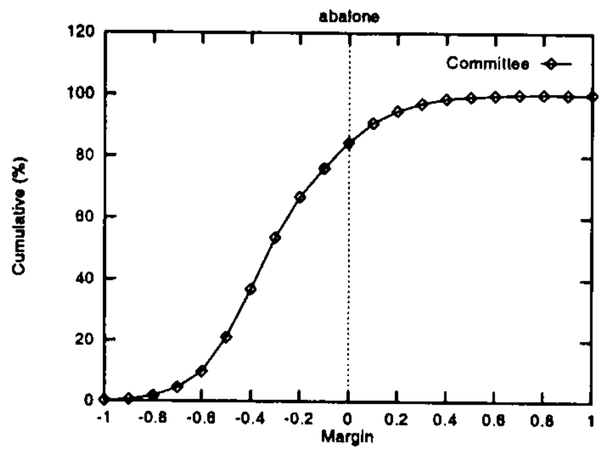
Table 5.2: Comparison between the classification error of each probed model and the average classification error of the corresponding local models.

NURSERY, SATELLITE, SEGMENT, and WAVEFORM data sets (the bottom part of Table 5.2).

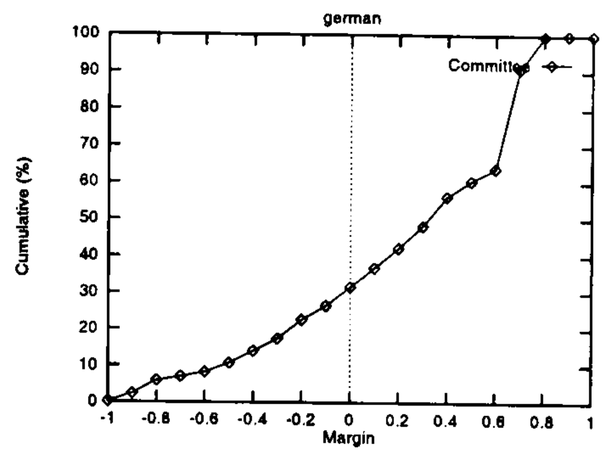
The corresponding margin distribution graphs of the model committees of the data sets in Table 5.2 are shown in Figure 5.1. This investigation reveals that the shapes of the margin distribution graphs are related to the performance of the probed models in these two groups of data sets.

In the case where the shape of margin distribution graph is close to the ideal shape[3], such as Graph (c), (d), (j), and (k) in Figure 5.1, despite the fact that three out of the four probed models have a lower error than the average errors of the corresponding local models, the improvement in the error is limited and the level of confidence in the difference of error is relatively low. The ideal margin distribution graph implies that the model committee is highly accurate and the local models are highly consistent and accurate. This is consistent with our analysis of margin distribution graph in Section 3.5 that in the case where the model committee has an ideal margin distribution graph, the effectiveness of DLKP is limited.

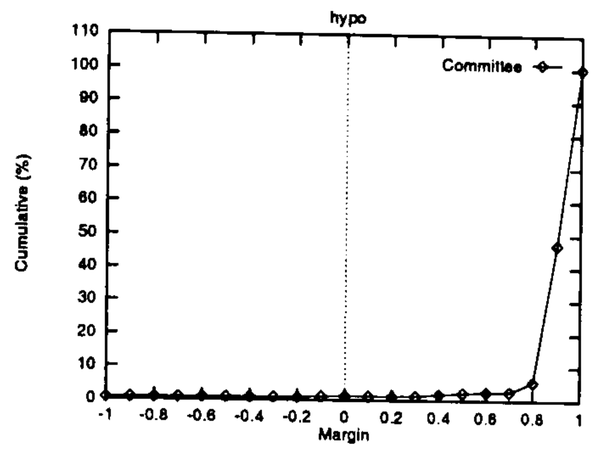
[3]With respect to the discussion in Section 2.5.



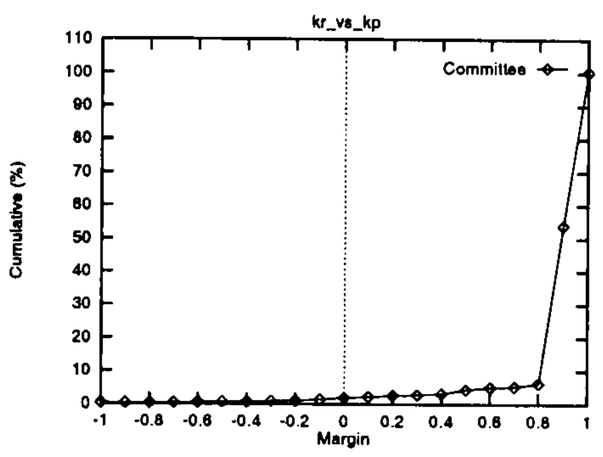
(a) ABALONE



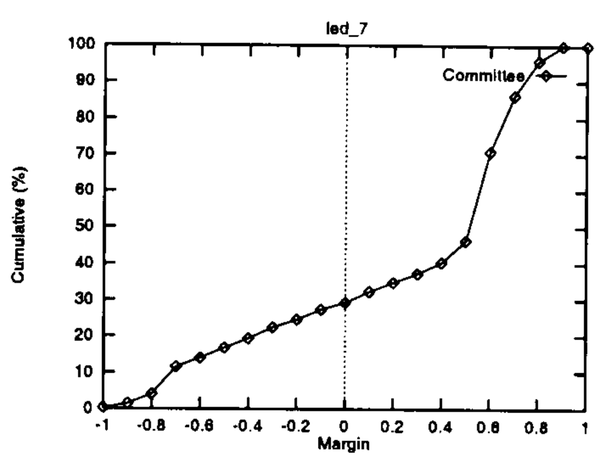
(b) GERMAN



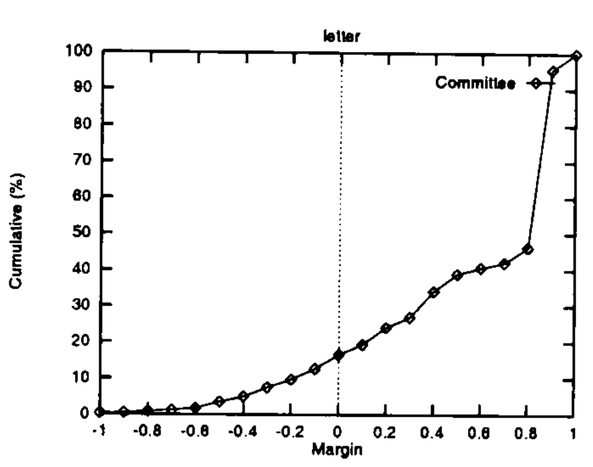
(c) HYPOTHYROID



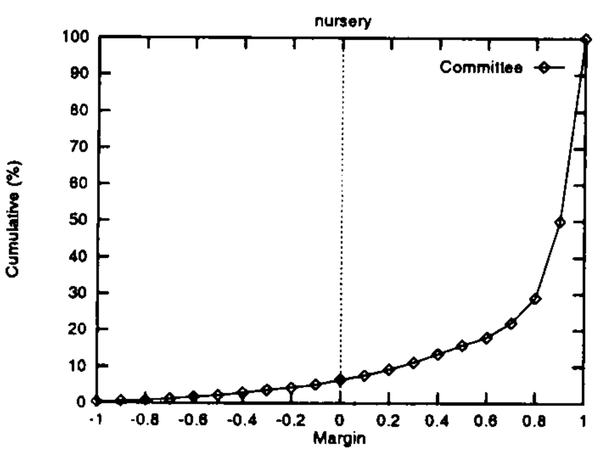
(d) CHESS



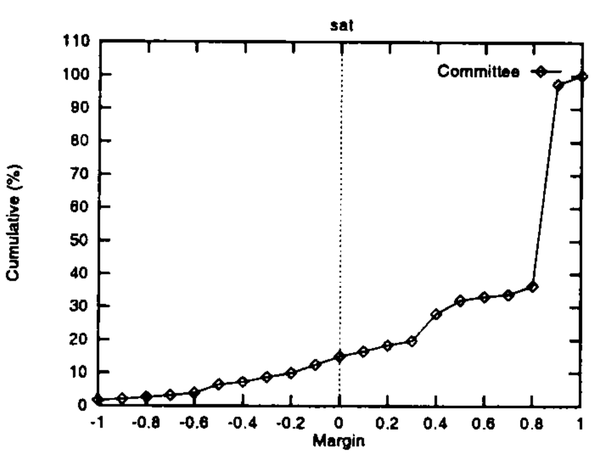
(e) LED7



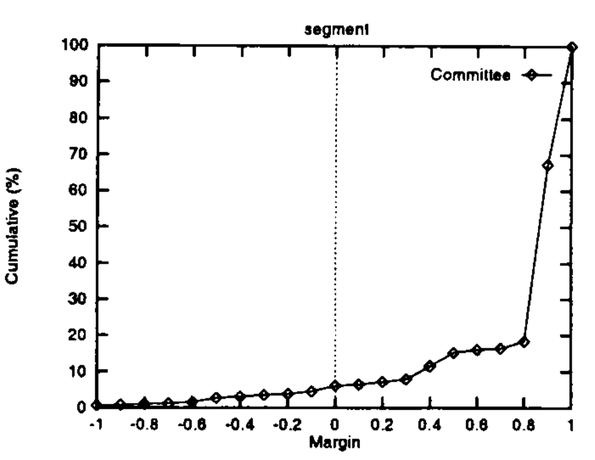
(f) LETTER



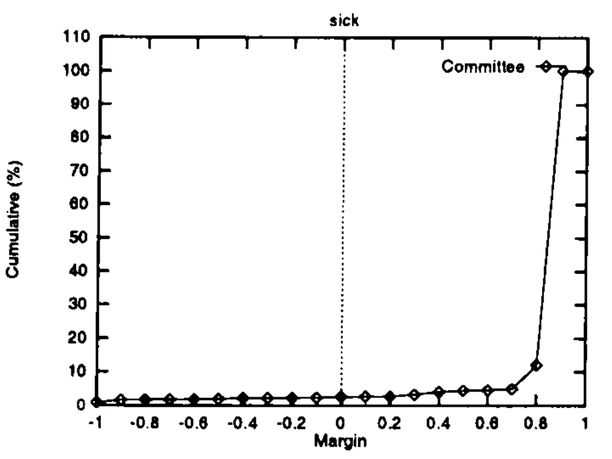
(g) NURSERY



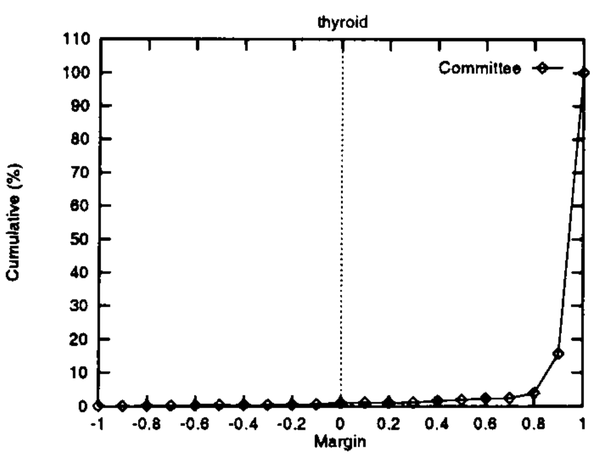
(h) SATELLITE



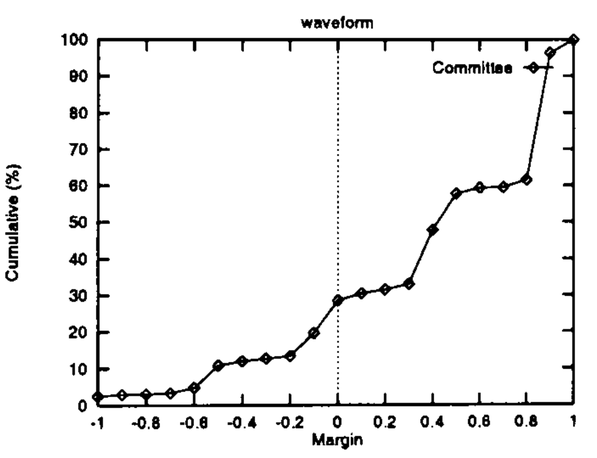
(i) SEGMENT



(j) SICK



(k) THYROID



(l) WAVEFORM

Figure 5.1: Margin distribution graphs of the twelve model committees on the test sets.

Data set	Graph label	Main coverage (%)	Margin range
ABALONE	(a)	90	-0.6 - 0.3
GERMAN	(b)	37	0.6 - 0.8
HYPOTHYROID	(c)	95	0.8 - 1.0
CHESS	(d)	94	0.8 - 1.0
LED7	(e)	50	0.5 - 0.8
LETTER	(f)	54	0.8 - 1.0
NURSERY	(g)	70	0.8 - 1.0
SATELLITE	(h)	64	0.8 - 1.0
SEGMENT	(i)	80	0.8 - 1.0
SICK	(j)	90	0.8 - 1.0
THYROID	(k)	97	0.8 - 1.0
WAVEFORM	(l)	40	0.8 - 1.0

Table 5.3: Analysis of the slope of the twelve model committees. Graph labels are based on those used in Figure 5.1

In the scenario where the margins of the predictions of the model committee are not ideally distributed, such as Graph (a), (b), (e), (f), (g), (h), (i) and (l) in Figure 5.1, the probed model appears to have a wider gap for improvement. Table 5.2 shows that in seven out of the eight data sets the errors of the probed models are lower than the average errors of the local models. In six out of the seven data sets (except the GERMAN data set), the probed models perform uniformly better with more than 95% confidence. The reason for this is the opposite of that for the first group of data sets. An imperfect margin distribution graph implies that the coverage of each local model is different and integrating them can, therefore, lead to an improvement in accuracy over an individual local model.

5.3.3 Improvement in accuracy by the analysis of model coverage

Applying the analysis of margin distribution graph to the twelve model committees in Figure 5.1, the coverage of each committee is shown in Table 5.3. According to the approach discussed in Section 3.5, the coverage of the twelve model committees in Table 5.3, can be categorised into four main groups.

- The first group consists of the model committee where more than 90% of its coverage has margin ranging between $0.8 - 1.0$. This group consists of the model committees of HYPOTHYROID (c), CHESS (d), SICK (j) and THYROID (k) data sets. The model committees of these data sets have ideal margin distribution graphs.

- The second group consists of the model committee where the margin of the main coverage ranges between $0.8 - 1.0$, but the percentage of the instances under this margin range is less than 90%. This group consists of the model committees of the LETTER (f), NURSERY (g), SATELLITE (h), SEGMENT (i) and WAVEFORM (l) data sets.

- The third group consists of the model committee whose margin of the main coverage is less than 0.8, but most of it is still in the *positive* margin area. This group consists of the model committee of the GERMAN (b) and LED7 (e) data sets.

- The fourth group consists of the model committee whose margin of its main coverage is less than 0.8 and most of it is in *negative* margin area. This group consists of the model committee of the ABALONE (a) data set.

Our analysis of model coverage in Section 3.5 suggests that for the first group of model committees, the final results can be arbitrarily selected from the local models. The results of the arbitrarily selection[4] of the final models for the four model committees in this group are shown in Table 5.4. Table 5.4 shows that with the arbitrarily selected models, the difference between the error of DLKP and the average error of the local models increases from 0.20% to 1.58%.

For the second group of model committees, our analysis in Section 3.5 suggests that the final model can be obtained by directly applying the Knowledge Probing phase. Table 5.2 shows that, in the second group, the improvement in classification error by DLKP over the average error of the local models is in the

[4]In this thesis, the local model that has the lowest classification error is chosen.

Data Set	Classification error (%)		
	Average local models	Probed model	Arbitrarily selected model
HYPOTHYROID	1.3240 ± 0.3390	1.1500 ± 1.5759	0.3000 ± 0.4830
CHESS	2.0830 ± 0.6135	2.2100 ± 0.9960	0.5000 ± 0.8498
SICK	2.7690 ± 0.9637	2.7400 ± 1.4144	1.6000 ± 1.0750
THYROID	1.0030 ± 0.4155	0.8000 ± 0.5735	0.1000 ± 0.3162

Table 5.4: Comparison of classification error between the average error of the local models, the probed model and the arbitrarily selected model of the data set whose local models are consistently accurate.

range of 1.59%. Note that in this analysis, although the SEGMENT data set is classified in the second group, its property is close to the data sets in the first group. That is, its main coverage covers 80% of instances with margin ranging between 0.8 to 1.0. This can be the reason why the probed model of SEGMENT data set has a higher error than the average error of the corresponding local models.

In the third and the fourth groups of model committees, the fact that their main coverage has margin lower than 0.8 implies that they are not covering the instance space with high confidence. This could be because the learning algorithm has explored the wrong instance space due to noise in the training data (e.g. the LED7 data set where 10.24% of the attribute values are noise) or due to an incompatibility between inductive bias and the problem domain (e.g. the ABALONE data set whose the average classification error of the models that are learnt by C4.5 is 79.26%). Our analysis in Section 3.5 suggests the application of the data redistribution technique to the model committee that has most of its main coverage in positive margin area, such as GERMAN and LED7 data sets and the application of the data contraction technique to the model committee that has most of its main coverage in negative margin area, such as ABALONE data set. The results of the application of these data distribution adjusting techniques[5] are shown in Table 5.5. Compared with the corresponding results in

[5]The details of the data redistribution technique and the data contraction technique can be found in Section 4.4.

Data Set	Classification error (%)		Confident level
	Average of local models	Probed model	
ABALONE	79.2580 ± 0.8964	74.7600 ± 1.5378	1.0000
GERMAN	30.3750 ± 1.5195	27.7000 ± 3.1287	0.9491
LED7	29.3390 ± 2.7170	26.9000 ± 2.8363	0.9965

Table 5.5: Comparison between the classification error of each probed model after the adjustment in local data distributions and the average classification error of the corresponding local models.

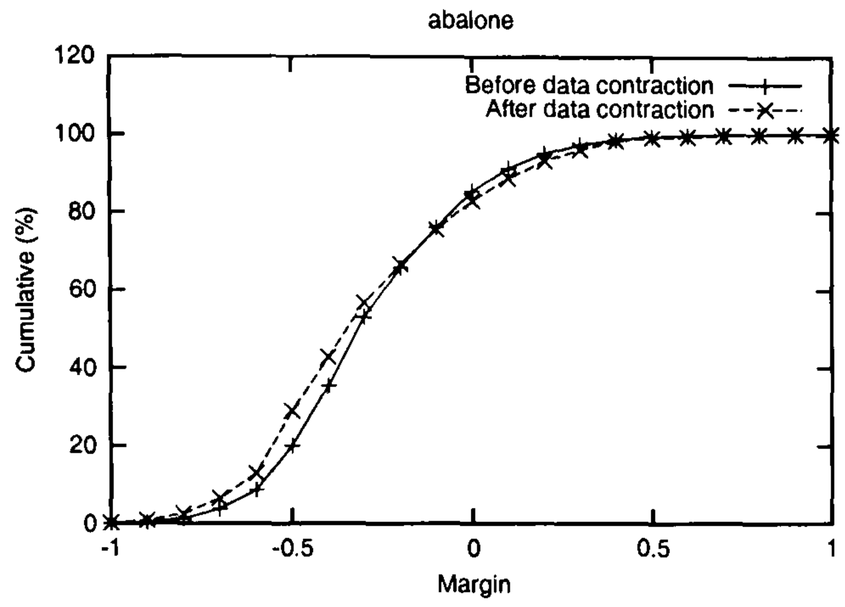
Table 5.2, the adjustment in data distributions leads to a 1.1% – 1.8% further decrease in classification error. This improvement can also be observed Figure 5.2, which is a comparison of margin distribution graphs of the model committee and the corresponding probed model before and after the adjustment in local data distributions. Figure 5.2 shows that the shapes of the margin distribution graphs of the model committees and the probed models after the adjustment in local data distributions are approaching the ideal shape.

5.3.4 Summary of the effectiveness study

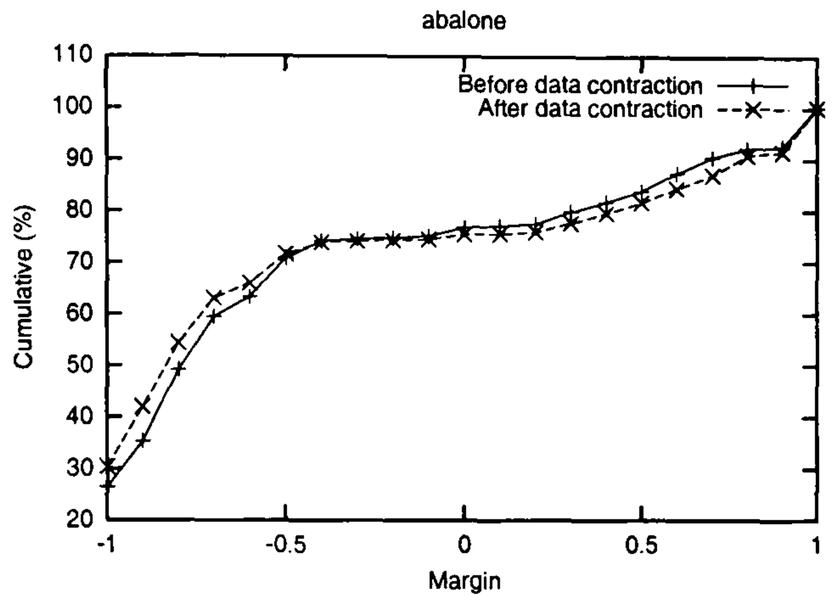
This set of empirical studies justifies our claim that the effectiveness of DLKP depends on the coverage of the local models, which can be observed from the margin distribution graph of the model committee. By analysing their coverage, the model committees of the twelve data sets in the study can be categorised into four main groups. The characteristics of the four groups of model committees and the improvement in classification errors by DLKP and its associated tools and techniques over the average error of the corresponding local models are summarised in Table 5.6. Table 5.6 shows that the minimum goal of DLKP has been achieved.

5.4 Bias and variance decomposition study

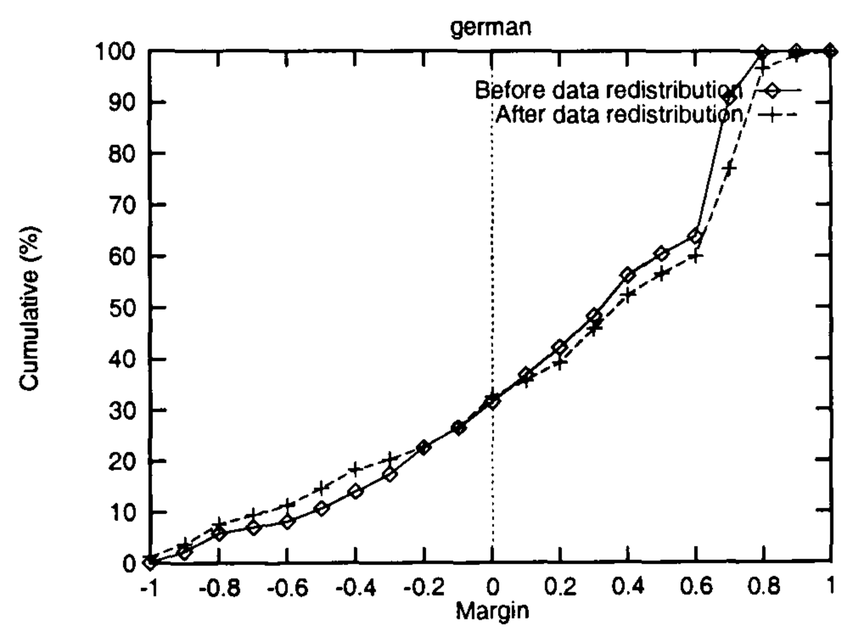
One of the standard approaches for analysing the improvement achieved by a model committee is to separate the expected error of the model committee into



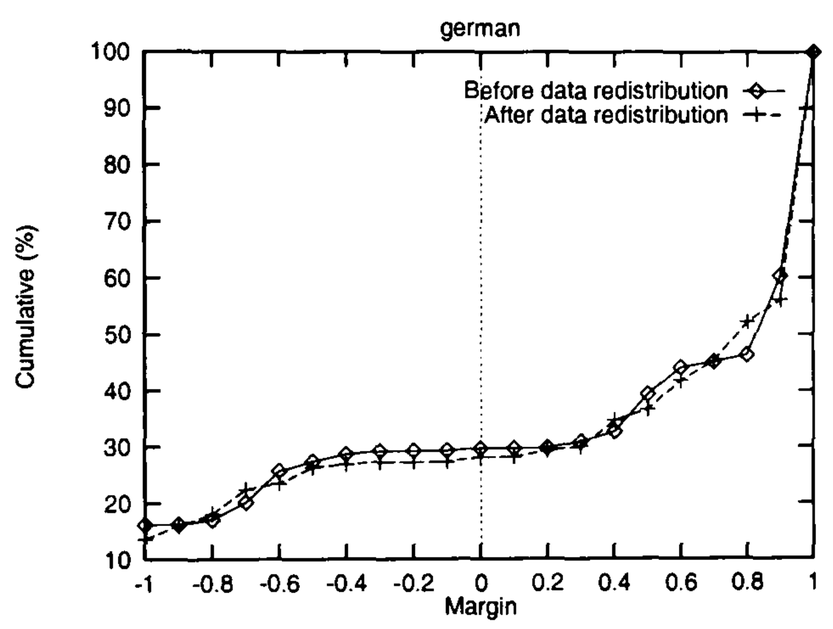
(a) ABALONE-MC



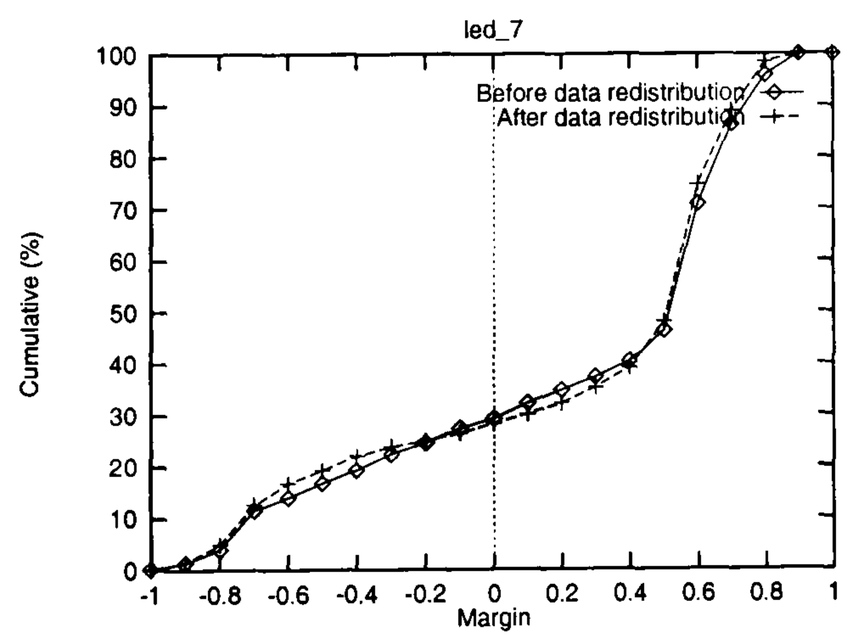
(b) ABALONE-P



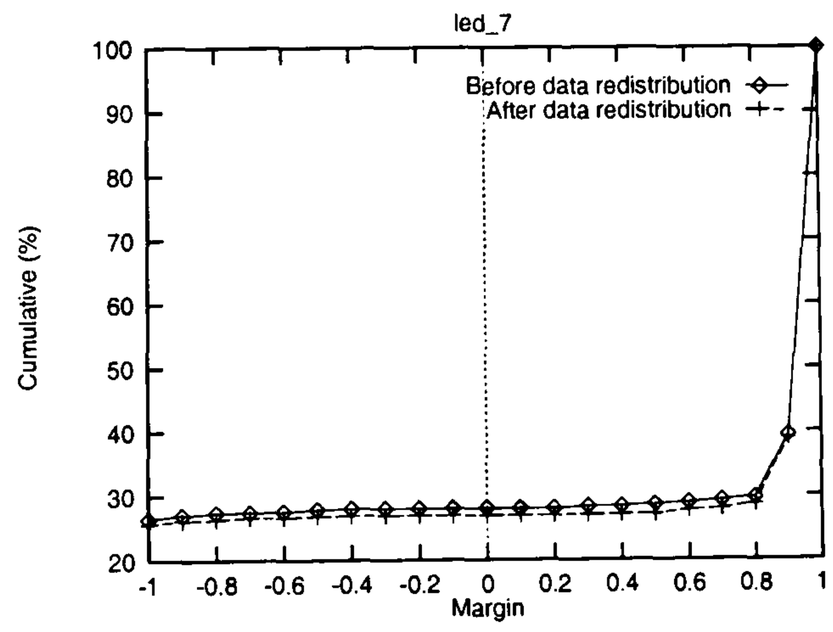
(c) GERMAN-MC



(d) GERMAN-P



(e) LED7-MC



(f) LED7-P

Figure 5.2: Margin distribution graphs of model committees and the corresponding probed models before and after the adjustment in the local data distributions. (MC and P represent *model committee* and *probed model*, respectively.)

Main coverage	Integration scheme	Improvement by DLKP (%)
More than 90% in 0.8 - 1.0 interval	Arbitrarily selection	0.03 - 1.58
Less than 90% in 0.8 - 1.0 interval	Knowledge Probing	0.46 - 1.60
Having margin less than 0.8 but in positive margin area	Data redistribution & Knowledge Probing	2.44 - 2.68
Having margin less than 0.8 and in negative margin area	Data contraction & Knowledge Probing	4.50

Table 5.6: The characteristics of the four groups of model committees and the improvement in classification errors by DLKP and its associated tools and techniques over the average errors of the corresponding local models.

bias and variance terms [82]. This is because a classification error can be decomposed into three quantities: the error caused by an inductive bias in the learning algorithm, the error caused by a variance of the learning algorithm due to a change in the training data, and the error caused by noise in the training data. Research on model integration [32, 12, 58, 60, 6] reveals that the error reduced by model integration techniques is mainly attributed to a decrease in variance. In the case of DLKP, while the integration of local models is expected to decrease variance, the Knowledge Probing phase is expected to induce additional representation bias to the final model. Therefore, this bias and variance decomposition study is aimed at providing an understanding of the behaviour of DLKP and investigating the amount of bias and variance that each of the two learning phases contributes to the classification error of DLKP.

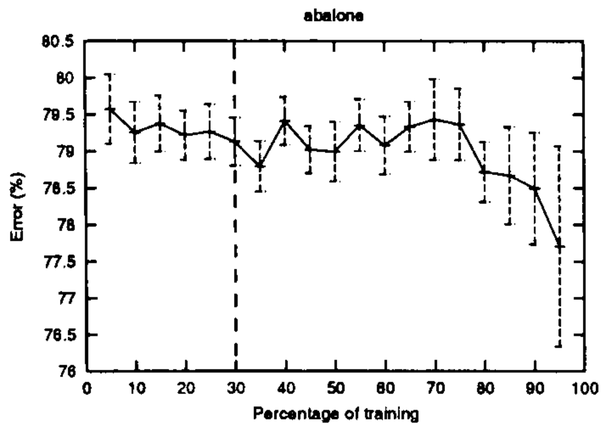
5.4.1 Design of the experiments

In this set of experiments, we use the bias and the variance of the centralised learning approach as bases of the comparison to those of DLKP. Thus, we design two corresponding sets of experiments to be executed in parallel. The details of

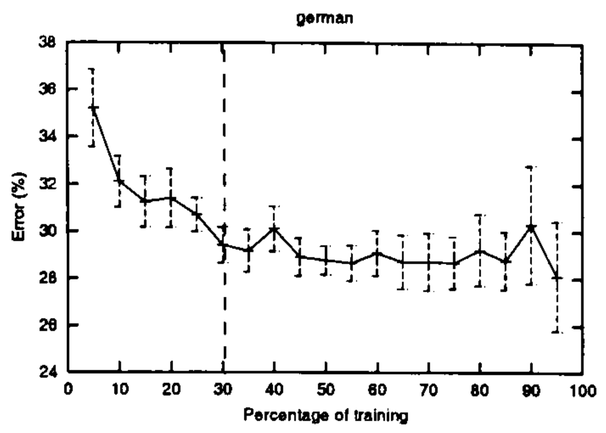
the design are as follows.

Selecting sizes of training set and test set

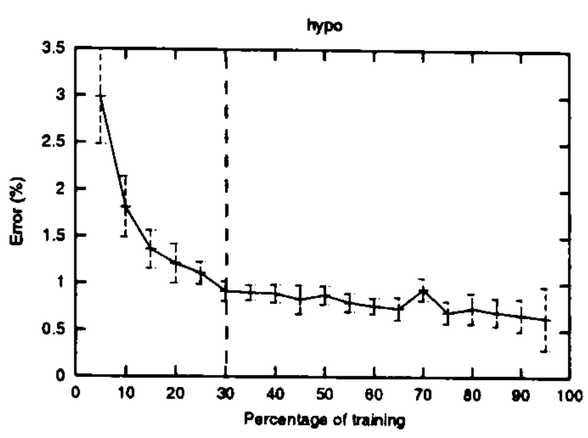
The experiments are performed on the twelve data sets that were used in the previous set of experiments. Again, we simulate distributed environments by mutually and exclusively partitioning a data set into four subsets. Similarly to the previous set of the experiments, some part of the data set is used as a probing set. A concern about data preparation in this experiment is the size of the test set. To make a reliable comparison of biases and variances between the two learning approaches, the approximations of biases and variances should have a small deviation. This requires the size of the test set to be relatively large. However, a reasonable amount of data should be reserved for training. To properly divide the training and test sets, we use the *Learning Curve* feature of $\mathcal{MLC}$++ [57] to generate a learning curve of different training and test set sizes for each data set. The learning curve of each data set is shown in Figure 5.3. Each point in the learning curve represents the error of a model (the value on the y-axis) when a specific percentage of data (the value on the x-axis) is used to train the model and the rest of the data is used to test it. Thus, the learning curve shows the change in classification error when different percentages of data are used to train the model. From each learning curve in Figure 5.3, firstly, we choose a size where the error is relatively stable. This is the value on the x-axis where the graph does not dramatically slope down as x increases. In other words, it is the size where increasing amount of training data does not significantly decrease classification error. Secondly, it should be the point where there is a reasonable amount of data left for a test set. Thirdly, in order to avoid the variability of learning from too small training set, it should be the point where the standard deviation of the estimate (the error bar in Figure 5.3) is small. Figure 5.3 shows that the point where 30% of data is used as a training set appears to satisfy the above three criteria. Thus, in this study, 30% of data is used for training.



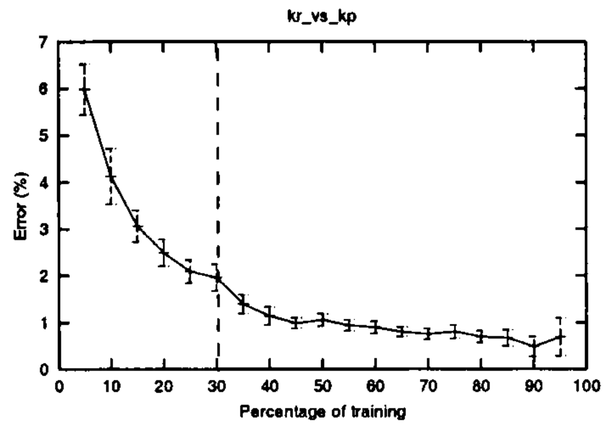
(a) ABALONE



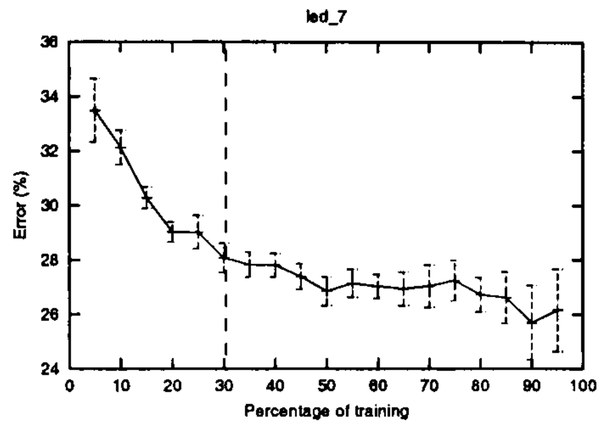
(b) GERMAN



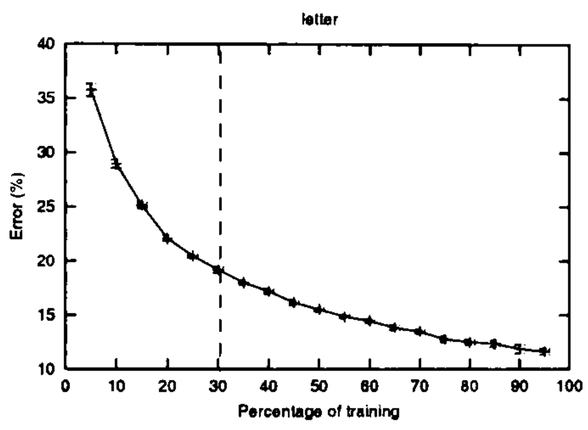
(c) HYPOTHYROID



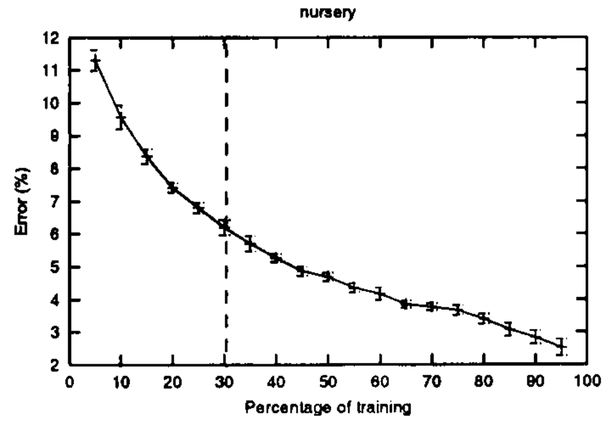
(d) CHESS



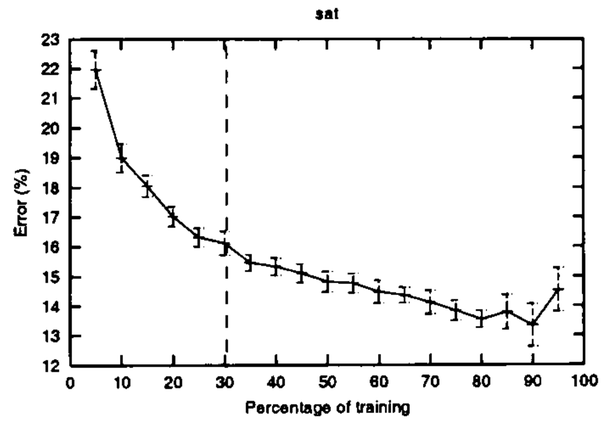
(e) LED7



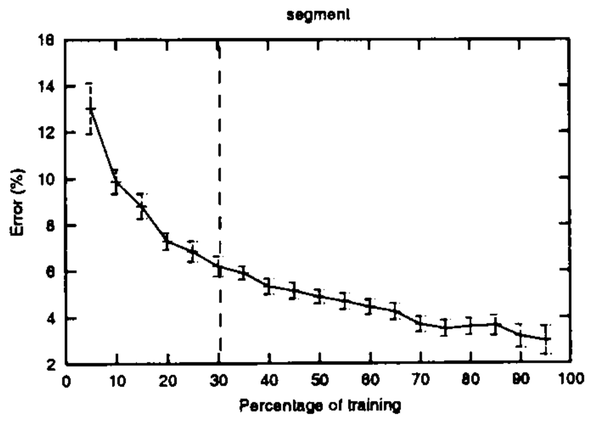
(f) LETTER



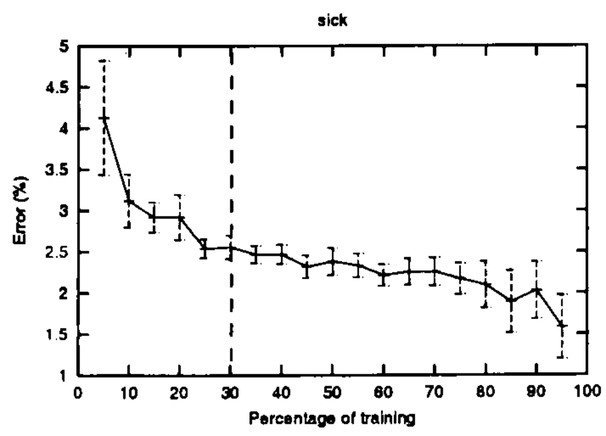
(g) NURSERY



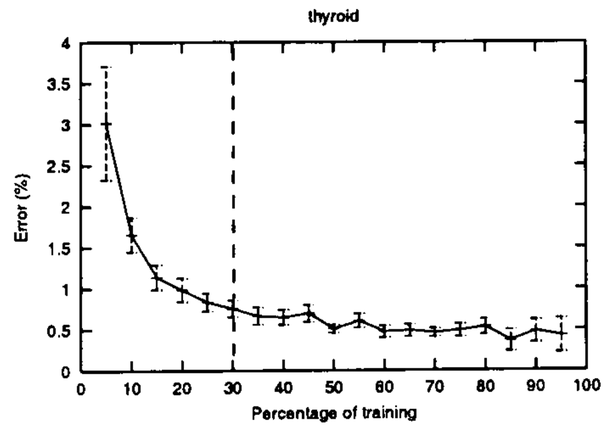
(h) SATELLITE



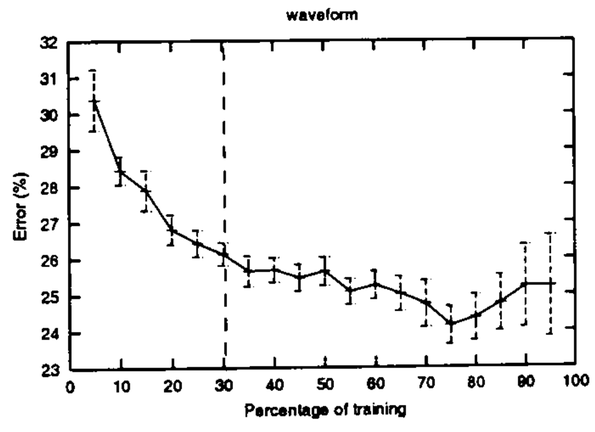
(i) SEGMENT



(j) SICK



(k) THYROID



(l) WAVEFORM

Figure 5.3: Learning curves of the twelve data sets. Each point represents the mean error rate of the twenty runs for the given training set size as tested on the hold-out sample. Each error bar shows the standard deviation of the estimated error. The vertical dashed bar shows the chosen training set size (30%).

Since we use Kohavi & Wolpert's bias-variance decomposition technique [58][6], which approximates bias and variance from different samples of data, we follow Kohavi and Wolpert's approach that the data to be used to further sample the training sets should be twice the size of the desired training sets. Given m as a desired size of the training set, this setting allows for $\binom{2m}{m}$ possible training sets which is enough to guarantee that there are not too many duplicate training sets, even for small values of m. Therefore, in this experiment, 40% of data are used as a test set and the training set, which is 30% of data, is randomly selected from the remaining 60% of the data.

Overall design

Each experiment is separated into three phases: Data Preparation, Local Model Learning phase and Knowledge Probing phase. Data preparation is performed once for each data set. The same training and test sets are used in both DLKP and the centralised learning approaches. The experiments of both approaches are computed in parallel. The statistical calculation is performed at the end of the Local Model Learning phase and the Knowledge Probing phase. This is to investigate the effect of distributed learning and the Knowledge Probing concepts in comparison with the corresponding phases in the centralised learning approach.

In each of the experiments, given a data set, a learning algorithm L (the same algorithm is used for learning local models and probing a final model), a prediction scheme PD and some parameters, i.e. number of partitions, the procedure for estimating $bias^2$, $variance$, and classification error can be separated into three phases as follows.

Data preparation phase The data preparation procedure is described in Figure 5.4. TD is assigned to be 60% of all the available data. Ten sample sets D_i are uniformly sampled from TD without replacement. A probing set PD_i is randomly sampled from each sample set D_i. The size of the probing set is assigned to be 15% plus the average size of the local training data in the Local Model

[6]The details were given in Section 2.3.

Input:

A data set D.

Test set selection:

Randomly select a test set ED from D, so $TD = D - ED$.

Training set selection:

For $i = 1$ to 10

randomly select D_i from TD.

Probing set selection:

For $i = 1$ to 10

randomly select PD_i from D_i.

Output:

Test set ED,
training set $D_1, \ldots, D_{10}$
probing set $PD_1, \ldots, PD_{10}$.

Figure 5.4: Data preparation procedure of the bias-variance decomposition experiments.

Learning phase. ED is used as the test set to approximate the terms in the decomposition. The size of ED is set to be 40% of D.

First phase: Local Model Learning The local model learning procedure is described in Figure 5.5. From Figure 5.5, the evaluation of μ_i using ED is the process of obtaining class predictions of μ_i on ED. That is, each constituent model of μ_i first makes a prediction of each instance in ED. The final prediction from μ_i is derived by using a prediction scheme PS to combine the predictions of the local models. BV is the bias-variance decomposition procedure that was described in Section 2.3.

Second phase: Knowledge Probing Note that only the DLKP experiment is continued in the Knowledge Probing phase. The Knowledge Probing procedure for this experiment is described in Figure 5.6.

Input:

Ten training set, $D_1, \ldots, D_{10}$,
a prediction scheme PS,
a learning algorithm L that provides a descriptive output,
a test set ED,
a bias-variance decomposition procedure BV.

DLKP (distributed environment simulation):

Data partitioning:

For $i = 1$ to 10

randomly partition D_i into $S_{i1}, \ldots, S_{i4}$.

Local Model Learning:

For $i = 1$ to 10

For $j = 1$ to 4

$Md_{ij} = L(S_{ij})$.

Model evaluation:

Let a model committee μ_i consist of $Md_{i1}, \ldots, Md_{i4}$ and PS.
For $i = 1$ to 10

Ed_i is a result of using ED to evaluate μ_i.

Bias-variance decomposition:

$(Bd, Vd) = BV(Ed_1, \ldots, Ed_{10}, ED)$.

Centralised learning approach (non-distributed environment simulation):

Model learning:

For $i = 1$ to 10

$Mnd_i = L(D_i)$.

Model evaluation:

For $i = 1$ to 10

End_i is a result of using ED to evaluate Mnd_i.

Bias-variance decomposition:

$(Bnd, Vnd) = BV(End_1, \ldots, End_{10}, ED)$.

Output:

bias-variance of the Local Model Learning phase Bd, Vd,
bias-variance of the centralised learning approach Bnd, Vnd.

Figure 5.5: The local model learning procedure and the centralised learning procedure for the bias-variance decomposition experiments.

Input:

Model committee $\mu_i, i = 1, \ldots, 10$,
probing set $PD_i, i = 1, \ldots, 10$,
a learning algorithm L which provides a descriptive output,
a test set ED,
a bias-variance decomposition procedure BV.

Prediction phase:

Creating the class values of instances in PD_i using μ_i.
For $i = 1$ to 10
$Pred_i$ is a prediction of μ_i on PD_i.

Probing phase:

Let PD_{probe_i} be a new data set constructed from instances in PD_i whose class values are assigned to $Pred_i$, $i = 1, \ldots, 10$.
For $i = 1$ to 10
$M_i^* = L(PD_{probe_i})$.

Model evaluation:

For $i = 1$ to 10
Ep_i is a result of using ED to evaluate M_i^*.

Bias-variance decomposition:
$(Bp, Vp) = BV(Ep_1, \ldots, Ep_{10}, ED)$.

Output:

Bias-variance after the Knowledge Probing phase Bp, Vp.

Figure 5.6: The Knowledge Probing procedure for the bias-variance decomposition experiments.

In this study, we use C4.5 as a learning algorithm (L) and a probabilistic prediction[7] as a prediction scheme (PD). The number of partitions is set to four.

5.4.2 The experimental results

The biases and variances of the two learning phases of DLKP are shown in Table 5.7. The biases and variances of DLKP and the centralised learning approaches are shown in Table 5.8. For a sanity check on the correctness of the

[7]The details of the probabilistic prediction scheme were given in Section 4.3.1.

Data Set	Local Model Learning Phase			Knowledge Probing Phase		
	$Bias^2$	Variance	Error	$Bias^2$	Variance	Error
ABALONE	46.6862	30.1341	76.8204	46.2611	30.3497	76.6108
GERMAN	22.0426	8.2581	30.3008	26.3860	4.9925	31.3784
HYPOTHYROID	0.7595	0.4548	1.2332	1.1563	0.6611	1.8419
CHESS	3.2731	1.2731	4.5462	3.8216	1.5462	5.3678
LED7	23.9089	6.9607	29.3116	24.4828	9.5000	31.8648
LETTER	11.3492	11.6622	22.9129	15.8725	16.6328	32.3765
NURSERY	6.6049	2.0291	8.6340	7.4727	1.9195	9.3922
SATELLITE	8.9907	6.0307	15.0214	10.3789	8.3113	18.6902
SEGMENT	5.5077	3.2752	8.6999	7.1020	4.8169	11.8310
SICK	2.3098	0.4861	2.7826	2.4408	0.5218	2.9565
THYROID	0.8468	0.3791	1.2202	1.1432	0.5490	1.6844
WAVEFORM	11.2846	10.5713	21.8559	13.5168	14.1571	27.6738
Average	11.9637	6.7929	18.6116	13.3362	7.8298	20.9724

Table 5.7: Comparison of $bias^2$ and variance between the Local Model Learning phase and the Knowledge Probing phases. All figures are in percent.

experiments, it can be seen that, in each data set, the summation of the decomposition results of the centralised learning approach in Table 5.8 is similar to the estimated error to the hold-out error estimation (repeated twenty times) in Figure 5.3 where 30% of data is used for training. Note that there is no noise reported in Table 5.7 and 5.8. This is because the technique that we used, generates a bias term that already includes the intrinsic noise in its frequency counting process.

It should be noted that although Kohavi and Wolpert observe that the estimation of $bias^2$ in their decomposition is itself biased upwards[8] and propose a correction approach, the results reported in this study are not corrected by that approach. This is because Kohavi and Wolpert note that such correction possibly leads to a negative estimate of $bias^2$. In addition, this study is not aimed at calculating the exact quantities of bias and variance, but rather to carry out comparative studies on the change in bias and variance as a result of the two learning phases of DLKP, and to compare bias and variance between DLKP and the centralised learning approach. The comparison can be achieved with-

[8]The estimation of $bias^2$ shrinks as number of sample sets increases. The infinite number of sample sets appears to give its correct value.

Data Set	Centralised learning approach			DLKP		
	$Bias^2$	Variance	Error	$Bias^2$	Variance	Error
ABALONE	46.5856	32.1449	78.7305	46.2611	30.3497	76.6108
GERMAN	19.2607	11.0902	30.3509	26.3860	4.9925	31.3784
HYPOTHYROID	0.3913	0.2992	0.6957	1.1563	0.6611	1.8419
CHESS	1.1338	1.0023	2.1362	3.8216	1.5462	5.3678
LED7	23.3633	7.5063	29.3742	24.4828	9.5000	31.8648
LETTER	8.8504	11.1822	19.9462	15.8725	16.6328	32.3765
NURSERY	3.6878	2.4071	6.0949	7.4727	1.9195	9.3922
SATELLITE	8.0299	8.0253	16.0552	10.3789	8.3113	18.6902
SEGMENT	3.2456	3.3991	6.5655	7.1020	4.8169	11.8310
SICK	1.9293	0.5727	2.4901	2.4408	0.5218	2.9565
THYROID	0.2612	0.3185	0.5769	1.1432	0.5490	1.6844
WAVEFORM	12.6533	13.0895	25.7429	13.5168	14.1571	27.6738
Average	10.7827	7.5864	18.2299	13.3362	7.8298	20.9724

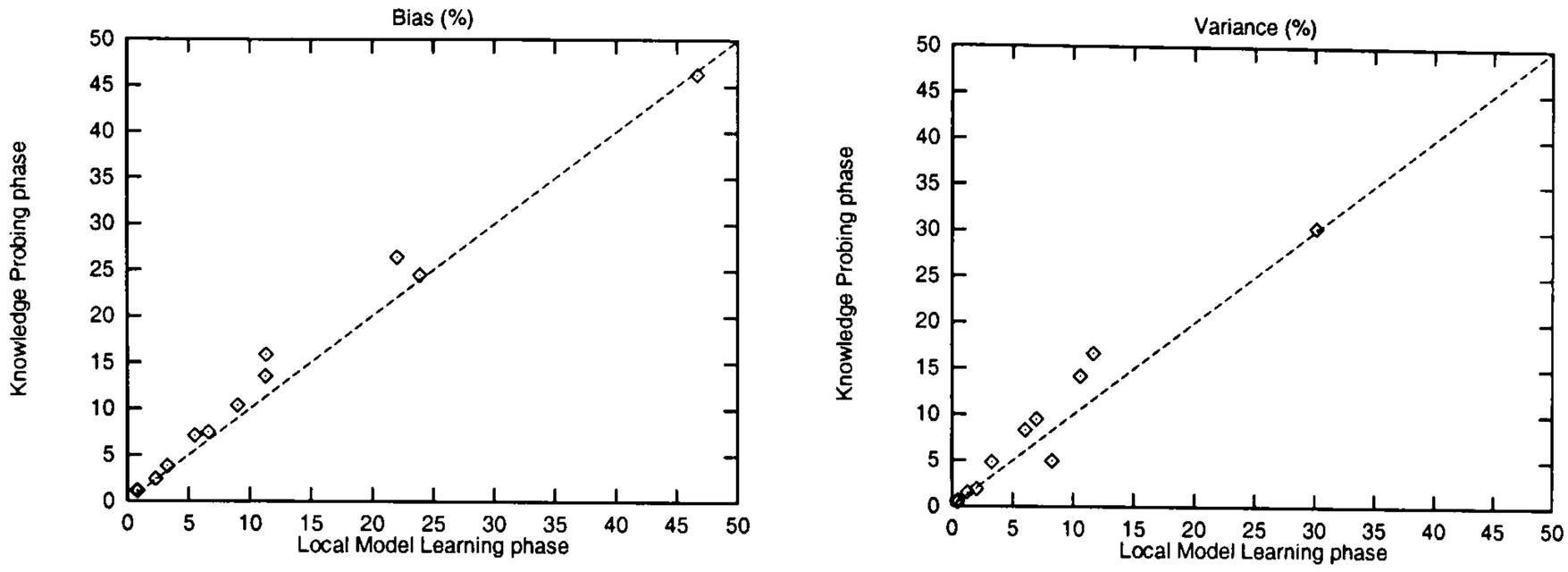
Table 5.8: Comparison of $bias^2$ and variance between the centralised learning approach and DLKP. All figures are in percent.

out the correction because the comparison is performed between models that are learnt and evaluated on exactly the same training and test sets.

5.4.3 Analysis of biases and variances between the two learning phases of DLKP

From the results in Table 5.7, the Knowledge Probing phase appears to induce additional bias in almost all the data sets, except the ABALONE data set. At the same time, it also shows an increase in variance in almost all data sets. This observation can be seen from Figure 5.7 where the biases and variances resulting from the Knowledge Probing phase are plotted against the corresponding biases and variances of the Local Model Learning phase. A dot that lies *above* the cross line implies that there is one data set where the Knowledge Probing phase has a *higher* quantity of interest (bias or variance) than the Local Model Learning phase. On the other hand, a dot lying *beneath* the cross line implies that there is one data set where the Knowledge Probing phase has a *lower* quantity of interest (bias or variance) than the Local Model Learning phase.

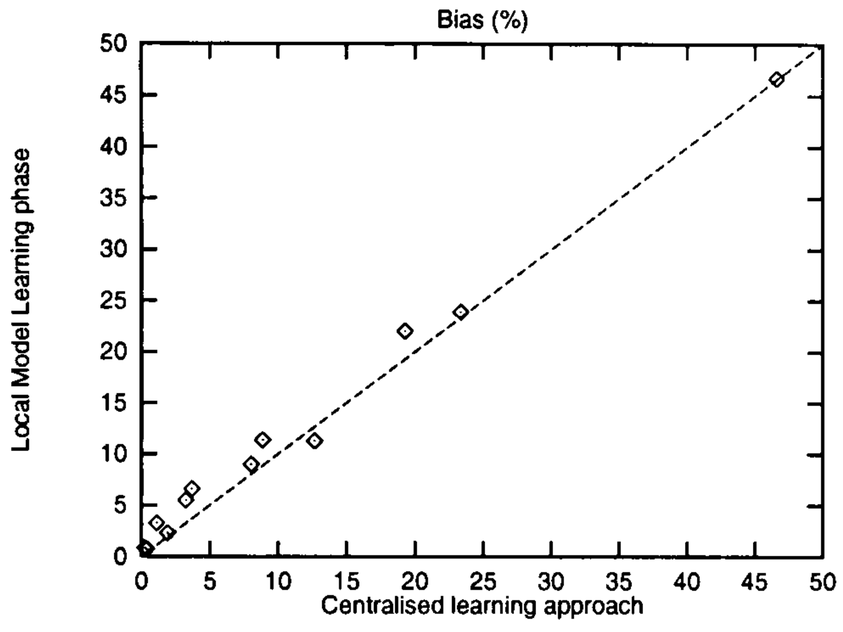
In Figure 5.8, the biases and variances resulting from both phases are plot-



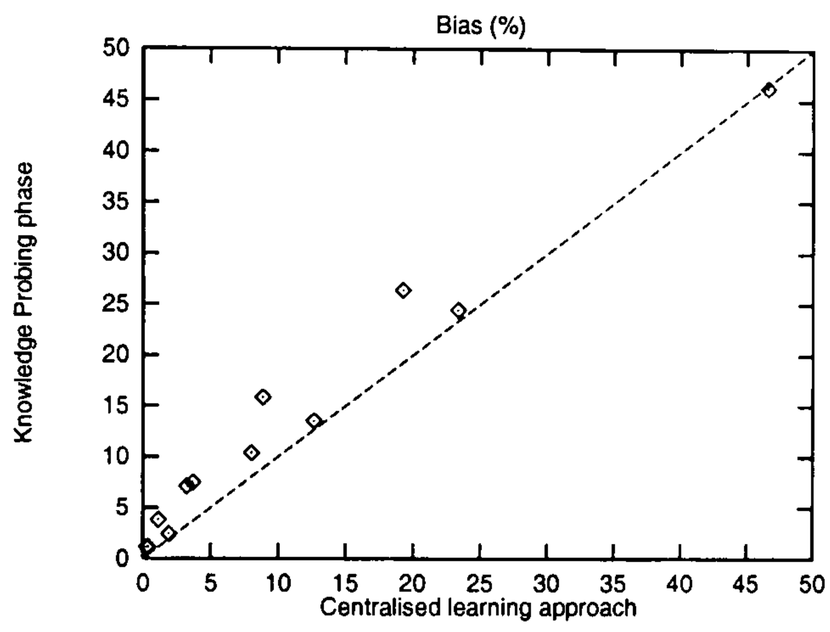
(a) Bias: LML - KP phase (b) Variance: LML - KP phase

Figure 5.7: Biases and variances of the Knowledge Probing phase (KP) plotted against the corresponding biases and variances of the Local Model Learning phase (LML).

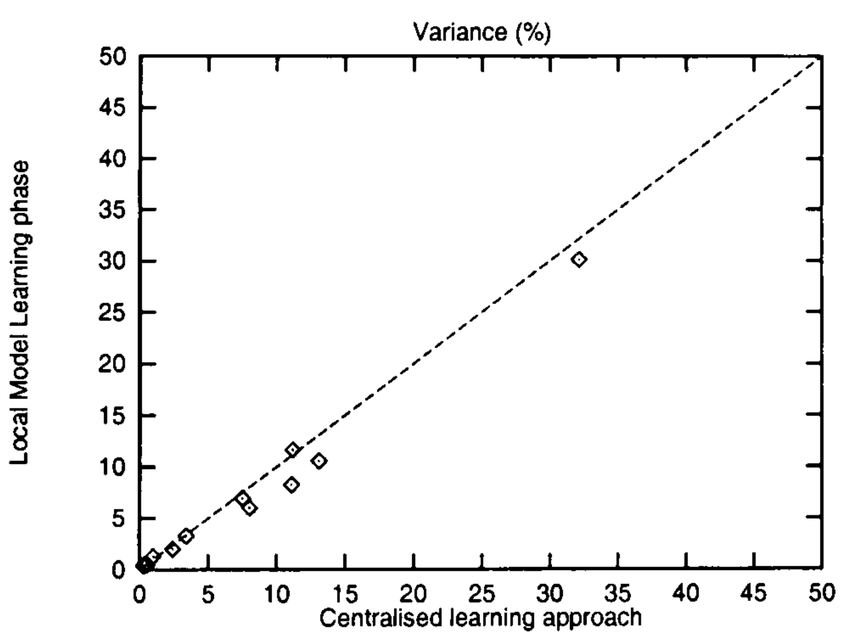
ted against the corresponding biases and variances of the centralised learning approach. An observation from Figure 5.8 reveals interesting responses of bias and variance between the two phases of DLKP. Again, a dot that lies *above* the cross line implies that there is one data set where DLKP has a *higher* quantity of interest (bias or variance) than the centralised learning approach. On the other hand, a dot lying *beneath* the cross line implies that there is one data set where DLKP has a *lower* quantity of interest (bias or variance) than the centralised learning approach. Graphs (a) and (b) in Figure 5.8 show that compared to the centralised learning approach, DLKP continues to increase the bias after the Local Model Learning phase until the end of the Knowledge Probing phase. This can be seen from the points in Figure 5.8 (a) that move further above the cross line in Figure 5.8 (b). In contrast, compared to the centralised approach, Figure 5.8 (c) and (d) show that the variance decreases after the Local Model Learning phase, but increases after the Knowledge Probing phase. This can be explained by Krogh & Sollich's finding [60] that the bias of a model committee (a result of the Local Model Learning phase) is the *average* bias of the constituent models, but the variance of a model committee is *upper bounded* by the average variance of the constituent models. Therefore, compared to the centralised learning approach, a committee of models is not normally expected to yield a



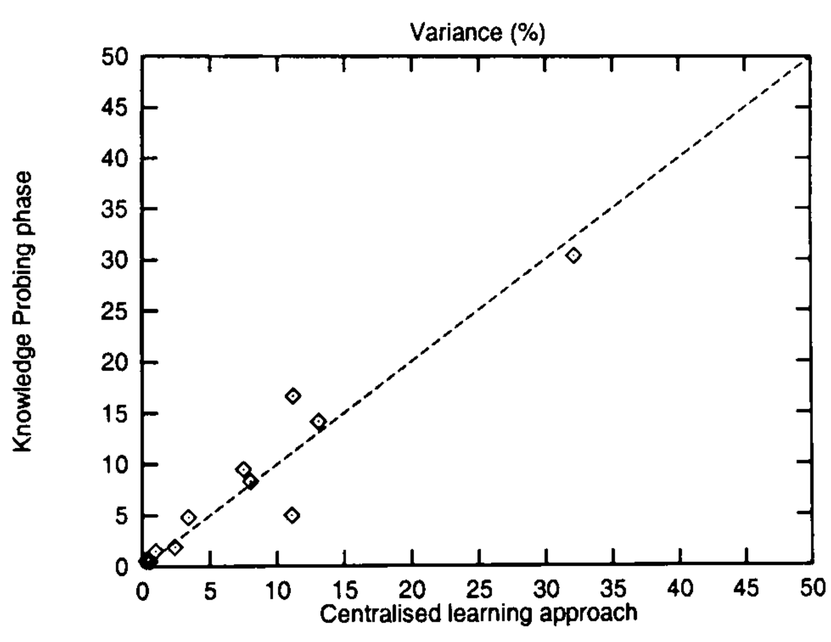
(a) Bias: CL - 1st DLKP



(b) Bias: CL - 2nd DLKP



(c) Variance: CL - 1st DLKP



(d) Variance: CL - 2nd DLKP

Figure 5.8: Biases and variances resulted from both phases of DLKP plotted against the corresponding biases and variances of the centralised learning approach (CL).

Scenario	Integration scheme	Local Model Learning Phase	
		Avr. $bias^2$	Avr. variance
1st	Arbitrarily selecting	1.7973	0.6483
2nd	Knowledge Probing	8.7474	6.7137
3rd	Data redistribution & Knowledge Probing	22.9757	7.6094
4th	Data contraction & Knowledge Probing	46.6862	30.1341

Table 5.9: Summary of the average bias and variance of data sets in different scenarios.

significant reduction in bias. In contrast, for the variance where its bound is saturated when the local models are similar and is varied when the local models are different, the model committee tends to have a wider gap for improvement. This leads to the decrease in the variance after the Local Model Learning phase.

5.4.4 Analysis of biases and variances in different scenarios

Table 5.9 shows the average bias and variance of the Local Model Learning phase of the data sets in each scenario of DLKP. The results in Table 5.9 are consistent with Krogh and Sollich's finding [60]. From Table 5.9, it can be seen that the average variance of the data sets that are classified in the first scenario (where the final model should be arbitrarily selected from the local models) is very low. According to our results, it is less than 1%. It implies that the Local Model Learning phase can make a very low contribution in reducing classification error of the model committees in this scenario. This effect will pass on to the probed model. Therefore, to arbitrarily select the final model from the local models appears to be a reasonable approach with respect to the costs and the quality of the final model.

In the second scenario where the Knowledge Probing technique can be directly applied to the model committee, the average bias and variance appear to be moderate. Between the second and the third scenario (where the data redistribution technique should be applied before the Knowledge Probing phase), it can be seen that the average variances of the two scenarios are not much different.

Their difference in average biases, however, are significant. The average bias of the third scenario is much higher. In this scenario, increasing the coverage of the model committee by the data redistribution technique can be one approach to reduce its bias.

In the fourth scenario (where the data contraction technique is applied before the Knowledge Probing phase), both the average bias and variance are extremely high when compared to those of the other three scenarios. This implies that it can be the case that the learning algorithm has tried to generalise the hard-to-realisable area. This can be observed from Figure 5.1 (a), which is the margin distribution graph of the model committee of the ABALONE data set. The graph shows that most of its slope is in the negative margin area. This shows not only that the model committee is not confident in its predictions, but also that it is inaccurate. This also reflects in the high error rate of the model committee. In this scenario, its coverage can be improved by the data contraction technique.

5.4.5 Summary of the bias-variance decomposition study

The study reveals that, compared to the centralised learning approach, the increase in bias by DLKP is much more significant than the variance. This implies that an error in the probed model can be mainly attributed to an increase in bias after both learning phases, although some error is compensated by a decrease in variance after the Local Model Learning phase.

The results we have seen so far appear to be as expected. In the Local Model Learning phase, additional bias is introduced as a result of learning from (local) models that already incorporate some inductive biases. This is especially true when the models are learnt from subsets of data. The decrease in variance after the Local Model Learning phase as an effect of learning from more than one models is also expected. In the Knowledge Probing phase, an additional bias is inherently introduced in an effort to extract the representation form from the model committee. The variance of the Knowledge Probing phase is increased as a result of learning from the probing set that is only a subset of the available

Data Set	Classification error (%)		
	Average local models	Model committee	Final model
ABALONE	79.2580 ± 0.8964	76.7616 ± 1.4644	74.7600 ± 1.5378 (C)
GERMAN	30.3750 ± 1.5195	26.9000 ± 3.7550	27.7000 ± 3.1287 (R)
HYPOTHYROID	1.3240 ± 0.3390	1.1030 ± 0.4268	0.3000 ± 0.4830 (A)
CHESS	2.0830 ± 0.6135	1.3727 ± 0.7395	0.5000 ± 0.8498 (A)
LED7	29.3390 ± 2.7170	27.4000 ± 2.8460	26.9000 ± 2.8363 (R)
LETTER	21.3170 ± 0.9652	13.2650 ± 0.9095	20.6200 ± 0.8991 (P)
NURSERY	7.1370 ± 0.4117	5.0540 ± 0.6407	6.6800 ± 0.6015 (P)
SATELLITE	16.5320 ± 0.5699	12.7229 ± 0.8566	15.1300 ± 1.2428 (P)
SEGMENT	6.9560 ± 1.0572	4.7619 ± 0.7358	7.5500 ± 1.4539 (P)
SICK	2.7690 ± 0.9637	2.3633 ± 0.9976	1.6000 ± 1.0750 (A)
THYROID	1.0030 ± 0.4155	0.6877 ± 0.4357	0.1000 ± 0.3162 (A)
WAVEFORM	26.6750 ± 0.8804	21.7200 ± 1.9848	25.0800 ± 1.9025 (P)

Table 5.10: Summary of classification errors in three different stages of DLKP framework. ((A), (P), (R) and (C) represent arbitrarily selected model, probed model, probed model after applying the data redistribution technique and probed model after applying the data contraction technique, respectively.)

data.

Thus, it can be concluded that there is a tradeoff between the classification error and the comprehensibility of the model. In the case of DLKP, although some error is reduced in the Local Model Learning phase, some classification error is concomitantly increased by the introduction of the representation bias in the Knowledge Probing phase. Therefore, since there are some costs associated with DLKP, it is important to investigate the criteria of the efficiency of DLKP and provide techniques for optimising DLKP in the scenario where its effectiveness is limited.

5.5 Discussion on the results of the empirical studies

To provide the overall picture, the classification errors of the three stages of DLKP are summarised in Table 5.10. Based on Table 5.10, the effectiveness of the four alternative approaches for different scenarios of DLKP is discussed in

the next three subsections. In the last subsection, the effectiveness of DLKP is compared against the centralised learning approach.

5.5.1 Model integration by selecting a final model

In the case where local models are consistently accurate, the study indicates that the final result of DLKP can be selected arbitrarily from the local models. This is because these local models are more or less the same. Therefore, each of them can be a representative of the others. According to the analysis of the bias-variance decomposition study in Section 5.4, to probe knowledge from such a model committee will unnecessarily introduce an additional representation bias to the final model. As a result, this will lead to a limited improvement or a decrease in classification accuracy.

Other supportive evidence of the limited improvement by probing a global model in this scenario is the fact that limited or small number of instances in the probing set are relabelled. The disagreement level[9] and the percentage of relabelled instances during the Knowledge Probing phase are shown in Table 5.11 and Table 5.12, respectively. It can be seen that the HYPOTHYROID, CHESS, SICK and THYROID data sets have low disagreement levels on the probing set (0.33% – 1.48%), which also reflects in the low number of relabelled instances (0.63% – 2.13%). The low disagreement level and the low number of relabelled instances show that the local models are both consistent and accurate.

Therefore, to select a final model arbitrarily or by using some criteria appears to be a reasonable approach for this scenario.

5.5.2 Model integration by Knowledge Probing

In the case where the margin distribution graph of a model committee does not have an ideal shape, but is in a reasonably good shape, the study reveals that the straightforward application of the Knowledge Probing phase can yield

[9]The disagreement level is calculated from the percentage that the prediction from each local model conflicts with the corresponding combined prediction. This is averaged over the number of local models.

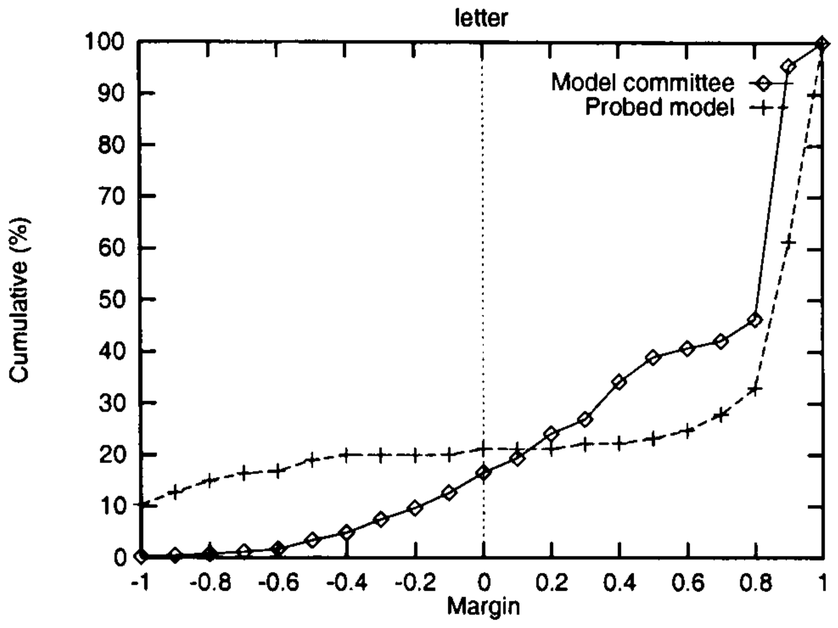
Data Set	Scenario	Disagreement level (%)	
		On probing set	On test set
ABALONE	C	51.2940	51.9590
GERMAN	R	16.4270	17.5750
HYPOTHYROID	A	0.3280	0.5510
CHESS	A	1.4830	1.4970
LED7	R	10.8240	10.7620
LETTER	P	15.5970	17.4260
NURSERY	P	4.2070	4.5010
SATELLITE	P	10.5590	11.0940
SEGMENT	P	4.5510	4.8810
SICK	A	0.7950	0.7880
THYROID	A	0.5280	0.5690
WAVEFORM	P	17.4660	18.9450

Table 5.11: The percentage of disagreement in the predictions between the local models and the corresponding model committee on the probing set and the test sets. Each figure is the average of ten sample sets. (A), (P), (R) and (C) represent arbitrarily selected model, probed model, probed model after applying the data redistribution technique and probed model after applying the data contraction technique, respectively.

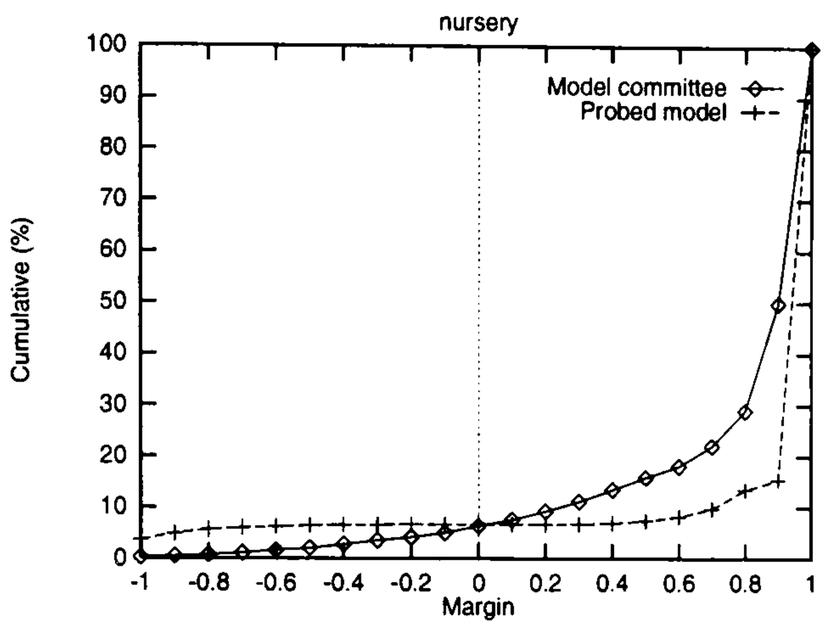
satisfying results. From Table 5.10, the probed model of the LETTER, NURSERY, SATELLITE, SEGMENT and WAVEFORM data sets have comparable classification errors to those of the corresponding local models. In four out of the five data sets, the probed models have $0.46\% - 1.60\%$ lower error than the average error of the local models. There is only one data set (SEGMENT) where the probed model has 0.59% higher error. One may observe from Table 5.10 that the arbitrarily selected model appears to have the lowest error of the three stages[10] and one may question why the arbitrarily approach is not applicable to this scenario. It should be noted, however, that in this case, there is no grounds for assuming that the local models are more or less the same. Therefore, it is not reasonable to use one of them as a representative of the others.

Figure 5.9 is the comparison of the margin distribution graphs between the model committee and the corresponding probed models of the five model committees in this scenario. It can be seen that at the zero margin line (the central

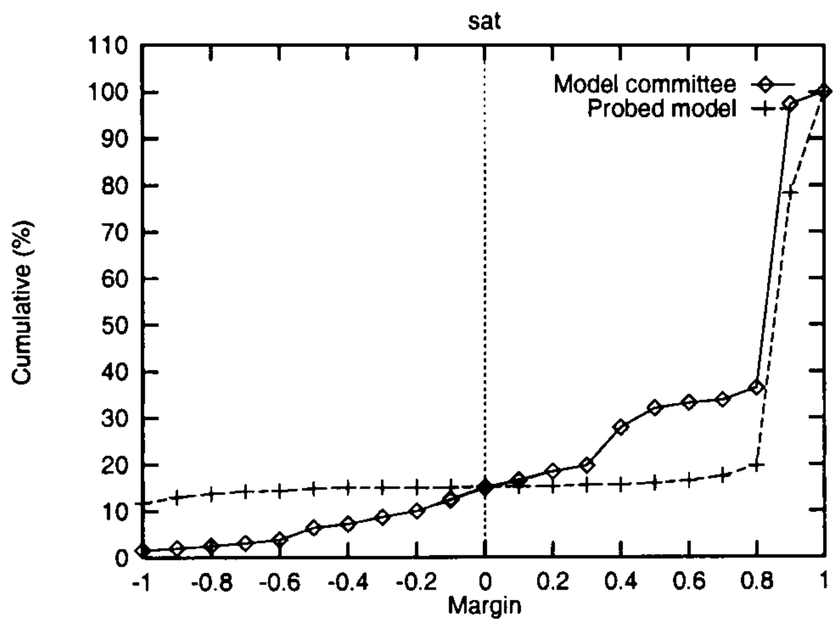
[10]The three columns in Table 5.10.



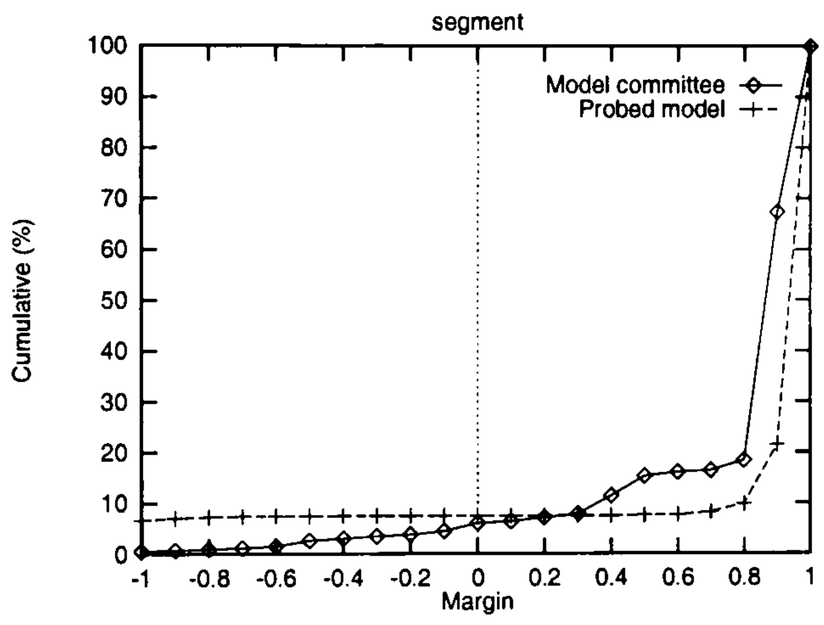
(a) LETTER



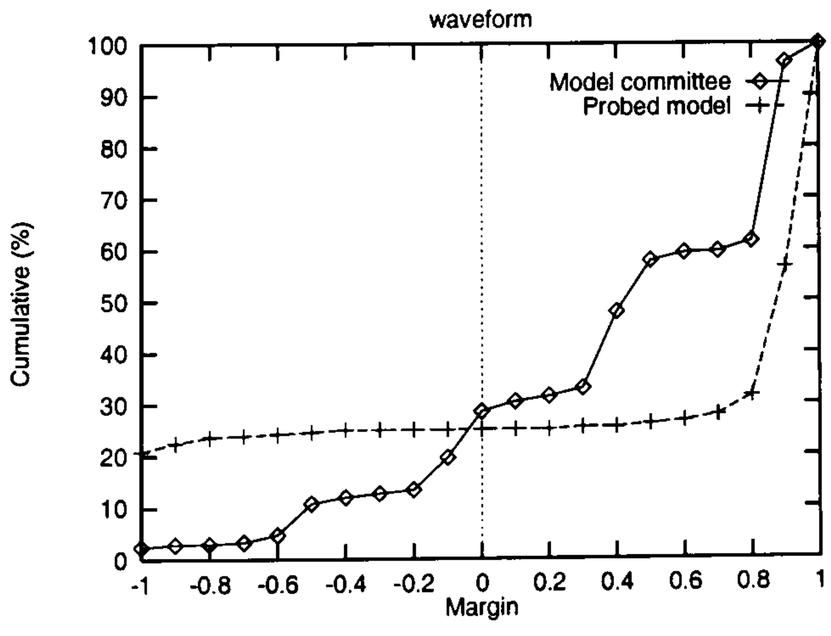
(b) NURSERY



(c) SATELLITE



(d) SEGMENT



(e) WAVEFORM

Figure 5.9: Margin distribution graphs of each model committee in the second scenario and its corresponding probed model.

Data Set	Scenario	Percentage of relabelled instances
ABALONE	C	64.59
GERMAN	R	19.19
HYPOTHYROID	A	0.83
CHESS	A	1.24
LED7	R	25.85
LETTER	P	8.86
NURSERY	P	3.71
SATELLITE	P	9.37
SEGMENT	P	3.18
SICK	A	2.13
THYROID	A	0.63
WAVEFORM	P	14.07

Table 5.12: Percentage of relabelled instances during the Knowledge Probing phase. Each figure is the average of ten sample sets. (A), (P), (R) and (C) represent arbitrarily selected model, probed model, probed model after applying the data redistribution process and probed model after applying the data contraction process, respectively.

vertical line), both graphs are close. This means both models have comparable accuracy. However, the shape of the margin distribution graph of the probed model appears to be closer to the ideal shape. This means that, although the probed model does not show much improvement with respect to the accuracy, its coverage on the instance space appears to be better.

5.5.3 Model integration by Knowledge Probing after adjusting local data distributions

In the scenario where the model committees have poor margin distribution graphs, the study (detailed in Section 5.3) indicates that an adjustment in the local data distribution can further improve the classification accuracy. This can be achieved by using the data redistribution technique to enable the local models to increasingly cover a common area in the instance space, i.e. the target area, or using the data contraction technique to enable them to gradually change their coverage. Table 5.13 shows the average error of local models in comparison with the error of probed model before and after the adjustment in local data distributions.

Data Set	Classification error (%)		
	Average local models	Probed model	
		Before adjustment	After adjustment
ABALONE	79.2580 ± 0.8964	75.7800 ± 2.5183	74.7600 ± 1.5378 (C)
GERMAN	30.3750 ± 1.5195	29.5000 ± 4.9272	27.7000 ± 3.1287 (R)
LED7	29.3390 ± 2.7170	28.0000 ± 2.5055	26.9000 ± 2.8363 (R)

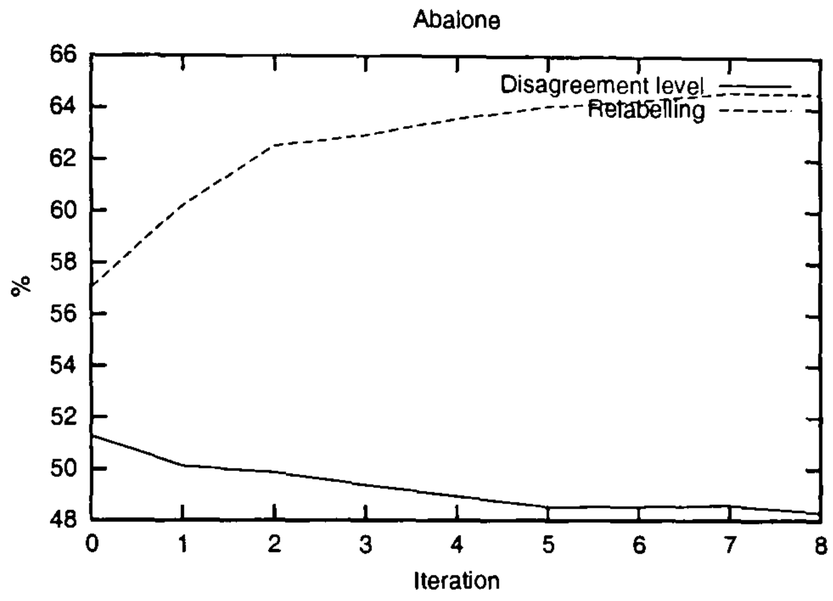
Table 5.13: Comparison of the average error of the local models, the error of the probed model before and after the adjustment in local data distributions in the scenario where margin distribution graphs of model committees are relatively poor. (C) and (R) represents the data contraction and the data redistribution techniques, respectively.

Table 5.13 shows that both techniques show an improvement in classification accuracy over the original distributions of local data.

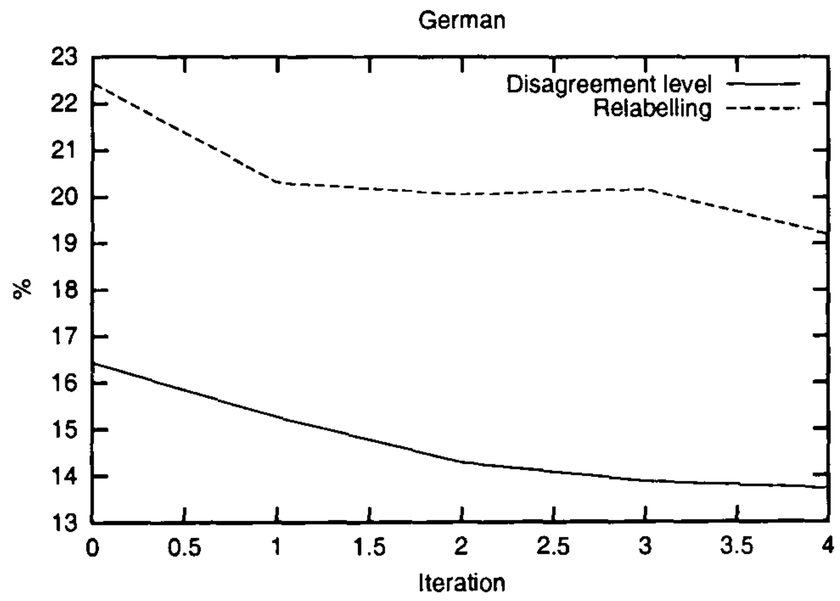
Further investigation reveals that although both the data redistribution and the data contraction techniques appear to increase the consistency among the local model as the iteration proceeds, they show an opposite effect on the percentage of relabelling. This behaviour can be observed from Figure 5.10, which shows the plots of the inconsistency level and the number of relabelled instances of the ABALONE, GERMAN and LED7 data sets as the data distribution adjusting process proceeds. The ascending graph shows that the disagreement level and the percentage of relabelling increase as the iteration proceeds. The descending graph shows the opposite trend.

In the data redistribution technique (Figure 5.10 (b) and Figure 5.10 (c)), the percentage of the relabelling decreases as the local models disagree less and become more consistent. This trend is a positive response to the aim of the design of the data redistribution technique, which is to enable the local models to share a common distribution to a certain level. For such a purpose, the local models should gradually become more consistent and lead to a decreasing percentage of relabelling.

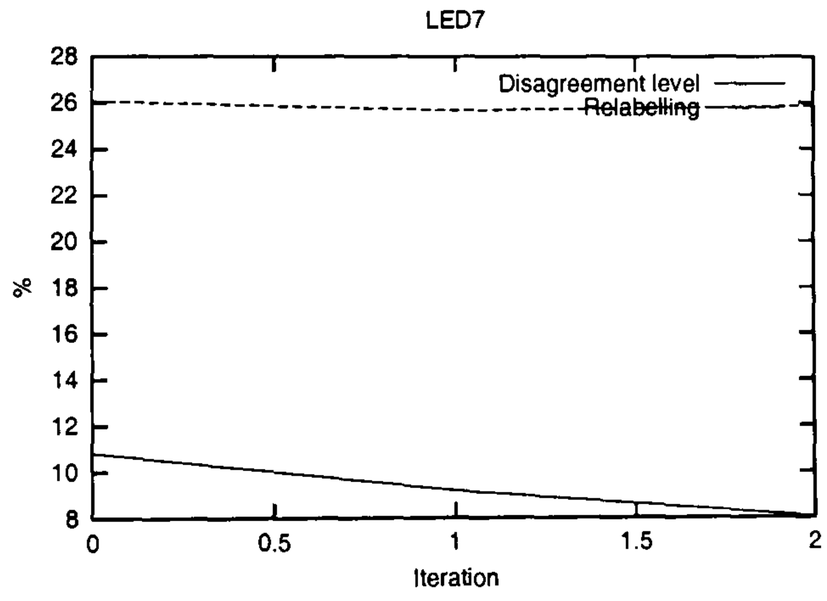
In contrast, the data contraction technique is designed to contract the coverage of the local model especially in the area where they have low confidence in



(a) ABALONE (C)



(b) GERMAN (R)



(c) LED7 (R)

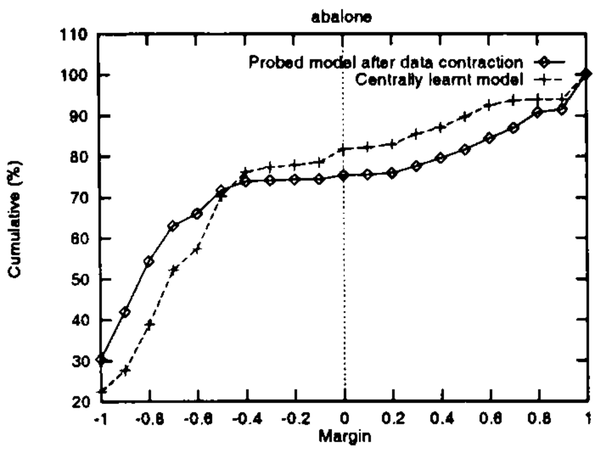
Figure 5.10: The changes in the inconsistency level and the relabelling percentages of ABALONE, GERMAN and LED7 data sets during the data distribution adjusting process. (C) and (R) represents the data contraction and the data redistribution techniques, respectively.

making predictions. These areas are likely to be the areas where the misleading instances are located i.e. the border area of the coverage of the local model. After the coverage of each local model is contracted from those areas, the local models are expected to become more consistent. This is because there is less fluctuation in their predictions as the problematic areas have been removed from the local training sets. The percentage of relabelling, however, increases due to the fact that instances in the probing set are randomly selected from the local data. It is inevitable that this may include some problematic instances. These instances are expected to be relabelled as the local models become more consistent. The opposite effect on the disagreement level and the percentage of relabelling can be observed from Figure 5.10 (a).

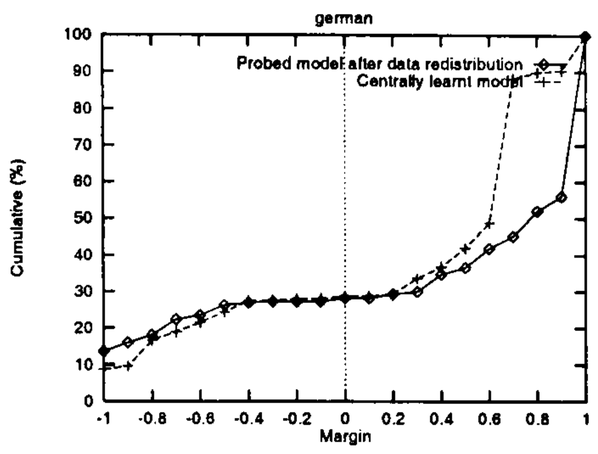
5.5.4 Comparison between DLKP and the centralised learning approach

The error of the final models of DLKP and the error of the corresponding centrally learnt models are shown in Table 5.14. Table 5.14 shows that the accuracy of the results of DLKP is comparable to those of the centralised learning approach. In seven out of the twelve data sets, the final models of DLKP have lower error rate than the centrally learnt models. Four (LETTER, SATELLITE, SEGMENT and WAVEFORM) out of the five data sets where DLKP are inferior to the centrally learnt models are the data sets whose all the attributes are continuous. Thus, it can be the case that this inferiority is a result of the approach which C4.5 uses to deal with continuous attributes. C4.5 requires to have a global view of all existing values of each continuous attribute in the training set [79]. Using C4.5 as the local learning algorithm, DLKP is disadvantaged in this respect because the local data do not provide a global view of the continuous attribute values. Therefore, the probed models are less accurate in predicting instances whose attributes are mainly continuous.

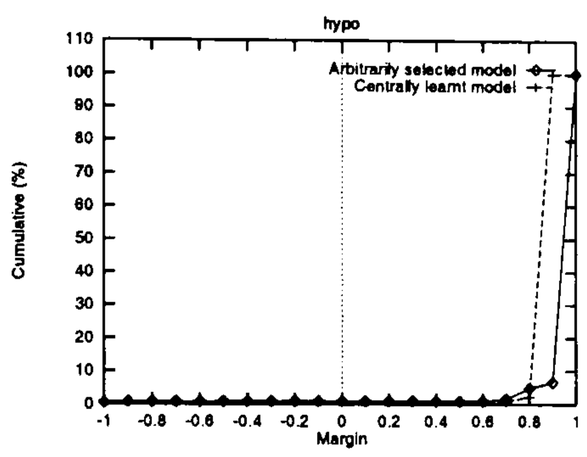
The margin distribution graphs of the two approaches are compared in Figure 5.11. It can be seen that, except the LED7 data set, margin distribution



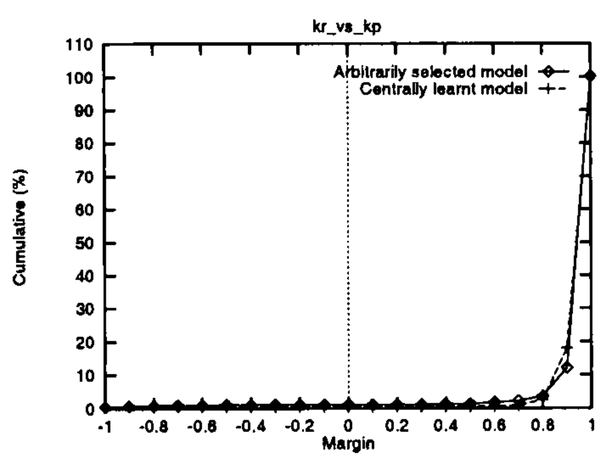
(a) ABALONE



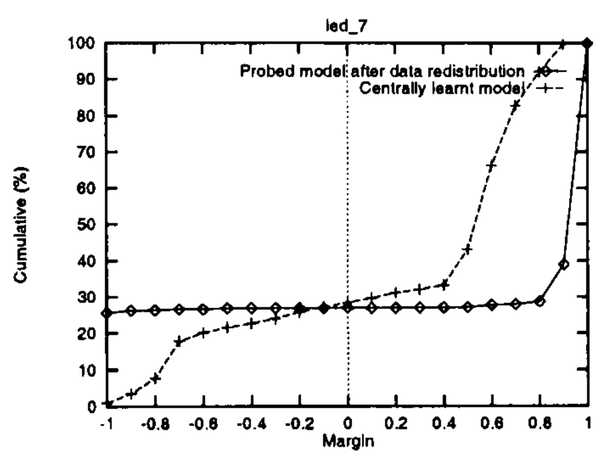
(b) GERMAN



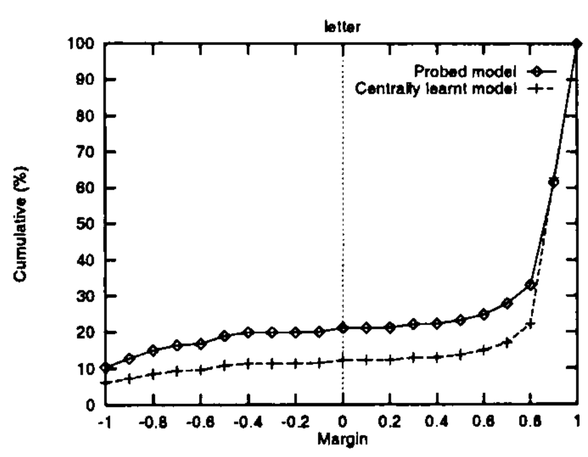
(c) HYPOTHYROID



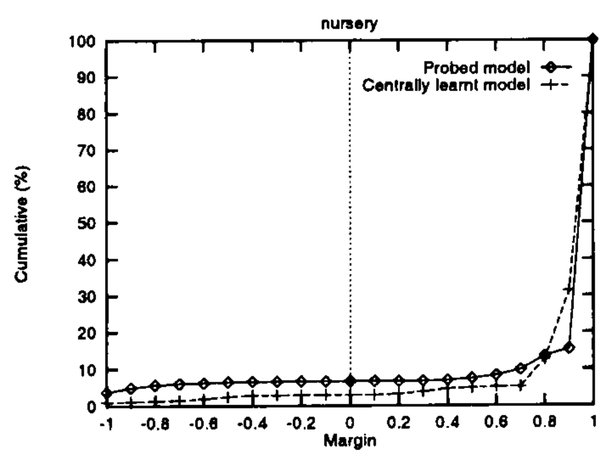
(d) CHESS



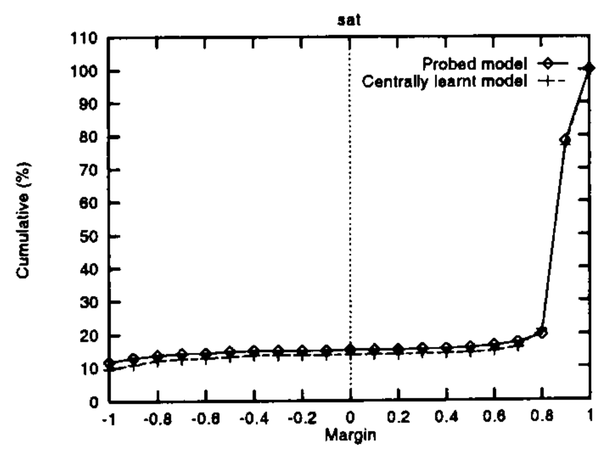
(e) LED7



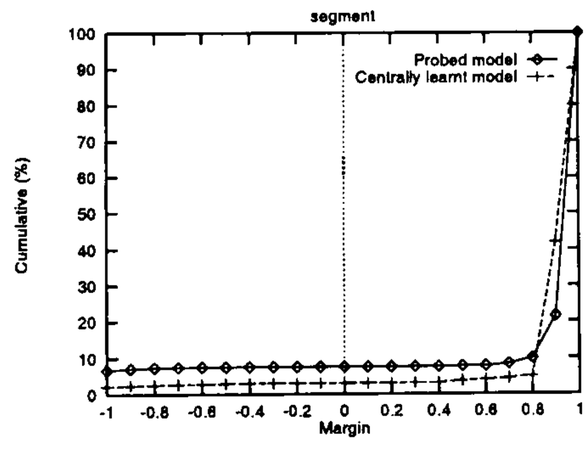
(f) LETTER



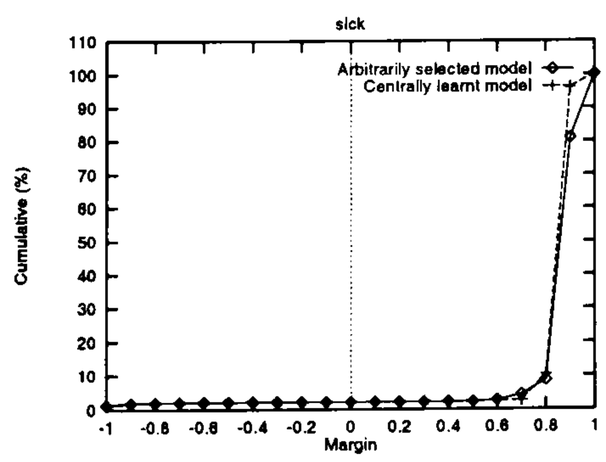
(g) NURSERY



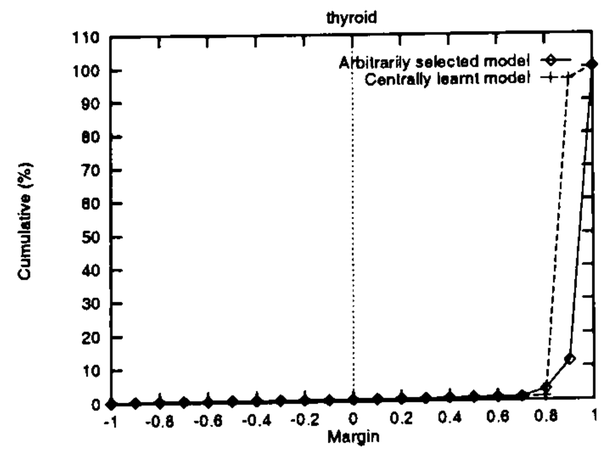
(h) SATELLITE



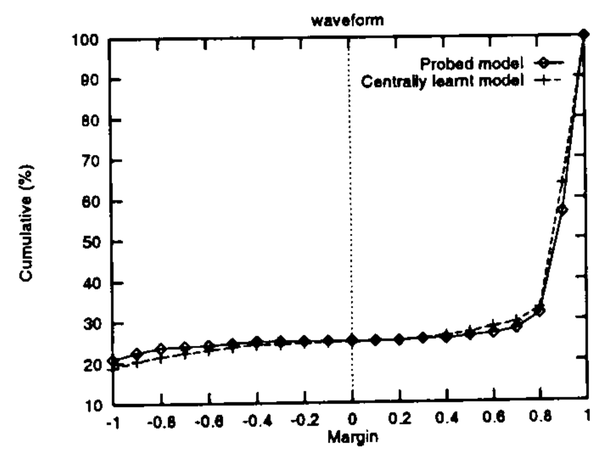
(i) SEGMENT



(j) SICK



(k) THYROID



(l) WAVEFORM

Figure 5.11: Comparison of the margin distribution graphs of the probed models and the corresponding centrally learnt models.

Data Set	Classification error (%)	
	DLKP's final model	CL-model
ABALONE	74.7600 ± 1.5378 (C)	79.5100 ± 2.0136
GERMAN	27.7000 ± 3.1287 (R)	28.4000 ± 2.8752
HYPOTHYROID	0.3000 ± 0.4830 (A)	0.8000 ± 0.4876
CHESS	0.5000 ± 0.8498 (A)	0.5400 ± 0.3950
LED7	26.9000 ± 2.8363 (R)	27.1500 ± 3.1890
LETTER	20.6200 ± 0.8991 (P)	11.8500 ± 1.0352
NURSERY	6.6800 ± 0.6015 (P)	2.9600 ± 0.4006
SATELLITE	15.1300 ± 1.2428 (P)	13.8100 ± 1.0514
SEGMENT	7.5500 ± 1.4539 (P)	2.9800 ± 1.1961
SICK	1.6000 ± 1.0750 (A)	2.1800 ± 0.7598
THYROID	0.1000 ± 0.3162 (A)	0.5900 ± 0.3315
WAVEFORM	25.0800 ± 1.9025 (P)	24.7600 ± 1.7199

Table 5.14: Comparison of classification error between the final models of DLKP framework and the corresponding centrally learnt models. CL-model represents the centrally learnt model. (A), (P), (R) and (C) represent arbitrarily selected model, probed model, probed model after applying the data redistribution technique, and probed model after applying the data contraction technique, respectively.

graphs of both approaches appear to have a similar shape. This means that, with much smaller training set size (roughly 1/4 according to the setting of this study), the probed model has more or less the same coverage as the model that is learnt from the complete data set.

The comparison of the model sizes between the two approaches is shown in Table 5.15. It can be seen that with comparable accuracy and coverage, the results of DLKP appear to be much more compact (121.35% – 544.39% smaller) than the corresponding centrally learnt models.

5.6 Summary

The empirical studies show that the use of the margin distribution graph for observing different forms of coverage of a model committee and the use of the two data distribution adjusting techniques together with DLKP are effective. The analysis of the bias and the variance of the framework has also shown supportive evidence to the design of DLKP. With these tools and techniques, DLKP has

Data Set	Tree size	
	Probed model	CL-model
ABALONE	378.00 ± 31.09	2057.80 ± 35.34
GERMAN	34.40 ± 13.49	128.60 ± 21.87
LED7	56.20 ± 5.51	68.20 ± 8.12
LETTER	967.40 ± 15.11	2331.80 ± 57.27
NURSERY	202.00 ± 19.82	508.40 ± 13.13
SATELLITE	152.80 ± 10.89	543.80 ± 18.55
SEGMENT	34.20 ± 4.92	80.00 ± 5.01
WAVEFORM	182.80 ± 8.97	580.60 ± 14.66

Table 5.15: Comparison of the model sizes between the probed models (probed trees) and the corresponding centrally learnt models. CL-model represents the centrally learnt model.

shown to be robust across a range data sets of varying characteristics. In all scenarios, the results of DLKP are, at least, as good as the average results of the Local Model Learning phase, which satisfies the minimum requirement in the design of DLKP. In addition, the results of DLKP appear to be comparable to the centrally learnt models with respect to accuracy and coverage, and appear to be better with respect to model size. It is important to note that with DLKP, a model that is learnt from roughly 1/4 (according to the setting of this study) of a complete data set has comparable properties to a model that is learnt from a complete data set. This means that with some compensation in computation and communication overhead, learning under distributed environments can be made more or less the same as learning under non-distributed environments. The computation and communication overhead of DLKP will be discussed in Chapter 7.

Concerning the contributions in bias and variance from each of the learning phases of DLKP to the final result of DLKP, our empirical study shows that although some variance is reduced in the Local Model Learning phase, the Knowledge Probing phase introduces some addition representation bias in its representation extraction process. Thus, there is a compensation in the error between both phases of learning. This compensation can be minimised by using a learning algorithm that induces a small amount of representation bias in the

Knowledge Probing phase.

In this chapter, the distributed data were simulated by applying a data partitioning technique to a complete data set of each problem domain. In the next chapter, we present a case study on applying DLKP to a set of real distributed data. The applicability of three different classification algorithms in DLKP is also investigated.

Chapter 6

Case Study

This chapter presents a case study on applying DLKP to a set of real distributed data and on applying three classification algorithms in the Local Model Learning phase of DLKP. The possibility of applying DLKP to the model committee that is a mix of models learnt by different classification algorithms is also investigated.

6.1 Introduction

So far, our empirical study is based on distributed data that are simulated by using a data partitioning technique and utilises one classification learning algorithm. The purpose of this case study is to evaluate the effectiveness of DLKP on real distributed data and also on three classification learning algorithms. The problem domain, the HEART DISEASE data, is first described. After that, we discuss the experimental approach in the case study. Then, the experimental results are reported and discussed.

6.2 The problem domain: Heart disease diagnosis data

In this section, we describe the problem domain in the case study, the HEART DISEASE data. We first describe the characteristics of data, which include the

Institution	Principle investigator
Cleveland Clinic Foundation, USA	Robert Detrano, M.D., Ph.D.
Hungarian Institute of Cardiology, Budapest	Andras Janosi, M.D.
V.A. Medical Center, Long Beach, USA	Robert Detrano, M.D., Ph.D.
University Hospital, Zurich, Switzerland	William Steinbrunn, M.D.
University Hospital, Basel, Switzerland	Matthias Pfisterer, M.D.

Table 6.1: The locations and names of the principle investigators responsible for the collection of the HEART DISEASE data.

number and the type of attributes, and the number of classes and their distributions. After that, we describe the requirements of an integrated model for the HEART DISEASE data.

6.2.1 Data description

The HEART DISEASE data were obtained from UCI Repository of machine learning databases [8]. It consists of four data sets concerning heart disease diagnosis, namely CLEVELAND, HUNGARIAN, LONG BEACH VA, and SWITZERLAND data sets. The names of the principle investigators responsible for the collection of the data from each location are shown in Table 6.1. Each data set in the HEART DISEASE data has the same format. Originally, it contained seventy five attributes, but all previously published experiments [30, 45] refer to the use of only thirteen of them. We use the same thirteen attributes in this case study, five of which are continuous attributes, the other eight being discrete attributes. The class value refers to the presence of heart disease in the patient. This is an integer valued from 0 to 4. The values 1, 2, 3, and 4 represent the presence of the disease, the value 0 represents the absence from the disease. The class distribution of each data set is shown in Table 6.2.

6.2.2 Why integrate the HEART DISEASE models ?

One may question why integrating models is better than integrating all the data together and learning a model from the integrated data. These HEART DISEASE data sets provide a good example to answer this question.

Data set	Class					Total
	0	1	2	3	4	
CLEVELAND	164	55	36	35	13	303
HUNGARIAN	188	37	26	28	15	294
LONG BEACH VA	51	56	41	42	10	200
SWITZERLAND	8	48	32	30	5	123

Table 6.2: Class distributions of the four data sets of the HEART DISEASE data.

Attribute	Missing values in data set (%)			
	CLEVELAND	HUNGARIAN	LONG BEACH VA	SWITZERLAND
1	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00
4	0.00	0.34	28.00	1.63
5	0.00	7.82	3.50	0.00
6	0.00	2.72	3.50	60.98
7	0.00	0.34	0.00	0.81
8	0.00	0.34	26.50	0.81
9	0.00	0.34	26.50	0.81
10	0.00	0.00	28.00	4.88
11	0.00	64.63	51.00	13.82
12	1.32	98.98	99.00	95.93
13	0.66	90.48	83.00	42.28

Table 6.3: The percentages of the missing values in the four HEART DISEASE data sets.

The first reason concerns the difference in the nature of these distributed data sets. These data sets are distributed in different locations and it is very unlikely that all locations have the same environment. For examples, it is not likely that different hospitals have the same diagnostic equipments and the same group of patients. These differences can be seen from Table 6.3, which shows the missing values of each attribute in each data set. Table 6.3 shows that these data sets are not equally complete. The values of some attributes in some data sets are almost totally missing. An example is the twelfth attribute, which stands for the number of major vessels that are coloured by fluoroscopy[1]. The CLEVELAND data

[1] A method of observing the form and motion of the deep structures of the body by means of x-ray shadows projected on a fluorescent screen.

Local data sets	Classification error (%)		
	Decision tree (C4.5)	Naive Bayes Classifier (NBC)	k-nearest neighbour (PEBLS)
CLEVELAND	45.70	52.17	**43.50**
HUNGARIAN	46.50	34.78	**32.60**
LONG BEACH VA	53.90	**39.13**	42.40
SWITZERLAND	60.90	**53.26**	69.60

Table 6.4: The classification errors of the local models that are learnt by a decision tree learning algorithm (C4.5), a naive Bayes classifier (NBC) and a k-nearest neighbour algorithm (PEBLS). The lowest classification error in each row is displayed in bold face.

set contains only 1.32% missing value whereas the values of this attribute in the other data sets are almost completely missing. It can be the case that the other three hospitals do not have the equipment to measure this value. Therefore, to straightforwardly combine the data sets that are different in sizes, are collected from different environments, and are different in underlying knowledge can lead to the learning being biased towards the data sets that are larger in size and are more complete.

The second reason concerns the suitability of each local data to different learning algorithms. This difference in suitability can be seen from Table 6.4, which shows the classification errors that are the results of applying three different learning algorithms (a decision tree learning algorithm, a naive Bayes classifier, and a k-nearest neighbour algorithm) to the four data sets. The lowest classification error among the results of the three learning algorithms in each data set is displayed in bold face. Table 6.4 shows that the classification errors for the three learning algorithms are different and are not related. Since some algorithms perform better than the others on some data sets, to combine all the data sets together and use a single learning algorithm can lead to an increase in classification error because some of the data sets in the integration may not fit with the chosen algorithm.

Therefore, between model integration and data integration, model integration appears to be more suitable for distributed environments especially when the

environments of the local data are different.

6.3 Approach of the study

This section describes in details the approach used in the case study. It includes the data preparation scheme, the three classification learning algorithms, and the design of the experiments.

6.3.1 Data preparation

Table 6.2 shows that distributed data sets in practical applications, i.e. the HEART DISEASE data, are different in sizes and class distributions. They are different from the distributed data sets in Chapter 5, which were simulated by using a data partitioning technique and had approximately equal sizes. It should be noted that the methodology of assigning the size of the probing set, the test set and the consistency test set in the case study is the same as that used in Chapter 5. That is, the size of the probing set and the size of the consistency test set are 115% and 100% of the average sizes of the local data sets, respectively, and the size of the test set is 10% of all the four data sets.

6.3.2 Three classification algorithms used in the case study

Three classification algorithms: a decision tree learning algorithm, a naive Bayes classifier, and a k-nearest neighbour algorithm, are used to investigate the applicability of different classification algorithms in DLKP. They represents three major classification learning approaches, which are search-based approach (decision tree learning algorithm), probability-based approach (naive Bayes classifier), and instance-based approach (k-nearest neighbour algorithm). These classification algorithms differ mainly in the assumptions that they make on the underlying target function, and the ways they form the hypothesis.

Decision tree learning algorithm Decision tree learning provides a practical method for learning discrete-valued functions. The most familiar decision tree learning algorithm is Quinlan's ID3 family [78]. Its successor, C4.5 [79], is the algorithm that is employed in the case study. C4.5 forms a decision tree (hypothesis) based on the assumption that between two trees of the same accuracy, the smaller tree has a better generalisation than the larger one. To find the smallest tree, C4.5 employs an information gain criterion to select the attribute that will be used as the next node of the tree. Information gain measures the expected increase in the purity[2] in subsets of data when a particular attribute is used as the next node. The attribute with the highest information gain is expected to maximise the purity in the branches. To find such an attribute, C4.5 calculates the information required to identify the class of an instance that belongs to the current node and that of an instance that belongs to a potential next node (i.e. a node that is a result of using a particular attribute as a test). The difference of these two quantities represents the information gained from using that attribute to branch the tree. Given D as a set of instances, $freq(c_j, D)$ as the number of instances in D that belong to class c_j, and $|D|$ as the number of instances in set D, the average amount of information needed to identify the class of an instance in D is

$$info(D) = -\sum_{j=1}^{k} \frac{freq(c_j, D)}{|D|} \times log_2\left(\frac{freq(c_j, D)}{|D|}\right), \tag{6.1}$$

where k is the number of possible classes. Applying the measurement in Equation 6.1 to $D_i, \ldots, D_o$, which is a result of partitioning D according to the o outcomes of a test X (an attribute X), the expected information required to identify the class of an instance in D is

$$info_X(D) = \sum_{i=1}^{o} \frac{|D_i|}{|D|} \times info(D_i).$$

[2]The similarity of the class values in a group of instances.

Thus, the information gained by using attribute X to branch the tree can be written as

$$gain(X) = info(D) - info_X(D). \tag{6.2}$$

C4.5 branches the tree depth first by choosing the attribute that has highest information gain as the next node. The process continues recursively until all the attributes are used in the path of the tree, or all instances at the leaf of the tree belong to the same class.

Naive Bayes classifier Bayesian reasoning provides a probabilistic approach for inference. It is based on the assumption that the quantities of interest are governed by probability distributions, and that the optimal decision can be made by reasoning about these probabilities together with the observed data. Given the attribute value $< x_1, x_2, \ldots, x_a >$ that describe an instance of k possible classes, the Bayesian approach to classifying a new instance is to assign the most probable class value (c_{MPC}),

$$c_{MPC} = \underset{j=1}{\overset{k}{argmax}}\ P(c_j|x_1, x_2, \ldots, x_a). \tag{6.3}$$

By using Bayes theorem, Equation 6.3 can be rewritten as

$$\begin{aligned} c_{MPC} &= \underset{j=1}{\overset{k}{argmax}}\ \frac{P(x_1, x_2, \ldots, x_a|c_j)P(c_j)}{P(x_1, x_2, \ldots x_a)} \\ &= \underset{j=1}{\overset{k}{argmax}}\ P(x_1, x_2, \ldots, x_a|c_j)P(c_j). \end{aligned} \tag{6.4}$$

One practical difficulty in applying this Bayesian method is that it typically requires initial knowledge of many probabilities. The number of required probabilities is equal to the number of possible instances times the number of possible class values. Thus, when the number of attributes increases, the estimation is only feasible when there is a very large set of training data available. The naive Bayes classifier (NBC) [46, 63] is a practical Bayesian learning method in such a scenario because it is based on the assumption that the attribute values are con-

ditionally independent given the class value i.e. $P(x_1, x_2, \ldots x_a|c_j) = \prod_{i=1}^{a} P(x_i|c_j)$. By substituting this into Equation 6.4, the most probable class value of the naive Bayes classifier is

$$c_{MPC} = \underset{j=1}{\overset{k}{argmax}}\ P(c_j) \prod_{i=1}^{a} P(x_i|c_j). \tag{6.5}$$

Therefore, for the naive Bayes classifier, the number of the required probabilities is reduced to the sum of the number of possible values of each attribute times the number of possible class values. Note that there is no explicit search through the space of possible hypotheses in the naive Bayes classifier. Instead, the hypothesis is formed simply by counting the frequency of the various combinations of data within the training examples. The version of naive Bayes classifier that is used in the case study has been developed at the data mining group, Imperial College.

k-nearest neighbour algorithm k-nearest neighbour [26, 38] is an instance-based learning algorithm for approximating real-valued target functions or discrete-valued target functions. Given a number of attributes equal to a, the algorithm assumes that each instance corresponds to a point in a a-dimensional Euclidean space. It stores the training instances and calculates the distances between the stored instances and a new instance when the new instance needs to be classified. The class value of the new instance is estimated from the class values of its k nearest training instances that are previously stored. The k-nearest neighbour algorithm used in the case study is PEBLS 3.0 [80].

PEBLS 3.0 employs Stanfill-Waltz VDM (Value Difference Metric) [86] in defining the distance (δ) between two values of a specific attribute. Under VDM, two values are similar if they occur with the same relative frequency for all classifications. Given X_1 and X_2 as instances of a data set that contains a attributes and k possible classifications, if x_{1j} and x_{2j} are the values of attribute j^{th} of X_1 and X_2, respectively, PEBLS 3.0 defines the distance between x_{1j} and

x_{2j} as

$$\delta(x_{1j}, x_{2j}) = \sum_{i=1}^{k} \left| \frac{F_{1i}}{F_1} - \frac{F_{2i}}{F_2} \right|, \tag{6.6}$$

where F_{1i} is the number of times that x_{1j} was classified into class i and F_1 is the total number of times that x_{1j} occurred. By applying Equation 6.6 to all a attributes, the total distance (Δ) between two instances, X_1 and X_2 is

$$\Delta(X_1, X_2) = w_{X_1} w_{X_2} \sum_{j=1}^{a} \delta(x_{1j}, x_{2j})^2, \tag{6.7}$$

where X_1 and X_2 correspond to a training instance and a new instance, and w_{X_1} and w_{X_2} are the weights assigned to X_1 and X_2, respectively. The details of the scheme to assign weights to the training instance and the new instance can be found in [25]. Note that the k-nearest neighbour algorithm is different from the previous two classification algorithms in that it does not form an explicit hypothesis of the entire target function from the available training data. It only gives the class prediction of a new instance based on the available training data, and the process only happens during classification time. Moreover, since PEBLS 3.0 gives a prediction as a single class rather than a set of the probabilities of all possible classes, when its prediction is applied to the probabilistic prediction scheme in this case study, we assign 1 to the probability of the class that is the prediction and assign 0 to the probabilities of all other classes.

6.3.3 Design of the experiments

This case study consists of four sets of the experiments that are different in the local learning algorithms. In all sets, C4.5 is used as the probing algorithms and the probabilistic prediction is used as the prediction scheme. C4.5 is chosen as the probing algorithm because it is the only algorithm of the three that provides a descriptive output. In the first three sets of the experiments, each of the three classification algorithms described in Section 6.3.2 is in turn used as a local learning algorithm. In the fourth set of the experiments, the algorithm that gives

the lowest classification error among the three learning algorithms on each data set is chosen to use as a local learning algorithm of that data set.

6.4 The effectiveness of DLKP on real distributed data

The experimental results of the first three sets of the experiments are shown in Figure 6.1, Table 6.5, and Figure 6.2. In this section, we discuss the performance of DLKP on the HEART DISEASE data sets and also the applicability of the three classification algorithms in DLKP.

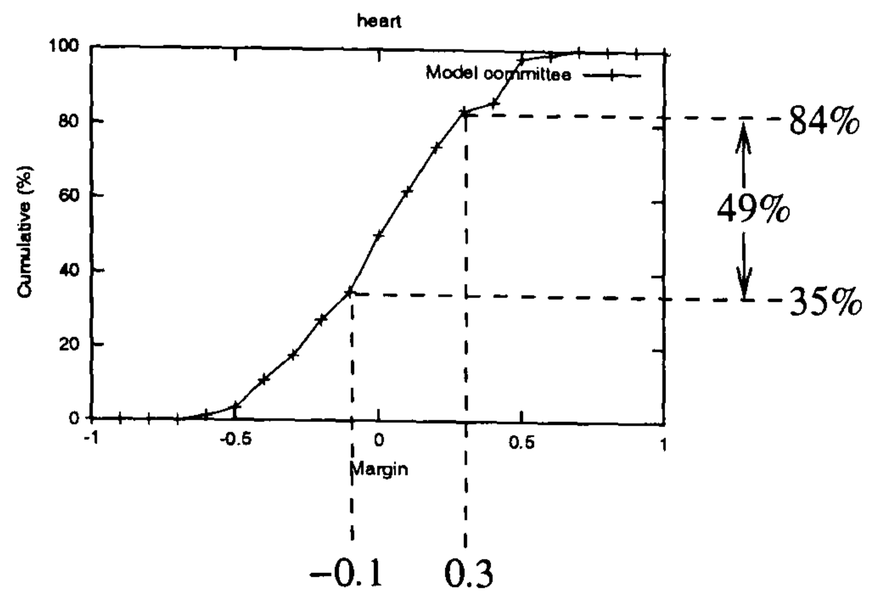
6.4.1 Analysis of margin distribution graphs

Figure 6.1 shows the margin distribution graphs of the model committees from the first three experiments. It can be seen that the graphs are similar in shape. However, the interpretations of their main slopes are very different.

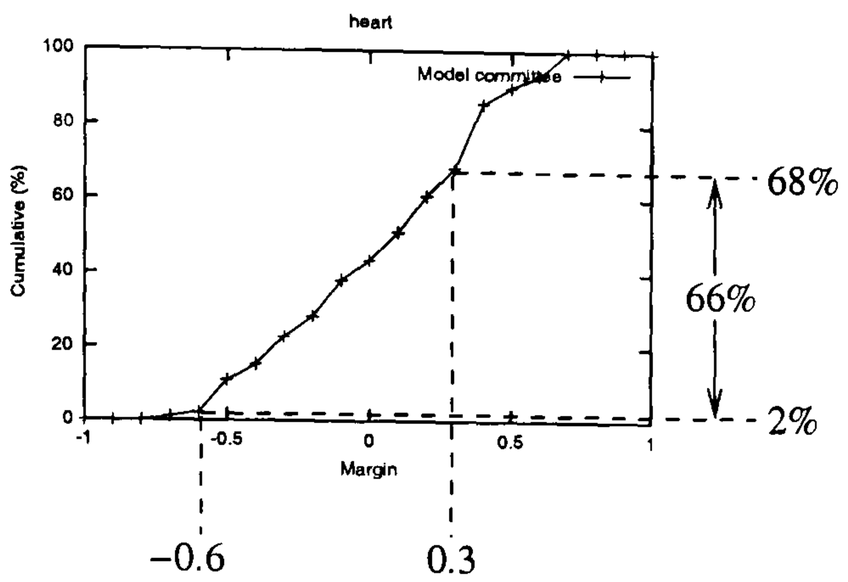
In Figure 6.1 (a), which is the graph of the model committee whose local models are learnt by C4.5, 49% of the coverage of the model committee has the margin ranging between -0.1 and 0.3. Because the margin range is less than 0.8 and its majority falls in the positive margin area, according to our approach, the data redistribution technique should be used to improve the consistency of the models in the committee before the Knowledge Probing phase.

Figure 6.1 (b), which is the graph of the model committee whose local models are learnt by the naive Bayes classifier, shows that 66% of the coverage of the model committee has the margin ranging between -0.6 and 0.3. Because the margin range is less than 0.8 and its majority is in the negative margin area, our study suggests that the data contraction technique should be used to increase the consistency of models in the committee.

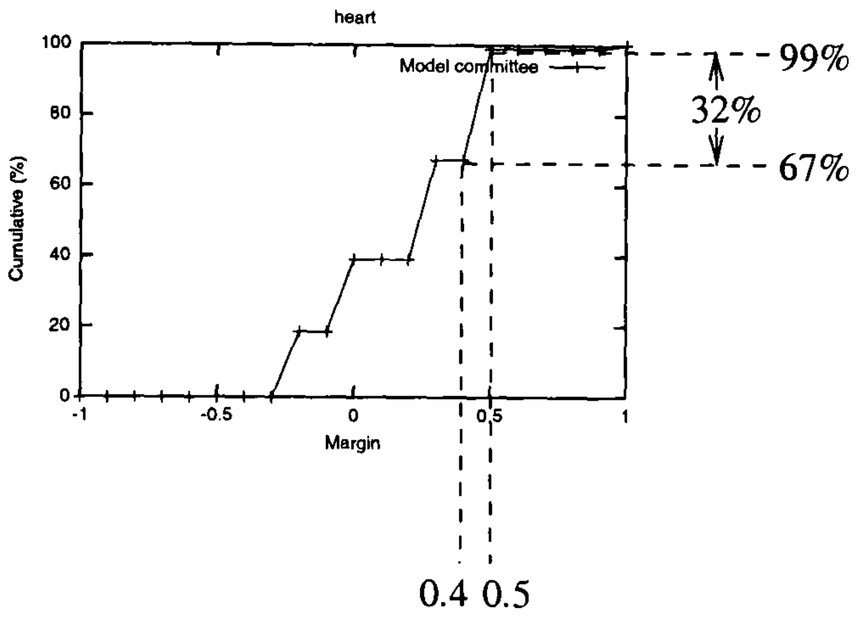
Compared with the first two graphs, Figure 6.1 (c), which is the graph of the committee whose local models are learnt by PEBLS, appears not very smooth. This is because PEBLS gives prediction as one single class rather than a set of



(a) C4.5



(b) NBC



(c) PEBLS

Figure 6.1: Margin distribution graphs of the model committees that are learnt by C4.5, NBC and PEBLS.

probabilities for each class. In Figure 6.1 (c), although the most prominent slope is not clearly observed, the rightmost slope appears to be more significant than the others. The rightmost slope can be interpreted as 32% of the coverage of the model committee has the margin range between 0.4 and 0.5. Since the main coverage has margin less than 0.8, but its majority falls in the positive margin area, according to our approach, the data redistribution technique should be used to increase the consistency of models in the committee before the Knowledge Probing phase.

In the next subsection, we discuss how the data distribution adjusting techniques can improve the quality of the probed models.

Local learning algorithms	Classification error (%)					
	Average local models		Model committee		Probed model	
	Before	After	Before	After	Before	After
C4.5 (R)	51.75	**31.50**	35.87	**23.91**	40.20	**34.80**
NBC (C)	**44.84**	45.38	38.04	**35.87**	38.00	38.00
PEBLS (R)	47.03	**35.60**	**30.43**	36.96	39.10	**37.00**

Table 6.5: Comparison of the classification errors in different stages of DLKP before and after the adjustment in local data distributions. C4.5, NBC and PEBLS are in turn used as the local learning algorithm. C4.5 is used as the probing algorithm. (R) and (C) in the first column represents the data redistribution and the data contraction techniques, respectively. The lower classification error in each pair is shown in bold face.

6.4.2 Improvement by the data distribution adjusting techniques

Based on the analysis of the margin distribution graphs in Section 6.4.1, we adjust the distributions of the local data of the three experiments. The classification errors of the three experiments before and after the adjustment in local data distributions are shown in Table 6.5. Note that the classification error of C4.5 on the joint data set is 40.20%.

Table 6.5 shows that the classification errors of the three probed models before the adjustment in local data distributions are 6.84% – 11.55% lower than the corresponding average errors of the local models. This means that even before the adjustment in local data distributions, the classification error of the probed models already reach the goal that we set out in Chapter 1. Compared to the classification error of C4.5 on the joint data set, which is 40.20%, the classification errors of the probed models before the adjustment in local data distributions are either comparable or lower.

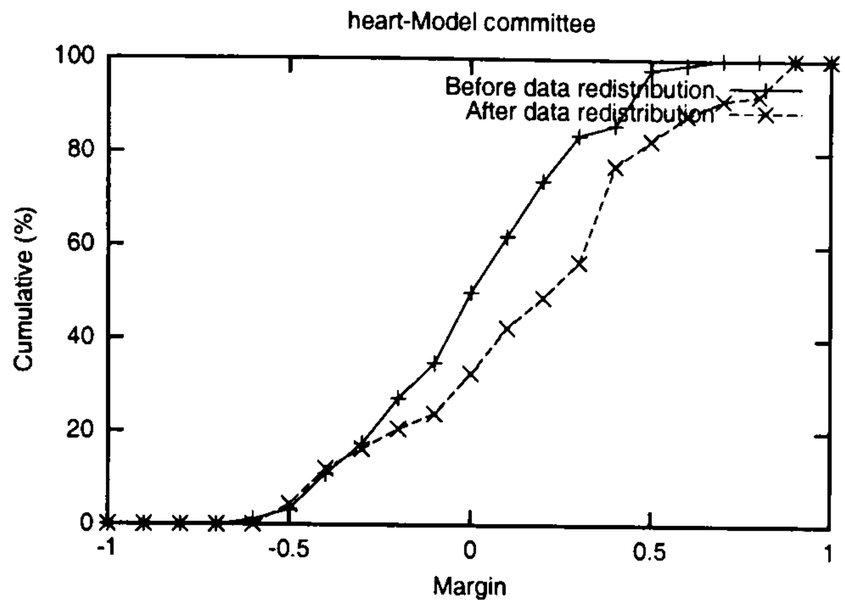
The adjustment in local data distributions leads to a further 5.4% decrease in classification error of the probed model that the local models are learnt by C4.5, and leads to a further 2.1% decrease in classification error of the probed model that the local models are learnt by PEBLS. In the experiment where local models are learnt by NBC, the adjustment in local data distribution leads to neither an

increase nor a decrease in the classification error of the probed model.

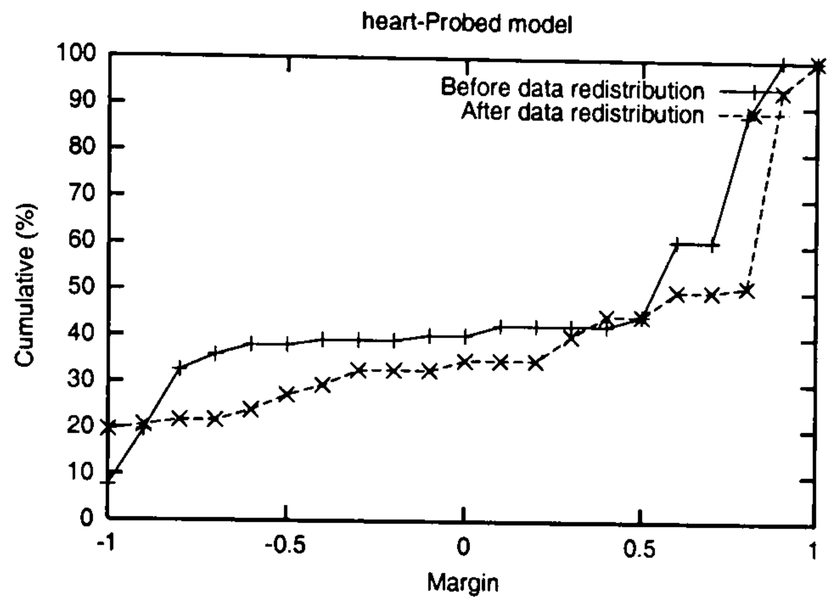
6.4.3 Effect of the stability of the learning algorithm on the data distribution adjusting techniques

Margin distribution graphs of the model committees and the probed models before and after the adjustment in local data distributions in the first three experiments are shown in Figure 6.2. In the experiments where C4.5 (Figure 6.2 (a) and (b)) and PEBLS (Figure 6.2 (e) and (f)) are used as the local learning algorithms, the graphs show that the adjustment in local data distributions makes margin distribution graphs closer to the ideal shape.

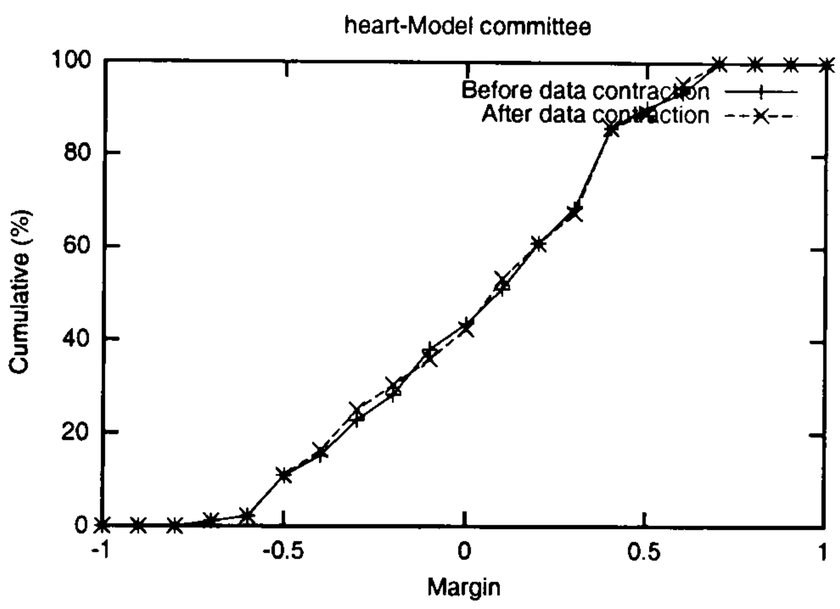
Figure 6.2 (c), which is the margin distribution graphs of the model committee whose local models are learnt by NBC, shows that the adjustment in local data distributions rarely affects the shape of the graph. This stability in margin distribution also leads to a small improvement in the shape of the graph in Figure 6.2 (d), which shows the margin distribution graphs of the probed models where the local models are learnt by NBC before and after the adjustment in local data distributions. This stability results in no improvement in the classification accuracy of the probed model after the adjustment in local data distributions. An investigation in the local models in Figure 6.2 (c) reveals that the data contraction technique has limited effect on them. This is due to the stability of NBC. Compared to other algorithms, such as C4.5, NBC is quite stable to changes in the training data [6]. This is because NBC forms a model from the prior probabilities that are calculated by counting the frequency of the quantity of interest from the whole training data. Since these prior probabilities are relative to the whole data, a small change in the training data will have a small effect on them and the output model. In contrast, the predictions of C4.5 and PEBLS are made locally. In C4.5, the degree of locality increases by the depth of the tree. This is because the tree is constructed based on a subset of data that belongs to the current branch during the tree construction process. This subset of data becomes smaller as the depth of the tree increases. For PEBLS, the degree of locality can-



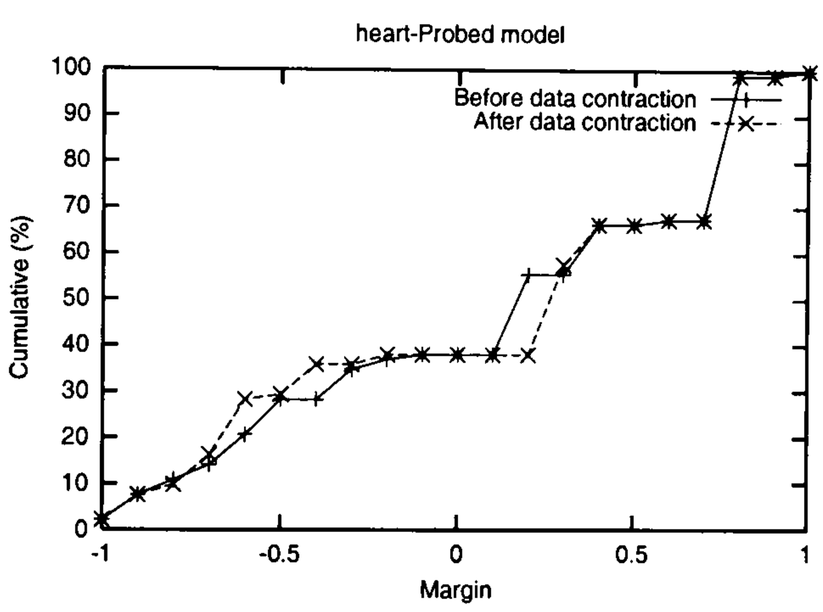
(a) C4.5-COMMITTEE



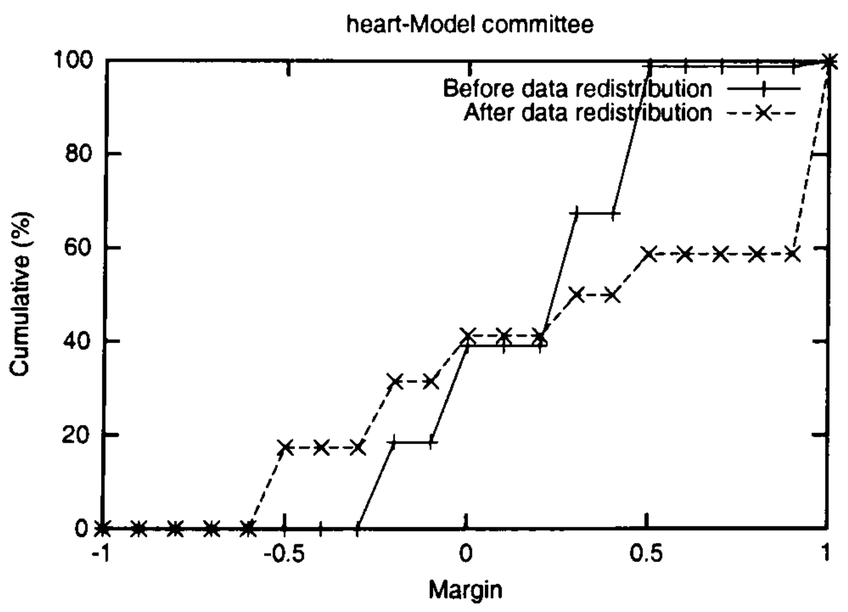
(b) C4.5-PROBE



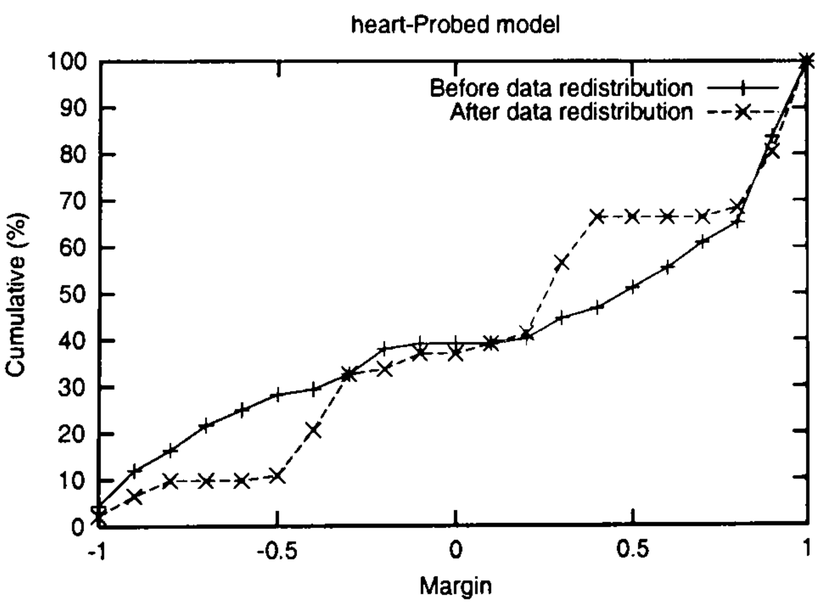
(c) NBC-COMMITTEE



(d) NBC-PROBE



(e) PEBLS-COMMITTEE



(f) PEBLS-PROBE

Figure 6.2: Margin distribution graphs of the model committees and the probed models before and after the adjustments in local data distributions.

Algorithm	Classification error (%)		
	Average local models	Model committee	Probed model
Mix	42.12	36.96	37.00

Table 6.6: Comparison of the classification errors in different stages of DLKP. The first two local models are learnt by PEBLS and the other two local models are learnt by NBC. The probed model is learnt by C4.5.

not be judged by the model because PEBLS does not explicitly form it. However, the predictions of PEBLS are decided by the class of the k nearest neighbour in the distance metric, which also introduces a degree of locality in its prediction process. Therefore, a small change in local data distributions tends to have more effect on C4.5 and PEBLS than NBC.

6.5 Integration of models learnt from different learning algorithms

In this section, we investigate the application of DLKP to a model committee whose local models are learnt by different learning algorithms. We apply different learning algorithms to each local data set, and use the model that has the lowest classification error as the representative model of that data. That is, we aim at probing model from the best model committee[3]. The classification errors resulting from applying C4.5, SBC and PEBLS to each local data set are shown in Table 6.4. From Table 6.4, the models whose errors are displayed in bold face are included in the model committee. The classification errors resulted from applying DLKP to this model committee are shown in Table 6.6. The classification error of the probed model in Table 6.6 is 5.12% lower than the average error of the corresponding local models and 3.20% lower than that of the model that is centrally learnt by C4.5. Compared with the error of the probed models whose local models are learnt by the same learning algorithm (the sixth column of Table 6.5: the classification errors of the probed models

[3]With respect to the classification accuracy.

before adjusting local data distributions), the probed model whose local models are learnt by different learning algorithms has the lowest classification error. Note that an adjustment in local data distributions is not performed on this experiment because two of the four chosen models are learnt by SBC, and the results from the first three experiments shows that an adjustment in local data distributions has small effect when SBC is used as the local learning algorithm.

6.6 Discussion on the results of the case study

From the results of the four experiments, it can be concluded that DLKP is applicable to real distributed data. The results of the experiments on the HEART DISEASE data sets show that DLKP reaches the goal set out in Chapter 1. For the HEART DISEASE data sets, the classification errors of the probed models are shown to be even lower than the average errors of the local models.

Also, DLKP can be applied to different classification learning algorithms and there is no constraint that the local learning algorithms have to be the same. The results from the fourth experiment show that the model committee whose local models are learnt from different learning algorithms yields a probed model of lower classification error than the committee whose local models are learnt from the same learning algorithm. However, the results of the experiments on using PEBLS as the local learning algorithm shows that when the local models do not give a class prediction as a set of probabilities for each class, the margin distribution graph of the committee of such models is not very smooth. This type of margin distribution graph (Figure 6.1 (c)) can raise a problem in the interpretation of the main slope.

With respect to an improvement achieved by the adjustment in local data distributions, the results from the first three experiments show that the adjustment in local data distributions is more effective when the local learning algorithms are sensitive to changes in the training data. These algorithms are likely to be the algorithms that generate models that make predictions locally rather than globally. However, the experimental results show that when the local learning al-

gorithm is insensitive to changes in the training data (i.e. NBC), the adjustment in local data distributions neither decreases the classification error of the probed model nor increases it.

6.7 Summary

In this chapter, DLKP is evaluated with real distributed data. The case study shows that DLKP is an effective model integration framework for real distributed data and is a generic framework for different classification learning algorithms. The data distribution adjusting phase of DLKP, however, is more effective when it is applied to a model committee whose local learning algorithms are sensitive to changes in the training data.

The case study also shows that DLKP can be an integration approach for different types of local models i.e. models learnt by different classification learning algorithms. This property of DLKP is beneficial in the situation where each local data is suitable to different learning algorithm.

In the next chapter, the application of DLKP on a practical distribution data mining system is illustrated. Its associated cost is analysed.

Chapter 7

Implementation of DLKP

This chapter identifies system requirements for implementing DLKP. We show an example of the implementation of DLKP on the KENSINGTON system, which is one of the foremost distributed data mining systems. The implementation of each of the four main scenarios of DLKP under KENSINGTON is described. The performance issues of DLKP are investigated.

7.1 Introduction

In the previous two chapters, we evaluated DLKP by the empirical studies. In this chapter, we discuss another important property in the design of the framework, its implementation on a practical system. Since DLKP focuses on the integration of data that are under the same schema, it is suitable to be applied in the environment where data are distributed by the nature of the applications, for example supermarket point-of-sale data that are distributed by locations of its branches. In this scenario, data are supposed to be collected in the same schema. Their distributions, however, are supposed to vary by the behaviour of the consumers of each branch. DLKP is also applicable to the environment where one large set of data is partitioned to speed up the learning process because, in this scenario, data are guaranteed to be under the same schema. In any of the two cases, however, users should be aware that the comprehensibility of the model integrated with DLKP comes with additional computation and communication

costs. Thus, in the scenario where comprehensibility of the integrated model is less concerned than the incurred costs, for example the situation where the communication cost is extremely high, prediction schemes, such as those listed in Section 4.3.1, can be used to directly combine the predictions of local models.

DLKP can be implemented on any system that supports distributed data mining i.e. any system that supports a set of independent data mining sites that can communicate to one another by transmitting and receiving of data. One such a system is the KENSINGTON system [22, 23], which has been developed by InforSense Ltd., a spin-off company of Imperial College. In this chapter, the practical implementation of DLKP on the KENSINGTON version 1.0 is discussed.

To provide a picture of where the processes of DLKP will take place, we first describe the architecture of KENSINGTON. This is followed by the system requirements for implementing DLKP on KENSINGTON. Then, the implementation of each of the four main scenarios of DLKP on KENSINGTON is illustrated. Its associated computation and communication costs are analysed. This is followed by an investigation in the performance issues of DLKP.

7.2 KENSINGTON architecture

KENSINGTON [23] is a distributed platform system for deploying enterprise data analysis solutions. It enables collaborative knowledge discovery and knowledge management on an Intranet and across the Internet. KENSINGTON has a three-tier client/server architecture which includes the client, the application server and the third-tier servers. The three-tier client/server components and their interaction are shown in Figure 7.1. The client (layer (a) in Figure 7.1) handles interactive creation of data mining tasks, visualisation of data and models, and sampling of the data. The application server (layer (b) in Figure 7.1) authenticates users, provides persistent disk storage and access control for objects, and controls task execution and data management. The third tier server (layer (c) in Figure 7.1) provides both databases and high-performance data mining services.

To maximise its portability and extensibility, KENSINGTON uses the Enter-

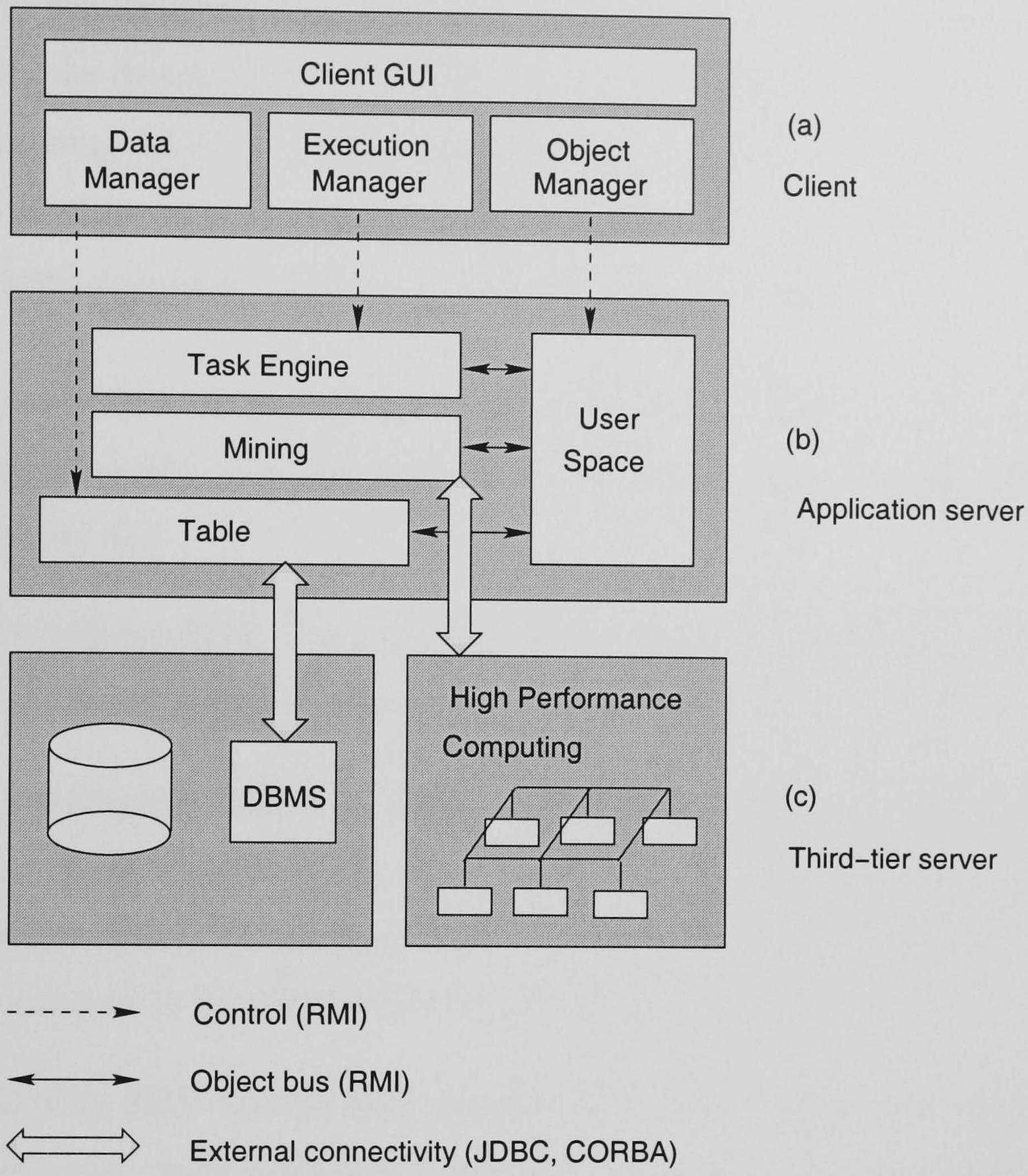


Figure 7.1: Kensington tree-tier client-server components and their interaction.

prise JavaBeans (EJB) component architecture and is implemented in the Java programming language. KENSINGTON uses the **C**ommon **O**bject **R**equest **B**roker **A**rchitecture (CORBA)[1] as an interface to parallel data mining application functions. The access of databases on the Internet is performed via a JDBC connection. JDBC is a Java **A**pplication **P**rograming **I**nterface (API) for executing SQL statements. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API. The interactive client is implemented using Java Beans, which are reusable software components that can be visually manipulated via the user interface. Further

[1]CORBA is the vendor-independent architecture and infrastructure that computer applications use to work together over networks.

technical details of the implementation of KENSINGTON can be found in [23].

Next, the design of the application server (EJB Server), which is where the control among distributed sites takes place will be discussed. This is to provide a picture of where the physical processes of DLKP will take place. The discussion will be used as a basis for the cost analysis in Section 7.4.

The application server consists of four EJB classes, which execute in an EJB server environment. The four EJB classes that constitute the application server functionality are:

- **UserSpace EJB** The UserSpace EJB manages a user's data mining tasks and results that are persistent.

- **TaskEngine EJB** The TaskEngine EJB is used as the central part for the client to control the execution of data mining tasks including data preprocessing, model generation, and model evaluation; execution can be controlled to be distributed between different sites.

- **Mining EJB** Mining EJB handles the interface with the mining components e.g. high performance parallel components that are accessed via a protocol, such as CORBA.

- **Table EJB** Table EJB is responsible for the import and the storage of the data set via JDBC.

A key advantage of the EJB architecture is the ability to run multiple cooperating servers at different sites. Therefore, a single application server configuration can contain multiple distributed sites for storing data and/or multiple distributed sites for mining data. A single user's data and group data can be distributed across the sites that contain a Table EJB, and the user can initiate a mining task from any of the sites that contain a Mining EJB. An example of the screen shot of KENSINGTON when performing distributed learning is shown in Figure 7.2.

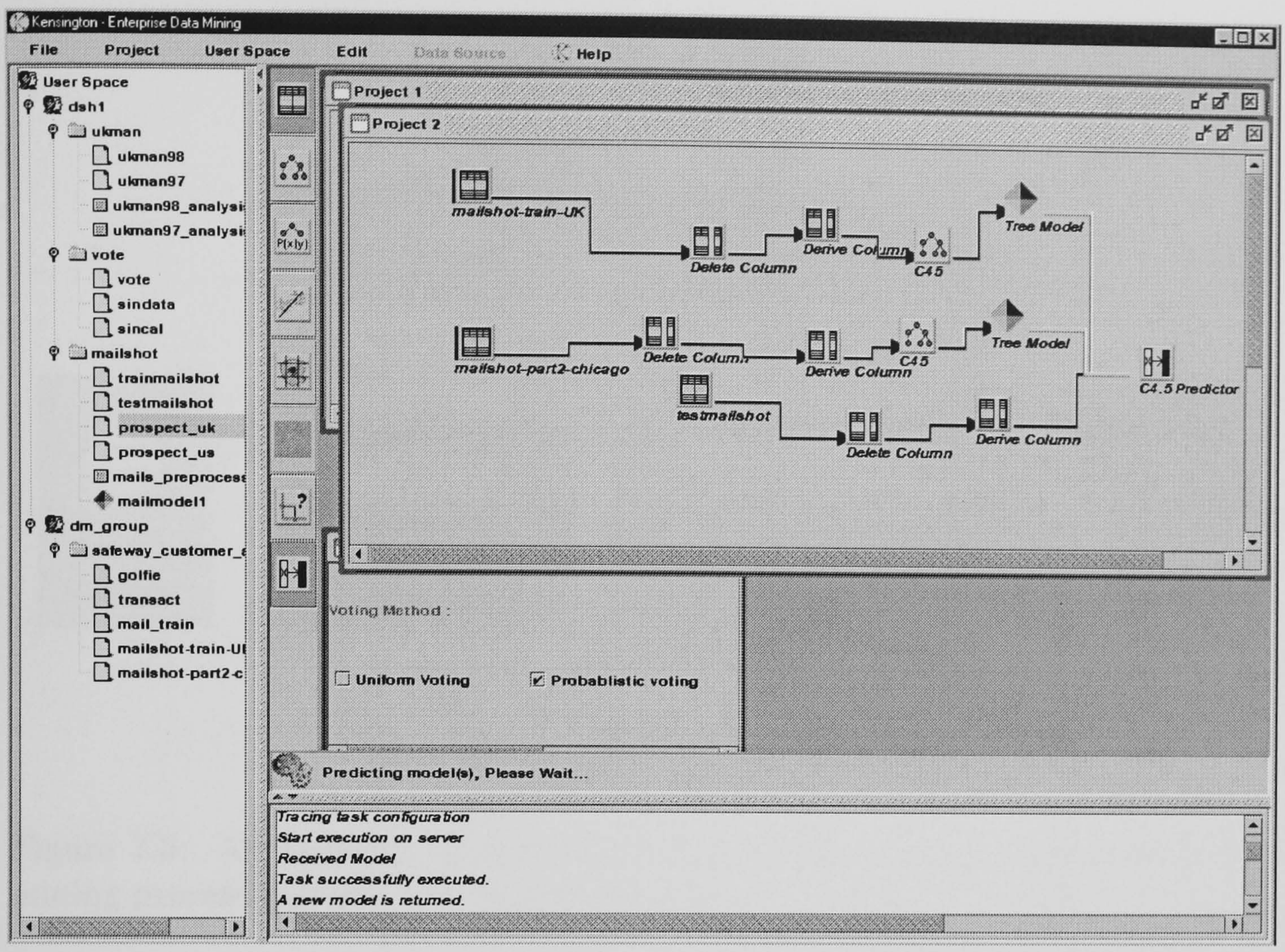


Figure 7.2: A screen shot of learning with a model committee on KENSINGTON.

7.3 System requirements for implementing DLKP on KENSINGTON

Figure 7.3 is an example of KENSINGTON configuration for a co-operative data mining process between two distributed sites. Each host site consists of a machine that runs an instance of the KENSINGTON EJBserver. The client is configured to access both server sites. This configuration enables users to access and mine data concurrently across the network. The most important server is the one that hosts the TaskEngine EJB and the UserSpace EJB because it will act as the coordinating site of the whole process. It is important to note that the TaskEngine EJB will act as an coordinator for control purpose only. Data that are distributed at each local site are loaded and mined locally.

The minimum requirements of the system for applying DLKP on KENSINGTON in the above configuration are as follows:

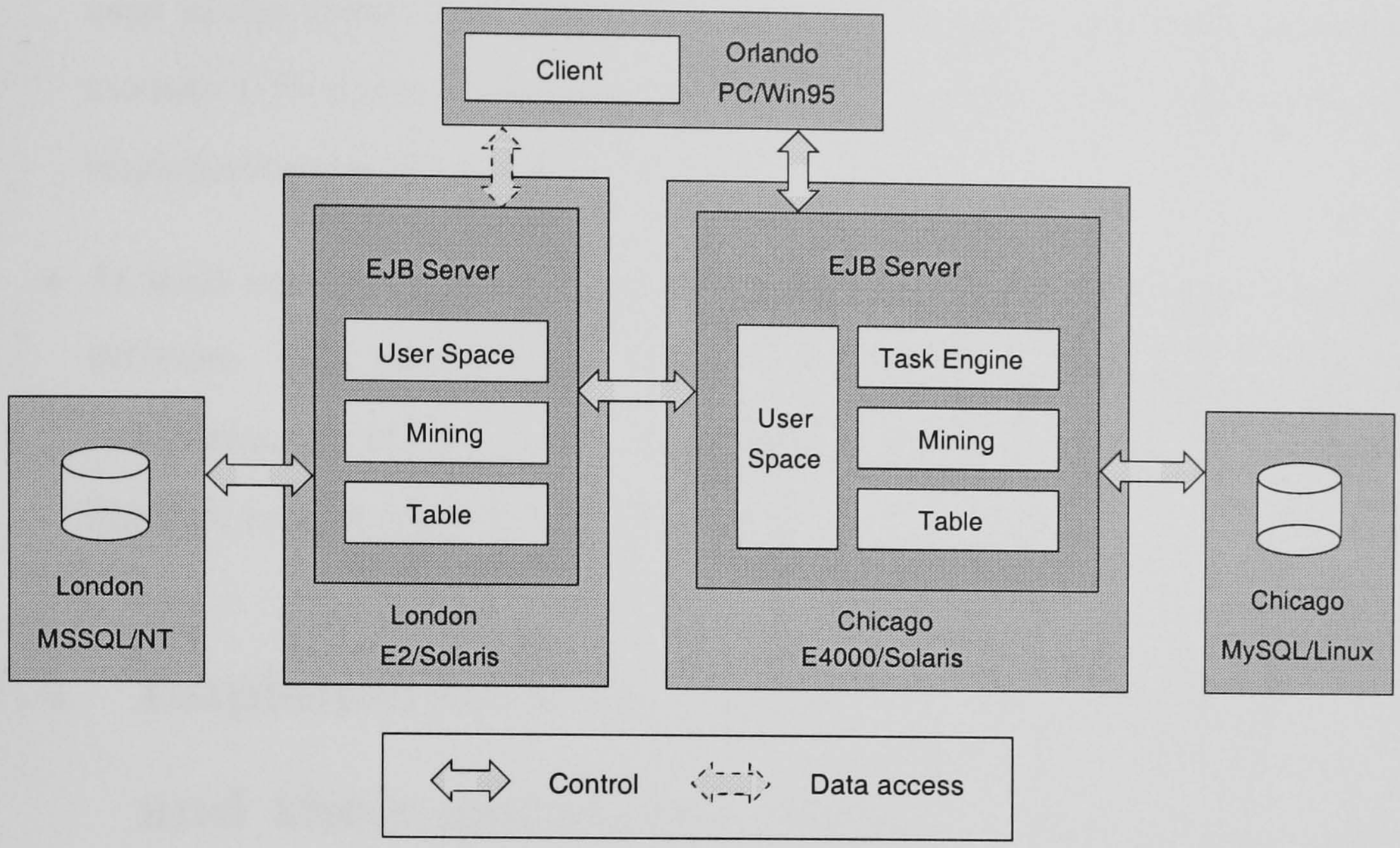


Figure 7.3: An example of KENSINGTON configuration for co-operating data mining process between two distributed sites.

- To be able to perform local mining task, each distributed site is required to have a machine that is capable of hosting a EJBServer that contains Table EJB, Mining EJB, UserSpace EJB and local databases.

- At least one site has a machine that is capable of hosting all the four components of the EJB Server. This site will act as a coordinator of the model integration process. Assume that there are n local sites and all n local databases are of roughly equal size, according to the setting used in our empirical study in Chapter 5, the Table EJB of the coordinating site is expected to be able to accommodate data at least three time the average size of the local databases[2]. That is, it should be able to contain its own local data, the probing set (during the Knowledge Probing phase), and the predictions from other sites (during prediction combining process). However, because the Table EJB of KENSINGTON is a disk-based component, the maximum size of data that is required to be loaded in the memory at one time is the size of the probing set. Thus, based on the configuration

[2]Provided that $n <$ number of attributes.

used in our study[3], the Table EJB of the coordinating site that can accommodate 1.15 times the average size of the local data sets is sufficient for the implementation of DLKP.

- At least one machine is capable of hosting the client component of KENSINGTON. This machine can be located at any place that has an Internet connection. It will act as the interface between the user and the coordinating site, and be used to view the result.

7.4 Implementations of DLKP on KENSINGTON and their associated costs

In the following subsections, the implementation of each scenario of DLKP is described. Its associated computation and communication costs are analysed. Since every scenario of DLKP starts from the preliminary evaluation process, which consists of the *Local Model Learning phase* and the process for *generating a margin distribution graph*, the preliminary evaluation process is first described. This is followed by the description of the rest of the processes in each of the four main scenarios. The analysis of the computation and communication costs for the rest of this chapter is based on the following assumptions:

- For computations that proceed concurrently at each distributed site, the computation time taken into consideration is equal to the computation time of a single local site.

- In the calculation of the cost of transmitting and receiving data between the coordinating site and other local sites, the communication cost taken into account is the communication cost between the coordinating site and the most remote local site.

The symbols used in the analysis are listed in Table 7.1.

[3]Provided that the learning algorithm requires the whole training data to be loaded into memory during the learning process.

Sym.	Meaning
n	Number of distributed sites
m	Number of instances in a local training set
m_i	Average number of instances involved in data distribution adjusting process
I	Number of iterations in data distribution adjusting process
$Size_d$	Size of an instance in words
$Size_p$	Size of a class prediction in words
$Size_i$	Size of a single index item in words
$Size_m$	Size of the largest local model in words
C_l	Unit of computation for learning a single instance
C_p	Unit of computation for predicting a single instance
C_c	Unit of computation for combining predictions of a single instance
C_m	Unit of computation for calculating margin of a single prediction
T_s	Start up time of communication latency
T_w	Per-word transfer time of communication latency to the most remote site

Table 7.1: Symbols used in the analysis of computation and communication costs.

7.4.1 Preliminary evaluation process

The physical processes of the preliminary evaluation can be described as follows:

- *Local Model Learning phase* This process is executed locally at each site. Each local learning process is performed on its own local data.

- *Generating a margin distribution graph* In practical applications, it is unlikely that there is an evaluation set existing. Thus, data that will be used to generate a margin distribution graph can be sampled from all local data and stored in the Table EJB of the coordinating site. This data will be referred to as the evaluation set. This evaluation set is then sent to the other local sites. Each local site loads the evaluation set into its Table EJB and uses its local model to predict the class values of the evaluation set. These predictions are sent back to the coordinating site. The coordinating site combines them and generates a margin distribution graph. The graph together with a recommendation will be displayed to the user at the Client site. The user decides which action he will take as the next step.

Process	Execution cost
Local model learning	$m \times C_l$
The coordinating site selects an evaluation set	$T_s + \frac{\frac{m}{n} \times Size_d}{T_w}$
The coordinating site transmits the evaluation set	$T_s + \frac{m \times Size_d}{T_w}$
Each local model makes predictions on the evaluation set	$m \times C_p$
The predictions are transmitted to the coordinating site	$T_s + \frac{m \times Size_p}{T_w}$
The coordinating site calculates margin and generates a margin distribution graph	$m \times C_m$

Table 7.2: The computation and communication cost of the preliminary evaluation process.

The computation and communication cost in each phase of the preliminary evaluation process is shown in Table 7.2. The cost of all phases in Table 7.2 can be summed to

$$3T_s + \frac{m}{T_w}\left(\frac{Size_d}{n} + Size_d + Size_p\right) + m\left(C_l + C_p + C_m\right). \tag{7.1}$$

This is a primary cost of all scenarios.

7.4.2 First scenario: Arbitrarily selecting the final model

In this scenario, the additional computation cost is zero if the user chooses to perform arbitrarily selection. That is, the TaskEngine EJB only arbitrarily selects one of the local models to use as a representation of the rest. However, there is

an additional communication cost of delivering that model from its location to the coordinating site for future use.

Therefore, the only additional cost from the preliminary evaluation phase is the communication cost of transferring the chosen model to the coordinating site, which is

$$T_s + \frac{Size_m}{T_w}. \tag{7.2}$$

7.4.3 Second scenario: Direct Knowledge Probing

In this scenario, the processes and their physical locations can be described as follows:

- *Probing set selection* Firstly, the probing set is required to be sampled from all local data. Following the approach used in our empirical study, the probing set is 15% larger than the average size of the local data. To minimise the communication cost, the evaluation set from the preliminary evaluation process can be used for the main part of the probing set. This is different from the empirical study where the probing set and the evaluation set are separated for the measurement purpose. In practical applications where the evaluation set does not exist, the probing set will be generated by the same approach as the evaluation set. Therefore, it is reasonable to use the evaluation set in the preliminary evaluation phase as the main part of the probing set. In this case, an additional 15% will be sampled to complete the probing set.

- *Model committee makes predictions on the probing set* The probing set is sent and stored in the Table EJBs of all other local sites. Each local site uses its local model to predict class values for the probing set. The predictions are sent back and stored in the Table EJB of the coordinating site.

Process	Execution cost
The coordinating site selects addition 15% of the probing set	$T_s + \frac{\frac{m}{n} \times 0.15 \times Size_d}{T_w}$
The coordinating site transmits the additional probing data	$T_s + \frac{m \times 0.15 \times Size_d}{T_w}$
Each local model makes predictions on the probing set	$m \times 1.15 \times C_p$
The predictions are transmitted to the coordinating site	$T_s + \frac{m \times 1.15 \times Size_p}{T_w}$
The coordinating site combines predictions, relabels the probing set, and probes a global model	$m \times 1.15 \times (C_c + C_l)$

Table 7.3: The computation and communication cost of the scenario of directly applying the Knowledge Probing phase.

- *Relabelling the probing set and probing the global model* The coordinating site combines the predictions and makes necessary relabelling. After that, it learns a probed model from the relabelled probing set and stores the probed model in its local UserSpace EJB for future use.

The details of the computation and communication cost of directly applying the Knowledge Probing phase are given in Table 7.3. The cost of all phases can be summed to

$$3T_s + \frac{m}{T_w}\left(\frac{0.15Size_d}{n} + 0.15Size_d + 1.15Size_p\right) + 1.15m\left(C_l + C_p + C_c\right). \tag{7.3}$$

7.4.4 The third scenario: Knowledge Probing after applying the data redistribution technique

Taking the same strategy as the previous scenario, the evaluation set is used as the main part of the probing set. However, under this scenario, a consistency test set of the same size as the local data set has to be randomly selected from the local data. Note that the consistency test set should not be shared with the probing set because it can lead to the over-fitting of the model committee to the probing set. The processes and their physical locations are listed as follows:

- *Consistency test set selection* A consistency test set is first sampled from all local data and kept in the Table EJB of the coordinating site.

- *Local models make predictions on the consistency test set* The consistency test set is distributed to other local sites[4] for them to make predictions. The predictions are returned and kept in the Table EJB of the coordinating site.

- *Data redistribution* The coordinating site combines the predictions and makes an index of the instances whose margins are between -0.2 and 0.2. After that, this index is distributed to other local sites.

- *Re-learning local models* Each local site adds instances (from the consistency test set) that are listed in the index to its own local data set and re-learns a local model.

- *Model committee makes predictions on the probing set* The coordinating site selects and sends the additional part of the probing set to all other sites[5]. Each local site makes predictions on the probing set. The predictions are sent back and stored in the Table EJB of the coordinating site.

- *Relabelling the probing set and probing the global tree* The coordinating site combines the predictions and makes any necessary relabelling. It learns a probed model from this relabelled probing set.

---

[4]Only in the first iteration.
[5]Only in the first iteration.

The process iterates until no instance is redistributed in the *data redistribution* phase or the classification error of the probed model increases.

The details of the computation and communication cost of Knowledge Probing after applying the data redistribution technique are shown in Table 7.4. The cost of all phases can be summed to

$$4T_s + \frac{m}{T_w}\left(\frac{1.15Size_d}{n} + 1.15Size_d\right) + \left(3T_s + \frac{m}{T_w}(2.15Size_p + Size_i) + m(2.15C_l + 2.15C_p + 1.15C_c + C_m) + m_iC_l\right)I. \tag{7.4}$$

7.4.5 The fourth scenario: Knowledge Probing after applying the data contraction technique

Taking the same strategy as the previous scenario, the evaluation set is used as the main part of the probing set. In this scenario, however, each local data set is used to contract the coverage of its own local model. The processes and their physical locations are as follows:

- *Data contraction* This process is performed locally at each site. Each local model makes predictions on its own local data. The instance whose margin of its class prediction is between -0.2 and 0.2 is removed from the local training set.

- *Re-learning local models* Each local site re-learns a local model from the contracted training set.

- *Model committee makes predictions on the probing set* The coordinating set selects and sends the additional probing data to all other local sites[6]. Each local site makes predictions on the probing set. The predictions are sent back and stored in the Table EJB of the coordinating site.

[6]Only in the first iteration.

Process	Execution cost
The coordinating site selects an additional 15% probing set	$T_s + \frac{\frac{m}{n} \times 0.15 \times Size_d}{T_w}$
The coordinating site transmits the additional probing data	$T_s + \frac{m \times 0.15 \times Size_d}{T_w}$
The coordinating site selects a consistency test set	$T_s + \frac{\frac{m}{n} \times Size_d}{T_w}$
The coordinating site transmits the consistency test set	$T_s + \frac{m \times Size_d}{T_w}$
(*) Each local model makes predictions on the consistency test set	$m \times C_p$
The predictions are transmitted to the coordinating site	$T_s + \frac{m \times Size_p}{T_w}$
The coordinating site calculates margin of the model committee's predictions on the consistency test set	$m \times C_m$
The coordinating site transmits the index of the redistributed instances	$T_s + \frac{m \times Size_i}{T_w}$
Each local site re-learns a local model from the expanded data set	$(m + m_i) \times C_l$
Each new local model makes predictions on the probing set	$m \times 1.15 \times C_p$
The predictions are transmitted to the coordinating site	$T_s + \frac{m \times 1.15 \times Size_p}{T_w}$
The coordinating site combines the predictions, relabels the probing set and probes a model	$m \times 1.15 \times (C_c + C_l)$
Repeat (*) until the stopping criterion is met	

Table 7.4: Computation and communication cost of the scenario where the data redistribution technique is applied before the Knowledge Probing phase.

- *Relabelling the probing set and probing the global tree* The coordinating site combines the predictions, makes any necessary relabelling, and learns a probed model from the relabelled probing set.

The process iterates until there is no further data to be contracted or the classification error of the probed model increases.

The details of the computation and communication cost of Knowledge Probing after applying the data contraction technique are given in Table 7.5. The cost of all phases can be summed to

$$2T_s + \frac{m}{T_w}\left(\frac{0.15Size_d}{n} + 0.15Size_d\right) + \left(T_s + m(2.15C_l + 2.15C_p + 1.15C_c + C_m + \frac{1.15Size_p}{T_w}) - m_iC_l\right)I. \tag{7.5}$$

7.5 Performance analysis

The previous section shows that the cost of applying DLKP on the KENSINGTON system ranges between

$$4T_s + \frac{m}{T_w}\left(\frac{Size_d}{n} + Size_d + Size_p + \frac{Size_m}{m}\right) + m\left(C_l + C_p + C_m\right) \tag{7.6}$$

and

$$7T_s + \frac{m}{T_w}\left(\frac{2.15Size_d}{n} + 2.15Size_d + Size_p\right) + m\left(C_l + C_p + C_m\right) + \left(3T_s + \frac{m}{T_w}(2.15Size_p + Size_i) + m(2.15C_l + 2.15C_p + 1.15C_c + C_m) + m_iC_l\right)I. \tag{7.7}$$

Cost function 7.6 is the result of the summation of the cost of the preliminary evaluation (Cost function 7.1) and the cost of the scenario where the final model is

Process	Execution cost
The coordinating site selects an additional 15% probing data	$T_s + \frac{\frac{m}{n} \times 0.15 \times Size_d}{T_w}$
The coordinating site transmits the additional probing data	$T_s + \frac{m \times 0.15 \times Size_d}{T_w}$
(*) Each local model makes predictions on its local data set	$m \times C_p$
Each local model calculates the margins of its predictions	$m \times C_m$
The contracting data is removed from the local data set and each local site re-learns a local model from its contracted data set	$(m - m_i) \times C_l$
Each local model makes predictions on the probing set	$m \times 1.15 \times C_p$
The predictions are transmitted to the coordinating site	$T_s + \frac{m \times 1.15 \times Size_p}{T_w}$
The coordinating site combines predictions, relabels the probing set, and probes a global model	$m \times 1.15 \times (C_c + C_l)$
Repeat (*) until the stopping criterion is met	

Table 7.5: Computation and communication cost of the scenario where the data contraction technique is applied before the Knowledge Probing phase.

Scenario	Costs
1st - Arbitrarily selecting	$O(Size_d + Size_p)$
2nd - Direct Knowledge Probing	$O(Size_d + Size_p)$
3rd - Knowledge Probing after Data Redistribution	$O(Size_d + Size_pI)$
4th - Knowledge Probing after Data Contraction	$O(Size_d + Size_pI)$

Table 7.6: Costs of the four main scenarios with respect to the characteristics of the problem domain.

arbitrarily selected from the local models (Cost function 7.2). This represents the lower bound of the cost of applying DLKP on KENSINGTON. Cost function 7.7 is the result of the summation of the cost of the preliminary evaluation (Cost function 7.1) and the cost of the scenario where the data redistribution technique is applied before the Knowledge Probing phase (Cost function 7.4). This represents the upper bound of the cost of applying DLKP on KENSINGTON.

7.5.1 Effect of varying problem characteristics on DLKP

The analysis of the four main scenarios of DLKP can be conducted by deriving their costs as a function of the characteristics of the problem domain. That is, we take the size of an instance ($Size_d$), the size of a class prediction ($Size_p$), and the number of iterations involved in data distribution adjusting process (I) as variables and take the other units as constants. $Size_d$ and $Size_p$ are related to the number of attributes and the number of classes[7] in the problem domain, respectively. Under this convention, the costs of the four main scenarios are shown in Table 7.6. Thus, the cost of DLKP will be between $O(Size_d + Size_p)$ and $O(Size_d + Size_pI)$.

With respect to the characteristics of the problem domain ($Size_d$, $Size_p$), the cost difference between the cheapest and the most expensive scenarios of DLKP is in $O(Size_pI)$. If we fix I as a constant, this means that the gap between the best and the worst scenarios of DLKP is wider when $Size_p$ increases, i.e. when the problem domain has a high number of classes. This leads to a trade-off

[7]In the case where a prediction is given as a set of the probabilities for each class.

between the accuracy and the cost especially in the third and the fourth scenarios where the Knowledge Probing phase should be applied after the data distribution adjusting process. That is, the accuracy can be improved by adjusting local data distributions with the cost that is a function of $Size_p$ in each iteration. This cost become prohibitive when the number of classes is high. Therefore, for problem domains with a high number of classes, the additional costs of applying the data distribution adjusting techniques can become an important performance issue.

7.5.2 Cost comparison between DLKP and the centralised learning approach

In order to compare the costs involved in applying DLKP and the centralised learning approach, we first define the cost of applying the centralised learning approach on KENSINGTON. The cost involves the communication cost of migrating all the local data to one site and the computation cost of learning from the combined data. Therefore, we have:

$$T_s + \frac{mSize_d}{T_w} + nmC_l. \tag{7.8}$$

Assume that both approaches are implemented on the same machine under the same environment and they are used to learn the same problem domain, Cost function 7.6, 7.7 and 7.8 can be simplified by taking the units of size ($Size_d$, $size_p$, $Size_i$, $Size_m$), the units of computation (C_l, C_p, C_c, C_m), and the units of communication cost (T_s, T_w) as constants. After simplification, the lower and the upper bound of the cost of applying DLKP on KENSINGTON is $O(m)$ and $O(mI)$, respectively. On the other hand, after simplification, the cost of applying the centralised learning approach is in $O(nm)$. Let $D = nm$, the cost of DLKP can be re-written as between $O(\frac{D}{n})$ and $O(\frac{DI}{n})$, and the cost of applying the centralised learning approach can be re-written as in $O(D)$.

As a result, we can conclude that in the cheapest scenario of DLKP, DLKP provides a speedup of n over the centralised learning approach. In the most

expensive scenario, DLKP will provide a speedup only when the number of local sites is greater than the number of iterations in the data distribution adjusting process i.e. when $n > kI$, where k is a constant[8]. However, our empirical study on the data sets that involve in the data distribution adjusting processes (ABALONE, GERMAN and LED7 data sets), shows that the average number of the iterations is 4.67 times. Therefore, based on our empirical results, the application of DLKP on KENSINGTON will have a speedup over the centralised learning approach when the number of the local models is relatively small.

7.6 Summary

In this chapter, the system requirements for implementing DLKP are addressed. We show an example of the implementation of the four main scenarios of DLKP on the KENSINGTON system, which is a system that satisfies the system requirements of DLKP. The analysis of the computation and communication costs of DLKP shows that it is feasible to implement DLKP on a practical distributed data mining system.

The analysis shows that among the four main scenarios of DLKP, the cost difference between the cheapest and the most expensive scenarios is $O(Size_p I)$. With respect to the characteristics of the problem domain, the gap becomes wider as the number of classes increases. Thus, for problem domains with a high number of classes, the application of the data distribution adjusting techniques should be applied with the awareness of the trade-off between an improvement in accuracy and the additional execution and communication costs.

With respect to the size of the local data and the number of the local models, given a fixed problem domain, the cost of applying DLKP on KENSINGTON ranges between $O(m)$ and $O(mI)$. The analysis shows that the cheapest scenario of DLKP provides a speedup of n over the centralised learning approach. The

[8]Note that there is a constant k associated with I because some constants are removed during simplification. If the unit symbols involved in the derivation of the cost functions, i.e. the symbols in Table 7.1, are given, k can be calculated.

most expensive scenario of DLKP will provide a speedup when the number of the local sites is greater than the average number of the iteration in the data distribution adjusting process. Our empirical study in Chapter 5, however, shows that the average number of such iterations (4.67 times) is small. Therefore, there is a high possibility that DLKP will provide a speedup over the centralised learning approach.

In the next chapter, related works of DLKP are reviewed in three aspects according to their relation to each of the three components of DLKP.

Chapter 8

Related Works

This chapter is aimed at providing information on current state of the art in the area of model integration in distributed data mining and multiple model learning. This also aims at comparing and contrasting DLKP with related works in these areas.

8.1 Introduction

Recently, there has been an increasing number of works on model integration. They are referred to with different terms i.e. *model averaging* [65], *model ensemble* [60], *stack generalisation* [90] and *arbiter/combiner trees* [20]. All these works share a commonality in generating multiple models, using them to predict the class value of an instance and combining the predictions with a prediction scheme. This also applies to the Local Model Learning phase of DLKP.

As described in Chapter 4, DLKP consists of three main components: the Local Model Learning phase, the Knowledge Probing phase, and the techniques for adjusting local data distributions. These three components involve three aspects of learning, which are

- the aspect of learning from data partitions (distributed learning)
- the aspect of extracting a representation form from a black box model

- the aspect of active learning which includes data selection and data manipulation.

To our knowledge, there is no work that covers all three aspects of DLKP. In this chapter, we compare and contrast the works that are related to DLKP in these aspects.

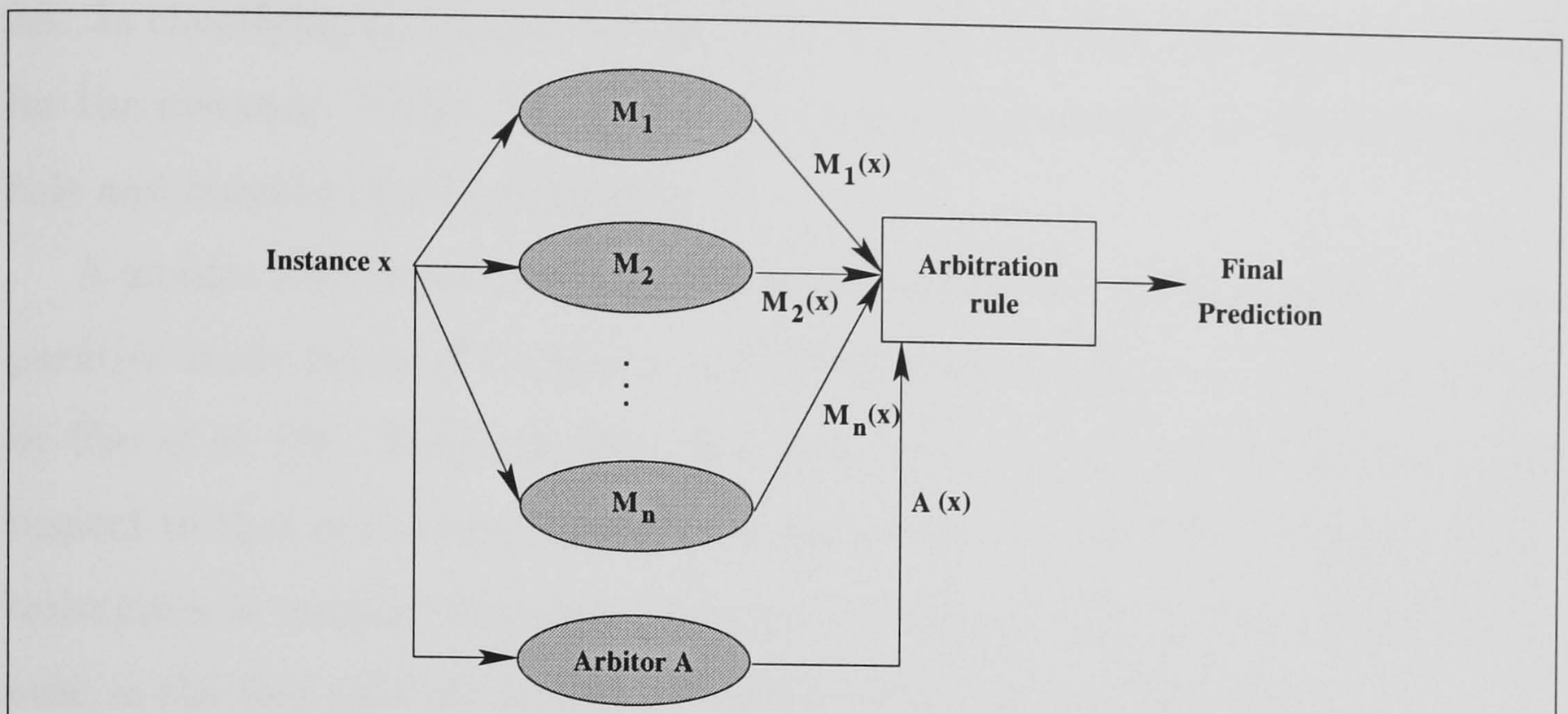
8.2 Related works in the aspect of learning from data partitions

With respect to learning from data partitions, related works include Chan & Stolfo [18, 16, 21, 19], Provost & Hennessy [77], Hall et al. [50], and Domingos [33].

Chan & Stolfo [18] propose the use of the *meta-model* as a mediator among the models that are independently learnt from data partitions. The *meta-model* is learnt from predictions of local models and, in some cases, some parts of training data. Chan & Stolfo's approach [18, 16, 21, 19] consists of two learning steps. In the first step, each local model is first learnt from its local data and then used to make predictions on an unseen data set. These predictions are used as a training set for the second learning step to generate a *meta-model*, which can be classified into two types: an *arbiter* and a *combiner*.

An **arbiter** (the top diagram in Figure 8.1) is used to arbitrate the predictions generated by the local models. Arbiters are different in a *selection rule*, a rule to select the training set, and *arbitration rule*, a scheme for combining predictions from the local models. For example, let x be an instance, $M_i(x)$ be the class predictions of x from the local models $M_i, i = 1, \ldots, n$, and $A(x)$ be the class prediction of x from an arbiter A, the selection rule and the arbitration rule can be:

- *selection rule:* An instance is selected if none of the class gathers a majority vote from n predictions.



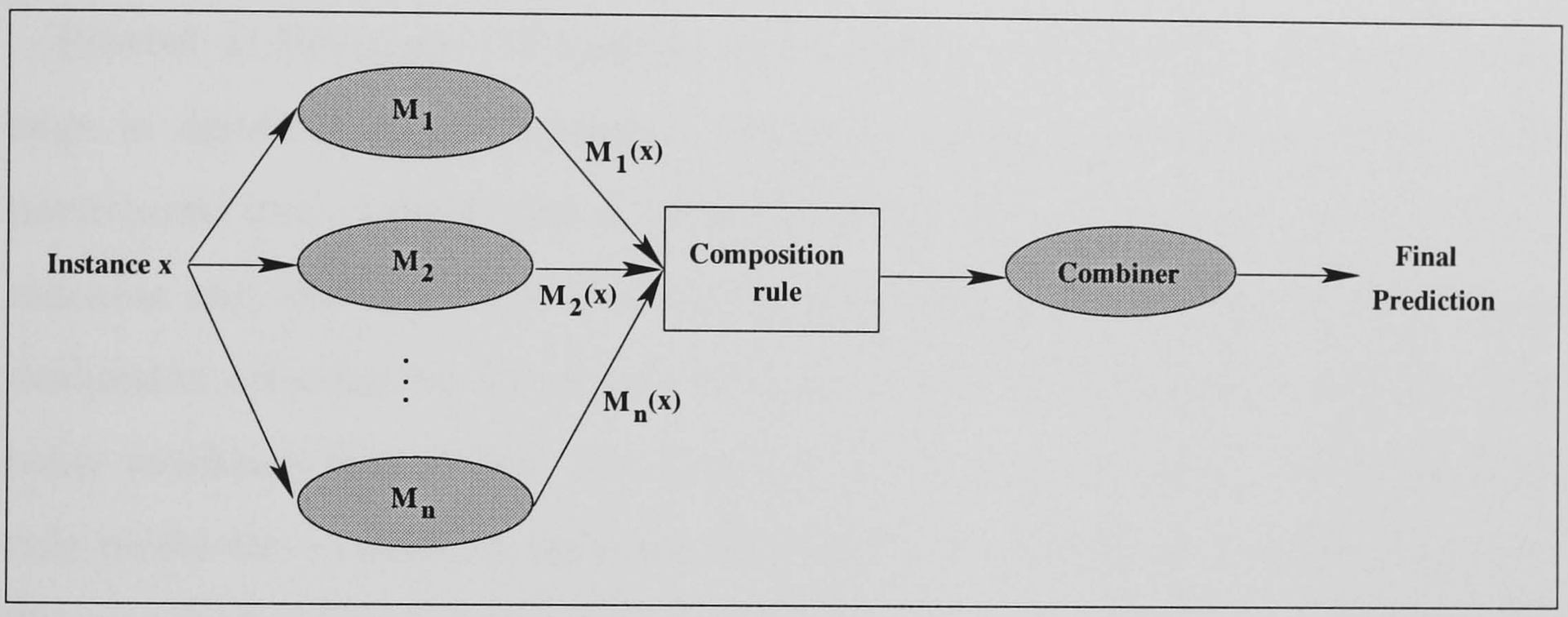


Figure 8.1: An illustration of an arbiter (the top diagram) and a combiner (the second diagram) with n local models.

- *arbitration rule:* Return the class with a plurality of vote in $M_1(x)$, $M_2(x)$, ..., $M_n(x)$ and $A(x)$ with preference given to the arbiter's prediction in the case of tie.

With these two rules, the arbiter is learnt from the instances that are confusing and can be used to resolve conflicts among the local models.

A **combiner** (the second diagram in Figure 8.1) is aimed at coalescing the predictions from local models by learning the relationship between their class predictions and the original class values. The content of the training set of the combiner is determined by a *composition rule*. One example of the composition rule is *class-combiner*. In the class-combiner, all predictions made from the local models, and the original class value of each instance, are included in the training

set. In classifying an unseen instance, the local models first give their predictions for the instance. These predictions are formatted according to the composition rule and classified by the combiner.

A similar idea is proposed in Wolpert's Stacked Generalisation [90]. A comparative study between Combiner and Stack Generalisation has been carried out by Fan et al. [39]. They conclude that both methods increase the accuracy with respect to that of the local models and the improvement in the accuracy of both techniques is roughly the same. A major problem of this type of learning, however, is the fact that the comprehensibility of the final result is lost.

Provost & Hennessy [77] employ a cooperation approach to integrate knowledge in distributed environments. In their approach, a training data is first partitioned into n disjointed subsets. Each subset is then assigned to different machine and learnt by a rule-based learning algorithm. When a rule meets the evaluation criterion on one subset of data, it will be broadcasted globally. The other machines then review the statistics of that rule on their local data. If the rule meets the evaluation criterion globally, it is retained as a satisfactory rule. The resulted rule set is shown to be a superset of the one generated when the learning is performed on the complete training set. This approach can be seen as a rule-based learning algorithm for distributed data rather than an approach for integrating existing local models.

Hall et al. [50] discuss an approach of integrating decision trees that are learnt independently from disjointed subsets of data. In their approach, each decision tree is first converted into a set of rules. These sets of rules are later merged into a single rule set. The combination process is based on the approach suggested by Williams [89], in which similar rules are combined into more general rules, but rules that match one or more examples but assign them to different classes are modified to resolve the conflict. The resulted rule set is shown to be comparable to the tree learnt from the complete data set. This approach provides an indirect way of combining decision trees. However, this is algorithm dependent because

local models are required to be in a representation form that can be converted into a set of rules. In addition, the final model is also restricted to be a set of rules only.

Domingos' RISE system [33] is a specific-to-general rule induction algorithm. RISE implements a covering algorithm in its rule expansion process. RISE can be used to generate a set of rules from each data partition. These sets of rules can either be merged into one rule set, or kept separately and used to vote for predictions. Domingos finds that the latter approach yields consistently better results than the first approach. The latter approach, however, has the same disadvantage as Chan & Stolfo's approach [18, 16, 21, 19] in that the result does not aid the understanding of the problem domain. Also, the result of RISE is restricted to be a set of rules.

8.2.1 Comparison to DLKP

DLKP is different from the works described in this section, in two main aspects: the generality of the approach and the level of the result. With respect to the generality of the approach, in all of these works, except for Chan & Stolfo [18, 19], the local models and the final results are required to be rule sets. DLKP and Chan & Stolfo's model are more generic in that the local models can be any type of classifiers. However, with respect to the level of the result, the result of Chan & Stolfo's approach is in the meta-level where a prediction of an instance can only be derived after the instance has been classified by the models in the base level (the local models). As a result of the introduction of an additional level of models, the comprehensibility of the prediction process is lost. In contrast, the result of DLKP is in the base level where it can be directly used to make predictions. In addition, the comprehensibility of the result is preserved.

8.3 Related works in the aspect of extracting the representation form of a model

The related works in the aspect of extracting the representation form of a model include Craven & Shavlik's TREPAN algorithm [27, 29], Breiman & Shang's born again tree [13], and Domingos' CMM algorithm [34].

One of the pioneering works in extracting a representation form from a black box model, is Craven & Shavlik's TREPAN algorithm [27, 29]. By aiming to extract a comprehensible representation form from neural networks, Craven & Shavlik [28] observe that to consider the extraction process as a *search* problem, poses limitations on the quality of the result. This is because many assumptions, such as the approximation of the threshold units and the assumption on the independence of the activations of the hidden units, have to be made during the extraction process. Improper assumptions can deteriorate the quality of the result. To avoid this limitation, Craven & Shavlik present TREPAN [29] as a query-driven algorithm for extracting a decision tree from trained neural networks. TREPAN views tree extraction as a *learning* task where the target concept is the function computed by the networks. The attributes of the data used in the extraction process are the same as those used to train the networks. The data used in the extraction process include the training data of the neural networks and some artificially generated data. TREPAN generates the artificial data with respect to the marginal distribution of each attribute that is modelled from the neural network training data. Although TREPAN is able to maintain the accuracy of the neural networks in a single decision tree, its data modelling approach can become complicated when applied to a set of data that are distributed and independent from one another.

Another related work with respect to model representation is Breiman & Shang's *born again tree* [13]. Breiman & Shang study the approach of extracting

a tree from a set of arced trees[1] in such a way that the the accuracy of the arced trees can be maintained within a single tree. Breiman & Shang use the *100*palt% smearing* technique to manufacture data that are used in the extraction process. In the *100*palt% smearing* technique, an example is randomly selected from the training data, and its attribute values are randomly changed to one of the existing values of the corresponding attributes, if a random number chosen during the process is less than the value of *palt*. Thus, the data used to learn the *born again* tree are not explicitly modelled from the training set as those used in TREPAN algorithm. It takes the mixed approach which preserves some of the correlations between attributes in the training set and, at the same time, allows the attribute value to be randomly changed to the value that exists in the training data. The level of the mixture between the preservation and the re-assignment of the attribute values depends on the value of *palt*. Although it is not clear what is the optimal *palt* value, Breiman & Shang [13] observe that using *palt* equal to 0.5 appears to work well in most of the data sets used in their empirical study.

Concerning the criteria of the effectiveness of the approach, Breiman & Shang state that the degree of the improvement by the *born again* tree over the tree learnt by the centralised learning approach, is data set dependent. They conjecture that the source of the problem is the number of classes in the problem domains used in their study. That is, the improvement is relatively small in the data set whose number of classes is relatively large (6 - 19 classes). The results of our empirical studies in Chapter 5, show that the number of classes is insufficient to indicate the improvement in this type of system. The improvement should depend on the density of the training data[2] in each problem domain of study. This density depends on many factors, such as number and type of attributes, number of classes, the amount of the available data and their distributions, and the inductive biases of the learning algorithm. Although this density cannot be

---

[1] A set of arced trees is a set of trees that are sequentially constructed. The construction of the $(l+1)$st tree depends on the performance of the l previously constructed trees [12]. An example of a set of arced trees is a set of trees generated in Freund & Shapire's boosting algorithm [41, 42].

[2] With respect to all combinations of attribute and class values.

calculated exactly, it can be indirectly represented by the *Vapnik-Chervónenkis* dimension (VC-dimension) [10]. However, the VC-dimension of some representation forms, such as decision trees, is difficult to derive. In DLKP, we use a margin distribution graph to indirectly observe the density of distributed data via the analysis of the coverage of the model committee.

Another related work, with respect to model representation, is Domingos' CMM algorithm [34]. The CMM algorithm is aimed at extracting a representation form from a set of multiple models that are generated from the same data set. CMM integrates the set of base models, by learning from the original data set, plus a set of artificially generated data whose class values are assigned by the combined predictions of the base models. The artificial data used in the extraction process of CMM are generated in such a way that their distributions are as close to the "true" distributions of the data set as possible. Domingos estimates the probability of finding an example in the region that is covered by each classification, with the number of training examples that are classified in that class. Given that n base models represent n approximations of the "true" distributions with equal weight and that m artificial examples are required, CMM generates m/n of these examples from each base model. For each classification of the model, if it covers r of s training examples, $(r/s) \cdot (m/n)$ of this type of examples are generated. These artificially generated examples represent the approximation of the instance space via the set of base models. However, CMM sometimes suffers from unnecessary large output size as a result of using both the original training data and the artificially generated data in the extraction process [34].

8.3.1 Comparison to DLKP

DLKP and all the three works discussed in this section, share a similarity in that they try to extract a representation form from a black box model. A set of decision trees or decision rules is used as the black box model in the empirical studies of all works, except Craven's TREPAN algorithm [27, 29] where the black

box model used is neural networks. These two types of black box models are chosen on the grounds that they are more accurate than the model learnt by the centralised learning approach, however they lack comprehensibility. In DLKP, a model committee is used as the black box model, because of the requirement of an integrated model in distributed data mining.

DLKP and these works mainly differ in the source of the data that is used in the extraction process. In Craven's [27] and Domingos' [34] works, the data used consists of the original training set, plus some artificially generated data. Craven [27] addresses the disadvantage of artificially generated examples by means of modelling the underlying data distribution, that those examples sometimes cannot be answered by the black box, because they may not exist in the real problem domain. DLKP avoids this problem by using the existing training data in the extraction process. Although not as good as Craven's and Domingos' approaches where the extraction can be controlled more systematically, to select a probing set from a relatively large data set, which is the approach taken in DLKP, should sufficiently represent the distributions of data to some extent.

Note that in Domingos' CMM algorithm, only the artificial data are involved in the assignment of the class values from the black box. Thus, CMM can be seen as a data integration technique that assigns different weights to each area in the instance space by means of sampling or generating different numbers of instances from different area. In DLKP, only some part of the training data is used as a probing set, all of which involve the relabelling process. Thus, DLKP can be seen as an integration of the predictive ability of the models in the committee, rather than as an integration of data. Breiman & Shang's *born again* tree [11] shares a similarity with DLKP in this aspect. The original training set is not included in the extraction of the *born again* tree, however some parts of the data used still consist of manipulated data.

Other concerns of the works in this group, are the issues of distributed data mining, which include: different distributions of data in each local data set and

the communication overhead of the extraction process. Not designed for distributed data mining, none of the works in this group address these issues. As a result, some parts of the designs, such as the use of the complete training set in the extraction process, are not conveniently applied in distributed environments. This is because it is, at least, equivalent to gathering distributed data into one piece to perform a learning at one single site, which opposes the essence of distributed computing. Also, because these works do not address the possibility of uneven distributions among local data, the quality of the black box is not their issue of concerns. As a result, the black box used in these works is not adjusted. Without adjusting the black box, theoretically the accuracy of the resulted model should be upper bounded to the accuracy of the black box itself[3]. In contrast, the quality of the black box is the issue of concerns in DLKP. In the scenario where the black box, i.e. the model committee, is not suitable to be integrated, DLKP employs the *data redistribution* technique and the *data contraction* technique to adjust the distributions of the local data. Our experimental results show that, in some cases, these adjustments lead to a model that is more accurate than the model committee.

8.4 Related works in the aspect of active learning

By aiming to reduce the cost of exploring unuseful instance space[4], many researches [5, 7, 87] have been carried out in the area of active learning via selective sampling or active learning via queries from examples. Most of these pieces of research address the advantage of selectively sampling data from the area that maximises some criteria of interest. Thrun & Möller [87] employ the heuristic

[3]The empirical results from [34], [29] and [27] have also justified this observation. However, the additional artificially generated data can increase the possibility that the resulted model would perform better than the black box if the data are properly generated in sufficiently large quantities.

[4]It can be the space that has already been explored or the space that is not realisable for learning.

that drives the search to less familiar regions. They show the benefit of this exploration technique on a robot navigation task. Freund et al. [43] design a query algorithm that maximises information gain. They show that if the committee of two members[5] achieves information gain with positive lower bound in each of its queries, the classification error decreases exponentially with the number of queries. Krogh & Vedelsby [61] experiment on active learning with neural network ensemble. They show that the learning curve for a simple test problem can be improved by including an instance that the ensemble highly disagrees within its training set. Cohn et al. [24] study the effect of active learning in improving the generalisation performance of neural networks in *SG-network*. Apart from the fact that *SG-network* does not deal with multiple models and that the techniques they employ as the two main mechanisms of active learning[6] are different from DLKP, both systems share some fundamental concepts of active learning.

8.4.1 Comparison to DLKP

SG-network and DLKP differ in the schemes that are used to impose an order on different regions in the instance space and in the techniques for approximating an area of uncertainty. *SG-network* expresses the order of region, by imposing a *general-specific* concept on all concepts via the use of concept membership. By maintaining the *most general* and the *most specific* concepts that are consistent with all training examples, *SG-network* uses the gap between the coverage of these two concepts to approximate the area of uncertainty. This area covers instances in which the two concepts disagree in their class predictions. *SG-network* is aimed at narrowing this gap by modifying the most general and the most specific concepts. The maintenance of a *general-specific* concept, which is conveniently applied to single model learning, can become complicated when applied to multiple model environments. In DLKP, the *margin* of a prediction is used to impose an order to regions in the instance space with respect to the current model committee,

[5]Two members with respect to two hypotheses that are consistent with the labelled examples seen so far.

[6]The details of the two main mechanisms of active learning were given in Section 2.6.

because the margin can be conveniently derived from the prediction of the model committee[7]. The values 1 to −1 of margin represents the highest and lowest confidence of the model committee in predicting the class value of an instance. Thus, the area of uncertainty in DLKP is represented by the area in the instance space where the margin of the class prediction of the instances is between −0.2 and 0.2.

Despite different in the mechanisms used, SG-network and DLKP share an interesting similarity in the way they respond to the region of uncertainty. In *SG-network*, if an instance in this region is a positive example (i.e. an example that is located in the target area of the generalisation), some concepts in the *most specific* group have to be generalised to cover this instance. On the other hand, if the instance is a negative example (i.e. an example that is located outside the target area of the generalisation), some concepts in the *most general* group have to be modified to exclude it. These two reactions respectively correspond to the *data redistribution* technique and the *data contraction* technique in DLKP. The *data redistribution* technique is aimed at expanding the coverage of the model committee to a necessary area, whilst the *data contraction* technique is aimed at contracting the coverage of the model committee from an unnecessary area.

8.5 Summary

In this chapter, we have discussed related works of DLKP in three aspects: the aspect of learning from data partition, the aspect of extracting the representation of a model, and the aspect of active learning. We compare and contrast these works with DLKP and show that none of them are perfectly suitable for model integration in distributed data mining. With respect to learning from data partitions, most of the related works are shown to be algorithm dependent. Although Chan & Stolfo work [18, 16, 21, 19] is more generic in this aspect, the results of their meta-model approach lacks comprehensibility. This two limitations can

[7]The details of how to derive a margin from the prediction of a model committee were given in Section 2.5.

be solved by introducing the representation extraction technique, which is the technique employed in the group of works involving the extraction of the representation form of a model. That is, an integrated model can be learnt from the data whose class values are assigned by the black box used. In this way, the integration is achieved through the *data* rather than the *learning algorithm.* Therefore, the integration process can be more generic with respect to the learning algorithm. However, since none of the works in the second group are designed for distributed learning, they do not concern with the characteristics of distributed learnt models. As a result, the criteria of efficient integration and approaches to improve the quality of the black box are not addressed in these works. These criteria are addressed in the design of DLKP. In the scenario where local models are not suitable to be integrated, DLKP employs the active learning concept, which is the technique employed in the third group of related works, to adjust the model committee.

The next chapter contains a summary of this thesis, in which the objectives of the study as addressed in Chapter 1, are reviewed. Some directions of further research on DLKP are also discussed

Chapter 9

Summary, Conclusions and Further Work

In this chapter, the contributions of this thesis are summarised. The objectives set out at the start of the thesis are reviewed. Some directions for further research are discussed.

9.1 Contributions

The original contributions of this thesis can be listed as follows:

- This thesis addresses characteristics of local models in distributed data mining that are different from the characteristics of base models assumed in most of the model integration research. The local models in distributed environments are learnt independently from the existing data sets where the data distributions of those data sets can be of any shape.

- As a result of the previous observation, we find that model integration in distributed data mining does not always yield a model with satisfying accuracy. Also, it can be viewed that there is no universal model integration procedure for every set of local models. A procedure that is superior[1] to the

[1] With respect to the accuracy of the result.

others in one problem domain can be inferior in other problem domains. Thus, we propose DLKP, and its associated tools and techniques as an adaptive framework for model integration in distributed data mining. We show that the effectiveness of DLKP depends on the coverage of the model committee of interest, which can be generally categorised into four main scenarios. In each scenario, the appropriate model integration approach with DLKP is proposed and empirically justified.

- We propose the interpretation of the current form of the coverage of the model committee from the shape of its margin distribution graph. This interpretation enables DLKP to provide effective model integration approaches for different forms of the coverage of the model committee.

- We use the margin to locate the position of an instance in the coverage of the model. We proposes the use of this information to adjust local data distributions in the scenarios where the coverage of the model committee is not suitable for model integration. We also proposes the novel *data redistribution* and the *data contraction* as techniques for such a purpose.

9.2 Conclusions

The main interest of this thesis is to integrate distributed learnt models into one single model that combines the various advantages of local model and, at the same time, provides the insight into the problem domain. However, uncertainties involved in machine learning and data mining, such as number and type of attributes, amount and quality of the available data, and the difficulty of the problem domain itself, make it almost implausible to develop a universal model integration approach that is effective in all problem domains and in any forms of distributions of local data. This thesis deals with these difficulties in integrating distributed learnt models. Our study reveals that the margin distribution graph can be effectively used as an indicator for different forms of the coverage of the model committees, which is the crucial information for effective model integra-

tion. With the use of the margin distribution graph, *Distributed Learning with Knowledge Probing* (DLKP) is proposed as a model integration framework that is adaptive to different forms of the coverage of model committees in distributed data mining. In the scenarios where the coverage of the model committee is not suitable to be integrated, two data distribution adjusting techniques are proposed as additional mechanisms to adapt the local models to the state where they are suitable for the integration.

To access whether the design of DLKP has achieved the objectives set out in Section 1.3.3, we reprise the objectives of this thesis as follows:

- This thesis focuses on knowledge integration in the environment where data are distributed by nature, i.e. where data are already distributed and prior knowledge of data distributions at each local site is unavailable. We aim at developing a framework of model integration for different types of local models in distributed data mining.
- The integration is aimed at generating a single model that is represented in an understandable form and can be a representative of all the local models.
- The integrated model should, at least, have the same accuracy as the average accuracy of the local models.

The objectives of the research defined in Section 1.3.3 have been addressed and achieved in the design of DLKP. In DLKP, prior knowledge of data distributions is not required. By using the margin distribution graph as the indicator, this thesis has demonstrated that there are four main scenarios of the coverage of a model committee. The appropriate integration approach for each scenario is proposed accordingly. The analysis in Chapter 5 has shown that, in all four scenarios, the result of DLKP is in an understandable form and its accuracy is, at least, equal to the average accuracy of the local models. The case study in Chapter 6 shows that DLKP is effective on real distributed data and can be a generic framework for different classification learning algorithms. The implemen-

tation study in Chapter 7 shows that it is feasible to implement DLKP on a practical distributed data mining system.

Compared to the centralised learning approach, the accuracy of the final model of DLKP is comparable to that of the centralised learning approach. According to the design choices used in the empirical study, it means that using roughly 1/4 of the training data, DLKP can yield a model of comparable quality to the model learnt from the complete data set.

9.3 Further works

The followings are some possible directions for extending DLKP either to improve its efficiency or to enable it to solve broader types of problems.

Optimal size of the probing set and an approach to select probing data The problem of using optimal training set size is a common problem in machine learning, data mining and related research areas. Haussler [52] studies the approach to derive the minimum number of data required to achieve a desired accuracy under the Valiant's probably approximately correct (PAC) framework [88]. The key component of his approach is the *Vapnik-Chervónenkis* dimension (VC-dimension). The VC-dimension does not measure the complexity of the hypothesis space H by the number of distinct hypothesis $|H|$, but by the number of distinct instances that can be completely discriminated using H. Therefore, the VC-dimension depends on the expressive power of H. However, it is often the case that the expressive power of H is weaker when the hypothesis space is formed by some representation languages. Thus, the problem is reduced to how to calculate VC-dimension for each representation form. This is also the case for decision tree where it is not clear how to derive its VC-dimension.

If the minimum size of training data can be derived, it can be used to decide the size of the probing set. With a proper probing set size, DLKP is expected to be able to exploit the benefit of a model committee more extensively. In addition, the probing set selection method may be improved by imposing an

iteration process to enable the selection of instances from some specific regions that have not been covered by a current probed model.

Integration of models that are learned from biased data In practical applications, distributed data that are physically similar, i.e. sharing the same schema, can be different in their semantics. For example, in a set of clinical data that are distributed by locations of hospitals, it can be the case that some hospitals are specialist hospitals, for example a hospital for only heart disease or male patients, whereas the other hospitals in the group are more general in the aspect of open clinic and group of patients [66]. This raises the question of the semantics and the applicability of the integrated model. A careful design of the level of contribution from each local data, for example in form of weight, to the integration process may be required to solve this problem.

Integration of models that are learned from data of different schemas This thesis focuses on the integration of distributed data that are assumed to have the same schema. In practical applications, each distributed data may be applied under different schemas. To apply DLKP to such a problem needs a careful design of the correlation between attributes from different schemas and a mechanism for conveying such a correlation. Such a scenario is not covered in this thesis, but further study in this direction will certainly extend the applicability of DLKP in practical applications. An example of this type of study can be found in [83].

Management of outliers in the data contraction technique In the data contraction technique, the problematic data are removed from each local training data to improve the generalisation of the local model. This process, however, also introduces a possibility of removing the irregular data or the outliers that contain useful knowledge of the problem domain from the training data. An extension of the data contraction technique to differentiating the beneficial outliers from the problematic ones will certainly improve the efficiency of the framework. An

example of an approach to reasoning about outliers by using domain knowledge to model noisy data can be found in [64, 92].

Use of different inductive biases between the two learning phases of DLKP Further study on DLKP can be conducted on applying different inductive biases between the two learning phases. Intuitively, such a treatment may lead to the problem of a mismatch of biases, which is also addressed in [34]. However, it is still interesting to observe the reaction of different biases, which will reflect in the probed model. The study on bias matching and compensation of different biases in DLKP can be another direction of further study.

Application of DLKP **for a scalable data mining problem** The application of DLKP can also be extended to solve the problem of scalable data mining. Although most modern operating systems support virtual memory when the size of problem exceeds the size of a memory of a single processor, the out-of-core implementation of some learning algorithms such as decision tree learning algorithms (e.g. SPRINT system [84]) may require large amount of disk-resident data, which can lead to inhibitive amount of I/O overhead. Also, although this problem can be solved with sampling techniques, the efficient and reasonable-cost sampling technique is required. DLKP can be a solution for this type of problem. Although DLKP may not guarantee a speedup over the centralised learning approach, at least, it provides a straightforward way to make such a learning process viable under limited memory. Further experimental study is required to justify this application.

9.4 Epilogue

Since distributed data has the special characteristic of being able to be distributed in any form, there are several issues of concerns in the design of an approach for model integration in distributed data mining. Firstly, owing to the fact that there is no single best approach for all forms of data distributions [91], such an integra-

tion approach should be adaptive or consist of alternatives for different scenarios of data distributions. Secondly, the integration should provide a comprehensible output model so that the user can gain insight in the problem domain and be confident in the actions suggested by the model. Finally, because locations of the distributed sites and the distance between them are one of the major concerns in distributed computing, such an integration approach should be implemented in a reasonable computation and communication costs. This thesis has shown that these properties can be achieved and DLKP has provided an approach to achieve them to a significant extent.

Appendix A

Glossary of Terms

Since data mining is an interdisciplinary research area, technical terms used in publications are different. The technical terms used in this thesis are herein summarised in six categories. An asterisk indicates that the definition is the same as that given by Kohavi & Provost [56].

A.1 Data descriptions

Attribute* A quantity describing an instance. The attribute type, which denotes the values that can be taken by an attribute, defines the attribute domain. Two common types of the attribute domain are:

- *Categorical* A finite number of discrete values. The type *nominal* denotes that there is no order between values in the domain, such as a domain of last names. The type *ordinal* denotes that there is an order between values in the domain, such as a domain of the values low, medium, high.
- *Continuous* A subset of real numbers where there is a measurable difference between the possible values. In practical problems, integers are usually classified in a continuous domain.

Missing value* The value for an attribute is not known or does not exist.

Schema* A description of attributes of a data set and their properties.

Instance / example* A single object of the domain from which a model will be learnt, or on which a model will be used (e.g. for prediction).

Data set* A schema and a set of instances matching the schema. Generally, no order of instances is assumed.

i.i.d. sample* A set of independent and identically distributed instances.

A.2 Learning related terms

Supervised learning* Techniques used to learn the relationship between independent attributes and a designated dependent attribute (a class value).

Unsupervised learning* Learning techniques that group instances without a pre-specified dependent attribute.

Learning algorithm* An algorithm that takes instances and produces a model that generalises beyond these instances.

Training set A data set that is used as an input of a learning algorithm in a learning process.

Model* A structure and corresponding interpretation that summarises or partially summarises a set of data for description or prediction. Most inductive algorithms generate models as output.

Test set A data set that is used to evaluate the performance of a model. It is usually an independent set that has not been used during the learning process.

Black box model A model that is represented in a form that is difficult for human users to comprehend e.g. a neural network model.

Classifier A model which is a result of supervised learning. Given un-labelled instances, classifier gives their most probable classes. A classifier consists of a representation form, i.e. a decision tree, and an interpretation procedure, i.e. how to derive a class value of a given instance from the decision tree.

Prediction / Class prediction An output of a classifier given an unlabelled instance.

Instance space A search space determined by a set of attributes in a data set.

Hypothesis space All possible search space that is determined by the type of expression allowed in specifying an output of a learning algorithm.

A.3 Model properties

Accuracy / Error rate* The rate of correct / incorrect predictions made by a model over a set of data. Accuracy is usually estimated by using an independent test set, not used during the learning process.

Coverage The area in the instance space where a classifier makes a prediction. It is interesting to know the accuracy of the classifier in the area under its coverage. This information can lead to a conclusion of whether the classifier correctly covers the instance space.

A.4 Learning approaches

Black box learning approach The learning approach that develops its own representation for approximating the target function. However, its internal representation is difficult for human users to interpret, so it does not provide insight into the problem domain. The black box learning approach typically involves numeric calculations of coefficients, distances or weights.

Centralised learning approach A learning process that is performed on a complete set of data and gives a single model as an output.

A.5 Experimental methodology

Cross-validation* A method for estimating the accuracy (or error) of a learning algorithm on a data set. It is achieved by dividing the data set into l mutually exclusive subsets of approximately equal size. The learning algorithm is trained and tested l times. Each time it is trained on $l-1$ subsets and then tested on the remaining subset of data. Each fold is in turn used as a test set. Accuracy is estimated by averaging the accuracy of l models.

A.6 DLKP related terms

Base model A model in multiple model learning environments.

Local model A model in distributed learning environments.

Prediction scheme A scheme that is used to combine predictions from local models.

Combined prediction A prediction combined by a prediction scheme, given a set of predictions from local models.

Model committee A set of local models and its prediction scheme.

Probing set A data set used to probe a descriptive model from a black box model, such as a model committee.

Probing instance An instance in a probing set.

Probing algorithm An algorithm that is used in the Knowledge Probing phase of DLKP. It is required to be the algorithm that provides a descriptive output.

Relabelling process The process that occurs when there is a probing instance having a negative margin. A probing instance will have a negative margin only when the prediction of the model committee on that instance conflicts with its original class value.

Percentage of relabelling The number of times that the relabelling process happens when a particular data set is used as a probing set.

Disagreement level The number of times that the prediction from each local model conflicts with the corresponding combined prediction on a particular data set. It is averaged over the number of local models.

Appendix B

Descriptions of data sets

The descriptions of the data sets that were used in this thesis are included in this appendix for reference. These descriptions and the corresponding data sets were obtained from the UCI Repository of machine learning [8]. Each of them includes sources of data, past usage, relevant information, number of instances, number of attributes, percentage of missing attribute values, and number of classes.

B.1 Abalone data set

Title: Abalone data

Sources:

1. Original owners of database: Marine Resources Division, Marine Research Laboratories - Taroona, Department of Primary Industry and Fisheries, Tasmania, GPO Box 619F, Hobart, Tasmania 7001, Australia (contact: Warwick Nash +61 02 277277, wnash@dpi.tas.gov.au)

2. Donor: Sam Waugh (Sam.Waugh@cs.utas.edu.au) Department of Computer Science, University of Tasmania, GPO Box 252C, Hobart, Tasmania 7001, Australia

3. Date received: December 1995

Past Usage:

1. S. Waugh. *Extending and Benchmarking Cascade-Correlation.* PhD thesis, Computer Science Department, University of Tasmania, 1995.

2. D. Clark, Z. Schreter, and A. Adams. A Quantitative Comparison of Dystal and Back-propagation. Submitted to *the Australian Conference on Neural Networks* (ACNN), 1996.

Relevant Information: The data set involves the prediction of the age of abalone from its physical measurements. The age of abalone is usually determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope, which is a time-consuming task. Thus, other measurements, which are easier to obtain, are used to predict the age.

From the original data set, examples with missing values were removed and the ranges of the continuous values have been scaled by dividing by 200. The original data can be found in

W. J. Nash, T. L. Sellers, S. R. Talbot, A. J. Cawthorn, and W. B Ford. The Population Biology of Abalone (Haliotis species) in Tasmania. I. Blacklip Abalone (H. rubra) from the North Coast and Islands of Bass Strait. Technical Report No. 48 (ISSN 1034-3288), Sea Fisheries Division, Tasmania, 1994.

Number of Instances: 4,177

Number of Attributes: 8 (1 categorical, 7 numerical)

Missing Attribute Values: None

Number of Classes: 29

B.2 GERMAN data set

Title: German Credit data

Source:

1. Original owners of database: Professor Hans Hofmann, Institut für Statistik und Ökonometrie, Universität Hamburg, FB Wirtschaftswissenschaften, Von-Melle-Park 5, 2000 Hamburg 13

2. Donor: -

3. Date received: -

Past Usage: -

Relevant Information: The original dataset, in the form provided by Professor Hofmann, contains categorical attributes. For the form that was used in the StatLog project, Strathclyde University has edited and added several indicator variables to make the data set suitable for algorithms that cannot cope with categorical variables. Several attributes that are ordered categorical have been coded as integer.

Number of Instances: 1,000

Number of Attributes: 20 (13 categorical, 7 numerical)

Missing Attribute Values: None

Number of Classes: 2

B.3 HYPOTHYROID data set

Title: Hypothyroid

Sources: -

Past Usage: -

Relevant Information: -

Number of Instances: 3,163

Number of Attributes: 25 (18 categorical, 7 numerical)

Missing Attribute Values: 6.74%

Number of Classes: 2

B.4 KR-VS-KP data set

Title: Chess End-Game

Sources:

1. Original owners of database: Alen Shapiro
2. Donor: Rob Holte (holte@uottawa.bitnet). The database was supplied to Holte by Peter Clark of the Turing Institute in Glasgow (pete@turing.ac.uk).
3. Date received: August 1989

Past Usage:

1. A. D. Shapiro. *Structured Induction in Expert Systems.* Addison-Wesley, 1987. This book is based on A. D. Shapiro. *The Role of Structured Induction in Expert Systems.* PhD thesis, University of Edinburgh, 1983.
2. S. Muggleton. Structuring Knowledge by Asking Questions. In I. Bratko and N. Lavrac, editors, *Progress in Machine Learning*, pages 218-229, Sigma Press, Wilmslow, England SK9 5BB, 1987.
3. R. C. Holte, L. Acker, and B. W. Porter. Concept Learning and the Problem of Small Disjuncts. In *Proceedings of International Joint Conferences in Artificial Intelligence*, 1989.

Relevant Information: The format of instances in this database is a sequence of 36 attributes and one class value. Each instance is a board-description for the chess end-game. The attributes describe the board. The class value is the classification: "win" or "no win".

Number of Instances: 3,196

Number of Attributes: 36 (all categorical)

Missing Attributes Values: None

Number of Classes: 2

B.5 LETTER data set

Title: Letter Image Recognition Data

Source:

1. Original owners of database: David J. Slate, Odesta Corporation, 1890 Maple Ave, Suite 115, Evanston, IL 60201

2. Donor: David J. Slate (dave@math.nwu.edu), (708) 491-3867

3. Date received: January, 1991

Past Usage: P. W. Frey and D. J. Slate. Letter Recognition Using Holland-style Adaptive Classifiers. *Machine Learning*, 6(2), March, 1991.

This article investigated the ability of several variations of Holland-style adaptive classifier systems to learn to correctly guess the letter categories associated with vectors of 16 integer attributes extracted from raster scan images of the letters.

Relevant Information: The data set involves identifying each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 to 15.

Number of Instances: 20,000

Number of Attributes: 16 (all numerical)

Missing Attribute Values: None

Number of Classes: 26

B.6 NURSERY data set

Title: Nursery Database

Sources:

1. Original owners of database: Vladislav Rajkovic et al. (13 experts)
2. Donors: Marko Bohanec (marko.bohanec@ijs.si) and Blaz Zupan (blaz.zupan@ijs.si)
3. Date: June, 1997

Past Usage:

1. M. Olave, V. Rajkovic, and M. Bohanec. An Application for Admission in Public School Systems. In I. Th. M. Snellen, W. B. H. J. van de Donk, and J.-P. Baquiast, editors, *Expert Systems in Public Administration*, pages 145-160. Elsevier Science Publishers (North Holland), 1989.

2. B. Zupan, M. Bohanec, I. Bratko, and J. Demsar. Machine Learning by Function Decomposition. *International Conference on Machine Learning*, Nashville, TN, 1997.

Relevant Information: Nursery Database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It was used during several years in 1980's when there was excessive enrolment to these schools in Ljubljana, Slovenia, and the rejected applications frequently needed an objective explanation. The final decision depended on three subproblems: occupation of parents and child's nursery, family structure and financial standing, and social and health picture of the family. The model was developed within expert system shell for decision making DEX (M. Bohanec and V. Rajkovic. Expert System for Decision Making. *Sistemica*, 1(1):145-157, 1990.).

Number of Instances: 12,960 (instances completely cover the attribute space)

Number of Attributes: 8 (all categorical)

Missing Attribute Values: None

Number of Classes: 5

B.7 SATELLITE data set

Title: Satellite data

Sources:

1. Original owners of database: The original Landsat data for this database was generated from data purchased from NASA by the Australian Centre for Remote Sensing, and used for research at: The Centre for Remote Sensing, University of New South Wales, Kensington, PO Box 1, NSW 2033 Australia.

2. Donor: The small sample database was provided by Ashwin Srinivasan (ross@uk.ac.turing), Department of Statistics and Modelling Science, University of Strathclyde, Glasgow, Scotland UK.

3. Date received: -

The sample database was generated taking a small section (82 rows and 100 columns) from the original data. The binary values were converted to their present ASCII form by Ashwin Srinivasan. The classification for each pixel was performed on the basis of an actual site visit by Ms. Karen Hall, when working for Professor John A. Richards, at the Centre for Remote Sensing at the University of New South Wales, Australia. Conversion to 3 x 3 neighbourhoods and splitting into test and training sets was done by Alistair Sutherland.

Past Usage: -

Relevant Information: The Landsat satellite data is one of the many sources of information available for a scene. The interpretation of a scene by integrating spatial data of diverse types and resolutions including multispectral and radar data, maps indicating topography, land use etc. is expected to assume significant importance with the onset of an era characterised by integrative approaches to remote sensing.

One frame of Landsat MSS imagery consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and two are in the (near) infra-red. Each pixel is a 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about 80m x 80m. Each image contains 2340 x 3380 such pixels.

The database is a (tiny) sub-area of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3 x 3 square neighbourhood of pixels completely contained within the 82 x 100 sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the 9 pixels in the 3 x 3 neighbourhood and a number indicating the classification label of the central

pixel. The data is given in random order and certain lines of data have been removed, so the original image cannot be reconstructed from this dataset.

In each line of data the four spectral values for the top-left pixel are given first followed by the four spectral values for the top-middle pixel and then those for the top-right pixel, and so on with the pixels read out in sequence left-to-right and top-to-bottom.

Number of Instances: 6,435

Number of Attributes: 36 (all numerical)

Missing Attribute Values: None

Number of Classes: 7

B.8 SEGMENT data set

Title: Image Segmentation data

Source:

1. Original owners of database: Vision Group, University of Massachusetts
2. Donor: Vision Group (Carla Brodley, brodley@cs.umass.edu)
3. Date received: November, 1990

Past Usage: -

Relevant Information: The instances were drawn randomly from a database of seven outdoor images. The images were hand-segmented to create a classification for every pixel. Each instance is a 3 x 3 region.

Number of Instances: 2,310

Number of Attributes: 19 (all numerical)

Missing Attribute Values: None

Number of Classes: 7

B.9 SICK data set

Title: Sick-euthyroid data

Sources:

1. Original owners of database: The Garavan Institute and J. Ross Quinlan, New South Wales Institute, Sydney, Australia.

2. Donor: -

3. Date received: 1987

Past Usage: -

Relevant Information: -

Number of Instances: 3,162

Number of Attributes: 25 (18 categorical, 7 numerical)

Missing Attribute Values: 6.74%

Number of Classes: 2

B.10 THYROID data set

Title: Hypothyroid

Sources:

1. Original owners of database: The Garavan Institute and J. Ross Quinlan, New South Wales Institute, Sydney, Australia.

2. Donor: -

3. Date received: 1987

Past Usage: -

Relevant Information: -

Number of Instances: 3,772

Number of Attributes: 29 (22 categorical, 7 numerical)

Missing Attribute Values: 5.54%

Number of Classes: 5

B.11 LED7 data set

Title: LED display domain

Sources:

1. Original owners of database: L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, Belmont, California, 1984.

2. Donor: David Aha

3. Date received: October 1988

Past Usage:

1. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, Belmont, California, 1984.

2. J. R. Quinlan. Simplifying Decision Trees. *International Journal of Man-Machine Studies*, 1987.

3. M. Tan, and L. Eshelman. Using Weighted Networks to Represent Classification Knowledge in Noisy Domains. In *Proceedings of the 5th International Conference on Machine Learning*, pages 121-134, Morgan Kaufmann, 1988.

Relevant Information: This simple domain contains seven Boolean attributes which represent seven light-emitting diodes on the LED display, and ten classes which constitute the set of decimal digits. All attribute values are either 0 or 1, according to whether the corresponding light is on or not for the decimal digit.

Number of Instances: 2,000 (This can be specified by user.)

Number of Attributes: 7 (all categorical)

Missing Attribute Values: None

Number of Classes: 10

B.12 WAVEFORM data set

Title: Waveform Database Generator (written in C)

Sources:

1. Original owners of database: L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, Belmont, California, 1984.

2. Donor: David Aha

3. Date received: October 1988

Past Usage: L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, Belmont, California, 1984.

Relevant Information: -

Number of Instances: 5,000 (This can be specified by user.)

Number of Attributes: 40 (all numerical)

Missing Attribute Values: None

Number of Classes: 3

B.13 HEART data set

Title: Heart Disease Databases

Sources:

1. Original owners of database:

   (a) Andras Janosi, M.D., Hungarian Institute of Cardiology, Budapest

   (b) William Steinbrunn, M.D., University Hospital, Zurich, Switzerland

   (c) Matthias Pfisterer, M.D., University Hospital, Basel, Switzerland

   (d) Robert Detrano, M.D., Ph.D., V.A. Medical Center, Long Beach and Cleveland Clinic Foundation

2. Donor: David W. Aha (aha@ics.uci.edu), (714) 856-8779

3. Date received: July 1988

Past Usage:

1. R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee, and V. Froelicher. International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease. *American Journal of Cardiology*, 64: 304-310, 1989.

2. D. W. Aha and D. Kibler, Instance-based prediction of heart-disease presence with the Cleveland database.

3. J. H. Gennari, P. Langley, and D. Fisher. Models of Incremental Concept Formation. *Artificial Intelligence*, 40:11–61, (1989).

Relevant Information: This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by machine learning researchers to this date. The class value refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish the presence of the disease (values 1,2,3,4) from the absence from the disease (value 0).

Number of Instances:

Database	Number of instances
CLEVELAND	303
HUNGARIAN	294
LONG BEACH VA	200
SWITZERLAND	123

Number of Attributes: 75 (Only 13 used, five of which are numerical.)

Missing Attribute Values: Several (Distinguished with value -9.0)

Number of Classes: 5

References

[1] K. Ali and M. Pazzani. HYDRA-MM: Learning multiple descriptions to improve classification accuracy. *International Journal on Artificial Intelligence Tools*, 4(1):115–133, January 1995.

[2] K. Ali and M. Pazzani. Classification using Bayes averaging of multiple relational rule-based models. In D. Fisher and H. Lenz, editors, *Learning from data: Artificial Intelligence and Statistics*, volume 5, pages 207–217. New York, NY: Springer-Verlag, 1996.

[3] K. M. Ali. *Learning Probabilistic Relationship Concept Descriptions.* Dissertation, Information and Computer Science, University of California at Irvine, California, 1996.

[4] K. M. Ali and M. J. Pazzani. Error reduction through learning multiple descriptions. *Machine Learning*, 24(3):173–202, September 1996.

[5] D. Angluin. Queries and concept learning. *Machine Learning*, 2(3):319–342, 1988.

[6] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36(1/2):105–139, July 1999.

[7] E. B. Baum and K. J. Lang. Constructing hidden units using examples and queries. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 904–910. Morgan Kaufmann Publishers, 1991.

[8] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[9] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

[10] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the *vapnik-chervónenkis* dimension. *Journal of the Association for Computing Machinery*, 36(4):929–965, October 1989.

[11] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.

[12] L. Breiman. [bias variance &] arcing classifiers. Technical Report 460, Department of Statistics, University of California at Berkeley, California, July 1996.

[13] L. Breiman and N. Shang. Born again trees. Technical report, Department of Statistics, University of California at Berkeley, California, 1996.

[14] C. E. Brodley and T. Lane. Creating and exploiting coverage and diversity. In *Proceedings of the AAAI-96 Workshop: Integrating Multiple Learned Models for Improving and Scaling Machine Learning*, pages 8–14, Portland, Oregon, 4-5 August 1996. AAAI Press.

[15] W. L. Buntine. *A Theory of Learning Classification Rules.* Dissertation, School of Computing Science, University of Technology, Sydney, Australia, 1992.

[16] P. Chan and S. Stolfo. Experiments on multistrategy learning by meta-learning. In *Proceeding of the Second International Conference on Information Knowledge Management*, pages 314–323, Washington, USA, 1-5 November 1993. ACM.

[17] P. Chan and S. Stolfo. Meta-learning for multistrategy and parallel learning. In *Proceeding of the Second International Workshop on Multistrategy Learning*, pages 150–165, Harpers Ferry, WV, 27-29 May 1993. Morgan Kaufmann.

[18] P. Chan and S. Stolfo. Toward parallel and distributed learning by meta-learning. In *Working Notes AAAI-93 Workshop Knowledge Discovery in Databases*, pages 227–240. AAAI, 1993.

[19] P. Chan and S. Stolfo. On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information System*, 8(1):5–28, January 1997.

[20] P. K. Chan and S. J. Stolfo. A comparative evaluation of voting and meta-learning on partitioned data. In *Proceedings of Twelfth International Conference on Machine Learning*, pages 90–98, Tahoe city, California, USA, 9-12 July 1995. Morgan Kaufmann.

[21] P. K. Chan and S. J. Stolfo. Sharing learned models among remote database partitions by local meta-learning. In E. Simoudis, J. Han, and U. Fayyad, editors, *The Second International Conference on Knowledge Discovery and Data Mining*, pages 2–7. AAAI Press, 1996.

[22] J. Chattratichat, J. Darlington, M. Ghanem, Y. Guo, H. Hüning, M. Köhler, J. Sutiwaraphun, H. W. To, and D. Yang. Large scale data mining: Challenges and responses. In D. Pregibon and R. Uthurusamy, editors, *Third International Conference on Knowledge Discovery and Data Mining, KDD-97*, pages 61–64, Newport Beach, CA, 1997. AAAI Press.

[23] J. Chattratichat, J. Darlington, Y. Guo, S. Hedvall, M. Köhler, and J. Syed. An architecture for distributed enterprise data mining. In *Proceedings of the seventh International Conference on High Performance Computing and Network Europe*, Amsterdam, the Netherlands, 12-14 April 1999.

[24] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[25] S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78, 1993.

[26] T. M. Cover and P. E. Hart. Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[27] M. W. Craven. *Extracting Comprehensible Models from Trained Neural Networks.* PhD thesis, Department of Computer Science, University of Wisconsin - Madison, 1996.

[28] M. W. Craven and J. W. Shavlik. Using sampling and queries to extract rules from trained neural network. In W. W. Cohen and H. Hirsh, editors, *Machine Learning: Proceedings of the Eleventh International Conference*, pages 37–45, New Brunswick, NJ, 1994. Morgan Kaufmann, San Francisco, CA.

[29] M. W. Craven and J. W. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems*, volume 8. MIT Press, Cambridge, MA, 1996.

[30] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64:304–310, 1989.

[31] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, August 2000.

[32] T. G. Dietterich and E. B. Kong. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical report, Department of Computer Science, Oregon State University, 303 Dearborn Hall, Corvallis, OR 97331-3202, 1995.

[33] P. Domingos. Using partitioning to speed up specific-to-general rule induction. In *Proceedings of the AAAI-96 Workshop: Integrating Multiple Learned Models for Improving and Scaling Machine Learning*, pages 29–34, Portland, Oregon, 4-5 August 1996. AAAI Press.

[34] P. Domingos. Knowledge acquisition from examples via multiple models. In

D. Fisher, editor, *Proceedings of the Fourteen International Conference on Machine Learning*, pages 98–106. Morgan Kaufmann, 1997.

[35] P. Domingos. Occam's two razors: The sharp and the blunt. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 37–43. AAAI Press, 1998.

[36] H. Drucker, C. Cortes, L. D. Jackel, Y. Lecun, and V. Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.

[37] R. Duda, J. Gaschnig, and P. Hart. Model design in the prospector consultant system for mineral exploration. In D. Michie, editor, *Expert Systems in the Micro-electronic Age*, pages 153–167. Edinburgh University Press, Edinburgh, England, 1979.

[38] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, New York, 1973.

[39] D. W. Fan, P. K. Chan, and S. J. Stolfo. A comparative evaluation of combiner and stacked generalization. In *Proceedings of the AAAI-96 Workshop: Integrating Multiple Learned Models for Improving and Scaling Machine Learning*, pages 40–46, Portland, Oregon, 4-5 August 1996. AAAI Press.

[40] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 1, pages 1–34. AAAI Press/The MIT Press, 1996.

[41] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In L. Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, Bari, Italy, 1996. Morgan Kaufmann.

[42] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[43] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Information, prediction, and query by committee. In S. H. et al., editor, *Advances in Neural Information Processing Systems*, volume 5, pages 483–490. Morgan Kaufmann, 1993.

[44] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[45] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40(1-3):11–61, September 1989.

[46] I. J. Good. The estimation of probabilities: An essay on modern bayesian methods. Research Monograph No. 30. MIT Press, Cambridge, Massachusetts, 1965.

[47] Y. Guo, S. Rüeger, J. Sutiwaraphun, and J. Forbes-Millott. Meta-learning for parallel data mining. In *Proceedings of the Seventh Parallel Computing Workshop*, pages P2I1–P2I11, The Australian National University, Canberra, Australia, 25-26 September 1997. Fujitsu Laboratories Ltd.

[48] Y. Guo and J. Sutiwaraphun. Knowledge probing in distributed data mining. In *Working Notes of the KDD-97 Workshop on Distributed Data Mining*, pages 61–69, New York City, 27-31 August 1998.

[49] Y. Guo and J. Sutiwaraphun. Distributed classification with knowledge probing: A new framework for distributed data mining. In H. Kargupta and P. Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*, chapter 4, pages 115–132. AAAI Press/The MIT Press, 2000.

[50] L. O. Hall, N. Chawla, and K. W. Bowyer. Combining decision trees learned in parallel. In *Working Notes of the KDD-98 Workshop on Distributed Data Mining*, pages 10–15, New York, New York, 31 August 1998.

[51] C. Harris-Jones and T. Haines. Sample size and misclassification: Is more always better ? Technical Report AMSCAT-WP-97-118, AMS Center for Advanced Technologies, December 1997.

[52] D. Haussler. Quantifying inductive bias: AI learning algorithms and valiant's learning framework. *Artificial Intelligence*, 36(2):177–221, September 1988.

[53] R. V. Hogg and E. A. Tanis. *Probability and Statistical Inference.* Prentice-Hall, Inc., Upper Saddle River, New Jersey, 5 edition, 1997.

[54] H. Kargupta, E. Johnson, E. R. Sanseverino, B.-H. Park, L. Silvestre, and D. Hershberger. Scalable data mining from distributed, vertically partitioned feature space using collective mining and gene expression based genetic algorithms. In H. Gargupta and P. Chan, editors, *KDD-98 Workshop on Distributed Data Mining in the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 70–91, New York, 31 August 1998.

[55] R. Kohavi. Technique selection in machine learning applications. Slides of invited talk at the ICML-98 Workshop on the Methodology of Applying Machine Learning, July 1998.

[56] R. Kohavi and F. Provost. Glossary of terms. *Machine Learning: The Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, 30(2/3):271–274, February/March 1998.

[57] R. Kohavi, D. Sommerfield, and J. Dougherty. Data mining using MLC++: A machine learning library in C++. In *Tools With Artificial Intelligence 1996*, pages 234–245. IEEE Computer Society Press, November 1996. `http://www.sgi.com/Technology/mlc`.

[58] R. Kohavi and D. Wolpert. Bias plus variance decomposition for zero-one loss functions. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 275–283. Morgan Kaufmann, 1996.

[59] E. B. Kong and T. G. Dietterich. Error-correcting output coding corrects bias and variance. In A. Prieditis and S. Russell, editors, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 313–321. Morgan Kaufmann, 1995.

[60] A. Krogh and P. Sollich. Statistical mechanics of ensemble learning. *Physical Review E*, 55:811–825, 1997.

[61] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 231–238. The MIT Press, 1995.

[62] S. Kwok and C. Carter. Multiple decision trees. In R. D. Shachter, T. S. Levitt, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, volume 4, pages 327–335. Elsevier Science, North Holland, 1990.

[63] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 223–228. AAAI Press and MIT Press, 1992.

[64] X. Liu, G. Cheng, and J. X. Wu. Noise and uncertainty management in intelligent data modelling. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 1, pages 263–268, Seattle, Washington, 31 July - 4 August 1994. The American Association for Artificial Intelligence, AAAI Press.

[65] D. Madigan, A. E. Raftery, C. T. Volinsky, and J. A. Hoeting. Bayesian model averaging. In *Proceedings of the AAAI-96 Workshop: Integrating Multiple Learned Models for Improving and Scaling Machine Learning*, pages 77–83, Portland, Oregon, 4-5 August 1996. AAAI Press.

[66] S. McClean. Personal communication. In PhD viva, April 2001.

[67] R. Meir. Bias, variance and the combination of least squares estimators. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 295–302. The MIT Press, 1995.

[68] R. Michalski, I. Bratko, and M. Kubat. *Machine Learning and Data Mining.* John Wiley & Son Ltd, 1998.

[69] R. S. Michalski. A theory and methodology of inductive learning. In J. Shavlik and T. G. Dietterich, editors, *Readings in Machine Learning*, Morgan Kaufmann series in machine learning, chapter 2, pages 70–95. Morgan Kaufmann, 1990.

[70] T. Mitchell. Generalization as search. In J. Shavlik and T. Dietterich, editors, *Readings in Machine Learning*, Morgan Kaufmann series in machine learning, chapter 2, pages 96–107. Morgan Kaufmann, 1990.

[71] T. M. Mitchell. The need for biases in learning generalizations. Technical Report CBM-TR-117, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1980.

[72] T. M. Mitchell. *Machine Learning.* Computer Science Series. McGraw-Hill, McGraw-Hill international edition, 1997.

[73] T. Oates and D. Jensen. The effects of training set size on decision tree complexity. In D. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 254–262. Morgan Kaufmann, 1997.

[74] T. Oates and D. Jensen. Large datasets leads to overly complex models: an explanation and a solution. In R. Agrawal and P. Stoiorz, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 294–298. AAAI Press, 1998.

[75] M. Plutowski. *Selecting Training Exemplars for Neural Network Learning.* PhD thesis, Computer Science and Engineering Department, University of California at San Diego, 1994.

[76] F. Provost and V. Kolluri. A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery*, 3(2):131–169, June 1999.

[77] F. J. Provost and D. N. Hennessy. Scaling up: Distributed machine learning with cooperation. In *Proceedings of the AAAI-96 Workshop: Integrating Multiple Learned Models for Improving and Scaling Machine Learning*, pages 107–112, Portland, Oregon, 4-5 August 1996. AAAI Press.

[78] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[79] J. R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[80] J. Rachlin, S. Kasif, S. Salzberg, and D. W. Aha. Towards a better understanding of memory-based reasoning systems. In *Machine Learning: Proceedings of the 11th International Conference*, pages 242–250, New Brunswick, NJ, 1994. Morgan Kaufmann.

[81] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[82] R. Schapire, Y. Freund, P. Bartlett, and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, October 1998.

[83] B. Scotney and S. McClean. Efficient knowledge discovery through the integration of heterogeneous data. *Information and Software Technology*, 41(9):569–578, 1999.

[84] J. Shafer, R. Agrawal, and M. Mehta. SPRINT: A scalable parallel classifier for data mining. In T. Vijayaraman, A. Buchmann, C. Mohan, and N. Sarda, editors, *Proceedings of the 22nd International Conference on Very Large Data Bases*, Mumbai (Bombay), India, September 1996. Morgan Kaufmann.

[85] P. Sollich and A. Krogh. Learning with ensembles: How over-fitting can be useful. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 190–196. The MIT Press, 1996.

[86] C. Stanfill and D. Waltz. Towards memory-based reasoning. *Communications of the Association for Computation Machinery*, 29(12):1213–1228, 1986.

[87] S. Thrun and K. Möller. Active exploration in dynamic environments. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 531–538. Morgan Kaufmann, 1992.

[88] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[89] G. Williams. *Inducting and Combining Multiple Decision Trees*. PhD thesis, Department of Computer Science, Australian National University, Canberra, Australia, 1990.

[90] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[91] D. H. Wolpert and W. G. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, The Santa Fe Institute, 1399 Hyde Park Road, February 1996.

[92] J. X. Wu, G. Cheng, and X. Liu. Reasoning about outliers by modelling noisy data. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis (IDA-97): Reasoning about Data*, Lecture Notes in Computer Science 1280, pages 549–558, London, UK, August 1997. Springer-Verlag: Berlin, Heidelberg.

[93] K. Yamanishi. Distributed cooperative bayesian learning strategies. In *Proceedings of the 1997 10th Annual Conference on Computational Learning Theory*, pages 250–262, Nashville, TN, July 1997. ACM, New York.