A PROBABILISTIC PERSPECTIVE ON ENSEMBLE DIVERSITY

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2010

By
Manuela Zanda
School of Computer Science

Contents

Word Count: 54,295

Abstract

We study diversity in classifier ensembles from a broader perspective than the 0/1 loss function, the main reason being that the bias-variance decomposition of the 0/1 loss function is not unique, and therefore the relationship between ensemble accuracy and diversity is still unclear. In the parallel field of regression ensembles, where the loss function of interest is the mean squared error, this decomposition not only exists, but it has been shown that *diversity can be managed* via the Negative Correlation (NC) framework. In the field of probabilistic modelling the expected value of the negative log-likelihood loss function is given by its conditional entropy; this result suggests that *interaction information might provide some insight into the trade off between accuracy and diversity.* Our objective is to improve our understanding of classifier diversity by focusing on two different loss functions – the mean squared error and the negative log-likelihood.

In a study of mean squared error functions, we reformulate the Tumer & Ghosh model for the classification error as a regression problem, and we show how *the NC learning framework can be deployed to manage diversity in classification problems.* In an empirical study of classifiers that minimise the negative log-likelihood loss function, we discuss *model diversity* as opposed to error diversity in ensembles of Naïve Bayes classifiers. We observe that *diversity in low-variance classifiers has to be structurally inferred.* We apply interaction information to the problem of monitoring diversity in classifier ensembles. We present empirical evidence that *interaction information can capture the trade-off between accuracy and diversity*, and that *diversity occurs at different levels of interactions* between base classifiers. We use interaction information properties to *build ensembles of structurally diverse* averaged Augmented Naïve Bayes classifiers. Our empirical study shows that this novel ensemble approach is computationally more efficient than an accuracy based approach and at the same time it does not negatively affect the ensemble classification performance.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns any copyright in it (the "Copyright") and s/he has given The University of Manchester the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

ii. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the John Rylands University Library of Manchester. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

iii. The ownership of any patents, designs, trade marks and any and all other intellectual property rights except for the Copyright (the "Intellectual Property Rights") and any reproductions of copyright works, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and exploitation of this thesis, the Copyright and any Intellectual Property Rights and/or Reproductions described in it may take place is available from the Head of School of Computer Science (or the Vice-President).

Acknowledgements

If there is one single person I have to thank, that is my supervisor Gavin Brown. He taught me what research is, and how to be strong through the up and downs of this job. On my very first day here we started a learning path together which I hope will continue after my PhD. We have spent the best moments having fun in front of a white board, figuring out triangles, coming up with new ideas, questioning what's beyond question. I couldn't have asked for a better supervisor.

A special thank goes to Giorgio Fumera and Fabio Roli for their useful chats and their share of experience before and during my PhD. In the end, it is also thanks to them if I have had the chance to do my PhD. Thanks for believing in me and helping me when I was going through difficult times. I also want to thank the person who started all my interest for computer science and maths, Gigi Uras. He inspired me, and somehow, predicted all this happening. Thanks for having been such a strict maths teacher, and for being so uniquely you.

To my family, who have always supported me in my studies and my aspirations. When I had the occasion to start a PhD abroad, they encouraged me to go for it without any hesitation. Despite the long distance, they only thought what was best for me. I am no easy person, but yet, they have always been there for me, my dad, my mum, Diego, Cico and Luna. If I am here today I owe it all to my mum. She taught me the importance of following my dreams at any cost or sacrifice. This PhD is for you mum, I hope that by fulfilling my aspirations, I have fulfilled some of yours too.

Last, but not least, there is Aled. I wouldn't have done it without your support. *Grazie di essere ancora qui accanto a me, e di avermi saputo aspettare.*

Notation

General Notation

p	dimensionality of the input space, number of features
c	dimensionality of the output space, number of classes
N	dimensionality of a dataset, number of samples
M	dimensionality of an ensemble, number of classifiers
$\lvert \cdot \rvert$	number of elements of a set, as in $\lvert\mathcal{D}_U\rvert$

Probabilistic Notation

$\boldsymbol{X}$	input random variable in a p-dimensional space $\mathbb{R}^p$
Y	class label random variable
$\boldsymbol{x}$	input vector in a p-dimensional space $\mathbb{R}^p$, realization of $\boldsymbol{X}$
y	class label, realization of Y
$\mathcal{X}$	set of N input vectors $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
$\mathcal{Y}$	set of N class labels $\mathcal{Y} = \{y_1, \ldots, y_N\}$
$\mathcal{D}$	dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\} = \{(\mathbf{x}_i, y_i)\}$ for $i = 1, \ldots, N$
$\mathcal{D}_L$	labelled Dataset $\mathcal{D}_L = \{\mathcal{X}_L, \mathcal{Y}_L\} = \{(\mathbf{x}_i, y_i)\}$ for $i = 1, \ldots, \lvert\mathcal{D}_L\rvert$
$\mathcal{D}_U$	unlabelled Dataset $\mathcal{D}_U = \mathcal{X}_U = \{\mathbf{x}_j\}$ for $j = 1, \ldots, \lvert\mathcal{D}_U\rvert$
$\mathrm{p}(\cdot)$	probability density function
$\hat{\mathrm{p}}(\cdot)$	estimate of a probability density function from data
$\mathrm{p}'(\cdot)$	first derivative of a probability density function $\mathrm{p}(\cdot)$
$\mathbb{E}[\cdot]$	expectation of a random variable $\boldsymbol{Z}$, as in $\mathbb{E}[\boldsymbol{Z}]$
$\mathcal{L}$	log-likelihood function

Decision Theoretic Notation

$\mathcal{H}$	space of possible hypotheses
f, g, h	classifiers or models or hypotheses
$h_1, \ldots, h_z$	hypotheses in the search space $\mathcal{H}$

h^*	optimal hypothesis
$f^1, \ldots, f^M$	base classifiers of an ensemble
$\mathrm{L}(\Phi, \tau)$	loss function, as the loss of predicting τ when the state of nature is Φ
$\mathcal{I}(\alpha, \beta)$	indicator function, taking value 1 if $\alpha = \beta$, and 0 otherwise
$\mathcal{F}$	combination rule
$\epsilon(x)$	noise associated with the input x
$\hat{y}$	classifier estimate of the class label y
$\hat{y}^{\mathrm{ens}}$	ensemble estimate of the class label y
$f_i^m(\mathbf{x})$	output of the $m-$th classifier for the $i-$th class label
$f_i^{\mathrm{ens}}(\mathbf{x})$	ensemble output for the $i-$th class label
$\bar{f}_i(\mathbf{x})$	output of a linearly combined ensemble for the $i-$th class label
Ω	set of class labels $\{\omega_1, \ldots, \omega_c\}$
ω_k	k-th class label, $\omega_k \in \Omega$
$f^{\mathrm{ens}}(\mathbf{x})$	output of an ensemble of estimators
$\bar{f}(\mathbf{x})$	output of an ensemble of linearly combined estimators
$f^m(\mathbf{x})$	output of the $m-$th estimator of an ensemble
d	target value, scalar $d \in \mathbb{R}$

Probabilistic Models

$\boldsymbol{\theta}$	set of model parameters $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_q\}$
$\boldsymbol{\mu}$	mean vector
σ^2	covariance
Σ	covariance Matrix
$\mathbf{I}$	identity Matrix
$\boldsymbol{\alpha}$	vector of parameters for the Dirichlet distribution, $\boldsymbol{\alpha} \in \mathbb{R}^+$
$\boldsymbol{\pi}$	vector of parameters for the multinomial distribution, $\sum_i \pi_i = 1$
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	normal distribution with mean μ and covariance matrix Σ
$\mathrm{Mult}(\boldsymbol{\pi})$	multinomial distribution with parameters $\boldsymbol{\pi}$
$\mathrm{Dir}(\boldsymbol{\alpha})$	Dirichlet distribution with parameters $\boldsymbol{\alpha}$
$\mathcal{W}(W, \boldsymbol{\nu})$	Wishart distribution with parameters W and $\boldsymbol{\nu}$

Mutual Information

$A, \ldots, Z$	random variables
$\{A, \ldots, Z\}$	set of random variables

$\mathrm{H}(\cdot)$	entropy of a random variable, as in $\mathrm{H}(A)$
$\mathrm{H}(\cdot\|\cdot)$	conditional entropy, as in $\mathrm{H}(A\|B)$
$\mathrm{I}(\cdot;\cdot)$	mutual information of a pair of random variables, as in $\mathrm{I}(A;B)$
$\mathrm{I}(\cdot;\cdot\|\cdot)$	conditional mutual information, as in $\mathrm{I}(A;B\|C)$
$\mathrm{KL}(\mathrm{p}\|\mathrm{q})$	Kullback-Leibler divergence of two probability distributions p and q
$\mathrm{I}(\{\cdot\})$	interaction information of a set of random variables, as in $\mathrm{I}(\{A,B,C\})$
$\mathrm{I}(\{\cdot\}\|\cdot)$	conditional interaction information, as in $\mathrm{I}(\{A,B,C\}\|D)$

Chapter 1

Introduction

Everyday life is a continual flow of *decisions* that are to be taken. Whether that is to plan our day or whether to buy a new pair of socks, we are all constantly called on to make decisions. It is common sense to ask for other people's opinions whenever the *cost* of a decision is high, as in where to invest our savings, or whenever our understanding of a problem is *weak*, as in whether to treat a recurrent headache or not. This idea, which was formalised in early studies of political science concerned with the conditions under which a committee can make better decisions than a single person [98], has nowadays become the foundation of democracy and it has been widely applied to many work, social and research areas. This thesis is concerned with the development of *committees of learning machines which can take autonomous decisions*, and therefore finds in the Machine Learning community its target audience. Interestingly, whereas most everyday tasks can be solved easily, the advent of a computing era has opened up the possibility to solve complex tasks which go beyond human knowledge.

In this chapter we introduce the general problem of building machines that can learn from examples and we point out the motivations for our work in this research area. We illustrate the research questions which we are going to address in the remaining chapters of this thesis and we conclude with an outline of this thesis.

1.1 Learning from Examples

Machine learning is the discipline concerned with the development of *learning algorithms*, that is, machines able to teach themselves from examples how to solve

a task in an automated way [47, 25]. These learners are basically algorithms which can learn some knowledge about a problem from a finite set of examples of the problem itself, and that are able to generalise their behaviour to *new* examples.

This thesis focuses on a specific class of decision problems, that is classification problems, where we want to label an object according to a predefined set of class labels. The class of algorithms which learn from examples how to classify unseen objects are known as classifiers.

An important point to make here is that classifiers cannot correctly classify every object, but will incur in a *classification error* [43, 25]. One possible reason for that is that a classifier learns to classify new objects from a finite set of *training* examples which may or may not represent in full the classification problem. Therefore, a classifier might be missing out important examples that would make the knowledge of the problem more *accurate*. Another reason is that the set of training examples might be affected by *noise*, and therefore a classifier might be *biased* towards a *wrong* representation of the problem. A third reason is the so called Bayes error, the minimum error that a classifier can make due to the overlapping nature of the classification problem. We will discuss Bayes error in more detail in Section 2.4.3. A final reason for the classification error is given by the fact that there exist no algorithm that is suited to solve all classification problems. On the other hand, there exists many classification algorithms that are suited to solve different classification problems. The difficult process of selecting the most accurate algorithm to solve a specific problem might therefore affect the classification performance, as the selected classifier might actually model the problem in a *wrong* way.

A classifier is therefore characterised by its ability to *generalise* its knowledge to unseen objects. The 0/1 loss function is one possible measure of this generalisation ability, as this function is a measure of the *cost* we incur when we classify an object, and in particularly, when we make a mistake about the real class label of an object. A way to choose a classification algorithm is to test different types of classifiers on the problem at hand and to select the one that makes fewer mistakes on new objects. However, the problem of estimating the real *generalisation ability* of a classifier is quite challenging as, for instance, this estimation might be affected by several factors, as for instance the finite number of examples [43, 25].

An alternative to this selection approach is given by *ensemble learning*. Ensemble learning combines strengths and weaknesses of accurate and different classifiers to solve the same classification problem [21]. Intuitively, in order to make the most of this cooperative approach, base classifiers should be different or, more precisely, *diverse*, that is, they should be expert on different examples. In this way, ensemble learning can combine and deploy the expertise of different classifiers to increase the generalisation ability of the learning algorithm [57].

1.2 The Challenge of Understanding Classifier Diversity

Studies have shown that ensemble learning can outperform the single classifier approach [37, 26, 39]. The rationale behind this methodology is that by combining different and accurate models, we may improve the ensemble decision over each single classifier decision, that is, we may make fewer errors.

The notion of *diversity* between base classifiers seems to play a key role in the success of ensemble learning techniques [56]. Intuitively classifiers are diverse *if* they make *different errors*. It seems therefore natural to try to understand the reduction of ensemble errors in terms of reductions of the ensemble 0/1 loss function, and quantify the balance between accuracy and diversity in terms of the same loss. However, *the link between ensemble accuracy and diversity is not so apparent*, the main reason being that *there is no unique way to express the* 0/1 *loss function in terms of the accuracy (or bias) and the variability (or variance) of a classifier* [48, 10, 52, 22, 86, 53, 82]. Nevertheless, the Tumer & Ghosh model for the analysis of the classification error was the first framework to link the ensemble classification error with the *correlation* (or existing dependencies) between base classifiers [86]. This framework suggests that base classifiers should have low bias and high variance for an ensemble to outperform the single classifier [34].

Despite these attempts, to date there is still no agreement on how to measure, define or even manage diversity for classifier ensembles [57]. This lack of a unifying approach differs from the parallel field of regression ensembles, where diversity is a well known problem, and where the NC learning framework, which is based on the mean squared error loss function rather than the 0/1 loss function, has been shown to manage diversity [15]. Understanding *what* diversity is, and *how* it affects

the ensemble performance through the trade-off between ensemble accuracy and diversity is a matter of primary importance, as it would provide us with a better understanding of the conditions under which an ensemble succeeds over a single classifier approach and overall might result in building more accurate ensembles.

1.3 Thesis Aims and Objectives

1.3.1 Thesis Questions

This thesis studies diversity in classifier ensembles from a broader perspective than error diversity, and has the main objective of improving the general understanding of classifier diversity. Although the idea of combining *diverse* classifiers is widely acknowledged in the Ensemble Learning community, the link between ensemble accuracy and diversity is still unclear, the main reason being that there is no unique way to decompose the 0/1 loss in terms of the bias and variance of a classifier [48]. Therefore, the main research question this thesis addresses is *"Can we use a loss function other than the* 0/1 *loss to understand and manage diversity in classifier ensembles?"*

An interesting result in classifier ensembles is given by the Tumer & Ghosh model for the classification error [86]. This framework was the first work to link the ensemble classification error with the *correlation* between base classifiers. Another interesting result is given by the the Negative Correlation (NC) learning framework for regressor ensembles [15]. Here, the loss function of interest is the mean square error rather than the 0/1 loss function. The NC framework, which has its foundations on the bias-variance-covariance decomposition of the mean squared error, can directly *manage* diversity in ensembles of regressors [15]. Following the main research question of this thesis, the first research question we address in this thesis is *"Can we deploy NC learning in the context of the Tumer & Ghosh framework so that we can manage diversity in classification problems?"* This question is of great importance, as the final aim of understanding diversity is to improve the ensemble performance, and yet to date there are no algorithms that can *actually manage* diversity between base classifiers (Chapter 2).

In the machine learning community, diversity has always been associated with the notion of *error diversity*. Intuitively, base classifiers are diverse if they misclassify different objects, that is they make *different errors*. Since error diversity

is an immediate consequence of model diversity, the second research question we address in this thesis is *"Is it possible to generate diverse classifiers by looking at model diversity rather than error diversity?"* Parametric probabilistic models, that is, models that minimise the negative log-likelihood loss function rather than the 0/1 loss function, are particularly suited to this purpose, as they *explicitly select the model bias* of a classifier. In these models, diversity can occur at two different levels: it can be parametric or structural. More specifically, we address this research question: address is therefore *"Is parametric diversity sufficient to build accurate and diverse stable classifiers?"* This question is important for two reasons. The first reason is that similarly to the mean squared error, negative log-likelihood loss functions can be used to understand classifier diversity. The second reason is that this question would clarify whether it is possible to generate diverse and stable (i.e. low variance) classifiers with ensemble techniques such as Bagging or Random Subspaces, as opposed to the idea that these techniques require unstable classifiers to succeed [10]. This would confirm the evidence that the stability of base classifiers depends on several aspects, such as for instance the size of the training set [77, 78, 1], and that traditional ensemble techniques can be successfully applied to Naïve Bayes classifiers [28, 83] (Chapter 4).

A third research question this thesis addresses is *"What kind of measure could we use to understand diversity?"* To date, there is no formal definition or measure of diversity for classifier ensembles. Mutual information [75] is a proxy to classification accuracy which is maximised when minimising the negative log-likelihood loss function. Interaction information [64], which extends mutual information to deal with more than two random variables, quantifies the existing *interactions* between random variables (Chapter 3). Our specific research question is then *"Can we use interaction information to understand diversity between base classifiers?"* If the answer to this question is positive, this would clarify that diversity occurs at different levels of interactions between random variables, and therefore pairwise and non-pairwise interactions contribute to the diversity of an ensemble (Chapter 5).

Since interaction information quantifies the interaction between random variables and that these interactions can be thought of as model dependencies between random variables, our final research question is *"Can we use interaction information to generate accurate and structurally diverse base classifiers in a sensible way?"* This question is of primary importance, as it would address the problem

of how to choose structurally diverse models in such a way that the ensemble classification accuracy is not negatively affected, and it would also indicate that interaction information can be used as a way to generate accurate and diverse classifiers (Chapter 6).

1.3.2 Contributions of the Thesis

This thesis proposes novel viewpoints on classification diversity. In particular, we show how different loss functions such as the mean squared error and the negative-log likelihood can be used to understand and manage diversity in classifier ensembles. Our contributions can be summarised as follows:

- We reformulate the Tumer & Ghosh model for the analysis of the classification error into a regression model that minimises the mean squared error loss function. We deploy the Negative Correlation framework in this context and we show that our novel algorithm can manage diversity in classification problems for ensembles of neural networks (Chapter 2).

- We study model diversity in ensembles of Naïve Bayes classifiers for the ensemble techniques of Bagging and Random Subspaces. Whereas Bagging generates parametric diversity between Naïve Bayes models, Random Subspaces introduce a certain level of structural diversity between Naïve Bayes models, as in the latter case base classifiers share the same model dependencies but are trained on different features. Our study shows that parametric diversity is not sufficient when combining stable classifiers such as Naïve Bayes models. In fact, in this case base classifiers are accurate but not diverse. Conversely, feature diversity introduced by Random Subspaces generates base classifiers which are diverse but not accurate enough to make the ensemble outperform the single classifier approach. We conclude that for stable classifiers, such as Naïve Bayes models, diversity has to be structurally inferred (Chapter 4).

- We show that the success of Bagging with stable classifiers such as Naïve Bayes classifiers depends on the size of the training set and on the model specifications. This is in line with results found for other stable classifiers [78, 1] (Chapter 4).

- We discuss empirical work showing that interaction information can capture the trade off between ensemble accuracy and diversity (Chapter 5).
- We present empirical evidence that diversity occurs at different levels of interactions between base classifiers, and therefore higher order interactions between base classifiers cannot be discarded. As a result, Bayesian Networks can only approximately model interactions between base classifiers (Chapter 5).
- We present empirical evidence that the sign of interaction information measured between features is a good proxy for classification accuracy of augmented Bayesian Networks and averaged Bayesian Networks, and that therefore it can be used to choose the structure of an augmented Bayesian network (Chapter 6).
- We propose a novel ensemble technique, irsADE, which exploits interaction information properties to generate accurate and structurally diverse averaged Bayesian networks. We present an empirical comparison of irsADE with another ensemble method which measures the accuracy of base classifiers, rather than interaction information. We show that irsADE does not negatively affect the ensemble accuracy but on the contrary is at least an order of magnitude faster than the accuracy based method (Chapter 6).

1.4 Thesis Outline

In Chapter 2 we introduce the research context for this thesis investigation, that is, ensemble learning for classification. We illustrate the importance and the difficulty of defining and measuring diversity, and we describe the Tumer & Ghosh framework, the first work to show a relationship between the ensemble accuracy and the correlation between classifiers [85]. We describe our novel contribution to the problem of diversity by showing how the Tumer & Ghosh framework can be reformulated as a regression problem, and how the NC learning framework [15] can be extended to manage diversity in a classification context [101]. We conclude this chapter by noticing how these existing frameworks explain the ensemble performance in terms of error loss functions, and by making the observation that the negative log-likelihood loss function is another possible candidate to improve our understanding of ensemble diversity.

In Chapter 3 we introduce probabilistic models, that is, learning models that directly minimise the negative log-likelihood loss function rather than error loss functions. As such, Bayesian networks are graphical models that represent statistical dependencies between random variables of a probability distribution [68], and are an object of investigation throughout this thesis. We introduce mutual information as a natural way to quantify statistical dependencies between random variables [75], and we point out that the main limitation of mutual information is that it can only measure dependencies between pairs of random variables. On the contrary, interaction information [64] can be used to understand interactions between any number of random variables. This makes interaction information a suitable candidate for understanding relationships between base classifiers.

In Chapter 4 we embark on an empirical study of diversity for ensembles of Naïve Bayes classifiers [100]. Our results show that our Bagging implementation can only generate parametric diversity between base classifiers, and that this level of diversity is not sufficient to generate stable diverse classifiers. On the contrary, Random Subspaces can generate models with the same structure on different subsets of the feature space, and this level of feature diversity is able to generate diverse but not sufficiently accurate classifiers. This analysis suggests that *structural* diversity could lead to the design of accurate and diverse ensembles of Bayesian Networks.

In Chapter 5 we show empirically that interaction information can be used to understand the trade off between accuracy and diversity of an ensemble. In particular, we find that diversity happens at different levels of interactions between base classifiers. Diversity is only approximately a pairwise measure, since higher order interactions also occur. An indirect consequence of this is that Bayesian Networks, which only model pairwise interactions, can only approximately model the existing interactions between base classifiers.

In Chapter 6 we show that interaction information can be used to build structurally diverse ensembles. We propose irsADE, an ensemble method that makes use of certain properties of interaction information to generate fast and accurate averaged Bayesian Networks ensembles. We compare our method with an analogous accuracy based method that selects base classifiers according to their classification accuracy. Our results show that the use of the prior knowledge provided by interaction information does not adversely affect the classification accuracy but on the contrary it reduces the computational time by at least an

order of magnitude.

In Chapter 7 we summarise our main contributions towards understanding base classifier diversity from an interaction information perspective, and we discuss future developments.

1.5 Publications Resulting from the Thesis

Part of the work presented in this thesis has been published in a number of papers:

[**101**] Manuela Zanda, Gavin Brown, Giorgio Fumera, and Fabio Roli.
"*Ensemble Learning in Linearly Combined Classifiers via Negative Correlation*", Seventh International Workshop on Multiple Classifier Systems. Prague, Czech Republic, May 2007.

[**100**] Manuela Zanda and Gavin Brown.
"*A Study of Semi-supervised Generative Ensembles*", Eighth International Workshop on Multiple Classifier Systems. Reykjavik, Iceland, June 2009.

Chapter 2

Traditional Viewpoints on Ensemble Diversity

In the previous chapter we introduced the general notion of a classifier, and we pointed out how its performance on a given task depends on its ability to generalise to unseen examples. The difficulty of assessing the generalisation ability of classifiers led us to introduce classifier ensembles, an approach that is based on combining many diverse classifiers rather than selecting the classifier with the best generalisation ability from a pool of candidate classifiers.

In this chapter we formally define these concepts, we focus on traditional viewpoints on diversity in classifier ensembles, an we propose a novel perspective to ensemble diversity. In Section 2.1 we formally define classifiers as algorithms which learn how to assign class labels to unseen objects by making the least number of mistakes. In Section 2.2 we discuss the difficulty of selecting the classifier with the best generalisation ability. As an alternative learning approach, in Section 2.3 we introduce classifier ensembles, which can solve the same classification task by combining the decisions of different and accurate base classifiers. One of the key concepts here is that base classifiers should be *diverse*, i.e. they should produce different outputs for the same input. In Section 2.4 we discuss the concept of diversity from a traditional viewpoint. Despite the empirical support from the literature, there is still little understanding of how to use or even quantify this diversity. Following this discussion in Section 2.5 we propose a novel perspective to ensemble diversity, in which we show that diversity can be managed. As we will see, the challenge of measuring diversity to build more accurate ensembles is linked to the form of the loss function, and it is far from being solved.

2.1 Classifiers and Loss Functions

Classification is the area of machine learning concerned with the problem of assigning class labels to unseen objects. An object is usually known as a *pattern*, and it is represented by a feature vector $\boldsymbol{x} \in \mathbb{R}^p$ and by its class label $y \in Y$. In general terms a classifier f can be thought of as a mapping from the feature space X to the space of possible class labels $\Omega = \{\omega_1, \ldots, \omega_c\}$:

$$f : \mathbb{R}^p \longrightarrow \Omega \ , \tag{2.1}$$

such that given an unseen pattern x, the classifier returns an estimate for its class label $\hat{y} = f(\mathrm{x})$.

This thesis takes a statistical approach to classification by assuming that data patterns are independent and identically-distributed (i.i.d.) from the joint probability distribution $\mathrm{p}(\boldsymbol{X}, Y)$ of the feature random variable $\boldsymbol{X}$ and of the class label random variable Y. While $\boldsymbol{X}$ can be either continuous or discrete, the class random variable Y is always discrete, and can take values from the finite set $\Omega = \{\omega_1, \ldots, \omega_c\}$. A pattern sample is completely defined by a pair $(\boldsymbol{x}, y)$, where $\boldsymbol{x} \in \mathbb{R}^{\mathrm{p}}$ is its feature vector and y is its class label. Each pair represents an instance i.i.d. from the joint probability distribution $\mathrm{p}(\boldsymbol{X}, Y)$.

A classifier can be seen as a two-step algorithm. In the *training phase* a classifier learns estimates of the true class posterior distributions. In the *testing phase*, a classifier uses these estimates of the true class posterior probabilities to predict the class labels of new unseen objects.

2.1.1 Training Phase: Learning From Examples

In the training phase a classifier learns an estimate of the class posterior distribution $\mathrm{p}(Y|\boldsymbol{X})$ from a finite set of training data $\mathcal{D}$. The way classifiers learn from data depends strongly on the type of data available for training. In fact, we can distinguish between three different learning scenarios:

Supervised Scenario The training data is a finite set of $|\mathcal{D}_L|$ labelled pattern samples $\mathcal{D} = \{\boldsymbol{\mathcal{X}}, \mathcal{Y}\} = \{(\boldsymbol{x}_i, y_i)\}$ for $i = 1, \ldots, |\mathcal{D}_L|$, which are assumed to be i.i.d. according to the joint probability distribution $\mathrm{p}(\boldsymbol{X}, Y)$.

Semi-supervised Scenario The training data is made of a finite set of $|\mathcal{D}_L|$ labelled patterns $\mathcal{D}_L = \{\boldsymbol{\mathcal{X}}_L, \mathcal{Y}_L\} = \{(\boldsymbol{x}_i, y_i)\}$ for $i = 1, \ldots, |\mathcal{D}_L|$ and a finite

set of $|\mathcal{D}_U|$ unlabelled data patterns $\mathcal{D}_U = \boldsymbol{\mathcal{X}}_U = \{\boldsymbol{x}_j\}$ for $j = 1, \ldots, |\mathcal{D}_U|$. It is quite often the case that the number of the unlabelled patterns is much bigger than the number of labelled patterns: $|\mathcal{D}_U| \gg |\mathcal{D}_L|$. In the semi-supervised case labelled patterns can be seen as i.i.d. samples from the joint probability distribution $\mathrm{p}(\boldsymbol{X}, Y)$, whereas unlabelled patterns can be seen as i.i.d. samples from the input probability distribution $\mathrm{p}(\boldsymbol{X})$.

Unsupervised Scenario The training data is a finite set of $|\mathcal{D}_U|$ unlabelled patterns $\boldsymbol{x} \in \boldsymbol{X}$, $\mathcal{D}_U = \boldsymbol{\mathcal{X}}_U = \{\boldsymbol{x}_i\}$ for $i = 1, \ldots, |\mathcal{D}_U|$. Each sample is assumed to be i.i.d. from the probability distribution $\mathrm{p}(\boldsymbol{X})$.

The partial or total availability of training labels affects the learning process to such an extent that learning algorithms have been specifically designed to deal with each of these case scenarios. In fact, it is often the case that missing labels not only makes the learning algorithm more complex but it also affects its classification performance [18].

Different classifiers are based on different learning paradigms and implement different algorithms. Neural networks, decision trees, Naïve Bayes and nearest neighbours are but a few examples of classifiers, and they are all based on different learning approaches. Despite this great variety of learning algorithms[1], every single classifier aims at solving the same problem, that is, to classify new objects.

2.1.2 Testing Phase: Classifying New Examples

In the testing phase a classifier can apply *Bayesian decision theory* to predict the class label of any unseen pattern $\boldsymbol{x}$ [25]. Bayesian decision theory is a statistical approach to classification which assign an unseen pattern to the class with highest posterior probability. For instance, if during its training phase a classifier has learnt some estimates of the class posterior distributions $\mathrm{p}(Y = \omega_1|X), \ldots, \mathrm{p}(Y = \omega_C|X)$, according to Bayesian decision theory a classifier will choose the class label $\hat{y} = \omega_j$ with highest posterior probability, that is for any $j \neq i,\ i = 1 \ldots c$:

$$\hat{y} = \omega_j \ \Leftrightarrow\ P\left(Y = \omega_j | X = x\right) > P\left(Y = \omega_i | X = x\right) \ . \tag{2.2}$$

In this way a classifier will minimise the classification error that is inevitably associated with the action of making any decision. In fact, even if the estimates of the

[1]The reader might refer to [25, 6, 43] for an extensive description of the state of the art of classifiers.

class posterior probabilities would perfectly match the true posterior probability distributions, the decision step would still incur an inevitable classification error known as *Bayes error*, which is an immediate consequence of taking a probabilistic approach in the decision process [25]. This point, which is discussed more in detail in Subsection 2.4.3, suggests that an important aspect of all classification problems is to assess the classification performance of different classifiers in order to choose the classifier that performs best on the task at hand.

2.1.3 Classifier Evaluation

A classifier is usually characterised by its *generalisation ability*, that is, its capacity to generalise its decisions to new data [43]. Intuitively, we would like to build a classifier that is able to classify unseen data correctly, but since we can only learn an estimate of the true class posterior distributions, any classifier will make mistakes, or classification errors.

Loss Functions

A *loss function* is a way to measure the loss a classifier incurs when it classifies a pattern. More generally, a loss function $L(\Phi, \tau)$ measures the loss of performance associated with a specific action, and which is caused by the mismatch between the true state of nature Φ and its estimate τ [3]. Some of the most used loss functions in Machine Learning are:

0/1 Loss This function quantifies the loss we incur when we classify a pattern. Specifically, given a pattern $\boldsymbol{x}$ whose true class label is y, the loss of a classifier f which predicts its class label as $\hat{y} = f(\boldsymbol{x})$ is 1 if the classifier classifies the pattern incorrectly (i.e., $\hat{y} \neq y$), and is 0 otherwise:

$$\mathrm{L}(y, \hat{y}) = \mathcal{I}(y \neq \hat{y}) \ . \tag{2.3}$$

Squared Error Loss This function quantifies the estimation loss we make by approximating a continuous target d associated with an input $\boldsymbol{x}$ with its estimate $f(\boldsymbol{x})$

$$\mathrm{L}(d, f(\boldsymbol{x})) = (d - f(\boldsymbol{x}))^2 \ . \tag{2.4}$$

The error function that is minimised in neural networks is an example of squared loss function.

Negative Log-likelihood This function measures the loss we incur when we learn an estimate $\hat{\mathrm{p}}(y|\boldsymbol{x})$ of the true class posterior distribution $\mathrm{p}(y|\boldsymbol{x})$ from the training data $\mathcal{D} = \{\boldsymbol{\mathcal{X}}, \mathcal{Y}\}$:

$$-\mathcal{L} = \mathrm{L}(\mathrm{p}(Y|\boldsymbol{X}), \hat{\mathrm{p}}(Y|\boldsymbol{X})) = -\log \hat{\mathrm{p}}(Y|\boldsymbol{X}) \ . \tag{2.5}$$

Probabilistic models such as Naïve Bayes or Mixture Models can learn their model parameters by minimising the negative log-likelihood loss function.

These three loss functions provide us with different ways to measure the loss we incur when we try to estimate the ground truth. Whereas the first two loss functions are aimed at minimising the estimation error, the latter is aimed at minimising the loss we incur in estimating the true model distribution. However, it is worth pointing out that the mean squared error loss function can be derived from the negative log likelihood under the assumption that the estimation error is Gaussian distributed [5].

The negative log-likelihood loss function follows the Maximum Likelihood principle of choosing the model that most likely fits the data. In fact, minimising the negative log-likelihood of a model is equivalent to maximising the likelihood of a model given the data at hand. The reason why the negative log-likelihood can be viewed as a loss function is that minimising the negative log-likelihood also minimises the Kullback-Leibler (KL) divergence between the true probability distribution $\mathrm{p}(y|\boldsymbol{x})$ that generated the data and the estimate $\hat{\mathrm{p}}(y|\boldsymbol{x})$ of this true distribution:

$$\mathrm{KL}(\mathrm{p}\|\hat{\mathrm{p}}) = \iint_{\boldsymbol{X},Y} \mathrm{p}(Y|\boldsymbol{X}) \log \frac{\mathrm{p}(Y|\boldsymbol{X})}{\hat{\mathrm{p}}(Y|\boldsymbol{X})} d\boldsymbol{X}\, dY \ . \tag{2.6}$$

The KL divergence is a non commutative measure of how two probability distributions are close to each other. In fact, the KL divergence between p and $\hat{\mathrm{p}}$ is 0 if and only if p and $\hat{\mathrm{p}}$ are exactly the same probability distribution, and is non-zero otherwise. The expected value of the negative log-likelihood in Equation (2.5), that is

$$\mathbb{E}[-\mathcal{L}] = \iint_{\boldsymbol{X},Y} \mathrm{p}(Y|\boldsymbol{X}) \log \hat{\mathrm{p}}(Y|\boldsymbol{X}) d\boldsymbol{X}\, dY \ , \tag{2.7}$$

can be related to the KL divergence between the true probability distribution and its estimate by rewriting Equation (2.6) as:

$$\mathrm{KL}(\mathrm{p}\|\hat{\mathrm{p}}) = \iint_{\boldsymbol{X},Y} \mathrm{p}(Y|\boldsymbol{X}) \log \mathrm{p}(Y|\boldsymbol{X}) d\boldsymbol{X} dY - \iint_{\boldsymbol{X},Y} \mathrm{p}(Y|\boldsymbol{X}) \hat{\mathrm{p}}(Y|\boldsymbol{X}) d\boldsymbol{X} dY \ , \tag{2.8}$$

and by noticing that the second term of Equation (2.8) corresponds to the expected value of the negative log-likelihood as in Equation (2.7). Since the first term in Equation (2.8) is fixed and depends on the true class posterior distribution, it is easy to conclude that by minimising the expected value of the negative log-likelihood, we are also minimising the KL divergence between the true class posterior distribution and the estimate of the class posterior distribution.

Training and Test Error

The training error of a classifier is defined as the average loss of a classifier over a set of training patterns $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \text{ for } i = 1, \ldots, N\}$:

$$\overline{err} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(y_i, \hat{y}_i) \ . \tag{2.9}$$

The error in Equation (2.9) cannot be used as a measure of generalisation performance, as the training error decreases as we increase the complexity of a classifier. In such cases where the complexity depends on the size of the training set, as we increase the number of training samples the training error of a classifier will asymptotically converge to zero. Conversely, the same classifier will lose its generalisation ability to classify unseen patterns, and the error on unseen data will increase. It is therefore of primal importance to evaluate the performance of a classifier on an independent *test set* [43].

The *test error* is the *generalisation error* of a classifier that has been trained on a training set $\mathcal{D}$, and that is it defined as the expected value of the classification error measured over a test set which is independent from the training set:

$$\mathrm{Err}_{\mathcal{D}} = \mathbb{E}[\mathrm{L}(\Phi, \tau)|\mathcal{D}] \ . \tag{2.10}$$

The quantity defined in Equation (2.10) is hard to estimate, as it requires the

calculation of a conditional expectation [43]. In practice we measure the classification error as the expected test error over every possible random variable:

$$\mathrm{Err} = \mathbb{E}[\mathrm{L}(\Phi, \tau)] = \mathbb{E}[\mathrm{Err}_{\mathcal{D}}] \ . \tag{2.11}$$

This quantity, which from now onwards we will simply refer to as test error or generalisation error, is actually the expectation of the test error over every possible random variable involved, and therefore it averages over every source of randomness in both the training data and the classifier, and it is calculated as the 0/1 loss function over the test set, as done in Equation (2.9) for the training data.

2.2 Model Selection

The goal of classification is to make use of the prior knowledge of a problem to learn the best estimate of the true class posterior distribution $\mathrm{p}(Y|\boldsymbol{X})$. The process of choosing the most appropriate classifier model from a set of possible candidates given this prior knowledge of the problem is known as *model selection*.

However, in order to provide an estimate of the true class posterior distribution, every classification algorithm has to make assumptions about this distribution, with the result that the classifier performance will strongly depend on the correctness of such assumptions. Whenever these assumptions are *wrong*, a *model mismatch* is said to occur, as the model estimated from the training data does not match the true model that generated the data.

At an abstract level any learning algorithm can be thought of as a search in the space $\mathcal{H}$ of *representable* hypotheses, as shown in Figure 2.1. In this representation a point or hypothesis $h \in \mathcal{H}$ corresponds to a specific instance of the model that can be generated by varying the algorithm parameters. The aim of classification is to define $\mathcal{H}$ such that the true model g belongs to this space and the learning algorithm is able to find the optimal approximation h^* to g. Therefore, a model mismatch can happen for two main reasons:

- The true model g does not belong to the space of learnable hypotheses $\mathcal{H}$, that is, there is a *structural* mismatch between the true model g and any model $h \in \mathcal{H}$ that can be achieved via the learning algorithm.

- The true model g belongs to the space of learnable hypotheses $\mathcal{H}$ but the

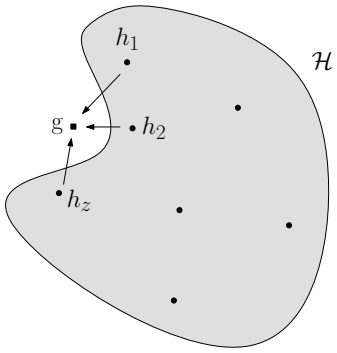


Figure 2.1: An ensemble of different models can explore a larger space of possible hypotheses by approximating the true hypothesis with a combination of wrong ones [21].

learning algorithm is not able to find the optimal estimate $h^* \in \mathcal{H}$ of the true model g.

Model selection is concerned with the process of estimating the performance of different classifiers in order to choose the model which is closest to the true model, and has therefore the best generalisation ability. If enough data is available, a validation set separate from the training set can be used to assess each model performance and choose the most accurate one. However, this is hardly ever the case, and many theoretical and empirical criteria have been formulated to quantify and adjust the match between the estimated model and the true model [43]. These methods use the prior knowledge of the problem (such as training data, prior information, prior distributions or cost functions) to select the optimal model [25].

2.3 Ensemble Learning

Ensemble learning tries to overcome the model mismatch limitations associated with the single model selection approach by replacing the single classifier with a combination of *accurate* and *different* models. The idea behind this framework is borrowed from early studies of political science [98], and can be summarised as follows: under certain assumptions a committee of expert classifiers can make better classification decisions than a single classifier. The following section motivates ensemble learning from a model selection viewpoint and provides an overview of the main concepts.

2.3.1 Motivation

Following the analogy proposed in Subsection 2.2 of thinking of a learning algorithm as a search in the space of possible hypotheses $\mathcal{H}$, ensemble learning can be thought of as a complex learning algorithm that can reach several points in the space at once, and that combines the models associated with these points to find a better estimate of the true function g.

Ensemble learning can be seen as an alternative approach to the problem of model selection. In a seminal paper [21], Dietterich identifies three main pitfalls of single model selection, and three different ways an ensemble learning approach could avoid them:

Statistical Problem The size of the training data is not large enough to select the optimal model on a validation set. Therefore, the learning algorithm might identify more than one point in the search space $\mathcal{H}$ showing optimal identical accuracies on the validation data. According to the no free lunch theorem [92, 91], without any other a priori knowledge of the problem there is no reason to choose one over another.

In an ensemble approach models showing identical accuracy can be combined together. Therefore, the risk of choosing the wrong classifier can be reduced.

Computational Problem Under the assumption that the size of the training data is large enough for the learning algorithm to converge to one single point, the landscape of the space of possible hypotheses $\mathcal{H}$ might be such that the algorithm might converge to a local optimum rather than exploring the full space of possible hypotheses and finding the global optimum solution.

In an ensemble approach it would be possible to start the learning algorithm from different points in the search space. Moreover, a combination of these models might be able to escape from local optima.

Representational Problem Due to the limited amount of training samples, the true model might not belong to the space of searchable hypotheses. In fact, although the learning algorithm might be able to learn the true model, the finite amount of training data might reduce the actual search space and hence exclude the true model from the same search space.

In an ensemble approach it might be the case that a weighted combination of different models $h_1, \ldots, h_M$ in the space of possible hypotheses $\mathcal{H}$ could lead to a better approximation of the true model g, even if this model does not belong to $\mathcal{H}$. This situation is illustrated in Figure 2.1.

2.3.2 Main Concepts

Ensemble learning is a learning approach where multiple classifiers are involved in solving a classification problem. This approach is very general and can be achieved in many different ways. For instance, a problem can be tackled in a cooperative or in a modular way such that base classifiers can either solve the full problem or disjoint parts of the same problem. Base classifiers can be arranged into parallel, serial or hybrid topologies. The combination rule can either be a fusion rule, where all the base classifiers contribute to the final ensemble decision, or a selection rule, where only a subset of the base classifier contributes to the final ensemble decision.

In this thesis we focus on parallel architectures, where base classifiers work in parallel on the same task, and on fusion combination rules, where all the base classifier outputs are combined in a cooperative way to solve the same problem. This ensemble scenario is depicted in Figure 2.2. As shown in this figure, the ensemble design can be split into (a) the design of the base classifiers and (b) the design of the combination rule. The reader might refer to [57] for a more detailed discussion of alternative ensemble architectures.

Base Classifiers

If base classifiers were all identical, there would be no need to build an ensemble out of them, as the ensemble decision would be exactly the same as each single base classifier. The key question of base classifier design is: *How do we generate different base classifiers from a given training set?*

With reference to this problem of generating different classifiers, a relevant type of classifiers is that of *unstable* classifiers. Breiman informally defines a classifier to be "unstable" when small perturbations in the training set or in the construction of the classifier can cause large changes in the classifier predictions [8, 9, 10]. More formally, unstable classifiers are classifiers characterised by *high variance*. Neural networks and decision trees are good examples of unstable

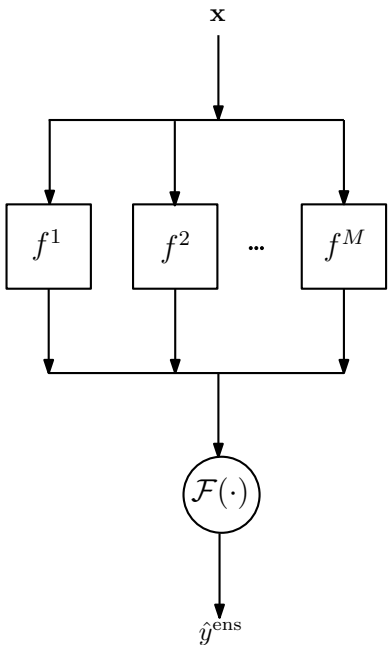


Figure 2.2: A parallel ensemble architecture. Given an unseen pattern $\mathbf{x}$ to the base classifiers $f^1, \ldots, f^M$, their outputs are combined according to some functional $\mathcal{F}$ to get the final ensemble decision $\hat{y}^{\text{ens}}$.

classifiers, whereas support vector machines and k-nearest neighbours are clear examples of stable classifiers. To give a simple example of unstable classifier, two identical neural networks trained on two slightly different training sets might give different predictions on the same unseen pattern. Similarly, two neural networks that only differ by the way their weights have been initialised and that have been trained on the same training set might give different predictions on the same unseen pattern.

There are two main ways of creating different classifiers: we can either train the same classifier algorithm on different training sets, or we can train different classifier algorithms on the same training set.

The first option, which is usually referred as *Perturb & Combine* approach [10], is the foundation of the most successful ensemble techniques, such as *Bagging*, *Adaboost* and *Random Subspace Methods* [9, 31, 45]. These techniques have been efficiently applied to unstable classifiers to build ensembles of weak (i.e. better than random guessing) classifiers [10]. One possible sampling strategy is to sample (with or without replacement) over the training patterns, so that base classifiers are trained on a slightly different *replica* of the whole training set — as in Bagging (uniform sampling), and Adaboost (non-uniform sampling). Another possible

sampling strategy to sample from joint or disjoint subsets of the feature space, so that base classifiers is trained on overlapping or disjoint subsets of the training set — as in Random Subspace Methods.

The second option can be achieved by changing the parameters of each base classifier or even combining different types of classifiers. For instance, we could combine neural networks with different number of hidden nodes or different staring weights, or more radically combine neural networks with decision trees. This approach, which would enforce *model diversity*, would probably reduce the correlation between the classifier bias (which, as we will see, is a condition that should be avoided) [86], but at the same time would make unclear what kind of gain or decision boundaries might arise from them.

Combination rules

Many combination rules have been studied for the purpose of ensemble learning [80, 81, 51, 26]. The choice of the combination rule is strongly affected by the level of information provided by the outputs of base classifiers [93]. In this thesis we are concerned with classifiers that provide two different levels of output information, that is (a) base classifiers that directly estimate the *class label* of an unseen pattern — such as support vector machines, and (b) base classifiers that provide estimates for the *class posterior probabilities* of an unseen pattern — such as linear discriminant classifiers. In the first case a classifier ensemble provides a set of possible label candidates (one for each classifier member) to the combination rule. *Simple majority vote* and *weighted majority vote* are but two examples of rules that combine crisp label outputs. In the second case the output of base classifiers not only provide a classifier decision (that according to Bayes' rule is the class with the highest probability) but it also provides a level of confidence of the classifier in making a decision. *Simple mean*, *weighted mean* and *product rule* are examples of combination rules that can take real valued outputs.

We now extend the notation introduced in Section 2.1 to classifier ensembles. We recall the supervised classification scenario, where a pattern is denoted by a pair $(\mathbf{x}, y)$, in which $\mathbf{x}$ is a feature vector $\mathbf{x} \in \mathbb{R}^p$ and y can assume one out of c class labels $\omega_1, \ldots, \omega_c$. We denote a set of M classifiers with $f^1, \ldots, f^M$. Given an unseen pattern $\mathbf{x}$, a base classifier f^m returns a $c-$dimensional output vector $[f_1^m(\mathbf{x}), \ldots, f_c^m(\mathbf{x})]$, where the single value $f_i^m(\mathbf{x})$ represents the support of the $m-$th classifier for the $i-$th class, for $i = 1, \ldots, c$.

Combining Votes. If base classifiers directly estimate class labels, the output vector $[f_1^m(\mathbf{x}), \ldots, f_c^m(\mathbf{x}))]$ of the $m-$th classifier is a vector of zeros and ones, where

$$f_i^m(\mathbf{x}) = \begin{cases} 1 & \text{if } f^m \text{ predicts class } \omega_i, \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

The *majority vote* rule decides for the class ω_k from $\Omega = \{\omega_1, \ldots, \omega_c\}$ that generates the highest consensus among all base classifiers:

$$f_k^{\text{ens}}(\mathbf{x}) = \arg\max_k \sum_{i=1}^{M} f_k^i(\mathbf{x}). \tag{2.13}$$

In this case each base classifier decision equally contributes to the final ensemble decision. On the contrary, a *weighted majority* rule can be used to increase the decision power of classifiers that make more accurate predictions. This can be done by associating each base classifier f^i with a *weight* w_i. The higher the weight, the stronger the classifier prediction will affect the ensemble decision. The weighted majority voting rule decides for the class label ω_k with highest weighted support $f_k^{\text{ens}}(\mathbf{x})$, defined as

$$f_k^{\text{ens}}(\mathbf{x}) = \arg\max_k \sum_{i=1}^{M} w_i f_k^i(\mathbf{x}) \ . \tag{2.14}$$

Although in theory weighted majority can be more accurate than simple majority, it has the downside that its performance relies on the actual weights. That is, a bad choice of weights might give lower accuracy than simple majority [57].

Combining real valued outputs. Continuous base classifier outputs can be thought of as the *evidence* a classifier assigns to classes after having observed a data point [71], or simply as estimates of the class posterior probabilities [46].

The *simple mean* rule calculates the overall support $f_i^{\text{ens}}(\mathbf{x})$ of the ensemble for each class ω_i as the mean average of each $j-$th base classifier support for that class:

$$f_i^{\text{ens}}(\mathbf{x}) = \frac{1}{M} \sum_{j=1}^{M} f_i^j(\mathbf{x}) \ . \tag{2.15}$$

The *product* rule calculates the overall support $f_i^{\text{ens}}(\mathbf{x})$ of the ensemble for each

class ω_i as the product of the support of each $j-$th base classifier:

$$f_i^{\text{ens}}(\mathbf{x}) = \prod_{j=1}^{M} f_i^j(\mathbf{x}) \ . \tag{2.16}$$

According to Bayesian decision theory, the ensemble decides for the class $\hat{y}$ with highest support:

$$\hat{y} = \max_{i=1}^{c} f_i^{\text{ens}}(\mathbf{x}) \ . \tag{2.17}$$

Classifiers have been combined mainly by averaging or by majority vote. Although experimental results [41, 79] have shown that linear methods outperform other existing rules, these two combination rules lack any intuitive support. Nevertheless, a general framework has shown that under the assumption of base classifiers trained on statistically independent feature subsets, the product rule can be derived from Bayes' theorem [51]. Moreover, the simple average rule as well as other linear combination rules can be derived from the product rule by imposing more restrictive assumptions on the closed form of the class posterior distributions. Experimental results have shown that if base classifiers are trained on independent feature spaces, the product rule will be more accurate than the simple mean [80]. However, if base classifiers are not trained on independent feature spaces, the simple mean, which is the combination rule derived under the most unrealistic assumption, outperforms the product rule since the simple mean is less sensitive to estimation errors than the product rule [81]. The same results show that the choice of the combination rule has to take into account the accuracy of the single base classifiers, as well as the nature of the classification problem such as number of classes or whether it is possible or not to split the subspace into statistically independent feature subsets.

2.4 Diversity in Ensemble Learning

If experimental results have often shown that ensemble learning can outperform the single classifier approach [37, 26, 39], the understanding of the conditions under which an ensemble outperforms a single classifier is still an open question. In line of principle base classifiers should be *specialised on different subsets* of the classification problem, so that by focusing on different parts of the problem, they can cooperate to achieve better generalisation performance than each single

base classifier. The notion of classifier *diversity* seems therefore to play a key role in ensemble learning. Intuitively diverse classifiers should perform differently if tested on the same data patterns. However, *how to measure, define, and possibly control diversity are still matters of ongoing research* [57].

Despite the lack of a formal definition of diversity, the research community has put a lot of effort on heuristically incorporating diversity into the ensemble learning process. Diversity measures have been more explicitly used to *monitor* the base classifiers [70] and to *prune* the base components of an ensemble [63, 20, 38, 72]. Resampling techniques have been used to *inject randomness* into the design of base classifiers. As such, one way to create diverse classifiers is to provide the *same inputs* to *different base classifiers*. Random Subspaces is a clear example of this approach [45]. Another alternative would be to provide *different input*s to the *same base classifiers*. This is the approach taken in Bagging and Adaboost [9, 31].

2.4.1 Understanding Diversity

In the ensemble community it is well known that base classifiers should be accurate and diverse. Diversity alone has been shown not to be beneficial to the ensemble performance, but there is often a trade-off between the ensemble accuracy and diversity, so that base classifiers should be at the same time accurate and diverse. In fact, diversity cannot be increased without negatively affecting the ensemble accuracy [40, 63]. Moreover, studies have shown that there exists "good" and "bad" diversity [42, 14].

Quantifying diversity among the base components of an ensemble is a matter of great importance, because it would (a) improve the understanding of how single components interact to reduce the ensemble error, and (b) point out practical guidelines for the design of ensembles with improved performance. With reference to the latter point, the knowledge about the base classifier diversity could be used in the design of the individual classifiers as well as in the choice of the combination rule.

There have been many attempts to define a relationship between the ensemble accuracy and the level of diversity among base components, but none has completely succeeded so far [58, 76]. Nevertheless, on a large scale (i.e. when diversity is forced to span the whole range of possible values), it has been shown that when base classifiers show similar levels of accuracy, a non linear relationship between

the ensemble accuracy and the base classifier diversity [56] exists. However, it is often the case that base classifiers generated with standard ensemble techniques show small diversity variation. Motivated by this evidence, we now discuss some theoretical results.

2.4.2 "Diversity of Diversity"

Most of the work done towards the understanding of the concept of diversity and towards an analysis of the numerous measures proposed in the literature can be found in Kuncheva's papers. To date, there is no strict definition of diversity, nor is there common agreement on how to measure it. As Kuncheva clearly stated in one of her papers, we are still in a "diversity of diversity" stage [56].

In the attempt to find a measure of diversity which could link diversity to the ensemble accuracy, Kuncheva has analysed ten diversity measures for binary classifier outputs[2], of which four are pairwise measures and six are non-pairwise measures [58]. Unlike non-pairwise measures, pairwise diversity measures involve the outputs of only two classifiers at a time. The overall diversity is the average of all possible pairwise measures. An example of pairwise measure is Yule's Q statistics [99]. Results have shown that the Q statistics is sensitive to small disagreement between classifiers but not to the accuracy of base classifiers [58], properties which are well suited for the detection of diversity in base classifiers. Furthermore, it has been shown that there exists a relationship between the pairwise Q statistics and the upper and lower limits of the accuracy achievable by majority voting [59]. It has to be pointed out that none of the diversity measures in [58] relate to loss functions.

We now focus on the Tumer & Ghosh model [85, 86], a framework which focuses on another pairwise diversity measure, that is the correlation between base classifiers.

2.4.3 The Tumer & Ghosh Framework

The Tumer & Ghosh framework [85, 86] is the first work to show that *correlations* between continuous classifier outputs have a quantifiable effect on the ensemble training error. This framework studies the effect of combining base classifiers on the distribution of the decision boundaries between two different classes, and

[2]That is, when classifiers output whether they are making a correct or incorrect prediction.

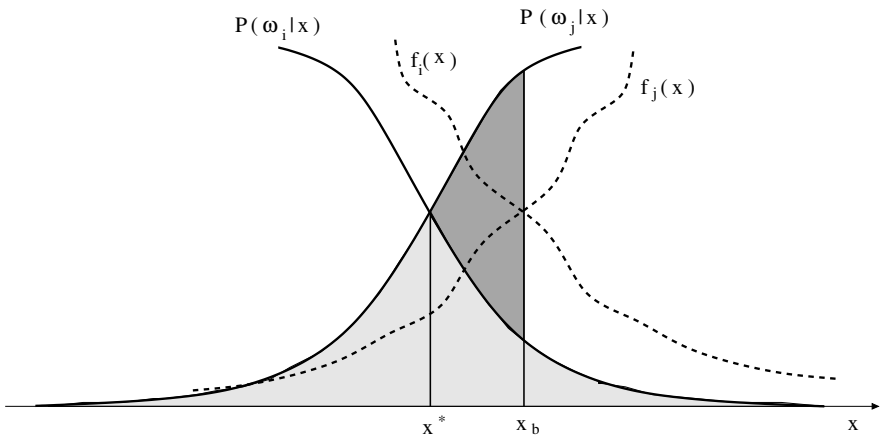


Figure 2.3: The estimates of the true class posterior distributions shift the ideal decision boundary x^* by $b = x_b - x^*$. The classification error is partly due to the irreducible Bayes error (light-grey area) and partly to the added error (dark-grey area).

shows how a linear combination of base classifiers might reduce the training error through a reduction of the overall error variance of the ensemble.

The classification error of any base classifier has its lower bound in the so called *Bayes error*, an irreducible quantity that depends on whether the true class posterior distributions overlap. For any pattern falling in these overlapping areas, Bayes' rule will always decide for the class with highest posterior probability, so that a classifier able to learn the true posterior distributions will incur the Bayes error, which is given by the area underneath these curves [25]. As such, the Bayes error is a measure of the best performance attainable by a classifier. However, this quantity is hard to estimate in practice, as it requires knowledge of the true probability distribution $\mathrm{p}(\boldsymbol{X}, Y)$ which generated the data. Since any classifier can only learn an estimate of the true class posterior distributions, a classifier will incur an *added error* as well as the Bayes error, as shown in Figure 2.3.

Tumer & Ghosh analyse the effects of estimating the class posterior distributions on the added error around a hypothetical decision boundary between two classes[3]. In particular they study the classification error of a single classifier and the classification error of a linearly combined ensemble of classifiers for the case where the estimates of the class posterior distributions produce a shift of

[3]The Tumer & Ghosh framework also applies to multi-class problems, as according to Bayes' rule, decision boundaries are usually dominated by the two classes with highest class posterior probabilities [85]

the decision boundary between the true class posterior distributions. This situation is shown in Figure 2.3, where we represent a hypothetical decision boundary between the two classes ω_i and ω_j. From this picture it is easy to notice that when the classifier chooses the class scoring the highest posterior probability estimate, non-optimal decisions are taken for patterns x where the decision made according to the posterior estimates disagrees with the decision made according to the true posterior distribution, i.e. $\arg\max_k f_k(x) \neq \arg\max_k \mathrm{p}(\omega_k|x)$, and that this disagreement introduces a further added error (dark-grey area) to the optimal Bayes error (light-grey area). The added error is just a portion of the overall classification error evaluated around the decision boundary x^*.

This framework[4] is based on the assumption that for a given pattern x and for any given class ω_k, where $k = 1, \ldots, c$, a classifier provides an estimate $f_k(x)$ of the true posterior distribution $\mathrm{p}(\omega_k|x)$ in the form:

$$f_k(x) = \mathrm{p}(\omega_k|x) + \epsilon_k(x) \quad . \tag{2.18}$$

The difference between the value of the true posterior distribution and its estimate at the point x is quantified by the error term $\epsilon_k(x)$. Tumer & Ghosh show that under the further assumptions that (a) the classifier estimates of the true posterior distributions produce a shift x_b of the ideal decision boundary x^* and (b) the true posterior distributions are monotonic around the ideal decision boundary x^*, the added error E for a single classifier is proportional to the square of the boundary shift $b = x_b - x^*$

$$\mathrm{E} = \frac{p(x^*)t}{2} b^2 \tag{2.19}$$

and that the shift itself can be expressed as a function of the estimation errors $\epsilon_i(x_b)$ and $\epsilon_j(x_b)$:

$$b = \frac{\epsilon_i\left(x_b\right) - \epsilon_j\left(x_b\right)}{t} \quad , \tag{2.20}$$

where t denotes the difference between the first order derivatives of the class posteriors at the optimal boundary x^*: $t = \mathrm{p}'(\omega_j|x^*) - \mathrm{p}'(\omega_i|x^*)$. They prove that the expected added error $\mathrm{E}_{\mathrm{add}} = \mathbb{E}\{\mathrm{E}\}$ for a single classifier can be decomposed

[4]For clarity of exposition we discuss this framework for a one-dimensional feature space. It is worth pointing out that this framework extends to the multi-dimensional case [84].

in terms of the *bias* β_b and the *variance* σ_b^2 of this shift b:

$$\mathrm{E}_{\mathrm{add}} = \frac{p(x^*)t}{2}\left(\beta_b^2 + \sigma_b^2\right) \quad , \tag{2.21}$$

where bias and variance are defined as $\beta_b = \frac{\beta_i - \beta_j}{t}$ and $\sigma_b^2 = \frac{\sigma_i^2 + \sigma_j^2}{t^2}$.

This analysis is further extended to a linear combination of M classifiers $f^1, \ldots, f^M$, such that the ensemble estimate for a given class ω_k is given by

$$\bar{f}_k(x) = \frac{1}{M}\sum_{m=1}^{M} f_k^m(x) = p(\omega_k|x) + \bar{\epsilon}_k(x) \quad , \tag{2.22}$$

where the average error $\bar{\epsilon}_k(x)$ denotes the linear combination of the single base classifier errors $\epsilon_k^m(x)$ and is defined as $\bar{\epsilon}_k(x) = \frac{1}{M}\sum_{i=1}^{M} \epsilon_k^m(x)$. In a similar way to the expression of the added error for a single classifier in Equation (2.19), the added error of a linearly combined ensemble can be expressed in terms of the averaged shift $\bar{b} = \frac{\bar{\epsilon}_i(x_b) - \bar{\epsilon}_j(x_b)}{t}$ as

$$\overline{\mathrm{E}} = \frac{p(x^*)t}{2}\bar{b}^2 \tag{2.23}$$

Tumer & Ghosh show that the expected added error $\bar{\mathrm{E}}_{\mathrm{add}} = \mathbb{E}\{\overline{\mathrm{E}}\}$ for an ensemble of linearly combined classifiers can be written as the sum of two separate terms, one accounting for the *bias* $\beta_{\bar{b}}$, and one accounting for the *variance* $\sigma^2{}_{\bar{b}}$ of the shift $\bar{b}$ of the linearly combined ensemble:

$$\overline{\mathrm{E}}_{\mathrm{add}} = \frac{p(x^*)t}{2}\left(\beta_{\bar{b}}^2 + \sigma^2{}_{\bar{b}}\right) \quad , \tag{2.24}$$

and that these two terms can be expressed in terms of the single base classifiers bias and variance. In particular, the ensemble error bias $\beta_{\bar{b}}$ is the average of the single base classifier bias terms β_b^m:

$$\beta_{\bar{b}} = \frac{1}{M}\sum_{i=1}^{M} \beta_{b^m} \quad . \tag{2.25}$$

Under the assumption that the estimation errors of different classifiers on different classes are uncorrelated, i.e. for ϵ_i^m and ϵ_j^n, and $i \neq j$, the ensemble error variance $\sigma_{\bar{b}}^2$ decomposes into a term involving the variance of single base classifiers $\sigma_{b^m}^2 =$

$\frac{(\sigma_i^m)^2+(\sigma_j^m)^2}{t^2}$ and into another term which involves pairwise *correlations* ρ_i^{mn} and ρ_j^{mn} between base classifiers[5] for the two classes ω_i and ω_j:

$$\sigma^2{}_{\bar{b}} = \frac{1}{M^2}\sum_{i=1}^{M}\sigma^2_{b^m} + \frac{1}{t^2}\frac{1}{M^2}\sum_{m=1}^{M}\sum_{n\neq m}(\rho_i^{mn}\sigma_i^m\sigma_i^n + \rho_j^{mn}\sigma_j^m\sigma_j^n) \ . \tag{2.26}$$

Overall, Equation (2.24) shows that the expected added error of a linearly combined ensemble depends on the bias and correlation between base classifiers. More specifically, Equation (2.25) shows that the bias component of the ensemble may not be reduced with respect to the base classifier bias components, nonetheless, it is guaranteed not to be larger than the largest base classifier bias.

If base classifiers are *unbiased* (i.e., $\beta_b^m = 0$ for any $m = 1, \ldots, M$) *but correlated*, and each classifier has identical variance $(\sigma_k^m)^2 = \sigma^2$, Equation (2.26) can be rewritten as

$$\sigma^2{}_{\bar{b}} = \frac{1 + (M-1)\delta_{ij}}{M}\sigma_b^2 \tag{2.27}$$

where $\delta_{ij} = \frac{1}{2}\frac{1}{M(M-1)}\sum_{m=1}^{M}\sum_{n\neq m}(\rho_i^{mn} + \rho_j^{mn})$. The quantity $\frac{1+(M-1)\delta_{ij}}{M}$ in Equation (2.27) is less or equal than 1. Therefore, the variance of the ensemble is reduced with respect to the variance of the single classifiers by a factor which depends on the correlations between base classifiers. This result, which has been extended to linearly and weighted averaged ensembles of unbiased and negatively correlated classifiers with non identical variances in [34], as well as different decision boundaries in [4], is very important, as it shows that *linearly combined ensembles of unbiased and negatively correlated base classifiers can reduce the variance of the estimation error.*

If base classifiers are *unbiased and uncorrelated* (i.e. $\rho_k^{mn} = 0$ for any class $k = 1, \ldots, c$ and for any pair m, n of classifiers, with $m, n = 1, \ldots, M$ and $m \neq n$), and they have the same variance $(\sigma_k^m)^2 = \sigma^2$ the expected added error for the linearly combined ensemble reduces to a single variance term

$$\overline{\mathrm{E}}_{\mathrm{add}} = K\ \frac{1}{M}\sigma_b^2 \ . \tag{2.28}$$

K is a multiplicative constant $K = \frac{p(x^*)t}{2}$ which does not dependent on the

[5]We adopt Pearson's correlation as our definition for correlation. The correlation ρ between two random variables X and Y is then defined as the ratio of the covariance between the two random variables over the product of the standard deviation of the random variables: $\rho^{X,Y} = \frac{cov(X,Y)}{\sigma_X\sigma_Y}$.

base classifier estimation errors. By inspecting Equation (2.28) it is easy to conclude that the expected added error is inversely proportional to the number of base classifiers in the ensemble, and that a linear combination of unbiased and uncorrelated classifiers can outperform a single classifier because it reduces the variance of the estimation error by a factor which depends on the size of the ensemble.

If base classifiers are *biased but uncorrelated*, the bias component in Equation (2.24) has a negative effect on the expected ensemble error, as a result, the overall ensemble error reduction is less than $\frac{1}{M}$.

The Tumer & Ghosh framework provides some guidelines on how base classifiers in an ensemble should be for the ensemble to outperform a single classifier approach. Ideally base classifiers should be unbiased and uncorrelated, as in this case the ensemble would benefit from the maximum estimation error reduction, as in Equation (2.28). However, this is hardly ever the case, as in practice it is very difficult to generate uncorrelated classifiers, the main reason being that classifiers are usually trained on different replicas of the same training set. Therefore, the Tumer & Ghosh analysis suggests that in practice base classifiers should have low bias and large variance, as a linear combination rule could reduce the ensemble estimation error by reducing the variance of base classifiers.

2.4.4 Diversity in Regression Problems

A parallel field to classifier ensembles is that of *regression ensembles*; that is, ensembles of estimators that solve a regression problem. In regression quantifying diversity among component individuals of an ensemble is a well-defined problem. It is interesting to point out that the loss function of interest in regression is not the 0/1 loss function as in the Tumer & Ghosh model, but instead the mean squared error (MSE). A central result here is the *bias-variance-covariance* decomposition of the mean squared error. This decomposition of the MSE shows that the performance of the ensemble is critically dependent on the three-way balance between bias, variance, and covariance; the latter accounting for correlations between estimators. This trade-off is the analog of the oft-cited "diversity" in the classifier ensemble literature.

Geman et al. [36] demonstrate that for regression problems the MSE can be

broken into separate components, namely the *bias* and the *variance* of the error:

$$\mathbb{E}\left\{(f(\boldsymbol{x})-d)^2\right\} = \left(\mathbb{E}\left\{f(\boldsymbol{x})\right\}-d\right)^2 + \mathbb{E}\left\{\left(f(\boldsymbol{x})-\mathbb{E}\left\{f(\boldsymbol{x})\right\}\right)^2\right\} \tag{2.29}$$

where f denotes the estimator, d denotes the target, and the expected value is calculated with respect to all possible training sets.

Ueda and Nakano [87] extend this concept to a linearly combined regression ensemble (i.e. where the estimator is $\bar{f}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} f(\boldsymbol{x})^m$), providing the *bias-variance-covariance* decomposition of the MSE.

Krogh and Vedelsby developed the *Ambiguity decomposition*, another important decomposition for the MSE [55]. They prove that at a single data point $\boldsymbol{x}$ the MSE can be broken into an accuracy and an "ambiguity" term:

$$\left(\bar{f}(\boldsymbol{x})-d\right)^2 = \frac{1}{M}\sum_{m=1}^{M}\left(f^m(\boldsymbol{x})-d\right)^2 - \frac{1}{M}\sum_{m=1}^{M}\left(f^m(\boldsymbol{x})-\bar{f}(\boldsymbol{x})\right)^2 \quad . \tag{2.30}$$

The first term is an index of the *accuracy* of the individuals, while the second one characterises *diversity* among individuals, being a measure of how individual output estimates differ from the ensemble output estimate for this single data point.

Brown et al. [15] show that the expected value of the Ambiguity decomposition corresponds to the bias-variance-covariance decomposition, and that in regression *there exists a common term which quantifies the accuracy-diversity trade-off*. Diversity cannot be maximised without affecting the accuracy of the individual components, and the oft-cited "diversity dilemma" is in fact a three-way balance between bias, variance, and covariance. Moreover, Brown [11] shows that the bias-variance-covariance (diversity) trade-off can be *explicitly managed* by Negative Correlation (NC) [62], a learning algorithm that introduces a penalty term in the error function of a classifier. In his formulation the NC algorithm uses the Ambiguity decomposition as it tries to minimise a "diversity-encouraging" error function [15]:

$$e^{\text{div}} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\left(f^m(\boldsymbol{x})-d\right)^2 - \gamma\frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\left(f^m(\boldsymbol{x})-\bar{f}(\boldsymbol{x})\right)^2 \quad . \tag{2.31}$$

It is easy to notice that, except for linear scaling factors, the last term in Equation (2.31) is equal to the ambiguity term in Equation (2.30).

2.5 A Novel Regression Perspective on Classifier diversity

The equivalence between the Ambiguity decomposition and the bias-variance-covariance decomposition [11] and its immediate application to regression ensembles via the NC learning algorithm [15] represent a well-grounded theoretical basis for the understanding of regression ensembles in terms of the trade off between accuracy and diversity.

In this section we propose a novel interpretation of the Tumer & Ghosh model by reformulating it as a regression problem. More precisely, we show that this framework can be interpreted in light of the Negative correlation framework, and that this result can be used to manage diversity for classification problems.

In the original Tumer & Ghosh model the random variable of interest is the boundary shift $b = x^* - x_b$. In fact, as shown in Figure 2.3, which for convenience has been reproduced here in Figure 2.4, as b decreases towards 0, the added error (dark grey area) decreases. When no boundary shift occurs there is no added error and the estimates of the class posterior distributions equal the true class posterior distributions.

Another key point of the Tumer & Ghosh model, is that although it analyses the bias and variance of the boundary shift b, this model differs from other bias-variance decompositions for classification problems, such as in [52], as it treats the error as a regression random variable [86].

The connection between the bias-variance-covariance and the Tumer & Ghosh model is not immediately apparent; the main question is: *what are the corresponding "estimator" and "target" variables in this framework?* In order to answer this question, we can first observe that the shaded area in Figure 2.4 has approximately the shape of a *triangle*. The area S of a triangle is given by $S = \frac{1}{2}$ (base $\times$ height). Secondly, we observe that by substituting Equation (2.20) into Equation (2.19), Equation (2.19) can be rewritten as:

$$\mathrm{E} = \mathrm{p}\left(x^*\right) \ \frac{1}{2} \ \left[\epsilon_i(x) - \epsilon_j(x)\right] \ \frac{\epsilon_i(x) - \epsilon_j(x)}{t} \ . \tag{2.32}$$

If we do not take into account the multiplicative constant $\mathrm{p}(x^*)$, it is easy to see that the added error is the area of a *pseudo-triangle* of base $(\epsilon_i(x) - \epsilon_j(x))$ and height $b = \frac{\epsilon_i(x) - \epsilon_j(x)}{t}$. The classifier estimate for the k-th class posterior

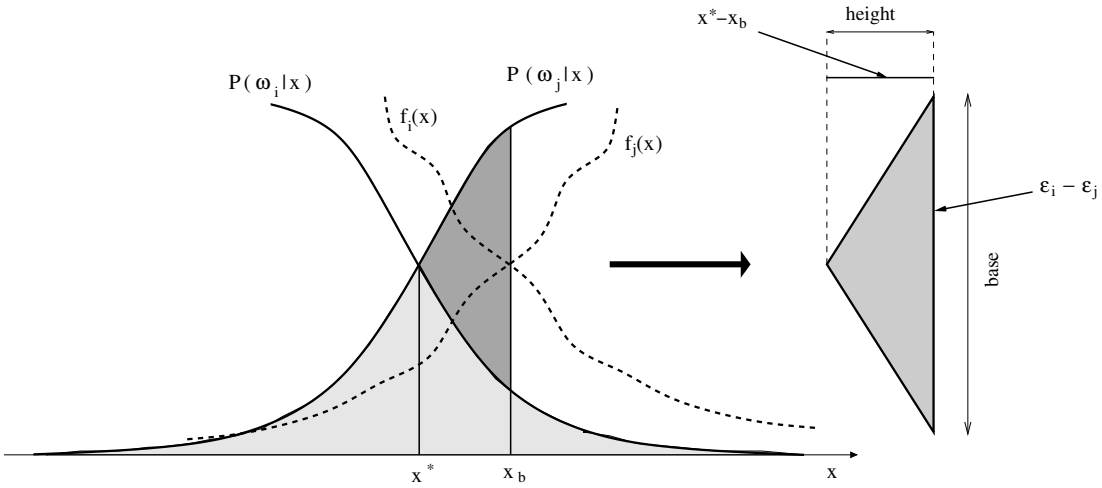


Figure 2.4: The added error has approximately the shape of a triangle.

probability can be written as $f_k(x) = \mathrm{p}(\omega_k|x) + \epsilon_k(x)$, from which we derive the estimation error for the k-th class posterior as:

$$\epsilon_k(x) = f_k(x) - \mathrm{p}(\omega_k|x) \ . \tag{2.33}$$

Given Equation (2.33), we can express the base $\epsilon_i(x) - \epsilon_j(x)$ of the triangle as the difference between two different quantities, the first one being the difference between estimates and the second one being the difference between the true class posterior probabilities:

$$\epsilon_i(x) - \epsilon_j(x) = [f_i(x) - f_j(x)] - [\mathrm{p}(\omega_i|x) - p(\omega_j|x)] \ . \tag{2.34}$$

By looking at the picture in Figure 2.4 the base of the triangle is not only proportional to b (it is t times b) but is also a more meaningful random variable. In fact the error, which is proportional to b^2, is equal to 0 whenever b is equal to 0. At the optimal boundary, that is where there is no shift and $b = 0$, the base of the triangle is equal to 0 as well:

$$[f_i(x) - f_j(x)] - [\mathrm{p}(\omega_i|x) - p(\omega_j|x)] = 0 \ . \tag{2.35}$$

In other words, the estimation error drops to 0 when the difference between the two probability estimates equals the difference between the true posterior probabilities.Therefore, *The Tumer & Ghosh model can be interpreted as a regression problem by simply considering the base instead of the height of the pseudo-triangle.*

Table 2.1: Some key aspects of the original Tumer & Ghosh (T&G) model are compared with our new interpretation in a regression context. All measures are point-wise but the input variable x has been omitted for simplicity.

	T & G Model	Novel Interpretation
Random Variable	$b = \frac{1}{t}(f_{ij} - d_{ij})$	f_{ij}
Target	0	d_{ij}
Bias	$\beta_b = \frac{1}{t}(\beta_i - \beta_j)$	$\beta_{ij} = t\beta_b + d_{ij}$
Variance	$\sigma_b^2 = \frac{1}{t^2}\left(\sigma_i^2 + \sigma_j^2\right)$	$\sigma_{ij}^2 = t^2\sigma_b^2$

The difference between the class posterior estimates $f_{ij}(x) = [f_i(x) - f_j(x)]$ can be thought of as the *estimator*, whereas the difference between the true class posterior probabilities can be thought of as the *target* $d_{ij} = [\mathrm{p}(\omega_i|x) - \mathrm{p}(\omega_j|x)]$ of our new regression problem. The *aim* of this regression problem is to make the estimator f_{ij} as close as possible to the new target d_{ij}, for any point $x \in \mathbb{R}$. This change of random variables, which makes it possible to point out a valid estimator function and a target, improves the understanding of the Tumer & Ghosh model in the sense that it not only allows the models to be redefined from another perspective, but it proposes a new interpretation of the bias variance decomposition, as shown in Table 2.1.

2.5.1 Optimising Diversity by NC Learning

We now describe how the NC learning framework can be applied to the regression formulation of the Tumer & Ghosh model via an algorithm that minimises the added error and increase the overall accuracy of a linearly combined classifier ensemble [101]. The NC algorithm works iteratively by performing a single weight update for each neural network in the ensemble, according to Equation (2.31), proceeding in a pattern-by-pattern updating scheme. The error function in Equation (2.31) allows us to train a linearly combined ensemble of estimators in parallel, rather than training each network independently, by putting[6] $\gamma = 0$. In a number of benchmark studies [11, 15] it was found that a γ value less than 1 showed significant improvements in both convergence speed and generalisation ability. It is easy to notice that, except for linear scaling factors, the last term is equal to the Ambiguity term from Equation (2.30). Given this, we now show how this algorithm can be extended to work on linearly combined classifier ensembles

[6]Equation (2.31) is equal to an independent MSE function for each network when $\gamma = 0$.

by exploiting the theoretical framework described earlier.

Let us assume we are given an ensemble of M classifiers combined by simple averaging, and that we are given a two class problem, where the two classes are respectively ω_i and ω_j. The ensemble class posterior probability estimate $\bar{f}_i(x)$ for class ω_i is:

$$\bar{f}_i(x) = \frac{1}{M}\sum_{m=1}^{M} f_i^m(x) \ , \tag{2.36}$$

whereas the difference $\bar{f}_{ij}(x) = \bar{f}_i(x) - \bar{f}_j(x)$ between the ensemble class posterior probability estimates for classes ω_i and ω_j is:

$$\bar{f}_{ij}(x) = \frac{1}{M}\sum_{m=1}^{M} \left(f_i^m(x) - f_j^m(x)\right) \ . \tag{2.37}$$

Following the Ambiguity decomposition in Equation (2.30) proposed by Krogh and Vedelsby for the MSE [55], we define the Ambiguity decomposition for the Tumer & Ghosh model for a given data point x as:

$$\left[\bar{f}_{ij}(x) - d_{ij}(x)\right]^2 = \frac{1}{M}\sum_{m=1}^{M}\left[f_{ij}^m(x) - d_{ij}(x)\right]^2 - \frac{1}{M}\sum_{m=1}^{M}\left[f_{ij}^m(x) - \bar{f}_{ij}(x)\right]^2 \ . \tag{2.38}$$

If we now apply the NC learning framework to Equation (2.38) we get the decomposition[7] $\mathrm{E_{ij}(x)} = \left(\bar{f}_{ij}(x) - d_{ij}(x)\right)^2$ as:

$$\mathrm{E_{ij}} = \left(\bar{f}_{ij} - d_{ij}\right)^2 = \frac{1}{M}\sum_{m=1}^{M}\left(f_{ij}^m - d_{ij}\right)^2 - \gamma\left\{\frac{1}{M}\sum_{m=1}^{M}\left(f_{ij}^m - \bar{f}_{ij}\right)^2\right\} \ , \tag{2.39}$$

where γ is a scaling factor that lets us vary the covariance component on E_{ij}. If we adopt a gradient descent procedure to minimise Equation (2.39), it follows that given two classes ω_i and ω_j the partial derivative for the m-th classifier and the i-th class is

$$\frac{\partial \mathrm{E_{ij}}}{\partial f_i^m} = \frac{2}{M}\left(f_{ij}^m - d_{ij}\right) - \frac{2}{M}\,\gamma\,\left(f_{ij}^m - \bar{f}_{ij}\right) \ . \tag{2.40}$$

In a real multi-class problem it is unknown which pair of classes will contribute to the added error around any point of the feature space. In this case, we have to take into account every possible pair of classes i, j such that $j \neq i$ and $i, j =$

[7] For clarity of exposition we omit the dependence of the class posterior estimates and the target from x, and we simply write $f_i^m(x) = f_i^m$ and $d_{ij}(x) = d_{ij}$ for any value of i, j, m.

$1, \ldots, c$:

$$\mathrm{E_{TOT}} = \sum_{i=1}^{c} \sum_{j>i} \left[\frac{1}{M} \sum_{m=1}^{M} \left(f_{ij}^{m} - d_{ij} \right)^2 \right] - \gamma \sum_{i=1}^{c} \sum_{j>i} \left[\frac{1}{M} \sum_{m=1}^{M} \left(f_{ij}^{m} - \bar{f}_{ij} \right)^2 \right] \quad . \tag{2.41}$$

The partial derivative of the overall error with respect to the class ω_i and the estimator function f^m is

$$\frac{\partial \mathrm{E_{TOT}}}{\partial f_i^m} = \frac{2}{M} \sum_{\substack{j=1 \\ j \neq i}}^{c} \left(f_{ij}^{m} - d_{ij} \right) - \gamma \Big[\frac{2}{M} \sum_{\substack{j=1 \\ j \neq i}}^{c} \left(f_{ij}^{m} - \bar{f}_{ij} \right) \Big] \quad . \tag{2.42}$$

Equation (2.42) simplifies to Equation (2.40) for a two class problem, and minimises the true added error for the two largest classes involved around a decision boundary. Equations (2.39) and (2.41) can be used for training a simple averaged ensemble of neural networks *in parallel*, like Equation (2.31) does in regression problems as an alternative to the standard independent training with the error function $\frac{1}{2} \sum_{m=1}^{M} \left(f^m - d \right)^2$.

2.5.2 Experimental Results

We apply this new NC learning algorithm on three real classification problems: Phoneme (two class problem), Wine (three class problem) and Heart Disease (two class problem). For a more detailed description of these datasets the reader can refer to [101].

In order to apply NC learning to classification problems, we implemented two different multilayer perceptron (MLP) algorithms that differ by the error function they minimise in their back-propagation training phase. The first MLP algorithm minimises the standard MSE error [5], whereas the second MLP algorithm minimises the error in Equations (2.39) or (2.41) as discussed in 2.5.1, depending on the number of classes of the problem at hand. From an ensemble perspective, the first algorithm trains a system of M MLPs independently from each other as in $\frac{1}{2} \sum_{m=1}^{M} \left(f^m - d \right)^2$, whereas the second algorithm trains all the MLPs simultaneously in order to minimise the overall ensemble error as in Equations (2.39) or (2.41). Both these algorithms are such that every MLP is trained with a learning rate of $\eta = 0.1$. The weights of every MLP are initialised randomly between -0.5 and 0.5. The activation function of each node is the Sigmoid function. We

Table 2.2: Mean (and 95% confidence intervals) improvement of ensembles of MLPs trained with the NC algorithm over ensembles of MLPs trained independently training. The number of epoch is fixed to 1000 epochs for low complexity ensembles ($H = 3$) and to 5000 epochs for high complexity ensembles ($H = 10$). Note that the best gains are made with large ensembles of relatively simple networks.

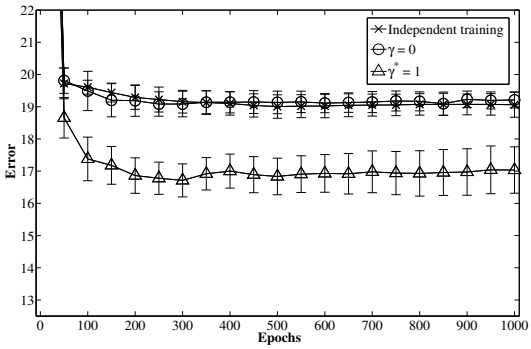
Dataset	M= 3, H= 3	M= 10, H= 3	M= 3,H= 10	M= 10, H= 10
Phoneme	2.0 (0.7)	3.8 (0.4)	−0.6 (1.1)	0.4 (0.3)
Wine	20.5 (2.1)	16.2 (1.4)	1.4 (0.5)	0.9 (0.1)
Heart	1.7 (0.1)	3.4 (0.5)	2.7 (0.4)	3.0 (0.2)

refer to the complexity of a base classifier as the number of hidden nodes H, as every MLP share the same network configuration. As a means to compare the effects of NC learning to a standard independently trained ensemble of MLPs, we decided not to use any regularisation criterion (such as a momentum) or not to adopt any stopping criterion. Instead, we fixed the number of epochs of each MLP to a predefined number. These choices are motivated by the aim of the experiment, that is to capture the improvement in classification performance which is solely due to NC learning.

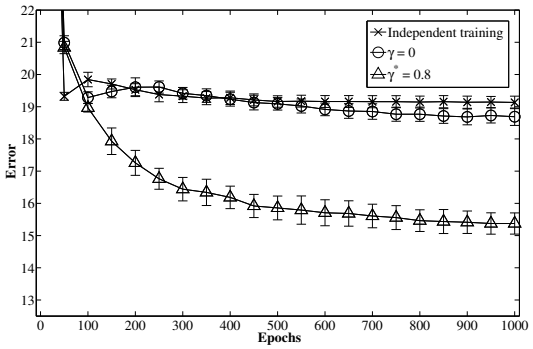
In order to understand the relationship between the number M of MLPs in the ensemble and the complexity of each base classifier, we test four different possible combinations of small/large ensembles made of low/high complexity MLPs, where we consider 3 and 10 to be respectively a suitable value for small/low and for large/high. Ten runs of the algorithm have been compared with the performance of a single classifier and with an identical ensemble[8] of individuals trained independently. Our experimental results show that every ensemble technique shows better performances than a single MLP. Table 2.2 summarises our results on different datasets—the largest improvement of NC over independent training is overall observed for a large ensemble of relatively simple MLPs, whereas the lowest improvement is overall observed for a small ensemble of complex MLPs.

In particular for Phoneme dataset the largest improvement (3.8%) is measured for a large ensemble of relatively simple MLPs, whereas the lowest improvement (−0.6%) is measured for small ensembles of relatively complex MLPs. This result seem to agree with the Tumer & Ghosh framework, where the best improvement of a linearly combined ensemble happens for low-bias and high-variance classifiers.

[8]That is an ensemble of same size and same complexity.

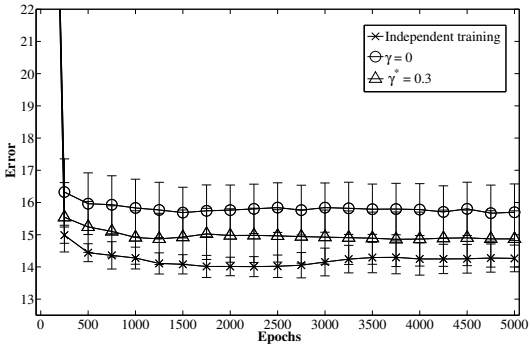


$M = 3$

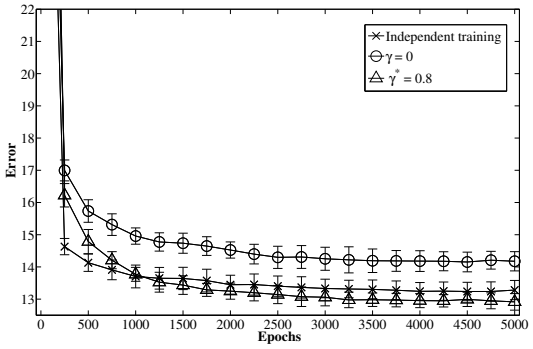


$M = 10$

Figure 2.5: Phoneme test error for an ensemble with relatively simple MLPs (each has 3 hidden nodes). On the left is an ensemble of size $M = 3$ (optimum $\gamma^* = 1$). On the right is a larger ensemble of size $M = 10$ (optimum $\gamma^* = 0.8$). The larger ensemble clearly faster convergence.



$M = 3$



$M = 10$

Figure 2.6: Phoneme test error for an ensemble with relatively complex MLPs (each has 10 hidden nodes). On the left is an ensemble of size $M = 3$ (optimum $\gamma^* = 0.3$). On the right is a larger ensemble of size $M = 10$ (optimum $\gamma^* = 0.8$). The NC technique shows no significant improvement over independent training with such complex networks.

Figure 2.5 shows the results obtained on the Phoneme dataset for ensembles of simple MLPs, whereas Figure 2.6 illustrates results obtained for ensembles of complex MLPs. These figures show the performance on the test set of the ensembles of base classifiers trained independently as well as the performance of the NC learning ensemble for the special case $\gamma = 0$ and the optimum value[9] $\gamma = \gamma^*$. If we compare the results obtained with $\gamma = 0$ and the optimum value of $\gamma = \gamma^*$, it is easy to observe that the ensemble with optimum $\gamma = \gamma^*$ always improve over the ensemble with $\gamma = 0$.

Our experimental results seem to point out that the NC learning algorithm

[9]That is, the γ which gives the best classification performance of the ensemble.

applied to the novel interpretation of the Tumer & Ghosh framework can be beneficial for the ensemble performance, and behaves similarly to the NC algorithm on regression problems [11]. Its success supports the original Tumer & Ghosh idea of decreasing correlations among classifiers as a tool for increasing the ensemble accuracy, also illustrating that this "diversity" can be *engineered* by an appropriate technique, in this case, the Negative Correlation Learning framework. Furthermore, these observations are consistent with the commonly held idea in the field that ensemble benefits are best levied from a large ensemble of relatively simple classifiers.

However, it has to be pointed out that this experimental study is a simple proof of concept that ensemble diversity can be managed, and that therefore some limitations are associated with this experimental analysis. This study should be extended to more datasets, and some form of regularisation should be introduced. More importantly, this algorithm minimises the mean squared error function and therefore makes the implicit assumption of *Gaussian* distributed noise on the posterior probability estimates, whereas in classification problems it is usual to assume binomial/multinomial noise, leading to the cross-entropy error function.

2.6 A Loss Function Perspective

Diversity seems to play a key role in the success of ensemble learning. A common aspect to all the efforts that have been made from the research community towards the understanding of diversity is that diversity is always quantified in terms of classification error. In fact, it is commonly acknowledged that the inability of base classifiers to make correct decisions should be spread on different patterns in order to make the most of an ensemble approach.

Since diversity is quantified in terms of different errors, it is closely connected to the particular loss function which is used to measure the loss in classification performance.

The 0/1 loss function, which measures the loss of predicting the wrong class label, is the quantity we usually want to minimise in a classification problem. However, the integral involved in calculating the probability of mislabelling the data can be solved only for special closed forms of the data probability distribution. Moreover, the bias-variance decomposition of the 0/1 loss function, which would provide us with a relationship between ensemble accuracy and diversity, is

not unique [48, 10, 52, 22, 53, 82].

The mean squared error is the quantity to be minimised in regression problems. It has been shown that the mean squared error of a single estimator can be decomposed in terms of two components, the bias and the variance [36]. This result has been extended to ensembles of regressors [87]. Moreover it has been shown that the expected error of the Ambiguity decomposition [55] decomposes into bias, variance and covariance [15] of the mean squared error.

Our novel contribution [101] shows that the Tumer & Ghosh model can be reformulated as a regression problem and that therefore it minimises the mean squared error loss function. It also shows that diversity can be successfully managed in classification problems. However, this model makes assumptions of Gaussian noise on the class posterior probability estimates, whereas in a classification problem the noise should follow a multinomial distribution.

The negative log-likelihood is a loss function that instead of minimising the classification error, it aims at maximising the match between our model estimate and the true model that originated the data. Probabilistic models, which naturally minimise the negative log-likelihood, will be our object of study for the remaining chapters of this thesis.

2.7 Chapter Summary

In this chapter we described the research context for this thesis investigation, that is, ensemble learning for classification. We illustrated the difficulties of defining and measuring diversity and we discussed the Tumer & Ghosh framework, the first work to show a relationship between the ensemble accuracy and the correlation between classifiers [85]. We described our novel contribution to the problem of diversity by showing how the Tumer & Ghosh framework can be reformulated as a regression problem, and how the NC learning framework [15] can be extended to manage diversity in a classification context [101].

We concluded this chapter by noticing how all these frameworks explain the ensemble performance in terms of error loss functions, and how the negative log-likelihood loss function might be another possible candidate to understand ensemble diversity. In the following chapter we focus on probabilistic classifiers, that is, models which minimise the negative log-likelihood rather than than the 0/1 loss function, and we introduce Information Theory, a framework that can

be used to understand these probabilistic models in terms of their statistical dependencies. As we will see, some Information Theory concepts are strictly related to the negative log-likelihood loss function, and might provide some insight into classifier diversity.

Chapter 3

Probabilistic Classifiers and Information Theory

In Chapter 2 we discussed the difficulties of measuring and explicitly managing diversity in classifier ensembles, and we pointed out how different theoretical frameworks explain the trade-off between accuracy and diversity in terms of decompositions of different loss functions such as the 0/1 loss or the mean squared error. In this chapter we introduce a novel probabilistic approach to the problem of understanding ensemble diversity by focussing on *negative log-likelihood* loss functions rather than traditional error loss functions.

In Section 3.1 we discuss a log-likelihood (or probabilistic) approach to classification, whereas in Section 3.2 we introduce probabilistic models, that is, learning algorithms that minimise the negative log-likelihood. We then focus our attention on Bayesian networks, a special class of probabilistic models which learn *interactions* between random variables (Section 3.3). In Section 3.4 we introduce mutual information as a way to quantify these interactions. We point out how mutual information can only quantify pairwise relationships between random variables and therefore it cannot take into account more than two base classifiers at a time. In Section 3.5 we introduce interaction information, a natural extension of mutual information to the multivariate case.

3.1 A Log-Likelihood Approach

The main goal of classification is to find the optimal estimate of the class posterior distribution $\mathrm{p}(Y|\boldsymbol{X})$ rather than the full joint probability distribution $\mathrm{p}(\boldsymbol{X}, Y)$

that is assumed to have generated the data. Nevertheless, if the joint probability distribution and the marginal probability distribution $\mathrm{p}(\boldsymbol{X})$ are known, it is possible to estimate the class posterior probability via Bayes' rule:

$$\mathrm{p}(Y|\boldsymbol{X}) = \frac{\mathrm{p}(\boldsymbol{X}|Y)\mathrm{p}(Y)}{\mathrm{p}(\boldsymbol{X})} \ . \tag{3.1}$$

Equation (3.1) tells us that there are two ways of learning a classification problem, as there are two main ways of learning the class posterior distribution $\mathrm{p}(Y|\boldsymbol{X})$ [19, 73, 65]. The first one is to take a *discriminative* approach by directly modelling the class posterior distribution $\mathrm{p}(Y|\boldsymbol{X})$. In practical terms, this corresponds to modelling our problem as decision boundaries between class regions. Typical examples of discriminative models are neural networks and decision trees. Usually these decision boundaries are learnt by minimising some error loss function, such as the 0/1 loss or the mean squared error. The second one consists of taking a *generative* approach by making explicit assumptions about the form of the class conditional distribution $\mathrm{p}(\boldsymbol{X}|Y)$ and the class prior $\mathrm{p}(Y)$, and learn the posterior distribution via Bayes' rule. Therefore, generative models learn the data distribution rather than decision regions. An example of generative models is Naïve Bayes, which is a probabilistic model based on the assumption that all the features are conditionally independent given the class. In a generative approach *the loss associated with classification is quantified in terms of the estimates of the posterior probability distributions rather than classification errors* [43].

3.2 Parametric Probabilistic Models

We now focus on parametric probabilistic models, that is models where the form of the probability distribution $\mathrm{p}(\boldsymbol{X}, Y, \boldsymbol{\theta})$ is assumed to be known and therefore completely governed by its set of parameters $\boldsymbol{\theta}$. In fact, once the model parameters have been estimated, the probability distribution is completely known. For example if the data is assumed to be continuous and normally distributed, the joint probability distribution of the data for each class variable is a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and the learning phase consists of estimating the mean $\boldsymbol{\mu}$ and the covariance matrix Σ for each value of the class label Y.

An interesting aspect of generative modelling is the possibility of explicitly expressing the way random variables factorise through repeated applications of

the product rule of probability. For instance a generative model $\mathrm{p}(\boldsymbol{X}, Y, \boldsymbol{\theta})$ can be expressed as the product of two probability terms, namely the class conditional probability $\mathrm{p}(\boldsymbol{X}|Y, \boldsymbol{\theta})$ and the class prior probability $\mathrm{p}(Y, \boldsymbol{\theta})$, which can itself be expressed as the product $\mathrm{p}(Y|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})$:

$$\mathrm{p}(\boldsymbol{X}, Y, \boldsymbol{\theta}) = \mathrm{p}(\boldsymbol{X}|Y, \boldsymbol{\theta})\mathrm{p}(Y|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}) \ . \tag{3.2}$$

The way a probabilistic generative model factorises can be graphically represented by *Bayesian Networks*. Bayesian Networks are probabilistic models that can be used to *learn existing dependencies* between the random variables of a probabilistic model [69].

The problem of learning a parametric probabilistic model given the prior knowledge of the problem and the training data consists of learning the parameters that *most likely* fit the training data. The initial knowledge about the model parameters is captured by the prior distribution $\mathrm{p}(\boldsymbol{\theta})$. Bayes' theorem can be used to update the uncertainty associated with a set of model parameters after having observed the training data $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$:

$$\mathrm{p}(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y}) = \frac{\mathrm{p}(\mathcal{X}, \mathcal{Y}|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{X}, \mathcal{Y})} \ . \tag{3.3}$$

The denominator of Equation (3.3) is a normalisation constant that does not depend on the model parameter, and can be estimated by integrating the joint probability distribution $\mathrm{p}(\mathcal{X}, \mathcal{Y}, \theta)$ over all possible values of the random variable θ:

$$\mathrm{p}(\mathcal{X}, \mathcal{Y}) = \int \mathrm{p}(\mathcal{X}, \mathcal{Y}|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})\boldsymbol{d\theta} \ . \tag{3.4}$$

The latter consideration implies that the problem of finding an estimate for the model parameters consists of finding an estimate of the posterior distribution of the parameters $\mathrm{p}(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y})$, which is proportional to the the product of the prior distribution $\mathrm{p}(\boldsymbol{\theta})$ and the likelihood function $\mathrm{p}(\mathcal{X}, \mathcal{Y}|\boldsymbol{\theta})$:

$$\mathrm{p}(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y}) \propto \mathrm{p}(\mathcal{X}, \mathcal{Y}|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}). \tag{3.5}$$

It is interesting to note that the likelihood function $\mathrm{p}(\mathcal{X}, \mathcal{Y}|\boldsymbol{\theta})$ plays a key role in the estimation of the parameter posterior probability.

The algorithms for parameter estimation can be classified into three main

estimation approaches of increasing complexity [6]:

Maximum Likelihood (**ML**). In this *frequentistic approach* the parameters of a model are fixed but have unknown values. Since they are not random variables, the estimation of the posterior distribution is simply proportional to the likelihood function $\mathrm{p}(\mathcal{X}, \mathcal{Y}|\boldsymbol{\theta})$. The ML approach aims to find the set of parameters θ_{ML} that maximise the probability of the data given the parameters:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\mathcal{X}, \mathcal{Y}|\boldsymbol{\theta}) \ . \tag{3.6}$$

This corresponds to finding the unbiased estimators of the parameters. For example if our generative model is normally distributed $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, the mean $\boldsymbol{\mu}$ and the variance Σ correspond to the sample mean and the sample covariance of the training data [25].

Maximum A Posteriori (**MAP**). In this approach parameters are treated as random variables governed by appropriate prior distributions, as in Equation (3.5). As for ML, the optimal parameter values are point estimates of the parameter set $\boldsymbol{\theta} = \boldsymbol{\theta}'$, and the integral at the denominator of Equation (3.4) is simply a multiplicative constant. If these priors are chosen to be *conjugate* of the class conditionals, the posterior distribution of the parameters $\mathrm{p}(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y})$ will have the same functional form as the parameter priors. Conjugate priors are chosen according to the form of the class conditional and to the specific set of unknown parameters. For example, if we model a Gaussian distribution $\mathrm{p}(\mathcal{X}, \mathcal{Y}|\boldsymbol{\theta})$, then the prior over the mean is a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu_0}, \Sigma_0)$, the prior over the covariance matrix Σ is a Wishart distribution $\mathcal{W}(W, \boldsymbol{\nu})$, and the prior over each class prior is a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha})$. The advantage of adding prior distributions for the parameters is that they introduce some form of regularisation, and therefore favour simpler models to more complicated ones [6].

Pure Bayesian Learning. As in the MAP approach, parameters are random variables, but their optimal values are not point estimates. This approach aims at solving the estimation problem of Equation (3.4) via sampling methods such as Markov Chain Monte Carlo or via approximation techniques such as variational inference or expectation propagation [6].

One of the strengths of generative models is their inherent ability to handle labelled and unlabelled data. This can be shown by simply rewriting Equation (3.2) for labelled and unlabelled data as:

$$\begin{aligned} \mathrm{p}(\mathcal{D}) &= \mathrm{p}(\boldsymbol{\mathcal{X}}_L, \mathcal{Y}_L, \boldsymbol{\mathcal{X}}_U, \theta) \\ &= \mathrm{p}(\boldsymbol{\mathcal{X}}_L, \mathcal{Y}_L|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\mathcal{X}}_U|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}) \\ &= \mathrm{p}(\boldsymbol{\mathcal{X}}_L|\mathcal{Y}_L, \boldsymbol{\theta})\mathrm{p}(\mathcal{Y}_L|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\mathcal{X}}_U|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}) \ , \end{aligned} \tag{3.7}$$

and by taking the logarithm of the likelihood $\mathrm{p}(\mathcal{D}|\boldsymbol{\theta}) = \mathrm{p}(\mathcal{Y}_L|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\mathcal{X}}_U|\boldsymbol{\theta})$ in Equation (3.7):

$$\log \mathrm{p}(\mathcal{D}|\boldsymbol{\theta}) = \sum_{i=1}^{|\mathcal{D}_L|} \log \mathrm{p}(x_i|y_i, \boldsymbol{\theta})\mathrm{p}(y_i|\boldsymbol{\theta}) + \sum_{j=1}^{|\mathcal{D}_U|} \log \Big(\sum_{k=1}^{c} \mathrm{p}(x_j|y_k, \boldsymbol{\theta})\mathrm{p}(y_k|\boldsymbol{\theta}) \Big) \ . \tag{3.8}$$

The log-likelihood in Equation (3.8) is made of two distinct terms, the first one accounting for the labelled data only and the second one accounting for the unlabelled data only.

This property of generative models to *naturally* handle labelled and unlabelled data is of great importance, as many real problems nowadays require large quantities of labelled data to design supervised classifiers with high accuracy, and at the same time are characterised by the difficulty and cost of collecting such data. A possible answer to this dilemma would be to consider *semi-supervised* algorithms, that is, techniques which are able to learn from a small amount of labelled data together with a large amount of unlabelled data [102].

3.3 Bayesian Networks

We now focus our attention on Bayesian networks, a class of probabilistic models that provide us with a way to graphically represent existing dependencies between random variables of a joint probability distribution [69].

More specifically Bayesian networks are directed acyclic graphs $G = (N, E)$ where N denotes the set of nodes in the graph and E represents the set of edges between the nodes in the graph. Each node $N_i \in N$ corresponds to a random variable X_i and each edge $E_i \in E$ represents a dependency between two random variables. A set of S random variables $\boldsymbol{X} = \{\mathrm{X}_1, \ldots, \mathrm{X}_S\}$ can be represented as a Bayesian Network if its joint probability distribution $\mathrm{p}(\boldsymbol{X})$ factorises into a

product of probabilities

$$\mathrm{p}(\boldsymbol{X}) = \prod_{i=1}^{S} \mathrm{p}\left(X_i|\mathrm{pa}(X_i)\right) \quad , \tag{3.9}$$

where $\mathrm{pa}(X_i)$ is the set of parent variables of X_i, that is the conditioning random variables.

From a Bayesian network perspective the problem of learning a probabilistic generative model from data can be seen as a search in the space of possible networks that can be generated by adding or removing arcs between random variables. In general, this search for the network configuration that best fits the training data is done by maximising a fitness criterion. One of the most applied criteria is Minimum Description Length, a principle based on mutual information that selects the model encoding the shortest description of the data. However, the problem of learning unrestricted Bayesian networks from data is not only intractable [16], but heuristic methods are extremely computationally demanding. As a result, research has mainly focussed on restricted network topologies. As such, Naïve Bayes classifiers are an example of simple Bayesian network that despite requiring unrealistic assumptions, have been shown to compete with decision trees or neural networks [60, 24, 67].

3.3.1 Naïve Bayes Classifiers

Naïve Bayes classifiers are probabilistic generative models of the joint probability distribution $\mathrm{p}(X, Y)$ that make the assumption that features $\boldsymbol{X}$ are statistically independent from each other given the class label Y [33, 25]. This assumption implies that the joint probability distribution factorises as $\mathrm{p}(\boldsymbol{X}, Y) = \mathrm{p}(\boldsymbol{X}|Y)\mathrm{p}(\mathrm{Y})$, where the class conditional distribution factorises into the product of each class conditional

$$\mathrm{p}(X|Y) = \prod_{i=1}^{p} \mathrm{p}(X_i|Y) \ . \tag{3.10}$$

Figure 3.1 illustrates the Bayesian network associated with a Naïve Bayes classifier with 3 features X_1, X_2, X_3 and a class label Y. This graphical model clearly visualises the assumption of statistical independence of each feature given the class label Y: the directed arc between Y and every feature denotes that each feature depends on the actual observation of Y, and the absence of arcs between

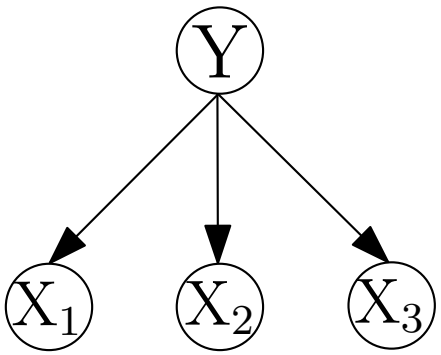


Figure 3.1: Naïve Bayes network. The feature random variables X_1, X_2 and X_3 depend on the class random variable Y but they are statistically independent from each other given the class Y.

features denotes that features are independent from each other given that the class label has been observed.

Example 1: Learning Discrete Naïve Bayes

In the case scenario of discrete features, each class conditional $\mathrm{p}(\boldsymbol{X}|Y)$ along with the class priors p(Y) are modelled by multinomial distributions. One way to estimate these distributions is by frequency counts.

For instance, let us consider a two class problem in a multi-dimensional input space, where each feature X_i can assume K different values from 1 to K, and the class Y can only take the two values ω_1 and ω_2. We are given N pattern samples, of which N_1 patterns belong to class ω_1 and N_2 patterns belong to class ω_2. The prior probability of a pattern to belong to class ω_1, i.e. $\mathrm{p}(\mathrm{Y} = \omega_1)$, is estimated as the fraction of data which is labelled as class ω_1: $\mathrm{p}(Y = \omega_1) = \frac{\mathrm{N}_1}{\mathrm{N}}$. Similarly the prior probability of a pattern to belong to class ω_2 is estimated as the fraction of the data which belongs to class ω_2: $\mathrm{p}(Y = \omega_2) = \frac{\mathrm{N}_2}{\mathrm{N}}$. We estimate the probability of a particular feature X_i to take the value k conditioned on the class label to take the value ω_j in the same way. For instance, the probability of a pattern that belongs to class ω_1 and has feature X_1 equal to k is calculated as:

$$\mathrm{p}(X_1 = k|Y = \omega_1) = \frac{\#\text{of patterns where } (\mathrm{X}_2 = k \text{ and } Y = \omega_1)}{\text{Total } \#\text{ patterns where } (\mathrm{X}_2 = k)}. \tag{3.11}$$

Once these probabilities have been calculated, Bayes' rule can be used to determine the class posterior probabilities of an unknown pattern and predict the class with maximum posterior probability.

Example 2: Learning Gaussian Naïve Bayes The Naïve assumption can be applied to continuous classification problems. For instance, a *Gaussian Naïve Bayes* is a probabilistic model where we assume each class conditional distribution to be a normal distribution. More explicitly, we make the assumption that each feature X_f, for $f = 1, \ldots, p$ is statistically independent from each other given the class, and that each class conditional distribution is Gaussian: $\mathrm{p}(X_f|Y_k, \theta) = \mathcal{N}_f(\mu_k, \sigma_k^2)$ for any class $k = \omega_1, \ldots, \omega_c$. The class prior is modelled by a Multinomial distribution $\mathrm{Mult}(\pi)$, where π are the distribution parameters constrained to $\sum_k \pi_k = 1$. In this case, the joint distribution of the model can be written as

$$\mathrm{p}(\boldsymbol{X}, Y_k) = \prod_{f=1}^{p} \mathcal{N}_f(\mu_k, \Sigma_k)\mathrm{Mult}(\pi_k)\mathrm{p}(\mu_k, \Sigma_k, \pi_k) \tag{3.12}$$

for any class for $k = \omega_1 \ldots \omega_c$.

3.3.2 Augmented Naïve Bayes

As their name suggests, Naïve Bayes classifiers are based on a very simplistic assumption; that features do not depend on each other. This is quite an unrealistic assumption that hardly ever occurs in real problems, because features measure different aspects of the same problem and the chances of an existing relationship between them are quite high.

Nevertheless, experimental results have often shown that these classifiers can often compete with classifiers which make less restrictive assumptions, such as decision trees or neural networks [60, 24]. The success of this simple but effective classifier has led researchers to try to understand the reasons why this classifier succeeds [23], and more importantly, to investigate whether relaxing the naïve assumption of Equation (3.10) might further increase classification accuracy [54, 74, 67, 33].

The assumptions of a Naïve Bayes model can be relaxed by adding dependencies between feature variables [49, 50]. An example of these *Augmented Naïve Bayes* classifiers is shown in Figure 3.2.

Augmented Naïve Bayes classifiers fall into the category of Bayesian networks where all the features contribute to the estimate of the class posterior distribution. Among these, we distinguish two main approaches to the problem of

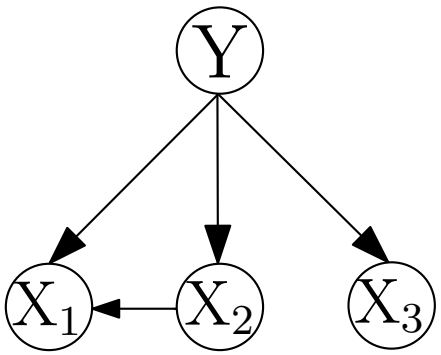


Figure 3.2: Augmented Naïve Bayes network. The feature random variables X_1, X_2 and X_3 depend on the class random variable Y. In addition, feature X_1 statistically depends on feature X_2.

learning a Bayesian Network – *distribution-based* approaches and *classification-based* approaches. The former approach has the main objective of providing an estimate for the class posterior distribution, whereas the latter approach has the simpler main objective of finding the model with highest classification accuracy.

Tree Augmented Naïve Bayes (TAN) are restricted Bayesian networks where every feature random variable depends on the class label random variable and is eventually allowed to depend on one other feature random variable. It has been shown that a distribution-based learning approach can learn the optimal structure of this type of networks in polynomial time [33, 32]. The alternative classification-based approach SuperParent (SP-TAN) performs a faster hill-climbing search over possible dependencies between features [49, 50]. The class label is the parent of all the features, whereas the feature that affects the other feature is renamed as super-parent. Experimental results [49, 50] show that this algorithm is faster and provides better classification results than distribution based algorithms.

Further restrictions to the networks structure are carried out in [89] with Super-Parent One-Dependency Estimators (SPODE), that is, augmented Naïve Bayes where each feature depends only on one common feature, namely the super-parent. The model search problem here is addressed by proposing Averaged One-Dependency Estimators (AODE), an ensemble technique where the ensemble model is the average of all possible AODEs trained on more than 30 training samples. One Dependency Estimators (ODE) are based on less restrictive assumptions than simple Naïve Bayes classifiers, but at the same time allowing at most one dependency between features guarantees better estimates of the class conditional probabilities. The issue of performing model selection rather

than model averaging on SPODEs has been further investigated in [94, 95, 96]. Recently Generalised Additive Models have been used to learn networks with higher-order dependencies between features [61].

3.4 Entropy and Mutual Information

Information theory was originally developed by Shannon to describe limits on the performance achievable by a transmission of information over a noisy channel [75]. The two key concepts in Information theory are entropy and mutual information.

The *entropy* of a random variable can be thought of as a measure of the uncertainty associated with the probability distribution of the random variable itself [17]. More specifically, for a discrete random variable X that can take $|X|$ possible values $X = x_i$ for $i = 1, \ldots, |X|$, the entropy $\mathrm{H}(X)$ is defined as[1]:

$$\mathrm{H}(X) = -\sum_{i=1}^{|X|} \mathrm{p}(x_i) \log \mathrm{p}(x_i) \ . \tag{3.13}$$

The base of the logarithm is arbitrary and defines the unity of measure of the entropy. If the logarithm is to base 2, the entropy is measured in *bits*, whereas if the logarithm is to natural base e, the entropy is measured in *nats*. For the rest of this thesis we assume to deal with logarithms to base e.

As an example, let us consider the case where X is a binary variable that can assume the two values x_1 and x_2. In this case Equation (3.13) reduces to $\mathrm{H}(X) = -\mathrm{p}(x_1) \log \mathrm{p}(x_1) - \mathrm{p}(x_2) \log \mathrm{p}(x_2)$. If X has exactly the same probability of taking any two values, i.e. $\mathrm{p}(x_1) = \mathrm{p}(x_2) = 0.5$, the entropy is $\mathrm{H}(X) = 0.69$ nats. On the other hand, if one value is more likely than the other one, as in $\mathrm{p}(x_1) = 0.8$ and $\mathrm{p}(x_2) = 0.2$ the entropy is $\mathrm{H}(X) = 0.35$ nats. If instead $\mathrm{p}(x_1) = 1$ and $\mathrm{p}(x_2) = 0$ the entropy is $\mathrm{H}(X) = 0$ nats. From these three cases we can easily see that the entropy is at its maximum when we are *totally uncertain* about the outcome of the random variable, as in the first case where the random variable has exactly the same probability of taking one of the two values. On the contrary the entropy is zero when we are *totally certain* about the outcome of the random variable, as in the third case.

The entropy of a random variable is always non negative, that is $\mathrm{H}(X) \geq 0$,

[1] For clarity of exposition we simplify our notation by denoting with x_i the fact that a random variable X takes its $i-$th value $X = x_i$, for $i = 1, \ldots, |X|$.

and it has its upper bound in the logarithm of the size of the alphabet[2] of the random variable $\mathrm{H}(X) \leq log|X|$. It is interesting to point out that the entropy of a random variable can be interpreted as the expected value of the negative logarithm of the probability distribution p(X)

$$\mathrm{H}(X) = \mathbb{E}_X \left\{ \log \frac{1}{\mathrm{p}(X)} \right\} = \mathbb{E}_X \left\{ -\log \mathrm{p}(X) \right\} \quad . \tag{3.14}$$

Another important concept is the *conditional entropy* of a random variable X conditioned on another random variable Y:

$$\mathrm{H}(Y|X) = -\sum_{i=1}^{|X|} \mathrm{p}(x_i) \sum_{j=1}^{|Y|} \mathrm{p}(y_j|x_i) \log \mathrm{p}(y_j|x_i) \tag{3.15}$$

that is a measure of the uncertainty left in Y once the random variable X has been observed, and which corresponds to the expected negative logarithm of the probability mass function p(Y|X):

$$\mathrm{H}(Y|X) = \mathbb{E}_{\mathrm{X,Y}} \left\{ -\log \mathrm{p}(\mathrm{Y}|\mathrm{X}) \right\} \quad . \tag{3.16}$$

Entropy has many properties, some of which are worth pointing out:

- The joint entropy of X and Y can be decomposed into the sum of the entropy $\mathrm{H}(X)$ and the conditional entropy $\mathrm{H}(Y|X)$: $\mathrm{H}(X,Y) = \mathrm{H}(X) + \mathrm{H}(Y|X)$. Alternatively, it is possible to swap X and Y, so that the alternative decomposition is also true: $\mathrm{H}(X,Y) = \mathrm{H}(Y) + \mathrm{H}(X|Y)$.

- The entropy of two random variables X and Y conditioned on a third random variable Z can be written as $\mathrm{H}(X,Y|Z) = \mathrm{H}(X|Z) + \mathrm{H}(Y|X,Z)$.

- The joint entropy can be generalised to more than two random variables, for which the following chain rule holds:

$$\mathrm{H}(X_1, \ldots, X_M) = \sum_{i=1}^{M} \mathrm{H}(X_i|X_{i-1}, \ldots, X_1) \quad . \tag{3.17}$$

Mutual information quantifies the information shared between two random variables [17]. For two discrete random variables X and Y it can be defined as

[2]The alphabet of a discrete random variable is the set of possible values it can take.

the difference between the entropy of Y and the conditional entropy of Y given the other random variable X:

$$\mathrm{I(X;Y)} = \mathrm{H}(Y) - \mathrm{H}(Y|X) \ , \tag{3.18}$$

and it can be thought of as the reduction in the uncertainty of the random variable Y due to the observation of another random variable X. The extension of mutual information to continuous random variables is out of the scope of this thesis; the reader might refer to [17] for a more detailed discussion.

Mutual information is a *commutative pairwise operator*, that is the order of the arguments is not important $\mathrm{I}(X;Y) = \mathrm{I}(Y;X)$. Since mutual information can be defined in terms of entropy measures, there are several equivalent ways of expressing mutual information in term of entropies:

$$\begin{aligned} \mathrm{I}(X;Y) &= \mathrm{H}(Y) - \mathrm{H}(Y|X) \\ &= \mathrm{H}(X) - \mathrm{H}(X|Y) \\ &= \mathrm{H}(Y) + \mathrm{H}(X) - \mathrm{H}(X,Y) \ . \end{aligned} \tag{3.19}$$

It is easy to show that the mutual information between two random variables X and Y corresponds to the KL divergence between the joint probability distribution $\mathrm{p}(X,Y)$ and the product of the marginal probability distributions $\mathrm{p}(X)$ and $\mathrm{p}(Y)$ [17]:

$$\begin{aligned} \mathrm{I}(X;Y) &= \sum_{y \in Y} \mathrm{p}(y) \sum_{x \in X} \mathrm{p}(x|y) \log \frac{\mathrm{p}(x,y)}{\mathrm{p}(x)\mathrm{p}(y)} \\ &= \mathrm{KL}\left(\mathrm{p}(X,Y) \parallel \mathrm{p}(X)\mathrm{p}(Y)\right). \end{aligned} \tag{3.20}$$

This latter expression tells us that mutual information measures how much the joint probability distribution $\mathrm{p}(X,Y)$ differs from the product of the marginals $\mathrm{p}(X)\mathrm{p}(Y)$. Since the joint probability distribution of two statistically random variables is the product of the marginal distributions, if the two random variables are statistically independent, by inspecting Equation (3.20) it is easy to conclude that the mutual information of two statistically independent random variables is always zero. If two random variables are statistically dependent, then their mutual information will be a non-negative quantity $\mathrm{I}(X;Y) \geq 0$, measuring *how dependent X and Y are on each other.*

The *conditional mutual information* of two random variables X_1 and X_2 being conditioned on another random variable Y is defined as:

$$\begin{aligned} \mathrm{I}(X_1; X_2|Y) &= \mathrm{H}(X_1|Y) - \mathrm{H}(X_1|X_2Y) \\ &= \sum_{y \in Y} \mathrm{p}(y) \sum_{x_1 \in \mathrm{X}_1} \sum_{x_2 \in \mathrm{X}_2} \mathrm{p}(x_1, x_2|y) \log \frac{\mathrm{p}(x_1, x_2|y)}{\mathrm{p}(x_1|y)\mathrm{p}(x_2|y)} \ , \end{aligned} \tag{3.21}$$

and it is a measure of the shared information between X_1 and X_2 once we have observed the random variable Y.

3.4.1 Why Use Mutual Information?

Although mutual information was originally developed in telecommunications for the problem of transmitting information through a noisy channel, some analogies can be drawn between a noisy channel and a classifier. The input data x can be seen as the signal encoding a message (the true class y) that a sender wants to communicate to a receiver. A classifier can be thought of as the transmission channel which decodes x through the mapping $\hat{y} = f(x)$. The message decoded by the classifier can be different than the original one: $\hat{y} \neq y$. The inability of a classifier to perfectly decode the signal x is measured through the classifier generalisation error and is due to the classifier being a mere estimate of the class posterior distribution. The generalisation error can be thought of as the channel noise, and the physical signal loss associated with signal transmission.

Studies [29, 44] have shown that information theory can be used to define an upper and a lower bound to the Bayes' error $\mathrm{p}(f(X) \neq Y)$ of any classifier f:

$$\frac{\mathrm{H}(Y) - \mathrm{I}(X;Y) - 1}{\log(|Y|)} \leq \mathrm{p}(f(X) \neq Y) \leq \frac{1}{2} \mathrm{H}(Y|X) \ . \tag{3.22}$$

The left hand inequality in Equation (3.22), which is known as Fano's inequality [29], shows that Bayes' error is minimised by increasing the mutual information shared between X and Y, the other quantities being merely constants which do not depend on the performance of the classifier. The right hand inequality in Equation (3.22), which is known as the Hellman-Raviv inequality [44], states that Bayes' error is always lower than or equal to half the conditional entropy $\mathrm{H}(Y|X)$. Figure 3.3 illustrate the relationship between the conditional entropy $\mathrm{H}(Y|X)$ and Bayes' error for a two class problem. For any value of the conditional

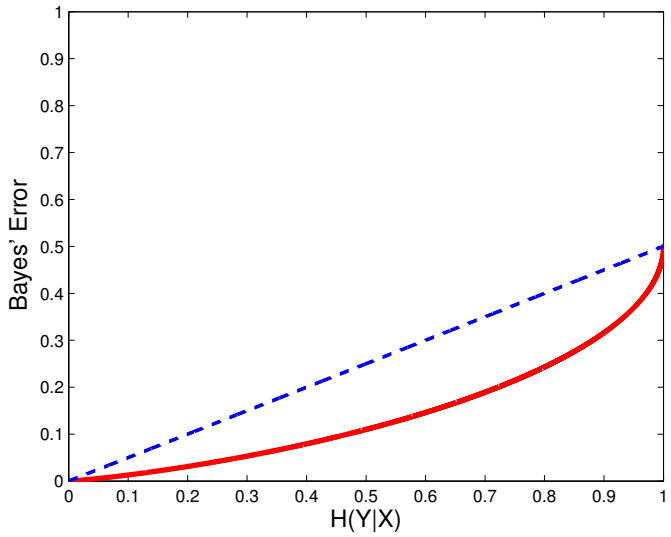


Figure 3.3: Boundaries on Bayes' error. The lower bound (red continuous line) is given by Fano's inequality, whereas the upper bound (dashed blue line) is given by Hellman-Raviv's inequality [29, 44].

entropy $\mathrm{H}(Y|X)$, Bayes' error will lie between the two curves.

The best case scenario is when the conditional entropy is zero: since there is no uncertainty left in Y once X has been observed, the classifier will make no errors. The worst case scenario from this bound perspective is when the conditional entropy is equal to $\frac{1}{2}$, since in this case the Bayes' error range is the largest. If we recall that mutual information can be written down as $\mathrm{I}(X;Y) = \mathrm{H}(Y) - \mathrm{H}(Y|X)$, we can notice that as the conditional entropy decreases, the mutual information of X and Y increases and Bayes' error decreases. These results show that mutual information can be used as a proxy to Bayes' error, and therefore it provides some indications about the best performance achievable by any classifier.

Another important consideration in support of mutual information can be given in terms of log-likelihoods. The negative log-likelihood of the class posterior distribution for a set of N data points (x_i, y_i) can be written as:

$$\begin{aligned} -\mathcal{L} = &- \frac{1}{N} \log \prod_{i=1}^{N} \mathrm{p}(y_i|x_i) \\ &- \frac{1}{N} \sum_{i=1}^{N} \log \mathrm{p}(y_i|x_i) \end{aligned} \tag{3.23}$$

In the limit of infinite data $\lim N \to \infty$, Equation (3.23) turns out to be:

$$\lim_{N\to\infty} -\frac{1}{N}\sum_{i=1}^{N} \log \mathrm{p}(y_i|x_i) = \mathrm{H}(Y|X). \tag{3.24}$$

The negative log-likelihood converges to the conditional entropy of the class random variable Y given the input random variable X. In other words, as we discussed for the properties of the conditional entropy, the conditional entropy of Y conditioned on X corresponds to the expectation over X and Y of the negative log-likelihood of the class posterior distribution $\mathrm{p}(Y|X)$:

$$\mathrm{H}(Y|X) = \mathbb{E}_{\mathrm{X,Y}}\left\{-\log \mathrm{p}(\mathrm{Y}|\mathrm{X})\right\} \tag{3.25}$$

Since the conditional entropy is equal to the mutual information of X and Y minus the entropy of Y, that is, $\mathrm{I}(X;Y) = \mathrm{H}(Y) - \mathrm{H}(Y|X)$, it is easy to see that if we minimise the negative log-likelihood we are actually minimising the conditional entropy of Y given X and therefore we are maximising the mutual information associated with the classifier.

3.4.2 Limitations of Mutual Information

Mutual information is a measure of how two random variables interact with each other. In Subsection 3.4.1 we discussed how these two random variables can be thought of as the input and the output random variables in a classification problem, and how mutual information can be used to provide bounds to the best performance achievable by a classifier. The existing relationship between accuracy and mutual information can be taken even further by noticing that maximising the negative log-likelihood of a model corresponds to maximising the mutual information associated with the same classification model.

However, mutual information can only measure pairwise interactions, and therefore it can only handle two random variables at a time. This is quite a limiting requirement, as machine learning applications usually involve multi-dimensional random variables.

For instance, let us consider the feature selection problem, which is a typical application of machine learning where we want to find the subset of features producing the classifier with highest classification accuracy. We are given p feature random variables $X_1, \ldots, X_p$, and a class random variable Y, and we want to find

the subset of features $X_{1:s}$ which maximise the mutual information of the *joint* random variable $X_{1:s}$ and the class label Y $\mathrm{I}(X_{1:s}; Y)$, as this would improve classification accuracy accordingly. This calculation requires the estimate of a high-dimensional probability distribution $\mathrm{p}(X_{1:s}, Y)$ which is hard to estimate in practice, as it would require a very large amount of data.

In the ideal case of being able to provide a good estimate of this mutual information, we would only be able to learn something about the information shared between the class label Y and the joint set of random variables as a *whole*. The mutual information $\mathrm{I}(X_{1:s}; Y)$ does not tell us anything about the relationships between single feature random variables, or between each feature and the class label.

If we now shift our attention to ensemble learning, where we deal with not one but many classifiers, and where we are interested in the interactions between *more than two* base classifiers, it is easy to notice that mutual information does not seem naturally suited to the purpose of analysing relationships between base classifiers for ensembles of more than two base classifiers.

3.5 Interaction Information

Mutual information cannot explain interactions in problems where more than two random variables are involved. This restriction has led researchers to the development of extensions of mutual information to handle more than two random variables and therefore to deal with multivariate probability distributions. As such, interaction information was proposed by McGill in 1954 [64]. Other multivariate mutual informations have been proposed [88, 97], but they are outside of the scope of this thesis.

Since interaction information is an extension of mutual information to more than two random variables, these two information theoretic measures share only some properties. One of the main differences between the two is the arity of the operation they perform. In fact, As we discussed earlier, mutual information is a binary operator $\mathrm{I}(\cdot;\cdot)$ that can take only two arguments or random variables. On the other hand, interaction information is a unary operator $\mathrm{I}(\cdot)$, that is, it takes only one argument. Another interesting difference is that whereas mutual information is always non negative, interaction information can be either a negative or positive quantity. We now formally define interaction information, and

then discuss its properties.

3.5.1 Definition

Interaction information is a unary operator that allows as its argument a *set* of random variables, which we denote with $\{\cdot, \ldots, \cdot\}$. For instance, $\mathrm{I}(\{X_1, X_2, X_3\})$ denotes the interaction information between three random variables X_1, X_2 and X_3.

This multivariate mutual information can be defined recursively from mutual information in the following way. The interaction information shared between two random variables is equivalent to the mutual information between the two random variables. The interaction information shared between three random variables X_1, X_2, X_3 is defined as the difference between the conditional interaction information of X_1 and X_2 given X_3 and the interaction information of X_1 and X_2:

$$\mathrm{I}(\{X_1, X_2, X_3\}) = \mathrm{I}(\{X_1, X_2\}|X_3) - \mathrm{I}(\{X_1, X_2\}) \ , \tag{3.26}$$

where the first term is the conditional interaction information between X_1 and X_2 given X_3 being observed, and the second terms is the interaction information of the random variables X_1 and X_2. These two interaction terms contain sets of two random variables each, and therefore they are simply defined as mutual information terms:

$$\mathrm{I}(\{X_1, X_2, X_3\}) = \mathrm{I}(X_1; X_3|X_2) - \mathrm{I}(X_1; X_3). \tag{3.27}$$

Although in his studies McGill explicitly defined interaction information for sets of up to 4 random variables, Equation (3.26) can be recursively extended to a set of M random variables $X_1 \ldots, X_M$ as the difference between the conditional interaction information of the first $M-1$ random variables given the M-th one and the interaction information of the first $M-1$ random variables:

$$\mathrm{I}(\{X_1, \ldots, X_M\}) = \mathrm{I}(\{X_1, \ldots, X_{M-1})|X_M\} - \mathrm{I}(\{X_1, \ldots, X_{M-1}\}) \tag{3.28}$$

3.5.2 The Three Random Variables Case

For the special case of three random variables, the interaction information defined in Equation (3.26), which we repeat here, simplifies to the difference of mutual

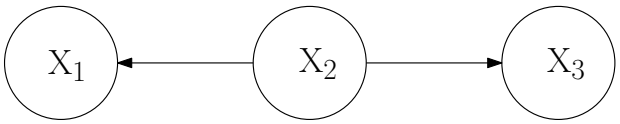


Figure 3.4: Fork configuration for three random variables

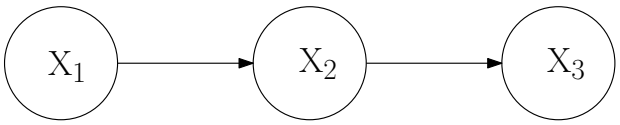


Figure 3.5: Chain configuration for three random variables

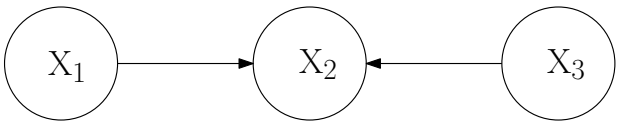


Figure 3.6: Collider configuration for three random variables

information terms:

$$\mathrm{I}(\{X_1, X_2, X_3\}) = \mathrm{I}(X_1; X_3|X_2) - \mathrm{I}(X_1; X_3). \tag{3.29}$$

This result shows that a three-wise interaction between three random variables simplifies to pairwise interactions between the same random variables. Since mutual information is a measure of mutual dependency between two random variables, for the case of three random variables we are able to relate interaction information to the statistical dependencies associated with these random variables.

Bayesian Networks represent the suitable candidates for studying pairwise dependencies. Without this reduction to pairwise terms, Bayesian Networks would not be able to capture higher order interactions between random variables. Given three random variables, there are only three possible dependency scenarios that do occur in a Bayesian Network:

- *Fork* configuration (as in Figure 3.4), where two random variables are conditioned on the third one.

- *Chain* configuration (as in Figure 3.5), where each variable is dependent on its predecessor.

- *Collider* configuration (as in Figure 3.6), where one random variable depends on the other two.

Among these configurations, we distinguish between *Markov chains* (such as forks and chains) and non Markov chains[3] (such as colliders). An interesting result here is that the interaction information of a Markov chain is always non positive [17]. Therefore, the sign of interaction information provides an indication of the inter-dependencies between random variables.

3.5.3 Properties

Since interaction information has been recursively derived from mutual information, it shares only some of its original properties. For instance, interaction information *preserves the mutual information property of being permutation invariant* to the random variables. As a result, there are different ways of expressing the interaction information between random variables. On the other hand, a distinctive feature of interaction information is that *its value does not restrict to non negative values*, but it depends on the existing statistical dependencies between random variables. We discuss these two properties in detail for the special case of three random variables.

Interaction Information can be negative

The interaction information $\mathrm{I}(\{X_1, X_2, X_3\})$ between three random variables X_1, X_2 and X_3 is defined as the difference between the conditional mutual information of any two random variables given the third one, and the mutual information between the same two random variables, as given in Equation (3.27). Since it is the difference between two positive quantities, it can either assume positive or negative values.

A *corollary* to the data process inequality states that if X_1, X_3, X_2 form a Markov chain in that order, then the mutual information of X_1 and X_3 conditioned on X_2 is lower or equal to the mutual information of X_1 and X_3: $\mathrm{I}(X_1; X_3|X_2) \leq \mathrm{I}(X_1; X_3)$. This quantity is actually the interaction information as measured in Equation (3.27). Therefore, this corollary implies that *the interaction information of a Markov chain is always non positive* [17]. The same

[3]Three random variables X, Y, Z form a Markov chain in that order if and only if $\mathrm{p}(X, Y, Z) = \mathrm{p}(X|Y)\mathrm{p}(Z|Y)\mathrm{p}(Y)$, that is if and only if X and Z are conditionally independent given Y [17].

corollary also implies that if the interaction information is always positive, then X_1, X_2, X_3 do not form a Markov chain (i.e. they form a collider chain).

We conclude that for a Bayesian Network of three random variables, the sign of interaction information is a consequence of the actual statistical dependencies between random variables. The positivity of interaction information is an indication of a non Markov chain network, whereas Markov chains lead to non positive interaction information.

This is an interesting result, as interaction information could be used in model selection techniques to decide which model is more likely to fit the data. As we have already explained in previous chapters, the general problem of probabilistic generative model selection is of great importance, as the choice of the underlying model strongly affects classification performance.

Permutation Invariance

Like mutual information, interaction information has the drawback of being permutation invariant with respect to the random variables in the set. For M random variables, there are M ways of calculating interaction information that are not distinguishable from one another.

We explain this property for three random variables X_1, X_2, and X_3. In this case, there are three different ways of conditioning two of these random variables on the third one, and for the permutation invariance property, there are three possible indistinguishable ways to calculate interaction information:

$$\begin{aligned} \mathrm{I}(\{X_1, X_2, X_3\}) &= \mathrm{I}(X_1; X_3|X_2) - \mathrm{I}(X_1; X_3) \\ &= \mathrm{I}(X_1; X_2|X_3) - \mathrm{I}(X_1; X_2) \\ &= \mathrm{I}(X_2; X_3|X_1) - \mathrm{I}(X_2; X_3) \end{aligned}$$

This property implies that although the sign of interaction information can tell us if the three random variables do form or do not form a Markov chain, it cannot tell us which is the conditioning random variable.

As an example, if we measure the interaction information of three random variables to be positive, then it follows that the three variables form a collider configuration, but the sign does not specify which one is being conditioned on the other two. This property is a direct consequence of the inability of mutual information to capture the direction of the dependency. In model terms, there

are three possible collider configurations that might correspond to a positive interaction information.

3.5.4 Relationship with Mutual Information

In a recent paper Brown [13] shows that given a set of M random variables $S = \{X_1, \ldots, X_M\}$ and another random variable Y, the mutual information $\mathrm{I}(X_{1:M}; Y)$ of a $(M+1)$ dimensional distribution $\mathrm{p}(X_1, \ldots, X_M, Y)$ can be expressed as a combination of interaction information terms:

$$\mathrm{I}(X_{1:M}; Y) = \sum_{T \subseteq S} \mathrm{I}(\{T \cup Y\}) \qquad |T| \geq 1 \ , \tag{3.30}$$

where the sum is over all possible subsets T of S. Equation (3.30) demonstrates that the mutual information between a joint random variable $X_{1:M}$ and another random variable Y decomposes into a finite number of interaction terms. As an example, for a joint random variable $X_{1:3}$ and Y, Equation (3.30) can be written as the sum of all possible *first order* interaction information terms, *second order* interaction information terms and *third order* interaction information terms:

$$\begin{aligned} \mathrm{I}(X_{1:3}; Y) \ &= \mathrm{I}(\{X_1, Y\}) + \mathrm{I}(\{X_2, Y\}) + \mathrm{I}(\{X_3, Y\}) \\ &\quad + \mathrm{I}(\{X_1, X_2, Y\}) + \mathrm{I}(\{X_1, X_3, Y\}) + \mathrm{I}(\{X_2, X_3, Y\}) \\ &\quad + \mathrm{I}(\{X_1, X_2, X_3, Y\}) \ . \end{aligned} \tag{3.31}$$

The decomposition in Equation (3.30) can be used to study different orders of interactions (first, second, third, M-th order) between *any* set of random variables. However, it has to be pointed out that the higher the order of interactions, the more data is needed to provide good estimates of these interaction terms.

Brown relates the decomposition in Equation (3.30) to the trade-off between accuracy and diversity of an ensemble of base classifiers [12]. He rewrites the mutual information between a set of base classifier outputs $X_1, \ldots, X_M$ and the true class label Y as

$$\mathrm{I}(X_{1:M}, Y) = \sum_{i=1}^{M} \mathrm{I}(X_i; Y) \ - \sum_{\substack{\boldsymbol{X} \subseteq S \\ |\boldsymbol{X}|=2,\ldots,M}} \mathrm{I}(\{\boldsymbol{X}\}) \ + \sum_{\substack{\boldsymbol{X} \subseteq S \\ |\boldsymbol{X}|=2,\ldots,M}} \mathrm{I}(\boldsymbol{X}|Y) \ . \tag{3.32}$$

Equation (3.32) breaks up into three separate summations of different mutual information quantities. The first term is the sum of the mutual information of each classifier with the class label. As we discussed in Subsection 3.4.1 mutual information is a proxy to classification accuracy and therefore this first summation can be interpreted as the *relevancy* of each base classifier to the true class label, as it indicates how relevant each base classifier is to the true class label, *independently* from the other classifiers. The second term is a *subtractive* sum over all possible interactions information terms $\mathrm{I}(\{\boldsymbol{X}\})$ and it can be thought of as a term which accounts for all the possible interactions between base classifier outputs, from pairwise interactions $\mathrm{I}(\{X_i, X_j\})$ to M-wise interactions $\mathrm{I}(\{X_1, \ldots, X_M\})$. This term can be interpreted as the *redundancy* between base classifier outputs, and since it subtracts the amount of information shared between base classifiers, it can be thought of as a measure of diversity between base classifier outputs. The last term is the sum over all possible class conditional interaction information terms $\mathrm{I}(\{\boldsymbol{X}|Y\})$. In line with the previous term, this can be thought of as *conditional redundancy*, but differently from redundancy it is an additive term. The conditional redundancy does not have a matching quantity in the accuracy/diversity trade-off described in Section 2.4. Overall we can refer to the sum of these two terms as diversity in an information theoretic sense. Equation (3.32) shows that in order to get a set of base classifiers with high mutual information, each base classifier should have high relevancy, low redundancy and high pairwise redundancy. This decomposition shows once more that there exists a trade-off between accuracy and diversity. Furthermore, this interpretation of the trade-off between accuracy and diversity in terms of interaction information between base classifier outputs seems to indicate that interactions occur at multiple levels of interactions between classifiers, and goes beyond the traditional pairwise analysis in terms of bias, variance and covariance.

3.6 Chapter Summary

In this chapter we introduced probabilistic models, that is, learning models that directly minimise the negative log-likelihood loss function rather than the 0/1 loss functions. As such, Bayesian networks are probabilistic graphical models that represent statistical dependencies between random variables of a probability distribution [68], and are going to be object of investigation throughout this

thesis. We introduce mutual information as a natural way to quantify statistical dependencies between random variables [75], and we point out that the main limitation of mutual information is that it can only measure dependencies between pairs of random variables. On the contrary, interaction information [64] can be used to understand interactions between any number of random variables. This makes interaction information a suitable candidate for understanding relationships between base classifiers.

We have now introduced the background which is necessary for the understanding of our thesis contributions. In the next chapter we propose an empirical investigation of traditional ensemble techniques applied to Naïve Bayes classifiers. Our objective is to understand what type of diversity is required by low variance probabilistic models to succeed in ensemble approaches. In Chapter 5, we embark on an empirical investigation of whether interaction information can be used to understand the trade-off between ensemble accuracy and diversity. Finally, in Chapter 6, we apply interaction information properties to build ensembles of averaged Bayesian networks in a more efficient way.

Chapter 4

Diversity in Naïve Bayes Ensembles

In Chapter 2 we discussed the link between ensemble methods and the bias-variance decompositions of the classification error [86]. According to the Tumer & Ghosh framework, base classifiers should have low bias and high variance, as linear combiners mainly reduce the variance of the ensemble classification error [34]. In Chapter 3 we shifted our attention to Bayesian networks, that is, models which minimise negative log-likelihood loss functions rather than error loss functions. In this chapter we study *model diversity* as opposed to *error diversity*. Since error diversity is an immediate consequence of model diversity, our goal is to understand whether it is possible to generate diverse classifiers by looking at model diversity rather than error diversity. Parametric probabilistic models are particularly suited to this purpose, as they *explicitly select the model bias*. In fact, diversity in probabilistic models can occur at two different levels: it can be parametric or it can also be structural. The question we try to address in this chapter is the following one: *is parametric diversity sufficient to build accurate and diverse ensembles of Naïve Bayes classifiers?* This research question is of great importance, as Naïve Bayes are classifiers characterised by high bias and low variance, and therefore they do not seem suited to be combined in an ensemble approach.

To this purpose we study Bagging and Random subspaces [9, 45], two ensemble techniques which are known to mainly reduce the variance of the classification error [10, 35]. Our experimental results show that parametric diversity is not sufficient to generate accurate and diverse ensembles, but on the contrary structural

diversity has a positive effect on the ensemble performance.

4.1 A Diversity Categorisation

In this thesis we analyse classifier diversity from different perspectives. It is well acknowledged that base classifiers are diverse if they make different errors. But *what is classifier diversity?*

A classifier is an algorithm that assigns class labels to unseen objects, or patterns. In order to perform this task a classifier first learns estimates of the true class posterior distributions from the training data. It then applies Bayes' rule to choose the class which scored the highest class posterior distribution estimate on the unseen pattern. This decision will also draw decision boundaries within regions of the feature space. In fact, each region will be associated with the class with highest probability estimate in the same region [25]. Within this dual probabilistic/decision boundary perspective, it makes sense to think of diverse classifiers as classifiers that have learnt *different estimates* of the true class posterior distributions, and hence *different decision boundaries* between feature regions.

If two classifiers have learnt relatively different estimates of the class posterior probabilities, they might classify the same pattern with different class labels, and therefore, these classifiers might classify different patterns correctly or incorrectly. In this thesis we refer to this diversity in taking classification decisions as *error or output diversity* or more generally, *diversity*. Error diversity is the actual result of two classifiers modelling our problem in two different ways.

Generally speaking, two classifiers are diverse if their estimates of the class posterior distributions are diverse. Many different learning algorithms have been developed for the purpose of solving classification problems [25], such as neural networks, decision trees, support vector machines or Naïve Bayes classifiers. As we discussed in Section 3.1, there are two main ways that a classifier can learn how to assign class labels to objects. It can either take a discriminative approach and directly estimate the class posterior distributions/decision regions from data, such as with neural networks and support vector machines, or it can take a generative approach, by solving the more general problem of learning an estimate of the true probability distribution $\mathrm{p}(X, Y)$ that generated the data and use Bayes' rule to learn estimates of the class posterior distributions, such as in Bayesian Networks.

The type of learning algorithm strongly affect our *perception* of what is a classifier, and how classifiers can be diverse. As an example, in *non-parametric models* such as k-nearest neighbours we perceive that two models are diverse if they draw different decision boundaries between feature regions. In *semi-parametric models* such as neural networks, we perceive that two models are diverse if for instance the model parameters are different. In other words, in addition to draw different boundaries between feature regions, we perceive that the final weights of these networks are different.

In this chapter we are concerned with the third category of *parametric models*, and in particular with parametric probabilistic classifiers. We acknowledge that two parametric probabilistic models might differ for three main different reasons.

The first—and more general reason, is that two parametric probabilistic models might not share the same dependencies between shared random variables. For instance, a model with three features X_1, X_2 and X_3 might make the assumption that all three features are statistically independent from each other, that is $\mathrm{p}(X_1, X_2, X_3) = \mathrm{p}(X_1)\mathrm{p}(X_2)\mathrm{p}(X_3)$, whereas another model with the same three features X_1, X_2 and X_3 might make the assumption that X_1 is statistically dependent on X_2 and X_3, that is $\mathrm{p}(X_1, X_2, X_3) = \mathrm{p}(X_1|X_2, X_3)\mathrm{p}(X_2, X_3)$. As a result, the estimates of the class posterior distributions will take different probability *forms* for the two models. In this case we say that classifiers are *structurally diverse*, as models have different structural dependencies between the same random variables.

The second reason is that two parametric models might share the same random variable dependencies, but the actual random variables in the models are different. For instance, a classifier might decide to model the problem at hand with three features X_1, X_2 and X_3, and make the assumption that all three features are statistically independent from each other, that is $\mathrm{p}(X_1, X_2, X_3) = \mathrm{p}(X_1)\mathrm{p}(X_2)\mathrm{p}(X_3)$, whereas another classifier might decide to model the same problem with three different features X_3, X_4 and X_5, and make the assumption that all these three features are statistically independent from each other, that is $\mathrm{p}(X_3, X_4, X_5) = \mathrm{p}(X_3)\mathrm{p}(X_4)\mathrm{p}(X_5)$. As a result, the estimates of the class posterior distributions will take the same probability *form* in the two cases, but the random variables of interest will be different. In this case we say that classifiers are *feature-diverse*, as they share the same model structure on different features.

The third reason is that two parametric models might share the same dependencies and the same features, but the parameters of each model might be different point estimates. For instance one model might have learnt a Gaussian distribution with mean $\mu = 1$ and identity covariance matrix as an estimate for the class posterior distributions, whereas another model might have learnt a Gaussian probability distribution with $\mu = 2$ and identity covariance matrix for the class posterior distributions. In this case we say that models are *parametrically diverse*. This situation might arise when models are trained on different datasets, or when their parameters are initialised in a different way, or with different values.

Therefore, in parametric probabilistic models we can clearly identify three types or *levels* of model diversity, that is structure, feature and parametric diversity. These levels of diversity usually combine together, as for instance models might have different features and different statistical dependencies.

To date there is no unique way to measure diversity in classifier ensembles, not only because a bias-variance-covariance decomposition of the classification error is not unique [48], but also because the source of diversity can be extensive, even within the same family of learning algorithms. Since there is no acknowledged way of measuring diversity, in this thesis we look at qualitative ways of recognising that two or more classifiers are diverse. To the purpose of providing a better understanding of the concept of model diversity, in this chapter we investigate whether different types of model diversity in parametric models can generate diverse classifiers. In order to understand what type of model diversity can generate error diversity in ensembles of parametric probabilistic models, we compare the performance of different classifier ensembles with the performance of a single classifier approach, and we consider any improvement of the ensemble generalisation error as we increase the number of base classifiers in the ensemble as a way to recognise classifier diversity.

4.2 Why Combine Probabilistic Models?

In Chapter 2 we discussed traditional viewpoints on ensemble diversity. One of the possible explanations for the success of ensemble techniques is based on the bias-variance decomposition of the classification error. Within this context, the Tumer & Ghosh framework shows that the performance of a linearly combined ensemble depends on the correlation between base classifiers [86]. This framework

for the analysis of the classification error suggests that we should linearly combine classifiers with low bias and high variance if we want to reduce the variance of the classification error [34]. Another possible explanation for the success of ensemble techniques is given in terms of search through the space of models that can be learnt within an ensemble. In a seminal paper [21] Dietterich suggests that an ensemble approach might *extend* the space of learnable models $\mathcal{H}$ and achieve a better estimate of the true model than the single classifier approach.

These two motivations anticipate that diversity between base classifiers is fundamental for the success of ensemble techniques, as there is no gain in combining identical classifiers. However, these two viewpoints are implicitly concerned with two different types of diversity: whereas the Tumer & Ghosh framework focuses on *error diversity*, Dietterich's interpretation is more general and encompasses *model diversity*. Nevertheless, it is easy to infer that if base classifiers are different models, they make different errors.

In this chapter we study the first type of diversity, that is model diversity. To this aim, we focus on parametric probabilistic models, that is models that minimise negative log-likelihood loss functions rather than error loss functions. In particular we want to address Naïve Bayes classifiers, as these models have shown to be able to compete with unstable classifiers such as decision trees [33]. At the same time, Naïve Bayes are stable classifiers, that is, they are characterised by low variance, and as such, they do not seem to meet the requirements of ensemble learning techniques. Nevertheless, studies have shown that the stability of classifiers depends on various factors such as the size of the training set or the inability of the model to solve the classification problem [78, 1, 77]. Moreover, results have shown that AdaBoost can reduce the bias of Naïve Bayes ensembles [28], and that Random Subspace methods can be modified to produce successful ensembles of Naïve Bayes classifiers [83].

The rationale for focussing on this type of classifier is that these models have the natural advantage of *explicitly selecting the model bias*, and therefore *define the boundaries of the hypothesis space* $\mathcal{H}$. If this space is large enough to contain *enough diverse* models, it might be possible to combine them and achieve better performance than the single classifier. Therefore, probabilistic models can be used to understand the conditions under which classifier ensembles succeed in outperforming single classifier approaches. The crucial question here is *how do we generate different models which are intrinsically stable?* Probabilistic parametric

models can either be *parametrically different*, that is, they model the same dependencies between the same random variables, but they are fitted with different parameter values, or they can be also be *structurally different*, that is, they model different dependencies on the same/different random variables.

We study these two *degrees of diversity* by investigating two different ensemble methods, Bagging and Random Subspaces [9, 45]. These two methods re-sample the training data in two different ways. In fact, both these methods train each base classifier on different replicas of the training set, but while Bagging samples over the patterns keeping the feature space unchanged, Random Subspace keeps the patterns unchanged and samples over the feature space. Our anticipation is that these two methods may affect model diversity in two different ways.

4.3 Methodology

We evaluate the generalisation error of the single classifier and the generalisation error of each ensemble technique according to a $5 \times 2-$fold cross-validation. Each generalisation error e is measured as mean $\bar{e}$ and 95% confidence interval over 5 repetitions of a 2-fold cross-validation. In particular the 95% confidence interval is calculated by assuming that the statistic of interest is the proportion of error e, and that this statistic follows a binomial distribution. Given a population of size N, the 95% confidence interval for the generalisation error e is given by $e \pm z_c \frac{\sqrt{\bar{e}(1-\bar{e})}}{N}$, with critical value z_c equal to 1.96 [7].

We now describe the classifier model, the ensemble techniques and the datasets that have been evaluated in these experiments.

4.3.1 Classifier Model

For our analysis we take a Maximum A Posteriori (MAP) learning approach in which our parametric model is represented by a joint probability distribution $\mathrm{p}(\boldsymbol{X}, Y, \boldsymbol{\theta}) = \mathrm{p}(\boldsymbol{X}|Y, \boldsymbol{\theta})\mathrm{p}(Y|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of model parameters, and $\mathrm{p}(\boldsymbol{\theta})$ is the prior distribution over the set of model parameters. We choose conjugate priors to preserve the form of the parameter posterior distribution $\mathrm{p}(\boldsymbol{\theta})$ in the class posterior distribution $\mathrm{p}(\boldsymbol{\theta}|\boldsymbol{X}, Y)$. This model can learn from both labelled and unlabelled data as described in Section 3.2.

The base model we choose for this analysis is a Gaussian Naïve Bayes classifier,

i.e. a Naïve Bayes classifier with normally distributed continuous features and covariance matrix $\mathrm{p}(\boldsymbol{X}|Y) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. The Naïve assumption restricts the covariance matrix to be diagonal. For simplicity we assume a uni-variate covariance matrix, i.e. we adopt the identity matrix as the covariance matrix $\Sigma = \mathrm{I}$. Since the covariance matrix is fixed, the only unknown parameter of the class conditional distribution is the mean $\boldsymbol{\mu}$: $\mathrm{p}(\boldsymbol{X}|Y) = \mathcal{N}(\boldsymbol{\mu})$. The class prior distribution is a multinomial distribution $\mathrm{p}(Y|\pi) = \mathrm{Mult}(\pi)$. The requirement of conjugate priors corresponds to assuming a normal distribution for the mean $p(\boldsymbol{\mu}|\boldsymbol{\mu_0}) = \mathcal{N}(\boldsymbol{\mu_0})$ and a Dirichlet distribution for the class prior $\mathrm{p}(\pi|\alpha) = \mathrm{Dir}(\alpha)$. The joint model can be written as:

$$\mathrm{p}(\boldsymbol{X}, Y, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu})\mathrm{Mult}(\pi)\mathcal{N}(\boldsymbol{\mu_0})\mathrm{Dir}(\alpha), \tag{4.1}$$

and the class posterior distribution $\mathrm{p}(Y|\boldsymbol{X}, \boldsymbol{\theta})$ is then derived by applying Bayes rule to Equation (4.1).

A preliminary set of experiments showed that the best way to initialise the mean hyper parameter $\boldsymbol{\mu}_0$ is to learn it from labelled data. The class prior hyper parameter α has been set to the fixed value $\alpha = 10$ for each class label. Preliminary experiments also identified scaled conjugate gradient descent as the most efficient way to learn the model parameters.

4.3.2 Ensemble Learning Approaches

We develop two different variants of Bagging: the first one – *BaggingL*, generates replicas of the training data by sampling with replacement from labelled data only; the second one – *BaggingLU*, generates replicas of the training data by sampling with replacement from labelled and unlabelled data.

We develop a Random Subspace Method that trains each base classifier on randomly selected feature subsets of the original feature subset. Each subset contains half of the original features[1] [45]. Each feature subset is sampled without replacement from the original feature set. As a consequence, different subsets might not be disjoint, and share a certain number of features.

We combine our base models according to the simple mean combination rule, as a linear combination of Gaussian Naïve Bayes models is still a Gaussian Naïve Bayes model. We compare Bagging and RSM with a single classifier trained on

[1] rounded to the nearest integer to systematically deal with odd feature numbers.

the original training set.

4.3.3 Dataset Description

We test our model on three different datasets. Ringnorm and Uniringnorm are artificial datasets, whereas Feltwell is a real dataset:

Uniringnorm Artificial dataset that represents a 2 class problem with 20 features and 10000 patterns which have been generated from two uni-variate normal distributions. This dataset is a *model match* for our classifier model, as data belonging to class 1 has been generated from a uni-variate normal distribution $\mathcal{N}(\boldsymbol{\mu_1}, \Sigma_1)$, with zero mean $\boldsymbol{\mu_1} = [0, \ldots, 0]$ and identity covariance matrix $\Sigma_1 = \mathrm{I}$ and data from class 2 has been generated from a uni-variate normal distribution $\mathcal{N}(\boldsymbol{\mu_2}, \mathrm{I})$, with mean $\boldsymbol{\mu_2} = [a, a, ..., a]$, where $a = \frac{2}{\sqrt{20}}$ and identity covariance matrix $\Sigma_2 = \mathrm{I}$.

Ringnorm Artificial dataset that implements Breiman's Ringnorm example [10]. This is a 2 class problem with 20 features and 7400 patterns which have been generated from two multivariate normal distributions. This dataset represents a *model mismatch* problem for our classifier model, as one of the two classes has not been generated from a uni-variate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \mathrm{I})$. In fact, whereas data from class 2 has been generated from a normal distribution $\mathcal{N}(\boldsymbol{\mu_2}, \Sigma_2)$ with mean $\boldsymbol{\mu_2} = [a, a, ..., a]$, with $a = \frac{2}{\sqrt{20}}$ and identity covariance matrix $\Sigma_2 = \mathrm{I}$, data from class 1 has been generated by a normal distribution $\mathcal{N}(\boldsymbol{\mu_1}, \Sigma_1)$ with zero mean $\boldsymbol{\mu_1} = [0, \ldots, 0]$ and covariance matrix equal to four times the identity matrix $\Sigma_1 = 4\,\mathrm{I}$.

Feltwell This real dataset has been generated by selecting 5124 patterns from Feltwell, a collection of multi-sensor remote-sensing images of an agricultural area near the village of Feltwell (UK). This dataset represents a 5 class problem with 15 features. In this case the true probability distribution that originated the data is not known.

4.3.4 Research Question

In this chapter we investigate how ensemble diversity is affected by two different factors, that is the correctness of the model assumptions and the size of the labelled training set. We further extend our investigation into ensemble diversity

to a semi-supervised case scenario, where the training set consists of labelled as well as unlabelled data. Our main research question is: *What kind of model diversity is sufficient to generate error diversity in ensembles of low variance classifiers such as Naïve Bayes classifiers?* and *under which conditions can this be observed?*

4.4 Results

We now present the results obtained comparing ensembles of Naïve Bayes classifiers with a single Naïve Bayes classifier, and discuss the limitations of our experimental study.

4.4.1 Bagging

Bagging is an ensemble method that trains base classifiers on datasets generated by sampling with replacement from the original training set of a given problem. From a model diversity perspective, this ensemble technique can only generate *parametric diversity* between parametric probabilistic classifiers, assuming that we are not varying the model parameters, as in this case the randomness comes solely from the training data. For instance, parametric probabilistic models such as a Naïve Bayes classifiers will simply learn two different sets of model parameters if trained on two different replicas of the original training set. In fact, these Naïve Bayes classifiers will share the same model assumptions, but will learn different point estimates for their mean vector.

In this subsection we investigate Bagging ensembles of Naïve Bayes classifiers in an attempt to answer the following question: *Is parametric diversity sufficient to generate diverse stable classifiers?*

Figure 4.1 compares the test error of a BaggingL ensemble of 10 classifiers with the test error of a single classifier on Uniringnorm dataset as the size of the training set increases from 10 to 200 labelled patterns in the case scenario of a model match between the true model that generated the data and our classifiers. As we would expected, this figure shows that the main effect of increasing the amount of training data is to reduce the test error of the ensemble and the test error of the single classifier. This figure also shows that when the true model g belongs to space of searchable hypotheses $\mathcal{H}$ and very little training data is available, the ensemble approach shows a lower test error than the single classifier.

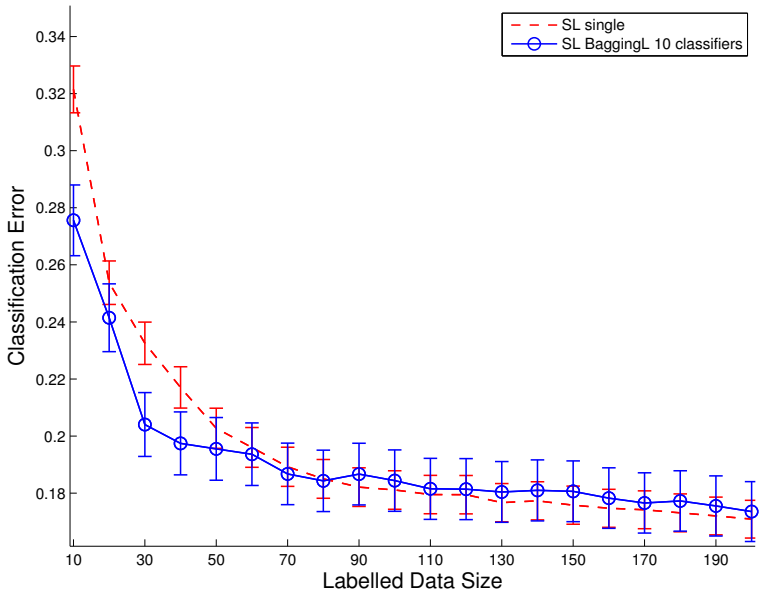
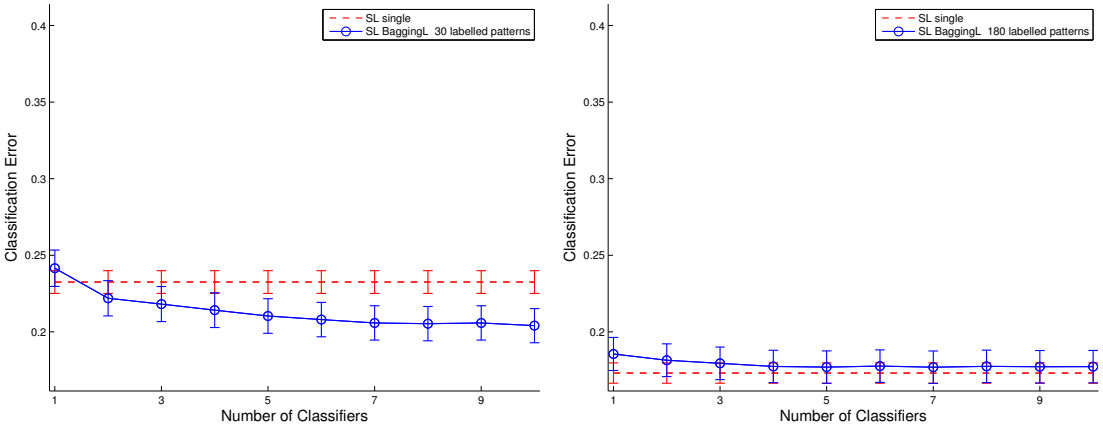


Figure 4.1: Uniringnorm dataset (*model match*), supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue line) and test error (mean and 95% confidence interval) of the single model (red dashed line) as we increase the amount of training data. Bagging outperforms the single classifier approach only when the training set is relatively small.



(a) 30 training patterns

(b) 180 training patterns

Figure 4.2: Uniringnorm dataset (*model match*), supervised learning – Test error (mean and 95% confidence interval) of the BaggingL ensemble (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the amount of base classifiers in the ensemble for a training set of 30 patterns (left) and 180 patterns (right). When the training set is small (left) the ensemble outperforms the single classifier approach, as the ensemble error decreases as we increase the number of components in the ensemble.

In this case scenario combining models with the right bias is more effective than a single model approach, as the ensemble method may explore a larger portion of the search space and therefore learn a closer approximation (i.e. closer point estimates of the parameters) to the true model that generated the data. However, as we increase the size of the training set, the single classifier shows comparable performance to the ensemble.

To understand this behaviour we look at how the ensemble size affects the test error when the training set is relatively small (i.e. 30 training patterns) and when the training set is larger (i.e. 180 training patterns). Figure 4.2a shows that with only 30 training patterns the ensemble test error decreases as we increase the number of base classifiers and eventually outperforms the single classifier. On the contrary, Figure 4.2b shows that with 180 training patterns the ensemble error does not improve as we increase the number of classifiers in the ensemble. The impact of the ensemble size on the ensemble error is illustrated for various amount of training data in Figure A.1. As we increase the amount of training data base classifiers become more accurate. However, this improvement in accuracy corresponds to a reduction in variance, which cancels out the diversity generated by the ensemble approach, as shown in Figure 4.2b: base classifiers show similar accuracies to the single classifier but no diversity, as the confidence intervals are quite small and averaging does not provide any improvement over a single base classifier. In other words, *for a model match problem under our experimental conditions Bagging Naïve Bayes is only beneficial when the training set is relatively small*. From a model diversity perspective, parametric diversity is beneficial only when base classifiers are unstable, that is when they are trained on small training sets.

In this case scenario base classifiers are *unbiased*, because they match the true model that generated the data, but they are *correlated*, as they are trained on replicas of the same training set. If we think in terms of the bias-variance-covariance decomposition of the classification error, Bagging seems to reduce the variance of classifiers trained on very small training sets. However, as the number of training data points increases, base classifiers become more stable and therefore the ensemble approach does not improve over the single classifier approach. This is in line with results obtained for linear discriminants [78], which show how certain types of linear discriminant classifiers are unstable when the training set is relatively small.

We now ask ourselves the further question: *What happens when we introduce some form of model mismatch between the base classifiers and the true probability distribution that generated the data?*

Figure 4.3 compares the test error of a BaggingL ensemble of 10 classifiers with the test error of a single classifier on Ringnorm dataset as the size of the training set increases from 10 to 200 labelled patterns in the case scenario of a model mismatch between the true model that generated the data and our classifiers. The ensemble approach is outperformed by the single classifier approach for small amounts of training data. As the size of the training data grows larger, the two methods converge to the same solution.

The impact of the ensemble size on the ensemble error is illustrated in Figure 4.4 for training sets of 30 and 180 patterns, and for various amount of training data in Figure A.3. As it is easy to inspect, increasing the number of base classifiers has no effect on the ensemble test error.

Similar results have been found for Feltwell, whose graphs are shown in Figures 4.5, 4.6 and A.5. In this case scenario base classifiers and single classifier are both *biased*. In particular, they are biased in the same way, since their mismatch with the true model is identically chosen. The only difference between the single and the base classifiers is in the training data they are provided. In fact, whereas the single classifier is provided with a 5×2-fold cross-validation repetition of the original training set, the base classifiers are provided with a dataset sampled at random with replacement from the repetition of the original training set. Since the probability of a pattern of not being selected from a training set of N patterns is $p = (1 - \frac{1}{N})$, therefore if N is large, the number of *distinct* patterns sampled will be approximately 63.2% of the patterns provided to the single classifier [27, 9]. Since Naïve Bayes classifiers update their probability distributions from distinct patterns, it follows that the reason why the single classifier outperforms the ensemble is because the former is given a larger number of distinct patterns. Moreover, base classifiers are *correlated*.

Our results show that when base classifiers are biased and correlated, and no source of diversity is within the model, parametric diversity is not sufficient to generate accurate and diverse classifiers, as the ensemble is always outperformed by the single classifier. In particular, the model bias seems to cancel out the improvement that could be gained from a small training set size. We conclude that under these experimental conditions *parametric diversity is not sufficient to*

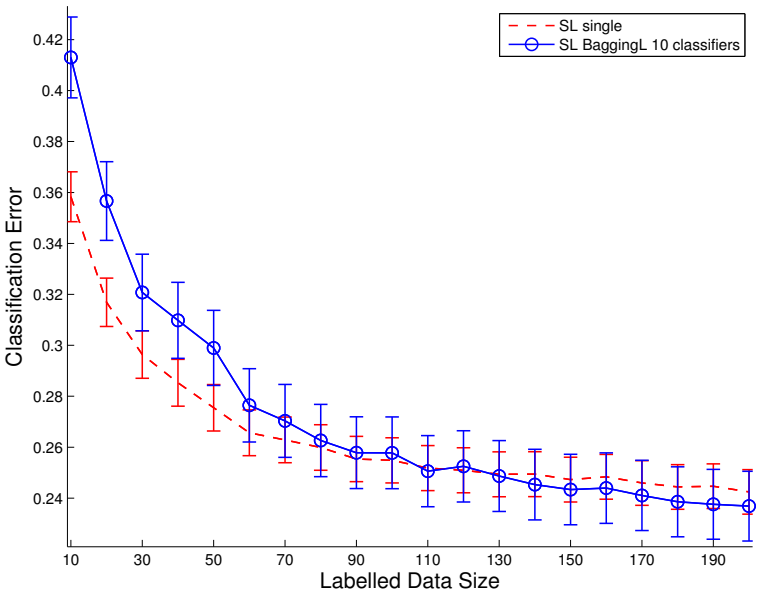
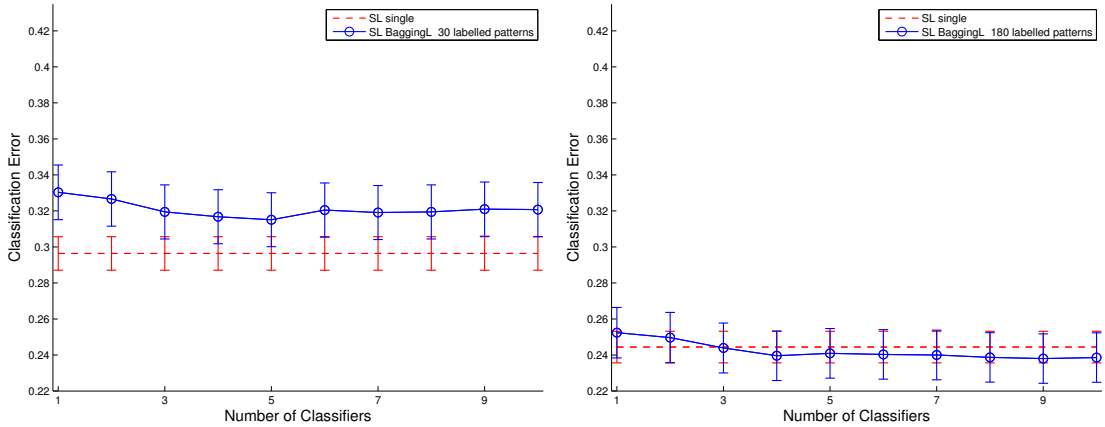


Figure 4.3: Ringnorm (*model mismatch*), supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue line) and test error (mean and 95% confidence interval) of the single model approach (red dashed line) as we increase the amount of training data from 10 to 200 patterns. The ensemble slightly outperforms the single classifier approach for increasing amounts of labelled data, but it does not learn a better hypothesis estimate than the single classifier.



(a) 30 training patterns

(b) 180 training patterns

Figure 4.4: Ringnorm (*model mismatch*), supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the amount of base classifiers in the ensemble for a training set of 30 patterns (left) and 180 patterns (right).

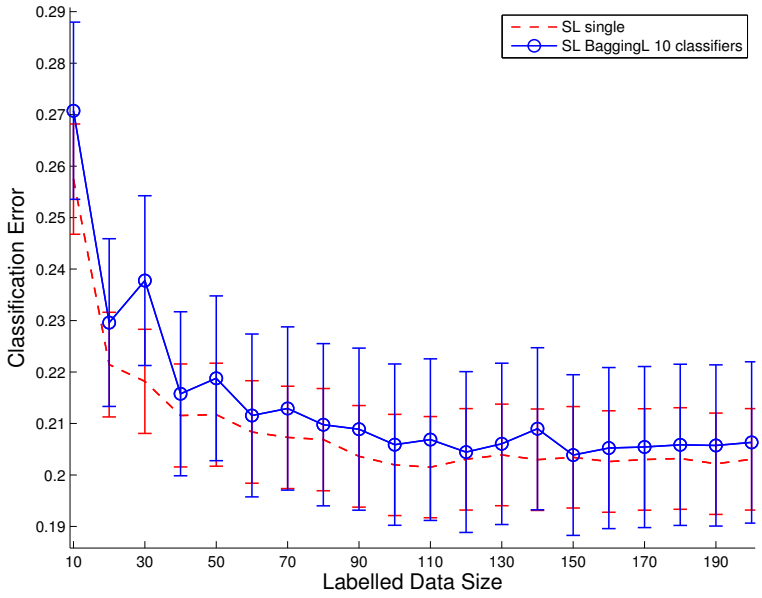


Figure 4.5: Feltwell supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue line) and test error (mean and 95% confidence interval) of the single model (red dashed line) as we increase the amount of training data from 10 to 200 patterns. BaggingL shows similar performances to the single classifier approach.

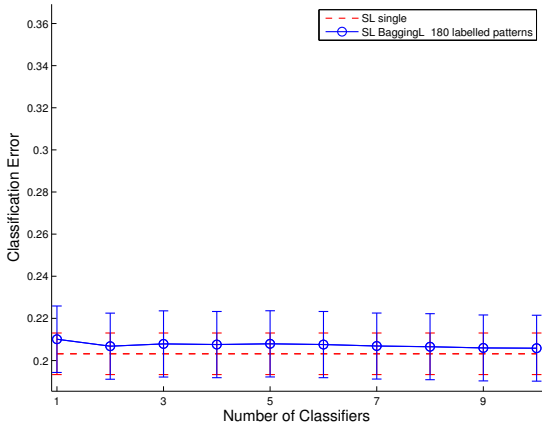


Figure 4.6: Feltwell, supervised learning – Test Error (mean and 95% confidence interval) of BaggingL test error (blue continuous line) and test error (mean and 95% confidence interval) of the single model approach (red dashed line) as we increase the amount of base classifiers from 1 to 10 for a training set of 180 patterns. The ensemble asymptotically reaches the base classifier performances as the number of base classifiers increases. This shows that the ensemble performance does not depend on the ensemble size.

generate enough diversity between stable classifiers, as the model stability reduces as we increase the size of the training set. Moreover, the model bias affects the ensemble performance. It has however to be pointed out that standard bootstrapping techniques such as Bagging require also a larger source of randomness from the training data in order to outperform the single classifier approach. Usually these bootstrap replicates (as well as the single classifier training set) are obtained from a random subset of the original training set (usually 75%). This way different repetitions of the validation procedure will share less samples, and eventually base classifiers will be less correlated.

4.4.2 Random Subspaces

The Random Subspace ensemble Method (RSM) generates diverse classifiers by training base classifiers on random subspaces of the original feature space. This ensemble technique introduces a form of *feature model diversity*, as it builds models with the same structure on different feature random variables.

Figure 4.7 compares the test error of a RSM ensemble of 10 Naïve Bayes classifiers with the test error of a single classifier on Uniringnorm dataset as the size of the training set increases from 10 to 200 labelled patterns. It has to be pointed out that whereas the former base classifiers have been trained on a random subset (50%) of the feature space, the latter has been trained on the whole feature space. This figure shows that although the single test error and the ensemble test error decrease as we increase the size of the training data, the ensemble is always outperformed by the single classifier.

Figure 4.8 illustrates how the ensemble size affects the test error when base classifiers are trained on dataset of 180 patterns. This figure shows that base classifiers are much less accurate than the single classifier, as the first base classifier is 25% inaccurate on the test error, whereas the single classifier is at least 7% more accurate than the first base classifier. In this case the ensemble error decreases with the number of averaged base classifiers, although it never reaches the performance of the single classifier. This behaviour can be observed for different amounts of training data in Figure A.2. This result seem to indicate that RSM cannot outperform the base classifier approach because base classifiers are diverse but not accurate enough to improve over the single classifier. It is worth pointing out that for a training set of 180 patterns Bagging shows the opposite behaviour (Figure 4.2b), as it generates accurate but not diverse classifiers. One

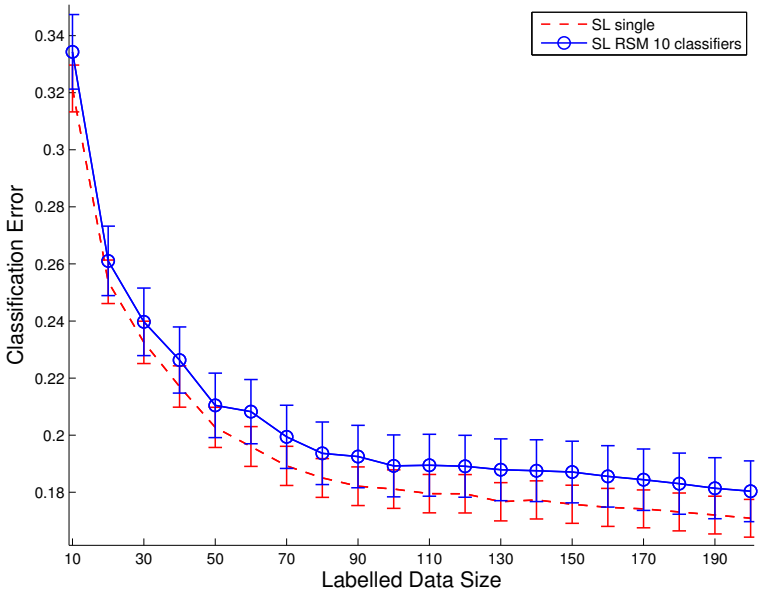


Figure 4.7: Uniringnorm (*model match*), supervised learning – Test Error (mean and 95% confidence interval) of RSM (blue line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) for increasing size of the training data. The RSM never outperforms the single classifier approach.

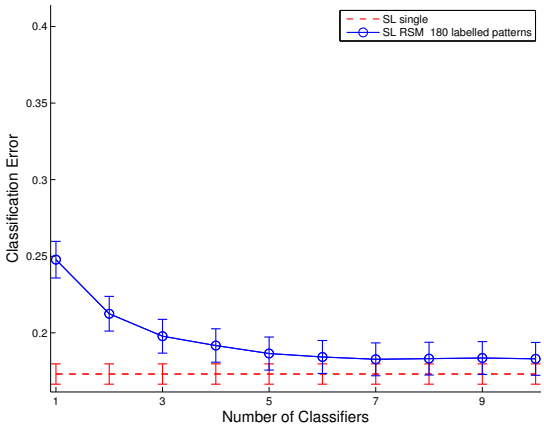


Figure 4.8: Uniringnorm (*model match*), supervised learning – Test error (mean and 95% confidence interval) of RSM (blue continuous line) and test error (mean and 95% confidence interval) of the single model approach (red dashed line) as we increase the amount of base classifiers from 1 to 10 for a training set of 180 patterns. The ensemble asymptotically reaches the performance of the single classifier as the number of base classifiers increases. This shows that the ensemble performance is positively affected by the ensemble size.

of the possible reason for this behaviour is that by generating base classifiers from random subspaces of the feature set, we are actually *introducing a model mismatch between the base classifiers and the true model which generated the data*. Therefore, the test error of base classifiers will be higher than the test error of a single model. In fact the latter matches the true model as it can learn from the whole feature set.

In this case scenario the single base classifier is *unbiased*, whereas the base classifiers are *biased* since they are trained on random subsets of the feature set and therefore learn *different* sub-models. Moreover these base classifiers are *correlated* as they are trained on overlapping feature subsets sampled from the same training set. Nevertheless, Figure 4.8 shows an interesting fact: it is possible to reduce the ensemble test error by combining low variance base classifiers. In other words, *feature diversity* seems to produce diverse base classifiers, although their high bias does not allow them to outperform the single classifier approach.

We now ask ourselves the further question: *what happens when we introduce some form of model mismatch between the base classifiers and the true probability that generated the data?*

Figure 4.9 compares the test error of RSM with the test error of the single model approach in Ringnorm dataset, for increasing amounts of training data. The Figure shows that overall, introducing some form of model mismatch between the single and base classifiers results in the ensemble method performing similarly to the single classifier for any amount of training data.

A comparison between RSM (Figure 4.9) and Bagging (Figure 4.3), shows that RSM is more accurate than Bagging on the model mismatch problem. This result seems to indicate that RSM, which introduces some extra form of model mismatch between base classifiers, is more successful than Bagging which on the contrary does not introduce any further model mismatch. This behaviour can be explained by looking at the effects of the ensemble size on RSM classification error. If we look at the ensemble test error as we increase the number of base classifiers (as depicted in Figure 4.10 for a training set of 180 labelled patterns), it is easy to conclude that the RSM error decreases as we increase the number of base classifiers, despite base classifiers showing low variance and a higher bias than the single classifier. This is true for various amounts of training data, as illustrated in Figure A.4. A further comparison between Figure 4.10 and Figure 4.4b seems to indicate that the RSM base classifiers are more diverse than the

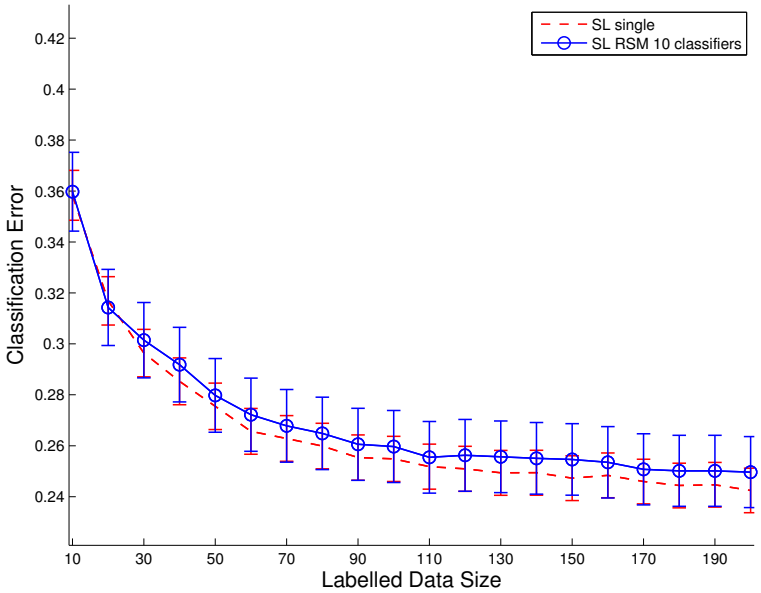


Figure 4.9: Ringnorm (*model mismatch*), supervised learning – Test error (mean and 95% confidence interval) of RSM (blue line) and test error (mean and 95% confidence interval) of the single model approach (red dashed line) for increasing size of the training data. The RSM never outperforms the single classifier approach.

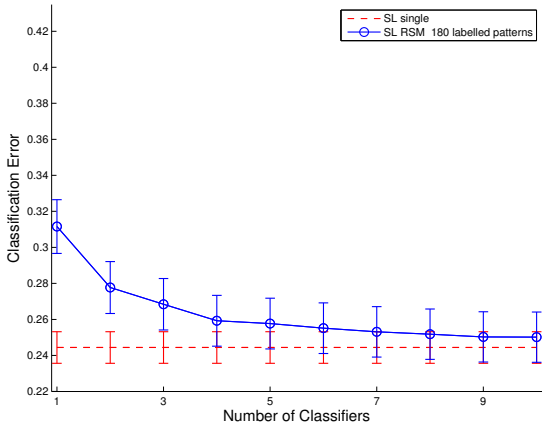


Figure 4.10: Ringnorm (*model mismatch*), supervised learning – RSM test error (blue continuous line) and single model approach test error (red dashed line) as we increase the amount of base classifiers from 1 to 10 for a training set of 180 patterns. The ensemble asymptotically reaches the performance of a single classifier as the number of base classifiers increases. This shows that the ensemble performance is positively affected by from the ensemble size.

Bagging base classifiers. In fact, whereas Bagging base classifiers show similar levels of generalisation error, a linear combination of these base classifiers do not seem to improve the generalisation error of the ensemble as the number of classifiers increases from 1 to 10. On the other hand, the generalisation error of a linear combination of the RSM base classifiers is positively affected by the size of the ensemble, as the ensemble generalisation error decreases as we increase the number of classifiers from 1 to 10. This result confirms the general idea that a model mismatch between base classifiers is beneficial to the ensemble performance. It also shows an interesting aspect of low variance classifiers: that *diversity between low-variance base classifiers lies within the structure of the model, and not within the training data.* Similar results have been found for Feltwell, the real dataset of which the true distribution is not known. These results are shown in Figures 4.11, 4.12 and A.6.

In the model mismatch case scenario base classifiers and single classifier are both *biased*. In particular, they are biased in a different way, as the single classifier models the whole feature space, whereas the base classifiers model only subsets of the feature space. Moreover, base classifiers are *correlated*. Our results seem to indicate that when base classifiers are biased and correlated, *feature diversity* can generate diverse, but not sufficiently accurate stable classifiers. This diversity becomes more and more evident as we increase the mismatch between the true model and the base models. However, RSM never outperforms the single classifier approach. A possible reason for that could be that the random selection of features might negatively affect the negative log-likelihood parameter estimation. In fact, it might be the case that the randomly selected features might not be relevant to the estimate of the class posterior distribution, whereas certain feature combinations might be more effective. In other words, a more informative way of selecting random subsets of features might be beneficial to the aim of building accurate classifiers.

We conclude that *feature diversity can generate enough diversity between stable classifiers, but base classifiers are not sufficiently accurate to improve over the single classifier approach.* Nevertheless, our experimental results point out that further diversity, in the form of dependency structure rather than feature structure might increase diversity among base classifiers.

We now look at another factor that might affect classifier diversity, i.e. the large availability of unlabelled training data.

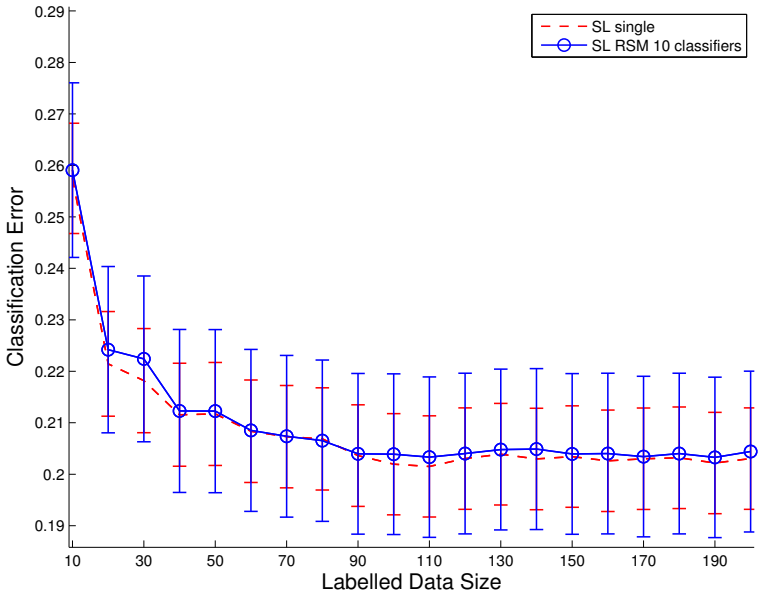


Figure 4.11: Feltwell supervised learning – Test error (mean and 95% confidence interval) of RSM (blue line) and test error (mean and 95% confidence interval) of the single model approach (red dashed line) for increasing size of the training data. The RSM shows similar accuracy to the single classifier approach.

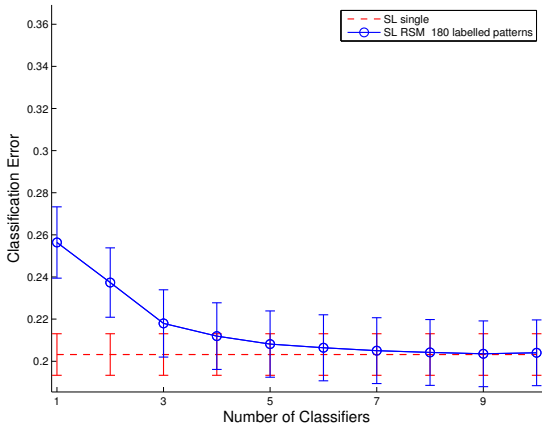


Figure 4.12: Feltwell, supervised learning – Test error (mean and 95% confidence interval) of RSM (blue continuous line) and test error (mean and 95% confidence interval) of the single model approach (red dashed line) as we increase the amount of base classifiers from 1 to 10 for a training set of 180 patterns. The ensemble asymptotically reaches the performance of the single classifier as the number of base classifiers increases. This shows that the ensemble performance is positively affected by the ensemble size.

4.4.3 Does Unlabelled Data Help?

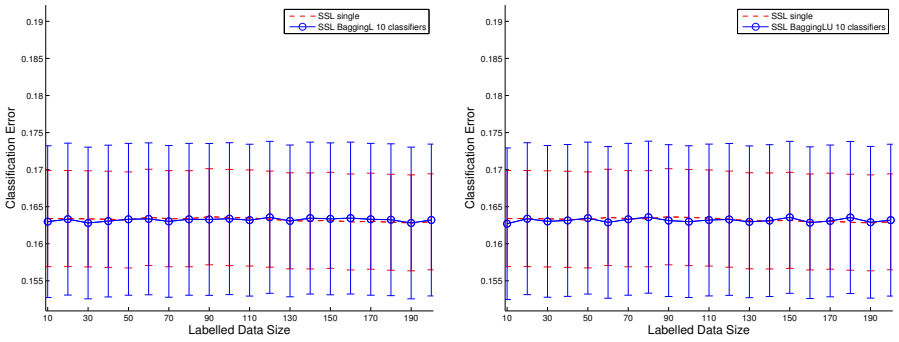
As we pointed out in Section 3.2, probabilistic models (such as Naïve Bayes classifiers) can *naturally* learn from labelled and unlabelled data. The question we want to investigate here is: *Can unlabelled data improve diversity in ensembles of stable classifiers?*

Parametric Diversity?

Figures 4.13, 4.14 and 4.15 study the effect of unlabelled data on parametric diversity, and more precisely on Bagging ensembles for the three different case scenarios of a model match, a model mismatch and a real dataset. The respective graphs that show the effects of the ensemble size on the classification accuracy are shown in Figures A.7, A.8, A.10, A.11 A.13, A.14 for different amounts of training data.

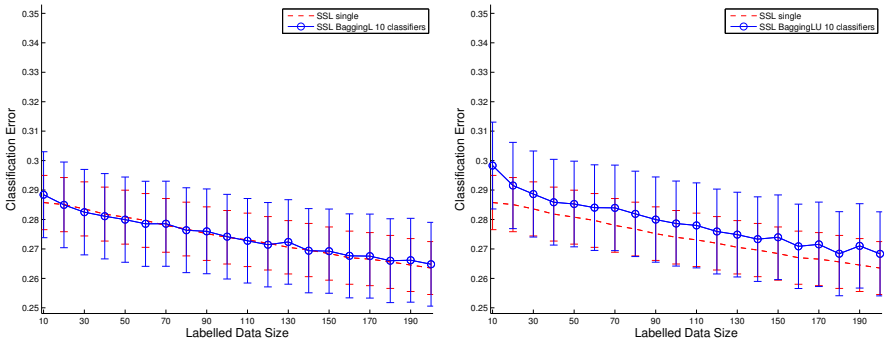
Figure 4.13 compares the test error of a Bagging ensemble to the test error of a single classifier on Uniringnorm (model match) when models are trained on a fixed amount of unlabelled data for increasing amounts of labelled data. This figure shows that Bagging labelled data (Figure 4.13a) shows identical performance to Bagging from labelled and unlabelled data (Figure 4.13b). Since classifiers are unbiased, adding unlabelled data to the training set has the effect of increasing the accuracy of base and single classifier models. However, it has also the effect of reducing the variance of these classifiers, and therefore there is no gain in averaging identical classifiers, as shown in Figure A.7 and A.8 for different amounts of training data.

Figure 4.14 compares the test error of a Bagging ensemble to the test error of a single classifier on Ringnorm (model mismatch) when models are trained on a fixed amount of unlabelled data for increasing amounts of labelled data. This figure shows that adding unlabelled data worsens both the single and the ensemble performance, and therefore adding unlabelled data has a negative effect on biased classifiers, whereas it has a positive effect if the model are unbiased. These results seem to confirm the idea that unlabelled data can be harmful when the model assumptions are wrong [66, 18]. This effect is even more evident if we sample with replacement from both labelled and unlabelled data, as in this case the bias of the ensemble test error is slightly worse than the test error of the single classifier (Figure 4.14b). If we look at the ensemble performance as we increase the number of base classifiers (Figures A.10 and A.11) we conclude that



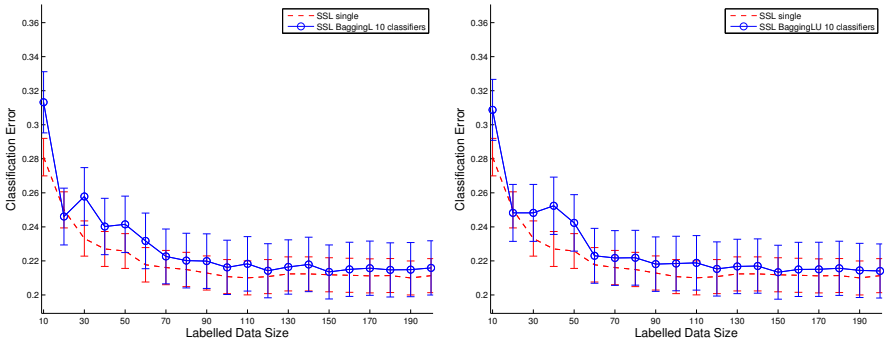
(a) BaggingL (b) BaggingLU

Figure 4.13: Uniringnorm (*model match*), semi-supervised learning – Test Error (mean and 95% confidence interval). On the left [right]: BaggingL (labelled data only) [BaggingLU (labelled and unlabelled data)] (blue line) and single classifier (red dashed line) for increasing size of the labelled training data.



(a) BaggingL (b) BaggingLU

Figure 4.14: Ringnorm (*model mismatch*), semi-supervised learning – Test Error (mean and 95% confidence interval). On the left [right]: BaggingL (labelled data only) [BaggingLU (labelled and unlabelled data)] (blue line) and single classifier (red dashed line) for increasing size of the labelled training data.



(a) BaggingL (b) BaggingLU

Figure 4.15: Feltwell, semi-supervised learning – Test Error (mean and 95% confidence interval). On the left [right]: BaggingL (labelled data only) [BaggingLU (labelled and unlabelled data)] (blue line) and single classifier (red dashed line) for increasing size of the labelled training data.

Bagging does not generate error diverse Naïve base classifiers when the model assumptions are wrong.

Finally Figure 4.15 illustrates the Bagging results for Feltwell. Again, if we look at how the ensemble size affects the ensemble error (Figures A.13 and A.14) it follows that *in ensemble techniques such as bagging, which can only produce parametric diversity, unlabelled data seem to adversely affect ensemble diversity.*

Feature Diversity?

Figures 4.16, 4.17 and 4.18 study the effect of unlabelled data on feature diversity, and more precisely on RSM ensembles for the three different case scenarios of a model match, a model mismatch and a real dataset. The respective graphs which show the effects of the size of the ensemble on the classification accuracy are shown in Figures A.9, A.12 and A.15, for different amounts of training data.

Figure 4.16 compares the test error of a RSM ensemble to the test error of a single classifier on Uniringnorm (model match) when models are trained on a fixed amount of unlabelled data for increasing amounts of labelled data. This graph shows that even if we add unlabelled data, there still exists a negative bias between the single classifier and the base classifier models. Figure A.9 shows how even with unlabelled data, base classifiers are less accurate than a single classifier, yet diversity is maintained between base classifiers.

Figure 4.17 compares the test error of a RSM ensemble to the test error of a single classifier on Ringnorm (model mismatch) when models are trained on a fixed amount of unlabelled data for increasing amounts of labelled data. This graph shows that as we increase the amount of training data both the ensemble and the single model approach test error decrease accordingly. For every amount of labelled training data, the test error of the RSM ensemble is higher than the test error of the single classifier and than the test error of any Bagging ensemble. This result is in contrast with the analogous result obtained for the supervised case scenario, where the RSM technique outperforms the Bagging technique. Since the supervised and the semi-supervised problems differ only on the training data, we conclude that unlabelled data effectively harms the ensemble performances, other than the single classifier. Figure A.12 illustrates the test error of the RSM ensemble with the test error of the single classifier as we increase the size of the ensemble from 1 to 10 classifiers. If we compare Figure A.12 with Figure A.10, we observe that the base classifiers of the RSM

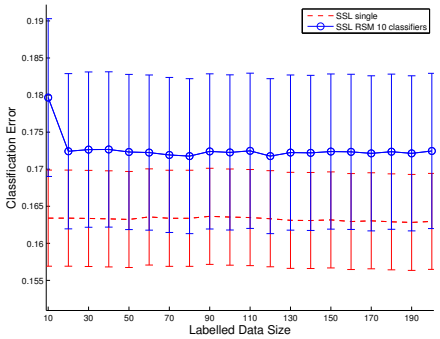


Figure 4.16: Uniringnorm (*model match*), semi-supervised learning – Test Error (mean and 95% confidence interval) of the RSM (blue line) and of the single classifier approach (red dashed line) for increasing size of the labelled training data.

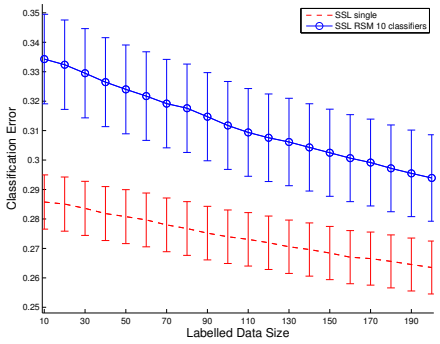


Figure 4.17: Ringnorm (*model mismatch*), semi-supervised learning – Test Error (mean and 95% confidence interval) of the RSM (blue line) and of the single classifier approach (red dashed line) for increasing size of the labelled training data.

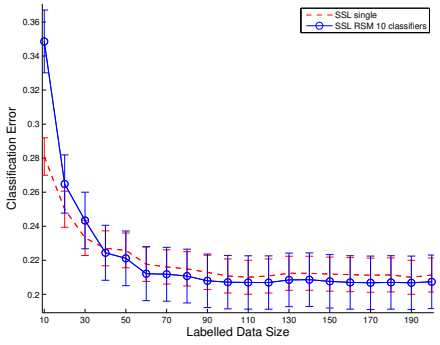


Figure 4.18: Feltwell, semi-supervised learning – Test Error (mean and 95% confidence interval) of the RSM (blue line) and of the single classifier approach (red dashed line) for increasing size of the labelled training data.

ensemble are less accurate than the base classifiers of the BaggingL ensemble, but unlike in the supervised scenario in Figure 4.10 the RSM base classifiers do not seem diverse, as the test error of the ensemble does not decrease as we increase the amount of base components. The RSM ensemble method combines simpler models than the BaggingL ensemble method since it trains each base classifier on random subspaces of the feature space. However the mismatch introduced by the RSM technique is not enough to build diverse classifiers from both labelled and unlabelled data. In fact, the variance of base models decreases with the size of the training data. This result is in line with the ideas that unlabelled data can harm the classification performances, and that increasing the amount of training data acts as a variance reducing factor between classifier models.

Figure 4.18 compares the test error of a RSM ensemble to the test error of a single classifier on Feltwell (real dataset) when models are trained on a fixed amount of unlabelled data for increasing amounts of labelled data. In this case, where the model mismatch is not known, the test error of the RSM ensemble is very similar to the test error of the single classifier. Figure A.15 illustrates the test error of the RSM ensemble with the test error of the single classifier as we increase the size of the ensemble from 1 to 10 classifiers. The model mismatch is such that base classifiers always show some level of diversity.

We conclude that *ensemble techniques such as RSM, which can produce some level of structural diversity, are adversely affected by unlabelled data in a way that the accuracy of the model is reduced, but not the diversity.*

4.5 Discussion

In this chapter we study in detail how different levels of model diversity affect error diversity in low variance classifiers such as Naïve Bayes classifiers. To this aim, we generate artificial datasets representing respectively a model match and a model mismatch case scenario and we use two different ensemble learning techniques to explicitly focus on the effects of parametric and feature diversity on the generalisation performance of the ensemble.

Our experimental results, which are summarised in Table 4.1, seem to indicate that models which are simply characterised by parametric diversity (such as in our Bagging implementations) do not seem to be enough diverse to generate error diversity between base classifiers. This result can be explained by the fact

Table 4.1: Summary of the effects of different types of model diversity on error diversity for low-variance classifiers in model match and model mismatch supervised case scenarios.

Diversity Type	Model Match	Model Mismatch
Parametric	Classifiers are diverse for small datasets only, where the ensemble error decreases with the number of base classifiers and eventually outperform the single classifier. As we increase the size of the training set, base classifiers become more stable and therefore less diverse. Parametric diversity alone does not seem to generate diverse classifiers.	Classifiers are not diverse, as the ensemble error does not decrease as we increase the number of base classifiers. The model mismatch seems to negatively affect the overall diversity of the base classifiers for any amount of training data. Parametric diversity alone does not seem to generate diverse classifiers.
Feature	The ensemble never outperforms the single classifier approach, but the ensemble error decreases as we increase the number of base classifiers. The (feature) structural model mismatch seems to preserve diversity between base classifiers.	The ensemble never outperforms the single classifier approach, but the ensemble error decreases as we increase the number of base classifiers. The structural model mismatch does not negatively affect the ensemble diversity.

that Naïve Bayes classifiers are low variance classifiers, and they therefore show some unstable behaviour only when base classifiers are trained on relatively small datasets. This behaviour is only evident when there is a model match between the classifier and the true model that generated the data. As soon as any form of model mismatch is introduced, the ensemble will tend to have the same generalisation ability as the single classifier, and the advantage of the ensemble method over the single classifier approach is cancelled out by the model mismatch. It has to be pointed out that the bootstrapping technique that we used in our experiments might have been improved, as for instance by bootstrapping on a random subset of the training set rather than the full dataset. However, the aim of the experiment was simply to assess whether under certain reproducible conditions

parametric diversity was sufficient to build error diverse classifiers.

On the other hand, building classifiers from random subsets of the feature space (as in RSM) has shown to be much more likely to generate error diverse base classifiers. More importantly, a certain level of model mismatch between base classifiers and the underlying distribution does not seem to negatively affect the ensemble generalisation error. It has been observed that despite base classifiers being low variance classifiers, the ensemble error decreases as we increase the number of base classifiers. Nevertheless, base classifiers which are trained on random subsets of the feature space have a higher bias (and therefore a larger generalisation error) than a single classifier trained on the full feature space, and although a linear combination of these models seems to reduce the generalisation error of the ensemble, the linear combination does not seem to make the classifier ensemble outperform the single classifier.

The effect of adding unlabelled data has the major effect of reducing the variance of base classifiers, as well as introducing some extra model bias which is due to the model mismatch. As a consequence, unlabelled data has the major drawback of reducing the ensemble diversity.

Overall, our experimental results for low variance classifiers such as Naïve Bayes classifiers seem to point out towards the direction of structural diversity, such as feature diversity, but more importantly, towards diversity of model dependencies.

4.6 Chapter Summary

In this chapter we embarked on an empirical study of diversity for ensembles of Naïve Bayes classifiers [100] in an attempt to study *model diversity* rather than *error diversity*.

The question we empirically tried to address in this chapter is the following one: *Is parametric diversity sufficient to build accurate and diverse ensembles of Naïve Bayes classifiers?* To this aim we investigated Bagging, a technique which builds ensembles of *parametrically* diverse Naïve Bayes classifiers and Random Subspaces, a technique which builds ensembles of Naïve Bayes classifiers on different subsets of the feature space. The latter method generates models which are *feature diverse* as it builds base models with the same dependency structure on different feature subsets. Our experimental results show that parametric

diversity as generated with our Bagging techniques is not sufficient to generate error diversity, as Bagging tends to generate accurate but stable classifiers. On the other hand, Random Subspaces generates error diverse but not sufficiently accurate classifiers to outperform the single classifier approach.

Our major conclusion is that ensembles of low variance classifiers such as Naïve Bayes classifiers benefit most from a combination of *structurally diverse* stable models rather than *parametrically diverse* stable models. Our empirical investigation seems to point out that *diversity has to lie in the structure of stable classifiers*. A key question here is *"How do we choose diverse model structure in a sensible way?"* We address this question in Chapter 6, where we show that interaction information can be used to build structurally diverse ensembles.

Chapter 5

An Interaction Information Perspective on Ensemble Learning

In Chapter 3 we introduced interaction information as a framework to understand the trade-off between ensemble accuracy and diversity in terms of interactions (correlations) between base classifier outputs [12]. Moreover, we proposed Bayesian networks as the natural candidate for graphically representing pairwise dependencies between the random variables of a joint probability distribution [32].

In this chapter we propose an empirical investigation of the following hypotheses: *Is interaction information a good measure of the trade-off between the ensemble accuracy and the base classifier diversity?*, and if so, *Can we model an ensemble as a Bayesian Network?* Since Bayesian networks can only model pairwise interactions, this question can also be rephrased as: *Is diversity a pairwise measure, that is, can we discard higher order interactions?* To answer these questions, we expand the mutual information of the joint outputs of the base classifiers and the true class label into all possible interaction terms, and we investigate these interaction terms for increasing values of the base classifier accuracy and for increasing values of the ensemble accuracy when base classifiers are combined via majority vote. We anticipate that our experimental results support the idea that higher order terms cannot be discarded, and therefore, it is not possible to model an ensemble with a Bayesian Network.

5.1 Modelling Ensembles via Bayesian Networks

Since different classifiers would give different outputs if provided with the same inputs, we can think of the outputs of base classifiers as random variables which are the result of some transformation of the input random variables.

We now make the assumption that we can abstract classifiers in terms of their outputs, that is, we can represent a classifier f solely in term of its output random variable X. In this chapter we adopt the random variable X for base classifier outputs to highlight the distinction between the estimates X of the true class random variable, and the true class random variable Y. For an ensemble of M base classifiers, we would have M random variables $X_1, \ldots, X_M$ and the true class label random variable Y. We now take this idea one level up, and we conjecture that the base classifier outputs $X_1, \ldots, X_M$ form a Bayesian network with the true class label Y.

There are two main reasons we would like to model an ensemble with a Bayesian network. First of all, this conjecture would imply that pairwise dependencies are able to model relationships between base classifiers, and that higher order interactions, which are hard to estimate in practice, can be reasonably discarded. Secondly it would be possible to apply a whole range of learning methods based on mutual information to select the base classifiers showing the most informative outputs with the true class. For instance, we could apply feature selection algorithms such as Markov Blanket discovery algorithms to identify the optimal subset of base classifiers from their outputs.

We now use the decomposition of the mutual information of a joint set of random variables in terms of multivariate information terms to investigate whether it is possible to abstract an ensemble in terms of a Bayesian network, and whether the same decomposition can be used to monitor the trade off between the ensemble accuracy and diversity.

5.2 Monitoring Ensembles via Interaction Information

Despite the notion of diversity being far from being completely understood [76, 58], an empirical investigation has shown that by forcing diversity between base classifiers with identical accuracy there exists a relationship between the ensemble

accuracy and the base classifier diversity [56].

In this chapter we study the relationship between the ensemble accuracy and the base classifier diversity from an information theoretic perspective, that is, by analysing the mutual information of the joint set of base classifier outputs and the true class label in terms of interaction information terms between the same random variables.

In particular, we consider ensembles of three base classifiers. We recall from Section 3.5 that the mutual information between the joint distribution of three base classifier output random variables X_1, X_2 and X_3 and the true class label Y can be decomposed into the sum of three levels of interactions:

$$\mathrm{I}(X_{1:3};Y) = \sum_{i=1}^{3} \mathrm{I}(X_i;Y) + \sum_{i=1}^{3}\sum_{j\neq i} \mathrm{I}(X_i, X_j, Y\}) + \mathrm{I}(\{X_1, X_2, X_3, Y\}) \ . \quad (5.1)$$

The first term of Equation (5.1) encapsulates the relevancy of each base classifier output to the true class label, and is a measure of each single base classifier accuracy. The second term, which sums up the second order interactions between pairs of base classifiers and the true class, measures the pairwise interactions between base classifier outputs. This second order interaction term can be written as the difference between two mutual information quantities, that is the second order redundancy and the second order conditional redundancy, as:

$$\sum_{i=1}^{3}\sum_{j\neq i} \mathrm{I}(\{X_i, X_j, Y\}) = \sum_{i=1}^{3}\sum_{j\neq i} \mathrm{I}(X_i; X_j|Y) - \sum_{i=1}^{3}\sum_{j\neq i} \mathrm{I}(X_i; X_j) \ . \quad (5.2)$$

The first term in the right hand of Equation (5.2) is a measure of how two classifier outputs are correlated given that we have observed the true class labels, whereas the second one is a measure of how two classifier outputs are correlated. Whereas the first order interaction is a measure of performance of *individual* classifiers, this second order interaction, as in Equation 5.2, can be thought of as a measure of the "two-way diversity".

The third term of Equation (5.1) accounts for the interactions between all the three base classifiers and the true class label. Similarly to the second order interaction, this term can be thought of as a measure of the "three-way diversity" of the ensemble. As we pointed out in the previous section, if we make the assumption that an ensemble can be modelled by a Bayesian network, we also

make the implicit assumption that third order interactions can be discarded. Therefore, if third order interactions have a significant effect on monitoring the interaction information of a joint set of random variables, it follows that ensembles can only be approximately modelled by Bayesian networks, and that mutual information cannot completely explain the ensemble behaviour.

5.3 An Empirical Study

We now empirically investigate the relationship between ensemble accuracy and base classifier diversity in terms of interaction information for the case of ensembles with three base classifiers with similar accuracies and combined according to majority vote.

The choice of the number of classifiers and the combination rule are the same as in an example that can be found in Kuncheva's book [57, Chapter 4.2.2 and 10.3.1] concerning the study of the limits of the majority vote ensemble accuracy and the study of a relationship between ensemble accuracy and base classifier diversity. However, this choice is also suitable for our study of interaction information, as by restricting the number of base classifiers to three, the decomposition stops at the third order interaction term and therefore there is no need for higher order interactions, which are hard to estimate in practice. The majority vote is a combination rule that can be associated with missing information as, in order to make a correct ensemble prediction, only $\frac{M}{2} + 1$ classifiers out of M have to be correct. The information in the remaining classifiers remains unused. Therefore, it might be the case that interaction information might be useful when dealing with this case of missing information [12].

5.3.1 Experimental Settings

We test $500,000$ ensembles of three base classifiers on 10 patterns. Base classifier outputs are combined via majority vote. We artificially generate ensembles of base classifiers with identical accuracy, ranging from ensembles of 50% accurate base classifiers to ensembles of 90% accurate base classifiers. In order to generate a representative sample of different output vectors from where to calculate interaction information, we restrict the number of patterns to 10 data points. In fact, to the extent of measuring interaction information terms, the actual value and order of the 10 outputs matter, as we are measuring interaction informations

between ordered output vectors. However, it has to be pointed out that such a small number of data points adversely affects the reliability of the interaction information estimates.

The ensemble accuracy of base classifiers with identical accuracy combined according to majority vote has upper and lower bounds which depend on the number of classifiers and on the accuracy of the base classifiers [59]. In particular the ensemble accuracy of three base classifiers which are 50% accurate ranges between 25% and 75%, whereas the ensemble accuracy of three base classifiers which are 90% accurate ranges from 85% to 100%. With only 10 data points the ensemble accuracy can only range from 30% to 100% with 10% intervals, as with only 10 test patterns it is not possible to generate a continuous space of possible accuracies but only multiples of 10%.

5.3.2 Results

We study first, second and third order interactions for (a) increasing values of base classifier accuracy and (b) for increasing values of ensemble accuracy. It has to be pointed out that Equation (5.1) decomposes the mutual information of the joint set of base classifier outputs in terms of interaction information between the same outputs, that is, it actually does not take into account the combination rule. This implies that by relating these quantities to the ensemble accuracy we are actually making the assumption that the application of the majority vote rule does not affect the results.

Figure 5.1 shows the first order interaction information (relevancy), second order interaction information and third order interaction information terms of ensembles of three base classifiers as we increase the base classifier accuracy from 50% to 90%. The bar chart shows the mean and the standard deviation values over $100,000$ ensembles. The relevancy of the base classifiers increase as we increase the accuracy of base classifiers, since the relevancy is the sum of the mutual information between each base classifier output and the class label, and since mutual information has been shown to be a proxy to classification accuracy [29, 44]. The relevancy shows a small standard deviation. On the other hand, the second order interaction term decreases considerably as we increase the accuracy of the base classifiers, going from being positive when base classifiers are like random guess or slightly better than random guess (that is, weak classifiers), to being negative for very accurate base classifiers. The third order interaction seems to be

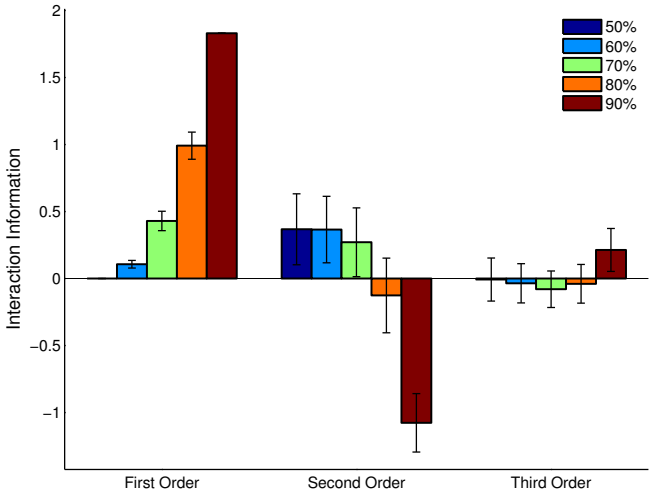


Figure 5.1: First, second and third order interaction terms (mean and standard deviation) for three base classifiers as the base classifier accuracy increases from 50% to 90%

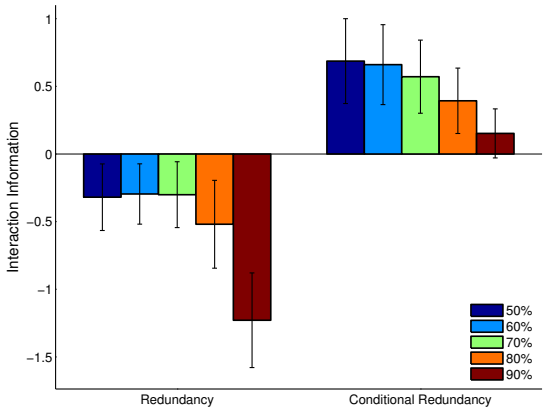


Figure 5.2: Second order redundancy and second order conditional redundancy (mean and standard deviation) for three base classifiers as the base classifier accuracy increases from 50% to 90%

negligible and close to zero for nearly all the accuracy values but the 90% accurate base classifiers. However the standard deviation overlaps among classifiers with different accuracies, and therefore no conclusions can be drawn. Figure 5.2 shows the second order interaction between base classifiers in more detail, by separating the second order redundancy from the second order conditional redundancy as described in Equation (5.2). The redundancy term increases (in absolute value) as we increase the accuracy of the base classifiers, showing that more accurate base classifiers are less diverse, whereas the conditional redundancy, which is a measure of how likely pairs of base classifiers are given the class, decreases. This behaviour makes sense if we recall from Section 3.5 that interaction information can be used as a way to explain the trade off between accuracy and diversity.

We now focus on the relationship between the ensemble accuracy and the interaction terms of Equation (5.1). Figure 5.3 shows the first order interaction information (relevancy), second order interaction information and third order interaction information terms of three base classifiers ordered according to the accuracy of the ensemble they generate when combined according to majority vote. The bars are ordered for increasing values of the ensemble accuracy, varying from 30% to 100% accurate ensembles. The relevancy of the base classifiers increases with the accuracy of the ensemble, whereas the mean of the second order interaction term seems to be constant for ensembles which are accurate up to 70%, but it then decreases and become negative for very accurate ensembles. If we observe the third order interaction, we notice that the mean of this term is actually non-zero, although the standard deviations are very high. If we focus on the second order redundancy and conditional redundancy as shown in Figure 5.4, we find that this graph shows a similar behaviour to Figure 5.2, if we restrict our attention to the same range of accuracy. The higher variability shown in Figures 5.3 and 5.4 is due to the fact that the number of sets of base classifiers contributing to each bar in these figure is variable, whereas it was constant in the bar charts of interaction information in terms of the base classifier accuracies. In particular the distribution of the ensemble accuracies for each base classifier accuracy is normally distributed around the base classifier accuracy, making it more difficult to generate ensembles whose accuracy is near to the accuracy bounds of majority vote.

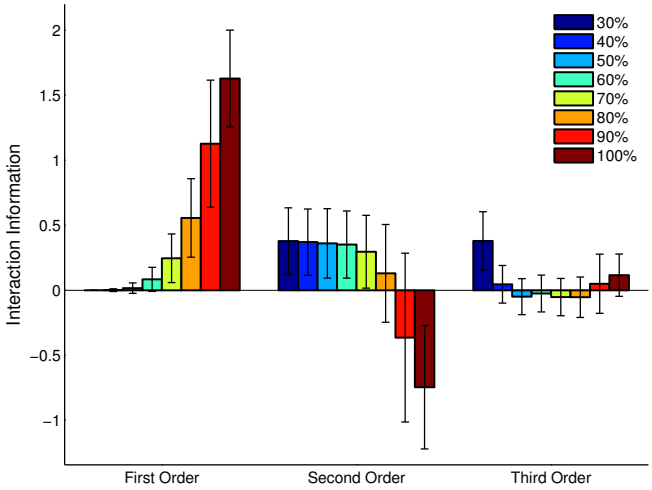


Figure 5.3: Relevancy, second order interaction information and third order interaction information as (mean and standard deviation) for an ensemble of three base classifiers combined via majority vote, as the ensemble accuracy increases from 30% to 100%

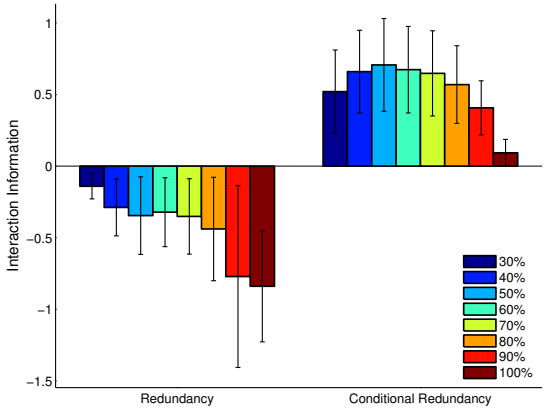


Figure 5.4: Second order redundancy and second order conditional redundancy (mean and standard deviation) for an ensemble of three base classifiers combined via majority vote, as the ensemble accuracy increases from 30% to 100%

Our experimental results seem to point out that first and second order interactions show that *interaction information can be used to monitor the trade-off between accuracy and diversity* of an ensemble of three base classifiers with identical accuracy. Moreover, they also show that non-pairwise, i.e. higher order interactions, are significant and cannot be discarded, as they contribute towards the way base classifiers interact. This result seems to confirm that *diversity plays a role at different levels in ensembles*, and seems to indicate that Bayesian networks, models that only assume pairwise interactions between random variables, can only approximately model the existing relationships between more than two base classifiers.

5.4 Chapter Summary

In this chapter we proposed an experimental analysis of the decomposition of the joint distribution of the ensemble base classifier outputs in terms of all possible levels of interactions between the base classifiers. Despite their limitations, our experimental results seemed to point out how first and second order interactions between three base classifier outputs can be used to monitor the trade-off between accuracy and diversity when base classifiers show similar accuracy. Moreover, they also highlighted that that non-pairwise interactions cannot be discarded in the understanding of ensemble diversity. This implies that Bayesian networks can only approximately model the existing interactions between base classifiers, and that Bayesian network learning algorithms such as Markov Blanket discovery algorithms or Bayesian network interpretations of combination rules can only approximately model ensemble learning approaches. In fact, *diversity seem to play a role at different levels in classifier ensembles.* In the next chapter we make use of interaction information to *build* classifier ensembles.

Chapter 6

Interaction Information for Ensemble Model Selection

In this chapter we empirically investigate two interaction information properties – sign and permutation invariance, and we show how these properties can be used to build ensembles in a more efficient way. More specifically, we try to address two main research questions: *Can we make use of the interaction information sign and permutation invariance to build more efficient classifiers?* and, *How can we make use of these classifiers in an ensemble framework?*

To this aim, we first focus our attention on two *structurally* different classes of Bayesian Networks, that is, fork augmented Naïve Bayes classifiers and collider augmented Naïve Bayes classifiers, and we empirically investigate whether (a) the sign of interaction information is a good predictor of the most accurate model and (b) whether this prediction is preserved by averaging over all possible models within the same model class.

We then propose a novel ensemble method which combines averaged augmented Naïve Bayes classifiers from random subsets of the feature space which have been chosen according to the sign of the interaction information shared between the feature subset. We study the impact of this ensemble method on the overall ensemble accuracy and on the training time by comparing this method with the corresponding accuracy based ensemble approach. Our experimental results show that not only do both methods exhibit a comparable generalisation performance, but that our method is computationally more efficient.

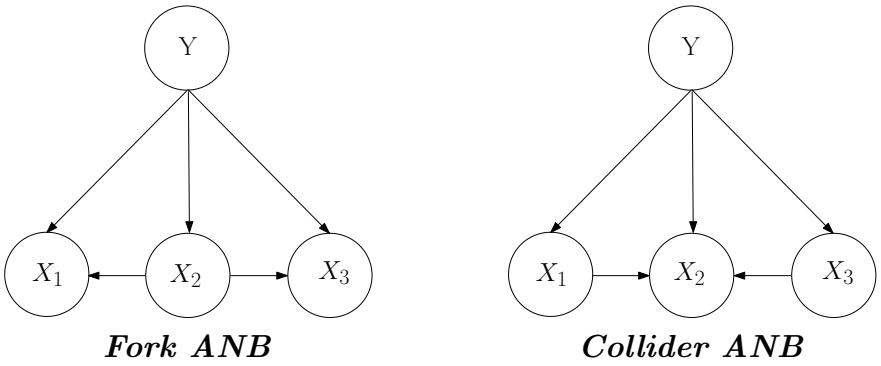
6.1 Single Model Selection

In Section 3.5.3 we studied the relationship between the Markov property of a set of three random variables and the sign of the interaction information shared between the same set of random variables. In this section we propose a possible application of this property to the problem of model selection for classification.

We restrict our analysis to augmented Naïve Bayes models for two main reasons. The first one is that Naïve Bayes models are particularly suited for classification problems, as every feature is statistically dependent on the class label, and therefore every feature contributes to the estimation of the class posterior distribution. The second one is that augmenting a Naïve model with extra dependencies might be a more realistic assumption than considering every feature to be statistically independent from each other. Since Naïve Bayes has been shown to compete with classifiers requiring less restrictive assumptions, such as decision trees, relaxing the naïve assumption might have a positive effect on classification accuracy [32].

More specifically we study two different augmented Naïve Bayes classifiers with only three features. The first model class makes the assumption that the features are connected into a fork structure (which is a Markov chain), whereas the second model class makes the assumption that the features are connected into a collider structure (which is a non Markov chain). For clarity of exposition we respectively refer to these networks as *fork Augmented Naïve Bayes* (forkANB) and *collider Augmented Naïve Bayes* (colliderANB). The probabilistic graphical models associated with these models are shown in Figure 6.1. These are two out of three possible augmented Naïve Bayes models that can be generated by assuming that the features factorise into one of the three dependency scenarios described in Section 3.5.3. Since interaction information does not let us distinguish between the two different types of Markov chain (i.e. fork and chain), we opt out for the fork configuration at this model has been widely used in augmented Naïve Bayes studies [89, 94, 95, 96]. For three feature random variables X_1, X_2, X_3, there are three ways of permuting features in a fork configuration, as well as three ways of permuting features in a collider configuration. If we restrict our space of possible models $\mathcal{H}$ to be the union of all possible forkANBs and all possible colliderANBs, the size of learnable models is $|\mathcal{H}| = 6$.

As we discussed in Chapter 3 probabilistic modelling is primarily concerned with learning the true model that generated the data at hand. If the right model



Fork ANB ***Collider ANB***

Figure 6.1: Two different ways of augmenting a Naïve Bayes classifier by assuming a *fork* (left) or a *collider* (right) dependency structure for the feature set X_1, X_2 and X_3. In the fork case the features form a Markov chain, whereas in the collider case, features do not form a Markov chain.

was known, this knowledge would have a positive effect on the generalisation ability of the model itself. However, the true model is usually not know in practice, and therefore the problem of learning a model from data relies strongly on the model assumptions that are made. In this chapter we want to take advantage of interaction information properties to exploit the prior knowledge in the the data to choose between two classes of classifier models. However, whereas the sign of interaction information can provide some insight into the dependency structure between random variables, the permutation invariance property information does not let us distinguish permutation invariant models within the same models class. For instance, if the interaction information measured between three random variables is positive, it follows that the random variables form a collider configuration, but it is not known which variable is conditioned on the other two. In other words, interaction information can only distinguish between Markov and non Markov chains, as the permutation invariance property does not let us distinguish between three inter-variable configurations.

It follows that the sign of interaction information is only *partially informative* in the context of model selection, as it cannot distinguish between models belonging to the same class. In this chapter we avoid this permutation problem by averaging over all possible permutation models within the same class. This approach has been successfully applied in [89].

6.1.1 Methodology

We carry out two preliminary studies to understand (a) to what extent the sign of interaction information is effectively associated with the class containing the most accurate model, and (b) whether this association can be extended to the averaged model from each model class, so that we can avoid the permutation invariance problem.

Relationship Between Accuracy and Interaction Information

We perform a first experiment to *verify to what extent the most accurate model is within the model class predicted by the sign of interaction information.*

To this aim, we make the hypothesis that if the interaction information measured on the training data is strictly positive, then this denotes that the features form a collider structure and therefore the colliderANB model class has to be preferred to the forkANB model class. As a direct consequence, there must exist at least one colliderANB classifier whose training error $\mathrm{err}_i^{(\mathrm{C})}$ where $i \in 1, \ldots, 3$ is lower or at least equal to the training error of any other forkANB $\mathrm{err}_j^{(\mathrm{F})}$ for any $j = 1, \ldots, 3$. If the sign of interaction information is a reliable predictor of the model class, the same result would apply to the generalisation error of those models: the same colliderANB model should have a better generalisation performance than any other forkANB model. We formalise this research question in the following hypothesis:

Hypothesis 6.1.1. *If the interaction information shared between three feature variables* $\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3$ *is strictly positive* $(\mathrm{I}\{\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3\}) > 0$ *then there must exist at least one i-th colliderANB model where* $i \in 1, \ldots, 3$ *whose generalisation error* $\mathrm{e}_i^{(\mathrm{C})}$ *is lower than or equal to any other j-th forkANB model error* $\mathrm{e}_j^{(\mathrm{F})}$*:*

$$\mathrm{e}_i^{(\mathrm{C})} \leq \mathrm{e}_j^{(\mathrm{F})} \quad \forall j = 1, \ldots, 3.$$

If the interaction information shared between the features is non positive $(\mathrm{I}\{\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3\}) \leq 0$, then this situation would account for a Markov chain relationship between the feature random variables. Since for three random variables there are two distinct Markov chains – i.e. fork and chain configuration, the sign of interaction information does not let us distinguish between the two model

classes forkANB and chainANB. Therefore, we opt to study only forkANB models and measure the association between a non positive interaction information and a forkANB outperforming any colliderANB according to the following hypothesis:

Hypothesis 6.1.2. *If the interaction information shared between three feature variables* $\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3$ *is non positive* $(\mathrm{I}\{\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3\}) \leq 0$ *then there must exist at least one i-th forkANB model where* $i \in 1, \ldots, 3$ *whose generalisation error* $\mathrm{e}_i^{(\mathrm{F})}$ *is lower than or equal to any other j-th colliderANB model error* $\mathrm{e}_j^{(\mathrm{C})}$*:*

$$\mathrm{e}_i^{(\mathrm{F})} \leq \mathrm{e}_j^{(\mathrm{C})} \quad \forall j = 1, \ldots, 3.$$

Extension to Averaged Models

We perform a second experiment to investigate whether the sign of interaction information applies to averaged ANB models obtained by averaging over all possible models within the same model class. This extension would avoid the permutation invariance problem associated with interaction information, as we would simply compare two averaged models rather than two sets of three possible models in search of the most accurate model. In other words, assuming that interaction information provides us with some indication about inter variable dependencies but cannot distinguish between all possible variable permutations within the selected configuration, we try to answer this question: *is the interaction information sign prediction robust averaging over all possible model configurations?* This question corresponds to investigating the following two hypotheses:

Hypothesis 6.1.3. *If the interaction information shared between three feature variables* $\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3$ *is strictly positive* $(\mathrm{I}\{\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3\}) > 0$ *then the averaged colliderANB model whose generalisation error* $\mathrm{e}^{(\mathrm{AC})}$ *is lower than or equal to the averaged forkANB model error* $\mathrm{e}^{(\mathrm{AF})}$*:*

$$\mathrm{e}^{(\mathrm{AC})} \leq \mathrm{e}^{(\mathrm{AF})}.$$

Similarly, for non positive interaction information:

Hypothesis 6.1.4. *If the interaction information shared between three feature variables* X_1, X_2, X_3 *is non positive* $(I\{X_1, X_2, X_3\}) \leq 0$ *then the averaged forkANB model whose generalisation error* $e^{(AF)}$ *is lower than or equal to the averaged colliderANB model error* $e^{(AC)}$*:*

$$e^{(AF)} \leq e^{(AC)}.$$

6.1.2 Experimental Results

We test Hypotheses 6.1.1 to 6.1.4 on 7 datasets taken from the UCI repository [2]. Table 6.1 illustrates some information about these datasets. Continuous datasets (segment, glass, magic4, hypothyroid and sickeuthyroid) have been discredited with a Weka [90] implementation of Fayyad multi-discretization algorithm [30].

Table 6.1: UCI Datasets Statistics – Mean and Max respectively denote the mean averaged attribute value and the maximum value that features can take.

Dataset	# Features	# Patterns	# Classes	Mean	Max
Congress	16	435	2	3	3
Glass	9	214	6	2.4	4
Magic4	10	19019	2	7.9	13
Segment	19	2310	7	9	15
Mushroom	21	8124	2	5.5	12
Hypothyroid	29	3772	4	2.6	7
Sickeuthyroid	29	3772	2	2.3	5

For every dataset but Glass and Magic4 we randomly select 500 subsets of 3 features from the set of possible features. In the case of Glass and Magic4, where there are respectively only 84 and 120 ways of selecting 3 features among all possible ones, we sample all possible subsets of 3 features.

For every selected feature subset we train and test all possible forkANB and all possible colliderANB, as well as the averaged forkANB model and the averaged colliderANB model that can be generated from these subsets according to a 5×2-fold cross-validation.

Single Model Analysis

Table 6.2 shows the mean and standard deviation percentage of the number of subsets where Hypothesis 6.1.1 is verified on the test set. We count for how many feature sets showing positive interaction there exists a colliderANB whose generalisation accuracy is higher or at least equal to any other forkANB.

Table 6.2: Percentage (mean and standard deviation) of subsets showing positive interaction information where colliderANB classifiers show lower generalisation error than any forkANB classifier.

Dataset	colliderANB
Congress	63.74 [7.03]
Glass	69.16 [10.34]
Hypothyroid	94.56 [1.28]
Magic4	19.01 [6.45]
Mushroom	77.07 [2.03]
Segment	29.85 [4.31]
Sickeuthyroid	97.84 [0.92]

It is interesting to point out that despite interaction information being a quantity measured on the training set, it is able to predict the model with highest generalisation ability on 5 out of 7 datasets. In fact, for each dataset but Magic4 and Segment, the number of occurrences of Hypothesis 6.1.1 is higher than 50% of cases. If we analyse the two cases where positive interaction information fails to predict the most accurate model class, we notice that Magic4 and Segment are quite different case scenarios. The total number of possible configurations for the Magic4 dataset is 120. Of these, interaction information is strictly positive only in the 19.75% of cases (over 10 runs: 237/1200 cases), and non negative for the remaining 80.25% of feature subspaces. The total number of possible configurations for Segment dataset is 500. Of these, interaction information is strictly positive only in the 46.84% of cases (over 10 runs: 2342/5000 cases), and non negative for the remaining 53.16% of feature subspaces.

Table 6.3 shows the mean and standard deviation percentage of the number of subsets where Hypothesis 6.1.2 is verified on the test set. For each subset of the feature space showing non positive interaction information, we quantify how many times there exists a forkANB classifier whose generalisation accuracy is higher or at least equal to any other colliderANB classifier. The experimental

Table 6.3: Percentage (mean and standard deviation) of subsets showing non positive interaction information where forkANB classifiers show lower generalisation error than any colliderANB classifier.

Dataset	forkANB
Congress	71.50 [6.35]
Glass	93.85 [2.68]
Hypothyroid	95.83 [1.10]
Magic4	79.33 [5.39]
Mushroom	79.11 [1.13]
Segment	90.58 [1.68]
Sickeuthyroid	96.89 [1.74]

results illustrated in Table 6.3 show that for every dataset, a non positive interaction information is associated with a forkANB classifier being the most accurate classifier on the test set.

Interaction information shows a higher prediction rate on the training set rather than on the test set. This can be explained by interaction information being a quantity estimated on the training data; therefore, its generalisation to the test set, which is based on the idea that both training and test are generated from the same data distribution can only be approximately correct. In general a positive value of interaction information is associated with a percentage of colliderANB classifiers outperforming forkANB classifiers on the training set which is higher than 50%. The only exception is given by Segment dataset, where a positive value of interaction information predicts a colliderANB only in 41.1% of cases on the training set. This percentage further reduces to 29.85% on the test set. On the other hand, in the Magic4 dataset this percentage drops from being 76.79 on the training set to 19.01% on the test set. This implies that the reason why interaction information does not act as a good model prediction on these two datasets is different. For Segment it seems that interaction information does not perform well on both training and test set, whereas for Magic4 it is simply a case of test set performance.

Table 6.4 illustrates the decrease in the prediction performance of interaction information measured by the difference Δ between the percentage of subsets scored in the training and test set, for both positive (Δ^{POS}) and non positive (Δ^{NEG}) interaction information. It is interesting to note that for non positive interaction information subsets, the only datasets where we observe an increase in

Table 6.4: $\Delta = \text{Train}_{\%} - \text{Test}_{\%}$

Dataset	Δ^{POS}	Δ^{NEG}
Congress	26.23	16.99
Glass	1.34	1.59
Hypothyroid	1.48	2.02
Magic4	57.78	**−31.56**
Mushroom	1.28	0.93
Segment	11.25	**−0.93**
Sickeuthyroid	1.48	2.32

percentage performance on the test set are Segment, which sees an increase from 89.65% to 90.58% and Magic4, which sees an increase from from 47.77 to 79.33%. These two datasets also show a significant prediction performance reduction for positive interaction information subsets. This result might be an indication that Segment and Magic4 datasets cannot be modelled by fork or collider augmented Naïve Bayes classifiers.

Averaged Model Analysis

We now consider classifiers that are obtained by averaging all possible models belonging to the same class, that is, models that are obtained by averaging over colliderANB classifiers or forkANB classifiers. Since our models are probabilistic and given an input x, they output the class posterior probabilities $p(Y|X)$ for $Y = 1, \ldots, \omega_c$, by averaging over possible models, we actually average over class posterior probability outputs. In summary, we compare an averaged colliderANB classifier with an averaged forkANB classifier, rather than comparing a set of three colliderANB classifiers with a set of forkANB classifiers.

Table 6.5 shows the mean and standard deviation percentage of the number of subsets where Hypothesis 6.1.3 is verified on a test set. For each feature subset showing positive interaction, we quantify how often an averaged colliderANB shows better generalisation accuracy than an averaged forkANB.

Table 6.6 summarises the mean and standard deviation percentage of the number of subsets where Hypothesis 6.1.4 is verified on the test set. For each feature subset showing non positive interaction we quantify how often an averaged forkANB classifier shows better generalisation accuracy than an averaged colliderANB classifier. Our experimental results show that the averaged models show a similar trend to the single model analysis. We conclude that the sign of

Table 6.5: Percentage (mean and standard deviation) of subsets showing positive interaction information where averaged colliderANB classifiers show lower generalisation error than averaged forkANB classifiers.

Dataset	averaged colliderANB
Congress	62.65 [5.74]
Glass	75.30 [11.56]
Hypothyroid	95.62 [0.92]
Magic4	16.04 [6.64]
Mushroom	73.43 [2.73]
Segment	36.73 [3.69]
Sickeuthyroid	97.21 [0.48]

Table 6.6: Percentage (mean and standard deviation) of subsets showing non positive interaction information where averaged forkANB classifiers show lower generalisation error than averaged colliderANB classifiers.

Dataset	averaged forkANB
congress	72.54 [7.45]
glass	93.87 [2.74]
hypothyroid	95.40 [1.46]
magic4	81.52 [5.98]
mushroom	78.67 [1.68]
segment	90.39 [3.68]
sickeuthyroid	97.06 [1.01]

interaction information can be used to choose not only between model classes, but also that its prediction ability can be used to approximate the spaces of possible models with the averaged models.

6.1.3 Discussion.

Our experimental results show that the sign of interaction information can be used not only to identify the model class with higher generalisation ability, but that the same result applies to the averaged models built from the same model classes. Nevertheless, results also show that is is not always possible to predict the model that best fits the data. In fact, Hypotheses 6.1.1 and 6.1.2 for the single ANB classifiers, as well as 6.1.3 and 6.1.4 for the averaged ANB classifiers, have been verified only up to a certain percentage. This can be explained by several factors.

First of all, this preliminary study is an attempt to understand whether the sign of interaction information measured between features provides some indication about the structure of a whole augmented Naïve Bayesian model. Hypotheses 6.1.1 and 6.1.2 do not take into consideration the fact that features are also dependent on the class label, and that therefore the interaction information of an augmented Naïve Bayes takes into account the class label Y, as well as the features X_1, X_2, X_3:

$$\mathrm{I}(\{X_1, X_2, X_3, Y\}) = \mathrm{I}(\{X_1, X_2, X_3\}|Y) - \mathrm{I}(\{X_1, X_2, X_3\}) \ . \tag{6.1}$$

On the contrary, our analysis takes into account only a portion of Equation 6.1, that is $\mathrm{I}(\{X_1, X_2, X_3\})$, and makes the assumption that the conditional interaction information $\mathrm{I}(\{X_1, X_2, X_3\}|Y)$ is negligible. It would be interesting to verify whether this implicit approximation is true, since experimental results seem to support our hypotheses.

Secondly, we are making the assumption that our data can be modelled by one of our models, either a colliderANB or a forkANB, whereas in practice, it might be the case that the data cannot be modelled by either of them, and that we are incurring a model mismatch in both cases.

Finally, Hypotheses 6.1.1 to 6.1.4 make the implicit assumption that we are given a non limited amount of data to estimate the model parameters as well as the interaction information, whereas in practice, we can only deal with finite

datasets of which we do not know the underlying distribution.

All these assumptions must be taken into account when discussing our experimental results, but overall we can conclude that there is some empirical evidence towards the utility of interaction information to predict the most accurate model class. In the next section, we make use of this property in building effective ensembles of averaged Bayesian networks.

6.2 A Novel Ensemble Approach

In Section 6.1 we studied ways of incorporating the prior knowledge provided by interaction information into augmented Naïve Bayes models of only three features. This restriction to three features makes the properties of interaction information of no practical use for single classifier approaches, as in real classification problems the number of features is usually larger. Nevertheless, this disadvantage can be turned into an advantage if paired with an ensemble learning approach like the Random Subspace method [45]. As we discussed in Chapter 4, RSM seeks diversity between base classifiers by training *different* classifiers on *different* random subsets of the feature space, the only requirement being to use *weak* base classifiers. Usually all base classifiers belong to the same model family, such as decision trees or neural networks, but as we have pointed out in Chapter 4, simple Bayesian networks such as naïve Bayes are *stable* classifiers, hence the need for combining *hybrid*, that is *structurally diverse*, Bayesian networks.

In this section we address the second research question of *how can we make use of classifiers based on interaction information properties to build more efficient ensemble methods?* To this aim, we propose **irsADE**, a *novel* ensemble method which effectively uses interaction information properties to build base classifiers. The acronym **irsADE** stands for **i**nteraction information **r**andom **s**ubspace **A**veraged **D**ependency **E**stimators. This approach combines hybrid base classifier models on random feature subsets of size three. Each base classifier is an averaged Bayesian network as described in Section 6.1, and depending on the sign of the interaction information measured between the feature subset, it can either be an averaged forkANB or an averaged colliderANB. Algorithm 1 describes the pseudo code for irsADE.

To quantify the effect of using interaction information on the ensemble performances, we compare irsADE with an accuracy based ensemble approach, where

Algorithm 1 Ensemble Method **irsADE**

Split dataset $\mathcal{D}$ into TR (train), TS (test)
for $i = 1 : \mathrm{T}$ **do**
  Randomly pick 3 features $S_i = \{X_1, X_2, X_3\}$
  if $\mathrm{I}(S_i) > 0$ **then**
    Build all possible colliderADE models from $\mathrm{TR}(S_i)$
  else
    Build all possible forkADE models from $\mathrm{TR}(S_i)$
  end if
  Build averaged ADE model f^i from selected models
end for
Combine averaged models $f^{\mathrm{ens}} = \mathcal{F}(f^i, \ldots, f^{\mathrm{T}})$
Test the ensemble f^{ens} on TS

the model class for a specific random subset is selected by training both averaged Bayesian network models and choosing the most accurate one on the training set, rather than using the sign of interaction information to decide which model to train. Similarly to irsADE, we name this approach **rsADE** , as for **r**andom **s**ubspace **A**veraged **D**ependency **E**stimators. Algorithm 2 illustrates how base classifiers are trained accordingly to rsADE. Ensemble methods irsADE and

Algorithm 2 Ensemble Method **rsADE**

Split dataset $\mathcal{D}$ into TR (train), TS (test)
for $i = 1 : \mathrm{T}$ **do**
  Randomly pick 3 features $S_i = \{X_1, X_2, X_3\}$
  Build all possible colliderANB models from $\mathrm{TR}(S_i)$
  Build all possible forkANB models from $\mathrm{TR}(S_i)$
  Choose the most accurate averaged model f^i on $\mathrm{TR}(S_i)$
end for
Combine averaged models $f^{\mathrm{ens}} = \mathcal{F}(f^i, \ldots, f^{\mathrm{T}})$
Test the ensemble f^{ens} on TS

rsADE are aimed at generating structurally diverse base classifiers. The main difference between these two methods is that the first one uses the sign of interaction information to exploit some prior knowledge about the data, whereas the second one is a decision directed method that relies only on the training accuracy of base classifiers.

6.2.1 Experimental Settings

We combine base classifiers generated via irsADE and rsADE according to simple mean rule, although base classifiers could be combined according to any combination rule. The size of the ensemble varies with the size of the training set, as we decide to train a number of base classifiers equal to the number of features. Appendix B shows the ensemble performance of irsADE and rsADE for ensembles of 50 base classifiers.

Methods irsADE and rsADE are tested according a 5×2-fold cross-validation on 7 different datasets which have already been described in Table 6.1. Table 6.7 illustrates the number of possible subsets of size 3 that can be generated for each dataset.

Table 6.7: Number of distinct feature subsets of size 3.

Dataset	Feature Subsets
glass	84
magic4	120
congress	560
segment	969
mushroom	1330
hypothyroid	3654
sickeuthyroid	3654

For each dataset but Glass and Magic4, each base model is trained on a different random subset of the feature space for every repetition of the experiments. Since Glass and Magic4 have a small number of features, and hence a small number of possible distinct subsets that can be generated, the base classifiers are trained on different random subsets of the feature space, but these subsets are identically repeated for each run of the 5×2-fold cross-validation. The reason we do not choose to apply this variant to the remaining datasets is due to the fact that Bayesian networks are low-variance classifiers, and by always using the same random subsets on each different run of the 5×2-fold cross-validation, we would train a base model on the same feature subspace of slightly different replicas of the training data. Overall models will be very similar on different runs and therefore this procedure might lead to too optimistic confidence intervals.

We now show the experimental results obtained by comparing irsADE with rsADE in terms of *generalisation error* and *training time*.

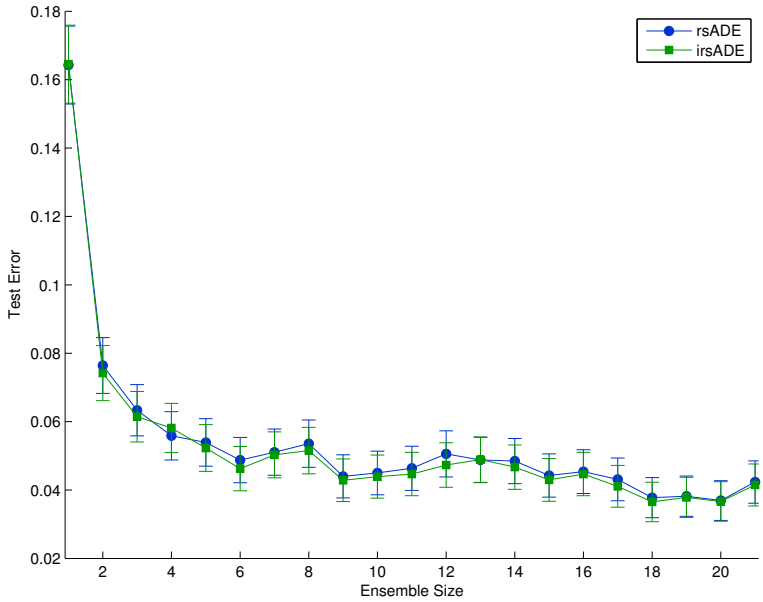


Figure 6.2: Mushroom dataset: irsADE vs rsADE – Test Error (mean and 95% confidence interval)

6.2.2 Generalisation Error

We now compare the generalisation ability of irsADE and rsADE. As expected, the ensemble techniques have a positive effect on the generalisation ability of some datasets, showing that there is no universal learning technique which is able to discriminate any classification problem.

Figures 6.2 and 6.3 respectively compare the generalisation error of irsADE and rsADE on Mushroom and Segment datasets as we increase the number of base classifiers. Figure 6.2 shows that the ensemble generalisation error decreases from 16% to 4% as we increase the number of classifiers in the ensemble. The ensemble methods irsADE and rsADE show similar classification performance for any size of the ensemble. Figure 6.3 shows a similar trend to Figure 6.2 for the Segment dataset. Overall the generalisation error decreases by more than 15% and both ensemble techniques show identical performance. Figures B.5 and B.4 in Appendix B illustrate how increasing the number of base classifiers up to 50% does not produce any change in the two ensemble method mutual behavior, as in both figures the classification error reaches a plateau from ensembles of 20 onwards although the classification mean error seems to be slightly improved. It is interesting to compare the base classifier generalisation ability with the ensemble generalisation ability as we increase the number of classifiers. Figures

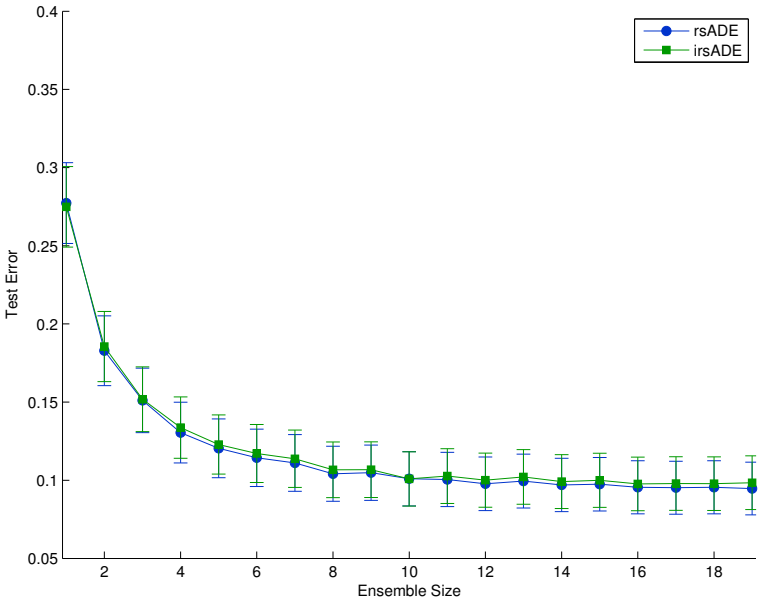


Figure 6.3: Segment dataset: irsADE vs rsADE – Test Error (mean and 95% confidence interval)

C.4 and C.5 in Appendix C show on the same graph the ensemble error for increasing numbers of base classifiers and each single base classifier error before being combined in the irsADE ensemble. For Mushroom and Segment datasets the ensembles succeed in reducing the generalisation error because they combine diverse base classifiers, since base classifiers show different levels of accuracy and must therefore be diverse. We conclude that when base classifiers are diverse, not only do both irsADE and rsADE ensemble techniques succeed in reducing the overall classification accuracy, but also that the sign of interaction information can be used as a base classifier model selection criterion, as it does not negatively affect the ensemble performance.

Figures 6.4, 6.5 and 6.6 are three cases where irsADE and rsADE do not succeed in reducing the generalisation error. Increasing the number of classifiers in the ensemble does not alter the ensemble performance, as shown in Figures B.2, B.3 and B.1. It is interesting to note that this occurs for Glass and Magic4, which are very different in sample size (the first one has only 107 training patterns, whereas the second one has 9509 training patterns) but both have a small of number of features (respectively 9 and 10). On the other hand Congress has a larger number of features (16) and a small number of training patterns (217). The reason why these ensemble methods do not succeed might be because the base

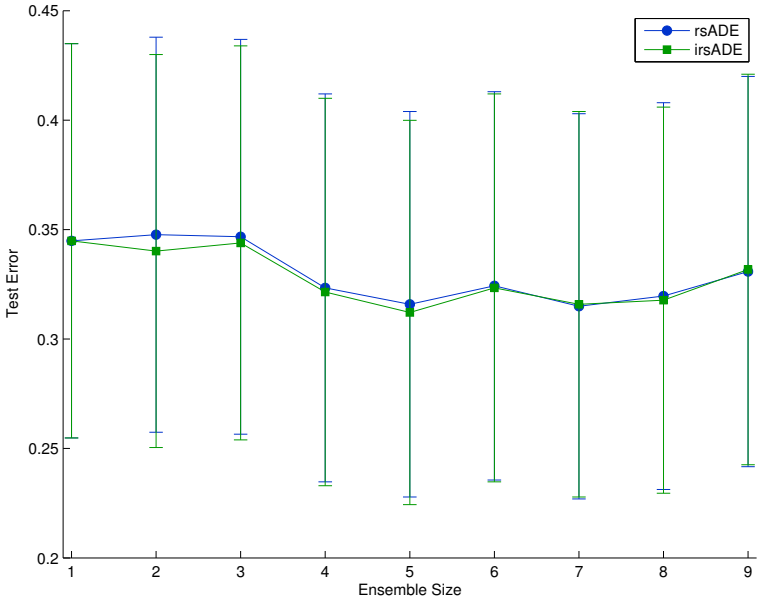


Figure 6.4: Glass dataset: irsADE vs rsADE – Test Error (mean and 95% confidence interval)

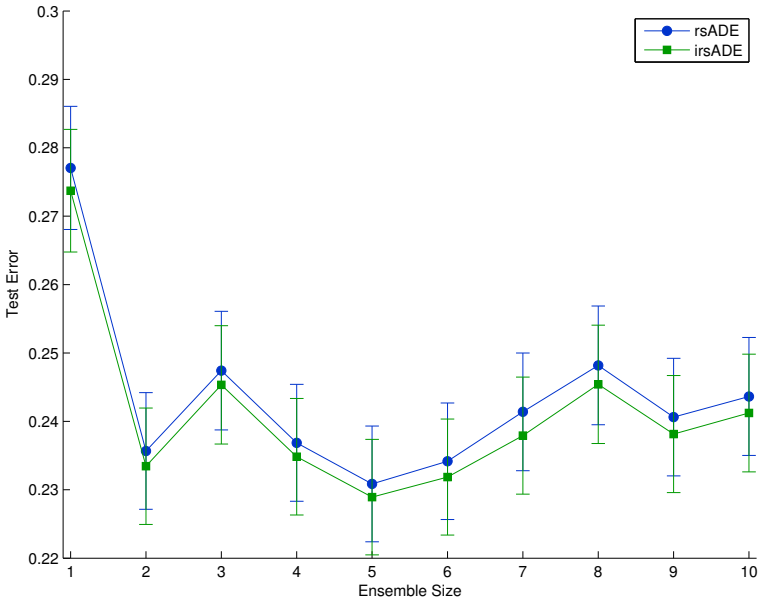


Figure 6.5: Magic4 dataset: irsADE vs rsADE – Test Error (mean and 95% confidence interval)

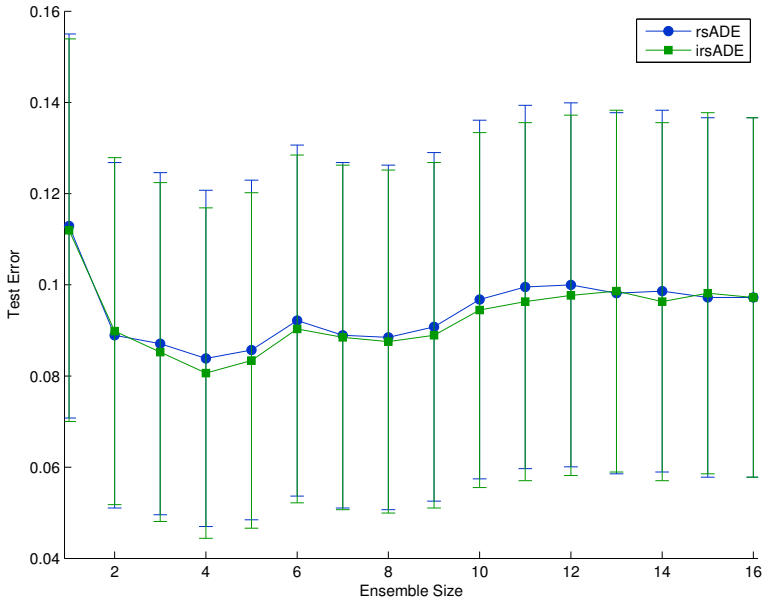


Figure 6.6: Congress dataset: irsADE vs rsADE – Test Error (mean and 95% confidence interval)

classifiers are not diverse from each other, as shown in Figures C.2 and C.3 and C.1: from these graphs it is easy to observe that the base classifiers show similar classification errors which are comparable to the irsADE ensemble error as we increase the number of classifiers in the ensemble. This behavior (which is less pronounced for Magic4) indicates that the base classifiers are not diverse from each other. However, both irsADE and rsADE show comparable generalisation error, which confirms how the sign of interaction information can be used as a proxy to classification accuracy.

Figures 6.7 and 6.8 compare the generalisation ability of irsADE and rsADE on Sickeuthyroid and Hypothyroid as we increase the number of base classifiers. For both datasets these ensemble approaches negatively affect the classification performance, as the generalisation error increases as we increase the number of classifiers.

Sickeuthyroid and Hypothyroid, are respectively the 2 class and 4 class version of the same classification problem. These datasets are particularly unbalanced. In Sickeuthyroid 93.9% of the data is of class 1 and only 6.1% is of class 2. Similarly, in Hypothyroid the data belongs to one out of 4 classes according to these percentages: 5.1%, 92.3%, 2.5%, and 0.1%. Moreover, only 40% of the 3772 patterns are distinguishable from each other. It is worth observing that the ensemble

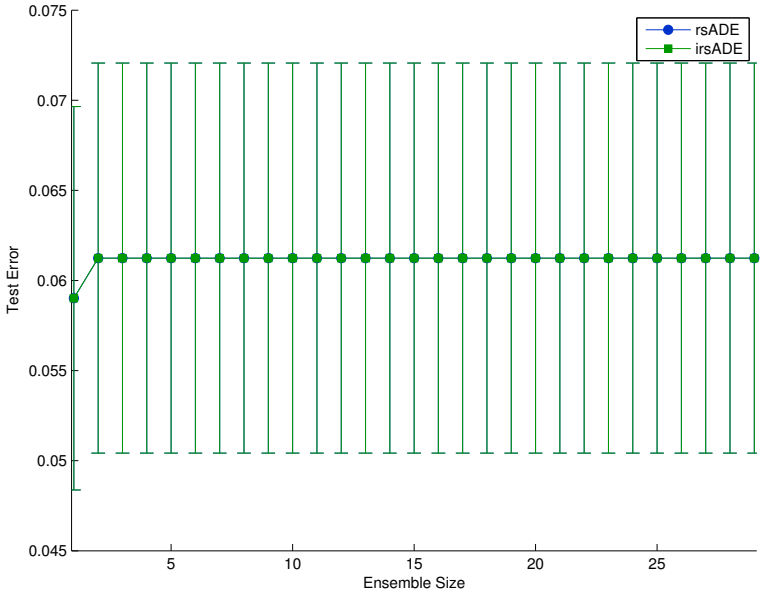


Figure 6.7: Sickeuthyroid dataset: irsADE vs rsADE – Test Error (mean and 95% confidence interval)

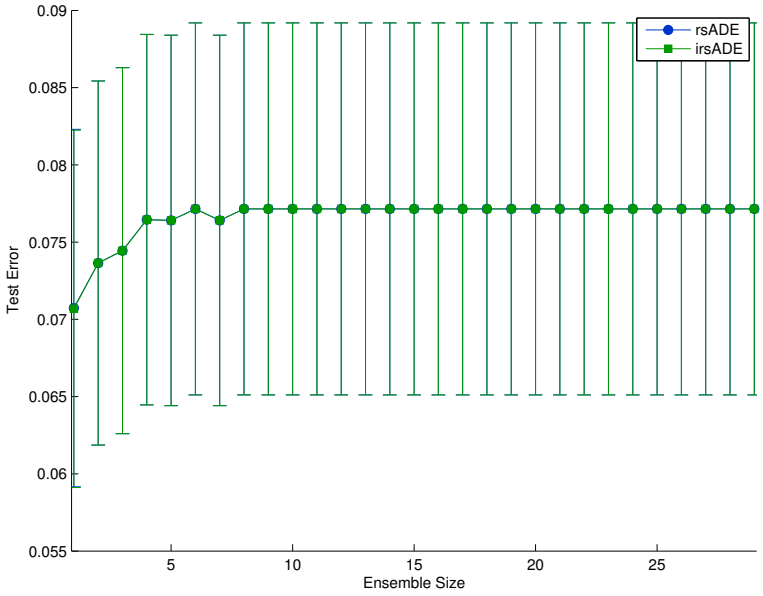


Figure 6.8: Hypothyroid dataset: irsADE vs rsADE – Test Error (mean and 95% confidence interval)

does not to improve over the single classifier as base classifiers make errors which are not statistically different from each other and the ensemble combination does not improve over the single accuracies, as shown in Appendix C in Figures C.6 and C.7. Moreover, it is interesting to point out that on average about 92% of test patterns are classified as class 1, which is in line with the data prior distribution. By analyzing the sensitivity and the specificity of base classifiers and the sensitivity and the specificity of the ensemble as we increase the number of classifiers we find out that the averaged sensitivity of base classifiers is about 97% and their average specificity is 9%. More accurate classifiers show higher levels of specificity as well as high levels of sensitivity. Regarding the irsADE ensemble behavior, the first base classifier shows similar values of sensitivity and specificity, but the simple mean rule improves the sensitivity of the ensemble up to 100%, and reduces the ensemble specificity to 0%.

6.2.3 Training Time

We now compare the ensemble approaches irsADE and rsADE in terms of training time. As we have already pointed out, irsADE make use of interaction information to select the model class for each base classifier. On the other hand, the accuracy based method rsADE builds an averaged model for each model class, and then picks the most accurate one on the training set. It is easy to see that the first one has to build only one averaged model, whereas the second one has to build two different models and then test them on the training set to decide which one is the most accurate.

Tables 6.8 shows the average training speed and standard deviation of each method over the 5×2-fold cross-validation experiment. The last column shows the ratio between the averaged rsADE training speed and the averaged irsADE training speed, that is, the speed improvement of irsADE with respect to rsADE. To compare our results on different datasets, we fixed the ensemble size to 50 classifiers for each dataset. These results clearly show that an interaction information ensemble based approach can noticeably improve the ensemble speed performance. As shown in Table 6.1 and in Table 6.8, different datasets are characterised by different speed improvements.

The main reason why the computational cost varies so greatly among different datasets is due to the choice of using probabilistic models as our base models. Training a probabilistic model consists of learning the form of its probability

Table 6.8: Average training seconds (mean and standard deviation) required to train an ensemble of 50 classifiers. The last column is the ratio between rsADE and irsADE training times.

Dataset	rsADE	irsADE	Improvement
congress	17.3 [0.0]	1.0 [0.0]	17.0
glass	10.3 [0.0]	1.3 [0.0]	7.7
magic4	790.3 [0.5]	40.3 [0.9]	19.6
mushroom	309.7 [3.6]	10.0 [1.6]	30.9
idaimage	89.6 [0.0]	4.4 [0.3]	20.5
segment	201.2 [13.7]	40.8 [8.2]	4.9
sickeuthyroid	134.7 [0.1]	1.4 [0.1]	98.0
hypothyroid	138.1 [0.4]	2.7 [0.3]	51.0

distribution. In our case all the random variables are discrete, and this implies that we have to learn the way a probability distribution factorises for any possible value of each random variable in the model. The learning process is strongly affected by number of training patterns and the alphabet of the training set, that is the number of classes and feature attribute values. As an example, if we want to learn a conditional probability distribution $\mathrm{p}(X|Y)$, where Y can take two different values and X can take up to 15 values, this will require to learn $2 \times 15 = 30$ estimates, one for each class conditional distribution. If Y can take two different values and X can take up to 3 values, this will require to learn $2 \times 3 = 6$ estimates, one for each class conditional distribution.

In addition to the dataset statistics, the structure of colliderANBs and forkANBs have a strong effect on the computational time. These two models learn different forms of the probability distributions. The main difference between the two is that whereas in forkANB models each random variable depends on at most one other random variable, in the collider ANB one random variable depends on all the remaining random variables. Whereas in the former case, we only calculate the probability of features conditioned on at most one random variable, in the latter we have to calculate probabilities of features being conditioned on all the remaining random variables, for any possible value in the alphabet of every random variable. To reuse the previous example, if X_1 can take 15 values, X_2 can take 3 values and X_3 can take 4 values, in a fork structure such that X_3 depends on X_2 and Y as in $\mathrm{p}(X_3|X_2, Y)$ we will have to calculate $4 \times 3 \times 2 = 24$ estimates, one for each class conditional distribution. On the other hand, in a collider structure such that X_3 depends on X_2, X_1 and Y as in $\mathrm{p}(X_3|X_2, X_1, Y)$ we will

have to calculate $4 \times 3 \times 15 \times 2 = 360$ estimates, one for each class conditional distribution.

Finally, whereas irsADE chooses only one of the two averaged models, rsADE estimates both averaged models. To summarise, irsADE does not simply reduce the computational time by a half with respect to the rsADE, as the computational time will be affected by the sign of the interaction information as well as the size of the alphabet of the features.

The dataset showing the least improvement from irsADE is Segment, which is a 7 class problem whose feature alphabet size can be up to 15. We tested irsADE and rsADE on Idaimage, which is another representation with 2 classes and maximum alphabet of 5, irsADE is 5 times faster than in segment. This can be explained by pointing out how the number of classes has more of an effect than the number of features, as the feature size varies between the features (and the subspaces), whereas the number of classes remains unchanged for all the base classifiers. This shows that changing the alphabet of the random variables, combined with the sign of interaction information, can have a huge effect on the computational improvement of irsADE over rsADE.

An important case is that one of Sickeuthyroid and Hypothyroid, that are the two and four class version of the same problem. Interestingly the range of the attribute size does not varies much between the two, as the mean attribute value for the two is respectively 2.3 and 2.6. In this case the irsADE improvement in Sickeuthyroid is nearly twice as the irsADE improvement in Hypothyroid, showing that the actual size of the alphabet has an important effect on the improvement of irsADE over rsADE.

We conclude that for averaged augmented Naïve Bayes classifiers such as averaged colliderANB and averaged forkANB, interaction information based ensembles are much more computationally efficient than accuracy based methods.

6.3 Chapter Summary

This chapter tried to address two main research questions, that is, *Can we use interaction information properties to build base classifiers?* and if so, *is there a way to make use of these findings in ensemble learning?* In Section 6.1 we empirically investigated the first question by using the sign of interaction information to predict the class with the most accurate classifier from two different classes

of Bayesian networks, that is colliderANB and forkANB. We then investigated whether this prediction was robust to averaging over all models belonging to these classes. Despite the many assumptions and the limitations due to the data availability, we showed that the sign of interaction information could be used to predict the most accurate averaged model from two different model classes.

We then proposed in Section 6.2 a practical application of these properties to random subspace ensemble learning techniques. We develop irsmADE, an ensemble technique which combines hybrid averaged Bayesian networks by looking at the sign of interaction information measures on the base classifier feature subset. Our results show that interaction information provide us with some a priori knowledge which can be used to choose hybrid base models in a more efficient way than an accuracy based approach. In particular we show that not only does the information theoretic approach irsADE not negatively affect the generalisation ability of the ensemble, but it is also computationally more efficient than the accuracy based counterpart ensemble method. Moreover our results contribute towards answering the question of *how we can generate structurally diverse base classifiers in a sensible way*, as we have developed a methodology that can create structurally diverse base classifiers which can improve the overall ensemble accuracy.

Chapter 7

Conclusions and Future Work

The primary research question which this thesis has answered is *"Can we use a loss function other than the* 0/1 *loss to understand and manage diversity in classifier ensembles?"* This question is of primary importance, as the link between ensemble accuracy and diversity is still unclear, the main reason being that there is no unique way to decompose the 0/1 loss in terms of the bias and variance of a classifier, as there is for the mean squared error [48].

7.1 Research Contributions

In order to answer our main research question, we focused on understanding classification diversity through two different loss functions, that is, the mean squared error and the negative log-likelihood. In Subsection 7.1.1 we summarise our findings about whether it is possible to manage classifier diversity through the mean squared error loss function. From Subsection 7.1.2 to Subsection 7.1.4 we summarise our findings about what we have understood about classifier diversity in this thesis with a special emphasis on model diversity and information theory.

7.1.1 Towards Managing Diversity in Classifier Ensembles

One of the main objectives of studying classifier diversity is to develop ensemble techniques which can *directly manage* the trade-off between accuracy and diversity. In Chapter 2 we linked the Tumer & Ghosh model for the classification error with the NC learning framework for regression problems, and we answered the question *"Can we deploy NC learning in the context of the Tumer & Ghosh*

framework so that we can manage diversity in classification problems?" To answer this question, we observed that unlike othe bias-variance analysis of the 0/1 loss, the Tumer & Ghosh model treats the classification error as a regression random variable. This observation enabled us to reformulate the Tumer & Ghosh model into a regression model. As a result, we developed an algorithm which can effectively manage diversity for classifier ensembles [101]. This algorithm trains the ensemble base classifiers simultaneously while trying to minimise the overall ensemble error.

7.1.2 Towards Understanding Classifier Model Diversity

In an effort to understand diversity, in Chapter 4 we embarked on an empirical investigation of *model diversity* as opposed to *error diversity*. Since error diversity is an immediate consequence of model diversity, our research question here was: *"is it possible to generate diverse classifiers by looking at model diversity rather than error diversity?"* We identified parametric probabilistic models as our object of investigation, as they can *explicitly select the model bias*. In these models, diversity can occur at two different levels: it can be parametric or it can also be structural. Therefore, the specific question we tried to address in this chapter was: *"is parametric diversity sufficient to build accurate and diverse ensembles of Naïve Bayes classifiers?"* Our experimental results showed that Bagging generates parametric diversity between Naïve Bayes models, whereas Random Subspaces introduced a certain level of structural diversity as base classifiers have the same model dependencies but are trained on different features. We found that parametric diversity is not sufficient when combining stable classifiers such as Naïve Bayes models. In fact, bagging base classifiers are accurate but not diverse. Conversely feature structure diversity introduced by random subspaces generates base classifiers which are diverse but not accurate enough to make the ensemble outperform the single classifier approach. Our results seemed to point towards the idea that *diversity in stable classifiers has to be structurally inferred* [100].

As a secondary result, we also found that the success of Bagging with stable classifiers such as Naïve Bayes classifiers depends on the training size and on the model specifications (Chapter 4). This is in line with results found for other stable classifiers [78, 1].

7.1.3 Towards Monitoring Diversity

As we discussed in Subsection 3.4.1, the expected value of the negative log-likelihood of the class posterior probability with respect to all the random variables corresponds to the conditional entropy for the same probability distribution. This link might indicate that interaction information, which is a multivariate extension of mutual information, could be used to understand classifier diversity. In Chapter 5 we addressed the question: *"Can we use interaction information to understand diversity between base classifiers?"* Our empirical investigation showed that interaction information is able to *capture the trade off between ensemble accuracy and diversity*. We presented empirical evidence that *diversity occurs at different levels of interactions between base classifiers*, and therefore higher order interactions between base classifiers cannot be discarded. As a consequence, Bayesian Networks can only *approximately* monitor interactions between base classifiers.

7.1.4 Towards Building Structurally Diverse Ensembles

In Chapter 6 we attempted to answer the question "*Can we use interaction information to generate more efficient ensembles?*" We first presented empirical evidence that the sign of interaction information measured between features is a good proxy for classification accuracy of augmented Naïve Bayes classifiers and averaged augmented Naïve Bayes classifiers, and that therefore it can be used to choose the structure of an augmented Bayesian network. We then proposed a *novel* interaction information based ensemble technique, *irsADE*, which exploits interaction information properties to generate accurate and structurally diverse averaged augmented Naïve Bayes classifiers. We presented an empirical comparison of irsADE with another ensemble method which measures accuracy rather than interaction information. We show that irsADE *does not negatively affect the ensemble accuracy but on the contrary is at least an order of magnitude faster than the accuracy based ensemble method.*

7.2 Future Work

7.2.1 Extension of Interaction Information Properties

In Chapter 6 we empirically investigated how interaction information properties which hold for three random variables, can be used to predict the accuracy of base classifiers, and that these properties can be exploited to generate more efficient ensembles. A very interesting research direction in this sense, would be to investigate whether these properties can be generalised to classifiers of more than three features. There are two reasons for addressing this limitation of interaction information properties to three random variables. The first one is that by increasing the number of features, we would increase the number of possible models that could be generated. The second one is that the accuracy of Bayesian networks improves with the number of features in the model.

One way to investigate this, would be to provide some heuristic procedure to increase the size of the feature subsets. For instance, we could build classifiers with more than three features, but only allow statistical dependencies between three of these features. Alternatively, we could use mutual information to rank the features and allow inter-dependencies between only the three most informative ones. *Another way* would be to investigate whether similar properties can be inferred for more than three random variables from the decomposition of the mutual information of the joint set of random variables in all possible interaction information terms, as in Equation (3.30), or at least in the special case that only three random variables in the feature set interact with each other.

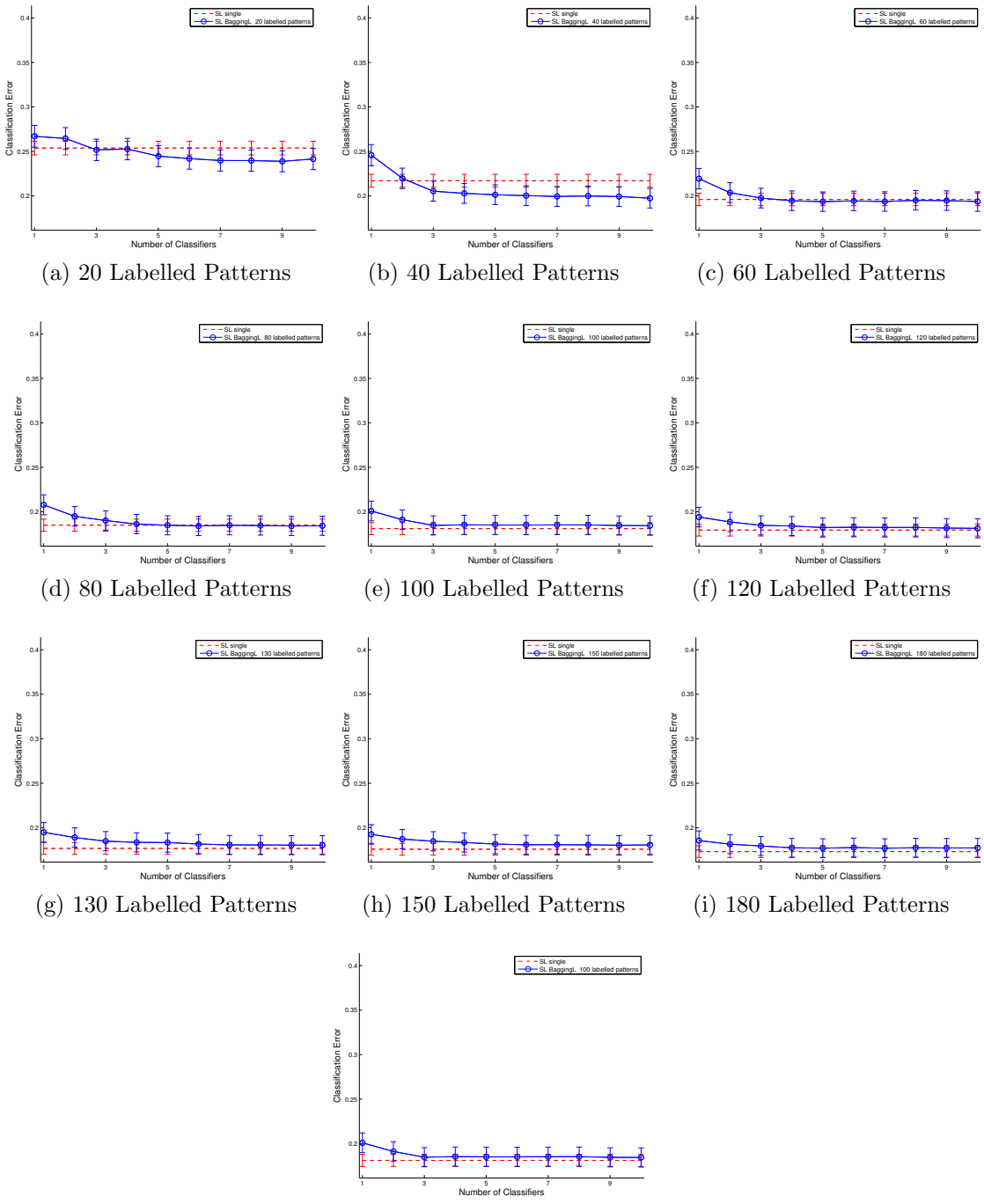
7.2.2 Using Interaction Information to Prune Ensembles

Whereas in Chapter 5 we discussed how interaction information can be used to monitor ensemble diversity between classifier outputs, in Chapter 6 we applied interaction information to generate accurate and structurally diverse base classifiers. Another attractive research direction that could be addressed via interaction information, would be to learn more complex Bayesian networks (for instance via hill climbing search, or Minimum Description Length) and investigate whether interaction information can be used to select accurate and diverse base classifiers, that is, classifiers showing lower bias than augmented Naïve Bayes classifiers. In other words, it would be interesting to study whether interaction information can

be used to prune the ensemble. An interesting result of feature selection that could be applied here is that most of the heuristic methods for feature ranking can be derived from the decomposition of the interaction information if this is truncated to the second order interactions [13]. In a similar way, we could for instance apply interaction information to rank ensemble members according to different levels of interactions between classifiers.

Appendix A

Naïve Bayes Ensembles



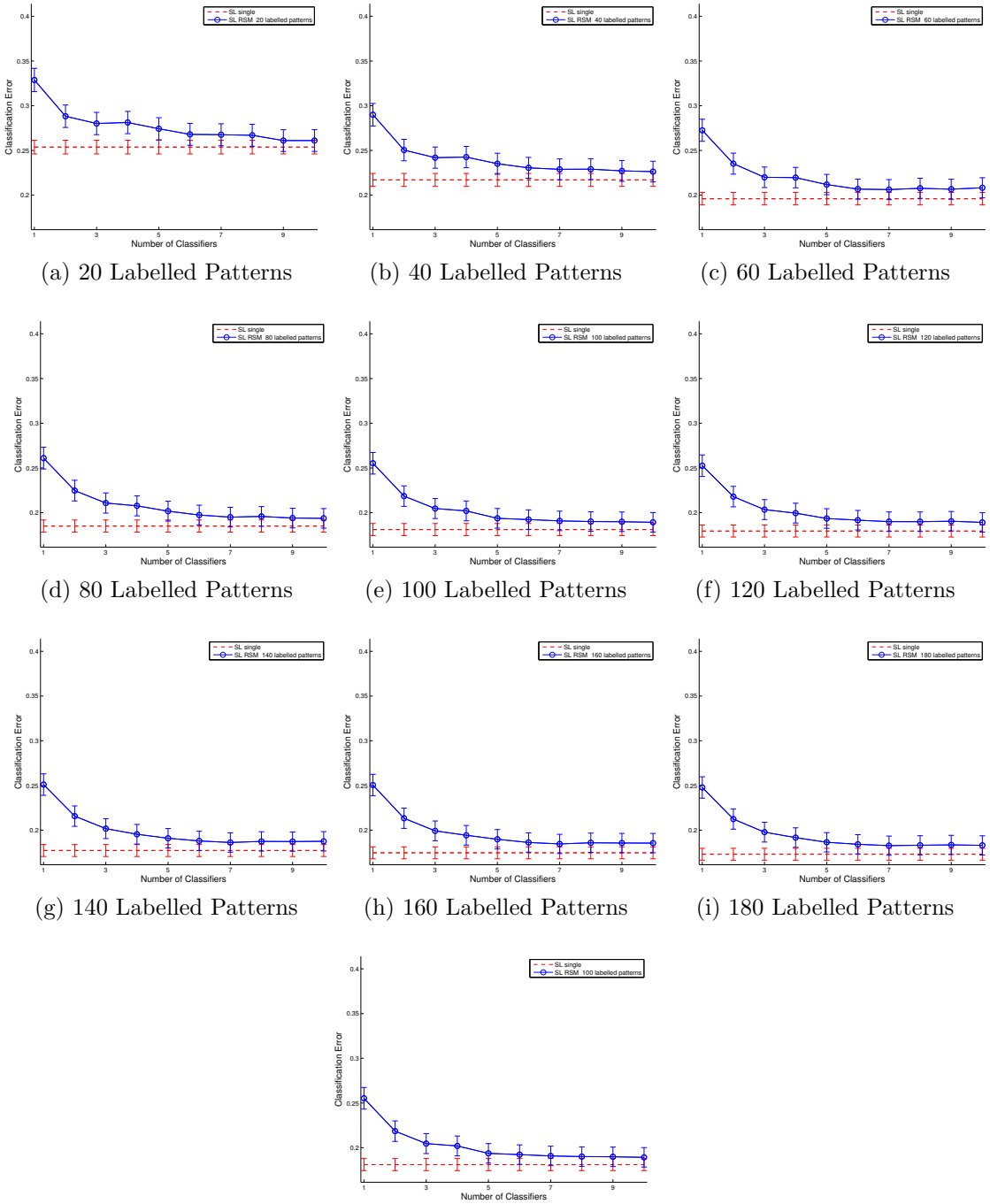
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 130 Labelled Patterns (h) 150 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.1: Uniringnorm dataset (model match problem), supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



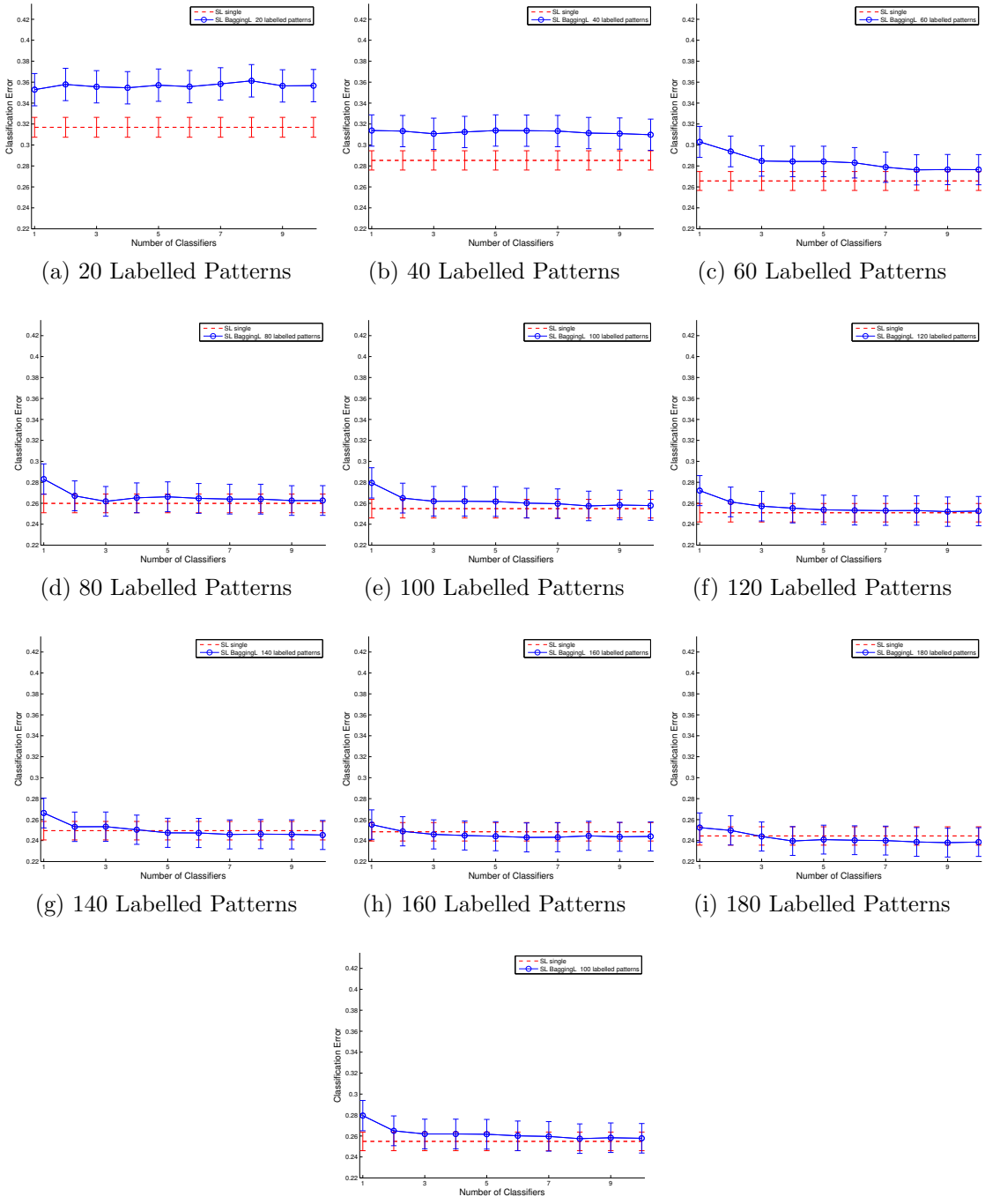
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.2: Uniringnorm dataset (model match problem), supervised learning – Test error (mean and 95% confidence interval) of RSM (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



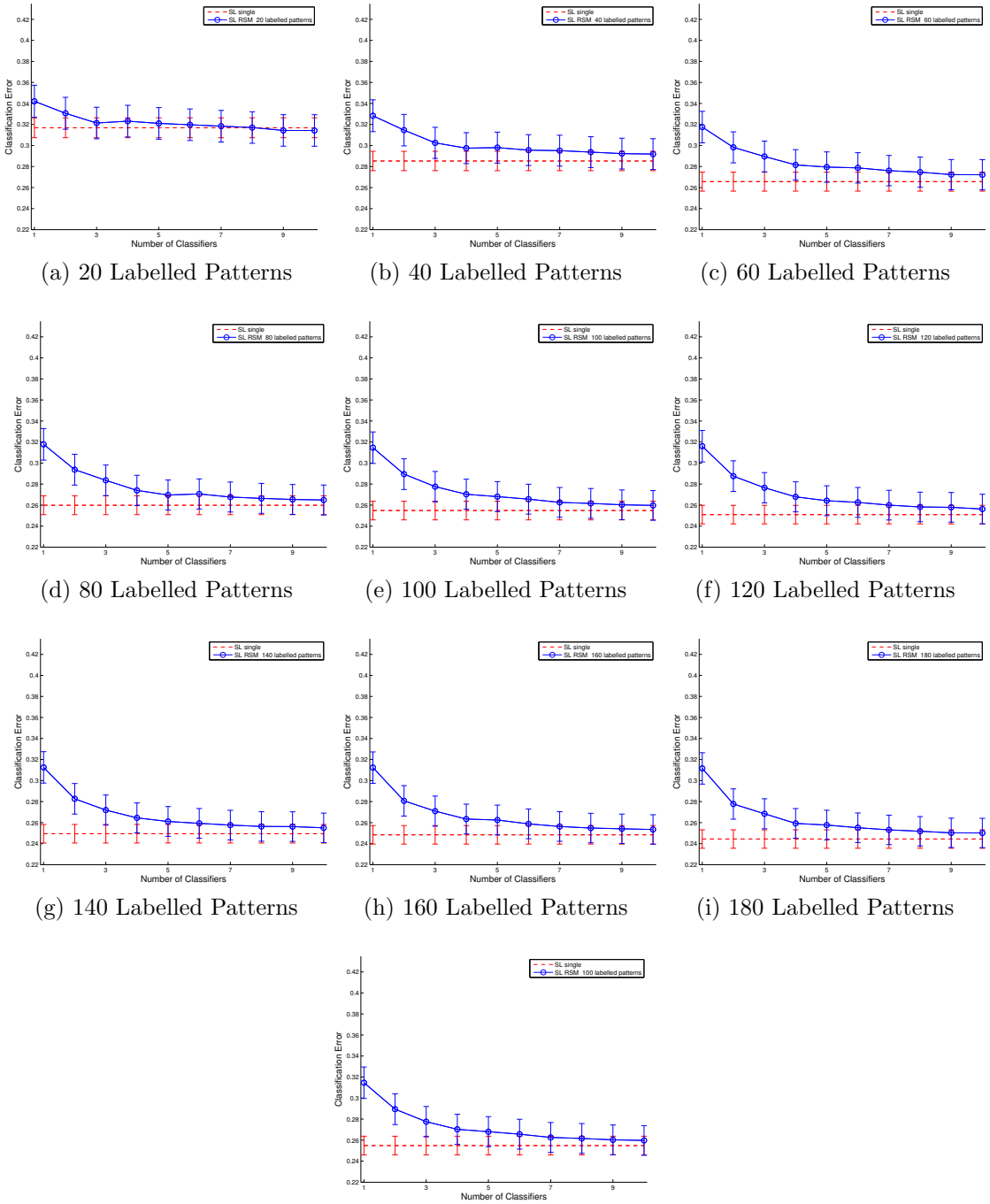
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.3: Ringnorm dataset (model mismatch problem), supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



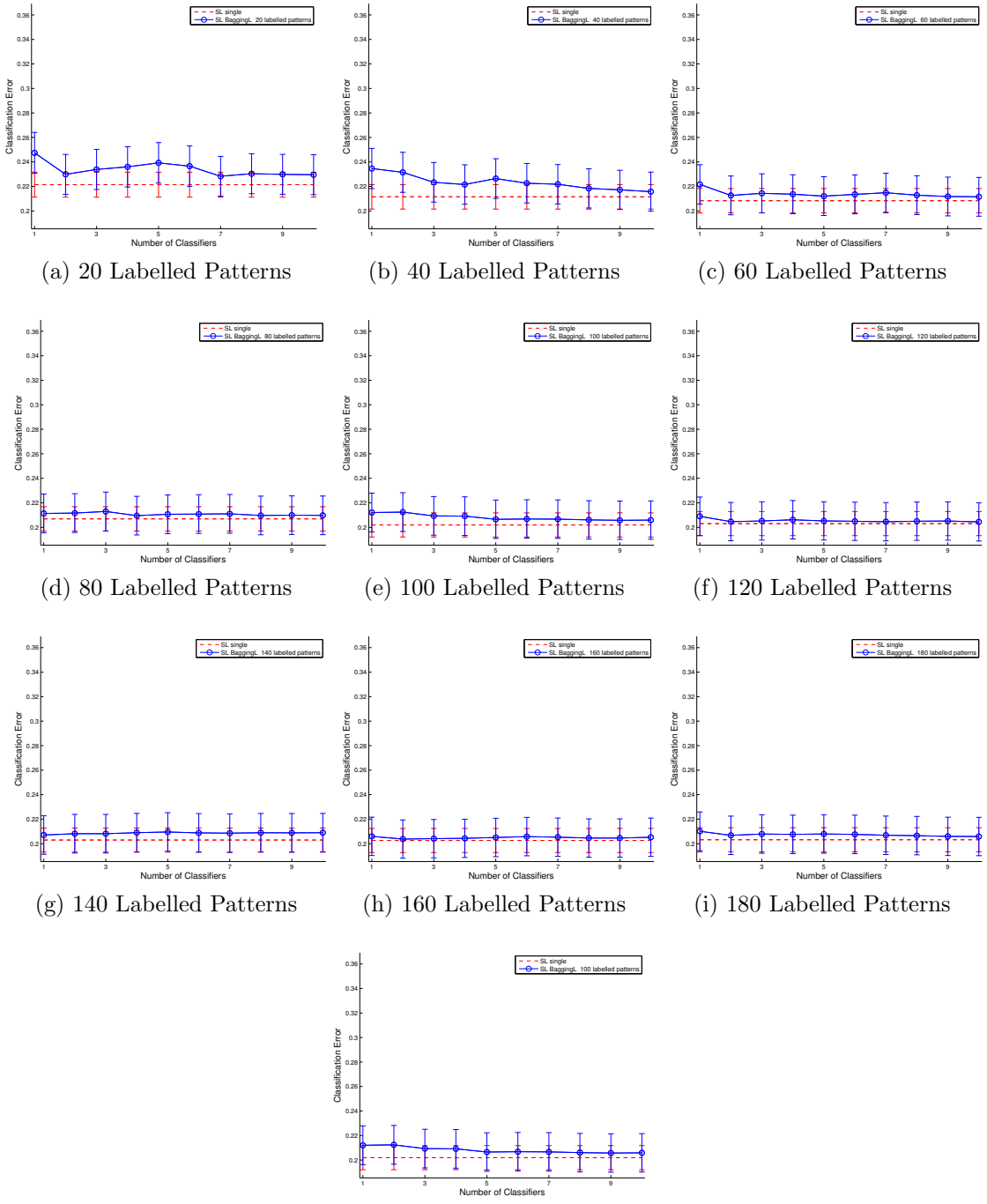
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.4: Ringnorm dataset (model mismatch problem), supervised learning – Test error (mean and 95% confidence interval) of RSM (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



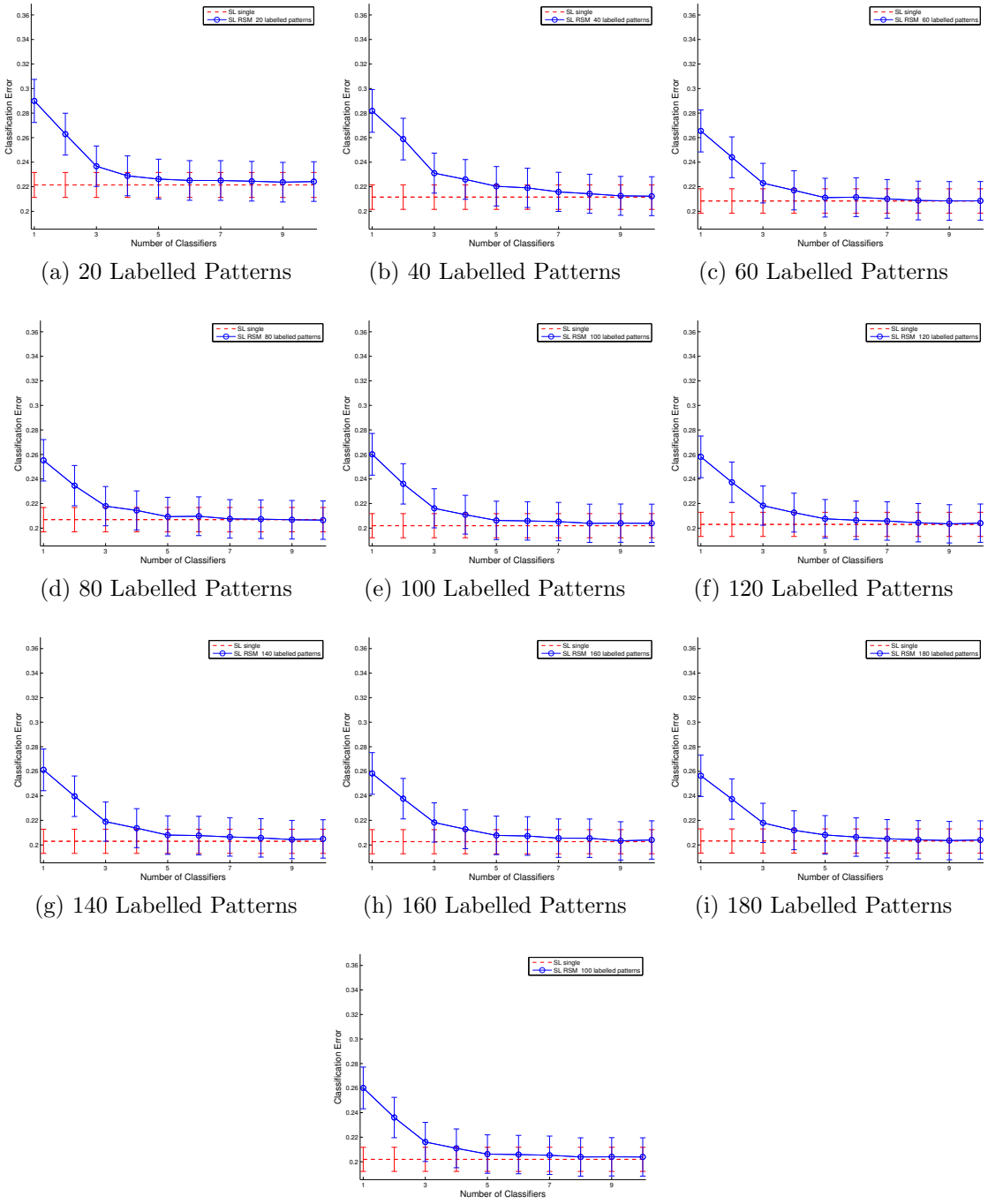
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.5: Feltwell dataset, supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



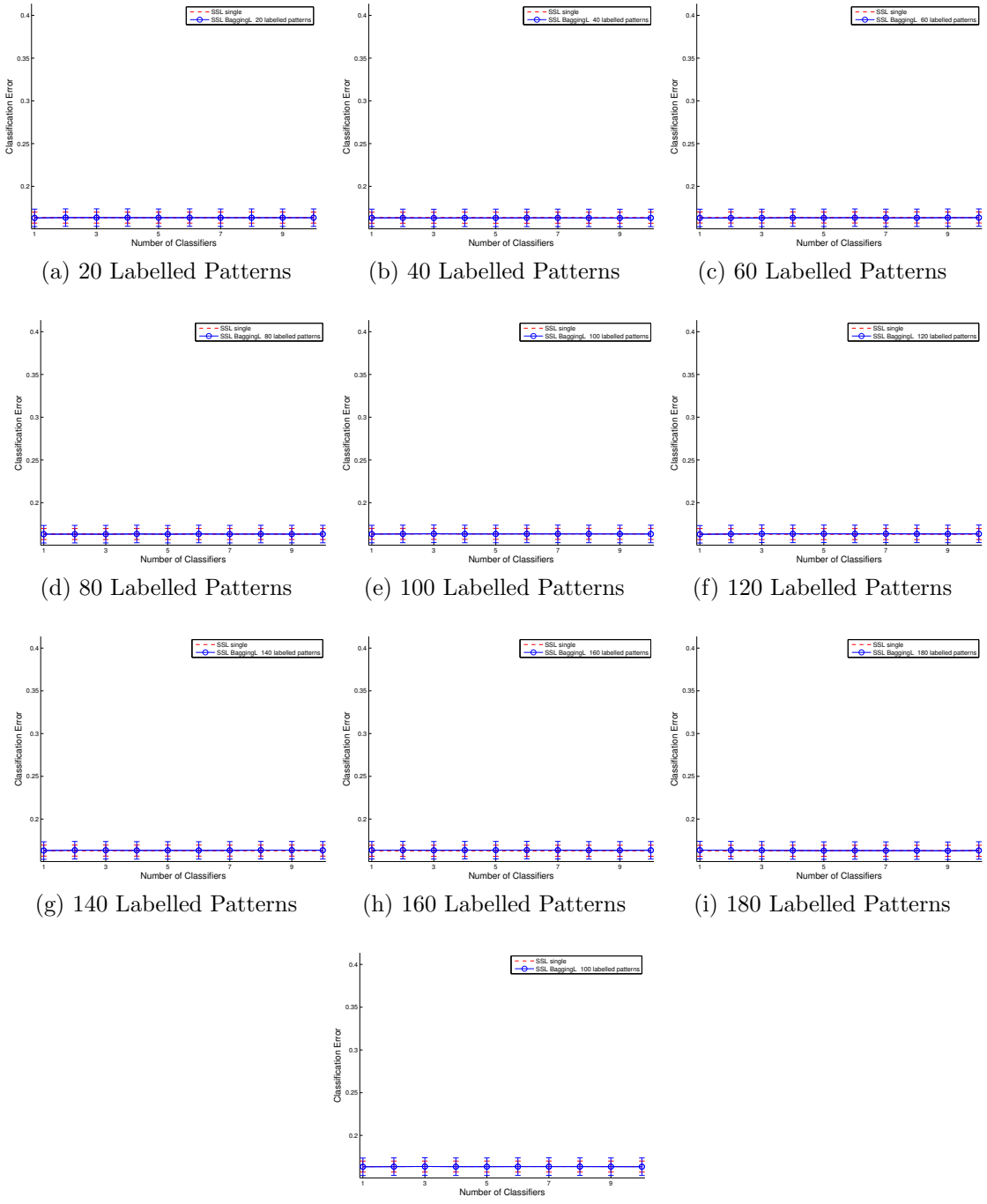
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.6: Feltwell dataset, supervised learning – Test error (mean and 95% confidence interval) of RSM (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



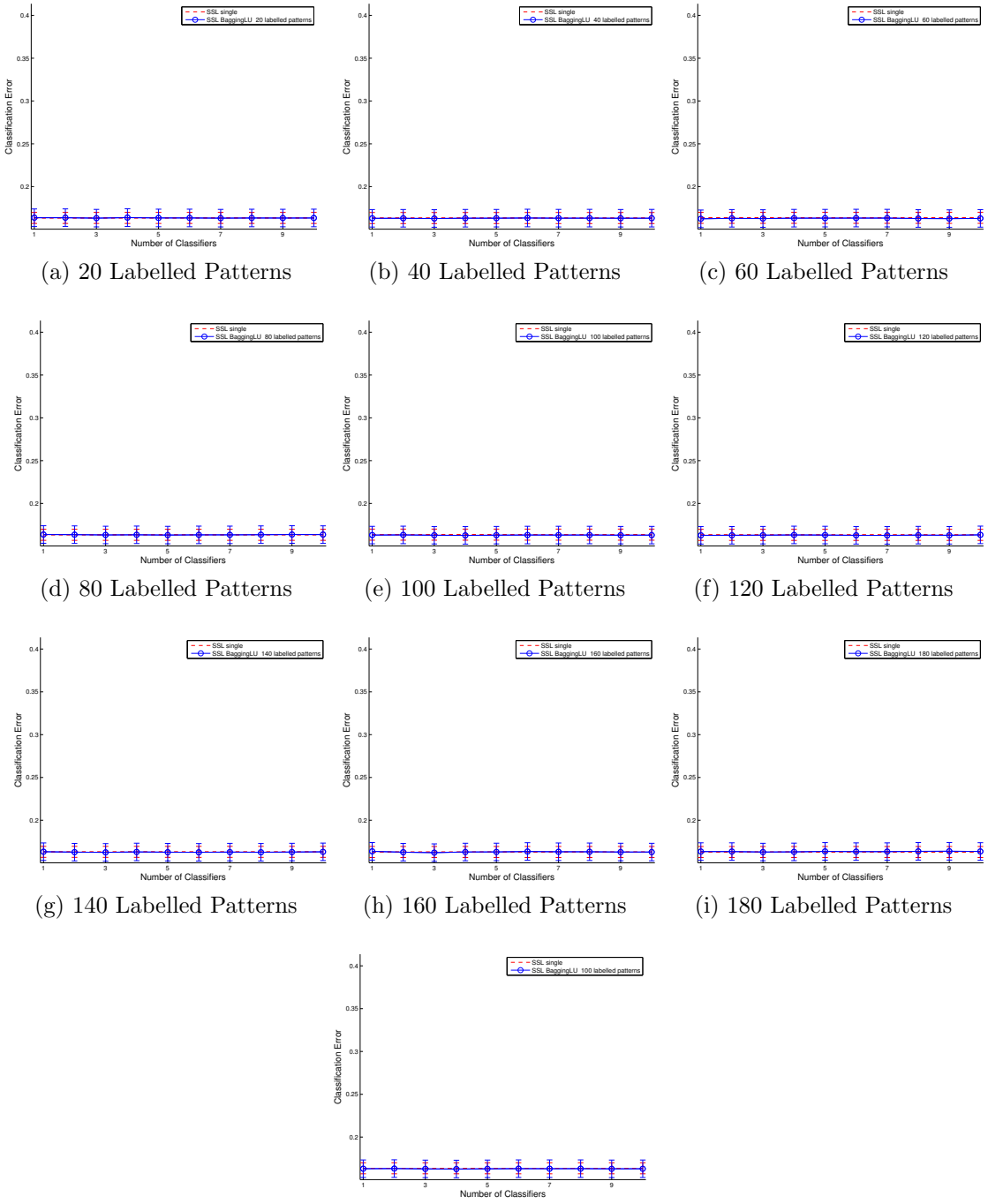
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.7: Uniringnorm dataset (model match problem), semi supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



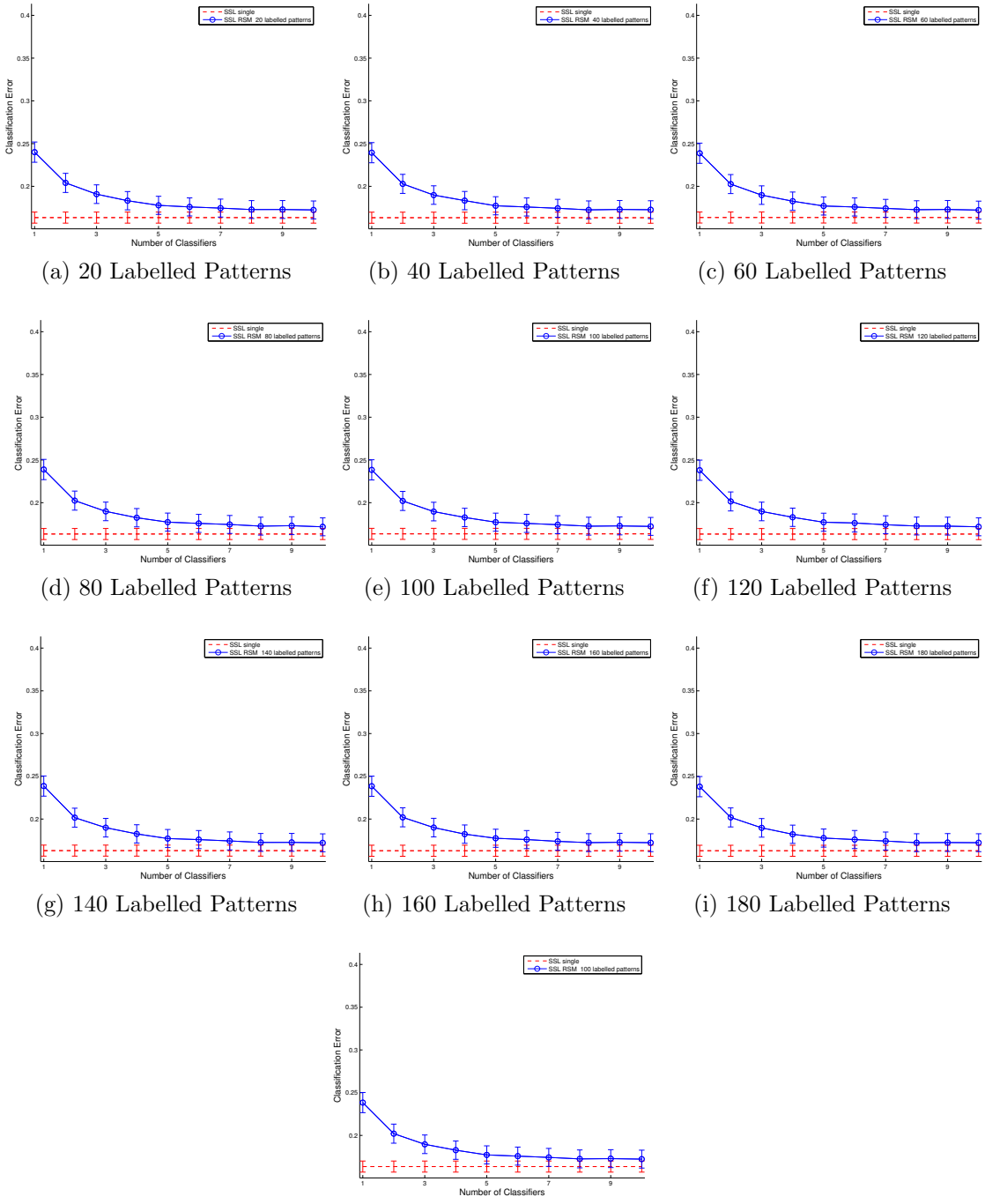
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.8: Uniringnorm dataset (model match problem), semi supervised learning – Test error (mean and 95% confidence interval) of BaggingLU (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



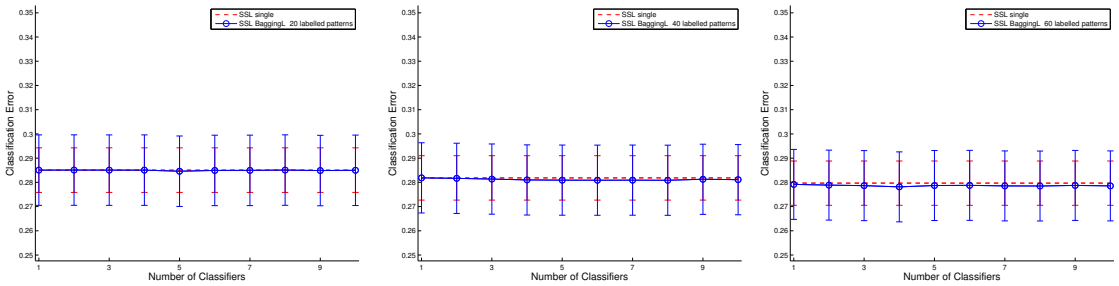
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

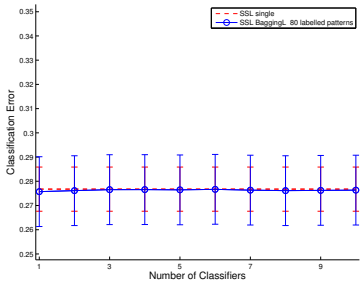
Figure A.9: Uniringnorm dataset (model match problem), semi supervised learning – Test error (mean and 95% confidence interval) of RSM (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



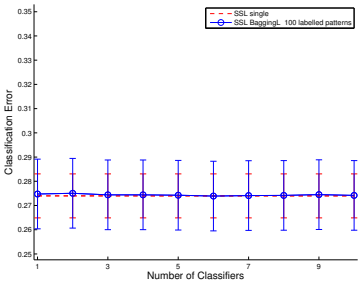
(a) 20 Labelled Patterns

(b) 40 Labelled Patterns

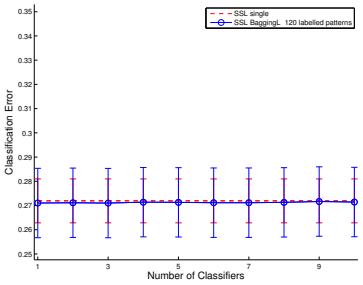
(c) 60 Labelled Patterns



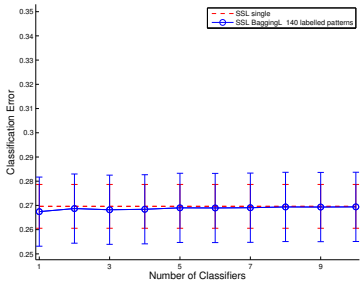
(d) 80 Labelled Patterns



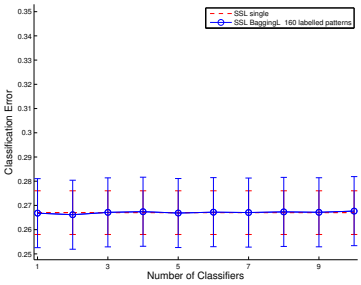
(e) 100 Labelled Patterns



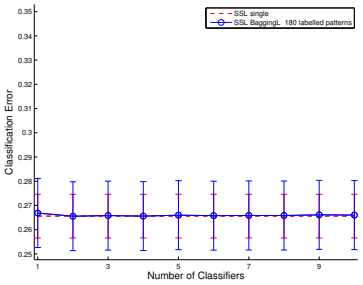
(f) 120 Labelled Patterns



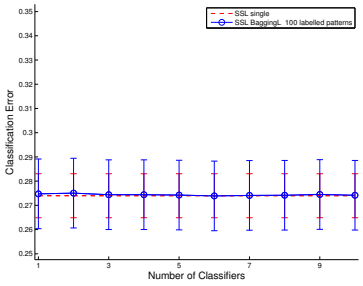
(g) 140 Labelled Patterns



(h) 160 Labelled Patterns

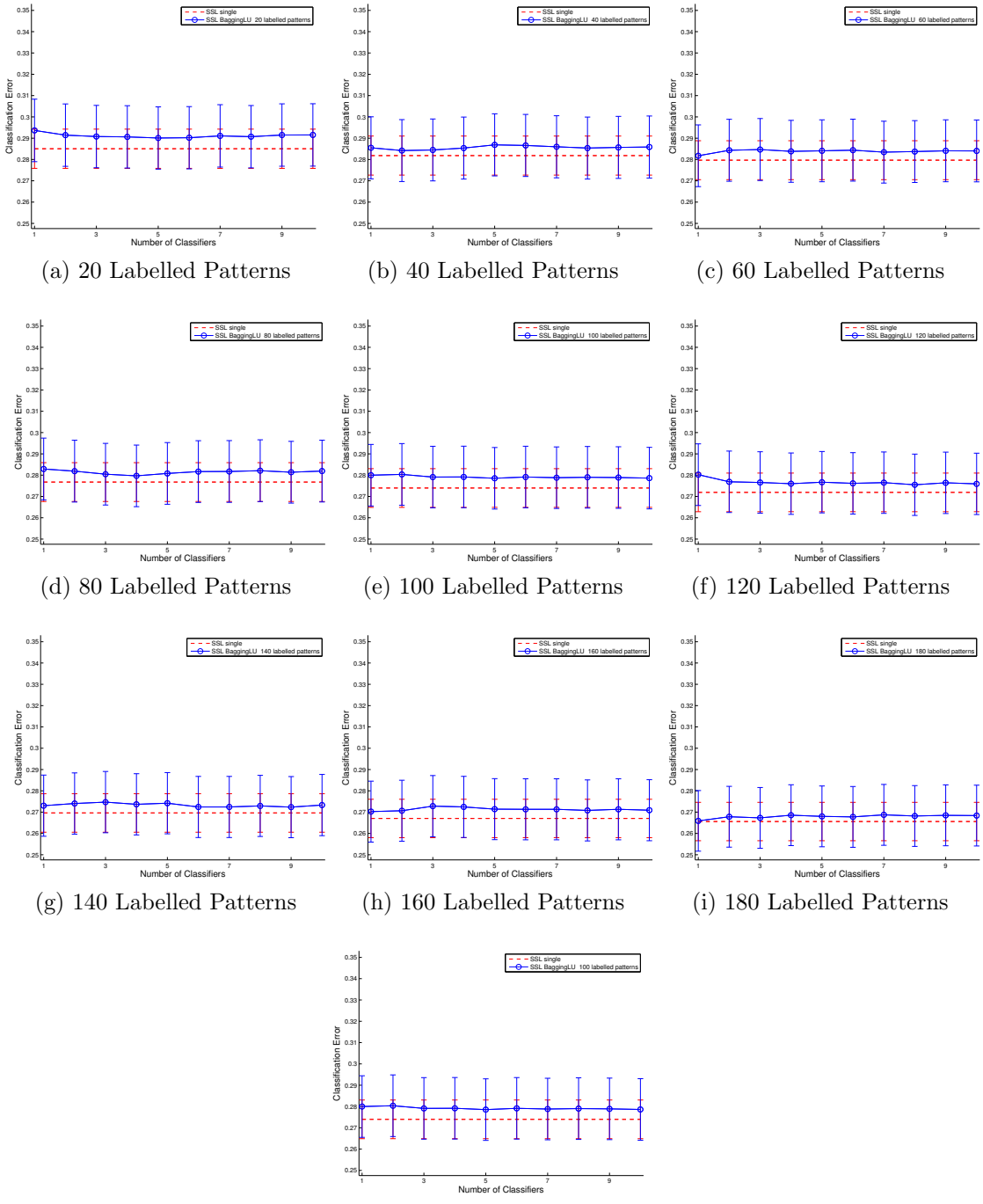


(i) 180 Labelled Patterns



(j) 200 Labelled Patterns

Figure A.10: Ringnorm dataset (Model mismatch problem) – Semi supervised Learning, BaggingL ensemble (blue continuous line) and single classifier (red dashed line) classification versus the size of the ensemble for increasing values of labelled data (20 to 200 patterns)



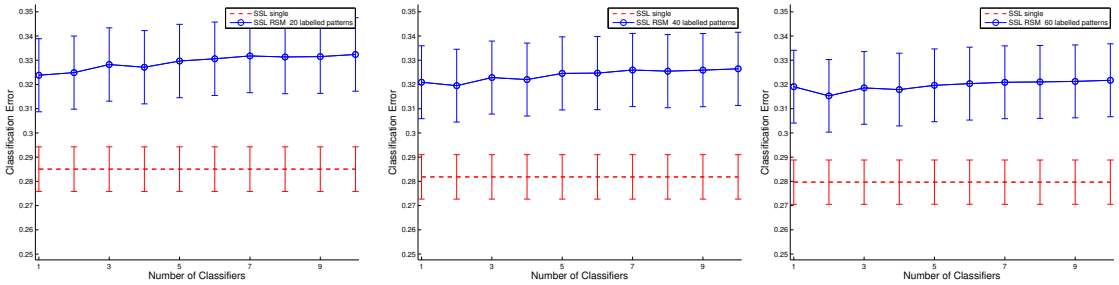
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

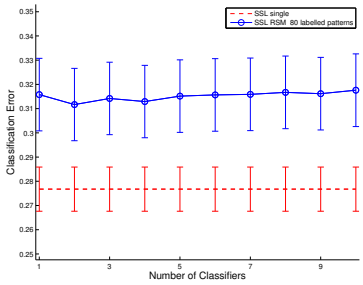
Figure A.11: Ringnorm dataset (Model mismatch problem) – Semi supervised Learning, BaggingLU ensemble (blue continuous line) and single classifier (red dashed line) classification versus the size of the ensemble for increasing values of labelled data (20 to 200 patterns)



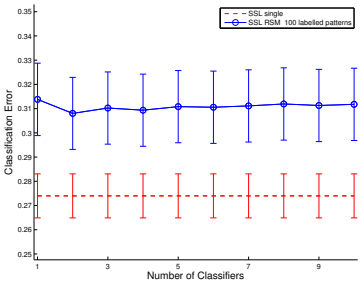
(a) 20 Labelled Patterns

(b) 40 Labelled Patterns

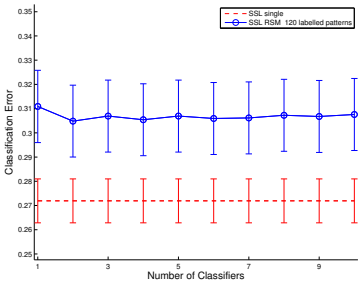
(c) 60 Labelled Patterns



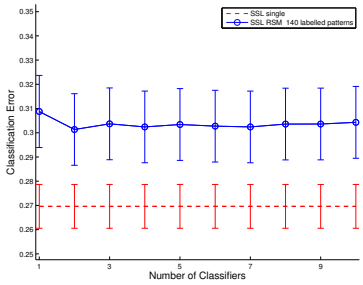
(d) 80 Labelled Patterns



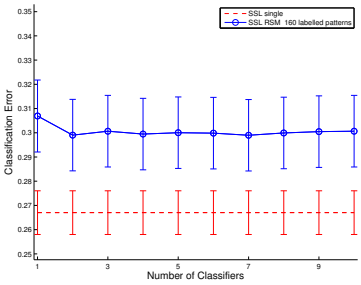
(e) 100 Labelled Patterns



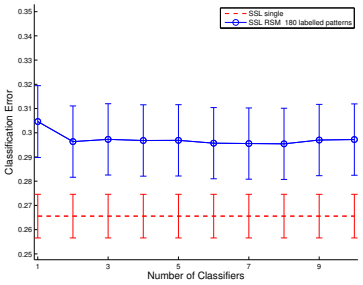
(f) 120 Labelled Patterns



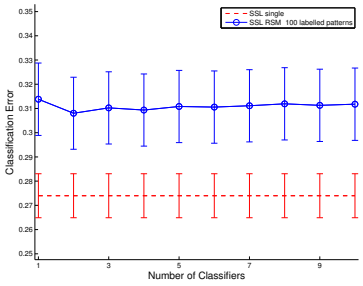
(g) 140 Labelled Patterns



(h) 160 Labelled Patterns

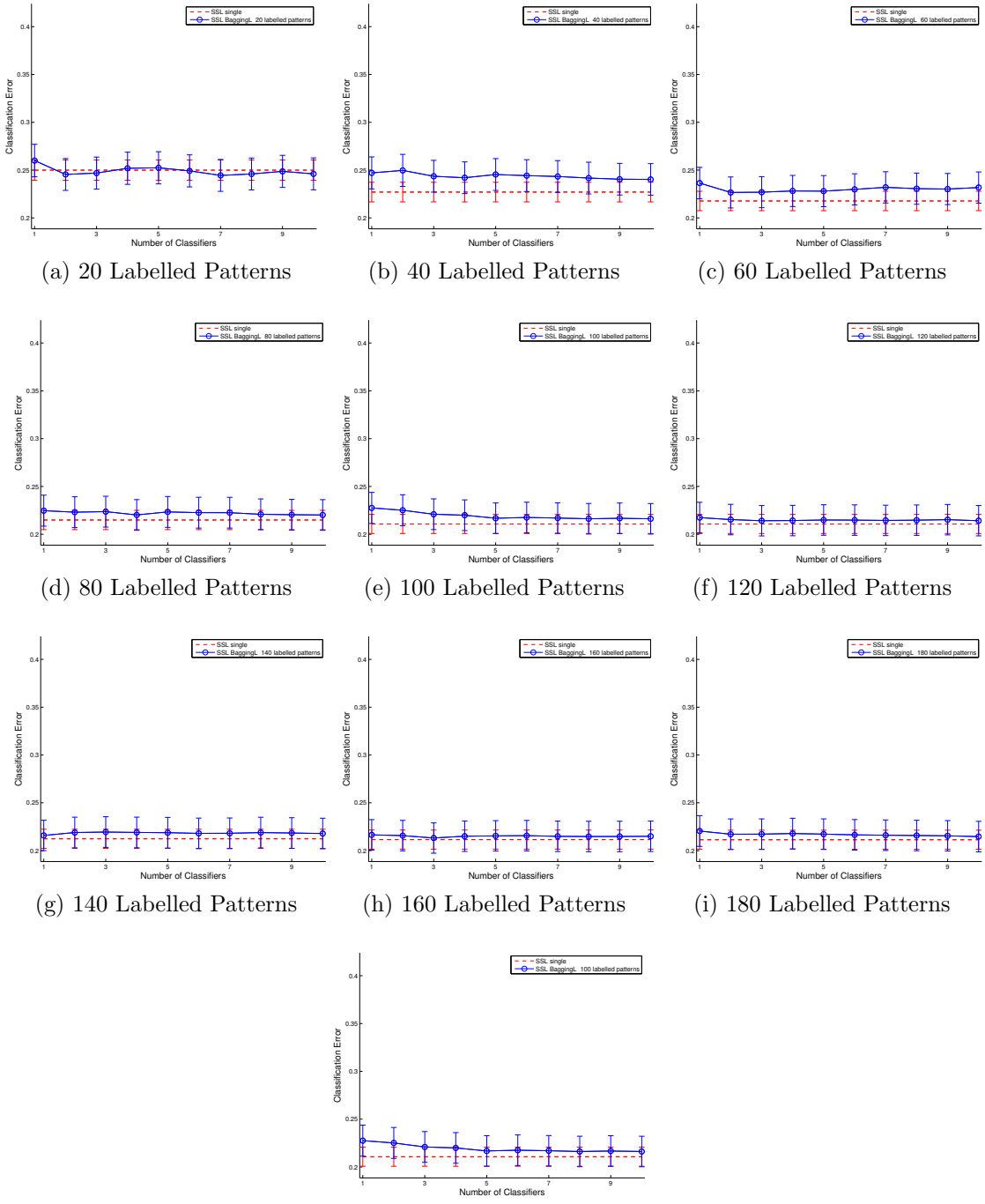


(i) 180 Labelled Patterns



(j) 200 Labelled Patterns

Figure A.12: Ringnorm dataset (Model mismatch problem) – Semi supervised Learning, RSM ensemble (blue continuous line) and single classifier (red dashed line) classification versus the size of the ensemble for increasing values of labelled data (20 to 200 patterns)



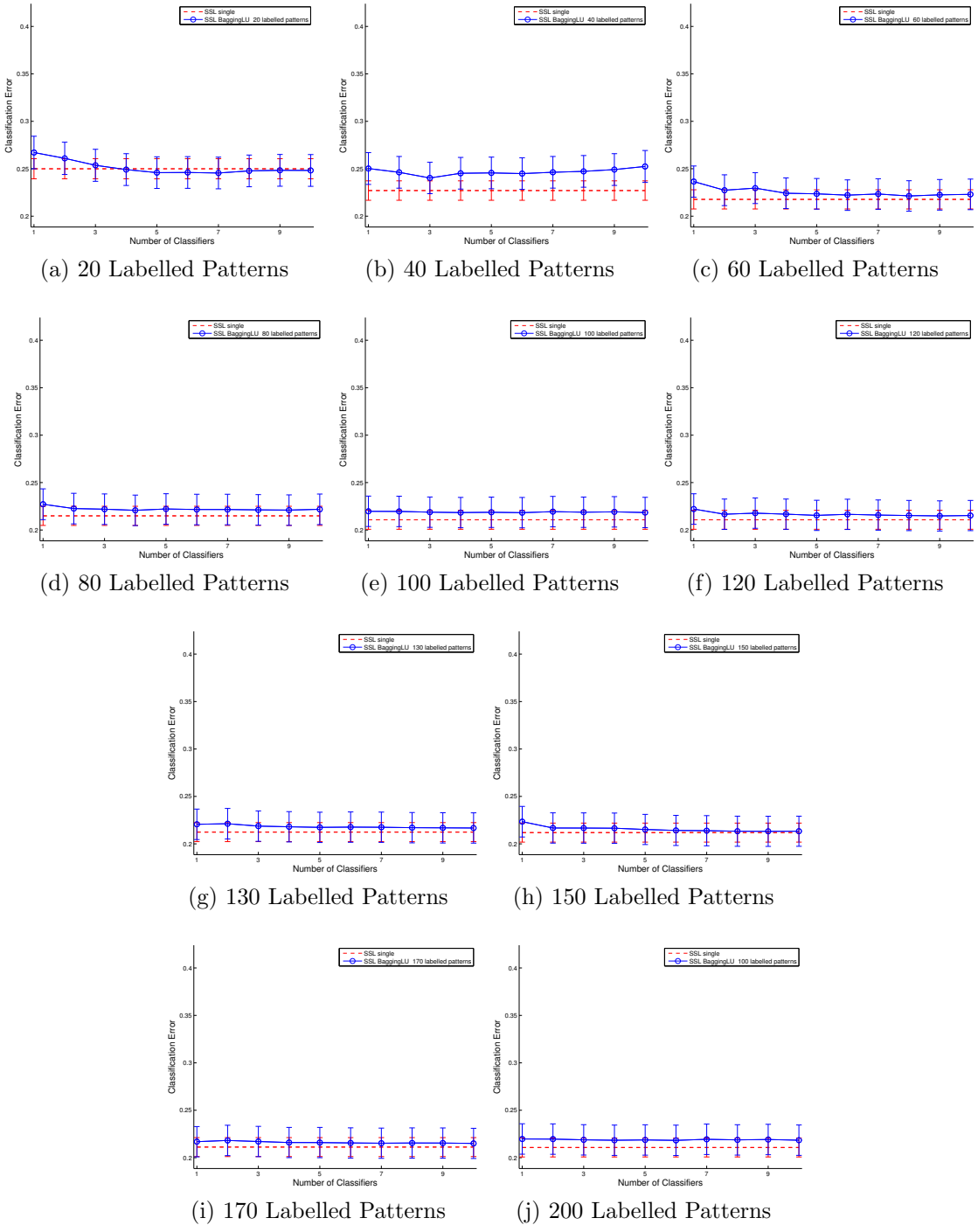
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.13: Feltwell dataset, semi supervised learning – Test error (mean and 95% confidence interval) of BaggingL (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



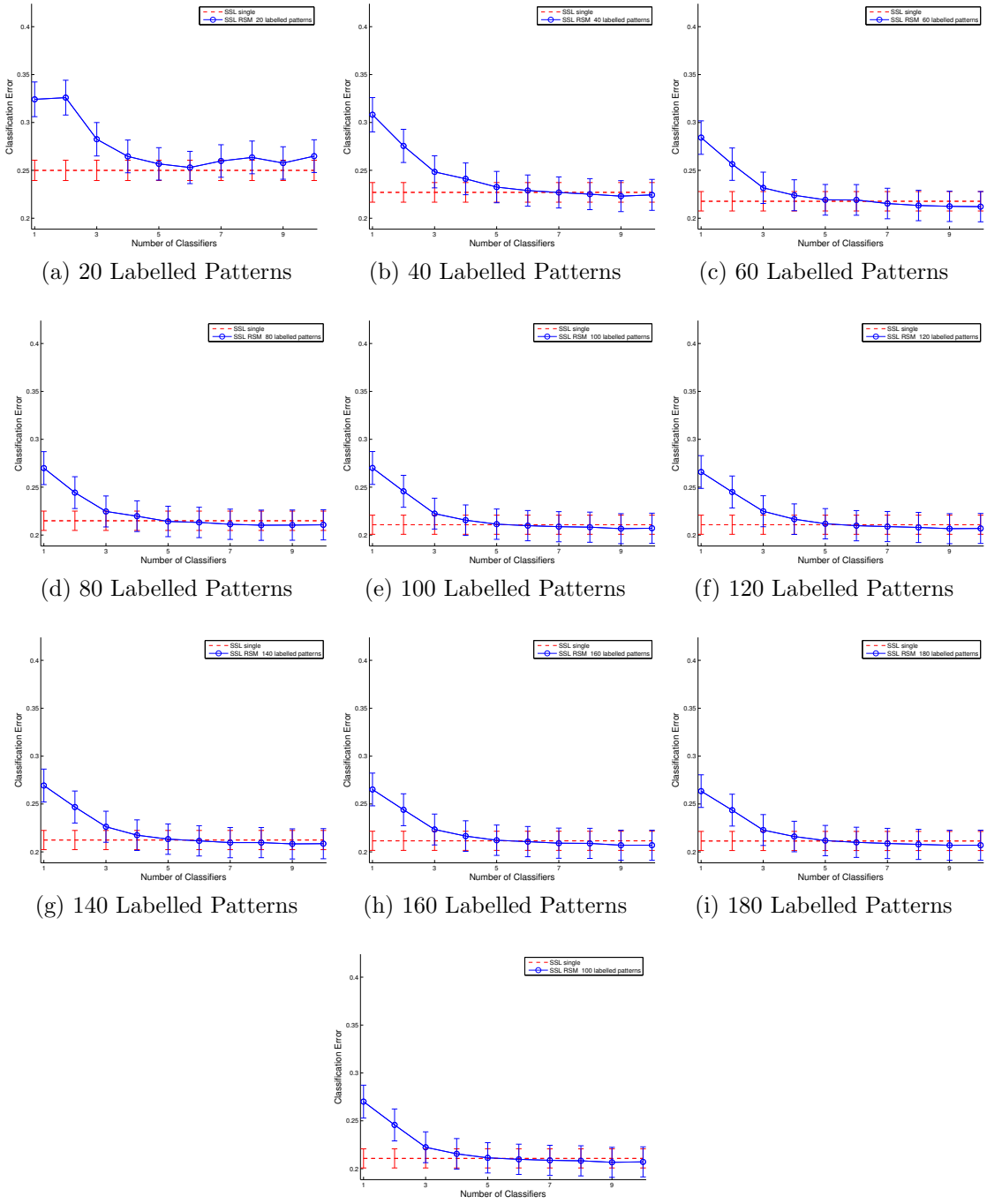
(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 130 Labelled Patterns (h) 150 Labelled Patterns

(i) 170 Labelled Patterns (j) 200 Labelled Patterns

Figure A.14: Feltwell dataset, semi supervised learning – Test error (mean and 95% confidence interval) of BaggingLU (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).



(a) 20 Labelled Patterns (b) 40 Labelled Patterns (c) 60 Labelled Patterns

(d) 80 Labelled Patterns (e) 100 Labelled Patterns (f) 120 Labelled Patterns

(g) 140 Labelled Patterns (h) 160 Labelled Patterns (i) 180 Labelled Patterns

(j) 200 Labelled Patterns

Figure A.15: Feltwell dataset, semi supervised learning – Test error (mean and 95% confidence interval) of RSM (blue continuous line) and test error (mean and 95% confidence interval) of the single classifier (red dashed line) as we increase the size of the ensemble, and for different amounts of labelled training data (20 to 200 patterns).

Appendix B

Test Errors for Ensembles of 50 ADE

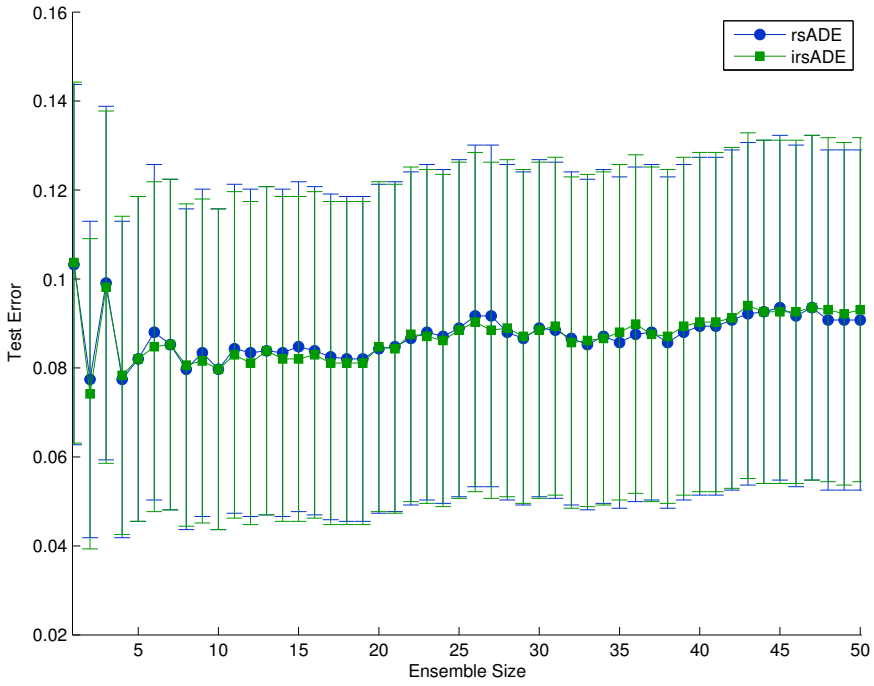


Figure B.1: Congress dataset: rsmADE vs irsmADE –50 classifiers– Test Error (mean and 95% confidence interval)

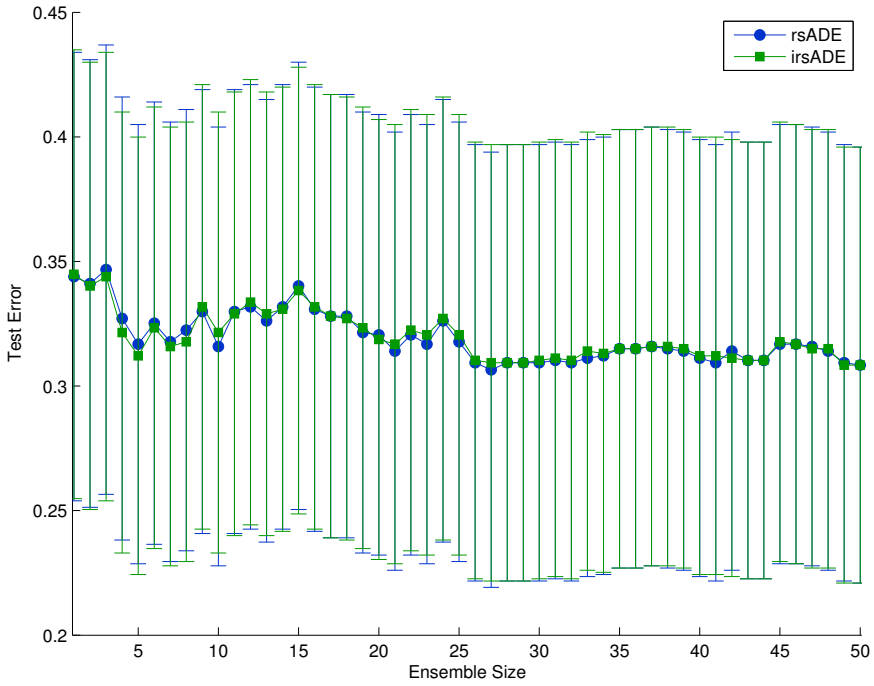


Figure B.2: Glass dataset: rsmADE vs irsmADE –50 classifiers– Test Error (mean and 95% confidence interval)

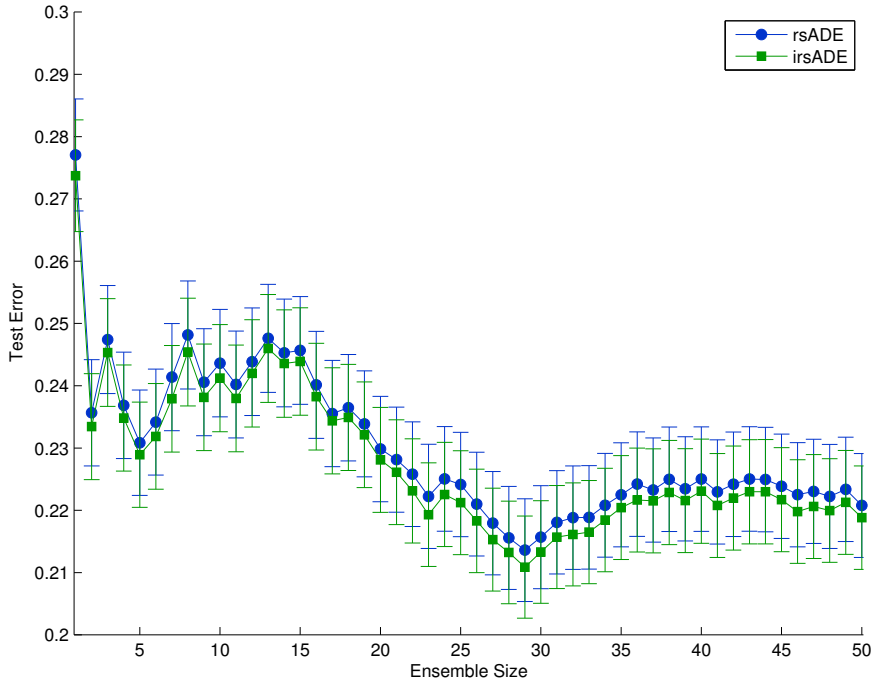


Figure B.3: Magic4dataset: rsmADE vs irsmADE –50 classifiers– Test Error (mean and 95% confidence interval)

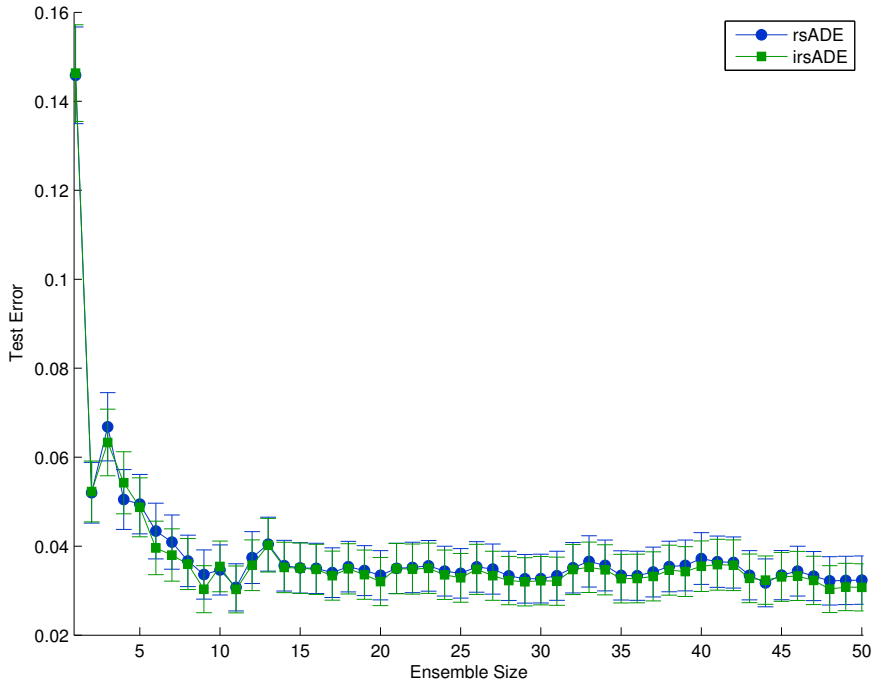


Figure B.4: Mushroom dataset: rsmADE vs irsmADE –50 classifiers– Test Error (mean and 95% confidence interval)

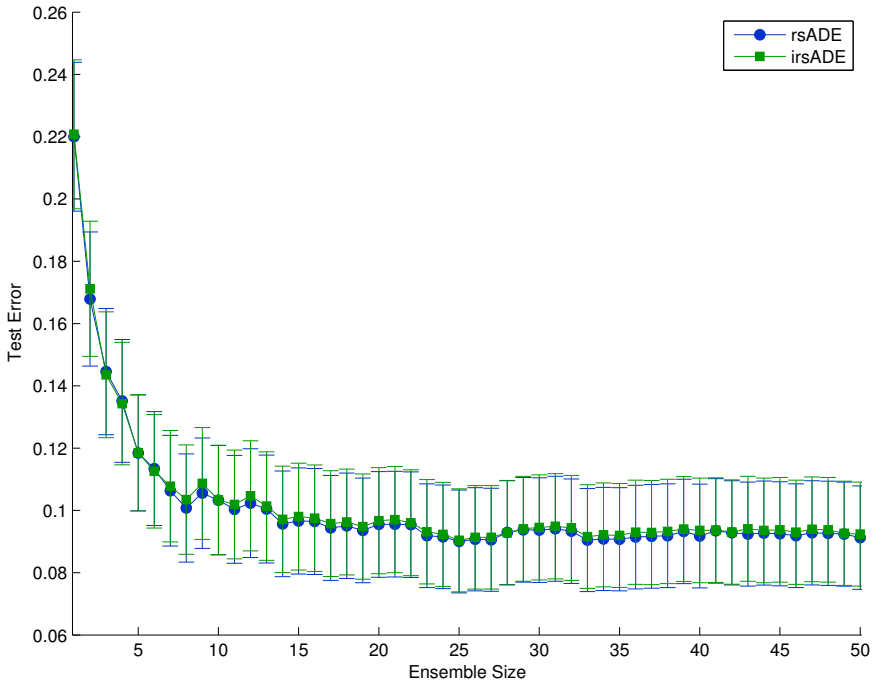


Figure B.5: Segment dataset: rsmADE vs irsmADE –50 classifiers– Test Error (mean and 95% confidence interval)

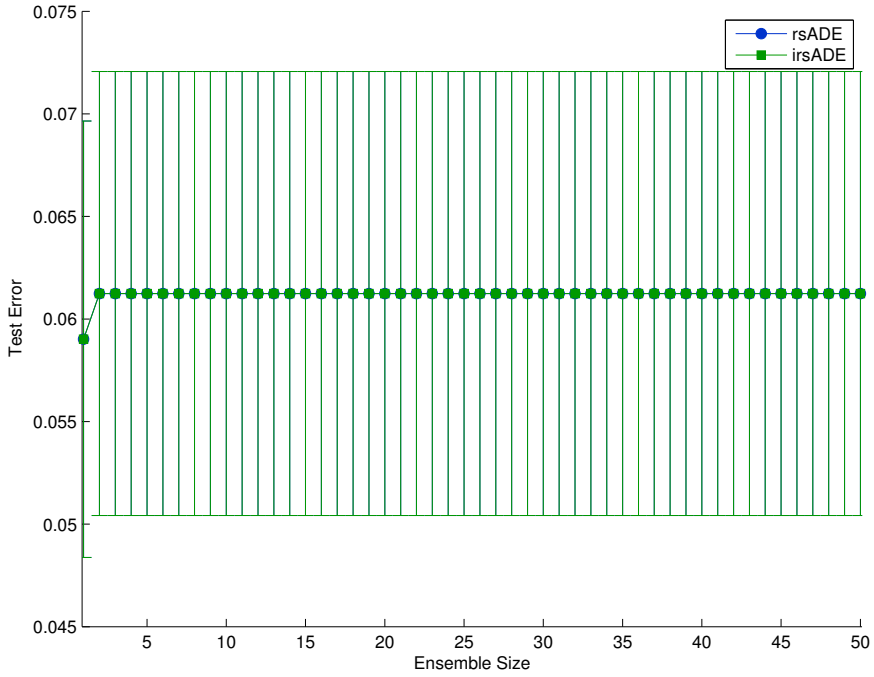


Figure B.6: Sickeuthyroid dataset: rsmADE vs irsmADE –50 classifiers– Test Error (mean and 95% confidence interval)

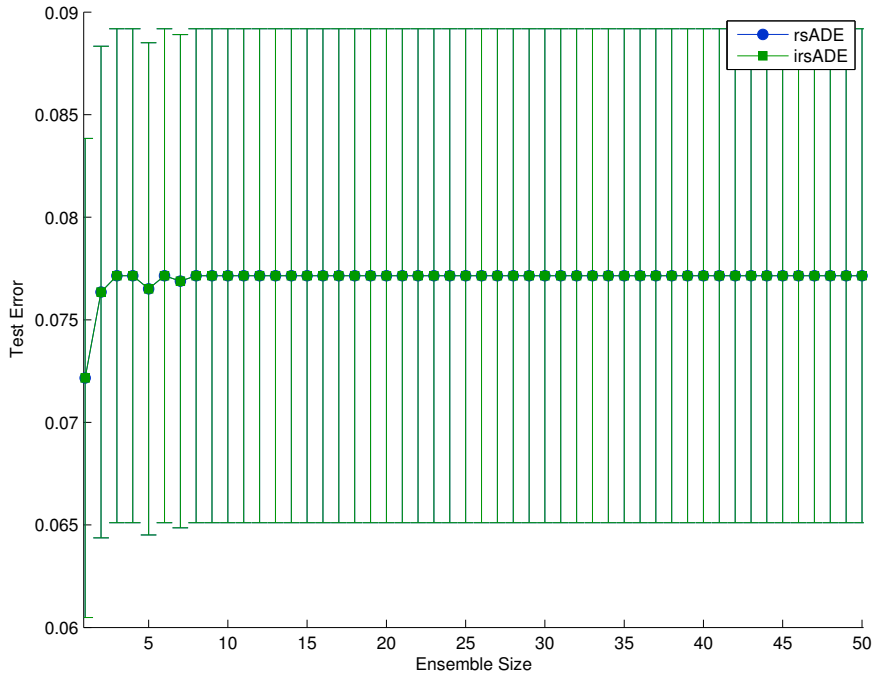


Figure B.7: Hypothyroid dataset: rsmADE vs irsmADE –50 classifiers– Test Error (mean and 95% confidence interval)

Appendix C

Test Errors for Single ADEs and Ensembles of ADE

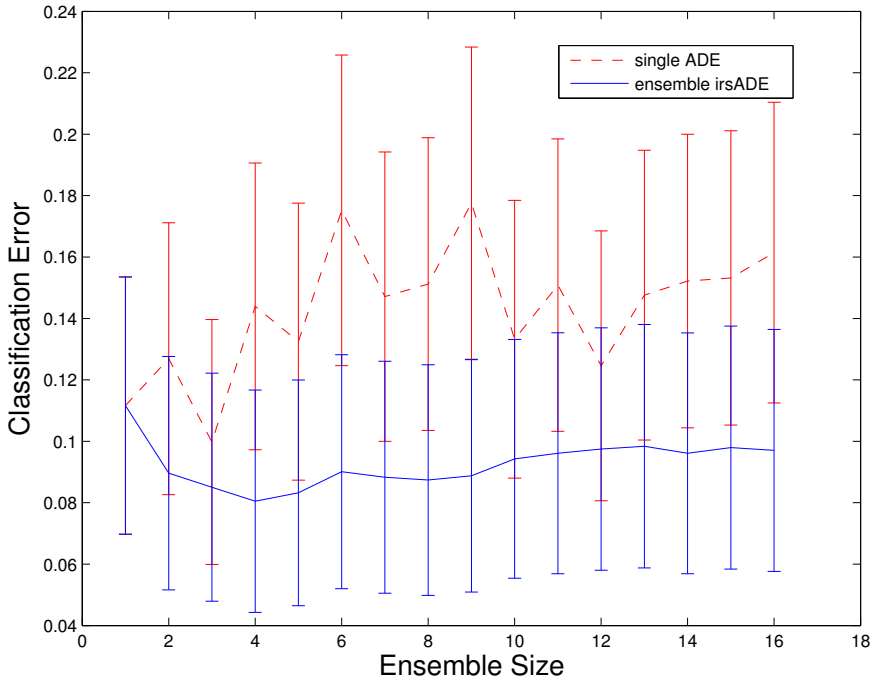


Figure C.1: Congress dataset: single ADE vs irsmADE –Test Error mean and 95% confidence interval

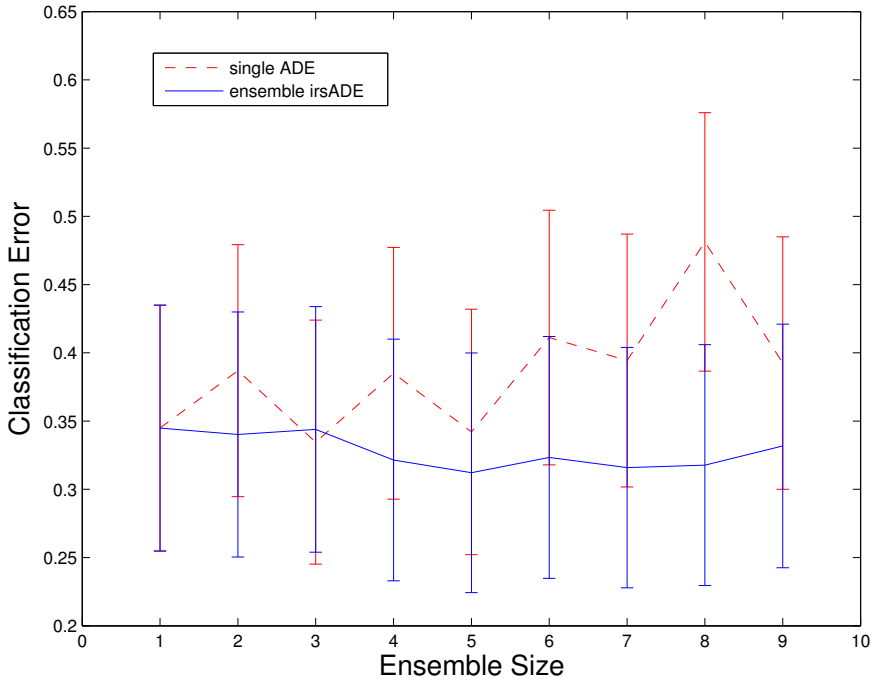


Figure C.2: Glass dataset: single ADE vs irsmADE –Test Error mean and 95% confidence interval

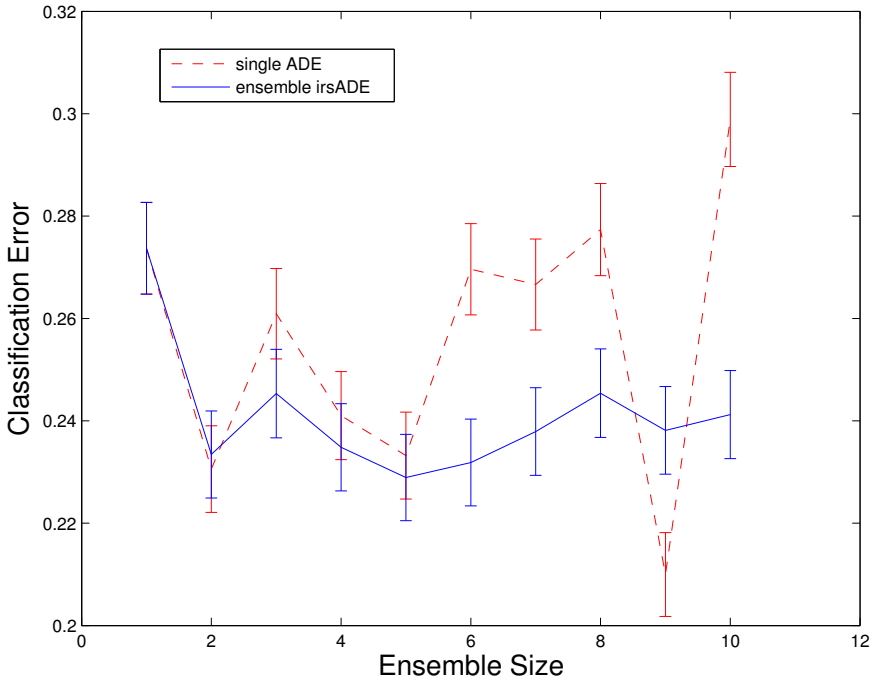


Figure C.3: Magic4dataset: single ADE vs irsmADE –Test Error mean and 95% confidence interval

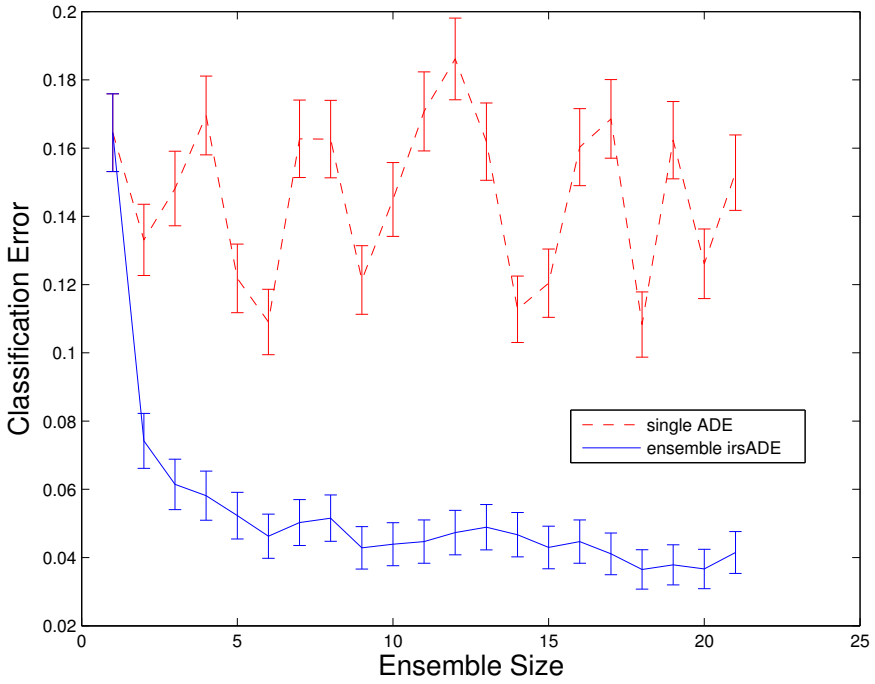


Figure C.4: Mushroom dataset: single ADE vs irsmADE –Test Error mean and 95% confidence interval

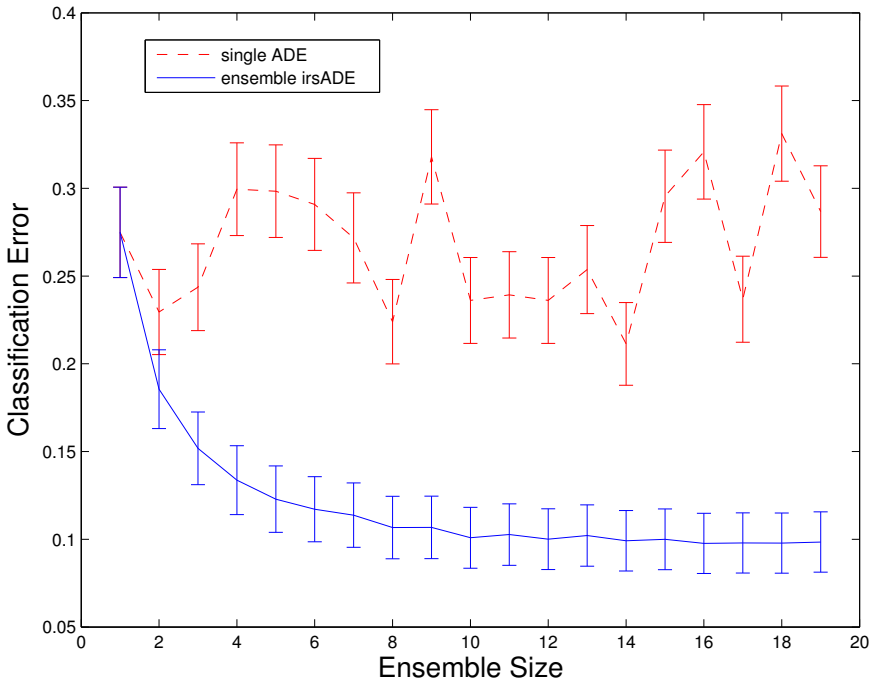


Figure C.5: Segment dataset: single ADE vs irsmADE –Test Error mean and 95% confidence interval

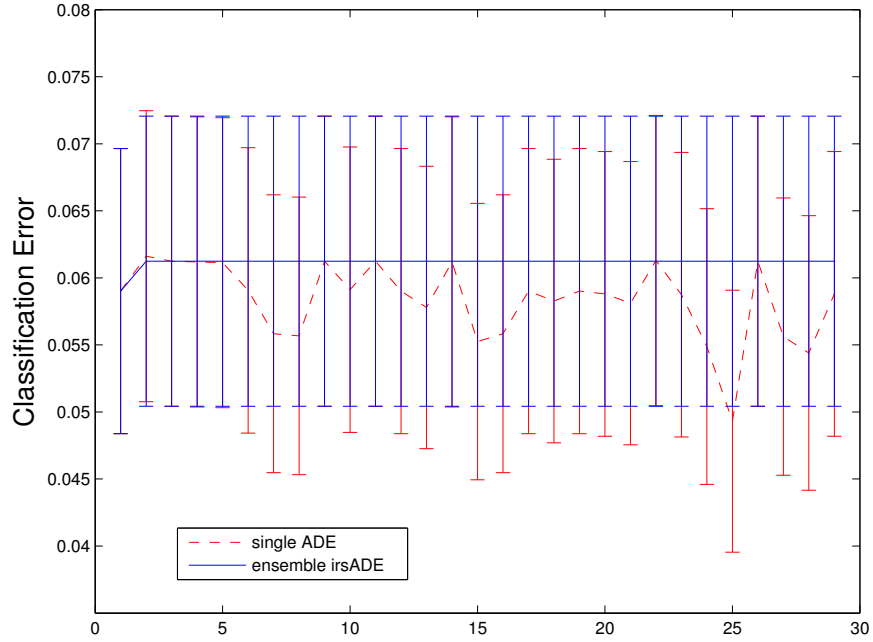


Figure C.6: Sickeuthyroid dataset: single ADE vs irsmADE –Test Error mean and 95% confidence interval

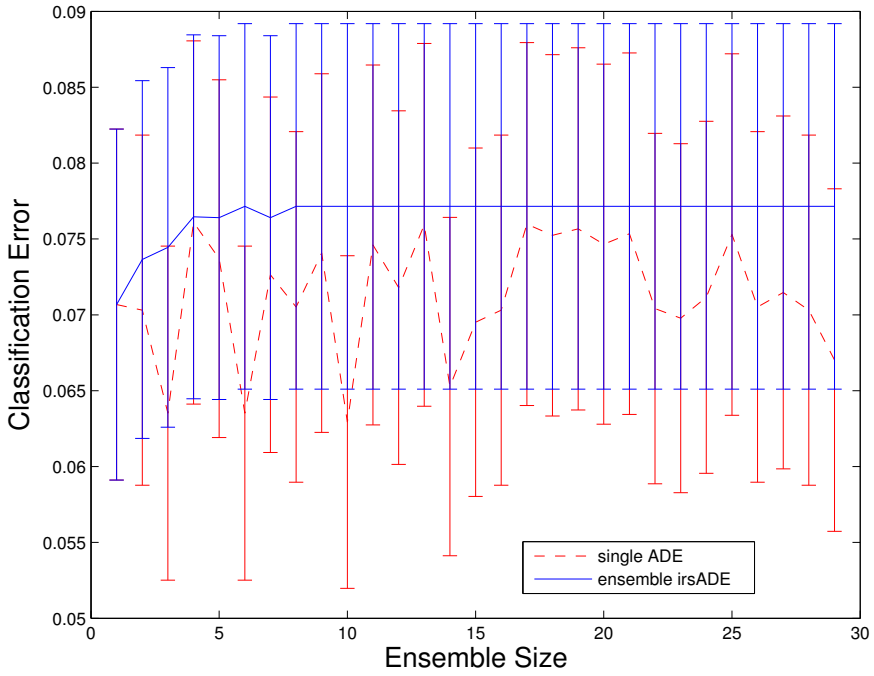


Figure C.7: Hypothyroid dataset: single ADE vs irsmADE –Test Error mean and 95% confidence interval

Bibliography

[1] Fuad M. Alkoot and Josef Kittler. Population bias control for bagging k-nn experts. In Belur V. Dasarathy, editor, *Sensor Fusion: Architectures, Algorithms, and Applications V*, volume 4385, pages 36–46. SPIE, 2001.

[2] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[3] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985.

[4] Battista Biggio, Giorgio Fumera, and Fabio Roli. Bayesian analysis of linear combiners. In *MCS '07, Proceedings of the 8th International Workshop on Multiple Classifier Systems*, pages 292–301. Springer, 2007.

[5] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1996.

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] Max Bramer. *Principles of Data Mining*. Springer, 2007.

[8] L. Breiman. Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6):2350–2383, 1996.

[9] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[10] Leo Breiman. Bias, Variance, and Arcing Classifiers. Technical report, Statistics Department, University of California, 1996.

[11] Gavin Brown. *Diversity in Neural Network Ensembles*. PhD thesis, School of Computer Science, University of Birmingham, 2004.

[12] Gavin Brown. An information theoretic perspective on multiple classifier systems. In *MCS '09: Proceedings of the 8th International Workshop on Multiple Classifier Systems*, pages 344–353, Berlin, Heidelberg, 2009. Springer-Verlag.

[13] Gavin Brown. A new perspective for information theoretic feature selection. In *12th International Conference on Artificial Intelligence and Statistics*, volume 5, pages 49–56, 2009.

[14] Gavin Brown and Ludmila I. Kuncheva. "Good" and "Bad" Diversity in Majority Vote Ensembles. In *MCS '10: Proceedings of the 9th International Workshop on Multiple Classifier Systems*, pages 124–133. Springer, 2010.

[15] Gavin Brown, Jeremy L. Wyatt, and Peter Tino. Managing diversity in regression ensembles. *Journal of Machine Learning Research*, 6:1621–1650, 2005.

[16] David M. Chickering, David Heckerman, and Christopher Meek. Large-sample learning of Bayesian networks is NP-hard. *The Journal of Machine Learning Research*, 5:1287–1330, 2004.

[17] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley, 1991.

[18] Fabio Gagliardi Cozman, Ira Cohen, and Marcelo Cesar Cirelo. Semi-supervised learning of mixture models. In *20th International Conference on Machine Learning*, pages 99–106, 2003.

[19] A. P. Dawid. Properties of diagnostic data distributions. *Biometrics*, 32(3):647–658, 1976.

[20] 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, 2000.

[21] Thomas G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.

[22] Pedro Domingos. A unified bias-variance decomposition for zero-one and squared loss. In *AAAI/IAAI*, pages 564–569, 2000.

[23] Pedro Domingos and Michael J. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2–3):103–130, 1997.

[24] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In Armand Prieditis and Stuart Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 194–202, 1995.

[25] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[26] Robert P. W. Duin and David M. J. Tax. Experiments with classifier combining rules. In *MCS '00, Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pages 16–29. Springer, 2000.

[27] B. Efron, R. Tibshirani, and R.J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall/CRC, 1993.

[28] C. Elkan. Boosting and Naïve Bayesian learning. *Proceedings of KDD-97, New Port Beach, CA*, 1997.

[29] Robert M. Fano. *Transmission of information: a statistical theory of communications*. M.I.T. Press & Wiley, London, 1961.

[30] Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference in AI*, pages 1022–1027, 1993.

[31] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[32] Nir Friedman, Dan Geiger, and Moisés Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2):131–163, 1997.

[33] Nir Friedman and Moisés Goldszmidt. Building classifiers using Bayesian networks. In *AAAI/IAAI, Vol. 2*, pages 1277–1284, 1996.

[34] Giorgio Fumera and Fabio Roli. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):942–956, 2005.

[35] Giorgio Fumera, Fabio Roli, and Alessandra Serrau. Dynamics of variance reduction in bagging and other techniques based on randomisation. In *MCS '05, Proceedings of the 6th International Workshop on Multiple Classifier Systems*, pages 316–325, 2005.

[36] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[37] Giorgio Giacinto and Fabio Roli. Dynamic classifier selection. In *MCS '00, Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pages 177–189. Springer, 2000.

[38] Giorgio Giacinto and Fabio Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9-10):699–707, 2001.

[39] Jiri Grim, Josef Kittler, Pavel Pudil, and Petr Somol. Combining multiple classifiers in probabilistic neural networks. In *MCS '00, Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pages 157–166. Springer, 2000.

[40] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[41] Sherif Hashem. Optimal Linear Combinations of Neural Networks. *Neural Networks*, 10(4):599–614, August 1997.

[42] Sherif Hashem. Treating harmful collinearity in neural network ensembles. *Combining Neural Networks*, pages 101–125, 1999.

[43] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2nd edition, 2008.

[44] Martin Hellman and Josef Raviv. Probability of error, equivocation, and the Chernoff bound. *IEEE Transactions on Information Theory*, 16(4):368–372, 1970.

[45] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[46] Robert A. Jacobs. Methods for combining experts' probability assessments. *Neural Computation*, 7(5):867–888, 1995.

[47] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[48] Gareth M. James. Variance and bias for general loss functions. In *Machine Learning*, pages 115–135, 2003.

[49] Eamonn J. Keogh and Michael J. Pazzani. Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, pages 225–230, 1999.

[50] Eamonn J. Keogh and Michael J. Pazzani. Learning the structure of augmented Bayesian classifiers. *International Journal on Artificial Intelligence Tools*, 11(4):587–602, 2002.

[51] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[52] Ron Kohavi and David H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In Lorenza Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 275–283. Morgan Kaufmann, 1996.

[53] E.B. Kong and T.G. Dietterich. Error-correcting output coding corrects bias and variance. In *Proceedings of the Twelfth International Conference on Machine Learning*, volume 313, page 321, 1995.

[54] Igor Kononenko. Semi-naïve Bayesian classifier. In *Proceedings of the European working session on learning on Machine learning*, pages 206–219. Springer, 1991.

[55] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, pages 231–238, 1995.

[56] Ludmila I. Kuncheva. That elusive diversity in classifier ensembles. *Pattern Recognition and Image Analysis*, pages 1126–1138, 2003.

[57] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley Press, 2004.

[58] Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

[59] Ludmila I. Kuncheva, Christopher J. Whitaker, and Catherine A. Shipp. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis & Applications*, 6(1):22–31, 2003.

[60] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *Proceedings of the National Conference on Artificial Intelligence*, pages 223–223, 1992.

[61] Nan Li, Yang Yu, and Zhi-Hua Zhou. Semi-naïve exploitation of one-dependence estimators. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM'09)*, Miami, FL, 2009.

[62] Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404, 1999.

[63] Dragos D. Margineantu and Thomas G. Dietterich. Pruning adaptive boosting. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 211–218, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[64] William J. McGill. Multivariate information transmission. *IEEE Transactions on Information Theory*, 4(4):93–111, September 1954.

[65] Adrew Ng and Michael Jordan. On generative vs. discriminative classifiers: A comparison of logistic regression and Naïve Bayes. In *Proceedings of Advances in Neural Information Processing*, volume 15, 2002.

[66] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3), 2000.

[67] Michael J. Pazzani. Searching for dependencies in Bayesian classifiers. In *International Workshop on Artificial Intelligence and Statistics*, pages 239–248. Springer-Verlag, 1996.

[68] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[69] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[70] Elzbieta Pekalska, Robert P. W. Duin, and Marina Skurichina. A discussion on the classifier projection space for classifier combining. In *MCS '02, Proceedings of the 3rd International Workshop on Multiple Classifier Systems*, pages 137–148. Springer, 2002.

[71] Galina Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

[72] Fabio Roli, Giorgio Giacinto, and Gianni Vernazza. Methods for designing multiple classifier systems. In *MCS '01, Proceedings of the 2nd International Workshop on Multiple Classifier Systems*, pages 78–87. Springer, 2001.

[73] Y. Dan Rubinstein and Trevor Hastie. Discriminative vs informative learning. In *KDD*, pages 49–53, 1997.

[74] Mehran Sahami. Learning limited dependence Bayesian classifiers. In *KDD*, pages 335–338, 1996.

[75] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–656, July, October 1948.

[76] Catherine A. Shipp and Ludmila I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3(2):135–148, 2002.

[77] D.B. Skalak. *Prototype selection for composite nearest neighbor classifiers*. PhD thesis, University of Massachusetts, Amherst, Massachusetts, 1997.

[78] Marina Skurichina and Robert P. W. Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135, 2002.

[79] Michiaki Taniguchi and Volker Tresp. Averaging regularized estimators. *Neural Computation*, 9(5):1163–1178, 1997.

[80] David M. J. Tax, Robert P. W. Duin, and M. van Breukelen. Comparison between product and mean classifier combination rules. In *In Proc. Workshop on Statistical Pattern Recognition*, pages 165–170, 1997.

[81] David M. J. Tax, Martijn van Breukelen, Robert P. W. Duin, and Josef Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern recognition*, 33(9):1475–1485, 2000.

[82] Robert Tibshirani. Bias, variance and prediction error for classification rules. Technical report, Department of Statistics, University of Toronto, 1996.

[83] Alexey Tsymbal, Seppo Puuronen, and David W. Patterson. Ensemble feature selection with the simple Bayesian classification. *Information Fusion*, 4(2):87–100, 2003.

[84] K. Tumer. *Linear and Order Statistics Combiners for Reliable Pattern Classification*. PhD thesis, The University of Texas, Austin, TX, May 1996.

[85] Kagan Tumer and Joydeep Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341–348, February 1996.

[86] Kagan Tumer and Joydeep Ghosh. Linear and order statistics combiners for pattern classification. In A. J. C. Sharkey, editor, *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, pages 127–162. Springer-Verlag, London, 1999.

[87] Naonori Ueda and Ryohei Nakano. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks*, pages 90–95, 1996.

[88] Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4(1):66–82, 1960.

[89] Geoffrey I. Webb, Janice R. Boughton, and Zhihai Wang. Not so naïve Bayes: aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.

[90] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition, 2005.

[91] D.H. Wolpert. The supervised learning no-free-lunch theorems. In *Proc. 6th Online World Conference on Soft Computing in Industrial Applications*, pages 25–42, 2001.

[92] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[93] Lei Xu, Adam Krzyzak, and Ching Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems Man and Cybernetics*, 22(3):418–435, 1992.

[94] Ying Yang, Kevin Korb, Kai Ting, and Geoffrey I. Webb. Ensemble selection for superparent-one-dependence estimators. *AI 2005: Advances in Artificial Intelligence*, pages 102–112, 2005.

[95] Ying Yang, Geoffrey I. Webb, Jesus Cerquides, Kevin Korb, Janice Boughton, and Kai Ming Ting. To select or to weigh: A comparative study of model selection and model weighing for spode ensembles. In *17th European Conference on Machine Learning (ECML)*, 2006.

[96] Ying Yang, Geoffrey I. Webb, Jesus Cerquides, Kevin B. Korb, Janice Boughton, and Kai Ming Ting. To select or to weigh: A comparative study of linear combination schemes for superparent-one-dependence estimators. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1652–1665, Dec. 2007.

[97] Raymond W. Yeung. A new outlook of Shannon's information measures. *IEEE Transactions on Information Theory*, 37(3):466–474, 1991.

[98] H. Peyton Young. Condorcet's theory of voting. *American Political Science Review*, 82(1231-1244), 1988.

[99] G.U. Yule. On the association of attributes in statistics: with illustrations from the material of the Childhood Society, &c. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 194:257–319, 1900.

[100] Manuela Zanda and Gavin Brown. A study of semi-supervised generative ensembles. In *MCS '09, Proceedings of the 8th International Workshop on Multiple Classifier Systems*, pages 242–251. Springer, 2009.

[101] Manuela Zanda, Gavin Brown, Giorgio Fumera, and Fabio Roli. Ensemble learning in linearly combined classifiers via negative correlation. In *MCS '07, Proceedings of the 7th International Workshop on Multiple Classifier Systems*, pages 440–449. Springer, 2007.

[102] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2008.