Probabilistic Multiple Kernel Learning

by

Theodoros Damoulas

A dissertation submitted to

The Department of Computing Science
of
The University of Glasgow

for the degree of

Doctor of Philosophy

October 2009

Abstract

The integration of multiple and possibly heterogeneous information sources for an overall decision-making process has been an open and unresolved research direction in computing science since its very beginning. This thesis attempts to address parts of that direction by proposing *probabilistic* data integration algorithms for *multiclass* decisions where an observation of interest is assigned to one of many categories based on a *plurality* of information channels.

Motivation for this thesis, from an application perspective, comes from the Automatic Currency Validation setting where the problem is to automatically classify currency notes, deposited in an Automated Teller Machine, to one of multiple classes while utilising information from multiple sensors. The adopted Bayesian probabilistic framework is motivated by the requirements for assessing decision-making costs, formal inclusion of prior knowledge and principled model selection. Requirements that are common across many fields, such as bioinformatics and robotics, where multiple sources of information are available for a multiclass classification decision.

There is a single light of science,
and to brighten it anywhere is
to brighten it everywhere.

Isaak Asimov

Acknowledgements

This thesis would not have been possible without the help and support of many people, to only some of whom it is possible to give mention and acknowledge here.

First of all I would like to thank my supervisor Prof. Mark Girolami for his guidance and support, for giving me directions and keeping me on track, and for his patience and understanding. Furthermore, I would like to thank him for giving me the opportunity to apply for the RAEng fellowship which taught me other aspects of research and academia and also to acknowledge the additional funding for the write up months that he provided. Thank you for everything Mark, I hope to make you proud in the future.

I was very lucky to also have the guiding hand of "Uncle Keith", my second supervisor Prof. C. J. Keith van Rijsbergen. I would like to thank him for being a role model for me, for giving me the appropriate hard time on my yearly progression examinations and for introducing me to the gems of I. J. Good and Bruno de Finetti.

Dr. Simon Rogers also had a significant impact on this thesis and my development through numerous discussions on Multiple Kernel Learning, support and direct feedback in many levels including corrections on drafts of this thesis and on my fellowship. I would like to thank him for also being a good friend throughout these years.

During this period I had the pleasure to collaborate with Dr. Colin Campbell and Dr. Yiming Ying from the University of Bristol. I would like to thank them for the numerous discussions and great time we had while developing some of the algorithms in this thesis.

This thesis was funded by NCR Financial Solutions Ltd and I benefited from interaction with research engineers from the NCR Labs. I would like to thank especially Dr. Chao He and Dr. Gary Ross for their help, support and for providing the necessary datasets and currency images.

Furthermore, thanks are due to the whole of the Inference Research Group and people from the Information Retrieval Group for their friendship, company and help. In particular, Ben, Billy, Dom, Iraklis, Keith, Tamara and Vlad (alphabetical order) helped me with research matters such as ESS scripts, thesis templates, programming, statistical advice, scientific discussions and coffee (beer) breaks. During the last year I had the pleasure of supervising the MSc

IT thesis of Yannis Psorakis and some of our work is included in this thesis. It is always a joy to interact with a great student and also gain a friend in the process.

On the final research side of things, I would like to thank my external and internal examiners Dr. Guido Sanguinetti and Dr. Paul Siebert for agreeing to read my thesis in due time and hence help me to a smoother transition from Glasgow to Cornell University while meeting the requirements and deadlines.

On a more personal and less technical note, I had the luck of being surrounded with people that supported me intellectually, emotionally and financially through these times. I owe a big thank you to my "extended" family (Damouleiko, Spaneiko, Tsakireiko) and my friends from back home (P.Club) and abroad. Special thanks to Vassiliki Grammenou for providing coffee and helping me during the correction phase. I also had the pleasure of meeting some new people this period that I would like to thank for their friendship and backing, especially Prof. Christos Papatheodorou and Periklis Vandoros. Finally, during the last year I was very lucky to have the attention and care of a lovely girl called Tamara Polajnar who has also helped me a lot in many ways, including help in binding and submitting this thesis.

Dedicated to my mother Ioanna Tsakiri

–Στην K.N.T.Δ

Declaration

All the work reported in this thesis has been performed by myself, unless specifically stated otherwise.

Theodoros Damoulas

October 2009.

Notation

Symbols

$\mathbb{R}^D$	Real D - dimensional space.
$\mathbb{R}^{D_s}$	The real D_s - dimensional space of information source $s \in \{1, \ldots, S\}$.
$\mathbb{R}^{N \times D}$	Real $N \times D$ - dimensional space.
$\mathbb{N}$	The set of natural numbers (positive integers).
x	Scalar $\in \mathbb{R}$.
$\mathbf{x}$	Column vector $\in \mathbb{R}^D$.
$\mathbf{X}$	Matrix $\in \mathbb{R}^{N \times D}$.
$\mathbf{x}^{(s)}$	Column vector $\mathbf{x} \in \mathbb{R}^{D_s}$ from the s^{th} information source.
$\mathbf{w}_c$	The c^{th} column vector of matrix[1] $\mathbf{W} \in \mathbb{R}^{N \times C}$.
x^α	x raised to the α power.
x_*	An "unseen" or new x.
$p(\mathbf{z})$	Probability density function (p.d.f) of $\mathbf{z}$.
$p(\mathbf{z}\|\mathbf{y})$	Conditional p.d.f of $\mathbf{z}$ given $\mathbf{y}$.
$p(\mathbf{z}, \mathbf{y})$	Joint p.d.f of $\mathbf{z}$ and $\mathbf{y}$.
$\mathbf{z} \sim p(\mathbf{z})$	$\mathbf{z}$ is distributed according to $p(\mathbf{z})$.
$\mathcal{O}(N)$	The computational complexity is order N operations.

Operators and functions

$\mathbf{A}^\mathsf{T}$	Transpose of matrix $\mathbf{A}$.
$\mathbf{A}^{-1}$	Inverse of matrix $\mathbf{A}$.
$\text{Tr}[\mathbf{A}]$	Trace of matrix $\mathbf{A}$.
$\lvert\mathbf{A}\rvert$	Determinant of matrix $\mathbf{A}$.
δ_i	Dirac delta function (impulse function).
$\mathbb{E}_{p(\mathbf{z})}(\mathbf{z})$	Expectation of the random variable $\mathbf{z}$ wrt. $p(\mathbf{z})$.
$\exp(\cdot)$	Exponential function.
$\log(\cdot)$	Naperian logarithmic function (ln).
$\min, \max$	Extrema with respect to an integer value.
$\underset{x}{\text{argmax}}$	The argument x that maximizes the operand.
$\underset{x}{\text{argmin}}$	The argument x that minimizes the operand.

[1] To simplify the notation we denote $\mathbf{w}_c$ as an equivalent to $\mathbf{W}_{1:N,c}$ and as an N dimensional column vector. That is, every vectorial representation will be denoted by lower-case bold (vice versa) and if an index is not appearing we are referring to all of its possible values. All vectors are column vectors.

Standard probability distributions

Binomial	$\mathcal{B}_k(n,p)$	$\binom{n}{k}p^k(1-p)^{n-k}$
Dirichlet	$\mathcal{D}_{\mathbf{x}}(\boldsymbol{\rho})$	$\frac{\Gamma\left(\sum_{i=1}^{S}\rho_i\right)}{\prod_{i=1}^{S}\Gamma(\rho_i)}\prod_{i=1}^{S}x_i^{\rho_i-1}$ with $\mathbf{x},\boldsymbol{\rho}\in\mathbb{R}^S$
Exponential	$\mathcal{E}_x(\lambda)$	$\lambda\exp(-\lambda x)$
Gamma	$\mathcal{G}_x(\alpha,\beta)$	$\frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}\exp(-\beta x)$
Gaussian	$\mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu},\boldsymbol{\Sigma})$	$\lvert 2\pi\boldsymbol{\Sigma}\rvert^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$ with $\mathbf{x},\boldsymbol{\mu}\in\mathbb{R}^N$ and $\boldsymbol{\Sigma}\in\mathbb{R}^{N\times N}$
Inverse Gamma	$\mathcal{IG}(\alpha,\beta)$	$\frac{\beta^\alpha}{\Gamma(\alpha)}x^{-\alpha-1}\exp(-\beta/x)$

Abbreviations

ANN	Artificial Neural Network.
ARD	Automatic Relevance Determination.
CDF	Cumulative Distribution Function.
CPU	Central Processing Unit.
EM	Expectation Maximisation.
GLM	Generalized Linear Model.
GP	Gaussian Process.
i.i.d	Independent and Identically Distributed.
IVM	Informative Vector Machine.
KLD	Kullback Leibler Divergence.
MAP	Maximum A Posteriori.
MCMC	Markov Chain Monte Carlo.
MH	Metropolis Hastings.
MKL	Multiple Kernel Learning.
ML	Maximum Likelihood.
p.d.f	Probability Density Function.
p.s.d	Positive Semi-Definite.
QP	Quadratic Programming.
RBF	Radial Basis Function.
RVM	Relevance Vector Machine.
SVM	Support Vector Machine.

Terminology

$\mathbf{x}$	Input sample, input variable, predictor, regressor.
$\mathbf{t},\mathbf{y}$	Response, dependent variable, output, target, label.

Contents

1 Introduction **22**
1.1 Learning from Multiple Sources 22
1.2 Contributions 23
1.3 Thought Process 25
1.4 Thesis Structure 26

2 Introduction to Multiple Kernel Learning **27**
2.1 Linear Regression and Nonlinear Responses 28
2.2 Learning and Bayesian Inference 30
2.2.1 Statistical Learning Theory 30
2.2.2 Towards Bayesian Inference 32
2.2.3 Bayesian Inference 33
2.3 The Kernel Trick and Kernel Regression 36
2.4 Classification 39
2.4.1 Logistic and Probit Regression 40
2.5 Markov Chain Monte Carlo 41
2.5.1 Importance Sampling 42
2.5.2 Metropolis Sampling 44
2.5.3 Metropolis-Hastings Sampling 44
2.5.4 Gibbs Sampling 45
2.6 Deterministic Approximations 46
2.6.1 Saddle-point (Laplace) Approximation 46
2.6.2 Variational Free Energy Minimisation 52
2.7 Sparsity and Shrinkage methods 54
2.7.1 Ridge Regression and the Lasso 55
2.7.2 Sparsity in Kernel Methods 56
2.7.3 Sparsity in Bayesian Inference 56

2.8 Ensemble Learning . . . 58
2.8.1 Classifier Combination . . . 59
2.8.2 Multiple Kernel Learning . . . 61

3 Probabilistic Multiple Kernel Learning 67
3.1 Introduction . . . 67
3.2 Constructing the Composite Kernel . . . 67
3.2.1 Fixed Combination . . . 68
3.2.2 Convex Linear Combination . . . 68
3.2.3 Binary Combination . . . 69
3.2.4 Product Combination . . . 69
3.2.5 Weighted Product Combination . . . 69
3.2.6 Theoretical Justification of Kernel Combinations . . . 70
3.3 Multinomial Probit Kernel Regression . . . 70
3.3.1 Multinomial Probit Likelihood . . . 71
3.3.2 Gauss-Hermite Quadrature . . . 73
3.3.3 Prior distributions and the graphical model . . . 73
3.4 Markov Chain Monte Carlo
Posterior Inference . . . 77
3.4.1 Gibbs Sampler . . . 77
3.4.2 Metropolis Hastings Sampler . . . 80
3.5 Marginal Likelihood for Model Selection . . . 82
3.6 Comparison of MCMC Sampling Schemes . . . 83
3.7 Toy Example Demonstration . . . 86
3.8 Computational Complexity . . . 87
3.9 Discussion . . . 87

4 Variational Bayes Inference 90
4.1 Mean Field Theory . . . 91
4.1.1 Variational Mean Field Theory for Classification . . . 91
4.2 Variational Bayes Probabilistic Multiple Kernel Learning . . . 93
4.2.1 $Q(\mathbf{Y})$: Approximate posterior for $\mathbf{Y}$ . . . 95
4.2.2 $Q(\mathbf{W})$: Approximate posterior for regression coefficients $\mathbf{W}$ 96
4.2.3 $Q(\mathbf{A})$: Approximate posterior of scales $\mathbf{A}$ . . . 97
4.2.4 $Q(\boldsymbol{\Theta})$: Approximate posterior for $\boldsymbol{\Theta}$ . . . 98
4.2.5 $Q(\boldsymbol{\beta})$: Approximate posterior for $\boldsymbol{\beta}$ . . . 98

4.2.6 $Q(\boldsymbol{\rho})Q(\boldsymbol{\pi})Q(\boldsymbol{\chi})$: Approximate posteriors for $\boldsymbol{\rho}, \boldsymbol{\pi}, \boldsymbol{\chi}$ . . . 99
4.2.7 Predictive Distribution . . . 100
4.3 Convergence and the Lower Bound . . . 101
4.4 Computational Complexity . . . 102
4.5 Variational Inference and Gibbs Sampling . . . 103
4.5.1 Synthetic Data sets . . . 103
4.6 Multinomial UCI Experiments . . . 107
4.7 Discussion . . . 108

5 MAP Estimators and mRVMs 110
5.1 MAP Estimation and EM Update Schemes . . . 111
5.2 Sparsity and Relevance Vector Machines . . . 114
5.3 Multiclass Multi-kernel Relevance Vector Machines . . . 115
5.4 Model Formulation . . . 116
5.4.1 mRVM_1 . . . 117
5.4.2 Computational Efficiency for mRVM_1 . . . 120
5.4.3 Informative Sample Selection for mRVM_1 . . . 121
5.4.4 Initialisation and Convergence for mRVM_1 . . . 123
5.4.5 mRVM_2 . . . 125
5.4.6 Initialisation and Convergence Criteria for mRVM_2 . . . 126
5.5 Preliminary Experimental Evaluation . . . 126
5.5.1 Experimental Setup . . . 127
5.5.2 Non-sparse Comparison . . . 127
5.5.3 Sparse Comparison . . . 128
5.5.4 Convergence, Sparsity and Predictive Power . . . 129
5.6 Discussion . . . 138

6 Automatic Currency Validation 140
6.1 Motivation . . . 140
6.2 ACV Literature Review . . . 142
6.2.1 Recognition and Verification of Currency . . . 142
6.3 ACV with Multiple Sources of Information . . . 148
6.4 Feature Extraction . . . 150
6.4.1 Image Channels . . . 150
6.4.2 Non-Image Channels . . . 151
6.5 Covariate Ranking . . . 151

6.5.1 Binary Classification . . . 152
6.5.2 Multinomial Classification . . . 156
6.6 VBpMKL Results . . . 157
6.6.1 Image Integration . . . 158
6.6.2 Image and Non-Image Integration . . . 163
6.7 mRVM Results . . . 167
6.8 Discussion . . . 171

7 Further Large Scale Applications 173
7.1 Handwritten Numeral Recognition . . . 174
7.1.1 Multiple Features Dataset: Gibbs Sampling . . . 175
7.1.2 Multiple Features Dataset: Variational Bayes . . . 180
7.2 Protein Fold Recognition . . . 182
7.2.1 Experimental Setup . . . 184
7.2.2 Results and Discussion . . . 186
7.3 Remote Homology Detection . . . 190
7.4 Protein Subcellular Localisation . . . 191
7.5 Discussion . . . 194

8 Diversity in Multiple Kernel Learning 196
8.1 The Flat Maximum Effect . . . 197
8.1.1 Linear regression model . . . 197
8.1.2 Extension to Multiple Kernel Learning . . . 199
8.2 The Ambiguity Decomposition . . . 200
8.3 Bias-Variance-Covariance Decomposition . . . 201
8.4 Diversity and Information . . . 203
8.5 Fisher Information for MKL . . . 205
8.5.1 Fisher Information of $\boldsymbol{\beta}$ . . . 206
8.5.2 Fisher Information of the regression coefficients $\mathbf{w}$ . . . 206
8.5.3 Maximisation of the Fisher Information . . . 207
8.6 Discussion . . . 209

9 Conclusions and Future Research Directions 210
9.1 Future Research Directions . . . 212

A Posterior Inference in MCMC **214**
A.1 Kernel Combination Parameters 214
A.1.1 Convex Linear Combination 214
A.1.2 Weighted Product Combination 214
A.1.3 Binary Combination 215
A.2 Kernel Parameters 216

B Variational Approximations **217**
B.1 Approximate posterior distributions 217
B.1.1 $Q(\mathbf{Y})$ 217
B.1.2 $Q(\mathbf{W})$ 218
B.1.3 $Q(\mathbf{A})$ 219
B.1.4 $Q(\boldsymbol{\beta}), Q(\boldsymbol{\rho}), Q(\boldsymbol{\Theta})$ 219
B.2 Posterior Expectations for the Auxiliary Variables 221
B.3 Predictive distribution 222
B.4 Lower bound 224

C Quadratic Program **225**

List of Figures

2.1 The supervised learning setting. 31
2.2 The intuition behind Multiple Kernel Learning and the differences with Classifier Combination methods. 62

3.1 Plates diagram of the model depicting the conditional relationships of model variables together with the dimensionality of corresponding plates. The dotted plates depict variations for the three parametric combination rules. 74
3.2 Plates diagram of the reduced model depicting the conditional relationships of model variables together with the dimensionality of corresponding plates. The dotted plates depict variations for the three parametric combination rules. 81
3.3 The artificial dataset. 84
3.4 Typical Autocorrelation from the Gibbs sampler. 85
3.5 Typical Autocorrelation from the Metropolis sampler. 86
3.6 Three combined sources with varying informational content. Notice how the the original informative kernel receives 80% of the weight, with the partially informative kernel receiving the rest 20% and the non-informative kernel being effectively discarded. . 87
3.7 The effect of conditioning on the Neal dataset. As the parameter space expands, the required steps of the Gibbs sampler for convergence increase. 88
3.8 Inferring $\boldsymbol{\theta}_i$ and hence learning the importance of the features. The uninformative features, as it can be seen, receive a very low weight and are effectively discarded. 88

4.1 Plates diagram of the model depicting the conditional relationships of model variables together with the dimensionality of corresponding plates. The dotted plates depict variations for the three parametric combination rules. 94
4.2 Linearly separable dataset with known regression coefficients defining the decision boundaries. C_n denotes the members of class n and Dec_{ij} is the decision boundary between classes i and j. 104
4.3 Gibbs posterior distribution of a decision boundary's (Dec_{12}) slope and intercept for a Markov chain of 100,000 samples. The cross describes the original decision boundary employed to sample the dataset. 105
4.4 The variational approximate posterior distribution for the same case as above. Employing 100,000 samples from the approximate posterior of the regression coefficients $\mathbf{W}$ in order to estimate the approximate posterior of the slope and intercept. 105
4.5 Decision boundaries from the Gibbs sampling solution on Neal's dataset. 106
4.6 Decision boundaries from the variational approximation on Neal's dataset. 107

5.1 Plates diagram of the model. 117
5.2 Neal dataset. Left: Uninformative sample selection. Right: informative sample selection 123
5.3 **Top**: Random Initialisation of $\mathbf{Y}$ and 50 cases that initialise contrary to the labels and probit link relation. **Bottom**: Aligned Initialisation of $\mathbf{Y}$ and 50 randomly selected cases (all follow the target labels from the start). 124
5.4 Typical Relevance vectors 128
5.5 Balance dataset. Top: mRVM_1 Bottom: mRVM_2 131
5.6 Glass dataset. Top: mRVM_1 Bottom: mRVM_2 132
5.7 Iris dataset. Top: mRVM_1 Bottom: mRVM_2 133
5.8 Soybean dataset. Top: mRVM_1 Bottom: mRVM_2 134
5.9 Vehicle dataset. Top: mRVM_1 Bottom: mRVM_2 135
5.10 Wine dataset. Top: mRVM_1 Bottom: mRVM_2 136

6.1 Multiple Sources of Automated Currency Validation. The deposited currency note produces crude sensory information from which features are extracted and later combined via the proposed pMKL methodology towards a final classification decision. Images not necessarily representative of sensor measurements. 149
6.2 Typical extraction masks for some Image channels. 151
6.3 Typical Markov chain from the GLMs. Top: Acceptance ratio tuned to 30%. Bottom: Samples from the regression coefficients posterior distribution. 153
6.4 Some posterior distributions (smoothened via a Parzen window type filter) from the Markov chain. Top row: Posteriors significantly deviating from the zero-mean prior. Bottom: Posteriors not deviating from the zero-mean prior. 153
6.5 Typical Z-scores for the binary classification between genuine and counterfeit notes. 154
6.6 Typical error progression with the logistic regression models. . . . 155
6.7 Typical error progression with the probit regression models. . . . 155
6.8 Typical Z-scores for the multiclass classification between genuine new, genuine old and counterfeit notes. 156
6.9 Typical error progression on the multiclass ACV case. 157
6.10 Learning curves on $50 currency notes. 159
6.11 Predictive likelihood progressions for varying training size on $50. 159
6.12 CPU time requirements for varying training size on $50. 160
6.13 Kernel combination parameters indicating the discriminative strength of each channel. 160
6.14 Learning curves on ¥100 currency notes. 161
6.15 Predictive likelihood progressions for varying training size on ¥100. 161
6.16 CPU time requirements for varying training size on ¥100. 162
6.17 Learning curves on £20 currency notes. 162
6.18 Predictive likelihood progressions for varying training size on £20. 163
6.19 CPU time requirements for varying training size on £20. 163
6.20 Predictive strength of fused channels on the US $50 front orientation. 164
6.21 Predictive strength of fused channels on the US $50 back orientation. 165
6.22 Predictive strength of fused channels on the ¥100. 166

6.23 Predictive strength of fused channels on the Scottish £10 currency. 167
6.24 EM Estimator: Error progression while varying the training size. Fixed test size of 500 notes. 169
6.25 mRVM1: Error progression while varying the training size. Fixed test size of 500 notes. 170
6.26 mRVM2: Error progression while varying the training size. Fixed test size of 500 notes. 170
6.27 mRVM1: Sparsity progression while varying the training size. . . 171
6.28 mRVM2: Sparsity progression while varying the training size. . . 171

7.1 Performance of the classifier combinations (Prod C, Sum C, Max C, Maj C). 176
7.2 Performance of the individual classifiers (FR, KL, Pix, ZM) against the best classifier combination (Prod C). 177
7.3 Performance of kernel combination methods (Fix K, Bin K, Weighted K, Prod K) and the single kernel (Single K). 178
7.4 Performance of the best kernel combination methods (Fix K, Weighted K) and the best performing classifier combination method (Prod C). 178
7.5 The mean and std of the multiple kernel weights from the convex linear method (Weighted K). 179
7.6 Tim-barrel 7-bladed beta-propeller Image Source: Wikipedia under a GNU Free Documentation License. 182
7.7 Combinatorial weights when all the feature spaces are employed. . 188
7.8 Confusion matrix with each element normalised to R_{ij} 189
7.9 ROC score (AUC) distributions for the proposed string combination method and two state-of-the-art string kernels with SVMs. Every point in the graph describes the number of families (y-axis) that achieve a specific ROC score (x-axis) by a single method. . . 192
7.10 Kernel combination weights when all the string kernels are fused. 192
7.11 Average kernel usage: PSORT+ 194
7.12 Average kernel usage: PSORT- 195

8.1 Varying the corruption level on a source while measuring the Frobenius inner product. Results are averaged over 10 randomly bootstrapped runs for every noise level. 204

List of Tables

3.1 Comparison of Gibbs versus Metropolis sampling through sampling Distance (mean ± std) and Effective Sampling Size (mean ± std). 85

4.1 CPU time (sec) comparison for 100,000 Gibbs samples versus a maximum of 100 variational iterations. Notice that the number of variational iterations needed for the lower bound to converge is typically less than 100. 107
4.2 Multinomial UCI datasets. N, C, D are respectively the number of samples, classes and attributes in each dataset. 108
4.3 10-fold cross-validated error percentages (mean±std) on standard UCI multinomial datasets. Top performance (not always statistically significant) in **bold**. 109
4.4 Running times (seconds) for computing 10-fold cross-validation results with unoptimised Matlab® codes. 109

5.1 Multinomial UCI datasets. N, C, D are respectively the number of samples, classes and attributes in each dataset. The best-performing kernel function for each problem is reported. 127
5.2 10 times 10-fold cross-validated recognition rates (mean±std) on standard UCI multinomial datasets with the EM scheme. Top performance from EM or MAP (not always statistically significant) in **bold**. 128
5.3 10 times 10-fold cross-validated recognition rates (mean±std) on standard UCI multinomial datasets with the mRVM schemes. Top performance (not always statistically significant) in **bold**. . . 129

6.1 Characteristics of the available ACV sensory information. Further details regarding sensor measurements are confidential to NCR Labs. 148
6.2 Generalised Linear Models employed for Covariate Ranking. . . . 152
6.3 The training ranges examined for the specific fixed test size. . . . 158
6.4 The training/test sample sizes examined. 164
6.5 *Fixed Integration in US50BA* : Combination of 2nd order polynomial kernels. Comparison between Integration with Image only channels versus total Integration with additional Non-Image channels on the US $50 (BA) currency. 164
6.6 *Fixed Integration in US50BC* : Combination of 2nd order polynomial kernels. Comparison between Integration with Image only channels versus total Integration with additional Non-Image channels on the US $50 (BC) currency. 165
6.7 *Weighted Integration in Chinese* : Combination of 2nd order polynomial kernels. Comparison between Integration with Image only channels versus total Integration with additional Non-Image channels on the Chinese ¥100 currency. 166
6.8 *Weighted Integration in SCT* : Combination of 2nd order polynomial kernels. Comparison between Integration with Image only channels versus total Integration with additional Non-Image channels on the Scottish £10 currency. 166
6.9 Comparison across Methods for US $50 (BA) currency. 168
6.10 Comparison across Methods for US $50 (BC) currency. 168
6.11 Comparison across Methods for Chinese ¥100 currency. 168
6.12 Comparison across Methods for Scottish £10 currency. 168

7.1 A roadmap for this Chapter regarding experiments, methods, problem main characteristics and experimental goals. Abbreviations: **HNR**-Handwritten Numeral Recognition, **PFR**-Protein Fold Recognition, **RHD**-Remote Homology Detection, **PSL**-Protein Sub-cellular Localisation, **MKL**(Sources S)-Multiple Kernel Learning, **MC**(Classes C)-Multiclass problem, **CC**-Classifier Combination methods, **Het.MKL**-Heterogeneous MKL. 174
7.2 Abbreviated names of ensemble methods. 175
7.3 Results on HNR from individual classifiers. 180

7.4 Results on HNR when combining classifiers. 181
7.5 Results on HNR with the pMKL methods. 181
7.6 Results on HNR with the VBpMKL methods. 181
7.7 Fold types (27 classes) in the dataset 185
7.8 The 12 Feature spaces. Sequence-alignment based features were computed with different gap penalties: SW_1 with scoring settings from Liao and Noble (2003) and SW_2 with penalties of 0.8. 185
7.9 Average Individual Feature Space Percentage Accuracy 186
7.10 Effect of F.S combination. % Accuracy reported. 187
7.11 CPU times (sec) for the VBKC 188
7.12 Best single run performances (% Accuracy) 188
7.13 ROC, ROC50 and median RFP scores. 191
7.14 Error and sparsity on PSORT+ 193
7.15 Error and sparsity on PSORT- 194

Chapter 1

Introduction

We are drowning in information and starving for knowledge.
–Rutherford D. Roger (former Yale librarian)

1.1 Learning from Multiple Sources

The longstanding need to extract and create knowledge from multiple uncertain observations of a common underlying phenomenon becomes non-trivial in the presence of multiple observers. This additional plurality motivates the urgent requirement for effective inference procedures in the presence of multiple and possibly heterogeneous information sources. The purpose of this thesis is to investigate and propose probabilistic approaches towards that end, within the context of Bayesian inference that permits plausible reasoning whilst handling uncertainty in a principled manner.

The particular (machine) learning scenario under investigation is classification, where the individual uncertain observations belong to a specific class within the unobserved phenomenon. Learning takes place on the basis of a supervisory process which provides initial examples of observations associated with a known class. An intuitive, but inexact, analogy is the learning process that takes place when parents teach their children to separate things by example. After learning has taken place, a prediction for the class of a novel observation can be obtained.

Under the classification setting, an observation or object may be represented by a set of characteristics that depend on its realisation within a specific information channel and its class. In the presence of multiple such channels the evidence is now multi-modal, in the sense of multiple modalities, as there are multiple sets

of characteristics with unknown discriminatory quality and information content. For an overall classification to take place that multi-modal evidence needs to be integrated in a formal, appropriate way such that both the discriminatory quality and the uncertainty associated with each channel is taken into account.

Until now, most approaches to learning under this scenario proposed to fragment the sources and learn individual models with individual predictions later combined in an ad-hoc post-processing manner. This leads to an exaggeration of the problem, multiple model fitting procedures, and the inability to formally infer the discriminatory strength of an information channel as the resulting individual predictions are now model dependent. Furthermore, the integration now occurs at the model level and not on the information source level loosing significant generality and model independent knowledge regarding the information channels.

The present work explores information integration close to the primal source level and in the multiclass setting where an observation may belong to one of a multitude of classes. Uncertainty is addressed through the adopted probabilistic Bayesian framework and formally expressed in parameter distributions and class predictions. Finally, this thesis proposes an overall probabilistic classification machine able to efficiently tackle multiple information sources.

The motivating application of this thesis is Automatic Currency Validation which describes the recognition and detection of counterfeit currency notes deposited in an Automated Teller Machine (ATM). The plurality of sensor modalities, the need for assessing the costs associated with a classification decision and the multiple currency note categories and conditions, motivate the requirement for probabilistic multiclass multiple kernel learning.

1.2 Contributions

The original contribution of this thesis is the proposal and investigation of probabilistic Bayesian approaches for multiclass classification with multiple sources of information. This is reflected in the following specific contributions:

Patents

- He, C., Damoulas, T. and Girolami, M. A.: 2009, Self-service terminals. USA Patent application, Serial number 11/899,381,

http://www.faqs.org/patents/app/20090057395.

Refereed Journal Articles

- Damoulas, T. and Girolami, M. A.: 2008, Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection, Bioinformatics 24(10), 1264 – 1270.

- Damoulas, T. and Girolami, M. A.: 2009a, Combining feature spaces for classification, Pattern Recognition 42(11), 2671 – 2683.

- Damoulas, T. and Girolami, M. A.: 2009c, Pattern recognition with a Bayesian kernel combination machine, Pattern Recognition Letters 30(1), 46 – 54.

- Psorakis, Y., Damoulas, T. and Girolami, M. A.: 2010, Multiclass relevance vector machines: An evaluation of sparsity and accuracy, IEEE Transactions on Neural Networks **Under Review**.

Book Chapters

- Damoulas, T. and Girolami, M. A.: 2009b, Combining information with a Bayesian multi-class multi-kernel pattern recognition machine, in R. K. De, D. P. Mandal and A. Ghosh (eds), Machine Interpretation of Patterns: Image Analysis, Data Mining and Bioinformatics, World Scientific Press. In Print.

Refereed Conference Articles

- Damoulas, T., Ying, Y., Girolami, M. A. and Campbel, C.: 2008, Inferring sparse kernel combinations and relevant vectors: An application to sub-cellular localisation of proteins, IEEE, International Conference on Machine Learning and Applications (ICMLA 08), pp. 577 – 582.

- Ying, Y., Campbell, C., Damoulas, T. and Girolami, M. A.: 2009, Class prediction from disparate biological data sources using a simple multi-class multi-kernel algorithm, Pattern Recognition in Bioinformatics (PRIB 09), pp. 427 – 438

Confidential Reports

- Damoulas, T.: 2006, Discriminative significance identification via Markov chain Monte Carlo on generalised linear regression models, Confidential Internal Report Rev. B. No. 002, NCR Labs.
- Damoulas, T.: 2008a, Feature selection of diverse signals for hierarchical Bayesian kernel machine, Confidential Internal Report Rev. B. No. 008, NCR Labs.
- Damoulas, T.: 2008b, Learning curve investigation for multinomial probit classifier, Confidential Internal Report Rev. B. No. 007, NCR Labs.
- Damoulas, T.: 2009, Inferring sparse kernel combinations and relevance vectors, Confidential Internal Report Rev. B. No. 010, NCR Labs.

Websites

- pMKL Website (University of Glasgow & NCR Labs):

  http://www.dcs.gla.ac.uk/inference/pMKL

1.3 Thought Process

The present thesis is underlined by a thought process which has been motivated by the aforementioned problem of learning from multiple sources and the inadequacies of past approaches. The starting point of this process follows the argument that in the presence of multiple information channels it is best to informatively fuse the sources instead of learning multiple (classification) models. This is justified on the basis of economy of computation, possible memory and processing restrictions, and theoretical basis. However, simply concatenating the sources, or some dimensionally reduced representation of them, is problematic as it does not allow us to learn their quality and fuse them accordingly. Furthermore, such concatenation is inefficient when the dimensionality of the sensory information is high and when the sources are heterogeneous.

From the above argument, the thesis progresses by transforming or embedding the information from individual channels to a common metric, individually constructed from each source, hence allowing for direct and informative fusion.

The underlying thought process and motivation then leads us to consider how this can be pursued within a probabilistic and multiclass framework. This gives rise to the proposed methodologies and the focus is placed on reducing the computational complexity, developing efficient algorithmic approaches and investigating the characteristics of multiple kernel learning.

1.4 Thesis Structure

Chapter 2 provides the necessary background and literature review for the thesis, emphasising the methodological motivation behind this work. Chapter 3 presents the first main contribution of this thesis by setting the framework for probabilistic multiple kernel learning and proposing kernel combination rules and Markov chain Monte Carlo inference procedures. Chapter 4 offers approximate inference methodology based on the variational free energy minimisation principles and explores its efficiency with respect to full Bayesian inference. Chapter 5 proposes further deterministic approximations based on point-estimators and generalises "The Relevance Vector Machine" to the multiclass and multiple kernel learning setting.

The motivating application for this thesis is presented in Chapter 6 with accompanied literature review and extensive experimental results on detecting counterfeit currency notes of various currencies and denominations. Further experimental results on large-scale bioinformatics and pattern recognition problems are reported in Chapter 7 with comparisons against classifier combination approaches and other competing heuristics. A theoretical insight on multiple kernel learning is offered in Chapter 8 with the decomposition of the ensemble loss and the observed flat maximum effect. Furthermore, a Fisher information maximisation approach for the linear regression case is proposed. Finally, Chapter 9 concludes and discusses future research directions that emerge from this thesis.

Chapter 2

Introduction to Multiple Kernel Learning

One of the main goals in machine learning, statistics and their intersection is to learn a relationship between input samples generated from a common underlying phenomenon and their responses which can be continuous or categorical variables. Consider for example as input samples the height of pine trees and as the response their corresponding age (regression) or their specific type (classification).

When the available information includes only input samples with their attributes and there is no dependent response variable, the problem reduces to learning an intrinsic pattern or grouping of the samples and it is defined as *unsupervised learning*. On the contrary, when a response variable (continuous or discrete) is associated with every input sample then the problem is in the domain of *supervised learning* and the goal is to predict the response variable for a novel input sample. Finally in between scenarios, where only part of the input samples are associated with a known response variable, belong to the category of *semi-supervised learning* where the goal is again to predict the response variable for a novel sample while this time utilising partially labelled information.

In the supervised learning scenario the typical experimental design is that a collection of *past* observations exists and it is used for model fitting and model selection. This initial observed collection is known as the *training set* and it contains all the information to be extracted by the learning algorithm. Any new or held-out set of observations that might be used for future prediction of their unknown response variables is known as the *test set*. It is worth noting that

a significant assumption of stationarity has taken place; both the training and any test samples are typically assumed independently and identically distributed (i.i.d).

This thesis addresses the problem of integrating multiple sources of information towards an overall classification decision and as such it is firmly within the supervised learning area of research. The specific research question addressed is how to *efficiently* classify input samples that have a multitude of attribute (feature) sets, produced from possibly heterogeneous sources or sensors. As an example consider classifying a deposited currency note in a bank as genuine or counterfeit based on information from light sensors, acoustic sensors and transaction history. Furthermore, this thesis investigates how the above question can be addressed within a *probabilistic* framework which comes with additional learning and decision-making benefits that are introduced in this Chapter together with the specific supervised learning problem.

Due to the connection with continuous response problems and their accommodating nature for inference, the introduction starts from a simple linear regression case and progresses through to classification, kernel methods, inference and the necessary background knowledge that the reader might require. It is not an exhaustive introduction to the supervised learning field as the material reviewed is the work that this thesis builds upon and for a more general introduction the reader is referred to Bishop (2006), MacKay (2003) or Denison et al. (2002) for machine learning, information theoretic or statistics perspectives respectively. It offers however a thorough introduction and review of the specific *multiple kernel learning* problem and associated research work to date.

2.1 Linear Regression and Nonlinear Responses

Consider N input predictor variables $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top$ with $\mathbf{x}_i \in \mathbb{R}^D$ where D is the number of attributes or features. The relationship between the predictors and the continuous response variables $\mathbf{y} = (y_1, \ldots, y_N)^\top$ is the point of interest in regression. Following the most common structural assumptions, this relationship is typically[1] assumed to be described by a deterministic function g and additional random error component $\boldsymbol{\epsilon}$ as:

[1]Not taking into account unobserved predictors known as *random effects* and assuming existing predictor variables are observed without error (Denison et al. 2002).

$$\mathbf{y} = g(\mathbf{X}) + \boldsymbol{\epsilon} \tag{2.1}$$

The true deterministic function g is unobserved and hence it is approximated by an estimating function $f(\mathbf{X})$ whose nature determines the type (linear or nonlinear) of regression employed. This problem-specific choice constitutes the main modelling assumption at this stage and it is crucial for the successful prediction of responses.

In *linear regression* the modelling assumption is that the functional relationship between the input predictor variables and the response variables is *linear in the parameters*. For a predictor $\mathbf{x}$ this implies

$$f(\mathbf{x}) = \mathbf{w}^{\mathsf{T}} h(\mathbf{x}) \tag{2.2}$$

where $\mathbf{w}$ are the parameters (regression coefficients) and h can be a linear or nonlinear function of the inputs. In the simplest case where h is just the input the relationship reduces to a hyper-plane:

$$f(\mathbf{x}) = \mathbf{w}^{\mathsf{T}} \mathbf{x} + b \tag{2.3}$$

with $\mathbf{w} \in \mathbb{R}^D$ and b the bias or intercept that makes the model translation invariant. From hereafter we will assume the bias term is included in the inner product $\mathbf{w}^{\mathsf{T}} h(\mathbf{x})$ by a simple augmentation of the attributes with a vector of ones.

Another common setting, analysed in depth in Denison et al. (2002) and Hastie et al. (2001), is to assume h as a set of k basis functions $B = (B_1, \ldots, B_k)$ e.g. splines as:

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i B_i(\mathbf{x}) \tag{2.4}$$

where now $\mathbf{w} \in \mathbb{R}^k$. Such modelling assumptions induce *nonlinear responses* via what is still a linear regression model, which for N input predictors and responses can be expressed as:

$$\mathbf{y} = \mathbf{B}\mathbf{w} + \boldsymbol{\epsilon} \tag{2.5}$$

where $\mathbf{y} = (y_1, \ldots, y_N)^{\mathsf{T}}$, $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_N)^{\mathsf{T}}$ and

$$\mathbf{B} = \begin{bmatrix} B_1(\mathbf{x}_1) & \cdots & B_k(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ B_1(\mathbf{x}_N) & \cdots & B_k(\mathbf{x}_N) \end{bmatrix}$$

Having introduced the setting for linear regression models we turn our attention to the main *learning* or *inference* methods and justify the Bayesian framework that is adopted in this thesis.

2.2 Learning and Bayesian Inference

The typical supervised learning experimental setting (Bishop 2006) consists of having a *training set* $\{\mathbf{x}_i, y_i\}_{i=1}^N$ of N predictor and response variables that are used to learn the parameters of the assumed model (model fitting). Assuming that the input predictors are i.i.d generated from a phenomenon and the dependent responses from a supervisory process, the goal of learning, as depicted in Figure 2.1 is to approximate the supervisory process and predict response y_* of novel input sample $\mathbf{x}_*$.

The learning process is driven by a loss function[2] $\mathcal{L}(y, \hat{y})$ between the estimated response $\hat{y}$ and the true response y which measures the deviation of the prediction with respect to the true target (evidence). At this point, the specific loss function and learning procedure deviates to two main schools of thought inspired from different branches of statistics and mathematics.

2.2.1 Statistical Learning Theory

In the *Statistical Learning Theory* (SLT) paradigm (Vapnik 1998, Hastie et al. 2001), the emphasis is on (typically convex) optimisation with respect to specific loss functions and penalising (regularisation) terms. As an example, the simple linear regression case can be approached with the Mean Squared Error (MSE) Loss:

$$\frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|^2 \tag{2.6}$$

[2]The term loss function is used generically here and it includes likelihood distributions.

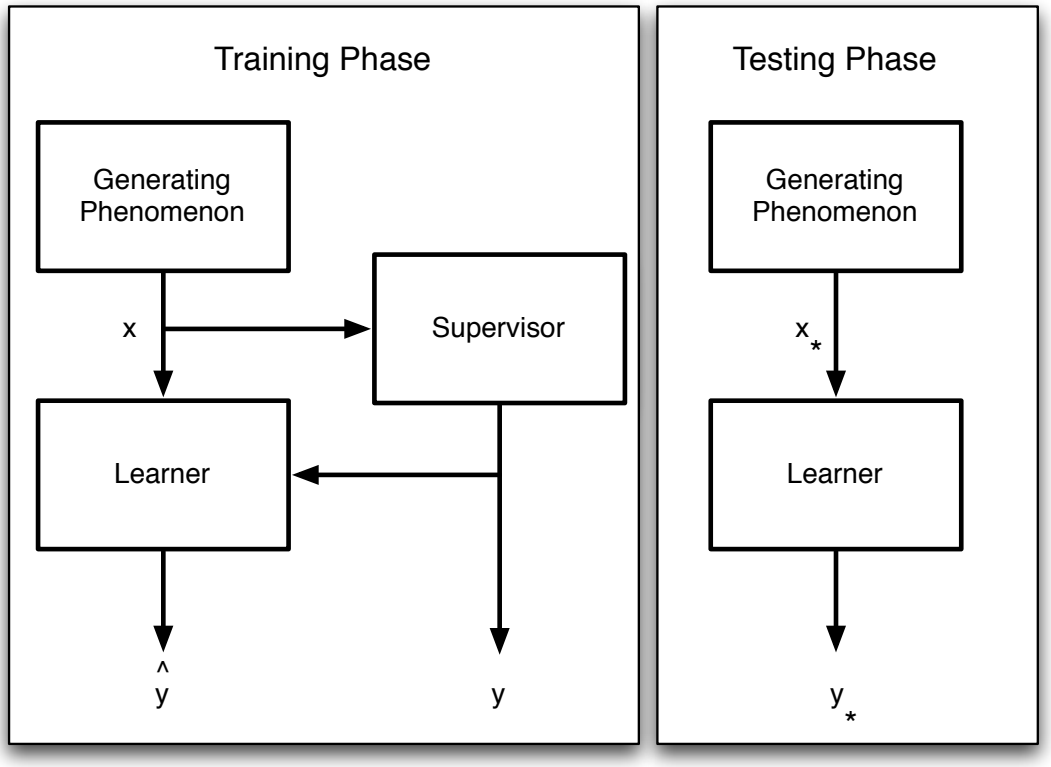

Figure 2.1: The supervised learning setting.

whose minimisation with respect to the parameters $\mathbf{w}$ and the estimating function $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ leads, see Hastie et al. (2001) for first and second order derivatives, to the well known global minimum solution:

$$\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \tag{2.7}$$

The MSE loss results to the ordinary least-squares method by[3] Gauss (1809) and the optimisation implicitly leads to minimising the noise error $\sum_{i=1}^{N} \epsilon_i^2$ for which we have made no assumptions so far.

To control over-fitting the estimating function to the target response (fitting the noise), regularisation via a penalising term is employed within the SLT framework[4]. In the linear regression case a typical regularisation is the squared-weight penalty $\frac{\lambda}{2} \sum_{d=1}^{D} w_d^2$ which leads to the penalised least squares (PLS) solution:

[3]Also claimed by Adrien-Marie Legendre in 1805

[4]In later sections we also draw the connection between regularisation and sparsity of the resulting solution.

$$\hat{\mathbf{w}}_{\text{PLS}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y} \tag{2.8}$$

where the parameter λ controls the trade-off between the smoothness of the function and the fit to the data.

Finally, having briefly described the training or learning phase for linear regression with a linear function within the SLT framework, prediction can be made based on the inferred parameters (from ordinary or penalised least squares), the novel predictor and our estimating linear function as:

$$y_* = \hat{\mathbf{w}}^{\mathsf{T}}\mathbf{x}_* \tag{2.9}$$

2.2.2 Towards Bayesian Inference

In this section we revisit the linear regression setting and introduce the basic concepts behind the *Bayesian* paradigm and the direct connections, e.g. (Tipping 2004), to the least squares solutions of the SLT framework.

So far we have made no modelling assumptions regarding the noise component ϵ which was implicitly minimised in the SLT case and could potentially lead to over-fitting without regularisation. In the Bayesian setting a probabilistic model over the noise component is placed which can be assumed to be normally distributed with σ^2 variance: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

This directly leads to a distribution over the responses, whose negative logarithm resembles a typical loss function, which is the *likelihood* of the linear model m:

$$L = p(\mathbf{y}|\mathbf{X}, m) = \mathcal{N}(f(\mathbf{X}), \sigma^2\mathbf{I}) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{w}^{\mathsf{T}}\mathbf{x}_i, \sigma^2) \tag{2.10}$$

This important distribution expresses how *likely* it is for the model to reproduce or generate the *evidence* $\mathbf{y}$.

Maximisation of the likelihood is equivalent to minimising the negative logarithm:

$$\mathcal{L} = -\log p(\mathbf{y}|\mathbf{X}, m) = \frac{N}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{N}\{y_n - \mathbf{w}^\intercal \mathbf{x}_i\}^2 \tag{2.11}$$

which leads to the *maximum likelihood* (ML) estimate for the parameters $\mathbf{w}$ that is equivalent to the ordinary least squares estimate from Equation 2.7:

$$\hat{\mathbf{w}}_{\text{ML}} = \underset{\mathbf{w}}{\text{argmax}}(\mathcal{L}) = \hat{\mathbf{w}}_{\text{OLS}} \tag{2.12}$$

and analogously for the noise variance:

$$\hat{\boldsymbol{\sigma}}_{\text{ML}} = \underset{\boldsymbol{\sigma}}{\text{argmax}}(\mathcal{L}) \tag{2.13}$$

However now we have resulted again in a point estimate for the parameters[5] and the response despite the initial placement of a distribution over the error component. Furthermore the ML estimate is prone to over-fitting (Ripley 1996), especially when the training size is small, as it is solely based on the data evidence.

2.2.3 Bayesian Inference

In order to retain a truly probabilistic framework we must also place distributions over the random variables before the model sees any evidence in the form of data. Such distributions are called prior distributions and express our *a priori beliefs* about the phenomenon we are trying to infer (as prior beliefs on the model parameters imply prior beliefs for the phenomenon). In order to update these prior beliefs to *a posteriori beliefs*, having seen the evidence, we need Bayes rule which is the foundation of Bayesian inference:

$$\underline{\text{Bayes Rule}}: \qquad P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.14}$$

where

- $P(A)$ - The prior belief for A independent of B.
- $P(B)$ - The prior belief for B independent of A. Also defined as the

[5]The variance of the estimate is available but has no contribution in the final prediction.

marginal likelihood as it is equivalent with integrating out A from the *joint likelihood* which is the numerator.

- $P(B|A)$ - The conditional probability of B given A which corresponds[6] to the *likelihood* of A for known B.
- $P(A|B)$ - The posterior belief for A after observing B.

Returning back to the linear regression framework we place a zero-mean Gaussian prior distribution over the parameters or regression coefficients $\mathbf{w}$:

$$p(\mathbf{w}|\alpha) = \prod_{j=1}^{D} \left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left\{-\frac{\alpha}{2} w_j^2\right\} \tag{2.15}$$

where α is a common scale or inverse variance across dimensions and the prior distribution expresses our prior belief that the evidence are generated from a relatively smooth phenomenon and hence smaller weights are preferred a priori.

Following Bayes rule and recalling the likelihood function in Equation 2.10 we can now update our beliefs for the parameters $\mathbf{w}$ to the posterior distribution (Tipping 2004):

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \alpha, \sigma^2) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2)p(\mathbf{w}|\alpha)}{p(\mathbf{y}|\mathbf{X}, \alpha, \sigma^2)} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.16}$$

where

$$\boldsymbol{\mu} = \left(\mathbf{X}^\intercal \mathbf{X} + \sigma^2 \alpha \mathbf{I}\right)^{-1} \mathbf{X}^\intercal \mathbf{y} \tag{2.17}$$

$$\boldsymbol{\Sigma} = \sigma^2 (\mathbf{X}^\intercal \mathbf{X} + \sigma^2 \alpha \mathbf{I})^{-1} \tag{2.18}$$

Hence now we have a closed form solution for the posterior over the parameters due to the accommodating nature of linear regression where both the likelihood and the prior can be described with Gaussian distributions that give rise to a Gaussian posterior. This unfortunately will not always be the case and we will have to resort to either sampling techniques or deterministic approximations that are described in later sections.

It is worth noting that the prior placed on the regression coefficients has an analogous function to the regularisation component within the SLT framework.

[6]When $P(B|A)$ is treated as a function of B given A it corresponds to a probability (distribution/density) function but when is treated as a function of A given B it is a likelihood function.

It places a bias for smooth estimating functions and hence ensures the model is not over-fitting the data. We can further see the analogy between the approaches by maximising over the posterior and examining the mode of that distribution:

$$\hat{\mathbf{w}}_{\text{MAP}} = \boldsymbol{\mu} = \hat{\mathbf{w}}_{\text{PLS}} \tag{2.19}$$

assuming $\lambda = \sigma^2\alpha$. Thus the *maximum a posteriori* (MAP) solution is equivalent to the PLS estimate and the parameter product $\sigma^2\alpha$ has a similar function to λ of penalising complex functions and avoiding over-fitting.

This analogy is only present when we restrict our probabilistic model to resulting point estimates such as the ML or MAP solutions. In reality we have a posterior distribution over the regression coefficients and we can make full use of it through the Bayesian tool of *marginalisation*:

$$p(y_*|\mathbf{y}, \mathbf{X}, \alpha, \sigma^2) = \int p(y_*|\mathbf{w}, \sigma^2)p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \alpha, \sigma^2)\ d\mathbf{w} \tag{2.20}$$

where we see that our final predictive function is an average over the *whole* of the regression coefficients posterior. In the case where integration cannot be performed in closed form, the *Monte Carlo* estimate can be employed. The above marginalisation provides another Bayesian benefit, that of explicitly taking into account the uncertainty for the parameters in the form of the posterior distribution (if it is concentrated or diffuse).

Finally, it is worth noting that we can place further prior distributions on the scales and the variance, propagating uncertainty into higher levels in the model and becoming "truly" Bayesian by marginalising over all model parameters. In some of these cases however we loose the benefit of having a closed form posterior distribution as the *joint* posterior over all parameters can become intractable. At this point, sampling or deterministic approximations become necessary for Bayesian inference and we will review such strategies later in this Chapter.

The Bayesian framework will be adopted for the remainder of this thesis on the basis of its advantages, most of which we have already seen. In a summary these are:

- **Prior beliefs** - Explicitly incorporate prior knowledge regarding the problem under consideration via the prior distributions placed on the model parameters. Bayesian inference is within the so-called subjective[7] probabil-

[7]There is a great history and interesting controversy in statistics between "Bayesians" and

ity theory field (Good 1983) and accommodates prior knowledge and also prior non-informative "objective" beliefs with appropriate distributions.

- **Probabilistic Responses** - Instead of a single point response, a distribution over responses is offered via the Bayesian framework. Therefore a direct measure of the confidence of the model's responses is offered which is crucial for decision making in critical applications such as health informatics or security.

- **Marginalisation** - Model parameters can be marginalised (integrated) out, effectively averaging over all their possible values. Very useful and informative quantities, as we shall see and employ later on, such as the marginal likelihood are based on marginalisation.

- **Uncertainty** - Posterior distributions directly express the uncertainty over model parameters which is taken directly into account via the process of marginalisation. Uncertainty can be encoded and propagated into higher levels of model hierarchy through the use of priors and hyper-priors (prior distributions over parameters from lower level prior distributions).

- **Formality** - Bayesian inference is firmly based on probability theory and the corresponding axioms of plausible reasoning (Jaynes 2003) providing a systematic and formal way of dealing with uncertainty.

2.3 The Kernel Trick and Kernel Regression

So far in this thesis we have concentrated on linear regression and how to perform *learning* through the two mainstream approaches of SLT and Bayesian inference, highlighting the probabilistic benefits of the latter. We have seen how non-linearity between the input predictors and the responses can be achieved through basis function expansions while retaining the appealing nature and identifiability of linear models. In this section we take a step further into possibly nonlinear embeddings of the original features and introduce the concept of *kernel substitution*, also known as the *kernel trick*, which has revolutionised

"Frequentists" on exactly the subjective nature of prior distributions. The interested reader is directed to (Jaynes 2003) and (Edwards 1992) for the Bayesian and Frequentist perspective respectively.

the field during the last decade (Schölkopf and Smola 2002, Shawe-Taylor and Cristianini 2004).

Consider the basis function expansion of the linear regression model in Equations 2.4 and 2.5 and generalise it to some possible nonlinear feature expansion $\mathbf{\Phi} \in \mathbb{R}^{N \times M}$ with the i^{th} row given by $\boldsymbol{\phi}(\mathbf{x}_i)^\mathsf{T}$:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \boldsymbol{\epsilon} \tag{2.21}$$

The likelihood and prior follow from Equations 2.10 and 2.15 after substituting the expansion $\mathbf{\Phi}$. Disregarding the marginal likelihood which is a constant term we can express the logarithm of the posterior[8] as the sum of the log-likelihood and log-prior:

$$\log p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \alpha, \sigma^2) \propto -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\{y_i - \mathbf{w}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}_i)\}^2 - \frac{\alpha}{2}\mathbf{w}^\mathsf{T}\mathbf{w} \tag{2.22}$$

maximising with respect to $\mathbf{w}$, setting to zero and solving for $\mathbf{w}$ we have:

$$\mathbf{w} = -\frac{1}{\alpha\sigma^2}\sum_{i=1}^{N}\{y_i - \mathbf{w}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}_i)\}\boldsymbol{\phi}(\mathbf{x}_i) = \mathbf{\Phi}^\mathsf{T}\mathbf{a} \tag{2.23}$$

where the vector $\mathbf{a}$ has elements $a_i = -\frac{1}{\alpha\sigma^2}\{y_i - \mathbf{w}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}_i)\}$. We can now reformulate the logarithm of the posterior with respect to the parameter $\mathbf{a}$ and obtain:

$$\log p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \alpha, \sigma^2) = -\frac{1}{2\sigma^2}\mathbf{a}^\mathsf{T}\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T}\mathbf{a} + \frac{1}{\sigma^2}\mathbf{a}\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T}\mathbf{y} - \frac{1}{2\sigma^2}\mathbf{y}^\mathsf{T}\mathbf{y} - \frac{\alpha}{2}\mathbf{a}^\mathsf{T}\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T}\mathbf{a} \tag{2.24}$$

and we can see that the feature expansion $\mathbf{\Phi}$ appears only as an inner product with itself. Hence this *dual* representation indicates that we actually only need inner products of the feature expansion and not the actual feature expansion per se. Defining the $N \times N$ Gram matrix $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\mathsf{T}$ as a symmetric matrix of vector inner products in an inner product space and setting the derivative to zero with respect to $\mathbf{a}$ we obtain the dual solution:

[8] We could directly formulate the closed form posterior as in 2.16 but we will maximise over it to introduce the dual formulation and the kernel trick.

$$\mathbf{a} = (\mathbf{K} + \alpha\sigma^2\mathbf{I})^{-1}\mathbf{y} \tag{2.25}$$

which, recalling Equation 2.9, leads to the prediction for a novel sample $\mathbf{x}_*$ as:

$$y_* = \mathbf{w}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}_*) = \mathbf{a}^\mathsf{T}\boldsymbol{\Phi}\boldsymbol{\phi}(\mathbf{x}_*) = \mathbf{k}(\mathbf{x}_*)^\mathsf{T}(\mathbf{K} + \alpha\sigma^2\mathbf{I})^{-1}\mathbf{y} \tag{2.26}$$

where $k(\mathbf{x}_*)$ denotes a vector of N inner products between the training set expansion $\boldsymbol{\Phi}$ and the test sample expansion $\phi(\mathbf{x}_*)$.

This transformation of the problem leads to two main observations. First, that we do not need to explicitly construct a feature embedding $\boldsymbol{\Phi}$ of the input samples but we only need to define a valid function that directly describes the inner product of some feature expansion. Secondly, the transformed regression parameters of the dual formulation are N dimensional now as they operate on the Gram matrix and they require an $\mathcal{O}(N^3)$ inversion for estimation. This appears initially disadvantageous as we were operating before on a space with dimensions equal to the number of basis functions, which are typically less than the number of samples, but it offers the advantage that implicitly now we can employ a very high (infinite in some cases) dimensional embedding.

Definition 2.1: [Kernel function] (Shawe-Taylor and Cristianini 2004)
A kernel is a function k that for all $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$ satisfies

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$$

where ϕ is a mapping from $\mathbf{X}$ to an (inner product) feature space F

$$\boldsymbol{\phi} : \mathbf{x} \mapsto \boldsymbol{\phi}(\mathbf{x}) \in F.$$

From Definition 2.1 we can see that the Gram matrix $\mathbf{K}$ is the corresponding *kernel* matrix and now we can employ any valid kernel function to implicitly produce high dimensional embeddings. The main kernel property of interest at this stage (see (Shawe-Taylor and Cristianini 2004) for a full treatment) is that the resulting kernels are symmetric *positive semi-definite* matrices.

Definition 2.2: [Positive Semi-definite Matrix]
A symmetric matrix $\mathbf{K}$ is positive semi-definite if its eigenvalues are all non-negative.

Some typical kernel functions $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$ that are employed in this thesis are summarised in Table 2.3:

Kernel Type	Function	Characteristics
Linear (Cosine)	$\left(\mathbf{x}_i^\top \mathbf{x}_j\right)$	Cosine follows by normalisation
Polynomial	$\left(\mathbf{x}_i^\top \mathbf{x}_j + 1\right)^n$	Degree n
Gaussian (RBF)	$\exp\left(-\frac{\|\|\mathbf{x}_i - \mathbf{x}_j\|\|^2}{2\sigma^2}\right)$	Infinite degree polynomial

Finally, revisiting the linear regression setting we can now reformulate it into a *kernel* regression problem:

$$\mathbf{y} = \mathbf{w}^\top \mathbf{K} + \boldsymbol{\epsilon} \tag{2.27}$$

and as before we obtain a closed form Gaussian posterior distribution for the regression coefficients as $p(\mathbf{w}|\mathbf{y}, \mathbf{K}, \alpha, \sigma^2) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with parameters defined with respect to the kernel matrix $\mathbf{K}$ as:

$$\boldsymbol{\mu} = \left(\mathbf{K}^\top \mathbf{K} + \sigma^2 \alpha \mathbf{I}\right)^{-1} \mathbf{K}^\top \mathbf{y} \tag{2.28}$$

$$\boldsymbol{\Sigma} = \sigma^2 (\mathbf{K}^\top \mathbf{K} + \sigma^2 \alpha \mathbf{I})^{-1} \tag{2.29}$$

The *kernel trick* offers a powerful and efficient way of producing high dimensional data embeddings that capture non-linearities of the modelling phenomenon and will be of especial interest to the classification setting that we visit next.

2.4 Classification

In classification the target or response variables[9] $\mathbf{t}$ are discrete real values associating input samples $\mathbf{x}_i$ to a single specific class $c \in \{1, \ldots, C\}$. The encoding for the target varies depending on the classification model employed and the number of classes. For binary classification problems where there are only two classes it is typically represented as $t_n \in \{0, 1\}$ or $t_n \in \{-1, +1\}$ whereas for multinomial problems it is either $t_n \in \{1, \ldots, C\}$ or follows a $1-of-C$ encoding scheme.

[9]Denoted by $\mathbf{t}$ to distinguish from the regression case where responses were defined as $\mathbf{y}$.

The interest lies in the joint distribution $p(\mathbf{t}, \mathbf{X})$ and there are two main categories of classification models depending on its decomposition:

$$p(\mathbf{t}, \mathbf{X}) = p(\mathbf{t}|\mathbf{X})p(\mathbf{X}) = p(\mathbf{X}|\mathbf{t})p(\mathbf{t}) \tag{2.30}$$

Following the first decomposition, we end up directly modelling the quantity of interest $p(\mathbf{t}|\mathbf{X})$ and such models are termed *discriminative*. In the second case we model the class conditional density $p(\mathbf{X}|\mathbf{t})$ and employ Bayes' rule to obtain again the distribution of interest:

$$p(\mathbf{t}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{t})p(\mathbf{t})}{p(\mathbf{X})} \tag{2.31}$$

Such approaches are termed *generative* as we are able to *generate* samples from the model's class conditional distribution. There are qualitative differences and merits for either approach (Duda et al. 2000, Bishop 2006) and in this thesis we concentrate on *discriminative* approaches which avoid the drawbacks of density estimation in high-dimensional spaces and directly model the quantity of interest $p(\mathbf{t}|\mathbf{X})$.

2.4.1 Logistic and Probit Regression

The standard probabilistic discriminative classification approach is to turn the output of a regression model into a class probability by the use of a *sigmoid* function[10]. This constrains the continuous real value output $[-\infty, +\infty]$ to the range [0,1], satisfying the requirements for a probabilistic representation of class membership.

For example, in the linear regression case the model becomes:

$$\mathbf{t} = \sigma\left(\mathbf{w}^{\mathsf{T}} h(\mathbf{x})\right) \tag{2.32}$$

where the sigmoid function σ typically takes one of the following forms:

[10] Also known as an activation function or inverse link function.

Type	Function	Case
Logistic	$\sigma(z) = \frac{\exp(z)}{1+\exp(z)}$	Binary
Softmax	$\sigma(z_c) = \frac{\exp(z_c)}{\sum_{i=1}^{C}\exp(z_i)}$	Multinomial
Probit	$\Phi(z) = \int_{-\infty}^{z} \mathcal{N}_x(0,1)dx$	Binary

These approaches belong to the family of Generalised Linear Models (GLMs) (McCullagh and Nelder 1989) and are specifically known as *logistic* or *probit* regression, according to the likelihood function employed[11].

Considering now the general probabilistic classification framework with GLMs we have the posterior for the parameters $\mathbf{w}$:

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \alpha) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\alpha)}{\int p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\alpha)d\mathbf{w}} \tag{2.33}$$

where the likelihood $p(\mathbf{t}|\mathbf{X}, \mathbf{w})$ is given by the specific choice of link function in Table 2.4.1.

Unfortunately the posterior cannot be obtained in closed form, in contrast with the accommodating nature of linear regression models, and hence *exact* inference is not possible. This is a typical obstacle in Bayesian inference for which *approximate* methods have been proposed and developed. In the next sections we review exactly such approximate inference techniques that will allow us to complete inference within the classification setting.

2.5 Markov Chain Monte Carlo

The first approximate inference scheme reviewed is the sampling approaches of Markov chain Monte Carlo (MCMC) which becomes exact in the limit of infinite samples. For an excellent practical introduction the reader is referred to Gelman et al. (2004). The intuition behind MCMC is to address our inability of obtaining closed form posterior distributions by instead drawing samples from them.

In most cases we are actually interested in calculating expectations with respect to the posterior distribution, such as the class predictions in classification.

[11]The softmax is the generalisation of the logistic link function to the multiclass setting.

Hence, assuming we can draw samples from the (joint) posterior of parameters[12] $\boldsymbol{\theta}$, these expectations can be approximated via the Monte Carlo estimate:

$$\mathbb{E}\{f|\mathbf{t},\mathbf{X}\} = \int f(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{t},\mathbf{X})d\boldsymbol{\theta} \approx \widetilde{f} = \frac{1}{L}\sum_{l=1}^{L} f(\boldsymbol{\theta}^l) \tag{2.34}$$

One important observation is that the variance of the Monte Carlo estimate is given by:

$$\text{var}\{\widetilde{f}\} = \frac{1}{L}\mathbb{E}\left\{(f - \mathbb{E}\{f\})^2\right\} \tag{2.35}$$

and hence the accuracy of the estimate is independent of the model's dimensionality.

The major hurdle of sampling from the posterior distribution has not been addressed so far and in the next section the four sampling approaches that will be employed in this thesis are reviewed.

2.5.1 Importance Sampling

One of the most classical, and straightforward to implement, Monte Carlo estimators for a function $f(\boldsymbol{\theta})$ with $\boldsymbol{\theta}$ distributed as $p(\boldsymbol{\theta})$ is given by the importance sampling approach that utilises an easy-to-sample from distribution $q(\boldsymbol{\theta})$ in the following way:

$$\mathbb{E}_{p(\boldsymbol{\theta})}\left\{f(\boldsymbol{\theta})\right\} = \int f(\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} = \int f(\boldsymbol{\theta})\frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}q(\boldsymbol{\theta})d\boldsymbol{\theta} = \mathbb{E}_{q(\boldsymbol{\theta})}\left\{w(\boldsymbol{\theta})f(\boldsymbol{\theta})\right\} \tag{2.36}$$

where $w(\boldsymbol{\theta}) = \dfrac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}$ is the *importance weight*.

The intuition behind this approach is to sample from the *importance* distribution $q(\boldsymbol{\theta})$ which can be conveniently chosen as long as it offers good support for the distribution of interest, and then to weight each sample with the associated importance weight resulting in the following estimator:

[12] In order to express any model parameter and not only regression coefficients $\mathbf{w}$, we denote parameters with the generic notation $\boldsymbol{\theta}$. We assume the classification setting as an example scenario with the parameter posterior of interest $p(\boldsymbol{\theta}|\mathbf{t},\mathbf{X})$.

$$\widetilde{f} = \frac{1}{L}\sum_{l=1}^{L} w(\boldsymbol{\theta}^l) f(\boldsymbol{\theta}^l) \tag{2.37}$$

In the (common) case (Andrieu 2003) where the distribution of interest $p(\boldsymbol{\theta})$ is only available in its unnormalized form $p_*(\boldsymbol{\theta})$ then the importance weights are modified to:

$$w(\boldsymbol{\theta}^l) = \frac{\frac{p_*\left(\boldsymbol{\theta}^l\right)}{q\left(\boldsymbol{\theta}^l\right)}}{\sum_{j=1}^{N} \frac{p_*\left(\boldsymbol{\theta}^j\right)}{q\left(\boldsymbol{\theta}^j\right)}} \tag{2.38}$$

Finally, the main dangers of importance sampling reside in the choice of the importance distribution. Ideally it should offer support so that the target distribution is efficiently explored and it should satisfy the condition that $p(\boldsymbol{\theta}) > 0 \Rightarrow q(\boldsymbol{\theta}) > 0$. The advantages of importance sampling are that it's easy to implement, it is parallelisable, and that it can be extended to sequential inference. Due to the latter property it is the cornerstone of sequential Monte Carlo (particle filters) techniques (Doucet et al. 2000) and the main approach in dealing with *covariate shift* (Quiñonero-Candela et al. 2009) where the i.i.d assumption is no longer valid.

In the next sections we introduce the Markov chain Monte Carlo techniques that construct an *ergodic* Markov chain that converges to a *stationary* distribution that is the target distribution. The following definitions introduce the basic concepts:

Definition 2.3: [First order Markov chain]
A first order Markov chain is a sequence of random variables $\theta^1, \theta^2, ..., \theta^n$ that for any $t \in \{1, \ldots, n\}$ the distribution of θ^t given all previous values of θ is dependent only on the previous value θ^{t-1}:

$$p(\theta^t|\theta^{t-1}, \theta^{t-2}, \ldots, \theta^1) = p(\theta^t|\theta^{t-1})$$

Definition 2.4: [Ergodicity]
A Markov chain converges to a unique stationary distribution if it is irreducible, aperiodic and not transient. Such a Markov chain is called ergodic.

Where aperiodicity and non-transiency hold for a random walk on any proper distribution (Gelman et al. 2004) and irreducibility dictates that there is a positive probability of reaching any state from any other state, in other words the Markov chain can reach all states from any state within a finite sequence of steps.

2.5.2 Metropolis Sampling

The first MCMC method considered is the Metropolis algorithm (Metropolis et al. 1953) which employs a *symmetric proposal* distribution (also known as transition or jump distribution) $q(\theta^t|\theta^{t-1}) = q(\theta^{t-1}|\theta^t)$ and accepts or rejects generated samples from that distribution based on the following criterion known as the *acceptance ratio*:

$$\mathcal{R} = \min\left\{1, \frac{p_*(\theta^t|\mathbf{t}, \mathbf{X})}{p_*(\theta^{t-1}|\mathbf{t}, \mathbf{X})}\right\} \tag{2.39}$$

where $p_*(\theta|\mathbf{t}, \mathbf{X})$ denotes the unnormalized parameter posterior which is equal to the product of the likelihood with the prior over the parameters.

The procedure is to draw a random number from a uniform distribution on the unit interval and if the number is smaller or equal to $mathcalR$ the proposed sample is accepted, else rejected. Hence the new state is given by:

$$\theta^t = \begin{cases} \theta^t & \text{with probability } \mathcal{R} \\ \theta^{t-1} & \text{otherwise} \end{cases} \tag{2.40}$$

2.5.3 Metropolis-Hastings Sampling

The straightforward generalisation of the Metropolis scheme to handle *asymmetric proposal* distributions is known as the Metropolis-Hastings (MH) method. The only significant difference is that the distribution $q(\theta)$ is asymmetric: $q(\theta^t|\theta^{t-1}) \neq q(\theta^{t-1}|\theta^t)$ and hence it is included in the acceptance ratio as:

$$\mathcal{R} = \min\left\{1, \frac{p_*(\theta^t|\mathbf{t}, \mathbf{X})q(\theta^{t-1}|\theta^t)}{p_*(\theta^{t-1}|\mathbf{t}, \mathbf{X})q(\theta^t|\theta^{t-1})}\right\} \tag{2.41}$$

Both the Metropolis and the Metropolis-Hastings MCMC methods have been proved (Hastings 1970) to converge to a stationary distribution that is the target distribution ($p(\theta|\mathbf{t}, \mathbf{X})$ here).

The main drawback of Metropolis based MCMC methods is the need to *tune* the proposal distribution in order to retain an acceptance ratio between $15 - 40\%$ (depending on the nature of the parameter sampling scheme) which is the recommended level to efficiently reach convergence (Gelman et al. 2004). Adaptive proposal distributions are usually employed that take into account the covariance structure of the posterior through initial exploratory samples that are later discarded (Burn-in period). In general, the engineering requirements of such methods make them less practical although more efficient sampling schemes based on gradient information through the Fisher information matrix are a major research topic (Girolami et al. 2009) and a promising direction.

2.5.4 Gibbs Sampling

The final MCMC approach reviewed is the Gibbs sampler (Geman and Geman 1984, Tanner and Wong 1987) also known as *alternating conditional sampling* (Andrieu 2003, Gelman et al. 2004) which can be seen as a special case of the Metropolis-Hastings method. The intuition behind it is to decompose the (unobtainable in closed form) posterior distribution of interest into conditional posterior distributions that are easy to sample from. Consider a parameter vector $\boldsymbol{\theta}$ and the sought after posterior distribution $p(\boldsymbol{\theta}|\mathbf{t}, \mathbf{X})$. If we decompose the joint posterior to conditional posterior distributions of the form:

$$p(\theta_i^t|\boldsymbol{\theta}_{-i}^{t-1}, \mathbf{t}, \mathbf{X}) \tag{2.42}$$

then iteratively sampling from these conditional distributions leads to sampling from the target joint posterior distribution. The notation $\boldsymbol{\theta}_{-i}^{t-1}$ denotes all the elements of $\boldsymbol{\theta}$ except the i^{th} one, from the current $(t-1)$ sample.

This principle can be extended to sampling *block* variables where the joint posterior can be decomposed to blocks of parameter sets (i.e. regression coefficients $\mathbf{w}$ and scales α in GLMs) whose conditional posterior distributions are easy to sample from. Such a *block-wise Gibbs sampling* approach will be employed in this thesis.

The advantage of Gibbs sampling is that no proposal distribution is necessary. It can be seen as a sub-case of MH sampling with an acceptance ratio equal to one, see e.g. Gelman et al. (2004) for proof, and this alleviates the need for tuning acceptance ratios and adapting proposal distributions for efficient exploration of

the stationary distribution. Further advantages of Gibbs sampling are discussed in Chapter 3. The main drawback is that it can lead to correlated posterior samples due to the coupling between the conditional posterior distributions.

Finally all of the MCMC methodology is computationally demanding due to the inherent sampling nature of the methods which might require anything from a few thousands to hundreds of thousands samples for convergence to the stationary distribution. A very important aspect for these methods is assessing and monitoring convergence which is described in detail in Chapter 3 together with the appropriate measures that are typically (Gelman et al. 2004) employed.

2.6 Deterministic Approximations

In this section further approximate Bayesian inference methods are introduced: the saddle-point or Laplace approximation and the variational Bayes methodology. Both approximations are deterministic in contrast with MCMC which stochastically explores the posterior space. The need for further approximations stems from the large computational requirements of MCMC methods that restricts their widespread application.

2.6.1 Saddle-point (Laplace) Approximation

In the previous section we went into the full length of the Bayesian problem and constructed samplers in order to sample our posterior distribution. An alternative way, which is perhaps the most straightforward, is to directly assume a specific functional form for the posterior distribution and approximate it. The saddle-point or Laplace's method (Duda et al. 2000, MacKay 2003) directly approximates the posterior with a Gaussian distribution which in the classification setting of interest follows as:

$$p\left(\mathbf{w}|\mathbf{t},\mathbf{X},\alpha\right) \approx N(\hat{\mathbf{w}}_{\text{MAP}},\boldsymbol{\Sigma})$$

where the mean $\hat{\mathbf{w}}_{\text{MAP}}$ is the *maximum a posteriori* estimate that we have introduced before and $\boldsymbol{\Sigma}$ is the curvature of the posterior at the maximum value.

The general intuition behind the saddle-point approximation is that we approximate an unnormalized distribution $f_*(\theta)$ by a Gaussian distribution based on the knowledge that the logarithm of a Gaussian distribution is a quadratic

function of the variables. Hence we want to express the logarithm of $f_*(\theta)$ in a quadratic form and then form back the Gaussian of interest. To do that we Taylor expand $\log f_*(\theta)$ around its mode $\hat{\theta}$:

$$\log f_*(\theta) \approx \log f_*(\hat{\theta}) - \frac{c}{2}(\theta - \hat{\theta})^2 \tag{2.43}$$

where $c = -\dfrac{\partial^2}{\partial\theta^2} \log f_*(\theta)\Big|_{\theta=\hat{\theta}}$ and by taking the exponential we form the unnormalized Gaussian:

$$f_*(\theta) \approx f_*(\hat{\theta}) \exp\left\{-\frac{c}{2}(\theta - \hat{\theta})^2\right\} \tag{2.44}$$

which considering the standard form of the normalising constant leads to the final saddle point approximation:

$$q(\theta) = \left(\frac{c}{2\pi}\right)^{1/2} \exp\left\{-\frac{c}{2}(\theta - \hat{\theta})^2\right\} \tag{2.45}$$

It might seem that we ended up were we started since we are expressing our approximation for the posterior based on parameters calculated from the posterior, but considering that Bayes rule gives us (since the marginal likelihood is a normalising term):

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Marginal Likelihood}} \propto (\text{Likelihood} \times \text{Prior}) = \text{Joint Likelihood} \tag{2.46}$$

we can calculate $\mathbf{w}_{MAP}$ and $\boldsymbol{\Sigma}$ (in analogy to $\hat{x}$ and c) for the maximum of the logarithm of the joint likelihood instead (in analogy to $\log f_*(x)$). Hence, we need the first and second derivatives of the (log) joint likelihood.

However, employing the typical sigmoid function for our likelihood in classification leads to a first derivative L which is a function of $\mathbf{w}$ and of nonlinear terms of $\mathbf{w}$. That implies that by setting it to zero we cannot retrieve its value explicitly. Instead we can use the Newton optimisation routine[13] which gives:

$$\mathbf{w}_{i+1} \longleftarrow \mathbf{w}_i - \left(\frac{\partial^2 L}{\partial \mathbf{w} \partial \mathbf{w}^\top}\right)^{-1} \frac{\partial L}{\partial \mathbf{w}} \tag{2.47}$$

Hence, since the covariance matrix $\boldsymbol{\Sigma}$ is also a function of derivatives of the logarithm of the joint likelihood :

[13] Or other suitable optimisation methods.

$$\boldsymbol{\Sigma} = -\left(\frac{\partial^2 L}{\partial \mathbf{w} \partial \mathbf{w}^\intercal}\right)^{-1} \tag{2.48}$$

we only need now to calculate the first and second derivatives of the logarithm of the joint likelihood for each case. In the following subsections we derive the Laplace approximation for some common classifiers employing the likelihood functions introduced in Section 2.4.1.

Binary Logistic Regression

For the logistic binary case and an expanded feature space $\boldsymbol{\phi}_n = \phi(\mathbf{x}_n) \in \mathbb{R}^M$ we have the joint likelihood as:

$$p(\mathbf{t}, \mathbf{w}|\mathbf{X}, \alpha) = \prod_{n=1}^{N} \frac{\exp(\mathbf{w}^T \boldsymbol{\phi}_n)^{t_n}}{1 + \exp(\mathbf{w}^T \boldsymbol{\phi}_n)} N_{\mathbf{w}}\left(\mathbf{0}, \alpha^{-1}\mathbf{I}\right) \tag{2.49}$$

where $t_n \in \{0, 1\}$, and the logarithm of the joint likelihood is:

$$L = \sum_{n=1}^{N} \left(t_n \mathbf{w}^\intercal \boldsymbol{\phi}_n - \log\left(1 + \exp\left(\mathbf{w}^\intercal \boldsymbol{\phi}_n\right)\right)\right) - \frac{\alpha}{2}\mathbf{w}^\intercal \mathbf{w} - \frac{M}{2}\log\left(\frac{2\pi}{\alpha}\right) \tag{2.50}$$

resulting in the following derivatives:

$$\frac{\partial L}{\partial \mathbf{w}} = \boldsymbol{\Phi}^\intercal \mathbf{t} - \boldsymbol{\Phi}^\intercal \mathbf{p} - \alpha \mathbf{w} \tag{2.51}$$

$$\frac{\partial^2 L}{\partial \mathbf{w} \partial \mathbf{w}^\intercal} = -\boldsymbol{\Phi}^\intercal \mathbf{V} \boldsymbol{\Phi} - \alpha \mathbf{I} \tag{2.52}$$

where $\mathbf{p} = \left[P(t_1 = 1|\mathbf{x}_1), \cdots, P(t_n = 1|\mathbf{x}_n), \cdots, P(t_N = 1|\mathbf{x}_N)\right]^\intercal$ is a $N \times 1$ vector, $\boldsymbol{\Phi}$ is a $N \times M$ matrix defined as:

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \ldots & \phi_M(\mathbf{x}_1) \\ . & & . \\ . & \phi_m(\mathbf{x}_n) & . \\ . & & . \\ \phi_1(\mathbf{x}_N) & \ldots & \phi_M(\mathbf{x}_N) \end{bmatrix}$$

and $\mathbf{V}$ is a diagonal $N \times N$ matrix with the non-zero diagonal elements defined as $\left[\mathrm{v}_{11}, \cdots, \mathrm{v}_{nn}, \cdots, \mathrm{v}_{NN}\right]^\intercal$ where each $\mathrm{v}_{nn} = P(t_n = 1|\mathbf{x}_n)(1 - P(t_n = 1|\mathbf{x}_n))$.

Multinomial Logistic Regression

The extension of the binary logistic case to the multinomial softmax case follows analogously for $c = 1, .., C$ classes where the joint likelihood is given by :

$$p(\mathbf{t}, \mathbf{w}_1, ..., \mathbf{w}_C | \mathbf{X}, \alpha) = \prod_{n=1}^{N} \prod_{c=1}^{C} \left[\frac{\exp\left(\mathbf{w}_c^{\intercal} \boldsymbol{\phi}_n\right)}{\sum_{c'} \exp\left(\mathbf{w}_{c'}^{\intercal} \boldsymbol{\phi}_n\right)} \right]^{t_{cn}} N_{\mathbf{w}_c}\left(\mathbf{0}, \alpha^{-1}\mathbf{I}\right) \quad (2.53)$$

$\mathbf{t}_n$ follows now a $1 - of - C$ encoding as a $C \times 1$ vector in which every element is zero, when the specific instance n does not belong to a specific class c, and one when it does. The logarithm of the joint likelihood is given by:

$$L = \sum_{n=1}^{N} \sum_{c=1}^{K} \left[t_{cn} \mathbf{w}_c^{\intercal} \boldsymbol{\phi}_n - \log \sum_{c'} \exp\left(\mathbf{w}_{c'}^{\intercal} \boldsymbol{\phi}_n\right) \right] - \frac{M}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha}{2} \mathbf{w}_c^{\intercal} \mathbf{w}_c \quad (2.54)$$

Taking derivatives with respect to $\mathbf{w}_c$:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}_c} &= \sum_{n=1}^{N} t_{cn} \boldsymbol{\phi}_n - \frac{\exp\left(\mathbf{w}_c^{\intercal} \boldsymbol{\phi}_n\right)}{\sum_{c'} \exp\left(\mathbf{w}_{c'}^{\intercal} \boldsymbol{\phi}_n\right)} \boldsymbol{\phi}_n - \alpha \mathbf{w}_c \\ &= \sum_{n=1}^{N} \left(t_{cn} - P\left(t_{cn} = 1 | \mathbf{x}_n\right)\right) \boldsymbol{\phi}_n - \alpha \mathbf{w}_c \end{aligned} \quad (2.55)$$

Now considering the second order derivatives for the Hessian[14] matrix:

$$\frac{\partial^2 L}{\partial \mathbf{w}_c \partial \mathbf{w}_c^{\intercal}} = \sum_{n=1}^{N} P(t_{cn} = 1 | \mathbf{x}_n) \left[P(t_{cn} = 1 | \mathbf{x}_n) - 1\right] \boldsymbol{\phi}_n \boldsymbol{\phi}_n^{\intercal} - \alpha \quad (2.56)$$

$$\frac{\partial^2 L}{\partial \mathbf{w}_c \partial \mathbf{w}_d^{\intercal}} = \sum_{n=1}^{N} P(t_{cn} = 1 | \mathbf{x}_n) P\left(t_{dn} = 1 | \mathbf{x}_n\right) \boldsymbol{\phi}_n \boldsymbol{\phi}_n^{\intercal} \quad (2.57)$$

Hence the Hessian matrix is an $MC \times MC$ symmetric matrix. Let $\boldsymbol{\Pi}$ be an $NC \times N$ block-matrix with diagonal matrices $\boldsymbol{P}_c$ of class likelihoods:

[14] Hessian is the square matrix with elements given by the second order partial derivatives of a function

$$\mathbf{\Pi} = \begin{bmatrix} \boldsymbol{P}_1 \\ \vdots \\ \boldsymbol{P}_c \\ \vdots \\ \boldsymbol{P}_C \end{bmatrix}$$

where the diagonal elements of $\boldsymbol{P}_c$ are:

$$[P(t_{c1} = 1|\mathbf{x}_1), \cdots, P(t_{cn} = 1|\mathbf{x}_n), \cdots, P(t_{cN} = 1|\mathbf{x}_N)]^\intercal$$

Then, if we also define a $\mathbf{\Psi}$ matrix to be an $NC \times MC$ diagonal block-matrix with the matrix $\mathbf{\Phi}$ $(N \times M)$ repeated C times:

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{\Phi} & & \\ & \mathbf{\Phi} & \\ & & \mathbf{\Phi} \end{bmatrix} \quad \text{where} \quad \mathbf{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1^\intercal \\ \vdots \\ \boldsymbol{\phi}_n^\intercal \\ \vdots \\ \boldsymbol{\phi}_N^\intercal \end{bmatrix}$$

we can write in matrix format the derivatives as:

$$\frac{\partial^2 L}{\partial \boldsymbol{w} \partial \boldsymbol{w}^\intercal} = \mathbf{\Psi}^\intercal \left(\mathbf{\Pi}\mathbf{\Pi}^\intercal - \mathbf{\Lambda} \right) \mathbf{\Psi} - \alpha \mathbf{I} \tag{2.58}$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = \mathbf{\Psi}^\intercal \left(\mathbf{t}^\dagger - \boldsymbol{\xi} \right) - \alpha \mathbf{w}^\dagger \tag{2.59}$$

$$\tag{2.60}$$

where $\mathbf{\Lambda}$ is a diagonal block instantiation of $\mathbf{\Pi}$, i.e an $NC \times NC$ matrix, and $\boldsymbol{\xi}$ is the diagonal of $\mathbf{\Lambda}$, i.e a $NC \times 1$ column vector that contains the concatenated diagonal elements of $\boldsymbol{P}_c \ \forall \ c \in \{1, \cdots, C\}$. Finally, $\mathbf{t}^\dagger$ and $\mathbf{w}^\dagger$ denote the target labels in a $NC \times 1$ and $MC \times 1$ column vector format in C stacks of N.

Binary Probit Regression

Again following the same approach, now with the binary probit sigmoid function and assuming a linear model $\mathbf{w}^\intercal \mathbf{x}_n$, the joint likelihood is given by:

$$p(\mathbf{t}, \mathbf{w}|\mathbf{X}, \alpha) = \prod_{n=1}^{N} P(t_n|\mathbf{x}_n, \mathbf{w}) N_{\mathbf{w}}\left(\mathbf{0}, \alpha^{-1}\mathbf{I}\right) \tag{2.61}$$

and the probit likelihood for a $t_n \in \{-1, 1\}$ encoding is defined as $\Phi\left(t_n \mathbf{w}^{\intercal} \mathbf{x}_n\right)$, where Φ the Gaussian CDF, leading to the logarithm of the joint likelihood:

$$L = \sum_{n=1}^{N} \left\{ \log \Phi\left(s_n \mathbf{w}^{\intercal} \mathbf{x}_n\right) \right\} - \frac{M}{2} \log\left(\frac{2\pi}{\alpha}\right) - \frac{\alpha}{2} \mathbf{w}^{\intercal} \mathbf{w} \tag{2.62}$$

Taking the first-order derivative:

$$\frac{\partial L}{\partial \mathbf{w}} = \sum_{n=1}^{N} \frac{N\left(t_n \mathbf{w}^{\intercal} \mathbf{x}_n|0, 1\right) t_n \mathbf{x}_n}{\Phi\left(t_n \mathbf{w}^{\intercal} \mathbf{x}_n\right)} - \alpha \mathbf{w} \tag{2.63}$$

which can be written in matrix format as:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{t}^{\intercal} \mathbf{\Lambda} \mathbf{X} - \alpha \mathbf{w} \tag{2.64}$$

with $\mathbf{X}$ the $N \times D$ design matrix and $\mathbf{\Lambda}$ the $N \times N$ diagonal matrix with non-zero elements $[\psi_1, \cdots, \psi_n, \cdots, \psi_N]^{\intercal}$ where ψ_n:

$$\psi_n = \frac{N\left(t_n \mathbf{w}^{\intercal} \mathbf{x}_n|0, 1\right)}{\Phi\left(t_n \mathbf{w}^{\intercal} \mathbf{x}_n\right)} \tag{2.65}$$

The second order derivatives, using the notation $y_n = t_n \mathbf{w}^{\intercal} \mathbf{x}_n$, are then given by:

$$\frac{\partial^2 L}{\partial w_i \partial w_i} = -\sum_{n=1}^{N} \psi_n^2 x_{ni}^2 - \sum_{n=1}^{N} \psi_n x_{ni}^2 y_n - \alpha \tag{2.66}$$

$$\frac{\partial^2 L}{\partial w_i \partial w_j} = -\sum_{n=1}^{N} \psi_n^2 x_{ni} x_{nj} - \sum_{n=1}^{N} \psi_n x_{ni} x_{nj} y_n \tag{2.67}$$

Hence, now we can form the Hessian matrix as:

$$\mathbf{X}^{\intercal} \mathbf{V} \mathbf{X} - \alpha \mathbf{I} \tag{2.68}$$

where $\mathbf{V}$ a $N \times N$ diagonal matrix with non-zero elements $\mathbf{v}_n = -\psi_n(\psi_n + t_n \mathbf{w}^{\intercal} \mathbf{x}_n)$

2.6.2 Variational Free Energy Minimisation

The Gaussian assumption of the saddle-point approximation is strong and in some cases significantly violated leading to poor estimates for regression and classification when the true posterior deviates from normality and log-concavity. Furthermore, the Laplace framework is derived on the basis of a point estimate (the mode) of the function and hence may ignore important global characteristics (Bishop 2006).

In this section we briefly review a further deterministic approach, *variational free energy minimisation*, which adopts a global perspective on the approximation of an intractable (posterior) distribution and avoids the basis dependence from which MAP and saddle-point approximations suffer[15]. Variational methods are defined as approximations (MacKay 2003) only due to the restriction of proposed functions to belong within a certain family (e.g. Gaussian (Opper and Archambeau 2009)) or to satisfy a structural assumption (mean field ensembles) as we shall see in Chapter 4. The resulting approximations tend to be more compact than the true distribution (MacKay 2003, Damoulas and Girolami 2009a) as they typically underestimate the covariance structure.

The intuition behind variational free energy minimisation is simple. Assume the distribution of interest is $P(x|\beta) = \frac{1}{Z}P_*(x|\beta)$ where Z is the normalising constant (partition function) that is unobtainable and we can only obtain the unnormalized $P_*(x|\beta)$ parameterised by β. The variational framework proposes an approximating distribution $Q(x,\theta)$ parameterised by θ and minimises the relative entropy (non-symmetric measure) between P and Q via adjusting[16] θ. The relative entropy is given by the *Kullback-Leibler* divergence:

$$\text{Relative Entropy} = D_{\text{KL}}\left(Q(x,\theta)||P(x|\beta)\right) \tag{2.69}$$

noting that the KL divergence satisfies Gibbs' inequality $D_{\text{KL}}\left(Q||P\right) \geq 0$ and that is not a metric due to the asymmetry:

$$D_{\text{KL}}\left(P||Q\right) \neq D_{\text{KL}}\left(Q||P\right) \tag{2.70}$$

[15]Despite this, it has been observed by MacKay (2001) that variational methods might not always perform better than ML or MAP point estimators due to model "pruning" of degrees of freedom (symmetry breaking).

[16]The parameter θ here is used generically as it can represent different functional forms or parameterisations within a specific family of functions or even different structural assumptions (i.e. factorised ensembles).

the relative entropy of interest is given by:

$$D_{\mathrm{KL}}\left(Q(x,\theta)||P(x|\beta)\right) = \int Q(x,\theta)\log\frac{Q(x,\theta)}{P(x|\beta)}dx \tag{2.71}$$

which can be decomposed to the following terms:

$$D_{\mathrm{KL}}\left(Q(x,\theta)||P(x|\beta)\right) = \underbrace{\int Q(x,\theta)\log\frac{Q(x,\theta)}{P_*(x|\beta)}dx}_{\text{Variational Free Energy}} - \underbrace{\log\frac{1}{Z}}_{\text{Free Energy}} \tag{2.72}$$

Hence, considering Gibb's inequality we can see that the variational free energy is an upper bound on the true free energy. The bound is minimised by minimising the variational free energy with appropriate θ, and it is zero for the obvious solution of $Q(x,\theta) = P(x|\beta)$. Hence, minimising the variational free energy is equivalent to maximising a *lower bound* on the normalising constant Z. The variational free energy can be further decomposed as follows:

$$\underbrace{\int Q(x,\theta)\log\frac{Q(x,\theta)}{P_*(x|\beta)}dx}_{\text{Variational Free Energy}} = \underbrace{\int Q(x,\theta)P_*(x|\beta)}_{\mathbb{E}_Q\{P_*(x|\beta)\}} - \underbrace{\int Q(x,\theta)\log\frac{1}{Q(x,\theta)}}_{\text{Entropy}} \tag{2.73}$$

where the first term is the expected value of the unnormalized density under the approximating distribution and the second term is the entropy of the approximating distribution Q.

In the Bayesian setting the unnormalized distribution of interest is the posterior distribution for which the normalising constant is the marginal likelihood (Baye's rule). Hence, the *variational Bayes* approaches follow the variational free energy minimisation principle and lower bound Z which corresponds to the marginal likelihood, also termed as *model evidence* (MacKay 1992b).

So far, no assumptions were introduced regarding the functional form, family or nature of the approximating distributions Q. In Chapter 4, where a variational Bayes approximation is employed for the problem addressed in this thesis, we will introduce the specific adoption of the variational method within the *mean field* framework where a specific factorised assumption on the approximating densities Q is employed.

2.7 Sparsity and Shrinkage methods

In the previous sections we reviewed approaches for regression and classification where the resulting model utilises the entire *training set* of past observations and attributes (denoted by the design matrix $\mathbf{X}$, the feature expansion $\phi(\mathbf{X})$ or the kernel matrix $\mathbf{K}$) for predicting novel responses. In many cases this is unfeasible and undesirable, due to memory and computing restrictions, and in this section *sparse* approaches that utilise a *subset* of observations and/or attributes are introduced.

The main reasons for aiming at sparse solutions are:

- **Scalability** - Methods that utilise the whole training set become computationally unfeasible for large data collections (either in number of attributes D or number of samples N). Kernel-based methods that are governed by an $\mathcal{O}(N^3)$ complexity, require sparse solutions to scale up for large application scenarios.

- **Interpretation** - Identifying the significant samples or attributes for the prediction task at hand can be crucial in some application areas such as bioinformatics, medical informatics and all cases where information and intuition about the problem's characteristics are more important than just a prediction output. The context in which a sample or attribute is judged as significant for the prediction task can be *statistical*, e.g. marginal likelihood (Tipping 1999, Damoulas et al. 2008) or predictive likelihood (Lawrence et al. 2003, Girolami and Rogers 2006), *information theoretic*, e.g. information gain (MacKay 1992a), or *geometric*, e.g. decision boundary construction (Vapnik and Chervonenkis 1964, Vapnik 1995).

- **Prediction Accuracy** - Sparse models can improve the prediction accuracy on a problem as they sacrifice bias (how well the model describes the specific training set of the phenomenon) in order to reduce variance (how much the resulting model will vary when trained on a different training set of the same phenomenon). This is achieved by obtaining a sparse solution that is based on a subset of *informative* observations or attributes and hence less likely to fit the noise.

In the following subsections a brief review of the main sparsity and shrinkage methods is offered together with the corresponding advantages and limitations.

2.7.1 Ridge Regression and the Lasso

In statistical learning theory sparsity is achieved via appropriate *regularisation* and different linear regression approaches have been developed according to the specific penalising term used. *Ridge Regression* (Hastie et al. 2001) adds to the OLS estimate a quadratic penalising term and hence it effectively minimises:

$$\sum_{i=1}^{N} \left(y_i - \mathbf{w}^\top \mathbf{x}_i \right)^2 + \lambda \sum_{d=1}^{D} w_d^2 \tag{2.74}$$

while the *Lasso* (Tibshirani 1996) employs a different nonlinear penalty term (L^1 norm) and minimises:

$$\sum_{i=1}^{N} \left(y_i - \mathbf{w}^\top \mathbf{x}_i \right)^2 + \lambda \sum_{d=1}^{D} |w_d| \tag{2.75}$$

The subtle differences in the penalising terms have a significant effect on the resulting estimates and obtained sparsity. The Lasso has better *interpretation* properties as it completely shrinks regression coefficients to zero and translates others (Tibshirani 1996), in contrast with ridge regression whose quadratic penalty term only scales all of the coefficients by a constant factor.

Both approaches use the same amount of shrinkage for each regression coefficient, as there is a global factor λ, and hence (coefficient) selection results can be inconsistent. Towards that direction, recent work by Zou (2006) proposed the *adaptive Lasso*, an extension that utilises individual shrinkage levels for each coefficient:

$$\sum_{i=1}^{N} \left(y_i - \mathbf{w}^\top \mathbf{x}_i \right)^2 + \sum_{d=1}^{D} \lambda_d |w_d| \tag{2.76}$$

The shrinkage levels are generally estimated through *cross-validation* (multiple partitions of the dataset to training and test sets) or an analytical unbiased estimate of risk (Tibshirani 1996, Berger 1985). For further theoretical analysis, convergence guarantees and direct connections to the standard penalised least squares estimator see Zou (2006), Wang and Leng (2007) and references within.

2.7.2 Sparsity in Kernel Methods

The previous section reviewed standard sparse methods on linear models with a linear estimating function. The same analysis directly applies to basis function or other feature expansions for obtaining nonlinear responses. However, the induced shrinkage from the penalty terms is with respect to the regression coefficients and acts on the features of each input sample and not on the size of the training set. Hence any sparsity is on the dimensionality of the regressors and identifies significant and non-significant attributes based on the MSE loss function.

In the kernel setting, similar penalising constraints on the regression coefficients results in *sample-wise* sparsity, i.e. a kernel-based Lasso (Roth 2004) that will identify significant and non-significant training samples instead of attributes. The general kernel-based lasso function to be minimised is:

$$\sum_{i=1}^{N} \left(y_i - \mathbf{w}^{\mathsf{T}} \mathbf{k}_i \right)^2 + \lambda \sum_{i=1}^{N} |w_i| \tag{2.77}$$

where now the regression coefficients $\mathbf{w} \in \mathbb{R}^N$ operate on the kernel matrix and the shrinkage effectively prunes out training samples.

One other prominent sparse kernel method is the *Support Vector Machine* (SVM) (Vapnik 1995) which is a geometric method that maximises the smallest distance between the decision boundary and the closest samples (margin). This results in a penalising term on the regression coefficients $\frac{1}{2}||\mathbf{w}||^2$ which is the L^2 norm. The resulting sparse solutions from SVMs retain only training samples that are close to the decision boundary, due to the initial assumptions of the model, and are termed as *support vectors* as they are responsible for defining or "supporting" the boundary.

SVMs have the drawback of not producing probabilistic outputs as they are "decision" machines (Bishop 2006) and the resulting sparsity levels are moderate when compared to other alternative sparse kernel methods such as the *Relevance Vector Machine* that is briefly described in the next section.

2.7.3 Sparsity in Bayesian Inference

In the Bayesian framework, the analogous sparsity-inducing role to regularisation is performed by the prior distributions placed on the model's variables. For

example, the Lasso approach is equivalent to placing a Laplace prior on the regression coefficients. Hence in this framework no ad-hoc penalising term needs to be introduced but we can formally place appropriate prior distributions that induce sparsity via the principle of Automatic Relevance Determination (ARD) (MacKay 2004).

ARD describes the Bayesian process by which sparsity inducing prior distributions on the parameters, such as the Laplace or the Student-t prior, effectively determine the "relevance" of a feature (or sample in kernel-based methods) based on the *evidence* from the data. The two dominant ARD approaches within the Bayesian paradigm and the Machine Learning community are the Relevance Vector Machines (RVMs) (Tipping 2001) and the Informative Vector Machines (Lawrence et al. 2003).

RVMs employ a hierarchical prior formulation with a zero-mean Gaussian distribution on the parameters and a Gamma distribution on the scales of the Gaussian. This results (by marginalising the scales) to an implicit Student-t distribution on the regression coefficients which, similarly to the Laplace, has probability mass at the mean (zero) and on the tails of the distribution. This enforces coefficients with no evidence to shrink to zero and significant ones to be non-zero.

The main driving force behind the RVM formalism is the maximisation of the marginal likelihood with respect to the hyper-parameters (regression):

$$p(\mathbf{y}|\boldsymbol{\alpha}, \sigma) = \int p(\mathbf{y}|\mathbf{w}, \sigma) p(\mathbf{w}|\boldsymbol{\alpha}) d\mathbf{w} \tag{2.78}$$

where as before $\boldsymbol{\alpha}$ are the scales and σ^2 is the noise term in regression. This maximisation is known as *type-II maximum likelihood* (type-II ML) and it leads to efficient and incremental ways (Tipping and Faul 2003, Faul and Tipping 2002) to prune out and include features or samples based on their contribution to the marginal likelihood. The resulting solutions are typically very sparse but the scalability to multiclass classification is problematic[17] due to the $MC \times MC$ Hessian matrix required for the Laplace approximation.

A further sparse Bayesian approach was suggested by Lawrence et al. (2003) within the context of Gaussian Processes (Rasmussen and Williams 2006) that model directly the estimating function $\hat{\mathbf{y}}$ by placing an appropriate (data depen-

[17]This thesis is addressing that issue with an efficient multiclass method in Chapter 5.

dent) zero-mean Gaussian distribution directly on the possible functions. The sparse approximation follows an information theoretic criterion based on the entropy contribution of each sample and it is competitive with SVMs in training processing times.

The criterion proposed is the *differential entropy score* and in effect favours samples that reduce the variance of the predictive distribution. Sparsity levels are comparable to SVMs with the additional benefit of probabilistic outputs. However, similarly to SVMs, they are binary classification methods and require multiple dichotomy of the solution space with one versus one or other ad-hoc procedures.

2.8 Ensemble Learning

So far we have considered the standard classification scenario where a training set of input predictors and target responses $\{\mathbf{x}_i, t_i\}_{i=1}^N$ is available and used for fitting a single classification model. *Ensemble Learning* methods[18] (Dietterich 2000b, Kuncheva 2004) are approaches that propose a collection of different models (e.g. classifiers, regressions, feature constructions or "experts") that are appropriately fused towards an overall response.

There are two main situations that motivate the use of ensemble learning methods; when there is a greater need to improve performance measures over individual models with less concern for the additional computational costs associated with multiple models, and when the input predictors have *multiple* (S) and possibly *heterogeneous* feature (attribute) sets or information sources $\{\mathbf{x}_i^{(s)}, t_i\}_{i,s=1}^{N,S}$, which is the problem setting of this thesis.

In the following subsections we review the two main categories for ensemble learning with discrete targets: *classifier combination* and *multiple kernel learning* methods. Both approaches can address the two motivating scenarios for ensemble learning, offering different advantages and disadvantages in each situation. According to these, we motivate the research that has been undertaken in this thesis and the specific research directions that have been addressed.

[18]The name is also used sometimes to describe variational methods that assume a factorised ensemble of approximate posteriors, see Chapter 4.

2.8.1 Classifier Combination

Some of the main approaches (Dietterich 2000b, Bishop 2006) in classifier combination methods are *boosting*, *bagging*, *stacking*, Bayesian *mixture of experts* and standard *voting rules*. They differ in their modelling assumptions and construction but they all arrive to the necessity for *diversity* between the individual classifiers that are used as base learners (Kuncheva and Whitaker 2003).

Boosting (Schapire 2003) manipulates the training set to generate multiple hypotheses by applying multiple base classifiers (weak learners) sequentially and weighing the data based on previous classification performance. In that way misclassified data from the previous classifier gains more weight in the next step and hence further emphasised and considered in the present classification level. The final classifier is a weighted vote of the individual classifiers based on their performance on the training set. Adaboost (Freund and Schapire 1996) is the best known example of boosting methods that are best performing in large training sets with relatively trivial classification noise (Dietterich 2000a).

Bagging (Breiman 1996) derives from *bootstrap aggregation* and employs the bootstrap procedure to sample by replacement copies of the training set on which base classifiers are trained. The main benefits of bagging are exploited when the base classifiers are relatively *unstable*, with respect to small changes in the training set, such as neural networks and decision trees.

Stacking (Wolpert 1992) is a meta-learning approach in which different base classifiers are trained typically on the same dataset and on the second level all the classifier outputs are used as a new feature space for further classification. This has the benefit of learning the combination of classifiers in a data-driven manner and avoids ad-hoc voting schemes. Stacking, like the previous methods, is computationally intensive due to the multiple training levels and does not always improve upon the best base learner (Džeroski and Ženko 2004).

A Bayesian approach to combining classifiers is *mixture of experts* (Jacobs et al. 1991) which can be seen as a mixture model of components that are conditioned on the input predictors and are associated to mixing coefficients drawn by appropriate gating (sigmoid) functions. The general framework for a mixture of experts is:

$$p(\mathbf{t}|\mathbf{X}) = \sum_{s=1}^{S} \pi_s(\mathbf{X}^{(s)}) p_s(\mathbf{t}|\mathbf{X}^{(s)}) \tag{2.79}$$

where $p_s(\mathbf{t}|\mathbf{X}^{(s)})$ are the S "experts", $\pi_s(\mathbf{X}^{(s)})$ are the gating coefficients, and $\mathbf{X} = \left\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(s)}, \ldots, \mathbf{X}^{(S)}\right\}$ defines the set of all the S feature spaces. The intuition behind this approach is that different components are better in describing different regions of the input space and the gating coefficients reflect that by choosing an expert according to the input. The initial framework was extended to a hierarchical scheme (Jordan and Jacobs 1994) with multi-level gating functions that results in a mixture of mixtures and is more flexible in describing partitions of the input space. The standard inference scheme follows an EM procedure for maximum likelihood estimation which was extended to a Bayesian variational procedure by Bishop and Svensen (2003).

Finally, standard *voting rules* can be employed for classifier fusion depending on their output nature. For models that do not produce posterior probabilities of class membership but crisp labels, several standard voting schemes such as majority rules and oracles have been proposed (Kuncheva 2004) but are outside the scope of this thesis. For probabilistic models or models that can produce "probabilistic" outputs through post-processing (squashing functions) the following combination rules (Kittler et al. 1998) have been proposed:

- **Summation Rule**: A simple average of the posterior probabilities over classes.

$$P\left(t_n = c|\mathbf{X}\right) = \frac{1}{S}\sum_{s=1}^{S} P_s\left(t_n = c|\mathbf{X}^{(s)}, \boldsymbol{\theta}^{(s)}\right) \tag{2.80}$$

- **Product Rule**: A normalised product of the posteriors over classes.

$$P\left(t_n = c|\mathbf{X}\right) = \frac{\prod_{s=1}^{S} P_s\left(t_n = c|\mathbf{X}^{(s)}, \boldsymbol{\theta}^{(s)}\right)}{\sum_{c'=1}^{C}\prod_{s'=1}^{S} P'_s\left(t_n = c'|\mathbf{X}^{(s')}, \boldsymbol{\theta}^{(s')}\right)} \tag{2.81}$$

- **Max Rule**: Select the class that has the maximum posterior probability over the S classifiers.

$$t_n = \max_{s,c=1}^{S,C} P_s\left(t_n = c|\mathbf{X}^{(s)}, \boldsymbol{\theta}^{(s)}\right) \tag{2.82}$$

- **Majority Rule**: Select the class that is predicted by the majority of the

S classifiers.

$$t_n = \underset{s=1}{\overset{S}{\text{maj}}} \max_{c=1}^{C} P_s\left(t_n = c|\mathbf{X}^{(s)}, \boldsymbol{\theta}^{(s)}\right) \tag{2.83}$$

where $P_s\left(t_n|\mathbf{X}^s, \boldsymbol{\theta}^s\right)$ is the class membership probability from the s^{th} classifier parameterised by $\boldsymbol{\theta}^s$ and trained on the s^{th} training set (feature set) in the case of multiple information sources.

The theoretical justification for the product rule comes from the *independence* assumption of the feature spaces, where $\mathbf{x}_n^i, \mathbf{x}_n^j \;\; \forall \;\; i, j \in \{1, \ldots, S\}$ are assumed to be uncorrelated, and the mean combination rule is derived on the opposite assumption of extreme correlation. The hope is that the individual errors of the classifiers will be different (diversity) and the synergetic effect of the combination will cancel them out and hence reduce the overall classification error.

2.8.2 Multiple Kernel Learning

In the previous subsection we reviewed classifier combination strategies that can be applied to different partitions of the training data from a single source or can be used to address problems with multiple feature sets generated from possibly heterogeneous information sources. However, such approaches are computationally demanding as they require multiple models and training regimes. Also, their theoretical justification is not always clear and the assumptions (e.g. independence of sources) employed towards that are most of the times unsupported and unrealistic. In this section a modern alternative approach, *multiple kernel learning*, that tackles the scenario of multiple sources of information is introduced, which will lead us into the problem formulation and goals of this thesis.

The intuition behind *multiple kernel learning* (MKL) is to create a common metric across the possibly heterogeneous sources S by embedding them into high-dimensional feature spaces via the kernel trick. Each source is then associated with a unique kernel matrix $\mathbf{K}_s$, the *base kernel*, which expresses the similarity between input samples based on the information from each source. Hence, this common metric of all the base kernels can now be combined into an overall composite kernel as depicted in Figure 2.2, where a single classifier operates. The definition of the MKL setting is to learn the kernel combination parameters associated with each base kernel and hence infer the contribution of each information source.

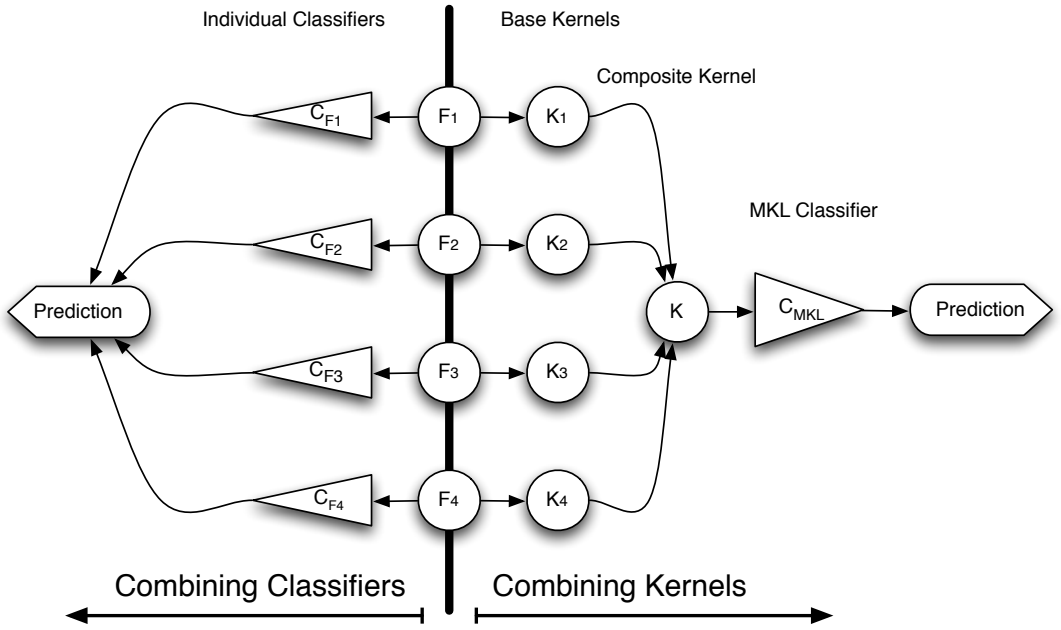

Figure 2.2: The intuition behind Multiple Kernel Learning and the differences with Classifier Combination methods.

Various MKL approaches have been proposed in the last decade and the great majority of them are from an optimisation and statistical learning theory (SLT) perspective employing Support Vector Machines (SVMs). We review the related research first and then introduce the few Bayesian exceptions in the last section where the motivation for this thesis is defined.

One of the first directions towards MKL was explored by Joachims et al. (2001) in the context of hypertext classification where general conditions based on a loose generalisation bound are derived for when the fixed combination of two base kernels is expected to improve upon any of the individual base ones. The scope of the work was limited to the case of soft margin SVMs with two base kernels and the drawn conclusions were that a combination is beneficial when the base kernels achieve approximately similar performance while their Support Vectors (SVs) are different.

The first MKL methodology was proposed by Lanckriet et al. (2002) (extended in (Lanckriet et al. 2004)) were an expensive semi-definite programming approach (SDP), which results in a quadratically constrained quadratic program (QCQP), with best-case complexity of $\mathcal{O}(SN^3)$ is employed to tackle convex linear combinations of base kernels:

$$\mathbf{K} = \sum_{s=1}^{S} \beta_s \mathbf{K}_s \tag{2.84}$$

where μ_s the kernel combination parameters, constrained such that $\mu_s \geq 0$, and S the number of base kernels $\mathbf{K}_s$ from a set of candidate kernels $\mathcal{K}$.

The approach has direct connections to the work of Cristianini et al. (2001) and the SDP approach reduces to the quadratically constrained quadratic program described in the latter for the optimisation of the alignment between the kernel and the target responses. In the alignment, the kernel is decomposed to eigenvector representations of the form $\mathbf{K} = \sum_{i=1}^{N} \mu_i \mathbf{u}_i \mathbf{u}_i^{\mathsf{T}}$, where $\mathbf{u}_i$ the i^{th} eigenvector of the original kernel, and their outer product can be seen as the base kernels to be informatively combined. Sun et al. (2004) showed that when the objective function is the target alignment proposed by Cristianini et al. (2001) the optimal solution for the convex linear combination rule results in a generalized eigenvalue problem.

Further work in the SLT discipline has focused on extending Lanckriet's approach to more *efficient* optimisations such as the work by Bach et al. (2004) where the QCQP is recast as a second-order cone programming that can be solved with sequential minimal optimisation (SMO) methods. Another approach by Sonnenburg, Ratsch and Schafer (2006), extended in Sonnenburg, Rätsch, Schäfer and Schölkopf (2006), recasts the same problem as a semi-infinite linear program (SILP) that can be solved by recycling existing SVM implementations. The above methods employ a block regularisation with a mixed L^1, L^2 norm on the regression coefficients in contrast with the work by Rakotomamonjy et al. (2007) (extended in Bach et al. (2008)) were an L^2 norm is employed on the block regularisation with a sparsity enforcing L^1 norm (Lasso type regularisation) on the kernel combination parameters. This has direct connections (Bach 2008) to the *Group Lasso* (Meier et al. 2008) where the L^1 regularising term that we have seen in previous sections is now applied to groups of attributes. A further alternative regularisation for MKL was proposed by Kloft et al. (2008) where an L^2 norm is employed for the kernel combination parameters leading to non-sparse combinations that are shown to be more robust in certain cases.

Another approach directly related to MKL, namely *hyperkernels*, was proposed by Ong et al. (2003) and later extended in Ong and Smola (2003) and Ong et al. (2005). The intuition is to address the general problem of learning

the *kernel function*, in contrast with the previous methods that learn the *kernel matrix*, by an optimisation on the space of kernels (resulting in a solution which defines a kernel on that space, hence the term hyperkernel). As the optimal solutions are linear combinations of base kernels, this can also be seen as an MKL problem in addition to the original goal of learning the optimal kernel within a parameterised family (e.g Gaussian kernels). The authors offer standard optimisation techniques such as QCQP and SILP in addition to ways of constructing loss functions for these spaces (e.g. the kernel target alignment of Cristianini et al. (2001)) that are termed as quality functionals. Further work by Kondor and Jebara (2007) offers closed form Gaussian and Wishart hyperkernel formulations and examines their use for dimensionality reduction. The hyperkernel methods are restricted though on optimisation within parameterised families of kernels which limits their applicability. Especially in the presence of heterogeneous sources that typically require very different kernel functions (Damoulas and Girolami 2008).

The reviewed approaches so far consider stationary combinations of base kernels, where the relative combination parameters of the base kernels do not vary among input examples. Research towards *non-stationary* kernel combinations was first reported by Lewis et al. (2006a) where the kernel combination parameters were sample-dependent within a generative latent variable model employing variational inference. The method follows a maximum entropy framework (Jaakkola, Meila and Jebara 1999) that minimises iteratively the divergence between the prior and the posterior over the parameters using an EM procedure. The drawback of employing sample-dependent kernel combination parameters is the increased computational complexity which restricts the scalability of the method.

An alternative *non-stationary* MKL approach by Gönen and Alpaydin (2008) follows similar ideas to the mixture of experts methodology in classifier combination and proposes a binary classifier with a gating function that learns different kernel combinations for regions of the input space. The model is an extension of the SVM approach by Bach et al. (2004) and the main assumption is that the different kernel combination regions of the input space are linearly separable. Another non-stationary approach with the SVM algorithm is followed by Lee et al. (2007) where Gaussian kernels with different width parameters form a "compositional" kernel matrix where the original base kernels are in the diagonal

and the mixtures of (Gaussian) base kernels in the off-diagonal. The approach however does not scale up as it results in an $SN \times SN$ matrix where S, N are the number of sources and samples respectively.

Further research on learning a more complex combination than the convex linear rule that has been employed in the above works was reported by Moguerza et al. (2004) and very recently by Varma and Babu (2009). The first authors employ a functional combination of kernels of the form $\mathbf{K} = \sum_{s=1}^{S} \mathbf{W}_s \mathbf{K}_s$ where the matrix $\mathbf{W}_s$ has elements $w_{i,j}$ as nonlinear functions of the input samples $\mathbf{x}_i, \mathbf{x}_j$. The composite kernel is not inferred during the training regime but pre-computed and a standard SVM algorithm is trained on the resulting kernel. The work in Varma and Babu (2009) builds directly on the previously aforementioned MKL developments within the SVM methodology and offers non linear combinations by generalising both the objective and the regularisation steps of SVMs. However, this comes at the cost of losing convexity and hence the ability to descend or ascend to the global minimum or maximum.

Other approaches to MKL that have been proposed include the work of Crammer et al. (2003) where boosting is employed for the construction of the composite kernel (computed outside the training phase) based on optimisation with respect to the kernel target alignment objective and the work by Fung et al. (2004) where the kernel fisher discriminant approach (kernelized version of standard LDA/LFD) is adopted to MKL, leading to a biconvex formulation.

So far the reviewed approaches to MKL have been developed within the SLT framework and the SVM classification approach that has three main drawbacks. First, such approaches are *non-probabilistic* and hence don't provide uncertainty estimates for the final predicted responses in either regression or classification scenarios that are crucial for risk assessment and further decision making. Secondly, the SVM classifier is by definition a *binary* classification method and it can only address multiclass MKL problems via additional assumptions or ad-hoc procedures such as the use of feature maps (Zien and Ong 2007) or decomposition of the problem to multiple binary ones (Ye et al. 2008). Finally, *prior knowledge* cannot be taken into account and integrated in a systematic way in either the objective function or the regularisation terms.

To that end, we now review the relatively few Bayesian approaches to MKL, besides the ones that this thesis will propose. One of the first Bayesian approaches towards MKL is the work by Zhang et al. (2004) where a Tanner-Wong

data augmentation algorithm is employed to learn the kernel matrix which encodes the test observations as "missing data". The base kernels are Wishart-distributed with hierarchical priors on the parameters and the method proposes convex linear combinations of these kernels to effectively infer the complete data. An alternative approach by Girolami and Rogers (2005) casts the problem in a standard classification scenario with heterogeneous sources of information and proposes a convex linear combination of base kernels with the logistic likelihood. Approximate inference via saddle-point and variational methods is offered for this binary classification setting.

Finally, the MKL problem has been also recently examined within the Gaussian Processes field of Bayesian inference by Girolami and Zhong (2007) where a convex linear summation of covariance functions was proposed for data integration and very recent unpublished work by Christoudias et al. (2009) follows the *localised* kernel combination and mixture of experts ideas and proposes a non-stationary combination of covariance functions. The proposed method however requires inference of $S(H + N^2)$ parameters, where H the number of hyper-parameters for each covariance function, and hence further low-rank approximations of the covariance matrices are employed.

The *methodological motivation* for this thesis stems from the lack of efficient probabilistic and multiclass MKL approaches that can easily handle heterogeneous sources of information, a need evidenced from the aforementioned existing literature. The emphasis therefore is placed on proposing explicitly multiclass approaches, alternative composite kernel construction rules and a collection of approximate inference procedures for *probabilistic multiple kernel learning*.

Chapter 3

Probabilistic Multiple Kernel Learning

3.1 Introduction

In this chapter[1] a probabilistic multiple kernel learning (pMKL) framework for multinomial classification is introduced. Alternative formulations for constructing the composite kernel together with Markov chain Monte Carlo (MCMC) sampling solutions are proposed, based on a hierarchical Bayesian approach that introduces prior distributions over random variables. The chapter offers a comparison between a Metropolis-Hastings (MH) and a Gibbs MCMC sampling scheme regarding posterior sampling efficiency, autocorrelation and the resulting effective sampling size. Finally, an approach for model selection through marginal likelihood estimation is presented together with the computational complexity of the algorithms and concluding remarks.

3.2 Constructing the Composite Kernel

Inference of the kernel combination parameters $\boldsymbol{\beta}$ is the main objective of MKL approaches that in most reviewed cases employ the convex linear combinatorial rule for the construction of the composite kernel. Another level of learning considers the kernel parameters $\boldsymbol{\theta}^{(s)}$ which, as mentioned before, control the level of smoothness applied to each feature space s and hence can be used to identify

[1]Parts of this work have already appeared in (Damoulas and Girolami 2009b, Damoulas and Girolami 2009c)

the significant features within each feature space in the manner of Automatic Relevance Determination (ARD) (Neal 1996).

In this section, besides revisiting the convex linear combinatorial approach, alternative rules for combining kernels are introduced and theoretically justified. Selection of the appropriate combinatorial rule can be achieved via prior knowledge, cross-validation or estimation of the marginal likelihood.

3.2.1 Fixed Combination

The baseline kernel combination rule consists of simply adding the kernels and dispensing the need for combination parameters $\boldsymbol{\beta}$ and associated inference procedures. The composite kernel elements i, j for S information sources with feature sets $\mathbf{X} = \left\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(s)}, \ldots, \mathbf{X}^{(S)}\right\}$ are given by:

$$k\left(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\Theta}\right) = \frac{1}{S} \sum_{s=1}^{S} k_s \left(\mathbf{x}_i^{(s)}, \mathbf{x}_j^{(s)}, \boldsymbol{\theta}^{(s)}\right) \tag{3.1}$$

where $\boldsymbol{\Theta}$ denotes all the parameters $\boldsymbol{\theta}^{(s)}$ of the s base kernel elements k_s. The advantage of this rule is the reduction of additional computation for inferring combination parameters. It has been shown (Lewis et al. 2006b, Damoulas and Girolami 2008) in certain cases to perform as well as parameterised combinations, when the individual sources lead to correlated similarities. Computation of the Frobenius inner product between kernels constructed from different sources is a good measure of whether a parameterised combination should be expected to improve.

3.2.2 Convex Linear Combination

This linear combination rule, which is the standard for existing MKL methods defines the composite kernel elements as:

$$k\left(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\beta}, \boldsymbol{\Theta}\right) = \sum_{s=1}^{S} \beta_s k_s \left(\mathbf{x}_i^{(s)}, \mathbf{x}_j^{(s)}, \boldsymbol{\theta}^{(s)}\right) \text{ with } \sum_{s=1}^{S} \beta_s = 1 \text{ and } \beta_s \geq 0 \; \forall \; s \tag{3.2}$$

Contrary to the fixed rule, an inference procedure is now required with additional computations, but it offers the benefits of learning the *significance* of

individual sources, and hence can accommodate cases where suspect or corrupted feature spaces might be present. The combination parameters $\boldsymbol{\beta}$ are defined over a simplex in order to ensure statistical weak identifiability and a p.s.d kernel matrix.

3.2.3 Binary Combination

In the case of the binary combination method, $\boldsymbol{\beta}$ is a binary vector switching base kernels on or off. The approach has been motivated by the work of Holmes and Held (2006) on covariate set uncertainty which we now modify as a *kernel set uncertainty*.

$$k\left(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\beta}, \boldsymbol{\Theta}\right) = \sum_{s=1}^{S} \beta_s k_s \left(\mathbf{x}_i^{(s)}, \mathbf{x}_j^{(s)}, \boldsymbol{\theta}^{(s)}\right) \text{ with } \beta_s \in \{0,1\} \ \forall \ s \tag{3.3}$$

The binary combination rule is suitable for large multi-feature problems where hard decisions of inclusion or dropping a specific kernel might be needed.

3.2.4 Product Combination

The product combination method is fundamentally different from the other approaches. The base kernels are no longer added together in a specific way but instead multiplied element-wise. The composite kernel elements are given by:

$$k\left(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\Theta}\right) = \prod_{s=1}^{S} k_s \left(\mathbf{x}_i^{(s)}, \mathbf{x}_j^{(s)}, \boldsymbol{\theta}^{(s)}\right) \tag{3.4}$$

There is no longer the need for combinatorial weights $\boldsymbol{\beta}$, which means that the model simplifies and inference is less computationally expensive but again the ability to infer the significance of the sources is lost.

3.2.5 Weighted Product Combination

Finally, another non-linear kernel construction is the parameterised extension of the product rule which retains the advantages of weighted rules and the computational trade-offs by exponentiating the kernel elements to the β_s power:

$$k\left(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\beta}, \boldsymbol{\Theta}\right) = \prod_{s=1}^{S} k_s^{\beta_s}\left(\mathbf{x}_i^{(s)}, \mathbf{x}_j^{(s)}, \boldsymbol{\theta}^{(s)}\right) \text{ with } \beta_s \geq 0 \ \forall \ s \tag{3.5}$$

It is worth noting that all the parameterised combination strategies have the *isolation property* (Rao 2001, Rao 2004) as they contain the case where only one of the base kernels can be selected.

3.2.6 Theoretical Justification of Kernel Combinations

As discussed in Chapter 2, the main property characterising a kernel is that it is a symmetric and positive semi-definite matrix, i.e all of its eigenvalues are non-negative. Given a symmetric matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ and its eigenvalue decomposition $\mathbf{Ku} = \boldsymbol{\lambda}\mathbf{u}$, then this matrix is a valid kernel if $\boldsymbol{\lambda}_n \geq 0 \ \forall \ n \in \{1, \ldots, N\}$.

Constructing a valid composite kernel relies on simple operations, or closure properties (Shawe-Taylor and Cristianini 2004), on the base kernels that lead to a symmetric positive semi-definite matrix. The closure properties utilised for the proposed combination strategies, given base kernels k_1, k_2 and input vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^(D)$, are:

$$\text{(i)} \ \ k(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{x}_i, \mathbf{x}_j) + k_2(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{(ii)} \ \ k(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{x}_i, \mathbf{x}_j) k_2(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{(iii)} \ \ k(\mathbf{x}_i, \mathbf{x}_j) = \alpha k_1(\mathbf{x}_i, \mathbf{x}_j)$$

where $\alpha \in \mathbb{R}$ is a scalar. The Fixed, Convex Linear and Binary combination rules are justified on the basis of properties (i) and (iii), while the Product and weighted Product rules follow properties (ii) and (iii). All the rules result in a symmetric, positive semi-definite composite kernel matrix with corresponding proofs given in Shawe-Taylor and Cristianini (2004).

3.3 Multinomial Probit Kernel Regression

Having described the composite kernel construction rules, this section progresses to the proposed kernel regression model that employes the multinomial probit likelihood and appropriate prior distributions. Emphasis is placed on the choice

and justification of priors as well as the characteristics of the likelihood and the role of the auxiliary variables that will be introduced into the model.

3.3.1 Multinomial Probit Likelihood

Consider a multinomial classification problem with an associated multi-feature training set $\{\mathbf{X}^{(s)}, \mathbf{t}\}$ where $\mathbf{X}^{(s)} \in \mathbb{R}^{N \times D_s}$, $s \in \{1, \ldots, S\}$, $t_n \in \{1, \ldots, C\}$ and S, N, D, C the $\mathbb{N}$ total number of sources, samples, features and classes respectively. Embedding each feature space via the *kernel trick* into a base kernel, assuming a specific combination rule and conditioning on specific kernel parameters $\boldsymbol{\beta}, \boldsymbol{\Theta}$ for clarity, leads to the construction of the composite kernel $\mathbf{K} \in \mathbb{R}^{N \times N}$.

The composite kernel allows informative integration of multiple sources via the inferred kernel parameters and provides a high-dimensional, possibly non-linear, embedding of the original features which can transform the problem to be linearly separable via hyper-planes.

The interest lies in modelling $p(\mathbf{t}|\mathbf{K})$ which, assuming a generalised linear model structure with parameters $\mathbf{W} \in \mathbb{R}^{N \times C}$, can be expressed as:

$$p(\mathbf{t}|\mathbf{K}) = \int p(\mathbf{t}|\mathbf{K}, \mathbf{W}) p(\mathbf{W}|\mathbf{t}, \mathbf{K}) d\mathbf{W} \tag{3.6}$$

where the first term in the integral is the likelihood of the model and the second term the parameter posterior distribution.

Following the approach by Albert and Chib (1993) and introducing auxiliary variables $\mathbf{Y} \in \mathbb{R}^{N \times C}$ as regression targets results in the following reformulation:

$$p(\mathbf{t}|\mathbf{K}) = \iint p(\mathbf{t}|\mathbf{Y}) p(\mathbf{Y}|\mathbf{K}, \mathbf{W}) p(\mathbf{W}|\mathbf{K}) d\mathbf{W} d\mathbf{Y} \tag{3.7}$$

The introduction of the auxiliary variables leads to a closed form posterior for the regression coefficients, as we now have a regression on $\mathbf{Y}$, and to an efficient Gibbs sampling scheme as we will see later in this Chapter. The regression on the auxiliary variables employs a standardised noise model $\mathbf{y}_c \sim \mathbf{K}\mathbf{w}_c + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ by definition[2] which results in the following products of univariate Gaussian distributions for the prior on auxiliary variables:

[2]The standardised noise model is not an assumption on noise but gives rise to the multinomial probit likelihood.

$$p\left(\mathbf{Y}|\mathbf{K},\mathbf{W}\right)=\prod_{n=1}^{N}\prod_{c=1}^{C}\mathcal{N}_{y_{nc}}\left(\mathbf{w}_c^{\mathsf{T}}\mathbf{k}_n,\mathbf{1}\right) \tag{3.8}$$

The probit link from the regression (continuous) target y_{nc} to the discrete target variable of interest $t_n \in \{1,\ldots,C\}$ is given by:

$$t_n=i \iff y_{ni}>y_{nj} \;\; \forall\, j\neq i \;\; \text{with} \;\; i,j\in\{1,\ldots,C\} \tag{3.9}$$

and hence we can express $p(\mathbf{t}|\mathbf{Y})$ as delta functions:

$$P(t_n=i|\mathbf{Y})=\delta^{t_n}=\begin{cases}1 & \iff y_{ni}>y_{nj}\;\forall\, j\neq i\\ 0 & \text{otherwise}\end{cases} \tag{3.10}$$

It is worth noting that the uncertainty on the prediction is not expressed in the link function $p(\mathbf{t}|\mathbf{Y})$ which *given* auxiliary variables for sample n simply assigns a specific class. The uncertainty is described in the auxiliary variable and regression coefficient posterior distributions which are marginalised out in Equation 3.7. Disregarding for now the regression coefficient posterior, and considering the auxiliary variable prior in Equation 3.8 we are led to the final expression, in an analogous manner to Girolami and Rogers (2006), for the multinomial probit likelihood in a GLM setting:

$$\begin{aligned}
&P\left(t_n=i|\mathbf{W},\mathbf{k}_n\right)\\
&= \int P\left(t_n=i|\mathbf{y}_n\right)p\left(\mathbf{y}_n|\mathbf{k}_n,\mathbf{W}\right)d\mathbf{y}_n\\
&= \int \delta^{t_n}\prod_{c=1}^{C}\mathcal{N}_{y_{nc}}\left(\mathbf{w}_c^{\mathsf{T}}\mathbf{k}_n,1\right)d\mathbf{y}_n\\
&= \int_{-\infty}^{+\infty}\int_{-\infty}^{y_{ni}}\mathcal{N}_{y_{ni}}\left(\mathbf{w}_i^{\mathsf{T}}\mathbf{k}_n,1\right)\prod_{j\neq i}^{C}\mathcal{N}_{y_{nj}}\left(\mathbf{w}_j^{\mathsf{T}}\mathbf{k}_n,1\right)dy_{nj}dy_{ni}\\
&\quad \text{setting } u=y_{ni}-\mathbf{w}_i^{\mathsf{T}}\mathbf{k}_n \text{ leads to}\\
&= \int_{-\infty}^{+\infty}\mathcal{N}_u(0,1)\prod_{j\neq i}^{C}\int_{-\infty}^{u+\mathbf{w}_i^{\mathsf{T}}\mathbf{k}_n-\mathbf{w}_j^{\mathsf{T}}\mathbf{k}_n}\mathcal{N}_{y_{nj}}(0,1)dy_{nj}du\\
&= \mathbb{E}_{p(u)}\left\{\prod_{j\neq i}\Phi\left(u+\left(\mathbf{w}_i-\mathbf{w}_j\right)^{\mathsf{T}}\mathbf{k}_n\right)\right\}
\end{aligned} \tag{3.11}$$

where $\mathbb{E}_{p(u)}$ is the expectation taken with respect to the standardised normal

distribution $p(u) = \mathcal{N}(0,1)$ and Φ is the cumulative density function.

The multinomial probit likelihood is conditioned on the regression coefficients $\mathbf{W}$ and it will be employed accordingly to each specific inference scheme considered in this thesis. In this Chapter "fully" Bayesian sampling approaches will be proposed that make use of the whole regression coefficient posterior distribution via Monte Carlo estimates. As such, the uncertainty over these coefficients which is expressed by their posterior distribution is taken into account.

We now have an explicit multiclass likelihood for multiple kernel learning that, as we shall see in this Chapter, gives rise to an efficient Gibbs sampling inference scheme and in later Chapters allows for further deterministic approximations that reduce computational complexity and memory requirements. The likelihood is in accordance with the motivation of this thesis for probabilistic and multiclass multiple kernel learning.

3.3.2 Gauss-Hermite Quadrature

The Multinomial probit likelihood in Equation 3.11 can be re-expressed as:

$$\begin{aligned} P(t_n = i|\mathbf{W}, \mathbf{k}_n) &= \mathbb{E}_{p(u)}\left\{\prod_{j\neq i}\Phi\left(u + (\mathbf{w}_i - \mathbf{w}_j)^{\mathsf{T}}\mathbf{k}_n\right)\right\} \\ &= \mathbb{E}_{p(u)}\{F(u)\} = \int F(u)\mathcal{N}_u(0,1)du \\ &= \frac{1}{\sqrt{2\pi}}\int F(u)e^{-u^2}du \end{aligned} \tag{3.12}$$

which directly leads to the standard Gauss-Hermite quadrature approximation with weights $\mathcal{W}(u) = e^{-u^2}$. Hence such an approximation offers an alternative to the Monte Carlo estimate of the expectation that is a computationally expensive sampling procedure.

3.3.3 Prior distributions and the graphical model

Having defined the kernel combination rules and derived the multinomial probit likelihood, the next step is to consider hierarchical prior distributions on the model parameters. The hierarchical approach adopted in this thesis places *hyper-prior* distributions on the prior distributions to propagate model uncertainty in

a higher and even less "subjective" manner. The justification for placing specific prior distributions on unknown model parameters will be based on one or more of the following reasons and conditions:

- **Prior Knowledge** - Prior information exists that dictates the numerical nature or scale of a parameter (i.e. the parameter is defined in $\mathbb{R}_+$), imposes a constraint (i.e. $\sum_{s=1}^{S} \beta_s = 1$) or specifies some expected range.
- **Conjugacy** - Conjugate pairs of distributions lead to closed form posteriors and hence are preferred (Gelman et al. 2004, Denison et al. 2002) when no other prior knowledge is available.
- **Sparsity** - When sparse solutions are encouraged based on either prior knowledge or desired outcome, appropriate prior settings and distributions can be employed to induce such sparsity. Such sparsity inducing prior formulations are adopted in Chapter 5.

Before going into the details of the specific prior distributions, the full graphical model is given in Figure 3.1 with variations for all the kernel combination rules. The plates diagram depicts the conditional dependancies and the dimensionality of model parameters that will be now introduced and justified.

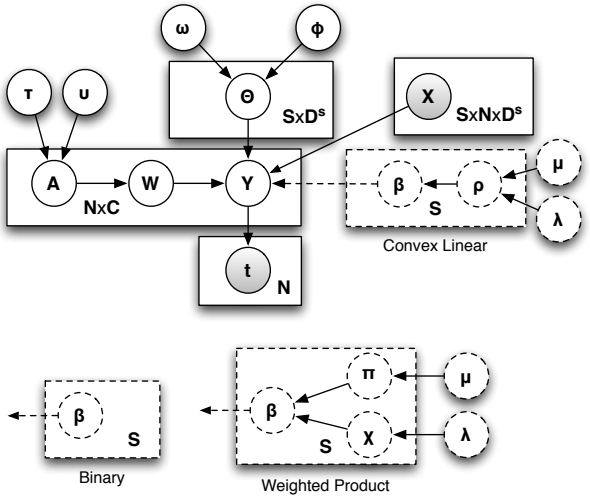

Figure 3.1: Plates diagram of the model depicting the conditional relationships of model variables together with the dimensionality of corresponding plates. The dotted plates depict variations for the three parametric combination rules.

Regression coefficients W and scales A

The regression coefficients $\mathbf{W}$ are defined typically (Hastie et al. 2001, Denison et al. 2002) as a product of zero-mean Gaussians with scales $\mathbf{A} \in \mathbb{R}_+^{N \times C}$ and a hyper-prior Gamma density (ensuring positive values) is placed on each scale α_{nc} (inverse of variance) with hyper-parameters τ and υ. This reflects the prior independence of the regression coefficients (product) and our *lack* of prior knowledge (zero mean normally distributed) regarding the sign of the regression coefficients. Such a hierarchical prior formulation induces sparsity (zero mean) and allows the scales of the distributions to be easily inferred from the data. As the prior distribution of the regression coefficients is only parameterised via the scale, which is described via the Gamma density, the only free-parameters to be defined at this stage are τ and υ which will be discussed further on.

The corresponding distributions are:

$$p(\mathbf{W}|\mathbf{A}) = \prod_{n,c=1}^{N,C} \mathcal{N}_{w_{nc}}\left(0, \alpha_{nc}^{-1}\right) \tag{3.13}$$

$$p(\mathbf{A}|\upsilon,\tau) = \prod_{n,c=1}^{N,C} \mathcal{G}_{\alpha_{nc}}(\upsilon,\tau) \tag{3.14}$$

where $\mathcal{N}$ and $\mathcal{G}$ are the Gaussian and Gamma distribution respectively as defined in the notation.

Kernel combination parameters $\boldsymbol{\beta}$

Only the parameterised combinatorial rules need to be considered as both the *Fixed* combination and the *Product* combination can be seen as specific sub-cases of the *Convex linear* and *Weighted product* combination rules under conditioning on specific values ($\beta_s = \frac{1}{S}$ and $\beta_s = 1$ respectively).

a) In the case of the *Convex linear* rule, the prior distribution on the combinatorial weights $\boldsymbol{\beta} \in \mathbb{R}^S$ is a Dirichlet distribution with parameters $\boldsymbol{\rho} \in \mathbb{R}^S$ and a hyper-prior Gamma distribution is placed on $\boldsymbol{\rho}$ with hyper-parameters μ and λ. The choice of the Dirichlet distribution, which confines our movement on a simplex, for the combinatorial weights $\boldsymbol{\beta}$ is based on the need to have a resulting positive semi-definite (p.s.d) composite kernel and weak statistical identifiability. These constraints, expressed via the Dirichlet prior, are:

$$\begin{cases} \beta_s \geq 0 \ \ \forall \ \ s \in \{1, \ldots, S\} \\ \sum_{s=1}^{S} \beta_s = 1 \end{cases}$$

Previous work (Girolami and Rogers 2005, Girolami and Rogers 2006) has shown that these conditions are necessary for avoiding unconstrained growth or reduction of the model parameters due to inherent coupling. The prior Gamma distribution in this case is justified on the restriction that $\boldsymbol{\rho}$ is defined on $\mathbb{R}_+^S$. The corresponding distributions are:

$$p\left(\boldsymbol{\beta}|\boldsymbol{\rho}\right) = \mathcal{D}_{\boldsymbol{\beta}}(\boldsymbol{\rho}) \tag{3.15}$$

$$p\left(\boldsymbol{\rho}|\lambda, \mu\right) = \prod_{s=1}^{S} \mathcal{G}_{\rho}(\lambda, \mu) \tag{3.16}$$

b) The *Weighted product* combination rule is described via the bottom right dashed plate in Figure 3.1 and places a Gamma distribution on the combinatorial weights $\boldsymbol{\beta}$ ensuring a p.s.d composite kernel and an exponential hyper-prior distribution on each of the prior parameters $\boldsymbol{\pi}, \boldsymbol{\chi}$ that are defined in $\mathbb{R}_+^S$.:

$$p\left(\boldsymbol{\beta}|\boldsymbol{\pi}, \boldsymbol{\chi}\right) = \prod_{s=1}^{S} \mathcal{G}_{\beta_s}(\pi_s, \chi_s) \tag{3.17}$$

$$p\left(\boldsymbol{\pi}|\mu\right) = \prod_{s=1}^{S} \mathcal{E}_{\pi_s}(\mu) \tag{3.18}$$

$$p\left(\boldsymbol{\chi}|\lambda\right) = \prod_{s=1}^{S} \mathcal{E}_{\chi_s}(\lambda) \tag{3.19}$$

c) Finally, in the *Binary* combination rule we employ the left dashed plate in Fig. 3.1 which places a binomial distribution on each β_s with equal probability of being 1 or zero (unless prior knowledge dictates otherwise). The small size of the possible 2^S states of the $\boldsymbol{\beta}$ vector allows for their explicit consideration in the inference procedure and hence there is no need to place any hyper-prior distributions:

$$p\left(\boldsymbol{\beta}\right) = \prod_{s=1}^{S} \mathcal{B}_{\beta_s}(n, p) \ \ \text{with} \ \ p = 0.5 \tag{3.20}$$

Base kernel parameters $\boldsymbol{\theta}^{(s)}$

The proposed model allows for inference on the base kernel parameters $\boldsymbol{\theta}$ by placing a Gamma distribution with parameters ω and ϕ. The kernel parameters are specific to each base kernel that describe a source of information, e.g the variances for the case of Gaussian kernels, and control the amount of smoothness level applied within each feature space. The specific prior distribution employed here reflects again the restriction in $\mathbb{R}_+$.

$$p\left(\boldsymbol{\Theta}|\phi,\omega\right) = \prod_{s,d=1}^{S,D} \mathcal{G}_{\theta_{sd}}(\omega,\phi) \tag{3.21}$$

Hyper-parameters

The *hyper-parameters* $\tau, \upsilon, \omega, \phi, \mu, \lambda$ can be set to uninformative values (Girolami and Rogers 2006) or can be inferred via the empirical Bayes approach of type-II maximum likelihood (Tipping 2004). In the specific case of τ, υ that dictate the prior form of the gamma distribution on the scales of the regression coefficients, we can induce sparsity via setting a flat prior ($\tau, \upsilon \to 0$) as in (Damoulas et al. 2008). This leads to the construction and generalisation of relevant vector machines (Tipping 1999) to the multiclass multiple kernel learning setting.

3.4 Markov Chain Monte Carlo Posterior Inference

We proceed by presenting Markov chain Monte Carlo (MCMC) sampling methods for the models that achieve exact posterior inference to the limit of infinite drawn samples. The introduction of the auxiliary variables $\mathbf{Y}$ allows a straightforward Gibbs sampling scheme which is compared against a standard Metropolis-Hastings method (Metropolis et al. 1953, Hastings 1970) for sampling characteristics and efficiency.

3.4.1 Gibbs Sampler

As we have seen in the previous chapter, Gibbs sampling exploits the re-expression of a desired joint posterior distribution to individual conditional posterior dis-

tributions from which is easy to draw samples. The benefits of this sampling scheme are ease of implementation and avoidance of acceptance ratio tuning when compared with the standard MH sampling methods (Gelman et al. 2004). Considering now the posterior distributions of the regression coefficients and auxiliary variables will further highlight the natural Gibbs sampler that arises.

Regressors

A closed form expression for the posterior of the regression coefficients $p(\mathbf{W}|\mathbf{Y},\mathbf{K},\mathbf{A}) = p(\mathbf{Y}|\mathbf{W},\mathbf{K})p(\mathbf{W}|\mathbf{A})/p(\mathbf{Y}|\mathbf{K},\mathbf{A})$ is readily available[3], see (Denison et al. 2002), as:

$$p(\mathbf{W}|\mathbf{Y},\mathbf{K},\mathbf{A}) = \prod_{c=1}^{C} \mathcal{N}(\mathbf{m_c},\mathbf{V}_c) \tag{3.22}$$

with $\mathbf{m}_c = \mathbf{V}_c(\mathbf{K}\mathbf{y}_c)$, $\mathbf{V}_c^{-1} = \mathbf{K}\mathbf{K}^{\mathsf{T}} + \mathbf{A}_c$ and $\mathbf{A}_c$ a diagonal matrix of the scales $\boldsymbol{\alpha}_c$ as depicted below:

$$\mathbf{A}_c = \begin{bmatrix} \alpha_{1c} & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & \dots & 0 \\ 0 & 0 & \alpha_{nc} & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & \dots & 0 & \alpha_{Nc} \end{bmatrix}$$

Auxiliary variables & hyper-parameters

Further on, a closed form expression (Girolami and Rogers 2006) for the posterior of the auxiliary variables $p(\mathbf{Y}|\mathbf{W},\mathbf{K},\mathbf{t}) = p(\mathbf{t}|\mathbf{Y})p(\mathbf{Y}|\mathbf{W},\mathbf{K})/p(\mathbf{t}|\mathbf{K},\mathbf{W})$ can be derived as a product of N, $C-$dimensional conically truncated Gaussians:

$$p(\mathbf{Y}|\mathbf{W},\mathbf{K},\mathbf{t}) = \prod_{n=1}^{N} \mathcal{N}_{\mathbf{y}_n}^{t_n}(\mathbf{W}^{\mathsf{T}}\mathbf{k}_n,\mathbf{I}) \tag{3.23}$$

where t_n indicates the dimension of truncation. Rejection sampling can be used to sample from truncated distributions as usual.

The conditional distributions in Equations 3.22 and 3.23 give rise to Gibbs sampling which is completed by a standard closed form expression for the poste-

[3]Dependence on $\boldsymbol{\beta}, \boldsymbol{\Theta}$ omitted for clarity

rior distribution of the scales $\mathbf{A}$, which is of the same form as the prior (Gamma) with updated hyper-parameters:

$$p(\mathbf{A}|\mathbf{W},\tau,\upsilon) = \prod_{n,c=1}^{N,C} \mathcal{G}_{\alpha_{nc}}(\tau+\frac{1}{2},\upsilon+\frac{1}{2}w_{nc}^2) \tag{3.24}$$

Finally, the kernel parameters $\boldsymbol{\Theta}$, combination weights $\boldsymbol{\beta}$ and associated hyper-parameters $\boldsymbol{\rho},\boldsymbol{\pi},\boldsymbol{\chi}$ do not have a closed form conditional posterior and we resort to Metropolis-Hastings (MH) (Hastings 1970) sub-samplers which are described in detail in Appendix A.1 and A.2. As we have seen in the previous chapter, Gibbs sampling can be viewed as a special case of MH sampling methods and hence it is natural to introduce MH steps into the Gibbs sampler. Furthermore, few steps (even a single one) of the sub-samplers for every Gibbs iteration are enough to lead to overall convergence to the stationary distribution (Neal 2003).

Predictive distribution

Having described posterior inference of model parameters via Gibbs sampling, we return to the original task of predicting the class label t_* of a new point $\mathbf{x}_*^{(s)}$ that is embedded in a composite kernel space $\mathbf{k}_*$. The predictive distribution is given by:

$$P\left(t_*=i|\mathbf{k}_*,\mathbf{K},\mathbf{t}\right) = \int P\left(t_*=i|\mathbf{W},\boldsymbol{\beta},\boldsymbol{\Theta},\mathbf{k}_*\right) P\left(\mathbf{W},\boldsymbol{\beta},\boldsymbol{\Theta}|\mathbf{K},\mathbf{t}\right) d\mathbf{W}\ d\boldsymbol{\beta}\ d\boldsymbol{\Theta} \tag{3.25}$$

where $\mathbf{k}_*$ is the composite test kernel created based on $\mathbf{x}_*^{(s)}$ and the inferred values for $\boldsymbol{\beta},\boldsymbol{\Theta}$. The training set $\{\mathbf{X}^{(s)},\mathbf{t}\}$ is used to infer the parameter posterior as described in the previous sections. The Monte Carlo estimate of the predictive distribution is used to assign the class probabilities according to L number of drawn samples[4]. Hence the estimated class probability is:

$$P\left(t_*=i|\mathbf{k}_*,\mathbf{K},\mathbf{t}\right) = \frac{1}{L}\sum_{l=1}^{L} \mathbb{E}_{p(u)}\left\{\prod_{j\neq i}\Phi\left(u+\left(\mathbf{w}_i^l-\mathbf{w}_j^l\right)^\intercal \mathbf{k}_*\right)\right\} \tag{3.26}$$

[4]In fact the number of samples equals the number of Gibbs steps minus some burn-in period. This is because in every Gibbs iteration we explore the posterior space of $\mathbf{W},\mathbf{Y},\boldsymbol{\Theta},\boldsymbol{\beta}$

and typically $1,000$ samples $u \sim \mathcal{N}_u(0,1)$ are enough to approximate the expectation or alternatively the Gauss-Hermite Quadrature from Equation 3.3.2 can be employed.

The pseudo-algorithm is given as:

Algorithm 1 Gibbs sampler

1: Initialise hyper-parameters $\tau, \upsilon, \omega, \phi, \mu, \lambda$
2: Sample parameters $\mathbf{A}, \boldsymbol{\rho}, \boldsymbol{\beta}$ from prior
3: Create train kernels
4: **for** Gibbs iterations **do**
5: Sample from regressor posterior, Equation 3.22
6: Sample from auxiliary variable posterior, Equation 3.23
7: Sample parameters $\mathbf{A}, \boldsymbol{\beta}, \boldsymbol{\Theta}$ and $\boldsymbol{\rho}, \boldsymbol{\pi}, \boldsymbol{\chi}$ with MH sub-samplers.
8: **end for**
9: Discard Burn-in period samples
10: Create test kernels $\mathbf{K}^*$ given $\boldsymbol{\beta}, \boldsymbol{\Theta}$ posteriors
11: Predict class from Equation 3.26

3.4.2 Metropolis Hastings Sampler

In order to assess the sampling efficiency and characteristics of the proposed Gibbs sampler, a standard Metropolis-Hastings (MH) sampling procedure can be readily derived for comparison. The introduction of auxiliary variables is no longer needed, as their sole role was to lead us to a Gibbs sampler via the posterior conditional distributions, and hence the model reduces to the graphical form depicted in Figure 3.2.

Having dispensed with the auxiliary variables, no closed form solution for the posterior distribution over the regression coefficients $\mathbf{W}$ can be offered and hence, we resort to standard MH sampling (Metropolis et al. 1953, Hastings 1970) with an appropriate acceptance ratio[5] given by

$$\text{Acceptance Ratio} = \min\left\{1, \frac{P(\mathbf{t}|\mathbf{W}^t, \mathbf{K})P(\mathbf{W}^t|\mathbf{A})Q(\mathbf{W}^{t-1}|\mathbf{W}^t)}{P(\mathbf{t}|\mathbf{W}^{t-1}, \mathbf{K})P(\mathbf{W}^{t-1}|\mathbf{A})Q(\mathbf{W}^t|\mathbf{W}^{t-1})}\right\} \quad (3.27)$$

where ^{t} symbolises the proposed move, $^{t-1}$ the current state and $Q(.,.)$ is the proposal or jumping distribution which in this scenario is typically a symmetric

[5] Again conditioning on parameters $\boldsymbol{\Theta}, \boldsymbol{\beta}$ for clarity.

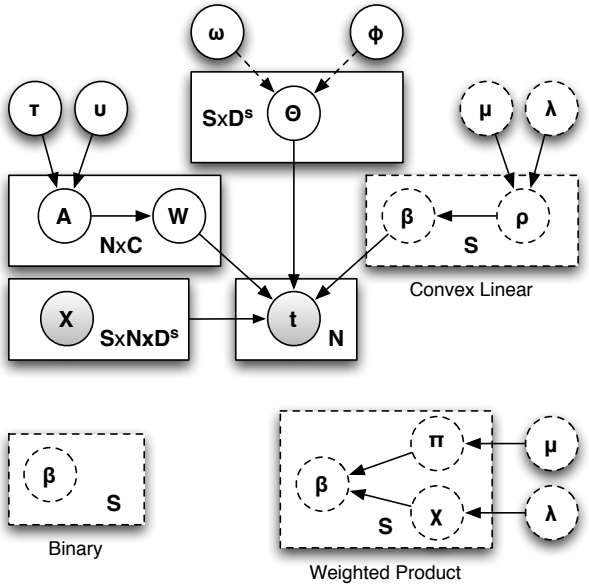

Figure 3.2: Plates diagram of the reduced model depicting the conditional relationships of model variables together with the dimensionality of corresponding plates. The dotted plates depict variations for the three parametric combination rules.

normal proposal and hence cancels out of the ratio leading to a Metropolis sampling scheme. As in the case with the Gibbs sampler, the kernel combination parameters and hyper-parameters are inferred via additional MH steps given in the Appendix A.1.

There are inherent drawbacks when adopting the Metropolis sampler instead of the Gibbs sampling scheme. First of all, the Metropolis scheme requires tuning of the jumping proposal in order to achieve the required proportion levels of sample acceptance ($20 - 30\%$ depending on dimensionality, see Gelman et al. (2004)). Although a large body of work has been devoted in the statistics community for designing adaptive and more efficient proposals, it introduces the need for additional computation and somewhat ad-hoc engineering procedures. Furthermore, the posterior samples from Metropolis or MH samplers are usually highly correlated due to the nature of the random walk and the proposal distribution. This, as we shall see, may lead to small effective sampling sizes and hence to the requirement for typically longer Markov chains. This would be necessary for achieving convergence to the stationary target distribution and for effective posterior inference.

Finally, as it can be seen from the acceptance ratio in Equation 3.27, the

Metropolis sampler is computationally more expensive *per sample* than the Gibbs scheme due to the required estimation of the likelihood $P(\mathbf{t}|\mathbf{W},\mathbf{K})$ for every proposed step. Recalling the definition of the multinomial likelihood in Equation 3.11, it is straightforward to understand the additional computational burden exerted by the required expectation.

The pseudo-algorithm is given below:

Algorithm 2 Metropolis sampler

1: Initialise hyper-parameters $\tau, \upsilon, \omega, \phi, \mu, \lambda$
2: Sample parameters $\mathbf{A}, \boldsymbol{\rho}, \boldsymbol{\beta}$ from prior
3: Create train kernels
4: **for** Metropolis iterations **do**
5: Sample regression coefficients with acceptance ratio in Equation 3.27
6: Sample parameters $\mathbf{A}, \boldsymbol{\beta}, \boldsymbol{\Theta}$ and associated hyper-parameters
7: **end for**
8: Discard Burn-in period samples
9: Create test kernels $\mathbf{K}^*$ given $\boldsymbol{\beta}, \boldsymbol{\Theta}$ posteriors
10: Predict class from Equation 3.26

3.5 Marginal Likelihood for Model Selection

The normalising constant of the posterior density plays an important role in Bayesian inference methods for model selection and the calculation of the so called "Bayes factors" (Berger 1985, Kass and Raftery 1995, Vyshemirsky and Girolami 2008). It is defined as the integral of the likelihood function with respect to the prior parameter density, hence its name marginal likelihood.

Bayes factors, defined as the ratio of the marginal likelihood under one model to the marginal likelihood of another model, are an elegant way to compare the fit of competing models associated with possibly different parameters that are marginalised out.

Due to this important role of the marginal likelihood a plethora of approaches for estimating it have been proposed in the literature, see (Vyshemirsky 2007) for a recent review. In this work we adopt the commonly used method developed by Chib (1995) which is efficient and easy to implement for the proposed Gibbs sampler and the multinomial probit model.

Chib's method, for parameters $\boldsymbol{\theta}$ and evidence or data $\mathbf{y}$, makes use of Bayes theorem and the resulting *basic marginal likelihood identity* (BMI) given by:

$$m(\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|\mathbf{y})} \tag{3.28}$$

where m the marginal likelihood, f the likelihood function, $\pi(\boldsymbol{\theta})$ the prior and $\pi(\boldsymbol{\theta}|\mathbf{y})$ the posterior parameter density.

Estimation of the marginal likelihood proceeds by selecting an appropriate high density point $\boldsymbol{\theta}^*$ (e.g. the mode or mean of the posterior Gibbs output) and employing the Naperian logarithm of the BMI identity to give:

$$\log \hat{m}(\mathbf{y}) = \log f(\mathbf{y}|\boldsymbol{\theta}^*) + \log \pi(\boldsymbol{\theta}^*) - \log \hat{\pi}(\boldsymbol{\theta}^*|\mathbf{y}) \tag{3.29}$$

Another benefit of the marginal likelihood estimation via the BMI identity is its use for assessing convergence by monitoring its stability during the Gibbs sampling (Chib 1995).

3.6 Comparison of MCMC Sampling Schemes

Having introduced the model and posterior inference via competing MCMC sampling methods, a study of their efficiency and sampling characteristics is offered on a multinomial classification dataset first introduced by Neal (1996). Following Holmes and Held (2006) an examination of the *effective sampling size* (ESS), *autocorrelation*, *average sample distance* and classification accuracy is performed. Multiple chains are randomly initialised for each case and convergence is monitored via the BMI progression and standard $\hat{R}$ values (Gelman et al. 2004).

Neal's dataset consists of objects $\mathbf{x}_n \in \mathbb{R}^4$ sampled from three 4-dimensional classes $t_n \in \{1, \ldots, C\}$ that form overlapping ellipses according to:

$$0.5 \geq x_{n,1}^2 + x_{n,2}^2 > 0.1 \quad \text{for } t_n = 1 \tag{3.30}$$

$$1.0 \geq x_{n,1}^2 + x_{n,2}^2 \geq 0.6 \quad \text{for } t_n = 2 \tag{3.31}$$

$$(x_{n,1}, x_{n,2})^\mathsf{T} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \quad \text{for } t_n = 3 \tag{3.32}$$

with two extra dimensions $x_{n,3}, x_{n,4}$ as added noise. The informative dimensions can be seen in Figure 3.3.

In order to compare the sampling characteristics of the two MCMC schemes, 100 objects from each class are used for training and 200 for testing. The

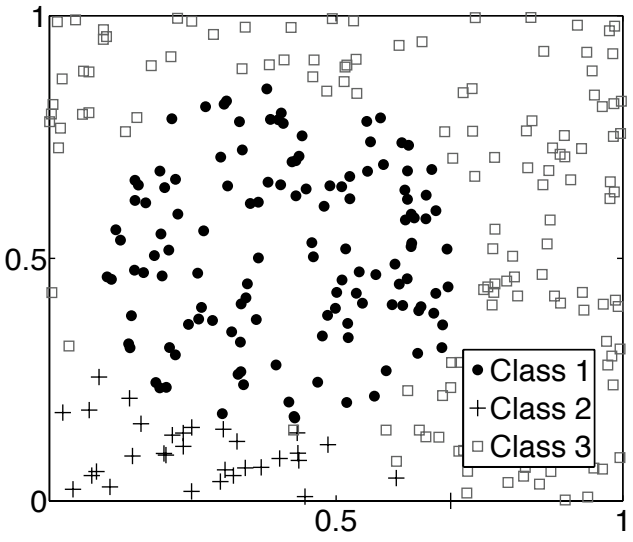

Figure 3.3: The artificial dataset.

Gibbs sampler is run for 20,000 samples with a burn-in period of 5,000 and the Metropolis for 100,000 samples with 20,000 burn-in period following recommended settings from Gelman et al. (2004). The Metropolis employs an adaptive proposal with a step based on the acceptance ratio to improve efficiency and the post burn-in samples are thinned retaining samples every 10 steps. Experiments are repeated over 10 randomly initialised runs and the comparison is performed across three main properties, following Holmes and Held (2006):

i) *Autocorrelation* - $\rho(k)$ monotone sample autocorrelations of lag k by the initial monotone sequence estimator proposed by Geyer (1995).

ii) *Distance* - Dist. $= \frac{1}{N-1}\sum_{i=1}^{N-1} \|\mathbf{W}^{(i)} - \mathbf{W}^{(i+1)}\|$ is the average Euclidean update distance between iterations.

iii) *Effective Sampling Size* - ESS $= L/\left(1 + 2\sum_{j=1}^{k} \rho(k)\right)$ is the effective sample size for a single coefficient w_{nc} while summing over the k monotone sample autocorrelations over a Markov chain of length L.

In Table 3.1 averaged results are reported. Both sampling schemes achieve a comparable classification performance of $2\% \pm 0.5$ error rate (average percentage of misclassified samples) on the specific dataset but the Gibbs approach is

shown to be superior in sampling characteristics. With an average ESS (over both covariates and runs) close to the full length of the post-burn chain and an average distance three orders of magnitude greater than the one from Metropolis sampling, it is clearly a more efficient sampling scheme.

MCMC Method	Dist.	ESS Lag 20	ESS Lag 2
Gibbs	$4,296.5 \pm 5.5$	$14,473 \pm 21.46$	$14,757 \pm 9.9182$
Metropolis	2.59 ± 0.87	$2,053 \pm 0.17$	$26,668 \pm 0.1497$

Table 3.1: Comparison of Gibbs versus Metropolis sampling through sampling Distance (mean $\pm$ std) and Effective Sampling Size (mean $\pm$ std).

In Figures 3.4 and 3.5, the typical autocorrelation progression for different lags is presented for both sampling schemes. As it can be seen, Gibbs sampling offers highly de-correlated samples from the posterior and hence smaller Markov chains are needed for convergence and a large ESS. In contrast, the Metropolis algorithm offers samples with high correlation due to the small local steps of the proposal distribution. Hence, it requires a large Markov chain and offers a small ESS and average Distance. Thinning the output by retaining samples periodically is the obvious solution which was adopted but still does not alleviate the need for a large Markov chain.

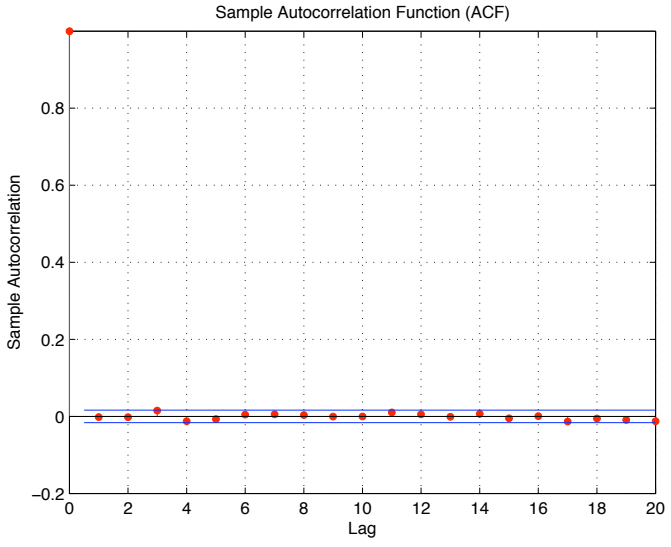

Figure 3.4: Typical Autocorrelation from the Gibbs sampler.

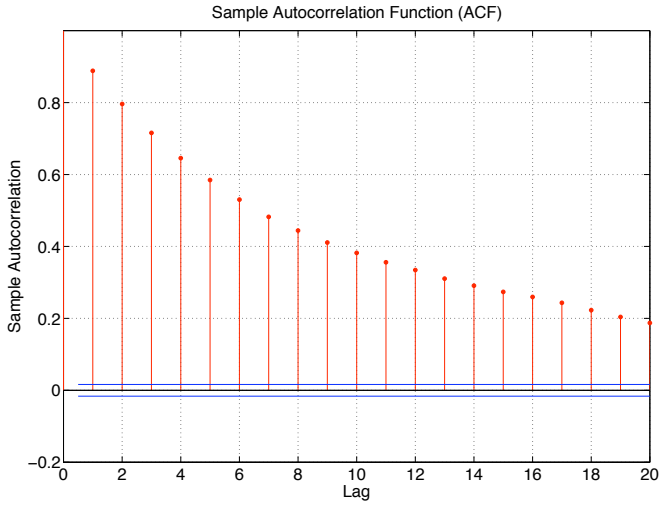

Figure 3.5: Typical Autocorrelation from the Metropolis sampler.

3.7 Toy Example Demonstration

In this section a brief demonstration for inference of kernel parameters $\boldsymbol{\Theta}$ and kernel combination parameters $\boldsymbol{\beta}$ is offered on the aforementioned Neal dataset (Neal 1996). The focus is on the convex linear combination rule as it is the standard MKL approach and the dataset is augmented by considering two additional noise kernels, with varying informational content. Inference is performed via the MH sub-samplers for $\boldsymbol{\Theta}$ and $\boldsymbol{\beta}$, learning both the importance of the individual attributes and the contribution of the kernels, the latter shown in Figure 3.6.

Results for the case of the convex linear combination rule with two noise kernels and one informative, and the sub-cases of conditioning on either only the prior variance of the regression coefficients, by setting all $\alpha_{nc}^{-1} = 1$, or on both $\mathbf{A}$ and the kernel parameters $\boldsymbol{\Theta}$, are depicted in Figure 3.7 and show the increase of the Markov chain needed for convergence as the parameter space expands.

As we have described, the model also allows one to infer knowledge on the importance of specific features (Automatic Relevance Determination (Neal 1996)) by learning the kernel parameters $\boldsymbol{\Theta}$. In Figure 3.8 the parameter values of the informative base kernel are depicted and as it is demonstrated the model correctly disregards the two uninformative features (the corresponding x_3 and x_4 dimension) and identifies x_1 and x_2 as the most important dimensions, depicted in Figure 3.3. Hence now, the composite kernel is successfully learned on both

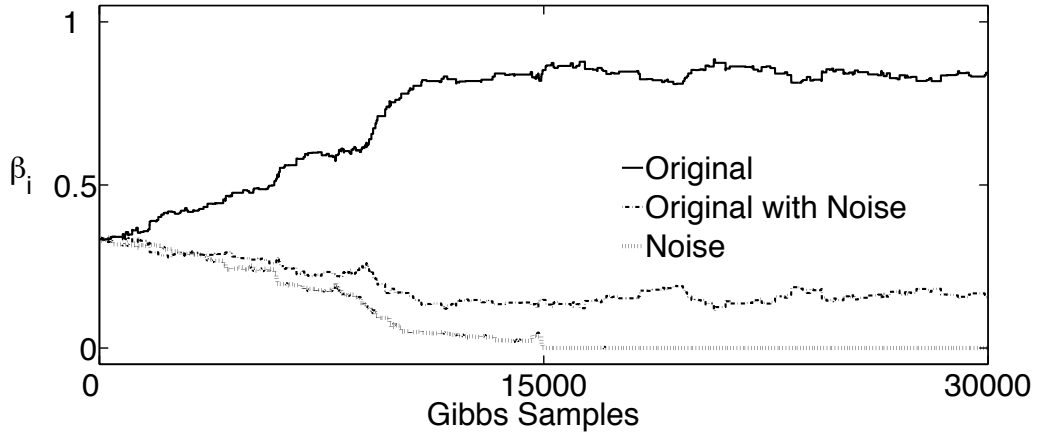

Figure 3.6: Three combined sources with varying informational content. Notice how the the original informative kernel receives 80% of the weight, with the partially informative kernel receiving the rest 20% and the non-informative kernel being effectively discarded.

levels, identifying the important base kernel and also the important attributes.

3.8 Computational Complexity

Markov chain Monte Carlo methods have a high computational cost due to the sampling nature of the methods. The Gibbs sampler introduced in this paper has a $\mathcal{O}(LCN^3)$ complexity, where L the Markov chain length, and with the dominating term N^3 arising from the typical matrix inversions in kernel settings. However, we can tackle both the sampling and the inversion training restrictions as efficient approximations such as MAP estimators, EM update schemes and variational treatments (Damoulas and Girolami 2008) can be derived from this framework and also sparse solutions such as RVMs (Tipping 1999) can be used that lead to reduced rank matrix inversions.

3.9 Discussion

In this chapter, a novel framework for probabilistic multiple kernel learning is presented. Different kernel combination rules are described from the well-known convex linear combination to novel product and binary rules, and are theoretically justified. The multinomial probit likelihood is proposed within the context

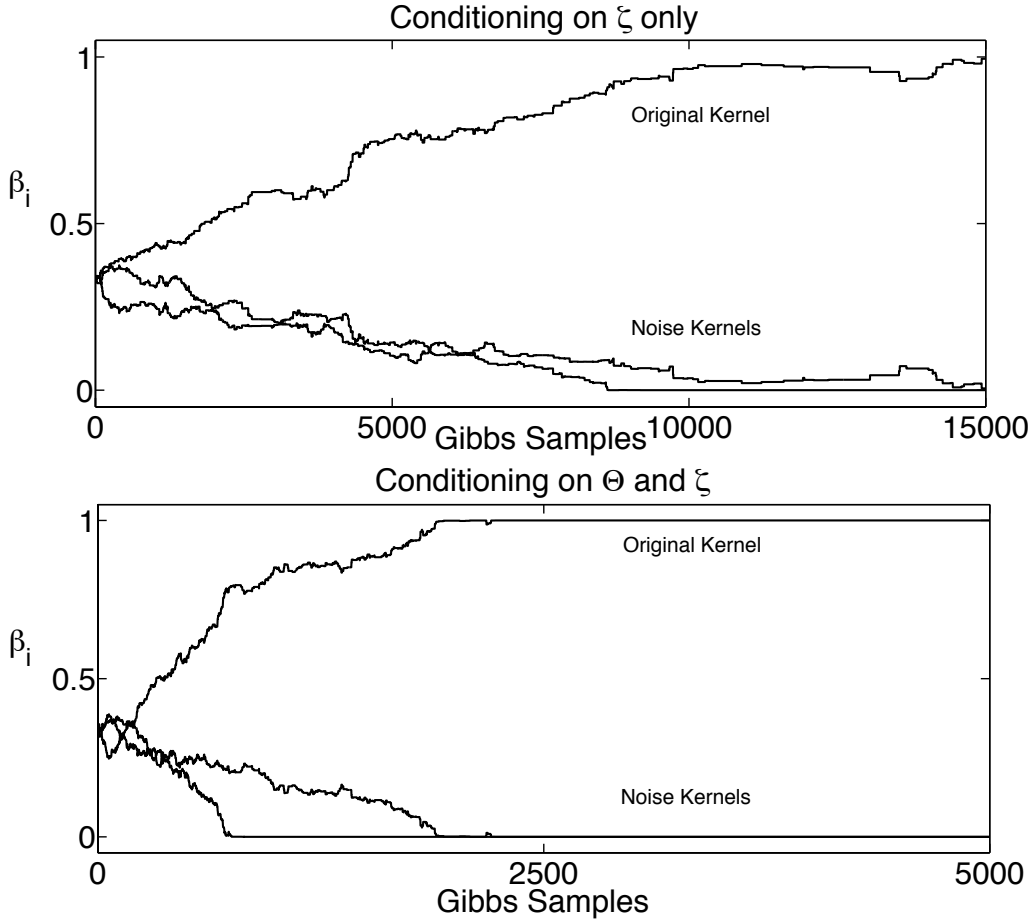

Figure 3.7: The effect of conditioning on the Neal dataset. As the parameter space expands, the required steps of the Gibbs sampler for convergence increase.

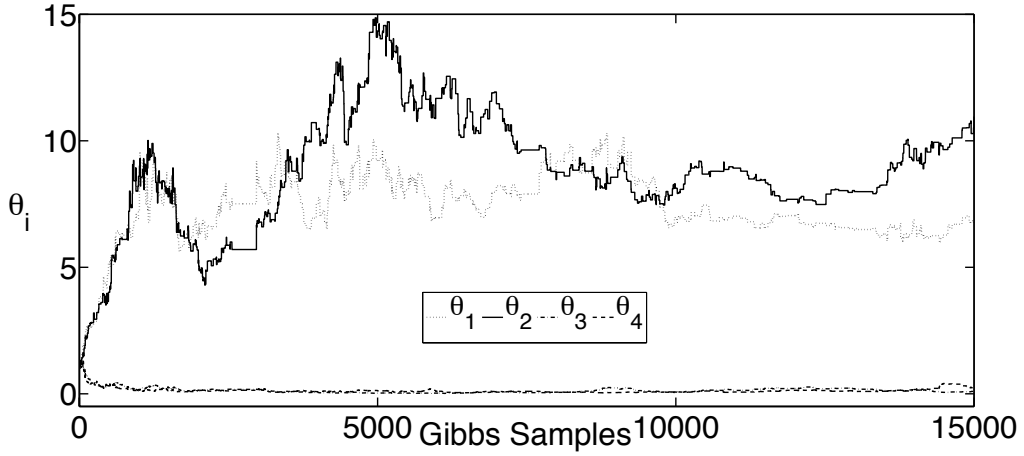

Figure 3.8: Inferring $\boldsymbol{\theta}_i$ and hence learning the importance of the features. The uninformative features, as it can be seen, receive a very low weight and are effectively discarded.

of Generalised Linear Models as it is an explicit multiclass likelihood giving rise to probabilistic classification assignments and efficient inference schemes. Two such Markov chain Monte Carlo (MCMC) solutions are derived and compared for sampling efficiency and convergence characteristics.

The proposed Gibbs sampling scheme is shown to outperform the Metropolis solution in all sampling efficiency measures and it will form the basis and adopted MCMC solution for the remainder of this thesis. Further sampling advancements and alternative schemes such as the adoption of a collapsed Gibbs sampling method (Liu 1994) are considered outside the scope of this thesis as the emphasis is now placed on proposing less computationally intensive solutions. The computational complexity of MCMC and kernel-based models is typically very high and hence deterministic approximations, such as variational Bayes and maximum-a-posteriori treatments, in addition to sparsity-inducing solutions will be the focus of the following Chapters.

Chapter 4

Variational Bayes Inference

The Markov chain Monte Carlo sampling schemes introduced in the previous chapter are resource intensive as evidenced by their dominant computational complexity $\mathcal{O}\left(LCN^3\right)$ and memory requirements $\mathcal{O}\left(SN^2 + 3NC\right)$, where S, L, C, N are the number of sources, drawn posterior samples (length of Markov chain), classes and samples (objects) respectively. As we have seen, the typical Markov chain length required for convergence in this application varies from some thousands, in the case of the Gibbs sampler, to tens of thousands, for the Metropolis scheme. Hence, considering that for each block sample drawn an inversion costing $\mathcal{O}(N^3)$ is performed, the need for approximate inference is apparent in order to scale up the proposed methodology for probabilistic multiple kernel learning.

In this chapter[1] the first step towards efficient and less expensive methodology is offered through a powerful deterministic approximation that belongs to the class of *Mean Field* methods (Parisi 1988, Opper and Winther 2001) and the sub-class of *variational Bayes* approaches (Jaakkola 2001, Ghahramani and Beal 2001, Beal 2003, MacKay 2003). First a brief examination of the main points of Mean Field methodology and specifically the "naive" variational mean field approach for factored ensembles of approximate posteriors is offered together with the intuition behind them. Then such an approximation is derived for the probabilistic multiple kernel learning model and it is assessed in comparison with the full MCMC solution previously introduced, with respect to the resulting approximate posterior distributions and computing times. Finally, we

[1] Parts of this work have already appeared in (Damoulas and Girolami 2008, Damoulas and Girolami 2009a)

conclude with an illustrative experimental section on multiclass UCI datasets comparing against published results from standard machine learning methods.

4.1 Mean Field Theory

The Mean Field (MF) theory, also known as *self consistent field theory* in physics, dates back to statistical mechanics work as early as 1935 with the Bethe approximation (Bethe 1935), that can be seen as an exact mean field theory on a tree, and also on later fundamental work by Thouless et al. (1977) that provided the TAP mean field equations for describing *spin glass* models in physics. The main characteristic of the MF approximation is that interactions and mutual influence between random variables is approximated by an effective "field" that acts independently on each of the individual random variables. This assumption provides us with convenient inference procedures that factorise the joint posterior distribution of interest to an ensemble of independent (but weakly coupled) approximate ones.

There are three main methodological streams of MF theory with distinct characteristics and applicability (Opper and Winther 2001). The most commonly adopted one in the information theory and machine learning areas is the *variational Bayes* approach because it can easily produce a lower bound on the model evidence, which as we shall see is necessary for inference, and it offers a better probabilistic interpretation than the *field theoretic* and *TAP* approaches that are outside the scope of this thesis. The interested reader can consult Parisi (1988) for a thorough exposition and review of the last methods as we will now focus on the proposed *variational* approach that has been introduced in Chapter 2 and is adopted for probabilistic multiple kernel learning.

4.1.1 Variational Mean Field Theory for Classification

In the *variational Bayes* framework[2] we seek to approximate the joint parameter posterior distribution with an ensemble of factored approximate posteriors that belong to a tractable family of distributions (typically the exponential family). Considering a classifier m with a set of parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_i, \ldots, \boldsymbol{\Theta}_I\}$ and a dataset $\mathcal{D} = \{\mathbf{t}, \mathbf{X}\}$ with labels $\mathbf{t}$ and input samples $\mathbf{X}$, the approximation can

[2]Also known as ensemble learning, see Bishop (2006) for a thorough treatment.

be expressed as:

$$p(\boldsymbol{\Theta}|\mathbf{t},\mathbf{X},m) \approx Q\left(\boldsymbol{\Theta}\right) = \prod_{i=1}^{I} Q_i\left(\boldsymbol{\Theta}_i\right) \tag{4.1}$$

The above factorisation of approximate posteriors does neglect statistical correlations between the random variables, i.e. $\mathbb{E}_Q\left\{\boldsymbol{\Theta}_i, \boldsymbol{\Theta}_I\right\} = \mathbb{E}_{Q_i}\left\{\boldsymbol{\Theta}_i\right\}\mathbb{E}_{Q_I}\left\{\boldsymbol{\Theta}_I\right\}$, but takes into account their inherent coupling as will be shown in the next section.

The approximate distribution $Q\left(\boldsymbol{\Theta}\right) = \prod_{i=1}^{I} Q_i\left(\boldsymbol{\Theta}_i\right)$ is chosen such that it minimises an appropriate divergence measure with respect to the true joint posterior distribution. That measure, as we have seen previously, is typically the *Kullback-Leibler (KL)* divergence[3] (Kullback and Leibler 1951) as it enables tractable computations:

$$\mathrm{KL}\left(Q\left(\boldsymbol{\Theta}\right) \parallel p\left(\boldsymbol{\Theta}|\mathbf{t},\mathbf{X},m\right)\right) = \int Q\left(\boldsymbol{\Theta}\right) \log \frac{Q\left(\boldsymbol{\Theta}\right)}{p\left(\boldsymbol{\Theta}|\mathbf{t},\mathbf{X},m\right)}\, d\boldsymbol{\Theta} \tag{4.2}$$

The main goal in *variational Bayes* methods is to approximate and lower bound the marginal likelihood or *model evidence* (MacKay 2003, Beal 2003). In doing so, the minimisation of the above KL divergence between the approximate and true posterior distribution is implicitly achieved. Consider the following decomposition of the marginal likelihood in our classification context:

$\log p\left(\mathbf{t}|\mathbf{X},m\right) =$

$$\underbrace{\int Q(\boldsymbol{\Theta}) \log\left\{\frac{p\left(\mathbf{t},\boldsymbol{\Theta}|\mathbf{X},m\right)}{Q(\boldsymbol{\Theta})}\right\} d\boldsymbol{\Theta}}_{\text{Lower Bound}} \underbrace{- \int Q(\boldsymbol{\Theta}) \log\left\{\frac{p(\boldsymbol{\Theta}|\mathbf{t},\mathbf{X},m)}{Q(\boldsymbol{\Theta})}\right\} d\boldsymbol{\Theta}}_{\text{KL Divergence}} \tag{4.3}$$

noting that the second term is the aforementioned KL divergence, the Naperian logarithm of the marginal likelihood can be re-expressed as:

$$\log p\left(\mathbf{t}|\mathbf{X},m\right) = \mathcal{L}(Q\left(\boldsymbol{\Theta}\right)) + \mathrm{KL}\left(Q\left(\boldsymbol{\Theta}\right) \parallel p\left(\boldsymbol{\Theta}|\mathbf{t},\mathbf{X},m\right)\right) \tag{4.4}$$

where $\mathcal{L}(Q\left(\boldsymbol{\Theta}\right))$ is a *lower bound* on the model evidence since the KL divergence is always greater or equal to zero. Hence, by maximising the lower bound we equivalently minimise the KL divergence (which appears in the decomposition of the lower bound as we have seen in Chapter 2) and achieve the desired goals

[3]Also known as the *relative entropy* or *information gain* (MacKay 2003)

of approximating the marginal likelihood and the posterior distributions.

Finally, for the factorised ensemble approximation that is adopted in this thesis, it has been shown in Chapter 2 that optimal approximate distributions that maximise the lower bound are of the following form:

$$Q_i(\boldsymbol{\Theta}_i) \propto \exp\left(\mathbb{E}_{i\neq j}\left\{\log p\left(\mathbf{t}, \boldsymbol{\Theta}|\mathbf{X}, m\right)\right\}\right) \tag{4.5}$$

where the expectation $\mathbb{E}_{i\neq j}$ is taken with respect to all factors Q except the j^{th} one.

4.2 Variational Bayes Probabilistic Multiple Kernel Learning

Having introduced the ideas behind the variational methodology and its adoption for the general classification framework, a variational treatment is now applied to the probabilistic MKL problem under consideration. In this section the focus is on the standard *convex linear* kernel combination rule with appropriate modifications given for the other previously proposed rules in previous Chapter.

Revisiting the proposed MKL model's plates diagram in Figure 4.1, we can define for convenience as $\boldsymbol{\Psi} = \{\mathbf{Y}, \mathbf{W}, \mathbf{A}, \boldsymbol{\Theta}, (\boldsymbol{\beta}), (\boldsymbol{\rho}), (\boldsymbol{\pi}, \boldsymbol{\chi})\}$ the set of all prior parameters and $\boldsymbol{\Xi} = \{\tau, \upsilon, \omega, \phi, (\mu), (\lambda)\}$ the set of all *hyper-prior* parameters[4].

Consider again the standard multinomial classification scenario on a multi-feature dataset $D = \left\{\mathbf{t}, \mathbf{X}^{(s)}\right\}$ where $\mathbf{X}^{(s)} \in \mathbb{R}^{N\times D_s}$, $s \in \{1, \ldots, S\}$, $t_n \in \{1, \ldots, C\}$ and $S, N, D, C \in \mathbb{N}$ the total number of sources, samples, features and classes respectively. After embedding the feature spaces into S base kernels $\mathbf{K}_s$ that will be combined into the composite kernel $\mathbf{K}$, the joint likelihood of the *convex linear* MKL model m is given by:

$$\text{JOINT LIKELIHOOD} \quad p\left(\mathbf{t}, \boldsymbol{\Psi}|\mathbf{K}_{s:1\ldots S}, \boldsymbol{\Xi}, m\right) =$$

$$p\left(\mathbf{t}|\mathbf{Y}\right) p\left(\mathbf{Y}|\mathbf{W}, \mathbf{K}_{s:1\ldots S}, \boldsymbol{\beta}, \boldsymbol{\Theta}\right) p\left(\mathbf{W}|\mathbf{A}\right) p\left(\mathbf{A}|\tau, \upsilon\right) p\left(\boldsymbol{\beta}|\boldsymbol{\rho}\right) p(\boldsymbol{\Theta}|\omega, \phi) p\left(\boldsymbol{\rho}|\mu, \lambda\right) \tag{4.6}$$

which is accordingly modified for the other parametric combination rules by substituting the corresponding prior distributions. The variational mean field approximation to the joint posterior is the factorable ensemble:

[4] $(\cdot)$ denotes a possibly undefined random variable for a given kernel combination rule.

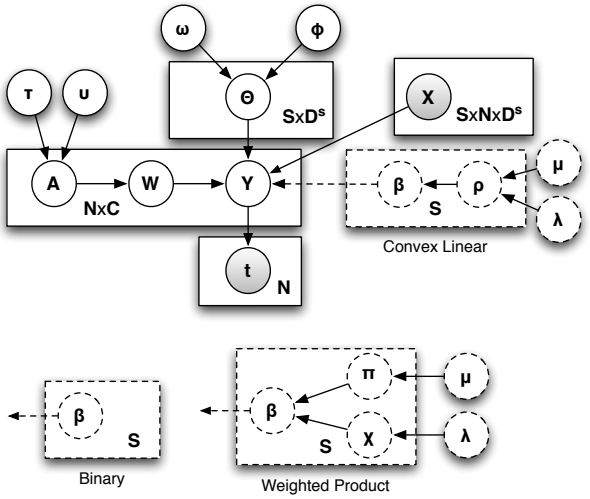

Figure 4.1: Plates diagram of the model depicting the conditional relationships of model variables together with the dimensionality of corresponding plates. The dotted plates depict variations for the three parametric combination rules.

$$p\left(\boldsymbol{\Psi}|\mathbf{K}_{s:1\ldots S},\boldsymbol{\Xi},\mathbf{t}\right)\approx Q\left(\boldsymbol{\Psi}\right)=Q\left(\mathbf{Y}\right)Q\left(\mathbf{W}\right)Q\left(\mathbf{A}\right)Q\left(\boldsymbol{\beta}\right)Q\left(\boldsymbol{\Theta}\right)Q\left(\boldsymbol{\rho}\right) \tag{4.7}$$

and following the decomposition in Equation 4.3 we can lower bound the marginal likelihood:

$$\textsc{lower bound}\quad \log p\left(\mathbf{t}|\mathbf{K}_{s:1\ldots S},\boldsymbol{\Xi},m\right)\geq\int Q(\boldsymbol{\Psi})\log\left\{\frac{p\left(\mathbf{t},\boldsymbol{\Psi}|\mathbf{K}_{s:1\ldots S},\boldsymbol{\Xi},m\right)}{Q(\boldsymbol{\Psi})}\right\}d\boldsymbol{\Psi} \tag{4.8}$$

or equivalently:

$$\log p\left(\mathbf{t}|\mathbf{K}_{s:1\ldots S},\boldsymbol{\Xi},m\right)\geq\mathbb{E}_{Q(\boldsymbol{\Psi})}\left\{\log p\left(\mathbf{t},\boldsymbol{\Psi}|\mathbf{K}_{s:1\ldots S},\boldsymbol{\Xi},m\right)\right\}-\mathbb{E}_{Q(\boldsymbol{\Psi})}\left\{\log Q(\boldsymbol{\Psi})\right\} \tag{4.9}$$

and minimise it, according to Equation 4.5, with distributions of the form:

$$Q\left(\boldsymbol{\Psi}_i\right)\propto\exp\left(\mathbb{E}_{Q(\Psi_{-i})}\{\log p\left(\mathbf{t},\boldsymbol{\Psi}|\mathbf{K}_{s:1\ldots S},\boldsymbol{\Xi},m\right)\}\right) \tag{4.10}$$

where $Q\left(\Psi_{-i}\right)$ is the factorable ensemble with the i^{th} component removed.

The derived closed form approximate posterior distributions for the model's random variables follows with full details in Appendix B.1. When a closed

form approximate posterior distribution is not available due to non-linear contributions to the expectation we resort to importance sampling techniques as previously employed in Lawrence et al. (2004) and Girolami and Rogers (2006).

4.2.1 $Q(\mathbf{Y})$: Approximate posterior for Y

A closed form approximate posterior for the auxiliary variables can be derived based on the multinomial probit link function $t_n = i \iff y_{ni} > y_{nj} \ \forall \ j \neq i$ with $j, i \in \{1, \ldots, C\}$, and the associated prior distribution on $\mathbf{Y}$ given in Equation 3.8. Considering the approximate posterior form in Equation 4.10, the model's joint likelihood in Equation 4.6 and its contributing terms to the required expectation we have:

$$Q(\mathbf{Y}) \propto \exp\left\{E_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\left\{\log p(\mathbf{t}|\mathbf{Y}) + \log p(\mathbf{Y}|\mathbf{W}, \mathbf{K}_{s:1\ldots S}, \boldsymbol{\beta}, \boldsymbol{\Theta})\right\}\right\} \tag{4.11}$$

$$Q(\mathbf{Y}) \propto \prod_{n=1}^{N} \delta(y_{ni} > y_{nj} \ \forall \ j \neq i)\, \delta(t_n = i)\, \mathcal{N}_{\mathbf{y}_n}\left(\widetilde{\mathbf{W}}^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}, \mathbf{I}\right) \tag{4.12}$$

which is a product of N $C-$dimensional conically truncated Gaussian distributions demonstrating independence across samples as expected from our initial i.i.d assumption. The proportionality indicates the unnormalized nature of the approximate posterior which can be corrected to account for the truncation by a normalising factor $\mathcal{Z}_n = P(\mathbf{y}_n \in \mathcal{C})$ where the cone is defined as $\mathcal{C} = \{\mathbf{y}_n : y_{ni} > y_{nj}\} \ \forall \ j \neq i$, see Appendix B.2 for full derivations, and results in:

$$Q(\mathbf{Y}) = \prod_{n=1}^{N} \mathcal{Z}_n^{-1} \prod_{c=1}^{C} \mathcal{N}_{y_{nc}}^{t_n}\left(\widetilde{\mathbf{w}}_c^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1\right) \tag{4.13}$$

where the superscript ^{t_n} denotes the dimension for truncation when $t_n = i$ and $c \neq i$. The normalising constant is given by:

$$\mathcal{Z}_n = \mathbb{E}_{p(u)}\left\{\prod_{j \neq i} \Phi\left(u + \widetilde{\mathbf{w}}_i^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{\mathbf{w}}_j^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}\right)\right\} \tag{4.14}$$

with $p(u) = \mathcal{N}_u(0,1)$ and Φ the standardised cumulative distribution function (CDF). The shorthand tilde notation denotes posterior expectations in the usual manner, i.e. $\widetilde{f(\boldsymbol{\beta})} = \mathbb{E}_{Q(\boldsymbol{\beta})}\{f(\boldsymbol{\beta})\}$, and the posterior expectations (details in Appendix B.2) for the auxiliary variable when sample n belongs to class $i \in \{1,\ldots,C\}$ follow as:

$$\widetilde{y}_{nc} = \widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \frac{\mathbb{E}_{p(u)}\left\{\mathcal{N}_u\left(\widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}, 1\right)\Phi_u^{n,i,c}\right\}}{\mathbb{E}_{p(u)}\left\{\Phi\left(u + \widetilde{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right)\Phi_u^{n,i,c}\right\}} \tag{4.15}$$

$$\widetilde{y}_{ni} = \widetilde{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \left(\sum_{c\neq i}\widetilde{y}_{nc} - \widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right) \tag{4.16}$$

where $\Phi_u^{n,i,c} = \prod_{j\neq i,c}\Phi\left(u + \widetilde{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_j^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right)$. It is worth noting the coupling between the approximate posterior expectations for the auxiliary variables and the regression coefficients, which ensures that appropriate (in the sense of following the probit link definition) class-conditional posterior dependencies are induced.

4.2.2 $Q(\mathbf{W})$: Approximate posterior for regression coefficients W

In an analogous manner, the approximate posterior distribution for the regression coefficients following Equation 4.10 is defined as:

$$Q(\mathbf{W}) \propto \exp\left\{E_{Q(\mathbf{Y})Q(\boldsymbol{\beta})Q(\mathbf{A})Q(\boldsymbol{\Theta})}\{\log p(\mathbf{Y}|\mathbf{W},\mathbf{K}_{s:1\ldots S},\boldsymbol{\beta},\boldsymbol{\Theta}) + \log p(\mathbf{W}|\mathbf{A})\}\right\} \tag{4.17}$$

and by substituting for the model's prior distributions, see Chapter 3, it can be decomposed as:

$$Q(\mathbf{W}) \propto \exp\left\{E_{Q(\mathbf{Y})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\left\{\log\prod_{n,c=1}^{N,C}\mathcal{N}_{\mathbf{y}_{nc}}\left(\mathbf{w}_c^{\mathsf{T}}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}, 1\right)\right\} + E_{Q(\mathbf{A})}\left\{\log\prod_{n,c=1}^{N,C}\mathcal{N}_{\mathbf{w}_{nc}}\left(0,\alpha_{nc}^{-1}\right)\right\}\right\} \tag{4.18}$$

which by taking the expectations and completing the square, see Appendix B.1.2 for full derivation, leads to the final closed form approximate posterior for the regression coefficients:

$$Q(\mathbf{W}) = \prod_{c=1}^{C} \mathcal{N}_{\mathbf{w}_c}\left(\mathbf{V}_c \mathbf{K}^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} \widetilde{\mathbf{y}}_c, \mathbf{V}_c\right) \tag{4.19}$$

where the covariance is defined as

$$\mathbf{V}_c = \left(\sum_{i=1}^{S}\sum_{j=1}^{S} \widetilde{\beta_i \beta_j} \mathbf{K}_i^{\widetilde{\boldsymbol{\theta}}_i} \mathbf{K}_j^{\widetilde{\boldsymbol{\theta}}_j} + \widetilde{\mathbf{A}}_c\right)^{-1} \tag{4.20}$$

and $\widetilde{\mathbf{A}}_c$ is a diagonal matrix of the expected variances $\widetilde{\alpha}_i \dots \widetilde{\alpha}_N$ for each class. The associated posterior mean for the regression coefficients is therefore $\widetilde{\mathbf{w}}_c = \mathbf{V}_c \mathbf{K}^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} \widetilde{\mathbf{y}}_c$ and we can see again the coupling between the auxiliary variable and regressor's posterior expectation.

4.2.3 $Q(\mathbf{A})$: Approximate posterior of scales A

The conjugacy between the Gamma prior of the scales and the Normal prior on the regression coefficients allows a closed form posterior for the scales as it was shown in Equation 3.24 in Chapter 3. Starting again from the same principles as above for deriving the approximate posterior:

$$Q(\mathbf{A}) \propto \exp\left\{E_{Q(\mathbf{W})}\left(\log p(\mathbf{W}|\mathbf{A}) + \log p(\mathbf{A}|\tau, \upsilon)\right)\right\} \tag{4.21}$$

which results, for a detailed derivation see Appendix B.1.3 or Denison et al. (2002), into the previously described Gamma posterior distribution, this time conditioned on the expected value of the regression coefficients:

$$Q(\mathbf{A}) = \prod_{n,c=1}^{N,C} \mathcal{G}_{\alpha_{nc}}\left(\tau + \frac{1}{2}, \upsilon + \frac{1}{2}\widetilde{w}_{nc}^2\right) \tag{4.22}$$

and hence the approximate posterior mean for the scale is given by

$$\widetilde{a}_{nc} = \frac{2\tau + 1}{2\upsilon + \widetilde{w}_{nc}^2} \tag{4.23}$$

4.2.4 $Q(\boldsymbol{\Theta})$: Approximate posterior for $\boldsymbol{\Theta}$

Following Equation 4.10, the approximate posterior for the kernel parameters $\boldsymbol{\Theta} \in \mathbb{R}^{S \times D^s}$, where D^s the dimensionality of each feature space s embedded into base kernel $\mathbf{K}_s$, is given by:

$$Q(\boldsymbol{\Theta}) \propto \exp\left\{E_{Q(\mathbf{Y})Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\mathbf{A})}\{\log p(\mathbf{Y}|\mathbf{W}, \mathbf{K}_{s:1\ldots S}, \boldsymbol{\beta}, \boldsymbol{\Theta}) + \log p(\boldsymbol{\Theta}|\phi, \omega)\}\right\} \tag{4.24}$$

where the normalising constant of this approximate posterior cannot be obtained in closed form. Hence the need to resort to importance sampling methods (Andrieu 2003), as previously employed within a variational framework (Lawrence et al. 2004, Girolami and Rogers 2006), in order to obtain the required expectations.

Following standard importance sampling methodology that was reviewed in Chapter 2, the required importance weights for the kernel parameters $\boldsymbol{\Theta}$, see Appendix B.1.4 for details, are given by:

$$\mathcal{W}\left(\boldsymbol{\Theta}^i\right) = \frac{\prod_{n,c=1}^{N,C} \mathcal{N}_{\widetilde{\mathbf{y}}_{nc}}\left(\widetilde{\mathbf{w}}_c^{\top}\mathbf{k}_n^{\boldsymbol{\Theta}^i}, 1\right)}{\sum_{i'=1}^{I}\prod_{n,c=1}^{N,C} \mathcal{N}_{\widetilde{\mathbf{y}}_{nc}}\left(\widetilde{\mathbf{w}}_c^{\top}\mathbf{k}_n^{\boldsymbol{\Theta}^{i'}}, 1\right)} \tag{4.25}$$

where $\boldsymbol{\Theta}^i$ is sampled from the Gamma prior $p(\boldsymbol{\Theta}|\phi, \omega)$ and the resulting estimator for the required expectations is:

$$\widetilde{f(\boldsymbol{\Theta})} \approx \sum_{i=1}^{I} f(\boldsymbol{\Theta}^i)\mathcal{W}\left(\boldsymbol{\Theta}^i\right) \tag{4.26}$$

with scaling per sample similar to gradient based methods that optimise the marginal likelihood (MacKay 2003).

4.2.5 $Q(\boldsymbol{\beta})$: Approximate posterior for $\boldsymbol{\beta}$

Again no closed form approximate posterior distribution can be obtained for the kernel combination parameters and in the exact same manner as above importance sampling weights are employed to provide an unbiased estimator of expectations. The importance weights, see Appendix B.1.4, are defined again as

a normalised likelihood ratio:

$$\mathcal{W}\left(\boldsymbol{\beta}^i\right) = \frac{\prod_{n,c=1}^{N,C} \mathcal{N}_{\tilde{\mathbf{y}}_{nc}}\left(\widetilde{\mathbf{w}}_c^{\mathsf{T}} \mathbf{k}_n^{\boldsymbol{\beta}^i}, 1\right)}{\sum_{i'=1}^{I} \prod_{n,c=1}^{N,C} \mathcal{N}_{\tilde{\mathbf{y}}_{nc}}\left(\widetilde{\mathbf{w}}_c^{\mathsf{T}} \mathbf{k}_n^{\boldsymbol{\beta}^{i'}}, 1\right)} \tag{4.27}$$

where $\boldsymbol{\beta}^i$ is sampled, for the convex linear combination rule, from the Dirichlet prior $p\left(\boldsymbol{\beta}|\boldsymbol{\rho}\right)$ and the estimator follows Equation 4.26 substituting for the kernel combination parameters $\boldsymbol{\beta}$. In the case of the two other parametric combination rules, the appropriate prior distribution is employed each time (Gamma for the weighted product rule and a binomial for the binary rule) for sample proposal with the same importance weights definition and resulting estimator.

4.2.6 $Q\left(\boldsymbol{\rho}\right) Q\left(\boldsymbol{\pi}\right) Q\left(\boldsymbol{\chi}\right)$: Approximate posteriors for $\boldsymbol{\rho}, \boldsymbol{\pi}, \boldsymbol{\chi}$

Finally, inference on the hyper-parameters leads again to unnormalized approximate posteriors with the normalising constant unobtainable in closed form. Hence, we resort again to importance sampling with proposed parameters sampled from the corresponding prior distributions and weights defined by normalised likelihood ratios. For the *convex linear* kernel combination rule the importance weights for the hyper-parameter associated to the Dirichlet distribution on $\boldsymbol{\beta}$ are:

$$\mathcal{W}\left(\boldsymbol{\rho}^i\right) = \frac{\mathcal{D}_{\widetilde{\boldsymbol{\beta}}}\left(\boldsymbol{\rho}^i\right)}{\sum_{i'=1}^{I} \mathcal{D}_{\widetilde{\boldsymbol{\beta}}}\left(\boldsymbol{\rho}^{i'}\right)} \tag{4.28}$$

where $\boldsymbol{\rho}$ is sampled from the prior Gamma distribution. In the case of the *weighted product* kernel combination rule, the hyper-parameters associated with the Gamma distribution on $\boldsymbol{\beta}$ are:

$$\mathcal{W}\left(\pi^i, \chi^i\right) = \frac{\prod_{s=1}^{S} \mathcal{G}_{\widetilde{\beta}_s}\left(\pi^i, \chi^i\right)}{\sum_{i'=1}^{I} \prod_{s=1}^{S} \mathcal{G}_{\widetilde{\beta}_s}\left(\pi^{i'}, \chi^{i'}\right)} \tag{4.29}$$

where π, χ are proposed from their corresponding prior exponential distribu-

tions defined in Chapter 3. The unbiased estimator for the required expectations follows from Equation 4.26, for each specific random variable considered, by making use of the associated importance weights.

4.2.7 Predictive Distribution

In the previous sections the approximate posterior distributions have been derived and analytically described. The strength of variational methodology, as previously emphasised, is the inference of (approximate) posterior distributions over parameters in contrast with simple point-estimate deterministic approximations (Bishop 2006) such as maximum likelihood (ML) or maximum-a-posteriori (MAP) methods. Approximate posterior inference of the model parameters during the training phase of the variational MKL method is now complete and we can now turn our attention to the prediction or testing phase of the algorithm.

Returning back to the goal of MKL classification scenarios, we are interested in making class predictions $\mathbf{t}_*$ for N_{test} new samples that are represented by S different information sources $\mathbf{X}_*^{(s)}$ embedded into Hilbert spaces as base test kernels $\mathbf{K}_{*s}$. Considering the limiting case of a single new sample $\mathbf{x}_*^{(s)}$ the composite test kernel element for the standard *convex linear* combination rule is defined as:

$$k_*\left(\mathbf{x}_i, \mathbf{x}_*, \boldsymbol{\theta}^{(s)}, \boldsymbol{\beta}\right) = \sum_{s=1}^{S} \beta_s k_{*s}\left(\mathbf{x}_i^{(s)}, \mathbf{x}_*^{(s)}, \boldsymbol{\theta}^{(s)}\right) \tag{4.30}$$

which generalises for multiple new samples to the overall composite test kernel $\mathbf{K}_* \in \mathbb{R}^{N \times N_{\text{test}}}$. The class predictive distribution for a single new object $\mathbf{x}_*$ is given by:

$$p\left(t_* = i|\mathbf{k}_*, \mathbf{K}, \mathbf{t}\right) = \int p\left(t_* = i|\mathbf{y}_*\right) p\left(\mathbf{y}_*|\mathbf{k}_*, \mathbf{K}, \mathbf{t}\right) d\mathbf{y}_* = \int \delta^{t_*} p\left(\mathbf{y}_*|\mathbf{k}_*, \mathbf{K}, \mathbf{t}\right) d\mathbf{y}_* \tag{4.31}$$

where $\delta^{t_*} = \delta\left(y_{*i} > y_{*j} \;\forall\; j \neq i\right) \delta\left(t_* = i\right)$ denotes the truncation induced by the test sample for class i membership. The second term is the predictive distribution for the auxiliary variable and is given by:

$$p\left(\mathbf{y}_*|\mathbf{k}_*, \mathbf{K}, \mathbf{t}\right) = \int p\left(\mathbf{y}_*|\mathbf{W}, \mathbf{k}_*\right) p\left(\mathbf{W}|\mathbf{K}, \mathbf{t}\right) d\mathbf{W} \tag{4.32}$$

which, by substituting for the approximate posterior on the regression coeffi-

cients $Q(W)$, gives rise (full derivation in Appendix B.3) to the final expression for the class predictive distribution:

$$p(t_* = i|\mathbf{k}_*, \mathbf{K}, \mathbf{t}) = E_{p(u)}\left\{\prod_{j\neq i}\Phi\left[\frac{1}{\widetilde{\nu}_j}\left(u\widetilde{\nu}_i + \widetilde{m}_i - \widetilde{m}_j\right)\right]\right\} \tag{4.33}$$

where, for the general case of N_{test} samples the variables $\widetilde{\mathbf{m}}_c$ and $\widetilde{\boldsymbol{\mathcal{V}}}_c$ are defined as:

$$\widetilde{\mathbf{m}}_c = \widetilde{\boldsymbol{\mathcal{V}}}_c \mathbf{K}_*^{\mathsf{T}}\left(\mathbf{K}_*\mathbf{K}_*^{\mathsf{T}} + \mathbf{V}_c^{-1}\right)^{-1}\mathbf{K}\widetilde{\mathbf{y}}_c \tag{4.34}$$

$$\widetilde{\boldsymbol{\mathcal{V}}}_c = \left(\mathbf{I} + \mathbf{K}_*{}^{\mathsf{T}}\mathbf{V}_c\mathbf{K}_*\right) \tag{4.35}$$

with $\mathbf{V}_c$ the covariance of the approximate regressor posterior as defined in Equation 4.20 and by dropping the notation for the dependence of the train $\mathbf{K} \in \mathbb{R}^{N\times N}$ and test $\mathbf{K}_* \in \mathbb{R}^{N\times N_{\text{test}}}$ kernels on $\widetilde{\boldsymbol{\Theta}}, \widetilde{\boldsymbol{\beta}}$ for clarity. In Algorithm 3 the variational approximation for probabilistic multiple kernel learning (VBpMKL) is summarised in a pseudo-algorithmic fashion.

4.3 Convergence and the Lower Bound

Convergence is typically monitored via the progression of the lower bound and its relative change. The variational lower bound, recalling Equation 4.8, can be derived by conditioning on current values of $\boldsymbol{\beta}, \boldsymbol{\Theta}, \mathbf{A}$ and associated hyper-parameters and by considering the relevant components of the joint likelihood in Equation 4.6 as follows:

$$\begin{aligned}\text{Lower Bound} \quad &= \mathbb{E}_{Q(\mathbf{Y})Q(\mathbf{W})}\left\{\log p(\mathbf{Y}|\mathbf{W}, \mathbf{K}_{s:1\ldots S})\right\} + \mathbb{E}_{Q(\mathbf{Y})Q(\mathbf{W})}\left\{\log p(\mathbf{W}|\mathbf{A})\right\} \\ &\quad - \mathbb{E}_{Q(\mathbf{Y})}\left\{\log Q(\mathbf{Y})\right\} - \mathbb{E}_{Q(\mathbf{W})}\left\{\log Q(\mathbf{W})\right\}\end{aligned} \tag{4.36}$$

which results in, see Appendix B.4 for derivation, to the final expression:

Algorithm 3 VBpMKL

Initialise $\boldsymbol{\Xi}$, sample $\boldsymbol{\Psi}$, create $\mathbf{K}_s|\beta_s, \theta_s$ and hence $\mathbf{K}|\boldsymbol{\beta}, \boldsymbol{\Theta}$
while Iterations < max & Convergence > Threshold **do**
  $\widetilde{\mathbf{w}}_c \leftarrow \mathbf{V}_c \mathbf{K} \widetilde{\mathbf{y}_c}$
  $\widetilde{y}_{nc} \leftarrow \widetilde{\mathbf{w}}_c^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \frac{\mathbb{E}_{p(u)}\{\mathcal{N}_u\left(\widetilde{\mathbf{w}}_c^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_i^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}, 1\right) \Phi_u^{n,i,c}\}}{\mathbb{E}_{p(u)}\{\Phi\left(u + \widetilde{\mathbf{w}}_i^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \widetilde{\mathbf{w}}_c^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right) \Phi_u^{n,i,c}\}}$
  $\widetilde{y}_{ni} \leftarrow \widetilde{\mathbf{w}}_i^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} - \left(\sum_{j \neq i} \widetilde{y}_{nj} - \widetilde{\mathbf{w}}_j^{\mathsf{T}} \mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}\right)$
  $\widetilde{\alpha}_{nc}^{-1} \leftarrow \frac{2\tau+1}{2\upsilon + \widetilde{w}_{nc}^2}$
  $\widetilde{\boldsymbol{\rho}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Theta}} \leftarrow \widetilde{\boldsymbol{\rho}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Theta}} | \widetilde{\mathbf{w}}_c, \widetilde{\mathbf{y}}_n$ by importance sampling
  Update $\mathbf{K}|\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Theta}}$ and $\mathbf{V}_c$
end while
Create composite test kernel $\mathbf{K}_*|\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\Theta}}$
$\widetilde{\boldsymbol{\mathcal{V}}_c} \leftarrow \left(\mathbf{I} + \mathbf{K}_*^{\mathsf{T}} \mathbf{V}_c \mathbf{K}_*\right)$
$\widetilde{\mathbf{m}}_c \leftarrow \widetilde{\boldsymbol{\mathcal{V}}}_c \mathbf{K}_*^{\mathsf{T}} \left(\mathbf{K}_* \mathbf{K}_*^{\mathsf{T}} + \mathbf{V}_c^{-1}\right)^{-1} \mathbf{K} \widetilde{\mathbf{y}}_c$
for $n = 1$ to N_{test} **do**
  for $i = 1$ to C **do**
    for $l = 1$ to L Samples **do**
      $u^l \leftarrow \mathcal{N}(0,1), \quad p_{ni}^l \leftarrow \prod_{j \neq i} \Phi\left[\frac{1}{\widetilde{\nu}_j}\left(u^l \widetilde{\nu}_i + \widetilde{m}_i - \widetilde{m}_j\right)\right]$
    end for
  end for
  $P(t_* = i | \mathbf{k}_*, \mathbf{K}, \mathbf{t}) = \frac{1}{L} \sum_{l=1}^{L} p_{ni}^l$
end for

$$\begin{aligned} \text{Lower Bound} = \frac{NC}{2} + \frac{1}{2} \sum_{c=1}^{C} \log |\mathbf{V}_c| + \sum_{n=1}^{N} \log \mathcal{Z}_n - \frac{1}{2} \sum_{c=1}^{C} \text{Tr}\left[\mathbf{A}_c^{-1} \mathbf{V}_c\right] \\ - \frac{1}{2} \sum_{c=1}^{C} \widetilde{\mathbf{w}}_c^{\mathsf{T}} \mathbf{A}_c^{-1} \widetilde{\mathbf{w}}_c - \frac{1}{2} \sum_{c=1}^{C} \log |\mathbf{A}_c| - \frac{1}{2} \sum_{c=1}^{C} \sum_{n=1}^{N} \mathbf{k}_n^{\mathsf{T}} \mathbf{V}_c \mathbf{k}_n \end{aligned} \quad (4.37)$$

4.4 Computational Complexity

The variational methodology adopted in this Chapter aims at reducing the computational complexity of the Markov chain Monte Carlo solutions of Chapter 3 while retaining similar levels of classification accuracy and performance metrics. The reduction achieved is from a computational complexity of $\mathcal{O}(LCN^3) \overset{\text{to}}{\rightarrow} \mathcal{O}(TCN^3)$ with T the number of the variational iterations and S, L, C, N are

the number of sources, drawn posterior samples (length of Markov chain), classes and samples (objects) respectively. Typically $T << L$ (e.g $T \leq 100$ and $L \geq 100,000$) as the number of required iterations for convergence of the lower bound is orders of magnitude smaller than the necessary Markov chain length.

Furthermore, although the apparent memory requirements remain the same as $\mathcal{O}(SN^2 + 3NC)$, the additional memory for storing the L posterior samples when computing convergence metrics for the MCMC solutions is now made redundant via the computation of the lower bound and the converged approximate posteriors.

4.5 Variational Inference and Gibbs Sampling

This section examines the performance of the variational Bayes approximation with respect to the full MCMC Gibbs sampling solution previously introduced in Chapter 3. The comparison is performed between the variational approximate posterior distribution and the Gibbs sampling posterior, classification accuracy and computational processing time on two artificial low-dimensional datasets, a linearly and a non-linearly separable one as introduced by (Neal 1998).

Furthermore, the convergence of the VBpMKL approximation was determined by monitoring the lower bound and the convergence occurred when there was less than 0.1% increase in the bound or when the maximum number of variational iterations was reached. The burn-in period for the Gibbs sampler was set to 10% of the total 100,000 of samples. Finally, all the CPU times reported in this study are for a 1.6 GHz Intel based PC with 2Gb RAM running unoptimised Matlab® codes.

4.5.1 Synthetic Data sets

In order to illustrate the performance of the variational approximation against the full Gibbs sampling solution, we employ two low dimensional datasets which enable us to visualise the decision boundaries and posterior distributions produced by either method. First we consider a linearly separable case in which we construct the dataset by fixing our regression coefficients $\mathbf{W} \in \mathbb{R}^{D \times C}$, with $C = 3$ and $D = 3$, to known values and sample two-dimensional covariates $\mathbf{X}$ plus a constant term. In that way, by knowing the true values of our regression

coefficients, we can examine the accuracy of both the Gibbs posterior distribution and the approximate posterior distribution of the variational method. In Figure 4.2 the dataset together with the optimal decision boundaries constructed by the known regression coefficients values can be seen.

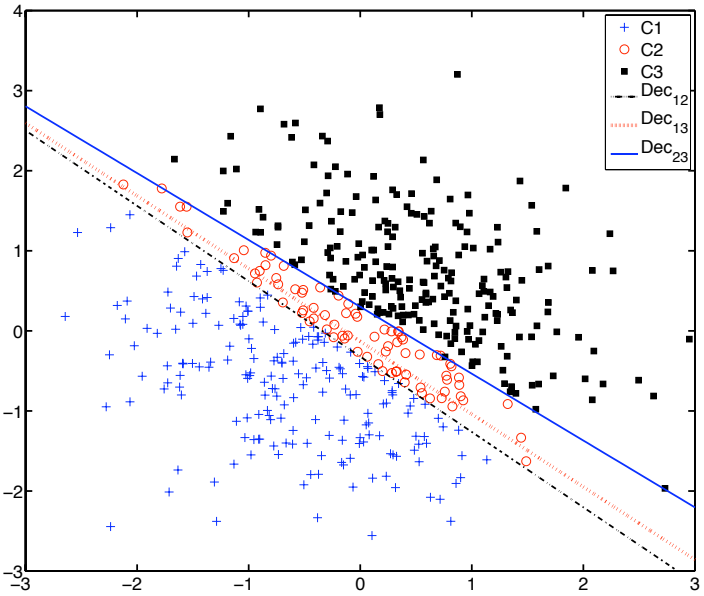

Figure 4.2: Linearly separable dataset with known regression coefficients defining the decision boundaries. C_n denotes the members of class n and Dec_{ij} is the decision boundary between classes i and j.

In Figures 4.3 and 4.4 the posterior distributions of one decision boundary's (Dec_{12}) slope and intercept based on both our obtained Gibbs samples and the approximate posterior of the regression coefficients $\mathbf{W}$ are plotted. As it can be seen, the variational approximation is in agreement with the mass of the Gibbs posterior and it successfully captures the predetermined regression coefficients values.

However, as it can be observed the approximation is over-confident in the prediction and produces a smaller covariance for the posterior distribution as expected (de Freitas et al. 2001). Furthermore, the probability mass is concentrated in a very small area due to the very nature of variational approximations and similar mean field methods that make extreme "judgements" as they do not explore the posterior space by Markov chains.

$$C_{\text{Gibbs}} = \begin{bmatrix} 0.16 & 0.18 \\ 0.18 & 0.22 \end{bmatrix} \quad C_{\text{VB}} = \begin{bmatrix} 0.015 & 0.015 \\ 0.015 & 0.018 \end{bmatrix} \tag{4.38}$$

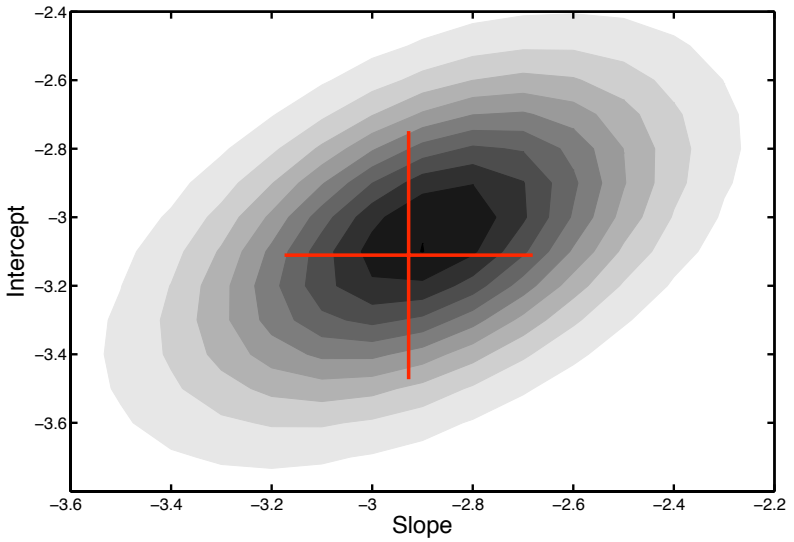

Figure 4.3: Gibbs posterior distribution of a decision boundary's (Dec_{12}) slope and intercept for a Markov chain of 100,000 samples. The cross describes the original decision boundary employed to sample the dataset.

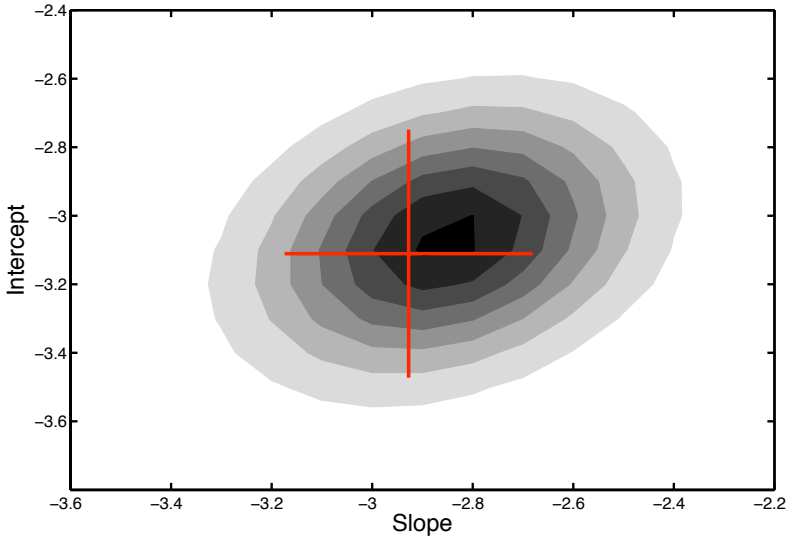

Figure 4.4: The variational approximate posterior distribution for the same case as above. Employing 100,000 samples from the approximate posterior of the regression coefficients $\mathbf{W}$ in order to estimate the approximate posterior of the slope and intercept.

The second synthetic dataset we employ is a 4-dimensional 3-class dataset $\{\mathbf{t}, \mathbf{X}\}$ with $N = 400$ samples, first described by (Neal 1998), which defines the first class as points in an ellipse $\alpha > x_1^2 + x_2^2 > \beta$, the second class as points below a line $\alpha x_1 + \beta x_2 < \gamma$ and the third class as points surrounding these areas, see Figure 4.5.

The problem is tackled by introducing a second order polynomial expansion on the original dataset $F(\mathbf{x}_n) = [1\ x_{n1}\ x_{n2}\ x_{n1}^2\ x_{n1}x_{n2}\ x_{n2}^2]$ while disregarding the uninformative dimensions x_3, x_4. Due to the aforementioned expansion which avoids the need for embedding the features into a high dimensional Hilbert space induced by a kernel, there is now a 2-dimensional decision plane that can be visualised and 6-dimensional regression coefficients $\mathbf{w}_c$ per class. In Fig. 4.5 we plot the decision boundaries produced from the full Gibbs solution by averaging over the posterior parameters after 100,000 samples and in Fig. 4.6 the corresponding decision boundaries from the variational approximation after a maximum of 100 iterations.

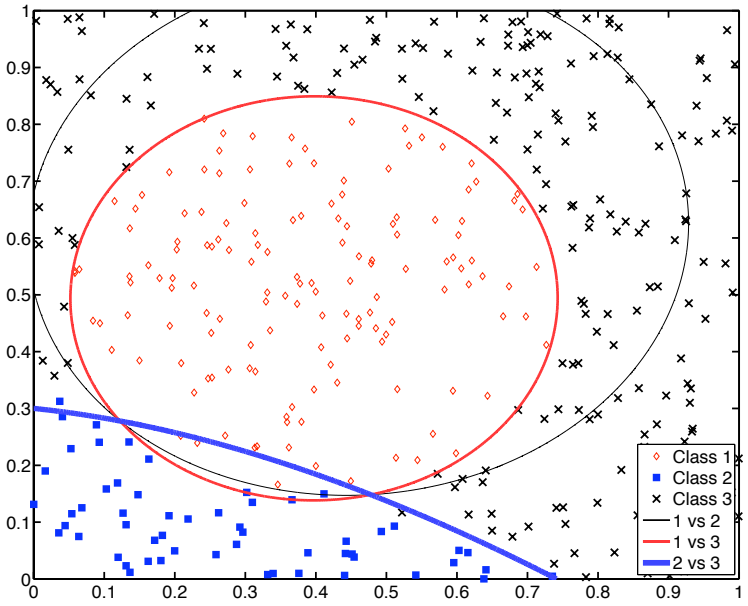

Figure 4.5: Decision boundaries from the Gibbs sampling solution on Neal's dataset.

As it can be seen, both the variational approximation and the MCMC solution produce similar decision boundaries leading to good classification performances of 2% error for both the Gibbs and the variational approximation. However, the Gibbs sampler produces typically tighter boundaries due to the

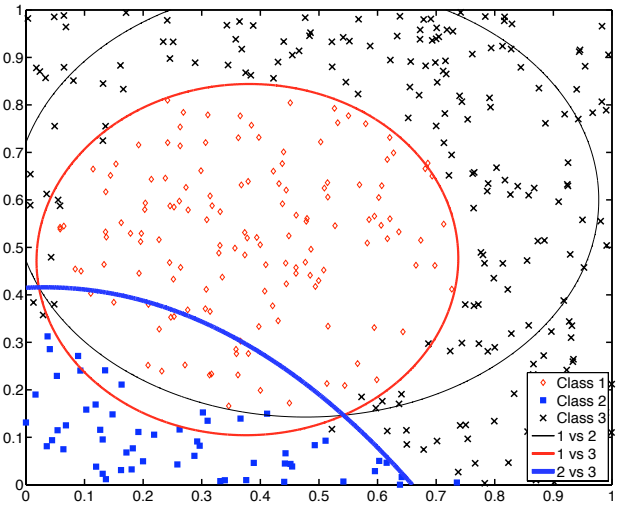

Figure 4.6: Decision boundaries from the variational approximation on Neal's dataset.

Markov Chain exploring the parameter posterior space more efficiently than the VB approximation.

The corresponding CPU times are given in Table 4.1

Gibbs	VB
41,720 (s)	120.3 (s)

Table 4.1: CPU time (sec) comparison for 100,000 Gibbs samples versus a maximum of 100 variational iterations. Notice that the number of variational iterations needed for the lower bound to converge is typically less than 100.

4.6 Multinomial UCI Experiments

In this final section the variational approximation is assessed on standard UCI (Newman et al. 1998) problems and compared against previously published results (Manocha and Girolami 2007) of probabilistic and standard nearest neighbour ($k-$nn) classifiers from the literature. It is worth noting that recent work on the probabilistic $k-$nn has offered a novel MCMC inference scheme based on *perfect sampling* (Cucala et al. 2009) that could potentially offer classification improvements. The aim here is to obtain a first picture of classification accuracy and computational processing time from the proposed variational approximation. At this stage, the problems considered have a single feature space

or information source and hence there is no need to employ the multiple kernel learning (MKL) part of the approximation. Extensive experimentation with multiple feature spaces will be reported in Chapter 7 where the various combination rules and the full MKL variational methodology are assessed on important bioinformatics and automatic currency validation problems.

In Table 4.2 the characteristics of the employed datasets are described. The experiments consider three standard kernel functions and the hyper-parameters were set to uninformative values. The Gaussian (RBF) kernel parameters were fixed to $1/D$ where D the number of attributes. We employ an RBF (VB RBF), a 2^{nd} order polynomial (VB P) and a linear kernel (VB L) with the variational approximation and report 10-fold cross-validated (CV) error percentages, in Table 4.3, and CPU times, in Table 4.4. Bold fonts denote the overall top classification accuracy across methods, which in most cases is not statistically significant due to the large CV variance observed.

Data set	N	C	D
Balance	625	3	4
Crabs	200	4	5
Glass	214	6	9
Iris	150	3	4
Soybean	48	4	35
Vehicle	846	4	18
Wine	178	3	13

Table 4.2: Multinomial UCI datasets. N, C, D are respectively the number of samples, classes and attributes in each dataset.

4.7 Discussion

In this Chapter a variational Bayes approximation for probabilistic multiple kernel learning (VBpMKL) was proposed and examined with respect to resulting posterior distributions and decision boundaries, classification accuracy and computational processing times. A direct comparison with the full MCMC Gibbs sampling solution demonstrates the over-confidence of the approximate posteriors that are typically narrower as they underestimate the covariance structure of the true posterior distribution. Preliminary results from the variational approximation on multinomial UCI datasets demonstrate competing classification

Data set	**VB RBF**	**VB L**	**VB P**	***K*-nn**	**P*K*-nn**
Balance	8.8 ± 3.6	12.2 ± 4.2	$\mathbf{7.0 \pm 3.3}$	11.5 ± 3.0	10.2 ± 3.0
Crabs	23.5 ± 11.3	$\mathbf{13.5 \pm 8.2}$	21.5 ± 9.1	15.0 ± 8.8	19.5 ± 6.8
Glass	27.9 ± 10.1	35.8 ± 11.8	28.4 ± 8.9	29.9 ± 9.2	$\mathbf{26.7 \pm 8.8}$
Iris	$\mathbf{2.7 \pm 5.6}$	11.3 ± 9.9	4.7 ± 6.3	5.3 ± 5.2	4.0 ± 5.6
Soybean	6.5 ± 10.5	6 ± 9.7	$\mathbf{4 \pm 8.4}$	14.5 ± 16.7	4.5 ± 9.6
Vehicle	$\mathbf{25.6 \pm 4.0}$	29.6 ± 3.3	26 ± 6.1	36.3 ± 5.2	37.2 ± 4.5
Wine	4.5 ± 5.1	2.8 ± 4.7	$\mathbf{1.1 \pm 2.3}$	3.9 ± 3.8	3.4 ± 2.9

Table 4.3: 10-fold cross-validated error percentages (mean±std) on standard UCI multinomial datasets. Top performance (not always statistically significant) in **bold**.

Data set	Balance	Crabs	Glass	Iris	Soybean	Vehicle	Wine
CPU time (s)	2,285	270	380	89	19	3,420	105

Table 4.4: Running times (seconds) for computing 10-fold cross-validation results with unoptimised Matlab® codes.

performances while retaining reasonable computational processing times when compared with CPU times reported in (Manocha and Girolami 2007) for nearest neighbour methods.

The main goal of reducing the computational burden of the MCMC methodology offered in Chapter 3 has been achieved by offering a smaller computational complexity of $\mathcal{O}(TCN^3)$ while retaining similar levels of classification accuracy and resulting (approximate) posterior distributions. This will be further demonstrated in Chapters 7 and 6 where the MCMC Gibbs sampling and the VBpMKL methods are applied on large scale bioinformatics, hand-written numeral recognition and automatic currency validation problems. In these problems, the various multiple kernel learning rules under the variational approximation will also be assessed. Finally, the memory requirements remain dominated by the multiple N^2 matrices required and the algorithms still require multiple N^3 inversions. These issues will be addressed in the following Chapter where *sparse* deterministic approximate methodology based on point-estimates will now be considered.

Chapter 5

MAP Estimators and mRVMs

In the previous chapters we have described an accurate (exact inference to the limit of infinite samples) MCMC inference methodology for multiple kernel learning which is computationally expensive in both processing and memory requirements. To address the processing burden, variational Bayesian methodology was proposed as a deterministic approximation that still retains the Bayesian benefits of (approximate) posterior distributions over parameters. However, despite the much improved computational complexity, the large memory requirements $\mathcal{O}(SN^2+3NC)$ and the typical dominant $\mathcal{O}(CN^3)$ scaling of multinomial kernel methods still present important and unsolved restrictions.

In order to address these issues and offer an efficient alternative, we resort to further deterministic approximations and sparse solutions via sparsity inducing prior formulations. In this chapter[1], such a deterministic maximum-a-posteriori (MAP) approximation is first introduced, which leads to a generalisation of the Relevance Vector Machine (Tipping 1999, Tipping 2001) to the multiclass multi-kernel setting. The MAP approximation can in principle be less accurate than the full MCMC solution and the variational Bayes approximation as it employes point-estimates instead of parameter distributions but the benefits are improved processing times and memory requirements. Throughout this chapter we concentrate on the standard kernel combination case of the convex linear summation rule that has been previously introduced.

First, the resulting MAP estimator of the model and also associated expectation - maximisation (EM) update schemes are described in detail. Then, two new

[1]Parts of this work have already appeared in (Damoulas et al. 2008, Ying et al. 2009) and have been submitted for publication (Psorakis et al. 2010)

formulations for multiclass multi-kernel relevance vector machines (mRVMs) are presented that explicitly lead to sparse solutions, both in samples and in number of kernels. This enables their application to large-scale multi-feature multinomial classification problems where there is an abundance of training samples, classes and feature spaces. Finally, the chapter concludes with experimental studies for convergence, performance and resulting sparsity on standard UCI datasets.

5.1 MAP Estimation and EM Update Schemes

Consider the regression nature of the multinomial probit model. In the previous chapter we have seen how the introduction of the auxiliary variables $\mathbf{Y}$ offers a closed form posterior distribution for the regression parameters $\mathbf{W}$. This is not possible with the standard softmax likelihood approach where the parameter posterior would be directly dependent on the class labels $\mathbf{t}$ and further approximations such as the Laplace (saddle-point approximation) are needed. Having that closed form posterior allows for a straightforward MAP estimator of the regression coefficients:

$$\text{M-STEP} \qquad \hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}\, p(\mathbf{W}|\mathbf{Y}, \mathbf{A}, \mathbf{K})$$

where again $\mathbf{K}$ is the composite kernel conditioned on specific kernel parameters $\boldsymbol{\Theta}, \boldsymbol{\beta}$ and $\mathbf{A}$ the scales of the zero-mean normally distributed parameters $\mathbf{W}$. The posterior is a multivariate Gaussian distribution and hence the MAP estimate of the regression coefficients is the mean of the posterior distribution, see [Appendix], given by:

$$\hat{\mathbf{w}}_c = \left(\mathbf{K}\mathbf{K}^\intercal + \mathbf{A}_c\right)^{-1} \mathbf{K}\mathbf{y}_c \tag{5.1}$$

The next step is to consider the auxiliary variable posterior distribution which, as we have seen in the previous chapter's Equation 3.23, is a product of N $C-$dimensional conically truncated Gaussians. As such, a MAP estimate and also an expectation step can be considered as:

$$\begin{cases} \text{M-STEP} & \hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmax}}\, p\left(\mathbf{Y}|\mathbf{W}, \mathbf{K}, \mathbf{t}\right) \\ \text{E-STEP} & \tilde{\mathbf{Y}} = \mathbb{E}_{p(\mathbf{Y}|\mathbf{W},\mathbf{K},\mathbf{t})}\{\mathbf{Y}\} \end{cases} \tag{5.2}$$

where the MAP estimate for a sample n that belongs to class i is given by

$\hat{y}_{ni} = \hat{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n$ and for $c \neq i$ is either the mean of the right truncated univariate normal (when the truncation is greater than the mean) or the truncation point itself (when the truncation is less than the mean). Hence it can be described as:

$$\begin{cases} \hat{y}_{nc} = \hat{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n & \text{if } \hat{y}_{ni} \geq \hat{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n \\ \hat{y}_{nc} = \hat{y}_{ni} & \text{if } \hat{y}_{ni} \leq \hat{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n \end{cases} \tag{5.3}$$

For the E-step, the posterior expectations of the auxiliary variable according to the object's class membership i are derived analytically in [Appendix]. The expectation of y_{nc} for all $c \neq i$ and again conditioning on specific kernel parameters $\boldsymbol{\Theta}, \boldsymbol{\beta}$, is given by:

$$\widetilde{y}_{nc} \leftarrow \mathbf{k}_n\hat{\mathbf{w}}_c - \frac{\mathbb{E}_{p(u)}\{\mathcal{N}_u\left(\hat{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n - \hat{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n, 1\right)\Phi_u^{n,i,c}\}}{\mathbb{E}_{p(u)}\{\Phi\left(u + \hat{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n - \hat{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n\right)\Phi_u^{n,i,c}\}} \tag{5.4}$$

and the expectation for the i^{th} class as:

$$\widetilde{y}_{ni} \leftarrow \hat{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n - \left(\sum_{j\neq i}\widetilde{y}_{nj} - \hat{\mathbf{w}}_j^{\mathsf{T}}\mathbf{k}_n\right) \tag{5.5}$$

where Φ is the cumulative distribution function and $\Phi_u^{n,i,c}$ is defined as:

$$\Phi_u^{n,i,c} = \prod_{j\neq i,c}\Phi\left(u + \hat{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n - \hat{\mathbf{w}}_j^{\mathsf{T}}\mathbf{k}_n\right) \tag{5.6}$$

Having derived and described the maximisation and expectation steps for the regression coefficients and auxiliary variables, we turn our attention to the scales $\mathbf{A}$ and the kernel parameters $\boldsymbol{\beta}$. Distinct M-step and E-step procedures can be derived for the scales given that the mode and mean of a Gamma distribution are different:

$$\begin{cases} \textsc{m-step} & \hat{\mathbf{A}} = \underset{\mathbf{A}}{\text{argmax}}\; p\left(\mathbf{A}|\mathbf{W}, \tau, \upsilon\right) \\ \textsc{e-step} & \widetilde{\mathbf{A}} = \mathbb{E}_{p(\mathbf{A}|\mathbf{W},\tau,\upsilon)}\{\mathbf{A}\} \end{cases} \tag{5.7}$$

As we have seen in the previous chapter and analytically in [Appendix], a closed form posterior distribution $p(\mathbf{A}|\mathbf{W}, \tau, \upsilon)$ is available for the scales:

$$\alpha_{nc}|w_{nc}, \tau, \upsilon \sim \mathcal{G}_{\alpha_{nc}}(\frac{1}{2} + \tau, \frac{w_{nc}^2}{2} + \upsilon) \tag{5.8}$$

which results in the following updates:

$$\begin{cases} \hat{\alpha}_{nc} = \dfrac{2\tau - 1}{w_{nc}^2 + 2\upsilon} \\ \\ \widetilde{\alpha}_{nc} = \dfrac{2\tau + 1}{w_{nc}^2 + 2\upsilon} \end{cases} \tag{5.9}$$

Finally, the inference schemes are completed with a MAP estimate for the kernel combinatorial parameters $\boldsymbol{\beta}$, assuming the standard case of the convex linear rule and a uniform Dirichlet prior distribution over the simplex. Considering the log of the joint likelihood it is easy to see (Appendix C) that maximisation with respect to $\boldsymbol{\beta}$ leads to the following linearly constrained quadratic program (QP):

$$\begin{cases} \hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\Omega}\boldsymbol{\beta} - \boldsymbol{\beta}^{\mathsf{T}}\mathbf{f} \\ \text{s.t} \quad \beta_i \geq 0 \; \forall \; i \;\; \text{and} \;\; \sum_{s=1}^{S} \beta_s = 1 \end{cases} \tag{5.10}$$

where $\boldsymbol{\Omega}_{ij} = \sum_{n,c}^{N,C} \mathbf{w}_c \mathbf{k}_{i\,n}^{\mathsf{T}} \mathbf{k}_{j_n} \mathbf{w}_c^{\mathsf{T}}$ is an $S \times S$ matrix, $f_i = \sum_{n,c}^{N,C} \mathbf{w}_c \mathbf{k}_{i\,n}^{\mathsf{T}} \tilde{y}_{nc}$ and $\mathbf{k}_{j_n}$ is the n^{th} vector of the j^{th} base kernel.

In Algorithms 4 and 5 pseudo-algorithmic formats of the resulting MAP and EM inference schemes are given. It is worth noting that preliminary initialisation of the auxiliary variables $\mathbf{Y}$ can be based on following the target labels $\mathbf{t}$ (remember the probit multinomial link is $t_n = i \iff y_{ni} > y_{nj} \;\; \forall \; j \neq i$ with $j, i \in \{1, \ldots, C\}$).

It is worth noting that the MAP estimation is over the augmented parameter space of both the regression coefficients and the auxiliary variables. Hence it is different from a standard MAP estimate, that would be over only the regression coefficients of the "reduced" model as given in Chapter 3: Figure 3.2, that does not employ auxiliary variables. However, the MAP estimate of the reduced model is unobtainable as there is no closed form posterior and thus inefficient Metropolis sampling would be required.

Having described the training phase of the standard MAP and EM schemes, attention is turned to the testing or predictive phase of the classifier. The resulting predictive likelihood for an unseen sample $\mathbf{x}_*^s$ embedded into S base

Algorithm 4 MAP estimator

1: Initialisation $(\tau, \upsilon, \boldsymbol{\beta}, \boldsymbol{A}_c, \mathbf{W})$
2: Sample $\mathbf{Y} \in \mathbb{R}^{C \times N}$ to follow target $\mathbf{t}$.
3: **while** Iterations $<$ max & Convergence $>$ Threshold **do**
4: M-Step for $\mathbf{W}$: Equation 5.1
5: M-Step for $\mathbf{Y}$: Equation 5.3
6: M-Step for $\boldsymbol{A}_c$: Equation 5.9
7: QP program for $\boldsymbol{\beta}$: Equation 5.10
8: **end while**

Algorithm 5 Expectation Maximisation scheme

1: Initialisation $(\tau, \upsilon, \boldsymbol{\beta}, \boldsymbol{A}_c, \mathbf{W})$
2: Sample $\mathbf{Y} \in \mathbb{R}^{C \times N}$ to follow target $\mathbf{t}$.
3: **while** Iterations $<$ max & Convergence $>$ Threshold **do**
4: M-Step for $\mathbf{W}$: Equation 5.1
5: E-Step for $\mathbf{Y}$: Equations 5.4 and 5.5
6: E-Step for $\boldsymbol{A}_c$: Equation 5.9
7: QP program for $\boldsymbol{\beta}$: Eq. 5.10
8: **end while**

kernels $\mathbf{k}_*^s$ is given by

$$p\left(t_* = i | \mathbf{x}_*^s, \mathbf{X}, \mathbf{t}\right) = \int \delta^{t_*} \mathcal{N}_{\mathbf{y}_*}\left(\mathbf{k}_*^{\hat{\boldsymbol{\beta}}} \hat{\mathbf{W}}, \mathbf{I}\right) d\mathbf{y}_* \tag{5.11}$$

$$= \mathbb{E}_{p(u)}\left\{\prod_{j \neq i} \Phi\left(u + \mathbf{k}_*^{\hat{\boldsymbol{\beta}}}\left(\hat{\mathbf{w}}_i - \hat{\mathbf{w}}_j\right)\right)\right\}. \tag{5.12}$$

Here the expectation $\mathbb{E}_{p(u)}$ is taken, in the usual manner, with respect to the standardised normal distribution $p(u) = \mathcal{N}(0, 1)$. Either the Monte Carlo estimate or the Gauss-Hermite quadrature from Chapter 3 can be employed to approximate the likelihood.

5.2 Sparsity and Relevance Vector Machines

All the models we have considered so far utilise the whole set of training samples when predicting the class of an unknown test sample. This is typical of many standard supervised learning algorithms such as K-nearest neighbours, naive bayes and Gaussian classifiers, and it leads to computational and memory problems as the training size increases. To deal with that, many sparse or sparsity-inducing models have been proposed in the past that utilise only a

selected subset of these samples for the final regression or classification solution. Within the statistics nomenclature such models are known as shrinkage and selection methods.

The most notable of such models are the Lasso (Tibshirani 1996), the Support Vector Machine (Vapnik and Chervonenkis 1964) and the Relevance Vector Machine (RVM) (Tipping 1999) that have been briefly reviewed in Chapter 2. In the following section we introduce such sparsity-inducing models based on the MAP and EM schemes considered so far. We employ a sparsity inducing prior or a novel *fast type-II Maximum Likelihood* procedure and offer a generalisation of the RVM to multiclass and multi-kernel problems.

5.3 Multiclass Multi-kernel Relevance Vector Machines

In the Bayesian paradigm, the functional form analogous to SVMs is the relevance vector machine (RVM) (Tipping 2001) which employs sparse Bayesian learning via an appropriate prior formulation. Maximisation of the marginal likelihood, a type-II maximum likelihood (ML) expression, gives sparse solutions which utilise only a subset of the basis functions: the *relevance vectors*. Compared to an SVM, there are relatively few relevance vectors and they are typically not close to the decision boundary but prototypical (Tipping 1999). However, until now, the multiclass adaptation of RVMs was problematic (Tipping 2001) due to the bad scaling of the type-II ML procedure with respect to C, the number of classes. Furthermore, although in regression problems the RVM offers a closed form posterior distribution for the regression parameters, in classification the employment of the softmax likelihood imposes the need for the Laplace or other saddle-point approximations.

In this section we describe two multiclass multi-kernel RVM methods which are able to address multi-kernel learning while producing both sample-wise and kernel-wise sparse solutions. In contrast to SVM approaches, they utilise the probabilistic framework of Bayesian inference, avoid pre-computation of margin trade-off parameters or cross-validation procedures and are able to produce posterior probabilities of class memberships without using ad-hoc post-processing methods.

In contrast with the original RVM (Tipping 1999, Tipping 2001), the pro-

posed methods employ the multinomial probit likelihood (Albert and Chib 1993) which, as we have seen, results in multiclass classifiers via the introduction of auxiliary variables. In one case we propose a multiclass extension of the fast type-II ML procedure in (Tipping and Faul 2003, Faul and Tipping 2002) and in the second case we *explicitly* employ a flat prior for the hyper-parameters that control the sparsity of the resulting model. In both cases, inference on the kernel combinatorial coefficients is enabled via a constrained QP procedure and an efficient expectation-maximisation (EM) scheme is adopted.

The two algorithms are suitable for different application scenarios based on the size of the initial training set and nature (streaming or not) of the data. As it will be further discussed, the first method (mRVM_1) is a "bottom-up" approach that starts with an empty set of basis functions and sequentially builds the model by adding or deleting such samples based on the marginal likelihood and utilising a novel fast multiclass type-II ML procedure. The second method (mRVM_2) is a "top-down" approach where the algorithm starts with the full model and prunes out basis functions that have insignificant contribution to model fitting. This is enforced by the implicit sparsity-enforcing prior which, as in the original RVM, is a Student-t distribution.

5.4 Model Formulation

Following the previously described settings we consider S feature spaces in which a D^s-dimensional sample $\mathbf{x}_n^s$ has an associated label $t_n \in \{1, \ldots, C\}$. Kernel substitution is applied in each feature space resulting in S base kernels $\mathbf{K}^s \in \mathbb{R}^{N \times N}$ that are combined into our composite kernel. Conditioning on specific[2] kernel parameters $\boldsymbol{\Theta}^s$ and assuming a uniform Dirichlet prior on the combination parameters $\boldsymbol{\beta}$ while adopting the multinomial probit likelihood results in the plates diagram in Figure 5.1.

The hierarchical Bayesian framework with a conjugate and *flat* ($\tau, \upsilon \to 0$) Gamma hyper-prior on the scale of the parameters' Gaussian prior results in an implicit Student-t distribution on the parameters (Tipping 2001) and therefore encourages sparsity. Together with appropriate Type-II ML inference of the scales $\mathbf{A}$, these two developments are the main focus of the RVM approaches and play an important role in both mRVM algorithms that are now proposed.

[2]Joint feature and sample sparsity methods are currently under research.

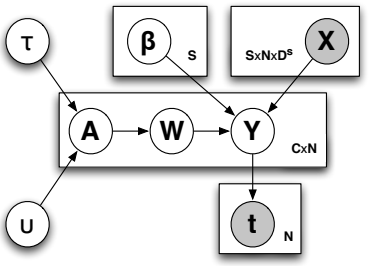

Figure 5.1: Plates diagram of the model.

5.4.1 mRVM$_1$

The first multiclass multi-kernel RVM we consider is based on the "constructive" variant of RVMs (Tipping and Faul 2003, Faul and Tipping 2002) which employs a fast type-II ML procedure. The maximisation of the marginal likelihood

$$p(\mathbf{Y}|\mathbf{K},\mathbf{A}) = \int p(\mathbf{Y}|\mathbf{K},\mathbf{W})p(\mathbf{W}|\mathbf{A})d\mathbf{W} \tag{5.13}$$

with respect to $\mathbf{A}$, and $\mathbf{K}$ conditioned on combination parameters $\boldsymbol{\beta}$, results in a criterion to either add a sample, delete or update its associated hyper-parameter α_n. Therefore, the model can start with a single sample and proceed in a constructive manner as detailed below. The (log) multiclass marginal likelihood is given by

$$\begin{aligned}\mathcal{L}(\mathbf{A}) &= \log p(\mathbf{Y}|\mathbf{K},\mathbf{A}) = \log \textstyle\int_{-\infty}^{+\infty} p(\mathbf{Y}|\mathbf{K},\mathbf{W})p(\mathbf{W}|\mathbf{A})d\mathbf{W} \\ &= \sum_{c=1}^{C} -\frac{1}{2}[N\log 2\pi + \log|\boldsymbol{\mathcal{C}}| + \mathbf{y}_c^{\mathsf{T}}\boldsymbol{\mathcal{C}}^{-1}\mathbf{y}_c]\end{aligned}$$

where $\boldsymbol{\mathcal{C}} = \mathbf{I} + \mathbf{K}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{K}$ and $\mathbf{A}$ is defined as a diagonal matrix with non-zero elements as $(\alpha_1,\ldots,\alpha_N)$. Here we have made the necessary assumption (allowing a well-behaved and differentiable marginal likelihood) that a common scale α_n is shared across classes for every sample n. This allows an effective type-II ML scheme based on the original binary scheme proposed by Tipping and Faul (Tipping and Faul 2003, Faul and Tipping 2002) and it couples the regression coefficients across classes to express the intuition that adding or removing a basis should be done on the joint evidence and support from all model classes.

The idea now is to decompose the marginal likelihood into contributing terms from each included sample (basis function). This gives the opportunity, as it will

be shown below, for a constructive methodology by maximising the marginal likelihood with respect to an individual sample. The decomposition of terms in $\boldsymbol{\mathcal{C}}$ follows exactly as in Tipping and Faul (2003) and Faul and Tipping (2002):

$$|\boldsymbol{\mathcal{C}}| = |\boldsymbol{\mathcal{C}}_{-i}| \; |1 + \alpha_i^{-1} \mathbf{k}_i^{\mathsf{T}} \boldsymbol{\mathcal{C}}_{-i}^{-1} \mathbf{k}_i|, \tag{5.14}$$

and

$$\boldsymbol{\mathcal{C}}^{-1} = \boldsymbol{\mathcal{C}}_{-i}^{-1} - \frac{\boldsymbol{\mathcal{C}}_{-i}^{-1} \mathbf{k}_i \mathbf{k}_i^{\mathsf{T}} \boldsymbol{\mathcal{C}}_{-i}^{-1}}{\alpha_i + \mathbf{k}_i^{\mathsf{T}} \boldsymbol{\mathcal{C}}_i^{-1} \mathbf{k}_i}. \tag{5.15}$$

where $\boldsymbol{\mathcal{C}}_{-i}^{-1}$ signifies the inverse of $\boldsymbol{\mathcal{C}}$ with the i^{th} basis removed. Hence the (log) marginal likelihood can also be decomposed as

$$\begin{aligned}
\mathcal{L}(\mathbf{A}) =& \sum_{c=1}^{C} -\frac{1}{2}[N \log 2\pi + \log |\boldsymbol{\mathcal{C}}_{-i}| + \mathbf{y}_c^{\mathsf{T}} \boldsymbol{\mathcal{C}}_{-i}^{-1} \mathbf{y}_c \\
& - \log \alpha_i + \log(\alpha_i + \mathbf{k}_i^{\mathsf{T}} \boldsymbol{\mathcal{C}}_{-i}^{-1} \mathbf{k}_i) - \frac{\left(\mathbf{k}_i^{\mathsf{T}} \boldsymbol{\mathcal{C}}_{-i}^{-1} \mathbf{y}_c\right)^2}{\alpha_i + \mathbf{k}_i^{\mathsf{T}} \boldsymbol{\mathcal{C}}_{-i}^{-1} \mathbf{k}_i}] \\
=& \mathcal{L}(\mathbf{A}_{-i}) + \sum_{c=1}^{C} \frac{1}{2} \left[\log \alpha_i - log(\alpha_i + s_i) + \frac{q_{ci}^2}{\alpha_i + s_i} \right] \\
=& \mathcal{L}(\mathbf{A}_{-i}) + l(\alpha_i)
\end{aligned} \tag{5.16}$$

where we follow Tipping and Faul (2003) in defining the "sparsity factor" s_i and also the new *multiclass* "quality factor" q_{ci} as:

$$s_i = \mathbf{k}_i^{\mathsf{T}} \boldsymbol{\mathcal{C}}_{-i}^{-1} \mathbf{k}_i \quad \text{and} \quad q_{ci} = \mathbf{k}_i^{\mathsf{T}} \boldsymbol{\mathcal{C}}_{-i}^{-1} \mathbf{y}_c. \tag{5.17}$$

It is worth noting that although the sparsity factor s_i can still be seen as a measure of overlap between sample $\mathbf{k}_i$ and the ones already included, the quality factor q_{ci} is now class-specific and it measures how good the sample is in helping to describe a specific class. The significant difference with the binary maximum solution of Tipping and Faul (2003) is that the quality of a sample is now assessed across classes through this novel multiclass formulation.

Having decomposed the marginal likelihood into sample specific contributions we can seek the maximum with respect to an α_i. The only term that is a function of α_i is $l(\alpha_i)$ and the only difference, in that term, with its binary definition is the extra summation over classes and the multiclass factor q_{ci}. The derivative

is:

$$\frac{\partial \mathcal{L}(\mathbf{A})}{\partial \alpha_i} = \frac{\partial l(\alpha_i)}{\partial \alpha_i} = \sum_{c=1}^{C} \frac{1}{2} \left[\frac{1}{\alpha_i} - \frac{1}{\alpha_i + s_i} - \frac{q_{ci}^2}{(\alpha_i + s_i)^2} \right] = \sum_{c=1}^{C} \frac{\alpha_i^{-1} s_i^2 - (q_{ci}^2 - s_i)}{2(\alpha_i + s_i)^2} \tag{5.18}$$

and by setting Equation 5.18 to zero we obtain the following stationary points:

$$\alpha_i = \frac{C s_i^2}{\sum_{c=1}^{C} q_{ci}^2 - C s_i}, \quad \text{if} \quad \sum_{c=1}^{C} q_{ci}^2 > C s_i \tag{5.19}$$

$$\alpha_i = \infty, \quad \text{if} \quad \sum_{c=1}^{C} q_{ci}^2 \leq C s_i \tag{5.20}$$

Following the same analysis as in (Faul and Tipping 2002) it is straightforward to show that the second derivative on the stationary point in Equation 5.19 is always negative and hence the solution is a unique maximum for the specific condition. For the second stationarity in Equation 5.20 as $\alpha_i \to \infty$ the sign of the gradient is given by $-(\sum_{c=1}^{C} q_{ci} - C s_i)$ and hence when $\sum_{c=1}^{C} q_{ci}^2 \leq C s_i$ this point is now the unique maximum.

Therefore, the maximisation of the marginal likelihood leads to the possible inclusion of a sample (with associated scale α_i by Equation 5.19), deletion of one (scale α_i by Equation 5.20) or updating its corresponding scale (α_i by Equation 5.19). Hence a constructive way of model building is available for the multiclass case and with little additional overhead to the original binary procedure as it only requires an extra summation over the 'quality factors' q_{ci}.

Furthermore, this novel multiclass formulation of the fast type-II ML procedure can be directly used for multinomial regression problems with $\mathbf{Y}$ as the continuous real-valued output. Thus, generalisations of recent work that have adopted the binomial procedure (Schmolck and Everson 2007, Tzikas et al. 2008, Tzikas et al. 2009) can be readily derived by adopting the proposed sequential multinomial scheme which is algorithmically described in Algorithm 6.

Returning back to our classification framework, the modified *constructive* M-step for the estimate $\hat{\mathbf{W}}$, conditioned on an E-step estimate of $\mathbf{Y}$ for example, is given by:

$$\hat{\mathbf{W}}_{\circ} = \left(\mathbf{K}_{\circ} \mathbf{K}_{\circ}^{\mathsf{T}} + \mathbf{A}_{\circ} \right)^{-1} \mathbf{K}_{\circ} \tilde{\mathbf{Y}}, \tag{5.21}$$

Algorithm 6 mRVM$_1$ and the Fast Multi-class Type-II ML procedure

Initialise $\mathbf{Y}$ to follow target labels $\mathbf{t}$, set all $\alpha_i = \infty$.
Initialise model with a single sample $\mathbf{k}_i$, setting $\alpha_i = \frac{\|\mathbf{k}_i\|^2}{\sum_{c=1}^{C} \|\mathbf{k}_i^{\mathsf{T}} \mathbf{y}_c\|^2 / C \|\mathbf{k}_i\|^2 - 1}$ from Equation 5.19.
while Convergence Criteria Unsatisfied **do**
  Select candidate sample $\mathbf{k}_i$.
  if $\sum_{c=1}^{C} q_{ci}^2 > C s_i$ and $\alpha_i < \infty$ **then**
    Update α_i from Equation 5.19 (sample already in the model).
  else if $\sum_{c=1}^{C} q_{ci}^2 > C s_i$ and $\alpha_i = \infty$ **then**
    Set α_i from Equation 5.19 (sample added in the model).
  else if $\sum_{c=1}^{C} q_{ci}^2 \leq C s_i$ and $\alpha_i < \infty$ **then**
    Set $\alpha_i = \infty$ from Equation 5.20 (sample deleted from the model).
  end if
  M-Step for $\hat{\mathbf{W}}_\circ$: Equation 5.21.
  QP program for $\boldsymbol{\beta}$: Equation 5.10 for reduced row rank $\hat{\mathbf{W}}_\circ, \mathbf{K}_\circ$.
  E-Step for $\mathbf{Y}$: Equation 5.3 or 5.4 and 5.5 for reduced row rank $\hat{\mathbf{W}}_\circ, \mathbf{K}_\circ$.
end while

where $\mathbf{K}_\circ \in \mathbb{R}^{M \times N}$ and $\mathbf{A}_\circ \in \mathbb{R}^{M \times M}$ are the reduced (composite) kernel and diagonal scale matrix respectively, which now utilise only M samples and corresponding scales for model fitting, with $M << N$. As the algorithm progresses, the selected samples are added to the initially empty model that upon convergence describes the whole training set while utilising a typically small fraction of it. The nature of this constructive procedure allows applications to large-scale datasets and scenarios where utilising the whole training set is prohibitive.

The E-step or M-step of the auxiliary variables follows directly from Equations 5.3 and 5.4, 5.5 by simply accommodating the reduced rank matrices of the regression coefficients and the composite kernel. Similarly, the kernel combination parameters $\boldsymbol{\beta}$ follow the standard quadratic program in Equation 5.10 with the sparse representations $\hat{\mathbf{W}}_\circ$ and $\mathbf{K}_\circ$.

5.4.2 Computational Efficiency for mRVM$_1$

The fast multiclass type-II ML procedure is based on sequential computation of the "sparsity" and *multiclass* "quality" factors described in Equation 5.17 which in turn require the inversion of matrix C_{-i} each time the model is updated (new samples have been included). This inversion, together with the one in Equation 5.21 govern the computational complexity of the algorithm as $\mathcal{O}(P2M^3)$ where

M the number of employed samples out of total N and assuming P proposals of samples to be considered for inclusion. Typically $M << N$ and hence the cubic contribution is not restricting but we note that by simple matrix identities and manipulations, that are now given based on (Tipping and Faul 2003), this complexity can be reduced by half to $\mathcal{O}(PM^3)$.

Assuming M samples have been included in the model, we follow Tipping and Faul (2003) in defining the (modified for our model) quantities S_m and Q_m as:

$$S_m = \mathbf{k}_m^{\mathsf{T}} \boldsymbol{\mathcal{C}}^{-1} \mathbf{k}_m \quad \text{and} \quad Q_{cm} = \mathbf{k}_m^{\mathsf{T}} \boldsymbol{\mathcal{C}}^{-1} \mathbf{y}_c \tag{5.22}$$

which, if the sample $\mathbf{k}_m$ is not included in the model (hence $\boldsymbol{\mathcal{C}}^{-1}$ is in fact $\boldsymbol{\mathcal{C}}_{-m}^{-1}$), correspond to the quantities of interest; i.e. $s_m = S_m$ and $q_{cm} = Q_{cm}$. On the other hand if the sample is included in the current model then the true sparsity and quality factors are given by:

$$s_m = \frac{\alpha_m S_m}{\alpha_m - S_m} \quad \text{and} \quad q_{cm} = \frac{\alpha_m Q_{cm}}{\alpha_m - S_m} \tag{5.23}$$

Therefore our only interest now is to maintain values for S_m and Q_{cm} $\forall$ $c = \{1, \ldots, C\}$. By decomposing these quantities, bearing in mind that $\boldsymbol{\mathcal{C}} = \mathbf{I} + \mathbf{K}_\circ^{\mathsf{T}} \mathbf{A}_\circ^{-1} \mathbf{K}_\circ$, and utilising the Woodbury identity we have:

$$S_m = \mathbf{k}_m^{\mathsf{T}} \mathbf{k}_m - \mathbf{k}_m^{\mathsf{T}} \mathbf{K}_\circ^{\mathsf{T}} \left(\mathbf{K}_\circ \mathbf{K}_\circ^{\mathsf{T}} + \mathbf{A}_\circ \right)^{-1} \mathbf{K}_\circ \mathbf{k}_m \tag{5.24}$$

$$Q_{cm} = \mathbf{k}_m^{\mathsf{T}} \mathbf{y}_c - \mathbf{k}_m^{\mathsf{T}} \mathbf{K}_\circ^{\mathsf{T}} \left(\mathbf{K}_\circ \mathbf{K}_\circ^{\mathsf{T}} + \mathbf{A}_\circ \right)^{-1} \mathbf{K}_\circ \mathbf{y}_c \tag{5.25}$$

The required inversion now for computation of sparsity and quality factors is the same inversion required for the MAP estimate of the regression coefficients $\hat{\mathbf{W}}_\circ$ in Equation 5.21. Hence, only a single inversion is needed overall, leading to a reduced computational complexity and an efficient algorithm.

5.4.3 Informative Sample Selection for mRVM$_1$

One unresolved issue so far with this constructive-type sparse methodology is how to propose new samples to be examined for possible inclusion in the model. A random proposal can be employed but it is obviously sub-optimal and will

lead to a slow convergence of the algorithm as the "relevance vectors" might be proposed very late in training.

An alternative informative sample proposal strategy for mRVM_1 can be derived based on the binary one of Tipping and Faul (2003). Revisiting Equations 5.19, 5.20 and defining $\theta_i = \sum_{c=1}^{C} q_{ci}^2 - Cs_i$ as the *contribution value* of sample i under the current model that includes M "active" samples, leads to the criterion of proposing the new sample that has the highest *positive* contribution value from the $\mathcal{I}$ "inactive" samples that are not currently included in the model.

In the case where all the inactive samples have negative *contribution value* and all the active samples have positive ones (hence the algorithm is close to convergence) the proposed sample is randomly selected from the active set therefore updating its corresponding scale. The informative selection procedure is summarised below in Algorithm 7.

Algorithm 7 Informative Sample Selection: mRVM_1

if $\exists \;\; \theta_i > 0 \;\;$ with $\;\; i \in \mathcal{I}$ **then**
    Select sample i with $\theta_i \geq \theta_j \;\; \forall \;\; i, j \in \mathcal{I}$ (include)
else if $\exists \;\; \theta_i \leq 0 \;\;$ with $\;\; i \in M$ **then**
    Select sample i with $\theta_i \leq \theta_j \;\; \forall \;\; i, j \in M$ (delete)
else
    Select random sample $i \in M$ (update α_i)
end if

In Figure 5.2 the informative sample selection procedure on the Neal dataset is compared against randomly selecting samples. As it can be seen, the proposed procedure typically allows for faster convergence and avoids local minima solutions with respect to sparsity whereas randomly selecting samples proves to be less efficient, as expected.

The procedure requires storage and update of the contributing values θ_i for *all* the samples ($N = (\mathcal{I} \cup M)$), active or inactive, and this comes to some additional overhead (only matrix multiplication and not inversions) as the corresponding values for S_m and Q_{cm} from Equations 5.24 are required for all the N samples. The benefits of informative sample selection counterbalance the additional overhead as they lead to good classification performances and appropriate (in the sense of stable and optimum) convergence measures (Psorakis et al. 2010) that will be described in the next sections.

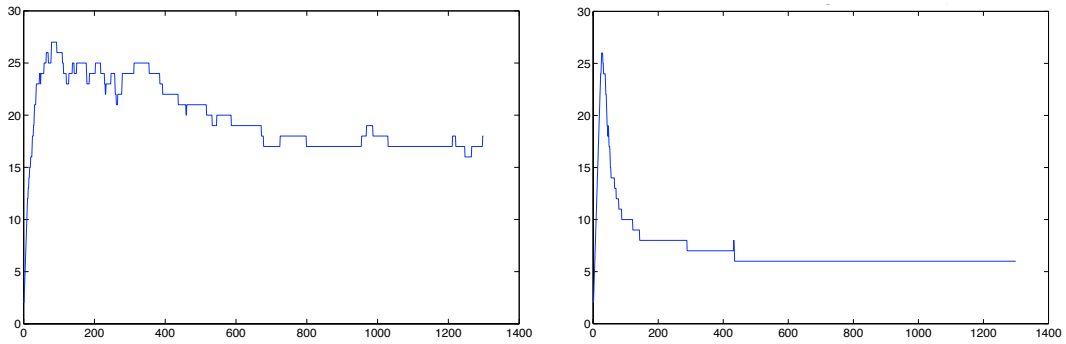

Figure 5.2: Neal dataset. Left: Uninformative sample selection. Right: informative sample selection

5.4.4 Initialisation and Convergence for mRVM$_1$

Because of the point-estimation nature of the algorithms introduced in this chapter, appropriate initialisation plays an important role for efficient training, avoiding numerical problems and eventual convergence. As stated in the pseudo-algorithmic descriptions, initialisation of the auxiliary variables should follow the target labels and the multinomial probit link $y_{ni} \geq y_{nj} \quad \forall \quad j \neq i$ if $t_n = i$. This enforces an initialisation close to the local maximum of the joint likelihood and speeds up convergence. This is illustrated in Figure 5.3 where a random and an "aligned" initialisation are compared on a binary classification problem (counterfeit detection on US50BA, Chapter 7). The recovery of the correct alignment for the randomly initialised auxiliary variables is still very fast for the considered binary problem but an aligned initialisation has a better convergence behaviour especially for large multinomial problems.

In the specific case of mRVM$_1$ where the type-II ML procedure is employed, additional convergence criteria are proposed that were found to provide better solutions in both sparsity and accuracy performance measures, as observed in (Psorakis et al. 2010) and summarised in the sections of this Chapter. The main convergence criterion is generalised from Tipping's binary Type-II ML method in (Tipping and Faul 2003) and together with a modified version that restricts early convergence are:

Criterion 1 $\log \alpha_m^* - \log \alpha_m^{\text{old}} \leq \text{threshold} \quad \forall \quad m \in \mathcal{M}$ and $\theta_i \leq 0 \quad \forall \quad i \in \mathcal{I}$

Criterion 2 $\log \alpha_m^* - \log \alpha_m^{\text{old}} \leq \text{threshold} \quad \forall \quad m \in \mathcal{M}$ and $\theta_i \leq 0 \quad \forall \quad i \in \mathcal{I}$ and minimum of N iterations.

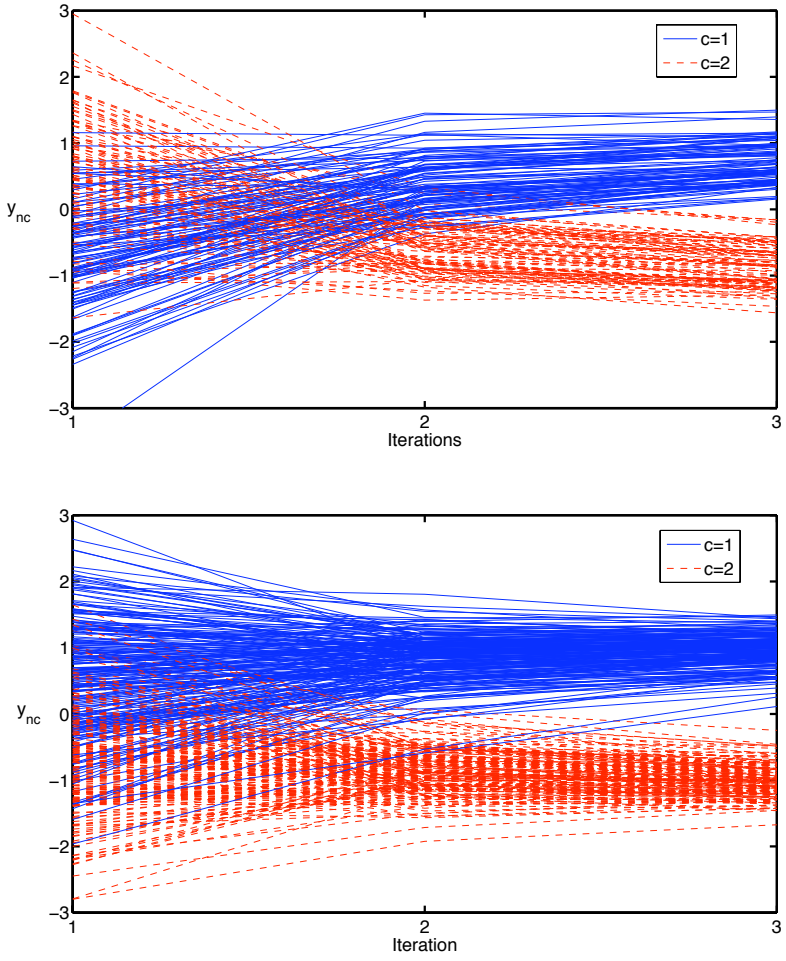

Figure 5.3: **Top**: Random Initialisation of $\mathbf{Y}$ and 50 cases that initialise contrary to the labels and probit link relation. **Bottom**: Aligned Initialisation of $\mathbf{Y}$ and 50 randomly selected cases (all follow the target labels from the start).

where the threshold is some small value, typically set to $1e^{-6}$. The convergence criterion assesses both the change on the scales of *active* samples and also the *contribution value* that *inactive* samples might still have. Intuitively, while the model proposes new samples, it can "see" how the inclusion of these individual samples changes the relevance of *all* inactive samples. Convergence occurs when all the inactive samples are judged irrelevant and the scales of the active samples remain unchanged to the threshold level accuracy.

5.4.5 mRVM$_2$

The second sparse model proposed follows directly from the MAP and EM algorithms considered so far with the only modification being a sparsity-inducing Gamma hyper-prior on the scales $\mathbf{A}$ and explicit pruning, while training, of insignificant samples. Note that the assumption for common scales across classes is not needed and $\mathbf{A} \in \mathbb{R}^{N \times C}$. Consider the properties of the *prior* Gamma distribution $\mathcal{G}_{a_{nc}}(\tau, \upsilon)$ with shape and rate parameter τ and υ respectively:

$$\text{Mean} = \frac{\tau}{\upsilon} \qquad \text{Variance} = \frac{\tau}{\upsilon^2} \tag{5.26}$$

By setting $\tau, \upsilon \to 0$ (for example $\tau = \upsilon = 1e^{-5}$) the prior on the scales becomes an un-informative (flat) distribution and encourages sparsity by allowing the data evidence to concentrate the posterior probability mass on very large scales (leading to sharply peaked regressor posteriors centred on zero) for "irrelevant" samples. This well-known prior setting follows the *automatic relevance determination* framework introduced by MacKay (2004).

The model now motivates a "top-down" procedure which starts with the full set of samples and results in a sparse solution through constant discarding of non-relevant samples via examination and thresholding of their associated scales and regressor posteriors. The training phase exhibits a speed-up during progression as the dominant computational complexity reduces from $\mathcal{O}(N^3)$ to $\mathcal{O}(M^3)$ due to the $N - M$ pruned out samples. A potential disadvantage of the algorithm, given in Algorithm 8, is that removed samples cannot be re-introduced into the model.

Algorithm 8 mRVM$_2$

1: Initialisation
2: Sample $\mathbf{Y} \in \mathbb{R}^{C \times N}$ to follow target $\mathbf{t}$.
3: **while** Iterations $<$ max & Convergence $>$ Threshold **do**
4: E-Step for α_{nc} still in the model: Eq. 5.9.
5: Prune $\mathbf{w}_i$, and $\mathbf{k}_i$ when $a_{ic} > 10^6 \; \forall \; c$
6: M-Step for $\hat{\mathbf{W}}_\circ$: Eq. 5.21
7: QP program for $\boldsymbol{\beta}$: Eq. 5.10 for reduced row rank $\hat{\mathbf{W}}_\circ, \mathbf{K}_\circ$.
8: E-Step for $\mathbf{Y}$: Equation 5.3 or 5.4 and 5.5 for reduced row rank $\hat{\mathbf{W}}_\circ, \mathbf{K}_\circ$
9: **end while**

It is worth mentioning that the model is theoretically more expressive than

mRVM_1 as there is an independent scale associated with every sample and every class. However, it is not a constructive algorithm and its application to large datasets would require splitting the training set into folds to select sets of *relevance vectors* within, that can be later combined for meta-learning an overall classifier.

5.4.6 Initialisation and Convergence Criteria for mRVM_2

Similar to the previous algorithms described in this Chapter, aligned initialisation of the auxiliary variables is beneficial although not necessary. The initialisation of the scales $\mathbf{A}$ for mRVM_2 should be performed irrespectively of the sparse prior placed upon them, by setting them to either uniformly sampled values on the unit interval or some small constant value. This is necessary for avoiding numerical problems when sampling from such a diffuse and improper prior.

Finally, convergence is monitored via the relative change of the auxiliary variables and regression coefficients every step, and by the progression of the logarithm of the joint likelihood. When convergence measures are below a threshold or a maximum number of iterations has been reached, the algorithm terminates. As it will be shown in the last sections of this chapter, convergence is typically reached within a few number of iterations for a variety of problems that are considered.

5.5 Preliminary Experimental Evaluation

In this section a preliminary evaluation of the proposed methods is performed on multinomial UCI (Newman et al. 1998) datasets. First, a comparison between the expectation-maximisation (EM) and the "full" maximum-a-posteriori (MAP) non-sparse solutions is offered that investigates their relative performance. Then, the two sparse models, mRVM_1 and mRVM_2, are compared against the same datasets in order to illustrate sparsity levels, classification accuracy and also any observed trade-off against the non-sparse estimators. Finally, further experimentation on large datasets is presented together with the accompanied algorithmic modifications necessary for scaling up.

5.5.1 Experimental Setup

Following the preliminary experimentation from the variational model in the previous Chapter, we employ the same UCI datasets and examine the performance of the algorithms via 10 times 10-fold cross-validation. The characteristics of the datasets are repeated in Table 5.1 to assist with interpretation of the results.

Dataset	N	C	D	**Kernel Function**
Balance	625	3	4	Polynomial (Order 2)
Crabs	200	4	5	Linear
Glass	214	6	9	Polynomial (Order 2)
Iris	150	3	4	Gaussian (scale $1/D$)
Soybean	48	4	35	Linear
Vehicle	846	4	18	Polynomial (Order 2)
Wine	178	3	13	Linear

Table 5.1: Multinomial UCI datasets. N, C, D are respectively the number of samples, classes and attributes in each dataset. The best-performing kernel function for each problem is reported.

Before reporting the experimental results, it is worth visualising the nature of the resulting *Relevance Vectors* (RVs). Employing the two dimensional Neal dataset (Neal 1996), as described in the previous chapters, we plot the typical resulting multiclass RVs from the sparse methods in Figure 5.4. Both methods retain *prototypical* samples that are representative of their class conditional distribution in contrast with other methods like support vector machines (Vapnik 1995) or informative vector machines (Lawrence et al. 2003) that retain samples, *support vectors* (SVs) or *informative vectors* (IVs) respectively, close to the decision boundary. This contrast stems from the difference in model fitting objectives. RVs are selected based on their contribution to the marginal (mRVM_1) or joint (mRVM_2) likelihood whereas SVs and IVs are pre-defined as boundary samples due to the geometric objective of maximising the margin or having lowest predictive likelihood respectively.

5.5.2 Non-sparse Comparison

First, the results from the EM algorithm and the MAP method are given in Table 5.2. As it can be seen the EM is clearly the best performing approach outperforming or matching the accuracy of the fully MAP method. The main

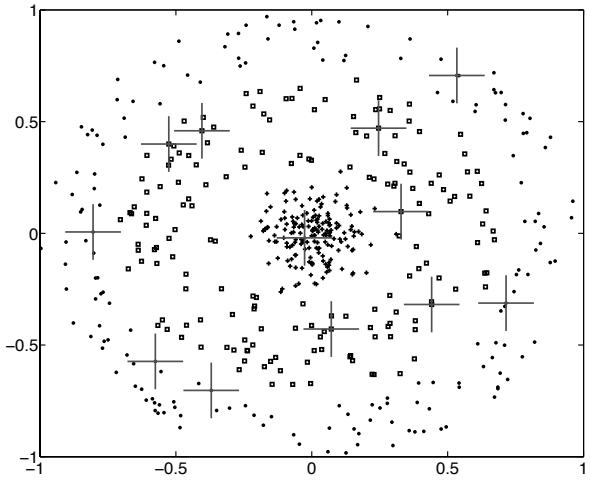

Figure 5.4: Typical Relevance vectors

reason for this observation is the less greedy nature of the EM scheme and the maximisation of the auxiliary variables under the truncated normal which exerts a less stable step than when taking the expected value.

	EM	**MAP**
Dataset	% Recognition Rate	% Recognition Rate
Balance	95.0 ± 3.6	$\mathbf{96.3 \pm 1.4}$
Crabs	$\mathbf{86.5 \pm 7.1}$	69.5 ± 7.6
Glass	$\mathbf{70.0 \pm 13.5}$	53.8 ± 9.3
Iris	$\mathbf{93.3 \pm 5.4}$	87.3 ± 9.1
Soybean	$\mathbf{97.5 \pm 7.9}$	87.5 ± 17.7
Vehicle	$\mathbf{75.9 \pm 5.4}$	68.5 ± 1.3
Wine	$\mathbf{95.9 \pm 3.9}$	92.9 ± 4.6

Table 5.2: 10 times 10-fold cross-validated recognition rates (mean±std) on standard UCI multinomial datasets with the EM scheme. Top performance from EM or MAP (not always statistically significant) in **bold**.

5.5.3 Sparse Comparison

For the case of the sparse mRVM methods, the 10 fold cross-validated recognition rates together with the resulting sparsity levels of the models are presented in Table 5.5.3. As it can be seen, mRVM_1 results in typically sparser solutions due to its constructive nature and the type-II fast maximum likelihood procedure.

The classification performance is not statistical different in most problems considered except the *Balance* and the *Vehicle* dataset where mRVM_1 outperforms the top-down approach of mRVM_2.

	mRVM$_1$		**mRVM$_2$**	
Dataset	% Recognition Rate	RVs used	% Recognition Rate	RVs used
Balance	**96.42 ±0.86**	8 ± 0	92.44 ± 1.54	13 ± 1
Crabs	85.2 ± 3.47	5 ± 0	**89.8 ± 2.42**	16 ± 1
Glass	65.9 ± 8.92	10 ± 2	67.57 ± 9.85	17 ± 2
Iris	93.73 ± 5.75	6 ± 1	**94.13 ± 2.31**	6 ± 0
Soybean	88 ± 17.58	7 ± 3	**97.75 ± 7.19**	6 ± 1
Vehicle	**77.52 ± 2.52**	10 ± 0	76.17 ± 1.2	27 ± 0
Wine	95.82 ± 0.98	3 ± 0	**95.94 ± 0.65**	5 ± 0

Table 5.3: 10 times 10-fold cross-validated recognition rates (mean±std) on standard UCI multinomial datasets with the mRVM schemes. Top performance (not always statistically significant) in **bold**.

5.5.4 Convergence, Sparsity and Predictive Power

In order to offer further insight to the nature and behaviour of the sparse algorithms we examine their characteristics during the training phase on the aforementioned UCI datasets. The full study and conclusions are described in (Psorakis et al. 2010). The experimental procedure adopted is to monitor four main properties while performing multiple 10 fold cross-validation and varying the training regime interval. These are:

- **Test Recognition Rate** - The (mean ± std.) percentage of correctly classified samples from the test set.
- **Predictive Likelihood** - The confidence of class membership predictions defined as $\sum_{n^*=1}^{N^*} \log(P_c)$ where P_c the probability of classifying test sample n^* to its correct class c^*.
- **Sparsity** - The number of Relevance Vectors in the model.
- **Model Fitting** - $\begin{cases} \textbf{mRVM}_1\text{: Marginal Likelihood progression.} \\ \textbf{mRVM}_2\text{: Joint Likelihood progression.} \end{cases}$

Furthermore, the previously proposed convergence criteria for both algorithms are recorded and their terminating point is displayed on each graph. For mRVM_1, Criterion 1 and Criterion 2 (Section 5.4.4) are denoted as "1" and "2" respectively, whereas for mRVM_2 "A" represents the point where the change in scales $\mathbf{A}$ is insignificant (Section 5.4.6) and "N" represents the point where a maximum number of iterations, equal to the training size, has been reached.

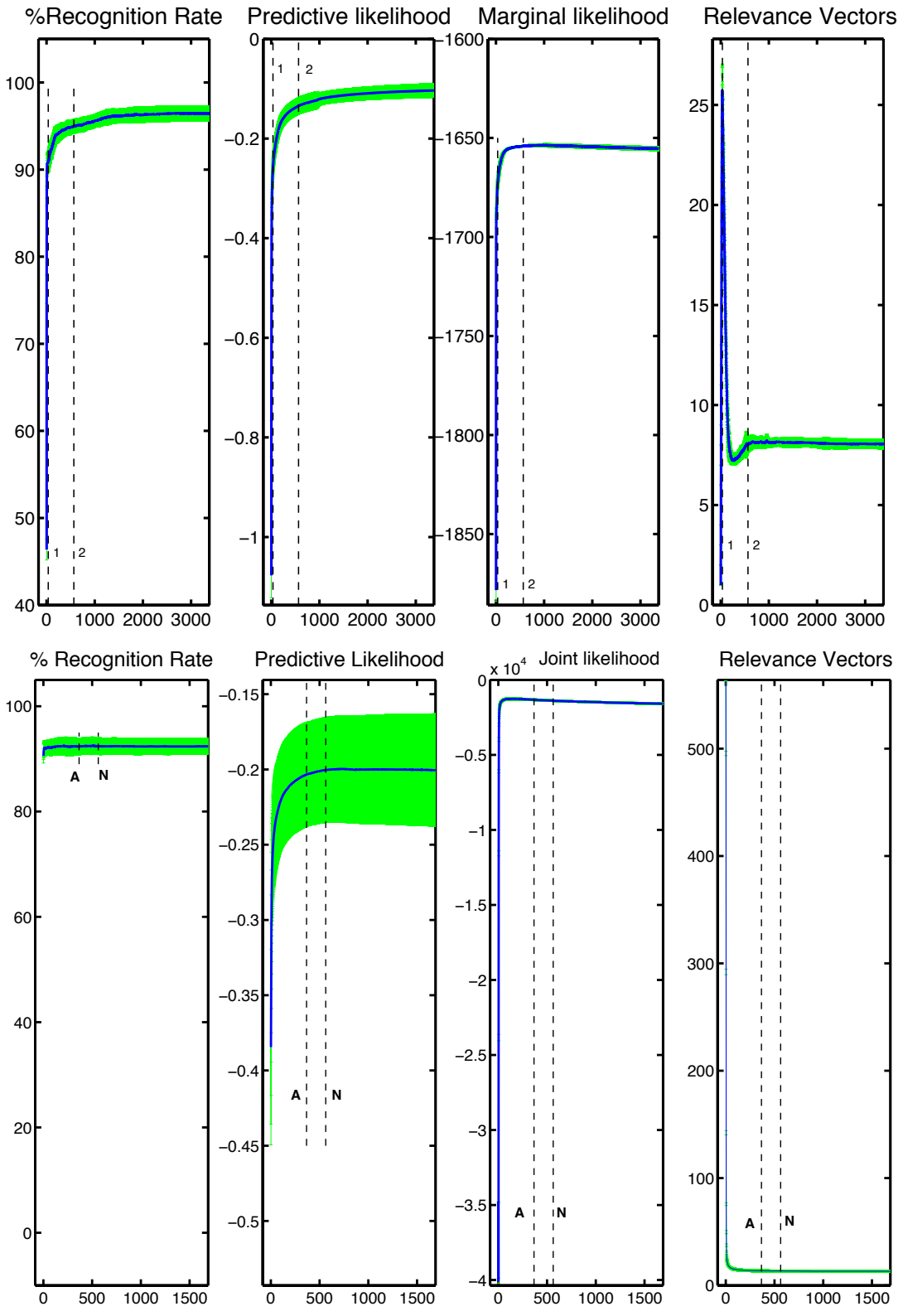

Figure 5.5: Balance dataset. Top: mRVM_1 Bottom: mRVM_2

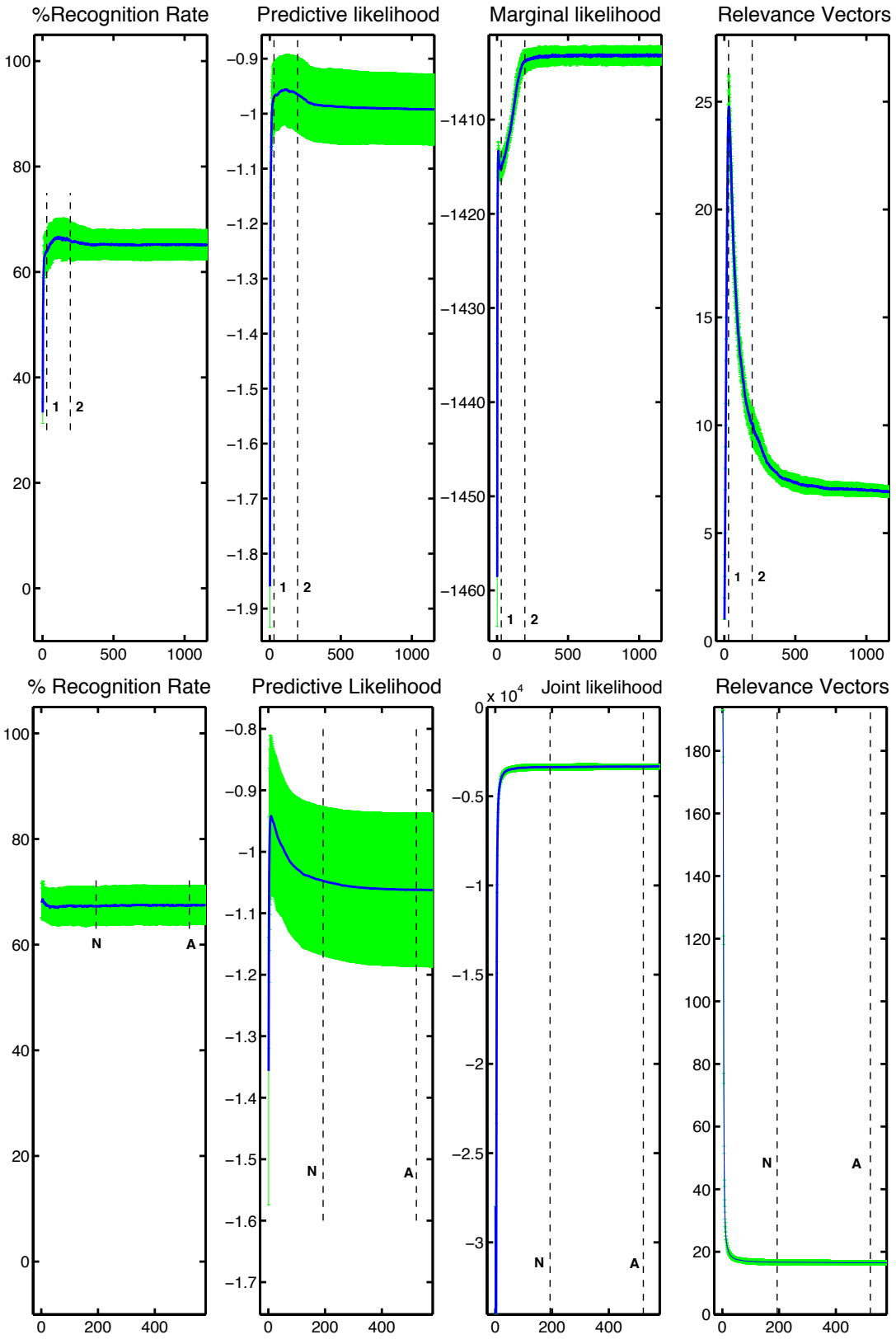

Figure 5.6: Glass dataset. Top: mRVM_1 Bottom: mRVM_2

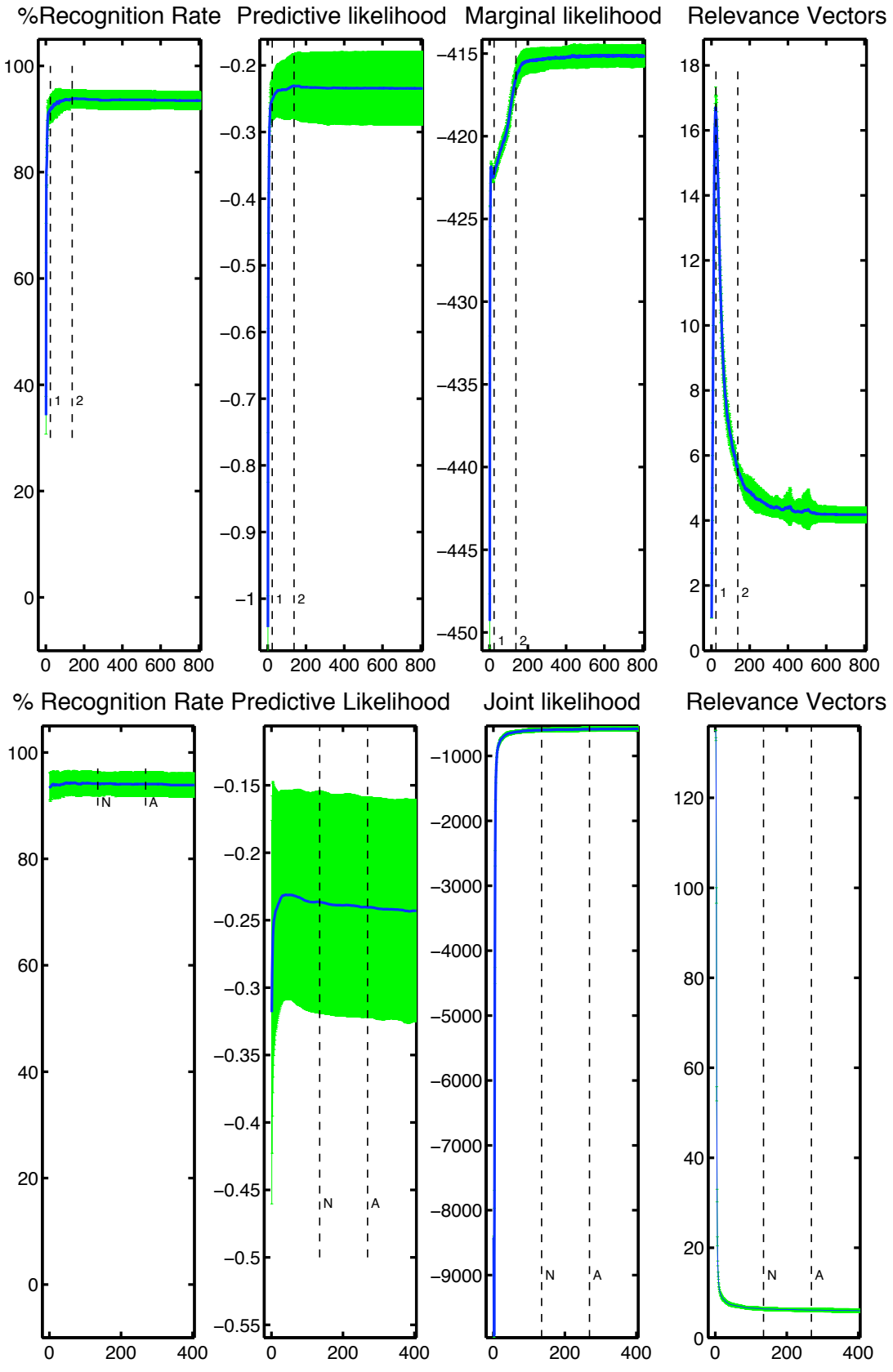

Figure 5.7: Iris dataset. Top: mRVM_1 Bottom: mRVM_2

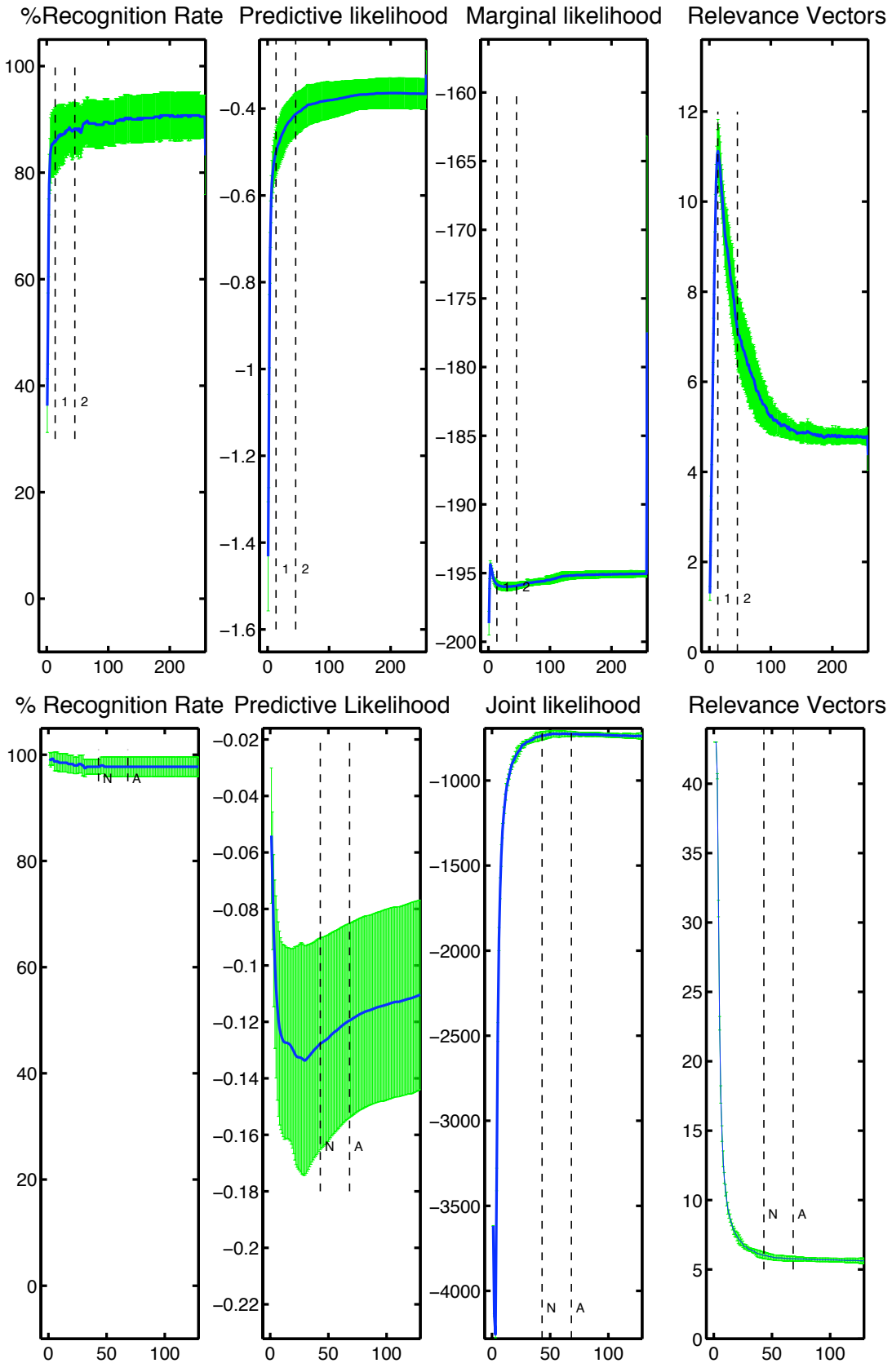

Figure 5.8: Soybean dataset. Top: mRVM_1 Bottom: mRVM_2

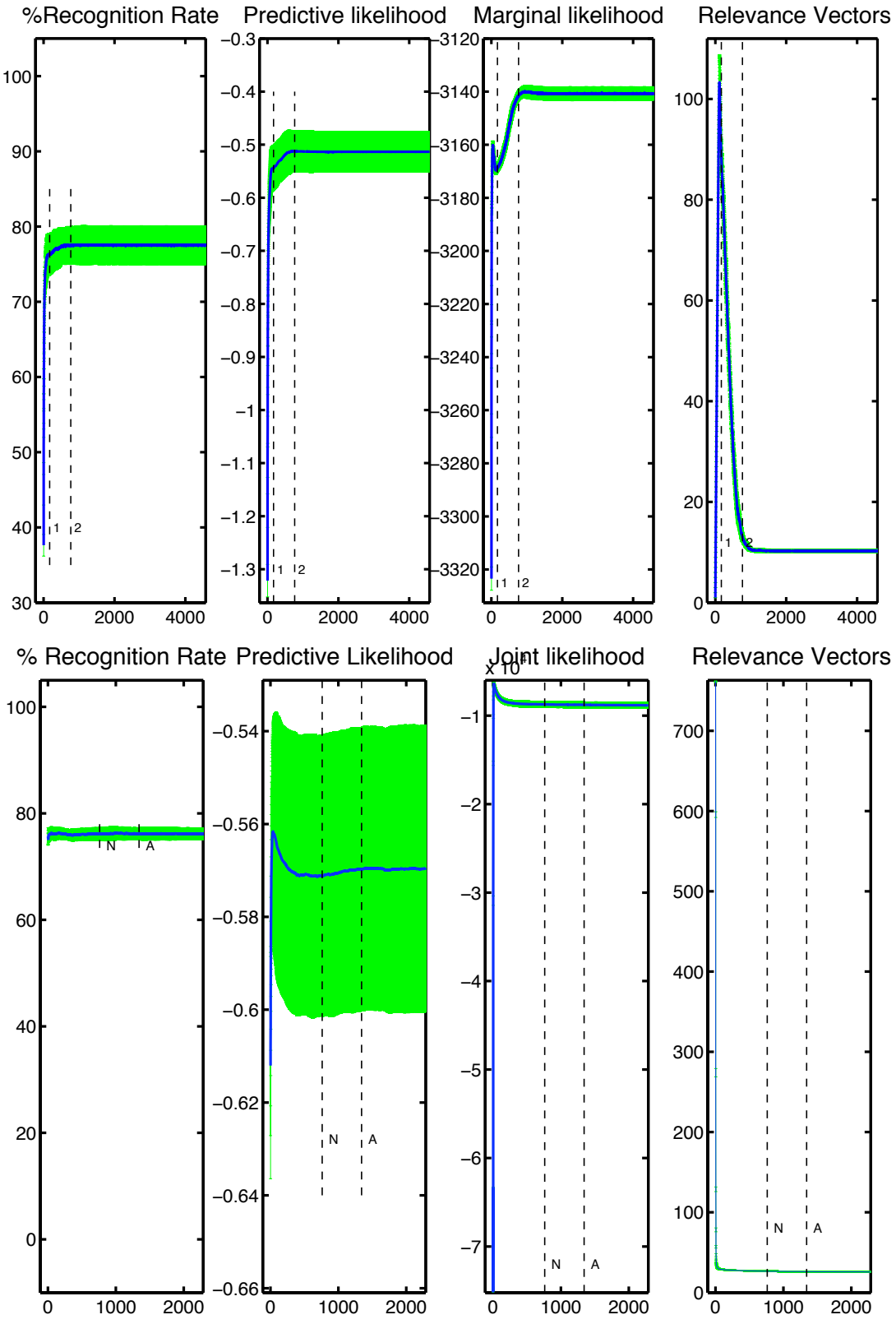

Figure 5.9: Vehicle dataset. Top: mRVM_1 Bottom: mRVM_2

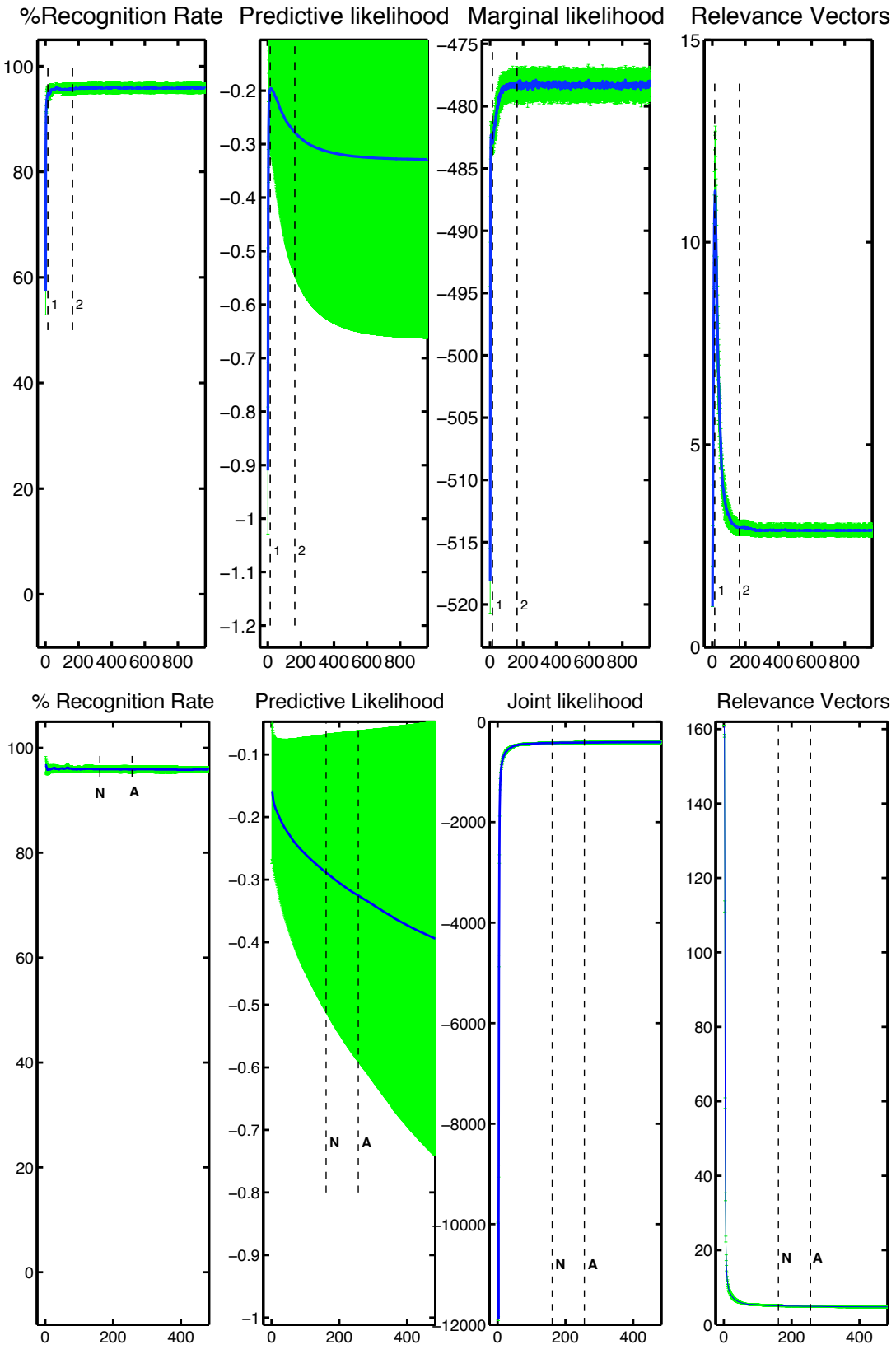

Figure 5.10: Wine dataset. Top: mRVM_1 Bottom: mRVM_2

Both methods produce very sparse solutions while retaining competitive performances with respect to both the variational approximation and other standard classification algorithms (Manocha and Girolami 2007). In some cases, such as for *Glass* and *Iris*, the number of relevance vectors approaches the number of classes while still retaining good levels or recognition rate and confidence in predictions. This problem-dependent characteristic implies that such datasets do not require overly complex decision boundaries and in fact reducing the bias and complexity via sparsity (predictions are based only on few RVs) can result in better generalisation to unseen novel samples. This is further supported by the progression of the model fitting measures (marginal and joint likelihood) in these datasets, that typically increase as the sparsity levels increase (less RVs) indicating that the models actually describe better the data when less samples are used.

mRVM_1 typically starts by adding the most informative samples from the inactive set and then starts pruning out the ones from the active set that are no longer informative enough (due to the inclusion of new samples). This behaviour, common across datasets, results in some interesting model fitting progressions especially captured on the *Vehicle* dataset. In that case, during the initial "build up" phase where informative samples are added the marginal likelihood is increased. Then, as the algorithm starts pruning out samples from the active set that are no longer judged relevant, the marginal likelihood is briefly reduced and then increases to the maximum level which corresponds to the end of the "prune out" period and stabilises.

mRVM_2 starts with the full model and very quickly (typically within 10 iterations) prunes out the majority of the training samples. The retained relevance vectors from each dataset considered indicate that this method results in *less* sparse solutions than the mRVM_1. However that sparsity level is still very high when considering competing methods such as SVMs and IVMs that are less efficient in multiclass problems due to their binary nature (which requires a set of vectors retained from each binary sub-problem considered).

The progression of the predictive likelihood appears to be dataset dependent but in most cases it reaches a maximum and either stabilises or declines. This indicates that the methods may become over-confident in the predictions as the model over-trains and possibly over-fits the data. The exception is the *Soybean* problem which however is the smallest dataset in this collection and the very

small number of training and testing points alters the expected decline of the predictive likelihood.

5.6 Discussion

In this final methodological Chapter, novel algorithms for sparse probabilistic MKL that are based on point-estimators were presented. Starting from a general expectation-maximisation scheme and a maximum-a-posteriori estimator the approach led to a generalisation of the well-known Relevance Vector Machine to the multiclass multi-kernel setting with the multinomial probit likelihood. Two different sparsity-inducing models were derived, the first (mRVM_1) being a constructive type that starts with an empty model and informatively selecting or deleting samples, and the second (mRVM_2) a top-down approach that starts with the full set of samples and prunes out the "irrelevant" ones.

The computational complexity gains from the sparse methods are significant. In the case of mRVM_1 the dominant order has been reduced to $\mathcal{O}(PM^3)$ where P the number of iterations, $P \ll N$ under the first convergence criteria, and the memory requirements reduced to $\mathcal{O}(SNM + (N + M)C)$ where S, C, N, M are the number of sources, classes, total samples and retained samples (relevance vectors) respectively. Finally, similar computational complexity reductions are achieved for mRVM_2 with the only difference being that initially the model is still governed by an N^3 order as it starts with the full training set. As we have seen though, within few iterations the number of utilised samples falls dramatically and converges to the final number of relevance vectors.

An unresolved aspect of the proposed models in this Chapter is the lack of appropriate inference procedures for the kernel parameters in the case of parameterised kernel functions such as the Gaussian. Cross-validation is the obvious but inefficient approach to this problem which however is adopted by other non-probabilistic popular models such as SVMs. An alternative but still inefficient approach is gradient-based methods as proposed in the original RVM (Tipping 2001). The potential benefit of inferring the Gaussian kernel parameters, besides a better smoothing of the composite feature space, is the identification of informative features through Automatic Relevance Determination (MacKay 2004). This will open the way for *joint* feature and sample selection in mRVMs.

Further experimentation on the proposed probabilistic multiple kernel learning setting through the motivating application and large scale datasets is reported in the following two Chapters that summarise the main experimental results of this thesis.

Chapter 6

Automatic Currency Validation

In this Chapter, experimental results on the motivating application for this thesis are presented together with the literature review and previous work on the area. Some of the experimental findings from the covariate ranking approaches, the pMKL methodology and the associated feature extraction procedures related to sensitive banknote information will remain unpublished for confidentiality and the self-service application of the methodology protected by US patent law (He et al. 2009). This Chapter summarises the main technical reports (Damoulas 2006, Damoulas 2008b, Damoulas 2008a, Damoulas 2009) submitted to NCR Labs as part of the project deliverables during the period of this thesis.

6.1 Motivation

Since the introduction of Automated Teller Machines (ATM) in the '60s the everyday transactions for customers of banks and financial institutions have become conveniently fast and efficient. The world's first cash machine was introduced in the UK in 1967 and current statistics[1,2] indicate that more than $58,000$ ATMs are employed in the UK alone with the corresponding figure for the US market rising above $370,000$. Taking into account that[3]:

1. Nowadays deposits account for $40-60\%$ of all branch transactions.

2. Teller costs are 5 times higher than automatic deposition through ATMs.

[1] http://www.apacs.org.uk/resources_publications/cash_machine_facts_and_figures.html
[2] http://www.atmwarehouse.com/ATMstatistics.htm
[3] Source: Dr. Chao He, NCR Labs

3. Current automated deposition methods save $30 - 75\%$ of the cost.

it is straightforward to see the need for ATM automatic depositions.

As with most new technological advancements, new needs were also created in conjunction with the development of the ATMs. One of the by-products of the ATM's broad use and development is the introduction of security risks associated with such transactions. In particular, certain types of fraud such as fake identification, counterfeit currency and even money laundering that in the past were dealt with by human personnel are now faced by the ATMs.

In parallel, the further "development" of the quality of counterfeit notes to the point of partially reproducing existing security features led to the re-introduction of the human factor in the process of currency transaction. Money deposited in ATMs have to be hand-checked by experienced personnel and the corresponding transactions have to be recorded for security reasons. That problem alone, disregarding other security issues of ATMs such as fraud by fake credit cards or theft of personal costumer information, is very costly in human processing time and eventually leads to a potential profit loss.

Furthermore, the very nature of the problem created a clear separation, on the basis of safety risks, between the two types of transactions: depositing and withdrawing money. Nowadays the main service of ATMs is to allow costumers to withdraw cash and not to deposit, partially due to the requirement for trustworthy and efficient Automatic Currency Validation (ACV) systems.

This requirement is the general motivation and application scenario for the research undertaken in this thesis. Important aspects of the ACV problem are the lack of availability and scarcity of counterfeit notes, the plurality of currency types, denominations and ATM input-orientations, the plurality of sensory measurements, and finally the natural fatigue and ageing process that used notes have undergone.

This thesis is motivated by, and attempts to address, the specific ACV problem of integrating ATM sensory inputs towards an overall, possibly multinomial, classification decision.

The Bayesian paradigm has been adopted to tackle this scenario due to the clear benefits of probabilistic reasoning that include: 1) formal inclusion of prior beliefs such as counterfeit scarcity or abundance, ATM location, currency-specific security characteristics, 2) probabilistic classification decisions amenable to further post-processing, risk assessment and decision making, 3) inference of

feature discrimination strength (covariate ranking), sensor discrimination ability (kernel combination parameters from pMKL) and sample descriptive power (relevance vectors from mRVMs).

In the following section I review the available past work on ACV before describing the specific problem of integrating a plurality of ATM sensory measurements for an overall decision.

6.2 ACV Literature Review

In this section we present an overview of published work related to the ACV problem under consideration. The commercial and security sensitive nature of the application scenario limits the available published work and hence this literature review is not necessarily representative of the actual progress in the area. Furthermore, the scarcity of counterfeit notes together with certain legal issues (possession and creation of counterfeits) and the expense of an ATM machine, restricts the availability of data. In that respect, past work on the general problem of the recognition and classification of currency notes is relatively limited.

It is worth noting that as the majority of the ATM detection signals are in fact images of the currency notes, part of the image processing literature is very relevant. However we will not review that literature here as we attack the problem from a statistical machine learning perspective that can in fact incorporate various feature extraction and construction methods (such as wavelets, edge detection features, splines) via the kernel trick embedding on each channel and later pMKL integration. Furthermore, the computational restrictions do not allow for expensive feature extraction approaches as it will be further discussed in a later section where the adopted feature extraction techniques are presented.

6.2.1 Recognition and Verification of Currency

The majority of the past work on ACV (Frosini et al. 1996, Ahmadi et al. 2003c, Omatu et al. 2001) has the goal of correctly recognising and classifying genuine used and new notes into classes according to their denomination value (i.e. £5 or £10) and a *single* source of information. Although directly relevant, this problem is less difficult than the general and realistic case of dealing with counterfeit notes at the same time and multiple detection signals. One could argue that a

counterfeit note would be rejected by such algorithms as not belonging to any class by producing a low probability of either class membership but this becomes less realistic when considering the quality of some counterfeits and the crudeness of the adopted feature extraction methods.

Artificial Neural Network (ANN) methods

The work reported by Glory Co., Ltd.[4] jointly with the Universities of Tokushima, Osaka and Okayama (Ahmadi et al. 2003c, Ahmadi et al. 2003a, Ahmadi et al. 2003b, Ahmadi, Omatu, Kosaka and Fujinaka 2004, Ahmadi, Omatu, Fujinaka and Kosaka 2004, Ahmadi, Omatu and Kosaka 2004, Kosaka and Omatu 1999, Kosaka et al. 1999, Kosaka and Omatu 2000b, Kosaka and Omatu 2000a, Omatu et al. 2001, Kosaka et al. 2001) presents results of different variants of a recognition system which employs a *Learning Vector Quantization* (LVQ) (Kohonen 1990, Hastie et al. 2001) classifier at its core. The LVQ method is a supervised learning classification method that was developed in the 90's based on a type of Kohonen network (Vector Quantization) and it is very similar to *k-means* clustering (the difference being that LVQ uses all the classes to decide on the positioning of the prototypes) and to the nearest-neighbour rule (both use the Euclidean metric). The main drawback of the LVQ methods is that they are defined by algorithms rather than optimisation of some fixed criteria and hence it is difficult to understand their properties (Hastie et al. 2001).

The developed algorithm receives as inputs part of the note image which in most cases is further compressed through a Principal Component Analysis (PCA) linear reduction method (Hastie et al. 2001, Bishop 1996, Bishop 2006). The pre-processing stage also includes operations such as shifting and rotation of the original note image in order to account for the variability of insertion to the ATM. Furthermore, the classifier takes into account different orientations of the note (e.g. upside-down) using sub-class allocation. For example, the £5 class includes 4 sub-classes for all the possible orientations of a £5 note. Finally, in certain cases the Self-Organising Map (SOM) clustering algorithm developed by Kohonen (Kohonen 1990) is employed to partition the input space and then apply PCA to each region instead of the whole note.

A further extension to their work is the development of axis-symmetrical

[4]Money handling company - Founded in 1918 in Japan and established in 1982 in the USA market as Glory Inc. http://www.glory-jpn.com & http://www.gloryusainc.com

"masks" that are used to extract features from the currency notes and are invariant to inversion and rotation, (Takeda et al. 1993, Takeda and Nishikage 2000, Takeda and Omatu 1995a, Takeda and Omatu 1995b, Takeda et al. 1994, Takeda et al. 1998, Tanaka et al. 1998). The idea is based on work by Widrow et al. (1988) in which *slab* values, i.e. values from a planar network configuration that uses a majority rule, are used for a network with translational and rotational invariance. The location of these masks are in most cases optimised by a Genetic Algorithm (GA) (Mitchell 1998) and after their application on the notes the result is passed as an input to an ANN for final classification.

In general, the work of the Glory group concentrates on specific methodologies for classification, employing ANNs and Kohonen networks for classification together with SOM clustering techniques. Their results report rough recognition rates and classification accuracies (training in 2/3 of the data and testing on the remaining 1/3) without reporting cross-validated errors and error variances. Furthermore, the reported recognition rates (of 100% in most cases) are for genuine notes only and imply that the system is just able to distinguish between two different currency types. The approach adopted offers little if any insight to the nature and characteristics of the data and the methods used are now mostly outdated by theoretically principled machine learning techniques firmly embedded within statistical methodologies.

A different ANN approach to the problem comes from (Frosini et al. 1996). This work reports recognition and verification techniques used in a banknote acceptor (BANK) that was implemented for accepting paper currency of different countries. Their approach offers a three-way verification mechanism:

- *Dimension Verification* - The first operation performed when the note is introduced in the machine, simply a measure of the note's dimension. Especially useful when a different dimension corresponds to a different value for the note in which case it is easier to perform the classification.

- *Verification Threshold* - Employing ANNs (feed-forward networks or Multilayer Perceptrons (MLPs)) for the classification of the notes based on a thresholding criterion. This is susceptible to misclassification errors related with the creation of open separation surfaces by MLPs, (Gori and Scarselli 1998).

- *Autoassociator-based Verification* - An ANN which is trained to reproduce

the inputs as outputs. This is used to overcome the aforementioned MLP problems of creating open separation surfaces. It offers a better description of the target class through the closed separation surfaces and can make use of negative examples to restrict the class description area.

The dataset employed in this work comprised 600 genuine notes and 200 photocopies (50 colour and 150 black and white) and was split to two sets, a training set with 300 genuine and 60 photocopies and the rest as testing set. The best result on the test set reported is a 0% test error rate (False Positives) with a corresponding 5.6% test rejection rate (False Negatives). Again the error reported is a rough test error and not a cross-validation one (or a bootstrap test error value (Efron and Tibshirani 1993)), failing to give an idea of the error variance and to decouple the results from the specific test set used. Most importantly, the experimental procedure of using photocopied notes as counterfeits for both training and testing does not simulate the quality of counterfeit notes and hence is over-optimistic, especially if it is to be employed against real state-of-the-art counterfeits.

Finally, an inherent problem with most of the reviewed work in this section is that the input data used is often limited or of poor quality. That is due to the type of sensory information which in (Frosini et al. 1996) is just two signals from sensors covering two parallel strips of the note and in the other cases such as (Kosaka and Omatu 1999, Kosaka et al. 1999, Kosaka and Omatu 2000b, Kosaka and Omatu 2000a, Omatu et al. 2001) is the original pixels of an image of the note (or part of it), further compressed. In certain cases it is almost certain that the majority of real life counterfeit notes would have no problem in getting misclassified as genuine and hence surpassing the obstacle of the proposed classifiers.

Statistical methods for classification of currency

A different approach to the ACV problem is offered through the perspective of *novelty detection* (Markou and Singh 2003, Filippone and Sanguinetti 2009) and *classifier combination* in the work of He et al. (2004). The main intuition behind the work is that "*It is difficult in the counterfeiting process to provide a uniform quality of imitation across the whole note and certain regions of the note may be more difficult than others to copy successfully*". Through that perspective, the

note is being segmented into sub-regions were individual classifiers are trained and their decision is combined into an overall classification. The optimised segmentation of the note and the combination of classifiers is achieved through a genetic algorithm (GA).

The one-class classifiers are built based on statistical hypothesis testing that maximise the log-likelihood ratio of the *null hypothesis* (i.e. that the data under consideration is drawn from the target genuine class) over the alternate hypothesis. The D^2 test (Hastie et al. 2001) is used when assuming a multivariate Gaussian distribution for the target class, and a mixture of Gaussians with bootstrap sampling (Efron and Tibshirani 1993) when assuming a non-Gaussian distribution for the target class. The combination of classifiers in a principled manner (Tax and Duin 2001) is achieved in this work by making use of a product combination decision rule which corresponds to unanimous voting (a note is classified as genuine only if all the region-specific classifiers agree). That scheme was selected on the basis of the nature of the problem which demands a low False Negative rate (less counterfeits being classified as genuine notes).

The work in (He et al. 2004) underlines some important concepts and characteristics of the problem in hand. First of all, the notion that local information from certain sub-regions of the note might hold specific importance and weight for the classification task and secondly that a combination of decision-makers may enhance the performance of the overall classification. In contrast with (Frosini et al. 1996) where the use of autoassociator networks was deemed necessary for the system to create closed separation surfaces, in (He et al. 2004) both methods reported provide such a closed surface either in the form of a multivariate Gaussian or a mixture of Gaussians.

The shortcomings of this work are mainly the restrictions imposed by the use of a rectangular grid to separate the regions of the note and the use of a GA procedure to optimise the combination of classifiers. The rectangular grid places an unjustified assumption of *a priori* separation of the note that is not based on regional information or a pixel grey-scale value correlation between the members of each area. Furthermore, the choice of a GA to optimise the combination of the classifiers is not a very efficient approach to the problem since it is a stochastic optimisation method employing local random search based on a predefined fitness function without making explicit use of problem-dependent characteristics.

Fatigue classification

Another aspect of the automatic currency classification problem that has been partially addressed by the Glory group is the ageing of the bank notes. The classification of the fatigue level of a note into one of certain number of categories based on acoustic signals derived while the notes were passing through a part of the ATM, has been studied in (Teranishi et al. 1999, Teranishi et al. 2000, Teranishi et al. 2002, Teranishi et al. 2005). The tensional acoustic signal (when the note is stretched) or the frictional one (the acoustic signal produced due to friction) has been used as the input basis for classification of the note's fatigue level.

The same type of classifier as in the case of value recognition was used (or a slight variant of it), namely LVQ (Kohonen 1990, Hastie et al. 2001), and the feature space was created considering different manipulations of these signals:

- Fourier Transformations of the acoustic signals, (Teranishi et al. 2005). The signal is divided into frames and the spectral components are calculated from a Fourier transformation and then used as the features.

- Cepstrum analysis and use of the cepstrum coefficients of the acoustic signal, (Teranishi et al. 1999). The signal is again divided into frames and for each frame the cepstrum coefficients, see (Bogert et al. 1963), are calculated and used as the features.

- Acoustic Energy Patterns of the signal, (Teranishi et al. 2000). The signal is divided into frames and the energy (in relation to the square of amplitude) of the signal is used as the features.

- Acoustic Wavelet Components, (Teranishi et al. 2002). A wavelet transformation (Graps 1995) is applied and the wavelet power pattern is used as the features.

The results reported seem to be in Teranishi et al. (2002) a 10-fold cross validated test error but without reporting the variance of the error, and in the rest of the cases a rough test error percentage. Therefore, it is not clear what is the real performance estimate of the methods and the conclusions that can be drawn. It appears that the wavelet transformation leads to better classification

rates than using the energy of the signal, and that using the tensional acoustic signal instead of the frictional one can lead to a lower misclassification rate.

The use of acoustic signals for the determination of the fatigue level is an interesting idea as it could also provide in the future an alternative source of information for the proposed pMKL methodology and it could also be used, in a similar manner to the reviewed work, to determine the fatigue level of the note. The expectation is that such an information source would particularly assist, and would be weighted accordingly from the pMKL methods, in the separation between new and circulated genuine notes.

6.3 ACV with Multiple Sources of Information

In this thesis we focus on the specific ACV problem of integrating the available sensory information towards an overall classification decision. A typical ATM is equipped with various sensors that collect information of the deposited currency notes during the transaction as depicted in Figure 6.1. Such sensors can be light based, in which case light emitted in different frequencies hits the note and it either reflects back or transmits through to the sensor, or non-light based such as sensors measuring physical characteristics of the note. The first produce an Image-like signal that retains specific characteristics and features of the original note, the latter are in the form of spatio-temporal signals or discrete measurements. The sensory information collected and employed in this thesis is summarised in Table 6.1.

ACV Sensory Information	
Sensor (Generic Abbreviation)	**Dimensions**
FS_1 (Image-like)	23760
FS_2 (Image-like)	23760
FS_3 (Image-like)	23760
FS_4 (Non-Image)	240
FS_5 (Non-Image)	1920
FS_6 (Non-Image)	720

Table 6.1: Characteristics of the available ACV sensory information. Further details regarding sensor measurements are confidential to NCR Labs.

Figure 6.1 gives a schematic description of the problem and the level on which the proposed pMKL methodology integrates the available information

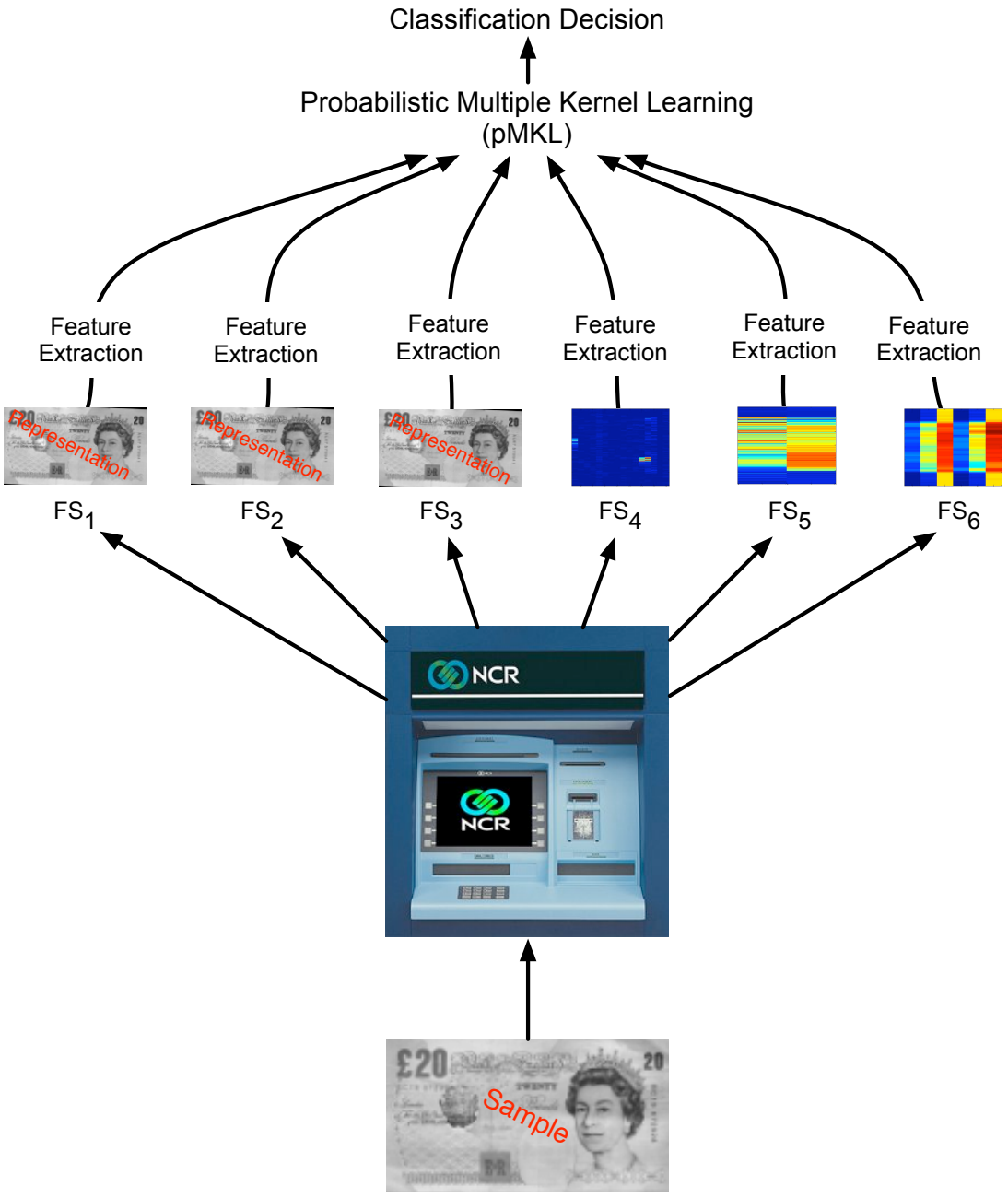

Figure 6.1: Multiple Sources of Automated Currency Validation. The deposited currency note produces crude sensory information from which features are extracted and later combined via the proposed pMKL methodology towards a final classification decision. Images not necessarily representative of sensor measurements.

sources towards an overall classification decision. Before presenting the experimental results we briefly consider the feature extraction methods considered for each signal and comment on the computational restrictions that guide the final choices.

6.4 Feature Extraction

The feature extraction procedures are briefly described in this section without an in depth investigation due to security and commercial sensitivity issues. For an in depth investigation the reader is directed to the confidential reports (Damoulas 2006, Damoulas 2008a). As it was shown in Table 6.1, the crude signals are high-dimensional and prohibitive for an online classification method in an as-is basis. Furthermore, the proposed feature extraction methods are able to capture the structure of the data and provide very discriminative feature spaces that, as it will be shown, produce state of the art ACV results across currencies and denominations.

6.4.1 Image Channels

The "Image" channels are the standard type ATM signals used for ACV and they resemble actual images of the note under different light conditions. Previous work (He and Ross 2006) on constructing discriminative features from these high dimensional sources offers an efficient solution to the feature extraction problem which has been further improved within this thesis. The approach consists of grouping characteristics within each Image channel in order to create a *segmentation template or "mask"* which can later be used to extract the corresponding features from an incoming note. Such masks can be seen in Figure 6.2 for specific Image channels.

The proposed feature extraction improvement is to simultaneously group characteristics from all the Image channels hence exploiting structure available from every Image source and result into one overall "mask" similar to the ones in Figure 6.2. This further reduces the memory requirements of storing these templates while retaining (and even improving) classification levels (Damoulas 2006).

Figure 6.2: Typical extraction masks for some Image channels.

6.4.2 Non-Image Channels

In the case of the Non-Image channels, alternative feature extraction approaches were examined despite the observation that a similar procedure of constructing "masks" offers the most discriminative feature space (Damoulas 2008a). The alternative approaches that were examined and subsequently adopted were chosen on the basis of very low dimensionality in order to satisfy computational and memory restrictions. The majority of the adopted feature extraction methods for these channels are histogram statistics within specific sub-areas of the channels (Damoulas 2008a). Given the information integration offered by the pMKL methodology, less than optimal feature extractions on the Non-Image channels are satisfactory in achieving state of the art results that are presented in the following sections.

6.5 Covariate Ranking

Before presenting the pMKL results on integrating the various Image and Non-Image channels for the ACV problem, we consider standard Bayesian Generalised Linear Models (GLMs) (Denison et al. 2002) applied to a concatenation of features extracted from the Image channels with a second order polynomial expansion. As the aforementioned segmentation procedure extracts clusters within these sources, the goal is to infer the significance or these covariates and their discriminative power. This first experimental study, summarised in (Damoulas 2006), offered to NCR Labs an insight into the information content of specific sub-regions of the notes for various currencies and denominations. In order to reconfirm the results and assess different inference schemes that were introduced in Chapter 2, the GLM models in Table 6.2 were considered.

GLM Model	Inference scheme(s)
Binary Logistic regression	Metropolis MCMC & Laplace Approximation
Binary Probit regression	Gibbs MCMC & Laplace Approximation
Multinomial Logistic regression	Metropolis MCMC & Laplace Approximation
Multinomial Probit regression	Gibbs MCMC

Table 6.2: Generalised Linear Models employed for Covariate Ranking.

The experimentation was performed on an NCR dataset of $\sim$ 700 English £20 notes with front face orientation performing 10 fold cross-validation and the binary classifiers were employed to classify genuine versus counterfeit notes while the multinomial classifiers addressed the three-way distinction between genuine new, genuine used and counterfeit notes. In the following subsections we summarise the main findings, see (Damoulas 2006) for full details, and concentrate on inference of informative covariates in a statistical significance sense.

6.5.1 Binary Classification

Treating the problem as a binary classification allows one to examine discriminative covariates between the general category of genuine notes and counterfeits. In Figure 6.3, a typical Markov chain for binary logistic regression is plotted. As it can be seen a specific regressor deviates significantly from the zero-mean prior hence indicating evidence of significance for the corresponding weighted covariate. The specific identified segmentation region of the note is consistently ranked across models as informative and provided insight to NCR regarding note design, security features and the difference between human and machine discrimination.

Examining further some of the resulting posteriors in Figure 6.4, we can see the difference between regressor posteriors of discriminative and non-discriminative covariates.

In order to assess statistical significance we employ the standard Z-scores (Denison et al. 2002) which is simply the ratio of the posterior mean over its standard deviation and hence penalises "vague" posteriors that might appear significant but have a small scale. The typical Z-scores for the binary case can be seen in Figure 6.5 where again the same covariate achieves a score well above two which indicates significant discriminatory strength.

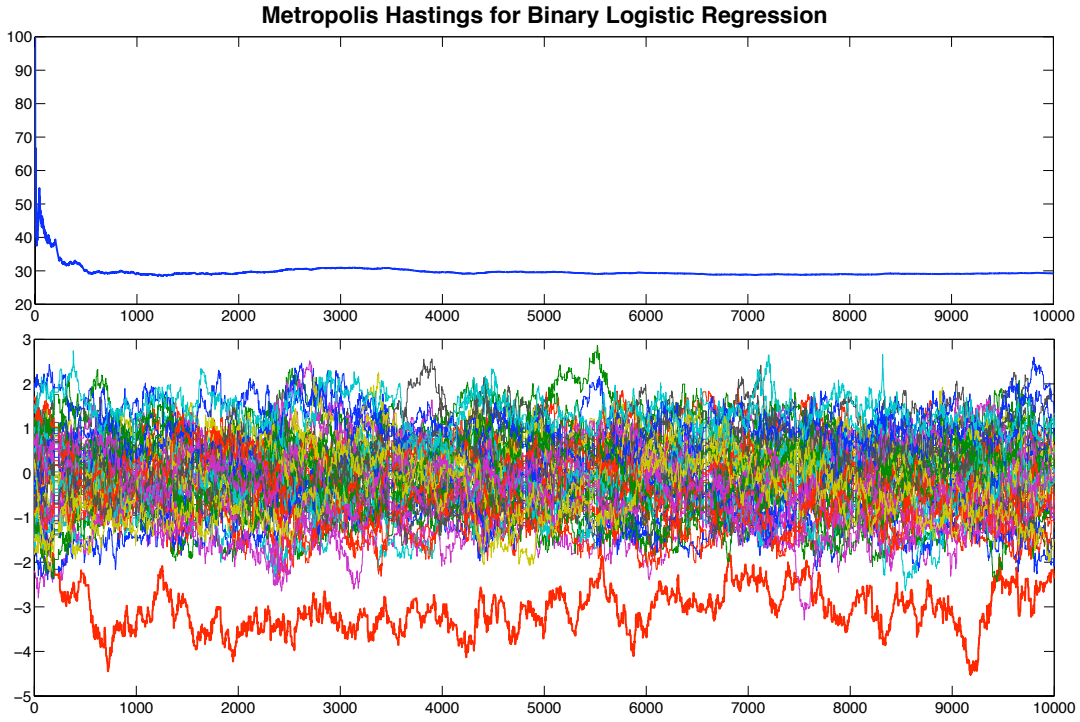

Figure 6.3: Typical Markov chain from the GLMs. Top: Acceptance ratio tuned to 30%. Bottom: Samples from the regression coefficients posterior distribution.

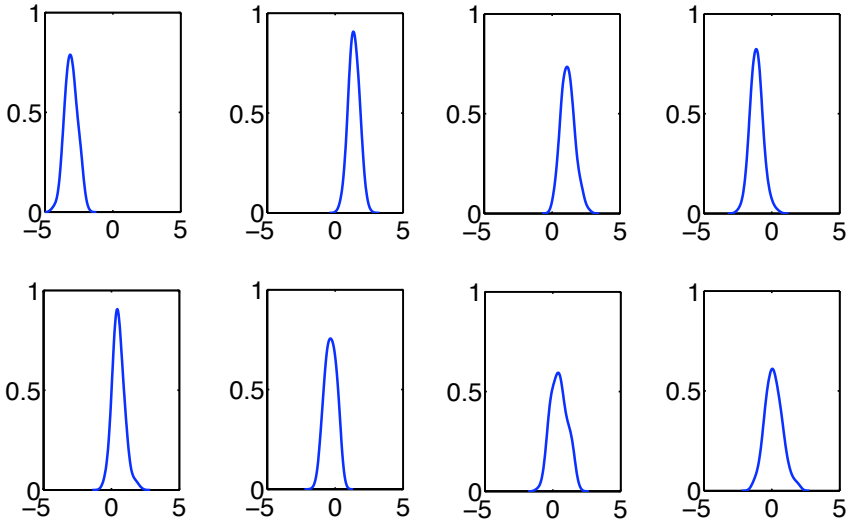

Figure 6.4: Some posterior distributions (smoothened via a Parzen window type filter) from the Markov chain. Top row: Posteriors significantly deviating from the zero-mean prior. Bottom: Posteriors not deviating from the zero-mean prior.

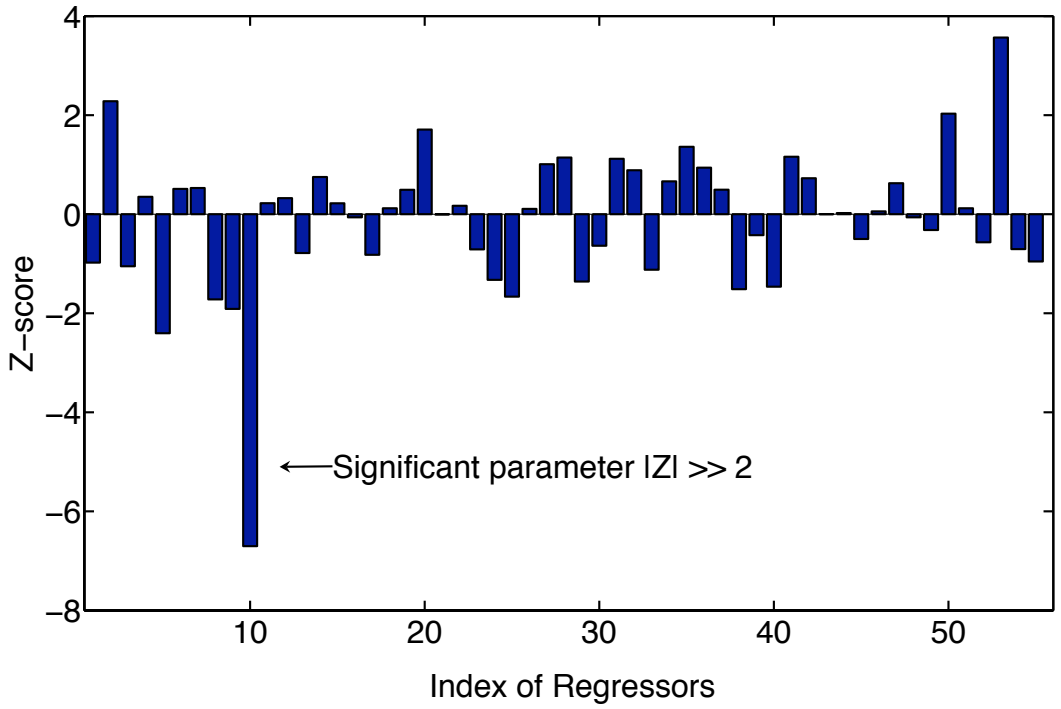

Figure 6.5: Typical Z-scores for the binary classification between genuine and counterfeit notes.

Typical classification error progressions can be seen in Figure 6.6 and Figure 6.7 for different segmentation levels. As it can be seen, the Image channels alone are able to produce an error percentage well below the 1% level for this specific currency and orientation (this is not the case across currencies as Non-Image information has been found to outperform these channels in certain cases). As the segmentation level increases the performance of the classifier improves and it is apparent that even simple GLM models operating on the extracted features (the cross and square terms of the polynomial expansion are not judged significant) are able to produce satisfactory recognition rates for this currency and orientation.

Finally, only a couple of covariates are judged statistically significant for the machine discrimination, indicating that a further reduction in dimensionality of the samples is attainable if further computational processing and memory reductions are needed. Considering that the original dimensionality of the concatenated Image channels is in the area of $70,000$ and that the feature extraction reduces that to less than 100 dimensions, it becomes apparent that the segmentation captures very discriminative areas that have now been specifically identified.

These areas offer insight into the genuine currency note design and most importantly on the shortcomings of state of the art counterfeit note construction.

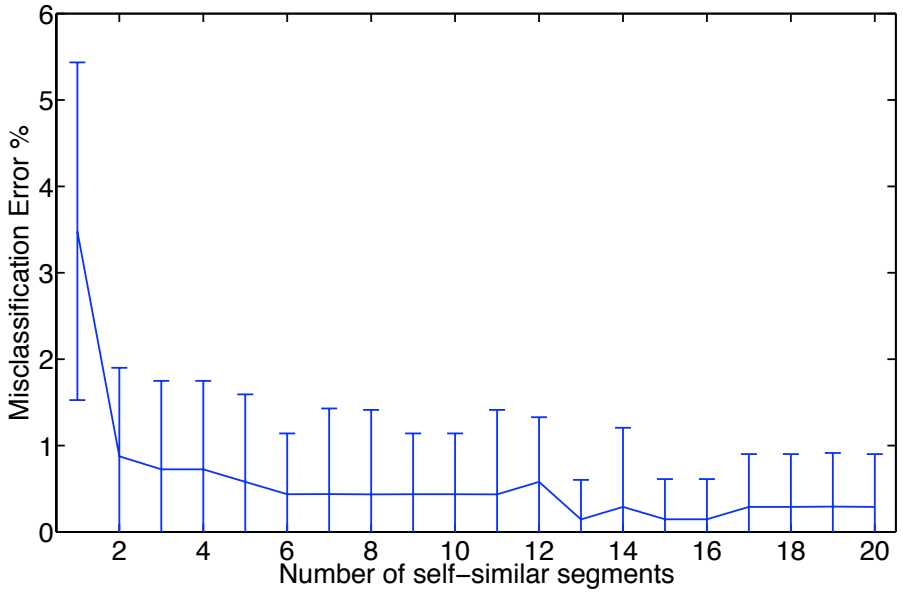

Figure 6.6: Typical error progression with the logistic regression models.

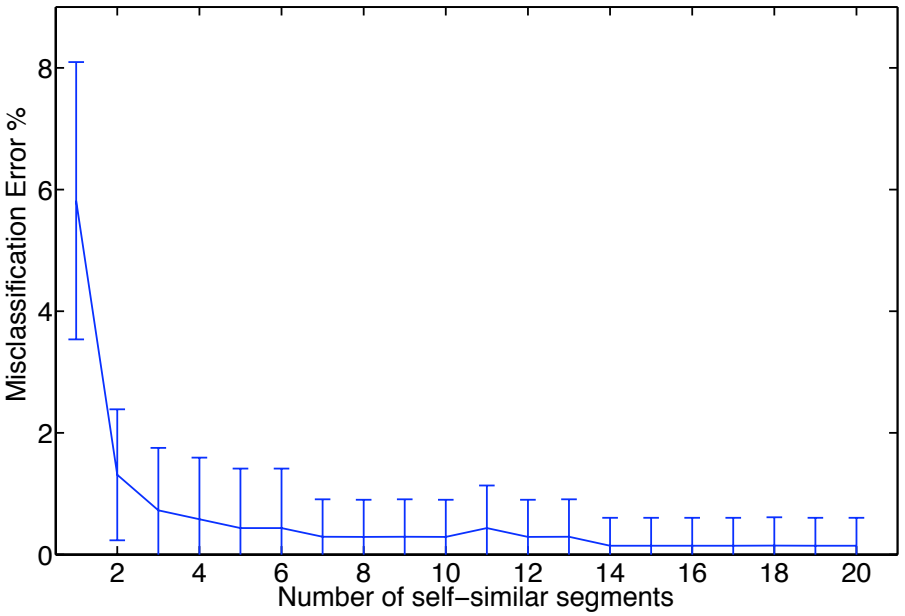

Figure 6.7: Typical error progression with the probit regression models.

6.5.2 Multinomial Classification

Following the same procedure as for the binary classification, we consider the Z-scores of the $C \times D$, where D the sample dimensions after the polynomial expansion, regression coefficients now for the three classes in order to identify discriminatory covariates. From Figure 6.8 it can be seen that only certain covariates achieve an absolute Z-score above a value of two and these indicate corresponding discriminative areas of the Image channels. It is worth noting that the 10th covariate is again significant for the discrimination of counterfeit notes, as in the binary case, but now also additional covariates contribute especially in the distinction between genuine new and genuine old notes.

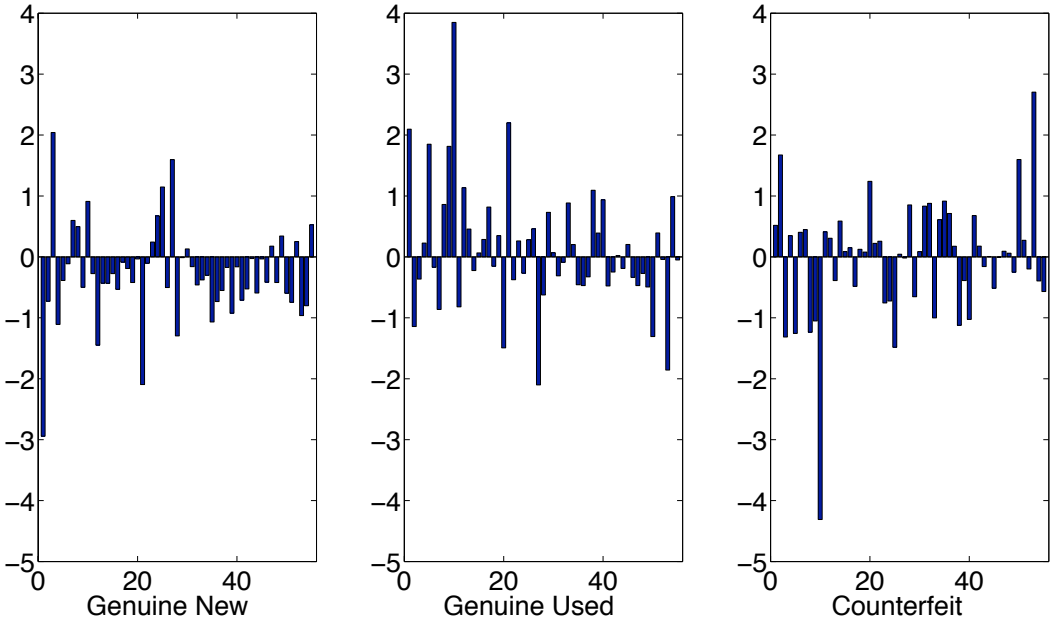

Figure 6.8: Typical Z-scores for the multiclass classification between genuine new, genuine old and counterfeit notes.

Finally, the error progressions follow the similar trend of improved performance when the number of segments increases. However the mean error progression is higher than the binary problem due to additional misclassifications between the genuine new and used classes. The initial results presented so far are for a fixed training size with an emphasis on covariate ranking; in the next sections we concentrate on the proposed pMKL methodologies and offer learning curves when varying the available training size in order to further assess the

behaviour of the proposed classifiers on the ACV problem.

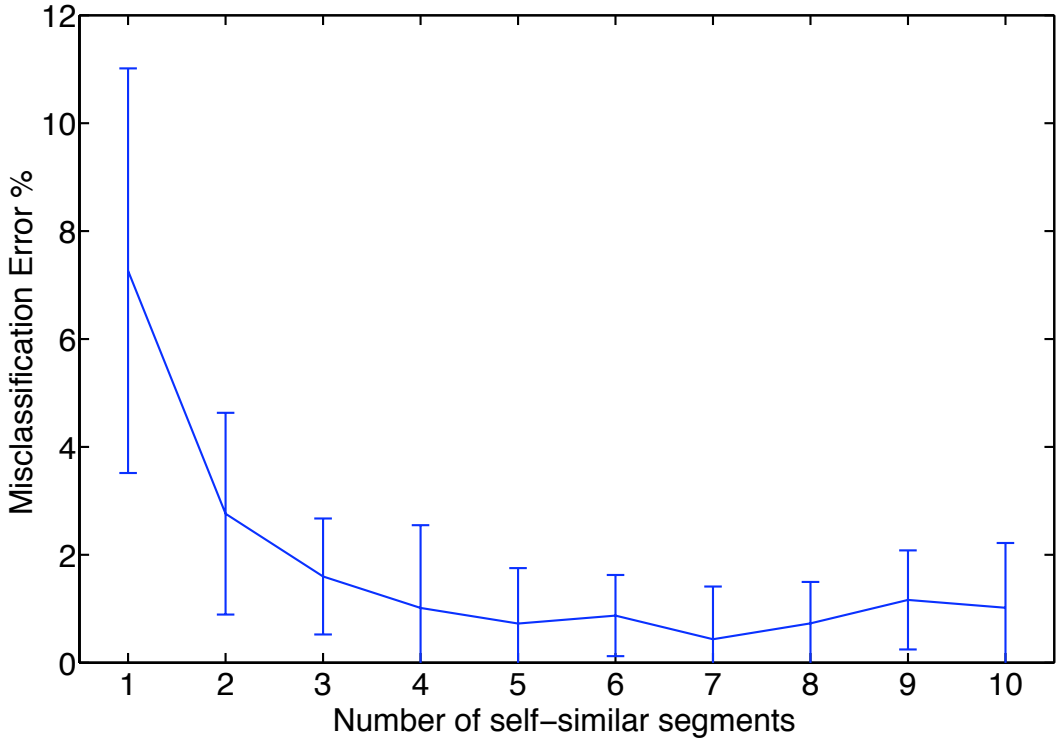

Figure 6.9: Typical error progression on the multiclass ACV case.

6.6 VBpMKL Results

In this section we present results from the variational approximation on the pMKL methodology and on various international currencies. First, the Image channels are integrated and training size dependent results are presented together with an assessment of the prediction confidence and inference of the contribution from each channel. In the next sub-section we extend the problem by including the Non-Image channels as well and further improve the classification performance measures on various currencies, denominations and orientations. These sections summarise results from the full experimental analysis in the confidential reports (Damoulas 2008b, Damoulas 2008a). Convergence is monitored via the lower bound progression at the 0.1% level with a maximum number of 100 iterations and experiments are repeated with random initialisations to offer statistics on the recorded performance measures.

6.6.1 Image Integration

The following datasets and training-test splits in Table 6.3 are considered with a bootstrap procedure of sampling a specific number of training notes and a fixed number of testing notes with replacement 20 times. Second order polynomial or Gaussian (RBF) kernels are employed as they were found to perform the best across currencies. We monitor the test error, the predictive likelihood which is our confidence in classifying a note and also the total CPU processing time. Some observed variability in CPU times is due to varying loads on different computer cluster nodes. Finally the results are presented over three different segmentation levels in order to assess the feature extraction procedure.

Currency	English £20	Chinese ¥100	US $50 (BC)
Training	10:10:400	10:10:900	10:10:200
Testing	288	500	148
Total	688	1483	348

Table 6.3: The training ranges examined for the specific fixed test size.

US $50

First the results on the US dollars dataset are given. Throughout the datasets considered, a higher segmentation level typically leads to a better (lower or smaller std) test error performance as in Figure 6.10. Such a higher segmentation level also implies more confident class membership probabilities as in Figure 6.11 and as expected higher computational costs, Figure 6.12, as the dimensionality of the constructed feature spaces increases.

The inferred kernel combination weights are presented in Figure 6.13 and as it can be seen there is a clear preference for the FS3 source , this is typical across most currencies considered, which contributes the most into the final composite kernel space. This is in agreement with past confidential work from the NCR Labs and the expected rankings of the sensor channels.

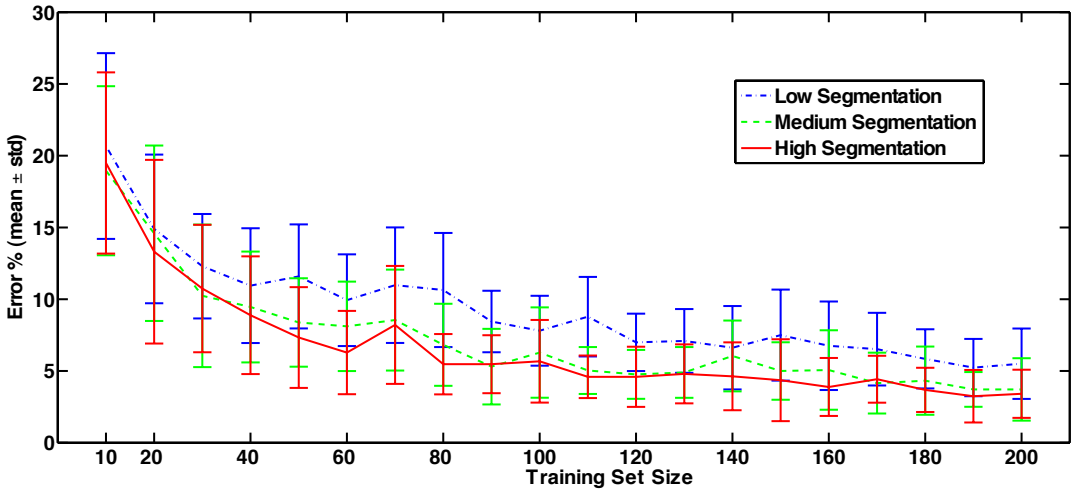

Figure 6.10: Learning curves on $50 currency notes.

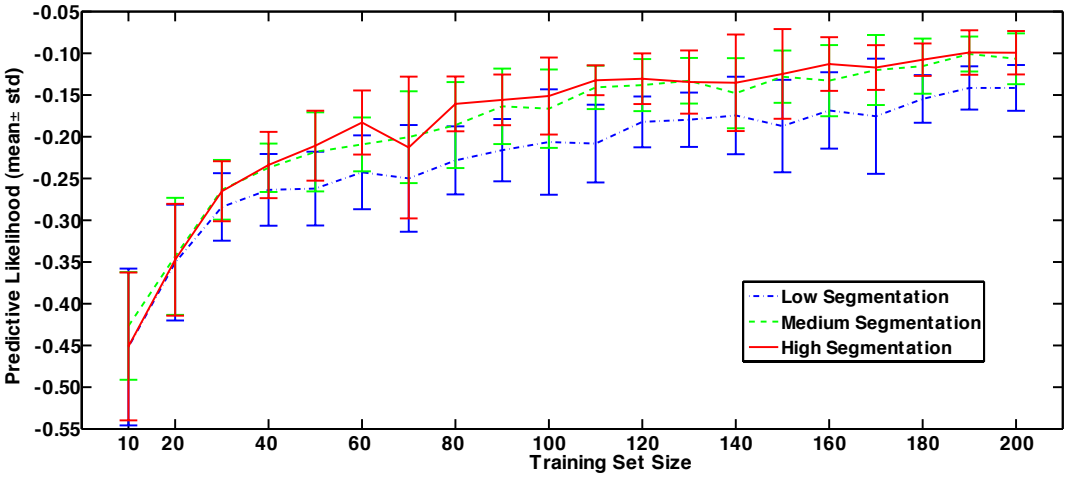

Figure 6.11: Predictive likelihood progressions for varying training size on $50.

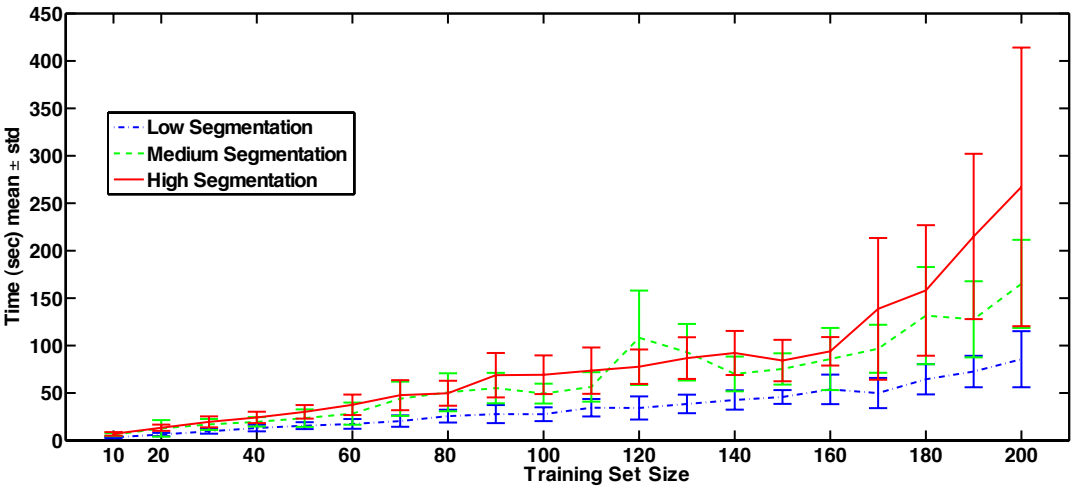

Figure 6.12: CPU time requirements for varying training size on $50.

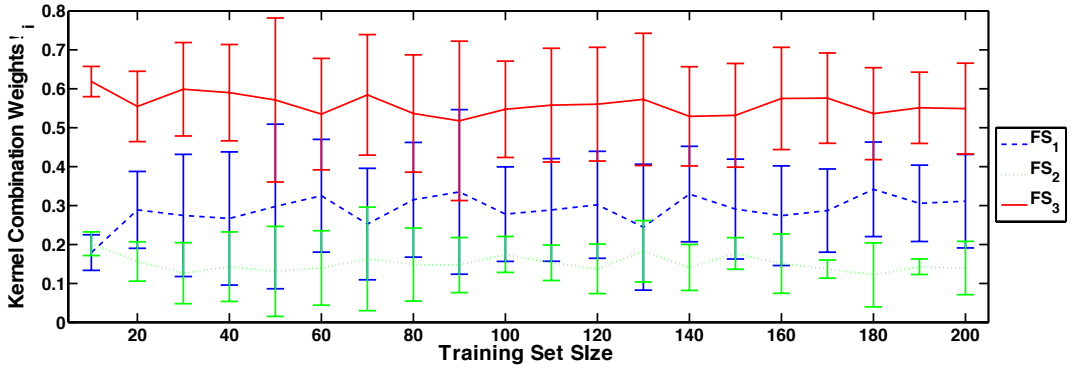

Figure 6.13: Kernel combination parameters indicating the discriminative strength of each channel.

Chinese ¥100

Next, learning curve results are presented for the Chinese ¥100 currency. The error progression in Figure 6.14 converges below an average 3% rate and this time without an improvement when increasing the segmentation level. The predictive likelihood follows the same trend, Figure 6.15, and the only significant effect of increasing the segmentation level is the additional processing requirements shown in Figure 6.16. The integration of Image channels alone is already achieving very high recognition rates that will be further improved via the introduction of the Non-Image information in the next sub-section.

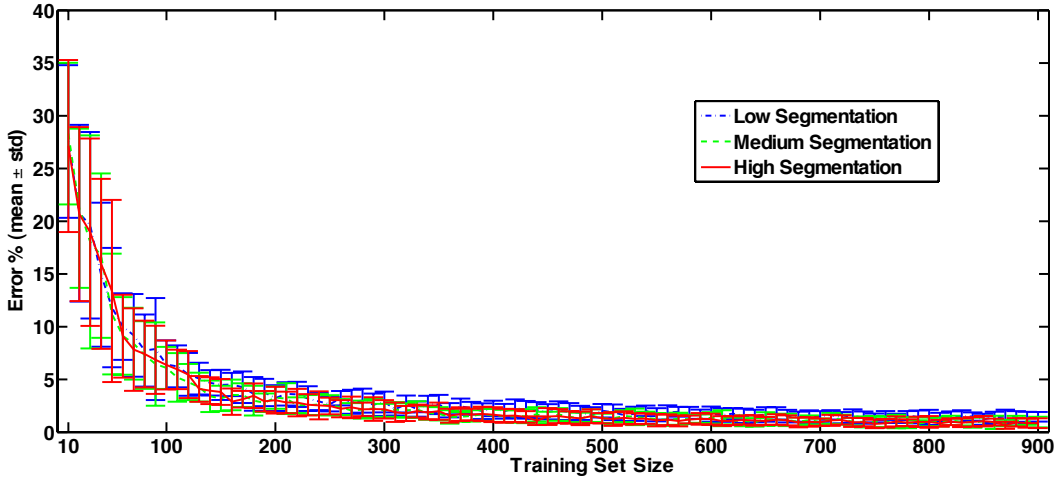

Figure 6.14: Learning curves on ¥100 currency notes.

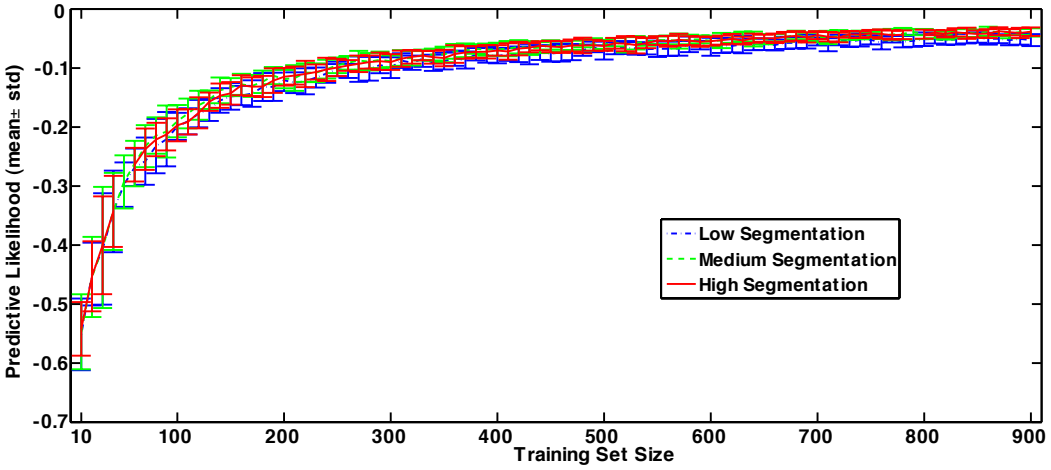

Figure 6.15: Predictive likelihood progressions for varying training size on ¥100.

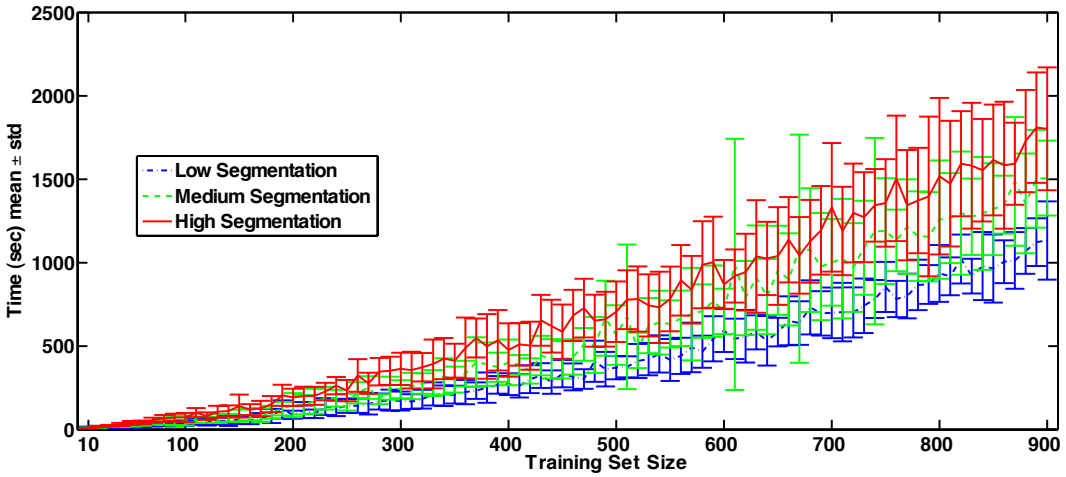

Figure 6.16: CPU time requirements for varying training size on ¥100.

English £20

Finally, the results for the £20 notes are given in Figures 6.17, 6.18 and 6.19. Similarly to the ¥100 case the error progression reaches an average rate below 3% while a low segmentation level is sufficient to achieve performances competitive to higher dimensional feature extractions. The recognition rates from the Image integration will again be improved via the further fusion of Non-Image information in the following sub-section.

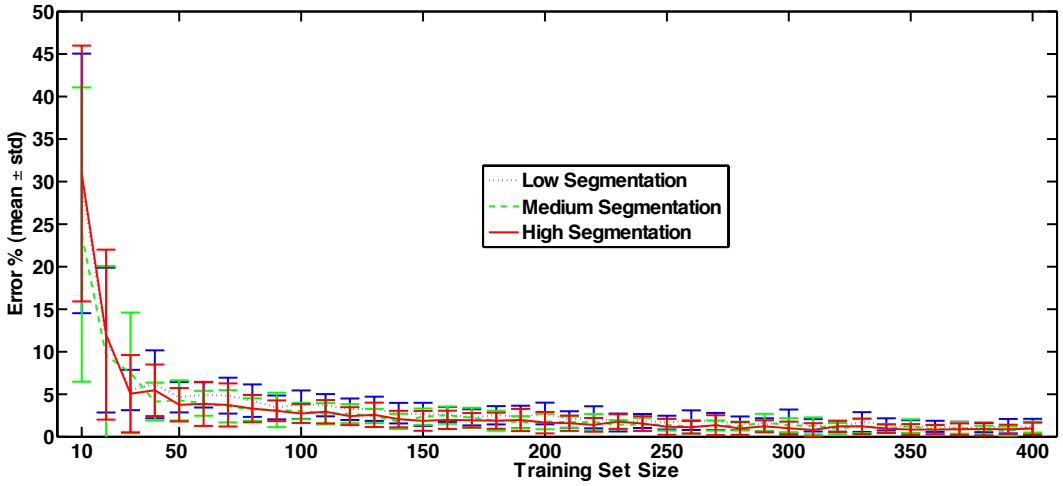

Figure 6.17: Learning curves on £20 currency notes.

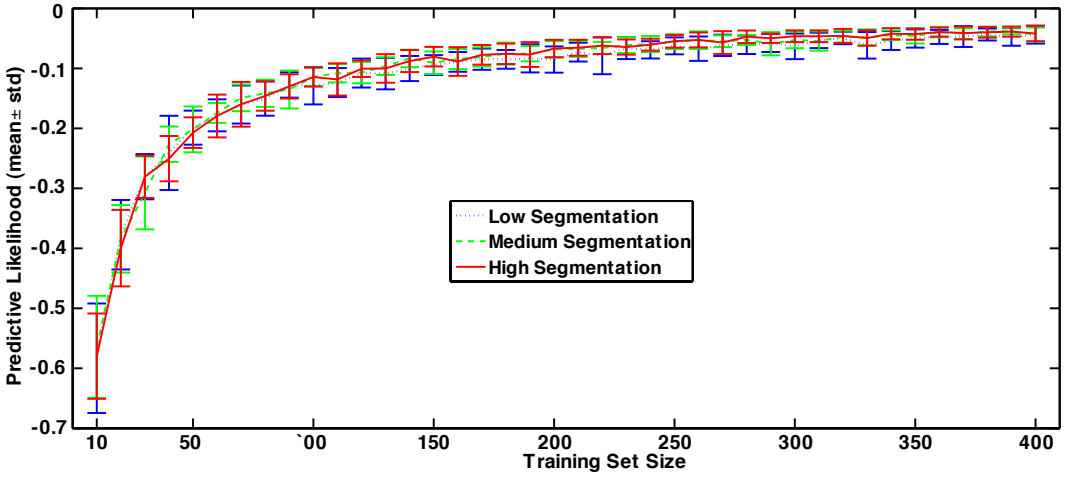

Figure 6.18: Predictive likelihood progressions for varying training size on £20.

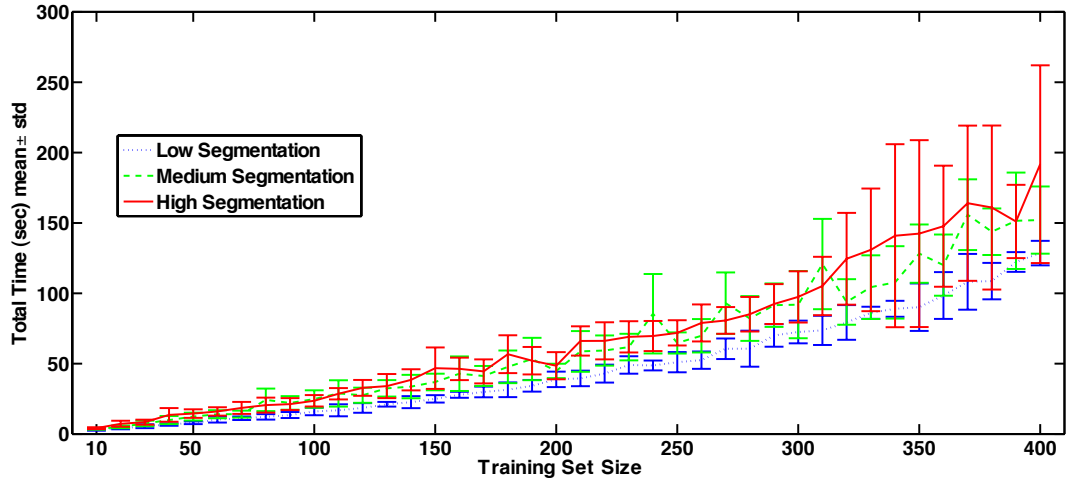

Figure 6.19: CPU time requirements for varying training size on £20.

6.6.2 Image and Non-Image Integration

Having combined the information from the Image channels the next step is to include the Non-Image signals and examine their contribution and the expected classification improvements. Following the same bootstrap setting of 20 repetitions, a fixed training and test split is now considered for every currency as described in Table 6.4. Results are presented for the best experimental settings and kernel types with the full analysis in confidential report (Damoulas 2008a).

Currency	US $50 (BC)	US $50 (BA)	Chinese ¥100	Scottish £20
Training	279	263	500	190
Testing	100	100	500	100
Total	379	363	1705	290

Table 6.4: The training/test sample sizes examined.

US $50 (BA) Front Orientation

In Table 6.5 the comparison between the Image-channels and the integration of all six channels is given. As it can be seen there is a statistically significant improvement that increases the recognition rate above the 99% level. The kernel combination parameters in Figure 6.20 show a high contribution from channels 1,4 and 5 offering additional intuition regarding the specific currency and orientation (Damoulas 2008a).

Channels:	Image Integration	Total Integration
% error (mean ± std)	1.65 ± 1.13	$\mathbf{0.65 \pm 0.67}$

Table 6.5: *Fixed Integration in US50BA* : Combination of 2nd order polynomial kernels. Comparison between Integration with Image only channels versus total Integration with additional Non-Image channels on the US $50 (BA) currency.

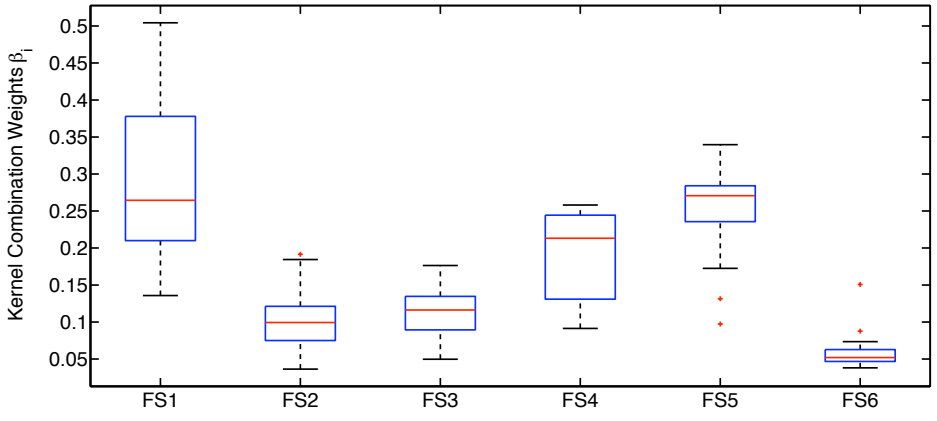

Figure 6.20: Predictive strength of fused channels on the US $50 front orientation.

US $50 (BC) Back Orientation

Similarly for the back orientation of the US $50 currency, a clear improvement is offered, Table 6.6, with an average recognition rate of 99.4%. The kernel combination parameters from Figure 6.21 indicate that different sources become discriminative for the back orientation face despite the same currency type. This is due to the different properties of the faces and specific security features that become more emphasised in different orientations. Again, confidentiality does not allow an in depth analysis but this phenomenon confirms the nature of the US dollars and their alternative name as "greenbacks".

Channels:	Image Integration	Total Integration
% error (mean ± std)	1.30 ± 0.73	$\mathbf{0.60 \pm 0.75}$

Table 6.6: *Fixed Integration in US50BC* : Combination of 2nd order polynomial kernels. Comparison between Integration with Image only channels versus total Integration with additional Non-Image channels on the US $50 (BC) currency.

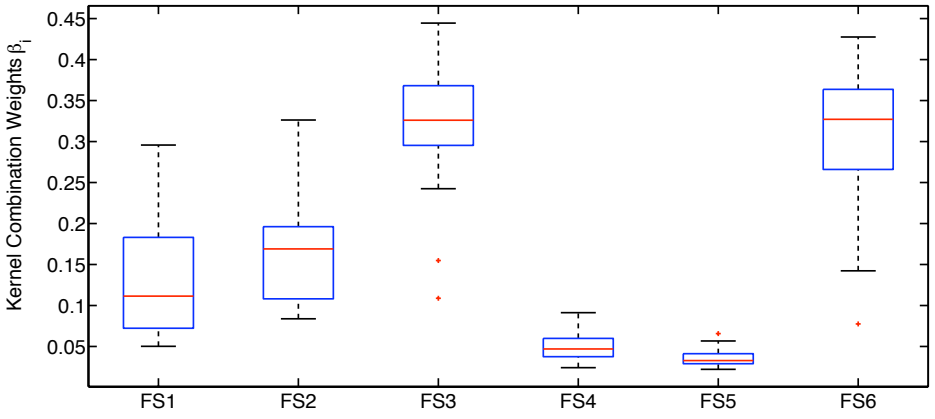

Figure 6.21: Predictive strength of fused channels on the US $50 back orientation.

Chinese ¥100

The third currency examined is the Chinese yen ¥100. The inclusion of the Non-Image channels leads to the best improvement so far as the average recognition rate increases from 97.8% to 99.4%, Table 6.7, due to the particular contribution from the Non-Image channels 5 and 6 as shown in Figure 6.22.

Channels:	Image Combination	Total Integration
% error (mean ± std)	2.71 ± 0.73	$\mathbf{0.57 \pm 0.27}$

Table 6.7: *Weighted Integration in Chinese* : Combination of 2nd order polynomial kernels. Comparison between Integration with Image only channels versus total Integration with additional Non-Image channels on the Chinese ¥100 currency.

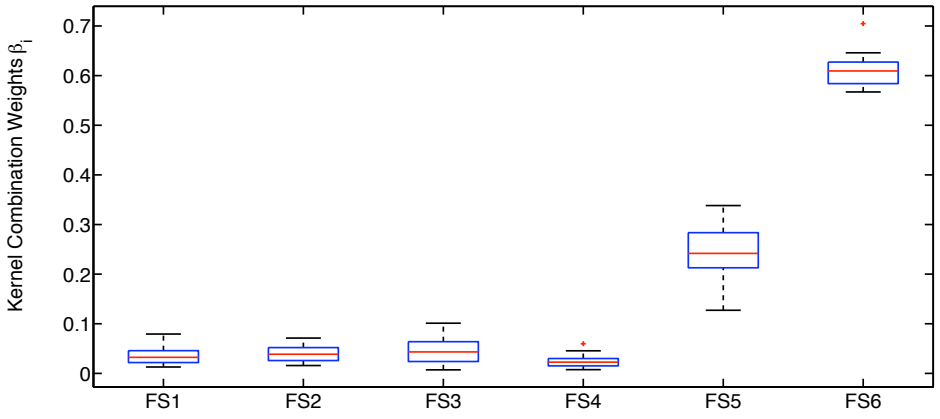

Figure 6.22: Predictive strength of fused channels on the ¥100.

Scottish £10

Finally, the Scottish £10 currency which already produces recognition rates of 99.2% with only the Image channels, Table 6.8, is again further improved to an average 99.95% level due to the contribution of the Non-Image channel 5 on the already highly discriminative Image channel 3 as depicted in Figure 6.23.

Channels:	Image Combination	Total Integration
% error (mean ± std)	0.75 ± 0.78	$\mathbf{0.05 \pm 0.22}$

Table 6.8: *Weighted Integration in SCT* : Combination of 2nd order polynomial kernels. Comparison between Integration with Image only channels versus total Integration with additional Non-Image channels on the Scottish £10 currency.

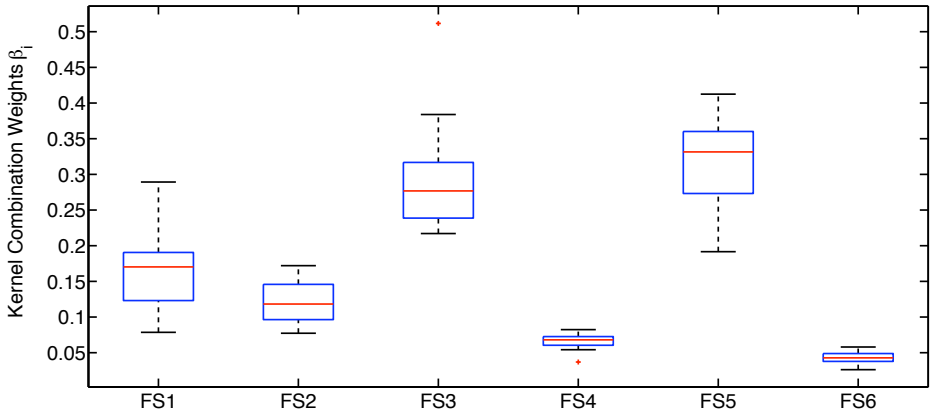

Figure 6.23: Predictive strength of fused channels on the Scottish £10 currency.

6.7 mRVM Results

In this section results are reported from the sparse pMKL methodology developed in Chapter 5 and the associated EM deterministic schemes. The same international currency datasets as in the previous section are employed and the focus is on the resulting recognition rates and sparsity levels. The latter is particularly emphasized as it leads to great computational savings in both memory and processing time, and for both training and testing phases of the algorithms. It is worth noting that the results reported here for mRVM_1 do not include the further development of informative sample selection and additional convergence measures that are described in Psorakis et al. (2010) and in Chapter 5. As a consequence, the following results from mRVM_1 can be considered sub-optimal especially on sparsity levels but still serve as an upper bound to the potential computational reduction benefits.

Second order polynomial kernels are employed across the six feature spaces as they were found to perform best from the previous methodologies. Convergence was monitored via the relative change of the regression coefficients and auxiliary variables (threshold denoted by T) in addition to maximum iteration number (denoted by "It") and a maximum proposal of samples (denoted as "It_{ML}") for mRVM_1. Experiments are repeated as previously over 20 randomly initialised trials. In Tables 6.9, 6.10, 6.11 and 6.12 the main results are depicted for all the currencies considered.

US $50 (BA) Front Orientation

Method	EM	mRVM1	mRVM2
Settings	It $= 100$, $T = 0.3$	It $= 2N$, $\text{It}_{ML} = 1$	It $= 100$, $T = 0.3$
Error %	0.6 ± 0.75	2.25 ± 1.4	1.4 ± 1.05
Relevance Vectors	All (263)	28.1 ± 6.93	8.75 ± 1.02

Table 6.9: Comparison across Methods for US $50 (BA) currency.

US $50 (BC) Back Orientation

Method	EM	mRVM1	mRVM2
Settings	It $= 100$, $T = 0.3$	It $= 2N$, $\text{It}_{ML} = 1$	It $= 200$, $T = 0.2$
Error %	0.55 ± 0.76	1.75 ± 1.3	1.4 ± 1.1
Relevance Vectors	All (279)	27.3 ± 7.9	10.6 ± 2.6

Table 6.10: Comparison across Methods for US $50 (BC) currency.

Chinese ¥100

Method	EM	mRVM1	mRVM2
Settings	It $= 100$, $T = 0.3$	It $= N$, $\text{It}_{ML} = 2$	It $= 100$, $T = 0.3$
Error %	0.32 ± 0.33	1.62 ± 0.8	1.02 ± 0.5
Relevance Vectors	All (500)	42.3 ± 9.4	14.7 ± 1.4

Table 6.11: Comparison across Methods for Chinese ¥100 currency.

Scottish £10

Method	EM	mRVM1	mRVM2
Settings	It $= 100$, $T = 0.3$	It $= 2N$, $\text{It}_{ML} = 1$	It $= 100$, $T = 0.3$
Error %	0 ± 0	0.65 ± 0.67	0.4 ± 0.7
Relevance Vectors	All (190)	20.8 ± 5.4	6 ± 0.8

Table 6.12: Comparison across Methods for Scottish £10 currency.

As it can be seen, the EM approximation retains high recognition rates competitive to the variational approximation and the mRVMs produce very sparse solutions with a minor trade-off on accuracy. The resulting Relevance Vectors (RVs) are less than 10% of the total training size and hence significantly reduce computational requirements and processing times. Considering that the results from mRVM_1 can be further improved via the informative sample selection strategies and better convergence measures that were proposed previously, these sparse models constitute a very efficient and accurate approach to the multi-feature problem of ACV with the additional benefits of probabilistic predictions.

Accuracy and Sparsity

Finally, the training size dependent error and sparsity progressions are depicted below for the considered models and on the largest dataset available (Chinese ¥100 currency). As expected in all cases the recognition rate improves when increasing the training size which especially for the sparse models indicates the possibility of retaining more descriptive RVs able to produce a better decision boundary.

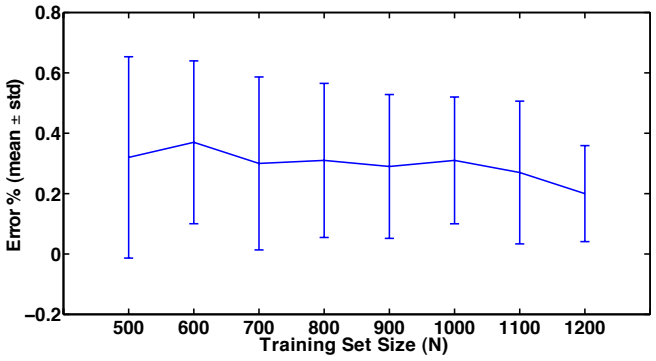

Figure 6.24: EM Estimator: Error progression while varying the training size. Fixed test size of 500 notes.

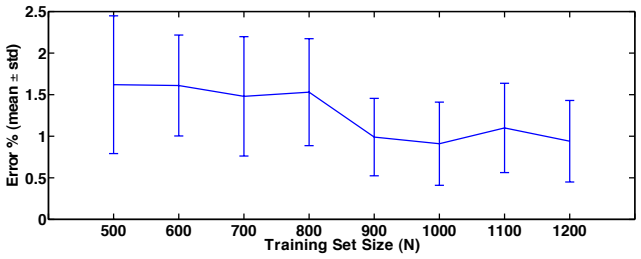

Figure 6.25: mRVM1: Error progression while varying the training size. Fixed test size of 500 notes.

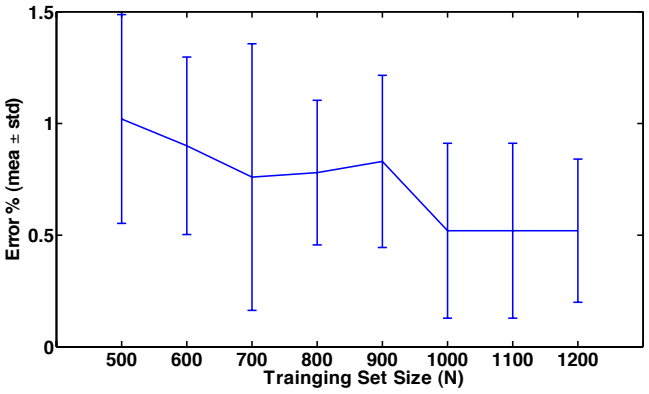

Figure 6.26: mRVM2: Error progression while varying the training size. Fixed test size of 500 notes.

Sparsity levels increase as the training size increases but the relationship in Figures 6.27 and 6.28 shows a linear trend with the mRVM_1 retaining 5% of the training set and mRVM_2 retaining just 2%.

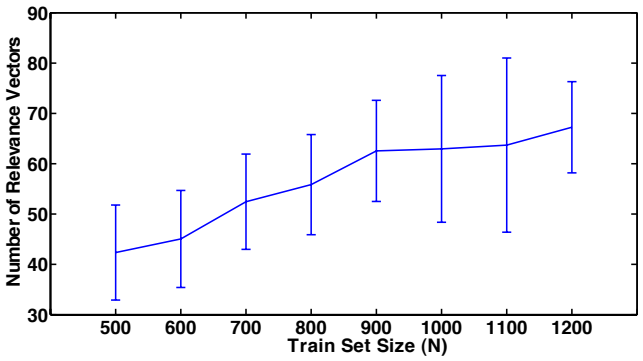

Figure 6.27: mRVM1: Sparsity progression while varying the training size.

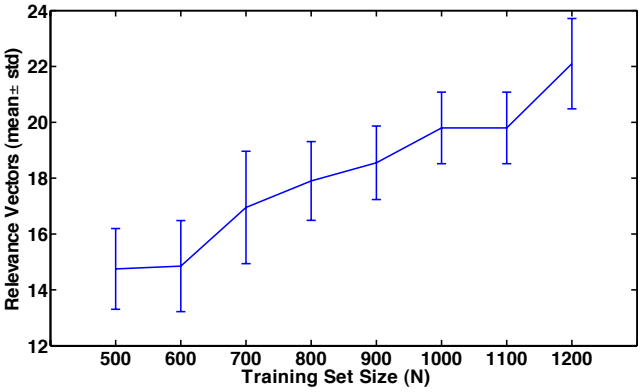

Figure 6.28: mRVM2: Sparsity progression while varying the training size.

6.8 Discussion

In this final experimental Chapter, the deterministic approximation methods VBpMKL and mRVMs, were employed to address the motivating application for this thesis, that of Automatic Currency Validation (ACV) with multiple sources of information. Following the literature review on ACV and the specific problem motivation, covariate ranking results were presented from standard generalised linear models under an MCMC or deterministic Laplace approximations.

By identifying statistically significant covariates we were able to assess the limitations of counterfeit currency and infer which specific genuine currency note features are responsible for efficient machine discrimination.

Further on, experimental results were presented for classification of a variety of international currencies producing multiple signals while being deposited in an Automated Teller Machine (ATM) and detected by the available sensors. The proposed probabilistic multiple kernel learning (MKL) methodologies were able to informatively integrate the available sensor measurements towards an overall accurate classification decision while offering reduced computational requirements through approximations and resulting sparsity.

Chapter 7

Further Large Scale Applications

This Chapter[1] presents further experimental results of all the proposed methods of this thesis. The emphasis is placed on large scale applications, in domains beyond Automatic Currency Validation, that have the same need for probabilistic multiple kernel learning (pMKL) methodology. The applications are on bioinformatics and pattern recognition problems such as protein folding prediction, hand-written numeral detection and protein sub-cellular localisation.

The main aims of the present Chapter are to offer an assessment of the various multiple kernel learning rules introduced in Chapter 3, the variational approximation introduced in Chapter 4 in comparison with the Gibbs sampling scheme of Chapter 3, the further deterministic approximations in Chapter 5 with the corresponding sparsity inducing models, and finally to provide large scale evidence of the scalability and efficiency of the proposed pMKL methods against classifier combination strategies.

The experimental results are presented in an order that follows the methodological developments of this thesis with an introduction on each of the large scale applications considered. Table 7.1 serves as a roadmap for the experiments considered in this chapter, the methodologies employed for each problem, the main problem characteristics and the experimental goal (what is demonstrated) in each section.

[1]Parts of this work have already appeared in (Damoulas and Girolami 2008, Damoulas and Girolami 2009a, Damoulas and Girolami 2009b, Damoulas and Girolami 2009c, Damoulas et al. 2008, Ying et al. 2009)

Experimental Roadmap			
Section	Methods	Characteristics	Goal
HNR	Gibbs & VBpMKL	MKL(4), MC(10)	MKL rules, MKL vs CC
PFR	VBpMKL & mRVMs	MKL(12), MC(27)	Het.MKL, MKL vs CC
RHD	VBpMKL	MKL(4), MC(54)	MKL on String Kernels
PSL	mRVMs	MKL(69), MC(4,5)	Het.MKL, Sparsity

Table 7.1: A roadmap for this Chapter regarding experiments, methods, problem main characteristics and experimental goals. Abbreviations: **HNR**-Handwritten Numeral Recognition, **PFR**-Protein Fold Recognition, **RHD**-Remote Homology Detection, **PSL**-Protein Sub-cellular Localisation, **MKL**(Sources S)-Multiple Kernel Learning, **MC**(Classes C)-Multiclass problem, **CC**-Classifier Combination methods, **Het.MKL**-Heterogeneous MKL.

7.1 Handwritten Numeral Recognition

The first application area considered is Handwritten Numeral Recognition (HNR), a well studied problem that has been heavily researched in the last decades (Chi et al. 1995, Tax et al. 2000) alongside the general problem areas of character and handwriting recognition (Tappert et al. 1990, Plamondon and Srihari 2000). The goal of HNR, as the name suggests, is to recognise (classify) handwritten numbers and the majority of the research is concentrated on improving recognition rates by proposing novel feature extraction methods (Trier et al. 1996, Shi et al. 2002) and classifier combination schemes (Tax et al. 2000) or other ensemble learning methods (Dietterich 2000b).

This work concentrates on offering an alternative to classifier combination methods on existing standard feature sets and hence further feature extraction and construction is not considered. The results reported are on the well known multiclass "Multiple Features" dataset from the UCI repository (Newman et al. 1998) which consists of features of 2000 handwritten numerals from 0 to 9 (10 classes with 200 examples each). Four different feature sets are used in accordance with past work (Tax et al. 2000), namely the Fourier descriptors (FR), the Karhunen-Loéve features (KL), the pixel averages (Pix) and the Zernike moments (ZM). In contrast with (Tax et al. 2000) we do not restrict the (ZM) features to 9 classes and allow the rotation invariance property to introduce further problems in the distinction between digits 6 and 9.

The first section of experiments employ the Gibbs sampling scheme intro-

duced in Chapter 3 and the later section performs a direct comparison between the latter and the variational approximation offered in Chapter 4.

7.1.1 Multiple Features Dataset: Gibbs Sampling

Experiments are repeated over 10 randomly initialised trials in order to report statistical properties. We report training set size dependent results for each trial by varying the number of training samples, from 10 to 100 per class, with a fixed test size of 20 test points per class. The classifier is trained on 5,000 Gibbs samples and we disregard the first 2,000 samples as burn-in period, see (Gelman et al. 2004) for details. In each trial we train individual classifiers on each set (denoted by the corresponding descriptor name, i.e FR), four standard (see Chapter 2) classifier combination methods (denoted by a C suffix), four proposed kernel combination methods (denoted by a K suffix) and a concatenation of all the object descriptors leading to a single kernel (denoted as *Single*). The abbreviations for the methods are given in Table 7.2.

Abbreviation	**Method**
Prod C	Product of Class Probabilities from Individual Classifiers
Sum C	Sum of Class Probabilities from Individual Classifiers
Max C	Maximum of Class Probabilities from Individual Classifiers
Maj C	Majority of Class Assignment from Individual Classifiers
Single K	Concatenating features into single Kernel
Fix K	Fixed Combination of Kernels
Weighted K	Convex Linear Combination of Kernels
Prod K	Product Combination of Kernels
WProd K	Weighted Product Combination of Kernels
Bin K	Binary Combination of Kernels

Table 7.2: Abbreviated names of ensemble methods.

In this section it is shown that the proposed pMKL approach with the Gibbs sampling method improves upon the individually trained classifiers on specific feature spaces, and matches the *best* performing classifier combination schemes while outperforming the rest. Furthermore, learning curves are offered for the various kernel combination rules and the methodology identifies *complementary* feature spaces for the handwritten numerals problem that explain previously reported results in (Tax et al. 2000). Throughout this experimental study, Gaussian (RBF) kernels with fixed kernel parameters $\theta_{sd} = 1/D^s$ are employed in

order to minimise the computational cost for producing learning curves from the Gibbs sampler.

In Figure 7.1 the performances of the classifier combination schemes are depicted and in Figure 7.2 the best performing of these is compared against the individually trained classifiers. We can see that the product and sum combination rule outperform all the individual classifiers and the maximum and majority combination schemes. The best performances offered by the two rules are in the range of $2.2 \pm 1\%$ with a single best performance achieving 0.5% error. Of the individual classifiers, the classifier trained on the Pix feature set performs the best followed by the KL classifier. The disagreement with (Tax et al. 2000) on the preference between the Pix and ZK is due to the class restriction employed in their work.

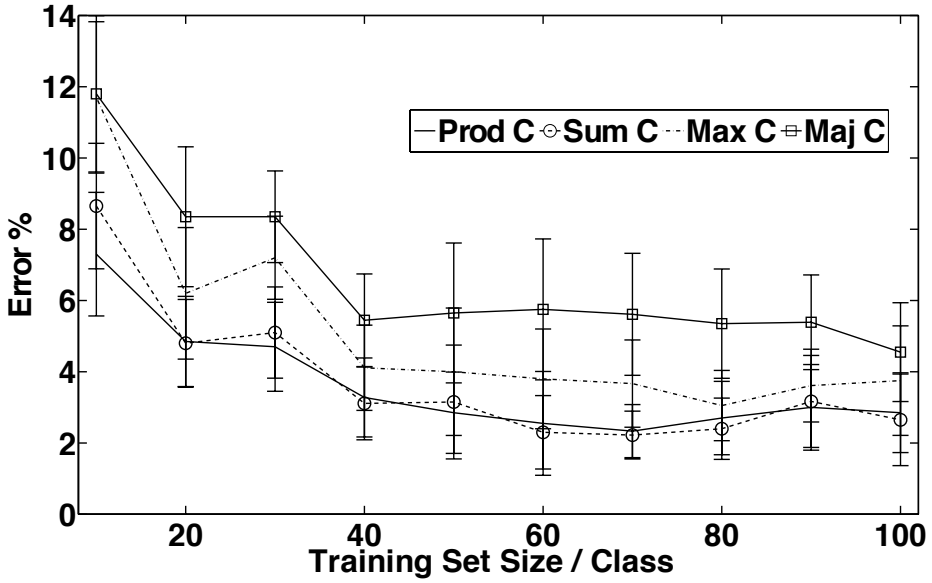

Figure 7.1: Performance of the classifier combinations (Prod C, Sum C, Max C, Maj C).

Turning now our attention to the kernel combination rules proposed in this thesis, we plot in Figure 7.3 their learning curves and in Figure 7.4 the best performing kernel combinations against the best performing classifier combination method (Prod C). First we can observe that when the training size is very small (10 samples per class), the weighting schemes for kernel combination (i.e. Weight and Bin) perform somewhat worse than a fixed combination of kernels. This is supported by the intuition that weighting schemes require more

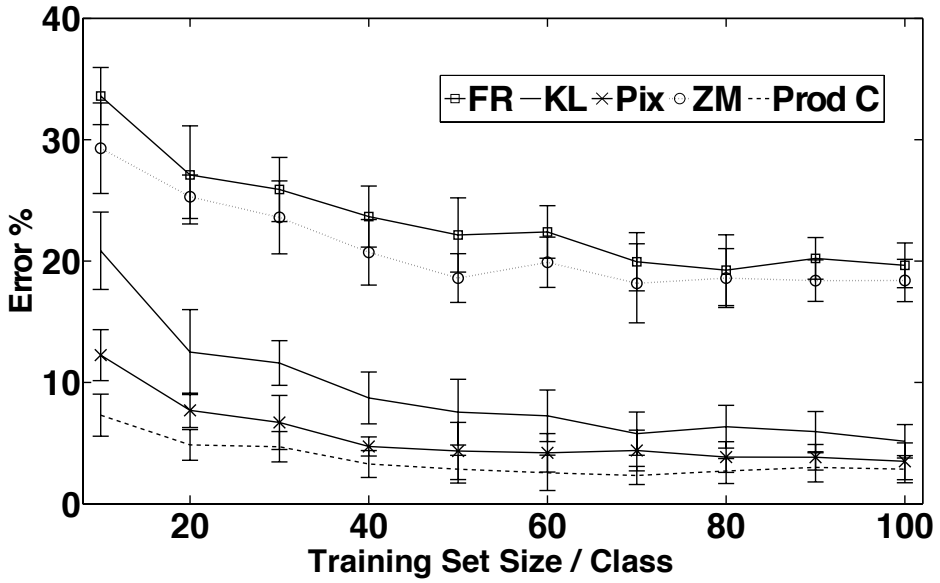

Figure 7.2: Performance of the individual classifiers (FR, KL, Pix, ZM) against the best classifier combination (Prod C).

evidence as the parameter space increases. Also, the product, and to a lesser extent the sum classifier combination methods perform significantly better in such small training size cases than the kernel combination approaches. This is due to the aforementioned drawback of weighting schemes and also on the fact that fixed kernel combination rules can be seen as combining information on the *prior* class membership level (the composite kernel is fixed from the start and carries prior class membership probabilities through the resulting similarities) whereas classifier combination methods operate on an *a posteriori* class membership level hence gaining on diversity as the information "bottleneck" occurs after the training regime.

When the training size is increased, the convex linear combinations of kernels match the fixed combination (while still offering insight on the significance of the sources) and the kernel combination methods match the *best* performing classifier combination methods and significantly outperform the best individual classifier (Pix) and the Maj and Max classifier combination rules.

Finally, it is worth noting that the binary kernel combination fails (on average) to improve on the pixel classifier when the training set is very small. This is due to the limited number of training samples and the "hard" nature of the binary switch which decides to effectively "switch off" certain kernels.

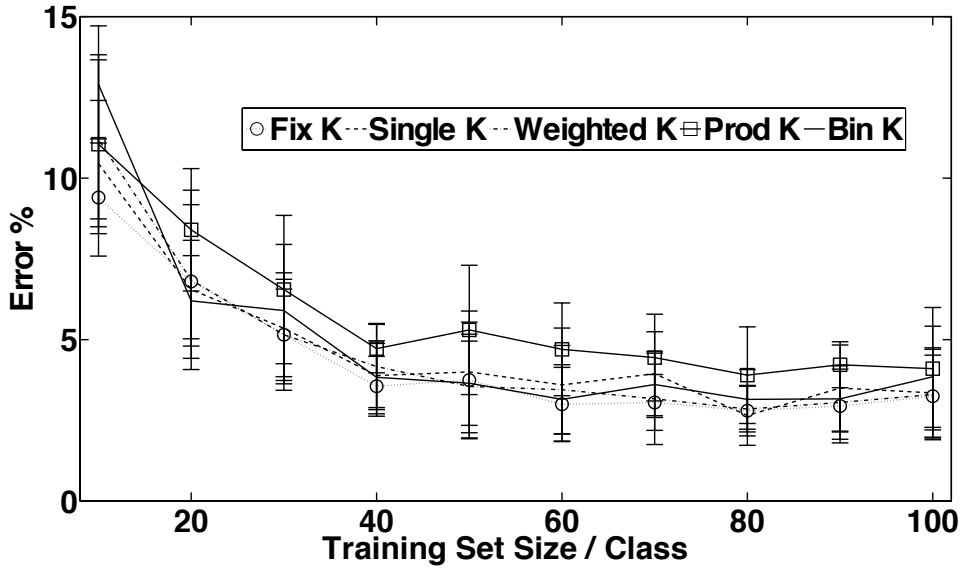

Figure 7.3: Performance of kernel combination methods (Fix K, Bin K, Weighted K, Prod K) and the single kernel (Single K).

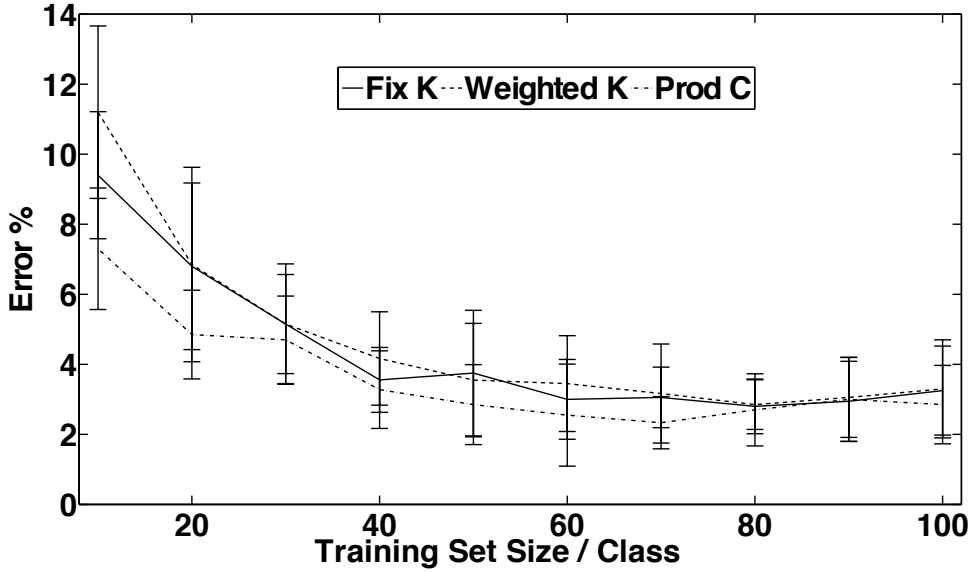

Figure 7.4: Performance of the best kernel combination methods (Fix K, Weighted K) and the best performing classifier combination method (Prod C).

The combinatorial weights $\boldsymbol{\beta}$ for the convex linear combination rule can give us an insight into the relative importance of base kernels and furthermore indicate which sources complement each other and are selected to be combined. In Figure 7.5 we plot the progression (mean $\pm$ std) of the weights, at the end of the Gibbs sampling procedure, for the training/test range and over the 10 trials.

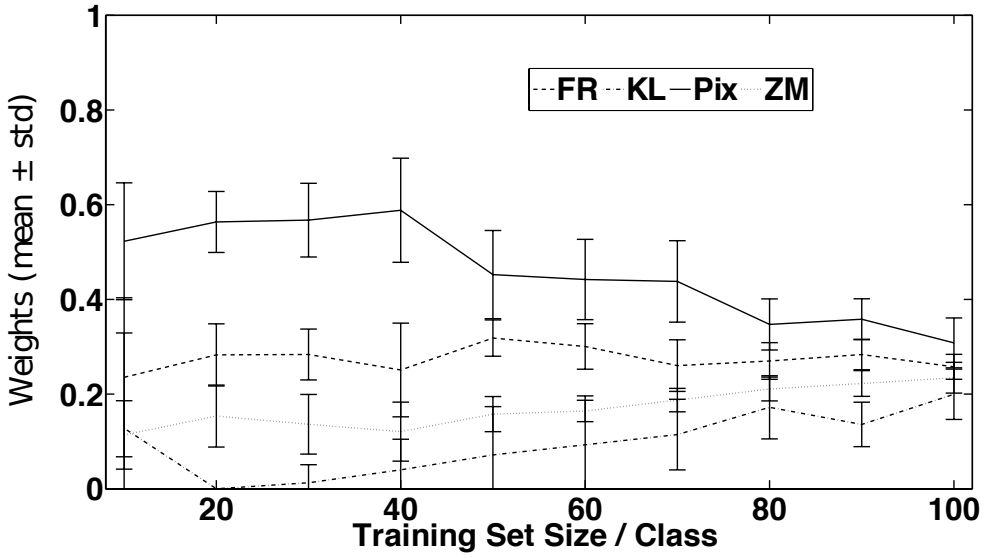

Figure 7.5: The mean and std of the multiple kernel weights from the convex linear method (Weighted K).

As expected, the pixel kernel (Pix) is the most important one followed by the fourier (FR) one. This is in agreement with the comparative performances of the individual classifiers. However, the Karhunen-Loéve kernel receives on average the lowest weight although it is the second best individual classifier and the Fourier kernel is weighted as more important for the combination. This is possibly due to lack of information diversity between the Pixel and the KL source. The underlying phenomenon is also depicted in the work of (Tax et al. 2000) where the performance reported is better by removing the Karhunen-Loéve classifier than by removing the Fourier one from the combination of classifiers.

Furthermore, it is interesting to note that for small training sizes the weights clearly prefer the two top-performing base kernels but as more objects are included the weighting scheme seems to converge towards a $1/S$ weighting. This is in agreement with our intuition for this specific problem as there are no counter-

informative base kernels and for small training sizes certain descriptors might be clearly preferred on the basis of their discriminative power for that training set but as more objects are presented, other complementary features start playing a significant role as well.

Finally, the zero-one loss match between the weighted and fixed kernel combination in Figure 7.4 is in agreement with previous work on kernel combination where a fixed procedure is as-good-as a weighted combination (Lewis et al. 2006b, Girolami and Zhong 2007) regarding the zero-one loss performance. This phenomenon is addressed in Chapter 8 via the theoretical analysis of multiple kernel learning methods.

7.1.2 Multiple Features Dataset: Variational Bayes

In this section we revisit the HNR problem but this time extending the comparison between the proposed pMKL methods and classifier combination schemes by also including the variational Bayes approximation (VBpMKL) that was introduced in Chapter 4. The aim is to assess any possible degradation of performance introduced by the deterministic approximation with respect to the full MCMC Gibbs sampling solution.

In Tables 7.3, 7.4, 7.5 and 7.6, we report experimental results over 50 repeated trials where we have randomly selected 20 training and 20 testing objects from each class. For each trial we employ a single classifier on each feature space, the classifier combination schemes and the proposed pMKL methods. In all cases we employ Gaussian (RBF) kernels with the aforementioned fixed parameters and for the VBpMKL method we monitor the lower bound convergence at the 0.1% level with a maximum of 100 iterations.

MCMC Gibbs sampling on Individual Feature Sets			
FR	KL	PX	ZM
27.3 ± 3.3	11.0 ± 2.3	7.3 ± 2	25.2 ± 3

Table 7.3: Results on HNR from individual classifiers.

The conclusions drawn for the comparisons between the Gibbs sampling scheme, the individual classifiers and the classifier combination schemes follow from the previous section. The interest is now on the variational approximation which, from Table 7.6, is shown to perform very well compared with the MCMC

MCMC Gibbs sampling for Combining Classifiers			
Prod C	Sum C	Max C	Maj C
5.1 ± 1.7	5.3 ± 2	8.4 ± 2.3	8.45 ± 2.2

Table 7.4: Results on HNR when combining classifiers.

MCMC Gibbs sampling for pMKL				
Bin	Fix K	Weighted K	Prod K	WProd K
5.7 ± 2	5.5 ± 2	5.8 ± 2.1	5.2 ± 1.8	5.9 ± 1.2

Table 7.5: Results on HNR with the pMKL methods.

Gibbs solution and with a standard t-test p-value of 0.47, between them there is no statistical difference.

VBpMKL				
Bin K	Fix K	Weighted K	Prod K	WProd K
5.53 ± 1.7	4.85 ± 1.5	6.1 ± 1.6	5.35 ± 1.4	6.43 ± 1.8

Table 7.6: Results on HNR with the VBpMKL methods.

Hence, from this section it is clear that pMKL methods are competitive with, and in cases outperform, classifier combination schemes with the additional benefit of inferring the significance of the contributing information sources. Furthermore, the variational approximation appears to retain classification performance levels when compared with the "full" MCMC Gibbs sampling approach.

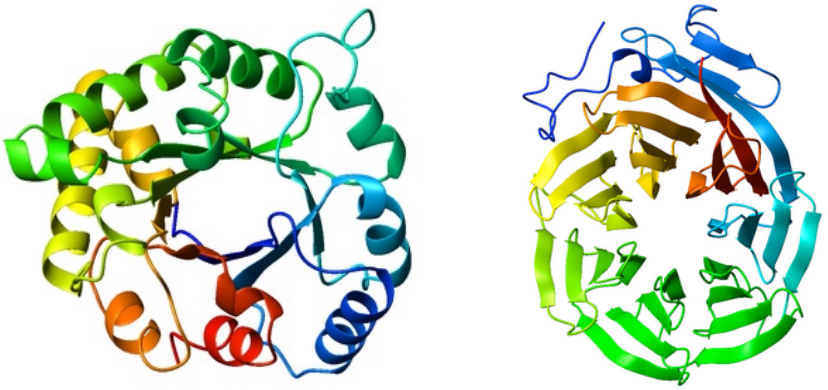

Figure 7.6: Tim-barrel 7-bladed beta-propeller
Image Source: Wikipedia under a GNU Free Documentation License.

7.2 Protein Fold Recognition

Much effort has been directed to the prediction of the three-dimensional structures of proteins for which no experimental structures are available (Baker and Sali 2001). Where there is sequence similarity to proteins of known structure, a comparative matching procedure is often adopted. However, where no such sequence similarity exists, the prediction problem is formidable, not least because the overall structure may be unlike that of any protein, the structure of which has been determined.

In this context, one approach, known as the *taxonomic* approach (Ding and Dubchak 2001, Shen and Chou 2006), has been to divide the problem of determining the overall three-dimensional structure into that of determining its 'fold'. The term 'fold' is used to denote a particular arrangement of a specific number of secondary structure components (usually alpha-helices and beta-strands) that is the basis of the overall structure of several different proteins which may have little or no amino acid sequence similarity. The appearances of some of these arrangements have given rise to names like 'barrel', 'bundle', 'sandwich' and 'propeller', although these tend to encompass several more specific folds e.g. the TIM beta/alpha barrel and the 7-bladed beta-propeller, Figure 7.6. Hence, protein fold prediction can be seen as a challenging multiclass recognition problem where proteins are classified into folds based on their characteristics and available measurements.

Past work on the problem of predicting protein folds has employed artificial

neural networks (ANNs), support vector machines (SVMs), Bayesian networks, Hidden Markov Models and k-nn classifiers (Chou and Zhang 1995, Dubchak et al. 1995, Jaakkola, Diekhans and Haussler 1999, Raval et al. 2002) with varying success. In (Ding and Dubchak 2001) an extensive study on a publicly available data-set, consisting of 27 SCOP folds (Lo Conte et al. 2000, Andreeva et al. 2004), was conducted exploring the use of various multiclass adaptations of the well-known binary SVM classifier methodology. In that work, the best methodology for combining binary SVMs was identified for the particular problem giving an accuracy of 56%, and furthermore, via an extensive experimental procedure the most *predictive* protein characteristics were selected from the initial group considered. These were found to be the amino-acid composition (C), the secondary structure (S) and the hydrophobicity (H).

Recently, (Shen and Chou 2006) proposed two modifications to the method of (Ding and Dubchak 2001) that raised the best performance accuracy from 56% to 62.1%. Firstly, they proposed a somewhat *ad-hoc* ensemble learning approach where multiclass k-nn classifiers individually trained on each feature space (such as C or S) were later combined and secondly, they proposed the use of 4 additional feature groups to replace the amino-acid composition. These were pseudo-amino acid compositions (Chou 2005) designed to capture sequence-order effects by using a correlation function between hydrophobicity and hydrophilicity in different intervals of the protein sequence.

In this thesis, I concentrate on the same benchmark dataset of (Ding and Dubchak 2001) with the extra groups of features proposed by (Shen and Chou 2006) and also include sequence-alignment[2] features via a *pairwise* kernel (Liao and Noble 2003), which essentially describes the sequence based similarity of the proteins. The VBpMKL method is employed as a single multiclass multi-kernel machine that is able to operate on all of these groups of features simultaneously and instructively combine them. This offers a new and efficient way of incorporating multiple feature characteristics of the proteins without an increase in the number of required classifiers. In addition, the importance and predictive power of the pseudo-amino acid compositions proposed by (Shen and Chou 2006) together with all the other available characteristics are assessed and hence further insight is gained on the protein fold recognition problem.

The best performance reported on the SCOP PDB-40D benchmark data-set

[2]Despite the apparent low homology dataset

is a 70% accuracy by combining all the available feature groups from global protein characteristics but also including sequence-alignment features. We offer an 8% improvement on the previously best reported performance that combines binary SVM classifiers while at the same time reducing computational costs and assessing the predictive power of the various available features.

7.2.1 Experimental Setup

The approach adopted is based on the motivation to reduce the number of classifiers needed for such challenging multiclass recognition problems where multiple feature sets are available, while improving performance. Combining binary classifiers as in the work by (Ding and Dubchak 2001) heavily increases the computational resources needed since, e.g for the best performing all-vs-all method, we need to deploy $S \times \frac{C(C-1)}{2} = 2106$ classifiers, where S is the number of feature spaces or *sources* (only 6 in their work) and C the number of classes.

Furthermore, even when employing multiclass classifiers in an ensemble learning framework such as the one proposed by (Shen and Chou 2006), we still need as many classifiers as there are available feature spaces. Considering the nature of the protein fold prediction problem, where the fold type of a protein can depend on a large number of protein characteristics and also noting that even in the *taxonomic* approach the number of fold types already approaches the thousand boundary, it is straightforward to see the need for a methodological framework that can cope with a large number of classes and can incorporate as many as there are available feature spaces while assessing their informational content.

The original dataset[3] from (Ding and Dubchak 2001) (based on SCOP PDB-40D) consists of 313 proteins for training and 385 proteins for testing with less than 35% sequence identity between any two proteins in the training and the test set. Furthermore, the extensions proposed by (Shen and Chou 2006) exclude 4 proteins from the original dataset, namely proteins 2SCMC and 2GPS from the training set plus 2YHX_1 and 2YHX_2 from the test set, due to lack of sequence records.

The 27 SCOP fold types (Dubchak et al. 1995) together with the original feature spaces in (Ding and Dubchak 2001), the 4 proposed by (Shen and Chou 2006) which describe pseudo-amino acid compositions (PseAA) estimated

[3]Available at http://crd.lbl.gov/-cding/protein.

on different intervals of the protein sequence, and the two local alignment Smith-Waterman (SW) based feature spaces, with different scoring matrices, are described in Tables 7.7 and 7.8.

(1) globin-like	(15) lipocalins
(2) cytochrome c	(16) TIM-barrel
(3) DNA binding 3-helical bundle	(17) FAD-binding motif
(4) 4-helical up-and-down bundle	(18) flavodoxin-like
(5) 4-helical cytokines	(19) Rossmann fold
(6) EF-hand	(20) P-loop
(7) immunoglobulin-like	(21) thioredoxin-like
(8) cupredoxins	(22) H-like motif
(9) viral coat & capsid proteins	(23) hydrolases
(10) conA-like glucanases	(24) periplasmic binding protein-like
(11) SH3-like barrel	(25) β-grasp
(12) OB-fold	(26) ferredoxin-like
(13) beta-trefoil	(27) small inhibitors, toxins, lectins
(14) trypsin-like serine proteases	

Table 7.7: Fold types (27 classes) in the dataset

Feature	Employed in	Dim
Amino Acid Composition (C)	D&D	20
PseAA $\lambda = 1$ (λ_1)	S&C	22
PseAA $\lambda = 4$ (λ_4)	S&C	28
PseAA $\lambda = 14$ (λ_{14})	S&C	48
PseAA $\lambda = 30$ (λ_{30})	S&C	80
Predicted Secondary Structure (S)	Both	21
Hydrophobicity (H)	Both	21
van der Waals volume (V)	Both	21
Polarity (P)	Both	21
Polarizability (Z)	Both	21
SW with BLOSUM62 (SW_1)	None	N
SW with PAM50 (SW_2)	None	N

Table 7.8: The 12 Feature spaces. Sequence-alignment based features were computed with different gap penalties: SW_1 with scoring settings from Liao and Noble (2003) and SW_2 with penalties of 0.8.

7.2.2 Results and Discussion

Reported results are averaged over 20 (fold recognition) randomly initialised trials in order to obtain statistical measures of accuracy and precision. We monitor convergence via the lower bound to the marginal likelihood and convergence is assumed when there is less than 0.01% increase of the lower bound progression or when a maximum of 100 iterations have been completed. Throughout this study, second order polynomial kernels for the global characteristics and inner product kernels for the local characteristics (SW) are employed as they were found to provide a better embedding of the feature spaces. CPU times reported are for a 2 GHz Intel based PC with 2Gb RAM running Matlab codes.

First, the performance from individual feature spaces is given to gain an overall understanding of their predictive abilities. This however does not draw the complete picture as complementary information may be shared across sources achieving low performances. In Table 7.9 the mean percentage accuracy with standard deviations from the proposed method (VBpMKL) is presented, together with the *best* ones reported by (Ding and Dubchak 2001) on the original dataset.

Table 7.9: Average Individual Feature Space Percentage Accuracy

Feature Space	VBpMKL	Ding and Dubchak
Amino Acid Composition (C)	51.2 ± 0.5	44.9
Predicted Secondary Structure (S)	38.1 ± 0.3	35.6
Hydrophobicity (H)	32.5 ± 0.4	36.5
Polarity (P)	32.2 ± 0.3	32.9
van der Waals volume (V)	32.8 ± 0.3	35
Polarizability (Z)	33.2 ± 0.4	32.9
PseAA $\lambda = 1$ (λ_1)	41.5 ± 0.5	-
PseAA $\lambda = 4$ (λ_4)	41.5 ± 0.4	-
PseAA $\lambda = 14$ (λ_{14})	38 ± 0.2	-
PseAA $\lambda = 30$ (λ_{30})	32 ± 0.2	-
SW with BLOSUM62 (SW_1)	59.8±1.9	-
SW with PAM50 (SW_2)	49±0.7	-

Regarding the original features employed by (Ding and Dubchak 2001) we are in agreement with their observations as the best performing feature space, seems to be the amino acid composition (C). The $\lambda = 1$ and $\lambda = 4$ PseAA achieve the second best *global* individual performance and as the "step" λ increases further, the individual performances decrease. Although according to

(Shen and Chou 2006) the PseAA composition "has the same form as the conventional amino acid composition, but contains much more information" it seems at this stage that none of the PseAA is as predictive as the conventional amino acid composition. Furthermore, the local characteristics (SW) surprisingly outperform every global one and SW_1 achieves a higher accuracy than the best SVM-combinations proposed by (Ding and Dubchak 2001). This is because although most of the proteins have less than 35% sequence similarity, this seems to be an adequate similarity level to achieve good accuracy.

In Table 7.10 the effect of sequentially adding the feature spaces in the order of (Ding and Dubchak 2001) is presented and extended to the addition of the PseAA compositions and finally the sequence similarity based features. The comparison is against the best performing SVM combination methodology as reported in (Ding and Dubchak 2001) and the ensemble method of (Shen and Chou 2006). As we can see in all the steps the proposed method outperforms the best reported accuracies and offers the current *state-of-the-art* in this data-set.

Table 7.10: Effect of F.S combination. % Accuracy reported.

Feature Spaces	VBpMKL	Ding & Dubchak (AvA)
C	51.2 ± 0.5	44.9
CS	55.7 ± 0.5	52.1
CSH	57.7 ± 0.6	56.0
CSHP	57.9 ± 0.9	56.5
CSHPV	58.1 ± 0.8	55.5
CSHPVZ	58.6 ± 1.1	53.9
CSHPVZλ_1	60 ± 0.8	-
CSHPVZ$\lambda_1\lambda_4$	60.8 ± 1.1	-
CSHPVZ$\lambda_1\lambda_4\lambda_{14}$	61.5 ± 1.2	-
CSHPVZ$\lambda^1\lambda^4\lambda^{14}\lambda^{30}$	62.2 ± 1.3	-
CSHPVZ$\lambda^1\lambda^4\lambda^{14}\lambda^{30}SW_1$	66.4±0.8	-
CSHPVZ$\lambda^1\lambda^4\lambda^{14}\lambda^{30}SW_1SW_2$	68.1±1.2	-
		Shen & Chou
SHPVZ$\lambda_1\lambda_4\lambda_{14}\lambda_{30}$	61.0 ± 1.4	62.1

The best performances can be seen in Table 7.12 in comparison with the best ones reported in the cited past work. An improvement over both past methods is achieved while employing a single multiclass kernel machine without resorting to combinations of multiple binary classifiers. The average CPU times can be seen in Table 7.11 together with standard deviations.

The convex linear combination of base kernels is able to infer the significance

F.S Combination	$\mu \pm \sigma$ over 20 runs
CSHPVZ	2,243 ±485
SHPVZ$\lambda_1\lambda_4\lambda_{14}\lambda_{30}$	2,844±644
CSHPVZ$\lambda_1\lambda_4\lambda_{14}\lambda_{30}$	2,713±453

Table 7.11: CPU times (sec) for the VBKC

Table 7.12: Best single run performances (% Accuracy)

Feature Spaces	Ding & Dubchak	Shen & Chou	VBpMKL
CSHP	56.5	-	59.3
SHPVZ$\lambda_1\lambda_4\lambda_{14}\lambda_{30}$	-	62.1	63.5
CSHPVZ$\lambda_1\lambda_4\lambda_{14}\lambda_{30}$	-	-	63.9
CSHPVZ$\lambda^1\lambda^4\lambda^{14}\lambda^{30}SW_1SW_2$	-	-	70
No. of Classifiers	2,106	9	1

of the corresponding feature descriptions. In Figure 7.7 a summarising plot of the weights over 20 runs depicting the lower quartile, median, and upper quartile values is given.

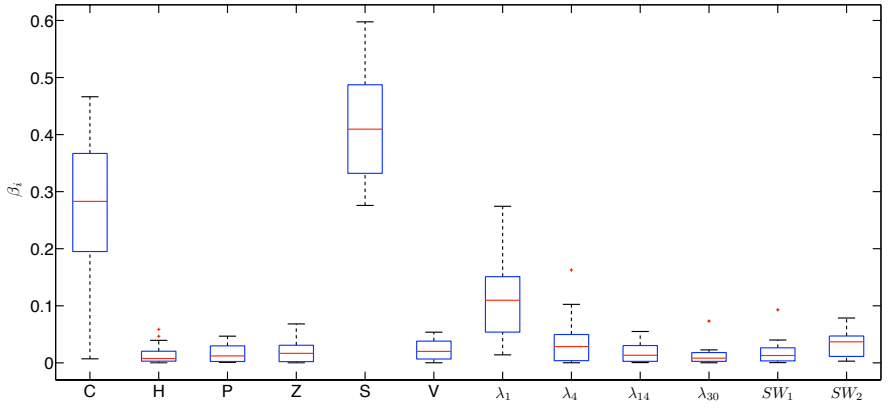

Figure 7.7: Combinatorial weights when all the feature spaces are employed.

As we can observe, the amino acid composition and the secondary structure are judged as more important, followed by the PseAA $\lambda = 1$. However, it is worth noting that by taking out the amino acid composition we have only a small loss in performance as we have seen in Table 7.10. These two observations suggest that the original amino acid (C) and the pseudo- ones (λ_i) carry redundant information. Furthermore, despite the individual accuracies of the SW features, they are not heavily weighted. This is because they depend solely

on the sequence similarity between proteins and their quality of discriminative information is strongly related to which end of the 0-35% sequence similarity the two proteins will belong. In reality, for the real "twilight-zone" of low-homology proteins (much less than 35% similarity) such features have little effect by definition.

In Figure 7.8 the confusion matrix for a single run is depicted. The values on the matrix are normalised according to $R_{ij} = \frac{P_j}{N_i}$ where N_i is the total number of proteins belonging in class i and P_j is the number of these N_i proteins that were predicted to belong to class j. For example when all of the proteins in class c were predicted correctly, then $R_{cc} = 1$ and $R_{cj} = 0 \; \forall \; j \neq c \in \{1, C\}$

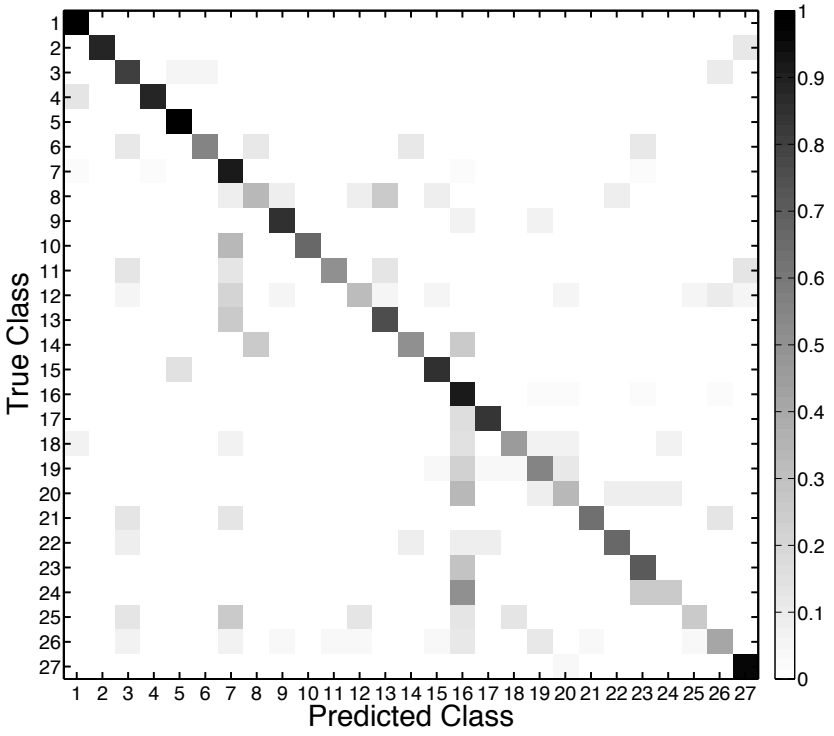

Figure 7.8: Confusion matrix with each element normalised to R_{ij}

First, it is worth noting that there are two areas where consistent misclassification occurs. The first one is when proteins of class 10 to 13 (conA-like barrel, SH3-like barrel, OB, beta-trefoil) are classified as class 7 (fold: immunoglobulin like) and the second one is when proteins of class 19-20 and 24 (Rossmann fold, P-loop, periplasmic binding protein-like) are classified as class 16 (fold: TIM-barrel). Noting that folds 7 and 16 are represented by the top two largest

numbers in the training set (30 and 29 proteins respectively) this seems to imply that these classes are over-represented in comparison with other folds (mean size of 10 proteins) and that features such as (pseudo- or not) amino acid composition and secondary structure offer little discriminative power on the distinction problem in these two areas.

Furthermore, besides the proteins in the fifth class (fold: 4-helical cytokines) that are all correctly classified as expected by previous observations by (Ding and Dubchak 2001), now the first class (fold: globin-like) is also achieving a 100% accuracy together with three more classes (7, 16, 27) (folds: immunoglobulin-like, TIM-barrel, small inhibitors) above the 90% level.

7.3 Remote Homology Detection

As a further generalisation of the proposed methodology to other challenging domains that have recently received a lot of attention we consider the *simulated* remote homology problem (RHD) as described in the works of (Liao and Noble 2003, Leslie et al. 2004, Saigo et al. 2004, Lingner and Meinicke 2004). RHD is the problem of detecting protein homology (proteins belonging into the same evolutionary family) in cases when there is a low sequence similarity between them. It is a formidable problem as by definition the sequence based similarity between homologs is less informative and there is no approach that works well in all cases (Ben-Hur and Brutlag 2003).

The SCOP 1.53 benchmark data-set[4] as described in (Liao and Noble 2003) is employed to simulate the RHD problem. It consists of 4,352 proteins belonging to one of 54 families and the positive training is performed on low-homologs while the positive testing on members of the same family. We consider four state-of-the-art string kernels, namely a *local alignment* (LA) kernel (Saigo et al. 2004), a *mismatch* (MM) kernel (Leslie et al. 2004), an *oligomer* kernel (Mono) (Lingner and Meinicke 2004) and a *pairwise* Smith-Waterman (SW) kernel (Liao and Noble 2003) that were previously individually employed in conjunction with discriminative SVM classifiers.

Following the MKL paradigm, the best performing case from each string kernel category is selected as a separate informational source to be combined with the proposed VBpMKL method. The lower bound is monitored at the 0.1

[4]Available from http://www.ccls.columbia.edu/compbio/svm-pairwise

% level with a maximum of 100 iterations and following the above past works we add a class-dependent regularisation parameter to the diagonal of the kernels to improve performance on this highly imbalanced problem. Adhering to the same performance measures as in the related works, we report the average AUC or ROC score (Area Under the Receiver Operating Characteristic Curve), for both the standard 100% and the 50% level, and also the median RFP (Rate of False Positives) score which is the fraction of negative test sequences (non-homologs) that score as high or better than the median-scoring positive test sequences (homologs). Results are averaged over 10 randomly initialised trials.

The results from the combination of the string kernels are depicted in Table 7.13 together with the best previously reported results within the SVM methodology. We achieve state-of-the-art performance via the combination of the kernels and match the overall best performing SVM method outperforming other string kernels.

Table 7.13: ROC, ROC50 and median RFP scores.

Method	Mean ROC	Mean ROC50	Mean mRFP
SVM (SW)	0.896	0.464	0.0837
SVM (LA)	**0.925**	**0.649**	**0.0541**
SVM (MM)	0.872	0.400	0.0837
SVM (Mono)	0.919	0.508	0.0664
VBpMKL	0.924	0.567	0.0661

In Figure 7.9 the number of families that achieve certain ROC scores is depicted in comparison with some of the best performing methods reported in the literature.

Furthermore, by employing the weighted combination we infer the contribution of each string kernel and as it can be seen from Figure 7.10 the Monomer (Mono) and the Local-alignment (LA) kernel are weighted most heavily as expected from Table 7.13 and previously reported results.

7.4 Protein Subcellular Localisation

The final application of the proposed pMKL methodology in this Chapter is on a very large multiple feature problem (69 attribute sets) which is attacked with the sparse mRVM models of Chapter 5. The problem addressed is the one of

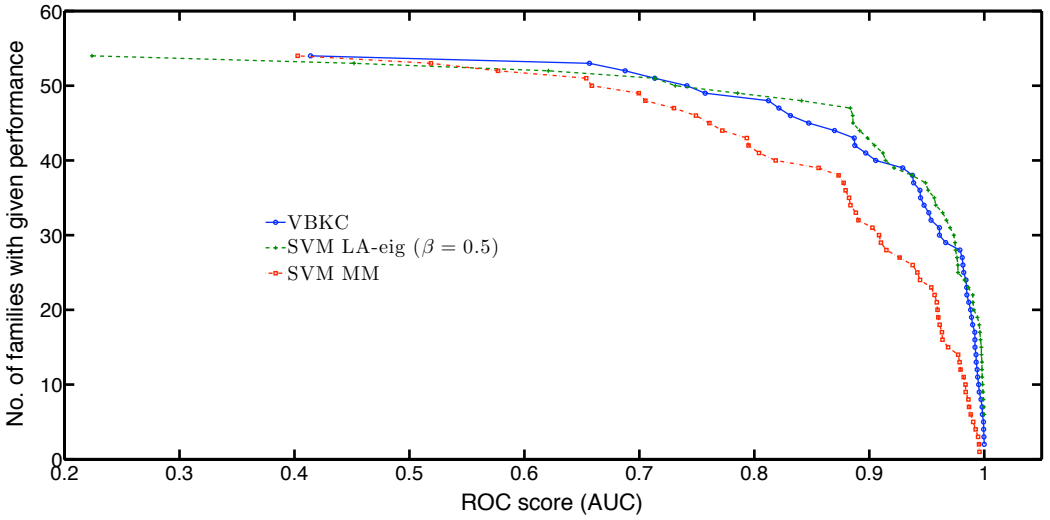

Figure 7.9: ROC score (AUC) distributions for the proposed string combination method and two state-of-the-art string kernels with SVMs. Every point in the graph describes the number of families (y-axis) that achieve a specific ROC score (x-axis) by a single method.

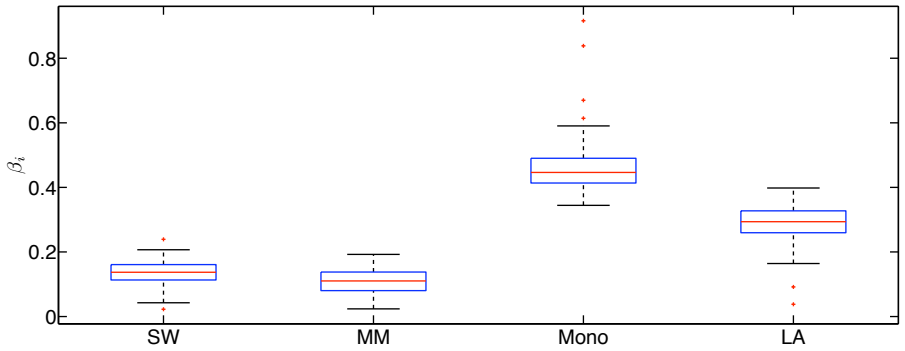

Figure 7.10: Kernel combination weights when all the string kernels are fused.

predicting subcellular protein localisation (the cell location of a protein) based on a set of disparate data sources, represented as a set of feature spaces and incorporated in the proposed method by a set of appropriate kernels.

Two problems are considered: predicting subcellular localisation for Gram positive (PSORT+) and Gram negative bacteria (PSORT-). Original state-of-the-art performance on this problem[5] was given by PSORTb (Gardy et al. 2005), a prediction tool utilising multiple SVMs and a Bayesian network which pro-

[5]Data and associated material in http://www.psort.org/

vides a prediction confidence measure for the method, compensating for the non-probabilistic formulation of standard SVMs. The confidence measure can be thresholded to perform class assignment or to indicate some samples as unclassifiable.

Following the experimental setup of Ong and Zien (Zien and Ong 2007) 69 feature spaces are proposed of which 64 are motif kernels computed at different sections of the protein sequence and the rest are pairwise string kernels based on BLAST E-values and phylogenetic profile kernels (Kuang et al. 2004). Their MKL method with SVMs (Zien and Ong 2007) claimed a new state-of-the art performance, on a reduced subset of the PSORTb dataset, with reported performances of 93.8 ± 1.3 on PSORT+ and 96.1 ± 0.6 on PSORT- using an average F1 score. However due to the non-probabilistic nature of SVMs the MKL method was augmented with a post-processing criteria to create class probabilities in order to leave out the 13% lowest confidence predictions for PSORT+ and 15% for PSORT-, thus approximating the unclassifiable assignment option of PSORTb.

Further comparison is reported with another multiclass multi-kernel learning algorithm proposed in (Ye et al. 2008) for regularised kernel discriminant analysis (RKDA). For this algorithm, we employ the semi-infinite linear programming (SILP) approach with a fixed regularisation parameter 5×10^{-4} as suggested there.

In Table 7.14 the average test-error percentage over 10 randomly initialised 80% training and 20% test splits on the PSORT+ subset for both mRVM methods is presented. Similarly Table 7.15 presents the results for the PSORT- case. The resulting average sample sparsity of the two models is very large, in cases requiring less than 20% of the total number of samples. It is worth pointing out that there are no analogous sparse relevant vectors in the RKDA kernel learning approach and the method relies on all the training samples.

Method	Test Error%	Relevance Vectors
mRVM_1	12.9 ± 3.7	27.9 ± 4.5
mRVM_2	10.4 ± 3.9	60.8 ± 4.3
RKDA-MKL	8.39 ± 1.46	$--$

Table 7.14: Error and sparsity on PSORT+

The further sparsity of the kernel combinations for PSORT+ can be seen from Figure 7.11, where the average kernel combination parameters $\boldsymbol{\beta}$ over the

Method	Test Error%	Relevance Vectors
mRVM_1	13.8 ± 4.5	109.2 ± 19.5
mRVM_2	11.9 ± 1.2	102.7 ± 7.4
RKDA-MKL	10.52 ± 2.56	$--$

Table 7.15: Error and sparsity on PSORT-

10 runs are shown. We are in general agreement with the selected kernels from previous studies as E-value kernels (3,4) and phylogeny kernels (68,69) are judged significant in these combinations.

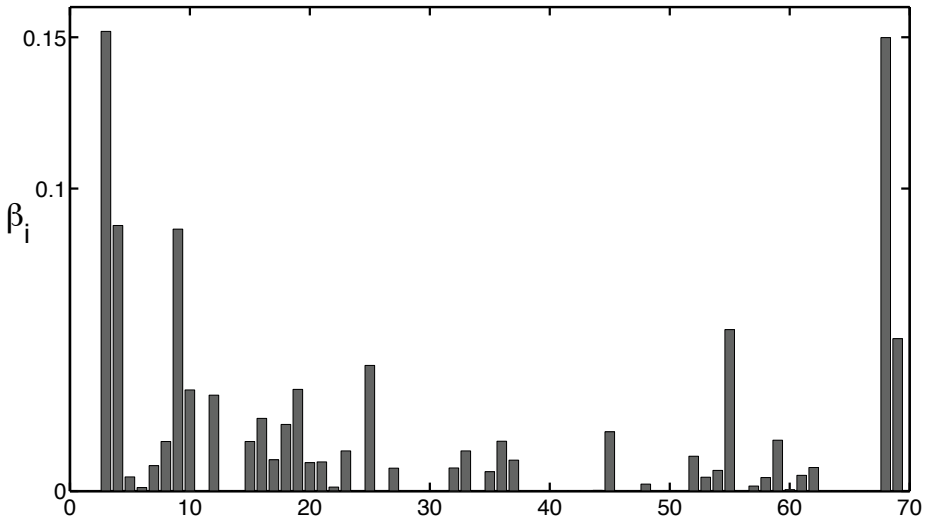

Figure 7.11: Average kernel usage: PSORT+

Similarly for PSORT-, Figure 7.12 indicates that the E-value and phylogeny kernels are significant contributors. Hence now both sample-wise and kernel-wise sparse solutions exist for the problem under consideration.

7.5 Discussion

In this Chapter the previously introduced probabilistic multiple kernel learning (pMKL) methodology has been applied to important large scale applications which benefit from both a computational and an informative perspective under such probabilistic fusion schemes. It was demonstrated that the proposed pMKL

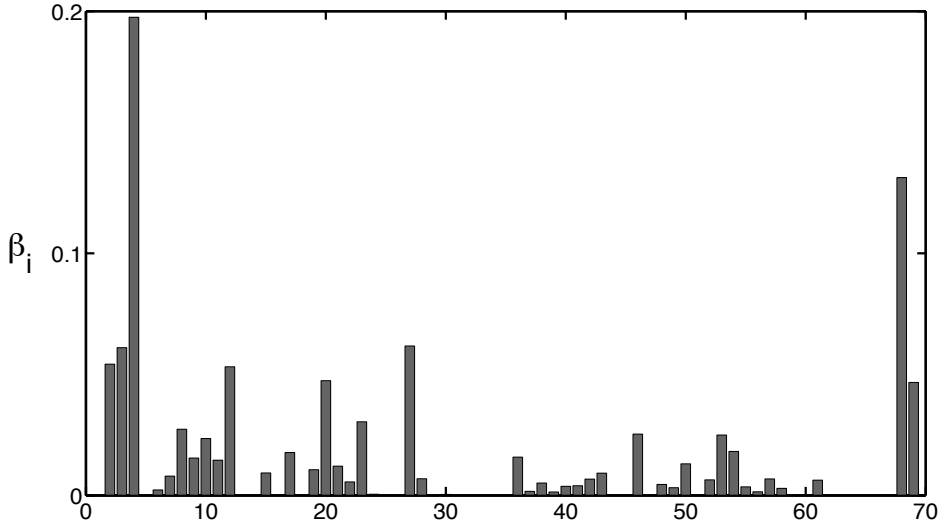

Figure 7.12: Average kernel usage: PSORT-

approaches produce state of the art results across multi-feature problem domains while retaining their efficiency and offering an appealing alternative to classifier combination and other ensemble learning methods.

Specifically, both the variational approximation and the Gibbs sampling approach performed competitively to previously introduced classifier combination rules on the problem of handwriting numeral recognition while inferring the significance of contributing sources. Furthermore, the variational approximation was shown to provide the state of the art on challenging bioinformatics problems such as protein fold recognition and remote homology detection where previous approaches required thousands of classifiers. Finally, the sparse methodology of mRVMs was shown to significantly reduce the required sample size while retaining few multiclass relevance vectors to be used for prediction on the problem of subcellular protein localisation.

In the following final main Chapter, an assessment of the underlying theoretical principles of MKL and a justification for the use of parameterised (weighted) or fixed kernel combination schemes is offered together with an attempt to formally define the conditions when the aforementioned integration schemes are expected to be beneficial on zero-one loss or predictive likelihood criteria.

Chapter 8

Diversity in Multiple Kernel Learning

Multiple Kernel Learning (MKL) methods aim at learning an optimal (in a predefined model-specific sense such as predictive likelihood or zero-one loss) combination of individual base kernels. Therefore such approaches follow the basic assumption that kernel combination parameter inference is crucial and beneficial over a fixed *a priori* combination. However, in a number of reported cases (Lewis et al. 2006b, Girolami and Zhong 2007, Damoulas and Girolami 2008) it has been observed that there is none or little such benefit. The opposite phenomenon of a significant performance improvement from an *a posteriori* combination rule has also been observed on other problems (Zien and Ong 2007, Damoulas et al. 2008) indicating a dataset dependent MKL behaviour. In this Chapter we attempt to address these issues and provide a formal reasoning behind this phenomenon. Borrowing ideas from classifier construction analysis (Hand 2006) and decomposition of the loss (Krogh and Vedelsby 1995, Ueda and Nakano 1996) we examine the conditions under which parameterised MKL methods are expected to improve over a priori fixed combinations.

Furthermore, an information theoretic perspective on MKL approaches is proposed via the Fisher Information and *Optimal Experimental Design* by Sir R. A Fisher (1935). This novel direction is considered as a significant future MKL research work that can lead to qualitative conclusions and further novel algorithms.

8.1 The Flat Maximum Effect

Hand (1997) first described the "Flat Maximum Effect" in the context of classifier performance as the phenomenon when "*often quite large deviations from the optimal set of weights will yield predictive performance not substantially worse than the optimal weights*". In Hand (2006) a set of regression coefficients are shown to be highly correlated between any other random set of regression coefficients *if* the predictor variables are correlated. This is generalised here for the kernel combination parameters and provides a starting argument for the need of *diversity* between individual information sources embedded as kernels.

The original case of simple linear combination of predictor variables, as described in Hand (2006) is revisited here, as it will form the basis for this section's analysis on multiple kernel learning.

8.1.1 Linear regression model

Let the predictor variables be $(x_1, \ldots, x_d)^T = \mathbf{x}$ and assume they are standardised to zero mean $\mathbb{E}(x_i) = 0$ and variance one $V(x_i) = 1$ for $i = 1, \ldots, d$. Considering non-negative weights $w_i, u_i \geq 0$ and requiring $\sum_{i=1}^{d} w_i = 1$ and $\sum_{i=1}^{d} u_i = 1$ then the two weighted sums can be written as

$$w = \sum_{i=1}^{d} w_i x_i \quad \text{and} \quad u = \sum_{j=1}^{d} u_j x_j \tag{8.1}$$

with the Pearson product-moment correlation coefficient for variables u, w given by

$$\rho(u, w) = \frac{\mathbb{E}(uw) - \mathbb{E}(u)\mathbb{E}(w)}{\sqrt{\mathbb{E}(u^2) - \mathbb{E}^2(u)}\sqrt{\mathbb{E}(w^2) - \mathbb{E}^2(w)}} \tag{8.2}$$

where $\mathbb{E}$ denotes expectation as usual. Substituting from Equation 8.1 and analysing terms leads to the expression:

$$\rho(u,w) =$$

$$= \frac{\mathbb{E}\left(\sum_{i=1}^{d} u_i x_i \sum_{j=1}^{d} w_j x_j\right) - \mathbb{E}\left(\sum_{i=1}^{d} u_i x_i\right)\mathbb{E}\left(\sum_{j=1}^{d} w_j x_j\right)}{\sqrt{\mathbb{E}\left(\left(\sum_{i=1}^{d} u_i x_i\right)^2\right) - \mathbb{E}^2\left(\sum_{i=1}^{d} u_i x_i\right)}\sqrt{\mathbb{E}\left(\left(\sum_{j=1}^{d} w_j x_j\right)^2\right) - \mathbb{E}^2\left(\sum_{j=1}^{d} w_j x_j\right)}}$$

$$= \frac{\sum_{i,j=1}^{d} u_i w_j \left[\mathbb{E}\left(x_i x_j\right) - \mathbb{E}\left(x_i\right)\mathbb{E}\left(x_j\right)\right]}{\sqrt{\sum_{i,j=1}^{d} u_i u_j \left[\mathbb{E}\left(x_i x_j\right) - \mathbb{E}\left(x_i\right)\mathbb{E}\left(x_j\right)\right]}\sqrt{\sum_{i,j=1}^{d} w_i w_j \left[\mathbb{E}\left(x_i x_j\right) - \mathbb{E}\left(x_i\right)\mathbb{E}\left(x_j\right)\right]}}$$

which by substituting for the covariance $\mathcal{V}_{ij} = \mathbb{E}\left(x_i x_j\right) - \mathbb{E}\left(x_i\right)\mathbb{E}\left(x_j\right)$ can be further simplified to

$$\rho(u,w) = \frac{\sum_{i,j=1}^{d} u_i w_j \mathcal{V}_{ij}}{\sqrt{\sum_{i,j=1}^{d} u_i u_j \mathcal{V}_{ij}}\sqrt{\sum_{i,j=1}^{d} w_i w_j \mathcal{V}_{ij}}} \tag{8.3}$$

Noting that $\rho(x_i, x_j) = \frac{\mathcal{V}_{ij}}{\sqrt{\mathcal{V}_{ii}}\sqrt{\mathcal{V}_{jj}}}$ and $\mathcal{V}_{ii} = 1 \;\forall\; i \in \{1, \ldots, d\}$ we have:

$$\rho(u,w) = \frac{\sum_{i,j=1}^{d} u_i w_j \rho(x_i, x_j)}{\sqrt{\sum_{i,j,k,\lambda=1}^{d} u_i u_j w_k w_\lambda \mathcal{V}_{ij} \mathcal{V}_{k\lambda}}} \tag{8.4}$$

and as the covariates are standardised, $\mathcal{V}_{ij} \leq 1 \;\forall\; i, j \in 1, \ldots, d$, and the regression coefficients sum to one, it leads to the final bound of the relationship:

$$\rho(u,w) \geq \sum_{i,j=1}^{d} u_i w_j \rho(x_i, x_j) \tag{8.5}$$

Equation 8.5 states that the correlation between *any* two sets of regression coefficients is bounded below by a function of the correlation between the co-

variates. Hence, when covariates are highly correlated then it does not really matter which regression coefficients are employed or in other words that a "flat maximum" region exists for all possible parameter values.

8.1.2 Extension to Multiple Kernel Learning

In a direct analogy to the linear regression case, a flat maximum effect can be observed for the case of multiple kernel learning and the standard convex linear combination rule. Consider S sources of information represented by S standardised kernels describing similarities between N training objects. For a test object $\mathbf{x}_i = \{x_1, \ldots, x_D\} \in \mathbb{R}^D$ the responses for two different linear combinations $\mathbf{b}$ and $\boldsymbol{\beta}$ are given by:

$$y_{\mathbf{b}} = \sum_{j=1}^{N} w_j \sum_{s=1}^{S} b_s k_s(\mathbf{x}_i, \mathbf{x}_j) \quad \text{and} \quad y_{\boldsymbol{\beta}} = \sum_{j=1}^{N} w_j \sum_{\tau=1}^{S} \beta_\tau k_\tau(\mathbf{x}_i, \mathbf{x}_j) \tag{8.6}$$

which by setting $\theta_s = \sum_{j=1}^{N} w_j k_s(\mathbf{x}_i, \mathbf{x}_j)$ results in

$$y_{\mathbf{b}} = \sum_{s=1}^{S} b_s \theta_s \quad \text{and} \quad y_{\boldsymbol{\beta}} = \sum_{\tau=1}^{S} \beta_\tau \theta_\tau \tag{8.7}$$

in direct correspondence with Equation 8.1 from the linear response case. Following the same analysis, the correlation between these two responses $y_{\mathbf{b}}$ and $y_{\boldsymbol{\beta}}$ is directly related to the correlation between the weighted base kernels. The response correlation is:

$$\rho(y_{\mathbf{b}}, y_{\boldsymbol{\beta}}) = \frac{\mathbb{E}(y_{\mathbf{b}} y_{\boldsymbol{\beta}}) - \mathbb{E}(y_{\mathbf{b}})\mathbb{E}(y_{\boldsymbol{\beta}})}{\sqrt{\mathbb{E}(y_{\mathbf{b}}^2) - \mathbb{E}^2(y_{\mathbf{b}})}\sqrt{\mathbb{E}(y_{\boldsymbol{\beta}}^2) - \mathbb{E}^2(y_{\boldsymbol{\beta}})}} \tag{8.8}$$

which by defining the correlation between θ_s and θ_τ as $\rho(\theta_s, \theta_\tau)$ and the covariance $\Delta_{s\tau} = \mathbb{E}(\theta_s \theta_\tau) - \mathbb{E}(\theta_s)\mathbb{E}(\theta_\tau)$ leads to

$$\rho(y_{\mathbf{b}}, y_{\boldsymbol{\beta}}) = \frac{\sum_{s,\tau=1}^{S} b_s \beta_\tau \rho(\theta_s, \theta_\tau)}{\sqrt{\sum_{s,\tau,\lambda,k=1}^{S} b_s b_\tau \beta_\lambda \beta_k \Delta_{s\tau} \Delta_{\lambda k}}} \tag{8.9}$$

and since $\Delta_{s\tau} \leq 1 \; \forall \; s, \tau \in \mathbb{R}$ the relationship can again be bounded as

$$\rho(y_{\mathbf{b}}, y_{\boldsymbol{\beta}}) \geq \sum_{s,\tau=1}^{S} b_s \beta_\tau \rho\left(\theta_s, \theta_\tau\right) \tag{8.10}$$

Hence, when the sources of information are highly correlated then any two responses y created by different sets of kernel combination weights $\mathbf{b}$ and $\boldsymbol{\beta}$ will be highly correlated, exhibiting the so called *Flat maximum effect* on MKL problems. This novel characteristic for kernel combinations is in agreement with both the intuition regarding correlated information sources that do not offer any additional gain and with similar arguments of required *diversity* for multiple classifier systems (Kuncheva and Whitaker 2003). A further insight into why *diverse* base kernels are expected to offer improvement in MKL is given by the decomposition of the loss in a regression setting which is directly connected to the proposed pMKL classifiers due to the implicit regression on the auxiliary variables. We consider such decompositions in the following two Sections that highlight exactly that need for diversity.

8.2 The Ambiguity Decomposition

The auxiliary variable regression nature of the proposed probabilistic MKL models is amenable to the ensemble regression analysis introduced by Brown and Wyatt (2003) which is an extension of the well-known bias-variance decomposition by Krogh and Vedelsby (1995). In this section the so called *ambiguity decomposition* analysis is adopted and applied on the MKL scenario.

Consider the (ensemble) response of the model for sample i under the standard convex linear combination of base kernels

$$\mathbf{y}_e = \sum_{j=1}^{N} w_j \sum_{s=1}^{S} \beta_s k_s(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s=1}^{S} \beta_s \mathbf{y}_s \tag{8.11}$$

with the individual base kernel response defined as $\mathbf{y}_s = \sum_{j=1}^{N} w_j k_s(\mathbf{x}_i, \mathbf{x}_j)$ and the typical linear constraint $\sum_{s=1}^{S} \beta_s = 1$. Defining $\hat{\mathbf{y}}$ as the target regression variable and analysing the expression $\sum_{s=1}^{S} \beta_s \left(\mathbf{y}_s - \hat{\mathbf{y}}\right)^T \left(\mathbf{y}_s - \hat{\mathbf{y}}\right)$

$$
\begin{aligned}
&= \sum_{s=1}^{S} \beta_s \left(\mathbf{y}_s - \mathbf{y}_e + \mathbf{y}_e - \hat{\mathbf{y}}\right)^T \left(\mathbf{y}_s - \mathbf{y}_e + \mathbf{y}_e - \hat{\mathbf{y}}\right) \\
&= \sum_{s=1}^{S} \beta_s \left[\left(\mathbf{y}_s - \mathbf{y}_e\right)^T \left(\mathbf{y}_s - \mathbf{y}_e\right) + \left(\mathbf{y}_e - \hat{\mathbf{y}}\right)^T \left(\mathbf{y}_e - \hat{\mathbf{y}}\right) + 2\left(\mathbf{y}_s - \mathbf{y}_e\right)^T \left(\mathbf{y}_e - \hat{\mathbf{y}}\right)\right] \\
&= \sum_{s=1}^{S} \beta_s \left(\mathbf{y}_s - \mathbf{y}_e\right)^T \left(\mathbf{y}_s - \mathbf{y}_e\right) + \left(\mathbf{y}_e - \hat{\mathbf{y}}\right)^T \left(\mathbf{y}_e - \hat{\mathbf{y}}\right) \qquad (8.12)
\end{aligned}
$$

Rearranging we have

$$
\underbrace{\left(\mathbf{y}_e - \hat{\mathbf{y}}\right)^T \left(\mathbf{y}_e - \hat{\mathbf{y}}\right)}_{\text{Composite Error}} = \underbrace{\sum_{s=1}^{S} \beta_s \left(\mathbf{y}_s - \hat{\mathbf{y}}\right)^T \left(\mathbf{y}_s - \hat{\mathbf{y}}\right)}_{\text{Weighted Ind. Error}} - \underbrace{\sum_{s=1}^{S} \beta_s \left(\mathbf{y}_s - \mathbf{y}_e\right)^T \left(\mathbf{y}_s - \mathbf{y}_e\right)}_{\text{Ambiguity}} \qquad (8.13)
$$

where the first term of the right hand side is the weighted average error of individual base kernel responses and the second term is the *Ambiguity* term which describes the variability or diversity of the individual base kernels with respect to the ensemble response.

Hence, in order to minimise the composite error, the individual base kernel errors must be minimised and their response diversity maximised. The preference therefore is for accurate but different and uncorrelated ensemble members that potentially capture different aspects of the underlying phenomenon while retaining a good overall performance. In the next section an alternative decomposition of the loss reveals how the "Ambiguity" term can be formally captured within the covariance of the base kernel responses.

8.3 Bias-Variance-Covariance Decomposition

This analysis follows the Bias-Variance-Covariance loss decomposition of Ueda and Nakano (1996). Retaining the same notation as previously, the bias-variance decomposition for a single regressor (individual base kernel) with output $\mathbf{y}_s$ can

be expressed as:

$$\mathbb{E}\left\{(\mathbf{y}_s - \hat{\mathbf{y}})^{\mathsf{T}} (\mathbf{y}_s - \hat{\mathbf{y}})\right\} =$$

$$\underbrace{(\mathbb{E}\{\mathbf{y}_s\} - \hat{\mathbf{y}})^{\mathsf{T}} (\mathbb{E}\{\mathbf{y}_s\} - \hat{\mathbf{y}})}_{\text{Bias}^{\mathsf{T}}\text{Bias}} + \underbrace{\mathbb{E}\left\{(\mathbf{y}_s - \mathbb{E}\{\mathbf{y}_s\})^{\mathsf{T}} (\mathbf{y}_s - \mathbb{E}\{\mathbf{y}_s\})\right\}}_{\text{Variance}} \tag{8.14}$$

Extending the decomposition now to the the ensemble (composite kernel) output $\mathbf{y}_e$:

$$\mathbb{E}\left\{(\mathbf{y}_e - \hat{\mathbf{y}})^{\mathsf{T}} ((\mathbf{y}_e - \hat{\mathbf{y}})\right\} =$$

$$\underbrace{(\mathbb{E}\{\mathbf{y}_e\} - \hat{\mathbf{y}})^{\mathsf{T}} (\mathbb{E}\{\mathbf{y}_e\} - \hat{\mathbf{y}})}_{\text{Term 1}} + \underbrace{\mathbb{E}\left\{(\mathbf{y}_e - \mathbb{E}\{\mathbf{y}_e\})^{\mathsf{T}} (\mathbf{y}_e - \mathbb{E}\{\mathbf{y}_e\})\right\}}_{\text{Term 2}} \tag{8.15}$$

and recalling that $\mathbf{y}_e = \sum_{s=1}^{S} \beta_s \mathbf{y}_s$ and $\sum_{s=1}^{S} \beta_s = 1$, we can further examine the resulting terms in analogy with the original bias-variance decomposition.

Analysing Term 1 leads to:

$$(\mathbb{E}\{\mathbf{y}_e\} - \hat{\mathbf{y}})^{\mathsf{T}} (\mathbb{E}\{\mathbf{y}_e\} - \hat{\mathbf{y}}) = \sum_{s=1}^{S} \sum_{\sigma=1}^{S} \beta_s^{\mathsf{T}} \beta_\sigma \underbrace{(\mathbb{E}\{\mathbf{y}_s\} - \hat{\mathbf{y}})^{\mathsf{T}} (\mathbb{E}\{\mathbf{y}_\sigma\} - \hat{\mathbf{y}})}_{\text{Bias}^{\mathsf{T}}\text{Bias}} \tag{8.16}$$

Analyzing Term 2 leads to:

$$\mathbb{E}\left\{\left(\sum_{s=1}^{S}\beta_s\mathbf{y}_s-\mathbb{E}\left\{\sum_{s=1}^{S}\beta_s\mathbf{y}_s\right\}\right)^{\mathsf{T}}\left(\sum_{s=1}^{S}\beta_s\mathbf{y}_s-\mathbb{E}\left\{\sum_{s=1}^{S}\beta_s\mathbf{y}_s\right\}\right)\right\} \tag{8.17}$$

$$=\sum_{s=1}^{S}\beta_s^{\mathsf{T}}\beta_s\underbrace{\mathbb{E}\left\{\left(\mathbf{y}_s-\mathbb{E}\left\{\mathbf{y}_s\right\}\right)^{\mathsf{T}}\left(\mathbf{y}_s-\mathbb{E}\left\{\mathbf{y}_s\right\}\right)\right\}}_{\text{Variance}} \tag{8.18}$$

$$+\sum_{s=1}^{S}\sum_{\sigma\neq s}^{S}\beta_s\beta_\sigma\underbrace{\mathbb{E}\left\{\left(\mathbf{y}_s-\mathbb{E}\left\{\mathbf{y}_s\right\}\right)^{\mathsf{T}}\left(\mathbf{y}_\sigma-\mathbb{E}\left\{\mathbf{y}_\sigma\right\}\right)\right\}}_{\text{Covariance}} \tag{8.19}$$

Hence, combining both terms back to our original expression we have the final expression for the decomposition of the loss from an ensemble of base kernels:

$$\boxed{\begin{aligned}\mathbb{E}\left\{\left(\mathbf{y}_e-\hat{\mathbf{y}}\right)^{\mathsf{T}}\left(\left(\mathbf{y}_e-\hat{\mathbf{y}}\right)\right\}&=\sum_{s=1}^{S}\sum_{\sigma=1}^{S}\beta_s^{\mathsf{T}}\beta_\sigma\text{Bias}^{\mathsf{T}}\text{Bias}\\&+\sum_{s=1}^{S}\beta_s^{\mathsf{T}}\beta_s\text{Variance}+\sum_{s=1}^{S}\sum_{\sigma\neq s}^{S}\beta_s\beta_\sigma\text{Covariance}\end{aligned}} \tag{8.20}$$

The above decomposition of the MKL regression loss follows other ensemble learning methods (Ueda and Nakano 1996, Kittler et al. 1998, Tax et al. 2000) in including an additional *Covariance* term between the ensemble members (base kernels in our case) which offers an alternative description for the effect of diversity and its contribution on the overall loss.

8.4 Diversity and Information

So far we have seen how the "diversity" (expressed through the covariance and ambiguity terms) in base kernels plays an important role in reducing the loss and also that when such diversity is absent, we can expect a parameterised kernel combination rule not to improve upon fixed combinations due to the *flat maximum effect*. A further case when parameterised combinations should be preferred is when an information source is corrupt. To illustrate that scenario, classification on the Neal dataset is investigated with two information sources. The first is the original data and the second source is a corrupted version of it

which is created by adding random noise with varying standard deviation.

In order to observe how the diversity of the sources increases when the noise term becomes more dominant in the corrupted source we consider the Frobenius inner product between the two base sources $\mathbf{K}_i, \mathbf{K}_j \in \mathbb{R}^{N \times N}$ which is an explicit measure of kernel (matrix) similarity defined as:

$$\mathcal{F} = \mathrm{Tr}\left[\mathbf{K}_i \mathbf{K}'_j\right] = \mathrm{Tr}\left[\mathbf{K}_j \mathbf{K}'_i\right] \tag{8.21}$$

and we can normalise it with respect to the self-similarity of a matrix as

$$\hat{\mathcal{F}} = \frac{\mathrm{Tr}\left[\mathbf{K}_i \mathbf{K}'_j\right]}{\sqrt{C_1 C_2}} \tag{8.22}$$

where $C_1 = \mathrm{Tr}\left[\mathbf{K}_i \mathbf{K}'_i\right]$ and $C_2 = \mathrm{Tr}\left[\mathbf{K}_j \mathbf{K}'_j\right]$

In Figure 8.1 the results from the convex linear and the fixed summation rule are shown while varying the noise and hence decreasing the similarity between the base kernels.

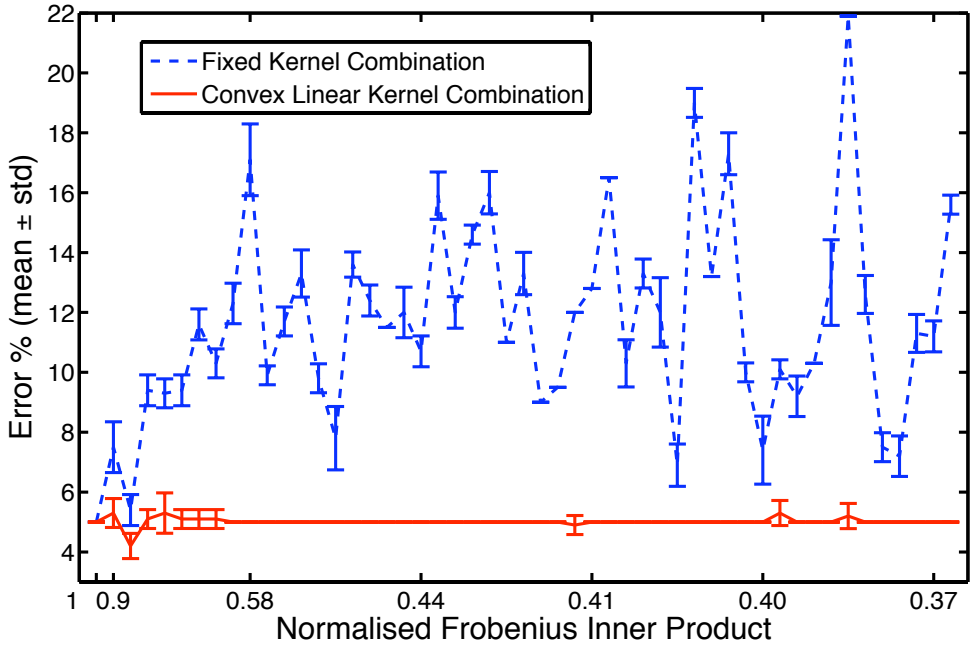

Figure 8.1: Varying the corruption level on a source while measuring the Frobenius inner product. Results are averaged over 10 randomly bootstrapped runs for every noise level.

As it can be seen, for low corruption levels and hence a normalised Frobenius

inner product close to one, both methods achieve same levels of performance. On the contrary, when the noise corruption level is increased the parameterised kernel combination rule clearly outperforms the fixed combination as it can down-weight the corrupted source and exploit the informative base kernel instead.

However, diversity or correlation do not describe the *information quality* of sources but only their dissimilarity and as we have seen here diversity may be present without additional information content. Towards that goal of assessing both the diversity and the information gain we propose Fisher information criteria in the following and last section.

8.5 Fisher Information for MKL

An alternative, information-theoretic, view for MKL can be offered via the Fisher Information (FI) and the *optimal experimental design* principles by Fisher (1935). In accordance with the latter we seek to maximise the information offered by the model parameters with respect to the evidence observed. This leads to the examination of the log-likelihood curvature which expresses the variance of the log-likelihood for small parameter permutations and hence acts as a measure of the information density in specific regions of the parameter space.

The Fisher Information for parameters $\boldsymbol{\theta}$, evidence y and log-likelihood $\mathcal{L} = \log p(y|\boldsymbol{\theta})$ is defined as:

$$\mathcal{F}(\boldsymbol{\theta}) = -\mathbb{E}\left\{\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}\right\} \tag{8.23}$$

In this section, a simple linear regression case for MKL is considered:

$$\mathbf{y} = \mathbf{K}_{\boldsymbol{\beta}}\mathbf{w} + \boldsymbol{\epsilon} \quad \text{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{8.24}$$

where the likelihood is given by:

$$L = C \exp\left\{-\frac{1}{2}\left(\mathbf{y} - \mathbf{K}_{\boldsymbol{\beta}}\mathbf{w}\right)^\top \left(\mathbf{y} - \mathbf{K}_{\boldsymbol{\beta}}\mathbf{w}\right)\right\} \tag{8.25}$$

and the log-likelihood, disregarding constant terms or terms independent of the parameters $\boldsymbol{\beta}$ and $\mathbf{w}$, is expressed as:

$$\mathcal{L} = \left(\mathbf{y}^\top \sum_{s=1}^{S} \beta_s \mathbf{K}_s \mathbf{w} - \frac{1}{2}\mathbf{w}^\top \sum_{s=1}^{S}\sum_{k=1}^{S} \beta_s \beta_k \mathbf{K}_s \mathbf{K}_k \mathbf{w}\right) \tag{8.26}$$

where $\mathbf{K}_s$ is the s^{th} base kernel $\in \mathbb{R}^{N\times N}$ and $\mathbf{K}_{\boldsymbol{\beta}}$ is the composite kernel $\in \mathbb{R}^{N\times N}$.

8.5.1 Fisher Information of $\boldsymbol{\beta}$

First we are interested in the Fisher information of the kernel combination parameters $\boldsymbol{\beta}$ which is given by:

$$\mathcal{F}(\boldsymbol{\beta}) = -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathsf{T}}}\right] \tag{8.27}$$

Hence we need the first derivative:

$$\frac{\partial \mathcal{L}}{\partial \beta_s} = \mathbf{y}^{\mathsf{T}} \mathbf{K}_s \mathbf{w} - \mathbf{w}^{\mathsf{T}} \sum_k \beta_k \mathbf{K}_s \mathbf{K}_k \mathbf{w} \tag{8.28}$$

which leads to the following second derivative:

$$\frac{\partial^2 \mathcal{L}}{\partial \beta_s \beta_k} = -\mathbf{w}^{\mathsf{T}} \mathbf{K}_s \mathbf{K}_k \mathbf{w} \tag{8.29}$$

and finally the Fisher Information matrix has elements given by:

$$\mathcal{F}_{sk}(\boldsymbol{\beta}) = \mathbf{w}^{\mathsf{T}} \mathbf{K}_s \mathbf{K}_k \mathbf{w} \tag{8.30}$$

8.5.2 Fisher Information of the regression coefficients w

Similarly for the parameters $\mathbf{w}$, the Fisher Information is given by:

$$\mathcal{F}(\mathbf{w}) = -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w} \partial \mathbf{w}^{\mathsf{T}}}\right] \tag{8.31}$$

Hence we need the first derivative:

$$\frac{\partial \mathcal{L}}{\partial w_i} = \mathbf{y}^{\mathsf{T}} \mathbf{k}_i - \sum_j w_j \mathbf{k}_i^{\mathsf{T}} \mathbf{k}_j \tag{8.32}$$

where $\mathbf{k}_i$ is the i^{th} column vector of the composite kernel $\mathbf{K}_{\boldsymbol{\beta}}$. This leads to the following second derivative:

$$\frac{\partial^2 \mathcal{L}}{\partial w_i w_j} = -\mathbf{k}_i^{\mathsf{T}} \mathbf{k}_j \tag{8.33}$$

and finally the Fisher Information matrix has elements given by:

$$\mathcal{F}_{ij}(\mathbf{w}) = \mathbf{k}_i^{\mathsf{T}} \mathbf{k}_j \tag{8.34}$$

where $\mathbf{k}_i = \sum_{s=1}^{S} \beta_s \mathbf{k}_i^s$. The Fisher Information matrix can therefore be expressed as:

$$\mathcal{F}(\mathbf{w}) = \mathbf{K}_{\boldsymbol{\beta}}^{\mathsf{T}} \mathbf{K}_{\boldsymbol{\beta}} \tag{8.35}$$

8.5.3 Maximisation of the Fisher Information

Having derived the Fisher Information matrices for the regression coefficients and the kernel combination parameters we seek to find the optimal parameters $\hat{\boldsymbol{\beta}}$ that maximise the FI of the regression coefficients. Following the *A-optimality* criterion by Fisher (1935):

$$\min \operatorname{Tr}\left[\mathcal{F}^{-1}\right] \tag{8.36}$$

which, in the case of the regression coefficients is expressed as:

$$\min \operatorname{Tr}\left[\mathbf{K}_{\boldsymbol{\beta}} \mathbf{K}_{\boldsymbol{\beta}}\right]^{-1} \tag{8.37}$$

each base kernel is eigen-decomposed as $\mathbf{K}_i = \mathbf{U}_i \boldsymbol{\Lambda} \mathbf{U}_i^{\mathsf{T}}$ and hence we have:

$$\mathbf{K}_{\boldsymbol{\beta}} \mathbf{K}_{\boldsymbol{\beta}} = \sum_{i,j=1}^{S} \mathbf{U}_i \boldsymbol{\Lambda}_{\beta_i} \mathbf{U}_i^{\mathsf{T}} \mathbf{U}_j \boldsymbol{\Lambda}_{\beta_j} \mathbf{U}_j^{\mathsf{T}} \tag{8.38}$$

where $\boldsymbol{\Lambda}_{\beta_i}$ is a diagonal eigenvalue matrix with elements $\lambda_n^i \beta_i$. Now the inverse of the above matrix product, due to the unitary nature of $\mathbf{U}_i$, is simply:

$$\left(\mathbf{K}_{\boldsymbol{\beta}} \mathbf{K}_{\boldsymbol{\beta}}\right)^{-1} = \sum_{i,j=1}^{S} \mathbf{U}_i \boldsymbol{\Lambda}_{\beta_i}^{-1} \mathbf{U}_i^{\mathsf{T}} \mathbf{U}_j \boldsymbol{\Lambda}_{\beta_j}^{-1} \mathbf{U}_j^{\mathsf{T}} \tag{8.39}$$

and finally the trace, using the property that $\operatorname{Tr}[\mathbf{ABC}] = \operatorname{Tr}[\mathbf{ACB}]$, where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ matrices, and that for unitary matrices $\mathbf{U}_i^{\mathsf{T}} \mathbf{U}_i = \mathbf{I}$, is given by:

$$\operatorname{Tr}\left[\mathbf{K}_{\boldsymbol{\beta}} \mathbf{K}_{\boldsymbol{\beta}}\right]^{-1} = \sum_{i,j=1}^{S} \operatorname{Tr}\left[\boldsymbol{\Lambda}_{\beta_i}^{-1} \boldsymbol{\Lambda}_{\beta_j}^{-1}\right] = \sum_{i,j,n}^{S,S,N} \frac{1}{\lambda_n^i \lambda_n^j \beta_i \beta_j} \tag{8.40}$$

Hence we want to maximise A:

$$A = \sum_{i,j,n}^{S,S,N} \lambda_n^i \lambda_n^j \beta_i \beta_j \tag{8.41}$$

where differentiating with respect to $\boldsymbol{\beta}$ and setting to zero in order to find the maximum leads to the following homogeneous linear system:

$$\boldsymbol{\Delta}\boldsymbol{\beta} = \mathbf{0} \tag{8.42}$$

where $\Delta_{ij} = \sum_n \lambda_n^i \lambda_n^j$

The solution of this system of equations is:

$$\begin{cases} \beta_i = 0 \ \ \forall \ \ i \in \{1, \cdots, S\} & \text{if } \ |\boldsymbol{\Delta}| \neq 0 \\ \text{Infinite solutions} & \text{if } \ |\boldsymbol{\Delta}| = 0 \end{cases} \tag{8.43}$$

and considering the constraints $\sum_{s=1}^{S} \beta_s = 1$ and $\beta_i \geq 0 \ \ \forall \ \ i \in \{1, \cdots, S\}$ this implies that if matrix $\boldsymbol{\Delta}$ is not singular there is no acceptable solution. If the matrix is singular than exists a solution other than the trivial solution ($\boldsymbol{\beta} = \mathbf{0}$) and iterative methods can be employed.

Therefore, the Fisher Information perspective on the simple linear regression MKL setting results in a condition for employing parameterised kernel combinations. Only if the matrix $\boldsymbol{\Delta}$ is singular there exists a solution to the system and in that case a convex linear combination with the resulting parameters should be used. In the contrary case where the matrix is non-singular the Fisher Information cannot be maximised and hence non-parametric combinations should be employed in order to reduce the dimensionality of the parameter space.

It is worth noting that when the matrix $\boldsymbol{\Delta}$ is singular, a specific relationship between the eigenvalues of the base kernels arises and the resulting optimal (in a Fisher Information context) solution for the parameters $\boldsymbol{\beta}$ is in direct relation with these base kernel eigenvalues. This preliminary result for the simple linear regression case shows a promising direction that is proposed as future research for probabilistic multiple kernel learning.

8.6 Discussion

In this final Chapter a theoretical analysis on the multiple kernel learning problem was offered. It was shown how the *Flat maximum effect* lower bounds the correlation of differently parameterised ensemble responses with the correlation between the ensemble members thus proving that highly correlated sources do not require parameterised combinations. When the base kernels are correlated, any set of kernel combination parameters produce correlated responses and hence a flat maximum region exists where no real improvement can be achieved by inferring such parameters. This offers a justification for many MKL problems where a simple fixed summation of base kernels has been observed to perform as well as a parameterised combination rule without the added burden of inference on that level and the expanded parameter space.

Further analysis with the *Ambiguity* and the *Bias-Variance-Covariance* decomposition of the loss expressed the direct link between the necessary diversity for base kernels and the resulting composite loss, in an analogy to other ensemble learning approaches such as classifier and regressor ensemble methods. Furthermore, it was demonstrated that when diverse but noisy information sources are present parameterised kernel combinations should be employed in order to address the corrupted signals.

This also motivated the need to assess the information content of the various base kernels besides relying on diversity measures, and led to an alternative information theoretic approach based on *Fisher Information* and *A-optimality* criteria. The preliminary investigation on the linear regression MKL scenario resulted in necessary conditions for employing a parameterised combination over a fixed non-parametric one. This offers a new perspective and a promising research direction for the future as this analysis can be extended on more complex classification scenarios where perhaps further sufficient conditions and optimal parameter settings can be derived towards a Fisher Information MKL methodology. This and other future research directions are proposed in the next final chapter together with the conclusions of this thesis.

Chapter 9

Conclusions and Future Research Directions

This thesis examined the problem of multiclass classification in the presence of multiple and possibly heterogeneous sources of information with Automatic Currency Validation (ACV) as the motivating application. Probabilistic multiple kernel learning (pMKL) approaches were proposed that are able to take into account uncertainty and effectively integrate the information sources towards an overall classification decision. The original contributions of this thesis have been presented in Chapters 3, 4, 5, 7, 6 and 8 with accompanying codes for some of the developments in http://www.dcs.gla.ac.uk/inference/pMKL.

In Chapter 3 the hierarchical Bayesian framework for pMKL was introduced with different kernel combination rules and full Markov chain Monte Carlo (MCMC) solutions. Employing the multinomial probit likelihood gave rise to an efficient Gibbs sampling scheme for multiclass classification that was complemented with additional Metropolis sub-samplers for inference on unobtainable parameter posterior distributions. Parameterised and fixed kernel combination rules were proposed from the standard convex linear summation to novel binary and product kernel combinations.

In Chapter 4 an efficient variational Bayes approximation was proposed to alleviate some of the computational burdens of MCMC while retaining classification performance. The accuracy of the approximation was examined in comparison with the Gibbs sampling solution and it was demonstrated that there is no statistical significant difference in classification performance despite the underestimation of the posterior covariance structure. Computational times for both

training and testing phases were greatly improved while retaining approximate posterior structure and uncertainty estimates.

Further deterministic approximations were proposed in Chapter 5 based on a Maximum a Posteriori and an Expectation Maximisation scheme. Subsequent sparse approximations offered a constructive and a top-down approach for tackling the problem of $\mathcal{O}(N^3)$ scaling, typical in kernel methods, by retaining a small prototypical set of input samples. That led to the generalisation of Relevance Vector Machines to the multiclass multiple kernel setting through the proposed mRVM methodologies that were shown to result in very sparse multiclass solutions while being competitive with methods that utilise the whole training set.

Large scale experimentation to applications in bioinformatics and pattern recognition problems was reported in Chapter 7 where state of the art classification performances with a single pMKL kernel machine were achieved. The proposed pMKL methods were shown to be competitive with multiple classifier methods with the additional benefits of inferring the contribution of each source, and hence assessing their discriminative power, while employing a single classifier.

The motivating application of ACV was presented in Chapter 6 together with a review of the specific application area. Extensive experimental validation of the pMKL methods on international currencies and denominations was reported and it was shown that an overall integration of the various modalities significantly improves recognition rates when discriminating genuine from counterfeit currency notes. Further insight on machine discriminative regions of the currency notes was gained and currency specific informative modalities were identified.

Finally, Chapter 8 offered a theoretical analysis of multiple kernel learning with respect to the decomposition of the ensemble loss and the flat maximum effect observed in kernel combination approaches. The diversity of information sources was identified and justified as a necessary condition for performance improvement via multiple kernel learning methods while a preliminary analysis with the Fisher information criteria on linear kernel regression provided the basis for one of the following promising future research directions.

9.1 Future Research Directions

As natural, model extensions, further issues and questions on probabilistic multiple kernel learning remain open. Some of them, judged as most pertinent are proposed here.

The theoretical analysis in Chapter 8, although promising, is still incomplete. Maximising the Fisher information seems a very promising direction and it has only been addressed in this thesis for the limited linear multiple kernel regression setting. Considering the sigmoidal likelihoods for classification will result in a measure for optimising the Fisher information for the truly interesting classification scenario. The hope is that an intuitive eigenvalue relationship, such as the result of Sun et al. (2004) for kernel target alignment optimisation, will emerge and provide an alternative procedure for pMKL based on Fisher optimality criteria. This is considered as an imminent research direction.

Furthermore, a limitation that has not been addressed in this thesis is stationarity of kernel combinations. The very recent work of Christoudias et al. (2009) follows the appropriate direction, similar to the hierarchical mixture of experts method (Jordan and Jacobs 1994), of introducing locality into the problem and designing covariance combinations dependent on the input space. However, a series of relatively unjustified approximations is followed to tackle the inference of covariance hyper-parameters and the hope is that by employing a gating function that partitions the input space, and avoid the problem of hyper-parameter covariance estimation with the adoption of the proposed framework, would be preferable and more efficient.

Another research direction stemming from the sparse models proposed in Chapter 5 is to pursue joint feature and sample sparsity. This will have the benefit of identifying both significant features and samples within a common framework and could potentially improve classification performances in problems where noisy uninformative dimensions exist. Such model would achieve sparsity in three levels and could potentially identify samples, attributes and feature sets (kernels) which would be of interest to bioinformatics and medical informatics research. However, this does not look trivial within the existing type-II maximum likelihood procedure as the marginal likelihood cannot be decomposed to individual contributions from sample attributes.

A direct, but arguably less interesting direction, would be to extend the

present methods to alternative sigmoidal likelihoods such as the softmax or the log-log models and further Bayesian approximations such as expectation propagation (Minka 2001) which scales favourably to competing approaches. This would extend the Bayesian approaches to pMKL and cover a wider spectrum of models and approximations.

The major research direction in MKL appears to be (see literature review) towards non-linear kernel combination rules and inference of the relationship directly from the evidence. Although increasing the combination complexity will prove beneficial in specific classification tasks, first a theoretical direction that will provide the conditions under which further non-linear combinations are expected to improve is needed. Summation of kernels implies concatenation of feature expansions and product rules result in tensor products of feature expansions, further non-linear combinations will require a better understanding of the implicit mapping.

For the motivating application of ACV, specific open research directions include the automatic inference of segmentation level via a wrapper feature extraction approach and the extension of MKL to novelty detection. Having offered a formal probabilistic framework for information integration in multiclass classification and computational viable solutions the interest lies now on risk assessment and decision making. These can and should be addressed via Bayesian decision theory (Berger 1985).

Finally, the i.i.d assumption, common to standard supervised learning problems, needs to be addressed for ACV. Currency validation is not a stationary process as the class conditional distribution of currency notes constantly changes (covariate shift) over time as the notes age in at least a geographical dependent way (environmental conditions and societal habits). Hence, even the inference of kernel combination parameters, signifying the discriminatory strength of various channels, needs to be non-stationary over time and adaptive. This research direction leads to online learning and tracking, which at the moment is addressed via importance sampling approaches such as sequential Monte Carlo, and it is a promising research avenue for proposing *adaptive* probabilistic multiple kernel learning.

Appendix A

Posterior Inference in MCMC

A.1 Kernel Combination Parameters

The Metropolis-Hastings sub-samplers and additional Gibbs steps are given for the parameters and associated hyper-parameters of each kernel combination case.

A.1.1 Convex Linear Combination

In the *convex linear* case the MH subsamplers employed, based on the prior distributions placed on the model in Chapter 3 and symmetric proposal distributions, have acceptance ratios:

$$\mathcal{R}\left(\boldsymbol{\beta}^{i}, \boldsymbol{\beta}^{*}\right) = \min\left(1, \frac{\prod_{n,c=1}^{N,C} \mathcal{N}_{\mathbf{y}_{nc}}\left(\mathbf{w}_{c}^{\mathsf{T}}\mathbf{k}_{n}^{\boldsymbol{\beta}^{*}}, 1\right)\mathcal{D}_{\boldsymbol{\beta}^{*}}(\boldsymbol{\rho})}{\prod_{n,c=1}^{N,C} \mathcal{N}_{\mathbf{y}_{nc}}\left(\mathbf{w}_{c}^{\mathsf{T}}\mathbf{k}_{n}^{\boldsymbol{\beta}^{i}}, 1\right)\mathcal{D}_{\boldsymbol{\beta}^{i}}(\boldsymbol{\rho})}\right) \tag{A.1}$$

$$\mathcal{R}\left(\boldsymbol{\rho}^{i}, \boldsymbol{\rho}^{*}\right) = \min\left(1, \frac{\mathcal{D}_{\boldsymbol{\beta}}\left(\boldsymbol{\rho}^{*}\right)\prod_{s=1}^{S}\mathcal{G}_{\rho_{s}^{*}}(\lambda,\mu)}{\mathcal{D}_{\boldsymbol{\beta}}\left(\boldsymbol{\rho}^{i}\right)\prod_{s=1}^{S}\mathcal{G}_{\rho_{s}^{i}}(\lambda,\mu)}\right) \tag{A.2}$$

with the proposed move symbolised by $*$ and the current state with i.

A.1.2 Weighted Product Combination

In the same manner and following the model's prior distributions for the *weighted product* case from Chapter 3 we have MH subsamplers with acceptance ratios:

$$\mathcal{R}\left(\boldsymbol{\beta}^i, \boldsymbol{\beta}^*\right) = \min\left(1, \frac{\prod_{n,c=1}^{N,C} \mathcal{N}_{\mathbf{y}_{nc}}\left(\mathbf{w}_c^\intercal \mathbf{k}_n^{\boldsymbol{\beta}^*}, 1\right) \prod_{s=1}^{S} \mathcal{G}_{\beta_s^*}(\pi_s, \chi_s)}{\prod_{n,c=1}^{N,C} \mathcal{N}_{\mathbf{y}_{nc}}\left(\mathbf{w}_c^\intercal \mathbf{k}_n^{\boldsymbol{\beta}^i}, 1\right) \prod_{s=1}^{S} \mathcal{G}_{\beta_s^i}(\pi_s, \chi_s)}\right) \tag{A.3}$$

$$\mathcal{R}\left(\boldsymbol{\pi}^i, \boldsymbol{\pi}^*\right) = \min\left(1, \frac{\prod_{s=1}^{S} \mathcal{G}_{\beta_s}(\pi_s^*, \chi_s) \prod_{s=1}^{S} \mathcal{E}_{\pi_s^*}(\mu)}{\prod_{s=1}^{S} \mathcal{G}_{\beta_s}(\pi_s^i, \chi_s) \prod_{s=1}^{S} \mathcal{E}_{\pi_s^i}(\mu)}\right) \tag{A.4}$$

$$\mathcal{R}\left(\boldsymbol{\chi}^i, \boldsymbol{\chi}^*\right) = \min\left(1, \frac{\prod_{s=1}^{S} \mathcal{G}_{\beta_s}(\pi_s, \chi_s^*) \prod_{s=1}^{S} \mathcal{E}_{\chi_s^*}(\lambda)}{\prod_{s=1}^{S} \mathcal{G}_{\beta_s}(\pi_s, \chi_s^i) \prod_{s=1}^{S} \mathcal{E}_{\chi_s^i}(\lambda)}\right) \tag{A.5}$$

A.1.3 Binary Combination

In the case of the *binary combination* inference of the kernel combination parameters is performed with an additional Gibbs step that depends on the conditional distribution of the auxiliary variables $\mathbf{Y}|\boldsymbol{\beta}$ marginalised over the model regression coefficients $\mathbf{W}$.

The conditional distribution which is the extra Gibbs step introduced, here for switching off kernels, $p\left(\beta_i = 0|\boldsymbol{\beta}_{-i}, \mathbf{Y}, \mathbf{K}_{s:1\ldots S}\right)$ is given by:

$$\frac{p\left(\mathbf{Y}|\beta_i = 0, \boldsymbol{\beta}_{-i}, \mathbf{K}_{s:1\ldots S}\right) p\left(\beta_i = 0|\boldsymbol{\beta}_{-i}\right)}{\sum_{j=0}^{1} p\left(\mathbf{Y}|\beta_i = j, \boldsymbol{\beta}_{-i}, \mathbf{K}_{s:1\ldots S}\right) p\left(\beta_i = j|\boldsymbol{\beta}_{-i}\right)}$$

where $\mathbf{K}_{s:1\ldots S}$ are all the base kernels. The case for switching on kernels follows logically from the above.

Finally, the marginal likelihood that the Gibbs step depends on, is given, see (Denison et al. 2002), by:

$$p\left(\mathbf{Y}|\boldsymbol{\beta}, \mathbf{K}_{s:1\ldots S}\right) = \prod_{c=1}^{C} (2\pi)^{-\frac{N}{2}} \mid \boldsymbol{\Omega}_c \mid^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\mathbf{y}_c^\intercal \boldsymbol{\Omega}_c \mathbf{y}_c\right\} \tag{A.6}$$

where $\mathbf{\Omega}_c = \mathbf{I} + \mathbf{K}^{\boldsymbol{\beta\Theta}}\mathbf{A}_c^{-1}\mathbf{K}^{\boldsymbol{\beta\Theta}}$

A.2 Kernel Parameters

Finally, the MH ratio for the kernel parameters following the prior Gamma distribution from Chapter 3 is:

$$\mathcal{R}\left(\mathbf{\Theta}^i, \mathbf{\Theta}^*\right) = \min\left(1, \frac{\prod_{c,n=1}^{C,N} \mathcal{N}_{\mathbf{y}_{cn}}\left(\mathbf{w}_c\mathbf{k}_n^{\boldsymbol{\beta\Theta}^*}, 1\right) \prod_{s,d=1}^{S,D} \mathcal{G}_{\theta^*_{sd}}(\omega, \phi)}{\prod_{c,n=1}^{C,N} \mathcal{N}_{\mathbf{y}_{cn}}\left(\mathbf{w}_c\mathbf{k}_n^{\boldsymbol{\beta\Theta}^i}, 1\right) \prod_{s,d=1}^{S,D} \mathcal{G}_{\theta^i_{sd}}(\omega, \phi)}\right) \tag{A.7}$$

again with the proposed move symbolised by $*$ and the current state with i.

Appendix B

Variational Approximations

B.1 Approximate posterior distributions

The full derivations are given here for the approximate posteriors of the pMKL model under the variational approximation. A first order approximation is employed for the kernel parameters $\boldsymbol{\Theta}$, i.e. $\mathbb{E}_{Q(\boldsymbol{\theta})}\{\mathbf{K}_i^{\boldsymbol{\theta}_i}\mathbf{K}_j^{\boldsymbol{\theta}_j}\} \approx \mathbf{K}_i^{\widetilde{\boldsymbol{\theta}}_i}\mathbf{K}_j^{\widetilde{\boldsymbol{\theta}}_j}$, to avoid nonlinear contributions to the expectation. The same approximation is applied to the *weighted product* kernel rule for the combinatorial parameters $\boldsymbol{\beta}$ where $\mathbb{E}_{Q(\boldsymbol{\beta})}\{\mathbf{K}_i^{\beta_i}\mathbf{K}_j^{\beta_j}\} \approx \mathbf{K}_i^{\widetilde{\beta}_i}\mathbf{K}_j^{\widetilde{\beta}_j}$

B.1.1 $Q(\mathbf{Y})$

$$\begin{aligned} Q(\mathbf{Y}) &\propto \exp\{\mathbb{E}_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\{\log p(\mathbf{t}|\mathbf{Y}) + \log p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta},\boldsymbol{\Theta})\}\} \\ &\propto \prod_{n=1}^{N} \delta^{t_n} \exp\{\mathbb{E}_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})} \log p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta},\boldsymbol{\Theta})\} \end{aligned} \tag{B.1}$$

where the exponential term, after denoting $\mathbb{E}_\dagger = \mathbb{E}_{Q(\mathbf{W})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}$ can be analysed as follows:

$$
\begin{aligned}
\exp\{\mathbb{E}_{\dagger}\log p\left(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta}\right)\} &= \exp\left\{\mathbb{E}_{\dagger}\log\prod_{n=1}^{N}\mathcal{N}_{\mathbf{y}_n}\left(\mathbf{W}^{\intercal}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}},\mathbf{I}\right)\right\} \\
&= \exp\left\{\mathbb{E}_{\dagger}\left\{\sum_{n=1}^{N}\log|2\pi\mathbf{I}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\left(\mathbf{y}_n-\mathbf{W}^{\intercal}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}\right)^{\intercal}\left(\mathbf{y}_n-\mathbf{W}^{\intercal}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}\right)\right)\right\}\right\} \\
&= \exp\left\{\mathbb{E}_{\dagger}\left\{\sum_{n=1}^{N}\left(-\frac{1}{2}\left(\mathbf{y}_n^{\intercal}\mathbf{y}_n-2\mathbf{y}_n^{\intercal}\mathbf{W}^{\intercal}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}+\left(\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}\right)^{\intercal}\mathbf{W}\mathbf{W}^{\intercal}\mathbf{k}_n^{\boldsymbol{\beta}\boldsymbol{\Theta}}\right)\right)\right\}\right\} \\
&= \exp\left\{\sum_{n=1}^{N}-\frac{1}{2}\left(\mathbf{y}_n^{\intercal}\mathbf{y}_n-2\mathbf{y}_n^{\intercal}\widetilde{\mathbf{W}}^{\intercal}\mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\boldsymbol{\Theta}}}+\sum_{i,j=1}^{S}\widetilde{\boldsymbol{\beta}_i\boldsymbol{\beta}_j}\left(\mathbf{k}_{in}^{\tilde{\boldsymbol{\theta}}_i}\right)^{\intercal}\widetilde{\mathbf{W}\mathbf{W}^{\intercal}}\mathbf{k}_{jn}^{\tilde{\boldsymbol{\theta}}_j}\right)\right\} \quad \text{(B.2)}
\end{aligned}
$$

where $\mathbf{k}_{in}^{\tilde{\boldsymbol{\theta}}_i}$ is the n^{th} N-dimensional column vector of the i^{th} base kernel with kernel parameters $\widetilde{\boldsymbol{\theta}}_i$. Now from this exponential term we can form the posterior distribution as a Gaussian and reach to the final expression:

$$
Q\left(\mathbf{Y}\right)\propto\prod_{n=1}^{N}\delta\left(y_{i,n}>y_{k,n}\forall k\neq i\right)\delta\left(t_n=i\right)\mathcal{N}_{\mathbf{y}_n}\left(\widetilde{\mathbf{W}}^{\intercal}\mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\boldsymbol{\Theta}}},\mathbf{I}\right) \quad \text{(B.3)}
$$

which is a C-dimensional conically truncated Gaussian.

B.1.2 $Q\left(\mathbf{W}\right)$

$$
Q\left(\mathbf{W}\right)\propto\exp\left\{\mathbb{E}_{Q(\mathbf{Y})Q(\boldsymbol{\beta})Q(\mathbf{A})Q(\boldsymbol{\Theta})}\{\log p\left(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta}\right)+\log p\left(\mathbf{W}|\mathbf{A}\right)\}\right\}
$$

$$
\begin{aligned}
&= \exp\left\{\mathbb{E}_{Q(\mathbf{Y})Q(\boldsymbol{\beta})Q(\boldsymbol{\Theta})}\left\{\sum_{c=1}^{C}-\frac{1}{2}\left(\mathbf{y}_c^{\intercal}\mathbf{y}_c-2\mathbf{y}_c^{\intercal}\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{w}_c+\mathbf{w}_c^{\intercal}\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{K}^{\boldsymbol{\beta}\boldsymbol{\Theta}}\mathbf{w}_c\right)\right\}\right. \\
&+ \left.\mathbb{E}_{Q(\mathbf{A})}\left\{\sum_{c=1}^{C}-\frac{1}{2}\log\prod_{n=1}^{N}\alpha_{nc}^{-1}-\frac{1}{2}\mathbf{w}_c^{\intercal}\left(\mathbf{A}_c\right)\mathbf{w}_c\right\}\right\} \\
&= \exp\left\{\sum_{c=1}^{C}-\frac{1}{2}\left\{\widetilde{\mathbf{y}_c^{\intercal}\mathbf{y}_c}-2\widetilde{\mathbf{y}_c}^{\intercal}\mathbf{K}^{\tilde{\boldsymbol{\beta}}\tilde{\boldsymbol{\Theta}}}\mathbf{w}_c+\mathbf{w}_c^{\intercal}\sum_{i=1}^{S}\sum_{j=1}^{S}\widetilde{\beta_i\beta_j}\mathbf{K}^{i\tilde{\boldsymbol{\theta}}_i}\mathbf{K}^{j\tilde{\boldsymbol{\theta}}_j}\mathbf{w}_c\right.\right. \\
&+ \left.\left.\sum_{n=1}^{N}\widetilde{\log\alpha_{nc}^{-1}}+\mathbf{w}_c^{\intercal}\left(\widetilde{\mathbf{A}}_c\right)\mathbf{w}_c\right\}\right\} \quad \text{(B.4)}
\end{aligned}
$$

Again we can form the posterior expectation as a new Gaussian:

$$Q(\mathbf{W}) \propto \prod_{c=1}^{C} \mathcal{N}_{\mathbf{w}_c}\left(\mathbf{V}_c \mathbf{K}^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}} \widetilde{\mathbf{y}}_c, \mathbf{V}_c\right) \tag{B.5}$$

where $\mathbf{V}_c$ is the covariance matrix defined as:

$$\mathbf{V}_c = \left(\sum_{i=1}^{S}\sum_{j=1}^{S} \widetilde{\beta_i \beta_j} \mathbf{K}^{i\widetilde{\boldsymbol{\theta}}_i} \mathbf{K}^{j\widetilde{\boldsymbol{\theta}}_j} + \widetilde{\mathbf{A}}_c\right)^{-1} \tag{B.6}$$

and $\widetilde{\mathbf{A}}_c$ is a diagonal matrix of the expected scales $\widetilde{\alpha_{ic}} \ldots \widetilde{\alpha_{Nc}}$ for each class.

B.1.3 $Q(\mathbf{A})$

$$\begin{aligned} Q(\mathbf{A}) &\propto \exp\left\{\mathbb{E}_{Q(\mathbf{W})}\left(\log p(\mathbf{W}|\mathbf{A}) + \log p(\mathbf{A}|\tau, \upsilon)\right)\right\} \\ &= \exp\left\{\mathbb{E}_{Q(\mathbf{W})}\left(\log p(\mathbf{W}|\mathbf{A})\right)\right\} p(\mathbf{A}|\tau, \upsilon) \end{aligned} \tag{B.7}$$

Analysing the exponential term only:

$$\begin{aligned} & \exp\left\{\mathbb{E}_{Q(\mathbf{W})}\left(\log \prod_{c=1}^{C}\prod_{n=1}^{N} \mathcal{N}_{w_{nc}}(0, \alpha_{nc})\right)\right\} \\ = & \exp\left\{\mathbb{E}_{Q(\mathbf{W})}\left(\sum_{c=1}^{C}\sum_{n=1}^{N} -\frac{1}{2}\log \alpha_{nc}^{-1} - \frac{1}{2} w_{nc}^2 \alpha_{cn}\right)\right\} \\ = & \prod_{c=1}^{C}\prod_{n=1}^{N} \alpha_{nc}^{\frac{1}{2}} \exp\left(-\frac{1}{2}\widetilde{w_{nc}^2}\alpha_{nc}\right) \end{aligned}$$

which combined with the $p(\mathbf{Z}|\tau, \upsilon)$ prior Gamma distribution leads to the approximate posterior distribution:

$$Q(\mathbf{A}) = \prod_{n,c=1}^{N,C} \mathcal{G}_{\alpha_{nc}}\left(\tau + \frac{1}{2}, \upsilon + \frac{1}{2}\widetilde{w_{nc}^2}\right) \tag{B.8}$$

B.1.4 $Q(\boldsymbol{\beta}), Q(\boldsymbol{\rho}), Q(\boldsymbol{\Theta})$

Importance sampling techniques (Andrieu 2003) are used to approximate these posterior distributions as they are intractable. We present the case of the *convex linear* composite kernel analytically:

For $Q(\boldsymbol{\rho})$ we have $p(\boldsymbol{\rho}|\boldsymbol{\beta}) \propto p(\boldsymbol{\beta}|\boldsymbol{\rho})\, p(\boldsymbol{\rho}|\mu,\lambda)$ and hence the unnormalised posterior is:

$Q^*(\boldsymbol{\rho}) = p(\boldsymbol{\beta}|\boldsymbol{\rho})\, p(\boldsymbol{\rho}|\mu,\lambda)$ with the importance weights defined as:

$$\mathcal{W}\left(\boldsymbol{\rho}^i\right) = \frac{\dfrac{Q^*\left(\boldsymbol{\rho}^{i'}\right)}{\prod_{s=1}^{S} \mathcal{G}_{\rho_s^{i'}}(\mu,\lambda)}}{\sum_{i=1}^{I} \dfrac{Q^*\left(\boldsymbol{\rho}^i\right)}{\prod_{s=1}^{S} \mathcal{G}_{\rho_s^{i}}(\mu,\lambda)}} = \frac{\mathcal{D}_{\widetilde{\boldsymbol{\beta}}}\left(\boldsymbol{\rho}^{i'}\right)}{\sum_{i=1}^{I} \mathcal{D}_{\widetilde{\boldsymbol{\beta}}}\left(\boldsymbol{\rho}^{i}\right)} \tag{B.9}$$

where I is the total number of samples of $\boldsymbol{\rho}$ taken until now from the product Gamma distributions and i' denotes the current (last) sample. So now we can estimate any function f of $\boldsymbol{\rho}$ based on:

$$\widetilde{f}(\boldsymbol{\rho}) = \sum_{i=1}^{I} f(\boldsymbol{\rho})\, \mathcal{W}\left(\boldsymbol{\rho}^i\right)$$

In the same manner as above but now for $Q(\boldsymbol{\beta})$ and $Q(\boldsymbol{\Theta})$ we can use the unnormalised posteriors $Q^*(\boldsymbol{\beta})$ and $Q^*(\boldsymbol{\Theta})$, where $p(\boldsymbol{\beta}|\boldsymbol{\rho},\mathbf{Y},\mathbf{W}) \propto p(\mathbf{Y}|\mathbf{W},\boldsymbol{\beta})\, p(\boldsymbol{\beta}|\boldsymbol{\rho})$ and $p(\boldsymbol{\Theta}|\omega,\phi,\mathbf{Y},\mathbf{W}) \propto p(\mathbf{Y}|\mathbf{W},\boldsymbol{\Theta})\, p(\boldsymbol{\Theta}|\omega,\phi)$ with importance weights defined as:

$$\mathcal{W}\left(\boldsymbol{\beta}^i\right) = \frac{\dfrac{Q^*\left(\boldsymbol{\beta}^{i'}\right)}{\mathcal{D}_{\boldsymbol{\beta}^{i'}}(\boldsymbol{\rho})}}{\sum_{i=1}^{I} \dfrac{Q^*\left(\boldsymbol{\beta}^i\right)}{\mathcal{D}_{\boldsymbol{\beta}^{i}}(\boldsymbol{\rho})}} = \frac{\prod_{n=1}^{N} \mathcal{N}_{\widetilde{\mathbf{y}}_n}\left(\widetilde{\mathbf{W}}^{\mathsf{T}} \mathbf{k}_n^{\boldsymbol{\beta}^{i'}}, \mathbf{I}\right)}{\sum_{i=1}^{I} \prod_{n=1}^{N} \mathcal{N}_{\widetilde{\mathbf{y}}_n}\left(\widetilde{\mathbf{W}}^{\mathsf{T}} \mathbf{k}_n^{\boldsymbol{\beta}^{i}}, \mathbf{I}\right)}$$

and

$$\mathcal{W}\left(\boldsymbol{\Theta}^i\right) = \frac{\dfrac{Q^*\left(\boldsymbol{\Theta}^{i'}\right)}{\mathcal{G}_{\boldsymbol{\Theta}^{i'}}(\omega,\phi)}}{\sum_{i=1}^{I} \dfrac{Q^*\left(\boldsymbol{\Theta}^i\right)}{\mathcal{G}_{\boldsymbol{\Theta}^{i}}(\omega,\phi)}} = \frac{\prod_{n=1}^{N} \mathcal{N}_{\widetilde{\mathbf{y}}_n}\left(\widetilde{\mathbf{W}}^{\mathsf{T}} \mathbf{k}_n^{\boldsymbol{\Theta}^{i'}}, \mathbf{I}\right)}{\sum_{i=1}^{I} \prod_{n=1}^{N} \mathcal{N}_{\widetilde{\mathbf{y}}_n}\left(\widetilde{\mathbf{W}}^{\mathsf{T}} \mathbf{k}_n^{\boldsymbol{\Theta}^{i}}, \mathbf{I}\right)}$$

and again we can estimate any function g of $\boldsymbol{\beta}$ and h of $\boldsymbol{\Theta}$ as:

$$\widetilde{g}(\boldsymbol{\beta}) = \sum_{i=1}^{I} g(\boldsymbol{\beta})\, \mathcal{W}\left(\boldsymbol{\beta}^i\right) \text{ and } \widetilde{h}(\boldsymbol{\Theta}) = \sum_{i=1}^{I} g(\boldsymbol{\Theta})\, \mathcal{W}\left(\boldsymbol{\Theta}^i\right)$$

B.2 Posterior Expectations for the Auxiliary Variables

As it was shown $Q(\mathbf{Y}) \propto \prod_{n=1}^{N} \delta(y_{n,i} > y_{n,k} \forall k \neq i)\, \delta(t_n = i)\, \mathcal{N}_{\mathbf{y}_n}\left(\widetilde{\mathbf{W}}^{\intercal}\mathbf{k}_n^{\widetilde{\boldsymbol{\beta}}\widetilde{\boldsymbol{\Theta}}}, \mathbf{I}\right)$. Hence $Q(\mathbf{y}_n)$ is a truncated multivariate Gaussian distribution and we need to calculate the correction to the normalizing term $\mathcal{Z}_n$ caused by the truncation. Thus, the posterior expectation can be expressed as

$$Q(\mathbf{y}_n) = \mathcal{Z}_n^{-1} \prod_{c=1}^{C} \mathcal{N}_{y_{nc}}^{t_n}\left(\widetilde{\mathbf{w}}_c^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1\right)$$

where the superscript t_n indicates the truncation needed so that the appropriate dimension i (since $t_n = i \iff y_{ni} > y_{nj} \;\; \forall\, j \neq i$) is the largest.

Now, $\mathcal{Z}_n = P(\mathbf{y}_n \in \mathcal{C})$ where $\mathcal{C} = \{\mathbf{y}_n : y_{in} > y_{jn}\}$ hence

$$\begin{aligned}
\mathcal{Z}_n &= \int_{-\infty}^{+\infty} \mathcal{N}_{y_{in}}\left(\widetilde{\mathbf{w}}_i\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1\right) \prod_{j\neq i} \int_{-\infty}^{y_{ni}} \mathcal{N}_{y_{jn}}\left(\widetilde{\mathbf{w}}_j\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1\right) dy_{nj} dy_{ni} \\
&= \mathbb{E}_{p(u)}\left\{\prod_{j\neq i} \Phi\left(u + \widetilde{\mathbf{w}}_i^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{\mathbf{w}}_j^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}\right)\right\}
\end{aligned}$$

with $p(u) = \mathcal{N}_u(0,1)$. The posterior expectation of y_{nc} for all $c \neq i$ (the auxiliary variables associated with the rest of the classes except the one that object n belongs to) is given by

$\widetilde{y}_{nc} = \mathcal{Z}_n^{-1} \int_{-\infty}^{+\infty} y_{nc} \prod_{j=1}^{C} \mathcal{N}_{y_{nj}}\left(\widetilde{\mathbf{w}}_j^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}\right) dy_{nj} =$

$$\begin{aligned}
&\mathcal{Z}_n^{-1} \int_{-\infty}^{+\infty}\int_{-\infty}^{y_{ni}} y_{nc}\mathcal{N}_{y_{nc}}\left(\widetilde{\mathbf{w}}_c^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}\right) \prod_{j\neq i,c} \mathcal{N}_{y_{ni}}\left(\widetilde{\mathbf{w}}_i^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1\right) \Phi\left(y_{ni} - \widetilde{\mathbf{w}}_j^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}\right) dy_{nc} dy_{ni} \\
&= \\
&\widetilde{\mathbf{w}}_c\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \mathcal{Z}_n^{-1}\mathbb{E}_{p(u)}\left\{\mathcal{N}_u\left(\widetilde{\mathbf{w}}_c^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{\mathbf{w}}_i^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}, 1\right) \prod_{j\neq i,c} \Phi\left(u + \widetilde{\mathbf{w}}_i^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}} - \widetilde{\mathbf{w}}_j^{\intercal}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}\right)\right\} \quad \text{(B.10)}
\end{aligned}$$

For the i^{th} class the posterior expectation y_{ni} (the auxiliary variable associated with the known class of the n^{th} object) is given by:

$$\begin{aligned}\widetilde{y}_{ni} = &= \mathcal{Z}_n^{-1}\int_{-\infty}^{+\infty} y_{ni}\mathcal{N}_{y_{ni}}\left(\widetilde{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}},1\right)\prod_{j\neq i}\Phi\left(y_{ni}-\widetilde{\mathbf{w}}_j^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}\right)dy_{ni}\\ &= \widetilde{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}+\mathcal{Z}_n^{-1}\mathbb{E}_{p(u)}\left\{u\prod_{j\neq i}\Phi\left(u+\widetilde{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}-\widetilde{\mathbf{w}}_j^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}\right)\right\}\\ &= \widetilde{\mathbf{w}}_i^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}+\sum_{c\neq i}\left(\widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n^{\widetilde{\beta}\widetilde{\Theta}}-\widetilde{y}_{nc}\right) \qquad \text{(B.11)}\end{aligned}$$

where we have made use of the fact that for a variable u $\mathcal{N}(0,1)$ and any differentiable function $g(u)$, $\mathbb{E}\{ug(u)\}=\mathbb{E}\{g'(u)\}$.

B.3 Predictive distribution

In order to make a prediction t_* for a new point $\mathbf{x}_*$ we need to know:

$$\begin{aligned}p(t_*=c|\mathbf{x}_*,\mathbf{X},\mathbf{t}) &= \int p(t_*=c|\mathbf{y}_*)\,p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{t})\,d\mathbf{y}_*\\ &= \int \delta_c^* p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{t})\,d\mathbf{y}_* \qquad \text{(B.12)}\end{aligned}$$

Hence we need to evaluate $p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{t})$

$$\begin{aligned}&= \int p(\mathbf{y}_*|\mathbf{W},\mathbf{x}_*)\,p(\mathbf{W}|\mathbf{X},\mathbf{t})\,d\mathbf{W}\\ &= \prod_{c=1}^{C}\int \mathcal{N}_{\mathbf{w}_c^{\mathsf{T}}\mathbf{K}_*}(\mathbf{y}_{c*},\mathbf{I})\,\mathcal{N}_{\mathbf{w}_c}(\mathbf{V}_c\mathbf{K}\widetilde{\mathbf{y}}_c,\mathbf{V}_c)\,d\mathbf{w}_c \qquad \text{(B.13)}\end{aligned}$$

We proceed by analysing the integral, gathering all the terms depending on $\mathbf{w}_c$, completing the square twice and reforming to

$$p(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{t}) = \prod_{c=1}^{C}\int \mathcal{N}_{\mathbf{y}_{c*}}\left(\widetilde{\mathcal{V}}_{c*}\mathbf{K}_*^{\mathsf{T}}\mathbf{\Lambda}_c\mathbf{K}\widetilde{\mathbf{y}}_c,\widetilde{\mathcal{V}}_{c*}\right)\mathcal{N}_{\mathbf{w}_c}(\mathbf{\Lambda}_c(\mathbf{K}\widetilde{\mathbf{y}}_c+\mathbf{K}_*\mathbf{y}_{c*}),\mathbf{\Lambda}_c) \qquad \text{(B.14)}$$

with

$$\widetilde{\mathcal{V}}_{c*} = \left(\mathbf{I} - \mathbf{K}_*^{\mathsf{T}} \boldsymbol{\Theta}_c \mathbf{K}_*\right)^{-1} \qquad \text{(Ntest} \times \text{Ntest)} \tag{B.15}$$

and

$$\boldsymbol{\Lambda}_c = \left(\mathbf{K}_* \mathbf{K}_*^{\mathsf{T}} + \mathbf{V}_c^{-1}\right)^{-1} \qquad \text{(N} \times \text{N)} \tag{B.16}$$

Finally we can simplify $\widetilde{\mathcal{V}}_{c*}$ by applying the Woodbury identity and reduce its form to:

$$\widetilde{\mathcal{V}}_{c*} = \left(\mathbf{I} + \mathbf{K}_*^{\mathsf{T}} \mathbf{V}_c \mathbf{K}_*\right) \tag{B.17}$$

Now the Gaussian distribution with respect to $\mathbf{w}_c$ integrates to one and we are left with

$$p\left(\mathbf{y}_* | \mathbf{x}_*, \mathbf{X}, \mathbf{t}\right) = \prod_{c=1}^{C} \mathcal{N}_{\mathbf{y}_{c*}}\left(\widetilde{\mathbf{m}}_{c*}, \widetilde{\mathcal{V}}_{c*}\right) \tag{B.18}$$

where $\widetilde{\mathbf{m}}_{c*} = \widetilde{\mathcal{V}}_{c*} \mathbf{K}_*^{\mathsf{T}} \boldsymbol{\Lambda}_c \mathbf{K} \widetilde{\mathbf{y}}_c$

Hence we can go back to the predictive distribution and consider the case of a single test point with associated scalars $\widetilde{m}_{c*}$ and $\widetilde{\nu_{c*}}$

$p\left(t_* = c | \mathbf{x}_*, \mathbf{X}, \mathbf{t}\right)$

$$\begin{aligned}
&= \int \delta_c^* \prod_{c=1}^{C} \mathcal{N}_{y_{c*}}\left(\widetilde{m_{c*}}, \widetilde{\nu_{c*}}\right) dy_{c*} \\
&= \int_{-\infty}^{+\infty} \mathcal{N}_{y_{c*}}\left(\widetilde{m_{c*}}, \widetilde{\nu_{c*}}\right) \prod_{j \neq c} \int_{-\infty}^{y_{c*}} \mathcal{N}_{y_{j*}}\left(\widetilde{m_{j*}}, \widetilde{\nu_{j*}}\right) dy_{j*} dy_{c*} \\
&= \int_{-\infty}^{+\infty} \mathcal{N}_{y_{c*} - \widetilde{m_{c*}}}\left(0, \widetilde{\nu_{c*}}\right) \prod_{j \neq c} \int_{-\infty}^{y_{c*} - \widetilde{m_{j*}}} \mathcal{N}_{y_{j*} - \widetilde{m_{j*}}}\left(0, \widetilde{\nu_{j*}}\right) dy_{j*} dy_{c*}
\end{aligned} \tag{B.19}$$

Setting $u = \left(y_{c*} - \widetilde{m_{c*}}\right) \widetilde{\nu_{c*}}^{-1}$ and $x = \left(y_{c*} - \widetilde{m_{j*}}\right) \widetilde{\nu_{j*}}^{-1}$ we have:

$$\begin{aligned}
p\left(t_* = c | \mathbf{x}_*, \mathbf{X}, \mathbf{t}\right) &= \int_{-\infty}^{+\infty} \mathcal{N}_u\left(0, 1\right) \prod_{j \neq c} \int_{-\infty}^{\left(u\widetilde{\nu_{c*}} + \widetilde{m_{c*}} - \widetilde{m_{j*}}\right)\widetilde{\nu_{j*}}^{-1}} \mathcal{N}_x\left(0, 1\right) dx du \\
&= \mathbb{E}_{p(u)} \left\{ \prod_{j \neq c} \Phi\left[\frac{1}{\widetilde{\nu_{j*}}} \left(u\widetilde{\nu_{c*}} + \widetilde{m_{c*}} - \widetilde{m_{j*}}\right) \right] \right\}
\end{aligned} \tag{B.20}$$

B.4 Lower bound

The variational lower bound, conditioning on current values of $\boldsymbol{\beta}, \boldsymbol{\Theta}, \boldsymbol{A}$ and $\boldsymbol{\rho}$ has the relevant components:

$$\begin{aligned} & \mathbb{E}_{Q(\mathbf{Y})Q(\mathbf{W})}\left\{\log p\left(\mathbf{Y}|\mathbf{W}, \boldsymbol{\beta}, \mathbf{K}\right)\right\} \\ + \quad & \mathbb{E}_{Q(\mathbf{Y})Q(\mathbf{W})}\left\{\log p\left(\mathbf{W}|\mathbf{A}, \mathbf{K}\right)\right\} \\ - \quad & \mathbb{E}_{Q(\mathbf{Y})}\left\{\log Q\left(\mathbf{Y}\right)\right\} \\ - \quad & \mathbb{E}_{Q(\mathbf{W})}\left\{\log Q\left(\mathbf{W}\right)\right\} \end{aligned} \tag{B.21}$$

which, by noting that the expectation of a quadratic form under a Gaussian is another quadratic form plus a constant, leads to the following expression for the lower bound

$$\begin{aligned} \text{Lower Bound} = & -\tfrac{NC}{2}\log 2\pi - \tfrac{1}{2}\textstyle\sum_{c=1}^{C}\sum_{n=1}^{N}\left\{\widetilde{y_{nc}^2} + \mathbf{k}_n^{\mathsf{T}}\widetilde{\mathbf{w}_c\mathbf{w}_c^{\mathsf{T}}}\mathbf{k}_n - 2\widetilde{y}_{nc}\widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n\right\} \\ & -\tfrac{NC}{2}\log 2\pi - \tfrac{1}{2}\textstyle\sum_{c=1}^{C}\log|\mathbf{Z}_c| - \tfrac{1}{2}\sum_{c=1}^{C}\widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{A}_c\widetilde{\mathbf{w}}_c - \tfrac{1}{2}\sum_{c=1}^{C}\mathrm{Tr}\left[\mathbf{A}_c\mathbf{V}_c\right] \\ & + \textstyle\sum_{n=1}^{N}\log\mathcal{Z}_n + \tfrac{NC}{2}\log 2\pi + \tfrac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{C}\left(\widetilde{y^2}_{cn} - 2\widetilde{y}_{cn}\widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n + \mathbf{k}_n^{\mathsf{T}}\widetilde{\mathbf{w}}_c\widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{k}_n\right) \\ & + \tfrac{NC}{2}\log 2\pi + \tfrac{1}{2}\textstyle\sum_{c=1}^{C}\log|\mathbf{V}_c| + \tfrac{NC}{2} \end{aligned} \tag{B.22}$$

which simplifies to our final expression

$$\begin{aligned} \text{Lower Bound} \quad = \quad & \frac{NC}{2} + \frac{1}{2}\sum_{c=1}^{C}\log|\mathbf{V}_c| + \sum_{n=1}^{N}\log\mathcal{Z}_n \\ - \quad & \frac{1}{2}\sum_{c=1}^{C}\mathrm{Tr}\left[\mathbf{A}_c\mathbf{V}_c\right] - \frac{1}{2}\sum_{c=1}^{C}\widetilde{\mathbf{w}}_c^{\mathsf{T}}\mathbf{A}_c\widetilde{\mathbf{w}}_c \\ - \quad & \frac{1}{2}\sum_{c=1}^{C}\log|\mathbf{A}_c^{-1}| - \frac{1}{2}\sum_{c=1}^{C}\sum_{n=1}^{N}\mathbf{k}_n^{\mathsf{T}}\mathbf{V}_c\mathbf{k}_n \end{aligned} \tag{B.23}$$

Appendix C

Quadratic Program

The joint likelihood of the model as depicted in Figure 5.1 is given by:

$$L = p\left(\mathbf{t}|\mathbf{Y}\right) p\left(\mathbf{Y}|\mathbf{K}, \mathbf{W}, \boldsymbol{\beta}\right) p\left(\mathbf{W}|\mathbf{A}\right) p\left(\mathbf{A}|\tau, \upsilon\right) p\left(\boldsymbol{\beta}\right) \tag{C.1}$$

where by placing a uniform Dirichlet prior on $\boldsymbol{\beta} \sim \mathcal{D}_{\boldsymbol{\beta}}(\boldsymbol{\rho})$ with $\rho_s = 1$, and disregarding irrelevant terms leads to the following expression for the logarithm of the joint likelihood:

$$\begin{aligned} \mathcal{L} &= \sum_{n=1}^{N} \delta^{t_n} - \frac{1}{2} \sum_{n,c=1}^{N,C} \left(y_{nc}^2 - 2 y_{nc} \mathbf{w}_c^{\mathsf{T}} \mathbf{k}_n + (\mathbf{w}_c^{\mathsf{T}} \mathbf{k}_n)^2 \right) \\ &+ \frac{1}{2} \sum_{n,c=1}^{N,C} \log \alpha_{nc} - \frac{1}{2} \sum_{n,c=1}^{N,C} \alpha_{nc} w_{nc}^2 \\ &+ (\tau - 1) \sum_{n,c=1}^{N,C} \log \alpha_{nc} - \upsilon \alpha_{nc} \end{aligned} \tag{C.2}$$

Recalling that the composite kernel is a function of the parameters $\boldsymbol{\beta}$, i.e. $\mathbf{K} = \sum_{s=1}^{S} \beta_s \mathbf{K}_s$, setting $m_{nc} = \mathbf{w}_c^{\mathsf{T}} \mathbf{k}_n = \sum_{s=1}^{S} \beta_s \mathbf{w}_c^{\mathsf{T}} \mathbf{k}_{sn} = \sum_{s=1}^{S} m_{snc}$ and maximizing the logarithm of the expected joint likelihood with respect to the kernel parameters $\boldsymbol{\beta}$ leads to:

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \sum_{n,c=1}^{N,C} \tilde{y}_{nc} m_{nc} - \frac{1}{2} m_{nc}^2 = \arg\min_{\boldsymbol{\beta}} \sum_{n,c=1}^{N,C} \frac{1}{2} m_{nc}^2 - \tilde{y}_{nc} m_{nc} \tag{C.3}$$

which is a Quadratic Program (QP) with inequality and equality constraints due to the Dirichlet prior on the kernel combination parameters. Finally, the QP can be expressed in matrix format as:

$$\begin{cases} \hat{\boldsymbol{\beta}} & = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\Omega}\boldsymbol{\beta} - \boldsymbol{\beta}^{\mathsf{T}}\mathbf{f} \\ \text{subject to} & \beta_s \geq 0 \text{ and } \sum_{s=1}^{S}\beta_s = 1 \end{cases} \tag{C.4}$$

where $\boldsymbol{\Omega}_{ij} = \sum_{n,c}^{N,C} m_{inc}m_{j_{nc}}$ is an $S \times S$ matrix and $f_i = \sum_{n,c}^{N,C} m_{inc}\tilde{y}_{nc}$.

Bibliography

Ahmadi, A., Omatu, S., Fujinaka, T. and Kosaka, T.: 2004, Improvement of reliability in banknote classification using reject option and local PCA, *Information Sciences* **168**, 277–293.

Ahmadi, A., Omatu, S. and Kosaka, T.: 2003a, A PCA based method for improving the reliability of bank note classifier machines, *ISPA 2003, 3rd International Symposium on Image and Signal Processing and Analysis*, pp. 494–499.

Ahmadi, A., Omatu, S. and Kosaka, T.: 2003b, A reliable method for recognition of paper currency by approach to local PCA, *IJCNN 2003, International Joint Conference on Neural Networks*, Vol. 2, pp. 1258–1262.

Ahmadi, A., Omatu, S. and Kosaka, T.: 2003c, A study on evaluating and improving the reliability of bank note neuro-classifiers, *SICE Annual Conference in Fukui*, Fukui University, Japan, pp. 2550–2554.

Ahmadi, A., Omatu, S. and Kosaka, T.: 2004, Improvement of the reliability of bank note classifier machines, *IJCNN 2004, International Joint Conference on Neural Networks*, Vol. 2, pp. 1313–1316.

Ahmadi, A., Omatu, S., Kosaka, T. and Fujinaka, T.: 2004, A reliable method for classification of bank notes using artificial neural networks, *Artificial Life Robotics* **8**(2), 133–139.

Albert, J. and Chib, S.: 1993, Bayesian analysis of binary and polychotomous response data, *Journal of the American Statistical Association* **88**, 669–679.

Andreeva, A., Howorth, D., Brenner, S., Hubbard, T., Chothia, C. and Murzin, A.: 2004, Scop database in 2004: refinements integrate structure and sequence family data, *Nucleic Acids Res.* **32**, 226–229.

Andrieu, C.: 2003, An introduction to MCMC for machine learning, *Machine Learning* **50**, 5–43.

Bach, F.: 2008, Consistency of the group Lasso and multiple kernel learning, *The Journal of Machine Learning Research* **9**, 1179–1225.

Bach, F., Lanckriet, G. and Jordan, M.: 2004, Multiple kernel learning, conic duality, and the SMO algorithm, *Proceedings of the twenty-first international conference on Machine learning*, ACM New York, NY, USA.

Bach, F., Rakotomamonjy, A., Canu, S. and Grandvalet, Y.: 2008, SimpleMKL, *Journal of Machine Learning Research* **9**, 2491–2521.

Baker, D. and Sali, A.: 2001, Protein structure prediction and structural genomics, *Science* pp. 93–96.

Beal, M. J.: 2003, *Variational Algorithms for approximate Bayesian Inference*, PhD thesis, The Gatsby Computational Neuroscience Unit, University College London.

Ben-Hur, A. and Brutlag, D.: 2003, Remote homology detection: a motif based approach, *Bioinformatics* **19**, 126–133.

Berger, J. O.: 1985, *Statistical Decision Theory and Bayesian Analysis*, Springer Series in Statistics, Springer.

Bethe, H. A.: 1935, Statistical theory of superlattices, *Proc. R. Soc. London* **Series A**(150), 552–575.

Bishop, C. M.: 1996, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK.

Bishop, C. M.: 2006, *Pattern Recognition and Machine Learning*, Springer, New York, USA.

Bishop, C. M. and Svensen, M.: 2003, Bayesian hierarchical mixtures of experts, *Proceedings Nineteenth Conference on Uncertainty in Artificial Intelligence*, pp. 57–64.

Bogert, B. P., Healy, M. J. R. and Tukey, J. W.: 1963, The quefrency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking, *Proc. Symposium Time Series Analysis*, pp. 209–243.

Breiman, L.: 1996, Bagging predictors, *Machine learning* **24**(2), 123–140.

Brown, G. and Wyatt, J.: 2003, The use of the Ambiguity Decomposition in neural network ensemble learning methods, *International Conference on Machine Learning (ICML '03)*.

Chi, Z., Wu, J. and Yan, H.: 1995, Handwritten numeral recognition using self-organizing maps and fuzzy rules, *Pattern Recognition* **28**(1), 59–66.

Chib, S.: 1995, Marginal likelihood from the Gibbs output, *Journal of the American Statistical Association* **90**(432).

Chou, K.: 2005, Using amphiphilic pseudo-amino acid composition to predict enzyme subfamily classes, *Bioinformatics* **21**, 10–19.

Chou, K. and Zhang, C.: 1995, Prediction of protein structural classes, *Critical Revi. Biochem. Mol. Biol.* **30**, 275–349.

Christoudias, M., Urtasun, R. and Darrell, T.: 2009, Bayesian localized multiple kernel learning, *Technical report no. ucb/eecs-2009-96*, Electrical Engineering and Computer Sciences, University of California at Berkeley.

Crammer, K., Keshet, J. and Singer, Y.: 2003, Kernel design using boosting, *Advances in Neural Information Processing Systems 17*, Citeseer, pp. 553–560.

Cristianini, N., Kandola, J. and Elissee, A.: 2001, On kernel target alignment, *Advances in Neural Information Processing Systems 14*.

Cucala, L., Marin, J.-M., Robert, C. P. and Titterington, D. M.: 2009, A Bayesian reassessment of Nearest-Neighbor classification, *Journal of the American Statistical Association* **104**(485), 263–273.

Damoulas, T.: 2006, Discriminative significance identification via Markov chain Monte Carlo on generalized linear regression models, *Confidential Internal Report Rev. B. No. 002*, NCR Labs.

Damoulas, T.: 2008a, Feature selection of diverse signals for hierarchical Bayesian kernel machine, *Confidential Internal Report Rev. B. No. 008*, NCR Labs.

Damoulas, T.: 2008b, Learning curve investigation for multinomial probit classifier, *Confidential Internal Report Rev. B. No. 007*, NCR Labs.

Damoulas, T.: 2009, Inferring sparse kernel combinations and relevance vectors, *Confidential Internal Report Rev. B. No. 010*, NCR Labs.

Damoulas, T. and Girolami, M. A.: 2008, Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection, *Bioinformatics* **24**(10), 1264–1270.

Damoulas, T. and Girolami, M. A.: 2009a, Combining feature spaces for classification, *Pattern Recognition* **42**(11), 2671–2683.

Damoulas, T. and Girolami, M. A.: 2009b, Combining information with a Bayesian multi-class multi-kernel pattern recognition machine, *in* R. K. De, D. P. Mandal and A. Ghosh (eds), *Machine Interpretation of Patterns: Image Analysis, Data Mining and Bioinformatics*, World Scientific Press. In Print.

Damoulas, T. and Girolami, M. A.: 2009c, Pattern recognition with a Bayesian kernel combination machine, *Pattern Recognition Letters* **30**(1), 46–54.

Damoulas, T., Ying, Y., Girolami, M. A. and Campbel, C.: 2008, Inferring sparse kernel combinations and relevant vectors: An application to subcellular localization of proteins, *IEEE, International Conference on Machine Learning and Applications (ICMLA '08)*, pp. 577–582.

de Freitas, N., Højen-Sørensen, P., Jordan, M. and Russell, S.: 2001, Variational MCMC, *Proceedings of the 17th conference in Uncertainty in Artificial Intelligence*, pp. 120–127.

Denison, D. G. T., Holmes, C. C., Mallick, B. K. and Smith, A. F. M.: 2002, *Bayesian Methods for Nonlinear Classification and Regression*, Wiley Series in Probability and Statistics, West Sussex, UK.

Dietterich, T.: 2000a, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Machine learning* **40**(2), 139–157.

Dietterich, T. G.: 2000b, Ensemble methods in machine learning, *Proceedings of the* 1^{st} *International Workshop on Multiple Classifier Systems*, pp. 1–15.

Ding, C. and Dubchak, I.: 2001, Multi-class protein fold recognition using support vector machines and neural networks, *Bioinformatics* **17**(4), 349–358.

Doucet, A., Freitas, N. and Gordon, N.: 2000, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, Springer.

Dubchak, I., Muchnik, I., Holbrook, S. and Kim, S.: 1995, Prediction of protein folding class using global decsription of amino acid sequence., *Proc. Natl. Acad. Sci.* **92**, 8700–8704.

Duda, R. O., Hart, P. E. and Stork, D. G.: 2000, *Pattern Classification*, 2nd edn, Wiley-Interscience, New York.

Džeroski, S. and Ženko, B.: 2004, Is combining classifiers with stacking better than selecting the best one?, *Machine Learning* **54**(3), 255–273.

Edwards, A. W. F.: 1992, *Likelihood*, The Johns Hopkins University Press.

Efron, B. and Tibshirani, R. J.: 1993, *An Introduction to the Bootstrap*, number 57 in *Monographs on Statistics and Applied Probability*, Chapman and Hall.

Faul, A. and Tipping, M.: 2002, Analysis of sparse Bayesian learning, *Advances in Neural Information Processing Systems 14*, MIT Press, Cambridge, MA, pp. 383–389.

Filippone, M. and Sanguinetti, G.: 2009, Information theoretic novelty detection, *Pattern Recognition* . In Press. doi:10.1016/j.patcog.2009.07.002.

Fisher, R. A.: 1935, *The Design of Experiments*, Oliver and Boyd, Edinburgh.

Freund, Y. and Schapire, R. E.: 1996, Experiments with a new boosting algorithm, *International Conference in Machine Learning*, pp. 148–156.

Frosini, A., Gori, M. and Priami, P.: 1996, A neural network-based model for paper currency recognition and verification, *IEEE Transactions on Neural Networks* **7**(6), 1482–1490.

Fung, G., Dundar, M., Bi, J. and Rao, B.: 2004, A fast iterative algorithm for fisher discriminant using heterogeneous kernels, *Proceedings of the twenty-first international conference on Machine learning*, ACM New York, NY, USA, pp. 313–320.

Gardy, J. L., Laird, M. R., Chen, F., Rey, S., Walsh, C. J., Ester, M. and Brinkman, F. S. L.: 2005, Psortb v.2.0: Expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis, *Bioinformatics* **21**(5), 617–623.

Gauss, C. F.: 1809, Thoeria motus corporum coelestium, *Pattern Recognition* . Translation reprinted as Theory of the Motions of the Heavenly Bodies Moving about the Sun in Conic Sections, Dover, New York, 1963.

Gelman, A., carlin, J. B., Stern, H. S. and Rubin, D. B.: 2004, *Bayesian Data Analysis*, Chapman & Hall/CRC.

Geman, S. and Geman, D.: 1984, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741.

Geyer, C. J.: 1995, Practical Markov chain Monte Carlo, *Statistical Science* **7**, 473–511.

Ghahramani, Z. and Beal, M. J.: 2001, Graphical models and variational methods, *in* M. Opper and D. Saad (eds), *Advanced Mean Field Methods: Theory and Practise*, The MIT Press, pp. 161–177.

Girolami, M. A., Calderhead, B. and Chin, S. A.: 2009, Riemannian manifold Hamiltonian Monte Carlo, *Technical report*, Department of Computing Science, University of Glasgow.

Girolami, M. and Rogers, S.: 2005, Hierarchic Bayesian models for kernel learning, *Proceedings of the* 22^{nd} *International Conference on Machine Learning*, pp. 241–248.

Girolami, M. and Rogers, S.: 2006, Variational Bayesian multinomial probit regression with Gaussian process priors, *Neural Computation* **18**(8), 1790–1817.

Girolami, M. and Zhong, M.: 2007, Data integration for classification problems employing Gaussian process priors, *in* B. Schölkopf, J. Platt and T. Hoffman (eds), *Advances in Neural Information Processing Systems 19*, MIT Press, Cambridge, MA, pp. 465–472.

Gönen, M. and Alpaydin, E.: 2008, Localized multiple kernel learning, *Proceedings of the 25th international conference on Machine learning*, ACM New York, NY, USA, pp. 352–359.

Good, I. J.: 1983, *Good Thinking: The Foundations of Probability and Its Applications*, Univ. of Minn. Press, Minneapolis.

Gori, M. and Scarselli, F.: 1998, Are multilayer perceptrons adequate for pattern recognition and verification?, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(11), 1121–1132.

Graps, A.: 1995, An introduction to wavelets, *IEEE Computational Science and Engineering* **2**(2), 50–61.

Hand, D. J.: 1997, *Construction and Assessment of Classification Rules*, Wiley, Chichester.

Hand, D. J.: 2006, Classifier technology and the illusion of progress, *Statistical Science* **21**(1), 1–15.

Hastie, T., Tibshirani, R. and Friedman, J.: 2001, *The Elements of Statistical Learning*, Springer Series in Statistics, Springer.

Hastings, W.: 1970, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**(1), 97–109.

He, C., Damoulas, T. and Girolami, M. A.: 2009, Self-service terminals. USA Patent application, Serial number 11/899,381, http://www.faqs.org/patents/app/20090057395.

He, C., Girolami, M. and Ross, G.: 2004, Employing optimised combinations of one-class classifiers for automated currency validation, *Pattern Recognition* **37**(6), 1085–1096.

He, C. and Ross, G.: 2006, Banknote validation. NCR, USA Patent application, US20070140551.

Holmes, C. C. and Held, L.: 2006, Bayesian auxiliary variable models for binary and multinomial regression, *Bayesian Analysis* **1**, 145–168.

Jaakkola, T., Diekhans, M. and Haussler, D.: 1999, Using the fisher kernel method to detect remote protein homologies, *Proceedings of the Seventh International Conference on Inteligent Systems in Molecular Biology*, AAAI Press, pp. 149–158.

Jaakkola, T., Meila, M. and Jebara, T.: 1999, Maximum entropy discrimination, *Advances in Neural Information Processing Systems 12*.

Jaakkola, T. S.: 2001, Tutorial on variational approximation methods, *in* M. Opper and D. Saad (eds), *Advanced Mean Field Methods: Theory and Practise*, The MIT Press, pp. 129–159.

Jacobs, R., Jordan, M., Nowlan, S. and Hinton, G.: 1991, Adaptive mixtures of local experts, *Neural computation* **3**(1), 79–87.

Jaynes, E. T.: 2003, *Probability Theory: The Logic of Science*, Cambridge University Press.

Joachims, T., De, T., Cristianini, N., Uk, N. and Ac, R.: 2001, Composite kernels for hypertext categorisation, *In Proceedings of the International Conference on Machine Learning (ICML*.

Jordan, M. I. and Jacobs, R. A.: 1994, Hierarchical mixtures of experts and the EM algorithm, *Neural computation* **6**(2), 181–214.

Kass, R. E. and Raftery, A. E.: 1995, Bayes factors and model uncertainty, *Journal of the American Statistical Association* **90**, 773–795.

Kittler, J., Hatef, M., Duin, R. P. W. and Matas, J.: 1998, On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3), 226–239.

Kloft, M., Brefeld, U., Laskov, P. and Sonnenburg, S.: 2008, Non-sparse Multiple Kernel Learning, *NIPS Workshop on Kernel Learning*.

Kohonen, T.: 1990, The Self Organizing Map, *Proceedings of the IEEE* **78**(9), 1464–1480.

Kondor, R. and Jebara, T.: 2007, Gaussian and wishart hyperkernels, *Advances in Neural Information Processing Systems 19*, MIT; 2007.

Kosaka, T. and Omatu, S.: 1999, Classification of the Italian Liras using the LVQ method, *International Conference on Systems, Man and Cybernetics*, Vol. 6, pp. 845–850.

Kosaka, T. and Omatu, S.: 2000a, Classification of the Italian Lira using the LVQ method, *International Conference on Systems, Man and Cybernetics*, Vol. 4, pp. 2769–2774.

Kosaka, T. and Omatu, S.: 2000b, Classification of the Italian Liras using the LVQ method, *IJCNN '00, International Joint Conference on Neural Networks*, Vol. 3, pp. 145–148.

Kosaka, T., Omatu, S. and Fujinaka, T.: 2001, Bill classification by using the LVQ method, *International Conference on Systems, Man and Cybernetics*, Vol. 3, pp. 1430–1435.

Kosaka, T., Taketani, N., Omatu, S. and Ryo, K.: 1999, Discussion of reliability criterion for US Dollar classification by LVQ, *Workshop in Soft Computing Methods in Industrial Applications*, pp. 28–33.

Krogh, A. and Vedelsby, J.: 1995, Neural network ensembles, cross validation, and active learning, *in* G. Tesauro, D. S. Touretzky and T. K. Leen (eds), *Advances in Neural Information Processing Systems 7*, MIT Press, Cambridge, MA, pp. 231–238.

Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y. and Leslie, C.: 2004, Profile-based string kernels for remote homology detection and motif extraction, *Computational Systems Bioinformatics*, pp. 146–154.

Kullback, S. and Leibler, R. A.: 1951, On Information and Sufficiency, *The Annals of Mathematical Statistics* **22**(1), 79–86.

Kuncheva, L. I.: 2004, *Combining Pattern Classifiers. Methods and Algorithms*, Wiley-Interscience.

Kuncheva, L. and Whitaker, C.: 2003, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, *Machine Learning* **51**(2), 181–207.

Lanckriet, G., Cristianini, N., Bartlett, P., El Ghaoui, L. and Jordan, M.: 2002, Learning the Kernel Matrix with Semidefinite Programming, *In Proceedings of the 19th International Conference on Machine Learning (ICML.*

Lanckriet, G., Cristianini, N., Bartlett, P., El Ghaoui, L. and Jordan, M.: 2004, Learning the Kernel Matrix with Semidefinite Programming, *Journal of Machine Learning Research* **5**, 27–72.

Lawrence, N. D., Milo, M., Niranjan, M., Rashbass, P. and Soullier, S.: 2004, Reducing the variability in cDNA microarray image processing by Bayesian inference, *Bioinformatics* **20**(4), 518–526.

Lawrence, N., Seeger, M. and Herbrich, R.: 2003, Fast sparse Gaussian process methods: The informative vector machine, *Advances in Neural Information Processing Systems*, Vol. 15.

Lee, W., Verzakov, S. and Duin, R.: 2007, Kernel combination versus classifier combination, *Lecture Notes in Computer Science* **4472**, 22.

Leslie, C. S., Eskin, E., Cohen, A., Weston, J. and Noble, W. S.: 2004, Mismatch string kernels for discriminative protein classification, *Bioinformatics* **20**(4), 467–476.

Lewis, D. P., Jebara, T. and Noble, W. S.: 2006a, Nonstationary kernel combination, *23rd International Conference on Machine Learning*, pp. 553–560.

Lewis, D. P., Jebara, T. and Noble, W. S.: 2006b, Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure, *Bioinformatics* **22**(22), 2753–2760.

Liao, L. and Noble, W. S.: 2003, Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships, *Journal of Computational Biology* **6**(6), 857–868.

Lingner, T. and Meinicke, P.: 2004, Remote homology detection based on oligomer distances, *Bioinformatics* **22**(18), 2224–2231.

Liu, J.: 1994, The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem., *Journal of the American Statistical Association* **89**(427).

Lo Conte, L., Ailey, B., Hubbard, T., Brenner, S., Murzin, A. and Chothia, C.: 2000, Scop: a structural classification of proteins database, *Nucleic Acids Res.* **28**, 257–259.

MacKay, D.: 1992a, Information-based objective functions for active data selection, *Neural computation* **4**(4), 590–604.

MacKay, D.: 2003, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press.

MacKay, D. J. C.: 1992b, The evidence framework applied to classification networks, *Neural Computation* **4**(5), 698–714.

MacKay, D. J. C.: 2001, Local minima, symmetry breaking and model prunning in variational free energy minimization. Unpublished manuscript: http://www.cs.toronto.edu/ mackay/abstracts/minima.html.

MacKay, D. J. C.: 2004, Bayesian methods for backpropagation networks, *in* E. Domany, J. L. van Hemmen and K. Schulten (eds), *Models of Neural Networks III*, Springer, chapter 6, pp. 211–254.

Manocha, S. and Girolami, M. A.: 2007, An empirical analysis of the probabilistic K-Nearest Neighbour classifier, *Pattern Recognition Letters* .

Markou, M. and Singh, S.: 2003, Novelty detection: A review part1: Statistical approaches, *Signal Processing* **83**(12), 2481–2497.

McCullagh, P. and Nelder, J. A.: 1989, *Generalised Linear Models*, Chapman & Hall, London.

Meier, L., van de Geer, S. and Buhlmann, P.: 2008, The group lasso for logistic regression, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(1), 53–71.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E.: 1953, Equations of state calculations by fast computing machines, *Journal of Chemical Physics* **21**(6), 1087–1092.

Minka, T.: 2001, *A family of algorithms for approximate Bayesian inference*, PhD thesis, MIT.

Mitchell, M.: 1998, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, USA.

Moguerza, J., Munoz, A. and de Diego, I.: 2004, Improving support vector classification via the combination of multiple sources of information, *Lecture notes in computer science*, Springer, pp. 592–600.

Neal, R.: 2003, Markov chain sampling for non-linear state space models using embedded hidden Markov models, *Arxiv preprint math/0305039* .

Neal, R. M.: 1996, *Bayesian Learning for Neural Networks*, Springer Verlag.

Neal, R. M.: 1998, Regression and classification using Gaussian process priors, *Bayesian Statistics* **6**, 475–501.

Newman, D., Hettich, S., Blake, C. and Merz, C.: 1998, UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/.

Omatu, S., Fujinaka, T., Kosaka, T., Yanagimoto, H. and Yoshioka, M.: 2001, Italian Lira classification by LVQ, *IJCNN '01, International Joint Conference on Neural Networks*, Vol. 4, pp. 2947–2951.

Ong, C. S., Smola, A. J. and Williamson, R. C.: 2005, Learning the kernel with hyperkernels, *Journal of Machine Learning Research* **6**, 1043–1071.

Ong, C. and Smola, A.: 2003, Machine learning using hyperkernels, *Proceedings of the International Conference on Machine Learning*, pp. 568–575.

Ong, C., Smola, A. and Williamson, R.: 2003, Hyperkernels, *Advances in Neural Information Processing Systems 17*, Citeseer, pp. 495–504.

Opper, M. and Archambeau, C.: 2009, The variational Gaussian approximation revisited, *Neural Computation* **21**(3), 786–792.

Opper, M. and Winther, O.: 2001, From naive Mean Field theory to the TAP equations, *in* M. Opper and D. Saad (eds), *Advanced Mean Field Methods: Theory and Practise*, The MIT Press, pp. 129–159.

Parisi, G.: 1988, *Statistical Field Theory*, Addison Wesley.

Plamondon, R. and Srihari, S.: 2000, On-line and off-line handwriting recognition: A comprehensive survey, *IEEE Transactions On Pattern Analysis and Machine Intelligence* **22**(1), 63–84.

Psorakis, Y., Damoulas, T. and Girolami, M. A.: 2010, Multiclass relevance vector machines: An evaluation of sparsity and accuracy, *IEEE Transactions on Neural Networks* **0**(0), 00–00. Submitted.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A. and Lawrence, N. D.: 2009, *Dataset Shift in Machine Learning*, Neural Information Processing series, MIT Press, Cambridge.

Rakotomamonjy, A., Bach, F., Canu, S. and Grandvalet, Y.: 2007, More efficiency in multiple kernel learning, *Proceedings of the 24th international conference on Machine learning*, ACM New York, NY, USA, pp. 775–782.

Rao, N. S. V.: 2001, On fusers that perform better than best sensor, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(8), 904–909.

Rao, N. S. V.: 2004, A generic sensor fucion problem: Classification and function estimation, *Multiple Classifier Systems*, pp. 16–30.

Rasmussen, C. E. and Williams, C. K. I.: 2006, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Massachusetts, USA.

Raval, A., Ghahramani, Z. and Wild, D. L.: 2002, A Bayesian network model for protein fold and remote homologue recognition, *Bioinformatics* **18**, 788–801.

Ripley, B. D.: 1996, *Pattern Recognition and Neural Networks*, Cambridge University Press, UK.

Roth, V.: 2004, The generalized LASSO, *IEEE Transactions on Neural Networks* **15**(1), 16–28.

Saigo, H., Vert, J.-P., Ueda, N. and Akutsu, T.: 2004, Protein homology detection using string alignment kernels, *Bioinformatics* **20**(11), 1682–1689.

Schapire, R. E.: 2003, The boosting approach to machine learning: An overview, *Lecture Notes in Statistics* pp. 149–172.

Schmolck, A. and Everson, R.: 2007, Smooth relevance vector machine: a smoothness prior extension of the RVM, *Machine Learning* **68**(2), 107–135.

Schölkopf, B. and Smola, A.: 2002, *Learning with Kernels*, The MIT Press, Cambridge, Massachusetts, USA.

Shawe-Taylor, J. and Cristianini, N.: 2004, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, England, UK.

Shen, H.-B. and Chou, K.-C.: 2006, Ensemble classifier for protein fold pattern recognition, *Bioinformatics* **22**(14), 1717–1722.

Shi, M., Fujisawa, Y., Wakabayashi, T. and Kimura, F.: 2002, Handwritten numeral recognition using gradient and curvature of gray scale image, *Pattern Recognition* **35**(10), 2051–2059.

Sonnenburg, S., Ratsch, G. and Schafer, C.: 2006, A general and efficient multiple kernel learning algorithm, *Advances in Neural Information Processing Systems 18: proceedings of the 2005 conference*, MIT.

Sonnenburg, S., Rätsch, G., Schäfer, C. and Schölkopf, B.: 2006, Large scale multiple kernel learning, *Journal of Machine Learning Research* **1**, 1–18.

Sun, J., Zhang, B., Chen, Z., Lu, Y., Shi, C. and Ma, W.: 2004, GE-CKO: A method to optimize composite kernels for Web page classification, *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society Washington, DC, USA, pp. 299–305.

Takeda, F. and Nishikage, T.: 2000, Multiple kinds of paper currency recognition using neural network and application for Euro currency, *IJCNN '00, International Joint Conference on Neural Networks*, Vol. 2, pp. 143–147.

Takeda, F., Nishikage, T. and Matsumoto, Y.: 1998, Characteristics extraction of paper currency using symmetrical masks optimized by GA and neuro-recognition of multi-national paper currency, *IJCNN '98, International Joint Conference on Neural Networks*, Vol. 1, pp. 634–639.

Takeda, F. and Omatu, S.: 1995a, High speed paper currency recognition by neural networks, *IEEE Transactions on Neural Networks*, Vol. 6, pp. 73–77.

Takeda, F. and Omatu, S.: 1995b, A neuro-paper currency recognition method using optimized masks by genetic algorithm, *ICSMC '95, International Conference on Systems, Man and Cybernetics*, Vol. 5, pp. 4367–4371.

Takeda, F., Omatu, S. and Onami, S.: 1993, Recognition system of US Dollars using a neural network with random masks, *IJCNN '93, International Joint Conference on Neural Networks*, Vol. 2, pp. 2033–2036.

Takeda, F., Omatu, S., Onami, S., Kadono, T. and Terada, K.: 1994, A paper currency recognition method by a small size neural network with optimized masks by GA, *ICNN '94, International Conference on Neural Networks*, Vol. 7, pp. 4243–4246.

Tanaka, M., Takeda, F., Ohkouchi, K. and Michiyuki, Y.: 1998, Recognition of paper currencies by hybrid neural network, *IJCNN '98, International Joint Conference on Neural Networks*, Vol. 3, pp. 1748–1753.

Tanner, M. A. and Wong, W. H.: 1987, The calculation of posterior distributions by data augmentation, *Journal of the American Statistical Association* **82**, 528–550.

Tappert, C. C., Suen, C. Y. and Wakahara, T.: 1990, The state of the art in on-line handwriting recognition, *IEEE Transactions On Pattern Analysis and Machine Intelligence* **12**(8), 787–808.

Tax, D. M. J. and Duin, R. P. W.: 2001, Combining one-class classifiers, *Multiple Classifier Systems, Proceedings Second International Workshop MCS 2001 (Cambridge, UK, July), Lecture Notes in Computer Science, Springer Verlag*, Vol. 2096, pp. 299–308.

Tax, D. M. J., van Breukelen, M., Duin, R. P. W. and Kittler, J.: 2000, Combining multiple classifiers by averaging or by multiplying?, *Pattern Recognition* **33**, 1475–1485.

Teranishi, M., Matsui, T., Omatu, S. and Kosaka, T.: 2005, Neuro-classification of fatigued bill based on tensional acoustic signal, *SMCIA '05, Mid-Summer Workshop on Soft Computing in Industrial Applications*, pp. 173–177.

Teranishi, M., Omatu, S. and Kosaka, T.: 1999, New and used bills classification for cepstrum patterns, *IJCNN '99, International Joint Conference on Neural Networks*, Vol. 6, pp. 3978–3980.

Teranishi, M., Omatu, S. and Kosaka, T.: 2000, Classification of bill fatigue levels by feature-selected acoustic energy pattern using competitive neural network, *IJCNN '00, International Joint Conference on Neural Networks*, Vol. 6, pp. 249–252.

Teranishi, M., Omatu, S. and Kosaka, T.: 2002, Neuro-classification of bill fatigue levels based on acoustic wavelet components, *ICANN '02, International Conference on Artificial Neural Networks*, pp. 1074–1079.

Thouless, D. J., Anderson, P. W. and Palmer, R. G.: 1977, Solution of a solvable model of a spin glass, *Phil. Mag* **35**, 593.

Tibshirani, R.: 1996, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society B* **58**(1), 267–288.

Tipping, M. E.: 1999, The relevance vector machine, *Advances in Neural Information Processing Systems 12*, pp. 652–658.

Tipping, M. E.: 2001, Sparse Bayesian learning and the relevance vector machine, *Journal of Machine Learning Research* **1**, 211–244.

Tipping, M. E.: 2004, Bayesian inference: An introduction to principles and practise in machine learning, *in* O. Bousquet, U. von Luxburg and G. Rätsch (eds), *Advanced Lectures on Machine Learning*, LNAI 3176 Springer, pp. 41–62.

Tipping, M. and Faul, A.: 2003, Fast marginal likelihood maximisation for sparse Bayesian models, *Proceedings of 9th AISTATS Workshop*, pp. 3–6.

Trier, O. D., Jain, A. K. and Taxt, T.: 1996, Feature extraction methods for character recognition - a survey, *Pattern Recognition* **29**(4), 641–662.

Tzikas, D., Likas, A. and Galatsanos, N.: 2008, Incremental relevance vector machine with kernel learning, *Hellenic Conference on Artificial Intelligence*, pp. 301–312.

Tzikas, D., Likas, A. and Galatsanos, N.: 2009, Sparse Bayesian modeling with adaptive kernel learning, *IEEE Transaction on Neural Networks* . to appear.

Ueda, N. and Nakano, R.: 1996, Generalization error of ensemble estimators, *IEEE International Conference on Neural Networks*.

Vapnik, V. and Chervonenkis, A.: 1964, A note on one class of perceptrons., *Automation and Remote Control,* **25**.

Vapnik, V. N.: 1995, *The Nature of Statistical Learning Theory*, Springer.

Vapnik, V. N.: 1998, *Statistical Learning Theory*, John Willey & Sons.

Varma, M. and Babu, B.: 2009, More generality in efficient multiple kernel learning, *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM New York, NY, USA, pp. 1065–1072.

Vyshemirsky, V.: 2007, *Probabilistic Reasoning and Inference for Systems Biology*, PhD thesis, Department of Computing SCience, University of Glasgow.

Vyshemirsky, V. and Girolami, M. A.: 2008, Bayesian ranking of biochemical system models, *Bioinformatics* **24**(6), 833–839.

Wang, H. and Leng, C.: 2007, Unified LASSO estimation by least squares approximation, *Journal of the American Statistical Association* **102**(479), 1039–1048.

Widrow, B., Winter, R. G. and Baxter, R. A.: 1988, Layered neural nets for pattern recognition, *IEEE Transactions on Acoustics, Speech, and Signal Processing* **36**(7), 1109–1118.

Wolpert, D. H.: 1992, Stacked generalization, *Neural networks* **5**(2), 241–259.

Ye, J., Ji, S. and Chen, J.: 2008, Multi-class discriminant kernel learning via convex programming, *JMLR* **9**, 719–758.

Ying, Y., Campbel, C., Damoulas, T. and Girolami, M. A.: 2009, Class prediction from disparate biological data sources using a simple multi-class multi-kernel algorithm, *Pattern Recognition in Bioinformatics (PRIB '09)*. Accepted.

Zhang, Z., Yeung, D. and Kwok, J.: 2004, Bayesian inference for transductive learning of kernel matrix using the Tanner-Wong data augmentation algorithm, *Proceedings of the twenty-first international conference on Machine learning*, ACM New York, NY, USA, pp. 935–942.

Zien, A. and Ong, C. S.: 2007, Multiclass multiple kernel learning, *ICML '07: Proceedings of the 24th international conference on Machine learning*, ACM, New York, NY, USA, pp. 1191–1198.

Zou, H.: 2006, The adaptive LASSO and its oracle properties, *Journal of the American Statistical Association* **101**(476), 1418–1429.