Submitted by Alexander Schmolck to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science,

May 2008

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

I certify that all material in this thesis which is not my own work has been identified and that no material is included for which a degree has previously been conferred upon me.

______________________________________________
Alexander Schmolck
Abstract

Regression tasks belong to the set of core problems faced in statistics and machine learning and promising approaches can often be generalized to also deal with classification, interpolation or denoising problems. Whereas the most widely used classical statistical techniques place severe a priori constraints on the type of function that can be approximated (e.g. only lines, in the case of linear regression), the successes of sparse kernel learners, such as the SVM (support vector machine) demonstrate that good results may be obtained in a quite general framework by enforcing sparsity. Similarly, even very simple sparsity-based denoising techniques, such as classical wavelet shrinkage, can produce surprisingly good results on a wide variety of different signals, because, unlike noise, most signals of practical interest share vital characteristics (such as smoothness, or the ability to be well approximated by piece-wise linear polynomials of a low order) that allow a sparse representation in wavelet space. On the other hand results obtained from SVMs (and classical wavelet-shrinkage) suffer from a certain lack of interpretability, since one cannot straightforwardly attach probabilities to them. By contrast regression, and even more importantly classification, in a Bayesian context always entails a probabilistic measure of confidence in the results, which, provided the model assumptions are reasonably accurate, forms a basis for principled decision-making. The relevance vector machine (RVM) combines these strengths by explicitly encoding the criterion of model sparsity as a (Bayesian) prior over the model weights and offers a single, unified paradigm to efficiently deal with regression as well as classification tasks. However the lack of an explicit prior structure over the weight variances means that the degree of sparsity is to a large extent controlled by the choice of kernel (and kernel parameters). This can lead to severe overfitting or oversmoothing – possibly even both at the same time (e.g. for the multiscale Doppler data). This thesis details an efficient scheme to control sparsity in Bayesian regression by incorporating a flexible noise-dependent smoothness prior into the RVM. The resultant smooth RVM (sRVM) encompasses the original RVM as a special case, but empirical results with a variety of popular data sets show that it can surpass RVM performance in terms of goodness of fit and achieved sparsity as well as computational performance in many cases. As the smoothness prior effectively makes it possible to use (highly efficient) wavelet kernels in an RVM setting this work also unveils a strong connection between Bayesian wavelet shrinkage and RVM regression and effectively further extends the applicability of the RVM to denoising tasks for up to millions of datapoints. We further discuss its applicability to classification tasks.
# Contents

1 Introduction .................................................. 14
   1.1 What, where, why ........................................ 14
   1.2 The sRVM for denoising: a 10 000 feet overview ........ 15
   1.3 The broader perspective: sRVM regression and classification .... 16
      1.3.1 Kernel space vs feature space ...................... 17
      1.3.2 What makes the (s)RVM attractive? ................. 17
   1.4 Research Overview .................................... 18
   1.5 Novel Contributions .................................. 18

2 Background .................................................. 20
   2.1 The Mechanics of Regression ............................. 20
      2.1.1 Regularization .................................... 22
      2.1.2 Regularization from a probabilistic perspective .... 27
      2.1.3 The LASSO ....................................... 30
      2.1.4 Wavelet Shrinkage .................................. 31
      2.1.5 Beyond wavelets: pursuit methods .................. 33
      2.1.6 Shortcomings of wavelets ........................... 34
   2.2 Classification and Kernels ............................... 34
      2.2.1 Gaussian processes ................................. 36
      2.2.2 Support Vector Machines ............................ 37

I Regression and the smooth Relevance Vector Machine ........ 39
   2.3 Shortcomings of the classical RVM ........................ 41
   2.4 Amending the RVM; outlook and overview ................. 43
   2.5 The smoothness prior .................................... 45
      2.5.1 Finding a suitable prior over α or wavelet shrinkage to the rescue .... 47
   2.6 An Implementation based on Tipping and Faul’s fRVM .......... 48
      2.6.1 Maximizing the marginal likelihood ................ 48
      2.6.2 Overview of MAP implementation properties and strategy .... 50
      2.6.3 Noise reestimation .................................. 52
      2.6.4 The algorithm ..................................... 53
      2.6.5 Results ............................................. 54
      2.6.6 Exploring local minima ............................ 63
      2.6.7 A not quite so detailed investigation of BlocksSinc .......... 67
   2.7 An attempt at a Markov Chain Monte Carlo implementation .... 69
      2.7.1 An introduction to MCMC ........................... 70
      2.7.2 The details of the MCMC scheme ................. 72
      2.7.3 The complete MCMC algorithm ..................... 74
II Classification and the smooth Relevance Vector Machine 80
2.9 Extending the RVM for regression for classification 81
2.10 IRLS and the Laplace approximation 81
2.10.1 IRLS in more detail 82
2.11 The derivative of the trace of $S$ in the general case 83

3 Classification Results and Summary 85
3.0.1 Something completely different: TP/FP-rate Evolutionary Optimization of $\ell_q$ for penalized logistic kernel regressors 87

4 Overall Conclusion 90

A Appendix 97
A.1 Kernel functions 97
A.2 Data sets used 98
A.3 Scale invariance for constant SNR 99
A.4 Uniqueness of local maximum 99
A.4.1 Asymptote from below: 99
A.4.2 Asymptote from above: 100
A.5 Saddle-points of $\hat{L}$ 100
A.6 Approaching the Jeffreys’ prior 100
A.7 Efficiently calculating the full likelihood and posterior 101
A.8 Computing $\frac{d}{d\alpha}(\text{tr } S)$ for $\Phi^T B \Phi \neq \sigma I$ 102

B Some useful Identities and Results 105
B.1 Matrix trace, inverse and determinant identities 105

C Some tedious calculations 106

D Notation and Glossary 107
List of Figures

1.1 10 000 feet overview of the sRVM. The smoothness prior $\pi(\alpha | c) \sim \exp(-c\text{DF})$ (2.59) is what sets the sRVM apart from the RVM. It penalizes complex models (models with high degrees of freedom DF, where DF is defined in (2.57)) and can be adjusted in its severity by the user-determined hyperparameter $c$. Particular values for $c$ can be related to classical model-choice criteria (see section 2.5.1).

2.1 Two ways to visualize the least squares fit $\hat{y}$ for the inputs $x = [1 3 4]$ and the targets $t = [-2 -2]$. Left panel: The familiar form in which the input/target pairs $(x_i, t_i)$ (represented as blue dots) are taken to be points in 2-dimensional space and $\hat{y}_i$ is chosen so that the pairs $(x_i, \hat{y}_i)$ lie on the line (shown in magenta) that minimizes the square of the vertical offset $\epsilon_i$ (shown in red) between $(x_i, t_i)$ and the corresponding point on the line at $x_i$. Right panel: The equivalent, but less familiar, linear algebra interpretation. Here the inputs $x$ are represented by a single vector in $(N = 3)$-dimensional space, as are the targets $t$, the estimate $\hat{y}$ and the noise estimate $\hat{\epsilon}$. The least squares estimate $\hat{y}$ is the orthogonal projection of $t$ into the $(D = 2)$-dimensional subspace (shown in green) spanned by $x$ and the bias vector $1 = [1 1 1]^T$; the noise estimate $\hat{\epsilon}$ is the distance between $\hat{y}$ and $t$.

2.2 (a) Sinc regression with polynomials of different degree $M$. Overfitting starts to occur well before $M = N = 20$. On the other hand $M = 3$ clearly lacks the flexibility required to fit the data and oversmoothes. $M = 9$ gives the best fit. (b) Same as pannel on the left with $M = 20$, but increasing level of regularization from top ($\alpha = 0$) to bottom ($\alpha = 2$). The middle panel ($\alpha = 0.23$) gives the lowest error.

2.3 Haar wavelet vs sinusoid. The Haar mother wavelet is given by the simple function $\phi_H(x) = -1$ if $0 \leq x < 1/2$ else 1 if $1/2 \leq x < 1$ else 0. Whilst a sinusoid is continuous and periodic, the Haar wavelet has finite support and is discontinuous. It is however possible to construct an orthogonal basis from shifted and scaled copies of either of these two functions, $\phi_H(x)$ or $\cos(x)$. Because the resulting bases are orthogonal, a signal expressed in either basis can be transferred into the other basis by means of a simple rotation (effected by pre-multiplying the signal with a matrix that spans the respective basis).
2.4 The effect of different $\ell_q$-norms over $w$ on sparsity, visualized with isocontour plots for $w = [w_1 \ldots w_2]$ and $q = 2^{-\infty}, 2^{-2}, 2^{-1}, 2^1, 2^2, 2^\infty$. Three values of $q$ are of particular interest: $\ell_2$ is the familiar Euclidian norm (and can be seen to be the only rotationally invariant norm), $\ell_1$ is also known as Manhattan distance (and from $\ell_q(w) = \|w\|_q = \sqrt[q]{\sum_i|w_i|^q}$ it is easy to see that it is the only norm that is invariant under reallocation of total weight mass $m_w = \sum_i |w_i|$). Finally $\ell_0$, is a pseudo-norm that corresponds to the number of selected basis functions $\phi_i$ (i.e. nonzero $w_i$). $q = 1$ is a natural middle point – smaller values of $q$ favour concentrating the absolute total weight-mass in a single $w_i$, thus encouraging sparsity whilst larger values favor spreading it equally over all $w_i$, thus encouraging “fatness” or inverse-sparsity. Note that $\ell_1$ takes a special place as the lowest (and only integer-valued-$q$) $\ell$-norm that is still convex (which can considerably ease optimization) but not obviously fattening. Actually, although $\ell_1$ is invariant w.r.t. to the allocation of total weight mass, it still turns out to be sparsening because when used as a weight penalty term in (2.16) it will generally have the effect of setting several $\hat{w}_i$ to exactly 0 (see text for an explanation).

2.5 The zero-mean Laplace distribution (solid red) assigns more probability mass to the region immediately around zero and to the tails than the zero mean Gaussian (solid black), thus favouring sparser results.

2.6 Achieving linear separability through mapping to a higher dimensional feature space. By mapping the one dimensional data set $\{x_n\}_N^{N}$ into two dimensions; $\{\phi x_n\}_N^{N}$ with an appropriate basis function (here $\phi(x) = [x \ x^2]^T$), it becomes linearly separable.

2.7 Classical RVM. The effect of dictionary choice on the smoothness of the regression result (Sinc data left, Bumps data right) when there is no prior over $\alpha$. Choosing a flexible symmlet-wavelet dictionary (top row) results in drastic overfitting for the Sinc data set (top left; $N=128$, SNR=2.0). To obtain the appropriate level of smoothing for the Sinc data one has to resort to a different dictionary type, such as lspline (bottom left). However an lspline dictionary cannot resolve the Bumps data (bottom right; $N=128$, SNR=7.0) at all.

2.8 Basis functions 1, 10, 23, 230 from $N = 512$ symmlet (left) and lspline dictionaries (right). Whereas the symmlet dictionary contains components at all frequencies, the lspline dictionary only offers low-frequency components. This explains why the classical RVM’s relatively weak sparseness enforcement suffices for lspline kernels, but not symmlets.

2.9 sRVM. The smoothness prior means that enforcing sparsity is no longer mostly relegated to the choice of kernel. A symmlet kernel (top row) no longer results in drastic overfitting for the Sinc data set (on the left). The bottom row shows that the smoothness prior typically has no adverse effect when smoothing is already mandated by the kernel. The data sets are identical to Figure 2.7.
2.10 The smoothness prior in logspace: \( \log \pi(\alpha_i \mid \sigma^2) = -c/(1 + \sigma^2 \alpha_i) \). As it increases monotonically with \( \alpha_i \) the prior encourages sparsity (associated with large/infinite values of \( \alpha_i \)). The noise-dependence of the prior is illustrated by showing plots for different values of \( \sigma^2(1/4, 1/2, 1; c = 1 \text{ for all figures}) \); higher noise is associated with more severe sparsity enforcement. This is as it should be, since (the unknown) noise \( e \) and (the unknown) underlying signal \( y \) effectively offer “competing explanations” for the observed data \( t \) and the smoothness prior helps to mediate the trade-off (Consider the two extremes: if there is no noise, no smoothing of \( t \) should take place in the computation of the posterior estimate \( \hat{y} \), which in that case should just be identical to \( t \); conversely if the observed data is all noise, the result should ideally be completely sparse, i.e. a flat line).

2.11 Log posteriors \( \hat{\ell}(\alpha_i) \) (solid), log likelihoods \( \ell(\alpha_i) \) (dashed), and log priors \( -c(1 + \sigma^2 \alpha_i)^{-1} \) (dotted) plotted versus log \( \alpha_i \) for the four possible scenarios of how the addition of the smoothness prior affects the pre-existing mode of \( \ell(\alpha_i) \). Firstly since the (s)RVM is based on MAP maximization and, in particular, in the case of the fast RVM implementation (Tipping and Faul, 2003), on the maximization of \( \hat{\ell}(\alpha_i) \) wrt \( \alpha_i \), one would hope to see that the introduction of the smoothness prior does not introduce additional (finite) modes (which could hamper optimization or even introduce ambiguities). Secondly, since the aim of the smoothness prior is to more stringently enforce sparsity, and since sparsity is associated with large (or more strictly, infinite) \( \alpha_i \), one would further hope to find that the mode of \( \hat{\ell}(\alpha_i) \) (viz the mode of the sparsity-prior-enhanced version of \( \ell(\alpha_i) \)) is always located on the right of \( \ell(\alpha_i) \). This is indeed the case, and the possible scenarios are illustrated in the above four panels: \textit{Top-left:} Prior nulls maximum in posterior: the addition of the smoothness prior removes the finite maximum, so that the optimal \( \alpha_i \) is now \( \infty \). \textit{Top-right:} Single turning point with \( \hat{\alpha}_i \) finite: the addition of the smoothness prior still gives a finite mode, but the regression result will still a little bit smoother since the mode is shifted to the right. \textit{Bottom-left and bottom-right:} although two turning points can occur in the posterior, at most one mode will be finite: the single finite mode can either be larger than \( 0 = \lim_{\alpha_i \to \infty} \hat{\ell}(\alpha_i) \) (bottom-left) or smaller (bottom-right).
2.12 The shortcomings of smoothness control via kernel choice (RVM, top, middle) in comparison to smoothness control via prior choice (sRVM, bottom), illustrated by denoising data with multiscale resolution (identical data set in all panels: Doppler data, N = 1024, SNR = 7.0). The Doppler data is characteristically multiscale: it goes from very high frequency on the very left side of the picture to mid and low frequencies towards the right end. In order to fully appreciate the result of a particular kernel/prior combination, the effects on both the high frequency part (the first 128 of the N data points – shown magnified in the inset at the top of each figure) and the mid/low frequency part (the remainder) should be studied: ideally both should show a good fit. The three panels in the top and middle rows were created with the classical RVM (i.e. None prior) and differ by kernel choice alone (Gaussian kernel with kernel width $r = 0.5$ in the top panel, and $r = 0.05$ in the left middle panel; symmlet in the right middle panel). They all show a bad fit: the top panel shows oversmoothing (visible in the inset) whereas the two middle panels show overfitting (visible particularly towards the right of each panel). The problem is that smoothness control via kernel width or type acts globally, whereas only part of the signal is respectively fine scale/large scale. Thus even though overfitting already starts to become apparent in the top panel, the fine scale information on the left side is still severely oversmoothed (as a glance at the inset reveals). Decreasing kernel width (middle) to improve resolution sufficiently to fit the fine scale details on the left is seen to be tied to drastic overfitting in the right part of the plot. By contrast, the bottom panel was created with the sRVM (BIC prior, symmlet) and shows that a smoothness prior in combination with a multi-resolution design matrix achieves an adaptive level of smoothing: there is neither overfitting in the inset nor oversmoothing towards the right of the figure.

2.13 Shrinkage plots. Plotting the least squares estimate of the weights $\mu_{LSQ}$ against the posterior weight estimates obtained with None (left) and BIC (right) priors clearly shows that the BIC smoothness prior is much more effective at weeding out small, irrelevant components by setting them to 0. These plots correspond to the middle right and bottom panel, respectively, of Figure 2.12 and are clipped to $|\mu_m| \leq 0.5$ (the few larger components are essentially unaffected by shrinkage and thus lie on the diagonal).

2.14 The sRVM (here with RIC prior) makes it possible to obtain very good results by using overcomplete dictionaries. The example data (“BlocksSinc”) is constructed by concatenating two signals with very different characteristics: Blocks and Sinc and adding Gaussian noise (SNR: 7.0). Whilst no standard kernel will give ideal results for this combination, thin-plate splines (tpsplines) are well suited for smooth, continuous curves such as Sinc (a), whilst the step-like nature of Haar wavelets makes them the ideal candidate for the Blocks subset (b). However, thanks to the smoothness prior, the sRVM can do a remarkably good job at automatically picking the appropriate components for each part of the signal from an overcomplete dictionary obtained by concatenating both these two sets of basis functions together (also see Figure 2.6.5).
2.15 sRVM regression as ad hoc blind-source separation (same toy BlocksSinc data set and overcomplete haar+tpspline dictionary as the previous figure, 2.14; but this time the contributions of each dictionary part to the final result are shown as separate, differently coloured curves (Haar, tpspline)). Left panel: The RVM’s posterior mean prediction $\hat{y}$ for the above setup: although the RVM achieves only a limited degree of separation (the Blocks part of the left half of the signal is largely, but not exclusively, reconstructed out of Haar basis functions and the Sinc part on the right is mostly made up out of tpspline components). It also fits a significant part of the noise (mostly with the Haar part of the overcomplete dictionary, in both halves of the signal). Right panel: The sRVM achieves near-perfect denoising and separation of the two signal components (Blocks and Sinc) into the respective dictionary parts (Haar and tpspline) in its posterior mean prediction $\hat{y}$ (note that this is just a different visualization of the result displayed in the bottom panel of Figure 2.14).

2.16 Top panel: Raw sleep EEG data (only the first half of $N = 4096$ data points is displayed for clarity) with human expert marked spindle regions (delimited by red lines). Note that whilst it is possible to make out some distinguishing characteristics in the spindle regions (more high frequency components and an increased amplitude, compared to the surrounding data), it is difficult to correctly identify the relevant regions by eye for a non-expert and neither criterion seems sufficient on its own. Bottom panel: The post-processed RIC sRVM denoised version of the same data as above. After denoising the data with the RVM (RIC prior), we removed all components outside the “second blob” in DCT- (viz. weight-) space, located around abscissa coordinates 880-1150 in Figure 2.17. This was done simply by setting the $w_i$ in this range to zero and pre-multiplying $\Phi$ to yield a modified posterior estimate $\hat{y}_{\text{pruned}}$. Subsequently we took the absolute value of the result in data-space, but this was done just to make it slightly easier to see how well the red lines of expert-determined spindle regions align with the peaks. We can see that the spindle regions are much easier to visually discern in the bottom graph – in particular for the displayed excerpt we could now detect the existence of a region by amplitude alone (in this case an ordinate threshold of 0.65 would work).

2.17 Unprocessed EEG data (top) and the sRVM denoised EEG data in DCT (discrete cosine transform) space (bottom). Low frequency components are on the right and high frequency components on the right. Note that whilst although even before denoising one can easily discern two, three distinct “blobs” as well as a lonely spike (in the right half); no obvious boundary between them is apparent till after denoising. Both visual inspection of the original signal and further experimentation suggests that the spindle waveforms are mostly located in the frequencies occupied by the middle blob (marked in the lower subplot). It should maybe also be explicitly stated that since we are using a $\Phi$ that implements the DCT, DCT-space is simply equivalent to weight space. In other words the second plot simply shows the posterior mean weight estimate $\mu$ and the first plot is identical to the least-squares estimate of the vector $m$ that solves $t = \Phi m$.

2.18 EEG spindle data: comparison between including only the second big blob in DCT space (centered around abscissa coordinate 1000 in the bottom of Figure 2.17) in the reconstruction (bottom, same as Figure 2.16 bottom) and excluding only the first blob in DCT space (centered at around 100 in the bottom of 2.17) from reconstruction. Since both plots looks rather similar, it appears the most salient aspects of the spindle regions (again marked by red bars) are indeed to be found in the second blob.

2.19 The 1000 BIC symmlet results.
2.20 The 1000 BIC 3.0 lspline results corresponding to Figure 2.19........64
2.21 A comprehensive inspection of the Sinc 3.0 lspline results. The results are sorted in by \( \hat{L}_{BIC}(\alpha) \) for and duplicates (as determined by \( S \)) are removed; all graphs are divided horizontally into 4 zones of interest. The first graph shows the number of steps that a run took to converge and the number of selected components. The second graph shows the posterior likelihood (BIC) and the MSE of the predictions \( \hat{y} \) to the underlying data \( t \) as well as the true signal \( y \). Note that the left and right abscissas have been truncated to remove the very low BIC results in partition 1 (which lie at around -1500) and the very large MSEs (around 0.1). The third and final graph displays the active components after each run, the development of what ends up as component 4 (marked as \( C_4 \)) is of special interest............65
2.22 The 1000 3.0 lspline results for a None prior.........................66
2.23 1000 runs of BlocksSinc (SNR=7.0, N=256, M=512) with a BIC prior and mixed Haar/lspline dictionary, left the \( \hat{y} \) values, right the corresponding \( \alpha \). Note that in contrast to the spline experiments, here different \( \alpha \) values really result in noticeably different predictions \( \hat{y} \)..................67
2.24 Rejection sampling: to sample from awkward distribution \( p \) use a proposal distribution \( q \) and a function \( p^* \) such that \( \forall x : p^*(x) \propto p(x) \land p^*(\alpha(x)) < q(x) \). Then generate a proposal \( x' \sim q \) and a uniform sample \( u \sim U[0, q(x')] \). Accept \( x' \) as a sample from \( p \) if \( u \) falls in the blue area of the graph (\( u \leq p^*(x') \)), else repeat. Informally, this procedure works because the likelihood of a sample lying in an infinitesimal small region around \( x' \), namely \( q(x') \) multiplied by its probability of being accepted, namely \( p^*(x')/q(x') \), simply yields \( p^*(x') \propto p(x') \) which shows that (an accepted) \( x' \) is an unbiased sample from \( p \)..................70
2.25 Left panel: Horizontally concatenated MH samples for \( \alpha \) (color-coded in log-scale) with different proposal standard deviations \( s \) (the values are 50, 10 and 2) for lspline 3.0 Sinc data (\( N=M=64, SNR=5 \)). Even with a quite high rejection rate (~0.05 for propStd 50), the sampler gets stuck in “furrows” and moreover these furrows are pretty arbitrary as is evinced by the fact that all three sample runs fell into quite different ones. Right panel: The corresponding averaged mean posterior predictions \( \hat{y} \)..................75
2.26 Different MCMC sampling schemes tested on the Sinc data using a large number of samples each time. Left panel: A long run of Reversible Jump Metropolis Hastings (RJMH) samples of \( \alpha \) (again color-coded on a log-scale) and the resultant mean posterior prediction \( \hat{y} \) for Sinc. The mixing is fairly poor and the model suddenly degenerates to a drastic underfit with only two active components in the right half of the figure. Middle panel: A long run of MH-only samples, again showing poor mixing. Right panel: RJ only with moves disabled – active \( \alpha \) are always set to \( 10^{-5} \). This approach gave us the best mixing behavior we observed, but somewhat defeats the point of our complex probabilistic model. Furthermore, even the RJ-only approach did not always mix well (see Figure 2.7.4, right panel)........75
2.27 BlocksSinc sampling analogous to Figure 2.7.4, using fewer samples and a M-reduced “pseudo-overcomplete” dictionary for reasons of computational expense. Left panel: The full RJMH sampling fails completely. Middle panel: MH-only sampling produces an acceptable mean prediction \( \hat{y} \) but gets trapped in a single local maxima. Right panel: RJ only – active \( \alpha \) are always set to \( 10^{-5} \), but this time even that approach fails to show any mixing..................76
2.28 Log likelihoods, $\ell(\alpha_i)$ (dashed), and log gamma prior, $\log G(\alpha_i \mid a, b)$ (dotted), and finally the log posteriors for a Gamma prior, $\hat{\ell}_G(\alpha_i)$ (solid), plotted versus $\log \alpha_i$, showing that in the case of a Gamma prior, with $q_i = 1$, $s_i = 2$, $a = 1$ and $b = 2$ the MAP $\alpha_i$ is less sparse than the maximum likelihood solution – something that will never occur with the smoothness prior and the corresponding posterior $\ell(\alpha_i)$.

3.1 Gausssnake fake data example. Gausssnake is a simple matlab script to create “wiggly” toy data from a mixture of Gaussians with known decision boundaries and class probabilities specifically to explore the effects of the smoothness prior for classification data. User adjustable parameters govern the separation of the two classes, and the number and shape of the Gaussians. The black line is the decision boundary, the exemplars for classes A and B are respectively displayed in blue and red, and the contour colors indicate the class probabilities.

3.2 Some gausssnake results with None prior (left) and with RIC prior (right).

3.3 Evolutionary multi-objective optimization on Ripley’s synthetic data, using the Lasso penalty.

3.4 Evolutionary multi-objective optimization on Ripley’s synthetic data, using the $L_0+$ penalty.
List of Tables

2.1 Empirical comparisons of different priors on standard data sets. Results are averaged over 10 runs (with different noise $\epsilon$ on each run). Results with lowest MSE appear in bold. ................................................................. 62
Acknowledgments

I wish to thank (in order of appearance): my parents, my long-suffering supervisor Richard Everson, as well as Mika Enslin and Michiru Sekiguchi. They all have, in different ways, supported me far more and for far longer than I could reasonably have asked (or they could have reasonably have wished) for and I want to express my heartfelt gratitude and debt to them.