12

Adaptive Sparse Regression

Mário A. T. Figueiredo[1]

Summary

In sparse regression, the goal is to obtain an estimate of the regression coefficients in which several of them are set exactly to zero. Sparseness is a desirable feature in regression problems, for several reasons. For example, in linear regression, sparse models are interpretable, that is, we find which variables are relevant; in kernel-based methods, like in *support vector* regression, sparseness leads to regression equations involving only a subset of the learning data. In all approaches to sparse regression, it is necessary to estimate parameters which will ultimately control the degree of sparseness of the obtained solution. This commonly involves cross-validation methods which waste learning data and are time consuming. In this chapter we present a sparseness inducing prior which does not involve any (hyper)parameters that need to be adjusted or estimated. Experiments with several publicly available benchmark data sets show that the proposed approach yields state-of-the-art performance. In particular, our method outperforms support vector regression and performs competitively with the best alternative techniques, both in terms of error rates and sparseness, although it involves no tuning or adjusting of sparseness-controlling hyper-parameters.

12.1 Introduction

The goal of supervised learning is to infer a functional relationship $y = f(\mathbf{x})$, based on a set of (possibly *noisy*) training examples $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. Usually, the inputs are vectors, $\mathbf{x}_i =$

[1] Mário A. T. Figueiredo is with the Institute of Telecommunications, and Department of Electrical and Computer Engineering Instituto Superior Técnico, 1049-001 Lisboa, Portugal (Email: mtf@lx.it.pt). This research was partially supported by the Portuguese Foundation for Science and Technology (FCT), under project POSI/33143/SRI/2000.

$[x_{i,1}, ..., x_{i,d}]^T \in \mathbb{R}^d$. When y is continuous (typically $y \in \mathbb{R}$), we are in the context of *regression*, whereas in *classification*, y is of categorical nature (*e.g.*, $y \in \{-1, 1\}$). Usually, the structure of $f(\cdot)$ is assumed fixed and the objective is to estimate a vector of parameters $\boldsymbol{\beta}$ defining it; accordingly we write $y = f(\mathbf{x}, \boldsymbol{\beta})$.

To achieve good *generalization* (*i.e.* to perform well on yet unseen data) it is necessary to control the *complexity* of the learned function (see [5], [6], [23] and [27], and the many references therein). In Bayesian approaches, complexity is controlled by placing a prior on the function to be learned, *i.e.*, on $\boldsymbol{\beta}$. This should not be confused with a *generative* (*informative*) Bayesian approach, since it involves no explicit modelling of the joint probability $p(\mathbf{x}, y)$. A common choice is a zero-mean Gaussian prior, which appears under different names, like *ridge regression* [11], or *weight decay*, in the neural learning literature [2]. Gaussian priors are also used in non-parametric contexts, like the Gaussian processes (GP) approach [6], [18], [28], [29], which has roots in earlier spline models [13] and regularized radial basis functions [22]. Very good performance has been reported for methods based on Gaussian priors [28], [29]. Their main disadvantage is that they do not control the structural complexity of the resulting functions. That is, if one of the components of $\boldsymbol{\beta}$ (say, a weight in a neural network) happens to be irrelevant, a Gaussian prior will not set it exactly to zero, thus pruning that parameter, but to some small value.

Sparse estimates (*i.e.*, in which irrelevant parameters are set exactly to zero) are desirable because (in addition to other learning-theoretic reasons [27]) they correspond to a structural simplification of the estimated function. Using Laplacian priors (equivalently, l_1-penalized regularization) is known to promote sparseness [4], [10], [25], [31]. *Support vector machines* (SVM) also lead to sparse regressors without explicitly adopting a sparseness inducing prior [6], [27]. Interestingly, however, it can be shown that the SVM and l_1-penalized regression are closely related [10].

Both in approaches based on Laplacian priors and in SVMs, there are hyper-parameters which control the degree of sparseness of the obtained estimates. These are commonly adjusted using cross-validation methods which do not optimally utilize the available data, and are time consuming.

In this chapter, we propose an alternative approach which involves no hyper-parameters. The key steps of our proposal are: (i) a hierarchical Bayes interpretation of the Laplacian prior as a *normal/independent* distribution (as used in robust regression [14]); (ii) a Jeffreys' non-informative second-level hyper-prior (in the same spirit as [9]) which expresses scale-invariance and, more importantly, is parameter-free [1]; (iii) a simple *expectation-maximization* (EM) algorithm which yields a *maximum a posteriori* (MAP) estimate of $\boldsymbol{\beta}$, and of the observation noise variance.

Our method is related to the *automatic relevance determination* (ARD) concept [18], [16], which underlies the recently proposed *relevance vector machine* (RVM) [3], [26]. The RVM exhibits state-of-the-art performance, and it seems to outperform SVMs both in terms of accuracy and sparseness [3], [26]. However, we do not resort to a *type-II maximum likelihood* approximation [1] (as in ARD and RVM); rather, our modelling assumptions lead to a marginal *a posteriori* probability function on $\boldsymbol{\beta}$ whose mode is located by a very simple EM algorithm. Related hierarchical-Bayes models were proposed in [12] and [24]; in those papers inference is carried out by *Markov chain Monte Carlo* (MCMC) sampling.

Experimental evaluation of the proposed method, both with synthetic and real data, shows that it performs competitively with (often better than) RVM and SVM.

12.2 Bayesian Linear Regression

12.2.1 Gaussian prior and ridge regression

We consider functional representations which are linear with respect to $\boldsymbol{\beta}$, that is,

$$f(\mathbf{x}, \boldsymbol{\beta}) = \boldsymbol{\beta}^T \mathbf{h}(\mathbf{x});$$

we will denote the dimensionality of $\boldsymbol{\beta}$ as k. This form includes:

(i) classical linear regression, where $\mathbf{h}(\mathbf{x}) = [1, x_1, ..., x_d]^T$;

(ii) nonlinear regression via a set of k basis functions, in which case $\mathbf{h}(\mathbf{x}) = [\phi_1(\mathbf{x}), ..., \phi_k(\mathbf{x})]^T$; this is the case, for example, of radial basis functions (with fixed basis functions), spline functions (with fixed knots), or even free knot splines (see [21]);

(iii) kernel regression, with $\mathbf{h}(\mathbf{x}) = [1, K(\mathbf{x}, \mathbf{x}_1), ..., K(\mathbf{x}, \mathbf{x}_n)]^T$, where $K(\mathbf{x}, \mathbf{y})$ is some (symmetric) kernel function [6] (as in SVM and RVM regression), though not necessarily verifying Mercer's condition.

We follow the standard assumption that

$$y_i = f(\mathbf{x}_i, \boldsymbol{\beta}) + w_i,$$

for $i = 1, ..., n$, where $[w_1, ..., w_n]$ is a set of independent zero-mean Gaussian variables with variance σ^2. With $\mathbf{y} \equiv [y_1, ..., y_n]^T$, the likelihood function is then

$$p(\mathbf{y}|\boldsymbol{\beta}) = \mathcal{N}(\mathbf{y}|\mathbf{H}\boldsymbol{\beta}, \sigma^2\mathbf{I}),$$

where $\mathbf{H}$ is the $(n \times k)$ *design matrix* which depends on the $\mathbf{x}_i$s and on the adopted function representation, and $\mathcal{N}(\mathbf{v}|\boldsymbol{\mu}, \mathbf{C})$ denotes a Gaussian density of mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$, evaluated at $\mathbf{v}$.

With a zero-mean Gaussian prior with covariance $\mathbf{A}$, that is $p(\boldsymbol{\beta}|\mathbf{A}) = \mathcal{N}(\boldsymbol{\beta}|0, \mathbf{A})$, it is well known that the posterior is still Gaussian; more specifically,

$$p(\boldsymbol{\beta}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\beta}|\widehat{\boldsymbol{\beta}}, \mathbf{D})$$

with mean and mode at

$$\widehat{\boldsymbol{\beta}} = (\sigma^2 \mathbf{A}^{-1} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}.$$

When $\mathbf{A}$ is proportional to identity, say $\mathbf{A} = \mu^2\mathbf{I}$, this is equivalent to *ridge regression* (RR) [11], although RR was proposed in a non-Bayesian context.

12.2.2 Laplacian prior, sparse Regression, and the LASSO

Let us now consider a Laplacian prior for $\boldsymbol{\beta}$,

$$p(\boldsymbol{\beta}|\alpha) = \prod_{i=1}^{k} \frac{\alpha}{2} \exp\{-\alpha\,|\beta_i|\} = \left(\frac{\alpha}{2}\right)^k \exp\{-\alpha\;\|\boldsymbol{\beta}\|_1\},$$

where $\|\mathbf{v}\|_1 = \sum_i |v_i|$ denotes the l_1 norm. The posterior $p(\boldsymbol{\beta}|\mathbf{y})$ is no longer Gaussian. The *maximum a posteriori* (MAP) estimate is now given by

$$\widehat{\boldsymbol{\beta}} = \arg\min\{\|\mathbf{H}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + 2\,\sigma^2\alpha\,\|\boldsymbol{\beta}\|_1\}, \tag{12.1}$$

where $\|\mathbf{v}\|_2$ is the Euclidean (l_2) norm. In linear regression, this is called the LASSO (*least absolute shrinkage and selection operator*) [25]. The main effect of the l_1 penalty is that some of the components of $\widehat{\boldsymbol{\beta}}$ may be exactly zero. If $\mathbf{H}$ is an orthogonal matrix, (12.1) can be solved separately for each β_i, leading to the *soft threshold* estimation rule, widely used in wavelet-based signal/image denoising [7]. The sparseness inducing nature of the Laplacian prior (or equivalently, of the l_1 penalty) has been exploited in several other contexts [4], [31], [21], [15], [19].

12.3 Hierarchical Interpretation of the Laplacian

Let us consider an alternative model: let each β_i have a zero-mean Gaussian prior $p(\beta_i|\tau_i) = \mathcal{N}(\beta_i|0, \tau_i)$, with its own variance τ_i (like in ARD and RVM). Now, rather than adopting a *type-II maximum likelihood* criterion (as in ARD and RVM [26]), let us consider hyper-priors for the τ_is and integrate them out. Assuming exponential hyper-priors $p(\tau_i|\gamma) = (\gamma/2)\exp\{-\gamma\,\tau_i/2\}$ (for $\tau_i \geq 0$, because these are variances) we obtain

$$p(\beta_i|\gamma) = \int_0^\infty p(\beta_i|\tau_i)p(\tau_i|\gamma)\,d\tau_i = \frac{\sqrt{\gamma}}{2}\exp\{-\sqrt{\gamma}\,|\beta_i|\}.$$

This shows that the Laplacian prior is equivalent to a 2-level hierachical model: zero-mean Gaussian priors with independent exponentially distributed variances. This equivalence has been previously exploited in robust *least absolute deviation* (LAD) regression [14].

The hierarchical decomposition of the Laplacian prior allows using the EM algorithm to implement the LASSO criterion in (12.1) by simply regarding $\boldsymbol{\tau} = [\tau_1, ..., \tau_k]$ as *hidden/missing data*. Let us define the following diagonal matrix: $\boldsymbol{\Upsilon}(\boldsymbol{\tau}) = \text{diag}(\tau_1^{-1}, ..., \tau_m^{-1})$. In the presence of $\boldsymbol{\Upsilon}(\boldsymbol{\tau})$, the complete log-posterior (with a flat prior for σ^2),

$$\log p(\boldsymbol{\beta}, \sigma^2 | \mathbf{y}, \boldsymbol{\tau}) \propto -n\,\sigma^2 \log \sigma^2 - \|\mathbf{y} - \mathbf{H}\boldsymbol{\beta}\|_2^2 - \sigma^2 \boldsymbol{\beta}^T \boldsymbol{\Upsilon}(\boldsymbol{\tau}) \boldsymbol{\beta}, \qquad (12.2)$$

would be easy to maximize with respect to $\boldsymbol{\beta}$ and σ^2. Since this complete log-posterior is liner with respect to $\boldsymbol{\Upsilon}(\boldsymbol{\tau})$, the E-step reduces to the computation of the conditional expectation of $\boldsymbol{\Upsilon}(\boldsymbol{\tau})$, given current (at iteration t) estimates $\widehat{\sigma^2}_{(t)}$ and $\widehat{\boldsymbol{\beta}}_{(t)}$. It can easily be shown that this leads to

$$\begin{aligned} \mathbf{V}_{(t)} &\equiv E[\boldsymbol{\Upsilon}(\boldsymbol{\tau}) \,|\, \mathbf{y}, \widehat{\sigma^2}_{(t)}, \widehat{\boldsymbol{\beta}}_{(t)}] \\ &= \gamma\, \text{diag}(|\widehat{\beta}_{1,(t)}|^{-1}, ..., |\widehat{\beta}_{k,(t)}|^{-1}). \end{aligned} \qquad (12.3)$$

Finally, the M-step consists in updating the estimates of σ^2 and $\boldsymbol{\beta}$ by maximizing the complete log-posterior, with $\mathbf{V}_{(t)}$ replacing the missing $\boldsymbol{\Upsilon}(\boldsymbol{\tau})$. This leads to

$$\widehat{\sigma^2}_{(t+1)} = \frac{1}{n} \|\mathbf{y} - \mathbf{H}\widehat{\boldsymbol{\beta}}_{(t)}\|_2^2 \qquad (12.4)$$

and

$$\widehat{\boldsymbol{\beta}}_{(t+1)} = (\widehat{\sigma^2}_{(t+1)} \mathbf{V}_{(t)} + \mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}. \qquad (12.5)$$

This EM algorithm is not the most efficient way to solve (12.1); faster special-purpose methods have been proposed in [20], [25]. Our main goal is to open the way to the adoption of different hyper-priors that do not corresponf to LASSO estimates.

12.4 The Jeffreys Hyper-Prior

One question remains: how to adjust γ, which is the main parameter controlling the degree of sparseness of the estimates? Our proposal is to remove γ from the model, by replacing the exponential hyper-prior by a non-informative Jeffreys hyper-prior

$$p(\tau_i) \propto \tau_i^{-1}. \qquad (12.6)$$

This prior expresses ignorance with respect to scale (see [9], [1]) and, most importantly, it is parameter-free. Of course this is no longer equivalent to a Laplacian prior on $\boldsymbol{\beta}$, but to some other prior [9]. As will be shown experimentally, this prior strongly induces sparseness and yields state-of-the-art performance in regression applications. Computationally, this choice leads to a minor modification of the EM algorithm described above: matrix

$\mathbf{V}_{(t)}$ is now given by

$$\mathbf{V}_{(t)} = \mathrm{diag}(|\widehat{\beta}_{1,(t)}|^{-2}, ..., |\widehat{\beta}_{k,(t)}|^{-2}). \tag{12.7}$$

(instead of Eq. (12.3)). Notice the absence of parameter γ.

Since several of the components of $\widehat{\beta}$ are expected to go to zero, it is not convenient to deal with $\mathbf{V}_{(t)}$ as defined in Eq. (12.7). However, defining a new diagonal matrix

$$\mathbf{U}_{(t)} = \mathrm{diag}(|\widehat{\beta}_{1,(t)}|, ..., |\widehat{\beta}_{k,(t)}|),$$

we can re-write Eq. (12.5) in the M-step as

$$\widehat{\boldsymbol{\beta}}_{(t+1)} = \mathbf{U}_{(t)}(\widehat{\sigma^2}_{(t+1)}\mathbf{I} + \mathbf{U}_{(t)}\mathbf{H}^T\mathbf{H}\mathbf{U}_{(t)})^{-1}\mathbf{U}_{(t)}\mathbf{H}^T\mathbf{y}. \tag{12.8}$$

This form of the algorithm avoids the inversion of the elements of $\widehat{\boldsymbol{\beta}}_{(t)}$. Moreover, it is not necessary to invert the matrix, but simply to solve the corresponding linear system, whose dimension is only the number of non-zero elements in $\mathbf{U}_{(t)}$.

12.5 Experiments

Our first example illustrates the use of the proposed method for variable selection in standard linear regression. Consider a sequence of 20 true $\boldsymbol{\beta}$s, having from 1 to 20 non-zero components (out of 20): from $[3, 0, 0, ..., 0]$ to $[3, 3, ..., 3]$. For each $\boldsymbol{\beta}$, we obtain 100 random (50×20) design matrices, following the procedure in [25], and for each of these, we obtain data points with unit noise variance. Fig. 12.1 shows the mean number of estimated non-zero components, as a function of the true number. Our method exhibits a very good ability to find the correct number of nonzero components in $\boldsymbol{\beta}$, in an adaptive manner.

Table 12.1. Relative (%) improvement in modelling error of several mehods.

Method	$\boldsymbol{\beta}_a$	$\boldsymbol{\beta}_b$
Proposed method	28%	74%
LASSO (CV)	13%	69%
LASSO (GCV)	30%	65%
Subset selection	13%	77%

We now consider two of the experimental setups used in [25]: $\boldsymbol{\beta}_a = [3, 1.5, 0, 0, 2, 0, 0, 0]$, with $\sigma = 3$, and $\boldsymbol{\beta}_b = [5, 0, 0, 0, 0, 0, 0, 0]$, with $\sigma = 2$. In both cases, $n = 20$, and the design matrices are generated as in [25]. In Table 12.1, we compare the relative modelling error ($ME = E[\|\mathbf{H}\widehat{\boldsymbol{\beta}} - \mathbf{H}\boldsymbol{\beta}\|^2]$) improvement (with respect to the least squares solution) of our method and of several methods studied in [25]. Our method

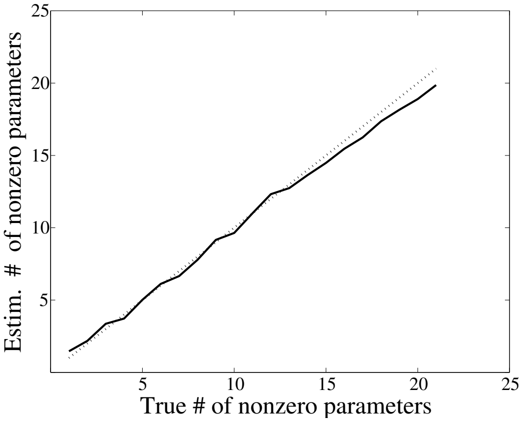

Figure 12.1. Mean number of nonzero components in $\widehat{\boldsymbol{\beta}}$ versus the number of nonzero components in $\boldsymbol{\beta}$ (the dotted line is the identity).

performs comparably with the best method for each case (LASSO tuned by generalized cross-validation, for $\boldsymbol{\beta}_a$, and subset selection, for $\boldsymbol{\beta}_b$), although it involves no tuning or adjustment of parameters, and is computationally faster.

We now study the performance of our method in kernel regression, using Gaussian kernels, *i.e.*, $K(\mathbf{x}, \mathbf{x}_i) = \exp\{-\|\mathbf{x} - \mathbf{x}_i\|^2/(2h^2)\}$. We begin by considering the synthetic example studied in [3] and [26], where the true function is $y = \sin(x)/x$ (see Fig. 12.2). To compare our results to the RVM and the variational RVM (VRVM), we ran the algorithm on 25 generations of the noisy data. The results are summarized in Table 12.2 (which also includes the SVM results from [3]). Of course the results depend on the

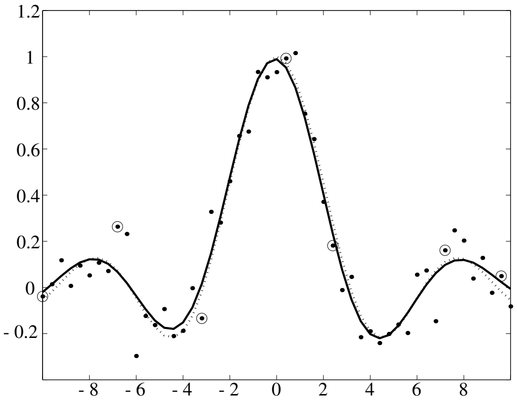

Figure 12.2. Kernel regression. Dotted line: true function $y = \sin(x)/x$. Dots: 50 noisy observations ($\sigma = 0.1$). Solid line: the estimated function. Circles: data points corresponding to the non-zero parameters.

Table 12.2. Mean root squared errors and mean number of kernels for the "$\sin(x)/x$" function example.

Method	MSE	No. kernels
New method	0.0455	7.0
SVM	0.0519	28.0
RVM	0.0494	6.9
VRVM	0.0494	7.4

Table 12.3. Mean root squared errors and mean number of kernels for the "Boston housing" example.

Method	MSE	No. kernels
New method	9.98	45.2
SVM	10.29	235.2
RVM	10.17	41.1
VRVM	10.36	40.9

choice of kernel width h (as do the RVM and SVM results), which was adjusted by cross validation.

Finally, we have also applied our method to the well-known *Boston housing* data-set (20 random partitions of the full data-set into 481 training samples and 25 test samples); Table 12.3 shows the results, again versus SVM, RVM, and VRVM regression (as reported in [3]). In these tests, our method performs better than RVM, VRVM, and SVM regression, although it doesn't require any tuning.

12.6 Concluding remarks

We have introduced a new sparseness inducing prior for regression problems which is related to the Laplacian prior. Its main feature is the absence of any hyper-parameters to be adjusted or estimated. Experiments have shown state-of-the-art performance, although the method involves no tuning or adjusting of sparseness-controlling hyper-parameters.

It is possible to apply the approach herein described to classification problems (*i.e.*, when the response variable is of categorical nature) using a generalized linear model [17]. In [8] we show how a very simple EM algorithm can be used to address the classification case, leading also to state-of-the-art performance.

One of the weak points of our approach (which is only problematic in kernel-based methods) is the need to solve a linear system in the M-step, whose computational requirements make it impractical to use with very large data sets. This issue is of current interest to researchers in kernel-based methods (*e.g.*, [30]), and we also intend to focus on it.

References

[1] J. Berger, *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, 1980.

[2] C. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[3] C. Bishop and M. Tipping, "Variational relevance vector machines," in *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, pp. 46–53, Morgan Kaufmann, 2000.

[4] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal of Scientific Computation*, vol. 20, no. 1, pp. 33–61, 1998.

[5] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods.* John Wiley & Sons, New York, 1998.

[6] N. Cristianini and J. Shawe-Taylor, *Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, 2000.

[7] D. L. Donoho and I. M. Johnstone, "Ideal adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[8] M. Figueiredo, "Adaptive sparseness using Jeffreys prior", in T. Dietterich, S. Becker, and Z. Ghahramani (editors), *Advances in Neural Information Processing Systems 14*, MIT Press, Cambridge, MA, 2002.

[9] M. Figueiredo and R. Nowak, "Wavelet-based image estimation: an empirical Bayes approach using Jeffreys' noninformative prior," in *IEEE Transactions on Image Processing*, vol. 10, pp. 1322-1331, 2001.

[10] F. Girosi, "An equivalence between sparse approximation and support vector machines," *Neural Computation*, vol. 10, pp. 1445–1480, 1998.

[11] A. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 1970.

[12] C. Holmes and D. Denison, "Bayesian wavelet analysis with a model complexity prior," in *Bayesian Statistics 6*, J. Bernardo, J. Berger, A. Dawid, and A. Smith (editors), Oxford University Press, 1999.

[13] G. Kimeldorf and G. Wahba, "A correspondence between Bayesian estimation of stochastic processes and smoothing by splines," *Annals of Math. Statistics*, vol. 41, pp. 495–502, 1990.

[14] K. Lange and J. Sinsheimer, "Normal/independent distributions and their applications in robust regression," *Journal of Computational and Graphical Statistics*, vol. 2, pp. 175–198, 1993.

[15] M. Lewicki and T. Sejnowski, "Learning overcomplete representations," *Neural Computation*, vol. 12, pp. 337–365, 2000.

[16] D. MacKay, "Bayesian non-linear modelling for the 1993 energy prediction competition," in *Maximum Entropy and Bayesian Methods*, G. Heidbreder (editor), pp. 221–234, Kluwer, 1996.

[17] P. McCullagh and J. Nelder, *Generalized Linear Models.* Chapman and Hall, London, U.K., 1989.

[18] R. Neal, *Bayesian Learning for Neural Networks.* Springer Verlag, New York, 1996.

[19] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[20] M. Osborne, B. Presnell, and B. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, pp. 389–404, 2000.

[21] M. Osborne, B. Presnell, and B. Turlach, "Knot selection for regression splines via the LASSO", in *Dimension Reduction, Computational Complexity, and Information*, S. Weisberg (editor), pp. 44-49, Interface Foundation of North America, Fairfax Station, VA, 1998.

[22] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, 1990.

[23] B. Ripley, *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[24] M. Smith and R. Kohn, "Nonparametric regression via Bayesian variable selection," in *Journal of Econometrics*, vol. 75, pp. 317-344, 1996.

[25] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society (B)*, vol. 58, 1996.

[26] M. Tipping, "The relevance vector machine," in *Advances in Neural Information Processing Systems 12*, S. Solla, T. Leen, and K.-R. Müller (editors), pp. 652-658, MIT Press, 2000.

[27] V. Vapnik, *Statistical Learning Theory.* John Wiley & Sons, New York, 1998.

[28] C. Williams, "Prediction with Gaussian processes: from linear regression to linear prediction and beyond," in *Learning in Graphical Models*, MIT Press, Cambridge, MA, 1998.

[29] C. Williams and D. Barber, "Bayesian classification with Gaussian priors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1342–1351, 1998.

[30] C. Williams and M. Seeger, "Using the Nyström method to speedup kernel machines," in *Advances in Neural Information Processing Systems 13*, T. Leen, T. Dietterich, and V. Tresp (editors), MIT Press, Cambridge, MA, 2001.

[31] P. Williams, "Bayesian regularization and pruning using a Laplace prior," *Neural Computation*, vol. 7, pp. 117–143, 1995.