DIMACS Technical Report 2003-08
April 2003

On Bayesian Learning of Sparse Classifiers

by

Wen-Hua Ju[1], David Madigan[2], and Steven L. Scott[3]
Avaya Labs[1,2], Rutgers University[2], University of Southern California[3]
Contact e-mail: *madigan@stat.rutgers.edu*

ABSTRACT

Figueiredo (2001) and Figueiredo and Jain (2001) described a particular sparseness-inducing Bayesian model for probit regression. For several standard datasets, they reported predictive performance for their model that was as a good as, or better than, previously reported results. In this paper we explore several aspects of the Figueiredo and Jain model in an attempt to better understand its performance.

1 Introduction

Inductive supervised learning infers a functional relation $y = f(\mathbf{x})$ from a set of training examples $D = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)\}$. In what follows the inputs are vectors $[x_{i_1}, \ldots, x_{i_d}]^T$ in $\Re^d$, $y \in \{-1, 1\}$, and we refer to f as a classifier. We assume that a vector of parameters, β defines f and we write $f(\mathbf{x}; \beta)$. The learning procedures we consider output either a point estimate of β or a posterior distribution for β.

Our objective is to produce a classifer that makes accurate predictions for future input vectors. Typically this requires the learning procedure to control complexity and avoid over-fitting the training data. The Bayesian approach to complexity places a prior distribution on β, typically resulting in estimates that shrink towards zero. Using so-called Laplacian priors can result in posterior modes for some components of β that are exactly zero. Such "sparse classifiers" can yield excellent predictive performance and are closely related to support vector machines (SVM) (see Girosi, 1998, and Zhang and Oles, 2001) and to Tibshirani's lasso (Tibshirani, 1995).

Figueiredo (2001) and Figueiredo and Jain (2001), hereafter FJ, considered classifiers of the form:

$$p(y = 1|\mathbf{x}) = \psi(\beta^{\mathbf{T}}\mathbf{h}(\mathbf{x})).$$

The logistic link function is a common choice for ψ, but following FJ, we instead adopt the probit model, $\psi(z) = \Phi(z)$, where Φ denotes the standard Gaussian distribution function. We consider two forms for $\mathbf{h}$, namely $\mathbf{h}(\mathbf{x}) = [1, x_1, \ldots, x_d]^T$ (i.e., the original input variables), and $\mathbf{h}(\mathbf{x}) = [1, K(\mathbf{x}, \mathbf{x_1}), \ldots, K(\mathbf{x}, \mathbf{x_n})]^T$, where $K(\mathbf{x}, \mathbf{y})$ is some symmetric kernel function not necessarily satisfying Mercer's condition. SVMs and the more recent Relevance Vector Machines (RVM, Bishop and Tipping, 2000) generally adopt the kernel representation.

FJ adopted a hierarchical prior for β, specifically. for $i = 1, \ldots, d$:

$$p(\beta_i|\tau_i) = N(0, \tau_i)$$

and

$$p(\tau_i) \propto 1/\tau_i,$$

the prior on the τ's being the Jeffreys' prior.

The Laplacian prior instead uses exponential priors for the τ's:

$$p(\tau_i|\gamma) = \frac{\gamma}{2}\exp(-\frac{\gamma}{2}\tau_i).$$

This is equivalent to adopting the following prior for β:

$$p(\beta_i|\gamma) = \frac{\sqrt{\gamma}}{2}\exp(-\sqrt{\gamma}|\beta_i|)$$

and the corresponding maximum *a posteriori* (MAP) estimates are the lasso estimates. Note that the Laplacian prior requires a choice for the hyperparameter γ, perhaps via cross-validation, whereas the FJ prior is more convenient insofar as it has no tunable hyperparameters.

In what follows we explore three questions:

1. FJ make "plug-in" predictions at the posterior mode of β. Alternatively one can adopt a fully Bayesian approach and integrate out β when making predictions. FJ find the posterior mode using an efficient EM algorithm. By contrast, integration over the posterior distribution of β requires a computationally expensive Monte Carlo algorithm. Does the fully Bayesian approach lead to better predictions?

2. FJ only report results using a kernel representation (specifically a Gaussian kernel). Does the kernel representation provide better predictive performance that the simpler non-kernel representation?

3. Does the FJ prior provide similar predictive performance to the Laplacian prior?

2 The Learning Algorithms

Here we describe FJ's EM algorithm for finding the posterior mode and a proposed MCMC algorithm for finding the posterior mean. Both algorithms make use of the latent variable representation for probit regression introduced by Albert and Chib

(1993) and we describe this first. Define n independent latent variables $z_1, \ldots, z_n$ where z_i has distribution $N(\beta^{\mathbf{T}}\mathbf{h}(\mathbf{x}), 1)$. Then define $y_i = 1$ if $z_i > 0$ and $y_i = -1$ if $z_i \leq 0$. It is straightforward to show that the y_i are then independent Bernoulli variables with $p(y_i = 1) = \Phi(\beta^{\mathbf{T}}\mathbf{h}(\mathbf{x_i})), i = 1, \ldots, n$, and we recover the standard probit model. Figure 1 shows the corresponding graphical Markov model using the BUGS plate notation.

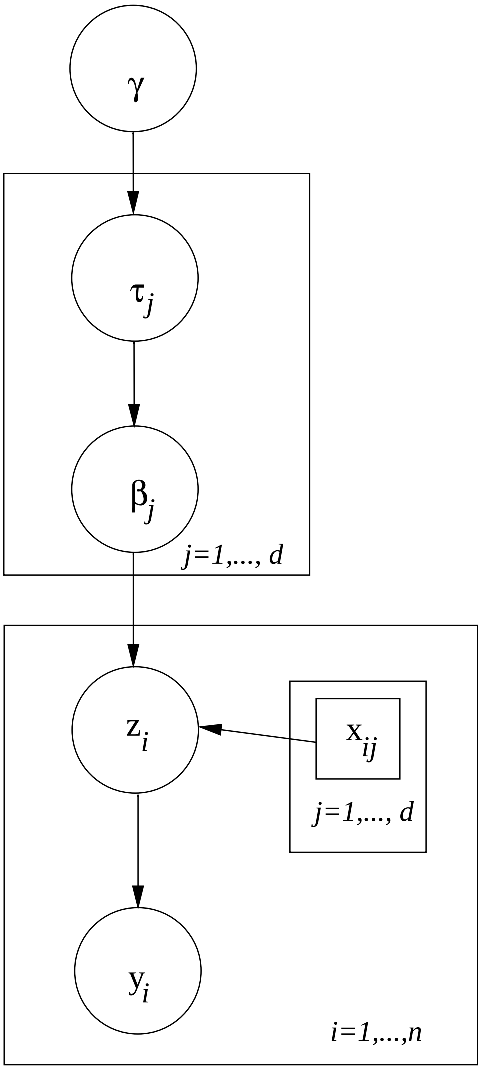

Figure 1: *Probit Model with Hierarchical Prior and Latent Variables.*

Observe that if the z_i are known, and β has a multivariate normal prior, then the posterior distribution for β is available in closed form using standard normal linear model results. The z_i are latent, but given the data y_i, z_i has truncated normal distribution. This setup facilitates both the EM algorithm of FJ and our proposed MCMC algorithm in an obvious way.

For both the Laplacian and FJ priors, FJ derive an EM algorithm that estimates β by treating treating both τ and $\mathbf{z}$ as missing. The complete data log-posterior is:

$$\begin{aligned} \log p(\beta|\mathbf{y}, \tau, \mathbf{z}) &\propto \log p(\mathbf{z}|\beta) + \log p(\beta|\tau) &(1)\\ &\propto -\|\mathbf{H}\beta - \mathbf{z}\|^2 - \beta^{\mathbf{T}}\mathbf{\Gamma}(\tau)\beta, &(2)\end{aligned}$$

where $\mathbf{\Gamma}(\tau) = \text{diag}(\tau_i^{-1}, \ldots, \tau_d^{-1})$ and H is the design matrix with rows $\mathbf{h}(\mathbf{x_1})^{\mathbf{T}}, \ldots, \mathbf{h}(\mathbf{x_n})^{\mathbf{T}}$. The E-step requires computing $E[\tau_i^{-1}|\hat{\beta}^{(t)}, \mathbf{y}, \gamma]$ and $E[z_i|\hat{\beta}^{(t)}, \mathbf{y}, \gamma]$. For the former, we have that:

$$p(\tau_i|\hat{\beta}^{(t)}, \mathbf{y}, \gamma) \propto p(\hat{\beta}^{(t)}|\tau_i)p(\tau_i|\gamma).$$

Then

$$\begin{aligned} \omega_i \equiv E[\tau_i^{-1}|\hat{\beta}^{(t)}, \mathbf{y}, \gamma] &= \frac{\int_0^\infty \frac{1}{\tau_i} p(\tau_i|\gamma)p(\hat{\beta}^{(\mathbf{t})}|\tau_i)d\tau_i}{\int_0^\infty p(\tau_i|\gamma)p(\hat{\beta}^{(\mathbf{t})}|\tau_i)d\tau_i} &(3)\\ &= \gamma|\hat{\beta}^{(t)}|^{-1}. &(4)\end{aligned}$$

For the latter, z_i is Gaussian with mean $\beta^{\mathbf{T}}\mathbf{h}(\mathbf{x_i})$, but left-truncated at zero if $y_i = 1$ and right-truncated at zero of $y_i = -1$. Thus:

$$v_i \equiv E[z_i|\hat{\beta}^{(t)}, \mathbf{y}, \gamma] =$$

$$\begin{cases} \beta^{\mathbf{T}}\mathbf{h}(\mathbf{x_i}) + \frac{N(\beta^T h(x_i))}{1-\Phi(-\beta^T h(x_i))} & \text{if } y_i = 1 \\ \beta^{\mathbf{T}}\mathbf{h}(\mathbf{x_i}) + \frac{N(\beta^T h(x_i))}{\Phi(-\beta^T h(x_i))} & \text{if } y_i = -1. \end{cases}$$

So the E-step then replaces Γ by its conditional expectation $\mathbf{\Omega} = \text{diag}(\omega_1, \ldots, \omega_d)$ and replaces $\mathbf{z}$ by $\mathbf{v}$.

The M-step carries out a maximization with respect to β leading to:

$$\hat{\beta}^{(t+1)} = \left(\mathbf{\Omega} + \mathbf{H^T H}\right)^{-1}\mathbf{H^T v}.$$

The EM procedure for the FJ prior requires minor modifications of the above algorithm. We refer the reader to FJ for details.

For fully Bayesian inference, we modify the Gibbs sampling procedure of Albert and Chib (1993). The basic idea is to draw a dependent sample from the target posterior distribution by drawing in turn from the conditional distribution of each of the unknowns given all the knowns and remaining unkowns. Specifically, the sampler draws in turn from:

$$z_i | z_{-i}, \mathbf{y}, \beta, \tau \sim TN_{y_i}(\beta^{\mathbf{T}} \mathbf{h}(\mathbf{x_i}))$$

where "TN" is a truncated normal as described above, and

$$\beta | \mathbf{z}, \mathbf{y}, \tau \sim N((\mathbf{\Gamma} + \mathbf{H^T H})^{-1} \mathbf{H^T z}, (\mathbf{\Gamma} + \mathbf{H^T H})^{-1}).$$

The conditional density of τ is not available in closed form so we introduce a Metropolis-within-Gibbs step (see, for example, Gilks *et al.*, 1996). For each component $i = 1, \ldots, d$, propose a candidate $\tau_i^{(t+1)}$ uniformly from a interval around $\tau_i^{(t)}$ of width h. For the Laplacian prior, accept the proposed move with probability:

$$\min\{1, \frac{N(\beta_i, 0, \tau_i^{(t+1)}) \exp(\gamma \tau_i^{(t+1)})}{N(\beta_i, 0, \tau_i^{(t)}) \exp(\gamma \tau_i^{(t)})}\}$$

otherwise set $\tau_i^{(t+1)} = \tau_i^{(t)}$. For the FJ prior, the acceptance probability is:

$$\min\{1, \frac{N(\beta_i, 0, \tau_i^{(t+1)}) 1/\tau_i^{(t+1)}}{N(\beta_i, 0, \tau_i^{(t)}) 1/\tau_i^{(t)}}\}.$$

In the experiments reported below, $h = 1$, and further tuning minimally impacted convergence behavior.

3 Classification Experiments

Following FJ and to facilitate comparison, we conducted experiments using three well-known benchmark datasets, namely Pima Indians Diabetes, Leptograpsus Crabs, and Wisconsin Breast Cancer (WBC). The first two datasets are available divided into training and test sets. Pima Indians is at `http://www.stats.ox.ac.uk/pub/PRNN/`.

Crabs is at `http://www.inference.phy.cam.ac.uk/is/data/ripley-class/`. Pima has 200 training observations and 332 test observations. Crabs gas 80 training observations and 120 test observations. The third data set comes from the UCI repository and following FJ we consider random splits into 300 training observations and 269 test observations. In each case we standardized the input variables. As with FJ, for the kernel models, we used a Gaussian kernel with bandwidth 4 for Pima and Crabs and 12 for WBC. All MCMC runs are of length 10^6 with 10^3 burn-in. The EM algorithm terminated when $\|\hat{\beta}^{(t+1)} - \hat{\beta}^{(t)}\|/\|\hat{\beta}^{(t)}\| < 10^{-3}$. Table 1 shows the results with the FJ (i.e., Jeffreys) prior.

	Method	**Kernel**	**Bayes?**	**Pima**	**Crabs**	**WBC**
1	Probit (F&J)	yes	plug-in	61	0	3.2%
2	Probit	yes	plug-in	72	2	4.1%
3	Probit	no	plug-in	70	3	5.0%
4	Probit	yes	full	72	3	2.8%
5	Probit	no	full	70	3	3.3%

Table 1: Number of test set errors (Pima and Crabs) or error rate (WBC). Line 1 reproduces the results reported in Figueiredo (2001). WBC results are averaged over 5 random splits into 300 training examples and 269 test examples. "Full Bayes" results used MCMC runs of length 10^6. "Plug-in" used the EM algorithm described in the text.

We were unable to reproduce the results of FJ (see lines 1 and 2 of Table 1). We created a second implementation in a different programming language to verify our results.

NOTE TO REVIEWERS: WE HAVE REQUESTED FIGEUEIREDO'S CODE AND HE HAS PROMISED TO SEND IT TO US.

3.1 Plug-in or Fully Bayesian Predictions?

Fully Bayesian inference via Markov chain Monte Carlo required at least 100 times the computing effort of the plug-in EM algorithm in these experiments. Generally, the EM algorithm provided parameter estimates in less than one minute on a 1GHz processor. By contrast, MCMC runs of 10^6 iterations took anything from 20 hours for the crabs data to 8 days for the WBC data with a kernel. For experiments not involving a kernel, MCMC convergence was obtained with as few as 25,000 iterations so the computing times are somewhat excessive. For those experiments that used a kernel, however, mixing was considerably less rapid, and parameter estimates exhibited some instability even at 10^6 iterations. However, in each case the number of test set errors had stabilized. One run of 10^7 iterations for the Crabs data showed identical test set errors after each million iterations.

Despite this computing effort, comparison of the predictive performance of the fully Bayesian versus plug-in performance yields mixed results. For the Pima data, performance was identical for the plug-in and fully Bayesian predictions. For the Crabs data, the plug-in predictions for kernel-based model outperformed the fully Bayesian predictions. Without a kernel, the two approaches performed identically. For the WBC data, the fully Bayesian predictions outperformed their plug-in counterparts.

We note that in the context of text categorizatioin with a Bayesian multinomial model, Rennie (2001) reported that the Bayesian plug-in approach predictively outperformed the fully Bayesian approach.

3.2 Kernel or No Kernel?

Using a Gaussian kernel instead of the original input variables gives mixed results. For MAP plug-in predictions, use of the kernel provides improved predictive performance for Crabs and WBC but poorer performance for Pima. For fully Bayesian prediction, the kernel provides improved predictive performance for WBC, poorer performance for Pima, and identical performance for Crabs.

We only experimented with a Gaussian kernel, and used the bandwidths suggested by FJ. Different kernels or cross-validated bandwidth selection may show the kernel versions in a better light.

3.3 Jeffreys Prior or Exponential Prior?

Table 2 shows our results for the Laplacian prior for different settings of the hyper-parameter γ.

Again the results are mixed. For the Pima data and the WBC data, choosing $\gamma = 1$ does provide improved predictive performance (compare lines 1 and 4 of Table 2. For Crabs, we were not able to find a value of γ that outperformed Jeffreys.

3.4 Other Comparisons

Table 3 reproduces the results from Seeger (2000) along with our own implementation of logistic regression via maximum likelihood and the best dataset-specific results from the tables above.

Linear discriminant analysis and logistic regression show relatively poor predictive performance with the WBC data, but otherwise no clear winner emerges.

4 Discussion

The probit model with the Laplacian prior is the lasso version of probit regression. Although software for the lasso has been available for many years, we are not aware of any large-scale evaluation of its predictive performance.

Taking advantage of the Gaussian-exponential hierarchical representation of the Laplacian prior, FJ replace the exponential component with a parameter-free Jeffreys prior. The only motivation FJ provided was to avoid setting a hyperparameter, and indeed we know of know theoretical justification for the Jeffreys prior in this context. Our experiments show that while the Jeffreys prior performs adequately in some situations, judicious hyperparameter choice can sometimes improve predictive performance.

The primary motivation of this work was to empirically compare the performance of plug-in (or MAP) prediction versus the more fully Bayesian procedure of integrating out the parameters to make predictions. The finding that the plug-in approach is competitive with its computationally intensive counterpart is perhaps surprising and certainly comforting for large scale applications such as text categorization where an MCMC approach is quite impractical.

Clearly our findings are preliminary and much further experimentation and theoretical exploration is required. Albert and Chib (1993) discussed a number of extensions to the basic probit model, such as replacing the Gaussian link with a mixture of Gaussians, one possibility being a t-distribution with random degrees of freedom. In fact the usual logistic link is close to a t-distribution with approximately 8 degrees of freedom. Another extension concerns a hierarchical model where β is distributed *a priori* $N(\mathbf{A}\beta^{\mathbf{0}}, \sigma^2\mathbf{I})$ with β_0 having dimension less than d. This represents the prior belief that β lie on linear subspace $\mathbf{A}\beta_{\mathbf{0}}$. A prior distribution for (β_0, σ^2) completes the hierarchy. Bayesian model averaging (see, for example, Madigan and York, 1995), averaging over different sets of input features may also improve predictive performance. Here, a complete MCMC scheme requires a somewhat complex dimension-jumping sampler (see Green, 1995). However, an algorithm that only uses MCMC to move through model space and uses EM for parameter estimation is substantially simpler and may be just as effective predictively. Raftery *et al.* (1996) used a similar approach in the context of survival analysis.

In summary, our findings are that "results are mixed." This seems to indicate that plug-in methods with default priors, while requiring relatively modest computational effort, may provide competitive predictive performance in many situations.

Software code for both algorithms described in this paper available from the second author.

Acknowledgements

We thank David D. Lewis and Werner Stuetzle for helpful discussions.

References

Albert, J.H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, **88**, 669–679.

Bishop, C.M. and Tipping, M. (2000). Variational relevance vector machines. In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, 46–53.

Carlin, B.P. and Louis, T.A. (2000). *Bayes and Empirical Bayes Methods for Data Analysis (second edition)* Chapman and Hall, New York.

Figueiredo, M.A.T. (2001). Adaptive sparseness using Jeffreys prior. *Neural Information Processing Systems*, Vancouver, December 2001.

Figueiredo, M.A.T. and Jain, A.K. (2001). Bayesian learning of sparse classifiers. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Hawaii, December 2001.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (1995). *Bayesian Data Analysis.* Chapman and Hall, London.

Gilks, W.R., Richardson, S., and Spiegelhalter, D.J. (1996). *Markov chain Monte Carlo in Practice.* Chapman and Hall, London.

Girosi, F. (1998). An equivlance between sparse approximation and support vector machines. *Neural Computation*, **10**, 1445–1480.

Green, P.J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711-32.

Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review*, **63**, 215–232.

Raftery, A.E., Madigan, D., and Volinsky, C.T. (1996). Accounting for model uncertainty in survival analysis improves predictive performance. In: Bernardo, J. M., Berger, J. O., Dawid, A. P. and Smith A. F. M., (eds.), *Bayesian Statistics V*, Oxford University Press, 323-350.

Rennie, J.D.M. (2001). Improving Multi-class Text Classification with Naive Bayes. *AI Technical Report, Massachusetts Institute of Technology* AITR-2001-004.

Seeger, M. (2000). Bayesian model selection for support vector machines, Gaussian Processes and other kernle classifier. In: *Advances in Neural Information Processing*, MIT Press, 603–609.

Tibshirani, R. (1995). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, **58**, 267-288.

Zhang, T. and Oles, F. (2001). Text categorization based on regularized linear classifiers. *Information Retrieval*, **4**, 5–31.

	Method	**Kernel**	**Bayes?**	γ	**Pima**	**Crabs**	**WBC**
1	*Probit*	*no*	*full*	*Jeffreys*	*70*	*3*	*3.3%*
2	Probit	no	full	0.1	-	4	-
3	Probit	no	full	0.5	-	3	-
4	Probit	no	full	1	65	3	2.8%
5	Probit	no	full	2	-	4	-
6	Probit	no	full	10	-	7	-
1	*Probit*	*no*	*plug-in*	*Jeffreys*	*70*	*3*	*4.0%*
2	Probit	no	plug-in	0.1	66	4	4.3%
3	Probit	no	plug-in	0.5	66	3	3.9%
4	Probit	no	plug-in	1	65	4	3.6%
5	Probit	no	plug-in	2	66	4	3.9%
6	Probit	no	plug-in	10	63	17	3.4%
1	*Probit*	*yes*	*full*	*Jeffreys*	*72*	*3*	*2.8%*
7	Probit	yes	full	1	-	6	-
1	*Probit*	*yes*	*plug-in*	*Jeffreys*	*72*	*2*	*4.1%*
2	Probit	yes	plug-in	0.1	74	8	3.5%
3	Probit	yes	plug-in	0.5	67	-	-
4	Probit	yes	plug-in	1	70	-	2.8%
5	Probit	yes	plug-in	2	68	-	-
5	Probit	yes	plug-in	4	64	-	-
6	Probit	yes	plug-in	10	109	-	-

Table 2: Number of test set errors (Pima and Crabs) or error rate (WBC). WBC results are averaged over 5 random splits into 300 training examples and 269 test examples. MCMC runs of length 10^6 in each case.

	Method	Kernel	Bayes?	Pima	Crabs	WBC
1	SVM	yes	plug-in	64	4	3.3%
2	Linear Discriminant	yes	plug-in	67	3	7.1%
3	Gaussian Process	yes	plug-in	67	3	3.0%
4	Logistic Regression	no	mle	66	5	7.7%
5	Best Probit Result	-	-	65	2	2.8%

Table 3: Number of test set errors (Pima and Crabs) or error rate (WBC). Lines 1 through 3 are from Seeger (2000).