Sparse Bayesian Classifiers for Text Categorization

Susana Eyheramendy[1], Alexander Genkin[2], Wen-Hua Ju[3], David D. Lewis[4,5], and David Madigan[1,2,3,4]

Department of Statistics, Rutgers University[1]; DIMACS[2]; Avaya Labs Research[3]; Ornarose Inc.[4]; Consulting Computer Scientist[5]

Abstract

This paper empirically compares the performance of different Bayesian models for text categorization. In particular we examine so-called "sparse" Bayesian models that explicitly favor simplicity. We present empirical evidence that these models retain good predictive capabilities while offering significant computational advantages.

1 Introduction

Text categorization algorithms assign texts to predefined categories. The study of such algorithms has a rich history dating back at least forty years. In the last decade or so, the statistical approach has dominated the literature. The essential idea is to infer a classifier (i.e. a rule that decides whether or not a document should be assigned to a category) from a set of labeled documents (i.e. documents with known category assignments). Standard statistical classification tools such as Naive Bayes, logistic regression, and decision trees are immediately relevant and have been used with some success. Sebastiani (2002) provides an overview.

Researchers in statistical text categorization face two particular challenges. First, the scale of text categorization applications causes problems for many standard learning algorithms. Documents are represented by vectors of numeric values, with one value for each word that appears in any training document. Document feature vectors therefore are typically of dimension $10^5 - 10^6$ or more. High dimensionality both increases processing time and increases the risk of *overfitting*, i.e. that the learning algorithm will induce a classifier that reflects accidental properties of the particular training examples rather than systematic relationships between the words and

categories.

The second challenge is how to incorporate human understanding of the categorization problem into the learning process. For instance, people with a need for text categorization almost always have some sense of words that would be good predictors for each category. Textual descriptions of the category content, or just the category name itself, can also provide clues. However, most learning approaches provide no way to combine this evidence with the evidence from labeled data. The result is to increase the expense of using text categorization, since larger amounts of training data must be labeled. Further, the most interesting categories in intelligence applications often have few known example documents, so unless prior knowledge can be combined with these documents, it may not be possible to learn a classifier with good effectiveness.

The challenge of high dimensionality led early work in text categorization to focus on learning algorithms that were both computationally efficient (for speed) and very restricted in the classifiers they could produce (to avoid overfitting). These include Naive Bayes (Lewis, 1998) and the Rocchio algorithm (Rocchio, 1971). More recently, increased computing power and a better theoretical understanding of classifier complexity have enabled algorithms to learn less restricted, and thus more accurate classifiers while simultaneously avoiding overfitting and maintaining sufficient speed. Examples include support vector machines (Joachims, 1998, Lewis, 2002, and Lewis *et al.*, 2003) and ridge logistic regression (Zhang and Oles, 2001).

The challenge of integrating knowledge with learning has attracted much less attention in text categorization. This has motivated our interest in Bayesian learning algorithms. Bayesian learning algorithms allow the user to specify a probability distribution over possible parameter values of the learned classifier. This not only provides one solution to the overfitting problem (since the algorithm can use prior distribution to regularize the classifier), but the prior also provides a mathematically well-justified way to allow domain knowledge to influence the parameter values that result from learning.

Our focus here is on developing a Bayesian approach which avoids overfitting, is computationally efficient, and gives state-of-the-art effectiveness. As such, we focus on experimental comparisons of competing models.

2 Formal Framework

Inductive supervised learning infers a functional relation $y = f(\boldsymbol{x})$ from a set of training examples $D = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_i}, y_i), \ldots, (\boldsymbol{x_n}, y_n)\}$. In what follows the inputs are vectors, e.g. $\boldsymbol{x_i} = [x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,d}]^T$ in $\Re^d$, $y_i \in \{-1, 1\}$, and we refer to f as a classifier. For text categorization, the coordinates $x_{i,j}$ of the vectors consist of transformed word frequencies from documents, but the mathematical approach could be applied to vectors representing any form of data. We assume that a vector of parameters, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_j, \ldots, \beta_d]$ defines f and we write $f(\mathbf{x}; \boldsymbol{\beta})$ to indicate the value output by the classifier for vector $\mathbf{x}$. The learning procedures we consider output either a point estimate of $\boldsymbol{\beta}$ or a posterior distribution that specifies the relative likelihood of different values of $\boldsymbol{\beta}$.

Our objective is to produce a classifer that makes accurate predictions for future input vectors. This requires the learning procedure to control complexity and avoid overfitting the training data ("regularization"). One Bayesian approach to avoiding overfitting is use a prior distribution for $\boldsymbol{\beta}$ that assigns a high probability that each β_j will take on a value near 0. In particular, so-called Laplacian (i.e., double exponential) priors can result in posterior modes for some components of $\boldsymbol{\beta}$ that are exactly zero. Such "sparse classifiers" yield excellent predictive performance on standard datasets. Furthermore, sparse classifiers, having fewer non-zero parameters, offer computational advantages and are simpler to understand. The models we discuss here are closely related to support vector machines (SVM) (see Girosi, 1998, and Zhang and Oles, 2001), relevance vector machines (RVM) (Tipping, 2001), and to the lasso (Tibshirani, 1995).

We consider classifiers of the form:

$$p(y = 1|\boldsymbol{x}) = \psi(\boldsymbol{\beta}^T \boldsymbol{x}) \tag{1}$$

These classifiers first compute the inner product of a set of parameters with the feature vector to produce a score, then use a link function ψ to map that score to a probability estimate between 0 and 1. For a text classification problem, that probability can be viewed as an estimate of the probability that the document should be assigned to the category. A decision of whether to assign the category can be based on comparing the probability estimate with a threshold or, more generally, by computing the decision

that maximizes expected utility.

The logistic link function is a common choice for ψ, but we instead adopt the very similar probit model, $\psi(z) = \Phi(z)$, where Φ denotes the standard Gaussian distribution function. The probit link function leads to simpler Bayesian learning algorithms. See Chambers and Cox (1967) for a comparison of the logistic and probit models.

One of the simplest Bayesian approaches to the probit model (or the logistic model for that matter) is to impose a univariate normal (Gaussian) prior on each parameter β_j:

$$p(\beta_j|\sigma_j) = N(0, \sigma_j^2). \tag{2}$$

Here the σ_j are constants that must be specified. (In the simplest case we let σ_j equal the same σ for all j.) The overall prior for $\boldsymbol{\beta}$ is simply the product of the priors for each of its components. Finding the MAP estimate of $\boldsymbol{\beta}$ with this prior is essentially equivalent to ridge regression (Hoerl and Kennard, 1970) for the probit model.

This Gaussian prior, however, does not favor values of $\boldsymbol{\beta}$ that are sparse. To achieve that, we adopt a normal-exponential hierarchical prior for $\boldsymbol{\beta}$. As above, the prior for each β_j is Gaussian:

$$p(\beta_j|\tau_j) = N(0, \tau_j). \tag{3}$$

However, here the variance τ_j may be different for each β_j. More importantly, the τ_j themselves are given prior distributions, rather than being specified explicitly. One choice of a prior for τ_j is an exponential distribution. In the simplest case we use the same exponential distribution for each τ_j:

$$p(\tau_j|\gamma) = \frac{\gamma}{2} \exp(-\frac{\gamma}{2}\tau_j). \tag{4}$$

As with the Gaussian approach, we have a single parameter to specify by hand, in this case γ.

In fact, when an exponential prior is used for τ_j, and all we are concerned with is the distribution of β_j, we can actually integrate out τ_j. This leaves a Laplacian or double exponential prior on each β_j:

$$p(\beta_j|\gamma) = \frac{\sqrt{\gamma}}{2} \exp(-\sqrt{\gamma}|\beta_j|). \tag{5}$$

Again the overall prior for $\boldsymbol{\beta}$ is simply the product of the priors for each of its components. Tibshirani (1995) was the first to suggest Laplacian priors in the regression context. He pointed out that the maximum *a posteriori* (MAP) estimates using the Laplacian prior are the same as the estimates produced by the lasso algorithm. The above approach is very similar to that of Tipping (2001), except that they use a gamma prior in place of the exponential.

We note in passing that if we had domain knowledge suggesting which words are likely to be good predictors of a category, we could use separate γ_j's for each β_j to favor β_j's for high quality words having values farther from 0.

Yet another approach is that of Figueiredo and Jain (2001), again using a Gaussian prior for β_j, but adopting a Jeffreys' prior for τ_j:

$$p(\tau_j) \propto 1/\tau_j. \tag{6}$$

Note that the Laplacian prior requires a choice for the hyperparameter γ, which might be done using the training data (perhaps by cross-validation) or could be chosen based on experience with similar problems. The Jeffreys prior is more convenient, insofar as it has no tunable hyperparameters. However, we show in Appendix A that while the Laplacian distribution leads to a posterior distribution for $\boldsymbol{\beta}$ that is convex, the Jeffreys prior can lead to a non-convex posterior distribution. This means that we in general cannot find the posterior mode of the distribution of $\boldsymbol{\beta}$, only local optima. Since both approaches have their advantages, we will compare them in this paper.

3 Learning Algorithms - EM

Figueiredo and Jain (2001) proposed an Expectation Maximization (EM) algorithm (Dempster, *et al.*, 1977) for finding the mode of the posterior distribution of $\boldsymbol{\beta}$ for the probit model. The algorithm makes 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 $z_1, \ldots, z_n$, one for each training example, where z_i has a Gaussian distribution $N(\boldsymbol{\beta}^T \boldsymbol{x_i}, 1)$. 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 $p(y_i = 1) = \Phi(\boldsymbol{\beta}^T \boldsymbol{x_i}), i = 1, \ldots, n$, and we recover the standard probit model. Figure 1 shows the corresponding graphical Markov model using the BUGS plate notation (Spiegelhalter *et al.*, 1999).

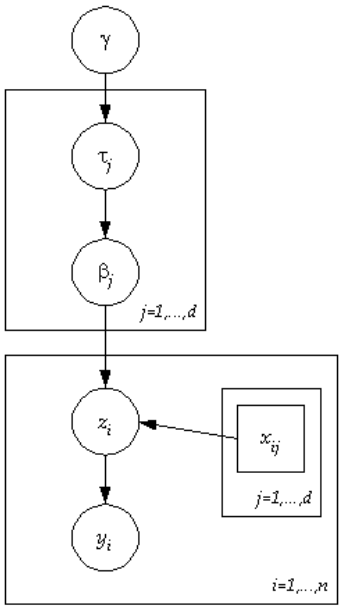

Figure 1: *Probit Model with Hierarchical Prior and Latent Variables.*

Observe that if $\boldsymbol{z} = [z_1, \ldots, z_i, \ldots, z_n]$ was known, we would have a normal (Gaussian) linear regression, rather than probit regression, problem. If $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_j, \ldots, \tau_d]$ was also known, we would have a known multivariate Gaussian prior for $\boldsymbol{\beta}$. For linear regression, a multivariate Gaussian prior on $\boldsymbol{\beta}$ leads to a closed form (in fact multivariate Gaussian) posterior distribution for $\boldsymbol{\beta}$ (Gelman *et al.*, 1995).

The difficulty is that $\boldsymbol{z}$ and $\boldsymbol{\tau}$ are latent rather than known. However, given the data y_i and a particular choice $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$, the z_i have a truncated normal distribution. Similarly, the distributions for the τ_j are known. This situation, where parameters of interest have a tractable distribution when latent variables are known, and latent variables have a tractable distribution when the parameter is known, enables the use of the Expectation Maximization (EM) algorithm to find the MAP estimate of the parameters. We now describe how the EM algorithm is applied to our models of interest.

3.1 Laplacian Prior

For the latent variable representation for probit under the Laplacian prior, both $\boldsymbol{z}$ and $\boldsymbol{\tau}$ are latent. The complete data log-posterior is:

$$\begin{aligned} \log p(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{\tau},\boldsymbol{z}) &\propto \log p(\boldsymbol{z}|\boldsymbol{\beta}) + \log p(\boldsymbol{\beta}|\boldsymbol{\tau}) \quad (7) \\ &\propto -\boldsymbol{\beta}^T\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{\beta} - 2\boldsymbol{z}) - \boldsymbol{\beta}^T\boldsymbol{\Gamma}\boldsymbol{\beta}, \quad (8) \end{aligned}$$

where $\boldsymbol{\Gamma} = \text{diag}(\tau_1^{-1}, \ldots, \tau_j^{-1}, \ldots, \tau_d^{-1})$ is the covariance matrix for the Gaussian prior on $\boldsymbol{\beta}$, and $\boldsymbol{X}$ is the design matrix whose rows are $\boldsymbol{x_1}^T, \ldots, \boldsymbol{x_i}^T, \ldots, \boldsymbol{x_n}^T$. Note the complete data log-likelihood is linear in $\boldsymbol{z}$.

The EM algorithm cycles through two steps. The first, or E-step, computes the new expected values for the latent variables given the current estimates of the model parameters. On iteration $t+1$ of the algorithm, the E-step computes $E[\tau_j^{-1}|\hat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{y}, \gamma]$ for each $\tau_j, j = 1, \ldots, d$, and computes $E[z_i|\hat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{y}, \gamma]$ for each $z_i, i = 1, \ldots, n$. Here $\hat{\boldsymbol{\beta}}^{(t)}$ is the estimate of $\boldsymbol{\beta}$ produced at step t of the computation.

The distribution of τ_j is:

$$p(\tau_j|\hat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{y}, \gamma) = p(\tau_j|\hat{\beta_j}^{(t)}, \gamma) \propto p(\hat{\beta_j}^{(t)}|\tau_j)p(\tau_j|\gamma). \quad (9)$$

Defining ω_j to be our desired expected value we then have:

$$\begin{aligned} \omega_j^{t+1} \equiv E[\tau_j^{-1}|\hat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{y}, \gamma] &= \frac{\int_0^\infty \frac{1}{\tau_j} p(\hat{\beta_j}^{(t)}|\tau_j)p(\tau_j|\gamma)d\tau_j}{\int_0^\infty p(\hat{\beta_j}^{(t)}|\tau_j)p(\tau_j|\gamma)d\tau_j} \\ &= \gamma|\hat{\beta_j}^{(t)}|^{-1}. \end{aligned}$$

The z_i have a Gaussian distribution with mean $\boldsymbol{\beta}^T\boldsymbol{x_i}$, but left-truncated at zero if $y_i = 1$ and right-truncated at zero of $y_i = -1$. Defining v_i to be the expected value of z_i we have:

$$v_i^{t+1} \equiv E[z_i|\hat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{y}, \gamma] = \begin{cases} \boldsymbol{\beta}^T\boldsymbol{x_i} + \frac{N(\boldsymbol{\beta}^T\boldsymbol{x_i})}{1-\Phi(-\boldsymbol{\beta}^T\boldsymbol{x_i})} & \text{if } y_i = 1 \\ \boldsymbol{\beta}^T\boldsymbol{x_i} + \frac{N(\boldsymbol{\beta}^T\boldsymbol{x_i})}{\Phi(-\boldsymbol{\beta}^T\boldsymbol{x_i})} & \text{if } y_i = -1. \end{cases} \quad (10)$$

Note that the class labels $\boldsymbol{y}$ affect the result through the expected value of $\boldsymbol{z}$. The E-step of the EM algorithm computes the above conditional expectations, which lead then to these conditional expectations for the covariance matrix $\boldsymbol{\Gamma}$ and latent variables $\boldsymbol{z}$:

$$E[\boldsymbol{\Gamma}|\hat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{y}, \gamma] = \boldsymbol{\Omega}^{t+1} = \text{diag}(\omega_1^{t+1}, \ldots, \omega_d^{t+1}) \quad (11)$$

$$E[\boldsymbol{z}|\hat{\boldsymbol{\beta}}^{(t)}, \boldsymbol{y}, \gamma] = \boldsymbol{v}^{t+1} = (v_1^{t+1}, \ldots, v_n^{t+1})^T \quad (12)$$

The M-step plugs the conditional expectations $\boldsymbol{\Omega}^{t+1}$ and $\boldsymbol{v}^{t+1}$ into the expression for the posterior probability of $\boldsymbol{\beta}$ (Equation 8). It then finds the value $\hat{\boldsymbol{\beta}}^{t+1}$ that maximizes this posteriori probability. The optimal value can be written in closed form as:

$$\hat{\boldsymbol{\beta}}^{(t+1)} = (\boldsymbol{\Omega}^{t+1} + \boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{v}^{t+1}. \quad (13)$$

The E- and M-steps are repeated until convergence, e.g. until successive values of $\hat{\boldsymbol{\beta}}^t$ differ by less than a specified tolerance. The convexity of the posterior distribution ensures that the algorithm converges to the posterior mode (McLachlan and Krishnan, 1996).

3.2 Jeffreys Prior

The above EM procedure can be used with the Jeffreys prior (Equation 6) as well. The steps are identical to those for the Laplace prior, except that:

$$\boldsymbol{\Omega} = \text{diag}(\omega_1, \ldots, \omega_d) = \text{diag}(|\hat{\boldsymbol{\beta}}_1^{(t)}|^{-2}, \ldots, |\hat{\boldsymbol{\beta}}_d^{(t)}|^{-2}). \quad (14)$$

However, because the posterior distribution of $\boldsymbol{\beta}$ under the Jeffreys prior is not convex, the EM procedure is gauranteed only to converge to a local optimum of the posterior distribution, not necessarily to the mode.

4 Results

We have conducted preliminary experiments using two standard test collections. The ModApte version of the Reuters−21578 collection of news stories[1] contains 7,769 documents in the training set and 3,019 in the testing set. We remove stopwords and punctuation marks before we do the analyses. The second test collection is the much larger Reuters Corpus Volume I - we discuss this below. Several published analyses of both datasets exist.

Table 1 and Figure 1 show the performance of the sparse Bayesian algorithm in comparison with results for Naive Bayes, logistic regression, and support vector machines (SVM) from Zhang and Oles (2001). The sparse Bayes results in this Table involved a preprocessing step that first selected a limited number features per category using a simple feature selection algorithm based on the absolute value of the Pearson correlation[2]:

$$r_{x^{(j)},y} = \frac{\sum_{i=1}^{n}(x_{ij} - \overline{x^{(j)}})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_{ij} - \overline{x^{(j)}})\sum_{i=1}^{n}(y_i - \overline{y})}, \tag{15}$$

where $x^{(j)} = [x_{1,j}, \ldots, x_{i,j}, \ldots, x_{n,j}]^T$ is the vector of TF values (see below) for the j-th term. The number of features for the reported experiments was between 300 and 1,000. Note that this potentially places us at a disadvantage when compared to the other algorithms' results shown in the Table, all of which used 10,000 features. Nonetheless, our algorithm is performing reasonably well with the exception of the "interest" catgeory. There was also a postprocessing step of tuning the threshold by minimizing the sum of errors (false positive plus false negative) on the training set.

Performance results here use the so-called "F1-measure," the harmonic mean of precision and recall (Lewis, 1995). Precision and recall are standard measures used in the text categorization literature:

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[2] Though not very popular in text categorization, this measure performed significantly better than measures based on binary data like chi-square or Yule's Q.

Topic	Naive Bayes	Log. Reg.	SVM	Sparse Bayes
earn	96.6	**98.4**	98.1	96.9
acq	91.7	95.2	**95.3**	90.3
money-fx	70.0	**75.2**	74.4	65.9
grain	76.7	88.4	89.6	**91.7**
crude	84.1	**85.9**	84.8	81.3
trade	52.3	72.9	73.4	**76.7**
interest	68.2	**78.2**	75.9	51.8
wheat	58.1	88.2	88.9	**89.6**
ship	76.4	81.9	**82.4**	78.8
corn	52.4	88.7	86.2	**90.9**
macro-average	72.6	**85.3**	84.9	81.4
micro-average	85.2	**91.4**	91.1	88.6

Table 1: F1 Performance results for the ModApte version of Reuters-21578. The Naive Bayes, logistic regression, and SVM results are Zhang & Oles' published results, using 10,000 features and their text processing. The Sparse Bayes (Laplace prior, $\gamma = 10$) results use our text processing, log TF weighting, and 300 features selected by Pearson's correlation. Bold-face indicates the top performer for each category.

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \times 100, \tag{16}$$

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \times 100. \tag{17}$$

We conducted a variety of experiments to explore different tunable aspects of the sparse Bayesian algorithm.

We compared two ways to define the weight of a term in a feature vector. Raw term frequency (TF) is simply the number of occurrences of a term in a document. Logarithmic TF is defined as $1 + \log(x)$, where x is raw TF. Table 2 shows the results. For most topics, using the logarithmic TF represenatation provides better F1 performance than using raw TF.

Topic	**Jeffreys raw TF**	**Jeffreys log TF**	**Laplace $\gamma = 10$ raw TF**	**Laplace $\gamma = 10$ log TF**
earn	96.6	97.1	96.9	96.9
acq	87.7	89.1	89.3	90.3
money-fx	55.0	59.7	59.3	65.9
grain	89.9	92.3	88.4	91.7
crude	77.7	81.1	82.4	81.3
trade	67.5	71.6	65.4	76.7
interest	49.0	43.9	49.5	51.8
wheat	89.7	89.7	87.9	89.6
ship	68.9	67.1	73.0	78.8
corn	88.2	90.3	83.6	90.9
macro-average	77.0	78.2	77.6	81.4
micro-average	86.0	87.4	87.1	88.6

Table 2: Role of term frequency measure (TF). Raw and log TF are compared for Jeffreys prior and Laplace priors with $\gamma = 10$. Results for the ModApte version of Reuters-21578.

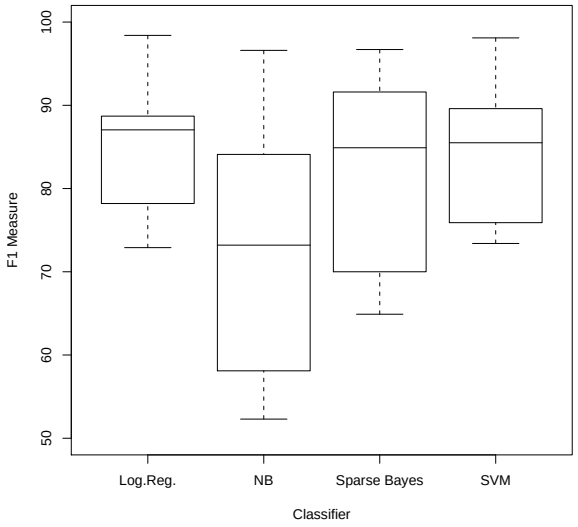

Figure 2: Boxplots of Table 1's results. Each boxplot shows, from top to bottom, the maximum, 75th percentile, median, 25th percentile, and minumum F1 measure across the 10 categories.

Table 3 compares different priors. One can see that for most classes the Laplacian prior outperforms the Jeffreys prior for a range of settings of the Laplacian hyperparameter γ.

Table 4 shows the sparsity achieved with different priors. Choosing, for example, $\gamma = 10$ results in substantial sparsity; anywhere from 27% (earn) to 86% (wheat) of the estimated regression co-efficients are zero. Nonetheless, the predictive performance remains competitive.

Table 5 looks at the value of increasing the number of input features from 300 to 1,000. It shows some moderate improvement for the Laplacian prior, while for the Jeffreys prior the impact is negligible.

It is interesting to make comparison with non-sparse models. We've done that with probit and logistic regression with a nonhierarchical Gaussian prior (Equation 2). We fit the probit regression with the Gaussian prior using the same EM algorithm used for sparse models. We fit the logistic regression model with the Gaussian prior using the column relaxation algorithm as described in Zhang and Oles (2001). Some

Topic	**Jeffreys**	**Laplace** $\gamma = 0.1$	**Laplace** $\gamma = 0.3$	**Laplace** $\gamma = 1$	**Laplace** $\gamma = 3$	**Laplace** $\gamma = 10$	**Laplace** $\gamma = 30$
earn	97.1	97.4	**97.5**	**97.5**	97.4	96.9	96.6
acq	89.1	90.5	90.0	90.4	**91.7**	90.3	89.0
money-fx	59.7	59.3	60.4	64.3	61.8	**65.9**	56.8
grain	**92.3**	88.9	88.1	90.6	90.7	91.7	90.0
crude	81.1	78.9	**82.3**	80.1	82.2	81.3	76.0
trade	71.6	66.1	67.2	70.5	73.8	**76.7**	69.5
interest	43.9	56.7	53.1	**62.7**	50.3	51.8	54.2
wheat	89.7	84.0	83.8	86.3	87.2	**89.6**	**89.6**
ship	67.1	59.7	65.3	71.9	75.8	**78.8**	76.3
corn	90.3	83.9	84.9	87.3	**91.4**	90.9	90.2
macro-average	78.2	76.5	77.3	80.2	80.2	**81.4**	78.8
micro-average	87.4	87.0	87.2	88.2	**88.7**	88.6	87.0

Table 3: Role of priors. Jeffreys prior compared with Laplace priors with different values of γ. Results for the ModApte version of Reuters-21578. Bold-face indicates the top performer(s) in each row.

Topic	**Jeffreys**	**Laplace** $\gamma = 0.1$	**Laplace** $\gamma = 0.3$	**Laplace** $\gamma = 1$	**Laplace** $\gamma = 3$	**Laplace** $\gamma = 10$	**Laplace** $\gamma = 30$
earn	55	301	300	288	256	220	186
acq	67	301	299	284	257	226	195
money-fx	32	200	291	276	235	177	111
grain	15	280	251	192	142	92	46
crude	25	282	266	205	174	90	62
trade	20	291	282	261	214	147	80
interest	22	294	270	249	205	136	85
wheat	3	275	203	127	90	41	14
ship	19	275	240	176	134	96	55
corn	6	259	232	170	110	47	17

Table 4: Sparsity achieved depending on type and parameter of prior. With 301 input parameters (300 selected features plus intercept), the table shows the number of parameters remaining (i.e., the number of parameters with non-zero posterior modes). Note that sparsity increases with γ, as expected. Results are for the ModApte version of Reuters-21578.

Topic	**Jeffreys 300 features**	**Jeffreys 1000 features**	**Laplace $\gamma = 10$ 300 features**	**Laplace $\gamma = 10$ 1000 features**
earn	97.1	96.8	96.9	**97.2**
acq	89.1	90.0	90.3	**92.2**
money-fx	59.7	63.7	65.9	**70.2**
grain	**92.3**	90.8	91.7	91.7
crude	81.1	76.6	81.3	**82.5**
trade	71.6	62.9	**76.7**	72.8
interest	43.9	56.0	51.8	**63.6**
wheat	**89.7**	**89.7**	89.6	88.9
ship	67.1	66.7	78.8	**79.7**
corn	90.3	90.3	**90.9**	90.0
macro-average	78.2	78.4	81.4	**82.9**
micro-average	87.4	87.3	88.6	**89.6**

Table 5: Role of the number of features. For each of Jeffreys prior and Laplace prior with $\gamma = 10$, two runs are compared: with 300 and 1,000 features selected by Pearson's correlation coefficient. Results for the ModApte version of Reuters-21578.

Topic	Probit Gaussian	Probit Laplace	Logistic Gaussian	Logistic Laplace
earn	96.8	96.9	97.6	**97.8**
acq	89.2	90.3	**94.0**	92.7
money-fx	65.1	65.9	**74.9**	64.2
grain	85.1	91.7	**91.8**	90.7
crude	82.9	81.3	85.7	**85.9**
trade	71.1	**76.7**	67.6	70.5
interest	58.1	51.8	**67.9**	67.3
wheat	89.0	**89.6**	87.7	85.7
ship	**83.3**	78.8	79.2	78.0
corn	86.7	**90.9**	90.4	90.4
macro-average	80.7	**81.4**	**83.7**	82.3
micro-average	87.9	**88.6**	**90.7**	89.8

Table 6: Comparing sparse and non-sparse binary regression with F1 measure. Gaussian priors used variance $\sigma^2 = 0.01$; Laplacian priors used $\gamma = 10$; logistic regression used IDF. All runs used 300 features selected by Pearson's correlation. Results for the ModApte version of Reuters-21578.

modification of this algorithm allowed us to implement logistic regression with the Laplacian prior. Table 6 shows the results. Perhaps surprisingly, the logistic model provides superior performance, at least for these data. Again, sparse models compared to non-sparse models show little decline in the effectiveness.

We have also performed experiments using the latest collection of news stories from Reuters, prepared and described by Lewis et al. (2003). This collection is significantly larger, it contains 23,149 documents in the training set and 781,265 in the testing set. We experimented with 101 "topics" categories, omitting the categories that have no positive examples in training. With this larger collection, the scalability properties of the column relaxation algorithm become important. We were thus able to increase the number of features for the logistic regression model up to 3,000. Table 7 shows average results and provides a comparison with several other methods reported in Lewis et al. (2003). Table 8 compares sparse and non-sparse binary regression results. Most observations made on the previous collection still hold here. The Laplace

	SVM	**kNN**	**Rocchio**	**Probit Jeffreys** *300 features*	**Probit Laplace** $\gamma = 25$ *300 features*	**Logistic Laplace** $\gamma = 25$ *3,000 features*
Macro-average	**61.9**	56.0	50.4	39.4	47.7	53.0
Micro-average	**81.6**	76.5	69.3	72.5	74.4	78.9

Table 7: Sparse Bayesian results compared to other methods reported in Lewis et al. (2003). SVM refers to Support Vector Machines. kNN refers to k-nearest neighbor. Results for the RCV1-v2 collection, 101 "topics" categories.

Method	**Probit**	**Probit**	**Probit**	**Logistic**	**Logistic**	**Logistic**
Priors	**Jeffreys**	**Laplace** $\gamma = 25$	**Gaussian** $\sigma^2 = 0.001$	**Laplace** $\gamma = 25$	**Laplace** $\gamma = 25$	**Gaussian** $\sigma^2 = 0.01$
Features	*300*	*300*	*300*	*300*	*3,000*	*3,000*
Macro-average	39.4	47.7	45.3	48.0	**53.0**	51.8
Micro-average	72.5	74.4	74.9	75.5	78.9	**79.7**

Table 8: Sparse and non-sparse binary regression results for the RCV1-v2 collection, 101 "topics" categories.

prior outperforms the Jeffreys prior. Effectiveness measures show little decline when moving from non-sparse to sparse models. Logistic regression has advantage over the probit even when the same number of features is used by both methods. Notice that in contrast with the sparse probit model, logistic regression performance improves significantly as the number of features increases.

5 Discussion and Future Work

Our experiments indicate that the sparse Bayesian approach is competitive with state-of-the-art techniques for text categorization. The logistic version of the model seems to outperform the probit version. Furthermore, tuning the hyperparameter γ and increasing the number of features improves performance.

We are currently investigating several topics related to this work. We are developing a block-EM algorithm (Meng and Rubin, 1993) that sequentially updates sub-vectors of $\boldsymbol{\beta}$. Note that Equation 13 requires inversion of a $d \times d$ matrix, a prohibitively expensive operation for large d. The block-EM algorithm instead requires inversion of several smaller matrices and can provide significant computational savings.

In many applications, labeled documents arrive sequentially creating a need for algorithms that learn in an online fashion. The goal of online learning in the sparse Bayesian classification context is to sequentially update the posterior distribution of the model parameters as each new labeled example arrives. The Bayesian paradigm supports this operation in a natural fashion; starting from the prior, the first example produces a posterior distribution incorporating the evidence from the first example. This then becomes the prior distribution awaiting the arrival of the second example, and so on. In practice, however, except in those cases where the posterior distribution has the same mathematical form as the prior distribution, some form of approximation is required to carry out the sequential updating. We are currently investigating a number of approaches.

Other related research topics include automated hyperparameter selection, embeding the logisitic and probit link functions in a general family, and incorporation of external knowledge via prior distributions.

Appendix A: Convexity

Consider first the following setting

$$\begin{aligned} y_i|\beta, x_i &\sim \text{Bern}(\Phi(x_i^T\beta)) \;\; i = 1, ..., n \\ \beta_j &\sim N(0, \sigma^2) \;\; j = 1, ..., d \end{aligned}$$

where $\Phi(x)$ denotes the standard normal cumulative distribution. The posterior probability for β and its log partial first and second order derivatives are given by

$$p(\beta|y) = \frac{p(y|\beta)p(\beta)}{p(y)}$$

$$
\begin{aligned}
&\propto p(y|\beta)p(\beta) \\
&= \Pi_{i=1}^{n}\Phi(x_i^T\beta)^{y_i}\Phi(-x_i^T\beta)^{1-y_i}\Pi_{j=1}^{d}\frac{e^{-\frac{\beta_j^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \\
\log(p(y|\beta)p(\beta)) &= \sum_{i=1}^{n}\{y_i\log(\Phi(x_i^T\beta)) + (1-y_i)\log(\Phi(-x_i^T\beta))\} - \sum_{j=1}^{d}\frac{\beta_j^2}{2\sigma^2} \\
\frac{\partial\log(p(\beta|y))}{\partial\beta_j} &= \sum_{i=1}^{n}\{x_{ij}y_i\frac{\phi(x_i^T\beta)}{\Phi(x_i^T\beta)} - x_{ij}(1-y_i)\frac{\phi(-x_i^T\beta)}{\Phi(-x_i^T\beta)}\} - \frac{\beta_j}{\sigma^2} \\
\frac{\partial^2\log(p(\beta|y))}{\partial\beta_j^2} &= \sum_{i=1}^{n}\{x_{ij}^2y_i\frac{\phi'(x_i^T\beta)\Phi(x_i^T\beta) - \phi^2(x_i^T\beta)}{\Phi^2(x_i^T\beta)} \\
&\quad + x_{ij}^2(1-y_i)\frac{\phi'(-x_i^T\beta)\Phi(-x_i^T\beta) - \phi^2(-x_i^T\beta)}{\Phi^2(-x_i^T\beta)}\} - \frac{1}{\sigma^2} \\
\frac{\partial^2\log(p(\beta|y))}{\partial\beta_j\partial\beta_k} &= \sum_{i=1}^{n}\{x_{ij}x_{ik}y_i\frac{\phi'(x_i^T\beta)\Phi(x_i^T\beta) - \phi^2(x_i^T\beta)}{\Phi^2(x_i^T\beta)} \\
&\quad + x_{ij}x_{ik}(1-y_i)\frac{\phi'(-x_i^T\beta)\Phi(-x_i^T\beta) - \phi^2(-x_i^T\beta)}{\Phi^2(-x_i^T\beta)}\}
\end{aligned}
$$

Define,

$$
r_{jk} = \sum_{i=1}^{n}\{x_{ij}x_{ik}y_i\frac{\phi'(x_i^T\beta)\Phi(x_i^T\beta) - \phi^2(x_i^T\beta)}{\Phi^2(x_i^T\beta)} + x_{ij}x_{ik}(1-y_i)\frac{\phi'(-x_i^T\beta)\Phi(-x_i^T\beta) - \phi^2(-x_i^T\beta)}{\Phi^2(-x_i^T\beta)}\}
$$

then

$$
z^T\frac{\partial p(\beta|y)}{\partial\beta}z = \sum_{k=1}^{d}z_k^2\sum_{j=1}^{d}r_{jk} - \frac{1}{\sigma^2}\sum_{k=1}^{d}z_k^2 \tag{18}
$$

and any solution of the equation $\frac{\partial\log(p(\beta|y))}{\partial\beta} = 0$ is a maximum of the posterior probability if $z^T\frac{\partial p(\beta|y)}{\partial\beta}z < 0$ for any β and z. It is easy to see that r_{ij} are always negative. Let $w_i = x_i^T\beta$

$$
r_{jk} = \sum_{i=1}^{n}\{x_{ij}x_{ik}y_if(w_i) + x_{ij}x_{ik}(1-y_i)f(-w_i)\} \tag{19}
$$

where $f(w_i) = \frac{\phi'(w_i)\Phi(w_i) - \phi^2(w_i)}{\Phi^2(w_i)}$ and is always negative. Figure 1 shows a graph of this function. For a detailed proof of the concavity of the posterior distribution see Appendix A.

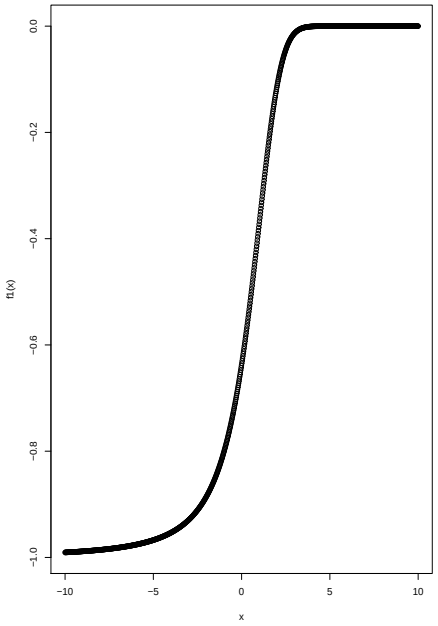

Figure 3: Plot of the f function.

Let's look at a hierarchical model now where

$$\begin{aligned} y_i|\beta, x_i &\sim \text{Bern}(\Phi(x_i^T\beta)) \;\; i = 1, ..., n \\ \beta_j|\tau_j &\sim N(0, \tau_j) \;\; j = 1, ..., d \\ \tau_j &\sim 1/\tau_j \;\; j = 1, ..., d. \end{aligned}$$

Then the marginal distribution for β is given by

$$\begin{aligned} p(\beta) &= \int_0^\infty p(\beta|\tau)p(\tau)d\tau = \int_0^\infty \Pi_{j=1}^d p(\beta_j|\tau_j)p(\tau_j)d\tau_j \\ &= \Pi_{j=1}^d \int_0^\infty p(\beta_j|\tau_j)p(\tau_j)d\tau_j = \Pi_{j=1}^d \frac{1}{|\beta_j|} \end{aligned}$$

and the posterior distribution with its log partial first and second order derivatives are given by

$$
\begin{aligned}
p(\beta|y) &\propto p(y|\beta)p(\beta) \\
&= \Pi_{i=1}^{n}\Phi(x_i^T\beta)^{y_i}\Phi(-x_i^T\beta)^{1-y_i}\Pi_{j=1}^{d}\frac{1}{|\beta_j|} \\
\log(p(y|\beta)p(\beta)) &= \sum_{i=1}^{n}\{y_i\log(\Phi(x_i^T\beta)) + (1-y_i)\log(\Phi(-x_i^T\beta))\} - \sum_{j=1}^{d}\log(|\beta_j|) \\
\frac{\partial\log(p(\beta|y))}{\partial\beta_j} &= \sum_{i=1}^{n}\{x_{ij}y_i\frac{\phi(x_i^T\beta)}{\Phi(x_i^T\beta)} - x_{ij}(1-y_i)\frac{\phi(-x_i^T\beta)}{\Phi(-x_i^T\beta)}\} - \frac{1}{\beta_j} \\
\frac{\partial^2\log(p(\beta|y))}{\partial\beta_j^2} &= \sum_{i=1}^{n}\{x_{ij}^2y_i\frac{\phi'(x_i^T\beta)\Phi(x_i^T\beta) - \phi^2(x_i^T\beta)}{\Phi^2(x_i^T\beta)} \\
&\quad + x_{ij}^2(1-y_i)\frac{\phi'(-x_i^T\beta)\Phi(-x_i^T\beta) - \phi^2(-x_i^T\beta)}{\Phi^2(-x_i^T\beta)}\} + \frac{1}{\beta_j^2} \\
\frac{\partial^2\log(p(\beta|y))}{\partial\beta_j\partial\beta_k} &= \sum_{i=1}^{n}\{x_{ij}x_{ik}y_i\frac{\phi'(x_i^T\beta)\Phi(x_i^T\beta) - \phi^2(x_i^T\beta)}{\Phi^2(x_i^T\beta)} \\
N \qquad &\quad + x_{ij}x_{ik}(1-y_i)\frac{\phi'(-x_i^T\beta)\Phi(-x_i^T\beta) - \phi^2(-x_i^T\beta)}{\Phi^2(-x_i^T\beta)}\}
\end{aligned}
$$

Note that the partial derivatives can only be evaluated in all $\beta \in \Re^d$ such that none of the components of β is equal to zero. Therefore, if the posterior probability contains a maximum at any of such points, we won't be able to find it by using derivatives.

In this case we have

$$
z^T\frac{\partial p(\beta|y)}{\partial\beta}z = \sum_{k=1}^{d}z_k^2\sum_{j=1}^{d}r_{jk} + \sum_{j=1}^{d}\frac{z_j^2}{\beta_j^2}, \tag{20}
$$

where the first component on the right side of the equal sign is always negative and the second one is always positive. Therefore we cannot say, as in the previous case, that any solution of the equation $\frac{\partial\log(p(\beta|y))}{\partial\beta} = 0$ is a maximum.

A third approach is to consider

$$
\begin{aligned}
y_i|\beta, x_i &\sim \text{Bern}(\Phi(x_i^T\beta)) \;\; i = 1, ..., n \\
\beta_j|\tau_j &\sim N(0, \tau_j) \;\; j = 1, ..., d
\end{aligned}
$$

$$\tau_j|\gamma \sim \frac{\gamma}{2}e^{-\frac{\gamma}{2}\tau_j} \quad j = 1, ..., d$$

In the same way we compute the marginal distribution for β and the posterior distribution with its log partial first and second order derivatives:

$$p(\beta|\gamma) = \Pi_{j=1}^{d}\frac{\sqrt{\gamma}}{2}e^{\sqrt{\gamma}|\beta_j|} \tag{21}$$

$$\begin{aligned}
p(\beta|y) &\propto p(y|\beta)p(\beta|\gamma) \\
&= \Pi_{i=1}^{n}\Phi(x_i^T\beta)^{y_i}\Phi(-x_i^T\beta)^{1-y_i}\Pi_{j=1}^{d}\frac{\sqrt{\gamma}}{2}e^{-\sqrt{\gamma}|\beta_j|} \\
\log(p(y|\beta)p(\beta)) &= \sum_{i=1}^{n}\{y_i\log(\Phi(x_i^T\beta)) + (1-y_i)\log(\Phi(-x_i^T\beta))\} - \sqrt{\gamma}\sum_{j=1}^{d}|\beta_j| \\
\frac{\partial\log(p(\beta|y))}{\partial\beta_j} &= \sum_{i=1}^{n}\{x_{ij}y_i\frac{\phi(x_i^T\beta)}{\Phi(x_i^T\beta)} - x_{ij}(1-y_i)\frac{\phi(-x_i^T\beta)}{\Phi(-x_i^T\beta)}\} - \sqrt{\gamma}I(\beta_j > 0) + \sqrt{\gamma}I(\beta_j < 0) \\
\frac{\partial^2\log(p(\beta|y))}{\partial\beta_j\partial\beta_k} &= \sum_{i=1}^{n}\{x_{ij}x_{ik}y_i\frac{\phi'(x_i^T\beta)\Phi(x_i^T\beta) - \phi^2(x_i^T\beta)}{\Phi^2(x_i^T\beta)} \\
&\quad + x_{ij}x_{ik}(1-y_i)\frac{\phi'(-x_i^T\beta)\Phi(-x_i^T\beta) - \phi^2(-x_i^T\beta)}{\Phi^2(-x_i^T\beta)}\}
\end{aligned}$$

In this case we have that

$$z^T\frac{\partial p(\beta|y)}{\partial\beta}z = \sum_{k=1}^{d}z_k^2\sum_{j=1}^{d}r_{jk} \tag{22}$$

Therefore, we have, as in the first case, that any solution of the equation $\frac{\partial\log(p(\beta|y))}{\partial\beta} = 0$ is a maximum.

In the first and third cases we can be certain that the solution that the EM or ECM algorithm find is the MAP, but not in the second case.

Acknowledgements

We are grateful to Bill DuMouchel, Colin Mallows, and Ilya Muchnik for helpful discussions. The work of Genkin, Lewis and Madigan was partially supported under funds provided by the KD-D group for a project at DIMACS on Monitoring Message Streams, funded through National Science Foundation grant EIA-0087022 to Rutgers University. The NSF also partially supported Madigan and Lewis's work through an ITR grant.

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.

Chai, K.M.A., Chieu, K.L., and Ng, H.T. (2002). Bayesian online classifiers for text classification and filtering ages.In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 97 - 104.

Chambers, E.A. and Cox, D.R. (1967). Discrimination between alternative binary response models. *Biometrika*, **54**, 573–578.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum lilelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society (Series B)*, **39**, 1–38.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2003). Least angle regression. *Annals of Statistics*, to appear.

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.

Fu, W.J. (1988). Penalized Regressions: The Bridge Versus the Lasso. *Journal of*

Computational and Graphical Statistics, **7**, 397–418.

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

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

Hoerl, A.E. and Kennard, R.W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, **12**, 55–67.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features Proceedings of ECML-98, 10th European Conference on Machine Learning, 137–142.

Ju, W., Madigan, D., and Scott, S. (2002). On sparse Bayesian classifiers. DIMACS Technical Report xxx, `http://www.stat.rutgers.edu/ madigan/PAPERS/sparse3.pdf`

Lewis, D.D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In: *ECML'98, The Tenth European Conference on Machine Learning*, 4–15.

Lewis, D.D. (2002). TREC.

Lewis, D. D., Yang, Y., Rose, T., Li, F. (2003). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, to appear.

McLachlan, G.J. and Krishnan, T. (1996). *The EM algorithm and extensions*, Wiley.

Meng, X.L. and Rubin, D.R. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika*, **80**, 267–278.

Neal, R.M. and Hinton, G.E. (1998). A new view of the EM algorithm that justifies incremental, sparse and other variants. In: *Learning in Graphical Models*, M.I. Jordan (Editor). Kluwer Academic Publishers, 355–368.

Opper, M. (1998). A Bayesian approach to online learning. In: *On-Line Learning in Neural Networks*, D. Saad (Editor, Cambridge University Press.

Osborne, M.R., Presnell, B., and Turlach, B.A. (2000). On the LASSO and its Dual. *Journal of Computational and Graphical Statistics*, **9**,319–338.

Ridgeway, G. and Madigan, D. (2003). A sequential Monte Carlo Method for Bayesian analysis of massive datasets. In *Journal of Data Mining and Knowledge Discovery*, to appear.

Rocchio, J. (1971). Relevance feedback information retrieval. In: Gerard Salton, editor, The Smart Retrieval System-Experiments in Automatic Document Processing. Prentice-Hall, 313–323.

Sato, M. and Ishii, S. (2000). On-line EM algorithm for the normalized gaussian network. *Neural Computation*, **12**, 407–432.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**, 1–47.

Smith, A.F.M. and Makov, U. (1978). A quasi-Bayes sequential procedure for mixtures. *Journal of the Royal Statistical Society (Series B)*, **40**, 106–112.

Smith, R.L. (1999). Bayesian and frequentist approaches to parametric predictive inference (with discussion). In: *Bayesian Statistics 6*, edited by J.M. Bernardo, J.O. Berger, A.P. Dawid and A.F.M. Smith. Oxford University Press, pp. 589-612.

Spiegelhalter, D.J., Thomas, A., and Best, N.G. (1999). *WinBUGS Version 1.2 User Manual*. MRC Biostatistics Unit.

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

Tipping, M.E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, **1**, 211-244.

Titterington, D.M. (1984). Recursive parameter estimation using incomplete data, *Journal of the Royal Statistical Society (Series B)*, **46**, 257–267.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization and retrieval. *Information Retrieval Journal*, **1**, 69–90.

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