

---

# Feature selection via sparse non-parametric Bayesian learning

---

Vikas C. Raykar

VIKAS.RAYKAR@SIEMENS.COM

CAD and Knowledge Solutions (IKM CKS), Siemens Healthcare, Malvern, PA 19355 USA

Linda H. Zhao

LZHAO@WHARTON.UPENN.EDU

Department of Statistics, University of Pennsylvania, Philadelphia, PA 19104 USA

## Abstract

One of the widely used methods to select features for classification consists of computing a score for each feature and ranking them in the order of decreasing relevance to the class label. Only the top few relevant features are used and the rest are discarded. The number of features to retain is often based on ad-hoc rules and/or domain knowledge. An important issue which we address in this paper is how to set this threshold between the relevant and irrelevant features. We propose a completely automatic sparse nonparametric Bayesian method to accurately estimate the number of relevant features. The novelty of the method is that we use the data to implicitly determine the prior. It shows much superior performance compared with Bayesian methods which use a parametric prior. Experiments on simulated and publicly available datasets show that the relevant features identified agree with the optimal number of features selected by conventional expensive cross-validation routines. The proposed method can also be used in other contexts especially in denoising applications in signal processing.

## 1. Introduction

In a typical two-class classification scenario we are given a training set  $\mathcal{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^N$  containing  $N$  instances, where  $\mathbf{x}_j \in \mathbb{R}^d$  is an instance (the  $d$ -dimensional feature vector) and  $y_j \in \mathcal{Y} = \{0, 1\}$  is the corresponding known class label. The task is to learn

a classification function  $f : \mathbb{R}^d \rightarrow \mathcal{Y}$  which generalizes well on unseen datasets.

During the past few decades it has become relatively easy to gather datasets with a huge amount of features. A few such examples include gene expression array, astronomical databases, internet databases, text documents, financial records, audio and video data. In such situations very often we would like the discriminant function  $f$  to use as few features as possible that are most predictive without any appreciable decrease in accuracy. Feature selection is very often beneficial for cost effectiveness and interpretability. In many situations it also increases the prediction accuracy by preventing over-fitting.

While many sophisticated methods have been proposed for feature selection (see Guyon and Elisseeff (2003) for a review), one of the earliest and the most widely used algorithms is *feature ranking*. This is a very simple and scalable method and has had considerable empirical success either as a stand-alone feature selection mechanism or as a pre-processing step for other methods.

Essentially, for each feature a score measuring the degree of relevance is computed. For example one commonly used score is a measure of the correlation between the feature and the class label. The features are then ranked in the order of decreasing scores. Only the top most relevant features are used and the rest are discarded. The number of features to retain is often based on *ad-hoc rules* and *domain knowledge*, or on *cross-validation* which can be computationally demanding for datasets with huge numbers of features. Classifiers built using ranked scores and primarily ad hoc selection rules can be found for example in (Golub et al., 1999; Furey et al., 2000; Slonim et al., 2000).

An important issue which we address in this paper is *how to set the threshold between the relevant and irrelevant features*. We propose a completely automatic

sparse non-parametric Bayesian method to accurately estimate the number of relevant features.

## 2. Feature relevance ranking

Let  $z_i$ ,  $i = 1, \dots, d$  be the computed ranking criterion for the  $i^{\text{th}}$  feature. Various different ranking scores have been used in different application domains (Guyon & Elisseeff, 2003). Commonly used scores are related with Fisher’s criterion or the T-test criterion but the form may vary. For example  $z_i$  can be a scaled difference between means among two classes.

$$z_i = \frac{m_i^+ - m_i^-}{\sqrt{\frac{(\sigma_i^+)^2}{N^+} + \frac{(\sigma_i^-)^2}{N^-}}}, \quad (1)$$

where  $m_i^+$  and  $m_i^-$  are the means,  $(\sigma_i^+)^2$  and  $(\sigma_i^-)^2$  are the variances, and  $N^+$  and  $N^-$  are the number of examples of the positive and negative class respectively. Note that if a feature is irrelevant then  $z_i$  will be very close to zero. Typically the scores  $z_i$  are sorted in the decreasing order of their relevance and the top (say  $t$ ) features are considered relevant and used for further analysis or classifier training. The problem we address in this paper is how to choose  $t$  automatically.

We will assume that each  $z_i$  is a noisy realization of some underlying  $\mu_i$ , *i.e.*,

$$z_i = \mu_i + \epsilon_i, \quad (2)$$

where the  $\epsilon_i$  are independent and identically distributed as  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ , *i.e.*, a normal distribution with mean zero and a known variance  $\sigma^2$  (which can be well estimated when it is unknown). The normality assumption of the score  $z_i$  is valid among most commonly used scores. Even though the features are not necessarily independent it is a reasonable assumption in the context of feature ranking methods. It has also been noticed that for datasets with a large number of features treating features independently give us as good a classifier or even better ones (See for example (Domingos & Pazzani, 1997; Lewis, 1998; Bickel & Levina, 2004)).

Note that if a feature is irrelevant then the corresponding  $\mu_i$  is zero. Relevant features have  $\mu_i \neq 0$ . Based on the observation  $\mathbf{z} = (z_1, z_2, \dots, z_d)$  we need to find a desirable estimate  $\hat{\boldsymbol{\mu}}$  of the unknown parameters  $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_d)$  and to determine how many of them are really zero.

## 3. Sparse non-parametric Bayesian learning

A Bayesian solution to the above model can be obtained by using a sparsity promoting prior on  $\mu_i$ . Sparse Bayesian Learning (SBL) (Tipping, 2001; Wipf et al., 2004) has proved to be very successful in regression and classification problems. A sparsity promoting parametric prior (often a zero mean Gaussian or Laplace) with unknown parameters is usually used for the weights which control the influence of each feature. The unknown parameters in the prior are then estimated through empirical Bayesian methods.

While SBL has shown considerable success, it is not very robust because of its specific assumption on the shape of the prior. In this paper we propose a Sparse Nonparametric Bayesian Learning (SNBL) method. Specifically, to promote sparsity we use a mixture prior on each  $\mu_i$  with an atom of probability  $w$  at zero (for the irrelevant features) and a completely unspecified density for the nonzero (relevant features) part (see §4.1). The prior used for the relevant features is completely nonparametric, *i.e.*, there is no specific functional form. We show that the non-zero part of the prior is only involved through the marginal density of the observations. A weighted non-parametric kernel density is used to estimate the marginal (see §5.2).

The mixing probability  $w$  which determines the amount of sparsity (or the fraction of irrelevant features) is considered as a hyperparameter and is estimated using an empirical Bayes approach (see §5.1). The posterior of  $\boldsymbol{\mu}$  is computed by plugging in the estimated hyperparameter and the posterior mean ( $\hat{\boldsymbol{\mu}}$ ) is used as a point estimate of  $\boldsymbol{\mu}$  (see §5.4). The final feature selection algorithm consists of sorting  $|\hat{\boldsymbol{\mu}}|$  and retaining the top  $(1 - \hat{w})d$  features (see §6).

In most situations the prior information we have is that some features are irrelevant. We do not know how many of them are irrelevant. We also do not have any prior information about the distribution of the true scores ( $\boldsymbol{\mu}$ ) for the relevant features. The proposed SNBL algorithm addresses both these issues—(1) It adapts to the sparsity of the problem by estimating  $w$  which determines the number of irrelevant features and (2) produces a completely data-guided prior which allows us to avoid selecting a specific prior family for the nonzero part of the mixture prior. We demonstrate the advantage of this approach by comparing it with an approach which uses a parametric prior (normal) for the non-zero part (see §8).

Simulation results (§8) show that the SNBL method adapts well to the sparsity. Experiments on six pub-

licly available datasets (see §9) show that the relevant features identified through SNBL agree with the optimal number of features identified by conventional expensive cross-validation routines.

## 4. The Bayesian setup

Without loss of generality we will assume that the  $z_i$  are scaled such that  $\sigma^2 = 1$ . From (2) we have  $p(z_i|\mu_i) = \mathcal{N}(z_i|\mu_i, 1)$ . Since  $\epsilon_i$  are independent the likelihood of the parameters  $\mu$  given the observations  $\mathbf{z}$  can be factored as

$$p(\mathbf{z}|\mu) = \prod_{i=1}^d p(z_i|\mu_i) = \prod_{i=1}^d \mathcal{N}(z_i|\mu_i, 1). \quad (3)$$

### 4.1. Mixture Prior

We assume that  $\mu_i$  comes from a mixture of a delta function with mass at zero and a completely unspecified nonparametric density  $\gamma$ , *i.e.*,

$$p(\mu_i|w, \gamma) = w\delta(\mu_i) + (1-w)\gamma(\mu_i), \quad (4)$$

where  $w \in [0, 1]$  is the mixture parameter and the  $\delta(\mu_i)$  is defined as having a probability mass of 1 at  $\mu_i = 0$  and zero elsewhere. This prior captures our belief that some of the  $\mu_i$ 's, are exactly zero. The mixing parameter  $w$  is the fraction of zeros in  $\mu$  and corresponds exactly to the number of irrelevant features. We treat  $w$  as a *hyperparameter* and estimate it using an empirical Bayes approach.

For the nonzero part of the prior  $\gamma$  we will leave it completely unspecified. We will later show that the non-zero part of the prior is only involved through the marginal density of the observations. A weighted non-parametric kernel density estimate is used to estimate the marginal.

### 4.2. Posterior

Given the hyper-parameter  $w$  and the prior  $\gamma$  the posterior of  $\mu$  given the data  $\mathbf{z}$  can be written as

$$p(\mu|\mathbf{z}, w, \gamma) = \frac{\prod_{i=1}^d p(z_i|\mu_i)p(\mu_i|w, \gamma)}{m(\mathbf{z}|w, \gamma)}, \quad \text{where} \quad (5)$$

$$m(\mathbf{z}|w, \gamma) = \prod_{i=1}^d \int p(z_i|\mu_i)p(\mu_i|w, \gamma)d\mu_i \quad (6)$$

is the marginal density of the data given the hyper-parameters. For the likelihood (3) and the mixture prior (4) we have

$$\int p(z_i|\mu_i)p(\mu_i|w, \gamma)d\mu_i = w\mathcal{N}(z_i|0, 1) + (1-w)g(z_i), \quad (7)$$

where

$$g(z_i) = \int \mathcal{N}(\mu_i|z_i, 1)\gamma(\mu_i)d\mu_i. \quad (8)$$

Note  $g$  is the marginal density of the non-zero  $\{z_i\}$ . The posterior in (5) can now be factored as follows  $p(\mu|\mathbf{z}, w, \gamma) = \prod_{i=1}^d p(\mu_i|z_i, w, \gamma)$ , where

$$p(\mu_i|z_i, w, \gamma) = \frac{w\delta(\mu_i)\mathcal{N}(z_i|0, 1) + (1-w)\gamma(\mu_i)\mathcal{N}(z_i|\mu_i, 1)}{w\mathcal{N}(z_i|0, 1) + (1-w)g(z_i)}. \quad (9)$$

Define

$$\tilde{p}_i = \frac{w\mathcal{N}(z_i|0, 1)}{w\mathcal{N}(z_i|0, 1) + (1-w)g(z_i)}. \quad (10)$$

Then

$$p(\mu_i|z_i, w, \gamma) = \tilde{p}_i\delta(\mu_i) + (1-\tilde{p}_i)G(\mu_i), \quad (11)$$

where

$$G(\mu_i) = \frac{\mathcal{N}(\mu_i|z_i, 1)\gamma(\mu_i)}{\int \mathcal{N}(\mu_i|z_i, 1)\gamma(\mu_i)d\mu_i}. \quad (12)$$

Notice that  $\tilde{p}_i$  is the posterior probability of  $\mu_i$  being 0 and  $G(\mu_i)$  is the posterior density of  $\mu_i$  when it is not 0, *i.e.*,

$$p(\mu_i = 0|z_i, w, \gamma) = \tilde{p}_i. \quad (13)$$

$$p(\mu_i|z_i, w, \gamma, \mu_i \neq 0) = G(\mu_i). \quad (14)$$

## 5. The empirical Bayesian approach

The hyperparameter  $w$  is estimated by maximizing the marginal likelihood. The posterior of  $\mu$  is then computed by plugging in the estimated  $\hat{w}$  and the mean of the posterior is used as a point estimate of  $\mu$ .

### 5.1. Iterative Type II MLE for $w$

The hyperparameter  $w$  is the fraction of zeros in  $\mu$ . It directly determines the number of irrelevant features. We use an empirical Bayes approach and chose  $w$  to maximize the marginal likelihood—which is the likelihood integrated over the model parameters.

$$\hat{w} = \arg \max_w m(\mathbf{z}|w, g) = \arg \max_w \log m(\mathbf{z}|w, g). \quad (15)$$

This is also known as the Type II maximum likelihood estimator in Bayesian literature (Berger, 1985). From (6) and (7) the log-marginal can be written as

$$\log m(\mathbf{z}|w, \gamma) = \sum_{i=1}^d \log [w\mathcal{N}(z_i|0, 1) + (1-w)g(z_i)], \quad (16)$$

where  $g$  is the marginal density of the non-zero  $\{z'_i s\}$ . Note that  $\gamma$ , the prior for the non-zero part is only involved through the marginal  $g(z_i) = \int \mathcal{N}(\mu_i|z_i, 1)\gamma(\mu_i)d\mu_i$ . If we can estimate  $g(z_i)$  directly then we do not have to specify any prior for the non-zero part.

However  $g$  is the marginal of the non-zero part. So to estimate  $g$  we need to know whether each  $z_i$  comes from the zero part or the non-zero part. We will use an alternate optimization procedure to estimate  $w$  and  $g$ . Given  $g$  we maximize (16) numerically to estimate  $w$ . Once  $w$  is chosen  $g$  will be re-estimated through a weighted kernel estimator as described below. This process is repeated till convergence.

### 5.2. Weighted kernel estimate of the marginal

In order to maximize (16) we need an estimate of  $g$ —the marginal density of the non-zero  $\{z'_i s\}$ . If we knew which  $\{z'_i s\}$  are non-zero we can estimate  $g$  through a kernel density estimator (Wand & Jones, 1995). We propose to use a *weighted* non-parametric kernel density estimate  $\hat{g}$  of the following form

$$\hat{g}(z) = \frac{1}{\tilde{d}h} \sum_{j=1}^d (1 - \tilde{p}_j) K\left(\frac{z - z_j}{h}\right), \quad (17)$$

where  $K$  is a function satisfying  $\int K(x)dx = 1$  called the *kernel*,  $h$  is a positive number called the *bandwidth* of the kernel, and  $\tilde{d} = \sum_{j=1}^d (1 - \tilde{p}_j)$  is the effective sample size. We have weighted each point by  $1 - \tilde{p}_i$ , where  $\tilde{p}_i$  is the posterior probability of  $\mu_i$  being 0 and is computed using (10), where we use the estimate  $\hat{g}(z_i)$  from the last iteration. The most widely used kernel is a normal density of zero mean and unit variance, *i.e.*,  $K(x) = \mathcal{N}(x|0, 1)$ . We set the bandwidth of the kernel using the normal reference rule (Wand & Jones, 1995) as  $h = O(d^{-1/5})$ .

### 5.3. Iterative algorithm

The final algorithm consists of the following three steps which are repeated till convergence.

- (1) Given the current estimate  $\hat{g}(z_i)$  maximize the log-marginal (16) numerically to obtain  $\hat{w}$ .
- (2) Compute  $\tilde{p}_i$  using the current estimate  $\hat{w}$  and  $\hat{g}(z_i)$ .

$$\tilde{p}_i = \frac{\hat{w}\mathcal{N}(z_i|0, 1)}{\hat{w}\mathcal{N}(z_i|0, 1) + (1 - \hat{w})\hat{g}(z_i)}. \quad (18)$$

- (3) Re-estimate  $\hat{g}(z_i)$  using the current estimate of  $\tilde{p}_i$ .

$$\hat{g}(z_i) = \frac{1}{\tilde{d}h} \sum_{j=1}^d (1 - \tilde{p}_j) K\left(\frac{z_i - z_j}{h}\right). \quad (19)$$

The optimization procedure can be initialized by setting  $\tilde{p}_i = 0$ .

### 5.4. Posterior mean

The estimated hyperparameter  $\hat{w}$  can be plugged into the posterior, and we use the mean of the posterior as a point estimate for  $\mu$ .

$$\hat{\mu}_i = (1 - \tilde{p}_i)E_G[\mu_i]. \quad (20)$$

The mean of the posterior also depends only on the marginal and its derivative. Notice that

$$E_G[\mu_i] = \frac{\int \mu_i \mathcal{N}(\mu_i|z_i, 1)\gamma(\mu_i)d\mu_i}{\int \mathcal{N}(\mu_i|z_i, 1)\gamma(\mu_i)d\mu_i} = z_i + \frac{g'(z_i)}{g(z_i)}. \quad (21)$$

The marginal  $g$  and its derivative  $g'$  are both estimated through a weighted kernel density estimate. The kernel density derivative estimate is given by

$$\hat{g}'(z) = \frac{1}{\tilde{d}h^2} \sum_{j=1}^d (1 - \tilde{p}_j) K'\left(\frac{z - z_j}{h}\right). \quad (22)$$

## 6. SNBL feature selection algorithm

The final algorithm is summarized as follows—

- (1) Compute  $\hat{w}$  by the iterative algorithm in § 5.3.
- (2) Plug in  $\hat{w}$  to compute the posterior mean  $\hat{\mu}$  (§5.4).
- (3) Sort  $|\hat{\mu}|$  and retain the top  $(1 - \hat{w})d$  features.

## 7. Parametric mixture prior

One crucial property of our method is that we avoid selecting a specific prior family for the nonzero part of the mixture prior and use the data to indirectly estimate it. In order to show the benefit of this approach we compare our procedure with one involving a zero mean normal prior with variance  $a^2$  for the non-zero part.

$$\gamma(\mu_i) = \mathcal{N}(\mu_i|0, a^2). \quad (23)$$

Here we consider both  $w$  and  $a$  as the hyperparameters and use the same empirical Bayesian approach to estimate them by maximizing the marginal likelihood which is given by  $\log m(\mathbf{z}|w, a) = \sum_{i=1}^d \log [w\mathcal{N}(z_i|0, 1) + (1 - w)g_a(z_i)]$ , where  $g_a(z_i) = \mathcal{N}(z_i|0, 1 + a^2)$ . The posterior mean is given by

$$\hat{\mu}_i = (1 - \tilde{p}_i) \frac{a^2}{1 + a^2} z_i. \quad (24)$$

This prior has been previously used by (Johnstone & Silverman, 2004) in a similar set up.

## 8. Simulations

In order to validate that the proposed procedure adapts well to varying sparsity and to show the advantage of using a non-parametric prior we design a simulation setup. A sequence  $\boldsymbol{\mu}$  of fixed length  $d = 100$  is generated with varying degree of sparsity controlled by a parameter  $R$ . The sequence has  $\mu_i = 0$  except at  $R$  random positions, where it takes a value  $V$ —representing the relevant features. The observation  $z_i$  is generated by adding  $\mathcal{N}(0, 1)$  noise for each  $\mu_i$ . Given  $\mathbf{z}$ ,  $w$  is estimated using the proposed SNBL algorithm with a non-parametric prior and also a normal prior. The number of selected features is estimated as  $(1 - \hat{w}) \cdot d$ . Given  $\hat{w}$  the mean of the posterior can also be computed to get an estimate of the underlying true sequence  $\boldsymbol{\mu}$ .

Figure 1(a) shows an example of such a sequence used in our simulations with  $R = 5$  (very sparse) and  $V = 5$ . Figure 1(c) shows our estimate (the mean of the posterior) for the proposed method using a non-parametric prior. Most of the values in the estimate are zero. Figure 1(d) shows the same for a normal parametric prior. The proposed non-parametric method was able to estimate  $w$  correctly as 0.95 thus selecting the 5 features, while the parametric method chose 11 features.

Figure 1(b) shows the behavior of the two estimators (posterior mean) for this signal as a function of the observation. Both the estimators have what is known as the shrinkage property, *i.e.*, the estimates shrink towards zero. The non-parametric method does a better job at finding when to shrink the values to zero. It is also interesting to note that this plot for the non-parametric estimator is not symmetric around zero unlike the parametric case. This is because the non-parametric method completely relies on the data-driven estimate of the marginal. In this simulation setup we only have large positive values.

Figure 2 shows the estimated marginal of the non-zero part for three different stages in the iteration process. When we start the iteration (Figure 2(a)) we assume all observations come from the non-zero part. Hence we see a large peak at zero since most of the values come from the zero part. It can be seen that as the optimization proceeds the  $w$  gets refined and marginal adapts to the non-zero part.

Figure 3 plots the number of selected features as a function of  $R$ —varying from 10 (very sparse setup) to 90 (most features are relevant). The results are averaged over 100 repetitions. It can be seen that the proposed SNBL method with non-parametric prior estimates the number of features quite accurately. It adapts well to

varying sparsity. It also has a small variability. On the other hand the commonly used normal prior yields a biased estimate.

## 9. Experiments

Table 1 summarizes the six publicly available datasets<sup>1</sup> used in our experiments. These datasets from different application domains have been previously used for feature selection challenges.

Our proposed method of selecting the relevant features can be used in conjunction with any ranking criterion having approximately a normal distribution. For our experiments we use the two-sample  $t$ -statistic (1) as the scores. We compare the number of features selected by the proposed method to those selected by the cross-validated area under the ROC curve (AUC) based criterion. The AUC based criterion is implemented as follows: The features are first sorted based on the absolute value of the two-sample  $t$ -statistic. A linear discriminant analysis (LDA) classifier is trained using the top  $t$  features. The performance of this classifier is tested on an independent validation set (see Table 1). We use the AUC as our performance metric. This is repeated for  $t$  varying from 1 to  $d$ —the number of features. The optimal number of features selected is the value of  $t$  where the AUC on the validation set is maximum.

Table 2 compares the number of features selected by the proposed method with those selected by the cross-validated AUC based criteria. The resulting AUC on the validation set is also compared. This table can be studied in conjunction with Figure 4 which plots the AUC on both the training and the validation set as a function of  $t$ . The following observations can be made—

(1) From Table 2 we see that for most datasets the number of features selected by the proposed SNBL algorithm is very close to those selected by the cross-validation based criterion. In cases where they differ the proposed method selects much less features and achieves the same AUC on the test set. The proposed method adapts well to the sparsity of the problem by estimating  $w$  correctly.

(2) From Figure 4 we see two types of behavior. For some datasets (see Figures 4(a), (b), and (c)) the AUC on the validation set reaches a peak and then starts decreasing. In such cases the number of features selected by the proposed algorithm is very close to the features selected by cross-validation.

<sup>1</sup>These datasets were downloaded from <http://www.nipsfsc.ecs.soton.ac.uk/datasets/> and <http://www.agnostic.inf.ethz.ch/datasets.php>.

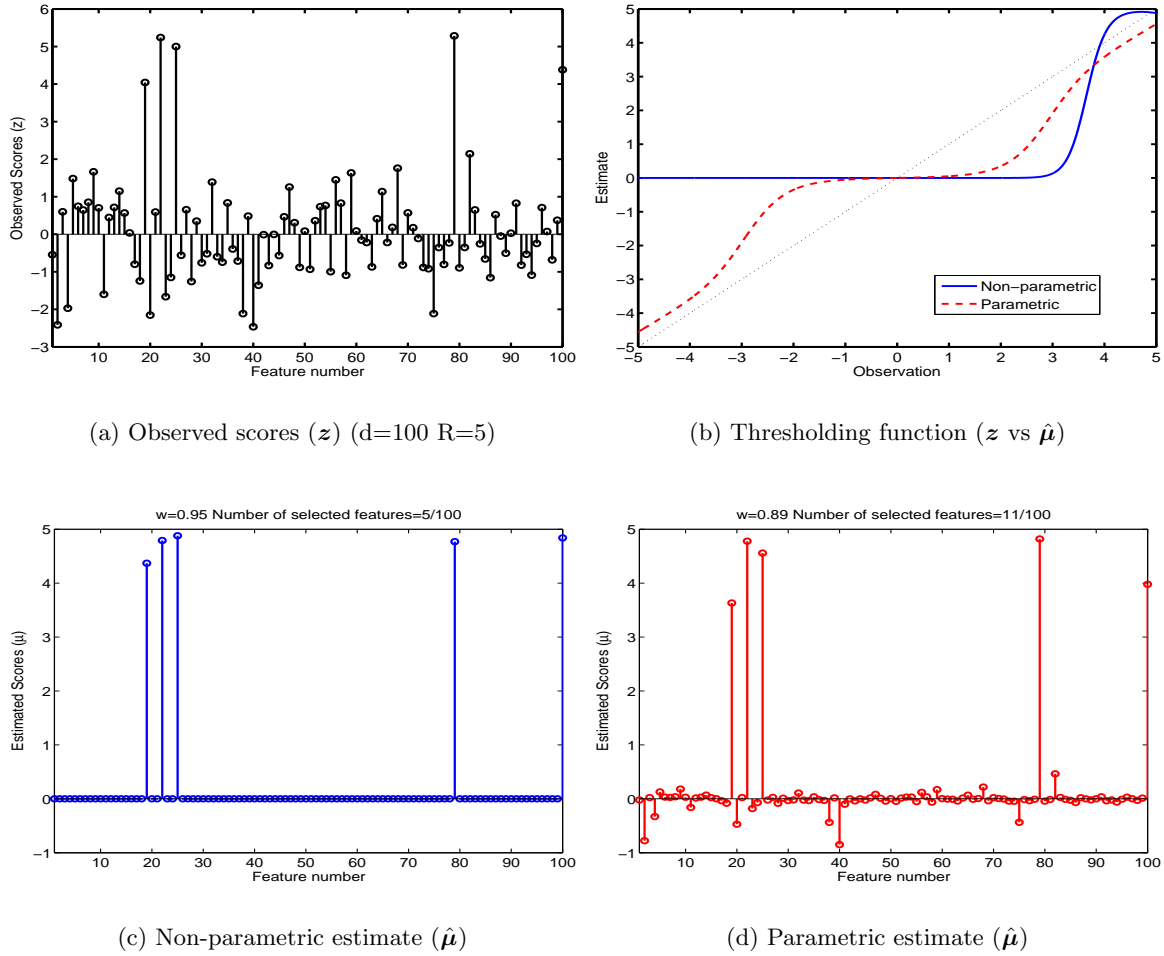


Figure 1. (a) A sample set of scores  $z$  used in our simulation studies. The sequence of length  $d = 100$  has  $R = 5$  values which are non-zero with signal strength  $V = 5$ . (c) The estimate  $\hat{\mu}$  obtained by the non-parametric method. (d) The estimate  $\hat{\mu}$  obtained by the parametric method. (b) The thresholding behavior of the two estimators. The proposed non-parametric method was able to estimate  $w$  correctly thus selecting the 5 features, while the parametric method chose 11 features.

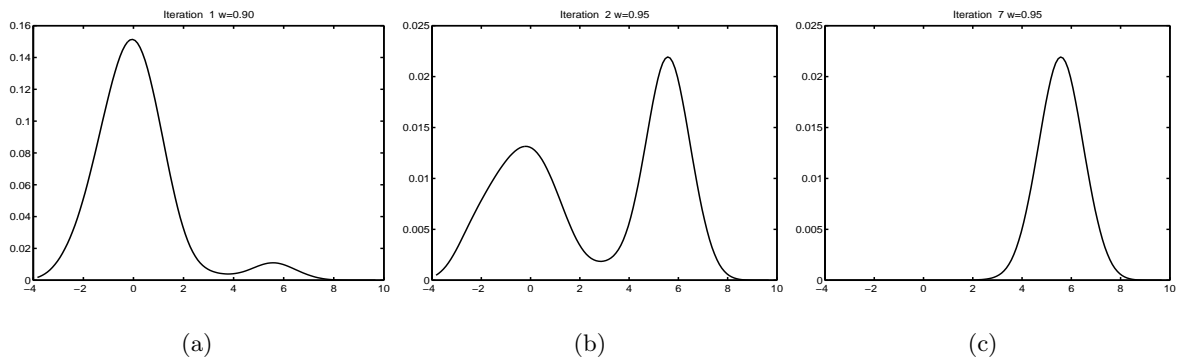


Figure 2. The marginal estimated by the non-parametric procedure for the setup described in Figure 1 during seven iterations of the optimization process.

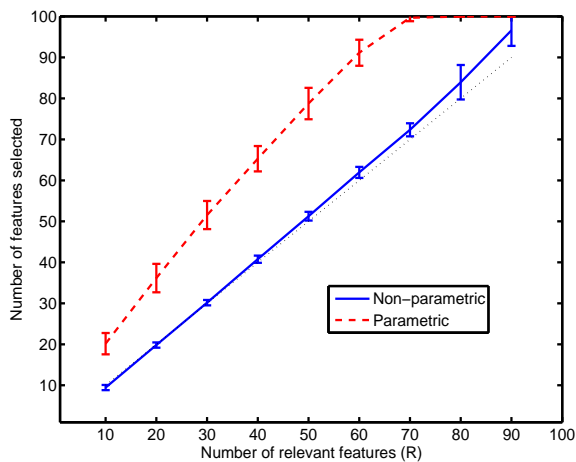


Figure 3. *Simulation Results* The mean and the error bars for number of selected features as a function of  $R$ -varying from 10 (very sparse setup) to 90 (most features are relevant). The 'Non-parametric' corresponds to the situation when the the non-zero part of the prior is completely unspecified The 'Parametric' assumes a normal prior. The results are averaged over 100 repetitions of the experiment.

(3) For some datasets (see Figures 4(d), (e), and (f)), the AUC on the validation set saturates after which there is no further significant improvement in the AUC by including more features. In such cases the proposed method selects much smaller features than the cross-validated criterion and at the same time achieving very similar AUC on the validation set.

(4) The SNBL algorithm shows good generalization properties. The number of features selected by the proposed algorithm leads to the best AUC on the *validation set*.

(5) SNBL is computationally efficient. It does not require any sort of cross-validation. The cross-validated AUC based criteria is very time and memory consuming especially if the number of features is large.

## 10. Conclusion

We proposed a sparse non-parametric empirical Bayesian method which automatically selects the number of relevant features. The proposed method of selecting the relevant features can be used in conjunction with commonly used ranking criterion. It adapts well to the sparsity of the problem. Our experiments on publicly available datasets show that the number of features selected by the proposed method is close to or smaller (without any reduction in the generalization error) than that selected by the popular, but time consuming, cross-validation criteria. A novel idea was

Table 1. The six data sets used in our experiments.

Dataset	Training examples	Validation examples	Number of Features	Domain
madelon	2000	600	500	Synthetic
gina	3153	315	970	Writing
nova	1754	175	16969	Text
ada	4147	415	48	Marketing
sylva	13086	1309	216	Ecology
gisette	6000	1000	5000	Writing

the use of a nonparametric prior which could be useful in other Bayesian approaches. While the proposed method was discussed in the context of feature selection, it can also be used in other contexts especially in denoising applications in signal processing.

## References

Berger, J. (1985). Statistical decision theory and Bayesian analysis. *Springer-Verlag*.

Bickel, P., & Levina, E. (2004). Some theory for Fisher’s linear discriminant function, ‘naive Bayes’, and some alternatives when there are many more variables than observations. *Bernoulli*, 10, 989–1010.

Domingos, P., & Pazzani, M. J. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29, 103–130.

Furey, T. S., Duffy, N., W, D., & Haussler, D. (2000). Some theory for Fisher’s linear discriminant function, ‘naive Bayes’, and some alternatives when there are many more variables than observations. *Bioinformatics*, 16, 906–914.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., & Bloomfield, C. D. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531–537.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3, 1157–1182.

Johnstone, I. M., & Silverman, B. W. (2004). Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences. *The Annals of Statistics*, 32, 1594–1649.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval (pp. 4–15. ). Springer Verlag.

- Slonim, D. K., Tamayo, P., Mesirov, J. P., Golub, T. R., & L, E. S. (2000). Class prediction and discovery using gene expression data. *4th Annual International Conference on Computational Molecular Biology (RECOMB)* (pp. 263–272).
- Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1, 211–244.
- Wand, M. P., & Jones, M. C. (1995). *Kernel smoothing*. Chapman and Hall.
- Wipf, D., Palmer, J., & Rao, B. (2004). Perspectives on sparse bayesian learning. In *Advances in neural information processing systems 16*.

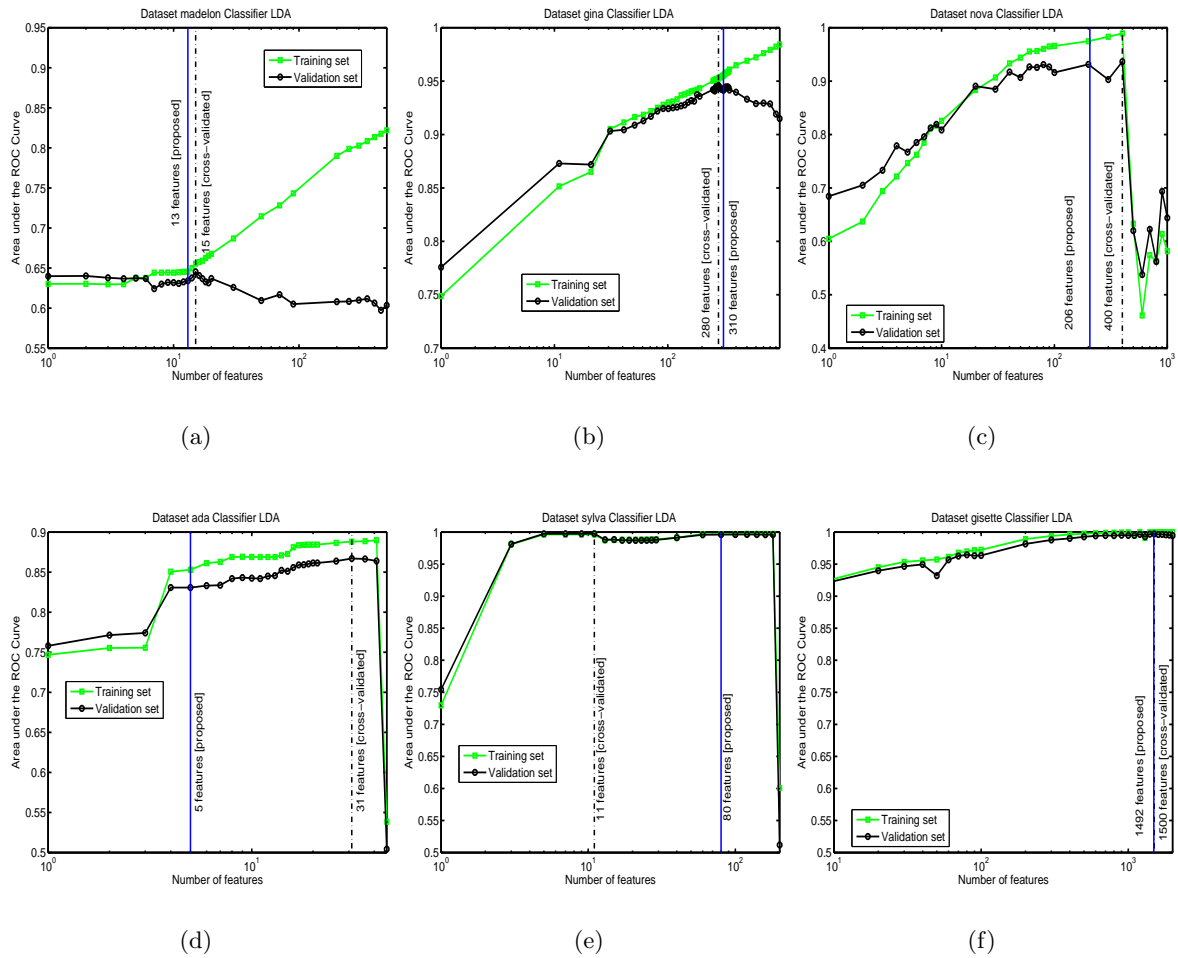


Figure 4. The AUC on both the training and the validation set as a function of the number of top features used to train the LDA classifier for different datasets. The number of features selected by the proposed method (which requires no cross-validation) is marked as a blue solid line. The number of features selected by the cross-validated AUC criterion is marked as a dotted black line.

Table 2. The number of features selected by the proposed method and those selected by the cross-validated AUC based criteria for the 6 different datasets. The resulting AUC on the validation set is also shown. The column  $\hat{w}$  is the estimate of the fraction of irrelevant features. Due to memory problems for datasets marked  $\star$  we ran the cross-validation procedure only till a limited number of features features.

Dataset	Features	Proposed SNBL procedure			Cross-validated AUC criterion	
		Features selected	Validation set AUC	$\hat{w}$	Features	Validation set AUC
madelon	500	13	0.634	0.97	15	0.645
gina	970	310	0.941	0.68	280	0.946
$\star$ nova	16969	206	0.931	0.99	400	0.936
ada	48	5	0.831	0.89	31	0.867
sylvia	216	80	0.996	0.63	11	0.997
gisette	5000	1492	0.997	0.70	1500	0.997