Recursive Sparse Estimation Using a Gaussian Sum Filter

Lachlan Blackhall* Michael Rotkowitz†

Abstract

We develop two recursive estimators that systematically arrive at sparse parameter estimates. The estimators leverage the Gaussian sum filter and sparse parameter estimates emerge by evaluating either of the maximum a posteriori (MAP) or maximum probability (MP) estimates of the posterior distribution. The estimators are computationally feasible for moderate parameter estimation problems and provide both sparse parameter estimates and credible Bayesian intervals for non-zero parameters in a recursive fashion. Simulations show extremely promising accuracy, as well as a robustness not enjoyed by other sparse estimators.

Key Words: Recursive sparse estimation, Recursive ℓ_1 penalized regression, Recursive identification, Bayesian methods.

1 Introduction

It is common to encounter parameter estimation problems with a large number of candidate parameters being equal to zero. This corresponds to a sparse solution of the estimation problem and is of significant interest as a high degree of sparsity corresponds to simpler models. The best way to summarize the myriad reasons why sparse estimates are often desirable, is that there are typically costs associated with the cardinality of the parameter which are not explicitly stated in the objective.

The solution to the sparse estimation problem has recently been the subject of much interest given the results of Candès, Tao and Romberg (a good survey of the results is given in [5] and [7]). This work, colloquially termed 'ℓ_1 -Magic', has provided significant insight into a methodology which has been widely known as the LASSO ([14]) in the statistical domain. This can now be considered as a special case of what is often referred to as 'ℓ_1 -Magic'; that is, the tendency of ℓ_1 minimization or penalization to produce parsimonious results in problems where enforcing that directly would yield computational intractability.

Using the LASSO requires all of the data to be obtained a priori and the resulting parameter estimates provide little information about the accuracy of the non-zero parameters. Having an algorithm that can be implemented recursively, like the Kalman filter, while systematically producing appropriately sparse parameter estimates and credible Bayesian statistics for non-sparse parameter estimates is highly desirable. Previous work by the authors in [3] showed how this could be achieved by taking the maximum a posteriori (MAP) estimate from the output of the Gaussian sum filter. These results were extended in [4] where it was shown how the MAP estimate was related to a maximum probability (MP) estimate which is also related to the approach detailed in [9].

The development and analysis of recursive sparse estimators using both MAP and MP approaches is detailed herein. A characterization and comparison of the performance of both MAP and MP estimators is presented. Extensive simulation results are included to support the theoretical development and to highlight the robust performance of both estimators.

*Research School of Information Sciences and Engineering, The Australian National University, Canberra, ACT, 0200, Australia (e-mail: lachlan.blackhall@anu.edu.au).

†Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC, 3010, Australia (e-mail: mcrotk@unimelb.edu.au).

2 Preliminaries

2.1 Distributions

We will make extensive use of the multivariate Gaussian or normal distribution throughout this paper and we give the following standard definition.

Definition 1 *Given a mean $\mu \in \mathbb{R}^n$ and a covariance $B \in \mathbb{R}^{n\times n}$ with $B > 0$, we say that a random variable X is normally distributed and denote $X \sim \mathcal{N}(\mu, B)$ if it has the following probability density function (pdf) for all $x \in \mathbb{R}^n$*

$$\mathcal{N}(x;\mu,B) = \frac{1}{(2\pi)^{N/2}|B|^{1/2}} \exp\left(-\frac{1}{2}\|B^{-1/2}(x-\mu)\|_2^2\right) \tag{1}$$

where $|B| = \det(B)$.

We also introduce the Laplace, or double exponential, distribution.

Definition 2 *Given a mean $\mu \in \mathbb{R}$ and a scale parameter $\tau > 0$, we say that a random variable X has the Laplace or double exponential distribution and denote $X \sim \mathcal{L}(\mu, \tau)$ if it has the following pdf for all $x \in \mathbb{R}$*

$$\mathcal{L}(x;\mu,\tau) = \frac{1}{2\tau} \exp\left(-\frac{|x-\mu|}{\tau}\right) \tag{2}$$

In this paper we only consider Laplace distributions with zero mean, and thus abbreviate our notation as $\mathcal{L}(\tau) \sim \mathcal{L}(0,\tau)$ and $\mathcal{L}(x;\tau) = \mathcal{L}(x;0,\tau)$.

2.2 Regression

We consider the following parameter estimation problem. Given $X \in \mathbb{R}^{N\times q}$, the rows of which are independent explanatory variables, and dependent response variables, or observations $y \in \mathbb{R}^N$, we assume that the observations are generated as

$$y \;=\; X\theta + \varepsilon, \tag{3}$$

where the noise may be considered normally distributed as $\varepsilon \sim \mathcal{N}(0, R)$, for some $R \in \mathbb{R}^{N\times N}$, $R > 0$. Typically the noise will be considered independent, and we then have $R = \sigma_\varepsilon^2 I$ for some $\sigma_\varepsilon > 0$. We then seek to estimate the underlying parameters $\theta \in \mathbb{R}^q$.

The standard solution to this problem is the ***(ordinary) least squares (OLS)*** estimator, obtained by solving $\theta^*_{\text{OLS}} = \arg\min_{\hat{\theta}} \|y - \hat{y}\|_2^2$ where $\hat{y} = X\hat{\theta}$ gives the fitted values. Where $\theta^*_{\text{OLS}} = (X^TX)^{-1}X^Ty$ is the closed-form solution.

2.2.1 Shrinkage

Since the least squares estimator only considers the goodness-of-fit, it tends to over fit the data. Shrinking the parameter, such as, by penalizing its size, typically performs better on out-of-sample data. A general way to achieve this is to enforce such a penalty as

$$\theta^* \;=\; \arg\min_{\hat{\theta}} \; \|y - \hat{y}\|_2^2 + \lambda\|\hat{\theta}\|_p^p \tag{4}$$

for some parameter $\lambda \geq 0$ and some norm $p \geq 1$. When we consider this estimator with $p = 2$, it becomes what is known in statistics as ***ridge regression (RR)***, and in some other fields as regularized least squares or Tikhonov regularization ([15]). Its popularity is due in large part to the fact that it too can be solved in closed-form, as $\theta^*_{\text{RR}} = \left(X^TX + \lambda I\right)^{-1} X^Ty$ In addition to improving out-of-sample performance, this

estimator has often been used to ensure that the inverse exists for possibly ill-posed problems, which cannot be guaranteed for the ordinary least squares estimate θ^*_{OLS}. This estimator also has a Bayesian interpretation; namely, that θ^*_{RR} arises as the maximum a posteriori (MAP) estimate if the parameters have independent prior Gaussian distributions of $\theta_i \sim \mathcal{N}(0, \sigma_\varepsilon^2/\lambda)$. Note that as the prior variance goes to infinity, we recover the least squares estimate θ^*_{OLS}.

If we instead solve (4) for $p = 1$, that estimator is known as the ***LASSO*** [14]. While a closed-form solution does not exist in general, solving for θ^* is still a convex optimization problem and readily solved.This estimator also has a Bayesian interpretation; namely, that it arises as the MAP estimate if the parameters have independent prior Laplace distributions of $\theta_i \sim \mathcal{L}(2\sigma_\varepsilon^2/\lambda)$.

This estimator has some attractive properties that will be discussed in the next section.

2.3 Sparse Estimators

Often the number of parameters (q) we are considering is greater than the number necessary to explain the data, and it is thus desirable to use an estimator that will systematically produce sparse estimates.

The classical way to achieve sparse estimates is known as subset selection, where for a desired parameter cardinality of $\tilde{q}$, the least squares estimator is found for all possible $\binom{q}{\tilde{q}}$ models, and then the best is chosen among them. This obviously scales terribly in the number of parameters, and still requires other means of determining the level of sparsity.

Of the estimators described in the previous section, only the LASSO gives sparse estimates. The fact that it yields sparse estimates systematically, combined with the fact that the estimates can be obtained via convex optimization in polynomial time, has made the LASSO a very popular option since its introduction. The conditions and reasons for which this occur have become much better understood in recent years (see the references cited in Section 1).

3 Recursive Estimation and the Gaussian Sum Filter

3.1 Recursive Parameter Estimation

It is often desirable to obtain a parameter estimate in a recursive or iterative fashion. This may be because the number of observations (N) is very large and it would not be possible to process them all at once, or it may be because on-line estimates are needed as the data becomes available.

If we have a parameter θ that follows a (prior) pdf of $f_0(\theta)$, and we observe a set of measurements Y with conditional density $h(Y|\theta)$, we then have the posterior pdf given by:

$$f(\theta|Y) \;=\; \frac{h(Y|\theta)f_0(\theta)}{\int h(Y|\theta)f_0(\theta)d\theta} \tag{5}$$

and the MAP estimate is then given as $\theta^*_{\text{MAP}} \;=\; \arg\max_{\hat{\theta}} f(\hat{\theta}|Y)$. Now let $Y_k = [y_1, \ldots, y_k]$ represent all of the measurements up to and including k. If we can express the posterior given these measurements in terms of the posterior given the previous set of measurements as

$$f_k(\theta|Y_k) \;=\; \frac{h(y_k|\theta)f_{k-1}(\theta|Y_{k-1})}{\int h(y_k|\theta)f_{k-1}(\theta|Y_{k-1})d\theta} \tag{6}$$

that is, if we can use the previous posterior as the new prior, then in theory, we can perform recursive estimation. This equivalence holds if the measurements are conditionally independent, where the measurements ($Y_k = [y_1, \ldots, y_k]$) are said to be conditionally independent provided $h(Y_k|\theta) = h(y_1, \ldots, y_k|\theta) = h(y_1|\theta) \cdots h(y_k|\theta)$.

To perform recursive estimation in practice, we also need for each subsequent distribution f_k to have the same form, parameterizable with a constant number of variables, so that we can just update those with each measurement.

If for example the prior has a Gaussian distribution, then each subsequent posterior distribution is also Gaussian, and thus it is possible to encapsulate all of the previous information in two parameters, mean and covariance. This is precisely what is achieved by the best known recursive estimator, the Kalman filter.

The LASSO, however, has no such recursive estimator, as a double exponential prior distribution yields a posterior which is not a double exponential nor any other easily characterizable distribution. The same is true for the other priors ([8]) known to induce sparse MAP estimates. The objective of this work is thus to systematically achieve sparse estimates, as we could with the LASSO, but in a recursive fashion, as we could with the Kalman filter.

3.2 The Gaussian Sum Filter

We outlined in the previous section that a parameter having a prior distribution that is Gaussian is easily recursively estimated, as the distribution can be simply parameterized as a mean and covariance. The Kalman filter is the most well known estimator in this instance. This same simple parametrization extends to multivariate Gaussian distributions in many dimensions. An extension of the Kalman filter is the Gaussian sum filter that allows non-Gaussian filtering to leverage the effectiveness of the Kalman filter. The Gaussian sum filter was outlined in [12] and [1] and further detailed in [2]. The primary motivation for using the Gaussian sum filter is that non-Gaussian parameter estimate priors can be accommodated. We now outline the form of the filter and our development is based upon a more general version in [2].

Similarly to the Kalman filter, the Gaussian sum filter is typically used for state estimation of a dynamic system; however, it is possible to use it for parameter estimation or system identification, and this can be considered a special case. This is achieved by assuming that there are no internal system dynamics and thus the parameter estimates can only change when a new measurement is obtained. It is worth noting throughout this section that if we chose the number of Gaussians as $M = 1$, we would recover the Kalman filter, and if we did so for the special case of parameter estimation, we would recover ridge regression. The Gaussian sum filter can be considered as a weighted bank of Kalman filters operating in parallel, where the weights change after each measurement is processed.

We will be assuming a linear measurement process and from this standpoint we have a measurement model:

$$y_k = X_k\theta + \varepsilon_k \tag{7}$$

where we have a Gaussian measurement noise process ($\varepsilon_k \sim \mathcal{N}(0, R_k)$) and a prior distribution of θ given by:

$$\theta \sim \sum_{i=1}^{M} \alpha_i \, \mathcal{N}(\mu_i, B_i) \tag{8}$$

where μ_i and B_i are the q-dimensional mean vector and $q \times q$ covariance matrix respectively.

Let us now assume that at a given point we receive a new measurement y_k, along with its corresponding explanatory variable X_k, and that the distribution of the parameter given all of the previous measurements is given as:

$$\theta|Y_{k-1} \sim \sum_{i=1}^{M} \alpha_{i,k-1} \, \mathcal{N}(\mu_{i,k-1}, B_{i,k-1}) \tag{9}$$

where $Y_k = [y_1, \ldots, y_k]$ again represents all of the measurements up to and including k. The distribution of the parameter given all of the measurements including the new one is then given by:

$$\theta|Y_k \;\sim\; \sum_{i=1}^{M} \alpha_{i,k}\, \mathcal{N}(\mu_{i,k}, B_{i,k}) \tag{10}$$

where the updated weights $\alpha_{i,k}$, means $\mu_{i,k}$, and covariances $B_{i,k}$ are given by:

$$\begin{aligned}
\Omega_{i,k} &= X_k B_{i,k-1} X_k^T + R_k \\
K_{i,k} &= B_{i,k-1} X_k^T \Omega_{i,k}^{-1} \\
B_{i,k} &= B_{i,k-1} - B_{i,k-1} X_k^T \Omega_{ik}^{-1} X_k B_{i,k-1}^T \\
\hat{y}_{i,k} &= X_k \mu_{i,k-1} \\
\mu_{i,k} &= \mu_{i,k-1} + K_{i,k}(y_k - \hat{y}_{i,k}) \\
\alpha_{i,k} &= \frac{\alpha_{i,k-1}\, \mathcal{N}(y_k;\, \hat{y}_{i,k},\, \Omega_{i,k})}{\sum_{j=1}^{M} \alpha_{j,k-1}\, \mathcal{N}(y_k;\, \hat{y}_{j,k},\, \Omega_{j,k})}
\end{aligned} \tag{11}$$

If we have a Gaussian mixture for our prior distribution given as (8), we can then set the initial weights as $\alpha_{i,0} = \alpha_i$, the initial means as $\mu_{i,0} = \mu_i$, and the initial covariances as $B_{i,0} = B_i$ for all $i \in \{1, \ldots, M\}$, run the above iteration for each new measurement received, and then arrive at the posterior distribution as

$$\theta|Y_N \;\sim\; \sum_{i=1}^{M} \alpha_{i,N}\, \mathcal{N}(\mu_{i,N}, B_{i,N}). \tag{12}$$

Finding the MAP estimate of θ then requires finding the mode of this posterior Gaussian mixture which is given by:

$$\theta^*_{\text{MAP}} \;=\; \arg\max_{\hat{\theta}} \sum_{i=1}^{M} \alpha_{i,N}\, \mathcal{N}(\hat{\theta}; \mu_{i,N}, B_{i,N}). \tag{13}$$

4 Gaussian Mixtures and the Laplacian Prior

4.1 Gaussian Mixtures

Having detailed the Gaussian sum filter it remains to show that it is possible to approximate a non-Gaussian distribution as a finite sum of Gaussians. In particular, our goal will be to represent the double exponential distribution in a form amenable to recursive propagation.

In [8] it was shown how to represent several priors, all known to induce sparse MAP estimates, as mixtures of Gaussian distributions in one dimension in the following form:

$$f(\theta) \;=\; \int_{\psi=0}^{\infty} g(\psi)\, \mathcal{N}(\theta; 0, \psi)\, d\psi \tag{14}$$

For a double exponential $\theta \sim \mathcal{L}(\tau)$ in particular, we have:

$$g(\psi; \tau) \;=\; \frac{1}{2\tau^2} \exp\left(-\frac{\psi}{2\tau^2}\right) \tag{15}$$

in other words, the hyper-prior has an exponential distribution.

It was shown in [12] that any probability density $f_{\text{des}}(\theta)$ can be approximated as closely as desired in the space $\ell_1(\mathbb{R}^n)$ by a fine enough Gaussian sum mixture:

$$f(\theta) \;=\; \sum_{i=1}^{M} \alpha_i \, \mathcal{N}(\theta; \mu_i, B_i) \tag{16}$$

where M is the number of Gaussians, $\alpha_i \in \mathbb{R}^+$ with $\sum_i^M \alpha_i = 1$ are the weights, $\mu_i \in \mathbb{R}^N$ are the means, and $B_i \in \mathbb{R}^{N \times N}$ are the covariances. The closeness of approximation corresponds to:

$$\int_{R^n} |f_{\text{des}}(\theta) - f(\theta)| \; d\theta \tag{17}$$

being arbitrarily small for a large enough number of Gaussians M.

Given a distribution which can be represented as an infinite Gaussian mixture (14), we can then approximate the distribution as a Gaussian sum (16) by selecting a range of variances ψ_i which are as representative as possible, and then choosing the associated weights as $\alpha_i \;\propto\; g(\psi_i)$ of course scaling them to ensure that $\sum_{i=1}^M \alpha_i = 1$.

4.2 The Laplacian Prior as a Gaussian Sum

As mentioned in the preliminaries, the LASSO estimate can be interpreted as the MAP estimate when the parameters have independent Laplace prior distributions. It was shown in [3] that the higher dimensional Laplacian prior could be approximated as a Gaussian sum by:

$$\begin{aligned}
f_0(\theta) \;&=\; \prod_{j=1}^{q} \mathcal{L}(\theta_j; \tau) \\
&=\; \prod_{j=1}^{q} \int_{\psi_j=0}^{\infty} g(\psi_j; \tau) \, \mathcal{N}(\theta_j;\, 0, \psi_j) \; d\psi_j \\
&\approx\; \prod_{j=1}^{q} \sum_{i_j=1}^{M} \alpha_{i_j} \, \mathcal{N}(\theta_j;\, 0, \psi_{i_j}) \\
&=\; \sum_{i_1=1}^{M} \cdots \sum_{i_q=1}^{M} \left(\prod_{j=1}^{q} \alpha_{i_j} \, \mathcal{N}(\theta_j;\, 0, \psi_{i_j}) \right) \\
&=\; \sum_{i_1=1}^{M} \cdots \sum_{i_q=1}^{M} \left(\prod_{j=1}^{q} \alpha_{i_j} \; \prod_{j=1}^{q} \mathcal{N}(\theta_j;\, 0, \psi_{i_j}) \right) \\
&=\; \sum_{i_1=1}^{M} \cdots \sum_{i_q=1}^{M} \alpha_{i_1,\ldots,i_q} \, \mathcal{N}(\theta;\, 0, B_{i_1,\ldots,i_q})
\end{aligned} \tag{18}$$

where the multivariate weightings and covariances are given, respectively, as

$$\begin{aligned}
\alpha_{i_1,\ldots,i_q} \;&=\; \prod_{j=1}^{q} \alpha_{i_j} \\
B_{i_1,\ldots,i_q} \;&=\; \operatorname{diag}(\psi_{i_1}, \ldots, \psi_{i_q}).
\end{aligned}$$

This shows how to approximate the prior distribution for the LASSO as a sum of Gaussian distributions. Thus we can utilize the Gaussian sum filter of Section 3.2 to recursively estimate the parameters with each new observation, which is not possible with the original distribution.

We unfortunately see that if we have q parameters to estimate and approximate each univariate double exponential with M Gaussians, then we end up using M^q total Gaussians in the final mixture. Addressing this challenge is discussed in more detail in Section 5 and Section 6.

5 Numerical Simulations

We are now able to implement, in MATLAB, the recursive sparse estimator developed thus far. We compare its performance with that of well-known aforementioned estimators using simulated data. In Section 5.1, we first test the algorithm with $q = 2$ parameters to estimate. This allows us to graphically present the posterior distributions, and also allows us to compare results for different values of the approximation fineness (M). We surprisingly see that we can proceed with $M = 2$, and then move on to consider higher-dimensional problems in Section 5.3.

5.1 Sparse Two Parameter Estimates

We first test our algorithm with $q = 2$ parameters. We will compare it to the least squares estimator, ridge regression, and of course, the LASSO. We test what happens for the three possible levels of sparsity by considering true coefficients of $[0\ 0]$, $[0\ 1]$, and $[1\ 1]$. In these examples we simulated fifty data sets where we have $N = 30$ data points, and the double exponential distribution is approximated by $M = 20$ initial variances, varying linearly between $\sigma^2_{\text{min}} = 1 \times 10^{-4}$ and $\sigma^2_{\text{max}} = 1$, corresponding to 400 Gaussian distributions in the Gaussian sum filter. The regressor matrix is composed of random values drawn from the uniform distribution on the unit interval (that is, $X_{ij} \sim U[0, 1]$, generated using the MATLAB *rand* command), the measurement noise is generated (using the MATLAB *randn* command) as $\varepsilon_k \sim \mathcal{N}(0, \sigma^2_\varepsilon)$ with $\sigma^2_\varepsilon = 0.5$, and the measurements are then generated as $y_k = X_k\theta + \varepsilon_k$.

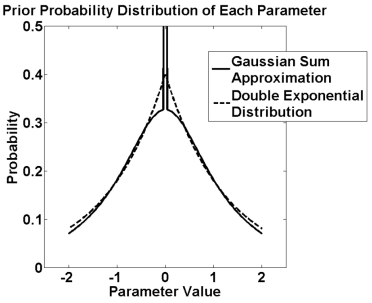


Figure 1: Prior probability density function (pdf) of each parameter prior to estimation commencing.

As in previous sections, λ is the penalty term for the one norm of the parameters in the LASSO and, equivalently, determines the shape of the double exponential prior distribution of the parameters being estimated. A comparison of the exact double exponential distribution and the Gaussian sum approximation in one dimension (for $\lambda = 0.8$) can be seen in Figure 1. The equivalence is very good except for very small absolute values of the parameter, where the approximation deviates mostly due to the smallest variance used in the Gaussian sum approximation. The peak of this deviation rises to a value of approximately 4 and could be seen as providing additional prior probability that the resulting parameter estimate will be sparse.

For the following analyses we choose the penalty term (λ) for the LASSO and the ridge regression using two-fold cross validation. We take 75 percent of the data set, vary λ, and for each value of λ, compute the coefficients (the model). The optimal λ is chosen as

the λ that corresponds to the model which yields the lowest mean squared error between the measurements and the fitted values in the remaining (out-of-sample) data. Utilizing this method we obtain $\lambda = 2.5,\ 0.8,\ 0.01$ as the penalty parameter for the LASSO, when the true underlying coefficients are 0-0, 0-1, and 1-1, respectively, and we similarly obtain $\lambda = 500,\ 0.05,\ 0.05$ as the penalty terms for the ridge regression. For our recursive algorithm, we use the same λ as those chosen for the LASSO. It could be seen as a significant disadvantage not to tune the parameter specifically for our algorithm, but we will see throughout this section that the algorithm enjoys great robustness with respect to its tuning parameters.

The LASSO is implemented using code from [11], and computation of the other estimators is straightforward. The comparison of the results from using this algorithm with different parameter combinations can be seen in Tables 1, 2 and 3. For each choice of parameters and regression algorithm we have computed the median mean squared error (MSE) of the coefficient estimates, percentage of correct zeros (where appropriate) and the percentage of incorrect zeros (also where appropriate) of the estimated parameters. Due to the computational nature of these algorithms, and the small but non-zero value of σ_{min} we define zero to be set at a threshold equal to $10\sigma_{\text{min}}$, thus provided the parameter estimates are below this threshold they are considered to be zero. This is appropriate because as $\sigma_{\text{min}} \neq 0$ we are only certain of the value of the parameter to the accuracy of the Gaussian distribution defined by σ_{min}. Setting $\sigma_{\text{min}} = 0$ is not possible in the current MAP framework, the reasons for which are discussed in greater detail in Section 6. We will also show in Section 6 that it is possible to set $\sigma_{\text{min}} = 0$ if we alternatively adopt a maximum probability (MP) approach instead of the current MAP methodology.

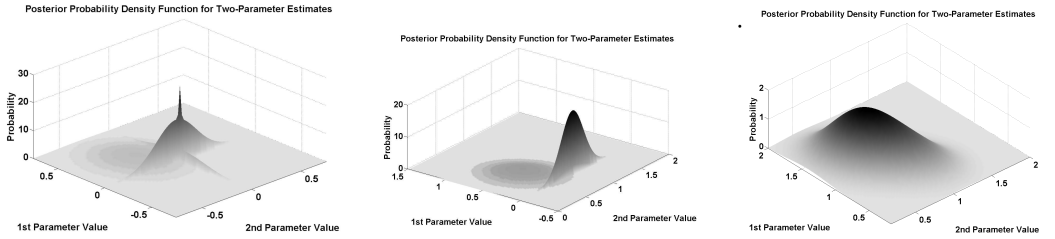
Method	Median MSE	Perc. True Zero Coeffs.
OLS	0.002	0%
Ridge	0.000	0%
LASSO	0.000	65%
RS	0.000	91%

Table 1: Results when the true coefficients are $[0\ 0]$. We compare least squares (OLS), ridge regression, LASSO, and recursive sparse (RS) estimates.

In Table 1 it is seen that the recursive sparse (RS) algorithm correctly identifies that both parameters are zero over 90 percent of the time. These examples nicely illustrate the dependence of the LASSO on its parameter λ. The LASSO estimator, defined as penalized least squares (4) with norm $p = 1$, is equivalent to finding the least squares estimate subject to a constraint of the form $\|\hat{\theta}\|_1 \leq t$, where the constrained and penalized forms are equivalent but the relationship between t and λ is not known a priori. In fact, the LASSO was first introduced in this form of constrained least squares in [14]. The much lower percentage of zeros identified by the LASSO in Table 1 represent that while the penalty term is sufficient to constrain one of the parameters to zero it is impossible for both parameters to be set to zero unless the penalty term approaches infinity.

In Figure 2a we show the resultant posterior distribution after a typical run of the recursive algorithm with a true underlying coefficient of $[0\ 0]$. We can see a large spike at the origin as the algorithm identifies this as the best estimate, and we see the contours from other Gaussians in the original mixture with severely diminished weights.

When the parameters are different and we have one zero parameter and one non-zero parameter we also obtain the correct sparse estimate substantially more often using the recursive approach to sparse parameter estimation. In this case the LASSO struggles to accurately identify the zero parameter. This is somewhat surprising given that cross validation was used to obtain an appropriate penalty term for this particular level of sparsity.



(a) Both parameters being estimated are equal to zero.

(b) One parameter equals zero and the other equals one.

(c) Both parameters are equal to one.

Figure 2: The posterior probability distributions for different sparse parameter combinations using the MAP sparse estimator with $M = 2$

One noteworthy point that can be seen in Table 2 is that the recursive sparse algorithm incorrectly identifies a zero on just one occasion in the fifty data sets analyzed.

Choosing a representative result from the analyzed data another important aspect of this algorithm can be observed. In the posterior distribution in Figure 2b we can observe essentially a one dimensional Gaussian distribution embedded in the resultant distribution. This occurs because the zero parameter has correctly been identified as being zero and thus has a small resultant covariance associated with it. The non-zero parameter has a complete posterior distribution providing credible Bayesian intervals for this particular parameter.

The LASSO only provides the MAP point estimate of a parameter value and so cannot provide confidence intervals for the non-zero parameters thus making it hard to ascertain the accuracy of those estimates. The potential of this algorithm to not only operate recursively, but to simultaneously identify zero parameters and provide statistical quantities about the non-zero parameters, is a major advantage going forward.

Method	Median MSE	Perc. True Zero Coeffs.	Perc. False Zero Coeffs.
OLS	0.005	0%	0%
Ridge	0.004	2%	0%
LASSO	0.002	28%	0%
RS	0.002	90%	2%

Table 2: Results when the true coefficients are $[0\ 1]$. We compare least squares (OLS), ridge regression, LASSO, and recursive sparse (RS) estimates.

In the final scenario both parameters are non-zero, and we present the results in Table 3. The recursive sparse algorithm is seen to occasionally falsely estimate a zero, and the results are otherwise very similar for the non-sparse scenario.

We again display the posterior distribution from a representative run of the algorithm in Figure 2c. In this case, the algorithm has correctly identified both parameters as non-zero, and a typical multivariate Gaussian distribution is the result. This distribution could then provide error estimates similar to those of a standard regression analysis.

The flexibility of this algorithm in performing parameter selection and parameter estimation is nicely observed through the changing shape of the posterior distributions across these three scenarios.

Method	Median MSE	Perc. False Zero Coeffs.
OLS	0.003	0%
Ridge	0.005	0%
LASSO	0.005	0%
RS	0.004	3%

Table 3: Results when the true coefficients are $[1\ 1]$. We compare least squares (OLS), ridge regression, LASSO, and recursive sparse (RS) estimates.

5.2 Number of Gaussians

Having achieved a very promising proof-of-concept in 2 dimensions, we now consider the real objective of recursively estimating sparse parameters in higher dimensions. If we estimate q parameters, and use $M = 20$ Gaussians for each parameter as in the previous section, we will have 20^q Gaussians in our algorithm, which will not be tractable for much larger values of q. However, while about that many Gaussians are necessary to approximate the Laplace distribution well in the prior distribution, it may be possible to achieve a similar posterior distribution with far fewer, which is what we ultimately care about. In fact, studying the behavior of the algorithm shows that a zero estimate arises from the weight on the Gaussian with the smallest variance (and thus biggest peak at zero) growing while the others diminish, and that for non-zero estimates, the Gaussians with non-trivial weights at the end coalesce around a similar estimate. This implies that we may be able to achieve similar results using only 2 Gaussians for each parameter, one with a very small variance corresponding to a prior probability of the coefficient being zero, and one with a larger variance, corresponding to a typical ridge regression if it is not. We now compare results for the same three scenarios as before using both $M = 20$ Gaussians for each parameter and $M = 2$.

True Parameters	M	Median MSE	Perc. True Zero Coeffs.	Perc. False Zero Coeffs.
0-0	20	0.000	91%	NA
0-0	2	0.000	98%	NA
0-1	20	0.002	90%	2%
0-1	2	0.001	94%	8%
1-1	20	0.004	NA	3%
1-1	2	0.052	NA	29%

Table 4: Comparison of the recursive sparse (RS) estimates for $q = 2$ parameters with $M = 20$ and $M = 2$ variances for each Gaussian mixture.

Table 4 has the results of this comparison and it can be seen that the ability of the algorithm to correctly identify the zero parameters is comparable even with many fewer Gaussians. The general effect of moving to 2 Gaussians and losing those with intermediate variances seems be that zeros are estimated a bit more often. The performance is thus a little better for the case where the true parameter is $0 - 0$, worse where it is $1 - 1$, and similar for the $0 - 1$ case with more zero estimates overall.

While the original motivation for the Gaussian sum was to approximate the prior distribution which corresponded to that of the LASSO, we see that we can achieve our end goal of systematically and accurately estimating sparse parameters perhaps just as well using a bi-Gaussian filter. This leads to 2^q total Gaussians in the algorithm, and while it still scales exponentially, it allows us to move up to higher dimensions more easily.

5.3 Sparse Higher Dimensional Estimates

Having shown that it is possible to use only two Gaussians for each parameter, we now illustrate the effectiveness of this method in higher dimensions by performing sparse parameter estimation on a parameter vector with $q = 10$ components. For comparison we also compute the LASSO and other parameter estimates for this problem. For this simulation we used $N = 30$ data points and the measurement noise had a variance of $\sigma_\varepsilon^2 = 0.5$. The probability of each parameter being equal to zero was 0.5. The non-zero parameters were then chosen from a uniform distribution on the interval $[0, 5]$. The penalty parameter for the LASSO ($\lambda = 0.5$) was chosen by cross validation for the case where five of the parameters

were equal to zero. The two variances chosen for the recursive Bayesian algorithm were again chosen to be $\sigma^2_{\text{min}} = 1 \times 10^{-4}$ and $\sigma^2_{\text{max}} = 1$.

Method	Median MSE	Perc. True Zero Coeffs.	Perc. False Zero Coeffs.
OLS	0.078	0%	0%
Ridge	0.090	0%	0%
LASSO	0.035	12%	3%
RS	0.026	68%	14%

Table 5: Results for $q = 10$ where we have an average of five zero parameters. We compare the least squares (OLS), ridge regression, LASSO, and recursive sparse (RS) estimates.

The results for this simulation are shown in Table 5. It is encouraging to realize the high accuracy with which the algorithm is able to select zero parameters recursively. As noted previously, the algorithm again occasionally finds false zeros. It can be shown that the percentage of true and false zeros is inherently related to the relationship between σ_{min}, σ_{max} and α.

It can be seen that the LASSO has difficulty extracting the correct sparse model. This is a demonstration of the high reliance of the LASSO on the penalty parameter (λ) chosen. In this example the average number of zero parameters is five, and the LASSO was tuned for this value, but in each of the fifty data sets the actual number of zeros varies. This variation reduces the LASSO's ability to correctly identify the sparsity, something not observed in the recursive algorithm, which appears much more robust to its choice of tuning parameters.

6 The Bi-Gaussian Filter

We saw previously that it was possible to set $M = 2$ in the recursive sparse estimator and obtain equivalent, although marginally higher, levels of sparsity compared to the case when $M >> 2$. We called this the bi-Gaussian filter. In the bi-Gaussian filter we have two Gaussian distributions, characterized by the variances σ^2_{min} and σ^2_{max} respectively and we typically use an a priori weighting of $\alpha = 0.5$ for both Gaussians rather than the weighting given by the hyper prior $g(\psi_j; \tau)$ as was done previously. This is a heuristic adjustment to the algorithm that represents the two potential outcomes for each parameter, that it is either zero or non-zero. If the parameter is non-zero we are interested in obtaining further Bayesian credible statistics. Conversely, if the parameter is zero then we need not consider it in the model. In our earlier work this left us needing to choose variances σ^2_{min} and σ^2_{max} as the variables that affected the performance of the recursive sparse estimator. It should be observed that $\sigma^2_{\text{min}} \neq 0$ means that we are not representing the potential for a zero parameter estimate exactly but rather that the parameter is small but finite. It was possible to show that this assumption still allowed sparse estimates to be recovered (Section 5). Furthermore, it was necessary to set $\sigma^2_{\text{min}} \neq 0$ in order to be able to compute a meaningful MAP estimate. We will show in Section 6.1.1 that such a MAP estimate is no longer meaningful, from the perspective of parameter estimation, if $\sigma^2_{\text{min}} = 0$.

It is of interest to understand the conditions under which it is possible to still achieve meaningful sparse estimates when we set $\sigma^2_{\text{min}} = 0$ and this is the topic of subsequent sections.

It should be noted that the bi-Gaussian filter (with $M = 2$) is related to the spike and slab prior distribution introduced in [10] for the purpose of Bayesian variable selection for linear regression problems. The bi-Gaussian filter has a prior distribution that is a mixture of two Gaussian distributions, whereas the prior introduced in [10] was a mixture of an impulse ($\sigma^2 = 0$) and a uniform distribution.

6.1 The zero-mean, zero-variance Impulse

We are now interested in examining the effect, on the Gaussian sum filter, of setting $\sigma^2_{\text{min}} = 0$ to represent the possibility of the parameter estimate being exactly equal to zero. This introduces some further challenges which will be addressed below. The equations for the Gaussian sum filter are presented in Eq. 11 and we proceed by evaluating the effect on each line of the algorithm when we have a zero-mean, zero variance impulse as one of the components of the i-th multivariate Gaussian.

For a given matrix (B) we will use standard MATLAB notation to indicate elements within the matrix. Thus $B(j,l)$ is the (j,l)-th entry of the matrix B and $B(j,:)$ and $B(:,j)$ represent the j-th row and column, respectively. Initially we can see that a zero-mean, zero-variance impulse as the j-th component of the i-th multivariate Gaussian implies that $\mu_i(j) = 0$, $B_i(j,:) = 0$, and $B_i(:,j) = 0$.

Beginning with the definition of $\Omega_{i,k} = X_k B_{i,k-1} X_k^T + R_k$ it is observed that the zero impulse component only influences the magnitude of $\Omega_{i,k}$ which it does through the first term of the previous equation.

The second line of the algorithm requires the computation of the vector K and it is here that we first observe the effect of the impulse. It is clear from the form of the vector $K_{i,k} = B_{i,k-1} X_k^T \Omega_{i,k}^{-1}$ that:

$$B_{i,k-1}(j,:) = 0 \Rightarrow K_{i,k}(j,:) = 0$$

We now proceed to update the covariance matrix and we have the following form for the update $B_{i,k} = B_{i,k-1} - B_{i,k-1} X_k^T \Omega_{ik}^{-1} X_k B_{i,k-1}^T$. If we have $B_{i,k-1}(j,:) = 0$ and $B_{i,k-1}(:,j) = 0$, that is the j-th row and column are already zero then we can see that:

$$\begin{aligned} B_{i,k-1}(j,:) = 0 &\Rightarrow B_{i,k}(j,:) = 0 \\ &\Rightarrow B_{i,k}(:,j) = 0 \end{aligned}$$

highlighting that once the covariance matrix has a row and column initialized to zero, as in the case of the impulse, it remains that way for all time.

The penultimate step in the algorithm is to update the Gaussian mean and from $\mu_{i,k} = \mu_{i,k-1} + K_{i,k}(y_k - \hat{y}_{i,k})$ it can be seen that $K_{i,k}(j,:) = 0$ and $\mu_{i,k-1}(j) = 0$ imply $\mu_{i,k}(j) = 0$. Essentially we can see that a zero-mean, zero-variance impulse remains as such for all time. This makes intuitive sense as zero-variance implies zero uncertainty, and hence any change in the mean or variance would contradict this assertion.

The final step in the algorithm is the evaluation of the weighting term ($\alpha_{i,k}$) for each multivariate Gaussian in the Gaussian sum filter. Essentially, in updating $\alpha_{i,k}$ we are determining how closely the mean ($\mu_{i,k}$) of the i-th Gaussian explains the observation (y_k). Thus larger values of $\alpha_{i,k}$ imply that the mean of the i-th Gaussian is a better estimate of the parameter vector of interest.

The analysis presented in this section shows that the Gaussian sum filter continues to provide meaningful output when a zero-mean, zero-variance component is used. It remains to show how to evaluate the multivariate Gaussian with a zero-mean, zero-variance component and this is the topic of the following section.

6.1.1 Evaluating a Gaussian Sum with a zero-mean, zero-variance Impulse

We saw earlier that the form of the multivariate Gaussian distribution is given by Eq. (1). In order to proceed we first present two standard lemmas, without proof, about the computation of matrix determinants:

Lemma 3 *Any matrix with a row or column equal to zero is singular.*

Lemma 4 *The determinant of a singular matrix is zero.*

In our previous work [3] we were able to compute the parameter estimates by finding the MAP estimate directly from the resulting Gaussian sum using what the authors of [6] called the gradient-quadratic method.

Using the previous two lemmas it is clear that when a zero-mean, zero-variance impulse is used as a component in the Gaussian sum filter the term $|B|^{-1/2}$ goes to infinity, implying that obtaining a MAP estimate from this Gaussian sum is now not useful in determining an accurate parameter estimate.

An alternative approach, similar to that in [13], that is suitable for a Gaussian sum with zero-mean, zero-variance components, is determining the component of the Gaussian mixture with the largest weighting. We will refer to this as the maximum probability (MP) estimate. In this methodology we can interpret $\alpha_{i,k}$ as the measure of accuracy of the mean ($\mu_{i,k}$) of the i-th multivariate Gaussian being the best estimate of the true parameter values θ. From this perspective it is possible to compute the sparse estimate as the mean of the multivariate Gaussian that has the largest weighting $\alpha_{i,k}$ and thus our maximum probability estimator is given by:

$$i^* = \arg\max_i \alpha_{i,N} \tag{19}$$

$$\theta^*_{\text{MP}} = \mu_{i^*,N} \tag{20}$$

This can be compared to the earlier MAP estimate given in Eq. 13. It is this maximum probability methodology that we will use to compare with the earlier recursive sparse estimator implementation of [3].

The maximum probability approach has an alternate interpretation as the estimator where we perform 2^q regressions and choose at each time step the regression output that maximizes some performance measure. There are some other points of interest in comparing between the two filters. In the original filter it was necessary to determine at each time step the mode of the Gaussian sum in order to determine the parameter estimates. This was a computationally expensive step as even with the approach detailed in [6] we are forced to compute gradient or fixed point search in a high dimensional space. In moving to the maximum probability estimate we are no longer required to determine the mode resulting in a computationally simpler task. In the following section we present numerical comparisons of the two estimator implementations in order to understand the effect on performance of moving from maximum a posteriori (MAP) to maximum probability (MP) estimate.

6.2 Maximum a Posteriori (MAP) vs Maximum Probability (MP) Estimates

We now compare the recursive sparse estimator using the MAP methodology with the recursive sparse estimator using the MP methodology (for both $\sigma^2_{\text{min}} = 0$ and $\sigma^2_{\text{min}} \neq 0$, where we choose $\sigma^2_{\text{min}} = 1 \times 10^{-4}$) as well as the least squares estimator (OLS), ridge regression (RR), and of course, the LASSO. In these examples we have $M = 2$, $q = 10$ and we simulated fifty data sets where we have $N = 30$ data points. The probability of each parameter being equal to zero was 0.5. The non-zero parameters were then chosen from a uniform distribution on the interval $[0, 5]$. The regressor matrix is composed of random values drawn from the uniform distribution on the unit interval (that is, $X_{ij} \sim U[0, 1]$, generated using the MATLAB *rand* command), the measurement noise is generated (using the MATLAB *randn* command) as $\varepsilon_k \sim \mathcal{N}(0, \sigma^2_\varepsilon)$ with $\sigma^2_\varepsilon = 0.5$, and the measurements are then generated as $y_k = X_k\theta + \varepsilon_k$. Fixed random seeds were chosen such that the results shown in Table 6 are directly comparable.

We now outline the implementation of the recursive sparse estimator detailed in this paper. The prior of each parameter is assumed to be independently distributed and represented as a sum of two Gaussian distributions with σ^2_{min} as outlined above, $\sigma^2_{\text{max}} = 25$

and equal a priori weightings ($\alpha = 0.5$) for both variance components. The mean of each multivariate Gaussian is a $1 \times q$ vector of zeros.

As presented previously, λ is the penalty term for the one norm of the parameters in the LASSO. We choose to implement the recursive sparse estimator with equal a priori weightings ($\alpha = 0.5$) and thus we are only required to choose the penalty terms (λ) for the LASSO and the ridge regression. The penalty terms were chosen by two-fold cross validation for the case where five of the parameters were equal to zero. Utilizing this method we obtain $\lambda = 0.5$ as the penalty parameter for the LASSO and we similarly obtain $\lambda = 0.05$ as the penalty term for the ridge regression.

Method	Median MSE	Perc. True Zero Coeffs.	Perc. False Zero Coeffs.
OLS	0.075	0%	0%
RR	0.08	0%	0%
LASSO	0.046	12%	3%
RS(MAP)	0.049	94%	26%
RS(MP)	0.030	95%	14%
RS(MP, $\sigma^2_{\text{min}} = 0$)	0.030	95%	14%

Table 6: A comparison of recursive sparse estimators using both MAP and MP methods of estimating the sparse estimate when $q = 10$. Results for the MP criterion are shown for both $\sigma^2_{\text{min}} = 0$ and $\sigma^2_{\text{min}} \neq 0$. The estimates obtained using these methods are further compared to the ordinary least squares (OLS), Ridge regression (RR) and the LASSO.

The LASSO is implemented using code from [11], and computation of the least squares and ridge regression estimators is straightforward. The comparison of the results from using this algorithm with different parameter combinations can be seen in Table 6. For each choice of parameters and regression algorithm we have computed the median mean squared error (MSE) of the coefficient estimates, percentage of actual zero parameters correctly estimated to be zero (Perc. True Zero Coeffs.) and the percentage of actual non-zero parameters incorrectly estimated to be zero (Perc. False Zero Coeffs.).

In Table 6 we observe that the MP implementation of the recursive sparse filter achieves a high level of accuracy in its own right as well as comparing favorably with the other estimators, including the MAP version of the same filter. For both $\sigma^2_{\text{min}} \neq 0$ and $\sigma^2_{\text{min}} = 0$ the maximum probability estimate is identical. This arises because whether starting with $\sigma^2_{\text{min}} = 0$ or a value very close to zero, $\sigma^2_{\text{min}} \neq 0$, this component of the Gaussian mixture will have the highest weighting and will thus be chosen when using the MP methodology for estimation.

Of most interest is the decrease in the percentage of false coefficients when moving from the MAP to MP methodologies. Previously, when using the MAP methodology, the Gaussian components with σ^2_{min} placed a large density around the zero estimate, resulting in sparse parameter estimates being favored. Conversely when we use the MP approach we are no longer relying on a mode estimate and thus we are choosing the estimator that best fits the measurements. This reduces the dependence of the estimator output on the value of σ^2_{min}, making it more likely to accurately choose the estimate with the correct level of sparsity and resulting in the observed decrease in false zero coefficients.

7 Conclusion

Two recursive sparse parameter estimators (using the MAP and MP methodologies), that systematically arrive at sparse parameter estimates, have been presented. In simulation, the algorithms performed well, correctly identifying zero and non-zero parameters, while

further providing Bayesian credible statistics of the non-zero parameters. The results presented show that the MP methodology compares favorably with the MAP methodology while providing some important computational efficiency.

While our main objective was to develop an algorithm that would, in a recursive fashion like the Kalman filter, arrive at a sparse estimate similar to that of the LASSO, we saw an important additional bonus; namely, that the performance of our estimator seems to be much more robust to its parameters than the LASSO.

The algorithm is recursive but does not scale with the amount of data points N, it does, however, scale as 2^q in the number of parameters being estimated. Obtaining a better understanding of how both MAP and MP estimators can be exploited for computational benefits in computing sparse parameter estimates recursively is the most important area of future work and will hopefully lead to further improvements in the size and scope of problems which can be solved using this recursive sparse parameter estimation framework.

Application of both MAP and MP algorithms to real world data sets is an important further step in benchmarking the performance of these algorithms.

References

[1] D. L. Alspach and H. W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, AC-17(4):439–448, August 1972.

[2] B. D. Anderson and J. B. Moore. *Optimal Filtering*. Dover Publications, Inc, 2005.

[3] L. Blackhall and M. Rotkowitz. Recursive sparse estimation using agaussian sum filter. In *Proceedings of the International Federation for Automatic Control Congress*, 2008.

[4] L. Blackhall and M. Rotkowitz. Maximum aposteriori vs maximum probability recursive sparse estimation. In *Proceedings of the European Control Conference*, 2009.

[5] E. J. Candès. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*. European Mathematical Society, 2006.

[6] M. A. Carreira-Perpinán. Mode-finding for mixtures of Gaussian distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1318–1323, November 2000.

[7] D. L. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Technical Report*, 2005.

[8] J. Griffin and P. Brown. Alternative prior distributions for variable selection with very many more variables than observations. *University of Kent Technical Report*, 2005.

[9] E. G. Larsson and Y. Selèn. Linear regression with a sparse parameter vector. *IEEE Transactions on Signal Processing*, 55(2):451–460, February 2007.

[10] T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, December 1988.

[11] M. Schmidt. Lasso matlab implementation (http://www.cs.ubc.ca/ schmidtm/software/lasso.html), 2005.

[12] H. Sorenson and D. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7:465–479, 1971.

[13] P. Stoica and Y. Selen. Model-order selection: A review of information criterion rules. *IEEE Signal Processing Magazine*, 21(4):36–47, July 2004.

[14] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

[15] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Sov. Math., Dokl.*, 5:1035–1038, 1963.