Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Recursive Sparse Estimation using a Gaussian Sum Filter

Lachlan Blackhall * **Michael Rotkowitz** **

* *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).

Recursive identification; Bayesian methods; Software for system identification.

Abstract: We develop a recursive estimator that systematically arrives at sparse parameter estimates. The algorithm is computationally feasible for moderate parameter estimation problems and leverages the Gaussian sum filter to 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.

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 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 Candès [2006] and Donoho and Tanner [2005]). This work, colloquially termed 'ℓ_1 -Magic', has provided significant insight into a methodology which has been widely known as the LASSO (Tibshirani [1996]) in the statistical domain. Using the LASSO requires all of the data to be obtained before the solution to the parameter estimation problem can be determined. While the solution is sparse there is little information provided about the accuracy of the non-zero parameters and this too would be beneficial when parameters are estimated. In this work we seek to provide an algorithm that can be implemented recursively, like the Kalman filter, while systematically producing appropriately sparse parameter estimates.

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

$$\theta^*_{\text{OLS}} \;=\; \arg\min_{\hat\theta} \; \|y - \hat y\|_2^2 \tag{4}$$

where $\hat y = X\hat\theta$ gives the fitted values. This has the following closed-form solution

$$\theta^*_{\text{OLS}} \;=\; (X^T X)^{-1} X^T y. \tag{5}$$

Shrinkage Since the least squares estimator only considers the goodness-of-fit, it tends to overfit 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

$$\theta^* \;=\; \arg\min_{\hat{\theta}} \; \|y - \hat{y}\|_2^2 + \lambda \|\hat{\theta}\|_p^p \tag{6}$$

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 (Tikhonov [1963]). Its popularity is due in large part to the fact that it too can be solved in closed-form, as

$$\theta^*_{\text{RR}} \;=\; \left(X^T X + \lambda I\right)^{-1} X^T y. \tag{7}$$

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 estimator (5). This estimator also has a Bayesian interpretation; namely, that (7) arises as the maximum a posteriori (MAP) estimate if the parameters have independent prior 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 (5).

If we instead solve (6) for $p = 1$, that estimator is known as the ***LASSO*** (Tibshirani [1996]). 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 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 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 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. As the parameter λ is increased, the resulting estimate becomes increasingly sparse. More intuition behind the relationship between λ and the resulting sparsity is provided in Section 5.

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. 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 penalisation to produce parsimonious results in problems where enforcing that directly would yield computational intractability. The conditions and reasons for which this occur have become much better understood in recent years (see the references cited in Section 1).

2.4 Gaussian Mixtures

This section discusses the idea of exactly expressing a non-Gaussian distribution as an infinite mix of Gaussians, and approximating 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, which will be discussed in subsequent sections; however, most of the discussion is more general.

In Griffin and Brown [2005] 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:

$$f(\theta) \;=\; \int_{\psi=0}^{\infty} g(\psi)\, \mathcal{N}(\theta; 0, \psi)\; d\psi \tag{8}$$

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

$$g(\psi; \tau) \;=\; \frac{1}{2\tau^2} \exp\left(-\frac{\psi}{2\tau^2}\right) \tag{9}$$

in other words, the hyperprior has an exponential distribution.

It is worth noting that it was shown in Sorenson and Alspach [1971] 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:

$$f(\theta) \;=\; \sum_{i=1}^{M} \alpha_i\, \mathcal{N}(\theta; \mu_i, B_i) \tag{10}$$

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:

$$\int_{R^n} |f_{\text{des}}(\theta) - f(\theta)|\; d\theta \tag{11}$$

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

Given a distribution which can be represented as an infinite Gaussian mixture (8), we can then approximate the distribution as a Gaussian sum (10) by selecting a range of variances ψ_i which are as representative as possible, and then choosing the associated weights as

$$\alpha_i \;\propto\; g(\psi_i), \tag{12}$$

of course scaling them to ensure that $\sum_{i=1}^M \alpha_i = 1$.

3. 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:

$$f(\theta|Y) = \frac{h(Y|\theta)f_0(\theta)}{\int h(Y|\theta)f_0(\theta)d\theta} \tag{13}$$

and the MAP estimate is then given as

$$\theta^*_{\text{MAP}} = \arg\max_{\hat{\theta}} f(\hat{\theta}|Y) \tag{14}$$

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

$$f_k(\theta|Y_k) = \frac{h(y_k|\theta)f_{k-1}(\theta|Y_{k-1})}{\int h(y_k|\theta)f_{k-1}(\theta|Y_{k-1})d\theta} \tag{15}$$

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.

To perform recursive estimation in practice, we also need for each subsequent distribution f_k to have the same form, parametrizable 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 (Griffin and Brown [2005]) 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.

3.1 Recursive Parameter Estimation using the Gaussian Sum Filter

An extension of the simple 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 Sorenson and Alspach [1971] and Alspach and Sorenson [1972] and further detailed in Anderson and Moore [2005]. The primary motivation for the use of this filter is its ability to use non-Gaussian measurement noise and parameter estimate priors, significantly extending the usefulness of the basic Kalman filter. We will outline the form of the filter below and our development is based upon a more general version in Anderson and Moore [2005].

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:

$$y_k = X_k\theta + \varepsilon_k \tag{16}$$

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

$$\theta \sim \sum_{i=1}^{M} \alpha_i \, \mathcal{N}(\mu_i, B_i) \tag{17}$$

where μ_i and B_i are the N-dimensional mean vector and $N \times N$ 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:

$$\theta|Y_{k-1} \sim \sum_{i=1}^{M} \alpha_{i,k-1} \, \mathcal{N}(\mu_{i,k-1}, B_{i,k-1}) \tag{18}$$

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:

$$\theta|Y_k \sim \sum_{i=1}^{M} \alpha_{i,k} \, \mathcal{N}(\mu_{i,k}, B_{i,k}) \tag{19}$$

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

$$\begin{aligned}
\Omega_{i,k} &= X_k^T B_{i,k-1} X_k + R_k \\
K_{i,k} &= B_{i,k-1} X_k \Omega_{i,k}^{-1} \\
B_{i,k} &= B_{i,k-1} - B_{i,k-1} X_k \Omega_{ik}^{-1} X_k^T B_{i,k-1}^T \\
\hat{y}_{i,k} &= X_k \mu_{i,k-1} \\
\mu_{i,k} &= \mu_{i,k-1} + K_{i,k}(y_k - \hat{y}_{i,k}) \\
\alpha_{i,k} &= \frac{\alpha_{i,k-1} \, \mathcal{N}(y_k;\ \hat{y}_{i,k},\ \Omega_{i,k})}{\sum_{j=1}^{M} \alpha_{j,k-1} \, \mathcal{N}(y_k;\ \hat{y}_{j,k},\ \Omega_{j,k})}
\end{aligned} \tag{20}$$

If we have a Gaussian mixture for our prior distribution given as (17), 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

$$\theta|Y_N \sim \sum_{i=1}^{M} \alpha_{i,N} \, \mathcal{N}(\mu_{i,N}, B_{i,N}). \tag{21}$$

Finding the MAP estimate of θ then requires finding the mode of this posterior Gaussian mixture. Two algorithms for finding such modes are given in Carreira-Perpinán [2000], and we utilise what the authors call the gradient-quadratic search methodology.

4. RECURSIVE ℓ_1 PENALIZED REGRESSION

We are now ready to show how to approximate the prior which yields the LASSO estimate as its MAP estimate. In this way we can find the posterior distribution, and thus, the MAP estimate, in a recursive fashion with each new measurement. As mentioned in the preliminaries, the LASSO estimate can be interpreted as the MAP estimate when the parameters have independent Laplace prior distributions. Thus we express the prior as follows:

$$\begin{aligned} f_0(\theta) &= \prod_{j=1}^{q} \mathcal{L}(\theta_j;\tau) \\ &= \prod_{j=1}^{q} \int_{\psi_j=0}^{\infty} g(\psi_j;\tau)\, \mathcal{N}(\theta_j;\ 0,\psi_j)\ d\psi_j \\ &\approx \prod_{j=1}^{q} \sum_{i_j=1}^{M} \alpha_{i_j}\, \mathcal{N}(\theta_j;\ 0,\psi_{i_j}) \\ &= \sum_{i_1=1}^{M} \cdots \sum_{i_q=1}^{M} \left(\prod_{j=1}^{q} \alpha_{i_j}\, \mathcal{N}(\theta_j;\ 0,\psi_{i_j}) \right) \\ &= \sum_{i_1=1}^{M} \cdots \sum_{i_q=1}^{M} \left(\prod_{j=1}^{q} \alpha_{i_j} \prod_{j=1}^{q} \mathcal{N}(\theta_j;\ 0,\psi_{i_j}) \right) \\ &= \sum_{i_1=1}^{M} \cdots \sum_{i_q=1}^{M} \alpha_{i_1,\ldots,i_q}\, \mathcal{N}(\theta;\ 0,B_{i_1,\ldots,i_q}) \end{aligned} \tag{22}$$

where the multivariate weightings are given as

$$\alpha_{i_1,\ldots,i_q} = \prod_{j=1}^{q} \alpha_{i_j} \tag{23}$$

and the multivariate covariances as

$$B_{i_1,\ldots,i_q} = \operatorname{diag}(\psi_{i_1},\ldots,\psi_{i_q}). \tag{24}$$

This shows how to approximate the prior distribution for the LASSO as a sum of Gaussian distributions. Thus we can utilise Section 3.1 to recursively estimate the parameter 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. However, we will see evidence in the next section that this number can be greatly reduced.

5. NUMERICAL SIMULATIONS

We are now able to implement the recursive sparse estimator developed in this paper in MATLAB, and 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 $\sigma^2_{\min} = 1 \times 10^{-4}$ and $\sigma^2_{\max} = 1$, 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 $y_k = X_k\theta + \varepsilon_k$.

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 will be sparse.

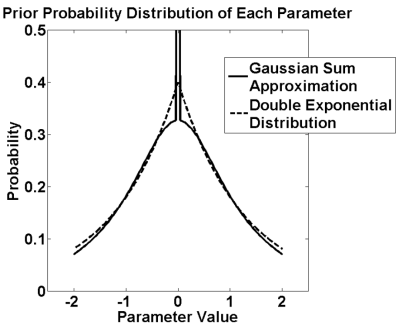

Fig. 1. Prior probability density function (pdf) of each parameter prior to estimation commencing.

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. Utilising 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 Schmidt [2005], 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 $\sigma_{\min}$ we define zero to be set at a threshold equal to $10\sigma_{\min}$, thus provided the parameter estimates are below this threshold they are considered to be zero. This is appropriate because as $\sigma_{\min} \neq 0$ we are only certain of the value of the parameter to the accuracy of the Gaussian distribution defined by $\sigma_{\min}$.

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.

It is worth noting that in all of the examples shown the recursive sparse parameter estimation algorithm significantly outperforms the other algorithms in choosing the correct sparse model. 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 (6) 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 penalised forms are equivalent but the relationship between t and λ is not known apriori. In fact, the LASSO was first introduced in this form of constrained least squares in Tibshirani [1996]. 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 2 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.

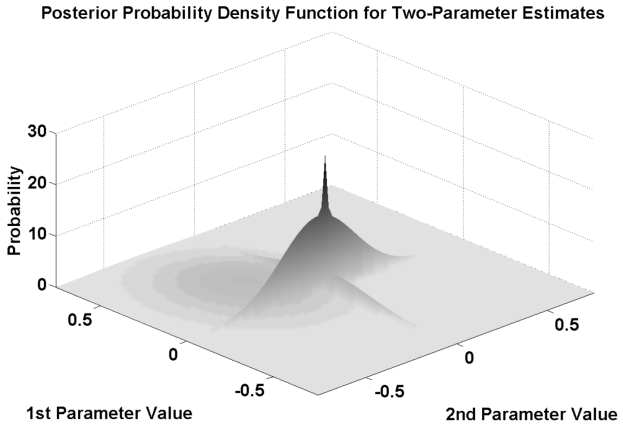

Fig. 2. The posterior probability distribution obtained using the proposed algorithm when both parameters being estimated are equal to zero.

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. 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 analysed.

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.

Choosing a representative result from the analysed data another important aspect of this algorithm can be observed. In the posterior distribution in Figure 3 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, could be a major advantage going forward.

In the final scenario both parameters are non-zero, and we present the results in Table 3. Our algorithm is seen

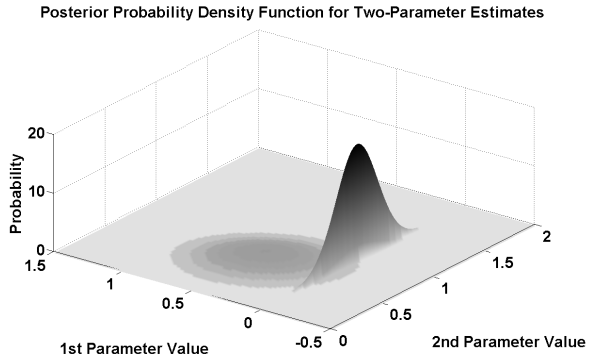

Fig. 3. The posterior probability distribution obtained using the proposed algorithm when one parameter equals zero and the other equals one.

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 4. 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.

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.

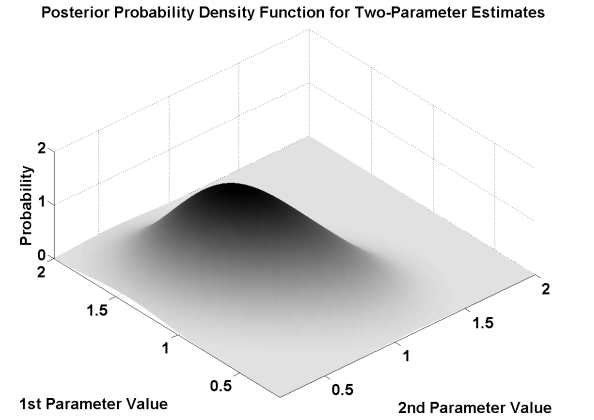

Fig. 4. The posterior probability distribution obtained using the proposed algorithm when both parameters are equal to one.

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 behaviour 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 at least 10 dimensions without a problem.

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 $\sigma_{\min}^2 = 1 \times 10^{-4}$ and $\sigma_{\max}^2 = 1$.

Method	Median MSE	Perc. True Zero Coeffs.	Perc. False Zero Coeffs.
OLS	0.078	0%	0%
Ridge	0.090	0%	0%
LASSO	0.035	22%	1%
RS	0.026	99%	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 realise the high accuracy with which the algorithm is able to select zero parameters recursively, finding nearly every one over all 50 runs. As noted previously, the algorithm again occasionally finds false zeros. It is possible that this can be reduced with a better understanding of how to tune $\sigma_{\min}$ and $\sigma_{\max}$, and the authors hope to more fully characterise this relationship in future work.

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.

To give further intuition into these estimators we provide the coefficient estimates for one typical sample run in Table 6. We can see the recursive algorithm correctly selects all of the zeros, while the LASSO selects most but not all of them.

Actual	OLS	Ridge	LASSO	RS
0	-1.785	-1.538	-0.632	**0.000**
3.541	3.810	3.673	3.570	3.407
0	1.042	0.958	0.404	**0.000**
2.351	2.422	2.305	1.998	1.940
0	-0.471	-0.377	**0**	**0.000**
0	0.645	0.463	**0**	**0.000**
0	-0.074	-0.007	**0**	**0.000**
0	-0.159	-0.119	**0**	**0.000**
2.215	3.394	3.252	2.749	2.623
0	-0.697	-0.504	**0**	**0.000**

Table 6. Parameter estimates for $q = 10$ where we have the actual values, along with the least squares (OLS), ridge regression, LASSO, and recursive sparse (RS) estimates.

6. CONCLUSION AND FUTURE WORK

This paper provides an algorithm for a recursive estimator that systematically arrives at sparse parameter estimates. In simulation, the algorithm performed extremely well, correctly identifying zero and non-zero parameters, while further providing accurate estimates of the non-zero parameters. 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.

The algorithm is recursive and thus does not scale with the amount of data points N, but does scale as 2^q in the number of parameters being estimated; however, as the algorithm eliminates and selects parameters as it runs, it no longer needs to be exponential in those parameters. There are thus several possible methods of exploiting this to extend this algorithm to much larger problems, and developing these is the most important area of future work.

Other areas of future work include the characterisation of the algorithm's performance with respect to $\sigma_{\min}$ and $\sigma_{\max}$. Utilising this algorithm with real data will also be an important step for validating its utility, and extending it to dynamical systems should be straightforward.

REFERENCES

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

Brian D.O. Anderson and John B. Moore. *Optimal Filtering*. Dover Publications, Inc, 2005.

Emmanuel J. Candès. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*. European Mathematical Society, 2006.

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

David L. Donoho and Jared Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Technical Report*, 2005.

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

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

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

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

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