Joint 48th IEEE Conference on Decision and Control and
28th Chinese Control Conference
Shanghai, P.R. China, December 16-18, 2009

Using an Information Filter to Speed Computation of Sparse Parameter Estimates

Lachlan Blackhall and Michael Rotkowitz

Abstract— **This paper discusses the development of a recursive estimator which systematically arrives at sparse parameter estimates. Prior work achieved this by utilizing a Gaussian sum filter. This paper shows the relationship between the implementation using a Gaussian sum filter, where the mean and covariance of each component is propagated, and the equivalent representation using an information filter. We see that the information filter representation requires only a single information filter to be updated for each new measurement instead of the exponential number of measurement updates that were required when using the Gaussian sum filter. We thus see that using the information filter provides computational benefits when recursively estimating sparse parameters, reducing running time as well as data storage.**

I. Introduction

Finding a recursive sparse estimator motivated by the LASSO has been the subject of previous work by the authors (details of which can be found in [1] and [2]). Previous work focused on the development of a recursive sparse estimator using the Gaussian sum filter. The Gaussian sum filter uses multiple Kalman filters to recursively achieve sparse parameter estimates but faces two distinct challenges. One being the appropriate initial choice of Gaussian distributions for the parameters being estimated. The second being the curse of dimensionality that arises from the current filter implementation.

One possible solution to these challenges is to use the information filter. This method is characterized by the use of information matrices and state information vectors rather than the typical covariance matrices and state vectors of the Kalman filter. In this paper we present the relationship between the Kalman and information filters, characterizing how we can use the information filter to recursively achieve sparse estimates. Of most importance we see that a single information filter is capable of representing all 2^q Gaussians in the current recursive sparse estimator. We present theoretical and computational results comparing the operation and performance of the two filters.

II. Preliminaries

A. 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 $\mathcal{L}(x;\tau) = \mathcal{L}(x;0,\tau)$.

B. Sparse Parameter Estimation

We consider the following parameter estimation problem

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

with regressor matrix $X \in \mathbb{R}^{N\times q}$ and observations $y \in \mathbb{R}^N$. We have additionally that the measurement noise ($\varepsilon \sim \mathcal{N}(0,R)$) is i.i.d Gaussian for some $R \in \mathbb{R}^{N\times N},\ R > 0$. We seek to estimate the underlying parameters $\theta \in \mathbb{R}^q$. Often the number of parameters (q) we consider 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 LASSO [3], obtained by solving

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

for some parameter $\lambda \geq 0$ and norm $p = 1$ is a sparse estimator that has achieved considerable attention recently [4] [5] due to some attractive qualities. This estimator also has a Bayesian interpretation; namely, that it arises as the MAP estimate if the parameters have independent prior distributions of $\theta_i \sim \mathcal{L}(2\sigma_\varepsilon^2/\lambda)$. The goal of the recursive sparse estimator of [1] and [2] was to find a recursive form of the LASSO. The approach is detailed in the following sections.

L. Blackhall is with The Research School of Information Sciences and Engineering, The Australian National University, Canberra, ACT, 0200, Australia lachlan.blackhall@anu.edu.au

M. Rotkowitz is with The Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC, 3010, Australia mcrotk@unimelb.edu.au

978-1-4244-3872-3/09/$25.00 ©2009 IEEE

III. Recursive Sparse Estimation using a Gaussian Sum Filter

A. Recursive Parameter Estimation

Given measurements up to and including k ($Y_k = [y_1, \ldots, y_k]$) and assuming we can express the posterior given these measurements in terms of the posterior given the previous set of measurements as

$$f_k(\theta|Y_k) \;=\; \frac{h(y_k|\theta) f_{k-1}(\theta|Y_{k-1})}{\int h(y_k|\theta) f_{k-1}(\theta|Y_{k-1}) d\theta} \tag{5}$$

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. The MAP estimate is then given as

$$\theta^*_{\text{MAP}} \;=\; \arg\max_{\hat{\theta}} f(\hat{\theta}|Y_k)$$

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 ([6]) 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.

B. 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 simple and frequently used recursive estimator in this instance. 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 [7] and [8] and further detailed in [9]. The primary motivation for using the Gaussian sum filter is that non-Gaussian distributions can be updated recursively. This significantly extends the usefulness of the Kalman filter. We now outline the form of the filter and our development is based upon a more general version in [9].

Using the Gaussian sum filter for parameter estimation can be considered a special case of its operation. 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. 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:

$$y_k = X_k\theta + \varepsilon_k \tag{6}$$

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

$$\theta \;\sim\; \sum_{i=1}^{M} \alpha_i \; \mathcal{N}(\mu_i, B_i) \tag{7}$$

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:

$$\theta|Y_{k-1} \;\sim\; \sum_{i=1}^{M} \alpha_{i,k-1} \; \mathcal{N}(\mu_{i,k-1}, B_{i,k-1}) \tag{8}$$

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:

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

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

$$\begin{aligned}
\Omega_{i,k} &= X_k B_{i,k-1} X_k^T + R_k \\
K_{i,k} &= B_{i,k-1} X_k^T \Omega_{i,k}^{-1} \\
B_{i,k} &= B_{i,k-1} - B_{i,k-1} X_k^T \Omega_{ik}^{-1} X_k B_{i,k-1}^T \\
\hat{y}_{i,k} &= X_k \mu_{i,k-1} \\
\mu_{i,k} &= \mu_{i,k-1} + K_{i,k}(y_k - \hat{y}_{i,k}) \\
\alpha_{i,k} &= \frac{\alpha_{i,k-1} \; \mathcal{N}(y_k;\; \hat{y}_{i,k},\; \Omega_{i,k})}{\sum_{j=1}^{M} \alpha_{j,k-1} \; \mathcal{N}(y_k;\; \hat{y}_{j,k},\; \Omega_{j,k})}
\end{aligned} \tag{10}$$

If we have a Gaussian mixture for our prior distribution given as (7), 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

$$\theta|Y_N \;\sim\; \sum_{i=1}^{M} \alpha_{i,N} \; \mathcal{N}(\mu_{i,N}, B_{i,N}). \tag{11}$$

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

$$\theta^*_{\text{MAP}} \;=\; \arg\max_{\hat{\theta}} \sum_{i=1}^{M} \alpha_{i,N} \; \mathcal{N}(\hat{\theta}; \mu_{i,N}, B_{i,N}). \tag{12}$$

C. The Laplacian Prior as a Gaussian Sum

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.

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 [1] that the higher dimensional Laplacian prior could be approximated as a Gaussian sum by:

$$f_0(\theta) = \sum_{i_1=1}^{M} \cdots \sum_{i_q=1}^{M} \alpha_{i_1,\ldots,i_q}\, \mathcal{N}(\theta;\ 0, B_{i_1,\ldots,i_q}) \qquad (13)$$

The multivariate weightings $\alpha_{i_1,\ldots,i_q}$ arise as the product of the prior probabilities of the q independent Gaussian prior distributions that compose each multivariate Gaussian distribution of Eq. (13). The corresponding multivariate covariances $B_{i_1,\ldots,i_q}$ are the diagonal matrices containing the initial variances of the same q univariate Gaussian distributions.

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 III-B 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.

D. The Bi-Gaussian Filter

We saw in [1] and [2] that it was possible to set $M = 2$ in the recursive sparse estimator and obtain similar performance compared to the case when $M >> 2$. We referred to this as the Bi-Gaussian filter. Setting $M = 2$ is a heuristic adjustment to the algorithm that represents the two potential outcomes for each parameter, that it is either zero or non-zero. This required us to choose variances σ^2_{min} and σ^2_{max} as the variances of the two Gaussian distributions used to model the two possible outcomes of a parameter estimate. In [1] it was necessary to choose $\sigma^2_{\text{min}} \neq 0$ in order to be able to compute a meaningful MAP estimate. However this means that we are not representing the potential for a zero parameter estimate exactly but rather that the parameter is small but finite.

In [2] we saw that when $\sigma^2_{\text{min}} = 0$ the Gaussian sum filter was still well defined. In this situation, however, it was not possible to compute a meaningful MAP estimate and we proceeded as in [10] using the Maximum Probability (MP) approach. 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:

$$i^* = \arg\max_i \alpha_{i,N} \qquad (14)$$

$$\theta^*_{\text{MP}} = \mu_{i^*,N} \qquad (15)$$

This can be compared to the earlier MAP estimate given in Eq. (12). It is this MP Bi-Gaussian estimator that we will be analyzing and extending throughout the remainder of this paper.

We detailed in [2] that when a parameter is initialized with zero mean and variance it would remain as such for subsequent iterations. That is given that the j-th element of $\mu_{i,0}$ is zero and that the j-th row and column of $B_{i,0}$ is zero then the j-th element of $\mu_{i,k}$ is zero and that the j-th row and column of $B_{i,k}$ remain zero for all k. From a computational perspective this implies that it is only necessary to compute the mean and covariance updates for elements of the i-th Kalman filter that are initialized with non-zero variance. This becomes an important consideration when we use the information filter that is detailed in the following section.

IV. The Information Filter

Having detailed the recursive sparse estimator using a Gaussian sum filter we now outline the information filter as described in [9]. Given a Kalman filter with mean (μ) and covariance (B) we define the information matrix and information state vector respectively as:

$$\begin{aligned} Z &= B^{-1} \\ z &= B^{-1}\mu \end{aligned} \qquad (16)$$

It is possible to obtain a recursion for these values using the matrix inversion lemma from the usual Kalman filter equations, as is presented in [9]. Similarly to the Kalman filter, the information 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. For this reason only the measurement update portion of the information filter will be presented. Readers are directed to [9] or [11] for a full presentation of the information filter.

We say that X_k and y_k are the regressor matrix and measurements respectively at iteration k, and thus we obtain the recursive formula for the information matrix and information state vector as:

$$\begin{aligned} Z_k &= Z_{k-1} + X_k^T R_k^{-1} X_k \\ z_k &= z_{k-1} + X_k^T R_k^{-1} y_k \end{aligned} \qquad (17)$$

It should be observed that further computations are required to obtain the covariance matrix and state vector obtained from the normal Kalman filter.

We have shown here the relationship between the Kalman and information filters but it remains to show how the information filter is related to the Gaussian sum filter. In the case of the Gaussian sum filter we have a multitude of Kalman filters running in parallel. However, as each Kalman filter in the Gaussian sum filter does not exchange information while it runs it is possible to convert each Kalman filter running in parallel to an information filter as follows:

$$\begin{aligned} Z_{i,k} &= B_{i,k}^{-1} \\ z_{i,k} &= B_{i,k}^{-1}\mu_{i,k} \end{aligned} \tag{18}$$

Using this relationship we see that it is possible to convert between the state vector and the information state vector as well as converting between the covariance matrix and the information matrix as necessary. We see in future sections, somewhat surprisingly, that it is possible to use only a single information matrix to represent the entire bank of Kalman filters from the Gaussian sum filter in the case of the MP Bi-Gaussian filter where we have $\sigma_{\text{min}} = 0$. Similarly to the Kalman filter it is only necessary to compute the information filter for those parameters that are initialized with non-zero variance, the reasons for which were detailed at the end of the previous section.

V. Relationship between Kalman and Information Filters

We now present the main result of this paper, namely that a single information filter can represent the evolution of the exponential number of Kalman filters that were used in the Gaussian sum filter. While we cannot overcome the exponential model selection problem the model update is no longer exponential, considerably reducing the storage and computational requirements of the algorithm. We show this as follows.

It is possible to consider the Gaussian sum filter as M^q Kalman filters in parallel. Using the Bi-Gaussian filter we have 2^q Kalman filters. In each of these Kalman filters we consider that some of the parameters are active and some inactive. The active parameters in a given filter of the Gaussian sum filter are those parameters that are initialized with non-zero mean and covariance in that filter. Conversely, the inactive parameters are those initialized with a mean and covariance of zero. We can see this as follows. If a parameter is initialized with zero mean and variance then we detailed in [2] that it will remain this way for subsequent iterations. As the parameter will always have a zero magnitude we say it is inactive, representing a reduction of the model order. Conversely a parameter that has non-zero variance will have non-zero mean and variance for subsequent iterations thus being an active member in the model represented by that Gaussian. We can formalize this as having a set of active (non-zero) model parameters $\mathcal{I}_i = \{p_1^{(i)}, \cdots, p_{s_i}^{(i)}\} \subseteq \{1, \cdots, q\}$, $\forall\ i \in \{1, \cdots, 2^q\}$ Kalman filters running in parallel where $1 \leq p_1^{(i)} < \cdots < p_{s_i}^{(i)} \leq q$ and clearly $s_i = |\mathcal{I}_i|$. From our earlier statement that initializing the covariance and mean of inactive parameters results in them remaining as such for subsequent iterations implies that $\forall i, k, l \ \ \forall j \notin \mathcal{I}_i$ we have $B_{i,k}(j,l) = B_{i,k}(l,j) = 0$.

We now, for each $i \in \{1, 2, \cdots, 2^q\}$, define extraction matrices $P_i \in \mathbb{R}^{s_i \times q}$ as:

$$P_i(j,l) = \begin{cases} 1 & \text{if } l = p_j^{(i)} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

for all $j \in \{1, \cdots, s_i\}$ and for all $l \in \{1, 2, \cdots, q\}$.

We can then define:

$$X_{i,k} = X_k P_i^T \tag{20}$$

such that $X_{i,k} \in \mathbb{R}^{r \times s_i}$ contains the columns of X_k corresponding to the active parameters in $\mathcal{I}_i$.

The information filter corresponding to the i-th Kalman filter of the Gaussian sum is then given by:

$$\begin{aligned} Z_{i,k} &= Z_{i,k-1} + X_{i,k}^T R_k^{-1} X_{i,k} \\ z_{i,k} &= z_{i,k-1} + X_{i,k}^T R_k^{-1} y_k \end{aligned} \tag{21}$$

Where $Z_{i,k} \in \mathbb{R}^{s_i \times s_i}$ and $z_{i,k} \in \mathbb{R}^{s_i \times 1}$ are the information matrix and information state vector respectively, $R_k \in \mathbb{R}^{r \times r}$ is the Gaussian measurement noise and $y_k \in \mathbb{R}^{r \times 1}$ are the observations at iteration k. We discussed earlier that it was not necessary to evolve the inactive parameters as they will remain zero for all iterations of the algorithm. For this reason we have that the Gaussian of the i-th Kalman filter is $|\mathcal{I}_i|$ dimensional.

We denote by $\bar{Z}_k$ and $\bar{z}_k$ the information matrix and information state vector if we assume all parameters are active. That is when $P_i = I$, where I is the $q \times q$ dimensional identity matrix. This gives us:

$$\begin{aligned} \bar{Z}_k &= \bar{Z}_{k-1} + X_k^T R_k^{-1} X_k \\ \bar{z}_k &= \bar{z}_{k-1} + X_k^T R_k^{-1} y_k \end{aligned} \tag{22}$$

where X_k is the complete set of regressors and we have dropped the subscript i to indicate that this is the complete information filter for the parameter estimation problem with q parameters.

This enables us to define:

$$\begin{aligned} Z_{i,0} &= P_i \bar{Z}_0 P_i^T \\ z_{i,0} &= P_i \bar{z}_0 \end{aligned} \tag{23}$$

as the initial information matrix and information state vector corresponding to the active parameters in $\mathcal{I}_i$. We can now state the main result of this paper.

Given the definitions previously we see that the following relations hold:

$$\begin{aligned} Z_{i,k} &= P_i \bar{Z}_k P_i^T \\ z_{i,k} &= P_i \bar{z}_k \end{aligned} \tag{24}$$

We show this by induction. We have already seen that the relation holds for $k = 0$ above in Eq. (23).

Now we assume the result holds for $k - 1$ and we show it for k.

$$\begin{aligned} P_i \bar{Z}_k P_i^T &= P_i(\bar{Z}_{k-1} + X_k^T R_k^{-1} X_k) P_i^T \\ &= P_i \bar{Z}_{k-1} P_i^T + P_i X_k^T R_k^{-1} X_k P_i^T \\ &= Z_{i,k-1} + X_{i,k}^T R_k^{-1} X_{i,k} \\ &= Z_{i,k} \end{aligned} \tag{25}$$

and finally,

$$\begin{aligned} P_i\bar{z}_k &= P_i(\bar{z}_{k-1} + X_k^T R_k^{-1} y_k) \\ &= P_i\bar{z}_{k-1} + P_i X_k^T R_k^{-1} y_k \\ &= z_{i,k-1} + X_{i,k}^T R_k^{-1} y_k \\ &= z_{i,k} \end{aligned} \tag{26}$$

as required.

This implies that if we extract the sub-matrix of $\bar{Z}_k$ where we take elements that are in both the rows and columns of $\mathcal{I}_i$ then we have the information matrix corresponding to the i-th Kalman filter in our Gaussian sum filter. Equivalently taking the sub-vector of the information state vector $\bar{z}_k$ where we take elements corresponding to the active parameters from $\mathcal{I}_i$ we have the information state vector corresponding to the i-th Kalman filter in our Gaussian sum filter.

What we have just shown is that it is possible to use a single information filter to represent all the information filters corresponding to the 2^q Kalman filters without having to represent each information filter explicitly. In the Gaussian sum filter we are updating 2^q Kalman filters with the same measurement which is computationally expensive. Additionally we need to store the covariance matrix and mean vector for each of the Gaussians we are iterating. In using the information filter we are required to update only a single filter, obtaining the corresponding mean and covariance only when required to determine the fitness of a given model in representing the measurements obtained. In this way we have a single information matrix and a single information state vector and yet it is possible to compute the parameter estimates and covariance matrices for all 2^q Kalman filters from the original Gaussian sum filter. In the context of sparse estimation this is a major computational advantage.

A. *Updating $\alpha_{i,k}$ using the Information Filter*

We have shown previously the form of the measurement update for the information filter. The update we have shown is only for the data accumulation step and does not give the updated covariance matrix ($B_{i,k}$) or parameter vector ($\mu_{i,k}$) as was obtained from the Gaussian sum filter of Eq. (10). Both $\mu_{i,k}$ and $B_{i,k}$ can obviously be obtained from the $Z_{i,k}$ and $z_{i,k}$ of the information filter.

The goal of the recursive sparse estimator is to compute the values of $\alpha_{i,k}$ that indicate the likelihood that the i-th Gaussian of the Gaussian sum filter, corresponding to the i-th parameter model, represents the true parameter vector θ after k observations have been processed. To compute $\alpha_{i,k}$ as in Eq. (10) requires knowledge of $\mu_{j,k-1}$ and $B_{j,k-1}$ for all j. In order to compute $\mu_{i,k-1}$ we must first compute:

$$B_{i,k-1} = Z_{i,k-1}^{-1} \tag{27}$$

so that we can compute

$$\mu_{i,k-1} = B_{i,k-1} z_{i,k-1} \tag{28}$$

thus allowing the computation of $\Omega_{i,k}$ and $\alpha_{i,k}$ as given in Eq. (10).

An additional advantage of the information filter seen in Table I is that it is possible to incorporate a large number of measurements into the information filter with a very small computational requirement, relative to that of the Gaussian sum filter. When using the Gaussian sum filter the measurement updates are exponential as all 2^q Gaussians must be updated with each measurement. In contrast it is computationally cheap to update the single information filter thus offering significant advantages to updating the information filter with many measurements before converting into the Gaussian sum filter representation.

B. *Computation and Storage Requirements*

In order to evaluate possible performance gains by using the information filter we can refer to [12] which details the number of computations in a single information and Kalman filter. In Table I we compare the number of computations (Multiplications, Divisions and Square Roots) necessary to update the Kalman and information filters for recursive sparse estimation where q parameters are being estimated and r observations are being incorporated each iteration.

Table I shows the computations required for updating the filter after a measurement is received through to computing the 2^q covariance matrices and parameter estimates allowing the computational complexity of both schemes to be directly compared. Table I does not include the computations for computing $\alpha_{i,k}$, however the computations are identical for both the Gaussian sum filter and the corresponding information filter and hence a direct comparison is not required.

Filter	**Multiplications**
Kalman	$\sum_{p=1}^{q}\binom{q}{p}(\frac{3}{2}p^2+\frac{9}{2}p)r$
Info.	$(\frac{q^2}{2}+\frac{5}{2}q)r+\sum_{p=1}^{q}\binom{q}{p}(\frac{p^3}{6}+\frac{3}{2}p^2-\frac{5}{3}p)$
Filter	**Divisions**
Kalman	$2^q r$
Info.	$\sum_{p=1}^{q}\binom{q}{p}2p-1$
Filter	**Square Roots**
Kalman	0
Info.	$\sum_{p=1}^{q}\binom{q}{p}p$

TABLE I: Number of computations for the Gaussian sum filter and the corresponding information filter. As a brief example we find that for $q = 10$ and $r = 1$ both filters require approximately 7×10^4 computations, with the Kalman filter requiring slightly fewer computations than the information filter. The computational advantage of the information filter over the Kalman filter for incorporating multiple measurements can be seen for $q = 10$ and $r = 2$. In this instance the information filter still requires approximately 7×10^4 computations with the Kalman filter now requiring over 1.3×10^5 computations. This computational advantage increases as the value of r increases.

The binomial coefficients in Table I arise due to the different number of active parameters in each of the 2^q models. In Table I we see that as the value of r increases the Kalman filter requires many more computations than the information filter. This implies that when updating the information filter with multiple measurements significant performance improvements over the current Kalman filter approach can be realized.

The information filter also requires significantly less storage as all 2^q Gaussians are represented in a single matrix for the measurement update. In this way we do not require storage for the exponential number of Gaussians as we did before and this makes the algorithm amenable to implementation on systems with storage constraints.

VI. Numerical Results

We now compare the computational performance of the recursive sparse estimators using both Kalman and information filters. 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 is 0.5. The non-zero parameters were chosen from a uniform distribution on the interval $[0, 5]$. The prior of each parameter is assumed to be independently distributed and represented as a sum of two Gaussian distributions ($M = 2$), both having an a priori weighting ($\alpha = 0.5$) for both variance components σ^2_{min} and σ^2_{max}. For the Kalman filter the two Gaussians have $\sigma^2_{\text{min}} = 0$ and $\sigma^2_{\text{max}} = 1 \times 10^4$. For the information filter the two Gaussians have $\sigma^2_{\text{min}} = 0$ and $\sigma^2_{\text{max}} = \infty$. Initializing the Gaussian corresponding to each of the q parameters of the information filter with $\sigma^2_{\text{max}} = \infty$ corresponds to initializing the information matrix as the $q \times q$ matrix of zeros. The initial mean of each multivariate Gaussian is a $q \times 1$ vector of zeros for both the Kalman and information filters. 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 $y_k = X_k\theta + \varepsilon_k$.

Filter	Eqn(s). Computed	Comp. Time (s)
Kalman	(10)	71.94
Information	(22), (27), (10)*	64.58
Information (Update)	(22)	0.05

TABLE II: Computation time for the Kalman and information filters in computing fifty iterations of parameter estimation with $q = 10$ and $N = 30$. Note that Eq. (10)* refers to the computation of $\Omega_{i,k}$ and $\alpha_{i,k}$ given in Eq. (10).

The computational results shown in Table II are for updating the filter(s) after a measurement is received through to computing the 2^q covariance matrices $B_{i,k}$, parameter estimates $\mu_{i,k}$ and weights $\alpha_{i,k}$. The final line of Table II shows the computational time for the update portion of the information filter (i.e. computing only Eq. (22)).

We see that the information filter is able to return identical parameter estimates as those obtained using the MP Bi-Gaussian filter, albeit in a faster time. This is because the computationally intensive step of updating the filter as a measurement is received reduces from updating 2^q Kalman filters to updating a single information filter. The primary computational burden of the information filter (as seen by the final line of the Table II) is the conversion from information filter to covariance matrices and parameter estimates. Finding methods that reduce the frequency or necessity of converting between information filter and Kalman filter representations would provide significant performance increases for computing sparse parameter estimates. Understanding more fully how this may be achieved is the most critical aspect of future research.

VII. Conclusion

We have shown here how the information filter can be used in the framework of recursive sparse parameter estimation to obtain identical sparse parameter estimates to those of the Gaussian sum filter approach, whilst reducing the computation time and storage requirements of the filter. The reduced computational and storage requirements thus improve the efficiency with which parameter estimates can be updated recursively. This approach suggests modifications that can be made to further increase the performance of the filter. Providing equivalent Kalman and information filter representations of the recursive sparse parameter estimation framework improves the flexibility as it is possible to convert between the two representations as required to achieve the desired performance.

Obtaining a better understanding of how this representation 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.

References

[1] L. Blackhall and M. Rotkowitz, “Recursive sparse estimation using a gaussian sum filter,” in *Proceedings of the International Federation for Automatic Control World Congress*, 2008.

[2] ——, “Maximum aposteriori vs maximum probability recursive sparse estimation,” in *Proceedings of the European Control Conference*, 2009.

[3] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[4] E. J. Candès, “Compressive sampling,” in *International Congress of Mathematicians*. European Mathematical Society, 2006.

[5] D. L. Donoho and J. Tanner, “Sparse nonnegative solution of underdetermined linear equations by linear programming,” *Tech. Report*, 2005.

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

[7] H. Sorenson and D. Alspach, “Recursive Bayesian estimation using Gaussian sums,” *Automatica*, vol. 7, pp. 465–479, 1971.

[8] D. L. Alspach and H. W. Sorenson, “Nonlinear Bayesian estimation using Gaussian sum approximations,” *IEEE Transactions on Automatic Control*, vol. AC-17, no. 4, pp. 439–448, August 1972.

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

[10] P. Stoica and Y. Selen, “Model-order selection: A review of information criterion rules,” *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 36–47, July 2004.

[11] G. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, 1998.

[12] G. Bierman, “A comparison of discrete linear filtering algorithms,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-9, no. 1, pp. 28–37, January 1973.