Variational Bayesian Dynamic Compressive Sensing

Hongwei Wang†*, Hang Yu*, Michael Hoy*, Justin Dauwels*, and Heping Wang†
*School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
†School of Aeronautics, Northwestern Polytechnical University, China

Abstract—Dynamic compressed sensing (DCS) has recently gained popularity as a successful approach to recovering dynamic sparse signals. In this paper, we attack the problem from a Bayesian perspective. The proposed model imposes sparse constraints on both the unknown sparse signal and its temporal innovation via t priors. Due to the conjugacy between the priors and likelihoods, we are able to propose a computationally efficient mean-field variational Bayes algorithm to learn the model without parameter tuning. We consider both the online and offline scenarios, and demonstrate via numerical experiments that the proposed methods are superior to alternatives in terms of both reconstruction accuracy and computational time.

I. INTRODUCTION

Compressive sensing (CS) provides an effective tool to reconstruct high-dimensional latent sparse signals from their noisy, low-dimensional, linear measurements. So far, most literature on CS focuses on static data, whereas there is also an important class of dynamic problems, including dynamic MRI [1], target tracking [2], and high-speed video capture [3]. It is therefore essential to develop DCS tools to deal with time-varying sparse signals.

One simple approach is to consider each time step independently and apply standard CS techniques, such as Danztig selector [4], basic pursuit denoising [5] and LASSO [6]. Although it is an efficient online algorithm, this simple CS method often fails since a significant amount of relevant data is discarded. On the contrary, the in-one-go method assumes that the signal is static and bundles the entire collection of measurements [7]. However, this assumption can jeopardize the performance when processing signals with strong dynamics.

As a compromise between the aforementioned approaches, Zachariah et al. [8] propose a dynamic iterative pursuit (DIP) algorithm. As in the Kalman filter (KF), it borrows the information of the latent signal at the previous time point t − 1 to aid in the estimation of the signal at t. After capturing the dependence between consecutive latent signals, this method significantly outperforms the simple CS and the in-one-go algorithm. Unfortunately, the KF prediction step in the method boils down to an ℓ2 norm penalty on the signal innovation (i.e., difference between two consecutive states). Thus, this method only allows a smooth variation of the latent signal across time. Compressive sensing on Kalman filtered residual (KF-CS-residual) [9] provides a recipe to this problem by employing a CS step on the filtering error. An extension to this method replaces the KF with a least squares (LS) estimator (i.e., LS-CS-residual) [10]. These methods are equivalent to imposing an ℓ1 norm penalty on the innovation. The resulting sparse innovation, however, does not guarantee that the latent signal is also sparse. In other words, the latent signal can become dense as time proceeds, leading to biased estimates.

On the other hand, Bayesian models are also available which combine two independent processes: a binary Markov process for support set estimation and a Gauss Markov process for amplitude estimation [11]. Efficient approximate message passing (AMP) is then developed to perform inference [11]. However, the approximation may result in accuracy loss.

In this paper, we propose a novel Bayesian framework to tackle the DCS problem. To ensure sparsity, we impose t-priors on the components of the latent sparse signals as well as the innovation. Since a t prior can be interpreted as a zero-mean Gaussian distribution with a conjugate Gamma prior, the resulting model allows conjugacy between all priors and likelihoods, thus facilitating the derivation of efficient mean-field variational Bayes (VB) algorithms. The resulting algorithm closely resembles KF. However, it has a significant benefit that the parameters controlling the signal dynamics (i.e., the penalty parameters) can be determined in an automatic manner without any tuning.

A framework similar to ours is proposed in [12]. Unfortunately, the authors impose Laplace priors rather than t priors, leading to non-conjugacy between the priors and likelihoods. As a consequence, they can only resort to the time-consuming sequential Monte-Carlo sampling method to learn the model. Furthermore, it is troublesome to select a proper sampler as pointed out in [11]. In contrast, we follow the VB approach and circumvent the issues of the sampling method. Another contribution of our paper is that we further consider both offline and online scenarios. For offline scenarios we provide a learning algorithm which is able to learn the covariance of the measurement noise from data.

Through numerical experiments with synthetic data, we show that the proposed methods can reliably reconstruct the latent sparse states, and are quite robust to relatively large temporal evolutions to the signal. We provide comparisons to the aforementioned works, and demonstrate that our method is able to yield more accurate results with a smaller computational time.

The remainder of the paper is organized as follows: we explain the Bayesian formulation of DCS in Section II, and

The first two authors contributed equally to this work.

This research was supported by MOE (Singapore) ACRF Tier 2 grant M4020187.
then develop efficient offline and online variational Bayes algorithms in Section III. Numerical results are presented in Section IV. Finally, conclusions are drawn in Section V.

II. BAYESIAN FORMULATION OF DYNAMIC COMPRESSIVE SENSING

A. State Space Model for DCS

We consider the following linear state-space model (SSM):

\begin{align}
\mathbf{x}_t &= F_t \mathbf{x}_{t-1} + \mathbf{u}_t, \\
\mathbf{y}_t &= H_t \mathbf{x}_t + \mathbf{v}_t,
\end{align}

where for all \( t \in \{1, \cdots, T\} \), \( F_t \in \mathbb{R}^{m \times n} \) is a known state transition matrix; \( H_t \in \mathbb{R}^{m \times n} \) is a known measurement matrix (or sensing matrix); \( \mathbf{v}_t \) is zero-mean white Gaussian noise with covariance \( R_t \) and \( \mathbf{u}_t \) is the innovation during the signal evolution. We emphasize that \( \mathbf{u}_t \) is also assumed to be sparse so as to handle drastic changes in support set at time \( t \).

The main task is to reconstruct dynamically changing sparse signals \( \mathbf{x}_{1:T} \) from noisy linear measurements \( \mathbf{y}_{1:T} \). The above problem can be equivalently written as [12]:

\[
\hat{\mathbf{x}}_t = \arg\min(\mathbf{y}_t - H_t \mathbf{x}_t)'R_t^{-1}(\mathbf{y}_t - H_t \mathbf{x}_t) + \beta_t ||\mathbf{x}_t - F_t \mathbf{x}_{t-1}||_1 + \lambda_t ||\mathbf{x}_t||_1,
\]

where \( \mathbf{x}' \) is the transpose of \( \mathbf{x} \), and \( \beta_t \) and \( \lambda_t \) are the penalty parameters which should be tuned carefully in order to recover the sparsity pattern of \( \mathbf{x}_t \). Moreover, we are often interested in learning the noise covariance matrix \( R_t \) from the data. To address these issues, we propose to formulate the above problem (3) from a Bayesian perspective.

B. Bayesian Representation

First, we express the \( \ell_1 \) norm constraints in a Bayesian manner. To achieve a sparse \( \mathbf{x}_t \), we impose independent zero-mean Gaussian priors with precision \( \lambda_t^1 \) on each entry \( x_t^i \) of \( \mathbf{x}_t \), that is,

\[
p(x_t^i | \lambda_t^1) \propto \sqrt{\lambda_t^1} \exp \left[-\frac{\lambda_t^1}{2} (x_t^i)^2 \right].
\]

Note that when \( \lambda_t^1 \) takes very large values, this prior can successfully shrink the corresponding entry \( x_t^i \) to zero, thus yielding a sparse \( \mathbf{x}_t \). We further impose conjugate Gamma hyperprior on the precisions \( \lambda_t^1 \):

\[
p(\lambda_t^1) = \text{Gamma}(\lambda_t^1; a_0, b_0) \propto \lambda_t^{a_0-1} \exp(-b_0 \lambda_t^1).
\]

The shape parameter \( a_0 \) and the rate parameter \( b_0 \) are set to be small values (e.g., \( 10^{-10} \)) such that the hyperprior is non-informative. Interestingly, after integrating out \( \lambda_t^1 \),

\[
p(x_t^i | a_0, b_0) = \int p(x_t^i | \lambda_t^1)p(\lambda_t^1) d\lambda_t^1
\]

\[
= \frac{\Gamma(a_0 + \frac{1}{2})}{\Gamma(a_0) \sqrt{2\pi b_0}} \left[ 1 + \frac{1}{2b_0} (x_t^i)^2 \right]^{-a_0 + \frac{1}{2}},
\]

where \( \Gamma(\cdot) \) is a Gamma function. Therefore, we essentially put a \( t \)-prior on \( x_t^i \). Such shrinkage prior is often used in the framework of sparse Bayesian learning [13]. Interestingly, the \( \ell_1 \) norm in Eq. (3) is equivalent to a Laplace prior [14]. Although Laplace priors can also be regarded as a scale mixture of zero-mean Gaussians [14], the hyperprior on the precisions \( \lambda_t^1 \) is an inverse Gamma distribution that is not conjugate to \( p(x_t^i | \lambda_t^1) \) (i.e., the likelihood of \( \lambda_t^1 \)). Thus, for the sake of tractability, we resort to \( t \)-priors. Akin to the \( \mathbf{x}_t \), we can model the state transmission probability as:

\[
p(\mathbf{x}_t | \mathbf{x}_{t-1}) = N(\mathbf{x}_t; F_t \mathbf{x}_{t-1}, \text{diag}(\beta_t)^{-1}),
\]

\[
p(\beta_t^1) = \text{Gamma}(\beta_t^1; c_0, b_0),
\]

where \( \text{diag}(\beta_t) \) is a diagonal matrix with \( \beta_t \) on the diagonal. Similarly, \( c_0 \) and \( d_0 \) are also set to be small.

On the other hand, it follows from Eq. (2) that

\[
p(\mathbf{y}_t | \mathbf{x}_t) = N(\mathbf{y}_t; H_t \mathbf{x}_t, R_t).
\]

In the scenario where \( R_t \) is unknown, we assume that \( R_t \) follows the non-informative Jeffrey’s prior, that is,

\[
p(R_t) \propto \text{det}(R_t)^{-\frac{n+2}{2}}.
\]

Furthermore, we need to assume \( R_t \) to be time-invariant in order to make the problem identifiable. Therefore, we simplify the notation as \( \tilde{R} \).

Altogether, the resulting graphical model can be depicted as in Fig 1.

---

Fig. 1: SSM based dynamic compressive sensing.

In the above figure, to facilitate the construction of the Bayesian network, we equivalently write \( p(x_t^i | \lambda_t^1) \) as \( p(z_t^i | x_t^i, \lambda_t^1) \), which is Gaussian distribution \( N(z_t^i; x_t^i, 1/\lambda_t^1) \) with mean \( x_t^i \) and precision \( \lambda_t^1 \), and further set \( z_t^i = 0 \) for all \( i \) and \( t \). \( z_t^i \) here can be regarded as pseudo observation. The resulting posterior distribution of all variables can be factorized as:

\[
p(\mathbf{x}_{1:T}, \lambda_{1:T}, \beta_{1:T}, R | \mathbf{y}_{1:T}, \mathbf{z}_{1:T})
\]

\[
\propto \prod_{t=1}^T p(\mathbf{x}_t | \mathbf{x}_{t-1}, \beta_t)p(\mathbf{y}_t | \mathbf{x}_t, R)p(\mathbf{z}_t | \mathbf{x}_t, \lambda_t)p(\lambda_t)p(\beta_t)p(R).
\]

III. VARIATIONAL BAYES INFERENCE

In this section, we develop two VB inference algorithms to learn the aforementioned model. The first one is an offline algorithm which can infer \( \mathbf{x}_{1:T} \) along with \( \lambda_{1:T}, \beta_{1:T}, \) and \( R \) given all the noisy measurements \( \mathbf{y}_{1:T} \). The second algorithm, on the other hand, can update \( \mathbf{x}_t, \beta_t \) and \( \lambda_t \) online given the measurements up to time \( t \) (i.e., \( \mathbf{y}_{1:t} \)). Under this scenario, we assume that the noise covariances \( R_t \) are given in advance.

A. Offline Algorithm

Our objective here is to approximate the intractable exact posterior \( p(\mathbf{x}_{1:T}, \lambda_{1:T}, \beta_{1:T}, R | \mathbf{y}_{1:T}, \mathbf{z}_{1:T}) \) by a tractable variational distribution \( q(\mathbf{x}_{1:T}, \lambda_{1:T}, \beta_{1:T}, R) \) for simplicity, we apply the mean-field approximation, and therefore, the variational distribution can be factorized as:

\[
q(\mathbf{x}_{1:T}, \lambda_{1:T}, \beta_{1:T}, R) = q(\mathbf{x}_{1:T})q(R) \prod_{t=1}^T \prod_{i=1}^n q(\lambda_t^i)q(\beta_t^1).
\]
Algorithm 1 Offline VB Algorithm for DCS

Input: \(y_{1:T}, q(x_0) = p(x_0)\).
Output: \(q(x_t), q(\lambda_t), q(\beta_t)\) for \(t = 1 : T\), and \(q(R)\).
Initialize the variational parameters \(a_t^{(0)}, b_t^{(0)}, c_t^{(0)}, d_t^{(0)},\ \Psi_R^{(0)},\ \nu_R^{(0)}\), and iteration number \(\kappa = 1\)
repeat
  1. Update \(\hat{Q}_t^{(\kappa)}, \hat{F}_t^{(\kappa)}, \hat{R}^{(\kappa)}\) as in (15)-(17).
  2. Compute \(q(x_t)\) and \(q(x_t, x_{t-1})\) by forward-backward Kalman filtering algorithm in the Gaussian linear SSM as defined in (13) and (14).
  3. Update the parameters of \(q(\lambda_{1:T}), q(\beta_{1:T})\) and \(q(R)\) as in (18), (19), and (21).
  4. \(\kappa = \kappa + 1\).
until convergence criterion is met

Note that we do not factorize \(q(x_{1:T})\) as \(\prod_{t=1}^T q(x_t)\) since otherwise we would lose crucial information about the latent Markov chain required for accurate inference [15].

Next, we can derive the VB update rules as follows. For the sparse signals \(x_{1:T}\),
\[
q(x_{1:T}) \propto \exp \left\{ \sum_{t=1}^T \left( \log p(x_t|x_{t-1}, \beta_t) \right) q(\beta_t) + \sum_{t=1}^T \left( \log p(y_t|x_t, R) \right) q(R) + \sum_{t=1}^T \left( \log p(z_t|x_t, \lambda_t) \right) q(\lambda_t) \right\},
\]
(12)
where \(\langle g(x) \rangle_{q(x)}\) is the expectation of \(g(x)\) over distribution \(q(x)\). This expression can be regarded as the joint density function of a Gaussian linear SSM. In this model, the transition and omission distribution are:
\[
q(x_t|x_{t-1}) = \mathcal{N}(x_t; \hat{F}_t x_{t-1}, \hat{Q}_t),
\]
(13)
\[
q(y_t|x_t) = \mathcal{N}(y_t; H_t x_t, \hat{\Sigma}),
\]
(14)
where
\[
\hat{Q}_t = \left( \langle \text{diag}(\beta_t) \rangle_{q(\beta_t)} + \langle \text{diag}(\lambda_t) \rangle_{q(\lambda_t)} \right)^{-1},
\]
(15)
\[
\hat{F}_t = \hat{Q}_t \left( \langle \text{diag}(\beta_t) \rangle_{q(\beta_t)} \right) F_t,
\]
(16)
\[
\hat{R} = (R^{-1})^{-1},
\]
(17)
We apply the forward covariance Kalman filter and backward information Kalman filter on the SSM to obtain \(q(x_t) = \mathcal{N}(x_t; m_t, \Sigma_t)\) and \(q(x_{t-1}, x_t)\).

We now focus on updating \(q(\lambda_t^{(\kappa)})\) and \(q(\beta_t^{(\kappa)})\). Specifically,
\[
q(\lambda_t^{(\kappa)}) = \text{Gamma}(\lambda_t^{(\kappa)}; a_t^{(\kappa)}, b_t^{(\kappa)}),
\]
(18)
\[
q(\beta_t^{(\kappa)}) = \text{Gamma}(\beta_t^{(\kappa)}; c_t^{(\kappa)}, d_t^{(\kappa)}),
\]
(19)
where \(a_t = a_0 + 1/2\), \(b_t = b_0 + (\langle x_t^2 \rangle_{q(x_t)})/2\), \(c_t = c_0 + 1/2\), \(d_t = d_0 + D_{ii}/2\), and \(D_{ii}\) is the \(i\)th diagonal element of the following matrix \(D\):
\[
D = (m_t - F_t m_{t-1}) (m_t - F_t m_{t-1})^\top + \Sigma_t + F_t \Sigma_{t-1} F_t^\top - 2 \Sigma_{t-1} \left\{ \left[ \hat{Q}_t (K_{t+1} H^\top \hat{R}^{-1} H + I)^{-1} \right] \right\}^\top.
\]
(20)
In the above expression, \(K_{t+1}\) is the precision matrix of \(q(x_t|y_{t+1}, \ldots, y_T)\) in the backward pass, and \(I\) is a \(n \times n\) unit matrix.

Finally, \(q(R)\) can be updated as an inverse Wishart distribution:
\[
q(R) = \mathcal{W}^{-1}(R; \Psi_R, \nu_R),
\]
(21)
where
\[
\Psi_R = \sum_{t=1}^T \left[ H_t (m_t m_t^\top + \Sigma_t) H_t^\top - 2 H_t m_t y_t^\top + y_t y_t^\top \right],
\]
are the scale matrices, and \(\nu_R = T\) is the degrees of freedom.

The resulting offline algorithm is summarized in Algorithm 1.

B. Online Algorithm

In this case, we aim to develop an algorithm analogous to Kalman filter for updating the sparse signals \(x_{1:t}\) online given noisy measurements \(y_{1:t}\). As mentioned before, we assume that covariances \(R_t\) are known. The VB inference method seeks the variational distribution \(q(x_{1:t}, \lambda_{1:t}, \beta_{1:t}) = q(x_{1:t}) \prod_{t=1}^T q(\lambda_t) p(\beta_t|y_t)\) that can best approximate the exact posterior \(p(x_{1:t}, \lambda_{1:t}, \beta_{1:t}|y_{1:t})\). However, to resemble Kalman filter, we need to sidestep the backward pass in the variational inference. In other words, at time \(t\), we only compute \(q(x_t|y_{1:t}), q(\beta_t|y_{1:t})\) and \(q(\lambda_t|y_{1:t})\). The resulting
online algorithm is listed in Algorithm 2 in detail. Note that $\odot$ denotes componentwise division in the algorithm.

IV. Numerical Results

In this section, we test the proposed online variational Bayesian dynamic compressive sensing (VBDCS) model against 1) simple CS [4], 2) KF-CS-residual [9], 3) LS-CS-residual [10], and the offline VBDCS against 4) approximate message passing (AMP-DCS) [11]. We use the Matlab code provided by the authors online for the four benchmark models. To evaluate the performance of different models, we consider computational time (CT), normalized mean square error (NMSE) and time-averaged NMSE (TNMSE) as metrics. The NMSE and TNMSE are defined as $1/n \sum_{i=1}^{n} ||x_i - \hat{x}_i||^2/||x_i||^2$ and $1/T \sum_{t=1}^{T} \text{NMSE}(t)$ respectively.

We generate synthetic data for scenarios of both fixed and varying support set. For fixed support set, we change the support set at every time point by setting $S_t = S_{t-1} \odot U$. We then draw $|S_1|$ samples from $S_1$ uniformly and $|S_+|$ and $|S_-|$ to be made to the previous support set $S_{t-1}$. Next, we compute $S_t = S_{t-1} \cup S_+ \setminus S_-$, where \ denotes set difference. After obtaining the support set, we set the amplitude of those non-zero entries as a Gaussian random walk with white noise $w_t$, and the measurements can be produced via $y_t = H x_t + v$. In all experiments, $T = 100$, $H$ is a $72 \times 256$ matrix whose entries are drawn randomly from $N(0, 1/72)$, initial $x_0$ and elements in $S_+$ follow a continuous uniform distribution $U_{c}(-20, 20)$, $w_t \sim N(0, 0.01)$ and $v \sim N(0, 0.01I)$.

A. Online Case

We summarize the results of the four online models in Table I. Specifically, we consider both scenarios of fixed and changing support set. For fixed support set, we change the cardinality of $S$ to check whether it influences the results of different models. For changing support set, we further consider two cases. In the first case, we only change the support every 5 time points and fix $|S_+| = |S_-| = 1$, such that the latent sparse signal changes slightly across time. In the second case, we change the support set at every time point by setting $|S_+|$ and $|S_-|$ to follow $U_d(1, 5)$, and therefore the latent sparse signal changes much more drastically. All the results are averaged over 100 runs, each with different randomly generated support set. We also depict the estimated cardinality of support set as well as the NMSE over time for slightly and drastically varying support set respectively in Fig. 2 and Fig. 3.

We can find that the simple CS is the fastest in all cases, since it simply performs CS separately for each time point. However, the sample size of measurements (that is only one) at every time point is too small to yield reliable estimates of the latent sparse signal $x_t$. Consequently, the reconstruction accuracy of simple CS is the lowest. On the other hand, after taking into account the sparsity of innovation, the KF-CS-residual and LS-CS-residual performs much better than the simple CS in the cases of fixed and slightly varying support set. However, in the case of slightly varying support set, although the NMSE is small, it keeps increasing with time. This can be explained by the fact that these two methods only enforce sparse constraints on innovation, and can lead to estimates denser than the ground truth in the end as shown in Fig. 2a. The problem is exacerbated in the setting of drastically changing support. The NMSE becomes huge as time proceeds (see Fig. 3b), and the estimated cardinality of the support set strays far from the ground truth (see Fig. 3a).

In contrast, the proposed online VBDCS approach attains the smallest reconstruction error in all cases. The computational time is always shorter than that of the KF-CS-residual.
and LF-CS-residual. Specifically, under the scenario of fixed support set, we can observe that our method is more robust to the increase of \(|S|\) than the KF-CS-residual and LF-CS-residual. On the other hand, the cardinality of the support set closely follows the ground truth, and the NMSE is very small across time, regardless of how drastic the support set changes. We notice that the cardinality of the support set is overestimated at the beginning of the time series, since only several measurements are available at those time points. However, the estimated amplitudes corresponding to the zero entries in the ground truth are small, thus, the NMSE is still small. Indeed, the TNMSE resulting from our VBDCS is several magnitudes smaller than that of benchmark models. By tuning the penalty parameters in a Bayesian manner, the online VBDCS can automatically detect the sparsity pattern of the latent signals, and is amenable to time-varying support sets.

**B. Offline algorithm**

We only consider here the drastically changing support set and compare the performance of the proposed offline VBDCS algorithm with the AMP-DCS in 100 independent trials. The computational time for these two methods is respectively

| Methods            | \(|S| = 10\) | \(|S| = 20\) | \(|S| = 30\) | varying support set |
|--------------------|-------------|-------------|-------------|---------------------|
| online VBDCS       | CT/s        | TNMSE       | CT/s        | TNMSE              |
| simple CS          | 3.5936      | 0.2236      | 3.7411      | 0.2341              |
| KF-CS-residual     | 117.6272    | 9.2896e-5   | 15.7471     | 1.6008e-4           |
| LS-CS-residual     | 117.4406    | 1.1096e-4   | 22.3634     | 1.0866e-4           |

Experimental results show that the proposed approach can reliably reconstruct the latent sparse signal and the computational time for these two methods is respectively

\[
\begin{align*}
\text{CT/s} & \approx 15.7471 \\
\text{TNMSE} & \approx 1.6008e-4
\end{align*}
\]

and \(117.4406\) respectively.

**Fig. 4:** The cardinality of the support set \(|S_t|\) (a) and the amplitude (b) as a function of time \(t\) under the offline scenario.

**REFERENCES**


