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 two 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 predict step in the method boils down to an ℓ_2 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 ℓ_1 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 variatinal 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):

$$\boldsymbol{x}_t = F_t \boldsymbol{x}_{t-1} + \boldsymbol{u}_t, \tag{1}$$

$$\boldsymbol{y}_t = H_t \boldsymbol{x}_t + \boldsymbol{v}_t, \tag{2}$$

where for all $t \in \{1, \cdots, T\}$, $F_t \in \Re^{n\times n}$ is a known state transition matrix; $H_t \in \Re^{m\times n}$ is a known measurement matrix (or sensing matrix); $\boldsymbol{v}_t$ is zero-mean white Gaussian noise with covariance R_t and $\boldsymbol{u}_t$ is the innovation during the signal evolution. We emphasize that $\boldsymbol{u}_t$ 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 $\boldsymbol{x}_{1:T}$ from noisy linear measurements $\boldsymbol{y}_{1:T}$. The above problem can be equivalently written as [12]:

$$\begin{aligned}\hat{x}_t = \operatorname{argmin}(\boldsymbol{y}_t - H_t\boldsymbol{x}_t)' R_t^{-1}(\boldsymbol{y}_t - H_t\boldsymbol{x}_t) \\ + \boldsymbol{\beta}_t \|\boldsymbol{x}_t - F_t\boldsymbol{x}_{t-1}\|_1 + \boldsymbol{\lambda}_t\|\boldsymbol{x}_t\|_1,\end{aligned} \tag{3}$$

where $\boldsymbol{x}'$ is the transpose of $\boldsymbol{x}$, and $\boldsymbol{\beta}_t$ and $\boldsymbol{\lambda}_t$ are the penalty parameters which should be tuned carefully in order to recover the sparsity pattern of $\boldsymbol{x}_t$. 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 ℓ_1 norm constraints in a Bayesian manner. To achieve a sparse $\boldsymbol{x}_t$, we impose independent zero-mean Gaussian priors with precision λ_t^i on each entry x_t^i of $\boldsymbol{x}_t$, that is,

$$p(x_t^i|\lambda_t^i) \propto \sqrt{\lambda_t^i}\exp\Big[-\frac{\lambda_t^i}{2}(x_t^i)^2\Big]. \tag{4}$$

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

$$p(\lambda_t^i) = \text{Gamma}(\lambda_t^i; a_0, b_0) \propto {\lambda_t^i}^{a_0-1}\exp(-b_0\lambda_t^i). \tag{5}$$

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 λ_t^i,

$$p(x_t^i|a_0, b_0) = \int p(x_t^i|\lambda_t^i)p(\lambda_t^i)d\lambda_t^i \tag{6}$$

$$= \frac{\Gamma(a_0+\frac{1}{2})}{\Gamma(a_0)\sqrt{2\pi b_0}}\left[\frac{1}{1+\frac{1}{2b_0}(x_t^i)^2}\right]^{a_0+\frac{1}{2}}, \tag{7}$$

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 ℓ_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 λ_t^i is an inverse Gamma distribution that is not conjugate to $p(x_t^i|\lambda_t^i)$ (i.e., the likelihood of λ_t^i). Thus, for the sake of tractability, we resort to t-priors. Akin to the $\boldsymbol{x}_t$, we can model the state transmission probability as:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; F_t\boldsymbol{x}_{t-1}, \text{diag}(\boldsymbol{\beta}_t)^{-1}), \tag{8}$$

$$p(\beta_t^i) = \text{Gamma}(\beta_t^i; c_0, b_0), \tag{9}$$

where $\text{diag}(\boldsymbol{\beta}_t)$ is a diagonal matrix with $\boldsymbol{\beta}_t$ 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

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t; H_t\boldsymbol{x}_t, R_t). \tag{10}$$

In the scenario where R_t is unknown, we assume that R_t follows the non-informative Jeffrey's prior, that is,

$$p(R_t) \propto \det(R_t)^{-\frac{m+1}{2}}. \tag{11}$$

Furthermore, we need to assume R_t to be time-invariant in order to make the problem identifiable. Therefore, we simplify the notation as R.

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

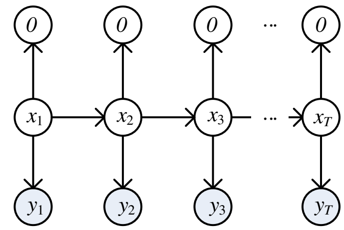


Fig. 1: SSM based dynamic compressive sensing.

In the above figure, to facilitate the construction of the Bayesian network, we equivalently write $p(x_t^i|\lambda_t^i)$ as $p(z_t^i|x_t^i, \lambda_t^i)$, which is Gaussian distribution $\mathcal{N}(z_t^i; x_t^i, 1/\lambda_t^i)$ with mean x_t^i and precision λ_t^i, 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:

$$p(\boldsymbol{x}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{\beta}_{1:T}, R|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$$

$$\propto \prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\beta}_t)p(\boldsymbol{y}_t|\boldsymbol{x}_t, R)p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{\lambda}_t)p(\boldsymbol{\lambda}_t)p(\boldsymbol{\beta}_t)p(R).$$

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 $\boldsymbol{x}_{1:T}$ along with $\boldsymbol{\lambda}_{1:T}$, $\boldsymbol{\beta}_{1:T}$, and R given all the noisy measurements $\boldsymbol{y}_{1:T}$. The second algorithm, on the other hand, can update $\boldsymbol{x}_t$, $\boldsymbol{\beta}_t$ and $\boldsymbol{\lambda}_t$ online given the measurements up to time t (i.e., $\boldsymbol{y}_{1:t}$). 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 $p(\boldsymbol{x}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{\beta}_{1:T}, R|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ by a tractable variational distribution $q(\boldsymbol{x}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{\beta}_{1:T}, R)$. For simplicity, we apply the mean-field approximation, and therefore, the variational distribution can be factorized as:

$$q(\boldsymbol{x}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{\beta}_{1:T}, R) = q(\boldsymbol{x}_{1:T})q(R)\prod_{t=1}^{T}\prod_{i=1}^{n} q(\lambda_t^i)q(\beta_t^i).$$

Algorithm 1 Offline VB Algorithm for DCS

Input: $\boldsymbol{y}_{1:T}$, $q(\boldsymbol{x}_0) = p(\boldsymbol{x}_0)$.
Output: $q(\boldsymbol{x}_t)$, $q(\boldsymbol{\lambda}_t)$, $q(\boldsymbol{\beta}_t)$ for $t = 1 : T$, and $q(R)$.
Initialize the varitional parameters $a_t^{i(0)}$, $b_t^{i(0)}$, $c_t^{i(0)}$, $d_t^{i(0)}$, $\Psi_R^{(0)}$, $\nu_R^{(0)}$, and iteration number $\kappa = 1$
repeat
1. Update $\tilde{Q}_t^{(\kappa)}$, $\tilde{F}_t^{(\kappa)}$, $\tilde{R}^{(\kappa)}$ as in (15)–(17).
2. Compute $q(\boldsymbol{x}_t)$ and $q(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})$ by forward-backward Kalman filtering algorithm in the Gaussian linear SSM as defined in (13) and (14).
3. Update the parameters of $q(\boldsymbol{\lambda}_{1:T})$, $q(\boldsymbol{\beta}_{1:T})$ and $q(R)$ as in (18), (19), and (21).
4. $\kappa = \kappa + 1$.

until convergence criterion is met

Note that we do not factorize $q(\boldsymbol{x}_{1:T})$ as $\prod_{t=1}^{T} q(\boldsymbol{x}_t)$ 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 $\boldsymbol{x}_{1:T}$,

$$
\begin{aligned}
q(\boldsymbol{x}_{1:T}) \propto \exp\Bigg\{ &\sum_{t=1}^{T} \langle \log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\beta}_t)\rangle_{q(\boldsymbol{\beta}_t)} \\
&+ \sum_{t=1}^{T} \langle \log p(\boldsymbol{y}_t|\boldsymbol{x}_t, R)\rangle_{q(R)} \\
&+ \sum_{t=1}^{T} \langle \log p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{\lambda}_t)\rangle_{q(\boldsymbol{\lambda}_t)} \Bigg\}, \qquad (12)
\end{aligned}
$$

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:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \tilde{F}_t\boldsymbol{x}_{t-1}, \tilde{Q}_t), \qquad (13)$$

$$q(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t; H_t\boldsymbol{x}_t, \tilde{R}), \qquad (14)$$

where

$$\tilde{Q}_t = \left[\langle \mathrm{diag}(\boldsymbol{\beta}_t)\rangle_{q(\boldsymbol{\beta}_t)} + \langle \mathrm{diag}(\boldsymbol{\lambda}_t)\rangle_{q(\lambda_t)}\right]^{-1}, \qquad (15)$$

$$\tilde{F}_t = \tilde{Q}_t\langle \mathrm{diag}(\boldsymbol{\beta}_t)\rangle_{q(\boldsymbol{\beta}_t)}F_t, \qquad (16)$$

$$\tilde{R} = \langle R^{-1}\rangle_{q(R)}^{-1}, \qquad (17)$$

We apply the forward covariance Kalman filter and backward information Kalman filter on the SSM to obtain $q(\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{m}_t, \Sigma_t)$ and $q(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t)$.

We now focus on updating $q(\lambda_t^i)$ and $q(\beta_t^i)$. Specifically,

$$q(\lambda_t^i) = \mathrm{Gamma}(\lambda_t^i; a_t^i, b_t^i), \qquad (18)$$

$$q(\beta_t^i) = \mathrm{Gamma}(\beta_t^i; c_t^i, d_t^i), \qquad (19)$$

where $a_t^i = a_0 + 1/2$, $b_t^i = b_0 + \langle (x_t^i)^2\rangle_{q(x_t^i)}/2$, $c_t^i = c_0 + 1/2$, $d_t^i = d_0 + \mathcal{D}_{ii}/2$, and $\mathcal{D}_{ii}$ is the ith diagonal element of the following matrix $\mathcal{D}$:

$$
\begin{aligned}
\mathcal{D} =& (\boldsymbol{m}_t - F_t\boldsymbol{m}_{t-1})(\boldsymbol{m}_t - F_t\boldsymbol{m}_{t-1})' + \Sigma_t + F_t\Sigma_{t-1}F_t' \\
&- 2\Sigma_{t-1}\Big\{\left[\tilde{Q}_t(K_{t|t+1} + H'\tilde{R}^{-1}H) + I\right]^{-1}\Big\}'. \qquad (20)
\end{aligned}
$$

In the above expression, $K_{t|t+1}$ is the precision matrix of $q(x_t|y_{t+1}, \cdots, y_T)$ in the backward pass, and I is a $n \times n$ unit matrix.

Finally, $q(R)$ can be updated as an inverse Wishart distribution:

$$q(R) = \mathcal{W}^{-1}(R; \Psi_R, \nu_R), \qquad (21)$$

where

$$\Psi_R = \sum_{t=1}^{T}\Big[H_t\big(\boldsymbol{m}_t\boldsymbol{m}_t' + \Sigma_t\big)H_t' - 2H_t\boldsymbol{m}_t\boldsymbol{y}_t' + \boldsymbol{y}_t\boldsymbol{y}_t'\Big],$$

are the scale matrices, and $\nu_R = T$ is the degrees of freedom. The resulting offline algorithm is summarized in Algorithm 1.

Algorithm 2 Online VB Algorithm for DCS

Input: $\boldsymbol{y}_{1:T}$, $q(\boldsymbol{x}_0) = p(\boldsymbol{x}_0)$, and $R_{1:T}$.
Output: $q(\boldsymbol{x}_t)$, $q(\boldsymbol{\lambda}_t)$, and $q(\boldsymbol{\beta}_t)$, for $t = 1 : T$.
Initialize the variational parameters: $a_t^{i(0)}$, $b_t^{i(0)}$, $c_t^{i(0)}$, and $d_t^{i(0)}$
for $t = 1, \cdots, T$ **do**
  $\kappa = 1$
  repeat
    $\langle \boldsymbol{\lambda}_t\rangle_{q(\boldsymbol{\lambda}_t)} = \boldsymbol{a}_t^{(\kappa-1)} \oslash \boldsymbol{b}_t^{(\kappa-1)}$;
    $\langle \boldsymbol{\beta}_t\rangle_{q(\boldsymbol{\beta}_t)} = \boldsymbol{c}_t^{(\kappa-1)} \oslash \boldsymbol{d}_t^{(\kappa-1)}$;
    $\tilde{Q}_t^{(\kappa)} = [\langle \mathrm{diag}(\boldsymbol{\beta}_t)\rangle_{q(\boldsymbol{\beta}_t)} + \langle \mathrm{diag}(\boldsymbol{\lambda}_t)\rangle_{q(\lambda_t)}]^{-1}$;
    $\tilde{F}_t^{(\kappa)} = \tilde{Q}_t\langle \mathrm{diag}(\boldsymbol{\beta}_t)\rangle_{q(\boldsymbol{\beta}_t)}F_t$;
    $\boldsymbol{m}_{t|t-1}^{(\kappa)} = \tilde{F}_t^{(\kappa)}\boldsymbol{m}_{t-1|t-1}$;
    $\Sigma_{t|t-1}^{(\kappa)} = \tilde{F}_t^{(\kappa)}\Sigma_{t-1|t-1}(\tilde{F}_t^{(\kappa)})' + \tilde{Q}_t^{(\kappa)}$;
    $G_t^{(\kappa)} = \Sigma_{t|t-1}^{(\kappa)}H_t'(H_t\Sigma_{t|t-1}^{(\kappa)}H_t' + R_t)^{-1}$;
    $\boldsymbol{m}_{t|t}^{(\kappa)} = \boldsymbol{m}_{t|t-1}^{(\kappa)} + G_t^{(\kappa)}(\boldsymbol{y}_t - H_t\boldsymbol{m}_{t|t-1}^{(\kappa)})$;
    $\Sigma_{t|t}^{(\kappa)} = (I - G_t^{(\kappa)}H_t)\Sigma_{t|t-1}^{(\kappa)}$;
    for $i = 1, \cdots, n$ **do**
      $a_t^{i(\kappa)} = a_0 + 1/2$;
      $b_t^{i(\kappa)} = b_0 + \big\{\big(m_{t|t}^{i\ (\kappa)}\big)^2 + \big[\Sigma_{t|t}^{(\kappa)}\big]_{ii}\big\}/2$;
      $\mathcal{D}^{(\kappa)} = \Sigma_t^{(\kappa)} + \boldsymbol{m}_t^{(\kappa)}\boldsymbol{m}_t^{(\kappa)\prime} + F_t\big(\Sigma_{t-1}^{(\kappa)} + \boldsymbol{m}_{t-1}^{(\kappa)}\boldsymbol{m}_{t-1}^{(\kappa)\prime}\big)\big(F_t - 2\tilde{F}_t^{(\kappa)}\big)'$;
      $c_t^{i(\kappa)} = c_0 + 1/2$;
      $d_t^{i(\kappa)} = d_0 + D_{ii}^{(\kappa)}/2$;
    end for
    $\kappa = \kappa + 1$.
  until convergence criterion is met
  $\boldsymbol{m}_{t|t} = \boldsymbol{m}_{t|t}^{(\kappa)}$;
  $\Sigma_{t|t} = \Sigma_{t|t}^{(\kappa)}$;
end for

B. Online Algorithm

In this case, we aim to develop an algorithm analogous to Kalman filter for updating the sparse signals $\boldsymbol{x}_{1:t}$ online given noisy measurements $\boldsymbol{y}_{1:t}$. As mentioned before, we assume that covariances R_t are known. The VB inference method seeks the variational distribution $q(\boldsymbol{x}_{1:t}, \boldsymbol{\lambda}_{1:t}, \boldsymbol{\beta}_{1:t}) = q(\boldsymbol{x}_{1:t})\prod_{j=1}^{t}\prod_{i=1}^{n} p(\lambda_j^i)p(\beta_j^i)$ that can best approximate the exact posterior $p(\boldsymbol{x}_{1:t}, \boldsymbol{\lambda}_{1:t}, \boldsymbol{\beta}_{1:t}|\boldsymbol{y}_{1:t}, \boldsymbol{z}_{1:t}, R_t)$. 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 $q(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$,$q(\boldsymbol{\beta}_t|\boldsymbol{y}_{1:t})$ and $q(\boldsymbol{\lambda}_t|\boldsymbol{y}_{1:t})$. The resulting online algorithm is listed in Algorithm 2 in detail. Note that $\oslash$ denotes componentwise division in the algorithm.

IV. Numerical Results

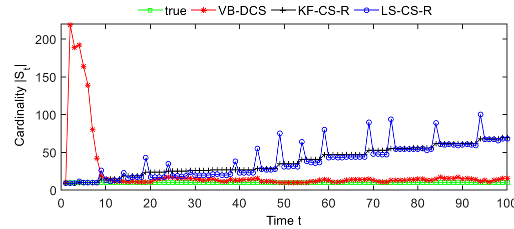
In this section, we test the proposed online variational Bayesian dynamic compressive sensing (VBDCS) model against 1) simple CS [4], 2) KF-CS-residual [9], 3) LS-CS-residual [10], and the offline VBDCS against 4) approximate massage passing (AMP-DCS) [11]. We use the Matlab code provided by the authors 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 $1/n \sum_{i=1}^{n} \|x_t^{(i)} - \hat{x}_t^{(i)}\|_2^2 / \|x_t^{(i)}\|_2^2$ and $1/T \sum_{t=1}^{T} \text{NMSE}(t)$ respectively.

We generate synthetic data for scenarios of both fixed and varying support set as follows. In the scenario of fixed support set, we pick $|S|$ elements in the support set S randomly from the discrete uniform distribution $U_d(1, 256)$, where $|S|$ is the predefined cardinality of S. On the other hand, in the scenario of varying support set, we begin with $|S_0| = 20$. At time t, we first determine the cardinality of additions S_+ and deletions S_- to be made to the previous support set S_{t-1}. We then draw $|S_-|$ samples for S_- from S_{t-1}, and $|S_+|$ samples for S_+ from S_{t-1}^C. Next, we compute $S_t = S_{t-1} \cup S_+ \backslash S_-$, where $\backslash$ 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 u_t^i, and the measurements can be produced via $\boldsymbol{y}_t = H\boldsymbol{x}_t + \boldsymbol{v}$. In all experiments, $T = 100$, H is a 72×256 matrix whose entries are drawn randomly from $\mathcal{N}(0, 1/72)$, initial $\boldsymbol{x}_0$ and elements in S_+ follow a continuous uniform distribution $U_c(-20, 20)$, $u_t^i \sim \mathcal{N}(0, 0.01)$ and $\boldsymbol{v} \sim \mathcal{N}(0, 0.01I)$.

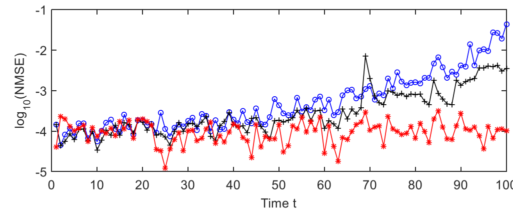
A. Online Case

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 $\boldsymbol{x}_t$. Consequently, the reconstruction accuracy of simple CS is the lowest. On the other hand, after

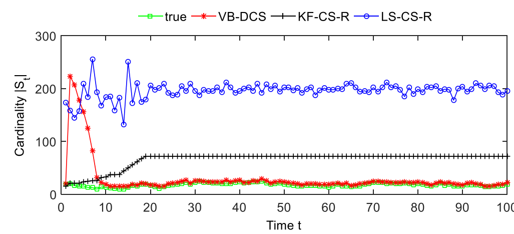


(a)

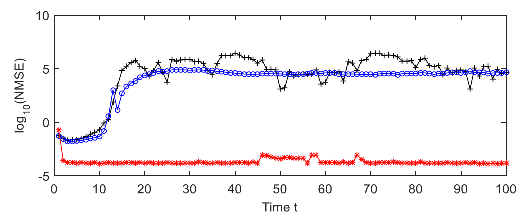


(b)

Fig. 2: The cardinality of the support set $|S_t|$ (a) and the NMSE (b) as a function of time t in the case of slightly varying support set.



(a)



(b)

Fig. 3: The cardinality of the support set $|S_t|$ (a) and the NMSE (b) as a function of time t in the case of drastically varying support set.

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

TABLE I: Comparison of Different Online Algorithms

Methods	fixed support set						varying support set			
	$\|S\| = 10$		$\|S\| = 20$		$\|S\| = 30$		slightly changing S		drastically changing S	
	CT/s	TNMSE	CT/s	TNMSE	CT/s	TNMSE	CT/s	TNMSE	CT/s	TNMSE
online VBDCS	15.7471	1.6008e-4	16.0050	1.4751e-4	17.0865	1.6985e-4	22.3634	1.0866e-4	32.1512	2.2237e-4
simple CS	3.9138	0.0066	3.3184	0.2155	3.5750	0.5147	3.4057	0.0018	3.7335	0.2430
KF-CS-resudial	117.6272	9.2992e-5	135.4112	5.5495e-4	306.4986	0.3381	188.8611	6.7835e-4	329.5027	3.4874e4(failed)
LS-CS-residual	117.4406	1.1096e-4	133.7619	5.8484e-4	317.3876	0.3413	160.0606	0.0024	340.6379	4.4011e5(failed)

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 patter 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 1.55×10^2s and 3.07×10^1s, while the TNMSE is 1.86×10^{-4} and 9.85×10^{-2}. The AMP-DCS is more efficient due to its linear complexity in n, m, and T, whereas the computational complexity of the VBCDS is linear in T but cubic in n and m. However, the reconstruction accuracy of the proposed method is higher than that of the AMP-DCS. We further show in Fig. 4 the estimated cardinality of support set and the amplitude of the latent signal across time. It is obvious that the AMP-DCS can estimate the support set exactly but the estimates of the amplitude is less accurate. On the contrary, although introducing some small false positives in the support set, the proposed VBDCS can infer the amplitude much more accurately than the AMP-DCS. Note that accuracy is often the priority under offline scenarios, and therefore, our VBDCS is preferable in practice.

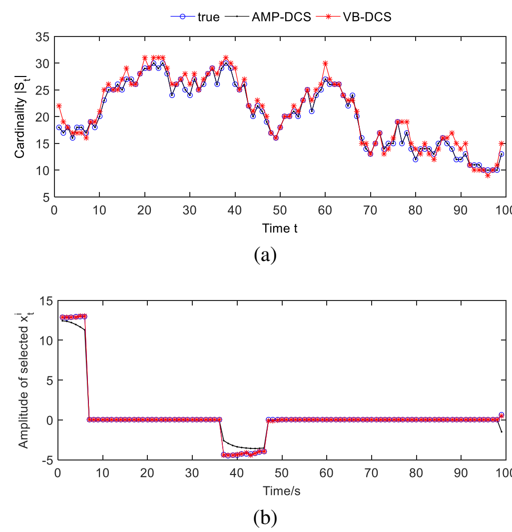


Fig. 4: The cardinality of the support set $|S_t|$ (a) and the amplitude (b) as a function of time t under the offline scenario.

V. Conclusions

We propose a novel Bayesian framework for DCS where a t-prior is enforced on the sparse latent signal and the the innovation to promote sparsity. Efficient variational Bayes algorithms are developed for online and offline inference. Experimental results show that the proposed approach can reliably reconstruct the latent sparse signal and the computational time is usually shorter than for state-of-the-art benchmark methods.

References

[1] C. Qiu and W. Lu and N. Vaswani, "Real-time dynamic MR image reconstruction using Kalman filtered compressed sensing", *Proc. ICASSP*, pp. 393-396, 2009.

[2] R. Sarkar, S. Das and N. Vaswani, "Tracking sparse signal sequence from nonlinear/non-gaussian measurements and aoolications in illumination motion tracking", *Proc. ICASSP*, pp.5954-5958, 2013.

[3] M. S. Asif, D. Reddy, P. Boufounos, and A. Veeraraghavan, "Streaming compressive sensing for high-speed periodic videos," *Proc. ICIP*, pp. 3373-3376. 2010.

[4] E. Candes and T. Tao, "The dantzig selector: statistical estimation when *p* is much larger than *n*", *Ann. Stat*, vol. 35, pp. 2313-2351, 2007.

[5] J. Tropp, "Just relax: Convex programming methods for identifying sparse signals", *IEEE Trans. Inf. Theory*, vol. 52, pp. 1030-1051, 2006.

[6] E. Candes, "The restricted isometry property and its implications for compressed sensing". *Comptes Rendus Mathematique*, pp. 589-592, 2008.

[7] U. Gamper, P. Boesiger and S. Kozerke, "Compressed sensing in dynamic MRI", *Magn. Reson. Med.*, Vol. 59, pp. 365-373, 2008.

[8] D. Zachariah, S. Chatterjee, and M. Jansson, "Dynamic iterative pursuit", *IEEE Trans. Signal processing*, Vol. 60, pp.4967-4972, 2012.

[9] N. Vaswani, "KF-CS: compressive sensing on Kalman filtered residual", arXiv:0912.1628, 2009.

[10] N. Vaswani, "LS-CS-Residual (LS-CS): Compressive Sensing on Least Squares Residual,", *IEEE Trans. Signal Process.*, pp. 3013-3016, 2010.

[11] J. Ziniel, and P. Schiniter, "Dynamic Compressive Sensing of Time-Varying Signals via Approximate Message Passsing," *IEEE Trans. Signal Process.*, vol. 61, no. 31, pp. 5270-5284, 2013.

[12] D. Sejdinovic, C. Andrieu and R. Piechocki, "Bayesian sequential compressed sensing in sparse dynamical systems", *Forty-Eighth Annual Allerton Conference*, pp. 1730-1736, 2010.

[13] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211-244, 2001.

[14] T. Park and G. Casella, "The bayesian lasso", *J. Amer. Stat. Assoc.*, Vol. 103, pp. 681-686, 2008.

[15] N. J. Foti, J. Xu, D. Laird, and E. B. Fox, "Stochastic Variational Inference for Hidden Markov Models," *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2014.