Efficient Forward Filtering and Backward Smoothing

Behrad Soleimani, Proloy Das
Department of Electrical and Computer Engineering, University of Maryland, College Park
Emails:{Behrad, Proloy}@umd.edu

I. Introduction

Forward filtering and backward smoothing is widely used in many different areas where one obtains a series of indirect linear measurements of some dynamically evolving sources and the goal is to infer the distribution of the underlying sources given the observations, recursively. However, the conventional filtering procedure, Kalman Filtering, would be computationally expensive, especially for large scale problems. Here, we propose an efficient filtering procedure using the steady-state covariances in order to tackle the computational complexity issue. The proposed method can shrink run-time x fold, for a very high dimensional problems, i.e. with xx dimensional states, xx time points. A MATLAB implementation of the proposed method can be accessed on Github [1].

II. Preliminaries and System Model

We consider a linear system whose measurement produces N_y dimensional observation vector $\mathbf{y}_t \in \mathbb{R}^{N_y}$ for time-points $t = 1, \cdots, T$. The measurement procedure that maps the source activities to the observations is considered as follows:

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{n}_t, \quad t = 1, \cdots, T, \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{N_y \times N_x}$ is the linear mapping (aka the sensing matrix), $\mathbf{x}_t \in \mathbb{R}^{N_x}$ is the source vector at time t, and $\mathbf{n}_t \in \mathbb{R}^{N_x}$ represents the observation noise at time t. The measurement noise is considered as a zero-mean Gaussian vector with covariance matrix $\mathbf{R}$, i.e., $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ and i.i.d. across the time.
The source dynamics is modeled as a vector auto-regressive process with p lags, VAR(p), as follows

$$\mathbf{x}_t = \sum_{k=1}^{p} \mathbf{A}_k \mathbf{x}_{t-k} + \mathbf{B}\mathbf{e}_t + \mathbf{w}_t, \quad t = 1, 2, \cdots, T, \tag{2}$$

where $\{\mathbf{A}_k\}_{k=1}^{p} \in \mathbb{R}^{N_x \times N_x}$ represents the VAR coefficients, $\mathbf{B} \in \mathbb{R}^{N_x \times N_e}$ captures the contribution from external factors, $\mathbf{e}_t \in \mathbb{R}^{N_e}$ is the process corresponding to the external factor, and $\mathbf{w}_t \in \mathbb{R}^{N_x}$ is the source (state) noise vector which is also assumed to be a zero-mean Gaussian vector with covariance matrix $\mathbf{Q}$, i.e., $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and i.i.d. across the time.
Given the observation and dynamical model in (1) and (2), the goal is to find non-causal beliefs $p(\mathbf{x}_t|\mathbf{y}_{1:T})$ for all $t = 1, 2, \cdots, T$. A conventional way to do that is utilizing standard Kalman filtering and backward smoothing, however, it might be computationally expensive for large scale systems. In fact, at every iteration a matrix inversion is needed which has cubic order of complexity with respect to the source and observation dimensions. It the next section, we describe an efficient instance of forward filtering and backward smoothing to tackle this issue.

III. Conventional Forward Filtering and Backward Smoothing

In order to use the conventional results on Kalman filtering and backward smoothing in [2], we need to construct an augmented model as follows

$$\mathbf{y}_t = \widetilde{\mathbf{C}}\widetilde{\mathbf{x}}_t + \mathbf{n}_t, \qquad t = p+1, p+2, \cdots, T, \tag{3}$$

$$\widetilde{\mathbf{x}}_t = \widetilde{\mathbf{A}}\widetilde{\mathbf{x}}_{t-1} + \widetilde{\mathbf{B}}\mathbf{e}_t + \widetilde{\mathbf{w}}_t, \quad t = p+1, p+2, \cdots, T, \tag{4}$$

where $\widetilde{\mathbf{x}}_t = [\mathbf{x}_t^\top, \mathbf{x}_{t-1}^\top, \cdots, \mathbf{x}_{t-p+1}^\top]^\top \in \mathbb{R}^{p \times N_x}$ is the augmented source vector, $\widetilde{\mathbf{w}}_t \in \mathbb{R}^{p \times N_x}$ represents the augmented noise vector with covariance matrix $\widetilde{\mathbf{Q}}$

$$\widetilde{\mathbf{Q}} := \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\epsilon} \end{bmatrix} \in \mathbb{R}^{pN_x \times pN_x}, \quad \boldsymbol{\epsilon} = \epsilon \mathbf{I}_{(p-1)N_x} \quad (\epsilon \approx 0) \tag{5}$$

Algorithm 1 Conventional forward filtering and backward smoothing

Input: $\{\mathbf{y}_t\}_{t=1}^T, \{\mathbf{e}_t\}_{t=1}^T, \widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{Q}}, \widetilde{\mathbf{C}}, \mathbf{R}, \mathbf{m}, \mathbf{V}$.

Output: $\overline{\mathbf{x}}_{t_1|T}, \mathbf{P}_{t_1,t_2|T}, \ \forall t_1, t_2 = 1, 2, \cdots, T$.

1: Set the initial condition $\widetilde{\mathbf{x}}_{p|p} = \mathbf{m}, \ \mathbf{\Sigma}_{p|p} = \mathbf{V}$.

2: Forward filter for $t = p, p+1, \cdots, T-1$:

$\star$ Prediction step : $\widetilde{\mathbf{x}}_{t+1|t} = \widetilde{\mathbf{A}}\widetilde{\mathbf{x}}_{t|t} + \widetilde{\mathbf{B}}\mathbf{e}_{t+1}$.

$\mathbf{\Sigma}_{t+1|t} = \widetilde{\mathbf{A}}\mathbf{\Sigma}_{t|t}\widetilde{\mathbf{A}}^\top + \widetilde{\mathbf{Q}}$.

Kalman gain: $\mathbf{K}_{t+1} = \mathbf{\Sigma}_{t+1|t}\widetilde{\mathbf{C}}^\top(\widetilde{\mathbf{C}}\mathbf{\Sigma}_{t+1|t}\widetilde{\mathbf{C}}^\top + \mathbf{R})^{-1}$.

$\star$ Update step: $\widetilde{\mathbf{x}}_{t+1|t+1} = \widetilde{\mathbf{x}}_{t+1|t} + \mathbf{K}_{t+1}(\mathbf{y}_{t+1} - \widetilde{\mathbf{C}}\widetilde{\mathbf{x}}_{t+1|t})$.

$\mathbf{\Sigma}_{t+1|t+1} = \mathbf{\Sigma}_{t+1|t} - \mathbf{K}_{t+1}(\widetilde{\mathbf{C}}\mathbf{\Sigma}_{t+1|t}\widetilde{\mathbf{C}}^\top + \mathbf{R})\mathbf{K}_{t+1}^\top$.

3: Backward smoothing for $t = T-1, T-2, \cdots, p$:

Smoothing gain: $\mathbf{S}_t = \mathbf{\Sigma}_{t|t}\widetilde{\mathbf{A}}^\top\mathbf{\Sigma}_{t+1|t}^{-1}$.

$\star$ Update step: $\widetilde{\mathbf{x}}_{t|T} = \widetilde{\mathbf{x}}_{t|t} + \mathbf{S}_t(\widetilde{\mathbf{x}}_{t+1|T} - \widetilde{\mathbf{x}}_{t+1|t})$.

$\mathbf{\Sigma}_{t|T} = \mathbf{\Sigma}_{t|t} + \mathbf{S}_t(\mathbf{\Sigma}_{t+1|T} - \mathbf{\Sigma}_{t+1|t})\mathbf{S}_t^\top$.

4: Covariance smoothing for $t_1, t_2 = T, T-1, \cdots, p+1, p$:

$$\widetilde{\mathbf{P}}_{t_1,t_2|T} = \begin{cases} \mathbf{\Sigma}_{t_1|T} & \text{if } t_1 = t_2 \\ \widetilde{\mathbf{P}}_{t_2,t_1|T}^\top & \text{if } t_1 > t_2 \\ \mathbf{S}_{t_1}\widetilde{\mathbf{P}}_{t_1+1,t_2|T} & \text{if } t_1 < t_2 \end{cases}$$

5: Extract the original model moments from the augmented one:

$\overline{\mathbf{x}}_{t_1|T} = \left[\widetilde{\mathbf{x}}_{t_1|T}\right]_{(p-1)N_x:pN_x}, \ \forall t_1 = 1, \cdots, T,$

$\mathbf{P}_{t_1,t_2|T} = \left[\widetilde{\mathbf{P}}_{t_1,t_2|T}\right]_{1:N_x,(p-1)Nx+1:pN_x}, \ \forall t_1, t_2 = 1, \cdots, T.$

and the augmented coefficient matrices are defined as

$$\widetilde{\mathbf{A}} := \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_{p-1} & \mathbf{A}_p \\ \mathbf{I}_{N_x} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N_x} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{N_x} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{pN_x \times pN_x}, \quad \widetilde{\mathbf{B}} := \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{pN_x \times N_e}. \tag{6}$$

Using the augmented model, one can utilize the Kalman filtering and backward smoothing. The procedure is written in Algorithm 1. Given the initial conditions $p(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p) \sim \mathcal{N}(\mathbf{m}, \mathbf{V})$, the forward filtering and backward smoothing procedure (including the prediction and update steps) can be done recursively for all time instances. Moreover, to obtain the smoothed cross-covariance terms $\mathbf{\Sigma}_{t,t-1|T} = \text{Cov}[\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:T}]$, we utilize the results in [3]. At the end, we need to extract the parameters of the original model for augmented model. The outputs of the Algorithm 1 are

$$\overline{\mathbf{x}}_{t_1|T} = \mathbb{E}[\mathbf{x}_t|\mathbf{y}_{1:T}], \quad \mathbf{P}_{t_1,t_2|T} = \text{Cov}[\mathbf{x}_{t_1}, \mathbf{x}_{t_2}|\mathbf{y}_{1:T}], \quad \forall t_1, t_2 = 1, \cdots, T. \tag{7}$$

The bottleneck of the Algorithm is the matrix inversion. For each time instance t, the Kalman and smoothing gains have matrix inversion with $\mathcal{O}\left(N_y^3\right)$ and $\mathcal{O}\left((pN_x)^3\right)$ computational complexity, respectively. So, the overall complexity corresponding to matrix inversion is $\mathcal{O}\left(T(N_y^3 + (pN_x)^3)\right)$.

IV. Efficient Forward Filtering and Backward Smoothing

Due to computational complexity of the inverision, using Algorithm 1 might have some issues, specially for large scale problems. One of the ideas that previously used in [4], is to find the steady-state covariance matrices. We define the following steady-state covariances

$$\lim_{t\to\infty} \mathbf{\Sigma}_{t+1|t} = \mathbf{\Sigma}^{(+)}, \quad \lim_{t\to\infty} \mathbf{\Sigma}_{t|t} = \mathbf{\Sigma}^{(-)}. \tag{8}$$

Algorithm 2 Efficient forward filtering and backward smoothing

Input: $\{\mathbf{y}_t\}_{t=1}^T, \{\mathbf{e}_t\}_{t=1}^T, \widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{Q}}, \widetilde{\mathbf{C}}, \mathbf{R}, \mathbf{m}, \mathbf{V}$.

Output: $\overline{\mathbf{x}}_{t_1|T}, \mathbf{P}_{t_1,t_2|T},\ \forall t_1, t_2 = 1, 2 \cdots, T$.

1: Set the initial condition $\widetilde{\mathbf{x}}_{p|p} = \mathbf{m}$.

2: Find the steady-state covariance matrices $\mathbf{\Sigma}^{(+)}$ and $\mathbf{\Sigma}^{(-)}$ via solving DARE with respect to $\mathbf{\Sigma}^{(-)}$:

$$\begin{cases}\mathbf{\Sigma}^{(+)} = \widetilde{\mathbf{A}}\mathbf{\Sigma}^{(-)}\widetilde{\mathbf{A}}^\top + \widetilde{\mathbf{Q}}, \\ \mathbf{\Sigma}^{(-)} = \mathbf{\Sigma}^{(+)} - \mathbf{\Sigma}^{(+)}\widetilde{\mathbf{C}}^\top(\widetilde{\mathbf{C}}\mathbf{\Sigma}^{(+)}\widetilde{\mathbf{C}}^\top + \mathbf{R})^{-1}\widetilde{\mathbf{C}}\mathbf{\Sigma}^{(+)}.\end{cases}$$

3: Compute Kalman and Smoothing gains:

$$\begin{cases}\mathbf{K} = \mathbf{\Sigma}^{(+)}\widetilde{\mathbf{C}}^\top(\widetilde{\mathbf{C}}\mathbf{\Sigma}^{(+)}\widetilde{\mathbf{C}}^\top + \mathbf{R})^{-1}, \\ \mathbf{S} = \mathbf{\Sigma}^{(-)}\widetilde{\mathbf{A}}^\top {\mathbf{\Sigma}^{(+)}}^{-1}\end{cases}$$

4: Forward filter for $t = p, p+1, \cdots, T-1$:

⋆ Prediction step : $\widetilde{\mathbf{x}}_{t+1|t} = \widetilde{\mathbf{A}}\widetilde{\mathbf{x}}_{t|t} + \widetilde{\mathbf{B}}\mathbf{e}_{t+1}$.

⋆ Update step: $\widetilde{\mathbf{x}}_{t+1|t+1} = \widetilde{\mathbf{x}}_{t+1|t} + \mathbf{K}(\mathbf{y}_{t+1} - \widetilde{\mathbf{C}}\widetilde{\mathbf{x}}_{t+1|t})$.

5: Backward smoothing for $t = T-1, T-2, \cdots, p$:

⋆ Update step: $\widetilde{\mathbf{x}}_{t|T} = \widetilde{\mathbf{x}}_{t|t} + \mathbf{S}(\widetilde{\mathbf{x}}_{t+1|T} - \widetilde{\mathbf{x}}_{t+1|t})$.
$\mathbf{\Sigma}_{t|T} = \mathbf{\Sigma}^{(-)} + \mathbf{S}(\mathbf{\Sigma}_{t+1|T} - \mathbf{\Sigma}^{(+)})\mathbf{S}^\top$.

6: Covariance smoothing for $t_1, t_2 = T, T-1, \cdots, p+1, p$:

$$\widetilde{\mathbf{P}}_{t_1,t_2|T} = \begin{cases}\mathbf{S}^{t_2-t_1}\mathbf{\Sigma}_{t_1|T}, & \text{if } t_1 \le t_2 \\ \widetilde{\mathbf{P}}^\top_{t_2,t_1|T}, & \text{otherwise}\end{cases}$$

7: Extract the original model moments from the augmented one:

$\overline{\mathbf{x}}_{t_1|T} = \left[\widetilde{\mathbf{x}}_{t_1|T}\right]_{(p-1)N_x:pN_x},\ \forall t_1 = 1, \cdots, T,$

$\mathbf{P}_{t_1,t_2|T} = \left[\widetilde{\mathbf{P}}_{t_1,t_2|T}\right]_{1:N_x,(p-1)Nx+1:pN_x},\ \forall t_1, t_2 = 1, \cdots, T.$

In order to obtain the steady-state predicted state covariance, one can find the solution of the discrete algebraic Riccati equation (DARE) with respect to: $\mathbf{\Sigma}^{(-)}$

$$\mathbf{\Sigma}^{(+)} = \widetilde{\mathbf{A}}\mathbf{\Sigma}^{(-)}\widetilde{\mathbf{A}}^\top + \widetilde{\mathbf{Q}}, \tag{9}$$

$$\mathbf{\Sigma}^{(-)} = \mathbf{\Sigma}^{(+)} - \mathbf{\Sigma}^{(+)}\widetilde{\mathbf{C}}^\top(\widetilde{\mathbf{C}}\mathbf{\Sigma}^{(+)}\widetilde{\mathbf{C}}^\top + \mathbf{R})^{-1}\widetilde{\mathbf{C}}\mathbf{\Sigma}^{(+)}. \tag{10}$$

After finding $\mathbf{\Sigma}^{(+)}$ and $\mathbf{\Sigma}^{(-)}$, Kalman and smoothing gains can be computed ahead of the recursive updates as following:

$$\mathbf{K} = \mathbf{\Sigma}^{(+)}\widetilde{\mathbf{C}}^\top(\widetilde{\mathbf{C}}\mathbf{\Sigma}^{(+)}\widetilde{\mathbf{C}}^\top + \mathbf{R})^{-1}, \tag{11}$$

$$\mathbf{S} = \mathbf{\Sigma}^{(-)}\widetilde{\mathbf{A}}^\top {\mathbf{\Sigma}^{(+)}}^{-1} \tag{12}$$

Given the steady-state covariances and smoothing gain, smoothed covariance matrices $\mathbf{\Sigma}_{t|T}$ can be updated recursively as:

$$\mathbf{\Sigma}_{t|T} = \mathbf{\Sigma}^{(-)} + \mathbf{S}(\mathbf{\Sigma}_{t+1|T} - \mathbf{\Sigma}^{(+)})\mathbf{S}^\top \tag{13}$$

for $t = T-1, \cdots, p$ starting with $\mathbf{\Sigma}_{T|T} = \mathbf{\Sigma}^{(-)}$. This computationally cheap scheme is presented in Algorithm 2 as a fast and concise alternative of the computationally intensive forward filtering-backward smoothing algorithm.

In terms of computational complexity, the Algorithm 2 needs only two matrix inversion. Moreover, two discrete algebraic Riccati equations can be solved efficiently via MacFarlane-Potter-Fath eigen-structure method [5].

References

[1] B. Soleimani, "Efficientffbs," https://github.com/BehradSol/EfficientFFBS, 2020. I

[2] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012. III

[3] P. D. JONG and M. J. Mackinnon, "Covariances for smoothed estimates in state space models," *Biometrika*, vol. 75, no. 3, pp. 601–602, 1988. III

[4] E. Pirondini, B. Babadi, G. Obregon-Henao, C. Lamus, W. Q. Malik, M. S. Hämäläinen, and P. L. Purdon, "Computationally efficient algorithms for sparse, dynamic solutions to the eeg source localization problem," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 6, pp. 1359–1372, 2017. IV

[5] C. K. Chui, G. Chen *et al.*, *Kalman filtering*. Springer, 2017. IV