Description of PWC Model

Luca Iten

January 30, 2024

This document describes the implementations of class *PWCModel*.

1 Model Description

The model described by the class *PWCModel* can be used to infer the means of piecewise constant data that is observed through an Additional White Gaussian Noise (AWGN) channel. These observations have general dimension D. To emphasize a piecewise constant solution, a sparsifying prior is applied to the state jumps (i.e., changes of the means) at each index. Prior knowledge about the initial mean (i.e., the mean at index 1) can be incorporated through a prior $\rho(\cdot)$. A factor graph depicting this (very general) model is shown in Figure 1. The nomenclature used there is consistent with the derivations in this document as well as the actual implementations.

2 Explanation of Implementations

This section describes the implementations of class *PWCModel*. In particular, it explains how the mean estimation is performed. This should in no way be seen

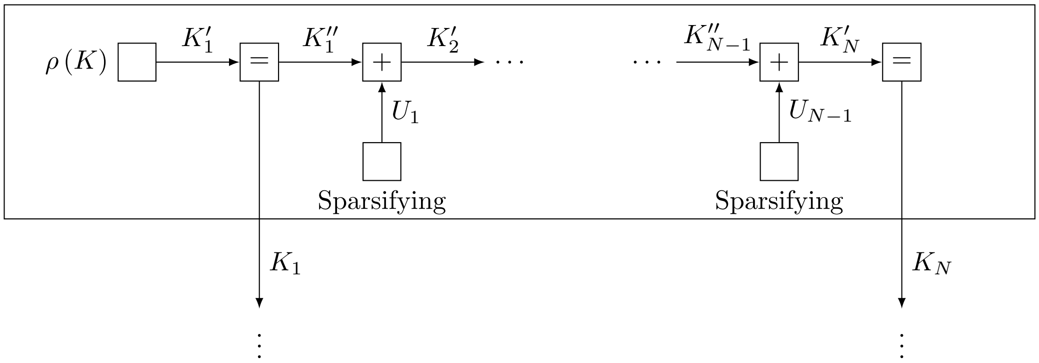

Figure 1: Factor graph of PWC model.

as a complete description of the algorithm, but as a starting point for someone trying to understand the implementations. In general, the used naming and directions of the messages are consistent in the factor graph shown in Figure 1, the following explanations, and the actual implementations.

2.1 Initializing an Object of *PWCModel*

When initializing an object of class *PWCModel*, the number of observations N, their dimension D, and the 'mode' $\in$ ['conventional', 'dual'] must be specified. The two different modes state whether either the mean and covariance matrix ('conventional') or the dual-mean and precision matrix ('dual') representation is used for message passing. One of these representations usually is the "more natural" choice, in the sense that all quantities are well defined. This choice heavily depends on the larger context the model is applied to!

Furthermore, the initial values of K_i and U_i (state vectors) can be specified. This can be done to incorporate prior knowledge about the model, for example, when some of the means are already known at some time indices. If no value is specified, the means of K_i and U_i are randomly initialized close to zero and their covariances are set high. representing (almost) no prior knowledge.

2.2 Sparsifying Prior

The sparsifying prior should strongly emphasize solutions of U_i that are close to zero while allowing occasional deviations. The natural choice for such a prior is the *log-cost* NUV prior as, for example, described in [1]. The resulting messages generated by such a NUV prior are

$$\overrightarrow{m}_{U_i} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{1}$$

$$\overrightarrow{V}_{U_i} = \sigma^2_{U_i} \cdot \mathbf{I}_M \,, \tag{2}$$

where

$$\sigma^2_{U_i} = \frac{\mathrm{Tr}\left\{\hat{V}_{S_i}\right\} + ||\hat{m}_{S_i}||^2}{\beta_u} \,. \tag{3}$$

Here, $\mathrm{Tr}\left\{\cdot\right\}$ denotes the trace and $||\cdot||$ the $L{-}2$ norm. Note that the Expectation Maximization (EM) rule has been used to derive the update in Equation (3). The parameter β_u can be used to "tune" the effect of this prior, where higher values correspond to a stronger prior. For the special case where β_u is selected to be equal to the dimension of the applied data, this prior is also referred to as the *plain* NUV. In the described implementations, this is usually the default value. For the prior to be sparsifying, its value must be positive!

2.3 Message Passing with MBF

If the estimator is initialized with 'mode' set to 'conventional', the given ingoing messages through K_i, $i \in \{1, \dots, N\}$ are assumed to be represented by their mean and covariance matrices. Therefore, forward- / backward- message passing is most efficiently handled by the Modified Bryson-Frazier (MBF) smoother (described in [2]). Accordingly, the forward messages are recursively computed as

$$\overrightarrow{m}_{K_i''} = \overrightarrow{m}_{K_i'} + \overrightarrow{V}_{K_i'} G_i \left(\overleftarrow{m}_{K_i} - \overrightarrow{m}_{K_i'} \right) \tag{4}$$

$$\overrightarrow{V}_{K_i''} = \overrightarrow{V}_{K_i'} - \overrightarrow{V}_{K_i'} G_i \overrightarrow{V}_{K_i'} \tag{5}$$

$$\overrightarrow{m}_{K_{i+1}'} = \overrightarrow{m}_{K_i''} \tag{6}$$

$$\overrightarrow{V}_{K_{i+1}'} = \overrightarrow{V}_{K_i''} + \overrightarrow{V}_{U_i} \, , \tag{7}$$

where

$$G_i = \left(\overleftarrow{V}_{K_i} + \overrightarrow{V}_{K_i'} \right)^{-1} \tag{8}$$

$$F_i = \mathbf{I}_D - \overrightarrow{V}_{K_i'} G_i \, . \tag{9}$$

$\mathbf{I}_D$ denotes the D-dimensional identity matrix. In the described implementations, $\overrightarrow{m}_{K_1'}$ and $\overrightarrow{V}_{K_1'}$ are initialized to incorporate any prior knowledge given to the estimator. If no prior knowledge is assumed, $\overrightarrow{V}_{K_1'}$ is initialized to a diagonal matrix with large (i.e., $\gg 0$) elements. Next, the backward recursion is computed as

$$\tilde{W}_{K_{i-1}'} = F_{i-1}^{\mathsf{T}} \tilde{W}_{K_i'} F_{i-1} + G_{i-1} \tag{10}$$

$$\tilde{\xi}_{K_{i-1}'} = F_{i-1}^{\mathsf{T}} \tilde{\xi}_{K_i'} - G_{i-1} \left(\overleftarrow{m}_{K_{i-1}} - \overrightarrow{m}_{K_{i-1}'} \right) \, , \tag{11}$$

where $\tilde{W}_{K_N'}$ and $\tilde{\xi}_{K_N'}$ are initialized as

$$\tilde{W}_{K_N'} = \left(\overrightarrow{V}_{K_N'} + \overleftarrow{V}_{K_N} \right)^{-1} \tag{12}$$

$$\tilde{\xi}_{K_N'} = \tilde{W}_{K_N'} \left(\overrightarrow{m}_{K_N'} - \overleftarrow{m}_{K_N} \right) \, . \tag{13}$$

Finally, the posterior estimates of K_i and U_i are calculated as

$$\hat{m}_{K_i} = \overrightarrow{m}_{K_i'} - \overrightarrow{V}_{K_i'} \tilde{\xi}_{K_i'} \tag{14}$$

$$\hat{V}_{K_i} = \overrightarrow{V}_{K_i'} - \overrightarrow{V}_{K_i'} \tilde{W}_{K_i'} \overrightarrow{V}_{K_i'} \tag{15}$$

$$\hat{m}_{U_i} = -\overrightarrow{V}_{U_i} \tilde{\xi}_{K_{i+1}'} \tag{16}$$

$$\hat{V}_{U_i} = \overrightarrow{V}_{U_i} - \overrightarrow{V}_{U_i} \tilde{W}_{K_{i+1}'} \overrightarrow{V}_{U_i} \, . \tag{17}$$

Note that the offset of indices in (16) and (17) are due to the definition of the indices in Figure 1.

2.4 Message Passing with BIFM

This approach is in some sense 'dual' to the algorithm described in the previous Subsection. It assumes the ingoing messages through K_i are given by their dual-mean and precision matrix representations. In the implementations of *PWCModel*, this method is used when the estimator is initialized with 'mode' = 'dual'. The following expressions are derived from the Backward Information Filter, forward with Marginals (BIFM) as described in [2]. The backward messages are recursively calculated as

$$\overleftarrow{\xi}_{K'_i} = \overleftarrow{\xi}_{K''_i} + \overleftarrow{\xi}_{K_i} \tag{18}$$

$$\overleftarrow{W}_{K'_i} = \overleftarrow{W}_{K''_i} + \overleftarrow{W}_{K_i} \tag{19}$$

$$\overleftarrow{\xi}_{K''_{i-1}} = \overleftarrow{\xi}_{K'_i} - \overleftarrow{W}_{K'_i}\ddot{h}_i \tag{20}$$

$$\overleftarrow{W}_{K''_{i-1}} = \overleftarrow{W}_{K'_i} - \overleftarrow{W}_{K'_i}\ddot{H}_i\overleftarrow{W}_{K'_i}\,, \tag{21}$$

where

$$\ddot{H}_i = \left(\overrightarrow{W}_{U_{i-1}} + \overleftarrow{W}_{K'_i}\right)^{-1} \tag{22}$$

$$\ddot{h}_i = \ddot{H}_i\overleftarrow{\xi}_{K'_i}\,. \tag{23}$$

$\overleftarrow{\xi}_{K''_N}$ and $\overleftarrow{W}_{K''_N}$ are both initialized to all-zero vectors / matrices. Next, the forward messages are recursively derived by

$$\tilde{F}_i = \mathbf{I}_D - \overleftarrow{W}_{K'_i}\ddot{H}_i \tag{24}$$

$$\hat{m}_{K_i} = \tilde{F}_i^{\mathsf{T}}\hat{m}_{K_{i-1}} + \ddot{h}_i \tag{25}$$

$$\hat{V}_{K_i} = \tilde{F}_i^{\mathsf{T}}\hat{V}_{K_{i-1}}\tilde{F}_i + \ddot{H}_i\,. \tag{26}$$

To incorporate given prior knowledge (represented by prior $\rho(K)$ in Figure 1), $\hat{m}_{K_1}$ and $\hat{V}_{K_1}$ are initialized as

$$\hat{V}_{K_1} = \left(\overrightarrow{W}_{K'_1} + \overleftarrow{W}_{K'_1}\right)^{-1} \tag{27}$$

$$\hat{m}_{K_1} = \hat{V}_{K_1}\left(\overrightarrow{\xi}_{K'_1} + \overleftarrow{\xi}_{K'_1}\right)\,, \tag{28}$$

where $\overrightarrow{\xi}_{K'_1}$ and $\overrightarrow{W}_{K'_1}$ directly follow from the values specifying $\rho(K)$. If no prior knowledge is assumed, both are set to all-zero vectors / matrices. In parallel, the following messages are also computed

$$\tilde{\xi}_{U_{i-1}} = \tilde{\xi}_{K'_i} \tag{29}$$

$$= \overleftarrow{W}_{K'_i}\hat{m}_{K_i} - \overleftarrow{\xi}_{K'_i} \tag{30}$$

$$\tilde{W}_{U_{i-1}} = \overrightarrow{\tilde{W}}_{K'_i} \tag{31}$$

$$= \overleftarrow{W}_{K'_i} - \overleftarrow{W}_{K'_i}\hat{V}_{K_i}\overleftarrow{W}_{K'_i}\,. \tag{32}$$

Note that the expressions given in Equations (25) and (26) already are the posterior estimates of K_i. Finally, the posterior estimates of the sparse inputs U_i are calculated as

$$\hat{m}_{U_i} = -\overrightarrow{V}_{U_i}\tilde{\xi}_{U_i} \tag{33}$$

$$\hat{V}_{U_i} = \overrightarrow{V}_{U_i} - \overrightarrow{V}_{U_i}\tilde{W}_{U_i}\overrightarrow{V}_{U_i}\,. \tag{34}$$

2.5 Estimation with IRLS

Subsection 2.2 described how the messages generated by the sparsifying prior are calculated in the implementations of class *PWCModel*. Subsections 2.3 and 2.4 then described how the corresponding posterior estimates of K_i and U_i are calculated by either MBF or BIFM, respectively. But how are the final estimates found?

For this, the algorithm relies on the idea of Iteratively Reweighed Least-Squared (IRLS) as described in [1]. Thereby, the generated messages by the sparsifying prior and the posterior estimates of K_i and U_i are iteratively improved. In the implementation of *PWCModel* this is done in a for-loop. The maximum number of iterations is set when calling the corresponding function. To check for convergence, the relative change of the absolute values of the posterior estimates from the current to the previous iteration are compared to a threshold. If this relative change falls below this specified threshold, the algorithm terminates.

References

[1] H.-A. Loeliger, "Lecture notes for model-based estimation and signal analysis," 2023.

[2] H. M. F. W. N. Z. Hans-Andreas Loeliger, Lukas Bruderer, "On sparsity by nuv-em, gaussian message passing, and kalman smoothing," 2016.