CS5340: Uncertainty Modeling in AI

Tutorial 7

Released: Mar. 9, 2023

Problem 1. (Linear Dynamical System)

In this tutorial, we will examine the popular *linear dynamical system*. This model has many real-world applications ranging from tracking to AI planning and decision-making. Specifically, we will look at linear-Gaussian state space models. The model is very similar to the hidden Markov model except that it has Gaussian latent variables. Some of you may have been introduced to this model when learning about the Kalman filter.

In this model, we have latent variables $\mathbf{z}_t$ and observed variables $\mathbf{x}_t$. In this model, we have initial latent variable

$$\mathbf{z}_1 = \boldsymbol{\mu}_0 + \mathbf{u} \tag{1}$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{V}_0)$. The system evolves via noisy linear equations:

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t \tag{2}$$
$$\mathbf{x}_t = \mathbf{C}\mathbf{z}_t + \mathbf{v}_t \tag{3}$$

where the noise terms are Gaussian,

$$\mathbf{w}_t \sim \mathcal{N}(\mathbf{w}_t|\mathbf{0}, \boldsymbol{\Gamma}) \tag{4}$$
$$\mathbf{v}_t \sim \mathcal{N}(\mathbf{v}_t|\mathbf{0}, \boldsymbol{\Sigma}) \tag{5}$$

The parameters of this model are

$$\boldsymbol{\theta} = \{\mathbf{A}, \boldsymbol{\Gamma}, \mathbf{C}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_0, \mathbf{V}_0\}$$

which we wish to learn via MLE. Similar to the HMM, we will have to do inference due to the latent variables, which results in an EM algorithm. For the remainder pf this tutorial, we index time spans using subscripts, e.g., $\mathbf{z}_{1:T} = \{\mathbf{z}_t, \mathbf{z}_2, \ldots, \mathbf{z}_T\}$ and likewise for $\mathbf{x}_{1:T}$.

Problem 1.a. Given the description above, draw the DGM corresponding to the linear-Gaussian state-space model.

Problem 1.b. Are the random variables $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ jointly Gaussian? In other words, is the concatenation of the all the random vectors drawn from a multivariate Gaussian? Why or why not?

Problem 1.c. Let us now consider how we might perform inference in this model. Specifically, how do you compute $p(\mathbf{z}_t|\mathbf{x}_{1:T}, \boldsymbol{\theta})$? State qualitatively how you might solve this and write down the major steps. *Hint:* Recall how this inference was done for HMMs. We will walk through the actual equations in the following subproblem.

Problem 1.d. Argue that in the model above, the forward messages that we propagate must be Gaussian, i.e.,

$$\widehat{\alpha}(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \mathbf{V}_t) \tag{6}$$

Hint: Recall that the scaled forward message is given by

$$c_t\widehat{\alpha}(\mathbf{z}_t) = p(\mathbf{x}_t|\mathbf{z}_t)\int \widehat{\alpha}(\mathbf{z}_{t-1})p(\mathbf{z}_t|\mathbf{z}_{t-1})d\mathbf{z}_{t-1} \tag{7}$$

For this tutorial, we will not be deriving everything by hand (which is tedious and you probably had enough of by now[1]). Instead, we will give the mean and covariance of the forward message:

$$\boldsymbol{\mu}_t = \mathbf{A}\boldsymbol{\mu}_{t-1} + \mathbf{K}_t(\mathbf{x}_t - \mathbf{C}\mathbf{A}\boldsymbol{\mu}_{t-1}) \tag{8}$$
$$\mathbf{V}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathbf{P}_{t-1} \tag{9}$$
$$c_t = \mathcal{N}(\mathbf{x}_t|\mathbf{C}\mathbf{A}\boldsymbol{\mu}_{t-1}, \mathbf{C}\mathbf{P}_{t-1}\mathbf{C}^\top + \boldsymbol{\Sigma}) \tag{10}$$

where

$$\mathbf{P}_{t-1} = \mathbf{A}\mathbf{V}_{t-1}\mathbf{A}^\top + \boldsymbol{\Gamma} \tag{11}$$
$$\mathbf{K}_t = \mathbf{P}_{t-1}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{t-1}\mathbf{C}^\top + \boldsymbol{\Sigma})^{-1} \tag{12}$$

Problem 1.e. The matrix $\mathbf{K}_t$ above is the famous *Kalman gain matrix*. Can you give an intuitive explanation of the update equation for $\boldsymbol{\mu}_t$ in Eq. (8) above? What is role of the Kalman gain matrix?

Problem 1.f. In the linear dynamical systems literature, the backward recursion is usually formulated in terms of

$$\gamma(\mathbf{z}_t) = \widehat{\alpha}(\mathbf{z}_t)\widehat{\beta}(\mathbf{z}_t)$$

where

$$c_{t+1}\widehat{\beta}(\mathbf{z}_t) = \int \widehat{\beta}(\mathbf{z}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{z}_{t+1})p(\mathbf{z}_{t+1}|\mathbf{z}_t)d\mathbf{z}_{t+1}.$$

Provide an argument that the message $\gamma(\mathbf{z}_t)$ must be Gaussian, i.e.,

$$\gamma(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t|\widehat{\boldsymbol{\mu}}_t, \widehat{\mathbf{V}}_t)$$

[1] but if you are bored on a Saturday evening, it is worth giving it a go at least once.

As before, we will give the mean and covariance of the backward messages without proof.

$$\widehat{\boldsymbol{\mu}}_t = \boldsymbol{\mu}_t + \mathbf{J}_t(\widehat{\boldsymbol{\mu}}_{t+1} - \mathbf{A}\boldsymbol{\mu}_T) \tag{13}$$

$$\widehat{\mathbf{V}}_t = \mathbf{V}_t + \mathbf{J}_t(\widehat{\mathbf{V}}_{t+1} - \mathbf{P}_t)\mathbf{J}_t^\top \tag{14}$$

where

$$\mathbf{J}_t = \mathbf{V}_t\mathbf{A}^\top\mathbf{P}_t^{-1}$$

Problem 1.g. Let us now consider learning the parameters $\boldsymbol{\theta}$. We will use the EM algorithm. State the two main steps of the EM algorithm.

Problem 1.h. For the EM algorithm we will need certain expectations, specifically, $\mathbb{E}[\mathbf{z}_t]$, $\mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^\top]$, and $\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top]$. We'll give you the latter two, i.e.,

$$\mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^\top] = \mathbf{J}_{t-1}\widehat{\mathbf{V}}_t + \widehat{\boldsymbol{\mu}}_t\widehat{\boldsymbol{\mu}}_{t-1}^\top \tag{15}$$

$$\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top] = \widehat{\mathbf{V}}_t + \widehat{\boldsymbol{\mu}}_t\widehat{\boldsymbol{\mu}}_t^\top \tag{16}$$

What is $\mathbb{E}[\mathbf{z}_t]$?

Problem 1.i. Show that the complete data likelihood $\log p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}|\boldsymbol{\theta})$ is given by:

$$\log p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}|\boldsymbol{\theta}) = \log p(\mathbf{z}_1|\boldsymbol{\mu}_0, \mathbf{V}_0) + \sum_{t=2}^{T} \log p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}, \boldsymbol{\Gamma}) + \sum_{t=1}^{T} \log p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{C}, \boldsymbol{\Sigma}) \tag{17}$$

Problem 1.j. Next, we need to find the Q function. What should be filled into the '?' below?

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{\text{old}}) = \mathbb{E}_{\mathbf{z}_{1:T}|\boldsymbol{\theta}_{\text{old}}}[\log p(?)]$$

Following our trend of not having to do tedious derivations, let's give you the update equations as well.

$$\boldsymbol{\mu}_0^{\text{new}} = \mathbb{E}[\mathbf{z}_1] \tag{18}$$

$$\mathbf{V}_0^{\text{new}} = \mathbb{E}[\mathbf{z}_1\mathbf{z}_1^\top] - \mathbb{E}[\mathbf{z}_1]\mathbb{E}[\mathbf{z}_1^\top] \tag{19}$$

$$\mathbf{A}^{\text{new}} = \left(\sum_{t=2}^{T} \mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^\top]\right)\left(\sum_{t=2}^{T} \mathbb{E}[\mathbf{z}_{t-1}\mathbf{z}_{t-1}^\top]\right)^{-1} \tag{20}$$

$$\boldsymbol{\Gamma}^{\text{new}} = \frac{1}{T-1}\sum_{t=2}^{T}\left(\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top] - \mathbf{A}^{\text{new}}\mathbb{E}[\mathbf{z}_{t-1}\mathbf{z}_t^\top] - \mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^\top](\mathbf{A}^{\text{new}})^\top + \mathbf{A}^{\text{new}}\mathbb{E}[\mathbf{z}_{t-1}\mathbf{z}_{t-1}^\top](\mathbf{A}^{\text{new}})^\top\right) \tag{21}$$

$$\mathbf{C}^{\text{new}} = \left(\sum_{t=1}^{T} \mathbf{x}_t\mathbb{E}[\mathbf{z}_t^\top]\right)\left(\sum_{t=1}^{T} \mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top]\right)^{-1} \tag{22}$$

$$\boldsymbol{\Sigma}^{\text{new}} = \frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{x}_t\mathbf{x}_t^\top - \mathbf{C}^{\text{new}}\mathbb{E}[\mathbf{z}_t]\mathbf{x}_t^\top - \mathbf{x}_t\mathbb{E}[\mathbf{z}_t^\top](\mathbf{C}^{\text{new}})^\top + \mathbf{C}^{\text{new}}\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top](\mathbf{C}^{\text{new}})^\top\right) \tag{23}$$

Problem 1.k. Try implementing the above update equations to perform EM for a linear dynamical system (similar to the PPCA that we have done). Try it first on some synthetic data. We will post some data on piazza next week. When does the method work well and when does it fail?

Problem 1.l. Why is there no need to consider a different algorithm for finding the maximum a posteriori configuration (like the Viterbi algorithm) for the linear dynamical system?