CS5340: Uncertainty Modeling in AI

Tutorial 7: Solutions

Released: Mar. 13, 2024

Problem 1. (Linear Dynamical System)

In this tutorial, we will examine the popular *linear dynamical system* (LDS). 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. If you want to read more about this model, consult Bishop's Pattern Recognition and Machine Learning, Chapter 13.3.

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

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

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., $\mathbf{z}_{1:T} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\}$ 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.

Solution: As shown in Figure 1.

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, does the random variable $\mathbf{y} = [\mathbf{z}_1; \mathbf{z}_2; \ldots, \mathbf{z}_T; \mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_T]$ (we concatenate the vectors column-wise) follow a multivariate Gaussian distribution? Why or why not?

Solution: The joint distribution is Gaussian.

Consider the PGM factorization,

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}|\theta) = p(\mathbf{z}_1)\left[\prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{z}_{t-1})\right]\prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_t) \tag{6}$$

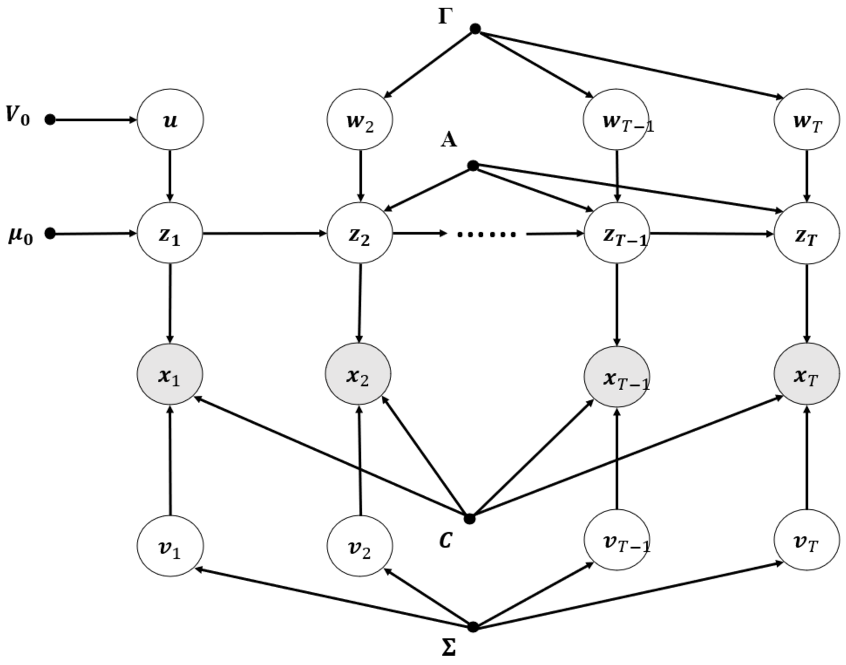

Figure 1: DGM of linear-Gaussian model

According to our definition, $\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t, \mathbf{x}_t = \mathbf{C}\mathbf{z}_t + \mathbf{v}_t, \mathbf{z}_1 = \boldsymbol{\mu}_0 + \mathbf{u}$, where $\mathbf{w}_t, \mathbf{v}_t, \boldsymbol{\mu}$ are Gaussian. Therefore, each node is a linear transformation of its parents. Since the root nodes $\boldsymbol{\mu}, \mathbf{z}_1, \mathbf{v}_1$ are gaussian, given the properties of linear Gaussian model, the joint distribution is Gaussian.

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.

Solution: Compute $p(\mathbf{z}_t|\mathbf{x}_{1:T}, \boldsymbol{\theta})$ for all $t \in [1:T]$ with alpha-beta algorithm.
First, factorize $p(\mathbf{z}_t|\mathbf{x}_{1:T}, \boldsymbol{\theta})$ according to forward messages and backward messages.

$$p(\mathbf{z}_t|\mathbf{x}_{1:T}) = \frac{p(\mathbf{x}_{1:T}|\mathbf{z}_t)p(\mathbf{z}_t)}{p(\mathbf{x}_{1:T})} \tag{7}$$

$$= \frac{p(\mathbf{x}_1, \cdots, \mathbf{x}_t, \mathbf{z}_t)p(\mathbf{x}_{t+1}, \cdots, \mathbf{x}_T|\mathbf{z}_t)}{p(\mathbf{x}_{1:T})} \tag{8}$$

$$= \frac{\alpha(\mathbf{z}_t)\beta(\mathbf{z}_t)}{p(\mathbf{x}_{1:T})} \tag{9}$$

Then, $\alpha(\mathbf{z}_t), \beta(\mathbf{z}_t)$ can be computed recursively (refer to lecture slides for proof):

$$\alpha(\mathbf{z}_t) = p(\mathbf{x}_t|\mathbf{z}_t) \int \alpha(\mathbf{z}_{t-1})p(\mathbf{z}_t|\mathbf{z}_{t-1})d\mathbf{z}_{t-1} \tag{10}$$

$$\beta(\mathbf{z}_t) = \int \beta(\mathbf{z}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{z}_{t+1})p(\mathbf{z}_{t+1}|\mathbf{z}_t)d\mathbf{z}_{t+1} \tag{11}$$

Now, we are able to compute $p(\mathbf{z}_t|\mathbf{x}_{1:T}, \boldsymbol{\theta})$ by computing $\alpha(\mathbf{z}_t)$ and $\beta(\mathbf{z}_t)$ recursively.

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

$$\widehat{\alpha}(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \mathbf{V}_t) \tag{12}$$

Hint: Recall that the scaled forward message is given by

$$c_t\widehat{\alpha}(\mathbf{z}_t) = p(\mathbf{x}_t|\mathbf{z}_t)\int \widehat{\alpha}(\mathbf{z}_{t-1})p(\mathbf{z}_t|\mathbf{z}_{t-1})d\mathbf{z}_{t-1} \tag{13}$$

Solution: Recall that $\widehat{\alpha}(\mathbf{z}_t) = p(\mathbf{z}_t|\mathbf{x}_1, \cdots, \mathbf{x}_t)$. Since $p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})$ is a Gaussian (as shown in Problem 1.b), the conditional probability $p(\mathbf{z}_t|\mathbf{x}_{1:t})$ is a Gaussian. As a result, the forward message $\widehat{\alpha}(\mathbf{z}_t)$ is a Gaussian.

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:

$$\boldsymbol{\mu}_t = \mathbf{A}\boldsymbol{\mu}_{t-1} + \mathbf{K}_t(\mathbf{x}_t - \mathbf{C}\mathbf{A}\boldsymbol{\mu}_{t-1}) \tag{14}$$

$$\mathbf{V}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathbf{P}_{t-1} \tag{15}$$

$$c_t = \mathcal{N}(\mathbf{x}_t|\mathbf{C}\mathbf{A}\boldsymbol{\mu}_{t-1}, \mathbf{C}\mathbf{P}_{t-1}\mathbf{C}^\top + \boldsymbol{\Sigma}) \tag{16}$$

where

$$\mathbf{P}_{t-1} = \mathbf{A}\mathbf{V}_{t-1}\mathbf{A}^\top + \boldsymbol{\Gamma} \tag{17}$$

$$\mathbf{K}_t = \mathbf{P}_{t-1}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{t-1}\mathbf{C}^\top + \boldsymbol{\Sigma})^{-1} \tag{18}$$

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. (14) above? What is role of the Kalman gain matrix?

Solution: [From Bishop's PRML Chp 13.3] We can interpret the steps involved in going from the posterior marginal over $\mathbf{z}_{t-1}$ to the posterior marginal over $\mathbf{z}_t$ as follows. In Eq. (14), we can view the quantity $\mathbf{A}\boldsymbol{\mu}_{t-1}$ as the prediction of the mean over $\mathbf{z}_t$ obtained by simply taking the mean over $\mathbf{z}_{t-1}$ and projecting it forward one step using transition probability matrix $\mathbf{A}$. This predicted mean would give a predicted observation for $\mathbf{x}_t$ given by $\mathbf{C}\mathbf{A}\mathbf{z}_{t-1}$ obtained by applying the emission probability matrix $\mathbf{C}$ to the predicted hidden state mean. We can view Eq. (14) for the mean of the hidden variable distribution as taking the predicted mean $\mathbf{A}\boldsymbol{\mu}_{t-1}$ and then adding a correction that is proportional to the error $\mathbf{x}_n - \mathbf{C}\mathbf{A}\mathbf{z}_{t-1}$ between the predicted observation and the actual observation. The coefficient of this correction is given by the Kalman gain matrix. Thus, we can view the Kalman filter as a process of making successive predictions and then correcting these predictions in the light of the new observations.

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

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.,

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

Solution: Since $\gamma(\mathbf{z}_t) = p(\mathbf{z}_t|\mathbf{x}_{1:T})$ and the LDS is a linear Gaussian model, $\gamma(\mathbf{z}_t)$ is Gaussian.

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

$$\widehat{\boldsymbol{\mu}}_t = \boldsymbol{\mu}_t + \mathbf{J}_t(\widehat{\boldsymbol{\mu}}_{t+1} - \mathbf{A}\boldsymbol{\mu}_T) \tag{19}$$

$$\widehat{\mathbf{V}}_t = \mathbf{V}_t + \mathbf{J}_t(\widehat{\mathbf{V}}_{t+1} - \mathbf{P}_t)\mathbf{J}_t^\top \tag{20}$$

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.

Solution: Denote the estimated parameter values at some particular cycle of the algorithm by $\boldsymbol{\theta}_{old}$.
E-step: Use the current parameter values $\boldsymbol{\theta}_{old}$ to find the posterior distribution of the latent variables given by $p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}, \boldsymbol{\theta}_{old})$. Then, use $p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}, \boldsymbol{\theta}_{old})$ to find the expectation of the complete-date log likelihood evaluated for some general parameter value $\boldsymbol{\theta}$. The expectation is denoted as $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{old})$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{old}) = \sum_{\mathbf{z}} p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}, \boldsymbol{\theta}_{old}) \ln p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}|\boldsymbol{\theta}) \tag{21}$$

$$= \mathbb{E}_{\mathbf{z}_{1:T}|\mathbf{x}_{1:T}, \boldsymbol{\theta}_{old}}[\ln p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}|\boldsymbol{\theta})] \tag{22}$$

M-step: maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{old})$ with respect to the component of $\boldsymbol{\theta}$.

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.,

$$\mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^\top] = \mathbf{J}_{t-1}\widehat{\mathbf{V}}_t + \widehat{\boldsymbol{\mu}}_t\widehat{\boldsymbol{\mu}}_{t-1}^\top \tag{23}$$

$$\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top] = \widehat{\mathbf{V}}_t + \widehat{\boldsymbol{\mu}}_t\widehat{\boldsymbol{\mu}}_t^\top \tag{24}$$

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

Solution: From problem 1.f, it is known that $\gamma(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t|\widehat{\boldsymbol{\mu}}_t, \widehat{\mathbf{V}}_t)$. $\mathbb{E}[\mathbf{z}_t]$ is the mean of the normal distribution $\widehat{\boldsymbol{\mu}}_t$, where $\widehat{\boldsymbol{\mu}}_t = \boldsymbol{\mu}_t + \mathbf{J}_t(\widehat{\boldsymbol{\mu}}_{t+1} - \mathbf{A}\boldsymbol{\mu}_T)$.

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

$$\log p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}|\boldsymbol{\theta}) = \log p(\mathbf{z}_1|\boldsymbol{\mu}_0, \mathbf{V}_0) + \sum_{t=2}^{T} \log p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{A}, \boldsymbol{\Gamma}) + \sum_{t=1}^{T} \log p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{C}, \boldsymbol{\Sigma}) \tag{25}$$

Solution: According to the DGM, we have:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}|\theta) = p(\mathbf{z}_1)\left[\prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{z}_{t-1})\right]\prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_t) \tag{26}$$

Take logarithm on both side leads to the answer.

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

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

Solution: $p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}|\theta)$

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

$$\boldsymbol{\mu}_0^{\text{new}} = \mathbb{E}[\mathbf{z}_1] \tag{27}$$

$$\mathbf{V}_0^{\text{new}} = \mathbb{E}[\mathbf{z}_1\mathbf{z}_1^\top] - \mathbb{E}[\mathbf{z}_1]\mathbb{E}[\mathbf{z}_1^\top] \tag{28}$$

$$\mathbf{A}^{\text{new}} = \left(\sum_{t=2}^{T} \mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^\top]\right)\left(\sum_{t=2}^{T} \mathbb{E}[\mathbf{z}_{t-1}\mathbf{z}_{t-1}^\top]\right)^{-1} \tag{29}$$

$$\boldsymbol{\Gamma}^{\text{new}} = \frac{1}{T-1}\sum_{t=2}^{T}\left(\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top] - \mathbf{A}^{\text{new}}\mathbb{E}[\mathbf{z}_{t-1}\mathbf{z}_t^\top] - \mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^\top](\mathbf{A}^{\text{new}})^\top + \mathbf{A}^{\text{new}}\mathbb{E}[\mathbf{z}_{t-1}\mathbf{z}_{t-1}^\top](\mathbf{A}^{\text{new}})^\top\right) \tag{30}$$

$$\mathbf{C}^{\text{new}} = \left(\sum_{t=1}^{T} \mathbf{x}_t\mathbb{E}[\mathbf{z}_t^\top]\right)\left(\sum_{t=1}^{T} \mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top]\right)^{-1} \tag{31}$$

$$\boldsymbol{\Sigma}^{\text{new}} = \frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{x}_t\mathbf{x}_t^\top - \mathbf{C}^{\text{new}}\mathbb{E}[\mathbf{z}_t]\mathbf{x}_t^\top - \mathbf{x}_t\mathbb{E}[\mathbf{z}_t^\top](\mathbf{C}^{\text{new}})^\top + \mathbf{C}^{\text{new}}\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top](\mathbf{C}^{\text{new}})^\top\right) \tag{32}$$

Problem 1.k. Try implementing the above update equations to perform EM for a linear dynamical system. When does the method work well and when does it fail? We have provided some sample source code to get you started on Canvas.

Solution: There could be several reasons for a model to fail.

- If the underlying "true" system is non-linear, modeling as a linear Gaussian dynamical system is likely work poorly.

- If we have very little data, the estimated dynamics may not be accurate (e.g., due to overfitting), which can lead to poor performance. However, using Bayesian methods can sometimes counteract some of this poor performance.

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?

Solution: The posteriori distribution for a linear dynamical system is a Gaussian. The MAP (mode) is the mean of the Gaussian posterior.