Chapter 1

Model definition

1.1 Notations

We consider the prediction task of a set of observations $(y^1, \cdots y^T)$ given a set of input $(u^1, \cdots u^T)$.

1.2 Model

We define a L layer RNN followed by a fully connected layer. At time step t,

$$\begin{cases} y_{t+1} = \tanh(W_y x_{t+1}^L + b_y) \\ x_{t+1}^l = \tanh(W_{xx}^l x_t^l + W_{xu}^l x_{t+1}^{l-1} + b_x^l) \quad \forall 1 \leq l \leq L \end{cases}$$

with $x_t^0 \equiv u_t \ \forall t$ and $x_0^l \equiv 0 \ \forall 1 \leq l \leq L$.

Let's consider the weights of the last RNN and fully connected layers as $\theta \equiv (W_{xx}^L, W_{xu}^L, b_x^L, W_y, b_y)$. We can define a new matrix y_t at each time step corresponding to the concatenation of all RNN layers: $x_t \equiv (x_t^1 \cdots x_t^L)$. We also introduce two sequences of random noises as i.i.d real valued random variables ϵ and η. We can now write our model in terms of two functions f and g as:

$$\begin{cases} y_{t+1} = f_\theta(x_{t+1}) + \epsilon_{t+1} & \text{observation model} \\ x_{t+1} = g_\theta(x_t, u_{t+1}) + \eta_{t+1} & \text{state model} \end{cases} \tag{1.1}$$

In the following section, we will focus on minimizing the log likelihood

$$\log \ p_\mu(X_{1:T}, y_{1:T}, u_{1:T}) \tag{1.2}$$

1.3 Minimization

In order to minimize 1.2, we apply an EM strategy. Let $\mu_p = (\theta_p, \Sigma_{x,p}, \Sigma_{y,p})$, we will compute at each EM step:

$$Q(\hat{\mu}_p, \mu) = \mathbb{E}_{\hat{\mu}_p} \left[\log\ p_\mu(X_{1:T}, y_{1:T}, u_{1:T}) | y_{1:T}\right] \tag{1.3}$$

We can start by developing the log likelihood:

$$\begin{aligned}
\log p_\mu(X_{1:T}, y_{1:T}, u_{1:T}) &= \frac{1}{T} \log \left(\prod_{k=1}^{T} p_\mu(x_k | x_{k-1}, u_k) p_\mu(y_k | x_k) \right) \\
&= \frac{1}{T} \sum_{k=1}^{T} \log\ p_\mu(x_k | x_{k-1}, u_k) + \log\ p_\mu(y_k | x_k) \\
&= \frac{1}{T} \sum_{k=1}^{T} \log \left(\det(2\pi\Sigma_x)^{-1/2} \exp(-\frac{1}{2}(x_k - g_\theta(x_{k-1}, u_k))^T \Sigma_x^{-1} (x_k - g_\theta(x_{k-1}, u_k)) \right) \\
&+ \log \left(det(2\pi\Sigma_y)^{-1/2} \exp(-\frac{1}{2}(y_k - f_\theta(x_k))^T \Sigma_y^{-1} (y_k - f_\theta(x_k)) \right) \\
&= -\frac{1}{2} \log |\Sigma_x| - \frac{1}{2} \log |\Sigma_y| \\
&- \frac{1}{2T} \sum_{k=1}^{T} (x_k - g_\theta(x_{k-1}, u_k))^T \Sigma_x^{-1} (x_k - g_\theta(x_{k-1}, u_k)) \\
&- \frac{1}{2T} \sum_{k=1}^{T} (y_k - f_\theta(x_k))^T \Sigma_y^{-1} (y_k - f_\theta(x_k))
\end{aligned}$$

We will jointly update Σ_x, Σ_y and θ iteratively. We can start by computing the explicit form of the minimum of both covariance matrices.

For Σ_y, we search the zeros of the derivate of the convex function $\Sigma_y \mapsto p_\mu(X_{1:T}, y_{1:T}, u_{1:T})$.

$$\frac{\partial p_\mu(X_{1:T}, y_{1:T}, u_{1:T})}{\partial \Sigma_y^{-1}} = \frac{1}{2}\Sigma_y - \frac{1}{2T} \sum_{k=1}^{T} (x_k - f_\theta(x_k)) \cdot (x_k - f_\theta(x_k))'$$

$$\frac{\partial p_\mu(X_{1:T}, y_{1:T}, u_{1:T})}{\partial \Sigma_y^{-1}} = 0 \implies \Sigma_y = \frac{1}{T} \sum_{k=1}^{T} (y_k - f_\theta(x_k))(y_k - f_\theta(x_k))'$$

$$\frac{\partial p_\mu(X_{1:T}, y_{1:T}, u_{1:T})}{\partial \Sigma_x^{-1}} = 0 \implies \Sigma_x = \frac{1}{T} \sum_{k=1}^{T} (x_k - g_\theta(x_{k-1}, u_k))(x_k - g_\theta(x_{k-1}, u_k))'$$

We now have an expression for minimizing both Σ matrices given a value of θ, but we can't compute an explicit form for minimizing θ. We can identify two approaches to jointly minimizing μ:

1. At each step of the EM algorithm, we can compute the maximum expectation for both covariant matrices given the previous value of θ, then approximate the new θ by minimizing an argmin, through gradient descent for example.

$$\Sigma_{y,p+1} = \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}_{\hat{\mu}_p}\left[(y_k - f_{\theta_p}(x_k))(y_k - f_{\theta_p}(x_k))'|y_{1:T}\right]$$

$$\Sigma_{x,p+1} = \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}_{\hat{\mu}_p}\left[(x_k - g_{\theta_p}(x_{k-1}, u_k))(x_k - g_{\theta_p}(x_{k-1}, u_k))'|y_{1:T}\right]$$

$$\begin{aligned}\theta_{p+1} = \underset{\theta}{\operatorname{argmin}} \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}_{\hat{\mu}_p}[&(y_k - f_{\theta_p}(x_k))'\Sigma_{y,p+1}^{-1}(y_k - f_{\theta_p}(x_k)) \\ &+ (x_k - g_{\theta_p}(x_{k-1}, u_k)'\Sigma_{x,p+1}^{-1}(x_k - g_{\theta_p}(x_{k-1}, u_k))|y_{1:T}]\end{aligned}$$

2. We can also ignore the explicit expression for the covariant matrices, and approximate both θ and Σ by gradient descent at each time step. Although we're putting aside a valuable result about Σ, this method could prove more efficient from an implementation perspective.

1.4 Sequential Monte Carlo Approach

We approximate the expectation conditional to the observations by using a particle filter. We sample M trajectories $\xi_{1:T}^m$ associated with the weight ω_T^m. At each time step k, we have:

$$\begin{aligned}\Phi_k^M[h] &= \mathbb{E}_{\hat{\mu}_p}\left[h(x_k)|y_{1:T}\right] \\ &= \frac{1}{\Omega_T^M}\sum_{m=1}^{M} \omega_T^m h(\xi_k^m)\end{aligned}$$

For all h measurable bounded, where $\Omega_T^M = \sum_{m=1}^{M} \omega_T^m$. In the following sections, we consider that the particles weights sum at 1.

1.5 Gradient descent

At each iteration of the EM algorithm, we start by generating a set of particles under the law $p(x|y)$, that allows us to compute a explicit value for the expectation. We can then minimize this expectation, in order to approximate the new θ candidate, using a gradient descent.

$$\begin{aligned}\theta_{p+1} = \underset{\theta}{\operatorname{argmin}} \frac{1}{T}\sum_{k=1}^{T}\sum_{m=1}^{M} &\omega_T^m (y_k - f_{\theta_p}(\xi_k^m))'\Sigma_{y,p+1}^{-1}(y_k - f_{\theta_p}(\omega_k^m)) \\ &+ \omega_T^m(\xi_k^m - g_{\theta_p}(\xi_{k-1}^m, u_k)'\Sigma_{x,p+1}^{-1}(\xi_k^m - g_{\theta_p}(\xi_{k-1}^m, u_k))\end{aligned}$$