PROJECT REPORT, PGM

NICOLAS JOUVIN, THOMAS KERDREUX, AND LOUIS THIRY

INTRODUCTION

This article [1] presents an algorithm to learn non-linear stochastic dynamical system with incomplete observation. This algorithm is an instance of EM algorithm, estimating the distributions of the hidden variables in the E-Step and learning the parameters of the non-linear functions in the M-Step. Actually, it uses very specific assumptions about the non linearities, namely that they can be decomposed as linear combination of Radial Basis Functions plus an affine term. This assumption allows to compute efficiently the exact M-steps in order to avoid sampling issues for integral calculations, which can be very costly if they need to be done at each step.

The article provides a few keys, as to derive the machinery, but most of the computations are left to the reader. With the help of the book of Simon Haykin [2], from which the article was published, and the lectures notes, we managed to derive the computations in the most general case[1]. Firstly, we present them, following the guideline of the article, and then present our numerical results on simulated data.

1. The model

This article deals with a non-linear stochastic dynamical system in discrete-time. The authors model with this systems with five sequences :

- $(x_t)_{t=1\dots T}$ (in $\mathbb{R}^p$) are called the **states** and are **unknown**.
- $(u_t)_{t=1\dots T}$ (in $\mathbb{R}^q$) are called the **inputs** and are **observed**.
- $(y_t)_{t=1\dots T}$ (in $\mathbb{R}^n$) are called the **outputs** and are **observed**.
- $(v_t)_{t=1\dots T-1}$ and $(w_t)_{t=1\dots T}$ (in $\mathbb{R}^p \times \mathbb{R}^n$) are zero-mean **Gaussian iid noises** of covariance matrices $Q \in \mathbb{R}^{p\times p}$ and $R \in \mathbb{R}^{n\times n}$, and are unknown.

[1] The absence of these computations in the initial article explains the length of this rapport, which we guess you hoped shorter. But they are critical for both the comprehension and the implementation so we chose to include at least the main ones.

and with two **non-linear differentiable** functions :

$$\begin{aligned} f\colon \mathbb{R}^p \times \mathbb{R}^q &\to \mathbb{R}^p \\ (x,u) &\mapsto f(x,u) \\ g\colon \mathbb{R}^p \times \mathbb{R}^q &\to \mathbb{R}^n \\ (x,u) &\mapsto g(x,u) \end{aligned}$$

The stochastic dynamics equations are:

$$x_{t+1} = f(x_t, u_t) + v_t \tag{1}$$

$$y_t = g(x_t, u_t) + w_t \tag{2}$$

This system can be represented with the graphical model:

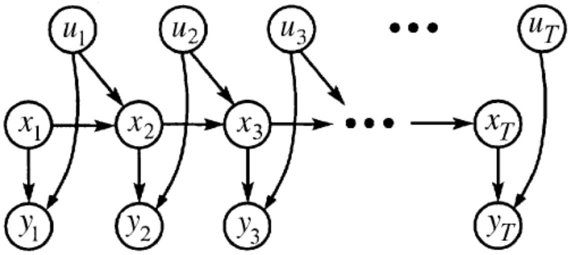


Figure 1. Graphical model of our system.

2. Presented algorithm

The goal of the algorithm presented in this article is to learn the dynamics of the system, which means to learn the functions f and g and the noise covariance matrices Q and R that are supposed to be unknown. The functions f and g are parametrized by θ_f and θ_g.

So our dynamical system is parametrized by $\theta = (\theta_f, \theta_g, Q, R)$ and we want to learn the value of θ with the maximum likelihood principle.

The complication is that both the parameter θ and the state sequence $(x_t)_{t=1\cdots T}$ are unknown. In such case where both states and parameters are unknown, a typical approach is to use an iterative Expectations Maximization (EM) algorithm:

- we begin with initial values for the parameter $\theta^{(0)} = \left(\theta_f^{(0)}, \theta_g^{(0)}, Q^{(0)}, R^{(0)}\right)$.
- For $k = 1 \dots K$, we iterate E-step and M-step:
  - The **E-step** consists in a *smoothing* problem, where we want to infer $p(x_t|y_1, \dots, y_T, u_1, \dots, u_T)$, the distributions of the states knowing observations y and inputs u. To that end, the article propose to use an Extended Kalman Smoother, in order to account for the non-linearity of the dynamic.
  - In the **M-step**, the updated parameter $\theta^{(k)} = \left(\theta_f^{(k)}, \theta_g^{(k)}, Q^{(k)}, R^{(k)}\right)$ is computed thanks to the states $(x_t)_{t=1\cdots T}$ inferred in the E-Step.

3. Parametrization of the functions

As we said above, the functions f and g must be parametrized. The authors propose the following for f and g :

$$f(x,u) = \sum_{i=1}^{I} \rho_i(x) h_i + Ax + Bu + b$$
$$g(x,u) = \sum_{j=1}^{J} \rho'_j(x) k_j + Cx + Du + d$$

$$\rho_i(x) \colon \begin{array}{ccc} \mathbb{R}^p & \to & \mathbb{R} \\ x & \mapsto & \frac{1}{(2\pi)^{d/2}|S_i|^{1/2}} \exp\left(-\frac{1}{2}(x-c_i)^T S_i^{-1}(x-c_i)\right) \end{array}$$

$$\rho'_j(x) \colon \begin{array}{ccc} \mathbb{R}^p & \to & \mathbb{R} \\ x & \mapsto & \frac{1}{(2\pi)^{d/2}|S'_j|^{1/2}} \exp\left(-\frac{1}{2}(x-c'_j)^T {S'_j}^{-1}(x-c'_j)\right) \end{array}$$

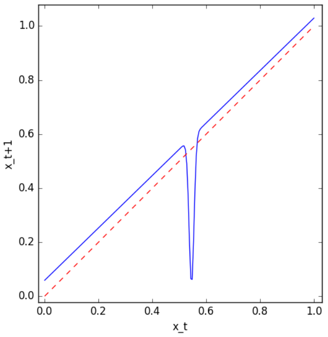
$$A \in \mathbb{R}^{p\times p},\ B \in \mathbb{R}^{p\times q},\ C \in \mathbb{R}^{n\times p},\ D \in \mathbb{R}^{n\times q}$$
$$h_i \in \mathbb{R}^p,\ b \in \mathbb{R}^p,\ k_j \in \mathbb{R}^n,\ d \in \mathbb{R}^n$$

The real valued functions ρ_i and ρ'_j are called radial basis functions (RBF). The *radial* terminology represent the dependency on the hilbertian distance (induced by the p.s.d matrix S_i) to a center c_i. **The centers c_i and c'_j of the RBF are supposed to be known, as well as the width S_i and S'_j.** Actually they are a sort of hyper-parameters for this method and we'll discuss later of a good way to set them but, for the moment, we suppose them as fixed since they will not be learned in the EM.

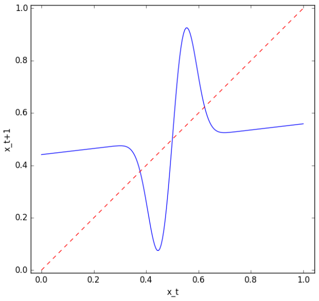
To sum things up, f is the sum of an affine function ($x \to Ax + b$) with a linear combination of I radial basis functions ($x \mapsto \rho_i(x)$, $i = 1 \cdots I$). The parameters θ_f and θ_g of our functions are :

$$\theta_f = (h_1, \ldots, h_I, A, B, b) \in \mathbb{R}^{p\times(I+p+q+1)}$$
$$\theta_g = (k_1, \ldots, k_J, C, D, d) \in \mathbb{R}^{n\times(J+p+q+1)}$$

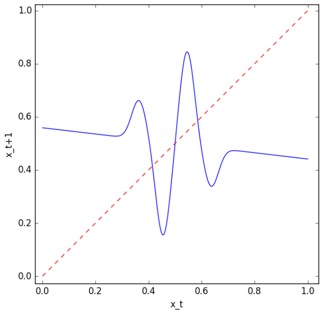
The following plots present examples of such functions, when we use no inputs u and set the state dimension to 1 ($p = 1$)The functions $f_I : \mathbb{R} \to \mathbb{R}$ where plotted on the segment $[0, 1]$:



I = 1



I = 2



I = 4

Now that the functions f and g are parametrized, we can begin with the study of the proposed EM algorithm.

4. E-Step

At iteration k, in the E-step, given the parameter $\theta^{(k-1)} = \left(\theta_f^{(k-1)}, \theta_g^{(k-1)}, Q^{(k-1)}, R^{(k-1)}\right)$ computed in the iteration $k-1$, we want to infer the sequences $(x_t)_{t=1\ldots T}$ and $(x_t, x_{t+1})_{t=1\ldots T-1}$ knowing the output sequence $(y_t)_{t=1\cdots T}$ and the input sequence $(u_t)_{t=1\cdots T}$. In other words we want to compute the conditional probabilities :

$$p_{\theta^{(k-1)}}\left(x_t \mid y_1, u_1, \ldots, y_T, u_T\right), t = 1\ldots T$$
$$p_{\theta^{(k-1)}}\left(x_t, x_{t+1} \mid y_1, u_1, \ldots, y_T, u_T\right), t = 1\ldots T-1$$

Essentially we are able to solve this inference problem (filtering or smoothing) when the dynamic is linear. However, if we use the non-linear dynamics equations with f and g, we have no guarantees that, starting from a Gaussian probability for x_1, we will have a Gaussian probability for x_t, and the computations become intractable. **To solve this problem, the authors propose to linearize the dynamics equations (1,2) by their first order Taylor expansion at each time step t and at a well-chosen point for filtering ($\tilde{x}_t$) and for smoothing ($\dot{x}_t$) :**

$$\begin{aligned}
x_{t+1} &= f(\tilde{x}_t, u_t) + A_{\tilde{x}_t}(x_t - \tilde{x}_t) + v_t \\
A_t &= \frac{\partial f}{\partial x}(\tilde{x}_t, u_t) \\
y_t &= g(\tilde{x}_t, u_t) + C_{\tilde{x}_t}(x_t - \tilde{x}_t) + w_t \\
C_t &= \frac{\partial g}{\partial x}(\tilde{x}_t, u_t).
\end{aligned}$$

Then the functions are locally linear, thus all probabilities are Gaussian and we can adapt the classical inferences algorithms (**Kalman filter** for filtering and **Rauch-Tung-Stribel smoother** for smoothing).

This forward-backward approach, involving a linearization at each time step, is called Extented Kalman Smoothing (EKS). **One of the important issues is to define $\tilde{\mathbf{x}}_\mathbf{t}$ and $\dot{\mathbf{x}}_\mathbf{t}$.**

4.1. **Kalman filter.** Let's first recall how Kalman filter proceed with inputs. It is a particular case where f and g are affine (e.g. $I = 0$ and $J = 0$) :

$$\begin{aligned} f(x,u) &= Ax + Bu + b \\ g(x,u) &= Cx + Du + d \end{aligned}$$

the Kalman filter algorithm computes dynamically the conditional probabilities :

$$\begin{aligned} &p_{\theta^{(k-1)}}\left(x_t \mid y_1, u_1, \dots, y_t, u_t\right), t = 1 \dots T \\ &p_{\theta^{(k-1)}}\left(x_t, x_{t+1} \mid y_1, u_1, \dots, y_t, u_t\right), t = 1 \dots T-1 \end{aligned}$$

Putting a gaussian probability on the first state x_0 and assuming the **additive** noise sequences (w_t) and (v_t) are Gaussian, since f and g are affine here, all the probabilities we are dealing with are then Gaussian. Thus we only need to compute means and covariances.

The authors derive the computations of the Kalman filter only in the case where B, b, D and d are all zero. They didn't give any reasons for these simplifications, but we extended the computations to take those into account.

As in [3], we denote by $\hat{x}_{t|t}$ the mean of x_t conditioned on $u_1; \cdots; u_t$ and for $\hat{P}_{t|t}$ the covariance of x_t given $u_1; \cdots; u_t$. It is straightforward to adapt these notations in the smoothing case ($\hat{P}_{t|T}$ and $\hat{x}_{t|T}$). We adopt the following notations :

$$\begin{aligned} \hat{x}_{t|t} &= \mathbb{E}_p\left(x_t \mid y_1, \dots, y_t, u_1, \dots, u_t\right) \\ \hat{x}_{t+1|t} &= \mathbb{E}_p\left(x_{t+1} \mid y_1, \dots, y_t, u_1, \dots, u_t\right) \\ \hat{P}_{t|t} &= \mathbb{E}_p\left((x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^\top \mid y_1, \dots, y_t, u_1, \dots, u_t\right) \\ \hat{P}_{t+1|t} &= \mathbb{E}_p\left((x_{t+1} - \hat{x}_{t+1|t})(x_{t+1} - \hat{x}_{t+1|t})^\top \mid y_1, \dots, y_t, u_1, \dots, u_t\right) \\ \hat{P}_{t,t+1|t} &= \mathbb{E}_p\left((x_t - \hat{x}_{t|t})(x_{t+1} - \hat{x}_{t+1|t})^\top \mid y_1, \dots, y_t, u_1, \dots, u_t\right) \end{aligned}$$

As already mentioned, computing these quantities amounts to infer the conditional probability distribution of the hidden states since we are working with gaussians distribution. The Kalman filter gives us a recursion equation to compute these quantities :

- **Initialization**

$$\begin{aligned}\hat{x}_{1|0} &= 0\\ \hat{P}_{1|0} &= \Sigma_0\end{aligned}$$

- **Forward recursion**
  For $t = 1 \ldots T$:

$$\begin{aligned}\hat{x}_{t+1|t} &= A\hat{x}_{t|t} + Bu_t + b\\ \hat{P}_{t+1|t} &= A\hat{P}_{t|t}A^\top + Q\\ \hat{P}_{t,t+1|t} &= \hat{P}_{t|t}A^\top\\ K_{t+1} &= \hat{P}_{t+1|t}C^\top (C\hat{P}_{t+1|t}C^\top + R^{(k-1)})^{-1}\\ \hat{x}_{t+1|t+1} &= \hat{x}_{t+1|t} + K_{t+1}\left(y_{t+1} - (C\hat{x}_{t+1|t} + Du_t + d)\right)\\ \hat{P}_{t+1|t+1} &= \hat{P}_{t+1|t} - K_{t+1}C\hat{P}_{t+1|t}\end{aligned}$$

We compute the matrix K_t (Kalman gain matrix) to simplify the computations.

In our case, f and g are not linear. So for $t = 1..T$ we linearize our functions f and g around the point $\tilde{x}_t$ and then we can apply the recursion equation below. We define $\tilde{x}_t$ as following :

$$\begin{aligned}\tilde{x}_1 &= \hat{x}_{1|0}\\ \tilde{x}_{t+1} &= f(\hat{x}_{t|t}, u_t)\end{aligned}$$

Then, the dynamics become :

$$\begin{aligned}A_t &= \frac{\partial f}{\partial x}(\tilde{x}_t, u_t) = A + \sum_{i=1}^{I} \rho_i(\tilde{x}_t) h_i (\tilde{x}_t - c_i)^\top S_i^{-1}\\ C_t &= \frac{\partial g}{\partial x}(\tilde{x}_t, u_t) = C + \sum_{j=1}^{J} \rho'_j(\tilde{x}_t) k_j (\tilde{x}_t - c'_i)^\top S'^{-1}_j\\ x_{t+1} &= A_t x_t + (f(\tilde{x}_t, u_t) - A_t\tilde{x}_t) + v_t\\ y_t &= C_t x_t + (g(\tilde{x}_t, u_t) - C_t\tilde{x}_t) + w_t\end{aligned}$$

in the recursion formula, we just have to replace the matrices A and C by the matrices A_t and C_t and we replace the vectors b and d by the vectors b_t and d_t :

$$\begin{aligned} b_t &= f(\tilde{x}_t, u_t) - A_{\tilde{x}_t}\tilde{x}_t \\ d_t &= g(\tilde{x}_t, u_t) - C_{\tilde{x}_t}\tilde{x}_t \end{aligned}$$

and we get the conditionnal probabilities of interest :

$$\begin{aligned} &p_{\theta^{(k-1)}}\left(x_t \mid y_1, u_1, \ldots, y_t, u_t\right), t = 1 \ldots T \\ &p_{\theta^{(k-1)}}\left(x_t, x_{t+1} \mid y_1, u_1, \ldots, y_t, u_t\right), t = 1 \ldots T-1 \end{aligned}$$

4.2. **Rauch-Tung-Stribel smoother.** The authors provide no computations for the RTS. So all the computations done here were inspired from the computations done in the lecture notes.

If the case where f and g are affine, the RTS smoother algorithm uses the conditionnal probabilities computed with Kalman filter to compute the conditional probabilities :

$$\begin{aligned} &p_{\theta^{(k-1)}}\left(x_t \mid y_1, u_1, \ldots, y_T, u_T\right), t = 1 \ldots T \\ &p_{\theta^{(k-1)}}\left(x_t, x_{t+1} \mid y_1, u_1, \ldots, y_T, u_T\right), t = 1 \ldots T-1 \end{aligned}$$

Once again, theses probabilities are Gaussian and only we need to compute their means and covariances.

The following notations follow the same rule as for Kalman filtering:

$$\begin{aligned} \hat{x}_{t|T} &= \mathbb{E}_p(x_t \mid y_1, \ldots, y_T, u_1, \ldots, u_T) \\ \hat{P}_{t|T} &= \mathbb{E}_p\left((x_t - \hat{x}_{t|T})(x_t - \hat{x}_{t|T})^\top \mid y_1, \ldots, y_T, u_1, \ldots, u_T\right) \\ \hat{P}_{t,t+1|T} &= \mathbb{E}_p\left((x_t - \hat{x}_{t|T})(x_{t+1} - \hat{x}_{t+1|T})^\top \mid y_1, \ldots, y_T, u_1, \ldots, u_T\right) \end{aligned}$$

With these quanities, we can compute the conditionnal probabilities of interest. The Kalman filter gives us a recursion equation to compute these quanities :

- **Initialization**
  For the initialization, we need to run a Kalman filter to get the quantities $\hat{x}_{t|t}, \hat{P}_{t|t}$ for $t = 1 \ldots T$ and $\hat{x}_{t+1|t}, \hat{P}_{t+1|t}, \hat{P}_{t,t+1|t}$ for $t = a \ldots T-1$.

- **Backward recursion**
  For $t = T - 1 \ldots 1$:

$$\begin{aligned} L_t &= \hat{P}_{t|t} A^\top \hat{P}_{t+1|t}^{-1} \\ \hat{x}_{t|T} &= \hat{x}_{t|t} + L_t(\hat{x}_{t+1|T} - \hat{x}_{t+1|t}) \\ \hat{P}_{t|T} &= \hat{P}_{t|t} + L_t(\hat{P}_{t+1|T} - \hat{P}_{t+1|t}) L_t^\top \\ \hat{P}_{t,t+1|T} &= \hat{P}_{t,t+1|t} - (\hat{x}_{t|t} - \hat{x}_{t|T})(\hat{x}_{t+1|t} - \hat{x}_{t+1|T})^\top \end{aligned}$$

  We compute the matrix L_t to simplify the computations.

In our case, f and g are not linear. We linearize them around the point $\dot{x}_t$ and then we can apply the recursion equation below. We define as follow :

$$\dot{x}_t = \hat{x}_{t|t} = \mathbb{E}_p(x_t \mid y_1, \ldots, y_t, u_1, \ldots, u_t)$$

As we see, $\dot{x}_t$ can be seen as the mean of probability $p_{\theta^{(k-1)}}(x_t \mid y_1, u_1, \ldots, y_t, u_T)$ computed by the Kalman filter. We also see that in the recursion forumla, the matrix C and the vectors b and d do not appear. So we just have to replace the matrix A by the matrix A_t :

$$A_t = \frac{\partial f}{\partial x}(\dot{x}_t, u_t)$$

The computation of A_t in term of our parameters $\theta_f^{(k-1)}$ and $\theta_g^{(k-1)}$ and of the centers and width of our RBF functions is the same as in the Kalman filter, replacing $\tilde{x}$ by $\dot{x}$.

So we end up with the means and covariance matrices of the probabilities

$$\begin{aligned} &p_{\theta^{(k-1)}}(x_t \mid y_1, u_1, \ldots, y_T, u_T), t = 1 \ldots T \\ &p_{\theta^{(k-1)}}(x_t, x_{t+1} \mid y_1, u_1, \ldots, y_T, u_T), t = 1 \ldots T - 1 \end{aligned}$$

which are exactly what we wanted to compute during the E-step. So we can move to the M-step.

5. M-step

At iteration k, in the M-step, we assume that the state sequence $(x_t)_{t=1\ldots T}$ follow the probability $q(x_1, \ldots, x_T)$. The sequence of states being a Markov chain, the probability q

is entirely defined by the marginal $q(x_t)$ and the joints $q(x_t, x_{t+1})$:

$$q(x_t) = \mathcal{N}\left(\hat{x}_{t|T}, \hat{P}_{t|T}\right), \ t = 1 \dots T$$
$$q(x_t, x_{t+1}) = \mathcal{N}\left(\left[\begin{array}{c} \hat{x}_{t|T} \\ \hat{x}_{t+1|T} \end{array}\right], \left[\begin{array}{cc} \hat{P}_{t|T} & \hat{P}_{t,t+1|T} \\ \hat{P}^{\top}_{t,t+1|T} & \hat{P}_{t+1|T} \end{array}\right]\right), \ t = 1 \dots T-1$$

where the means $\hat{x}_{t|T}$ and covariance matrices $\hat{P}_{t|T}, \hat{P}_{t,t+1|T}$ were computed in the E-step (we choose smoothing). To simplify the following development we adopt the notations:

$$q_t(x) = q(x = x_t)$$
$$q_{t,t+1}(x, x') = q\left((x, x') = (x_t, x_{t+1})\right)$$

Now we want to compute the updated parameter $\theta^{(k)} = \left(\theta_f^{(k)}, \theta_g^{(k)}, Q^{(k)}, R^{(k)}\right)$.

The updated parameter $\theta^{(k)}$ is the maximizer of the expected complete likelihood L_q which is equal to the maximizer of the expected complete log-likelihood l_q:

$$\theta^{(k)} = \underset{\theta}{\text{argmax}}\ L_q(\theta) = \underset{\theta}{\text{argmax}}\ l_q(\theta)$$

First we need to express $l_q(\theta)$ using the graph factorization. Our graphical model is the following :

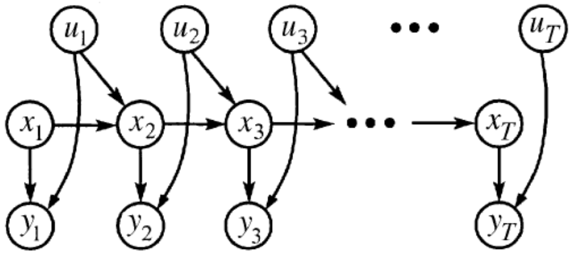


Figure 5. Graphical model of our system.

Firstly, the inputs u_t are not modeled as random variable, so we won't take them into account in the factorization. Thus, the factorization over the graph yields :

$$p_\theta(x_1, \dots x_T, y_1, \dots, y_T) = p_\theta(x_1) \prod_{t=1}^{T-1} p_\theta(x_{t+1} \mid x_t) \prod_{t=1}^{T} p_\theta(y_t \mid x_t)$$

Thus, the likelihood of the observed the sequence $\overline{y} = (\overline{y}_t)_{t=1\dots T}$ is :

$$p_\theta(x_1, \dots x_T, \overline{y}_1, \dots, \overline{y}_T) = p_\theta(x_1) \prod_{t=1}^{T-1} p_\theta(x_{t+1} \mid x_t) \prod_{t=1}^{T} p_\theta(\overline{y}_t \mid x_t)$$

Taking the log and expectation of this likelihood over the probability $q\,(x_1, \dots, x_T)$ gives us the expected log-likelihood:

$$l_q(\theta) \quad = \mathbb{E}_q(\log(p_\theta(x_1))) + \textstyle\sum_{t=1}^{T-1} \mathbb{E}_q\left(\log(p_\theta(x_{t+1} \mid x_t))\right) + \sum_{t=1}^{T} \mathbb{E}_q\left(\log(p_\theta(\overline{y}_t \mid x_t))\right)$$

And from the dynamic equation $(1, 2)$:

$$\begin{aligned}
\mathbb{E}_q(\log(p_\theta(x_1))) &= -\frac{1}{2} \int_x q_1(x)(x - \hat{x}_{1|1})^\top \hat{P}_{1|1}^{-1}(x - \hat{x}_{1|1})dx \\
\mathbb{E}_q\left(\log(p_\theta(x_{t+1} \mid x_t))\right) &= -\frac{1}{2} \int_{x,x'} q_{t,t+1}(x, x')((x' - f(x, u_t))^\top Q^{-1}(x' - f(x, u_t))dxdx' \\
\mathbb{E}_q\left(\log(p_\theta(\overline{y}_t \mid x_t))\right) &= -\frac{1}{2} \int_x q_t(x)(\overline{y}_t - g(x, u_t))^\top R^{-1}(\overline{y}_t - g(x, u_t))dx
\end{aligned}$$

Firstly, to get rid of the multiplying constants $-\frac{1}{2}$, we multiply the expected log likelihood by -2 such that our maximization problem becomes a minimization problem :

$$\theta^{(k)} = \underset{\theta}{\text{argmin}}\ l(\theta)$$

Then we notice that the expectation $\mathbb{E}_q(\log(p_\theta(x_1)))$ is constant with respect to θ so it doesn't account for the maximization. Introducing the author notations :

$$\begin{aligned}
\Phi_f(x, u) &= [\rho_1(x), \dots, \rho_I(x), x^\top, u^\top, 1]^\top \\
\Phi_g(x, u) &= [\rho'_1(x), \dots, \rho'_J(x), x^\top, u^\top, 1]^\top \\
\langle F(.)\rangle_t &= \int_x q_t(x)F(x)dx = \mathbb{E}_q\left(F(x_t)\right) \\
\langle F(.)\rangle_{t,t+1} &= \int_{x,x'} q_{t,t+1}(x, x')F(x, x')dxdx' = \mathbb{E}_q\left(F(x_t, x_{t+1})\right)
\end{aligned}$$

such that:

$$\begin{aligned}
f(x, u) &= \theta_f \Phi_f(x, u) \\
g(x, u) &= \theta_g \Phi_g(x, u)
\end{aligned}$$

and the functions l_q^1 and l_q^2:

$$l_q^1(\theta_f, Q) = \sum_{t=1}^{T-1} \Big\langle (x, x') \mapsto (x' - \theta_f \Phi_f(x, u_t))^\top Q^{-1} (x' - \theta_f \Phi_f(x, u_t)) \Big\rangle_{t,t+1} + (T-1)\log(|Q|)$$

$$l_q^2(\theta_g, R) = \sum_{t=1}^{T} \Big\langle x \mapsto (\overline{y}_t - \theta_g \Phi_g(x, u_t))^\top R^{-1} (\overline{y}_t - \theta_g \Phi_g(x, u_t)) \Big\rangle_{t} + T\log(|R|)$$

our minimization problem becomes:

$$\left(\theta_f^{(k)}, Q^{(k)}\right) = \operatorname*{argmin}_{\theta_k, Q} l_q^1(\theta_f, Q)$$
$$\left(\theta_g^{(k)}, R^{(k)}\right) = \operatorname*{argmin}_{\theta_g, R} l_q^2(\theta_g, R)$$

At the minimum point, the derivatives are zero :

$$\frac{\partial l_q^1}{\partial \theta_f} = 0$$
$$\frac{\partial l_q^1}{\partial Q} = 0$$
$$\frac{\partial l_q^2}{\partial \theta_g} = 0$$
$$\frac{\partial l_q^2}{\partial R} = 0$$

which yields:

$$\theta_f^{(k)} = \left(\sum_{t=1}^{T-1} \Big\langle (x, x') \mapsto x' \Phi_f(x, u_t)^\top \Big\rangle_{t+1}\right) \left(\sum_{t=1}^{T-1} \Big\langle x \mapsto \Phi_f(x, u_t) \Phi_f(x, u_t)^\top \Big\rangle_{t}\right)^{-1}$$

$$Q^{(k)} = \frac{1}{T-1} \left(\sum_{t=1}^{T-1} \Big\langle x' \mapsto x' x'^\top \Big\rangle_{t+1} - \theta_f^{(k)} \sum_{t=1}^{T-1} \Big\langle (x, x') \mapsto \Phi_f(x, u_t) x'^\top \Big\rangle_{t,t+1}\right)$$

$$\theta_g^{(k)} = \left(\sum_{t=1}^{T} \Big\langle x \mapsto \overline{y}_t \Phi_g(x, u_t)^\top \Big\rangle_{t}\right) \left(\sum_{t=1}^{T} \Big\langle x \mapsto \Phi_g(x, u_t) \Phi_g(x, u_t)^\top \Big\rangle_{t}\right)^{-1}$$

$$R^{(k)} = \frac{1}{T} \left(\sum_{t=1}^{T} \overline{y}_t \overline{y}_t^\top - \theta_g^{(k)} \sum_{t=1}^{T} \Big\langle x \mapsto \Phi_f(x, u_t) \overline{y}_t^\top \Big\rangle_{t}\right)$$

Now the keypoint is to compute these expectations. To compute $\theta_f^{(k)}$ and $Q^{(k)}$, we need to compute the expectations:

$$\left\langle (x,x') \mapsto x'\Phi_f(x,u_t)^\top \right\rangle_{t,t+1} = \left\langle (x,x') \mapsto [x'\rho_1(x),\ldots,x'\rho_I(x),x'x^\top,x'u_t^\top,x'] \right\rangle_{t,t+1}$$

$$\left\langle (x,x') \mapsto \Phi_f(x,u_t)x'^\top \right\rangle_{t,t+1} = \left\langle (x,x') \mapsto [x'\rho_1(x),\ldots,x'\rho_I(x),x'x^\top,x'u_t^\top,x']^\top \right\rangle_{t,t+1}$$

$$\left\langle x \mapsto \Phi_f(x,u_t)\Phi_f(x,u_t)^\top \right\rangle_t = \left\langle x \mapsto \begin{bmatrix} \rho_1(x)\rho_1(x) & \ldots & \rho_1(x)\rho_I(x) & \rho_1(x)x^\top & \rho_1(x)u_t^\top & \rho_1(x) \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \rho_I(x)\rho_1(x) & \ldots & \rho_I(x)\rho_I(x) & \rho_I(x)x^\top & \rho_I(x)u_t^\top & \rho_I(x) \\ \rho_1(x)x & \ldots & \rho_I(x)x & xx^\top & xu_t^\top & x \\ \rho_1(x)u_t & \ldots & \rho_I(x)u_t & u_tx^\top & u_tu_t^\top & u_t \\ \rho_1(x) & \ldots & \rho_I(x) & x^\top & u_t^\top & 1 \end{bmatrix} \right\rangle_t$$

$$\left\langle x' \mapsto x'x'^\top \right\rangle_{t+1}$$

So we need to compute the elementary expectations:

$$\langle (x,x') \mapsto \rho_i(x)x' \rangle_{t,t+1}$$
$$\left\langle (x,x') \mapsto x'x^\top \right\rangle_{t,t+1}$$
$$\langle (x,x') \mapsto x' \rangle_{t,t+1}$$
$$\left\langle x' \mapsto x'x'^\top \right\rangle_{t+1}$$
$$\langle x \mapsto \rho_i(x)\rho_j(x) \rangle_t$$
$$\langle x \mapsto \rho_i(x)x \rangle_t$$
$$\langle x \mapsto \rho_i(x) \rangle_t$$
$$\left\langle x \mapsto xx^\top \right\rangle_t$$
$$\langle x \mapsto x \rangle_t$$

We begin with the simple ones :

$$\begin{aligned}\left\langle x' \mapsto x' \right\rangle_{t+1} &= \hat{x}_{t+1|T} \\ \left\langle x' \mapsto x' x'^{\top} \right\rangle_{t+1} &= \hat{x}_{t+1|T} \hat{x}_{t+1|T}^{\top} + \hat{P}_{t+1|T} \\ \left\langle (x, x') \mapsto x' x^{\top} \right\rangle_{t,t+1} &= \hat{x}_{t+1|T} \hat{x}_{t|T}^{\top} + \hat{P}_{t,t+1|T}^{\top} \\ \left\langle x \mapsto x x^{\top} \right\rangle_{t} &= \hat{x}_{t|T} \hat{x}_{t|T}^{\top} + \hat{P}_{t|T} \\ \left\langle x \mapsto x \right\rangle_{t} &= \hat{x}_{t|T}\end{aligned}$$

For the other expectations, the authors make us observe that when we multiply the RBF function ρ_i and q_t, we obtain a Gaussian density in $\mathbb{R}^p$ with mean μ_t^i and covariance Σ_t^i :

$$\begin{aligned}\Sigma_t^i &= \left(\hat{P}_{t|T}^{-1} + S_i^{-1} \right)^{-1} \\ \mu_t^i &= \Sigma_t^i \left(\hat{P}_{t|T}^{-1} \hat{x}_{t|T} + S_i^{-1} c_i \right)\end{aligned}$$

multiplied by the constant β_t^i (due to non-normalization):

$$\begin{aligned}\beta_t^i &= (2\pi)^{-p/2} |S_i|^{-1/2} |\hat{P}_{t|T}|^{-1/2} |\Sigma_t^i|^{1/2} \exp(-\frac{1}{2}\delta_t^i) \\ \delta_t^i &= c_i^{\top} S_i^{-1} c_i + \hat{x}_{t|T}^{\top} \hat{P}_{t|T}^{-1} \hat{x}_{t|T} - {\mu_t^i}^{\top} \Sigma_t^i \mu_t^i\end{aligned}$$

Using these notations, the expectations involving x_t and ρ_i can be computed:

$$\begin{aligned}\left\langle x \mapsto \rho_i(x) x \right\rangle_t &= \beta_t^i \mu_t^i \\ \left\langle x \mapsto \rho_i(x) \right\rangle_t &= \beta_t^i\end{aligned}$$

Identically, when we multiply the product $\rho_i \rho_j$ and q_t, we obtain a Gaussian density in $\mathbb{R}^p$ with mean $\mu_t^{i,j}$ and covariance $\Sigma_t^{i,j}$:

$$\begin{aligned}\Sigma_t^{i,j} &= \left(\hat{P}_{t|T}^{-1} + S_i^{-1} + S_j^{-1} \right)^{-1} \\ \mu_t^{i,j} &= \Sigma_t^{i,j} \left(\hat{P}_{t|T}^{-1} \hat{x}_{t|T} + S_i^{-1} c_i + S_j^{-1} c_j \right)\end{aligned}$$

multiplied by the constant $\beta_t^{i,j}$ (due to non-normalization):

$$\begin{aligned}
\beta_t^{i,j} &= (2\pi)^{-p}|S_i|^{-1/2}|S_j|^{-1/2}|\hat{P}_{t|T}|^{-1/2}|\Sigma_t^{i,j}|^{1/2}\exp(-\frac{1}{2}\delta_t^{i,j}) \\
\delta_t^{i,j} &= c_i^\top S_i^{-1}c_i + c_j^\top S_j^{-1}c_j + \hat{x}_{t|T}^\top \hat{P}_{t|T}^{-1}\hat{x}_{t|T} - {\mu_t^{i,j}}^\top {\Sigma_t^{i,j}}^{-1}\mu_t^{i,j}
\end{aligned}$$

and the expectation involving the product $\rho_i\rho_j$ is:

$$\langle x \mapsto \rho_i(x)\rho_j(x)\rangle_t = \beta_t^{i,j}$$

Finally, when we multiply the RBF function ρ_i and $q_{t,t+1}$, we obtain a Gaussian density in $\mathbb{R}^{2p}$ with mean $\mu_{t,t+1}^i$ and covariance $\Sigma_{t,t+1}^i$:

$$\begin{aligned}
\mathcal{P}_{t,t+1|T} &= \left[\begin{array}{cc} \hat{P}_{t|T} & \hat{P}_{t,t+1|T} \\ \hat{P}_{t,t+1|T}^\top & \hat{P}_{t+1|T} \end{array}\right] \\
\mathcal{X}_{t,t+1|T} &= \left[\begin{array}{c} \hat{x}_{t|T} \\ \hat{x}_{t+1|T} \end{array}\right] \\
\Sigma_{t,t+1}^i &= \left(\mathcal{P}_{t,t+1|T}^{-1} + \left[\begin{array}{cc} S_i^{-1} & 0 \\ 0 & 0 \end{array}\right]\right)^{-1} \\
\mu_{t,t+1}^i = \left[\begin{array}{c} m_{t,t+1}^i \\ n_{t,t+1}^i \end{array}\right] &= \Sigma_{t,t+1}^i\left(\mathcal{P}_{t,t+1|T}^{-1}\mathcal{X}_{t,t+1|T} + \left[\begin{array}{c} S_i^{-1}c_i \\ 0 \end{array}\right]\right)
\end{aligned}$$

multiplied by a constant $\beta_{t,t+1}^i$:

$$\begin{aligned}
\beta_{t,t+1}^i &= (2\pi)^{-p/2}|S_i|^{-1/2}|\mathcal{P}_{t,t+1|T}|^{-1/2}|\Sigma_{t,t+1}^i|^{1/2}\exp(-\frac{1}{2}\delta_{t,t+1}^i) \\
\delta_{t,t+1}^i &= c_i^\top S_i^{-1}c_i + \mathcal{X}_{t,t+1|T}^\top \mathcal{P}_{t,t+1|T}^{-1}\mathcal{X}_{t,t+1|T} - {\mu_{t,t+1}^i}^\top {\Sigma_{t,t+1}^i}^{-1}\mu_{t,t+1}^i
\end{aligned}$$

Thus the last expectation is :

$$\left\langle (x,x') \mapsto \rho_i(x)x' \right\rangle_{t,t+1} = \beta_{t,t+1}^i n_{t,t+1}^i$$

We notice that the interest of the RBF functions is that the expectations can be computed analytically and efficiently.

Now, to compute $\theta_g^{(k)}$ and $R^{(k)}$, we need to compute the expectations:

$$\left\langle x \mapsto \overline{y}_t \Phi_g(x,u_t)^\top \right\rangle_t = \left\langle x \mapsto [\overline{y}_t \rho'_1(x), \dots, \overline{y}_t \rho'_J(x), \overline{y}_t x^\top, \overline{y}_t u_t^\top, \overline{y}_t] \right\rangle_t$$

$$\left\langle x \mapsto \Phi_g(x,u_t) \overline{y}_t^\top \right\rangle_t = \left\langle x \mapsto [\overline{y}_t \rho'_1(x), \dots, \overline{y}_t \rho'_J(x), \overline{y}_t x^\top, \overline{y}_t u_t^\top, \overline{y}_t]^\top \right\rangle_t$$

$$\left\langle x \mapsto \Phi_g(x,u_t)\Phi_g(x,u_t)^\top \right\rangle_t = \left\langle x \mapsto \begin{bmatrix} \rho'_1(x)\rho'_1(x) & \dots & \rho'_1(x)\rho'_J(x) & \rho'_1(x)x^\top & \rho'_1(x)u_t^\top & \rho'_1(x) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \rho'_J(x)\rho'_1(x) & \dots & \rho'_J(x)\rho'_J(x) & \rho'_J(x)x^\top & \rho'_J(x)u_t^\top & \rho'_J(x) \\ \rho'_1(x)x & \dots & \rho'_J(x)x & xx^\top & xu_t^\top & x \\ \rho'_1(x)u_t & \dots & \rho'_J(x)u_t & u_t x^\top & u_t u_t^\top & u_t \\ \rho'_1(x) & \dots & \rho'_J(x) & x^\top & u_t^\top & 1 \end{bmatrix} \right\rangle_t$$

For that, we need to compute the elementary expectations :

$$\left\langle x \mapsto \rho'_j(x)\rho'_k(x) \right\rangle_t$$
$$\left\langle x \mapsto \rho'_j(x)x \right\rangle_t$$
$$\left\langle x \mapsto \rho'_j(x) \right\rangle_t$$
$$\left\langle x \mapsto xx^\top \right\rangle_t$$
$$\left\langle x \mapsto x \right\rangle_t$$

The formulas for these expecations are the same as previously, replacing c_i, S_i, c_j, S_j by c'_j, S'_j, c'_k, S'_k.

6. The Initialization

In our models we are faced with two kind of parameters.

- **Hyper-parameters** are those that we are to set ourselves by build-out methods (cross-validation, selection criterion or home-made methods). In our case, I, J, (c_i, S_i),(c'_j, S'_j) and the dimension of the hidden variables are to be set. The article provides an heuristic way to set (c_i, S_i) and (c'_j, S'_j), that we will describe precisely.
- **Parameters to be learned** are those that the EM-algorithm will output, e.g. (θ_f, θ_g). **These parameters need to be initialized**. Note that the juncture between these two categories is quite porous. Indeed the centers and variances of the RBF Kernels could also be learn with an adaptive procedure inside the EM-algorithm. **Thus the setting of this parameters would become an initialization.**

Setting the Hyper-parameters is always a difficult issue, in particular in methods based of maximum-likelihood maximization because it cannot be used to discriminate the models.

Also initialization of the parameters we want to learn is a crucial issue, especially for EM-algorithm that converges to local minima. It is thus of interest to start the algorithm with a good guess.

6.1. Hyper-parameters:Fit the center and variance of RBF Kernel. Note that we will work with small (e.g. smaller than 4) for the dimension of the hidden variables.

En fait ca c'est pour haute dimension, en petite il suffit de grider l'espace...(pas exactement clair)

1. First run the EM with linear dynamics (e.g. $x_{k+1} = Ax_k + Bu_k + w_k$ and $y_k = Cx_k + Du_k + v_k$). **don't know how to initialize it**.
2. Run the E-step with the parameters learn by the previous EM-algorithm.
3. Pick at random I means of the sequence of states. Check that they are not to close to one another. **Ici aussi ce n'est pas clair la distance prise**. If some are too close, pick new ones. These will be the given sequence for the centers of the RBF Kernel.
4. **Set the width (e.g. the variance matrix S_i) by "once we have the spacing of their centers by attempting to make neighbouring kernels cross when their outputs are half of their peak value."**

6.2. **Initialization of (θ_g, θ_f).** So we need to initialize the set of parameters (**Est-ce que Q et R y sont?**):

$$\begin{aligned}\theta_f &= (f_1; \cdots; f_I; A; B; b; Q)\\ \theta_g &= (g_1; \cdots; g_J; C; D; d; R)\end{aligned}$$

6.2.1. *First case.* Consider the case in which the dynamics is non-linear with approximately linear output function:

$$x_{k+1} = \sum_{i=1}^{I} f_i \rho_i(x_k) + Ax_k + Bu_k + b + w_k \tag{3}$$

$$y_k = Cx_k + Du_k + d + v_k \tag{4}$$

The steps of the initialization are:

- Approximate the dynamics in order to apply a factor analysis. Namely (4) is not changed, while we consider that x_k follows $\mathcal{N}(0, R)$♣ *I don't understand why it is relevant* ♣ . Through an EM-algorithm, we come up with an estimate for C and an estimate for the states.
- Using regression to identify the θ_f parameters.

The factor analysis: Consider that x_k follows the law $\mathcal{N}(0, R)$ ♣ *Not sure if we are compelled that they follow the same law/ follow essentially the law of w_k* ♣ . Then from (4) we have that $y_k|x_k \sim \mathcal{N}(Cx_k + Du_k + d, R)$. ♣ *Note that in the coursework setting,*

$x_k \sim \mathcal{N}(0, I)$/ also Factor Analysis asks for R to be diagonal/ though was is important is that the EM estimation can be done correctly ♣ **A developper: DEMANDE UN GROS BOULOT!!!**
T

The regression step: We can then try to find the set of parameters by regressing the state at time k by the state estimation at the previous time. For instance finding the parameters that minimize:

$$\sum_{t=1}^{T-1} \mathbb{E}(||x_{k+1} - x_k||^2) \quad = \quad \sum_{t=1}^{T} \mathbb{E}||\sum_{i=1}^{I} h_i \rho_i(x_k) + Ax_k + Bu_k + b + w_k - x_k||^2 \tag{5}$$

where the expectation is taken over the inferred distribution of x_k in the factor analysis.

6.2.2. *Second case.* Now the case:

$$x_{k+1} \quad = \quad Ax_k + Bu_k + w_k \tag{6}$$

$$y_k \quad = \quad \sum_{i=1}^{J} g_i \rho_i(x_k) + Cx_k + Du_k + d + v_k \tag{7}$$

6.2.3. *In the general case?*

References

[1] Sam Roweis and Zoubin Gharamani. Learning non-linear dynamics using an EM algorithm. In *Kalman Filtering and Neural Networks*, chapter 6. Wiley Blackwell, 2001.

[2] Simon Haykin. Kalman filters. In *Kalman Filtering and Neural Networks*, chapter 1. Wiley Blackwell, 2001.

[3] Michael Jordan. Kalman filters. In *An introduction to graphical models*.