EECS 225A Statistical Signal Processing Lecture 10 & 11 - 02/20/2020

State Estimation for Hidden Markov Processes II

Lecturer: Jiantao Jiao *Scribe: Jiantao Jiao*

1 Recap of Hidden Markov Processes

Recall that a hidden Markov process has the following joint density

$$\begin{aligned} p(x^n, y^n) &= p(x^n)p(y^n \mid x^n) && (1)\\ &= \left(\prod_{t=1}^{n} p(x_t \mid x_{t-1})\right)\left(\prod_{t=1}^{n} p(y_t \mid x_t)\right) && (2)\end{aligned}$$

We have the following conditional independence relations.

$$\begin{aligned} (X^{t-1}, Y^t) - X_t - (X_{t+1}^n, Y_{t+1}^n) && \text{(a)}\\ (X^{t-1}, Y^{t-1}) - X_t - (X_{t+1}^n, Y_t^n) && \text{(b)}\\ X^{t-1} - (X_t, Y^{t-1}) - (X_{t+1}^n, Y_t^n) && \text{(c)}\\ X^{t-1} - (X_t, Y^n) - X_{t+1}^n && \text{(d)}\end{aligned}$$

1.1 Causal inference with forward recursion

In causal inference with forward recursion, we defined

- $\beta_t(x_t) = p(x_t \mid y^{t-1})$ posterior on the state x_t given past observations y^{t-1},
- $\alpha_t(x_t) = p(x_t \mid y^t)$ posterior on the state x_t given all observations y^t up to index t,

and the obtained the following algorithm:

```
Algorithm: Forward Recursion
Initialize: $\beta_1(x_1) = p(x_1)$    (prior probability on state)
For $t \geq 1$:
  1. $\alpha_t(x_t) = \frac{\beta_t(x_t)p(y_t \mid x_t)}{\sum_{x_t} \beta_t(x_t)p(y_t \mid x_t)}$    (measurement update)
  2. $\beta_{t+1}(x_{t+1}) = \sum_{x_t} \alpha_t(x_t)p(x_{t+1} \mid x_t)$    (time update)
```

1.2 Non-causal Inference with Backward Recursion

The causal forward recursion algorithm makes sense in applications where we are only allowed to use the state measurements up until the current index. In other applications, where causality is not a requirement (such as in image processing), we can also incorporate future measurements. In other words, we can calculate

$p(x_t \mid y^n)$, $t \leq n$ instead of just $p(x_t \mid y^t)$. We write,

$$\begin{aligned}
p(x_t \mid y^n) &= \sum_{x_{t+1}} p(x_t, x_{t+1} \mid y^n) \\
&= \sum_{x_{t+1}} p(x_{t+1} \mid y^n) p(x_t \mid x_{t+1}, y^t, y^n_{t+1}) \\
&= \sum_{x_{t+1}} p(x_{t+1} \mid y^n) p(x_t \mid x_{t+1}, y^t) \\
&= \sum_{x_{t+1}} p(x_{t+1} \mid y^n) \frac{p(x_t \mid y^t) p(x_{t+1} \mid x_t, y^t)}{p(x_{t+1} \mid y^t)}
\end{aligned}$$

where in the third step we used the independence property $X_t \,\text{---}\, (X_{t+1}, Y^t) \,\text{---}\, X^n_{t+1}$ (property (c)) and the last step follows from Bayes' rule conditioned on y^t. We can neglect the conditional dependence on y^t in the term $p(x_{t+1} \mid x_t, y^t)$, because of the independence property $Y^t \,\text{---}\, X_t \,\text{---}\, X_{t+1}$ (property (a)), thus, if we make the additional definition

- $\gamma_t(x_t) = p(x_t \mid y^n)$ posterior on the state x_t given all measurements y^n,

then

$$\gamma_t(x_t) = \sum_{x_{t+1}} \gamma_{t+1}(x_{t+1}) \frac{\alpha_t(x_t) p(x_{t+1} \mid x_t)}{\beta_{t+1}(x_{t+1})}. \tag{3}$$

Here, $\gamma_t(x_t)$ depends only on $\gamma_{t+1}(x_{t+1})$, the state transition probabilities $p(x_{t+1} \mid x_t)$, and the (known) posteriors $\alpha_t(x_t), \beta_{t+1}(x_{t+1})$. The backward recursion algorithm is presented below.

```
Algorithm: Backward Recursion
Initialize: $\gamma_n(x_n) = p(x_n \mid y^n) = \alpha_n(x_n)$   (from forward recursion)
For $t = n - 1$ to 1:
  1. $\gamma_t(x_t) = \sum_{x_{t+1}} \gamma_{t+1}(x_{t+1}) \frac{\alpha_t(x_t) p(x_{t+1} \mid x_t)}{\beta_{t+1}(x_{t+1})}$.
```

1.3 Reconstructions

Once we have the posteriors $p(x_t \mid y^t)$ or $p(x_t \mid y^n)$ from forward or backward recursion, we can make an estimate $\hat{x}_t$ of the underlying state in several ways, depending on the application.

- To minimize the probability of error (discrete state space):
  $\hat{x}_t = \arg\max_{x_t} p(x_t \mid y^t)$ (causal)
  $\hat{x}_t = \arg\max_{x_t} p(x_t \mid y^n)$ (non-causal)

- Conditional expectation, assuming $\mathcal{X} \subseteq \mathbb{R}$:
  $\hat{x}_t = \sum_{x_t} x_t p(x_t \mid y^t)$ (causal)
  $\hat{x}_t = \sum_{x_t} x_t p(x_t \mid y^n)$ (non-causal)

- The actual probabilities of each state:
  $\hat{x}_t = \{p(x_t \mid y^t)\}_{x_t}$ (causal)
  $\hat{x}_t = \{p(x_t \mid y^n)\}_{x_t}$ (non-causal)

- Samples from the posteriors:
  $\hat{x}_t \sim p(x_t \mid y^t)$ (causal)
  $\hat{x}_t \sim p(x_t \mid y^n)$ (non-causal)

2 Most likely sequence of hidden states: the Viterbi Algorithm

The most likely sequence of hidden states $\hat{x}^n$ is defined as

$$\hat{x}^n \triangleq \arg\max_{x^n} p(x^n|y^n).$$

The Viterbi algorithm computes $\hat{x}^n$ efficiently using the idea of dynamic programming. It defines the value function

$$V_t(x_t) \triangleq \max_{x^{t-1}} p(x^t, y^t) \tag{4}$$

and iteratively computes it. Clearly,

$$V_1(x_1) = p(x_1)p(y_1|x_1).$$

We have

$$\begin{aligned} p(x^t, y^t) &= p(x^{t-1}, y^{t-1})p(x_t, y_t|x^{t-1}, y^{t-1}) \\ &= p(x^{t-1}, y^{t-1})p(x_t|x^{t-1}, y^{t-1})p(y_t|x_t, x^{t-1}, y^{t-1}) \\ &= p(x^{t-1}, y^{t-1})p(x_t|x_{t-1})p(y_t|x_t), \end{aligned}$$

where in the last step we used conditional independence relation (a) to simplify $p(x_t|x^{t-1}, y^{t-1})$ and relation (b) to simplify $p(y_t|x_t, x^{t-1}, y^{t-1})$. Hence,

$$\begin{aligned} V_t(x_t) &= \max_{x^{t-1}} p(x^t, y^t) \\ &= \max_{x^{t-1}} p(x^{t-1}, y^{t-1})p(x_t|x_{t-1})p(y_t|x_t) \\ &= p(y_t|x_t) \max_{x^{t-1}} p(x^{t-1}, y^{t-1})p(x_t|x_{t-1}) \\ &= p(y_t|x_t) \max_{x_{t-1}} [p(x_t|x_{t-1}) \max_{x^{t-2}} [p(x^{t-1}, y^{t-1})]] \\ &= p(y_t|x_t) \max_{x_{t-1}} [p(x_t|x_{t-1})V_{t-1}(x_{t-1})] \end{aligned}$$

We just derived the iteration that computes V_t based on V_{t-1}, which efficiently produces V_t for all $1 \leq t \leq n$.

Now we traceback the optimum achieving states. Clearly,

$$\hat{x}_n = \arg\max_{x_n} V_n(x_n),$$

and it follows from the equation

$$V_t(x_t) = p(y_t|x_t) \max_{x_{t-1}} [p(x_t|x_{t-1})V_{t-1}(x_{t-1})] \tag{5}$$

that we can compute $\hat{x}_{t-1}$ based on $\hat{x}_t$ using

$$\hat{x}_{t-1} = \arg\max_{x_{t-1}} [p(\hat{x}_t|x_{t-1})V_{t-1}(x_{t-1})]. \tag{6}$$

The pseudocode for the Viterbi algorithm is below.

A few remarks are in order. To avoid underflow in numerical computations, usually one computes the logarithmic of all the probabilities and transform the product into sums. If the states x_t take values in a finite set with cardinality K, then the space complexity of the Viterbi algorithm is $O(nK)$ since we need to store n value functions and each one is a vector of dimension K. Its time complexity is $O(nK^2)$ since during each iteration we need to sweep over all the entries of a specific value function, and to compute each entry of a specific value function we need to compute the max operator, which again requires $O(K)$ operations.

```
function VITERBI
    $V_1(x_1) \leftarrow p(x_1)p(y_1|x_1)$                              ▷ Initialization of value function
    $\hat{x}^n \leftarrow \emptyset$                                     ▷ Initialization of the MAP estimator
    for $t = 2, \cdots, n$ do
        $V_t(x_t) = p(y_t|x_t)\max_{x_{t-1}}[p(x_t|x_{t-1})V_{t-1}(x_{t-1})]$    ▷ Forward computation of value function
    end for
    $\hat{x}_n = \arg\max_{x_n} V_n(x_n)$                                ▷ Maximum of overall log-likelihood
    for $t = n, \cdots, 2$ do
        $\hat{x}_{t-1} = \arg\max_{x_{t-1}}[p(\hat{x}_t|x_{t-1})V_{t-1}(x_{t-1})]$    ▷ Overall maximizing sequence
    end for
end function
```

3 Defining functions F and G

Throughout this section, random variables $X \in \mathcal{X}, Y \in \mathcal{Y}$, and a distribution on $\mathcal{X}$ is viewed as a member of $\mathcal{P}(\mathcal{X})$, and a conditional distribution $p_{Y|X}$ is viewed as a function mapping $\mathcal{X} \mapsto \mathcal{P}(\mathcal{Y})$. Here $\mathcal{P}(\mathcal{X})$ denotes the space of all probability distributions on $\mathcal{X}$. We denote the set of functions mapping from set A to B as B^A.

Consider a random variable X with known statistics, i.e., known $p_X(x)$. Let this random variable pass through a channel with known statistics $p_{Y|X}(y \mid x)$ in Figure 1.

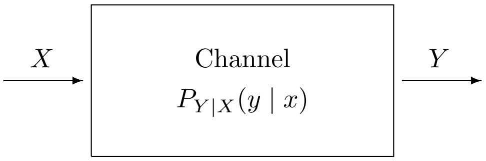

Figure 1: Pass random variable X through channel $P_{Y|X}$

The posterior of x given y is then given by the Bayes' rule:

$$p_{X|Y}(x \mid y) = \frac{p_X(x)p_{Y|X}(y \mid x)}{\sum_{\tilde{x}} p_X(\tilde{x})p_{Y|X}(y \mid \tilde{x})} \tag{7}$$

The marginal of Y is given by

$$p_Y(y) = \sum_{\tilde{x}} p_{X,Y}(\tilde{x}, y) = \sum_{\tilde{x}} p_X(\tilde{x})p_{Y|X}(y \mid \tilde{x}) \tag{8}$$

3.1 Function F

Definition 1. *We define* **F** *as a function that takes tuple* $(p_X, p_{Y|X})$ *as input to produce posterior* $p_{X|Y}$*. In other words,*

$$\mathbf{F} : \mathcal{P}(\mathcal{X}) \times (\mathcal{P}(\mathcal{Y}))^{\mathcal{X}} \mapsto (\mathcal{P}(\mathcal{X}))^{\mathcal{Y}}$$

We also introduce the notation $\mathbf{F}(p_X, p_{Y|X})(y) = p_{X|Y}(\cdot|y) \in \mathcal{P}(\mathcal{X})$. The block diagram representation of function **F** is shown in Figure 2.

3.2 Function G

Definition 2. *We define* **G** *as a function that takes tuple* $(p_X, p_{Y|X})$ *as input to output* p_Y*. In other words,*

$$G : \mathcal{P}(\mathcal{X}) \times (\mathcal{P}(\mathcal{Y}))^{\mathcal{X}} \mapsto \mathcal{P}(\mathcal{Y}).$$

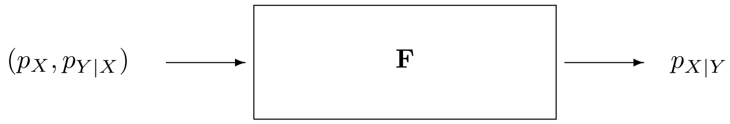

Figure 2: Block diagram representation for function **F**

The block diagram representation of function **G** is shown in Figure 3 .

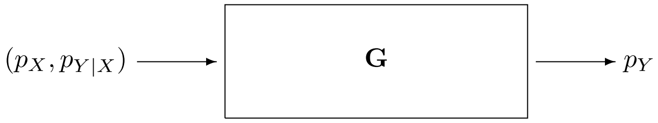

Figure 3: Block diagram representation for function **G**

4 Interpretation of forward recursion

$$\beta_{t+1}(x_{t+1}) = \sum_{x_t} p(x_t \mid y^t) p(x_{t+1} \mid x_t) \tag{9}$$

$$\alpha_t(x_t) = \frac{\beta_t(x_t) p(y_t \mid x_t)}{\sum_{x_t} \beta_t(x_t) p(y_t \mid x_t)} \tag{10}$$

- $\beta_t(x_t) = p(x_t \mid y^{t-1})$ is the posterior of the present state x_t given all the previous state observations, y^{t-1}. Equation (9) is called ***time update*** step.
- $\alpha_t(x_t) = p(x_t \mid y^t)$ posterior of the state x_t given all causal observations y^t. Equation (10) is called ***measurement update*** step.

Now, we would interpret the measurement and time update steps of the forward recursion algorithm in terms of the functions **F** and **G**.

4.1 Measurement update

Note from equation (10)

$$\alpha_t(x_t) = \frac{\beta_t(x_t) p(y_t \mid x_t)}{\sum_{x_t} \beta_t(x_t) p(y_t \mid x_t)}$$

We can construct a block diagram representation of equation (10) analogous to Figure 2 as shown in Figure 4.

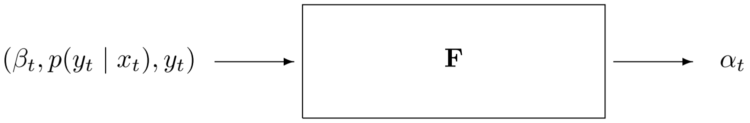

Figure 4: Interpretation of measurement update step in forward recursion

The above setting can now be viewed as if a random variable X_t with prior distribution β_t is passed through a known channel, i.e. known $p_{Y_t|X_t}$. We observe $Y_t = y_t$ as the output from this channel. We can identify these known quantities as of type same as inputs to function **F**. Also, we identify the output obtained by operating function **F** on $(\beta_t, p(y_t \mid x_t), y_t)$ as the posterior distribution of X_t given $Y_t = y_t$ which is same as the required quantity, $\alpha_t(x_t)$.

Thus, we can write

$$\alpha_t = \mathbf{F}(\beta_t, p_{Y_t|X_t})(y_t)$$

4.2 Time update

Note from equation (9), we have

$$\begin{aligned}\beta_{t+1}(x_{t+1}) &= \sum_{x_t} p(x_t \mid y^t) p(x_{t+1} \mid x_t) \\ &= \sum_{x_t} \alpha_t(x_t) p(x_{t+1} \mid x_t)\end{aligned}$$

We can now interpret the above equation for $\beta_{t+1}(x_{t+1})$ in terms of the setting of function **G** as described in Figure (5).

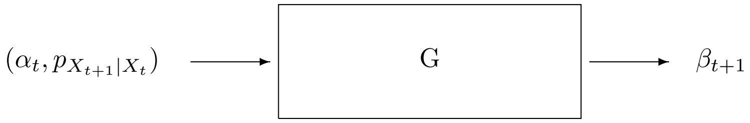

Figure 5: Interpretation of time update step in forward recursion

We can view α_t as the known prior distribution of a random variable which is applied to a known channel. The time update step then calculates the probability distribution of the output observed. Hence, we now write

$$\beta_{t+1} = \mathbf{G}(\alpha_t, p_{X_{t+1}|X_t})$$

5 Interpretation of backward recursion

In the previous lecture, we described that when we have all the observations in hand, the causality is not a requirement to estimate the underlying states. One such application is noisy image reconstruction. Backward recursion algorithm provides computationally efficient way to calculate the posterior of the underlying state at time t, x_t given all measurements y^n. In order to provide continuity, we revisit the notations and main points of the algorithm.

$$p(x_t \mid y^n) = \sum_{x_{t+1}} p(x_{t+1} \mid y^n) \frac{p(x_t \mid y^t) p(x_{t+1} \mid x_t, y^t)}{p(x_{t+1} \mid y^t)} \tag{11}$$

$$\gamma_t(x_t) = \sum_{x_{t+1}} \gamma_{t+1}(x_{t+1}) \frac{\alpha_t(x_t) p(x_{t+1} \mid x_t)}{\beta_{t+1}(x_{t+1})} \tag{12}$$

$$= \overbrace{\sum_{x_{t+1}} \gamma_{t+1}(x_{t+1}) \underbrace{\left[\frac{\alpha_t(x_t) p(x_{t+1} \mid x_t)}{\sum_{x_t} \alpha_t(x_t) p(x_{t+1} \mid x_t)} \right]}_{\mathbf{F}(\alpha_t, p_{X_{t+1}|X_t})}}^{\mathbf{G}(\gamma_{t+1}, \mathbf{F}(\alpha_t, p_{X_{t+1}|X_t}))} \tag{13}$$

Thus, we can write

$$\gamma_t = \mathbf{G}(\gamma_{t+1}, \mathbf{F}(\alpha_t, p_{X_{t+1}|X_t})) \tag{14}$$

The equation (14) can be interpreted in two steps:

1. A random variable X_t with known prior distribution α_t is passed through known channel, i.e. known $p_{X_{t+1}|X_t}$. The posterior of X_t given the channel output X_{t+1} can be calculated using function $\boldsymbol{F}$ interpretation as $p_{X_t|X_{t+1}} = \mathbf{F}(\alpha_t, p_{X_{t+1}|X_t})$.
2. Now, we can view another random variable with known prior distribution γ_{t+1} being passed through channel with known statistics, $\mathbf{F}(\alpha_t, p_{X_{t+1}|X_t})$. The pmf of output at value x_t, $\gamma_t(x_t)$ is then given by $\boldsymbol{G}$-interpretation, $\mathbf{G}(\gamma_{t+1}, \mathbf{F}(\alpha_t, p_{X_{t+1}|X_t}))$.

Thus, we see that the complex looking equation (14) can in fact be viewed as transformation of distributions α_t and γ_{t+1} into γ_t invoking the **F** and **G** function interpretations.

The interpretations given above using functions **F** and **G** are in general valid even if the states and observations are continuous but impractical in general(because we can not store the whole value of continuous function). However, if all variables are Gaussian and operations are linear, we can apply this. We will see that this property allows us to derive Kalman Filter as a special case of forward recursion in Gaussian state space setting.

6 Kalman Filter

If random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ are jointly Gaussian, then we write

$$(\boldsymbol{X}_{k\times 1}, \boldsymbol{Y}_{n\times 1}) \sim \mathcal{N}(\boldsymbol{\mu}_{(k+n)\times 1}, \Sigma_{(k+n)\times(k+n)})$$

where,

$$\boldsymbol{\mu}_{(k+n)\times 1} = \begin{pmatrix} \boldsymbol{\mu_X} \\ \boldsymbol{\mu_Y} \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}, \quad \Sigma_{YX} = \Sigma_{XY}^T$$

The matrix Σ_{XY} is also know as cross covariance matrix of $\boldsymbol{X}$ and $\boldsymbol{Y}$. Then, it follows from the Gauss–Markov theorem in Lecture 3 that the conditional distribution of $\boldsymbol{X}$ conditioned on $\{\boldsymbol{Y} = \boldsymbol{y}\}$ is given as

$$\boldsymbol{X} \mid \{\boldsymbol{Y} = \boldsymbol{y}\} \sim \mathcal{N}(\boldsymbol{\mu_X} + \Sigma_{XY}\Sigma_Y^{-1}(\boldsymbol{y} - \boldsymbol{\mu_Y}), \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}) \tag{15}$$

The following interpretations may be useful. Denote the conditional expectation $\mathbb{E}[X|Y]$ as $\hat{X}(Y)$, then we know

$$\hat{\boldsymbol{X}}(\boldsymbol{Y}) = \boldsymbol{\mu_X} + \Sigma_{XY}\Sigma_Y^{-1}(\boldsymbol{Y} - \boldsymbol{\mu_Y}).$$

The law of total variance gives us

$$\begin{aligned}\Sigma_X &= \mathbb{E}[\Sigma_{X|Y}] + \Sigma_{\mathbb{E}[X|Y]} \qquad (16)\\ &= \mathbb{E}[\Sigma_{X|Y}] + \Sigma_{\hat{X}}, \qquad (17)\end{aligned}$$

where

$$\Sigma_{X|Y} \triangleq \mathbb{E}[(X - \mathbb{E}[X|Y])(X - \mathbb{E}[X|Y])^T|Y],$$

which implies that

$$\begin{aligned}\mathbb{E}[\Sigma_{X|Y}] &= \mathbb{E}[(X - \mathbb{E}[X|Y])(X - \mathbb{E}[X|Y])^T]\\ &= \mathbb{E}[(X - \hat{X}(Y))(X - \hat{X}(Y))^T],\end{aligned}$$

which is equal to the covariance matrix of the error vector $X - \hat{X}(Y)$.

The matrix $\Sigma_{\hat{X}}$ satisfies

$$\begin{aligned}\Sigma_{\hat{X}} &\triangleq \mathbb{E}[(\hat{X} - \mathbb{E}[\hat{X}])(\hat{X} - \mathbb{E}[\hat{X}])^T]\\ &= \mathbb{E}[(\hat{X} - \mu_X)(\hat{X} - \mu_X)^T]\\ &= \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}\end{aligned}$$

as was shown in Lecture 2.

The interesting fact is that $\Sigma_{X|Y}$, which in general is random matrix, is in fact equal to a deterministic matrix almost surely:

$$\Sigma_{X|Y} = \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX}.$$

6.1 Affine transformation of Gaussian random vector

Consider the affine transformation of $\boldsymbol{X}$

$$\boldsymbol{Y} = \boldsymbol{AX} + \boldsymbol{N} \tag{18}$$

where we assume $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu_X}, \Sigma_X)$ and $\boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_N)$ are independent. We also assume that the dimensions of the involved matrices are consistent. Then following relations hold

$$\begin{aligned}\Sigma_{XY} &= \Sigma_X \boldsymbol{A^T} \qquad (19)\\ \Sigma_Y &= \boldsymbol{A}\Sigma_X\boldsymbol{A^T} + \Sigma_N \qquad (20)\\ \mu_Y &= \boldsymbol{A}\mu_X \qquad (21)\end{aligned}$$

From the above relations, we can write

$$\begin{aligned}E(\boldsymbol{X} \mid \boldsymbol{Y}) &= \boldsymbol{\mu_X} + \Sigma_X\boldsymbol{A^T}(\boldsymbol{A}\Sigma_X\boldsymbol{A^T} + \Sigma_N)^{-1}(\boldsymbol{Y} - \boldsymbol{A\mu_X})\\ &\triangleq \boldsymbol{F}_\mu(\boldsymbol{\mu_X}, \Sigma_X, \boldsymbol{A}, \Sigma_N, \boldsymbol{Y}) \qquad (22)\\ \Sigma_{X|Y} &= \Sigma_X - \Sigma_X\boldsymbol{A^T}(\boldsymbol{A}\Sigma_X\boldsymbol{A^T} + \Sigma_N)^{-1}\boldsymbol{A}\Sigma_X\\ &\triangleq \boldsymbol{F_\Sigma}(\Sigma_X, \boldsymbol{A}, \Sigma_N) \qquad (23)\end{aligned}$$

Here, $E(\boldsymbol{X} \mid \boldsymbol{Y})$ and $\Sigma_{X|Y}$ are conditional mean and covariance matrices of $\boldsymbol{X}$ given observed vector $\boldsymbol{Y}$. Now, $\boldsymbol{F_\mu}$ and $\boldsymbol{F_\Sigma}$ functions can be thought of as analogies of the generic function $\boldsymbol{F}$ defined previously. Here, we essentially have the same setting: **Gaussian** random vector $\boldsymbol{X}$ is passed through a known channel, $f_{\boldsymbol{Y}|\boldsymbol{X}}$. The functions $\boldsymbol{F_\mu}$ and $\boldsymbol{F_\Sigma}$ then completely describe the posterior $f_{\boldsymbol{X}|\boldsymbol{Y}}$. This is because the posterior distribution is also Gaussian and is completely characterised by the conditional mean and covariance matrix as in equation (15).

Also, we have

$$\begin{aligned}
\boldsymbol{Y} &\sim \mathcal{N}(\boldsymbol{\mu_Y}, \Sigma_Y) \\
\boldsymbol{\mu_Y} &= \boldsymbol{A\mu_X} \\
&\triangleq \boldsymbol{G_\mu(\mu_X, A)} && (24) \\
\Sigma_Y &= \boldsymbol{A}\Sigma_X \boldsymbol{A^T} + \Sigma_N \\
&\triangleq \boldsymbol{G_\Sigma(\Sigma_X, A, \Sigma_N)} && (25)
\end{aligned}$$

Here, we can interpret the affine transformation in a setting where **Gaussian** random vector $\boldsymbol{X}$ with prior distribution $f_{\boldsymbol{X}}$ is passed through a known channel, $f_{\boldsymbol{Y}|\boldsymbol{X}}$. The functions $\boldsymbol{G_\mu}$ and $\boldsymbol{G_\Sigma}$ as described in above equations then completely determine the distribution of the output vector $\boldsymbol{Y}$. This follows since the output is also Gaussian random vector, whose distribution is completely characterized by the mean and covariance matrices given by functions $\boldsymbol{G_\mu}$ and $\boldsymbol{G_\Sigma}$.

6.2 Kalman Filter

We now proceed to describe Kalman Filter. We assume the following setup of the state process and its observation.

$$\begin{aligned}
\text{state process :} \quad & \boldsymbol{X_{t+1}} = \boldsymbol{A_t X_t} + \boldsymbol{W_t} \quad \forall t \geq 1 && (26) \\
\text{observation process :} \quad & \boldsymbol{Y_t} = \boldsymbol{H_t X_t} + \boldsymbol{N_t} \quad \forall t \geq 1 && (27)
\end{aligned}$$

The dimensions of the vectors and matrices are given as below.

$\boldsymbol{X_t} \in \mathbb{R}^k$	: state process
$\boldsymbol{A_t} \in \mathbb{R}^{k \times k}$	: state transition matrix
$\boldsymbol{W_t} \in \mathbb{R}^k$	: process noise
$\boldsymbol{Y_t} \in \mathbb{R}^m$	: observed process
$\boldsymbol{H_t} \in \mathbb{R}^{m \times k}$	: output transition matrix
$\boldsymbol{N_t} \in \mathbb{R}^m$	: measurement noise

Assumptions of the model

1. $\boldsymbol{X_1 \sim \mathcal{N}(0, \Pi_1)}$
2. $\boldsymbol{W_t} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\boldsymbol{W_t}})$. Also, $\{\boldsymbol{W_t}\}_{t\geq1}$ are independent.
3. $\boldsymbol{N_t} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\boldsymbol{N_t}})$. Also, $\{\boldsymbol{N_t}\}_{t\geq1}$ are independent.
4. Process $\{\boldsymbol{W_t}\}_{t\geq1}$, $\{\boldsymbol{N_t}\}_{t\geq1}$, and $\boldsymbol{X_0}$ are mutually independent

5. Matrices $\boldsymbol{A_t}, \boldsymbol{H_t}, \Sigma_{\boldsymbol{W_t}}$ and $\Sigma_{\boldsymbol{N_t}}, \forall t \geq 1$, are deterministic and are known to the estimator.

Note

1. $\{\boldsymbol{X}_t\}$ is a Markov process
2. $\{\boldsymbol{Y}_t\}$ is related to $\{\boldsymbol{X}_t\}$ through a memoryless channel. It is an HMP!

The (forward recursion) Kalman Filter finds the linear minimum mean square error estimate of $\boldsymbol{X_t}$ given $\boldsymbol{Y^t}$ by applying the forward recursion algorithm tailored to the jointly Gaussian random vectors. This approach is justified as the process setup is an instance of the HMP with jointly Gaussian random vectors.

Due to Gaussianity, we can characterize the posterior probabilities using the first and second moments only. Thus the quantities α_t and β_t are totally determined by $(\boldsymbol{\mu}_{\boldsymbol{X_t|Y^t}}, \Sigma_{\boldsymbol{X_t|Y^t}})$ and $(\boldsymbol{\mu}_{\boldsymbol{X_t|Y^{t-1}}}, \Sigma_{\boldsymbol{X_t|Y^{t-1}}})$ respectively. Just like the forward recursion for the general HMP, Kalman Filter consists of measurement update and time update phase. We denote $\hat{X}_{i|j} = \boldsymbol{\mu}_{\boldsymbol{X_i|Y^j}}, P_{i|j} = \Sigma_{\boldsymbol{X_i|Y^j}}$.

6.2.1 Measurement Update

In general HMP, we have:

$$\alpha_t = \mathbf{F}(\beta_t, p_{Y_t|X_t})(Y_t) \tag{28}$$

In Kalman Filter, we express $\boldsymbol{\mu}_{\boldsymbol{X_t|Y^t}}$ and $\Sigma_{\boldsymbol{X_t|Y^t}}$ using $\boldsymbol{F_\mu}$ and $\boldsymbol{F_\Sigma}$.

$$\boldsymbol{\mu}_{\boldsymbol{X_t|Y^t}} = \boldsymbol{F_\mu}(\boldsymbol{\mu}_{\boldsymbol{X_t|Y^{t-1}}}, \Sigma_{\boldsymbol{X_t|Y^{t-1}}}, \boldsymbol{H_t}, \Sigma_{\boldsymbol{N_t}}, \boldsymbol{Y_t}) \tag{29}$$

$$\Sigma_{\boldsymbol{X_t|Y^t}} = \boldsymbol{F_\Sigma}(\Sigma_{\boldsymbol{X_t|Y^{t-1}}}, \boldsymbol{H_t}, \Sigma_{\boldsymbol{N_t}}) \tag{30}$$

Writing explicitly, if we define

$$K_{f,t} = P_{t|t-1}H_t^T(H_tP_{t|t-1}H_t^T + \Sigma_{N_t})^{-1}, \tag{31}$$

we have

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_{f,t}(Y_t - H_t\hat{X}_{t|t-1}) \tag{32}$$

$$P_{t|t} = P_{t|t-1} - K_{f,t}H_tP_{t|t-1}. \tag{33}$$

The quantity $K_{f,t}$ is called *filtered Kalman gain.*

6.2.2 Time Update

In general HMP, we have:

$$\beta_{t+1} = \mathbf{G}(\alpha_t, p_{X_{t+1}|X_t}) \tag{34}$$

In Kalman Filter, we express $\boldsymbol{\mu}_{\boldsymbol{X_{t+1}|Y^t}}$ and $\Sigma_{\boldsymbol{X_{t+1}|Y^t}}$ using $\boldsymbol{G_\mu}$ and $\boldsymbol{G_\Sigma}$.

$$\boldsymbol{\mu}_{\boldsymbol{X_{t+1}|Y^t}} = \boldsymbol{G_\mu}(\boldsymbol{\mu}_{\boldsymbol{X_t|Y^t}}, \boldsymbol{A_{t+1}}) \tag{35}$$

$$\Sigma_{\boldsymbol{X_{t+1}|Y^t}} = \boldsymbol{G_\Sigma}(\Sigma_{\boldsymbol{X_t|Y^t}}, \boldsymbol{A_{t+1}}, \Sigma_{\boldsymbol{W_t}}) \tag{36}$$

Writing explicitly, we have

$$\hat{X}_{t+1|t} = A_{t+1}\hat{X}_{t|t} \tag{37}$$

$$P_{t+1|t} = A_{t+1}P_{t|t}A_{t+1}^T + \Sigma_{W_t}. \tag{38}$$

Kalman Filter is initialized by assigning

$$(\boldsymbol{\mu}_{\boldsymbol{X_1}}, \Sigma_{\boldsymbol{X_1}}) = (\boldsymbol{0}, \boldsymbol{\Pi_1}) \tag{39}$$