10-708: Probabilistic Graphical Models 10-708, Spring 2014

11 : Factor Analysis and State Space Models

Lecturer: Eric P. Xing *Scribes: Dallas Card, Swabha Swayamdipta*

1. Overview

These notes present an overview of Factor Analysis and State Space Models. The factor analysis model is a simple latent variable model, where the latent variable is assumed to lie on a lower-dimensional linear subspace of the space of the observed variable. The graphical model for factor analysis is the same as a mixture model, except that both the observed and latent variables are assumed to be continuous. In particular, both the latent variable, and the (noisy) observations of that variable, are assumed to have Gaussian distributions, which makes for relatively simple estimation and inference. Factor Analysis is an old model, but is much more well understood today, thanks to the unifying framework of graphical models.

The State Space Model (SSM) can be seen as either a linear chain of factor analysis models, or a generalization of the Hidden Markov Model (HMM), in which the latent variables take on continuous, rather than discrete, values. The inference problem in both SSMs and HMMs is the same - calculating the probability of the latent variables given the observations. The model can be used for two types of inference - forward ("filtering") and backwards ("smoothing"). Starting with this model will allow us to build more complex models, such as a Switching SSM, where multiple layers of latent states are controlled by a master switching variable at each time point.

The notes are organized as follows: Section 2 provides the necessary mathematical background, Section 3 presents Factor Analysis, Section 4 presents the State Space Model in general, and Section 5 covers the details of Kalman filters.

2. Mathematical Background

In order to work with Gaussian distributions in the context of Factor Analysis and SSMs, it is useful to have a few mathematical tools in place.

2.1 Marginal and Conditional probabilities of Multivariate Gaussians

First, it is important to remember how a the marginal and conditional probabilities of a joint multivariate Gaussian can be written in terms of block elements. If we write the distribution of a multivariate Gaussian as:

$$p\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} | \mu, \Sigma\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} | \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right) \tag{1}$$

then we can write the marginal and conditional distributions of one of the components as:

$$p(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1 | \mu_1, \Sigma_{11}) \tag{2}$$

and

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1|\mathbf{m}_{1|2}, \mathbf{V}_{1|2}) \tag{3}$$

where

$$\mathbf{m}_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2) \tag{4}$$

and

$$\mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{5}$$

2.2 Matrix Inversion

It is also important to remember how to express matrix inversion in terms of the inverse of simpler block components of a matrix. In particular, if we consider a matrix:

$$M = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \tag{6}$$

We can write the inverse of this matrix as:

$$M^{-1} = \begin{bmatrix} E^{-1} + E^{-1}F(M/E)^{-1}GE^{-1} & -E^{-1}F(M/E)^{-1} \\ -(M/E)^{-1}GE^{-1} & (M/E)^{-1} \end{bmatrix} \tag{7}$$

where we have used the matrix inversion lemma:

$$(E - FH^{-1}G)^{-1} = E^{-1} + E^{-1}F(H - GE^{-1}F)^{-1}GE^{-1} \tag{8}$$

2.3 Matrix Algebra

Finally, it is useful to remember a few key formulas involving the trace and determinant.

$$tr[ABC] = tr[CAB] = tr[BCA] \tag{9}$$

$$\frac{\partial}{\partial A} tr[BA] = B^T \tag{10}$$

$$\frac{\partial}{\partial A} tr[x^T Ax] = \frac{\partial}{\partial A} tr[xx^T A] = xx^T \tag{11}$$

and

$$\frac{\partial}{\partial A} \log |A| = A^{-1} \tag{12}$$

3. Factor Analysis

3.1 The Factor Analysis Model

The Factor Analysis model can be thought of as an unsupervised linear regression model. In particular, it assumes that an unobserved variable, X, is generated from a Gaussian distribution over a linear subspace.

The observed variable, Y, is then generated from a Normal distribution conditioned on X, in a higher dimensional space. In other words, we have the following graphical model: $X \rightarrow Y$, where Y is observed. We begin with a marginal probability for X and a conditional probability for $Y|X$:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; 0, I)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mu + \Lambda\mathbf{x}, \Psi)$$

where Λ is called the *loading matrix*, and Ψ is a diagonal covariance matrix.

Geometrically, this model can be thought of in terms of generating a point, x, on a linear manifold, and then taking a noisy observation, y centred at x. This process is illustrated in the Figure 1 (from Michael Jordan's unpublished notes).

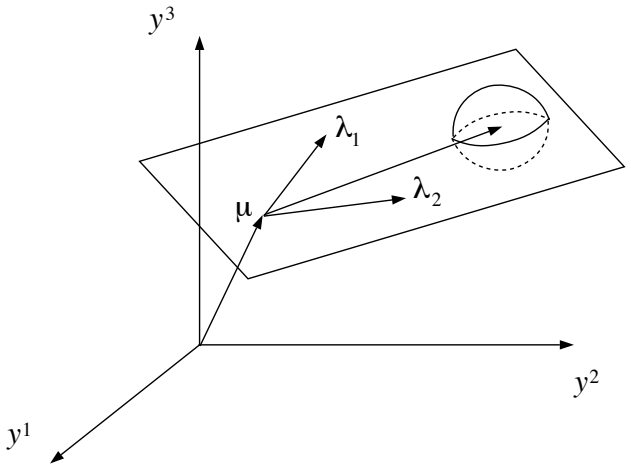

Figure 1: Dimensionality reduction with the Factor Analysis model

The conventional setting for this type of model is when we wish to project noisy observations into a lower dimensional space. Modern machine learning, however, is also reversing this, blowing low dimensional spaces up into high a higher dimensional space.

An advantage of this model is that since both X and $Y|X$ are Gaussian, all marginal, conditional, and joint distributions of interest will also be Gaussian. As a result, we can fully determine any of these distributions simply by computing its mean and variance. Using the notation developed above, we can represent the joint distribution of X and Y as:

$$p\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} | \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right)$$

We have already assumed values for μ_x and Σ_{xx}, so we can proceed to calculate the remaining quantities of interest (assuming the noise is uncorrelated with the data, i.e. $W \sim \mathcal{N}(0, \Psi)$):

$$\begin{aligned} \mu_y &= \mathbb{E}[Y] = \mathbb{E}[\mu + \Lambda X + W] \\ &= \mu + \Lambda\mathbb{E}[X] + \mathbb{E}[W] \\ &= \mu + \Lambda 0 + 0 = \mu \end{aligned}$$

$$\begin{aligned}\Sigma_{yy} = Var[Y] &= \mathbb{E}[(Y-\mu)(Y-\mu)^T] \\ &= \mathbb{E}[(\mu + \Lambda X + W - \mu)(\mu + \Lambda X + W - \mu)^T] \\ &= \mathbb{E}[(\Lambda X + W)(\Lambda X + W)^T] \\ &= \Lambda\mathbb{E}[XX^T]\Lambda^T + \mathbb{E}[WW^T] \\ &= \Lambda\Lambda^T + \Psi\end{aligned}$$

It is worth noting here that Y is the summation of two parts: a diagonal (covariance) matrix (Ψ), and the outer product of a tall skinny matrix with itself (Λ). Thus, although Y will be high-dimensional, it may actually have a low-rank structure.

Finally, we need the covariance between X and Y, which is given by:

$$\begin{aligned}\Sigma_{xy} = Cov[X,Y] &= \mathbb{E}[(X-0)(Y-\mu)^T] \\ &= \mathbb{E}[X(\mu + \Lambda X + W - \mu)^T] \\ &= \mathbb{E}[XX^T\Lambda^T + XW^T] = \Lambda^T\end{aligned}$$

Thus, the full joint distribution of X and Y can be written as:

$$p\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \mid \begin{bmatrix}0\\ \mu\end{bmatrix}, \begin{bmatrix}I & \Lambda^T \\ \Lambda & \Lambda\Lambda^T + \Psi\end{bmatrix}\right)$$

Using equations (4) and (5) presented above we can now easily compute the conditional probability $p(X|Y) = \mathcal{N}(X|\mathbf{m}_{x|y}, \mathbf{V}_{x|y})$. Note that we will use the matrix inversion lemma (8) here, because computing $(I + \Lambda^T\Psi^{-1}\Lambda)^{-1}$ will be much easier than computing $(\Lambda\Lambda^T + \Psi)^{-1}$. The computations follow directly from the equations presented above:

$$p(X|Y) = \mathcal{N}(X|\mathbf{m}_{x|y}, \mathbf{V}_{x|y})$$

$$\begin{aligned}\mathbf{V}_{x|y} &= \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} \\ &= I - \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\Lambda \\ &= (I + \Lambda^T\Psi^{-1}\Lambda)^{-1}\end{aligned}$$

$$\begin{aligned}\mathbf{m}_{x|y} &= \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(Y - \mu_y) \\ &= \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}(Y-\mu) \\ &= \left[(\Lambda\Lambda^T + \Psi)(\Lambda^T)^{-1}\right]^{-1}(Y-\mu) \\ &= \left[\Lambda + \Psi(\Lambda^T)^{-1}\right]^{-1}(Y-\mu) \\ &= \left[\Psi(\Psi^{-1}\Lambda + (\Lambda^T)^{-1})\right]^{-1}(Y-\mu) \\ &= \left[\Psi(\Lambda^T)^{-1}(\Lambda^T\Psi^{-1}\Lambda + I)\right]^{-1}(Y-\mu) \\ &= (I + \Lambda^T\Psi^{-1}\Lambda)^{-1}\Lambda^T\Psi^{-1}(Y-\mu) \\ &= \mathbf{V}_{x|y}\Lambda^T\Psi^{-1}(Y-\mu)\end{aligned}$$

Note that the posterior covariance does not depend on the observed data! Moreover, computing the posterior mean is just a linear operation. This is equivalent to projecting Y onto the subspace of X, a lower-dimensional subspace which is spanned by the loading matrix Λ. Since the move out of this subspace is assumed to be the result of an independent noise term, it thus makes sense that the posterior covariance does not depend on the observed data.

3.2 Learning Factor Analysis Models

We have now derived how to estimate $p(X|Y)$, but we still need to learn the parameters Λ, Ψ, and μ. We would like to solve for these via maximum likelihood estimation. Unfortunately, we have a latent variable, thus we need to resort to something like expectation-maximization (EM).

The incomplete log likelihood function is given by:

$$\begin{aligned} l(\theta, \mathcal{D}) &= -\frac{N}{2}\log|\Lambda\Lambda^T + \Psi| - \frac{1}{2}\sum_n (y_n - \mu)^T(\Lambda\Lambda^T + \Psi)^{-1}(y_n - \mu) \\ &= -\frac{N}{2}\log|\Lambda\Lambda^T + \Psi| - \frac{1}{2}tr[(\Lambda\Lambda^T + \Psi)^{-1}S] \end{aligned}$$

where $S = \sum_n (y_n - \mu)(y_n - \mu)^T$

Estimating μ is trivial:

$$\hat{\mu}_{ML} = \frac{1}{N}\sum_n y_n$$

The remaining variables, unfortunately, are coupled together in a highly non-linear way: $(\Lambda\Lambda^T + \Psi)^{-1}$. In this case, we cannot decouple the parameters. However, if we can pretend that everything is observed, that will give us something that is still coupled, but in a linear fashion. To simplify the derivation, we will assume that the data has been normalized, i.e. $y_i \leftarrow (y_i - \hat{\mu})$, such that $Y|x \sim \mathcal{N}(\Lambda x, \Psi)$.

The complete log-likelihood is given by:

$$\begin{aligned} l_c &= \sum_n \log p(x_n, y_n) = \sum_n \log p(x_n) + logp(y_n|x_n) \\ &= -\frac{N}{2}\log|I| - \frac{1}{2}\sum_n x_n^T x_n - \frac{N}{2}\log|\Psi| - \frac{1}{2}\sum_n (y_n - \Lambda x_n)^T\Psi^{-1}(y_n - \Lambda x_n) \\ &= -\frac{N}{2}\log|\Psi| - \frac{1}{2}\sum_n tr[x_n x_n^T] - \frac{N}{2}tr[S\Psi^{-1}] \end{aligned}$$

where $S = \frac{1}{N}\sum_n (y_n - \Lambda x_n)(y_n - \Lambda x_n)^T$

We can replace the unknown variables with their expectations (making use of the law of total variance, $Var(Y) = Var(\mathbb{E}(Y|X)) + \mathbb{E}(Var(Y|X))$, for $\langle X_n X_n^T\rangle$):

$$\langle S\rangle = \frac{1}{N}\sum_n (y_n y_n^T - y_n\langle X_n^T\rangle\Lambda^T - \Lambda\langle X_n^T\rangle y_n^T + \Lambda\langle X_n X_n^T\rangle\Lambda^T)$$

$$\langle X_n\rangle = \mathbb{E}[X_n|y_n]$$

$$\langle X_n X_n^T\rangle = Var[X_n|y_n] + \mathbb{E}[X_n|y_n]\mathbb{E}[X_n|y_n]^T$$

where $\langle X_n \rangle = \mathbf{m}_{x_n|y_n}$ and $\langle X_n X_n^T \rangle = \mathbf{V}_{X_n|Y_n} + \mathbf{m}_{x_n|y_n} \mathbf{m}_{x_n|y_n}^T$ are our sufficient statistics, as defined above. These estimates constitute the E-step of our EM algorithm.

For the M-step, we take partial derivatives of the complete log-likelihood function with respect to the two parameters of interest. After some algerba, we obtain the two update rules for the M-step:

$$\Psi^{t+1} = \langle S \rangle$$

and

$$\Lambda^{t+1} = \left(\sum_n y_n \langle X_n^T \rangle \right) \left(\sum_n \langle X_n X_n^T \rangle \right)^{-1}$$

Finally, we note that there is degeneracy in the Factor Analysis model. Since the loading matrix Λ only appers in an outer product with itself ($\Lambda\Lambda^T$), the model is invariant to rotation and flips of these basis vectors which define the latent manifold. In particular, we can replace Λ with ΛQ for any orthonormal matrix Q and the model remains the same: $(\Lambda Q)(\Lambda Q)^T = \Lambda(QQ^T)\Lambda^T = \Lambda\Lambda^T$.

This means that there is no one best setting for this parameter. If our purpose is to find a low-dimensional subspace to handle our data, then this suits our purpose. However, if our goal is to seek an implementation of the process which generated our data (such as using Latent Dirichlet Allocation to generate a lower-dimensional projection), this is not a very safe practice, because rotation can change the meaning. In general, these models are called *unidentifiable* since two people fitting parameters to identical data are not guaranteed to come up with the same values.

4. State Space Models

A State Space Model (SSM) is a dynamical generalization of the Factor Analysis model. In fact, it is a collection of factor analysers connected as a chain in the time domain, with one factor analyser model per time instance. SSMs are structurally identical to Hidden Markov Models - and hence have the same independence assumptions. The only difference is that the variables in SSM follow continuous (Gaussian) instead of discrete (Multinomial) distributions as in an HMM. As we will see, despite following continuous distributions, the derivation for inference under this model does not involve complex calculus, thanks to the properties of the Gaussian distribution.

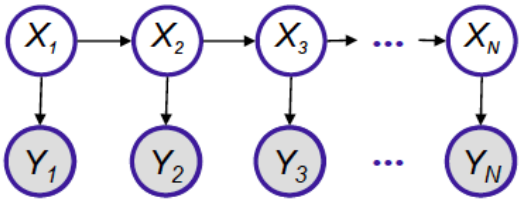

Figure 2: Graph for a State Space Model

In this model, we observe a sequence $y = (y_1, y_2, ..., y_t, ...)$ where each y_t is a continuous random variable for an instance of time, t. We assume there is a latent sequence $x = (x_1, x_2, ...x_t, ...)$ that generates this observation, where each x_t is also Gaussian. The graphical model is illustrated in Figure 2.

We introduce a transition matrix that determines the relationship between the latent variables, such that the mean of the state at time t, x_t is linear in the mean of the state at time $t-1$.

$$x_t = Ax_{t-1} + Gw_t$$

Here, $w_t = \mathcal{N}(0, Q)$ is the Gaussian noise we have introduced into the model. Since a linear combination of Gaussians is also Gaussian, x_t is Gaussian.

To describe the output, we use the Factor Analysis model at each point. The loading matrix, say C is shared across all x_t, y_t pairs. We assume that all the data points are in the same low-dimensional space. We have

$$y_t = Cx_t + v_t$$

where $v_t = \mathcal{N}(0, R)$ is some Gaussian noise. Note that we do not make any assumptions on the Q and R matrices, these could either be full rank or low rank.

Finally, we set the starting point, $x_0 = \mathcal{N}(0, \Sigma_0)$.

4.1. Application - LDS for 2D tracking

Unlike factor analyzers, SSMs are typically not used for dimensionality reduction. Below we describe an application for latent space inference.

Consider a point moving in 2D space. The true trajectory x is fully determined by the position and velocity of the particle (by Newton's law). Our observation of the trajectory, y, however, is limited to a noisy estimate of the truth. Thus x is the latent variable sequence corresponding to the true path. The true path is given by new position = old position + Δ(old velocity)+ noise, whereas our observations are given by observed position = true position + noise. Since the new position is a linear combination of the old position and the velocity, in practice, we can apply the SSM (in particular Kalman Filtering) to tracking the trajectory of a plane when we observe radar signals from it at different points in time.

4.2. Inference problems

The inference problem in this model is the same as that of the Factor Analysis model, i.e. how to compute $p(x|y)$ where y is an observed variable and x is a latent variable. However, now x is a sequence of random variables, $x_1, x_2, ...x_t, ...$ and similarly y is $y_1, y_2, ..., y_t, ...$, where t is an instance of time. This changes the inference problem slightly and we discuss the two variations here.

Filtering: Compute $p(x_t|y_{1:t})$

Smoothing: Compute $p(x_t|y_{1:T})$ where $t < T$

In our 2D tracking problem for planes using an SSM, we could attempt to infer the plane's true position at a given time based on a series of observations up to that time ("filtering"), or where the plane *was* at a previous point in time, based on previous and subsequent observations ("smoothing"). Filtering provides an acceptable estimate, but smoothing improves this estimate substantially as seen in the left and the right graphs of Figure 3. The circles represent the Gaussian distribution of the observed radar.

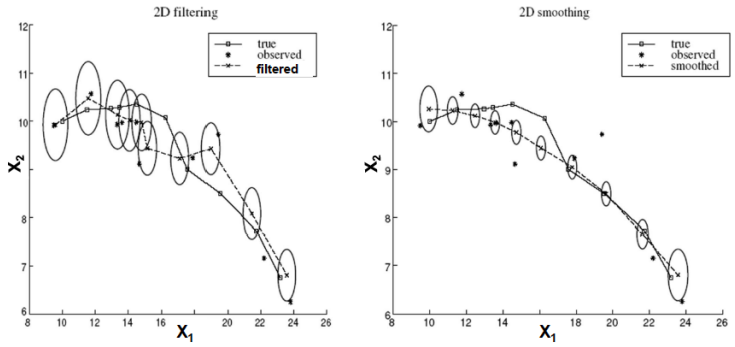

Figure 3: 2D tracking

5. Kalman Filtering

Given a sequence of observations $y_1, y_2, ..., y_t$, we have to infer the latent state at time t. This inference problem is also known as Kalman Filtering. Historically, Kalman Filtering used to be considered a stand-alone inference technique. However after graphical models gained popularity, it was clear that it is only a Gaussian analogue of the forward inference for HMMs (see Figure 4). The following equation shows this analogy:

$$p(x_t|y_{1:t}) = \alpha_t{}^i \propto p(y_t|x_t)\Sigma_{x_{t-1}} p(x_t|x_{t-1})\alpha^j{}_{t-1}$$

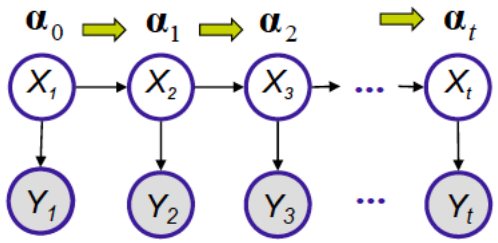

Figure 4: Kalman Filtering as Forward Inference on SSMs

Kalman Filtering for inference is widely applied in psychology. Our observation of the world through visual images, say y, is in two dimensional space even though the truth, x is in three dimensional space. This means y is a noisy version of the ground truth. However, when our brains recreate this observation, they do so in three-dimensional space. This happens dynamically in time as the brain cannot look forward in time before recreating the images. Hence, this is the Kalman Filtering estimation of $p(x_t|y_{1:t})$.

5.1 Kalman Filtering derivation

The key observation in the SSM is that every distribution in it is Gaussian. Therefore, the distribution of interest $p(x_t|y_{1:t})$ is also a Gaussian. The task is then to estimate the mean, $\mu_{1:t} = \mathbb{E}(x_t|y_{1:t})$ and the covariance, $P_{1:t} = \mathbb{E}(x_t - \mu_{1:t})^T(x_t - \mu_{1:t})$ of this distribution.

The estimation is done in two steps to simplify computation.

- Predict step - Compute $p(x_{t+1}|y_{1:t})$ from $p(x_t|y_{1:t})$. This is equivalent to moving one step ahead of the current observation sequence. It is also called time update.
- Update step - Update the prediction in the previous step by including the new evidence in the data. The new evidence is computed according to a new observation y_{t+1} and the model parameter $p(y_{t+1}|x_{t+1})$.

The high level idea behind the estimation is the following: We are given two Gaussian vectors z_1 and z_2, which are distributed as below:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \sim \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

We shall use z_1 to generate the joint z_1, z_2 and either marginalize the joint to obtain z_2 or use the marginal to compute the conditional distribution, $z_2|z_1$.

In the prediction step, we start with $p(x_t|y_{1:t})$ and use the transition information of the model, $x_t = Ax_{t-1} + Gw_t$ to compute $p(x_{t+1}|y_{1:t})$.

In the update step, we obtain $p(y_{t+1}|x_{t+1})$ using the emission information of the model, $y_t = Cx_t + v_t$. We use this evidence to obtain a joint $p(x_{t+1}, y_{t+1}|y_{1:t})$. Finally, we invert the model to obtain $p(x_{t+1}|y_{1:t+1})$.

5.1.1. Predict step

To calculate the mean, $\mu_{t+1|t}$ and the variance, $P_{t+1|t}$ of the joint distribution $p(x_{t+1}, y_{t+1}|y_{1:t})$ for the **Dynamical Model**, we proceed as below:

Mean:

$$\mathbb{E}(x_{t+1}|y_{1:t}) = \mathbb{E}(Ax_t + Gw_t) = A\mu_{1:t} + 0 = \mu_{1:t+1|t}$$

Covariance:

$$\begin{aligned} \mathbb{E}(x_{t+1} - \mu_{t+1|t})^T(x_{t+1} - \mu_{t+1|t}) &= \mathbb{E}(Ax_t + Gw_t - \mu_{t+1|t})^T(Ax_t + Gw_t - \mu_{t+1|t}) \\ &= A\mathbb{E}(x_t - \mu_{t+1|t})^T(x_t - \mu_{t+1|t})A^T + GQG^T \\ &= AP_{t+1|t}A^T + GQG^T \end{aligned}$$

To calculate the mean and the variance of the joint distribution $P(x_{t+1}, y_{t+1}|y_{1:t})$ for the **Observation Model**, we proceed as below:

Mean:

$$\mathbb{E}(y_{t+1}|y_{1:t}) = \mathbb{E}(Cx_{t+1} + v_{t+1}|y_{1:t}) = C\mu_{t+1|t}$$

Covariance:

$$\begin{aligned} \mathbb{E}[(y_{t+1} - \hat{y}_{t+1|t})(y_{t+1} - \hat{y}_{t+1|t})^T|y_{1:t}] &= CP_{t+1|t}C^T + R \\ \mathbb{E}[(y_{t+1} - \hat{y}_{t+1|t})(x_{t+1} - \mu_{t+1|t})^T|y_{1:t}] &= CP_{t+1|t} \end{aligned}$$

5.1.2. Update step

From the quantities computed in the previous step, we proceed here to compute the mean and the variance of the conditional distribution, $p(X_{t+1}|Y_{t+1})$ using the formulae for conditional Gaussian distributions. $\mu_{t|t}$ and $P_{t|t}$ are the mean and covariance respectively of the distribution $p(X_t|Y_t)$.

Time Updates:

$$\mu_{t+1|t} = A\mu_{t|t}$$
$$P_{t+1|t} = A^T P_{t|t} A + GQG^T$$

Measurement updates:

$$\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1}(y_{t+1} - C\mu_{t+1|t})$$
$$P_{t+1|t+1} = P_{t+1|t} - K_{t+1} C P_{t+1|t}$$

where $K_{t=1}$ is the Kalman gain. This quantity calibrates or adjusts the observed y_t such that our prediction for the next state is not biased. Kalman gain provides a trade-off between the prior and any new observation because in cases where either the prior is unreliable or the observations are noisy. The term $(y_{t+1} - C\mu_{t+1|t})$ is called the **innovation**, because it brings in additional information to the model.

Being independent of the data, the Kalman Gain can be precomputed using the following:

$$K_{t+1} = P_{t+1|t} C^T (C P_{t+1|t} C^T + R)^{-1}$$

5.2. 1D example

Consider noisy observations of a 1D particle doing a random walk. We have an initial estimate, which is a Gaussian centered at our best guess as to the true location, with some uncertainty as represented by the variance.

$$x_{t|t-1} = x_{t-1} + w, w \sim \mathcal{N}(0, \sigma_x)$$
$$y_t = x_t + v, v \sim \mathcal{N}(0, \sigma_y)$$

Note that both the transition matrix, A and the loading matrix, C are equal to the identity matrix I here.

$$P_{t+1|t} = A^T P_{t|t} A + GQG^T = \sigma_t + \sigma_x$$
$$\mu_{t+1|t} = A\mu_{t|t} = \mu_{t|t}$$
$$K_{t+1} = P_{t+1|t} C^T (C P_{t+1|t} C^T + R)^{-1} = (\sigma_t + \sigma_x)(\sigma_t + \sigma_x + \sigma_y)$$
$$\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1}(y_{t+1} - C\mu_{t+1|t}) = \frac{(\sigma_t + \sigma_x) y_t + \sigma_y \mu_{t|t}}{\sigma_t + \sigma_x + \sigma_y}$$
$$P_{t+1|t+1} = P_{t+1|t} - K_{t+1} C P_{t+1|t} = \frac{(\sigma_t + \sigma_x)\sigma_y}{\sigma_t + \sigma_x + \sigma_y}$$

After a time update, our estimate of the mean will not change, but the uncertainty (variance) will increase. After a measurement update, we will update our belief about the mean, based on new information, and decrease our uncertainty (variance).

5.3. Complexity

Let $x_t \in \Re^{N_x}$ and $y_t \in \Re^{N_y}$. Computing the new variance from the old variance takes $O(N_x^2)$ time:

$$P_{t+1|t} = A^T P_{t|t} A + GQG^T$$

Pre-computing the Kalman Gain takes $O(N_y^3)$ time:

$$K_{t+1} = P_{t+1|t} C^T (C P_{t+1|t} C^T + R)^{-1}$$

Hence, the overall time complexity is $\max(O(N_x^2), O(N_y^3))$. This makes Kalman Filtering quite an expensive inference algorithm for moderately high dimensional problems.

Kalman Filters are not popular these days due to high complexity. For instance, consider signals from 1000 aircraft coming at the same time. We need to consider all of them independent, and predicting each different trajectory is going to be highly expensive. In such cases, we should consider more complex models like Switching SSMs, which have multiple sequences of latent variables and a particular observation might depend on any combination of the latent sequences.

5.4 Smoothing

As described above, the smoothing problem is to estimate $x_t (t < T)$ given $y_1, \ldots, y_T$. This is the Gaussian analog of the backwards algorithm for HMMs. This process is known as the Rauch-Tung-Strievel smoother. The inference takes the following form:

$$p(X_t = i | y_{1:T}) = \gamma_t^i \propto \sum_j a\_t^i p(X_{t+1}^j | X_t^j) \gamma_{t+1}^j$$

We proceed in a manner similar to the Kalman Filtering process to get the estimates of the posterior distribution, $p(X_t = i | y_{1:T})$, the details of which are not covered in this document.

6. Conclusion

The above sections have presented an overview of Factor Analysis (FA) and State Space models (SSM), the analogues of Mixture and Hidden Markov Models (HMM) for the case where the latent variables take on continuous, rather than discrete, values. To summarize, Factor Analysis assumes a latent variable is generated from a Gaussian distribution in a linear subspace, and that a noisy observation is generated from a conditional Gaussian distribution in a higher-dimensional space, centered about the true value. SSMs are a dynamic generalization of FA models, in the form of a linear chain, where each latent state is a linear function of the previous state, plus an independent Gaussian noise term. As with HMMs, we can decompose the inference problem into a forward and a backward problem. The former is known as Kalman filtering,

and the later can be thought of as smoothing. Kalman filters are still a useful tool in time series analysis, and there is much to be explored beyond the scope of these notes.