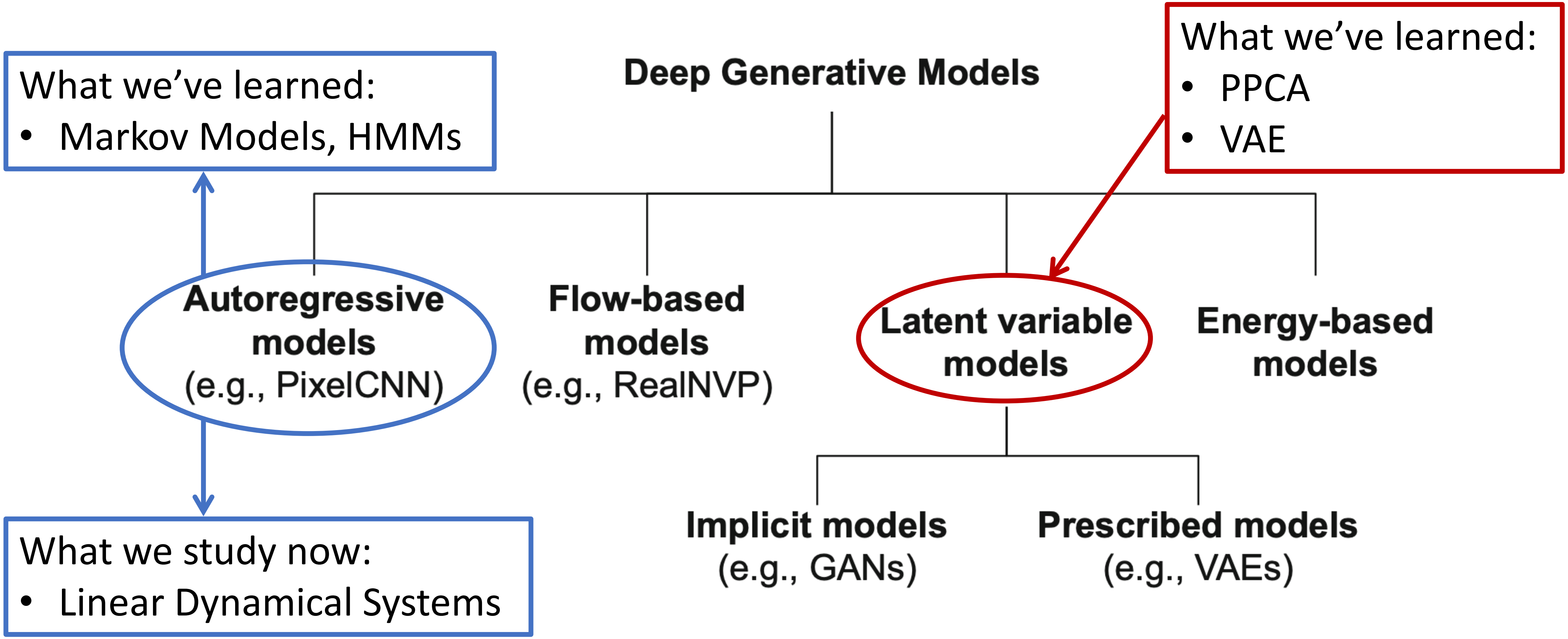
Deep Generative Models: Linear Dynamical Systems

Fall Semester 2025

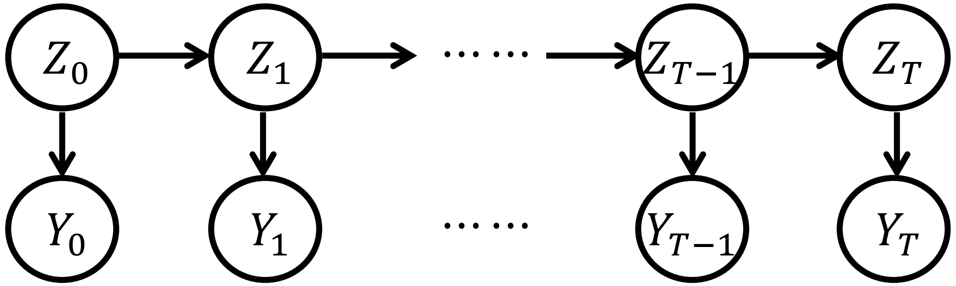
René Vidal
Director of the Center for Innovation in Data Engineering and Science (IDEAS),
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

Taxonomy of Generative Models



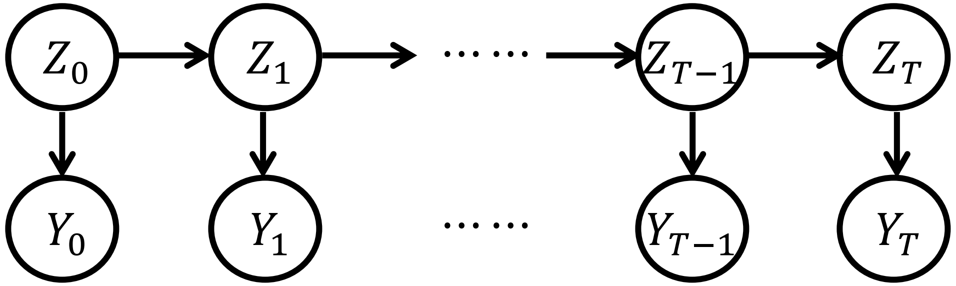
HMMs and Linear Dynamical Systems

Hidden Markov Models (HMMs)



- Hidden state Z_t and observation Y_t are discrete random scalar variables
- State transition and emission are discrete

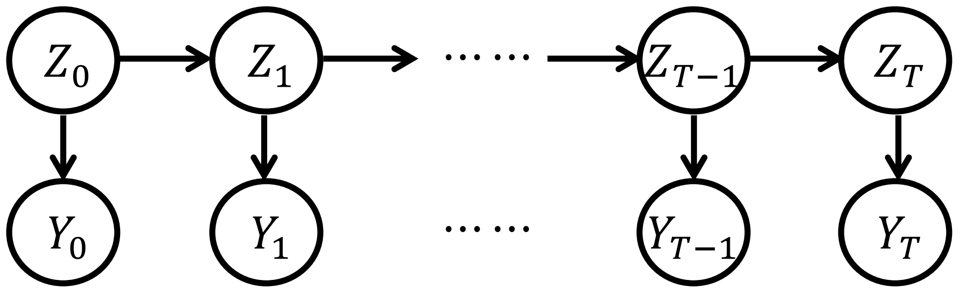
Linear Dynamical Systems (LDSs)



- Hidden state Z_t and observation Y_t are continuous (random) vectors
- State transition and emission are linear

Linear Dynamical Systems

Linear Dynamical Systems (LDSs)



- Hidden state Z_t and observation Y_t are continuous (random) vectors
- State transition and emission are linear

Model Parameters:

- $\theta \coloneqq (\pi_0, \Sigma_0, A, Q, C, R)$

d: state dimension
D: output dimension

$S_0 \in \mathbb{R}^{d \times d}$
$S_0 \succ 0$

$A \in \mathbb{R}^{d \times d}$
$C \in \mathbb{R}^{D \times d}$

$Q \in \mathbb{R}^{d \times d}$
$Q \succ 0$

$R \in \mathbb{R}^{D \times D}$
$R \succ 0$

- Initial Distribution:

$$\mathbb{P}(Z_0) = \mathcal{N}(\pi_0, \Sigma_0)$$

- State Transition:

$$\mathbb{P}(Z_t | z_{t-1}) = \mathcal{N}(A z_{t-1}, Q)$$

Why → this implies "Markov Property"

- State Emission:

$$\mathbb{P}(Y_t | z_t) = \mathcal{N}(C z_t, R)$$

Why → this implies "Output Independence"

Filtering and Smoothing

- P1: Filtering. Given θ and $(y_0, \ldots, y_t)$, infer the current state z_t, that is to compute
$$p_\theta(z_t | y_0, \ldots, y_t)$$
  - e.g., what is the current state of the missile given its position over some past time?
- P2: Smoothing. Given θ and $(y_0, \ldots, y_T)$, infer the past state z_t, that is to compute
$$p_\theta(z_t | y_0, \ldots, y_T)$$
  - e.g., where did the missile originate given we observed it over some time?
- Remark. You may find P1 and P2 familiar
  - In HMMs, we solved them via recursively updating $\alpha_j(t), \gamma_j(t)$

Background

- Before solving the filtering and smoothing problem, we will study (review) some basic properties about LDSs and Gaussian variables

Law of Total Expectation and of Total Variance

- We will heavily use the following basic results:

  - Law of Total Expectation (LoTE)

$$\mathbb{E}[x] = \mathbb{E}_y\big[\mathbb{E}_x[x|y]\big]$$

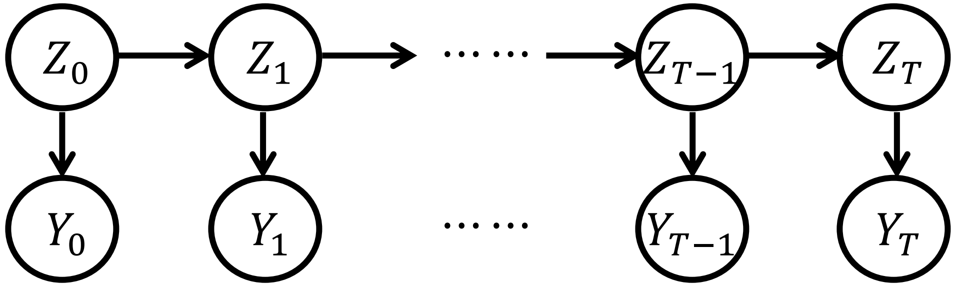
  - Law of Total Covariance (LoTC)

$$\mathrm{Cov}(x) = \mathbb{E}[\mathrm{Cov}(x|y)] + \mathrm{Cov}(\mathbb{E}[x|y])$$

$$\boxed{\begin{aligned}\mathrm{Cov}(x,y) &\coloneqq \mathbb{E}[(x-\mathbb{E}[x])(y-\mathbb{E}[y])^\top] \\ \mathrm{Cov}(x) &\coloneqq \mathrm{Cov}(x,x)\end{aligned}}$$

Basic Properties

Linear Dynamical Systems (LDSs)



- Hidden state Z_t and observation Y_t are continuous (random) vectors
- State transition and emission are linear

Model Parameters:
- $\theta \coloneqq (\pi_0, \Sigma_0, A, Q, C, R)$

d: state dimension
D: output dimension

$S_0 \in \mathbb{R}^{d\times d}$
$S_0 \succ 0$

$A \in \mathbb{R}^{d\times d}$
$C \in \mathbb{R}^{D\times d}$

$Q \in \mathbb{R}^{d\times d}$
$Q \succ 0$

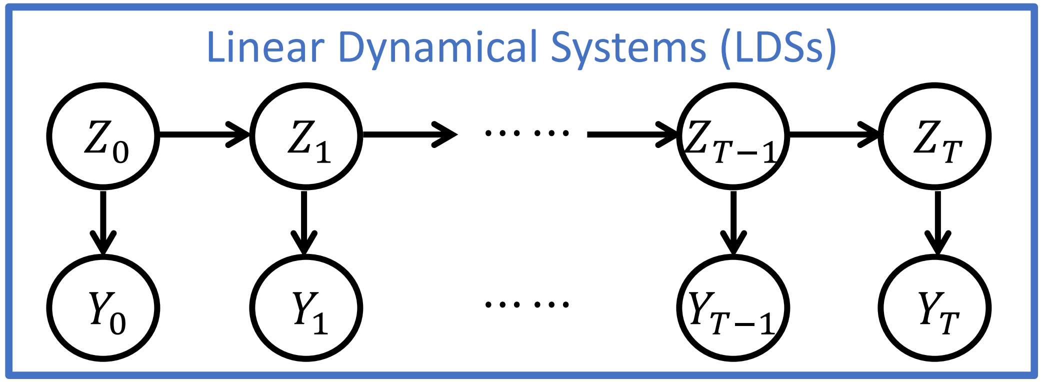
$R \in \mathbb{R}^{D\times D}$
$R \succ 0$

- Equivalent Descriptions:

	Probabilistic Description	Algebraic Description
State Transition	$\mathbb{P}(Z_t \mid z_{t-1}) = \mathcal{N}(Az_{t-1}, Q)$	$Z_t = Az_{t-1} + w_t$ with $w_t \sim \mathcal{N}(0, Q)$
State Emission	$\mathbb{P}(Y_t \mid z_t) = \mathcal{N}(Cz_t, R)$	$Y_t = Cz_t + v_t$ with $v_t \sim \mathcal{N}(0, R)$

- We assume w_t, v_t are independent from each other and independent from z_0

Basic Properties



- Hidden state Z_t and observation Y_t are continuous (random) vectors
- State transition and emission are linear

Model Parameters:
- $\theta \coloneqq (\pi_0, \Sigma_0, A, Q, C, R)$

d: state dimension
D: output dimension

$S_0 \in \mathbb{R}^{d \times d}$
$S_0 \succ 0$

$A \in \mathbb{R}^{d \times d}$
$C \in \mathbb{R}^{D \times d}$

$Q \in \mathbb{R}^{d \times d}$
$Q \succ 0$

$R \in \mathbb{R}^{D \times D}$
$R \succ 0$

- Joint distribution is Gaussian:

Gaussian

$$p_\theta(y_0, \dots, y_T, z_0, \dots, z_T) = \boxed{p_\theta(z_0)} \prod_{t=0}^{T} \boxed{p_\theta(y_t | z_t)} \prod_{t=1}^{T} \boxed{p_\theta(z_t | z_{t-1})}$$

- Therefore, “any conditional distribution of it” is Gaussian
  - Vague, but look at your question 1 of homework 1 (next page)

Gaussian Conditioning (Problem 1c & 1d, HW 1)

- If $\begin{bmatrix} a \\ b \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}\right)$, then the conditional distribution $p(a|b)$ is Gaussian with mean $\mu_{a|b}$ and covariance $\Sigma_{a|b}$ given by

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(b - \mu_b)$$

original mean & variance of a

correction upon observing b

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

Gaussian Combining (Extension of Problem 1b, HW 1)

Gaussian Combining: If $y = Cz + v$ with $z \sim \mathcal{N}(\mu_z, \Sigma_z)$ and $v \sim \mathcal{N}(0, \Sigma_v)$ then

$$\begin{bmatrix} z \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_z \\ C\mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_z & \Sigma_z C^\top \\ C\Sigma_z & C\Sigma_z C^\top + \Sigma_v \end{bmatrix}\right)$$

- Proof: The proof is finished by computing the following quantities:
  - $\mathbb{E}[y] = \mathbb{E}_z\left[\mathbb{E}_y[y|z]\right] = \mathbb{E}_z[Cz + v] = \mathbb{E}_z[Cz] = C\mu_z$
  - $\mathrm{Cov}(z, y) = \mathbb{E}[(z - \mu_z)(y - C\mu_z)^\top] = \mathbb{E}[(z - \mu_z)(Cz + v - C\mu_z)^\top] = \Sigma_z C^\top$
  - $\mathrm{Cov}(y) = \mathbb{E}[(y - C\mu_z)(y - C\mu_z)^\top] = \mathbb{E}_z\left[\mathbb{E}_y[(y - C\mu_z)(y - C\mu_z)^\top | z]\right]$

$$= \mathbb{E}_{z,v}[(Cz + v_t - C\mu_z)(Cz + v_t - C\mu_z)^\top]$$

$$= \mathbb{E}_z[(Cz - C\mu_z)(Cz - C\mu_z)^\top] + R$$

$$= C \cdot \mathbb{E}_z[(z - \mu_z)(z - \mu_z)^\top] \cdot C^\top + R = C\Sigma_z C^\top + R$$

Remark. In the proof, LoTE is used at the colored equality

Filtering and Smoothing

- P1: Filtering. Given θ and $(y_0, \ldots, y_t)$, compute
$$p_\theta(z_t | y_0, \ldots, y_t)$$

- P2: Smoothing. Given θ and $(y_0, \ldots, y_T)$, compute
$$p_\theta(z_t | y_0, \ldots, y_T)$$

- Since $p_\theta(z_s | y_0, \ldots, y_t)$ is Gaussian ($\forall s, t$), so it suffices to compute
$$\hat{z}_{s|t} := \mathbb{E}[z_s | y_0, \ldots, y_t]$$
$$\hat{\Sigma}_{s|t} := \mathbb{E}\left[\left(z_s - \hat{z}_{s|t}\right)\left(z_s - \hat{z}_{s|t}\right)^\top \middle| y_0, \ldots, y_t\right] = \mathrm{Cov}(z_s | y_0, \ldots, y_t)$$
  - and we will do so recursively (first for filtering and then for smoothing)

Filtering: Compute $\hat{z}_{0|0}, \hat{\Sigma}_{0|0}$

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$

- P1: Filtering. Given θ and $(y_0, \dots, y_t)$, compute

$$p_\theta(z_t | y_0, \dots, y_t)$$

- Let's begin with the simplest case:
  - What are the mean $\hat{z}_{0|0} = \mathbb{E}[z_0 | y_0]$ and covariance $\hat{\Sigma}_{0|0} = \mathrm{Cov}(z_0 | y_0)$ of z_0 given y_0?

- High-level Idea.
  1. find the mean and covariance of $\begin{bmatrix} z_0 \\ y_0 \end{bmatrix}$ via Gaussian combining
  2. find the mean $\hat{z}_{0|0}$ and covariance $\hat{\Sigma}_{0|0}$ of $z_0 | y_0$ via Gaussian conditioning

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(b - \mu_b), \ \Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

Filtering: Compute $\hat{z}_{0|0}, \hat{\Sigma}_{0|0}$

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$

$\mathbb{P}(Z_0) = \mathcal{N}(\pi_0, \Sigma_0)$
$\mathbb{P}(Y_t | z_t) = \mathcal{N}(C z_t, R)$

- Step 1: find the mean and covariance of $\begin{bmatrix} z_0 \\ y_0 \end{bmatrix}$

Gaussian Combining: If $y = Cz + v$ with $z \sim \mathcal{N}(\mu_z, \Sigma_z)$ and $v \sim \mathcal{N}(0, \Sigma_v)$ then

$$\begin{bmatrix} z \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_z \\ C\mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_z & \Sigma_z C^\top \\ C\Sigma_z & C\Sigma_z C^\top + \Sigma_v \end{bmatrix} \right)$$

- Applying Gaussian combining yields

$$\begin{bmatrix} z_0 \\ y_0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \pi_0 \\ C\pi_0 \end{bmatrix}, \begin{bmatrix} \Sigma_0 & \Sigma_0 C^\top \\ C\Sigma_0 & C\Sigma_0 C^\top + R \end{bmatrix} \right)$$

Filtering: Compute $\hat{z}_{0|0}$, $\hat{\Sigma}_{0|0}$

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$

- Step 2: apply Gaussian conditioning to $\begin{bmatrix} z_0 \\ y_0 \end{bmatrix}$

$$\begin{bmatrix} z_0 \\ y_0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \pi_0 \\ C\pi_0 \end{bmatrix}, \begin{bmatrix} \Sigma_0 & \Sigma_0 C^\top \\ C\Sigma_0 & C\Sigma_0 C^\top + R \end{bmatrix} \right)$$

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(b - \mu_b), \; \Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

- We have

$$\hat{z}_{0|0} = \pi_0 + \Sigma_0 C^\top (C\Sigma_0 C^\top + R)^{-1}(y_0 - C\pi_0)$$

$$\hat{\Sigma}_{0|0} = \Sigma_0 - \Sigma_0 C^\top (C\Sigma_0 C^\top + R)^{-1} C\Sigma_0$$

Filtering: From 0 to t

$$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$$
$$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$$

- P1: Filtering. Given θ and $(y_0, \dots, y_t)$, compute
$$p_\theta(z_t | y_0, \dots, y_t)$$

- We've now computed $\hat{z}_{0|0}$ and $\hat{\Sigma}_{0|0}$. This solves P1 for the case $t = 0$

- To proceed, we will update $\hat{z}_{t-1|t-1}, \hat{\Sigma}_{t-1|t-1}$ into $\hat{z}_{t|t}, \hat{\Sigma}_{t|t}$ for every t

The planned computational flow

$$\begin{array}{cccc}
\hat{z}_{0|0} & \hat{z}_{0|1} & \hat{z}_{0|2} & \hat{z}_{0|3} \\
\hat{z}_{1|0} & \hat{z}_{1|1} & \hat{z}_{1|2} & \hat{z}_{1|3} \\
\hat{z}_{2|0} & \hat{z}_{2|1} & \hat{z}_{2|2} & \hat{z}_{2|3} \\
\hat{z}_{3|0} & \hat{z}_{3|1} & \hat{z}_{3|2} & \hat{z}_{3|3}
\end{array}$$

Filtering: From 0 to t

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$

- To compute $\hat{z}_{0|0}$ and $\hat{\Sigma}_{0|0}$, we
  - (Step 0) found the mean and covariance of z_0 (already known)
  - (Step 1) found the mean and covariance of $\begin{bmatrix} z_0 \\ y_0 \end{bmatrix}$ via Gaussian combining
  - (Step 2) found the mean and covariance of $z_0 | y_0$ via Gaussian conditioning

Question: How can we generalize these steps for general t?

- To update $\hat{z}_{t-1|t-1}, \hat{\Sigma}_{t-1|t-1}$ into $\hat{z}_{t|t}, \hat{\Sigma}_{t|t}$, we will condition on $y_0, \dots, y_{t-1}$ and
  - (Step 0) find the mean and covariance of $z_t | y_0, \dots, y_{t-1}$ (using $\hat{z}_{t-1|t-1}, \hat{\Sigma}_{t-1|t-1}$)
  - (Step 1) find the mean and covariance of $\begin{bmatrix} z_t \\ y_t \end{bmatrix} | y_0, \dots, y_{t-1}$ via Gaussian combining
  - (Step 2) find the mean and covariance of $z_t | y_t, y_0, \dots, y_{t-1}$ via Gaussian conditioning

Step 0 and conditioning on $y_0, \dots, y_{t-1}$ are the only differences
- You should be able to figure out all the details without looking at the rest slides

Filtering: Compute $\hat{z}_{t|t}, \hat{\Sigma}_{t|t}$

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$

$\mathbb{P}(Z_t | z_{t-1}) = \mathcal{N}(Az_{t-1}, Q)$

- Step 0: find the mean and covariance of $z_t | y_0, \dots, y_{t-1}$
  - By definition, this is to compute $\hat{z}_{t|t-1}, \hat{\Sigma}_{t|t-1}$

- We have

$$\hat{z}_{t|t-1} = \mathbb{E}[z_t | y_0, \dots, y_{t-1}] = \mathbb{E}_{z_{t-1}}\left[\mathbb{E}_{z_t}[z_t | z_{t-1}, y_0, \dots, y_{t-1}]\right]$$
$$= \mathbb{E}[Az_{t-1} | y_0, \dots, y_{t-1}] = A\hat{z}_{t-1|t-1}$$

$$\hat{\Sigma}_{t|t-1} = \mathbb{E}\left[\left(z_t - \hat{z}_{t|t-1}\right)\left(z_t - \hat{z}_{t|t-1}\right)^\top \middle| y_0, \dots, y_{t-1}\right] = \cdots = A\hat{\Sigma}_{t-1|t-1}A^\top + Q$$

similar to how we computed $\mathrm{Cov}(y)$ in the proof of Gaussian combining

Filtering: Compute $\hat{z}_{t|t}, \hat{\Sigma}_{t|t}$

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$

$\hat{z}_{t|t-1} = A\hat{z}_{t-1|t-1}$
$\hat{\Sigma}_{t|t-1} = A\hat{\Sigma}_{t-1|t-1}A^\top + Q$

$\hat{z}_{0|-1} \coloneqq \pi_0, \ \hat{\Sigma}_{0|-1} \coloneqq \Sigma_0$

- Step 1: find the mean and covariance of $\begin{bmatrix} z_t \\ y_t \end{bmatrix} | y_0, \dots, y_{t-1}$

- We applied Gaussian combining to $\begin{bmatrix} z_0 \\ y_0 \end{bmatrix}$ and obtained
  - $\begin{bmatrix} z_0 \\ y_0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{z}_{0|-1} \\ C\hat{z}_{0|-1} \end{bmatrix}, \begin{bmatrix} \hat{\Sigma}_{0|-1} & \hat{\Sigma}_{0|-1}C^\top \\ C\hat{\Sigma}_{0|-1} & C\hat{\Sigma}_{0|-1}C^\top + R \end{bmatrix} \right)$

- Similarly, now, applying Gaussian combining to $\begin{bmatrix} z_t \\ y_t \end{bmatrix} | y_0, \dots, y_{t-1}$ gives:
  - $\begin{bmatrix} z_t \\ y_t \end{bmatrix} | y_0, \dots, y_{t-1} \sim \mathcal{N}\left(\begin{bmatrix} \hat{z}_{t|t-1} \\ C\hat{z}_{t|t-1} \end{bmatrix}, \begin{bmatrix} \hat{\Sigma}_{t|t-1} & \hat{\Sigma}_{t|t-1}C^\top \\ C\hat{\Sigma}_{t|t-1} & C\hat{\Sigma}_{t|t-1}C^\top + R \end{bmatrix} \right)$

Filtering: Compute $\hat{z}_{t|t}, \hat{\Sigma}_{t|t}$

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$

$\hat{z}_{t|t-1} = A\hat{z}_{t-1|t-1}$
$\hat{\Sigma}_{t|t-1} = A\hat{\Sigma}_{t-1|t-1}A^\top + Q$

$\hat{z}_{0|-1} \coloneqq \pi_0, \ \hat{\Sigma}_{0|-1} \coloneqq \Sigma_0$

- Step 2: apply Gaussian conditioning to $\begin{bmatrix} z_t \\ y_t \end{bmatrix} | y_0, \dots, y_{t-1}$

- We applied Gaussian conditioning to $\begin{bmatrix} z_0 \\ y_0 \end{bmatrix}$ and obtained:

$$\hat{z}_{0|0} = \hat{z}_{0|-1} + \hat{\Sigma}_{0|-1} C^\top \left(C \hat{\Sigma}_{0|-1} C^\top + R \right)^{-1} (y_0 - C \hat{z}_{0|-1})$$
$$\hat{\Sigma}_{0|0} = \hat{\Sigma}_{0|-1} - \hat{\Sigma}_{0|-1} C^\top \left(C \hat{\Sigma}_{0|-1} C^\top + R \right)^{-1} C \hat{\Sigma}_{0|-1}$$

- Similarly, now, applying Gaussian conditioning to $\begin{bmatrix} z_t \\ y_t \end{bmatrix} | y_0, \dots, y_{t-1}$ gives:

$$\hat{z}_{t|t} = \hat{z}_{t|t-1} + \hat{\Sigma}_{t|t-1} C^\top \left(C \hat{\Sigma}_{t|t-1} C^\top + R \right)^{-1} (y_0 - C \hat{z}_{t|t-1})$$
$$\hat{\Sigma}_{t|t} = \hat{\Sigma}_{t|t-1} - \boxed{\hat{\Sigma}_{t|t-1} C^\top \left(C \hat{\Sigma}_{t|t-1} C^\top + R \right)^{-1}} C \hat{\Sigma}_{t|t-1}$$

"Kalman gain matrix". Let us denote it by K_t

Summary: Filtering for LDSs

$$\hat{z}_{s|t} := \mathbb{E}[z_s | y_0, \dots, y_t]$$
$$\hat{\Sigma}_{s|t} := \mathrm{Cov}(z_s | y_0, \dots, y_t)$$

$$\mathbb{P}(Z_0) = \mathcal{N}(\pi_0, \Sigma_0)$$
$$\mathbb{P}(Z_t | z_{t-1}) = \mathcal{N}(A z_{t-1}, Q)$$
$$\mathbb{P}(Y_t | z_t) = \mathcal{N}(C z_t, R)$$

- Putting everything together gives Kalman Filter:

- Initialization: $\hat{z}_{0|-1} := \pi_0, \ \hat{\Sigma}_{0|-1} := \Sigma_0$
- Recursion ($\forall t = 0, \dots, T$):

“Correction”:

$$K_t = \hat{\Sigma}_{t|t-1} C^\top \left(C \hat{\Sigma}_{t|t-1} C^\top + R \right)^{-1}$$
$$\hat{z}_{t|t} = \hat{z}_{t|t-1} + K_t (y_0 - C \hat{z}_{t|t-1})$$
$$\hat{\Sigma}_{t|t} = \hat{\Sigma}_{t|t-1} - K_t C \hat{\Sigma}_{t|t-1}$$

“Prediction”:

$$\hat{z}_{t+1|t} = A \hat{z}_{t|t}$$
$$\hat{\Sigma}_{t+1|t} = A \hat{\Sigma}_{t|t} A^\top + Q$$

How To Derive Kalman Filter

$$\begin{tikzcd}
p(z_{t-1} | y_0, \dots, y_{t-1}) \arrow[d, "\text{Prediction}"] \\
p(z_t | y_0, \dots, y_{t-1}) \arrow[d, "\text{Gaussian Combining}"] \arrow[ddd, dotted, bend left, "\text{Correction}"] \\
p(z_t, y_t | y_0, \dots, y_{t-1}) \arrow[d, "\text{Gaussian Conditioning}"] \\
p(z_t | y_0, \dots, y_{t-1}, y_t)
\end{tikzcd}$$

Computational Trajectory of Kalman Filter

$$\begin{tikzcd}
\hat{z}_{0|0} \arrow[d] & \hat{z}_{0|1} & \hat{z}_{0|2} & \hat{z}_{0|3} \\
\hat{z}_{1|0} \arrow[r] & \hat{z}_{1|1} \arrow[d] & \hat{z}_{1|2} & \hat{z}_{1|3} \\
\hat{z}_{2|0} & \hat{z}_{2|1} \arrow[r] & \hat{z}_{2|2} \arrow[d] & \hat{z}_{2|3} \\
\hat{z}_{3|0} & \hat{z}_{3|1} & \hat{z}_{3|2} \arrow[r] & \hat{z}_{3|3}
\end{tikzcd}$$

From Filtering to Smoothing

$\hat{z}_{s|t} := \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} := \mathrm{Cov}(z_s | y_0, \dots, y_t)$

- P1: Filtering. Given θ and $(y_0, \dots, y_t)$, compute
$$p_\theta(z_t | y_0, \dots, y_t)$$
- P2: Smoothing. Given θ and $(y_0, \dots, y_T)$, compute
$$p_\theta(z_t | y_0, \dots, y_T)$$

- Since everything is Gaussian, to solve P2 it suffices to compute $\hat{z}_{t|T}, \hat{\Sigma}_{t|T}$ for all t

What Kalman Filter gives us:

$$\begin{tikzcd}
\hat{z}_{0|0} \arrow[d] & \hat{z}_{0|1} & \hat{z}_{0|2} & \hat{z}_{0|3} \\
\hat{z}_{1|0} \arrow[r] & \hat{z}_{1|1} \arrow[d] & \hat{z}_{1|2} & \hat{z}_{1|3} \\
\hat{z}_{2|0} & \hat{z}_{2|1} \arrow[r] & \hat{z}_{2|2} \arrow[d] & \hat{z}_{2|3} \\
\hat{z}_{3|0} & \hat{z}_{3|1} & \hat{z}_{3|2} \arrow[r] & \hat{z}_{3|3}
\end{tikzcd}$$

What we will do to solve P2:

$$\begin{tikzcd}
\hat{z}_{0|0} \arrow[d] & \hat{z}_{0|1} & \hat{z}_{0|2} & \hat{z}_{0|3} \\
\hat{z}_{1|0} \arrow[r] & \hat{z}_{1|1} \arrow[d] & \hat{z}_{1|2} & \hat{z}_{1|3} \arrow[u] \\
\hat{z}_{2|0} & \hat{z}_{2|1} \arrow[r] & \hat{z}_{2|2} \arrow[d] & \hat{z}_{2|3} \arrow[u] \\
\hat{z}_{3|0} & \hat{z}_{3|1} & \hat{z}_{3|2} \arrow[r] & \hat{z}_{3|3} \arrow[u]
\end{tikzcd}$$

Goal: Given $\hat{z}_{t|t}, \hat{\Sigma}_{t|t}$ and $\hat{z}_{t|t-1}, \hat{\Sigma}_{t|t-1}$ for every t, update $\hat{z}_{t|T}, \hat{\Sigma}_{t|T}$ into $\hat{z}_{t-1|T}, \hat{\Sigma}_{t-1|T}$

Smoothing

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s|y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s|y_0, \dots, y_t)$

Goal: Given $\hat{z}_{t|t}$, $\hat{\Sigma}_{t|t}$ and $\hat{z}_{t|t-1}$, $\hat{\Sigma}_{t|t-1}$ for every t, update $\hat{z}_{t|T}$, $\hat{\Sigma}_{t|T}$ into $\hat{z}_{t-1|T}$, $\hat{\Sigma}_{t-1|T}$

- Observation:
  - Since z_{t-1} is independent of $y_t, \dots y_T$ given z_t, we have

$$p_\theta(z_{t-1}|z_t, y_0, \dots, y_{t-1}) = p_\theta(z_{t-1}|z_t, y_0, \dots, y_T)$$

- High-level Idea:
  1. Compute $p_\theta(z_{t-1}|z_t, y_0, \dots, y_{t-1})$ via Gaussian combining and Gaussian conditioning
     - This gives us $p_\theta(z_{t-1}|z_t, y_0, \dots, y_T)$
  2. Given $p_\theta(z_{t-1}|z_t, y_0, \dots, y_T)$, update $\hat{z}_{t|T}$, $\hat{\Sigma}_{t|T}$ into $\hat{z}_{t-1|T}$, $\hat{\Sigma}_{t-1|T}$ via LoTE and LoTC

Smoothing

$\mathbb{P}(Z_t|z_{t-1}) = \mathcal{N}(Az_{t-1}, Q)$

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s|y_0,\dots,y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s|y_0,\dots,y_t)$

- Step 1: Compute $p_\theta(z_{t-1}|z_t, y_0, \dots, y_{t-1})$

Gaussian Combining: If $y = Cz + v$ with $z \sim \mathcal{N}(\mu_z, \Sigma_z)$ and $v \sim \mathcal{N}(0, \Sigma_v)$ then

$$\begin{bmatrix} z \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_z \\ C\mu_z \end{bmatrix}, \begin{bmatrix} \Sigma_z & \Sigma_z C^\top \\ C\Sigma_z & C\Sigma_z C^\top + \Sigma_v \end{bmatrix}\right)$$

Gaussian Conditioning: $\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(b - \mu_b),\ \Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$

  - Step 1.1: Applying Gaussian combining to $\begin{bmatrix} z_{t-1} \\ z_t \end{bmatrix} | y_0, \dots, y_{t-1}$ gives:

$$\begin{bmatrix} z_{t-1} \\ z_t \end{bmatrix} | y_0, \dots, y_{t-1} \sim \mathcal{N}\left(\begin{bmatrix} \hat{z}_{t-1|t-1} \\ \hat{z}_{t|t-1} \end{bmatrix}, \begin{bmatrix} \hat{\Sigma}_{t-1|t-1} & \hat{\Sigma}_{t-1|t-1}A^\top \\ A\hat{\Sigma}_{t-1|t-1} & \hat{\Sigma}_{t|t-1} \end{bmatrix}\right)$$

$\hat{\Sigma}_{t|t-1} = A\hat{\Sigma}_{t-1|t-1}A^\top + Q$

  - Step 1.2: From Gaussian conditioning we see $z_{t-1}|z_t, y_0, \dots, y_{t-1}$ has distribution:

$$\mathcal{N}\left(\hat{z}_{t-1|t-1} + L_{t-1}(z_t - \hat{z}_{t|t-1}), \hat{\Sigma}_{t-1|t-1} - L_{t-1}A\hat{\Sigma}_{t-1|t-1}\right)$$

$L_{t-1} \coloneqq \hat{\Sigma}_{t-1|t-1}A^\top\hat{\Sigma}_{t|t-1}^{-1}$

Remark. Note that $L_{t-1}A\hat{\Sigma}_{t-1|t-1} = L_{t-1}\hat{\Sigma}_{t|t-1}L_{t-1}^\top$, so the covariance $\hat{\Sigma}_{t-1|t-1} - L_{t-1}A\hat{\Sigma}_{t-1|t-1}$ is symmetric

Smoothing

$$L_{t-1} = \hat{\Sigma}_{t-1|t-1} A^\top \hat{\Sigma}_{t|t-1}^{-1}$$

$$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \ldots, y_t]$$
$$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \ldots, y_t)$$

- We have obtained

$$p_\theta(z_{t-1} | z_t, y_0, \ldots, y_T) = \mathcal{N}\left(\hat{z}_{t-1|t-1} + L_{t-1}(z_t - \hat{z}_{t|t-1}), \hat{\Sigma}_{t-1|t-1} - L_{t-1} \hat{\Sigma}_{t|t-1} L_{t-1}^\top\right)$$

- Step 2: update $\hat{z}_{t|T}$, $\hat{\Sigma}_{t|T}$ into $\hat{z}_{t-1|T}$, $\hat{\Sigma}_{t-1|T}$ via LoTE and LoTC

$$\begin{aligned}
\hat{z}_{t-1|T} &= \mathbb{E}[z_{t-1} | y_0, \ldots, y_T] = \mathbb{E}_{z_t}\left[\mathbb{E}_{z_{t-1}}[z_{t-1} | z_t, y_0, \ldots, y_T]\right] \\
&= \mathbb{E}_{z_t}\left[\hat{z}_{t-1|t-1} + L_{t-1}(z_t - \hat{z}_{t|t-1}) | y_0, \ldots, y_T\right] \\
&= \hat{z}_{t-1|t-1} + L_{t-1}\left(\hat{z}_{t|T} - \hat{z}_{t|t-1}\right)
\end{aligned}$$

$$\begin{aligned}
\hat{\Sigma}_{t-1|T} &= \mathrm{Cov}(z_{t-1} | y_0, \ldots, y_T) = \mathbb{E}[\mathrm{Cov}(z_{t-1} | z_t, y_0, \ldots, y_T)] + \mathrm{Cov}(\mathbb{E}[z_{t-1} | z_t, y_0, \ldots, y_T]) \\
&= \mathbb{E}\left[\hat{\Sigma}_{t-1|t-1} - L_{t-1} \hat{\Sigma}_{t|t-1} L_{t-1}^\top\right] + \mathrm{Cov}\left(\hat{z}_{t-1|t-1} + L_{t-1}(z_t - \hat{z}_{t|t-1})\right) \\
&= \hat{\Sigma}_{t-1|t-1} - L_{t-1} \hat{\Sigma}_{t|t-1} L_{t-1}^\top + \mathrm{Cov}\left(\hat{z}_{t-1|t-1} + L_{t-1}(z_t - \hat{z}_{t|t-1}) | y_0, \ldots, y_T\right) \\
&= \hat{\Sigma}_{t-1|t-1} + L_{t-1}(\hat{\Sigma}_{t|T} - \hat{\Sigma}_{t|t-1}) L_{t-1}^\top
\end{aligned}$$

Summary: Smoothing for LDSs

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s|y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s|y_0, \dots, y_t)$

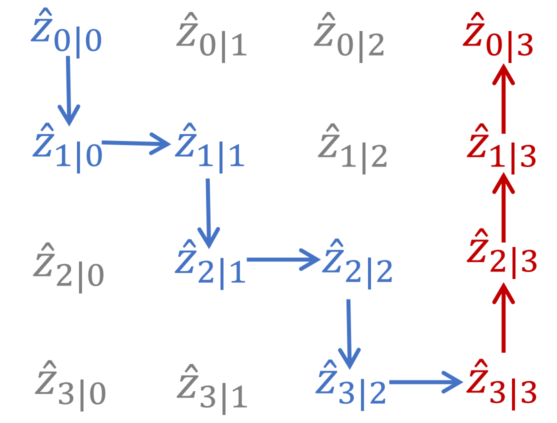
- **Goal**. Given θ and $(y_0, \dots, y_T)$, compute

$$p_\theta(z_t|y_0, \dots, y_T)$$

- **Algorithm** (known as "Rauch-Tung-Striebel smoother").
  1. (Forward Pass) Run Kalman filtering to compute $\hat{z}_{t|t}, \hat{\Sigma}_{t|t}$ and $\hat{z}_{t+1|t}, \hat{\Sigma}_{t+1|t}$ for all t
  2. (Backward Pass) For $t = T, \dots, 1$, compute the following:
     - $L_{t-1} = \hat{\Sigma}_{t-1|t-1} A^\top \hat{\Sigma}_{t|t-1}^{-1}$
     - $\hat{z}_{t-1|T} = \hat{z}_{t-1|t-1} + L_{t-1}(\hat{z}_{t|T} - \hat{z}_{t|t-1})$
     - $\hat{\Sigma}_{t-1|T} = \hat{\Sigma}_{t-1|t-1} + L_{t-1}(\hat{\Sigma}_{t|T} - \hat{\Sigma}_{t|t-1})L_{t-1}^\top$

Remark: L_{t-1} might also be computed in the forward pass

Computational Trajectory of Smoothing:



State Estimation and Learning

- Now that we've studied algorithms for filtering and smoothing, we are prepared to perform more complicated tasks

- State Estimation ("Decoding"). Given θ and $(y_0, \dots, y_T)$, solve:

$$\underset{z_0,\dots,z_T}{\operatorname{argmax}}\, p_\theta(z_0, \dots, z_T | y_0, \dots, y_T)$$

- Learning. Given N observations $\{\boldsymbol{y}^{(n)}\}_{n=1}^{N}$, find best θ:

$$\max_\theta \prod_{n=1}^{N} p_\theta(\boldsymbol{y}^{(n)})$$

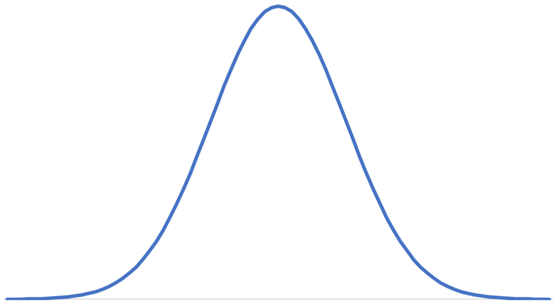
State Estimation

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s|y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s|y_0, \dots, y_t)$

- **State Estimation**. Given θ and $(y_0, \dots, y_T)$, solve:

$$\operatorname*{argmax}_{z_0, \dots, z_T} p_\theta(z_0, \dots, z_T | y_0, \dots, y_T)$$

- **Solution**:
  - Since $p_\theta(z_0, \dots, z_T | y_0, \dots, y_T)$ is Gaussian, the optimal solution to state estimation is

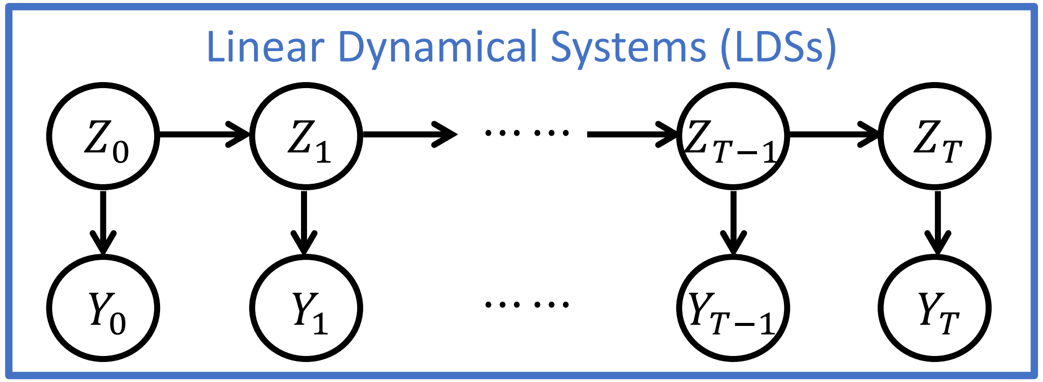


$$\mathbb{E}\left[\begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_T \end{bmatrix} \middle| y_0, \dots, y_T\right] = \begin{bmatrix} \mathbb{E}[z_0|y_0, \dots, y_T] \\ \mathbb{E}[z_1|y_0, \dots, y_T] \\ \vdots \\ \mathbb{E}[z_T|y_0, \dots, y_T] \end{bmatrix} = \begin{bmatrix} \hat{z}_{0|T} \\ \hat{z}_{1|T} \\ \vdots \\ \hat{z}_{T|T} \end{bmatrix}$$

  - Therefore, the state estimation problem can be solved by smoothing

- Similarly, we can prove the Kalman filter gives the optimal solution $\hat{z}_{t|t}$ to $\max_{z_t} p_\theta(z_t|y_0, \dots, y_t)$

Learning



Model Parameters:
- $\theta \coloneqq (\pi_0, \Sigma_0, A, Q, C, R)$

- Learning. Given N observations $\{\boldsymbol{y}^{(n)}\}_{n=1}^{N}$, find best θ:

$$\max_{\theta} \prod_{n=1}^{N} p_\theta(\boldsymbol{y}^{(n)})$$

Learning (Review Lecture 2)

- The likelihood $\prod_{i=1}^{N} p_\theta(x_i) = \dfrac{\exp\left(-\frac{1}{2}\sum_{i=1}^{N}(x_i-\mu)^\top\Sigma^{-1}(x_i-\mu)\right)}{(2\pi)^{\frac{ND}{2}}\det(\Sigma)^{\frac{N}{2}}}$ is maximized at

$$\mu^* = \frac{\sum_{i=1}^{N} x_i}{N}, \qquad \Sigma^* = \frac{\sum_{i=1}^{N}(x_i-\mu^*)(x_i-\mu^*)^\top}{N}$$

empirical mean

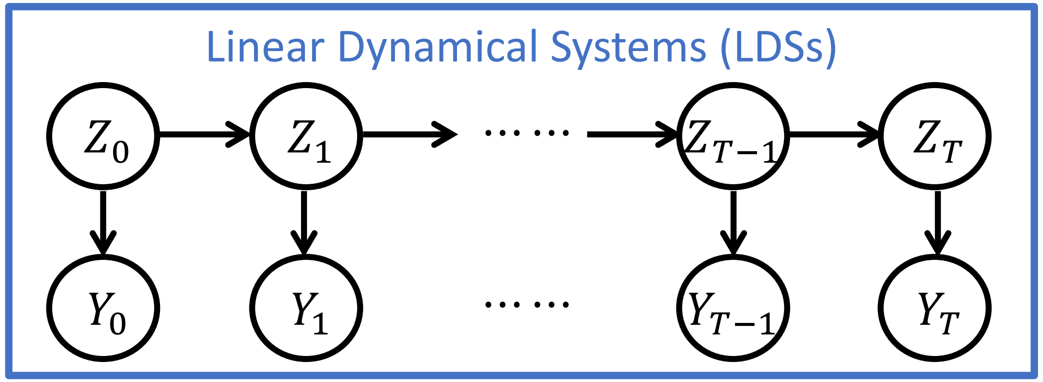
empirical covariance

- How did we obtain μ^* and Σ^*?
  - Step 0: Rewrite the objective
    - maximizing log-likelihood is minimizing $N \log\det\Sigma + \sum_{i=1}^{N}(x_i-\mu)^\top\Sigma^{-1}(x_i-\mu)$
  - Step 1: Set the derivative w.r.t. μ to 0
    - solving it gives μ^*
  - Step 2: Substitute $\mu = \mu^*$ into the objective, and set the derivative w.r.t. Σ^{-1} to 0

$m \cdot \log\det\Sigma + \mathrm{tr}(S\Sigma^{-1})$ *is minimized at* $\Sigma = S/m$

Learning

Model Parameters:
- $\theta \coloneqq (\pi_0, \Sigma_0, A, Q, C, R)$



- Learning. Given N observations $\{\boldsymbol{y}^{(n)}\}_{n=1}^{N}$, find best θ:

$$\max_{\theta} \prod_{n=1}^{N} p_\theta(\boldsymbol{y}^{(n)})$$

- We are going to apply the EM algorithm (iteration: k):

E-step:
$$q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)}) = p_{\theta^k}(\boldsymbol{z}\,|\boldsymbol{y}^{(n)})$$

M-step:
$$\theta^{k+1} = \operatorname{argmax}_\theta \sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}\left[\log p_\theta(\boldsymbol{y}^{(n)}, \boldsymbol{z})\right]$$

Guessing

Model Parameters:
- $\theta := (\pi_0, \Sigma_0, A, Q, C, R)$

E-step:

$$q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)}) = p_{\theta^k}(\boldsymbol{z}\,|\boldsymbol{y}^{(n)})$$

M-step:

$$\theta^{k+1} = \mathrm{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}\left[\log p_\theta(\boldsymbol{y}^{(n)}, \boldsymbol{z})\right]$$

$\mathbb{P}(Z_0) = \mathcal{N}(\pi_0, \Sigma_0)$ $\quad$ $\mathbb{P}(Z_t|z_{t-1}) = \mathcal{N}(Az_{t-1}, Q)$ $\quad$ $\mathbb{P}(Y_t|z_t) = \mathcal{N}(Cz_t, R)$

- Let us exercise our intuition and guess a solution to the M-step…
  - Since $\pi_0 = \mathbb{E}[z_0]$, we guess… :

$$\pi_0^{k+1} = \sum_{n=1}^N \frac{\mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_0]}{N}$$

"empirical mean"

  - And we guess the covariance should be

$$\Sigma_0^{k+1} = \sum_{n=1}^N \frac{\left(\mathbb{E}_{q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_0] - \pi_0^{k+1}\right)\left(\mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_0] - \pi_0^{k+1}\right)^{\mathsf{T}}}{N}$$

"empirical covariance"

    - Try having a guess for $A^{k+1}, Q^{k+1}, C^{k+1}, R^{k+1}$ yourself…

E-step (iteration: k)

$\hat{z}_{s|t} := \mathbb{E}[z_s|y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} := \mathrm{Cov}(z_s|y_0, \dots, y_t)$

E-step:
$$q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)}) = p_{\theta^k}(\boldsymbol{z}\,|\boldsymbol{y}^{(n)})$$

M-step:
$$\theta^{k+1} = \mathrm{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[\log p_\theta(\boldsymbol{y}^{(n)}, \boldsymbol{z})]$$

- In E-step, we will need to compute the expectation $\mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[\cdot]$.

- *"It turns out that"* …… we only need to compute the following expectations:

$$\mathbb{E}_k^{(n)}[z_t] := \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t]$$
$$\mathbb{E}_k^{(n)}[z_t z_t^\top] := \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t z_t^\top]$$
$$\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] := \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t z_{t-1}^\top]$$

  - later you will see why…

E-step (iteration: $\boldsymbol{k}$)

$\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
$\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$

E-step:
$$q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)}) = p_{\theta^k}(\boldsymbol{z}\,|\boldsymbol{y}^{(n)})$$

M-step:
$$\theta^{k+1} = \mathrm{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[\log p_\theta(\boldsymbol{y}^{(n)}, \boldsymbol{z})]$$

$$\mathbb{E}_k^{(n)}[z_t] \coloneqq \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t]$$
$$\mathbb{E}_k^{(n)}[z_t z_t^\top] \coloneqq \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t z_t^\top]$$
$$\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] \coloneqq \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t z_{t-1}^\top]$$

- These expectations can all be computed via smoothing using θ^k and $\boldsymbol{y}^{(n)}$
  - To see this, dropping indices k, n for clarity, we have

$$\mathbb{E}[z_t|\boldsymbol{y}] = \mathbb{E}[z_t|y_0, \dots, y_T] = \hat{z}_{t|T}$$
$$\begin{aligned}\mathbb{E}[z_t z_t^\top|\boldsymbol{y}] &= \mathrm{Cov}(z_t|y_0, \dots, y_T) + \mathbb{E}[z_t|y_0, \dots, y_T]\mathbb{E}[z_t^\top|y_0, \dots, y_T] \\ &= \hat{\Sigma}_{t|T} + \hat{z}_{t|T}\hat{z}_{t|T}^\top\end{aligned}$$
$$\mathbb{E}[z_t z_{t-1}^\top|\boldsymbol{y}] = \hat{\Sigma}_{t|T} L_{t-1}^\top + \hat{z}_{t|T}\hat{z}_{t-1|T}^\top$$

homework

$L_{t-1} = \hat{\Sigma}_{t-1|t-1} A^\top \hat{\Sigma}_{t|t-1}^{-1}$

M-step (iteration: k)

$$\mathbb{E}_k^{(n)}[z_t] := \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t]$$
$$\mathbb{E}_k^{(n)}[z_t z_t^\top] := \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t z_t^\top]$$
$$\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] := \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t z_{t-1}^\top]$$

$$\mathbb{P}(Z_0) = \mathcal{N}(\pi_0, \Sigma_0)$$
$$\mathbb{P}(Z_t|z_{t-1}) = \mathcal{N}(Az_{t-1}, Q)$$
$$\mathbb{P}(Y_t|z_t) = \mathcal{N}(Cz_t, R)$$

Model Parameters:
- $\theta := (\pi_0, \Sigma_0, A, Q, C, R)$

$$\theta^{k+1} = \operatorname{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}\left[\log p_\theta\left(\mathbf{y}^{(n)}, \mathbf{z}\right)\right]$$

- Observation. In the joint log-likelihood

$$p_\theta(y_0, \dots, y_T, z_0, \dots, z_T) = p_\theta(z_0) \prod_{t=0}^T p_\theta(y_t|z_t) \prod_{t=1}^T p_\theta(z_t|z_{t-1}),$$

  - π_0, Σ_0 only appear in $p_\theta(z_0)$
  - A, Q only appear in $\prod_{t=1}^T p_\theta(z_t|z_{t-1})$
  - C, R only appear in $\prod_{t=0}^T p_\theta(y_t|z_t)$

  - So the objective of the M-step is separable (as in HMMs), this gives … (next page)

M-step (iteration: k)

$$\mathbb{P}(Z_0) = \mathcal{N}(\pi_0, \Sigma_0)$$
$$\mathbb{P}(Z_t|z_{t-1}) = \mathcal{N}(Az_{t-1}, Q)$$
$$\mathbb{P}(Y_t|z_t) = \mathcal{N}(Cz_t, R)$$

Model Parameters:
- $\theta := (\pi_0, \Sigma_0, A, Q, C, R)$

$$\mathbb{E}_k^{(n)}[z_t] := \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t]$$
$$\mathbb{E}_k^{(n)}[z_t z_t^\top] := \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t z_t^\top]$$
$$\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] := \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t z_{t-1}^\top]$$

$$\theta^{k+1} = \operatorname{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}\left[\log p_\theta\left(\boldsymbol{y}^{(n)}, \boldsymbol{z}\right)\right]$$

- We can therefore decompose M-step into 3 optimization problems (as in HMMs):

M-step (π_0, Σ_0):
$$(\pi_0^{k+1}, \Sigma_0^{k+1}) = \operatorname{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[\log p_\theta(z_0)]$$

M-step (C, R):
$$(C^{k+1}, R^{k+1}) = \operatorname{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}\left[\sum_{t=0}^T \log p_\theta\left(y_t^{(n)}\middle|z_t\right)\right]$$

M-step (A, Q):
$$(A^{k+1}, Q^{k+1}) = \operatorname{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}\left[\sum_{t=1}^T \log p_\theta(z_t|z_{t-1})\right]$$

- We will address them one by one next

M-step (π_0, Σ_0)

$$\mathbb{P}(Z_0) = \mathcal{N}(\pi_0, \Sigma_0)$$

$$\mathbb{E}_k^{(n)}[z_t] := \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t]$$
$$\mathbb{E}_k^{(n)}[z_t z_t^\top] := \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t z_t^\top]$$
$$\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] := \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t z_{t-1}^\top]$$

$$(\pi_0^{k+1}, \Sigma_0^{k+1}) = \operatorname{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[\log p_\theta(z_0)]$$

- Since $p_\theta(z_0)$ is Gaussian, we have:

$$\log p_\theta(z_0) \propto -\log\det\Sigma_0 - (z_0 - \pi_0)^\top \Sigma_0^{-1}(z_0 - \pi_0)$$

- And now we get this:

$$(\pi_0^{k+1}, \Sigma_0^{k+1}) = \operatorname{argmin}_\theta N \log\det\Sigma_0 + \sum_{n=1}^N \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[(z_0 - \pi_0)^\top \Sigma_0^{-1}(z_0 - \pi_0)]$$

M-step (π_0, Σ_0)

$$\mathbb{E}_k^{(n)}[z_t] \coloneqq \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t]$$
$$\mathbb{E}_k^{(n)}[z_t z_t^\top] \coloneqq \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t z_t^\top]$$
$$\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] \coloneqq \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[z_t z_{t-1}^\top]$$

$$(\pi_0^{k+1}, \Sigma_0^{k+1}) = \operatorname{argmin}_\theta N \log\det\Sigma_0 + \sum_{n=1}^N \mathbb{E}_{\boldsymbol{z}\sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}[(z_0 - \pi_0)^\top \Sigma_0^{-1}(z_0 - \pi_0)]$$

use the definitions of $\mathbb{E}_k^{(n)}[z_0]$ and $\mathbb{E}_k^{(n)}[z_0 z_0^\top]$

$$(\pi_0^{k+1}, \Sigma_0^{k+1}) = \operatorname{argmin}_\theta N \log\det\Sigma_0 + N\pi_0^\top \Sigma_0^{-1}\pi_0 + \sum_{n=1}^N \operatorname{tr}\left(\mathbb{E}_k^{(n)}[z_0 z_0^\top]\Sigma_0^{-1}\right) - 2\sum_{n=1}^N \pi_0^\top \Sigma_0^{-1}\mathbb{E}_k^{(n)}[z_0]$$

- Setting the derivative with respect to π_0 to 0 yields $\pi_0^{k+1} = \dfrac{\sum_{n=1}^N \mathbb{E}_k^{(n)}[z_0]}{N}$

Use π_0^{k+1}

$$\Sigma_0^{k+1} = \operatorname{argmin}_\theta N \cdot \log\det\Sigma_0 + \sum_{n=1}^N \operatorname{tr}\left(\mathbb{E}_k^{(n)}[z_0 z_0^\top]\Sigma_0^{-1}\right) - N \cdot \operatorname{tr}\left(\pi_0^{k+1}(\pi_0^{k+1})^\top \Sigma_0^{-1}\right)$$

$m \cdot \log\det\Sigma + \operatorname{tr}(S\Sigma^{-1})$ *is minimized at* $\Sigma = S/m$

$$\Sigma_0^{k+1} = \frac{\sum_{n=1}^N \mathbb{E}_k^{(n)}[z_0 z_0^\top]}{N} - \pi_0^{k+1}(\pi_0^{k+1})^\top$$

M-step (C, R)

$\mathbb{P}(Y_t|z_t) = \mathcal{N}(Cz_t, R)$

$\mathbb{E}_k^{(n)}[z_t] \coloneqq \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t]$

$\mathbb{E}_k^{(n)}[z_t z_t^\top] \coloneqq \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t z_t^\top]$

$\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] \coloneqq \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t z_{t-1}^\top]$

$$(C^{k+1}, R^{k+1}) = \operatorname{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}\left[\sum_{t=0}^T \log p_\theta\left(y_t^{(n)}\middle|z_t\right)\right]$$

$p_\theta\left(y_t^{(n)}\middle|z_t\right)$ is Gaussian

$$(C^{k+1}, R^{k+1}) = \operatorname{argmin}_\theta N(T+1)\log\det R + \sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_{z_t\sim q^k(z_t|\mathbf{y}^{(n)})}\left[\left(y_t^{(n)} - Cz_t\right)^\top R^{-1}\left(y_t^{(n)} - Cz_t\right)\right]$$

use the definitions of $\mathbb{E}_k^{(n)}[z_t]$ and $\mathbb{E}_k^{(n)}[z_t z_t^\top]$

$$\min_\theta N(T+1)\log\det R + \sum_{n=1}^N \sum_{t=0}^T \left\{\operatorname{tr}\left(\left(C\mathbb{E}_k^{(n)}[z_t z_t^\top]C^\top + y_t^{(n)}\left(y_t^{(n)}\right)^\top\right)R^{-1} - 2y_t^{(n)}\mathbb{E}_k^{(n)}[z_t]^\top C^\top R^{-1}\right)\right\}$$

- Setting the derivative with respect to C to 0 yields

$$C^{k+1} = \left(\sum_{n=1}^N \sum_{t=0}^T y_t^{(n)}\,\mathbb{E}_k^{(n)}[z_t]^\top\right)\left(\sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_k^{(n)}[z_t z_t^\top]\right)^{-1}$$

Use C^{k+1}

$$\min_\theta N(T+1)\log\det R + \sum_{n=1}^N \sum_{t=0}^T \left\{\operatorname{tr}\left(y_t^{(n)}\left(y_t^{(n)}\right)^\top R^{-1} - y_t^{(n)}\mathbb{E}_k^{(n)}[z_t]^\top (C^{k+1})^\top R^{-1}\right)\right\}$$

$$\frac{\partial \operatorname{tr}(C^\top X)}{\partial C} = X$$

$$\frac{\partial \operatorname{tr}(C^\top X C Y)}{\partial C} = XCY + X^\top C Y^\top$$

$m \cdot \log\det\Sigma + \operatorname{tr}(S\Sigma^{-1})$ is minimized at $\Sigma = S/m$

$$R^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T y_t^{(n)}\left(y_t^{(n)}\right)^\top - C^{k+1}\sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_k^{(n)}[z_t]\left(y_t^{(n)}\right)^\top}{N(T+1)}$$

M-step (A, Q)

$$\mathbb{P}(Z_t|z_{t-1}) = \mathcal{N}(Az_{t-1}, Q)$$

$$\mathbb{E}_k^{(n)}[z_t] \coloneqq \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t]$$
$$\mathbb{E}_k^{(n)}[z_t z_t^\top] \coloneqq \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t z_t^\top]$$
$$\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] \coloneqq \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[z_t z_{t-1}^\top]$$

$$(A^{k+1}, Q^{k+1}) = \mathrm{argmax}_\theta \sum_{n=1}^N \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[\sum_{t=1}^T \log p_\theta(z_t|z_{t-1})]$$

$p_\theta(z_t|z_{t-1})$ is Gaussian

$$(A^{k+1}, Q^{k+1}) = \mathrm{argmin}_\theta\, NT \log\det Q + \sum_{n=1}^N \sum_{t=1}^T \mathbb{E}_{\mathbf{z}\sim q^k(\mathbf{z}|\mathbf{y}^{(n)})}[(z_t - Az_{t-1})^\top Q^{-1}(z_t - Az_{t-1})]$$

use the definitions of $\mathbb{E}_k^{(n)}[z_t z_t^\top]$ and $\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top]$

$$\min_\theta NT \log\det Q + \sum_{n=1}^N \sum_{t=1}^T \left\{\mathrm{tr}\left(\left(A\left(\mathbb{E}_k^{(n)}[z_{t-1} z_{t-1}^\top]\right)A^\top + \mathbb{E}_k^{(n)}[z_t z_t^\top]\right)Q^{-1} - 2\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top]A^\top Q^{-1}\right)\right\}$$

- Setting the derivative with respect to A to 0 yields

$$A^{k+1} = \left(\sum_{n=1}^N \sum_{t=1}^T \mathbb{E}_k^{(n)}[z_t z_{t-1}^\top]\right)\left(\sum_{n=1}^N \sum_{t=1}^T \mathbb{E}_k^{(n)}[z_{t-1} z_{t-1}^\top]\right)^{-1}$$

Use A^{k+1}

$$\min_\theta NT \log\det Q + \sum_{n=1}^N \sum_{t=1}^T \left\{\mathrm{tr}\left(\mathbb{E}_k^{(n)}[z_t z_t^\top]Q^{-1} - \mathbb{E}_k^{(n)}[z_t z_{t-1}^\top](A^{k+1})^\top Q^{-1}\right)\right\}$$

$m \cdot \log\det\Sigma + \mathrm{tr}(S\Sigma^{-1})$ *is minimized at* $\Sigma = S/m$

$$\frac{\partial\, \mathrm{tr}(A^\top X)}{\partial A} = X$$
$$\frac{\partial\, \mathrm{tr}(A^\top XAY)}{\partial A} = XAY + X^\top AY^\top$$

$$Q^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_k^{(n)}[z_t z_t^\top] - A^{k+1}\sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_k^{(n)}[z_{t-1} z_t^\top]}{NT}$$

Summary: EM for LDSs (iteration: k)

- $\hat{z}_{s|t} \coloneqq \mathbb{E}[z_s | y_0, \dots, y_t]$
- $\hat{\Sigma}_{s|t} \coloneqq \mathrm{Cov}(z_s | y_0, \dots, y_t)$
- $L_{t-1} \coloneqq \hat{\Sigma}_{t-1|t-1} A^\top \hat{\Sigma}_{t|t-1}^{-1}$

E-step

Given θ^k, for each $\boldsymbol{y}^{(n)}$, use Kalman Filter & Smoothing to compute:

- $\mathbb{E}_k^{(n)}[z_t] \coloneqq \mathbb{E}_{\boldsymbol{z} \sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}\left[z_t | \boldsymbol{y}^{(n)}\right]$
- $\mathbb{E}_k^{(n)}[z_t z_t^\top] \coloneqq \mathbb{E}_{\boldsymbol{z} \sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}\left[z_t z_t^\top | \boldsymbol{y}^{(n)}\right]$
- $\mathbb{E}_k^{(n)}[z_t z_{t-1}^\top] \coloneqq \mathbb{E}_{\boldsymbol{z} \sim q^k(\boldsymbol{z}|\boldsymbol{y}^{(n)})}\left[z_t z_{t-1}^\top | \boldsymbol{y}^{(n)}\right]$

add indices n, k

Kalman Filter & Smoothing can compute

- $\mathbb{E}[z_t | \boldsymbol{y}] = \hat{z}_{t|T}$
- $\mathbb{E}[z_t z_t^\top | \boldsymbol{y}] = \hat{\Sigma}_{t|T} + \hat{z}_{t|T} \hat{z}_{t|T}^\top$
- $\mathbb{E}[z_t z_{t-1}^\top | \boldsymbol{y}] = L_{t-1} \hat{\Sigma}_{t|T} + \hat{z}_{t|T} \hat{z}_{t-1|T}^\top$

via forward & backward passes

$$
\begin{tikzcd}
\hat{z}_{0|0} \arrow[d] & \hat{z}_{0|1} & \hat{z}_{0|2} & \hat{z}_{0|3} \\
\hat{z}_{1|0} \arrow[r] & \hat{z}_{1|1} \arrow[d] & \hat{z}_{1|2} & \hat{z}_{1|3} \arrow[u] \\
\hat{z}_{2|0} & \hat{z}_{2|1} \arrow[r] & \hat{z}_{2|2} \arrow[d] & \hat{z}_{2|3} \arrow[u] \\
\hat{z}_{3|0} & \hat{z}_{3|1} & \hat{z}_{3|2} \arrow[r] & \hat{z}_{3|3} \arrow[u]
\end{tikzcd}
$$

M-step

Update parameters:

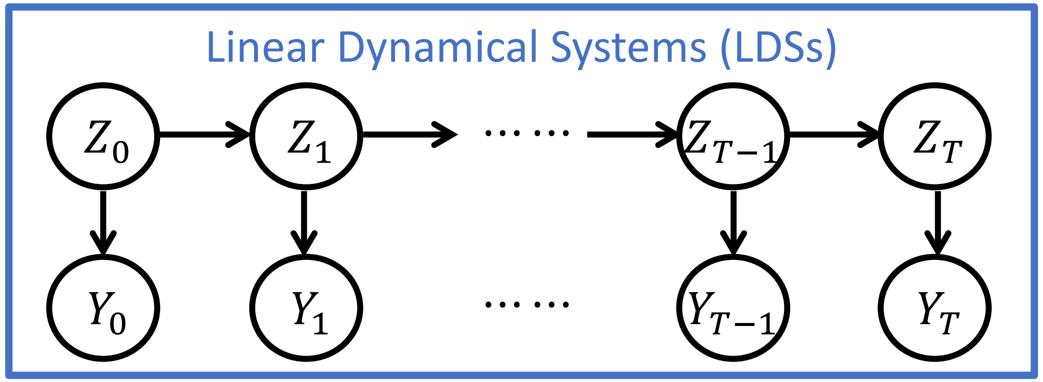
- $\pi_0^{k+1} = \dfrac{\sum_{n=1}^N \mathbb{E}_k^{(n)}[z_0]}{N}$
- $\Sigma_0^{k+1} = \dfrac{\sum_{n=1}^N \mathbb{E}_k^{(n)}[z_0 z_0^\top]}{N} - \pi_0^{k+1} \left(\pi_0^{k+1}\right)^\top$
- $C^{k+1} = \left(\sum_{n=1}^N \sum_{t=0}^T y_t^{(n)} \mathbb{E}_k^{(n)}[z_t]^\top\right) \left(\sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_k^{(n)}[z_t z_t^\top]\right)^{-1}$
- $R^{k+1} = \dfrac{\sum_{n=1}^N \sum_{t=0}^T y_t^{(n)} \left(y_t^{(n)}\right)^\top - C^{k+1} \sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_k^{(n)}[z_t] \left(y_t^{(n)}\right)^\top}{N(T+1)}$
- $A^{k+1} = \left(\sum_{n=1}^N \sum_{t=1}^T \mathbb{E}_k^{(n)}[z_t z_{t-1}^\top]\right) \left(\sum_{n=1}^N \sum_{t=1}^T \mathbb{E}_k^{(n)}[z_{t-1} z_{t-1}^\top]\right)^{-1}$
- $Q^{k+1} = \dfrac{\sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_k^{(n)}[z_t z_t^\top] - A^{k+1} \sum_{n=1}^N \sum_{t=0}^T \mathbb{E}_k^{(n)}[z_{t-1} z_t^\top]}{NT}$

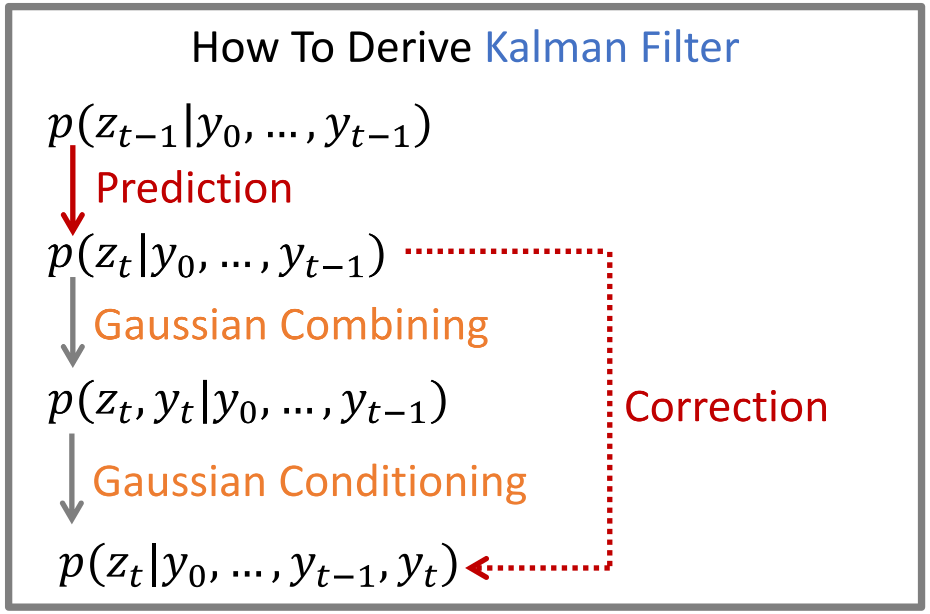
Model Parameters:

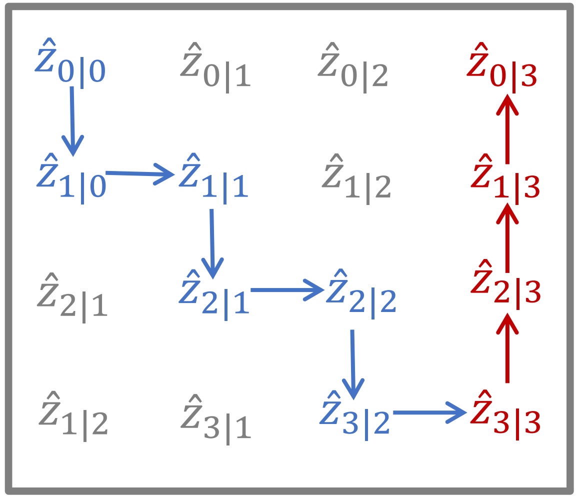
- $\theta \coloneqq (\pi_0, \Sigma_0, A, Q, C, R)$

$$
\begin{aligned}
\mathbb{P}(Z_0) &= \mathcal{N}(\pi_0, \Sigma_0) \\
\mathbb{P}(Z_t | Z_{t-1}) &= \mathcal{N}(A z_{t-1}, Q) \\
\mathbb{P}(Y_t | Z_t) &= \mathcal{N}(C z_t, R)
\end{aligned}
$$

Possible Extensions of Linear Dynamical Systems







- What if ……
  - we do not have Gaussians?
  - we have time-varying dynamics?
    - $Z_t = A_t z_{t-1} + w_t,\ Y_t = C_t z_t + v_t$
  - we have control over states?
    - $Z_t = A z_{t-1} + U x_t + w_t$
  - we have nonlinear dynamics?
    - $Z_t = f(z_{t-1}) + w_t,\ Y_t = g(z_t) + v_t$

Deep Generative Models: Dynamic Textures

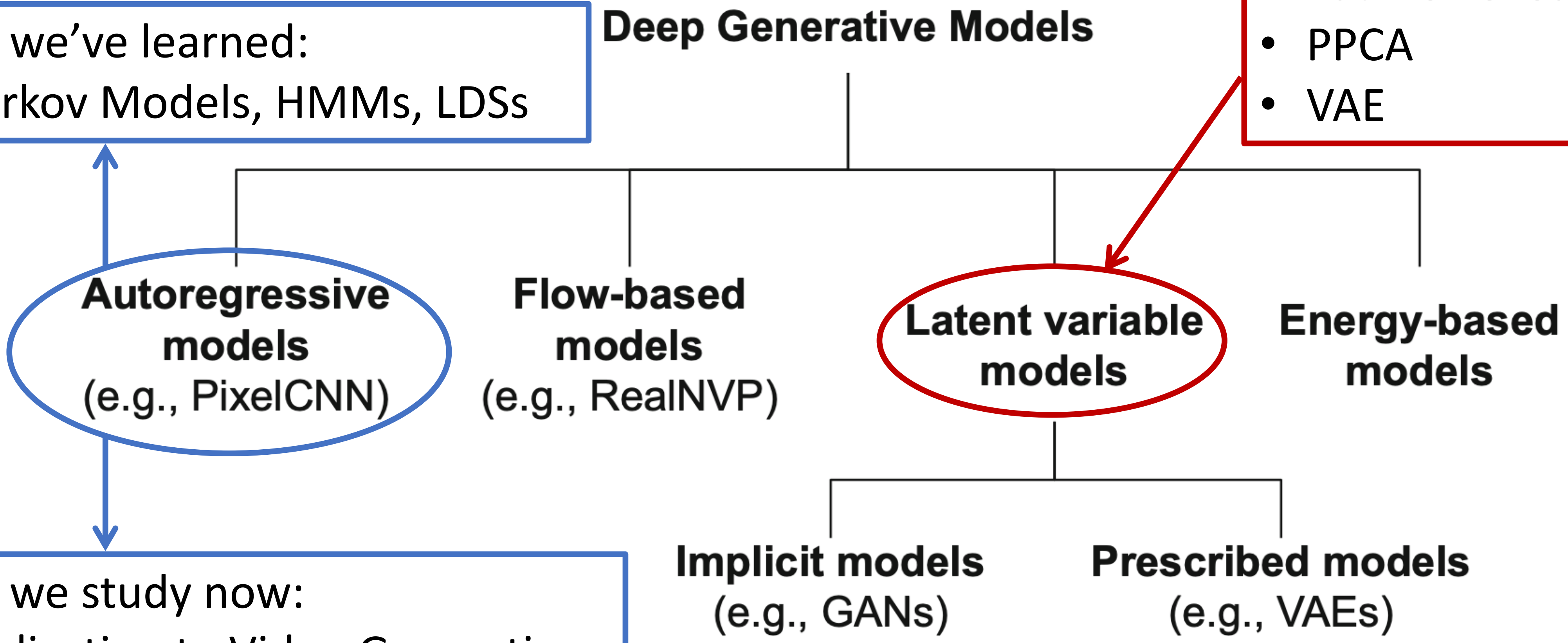
Fall Semester 2024

René Vidal
Director of the Center for Innovation in Data Engineering and Science (IDEAS),
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

Taxonomy of Generative Models

What we've learned:
- Markov Models, HMMs, LDSs

What we've learned:
- PPCA
- VAE



What we study now:
- Application to Video Generation: Dynamic Textures

Motivation: Dynamic Textures

- Dynamic textures are videos of nonrigid deformable objects such as water, fire, smoke, flags moving with the wind, etc.



- Dynamic textures are characterized by appearance + dynamics


Need a model that captures the appearance and dynamics

Dynamic Texture Model

- A video is a sequence of T images $\{I_t \in \mathbb{R}^D\}_{t=0}^{T}$, where D is the number of pixels.
- Suppose at each time instant we observe a noisy version of the image

$$y_t = I_t + v_t$$

where $v_t \sim \mathcal{N}(0, Q)$ is an i.i.d. sequence Gaussian noise.

- Suppose each image is an affine combination of d basis functions $\{C_i \in \mathbb{R}^D\}_{i=1}^{d}$

$$I_t = C_0 + C z_t$$

- If $Q = \sigma_v^2 I_D$ and $z_t \sim^{i.i.d.} \mathcal{N}(0, I_d)$, then $y_t = C_0 + C z_t + v_t$ is a PPCA model,
  - C_0 is the mean image/average video frame and C_i is the ith principal direction/eigenimage.
- However, this is not a good model for video because all frames are i.i.d.

Dynamic Texture Model

- To capture the temporal evolution of a video, instead of assuming z_t i.i.d., we assume that z_t is a linear autoregressive model $z_{t+1} = Az_t + w_t$.

- We say the video sequence $\{y_t \in \mathbb{R}^D\}_{t=0}^T$ is a **linear dynamic texture** if

$$z_{t+1} = Az_t + w_t$$
$$y_t = C_0 + Cz_t + v_t$$

where

- D is the number of pixels and T is the number of frames
- $z_t \in \mathbb{R}^d$ is the hidden state, d is its dimension, and z_0 is the initial state
- $w_t \sim \mathcal{N}(0, Q)$ is the state noise, i.i.d. sequence, assumed to be independent from z_t
- $v_t \sim \mathcal{N}(0, R)$ is the output noise, i.i.d. sequence, assumed to be independent from z_t
- $A \in \mathbb{R}^{d \times d}$, $C_0 \in \mathbb{R}^D$, $C \in \mathbb{R}^{D \times d}$, $Q \in \mathbb{R}^{d \times d}$ and $R \in \mathbb{R}^{D \times D}$ are the model parameters

Dynamic Textures

$$z_{t+1} = Az_t + w_t$$
$$y_t = C_0 + Cz_t + v_t$$

(Soatto ICCV 01, Doretto IJCV 03)

$C_0 \in \mathbb{R}^D$
$C \in \mathbb{R}^{D \times d}$

$A \in \mathbb{R}^{d \times d}$
$z_0 \in \mathbb{R}^d$

$Q \in \mathbb{R}^{d \times d}$
$R \in \mathbb{R}^{D \times D}$

APPEARANCE

DYNAMICS

NOISE

Learning Dynamic Textures: MLE

- Let $\{y_t\}_{t=0\ldots T},\ y_t \in \mathbb{R}^D$ be an observed video sequence. Our goal is to learn the model parameters A, C_0, C, Q, R, a.k.a. the system identification problem.
- One method we have learned is Maximum Likelihood (using EM)
  - Given $y_0, \ldots, y_T$, solve

$$\hat{A}, \hat{C}_0, \hat{C}, \hat{Q}, \hat{R} = \underset{A,C_0,C,Q,R}{\operatorname{argmax}} \log p(y_0, \ldots, y_T)$$

  - subject to

$$\begin{aligned} z_{t+1} &= Az_t + w_t \qquad t = 0, \ldots, T-1, \quad w_t \sim^{i.i.d.} \mathcal{N}(0, Q) \qquad t = 0, \ldots, T \\ y_t &= C_0 + Cz_t + v_t \qquad t = 0, \ldots, T, \qquad v_t \sim^{i.i.d.} \mathcal{N}(0, R) \qquad t = 0, \ldots, T \end{aligned}$$

- Note that if $\mathbb{E}[z_0] = 0$, then $\mathbb{E}[z_{t+1}] = \mathbb{E}[Az_t] + \mathbb{E}[w_t] = A\mathbb{E}[z_t] = A^t\mathbb{E}[z_0] = 0$
- Note also that $\mathbb{E}[y_t] = C_0 + C\mathbb{E}[z_t] + \mathbb{E}[v_t] = C_0$.
- Thus, we can estimate C_0 as the mean video $\hat{C}_0 = \frac{1}{T}\sum y_t$, subtract the mean from each frame so that $y_t \leftarrow y_t - \hat{C}_0$ is an LDS and we can use EM to learn A, C, Q, R.

Learning Dynamic Textures: Min. Prediction Error

- Another method is to use a one step Prediction Error
  - Predict the best you can for the next step given your previous steps

- E.g., One can consider using the mean squared error.
- Suppose $\hat{z}_{t+1|t} = \mathbb{E}[z_{t+1}|y_0 \ldots y_t]$

  - Given $y_0, \ldots, y_T$, solve

$$\hat{A}, \hat{B}, \hat{C}, \hat{Q}, \hat{R} = \underset{A,B,C,Q,R}{\operatorname{argmin}} \mathbb{E}||y_{t+1} - C\hat{z}_{t+1|t}||^2$$

  - subject to

$$z_{t+1} = Az_t + w_t \quad t = 0, \ldots, T-1, \quad w_t \sim^{i.i.d.} \mathcal{N}(0, Q) \quad t = 0, \ldots, T$$
$$y_t = Cz_t + v_t \quad t = 0, \ldots, T, \quad v_t \sim^{i.i.d.} \mathcal{N}(0, R) \quad t = 0, \ldots, T$$

Formal Problem Statement

- For our given process $\{z_t\}, z_t \in \mathbb{R}^d$, guided by

$$z_{t+1} = Az_t + w_t \qquad w_t \sim \mathcal{N}(0, Q) \qquad z(0) = z_0$$
$$y_t = Cz_t + v_t \qquad v_t \sim \mathcal{N}(0, R)$$

with initial condition $z(0) = z_0$, positive definite matrices R and Q.

- We seek the estimate parameters $A \in \mathbb{R}^{d\times d}, C \in \mathbb{R}^{D\times D}, Q \in \mathbb{S}_+^{d\times d}, R \in \mathbb{S}_+^{D\times D}$ from the given samples $y_1, \ldots, y_T$ that maximizes the log-likelihood

$$\hat{A}_T, \hat{C}_T, \hat{Q}_T, \hat{R}_T, = \underset{A,C,Q,R}{\operatorname{argmax}} \log p(y_0, \ldots, y_T)$$

- As it turns out, if we use the aforementioned **predictor error approach**, there is a closed-form solution to the problem.
- And we can solve for the A, C, Q, R independently

Closed Form Solutions

- Let $Y_{0:T} = [y_0, \ldots, y_T] \in \mathbb{R}^{D\times(T+1)}$, $Z_{0:T} = [z_0, \ldots, z_T] \in \mathbb{R}^{d\times(T+1)}$ with $T+1 > d$, $W_{0:T} = [w_0, \ldots, w_T] \in \mathbb{R}^{d\times(T+1)}$
- Our assumption $D \gg d, \mathrm{rank}(C) = d$ and $C^T C = I_d$ allows us to rewrite the output equation as

$$Y_{0:T} = CZ_{0:T} + W_{0:T}; \qquad C \in \mathbb{R}^{D\times d}; C^\top C = I_d$$

- The best estimate of C can be found by the following

$$\hat{C}_T, \hat{Z}_{0:T} = \mathrm{argmin}_{C, Z_{0:T}} \left|\left|Y_{0:T} - CZ_{0:T}\right|\right|_F^2 \text{ subject to } C^\top C = I_d$$

- To which the solution can be found using SVD. Let $Y_{0:T} = U\Sigma V^\top$, then

$$\hat{C}_T = U \qquad \hat{Z}_{0:T} = \Sigma V^\top$$

- This solution is very close to learning a PPCA model for $Y_{0:T}$.

Closed-form Solution

$$\begin{aligned} Y_{0:T} &= U\Sigma V^\top \\ \hat{Z}_{0:T} &= \Sigma V^\top \end{aligned}$$

- The best estimate for A can be solved by the following

$$\boxed{\hat{A}_T = \mathrm{argmin}_A \left\|\hat{Z}_{1:T} - A\hat{Z}_{0:T-1}\right\|_F = \hat{Z}_{0:T}\hat{Z}_{0:T}^\top\left(\hat{Z}_{0:T}\hat{Z}_{0:T}^\top\right)^{-1}}$$

- The solution to the above can be unique determined (assuming eigenvalues of A are distinct) by

$$\boxed{\hat{A}_T = \Sigma V^\top D_1 V (V^\top D_2 V)^{-1}\Sigma^{-1}}$$

- Finally the state and output covariances can be estimated as

$$\boxed{\hat{Q}_T = \frac{1}{T}\sum_{t=0}^{T-1} \widehat{w}_t \widehat{w}_t^\top} \qquad \boxed{\hat{R}_T = \frac{1}{T}\sum_{t=0}^{T-1} \hat{v}_t \hat{v}_t^\top}$$

where $\widehat{w}_t \doteq y_t - \hat{C}_T \hat{z}_t$ and $\hat{v}_t \doteq \hat{z}_{t+1} - \hat{A}_T \hat{z}_t$.

Closed-form solution: Pseudocode

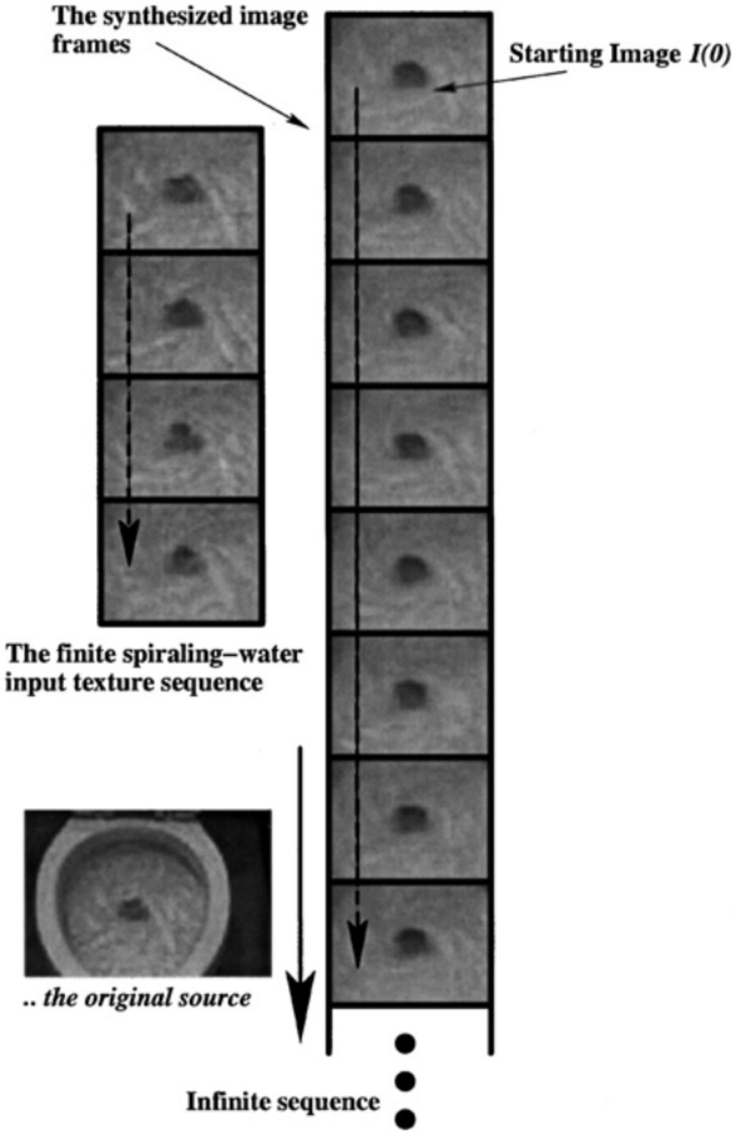
- Therefore, everything can be computed in closed-form

- In the original work, the algorithm is implemented in fewer then 20 lines

```
function [x0,Ymean,Ahat,Bhat,Chat] = dytex(Y,n,nv)

% Suboptimal Learning of Dynamic Textures;
% (c) UCLA, March 2001.

  tau = size(Y,2); Ymean = mean(Y,2);
  [U,S,V] = svd(Y-Ymean*ones(1,tau),0);
  Chat=U(:,1:n); Xhat = S(1:n,1:n)*V(:,1:n)';
  x0=Xhat(:,1);
  Ahat = Xhat(:,2:tau)*pinv(Xhat(:,1:(tau-1)));
  Vhat = Xhat(:,2:tau)-Ahat*Xhat(:,1:(tau-1));
  [Uv,Sv,Vv] = svd(Vhat,0);
  Bhat  = Uv(:,1:nv)*Sv(1:nv,1:nv)/sqrt(tau-1);

function [I] = synth(x0,Ymean,Ahat,Bhat,Chat,tau)

% Synthesis of Dynamic Textures;
% (c) UCLA, March 2001.

  [n,nv] = size(Bhat);
  X(:,1) = x0;
  for t = 1:tau,
    X(:,t+1) = Ahat*X(:,t)+Bhat*randn(k,1);
    I(:,t) = Chat*X(:,t)+Ymean;
  end;
```

Experimental Results

- By learning the parameters using a finite number of sequence. Our system can then help us synthesize *infinite sequences*



Experimental Results

- Top: The original sequence of water waves
- Left: As you increase the number of images in your original sequence, you are better at reconstructing the original sequence with your system
- Right: extrapolation error reduces as number of time steps increases

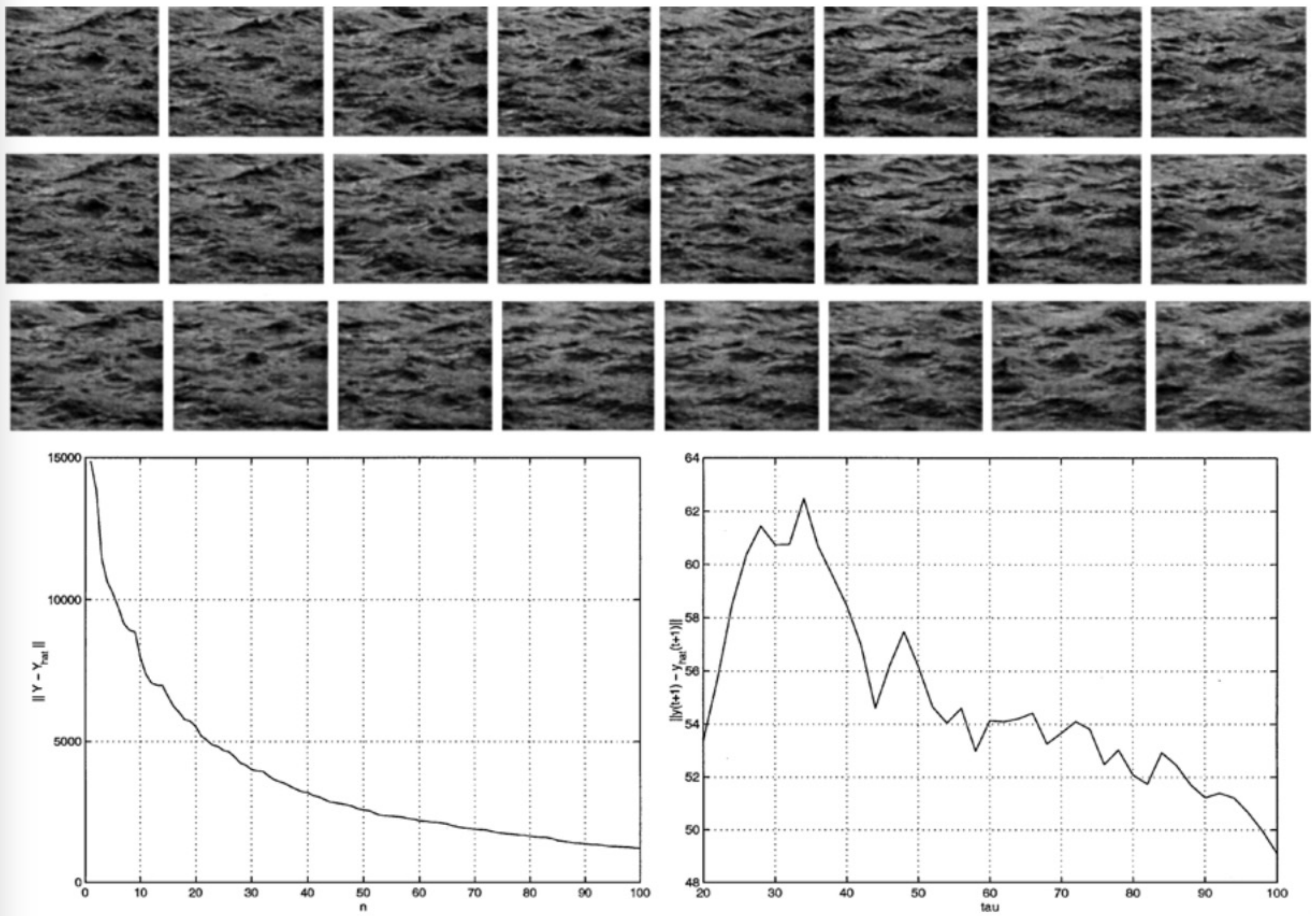


Figure 3. River. From top to bottom: Samples of the original sequence, corresponding samples of the compressed sequence (compression ratio: 2.53), samples of extrapolated sequence (using $n = 50$ components, $\tau = 120$, $m = 170 \times 115$), compression error as a function of the dimension of the state space n, and extrapolation error as a function of the length of the training set τ. The data set used comes from the *MIT Temporal Texture database*.

https://www.youtube.com/watch?v=bXHpOodkUv0