Planning and Optimal Control

3. State Space Models

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Syllabus

		A. Models for Sequential Data
Tue. 22.10.	(1)	1. Markov Models
Tue. 29.10.	(2)	2. Hidden Markov Models
Tue. 5.11.	(3)	3. State Space Models
Tue. 12.11.	(4)	3b. (ctd.)
		B. Models for Sequential Decisions
Tue. 19.11.	(5)	1. Markov Decision Processes
Tue. 26.11.	(6)	1b. (ctd.)
Tue. 3.12.	(7)	1c. (ctd.)
Tue. 10.12.	(8)	2. Monte Carlo and Temporal Difference Methods
Tue. 17.12.	(9)	3. Q Learning
Tue. 24.12.	—	— *Christmas Break* —
Tue. 7.1.	(10)	4. Policy Gradient Methods
Tue. 14.1.	(11)	tba
Tue. 21.1.	(12)	tba
Tue. 28.1.	(13)	8. Reinforcement Learning for Games
Tue. 4.2.	(14)	Q&A

Outline

Outline

Linear Transformation of a Gaussian

The linear transformation of a Gaussian is again a Gaussian:

$$p(x) := \mathcal{N}(x \mid \mu, \Sigma), \qquad \mu \in \mathbb{R}^N, \Sigma \in \mathbb{R}^{N \times N}$$

$$y := Ax + a, \qquad A \in \mathbb{R}^{M \times N}, a \in \mathbb{R}^M$$

$$\rightsquigarrow \quad p(y) = p_y(Ax + a) = \mathcal{N}(y \mid A\mu + a, A\Sigma A^T)$$

Proof:

$$\mathbb{E}(y) = \mathbb{E}(Ax + a) = A\mathbb{E}(x) + a = A\mu + a$$

$$\begin{aligned}\mathbb{V}(y) &= \mathbb{E}((y - \mathbb{E}(y))(y - \mathbb{E}(y))^T) \\ &= \mathbb{E}(A(x - \mu)(A(x - \mu))^T) \\ &= A\mathbb{E}((x - \mu)(x - \mu)^T)A^T \\ &= A\Sigma A^T\end{aligned}$$

Product of two Gaussian PDFs

The product of two Gaussian PDFs is again Gaussian:

$$\mathcal{N}(x \mid \mu_1, \Sigma_1) \cdot \mathcal{N}(x \mid \mu_2, \Sigma_2) \propto \mathcal{N}(x \mid \mu, \Sigma)$$

$$\text{with} \quad \Sigma := (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$$

$$\mu := \Sigma(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2)$$

Proof: elementary:

- $\log p$ is quadratic in x.
- complement squares.

Do not confuse this with

- $\mathcal{N}(x \mid \mu_1, \Sigma_1) \cdot \mathcal{N}(y \mid \mu_2, \Sigma_2) \propto \mathcal{N}(\begin{pmatrix} x \\ y \end{pmatrix} \mid \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix})$
- $p(x^2)$ for $x \sim \mathcal{N}(x \mid \mu, \Sigma)$.

Conditional Distributions of Multivariate Normals (Review)

Let y_A, y_B be jointly Gaussian

$$y := \begin{pmatrix} y_A \\ y_B \end{pmatrix} \sim \mathcal{N}(\begin{pmatrix} y_A \\ y_B \end{pmatrix} \mid \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix}, \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix})$$

then the **conditional distribution** is

$$p(y_B \mid y_A) = \mathcal{N}(y_B \mid \mu_{B|A}, \Sigma_{B|A})$$

with

$$\mu_{B|A} := \mu_B + \Sigma_{BA}\Sigma_{AA}^{-1}(y_A - \mu_A)$$
$$\Sigma_{B|A} := \Sigma_{BB} - \Sigma_{BA}\Sigma_{AA}^{-1}\Sigma_{AB}$$

Conditional Distr. of Multiv. Normals / Information Form

Let y_A, y_B be jointly Gaussian

$$y := \begin{pmatrix} y_A \\ y_B \end{pmatrix} \sim \mathcal{N}(\begin{pmatrix} y_A \\ y_B \end{pmatrix} \mid \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix}, \Lambda = \begin{pmatrix} \Lambda_{AA} & \Lambda_{AB} \\ \Lambda_{BA} & \Lambda_{BB} \end{pmatrix})$$

then the **conditional distribution** is

$$p(y_B \mid y_A) = \mathcal{N}(y_B \mid \mu_{B|A}, \Lambda_{B|A})$$

with

$$\mu_{B|A} := \mu_B + \Lambda_{BB}^{-1}\Lambda_{BA}(y_A - \mu_A)$$
$$\Lambda_{B|A} := \Lambda_{BB}$$

Linear Gaussian System

$$p(x) := \mathcal{N}(x \mid \mu_x, \Sigma_x)$$
$$p(y \mid x) := \mathcal{N}(y \mid Ax + b, \Sigma_y)$$

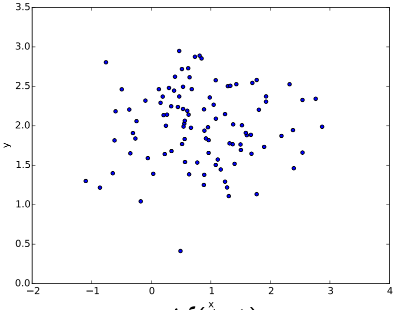
where

- x a multivariate Gaussian distributed random variable
  - $\mu_x \in \mathbb{R}^M, \Sigma_x \in \mathbb{R}^{M \times M}$
- y a multivariate Gaussian distributed random variable
  - $\mu_y := A\mu_x + b \in \mathbb{R}^L, \Sigma_y \in \mathbb{R}^{L \times L}$
  - $A \in \mathbb{R}^{L \times M}, b \in \mathbb{R}^L$
- y depends linearly on x

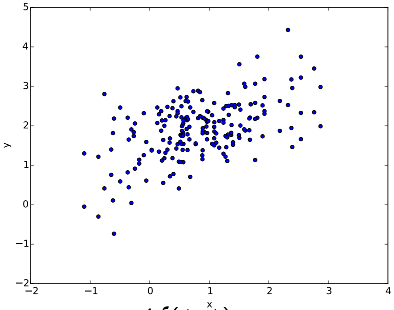
Linear Gaussian System

- LGS = multivariate multiple regression $(y|x)$ plus a Gaussian model for x.

- together, a generative Gaussian model.



$$x \sim \mathcal{N}(1, 1)$$
$$y \sim \mathcal{N}(2, 0.5)$$



$$x \sim \mathcal{N}(1, 1)$$
$$y \sim \mathcal{N}(x + 1, 0.5)$$

LGS as Joint Gaussian

An LGS

$$p(x) := \mathcal{N}(x \mid \mu_x, \Sigma_x)$$
$$p(y \mid x) := \mathcal{N}(y \mid Ax + b, \Sigma_y)$$

is equivalent to a jointly Gaussian distribution:

$$p\left(\begin{pmatrix} x \\ y \end{pmatrix}\right) = \mathcal{N}\left(\begin{pmatrix} \mu_x \\ A\mu_x + b \end{pmatrix}, \begin{pmatrix} \Sigma_x^{-1} + A^T\Sigma_y^{-1}A & -A^T\Sigma_y^{-1} \\ -\Sigma_y^{-1}A & \Sigma_y^{-1} \end{pmatrix}^{-1}\right)$$

LGS as Joint Gaussian / Information Form

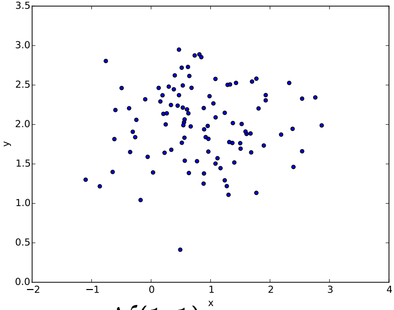
An LGS

$$p(x) := \mathcal{N}(x \mid \mu_x, \Lambda_x)$$
$$p(y \mid x) := \mathcal{N}(y \mid Ax + b, \Lambda_y)$$

is equivalent to a jointly Gaussian distribution:

$$p(\begin{pmatrix} x \\ y \end{pmatrix}) = \mathcal{N}(\begin{pmatrix} \mu_x \\ A\mu_x + b \end{pmatrix}, \begin{pmatrix} \Lambda_x + A^T\Lambda_y A & -A^T\Lambda_y \\ -\Lambda_y A & \Lambda_y \end{pmatrix})$$

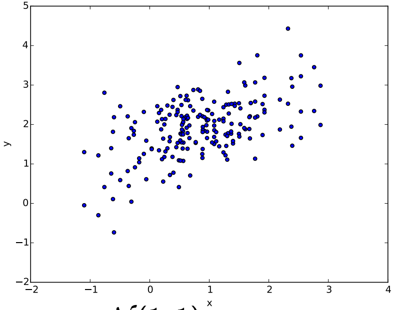
LGS as Joint Gaussian / Example



$$x \sim \mathcal{N}(1, 1)$$
$$y \sim \mathcal{N}(2, 0.5)$$

or equivalently

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathcal{N}(\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix})$$



$$x \sim \mathcal{N}(1, 1)$$
$$y \sim \mathcal{N}(x + 1, 0.5)$$

or equivalently

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathcal{N}(\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 1.5 \end{pmatrix})$$

Note: $\begin{pmatrix} 3 & -2 \\ -2 & 2 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 1 \\ 1 & 1.5 \end{pmatrix}$

LGS as Joint Gaussian / Proof

$$
\begin{aligned}
&\log p(x,y)\\
&= \log p(x) + \log p(y \mid x)\\
&\propto (x-\mu_x)^T \Lambda_x (x-\mu_x) + (y - Ax - b)^T \Lambda_y (y - Ax - b)\\
&= (x-\mu_x)^T \Lambda_x (x-\mu_x)\\
&\quad + (y - A\mu_x - b - A(x-\mu_x))^T \Lambda_y (y - A\mu_x - b - A(x-\mu_x))\\
&= (x-\mu_x)^T (\Lambda_x + A^T \Lambda_y A)(x-\mu_x)\\
&\quad + (y - A\mu_x - b)^T \Lambda_y (y - A\mu_x - b)\\
&\quad - 2(y - A\mu_x - b)^T \Lambda_y A (x-\mu_x)\\
&= \begin{pmatrix} x-\mu_x \\ y - A\mu_x - b \end{pmatrix}^T \begin{pmatrix} \Lambda_x + A^T\Lambda_y A & -A^T \Lambda_y \\ -\Lambda_y A & \Lambda_y \end{pmatrix} \begin{pmatrix} x-\mu_x \\ y - A\mu_x - b \end{pmatrix}
\end{aligned}
$$

Note: With $\Lambda_x := \Sigma_x^{-1}, \Lambda_y := \Sigma_y^{-1}$ precision matrices.

Bayes Rule for Linear Gaussian Systems

For an LGS

$$p(x) := \mathcal{N}(x \mid \mu_x, \Sigma_x)$$
$$p(y \mid x) := \mathcal{N}(y \mid Ax + b, \Sigma_y)$$

Bayes' Rule reads:

$$p(x \mid y) = \mathcal{N}(x \mid \mu_{x|y}, \Sigma_{x|y})$$
$$\text{with} \quad \Sigma_{x|y} := (\Sigma_x^{-1} + A^T \Sigma_y^{-1} A)^{-1}$$
$$\mu_{x|y} := \Sigma_{x|y} \left(A^T \Sigma_y^{-1} (y - b) + \Sigma_x^{-1} \mu_x \right)$$

Bayes Rule for Linear Gaussian Systems / Proof

- LGS is equivalent to joint Gaussian:

$$p(\begin{pmatrix} x \\ y \end{pmatrix}) = \mathcal{N}(\begin{pmatrix} \mu_x \\ A\mu_x + b \end{pmatrix}, \Lambda = \begin{pmatrix} \Lambda_x + A^T \Lambda_y A & A^T \Lambda_y \\ \Lambda_y A & \Lambda_y \end{pmatrix})$$

- conditional of a joint Gaussian:

$$p(x \mid y) = \mathcal{N}(x \mid \mu_{x|y}, \Lambda_{x|y})$$

with

$$\begin{aligned} \Lambda_{x|y} :=& \Lambda_{x,x} \\ \mu_{x|y} :=& \mu_x + \Lambda_{x,x}^{-1} \Lambda_{x,y}(y - \mu_y) \\ =& \Lambda_{x,x}^{-1}(\Lambda_{x,x}\mu_x + \Lambda_{x,y}(y - \mu_y)) \\ =& \Lambda_{x,x}^{-1}(\Lambda_x \mu_x + A^T \Lambda_y A \mu_x + A^T \Lambda_y (y - A\mu_x - b)) \\ =& \Lambda_{x,x}^{-1}(\Lambda_x \mu_x + A^T \Lambda_y (y - b)) \end{aligned}$$

Example: Inference from Noisy Measurements

- underlying quantity x
  - prior

$$p(x) := \mathcal{N}(x \mid \mu_x, \lambda_x^{-1})$$

- L noisy measurements $y_{1:L}$:

$$p(y_\ell \mid x) := \mathcal{N}(y_\ell \mid x, \lambda_y^{-1}), \quad \ell \in 1:L$$

  - scalar LGS: $N = M := 1$, $A := 1$ and $b := 0$: $\mu_y|x = Ax + b = x$
  - vector LGS: $N := 1, M := L$, $\mathbf{y} := y_{1:L}$, $\Lambda_y := \lambda_y \cdot I_{L\times L}$, $A := \mathbf{1}_L$, $\mathbf{b} := \mathbf{0}_L$,

$$\boldsymbol{\mu}_\mathbf{y}|\mathbf{x} = Ax + \mathbf{b} = x \cdot \mathbf{1}_L$$

Note: $I_{N\times N} := (\mathbb{I}(n = m))_{n,m\in 1:N}$ identity matrix.

Example: Inference from Noisy Measurements

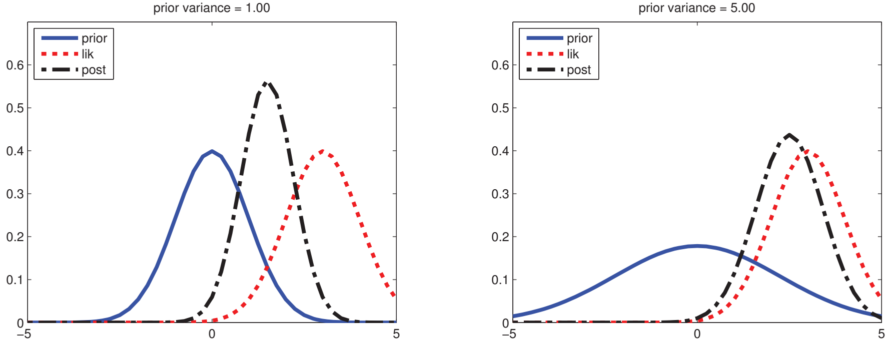
- vector LGS: $N = M := L$, $\mathbf{y} := y_{1:L}$, $\Lambda_y := \lambda_y \cdot I_{L\times L}$, $A := \mathbf{1}_L$, $\mathbf{b} := \mathbf{0}_L$,

$$\boldsymbol{\mu}_{\mathbf{y}}|\mathbf{x} = Ax + \mathbf{b} = x \cdot \mathbf{1}_L$$

- Bayes rule:

$$\begin{aligned}
p(x \mid y) &= \mathcal{N}(x \mid \mu_{x|y}, \Sigma_{x|y}) \\
\text{with} \quad \Sigma_{x|y}^{-1} &:= \Sigma_x^{-1} + A^T \Sigma_y^{-1} A \\
&= \lambda_x + L\lambda_y \\
\mu_{x|y} &:= \Sigma_{x|y} \left(A^T \Sigma_y^{-1} (y - b) + \Sigma_x^{-1} \mu_x \right) \\
&= (\lambda_x + L\lambda_y)^{-1} (\lambda_y \sum_{\ell=1}^{L} y_\ell + \lambda_x \mu_x) \\
&= \frac{\lambda_x}{\lambda_x + L\lambda_y} \mu_x + \frac{L\lambda_y}{\lambda_x + L\lambda_y} \frac{1}{L} \sum_{\ell=1}^{L} y_\ell
\end{aligned}$$

Example: Inference from Noisy Measurements



[source: Murphy 2012, p.121]

$$p(x) := \mathcal{N}(x \mid 0, \sigma^2 \in \{1,5\}), \quad p(y \mid x) := \mathcal{N}(y \mid x, 1), \qquad y = 3$$

prior: $p(x)$, MLE: $\mathcal{N}(x \mid y, 1)$, posterior: $p(x \mid y)$

Learning LGMs from Data

- data:

$$p(x) := \mathcal{N}(x \mid \mu_x, \Sigma_x)$$
$$p(y \mid x) := \mathcal{N}(y \mid Ax + b, \Sigma_y)$$

$$\mathcal{D} := \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subseteq \mathbb{R}^M \times \mathbb{R}^L$$

- multivariate linear regression of $y_{n,.}$ on $x_{n,.}$ (over all n):

$$X := (x_n^T)_{n=1:N} \in \mathbb{R}^{N \times M}, \quad Y := (y_n^T)_{n=1:N} \in \mathbb{R}^{N \times L}$$
$$\hat{A} := (X^T X)^{-1} X^T Y \quad (\text{for } \hat{b} := 0)$$
$$\hat{\Sigma}_y := \frac{1}{N-M} Y^T (I - X(X^T X)^{-1} X^T) Y$$

- multivariate normal density estimation of $x_{n,.}$ (over all n):

$$\hat{\mu}_x := \frac{1}{N} \mathbf{1}_{M \times N} X$$
$$\hat{\Sigma}_x := \frac{1}{N-1} (X - \hat{\mu}_x)(X - \hat{\mu}_x)^T$$

Learning LGMs from Data

$$\textbf{learn-lgm}(\mathcal{D} := \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subseteq \mathbb{R}^M \times \mathbb{R}^L):$$

$$\quad X := (x_n^T)_{n=1:N} \in \mathbb{R}^{N \times M}, \quad Y := (y_n^T)_{n=1:N} \in \mathbb{R}^{N \times L}$$

$$\quad \hat{\mu}_x := \frac{1}{N}\mathbf{1}_{M \times N} X$$

$$\quad \hat{\Sigma}_x := \frac{1}{N-1}(X - \hat{\mu}_x)(X - \hat{\mu}_x)^T$$

$$\quad \tilde{X} := (1_N, X)$$

$$\quad \tilde{A} := (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T Y$$

$$\quad \hat{b} := \tilde{A}_{.,1}, \quad \hat{A} := \tilde{A}_{.,2:}$$

$$\quad \hat{\Sigma}_y := \frac{1}{N-M} Y^T (I - \tilde{X}(\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T) Y$$

$$\quad \text{return} \;\; \hat{\mu}_x, \hat{\Sigma}_x, \hat{A}, \hat{b}, \hat{\Sigma}_y$$

Learning LGMs from Data with Uncertainties

- cases

$$x_n, \Sigma^x_n, y_n \quad \text{with } x_n^{\text{true}} \sim \mathcal{N}(x_n, \Sigma^x_n)$$

- normal equations:

$$\begin{aligned}
(\sum_n x_n^{\text{true}} x_n^{\text{true}\,T})\hat{A} &= \sum_n x_n^{\text{true}} y_n^T \quad |E(\ldots) \\
(\sum_n x_n x_n^T + \Sigma^x_n)\hat{A} &= \sum_n x_n y_n^T \\
\rightsquigarrow \quad \hat{A} &= (X^T X + \sum_n \Sigma^x_n)^{-1} X^T Y
\end{aligned}$$

Note: formula for A looks wrong. Where is the mistake?

Outline

1. Linear Gaussian Systems

2. State Space Models

3. Inference I: Kalman Filtering

4. Inference II: Kalman Smoothing

5. Learning via EM

State Space Model

$$z_t = g(z_{t-1}) \quad \textbf{transition model}$$

$$x_t = h(z_t) \quad \textbf{observation model}$$

$$z_t \in \mathbb{R}^K \quad \textbf{hidden state}$$

$$x_t \in \mathbb{R}^M \quad \textbf{observation}$$

- like HMM, but with continuous hidden state z_t
- g, h stochastic functions
  - $=$ parametric distributions:
    - parameters $=$ functions of the arguments

Linear-Gaussian State Space Model

$p(z_t \mid z_{t-1}) := \mathcal{N}(z_t \mid A_t z_{t-1} + a_{t-1}, \Sigma_{z,t})$	**transition model**
$p(x_t \mid z_t) := \mathcal{N}(x_t \mid B_t z_t + b_t, \Sigma_{x,t})$	**observation model**
$z_t \in \mathbb{R}^K$	**hidden state**
$x_t \in \mathbb{R}^M$	**observation**
$A_t \in \mathbb{R}^{K \times K}$	transition matrix at time t
$B_t \in \mathbb{R}^{M \times K}$	observation matrix at time t
$\Sigma_{z,t} \in \mathbb{R}^{K \times K}$	state/system noise at time t
$\Sigma_{x,t} \in \mathbb{R}^{M \times M}$	observation noise at time t

- transition and observation function is linear
  - bias term often dropped: $a_{t-1} := 0$, $b_t := 0$.
- state and observation noise is Gaussian
- also called **linear Gaussian system**

Stationary Linear-Gaussian State Space Model

$p(z_t \mid z_{t-1}) := \mathcal{N}(z_t \mid Az_{t-1}, \Sigma_z)$	**transition model**
$p(x_t \mid z_t) := \mathcal{N}(x_t \mid Bz_t, \Sigma_x)$	**observation model**
$z_t \in \mathbb{R}^K$	**hidden state**
$x_t \in \mathbb{R}^M$	**observation**
$A \in \mathbb{R}^{K\times K}$	transition matrix
$B \in \mathbb{R}^{M\times K}$	observation matrix
$\Sigma_z \in \mathbb{R}^{K\times K}$	state/system noise
$\Sigma_x \in \mathbb{R}^{M\times M}$	observation noise

- **stationary**, **time-invariant**:
  - transition and observation matrices do not depend on time t

Initial State Distribution

All models need to be complemented by an **initial state distribution**:

$$p(z_1) := \mathcal{N}(z_1 \mid \mu_{z_1}, \Sigma_{z_1})$$

Example

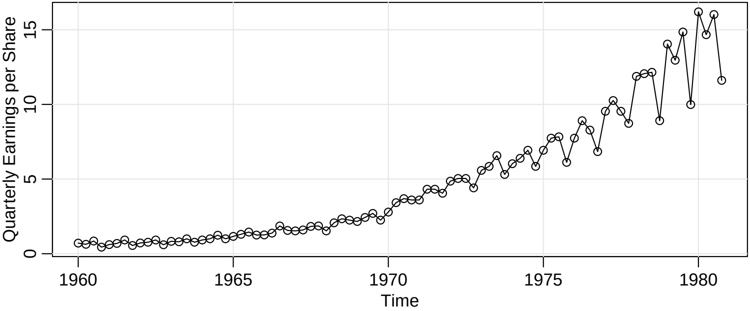


Fig. 1.1. *Johnson & Johnson quarterly earnings per share, 84 quarters, 1960-I to 1980-IV.*

[source: Shumway and Stoffer 2017, p.2]

Example

- decompose quarterly earnings E_t of a company
  into a trend T_t and
  a seasonal component S_t:

$$\begin{aligned} E_t &\sim \mathcal{N}(T_t + S_t, \sigma_E^2) \\ T_t &\sim \mathcal{N}(\beta T_{t-1}, \sigma_T^2) \\ S_t + S_{t-1} + S_{t-2} + S_{t-3} &\sim \mathcal{N}(0, \sigma_S^2) \end{aligned}$$

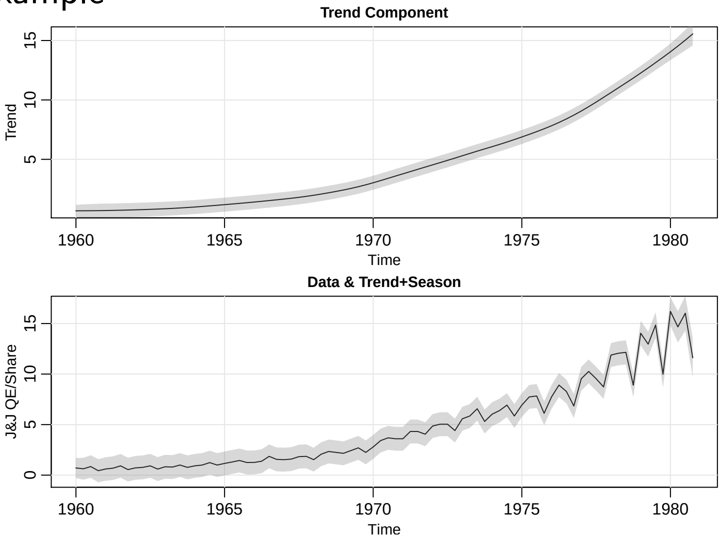
- as LGSSM:

$$x_t := E_t, \quad z_t := (T_t, S_t, S_{t-1}, S_{t-2})^T$$

$$B := (1, 1, 0, 0)^T, \quad b := 0, \quad \Sigma_x = (\sigma_E^2)$$

$$A := \begin{pmatrix} \beta & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad a := 0, \quad \Sigma_y := \text{diag}(\sigma_T^2, \sigma_S^2, 0, 0)$$

Example



[source: Shumway and Stoffer 2017, p.317]

Outline

Infering Posterior State Distributions $p(z_t \mid x_{1:t})$

Posterior hidden states can be computed sequentially:

$$p(z_t \mid x_{1:t}) = \mathcal{N}(z_t \mid \mu_t^\alpha, \Sigma_t^\alpha)$$

$$\text{with} \quad \Sigma_t^\alpha := ((A\Sigma_{t-1}^\alpha A^T)^{-1} + B^T \Sigma_x^{-1} B)^{-1}$$
$$\mu_t^\alpha := \Sigma_t^\alpha((A\Sigma_{t-1}^\alpha A^T)^{-1} A\mu_{t-1}^\alpha + B^T \Sigma_x^{-1} x_t)$$

$$\text{and} \quad \Sigma_1^\alpha := (\Sigma_{z_1}^{-1} + B^T \Sigma_x^{-1} B)^{-1}$$
$$\mu_1^\alpha := \Sigma_1^\alpha(\Sigma_{z_1}^{-1}\mu_{z_1} + B^T \Sigma_x^{-1} x_1)$$

Infering $p(z_t \mid x_{1:t})$ / Proof

- for $t = 1$:

$$p(x_1 \mid z_1) = \mathcal{N}(x_1 \mid Bz_1, \Sigma_x)$$
$$p(z_1) = \mathcal{N}(z_1 \mid \mu_{z_1}, \Sigma_{z_1})$$

Bayes rule $\rightsquigarrow$

$$p(z_1 \mid x_1) = \mathcal{N}(z_t \mid \mu_1^\alpha, \Sigma_1^\alpha)$$

$$\text{with} \quad \Sigma_1^\alpha := \Sigma_{z_1 \mid x_1} = (\Sigma_{z_1}^{-1} + B^T \Sigma_x^{-1} B)^{-1}$$
$$\mu_1^\alpha := \mu_{z_1 \mid x_1} = \Sigma_1^\alpha (\Sigma_{z_1}^{-1} \mu_{z_1} + B^T \Sigma_x^{-1} x_1)$$

- for $t > 1$:

$$p(x_t \mid z_t) = \mathcal{N}(x_t \mid Bz_t, \Sigma_x)$$
$$p(z_t \mid x_{1:t-1}) = \mathcal{N}(z_t \mid A\mu_{t-1}^\alpha, A\Sigma_{t-1}^\alpha A^T)$$

Bayes rule $\rightsquigarrow$

$$p(z_t \mid x_{1:t}) = \mathcal{N}(z_t \mid \mu_t^\alpha, \Sigma_t^\alpha)$$

$$\text{with} \quad \Sigma_t^\alpha := \Sigma_{z_t \mid x_{1:t}} = ((A\Sigma_{t-1}^\alpha A^T)^{-1} + B^T \Sigma_x^{-1} B)^{-1}$$
$$\mu_t^\alpha := \mu_{z_t \mid x_{1:t}} = \Sigma_t^\alpha ((A\Sigma_{t-1}^\alpha A^T)^{-1} A\mu_{t-1}^\alpha + B^T \Sigma_x^{-1} x_t)$$

Precomputing Posterior Variances

- Σ_t^α does not depend on the observations $x_{1:t}$
  - thus can be precomputed
- Σ_t^α depends on t only through the time since the initial state
  - if we assume states long after the initial state, use

$$\Sigma^\alpha := \lim_{t\rightarrow\infty} \Sigma_t^\alpha$$

    for all t.
  - Σ^α can be computed via fixpoint iterations

$$(\Sigma^\alpha)^{(0)} := (\Sigma_{z_1}^{-1} + B^T \Sigma_x^{-1} B)^{-1}$$
$$(\Sigma^\alpha)^{(t)} := ((A(\Sigma^\alpha)^{(t-1)} A^T)^{-1} + B^T \Sigma_x^{-1} B)^{-1}$$

Computing Variances with a Single Matrix Inversion

- in its previous form, computing variances Σ_t^α requires two matrix inversions:

$$\Sigma_t^\alpha := ((A\Sigma_{t-1}^\alpha A^T)^{-1} + B^T \Sigma_x^{-1} B)^{-1}$$

- more efficient computation with a single matrix inversion:

$$\begin{aligned}\Sigma_{t|t-1} &:= A\Sigma_{t-1}^\alpha A^T \\ \Sigma_t^\alpha &= (I - \underbrace{\Sigma_{t|t-1} B^T (\Sigma_x + B\Sigma_{t|t-1} B^T)^{-1}}_{=:K_t} B)\Sigma_{t|t-1} \\ &= (I - K_t B)\Sigma_{t|t-1}\end{aligned}$$

Proof: apply the matrix inversion lemma

$$(A - BD^{-1}C)^{-1} = (I + A^{-1}B(D - CA^{-1}B)^{-1}C)A^{-1}$$

to $(\Sigma_{t|t-1}^{-1} + B^T \Sigma_x^{-1} B)^{-1}$

Computing Means without Additional Matrix Inversion

- also the original mean formula contains a matrix inversion:

$$\mu_t^\alpha := \Sigma_t^\alpha(B^T\Sigma_x^{-1}x_t + \Sigma_{t|t-1}^{-1}A\mu_{t-1}^\alpha)$$

- can be simplified, reusing the matrix inversion from the variance:

$$\mu_{t|t-1} := A\mu_{t-1}^\alpha$$
$$\mu_t^\alpha = \mu_{t|t-1} + K_t(x_t - B\mu_{t|t-1})$$

proof: left term: using 2nd matrix inversion fomula

$$\begin{aligned}\Sigma_t^\alpha B^T\Sigma_x^{-1} \\ = (\Sigma_{t|t-1}^{-1} + B^T\Sigma_x^{-1}B)^{-1}B^T\Sigma_x^{-1} &= \Sigma_{t|t-1}B^T(\Sigma_x + B\Sigma_{t|t-1}B^T)^{-1} \\ &= K_t\end{aligned}$$

$$(A - BD^{-1}C)^{-1}BD^{-1} = A^{-1}B(D - CA^{-1}B)^{-1}$$

right term:

$$\Sigma_t^\alpha\Sigma_{t|t-1}^{-1} = (I - K_tB)\Sigma_{t|t-1}\Sigma_{t|t-1}^{-1} = (I - K_tB)$$

Kalman Filtering (Single Inversion)

- prediction step:

$$\Sigma_{t|t-1} := A\Sigma^{\alpha}_{t-1}A^T$$
$$\mu_{t|t-1} := A\mu^{\alpha}_{t-1}$$

- measurement step:

$$K_t := \Sigma_{t|t-1}B^T(\Sigma_x + B\Sigma_{t|t-1}B^T)^{-1}$$
$$\mu^{\alpha}_t = \mu_{t|t-1} + K_t(x_t - B\mu_{t|t-1})$$
$$\Sigma^{\alpha}_t := (I - K_tB)\Sigma_{t|t-1}$$

Kalman Filtering / Algorithm

infer-filtering-kalman$(x, A, \Sigma_z, B, \Sigma_x, \mu_{z_1}, \Sigma_{z_1})$:
  $T := |x|$
  $\Sigma_1^\alpha := (\Sigma_{z_1}^{-1} + B^T \Sigma_x^{-1} B)^{-1}$
  $\mu_1^\alpha := \Sigma_1^\alpha (B^T \Sigma_x^{-1} x_1 + \Sigma_{z_1}^{-1} \mu_{z_1})$
  for $t = 2, \ldots, T$:
    $\Sigma_{t|t-1} := A \Sigma_{t-1}^\alpha A^T$
    $\mu_{t|t-1} := A \mu_{t-1}^\alpha$
    $K_t := \Sigma_{t|t-1} B^T (\Sigma_x + B \Sigma_{t|t-1} B^T)^{-1}$
    $\mu_t^\alpha = \mu_{t|t-1} + K_t (x_t - B \mu_{t|t-1})$
    $\Sigma_t^\alpha := (I - K_t B) \Sigma_{t|t-1}$
  return $\mu_{1:T}^\alpha, \Sigma_{1:T}^\alpha$

where

- $x \in (\mathbb{R}^M)^*$ observed sequence
- $A, \Sigma_z, B, \Sigma_x, \mu_{z_1}, \Sigma_{z_1}$ linear-Gaussian state space model

yields $p(z_t \mid x_{1:t}) = \mathcal{N}(z_t \mid \mu_t^\alpha, \Sigma_t^\alpha), t = 1 : T$ PDFs of filtered latent states

Outline

Infering Posterior State Distributions $p(z_t \mid x_{1:T})$

$$p(z_t \mid x_{1:T}) = \mathcal{N}(z_t \mid \mu_t^\gamma, \Sigma_t^\gamma)$$
$$\mu_t^\gamma := \mu_t^\alpha + J_t(\mu_{t+1}^\gamma - \mu_{t+1|t})$$
$$\Sigma_t^\gamma := \Sigma_t^\alpha + J_t(\Sigma_{t+1}^\gamma - \Sigma_{t+1|t})J_t^T$$
$$J_t := \Sigma_t^\alpha A^T \Sigma_{t+1|t}$$ **backwards Kalman gain matrix**

with

$$p(z_{t+1} \mid x_{1:t}) = \mathcal{N}(z_t \mid \mu_{t+1|t}, \Sigma_{t+1|t})$$ **prediction**
$$\mu_{t+1|t} = A\mu_t^\alpha$$
$$\Sigma_{t+1|t} = A\Sigma_t^\alpha A^T + \Sigma_x$$

initialized by $p(z_T \mid x_{1:T})$, i.e.,

$$\mu_T^\gamma := \mu_T^\alpha, \quad \Sigma_T^\gamma := \Sigma_T^\alpha$$

Infering Posterior State Distr. $p(z_t \mid x_{1:T})$ / Proof

$$p(z_t \mid x_{1:T}) = \int_{z_{t+1}} p(z_{t+1} \mid x_{1:T})\, p(z_t \mid x_{1:t}, \cancel{x_{t+1:T}}, z_{t+1}) dz_{t+1}$$

$$p(z_t, z_{t+1} \mid x_{1:t}) = \mathcal{N}(\begin{pmatrix} z_t \\ z_{t+1} \end{pmatrix} \mid \begin{pmatrix} \mu_t^\alpha \\ \mu_{t+1|t} \end{pmatrix}, \begin{pmatrix} \Sigma_t^\alpha & \Sigma_t^\alpha A^T \\ A\Sigma_t^\alpha & \Sigma_{t+1|t} \end{pmatrix})$$

filtered two-slice posteriors

Gaussian conditioning yields

$$p(z_t \mid x_{1:t}, z_{t+1}) = \mathcal{N}(z_t \mid \mu_t^\alpha + J_t(z_{t+1} - \mu_{t+1|t}), \Sigma_t^\alpha - J_t \Sigma_{t+1|t} J_t^T)$$

and finally

$$\begin{aligned} \mu_t^\gamma &= \mathbb{E}(\mathbb{E}(z_t \mid z_{t+1}, x_{1:T}) \mid x_{1:T}) \\ &= \mathbb{E}(\mathbb{E}(z_t \mid z_{t+1}, x_{1:t}) \mid x_{1:T}) \\ &= \mathbb{E}(\mu_t^\alpha + J_t(z_{t+1} - \mu_{t+1|t}) \mid x_{1:T}) \\ &= \mu_t^\alpha + J_t(\mu_{t+1}^\gamma - \mu_{t+1|t}) \end{aligned}$$

Infering Posterior State Distr. $p(z_t \mid x_{1:T})$ / Proof

$$\begin{aligned}\Sigma_t^\gamma &= \mathbb{V}(\mathbb{E}(z_t \mid z_{t+1}, x_{1:T}) \mid x_{1:T}) + \mathbb{E}(\mathbb{V}(z_t \mid z_{t+1}, x_{1:T}) \mid x_{1:T}) \\ &= \ldots \\ &= \Sigma_t^\alpha + J_t(\Sigma_{t+1}^\gamma - \Sigma_{t+1|t})J_t^T\end{aligned}$$

Outline

Learning SSMs from Fully Observed Data

- just estimate
  - the LGS / multivar. linear regression $p(x_t \mid z_t)$,
  - the LGS / multivar. linear regression $p(z_{t+1} \mid z_t)$ and
  - the multivar. normal density $p(z_1)$

Learning LGMs from Fully Observed Data

learn-ssm-fully-observed$(\mathcal{D} := \{(x_1, z_1), (x_2, z_2), \ldots, (x_N, z_N)\} \subseteq (\mathbb{R}^M \times \mathbb{R}^K)^*)$:

$\quad \_, \_, \hat{B}, \hat{b}, \hat{\Sigma}_x :=$ learn-lgm$(\{(z_{n,t}, x_{n,t}) \mid n = 1 : N, t = 1 : T_n\})$

$\quad \_, \_, \hat{A}, \hat{a}, \hat{\Sigma}_z :=$ learn-lgm$(\{(z_{n,t}, z_{n,t+1}) \mid n = 1 : N, t = 1 : T_n - 1\})$

$\quad \hat{\mu}_{z_1}, \hat{\Sigma}_{z_1}, \_, \_, \_ :=$ learn-lgm$(\{(z_{n,1}, x_{n,1}) \mid n = 1 : N\})$

$\quad$ return $\hat{\mu}_{z_1}, \hat{\Sigma}_{z_1}, \hat{A}, \hat{a}, \hat{\Sigma}_z, \hat{B}, \hat{b}, \hat{\Sigma}_x$

Note: where $T_n := |x_n|$ denotes the length of sequence n.

Learning SSMs via EM

- E-step:
  estimate via Kalman smoothing:

$$p(z_{n,t} \mid x_{n,1:T}) = \mathcal{N}(\mu^\gamma_{n,t}, \Sigma^\gamma_{n,t})$$
$$p(z_{n,t+1}, z_{n,t} \mid x_{n,1:T}) = \mathcal{N}(\mu^\xi_{n,t,t+1}, \Sigma^\xi_{n,t,t+1})$$

- M-step:
  learn observation model $x = Bz + b$ from

$$(\mu^\gamma_{n,t}, \Sigma^\gamma_{n,t}, x_{n,t}) \mid n = 1:N, t = 1:T_n$$

  learn transition model $z_{t+1} = Az_t + a$ from

$$((\mu^\xi_{n,t,t+1})_{1:K}, (\mu^\xi_{n,t,t+1})_{K+1:2K}, \Sigma^\xi_{n,t,t+1}) \mid n = 1:N, t = 1:T_n - 1$$

  estimate starting density $p(z_1)$ from

$$(\mu^\gamma_{n,1}, \Sigma^\gamma_{n,1}) \mid n = 1:N$$

Note: for $\Sigma^\xi_{n,t,t+1}$ see Ghahramani/Hinton 1996b, eq. 34.

Summary

- Linear Gaussian Systems describe linear dependencies between continuous, normally distributed variables.
  - Continuous markov models.
- Linear Gaussian State Space Models (LGSSMs) describe linear dependencies between observed and latent, continuous normally distributed variables.
  - Continuous hidden markov models.
- For LGSSMs there exist simple algorithms to
  - infer the last latent state (**Kalman filtering**)
  - infer any intermediate latent state (**Kalman smoothing**)
  - forecast future observations (using Kalman filtering)
- LGSSMs can be learned via EM.
  (not covered by my slides currently.)

Further Readings

- Inference in jointly Gaussian distributions:
  - lecture Machine Learning 2, ch. A.2 Gaussian Processes
  - Murphy 2012, chapter 4.3.
- Linear Gaussian Systems:
  Murphy 2012, chapter 4.4.
- State Space Models:
  - Murphy 2012, chapter 18.
  - Shumway and Stoffer 2017, chapter 6.

References

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Springer International Publishing, 4 edition, 2017. ISBN 978-3-319-52451-1. doi: 10.1007/978-3-319-52452-8.