State space models

- Making some use of Murphy, chapter 18, as well as Bishop 13.3
- State space models are Like HMMs except that the states are continuous variables
- Example
  - A kicked soccer ball traveling under the influence of gravity (ignore air friction for now---this soccer match is on the moon).
  - One representation of state is position and velocity (all these are continuous variables)

State space models

- Notation
  - At a time, *t*, the state vector is $\mathbf{z}_t$
  - At each (discrete) time point, we make measurements $\mathbf{y}_t$
  - The system could also be influenced by a time varying control signal, which we will ignore in this course.

$$\mathbf{z}_t = g\left(\mathbf{z}_{t-1}, \varepsilon_t\right)$$

$$\mathbf{y}_t = h\left(\mathbf{z}_t, \delta_t\right)$$

where ε_t is the "system noise" and

δ_t is the "observation noise"

Linear dynamical systems (LDS)

- Special case of state-space models where the transition function g() is linear, and all random processes are Gaussian
  - Also known as linear-Gaussian SSM (LG-SSM)

$\mathbf{z}_t = A_t \mathbf{z}_{t-1} + \varepsilon_t$ where $\varepsilon_t \sim \mathcal{N}(0, Q_t)$ and A is a transition matrix

$\mathbf{y}_t = C_t \mathbf{z}_t + \delta_t$ where $\delta_t \sim \mathcal{N}(0, R_t)$

where ε_t is the "system noise" and δ_t is the "observation noise"

Often we assume that the parameters do not change over time.

This is known as a stationary model. Here,

$A_t = A \quad C_t = C \quad Q_t = Q \quad R_t = R$

Linear dynamical systems (LDS)

- For example, let **z** be the position in 2D of a hockey puck on the ice, moving with constant velocity.
- What is A?

Ignoring noise, we have

$\mathbf{X}_t = \mathbf{X}_{t-1} + \mathbf{V}_{t-1} \bullet \Delta t$

$\mathbf{V}_t = \mathbf{V}_{t-1}$

$$A = \begin{pmatrix} 1 & & \Delta t & \\ & 1 & & \Delta t \\ & & 1 & \\ & & & 1 \end{pmatrix}$$

Linear dynamical systems (LDS)

- For example, let **z** be the position in 2D of a hockey puck on the ice, moving with constant velocity.
- If the ice is rough, then **z** might be buffeted about. Then our system noise component becomes relevant

$$\mathbf{X}_t = \mathbf{X}_{t-1} + \mathbf{V}_{t-1} \bullet \Delta t \ + \ \varepsilon_t$$

$$\mathbf{V}_t = \mathbf{V}_{t-1} + \ \varepsilon_t$$

- (sometimes called random acceleration model)

Linear dynamical systems (LDS)

- Finally, the observations are a linear function of the state variable, with added Gaussian noise. But perhaps we only measure position. Then

$$\mathbf{y}_t = C\,\mathbf{z}_t + \delta_t$$

$$\text{Where} \quad \text{C=}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

LDS computational problems

Given data, what is the LDS (**learning**).

Given an LDS, what is the **distribution over the state** variables. Also, **how likely** are the observations, given the model (how good is the model).

Unlike an HMM, the Gaussian posterior for LDS means the most likely **state sequence** for the data is simply the most likely states computed from the previous.

LDS computational problems

Learning the LDS can be accomplished by EM in analogy with HMM, as well as other means.

Traditionally, given an LDS, the **distribution over the state** variables is computed using the Kalman filter and the Kalman smoother (like alpha and beta respectively).

Conventionally, the Kalman filter is analogous to computing the rescaled alphas

$$\hat{\alpha}(z_n) = p(z_n|x_1,...,x_n) = \frac{\alpha(z_n)}{p(x_1,...,x_n)}$$

and the Kalman smoother is analogous to computing the products

$$\gamma(z_n) = \hat{\alpha}(z_n)\hat{\beta}(z_n)$$

Recall that $\hat{\beta}(z_n) = \dfrac{p(x_{n+1},...,x_N|z_n)}{p(x_{n+1},...,x_N|x_1,...,x_n)}$

The complete log likelihood

In analogy with HMM, we have

$$\begin{aligned} p(X,Z|\theta) &= p(\mathbf{z}_1)\left[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1})\right]\prod_{m=1}^{N} p(\mathbf{x}_m|\mathbf{z}_m) \\ &= \mathcal{N}(\mathbf{z}_1|\boldsymbol{\mu}_0,\mathbf{V}_0)\left[\prod_{n=2}^{N}\mathcal{N}(\mathbf{z}_n|\mathbf{A}\,\mathbf{z}_{n-1},\ \Gamma)\right]\prod_{m=1}^{N}\mathcal{N}(\mathbf{x}_m|\mathbf{C}\,\mathbf{z}_m,\ \Sigma) \end{aligned}$$

Manipulating Gaussians

Recall that for multivariate Gaussians, if we partition the variables into two sets, **x** and **y**, then:

$$p(\mathbf{x}|\mathbf{y}) \sim \mathcal{N}(\quad)$$

$$p(\mathbf{x}) = \int_y p(\mathbf{x},\mathbf{y}) \sim \mathcal{N}(\quad)$$

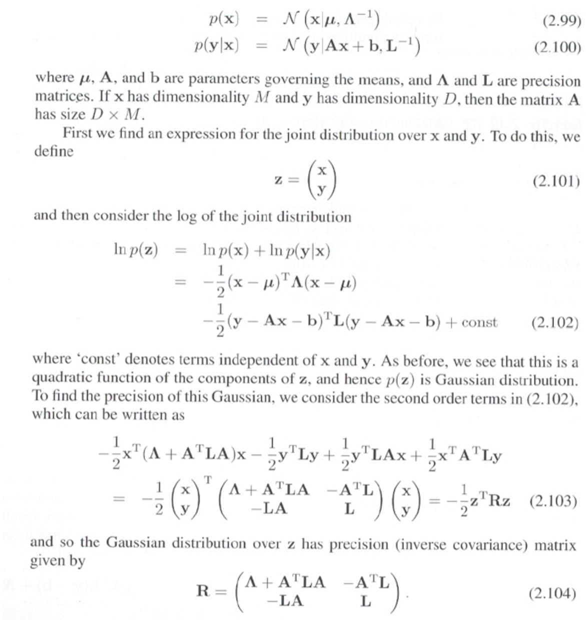
Manipulating Gaussians

In addition, if

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\Lambda^{-1})$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\,\mathbf{x}+\mathbf{b},\mathbf{L}^{-1})$$

then

$$p(\mathbf{x},\mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\quad)$$



$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right) \quad (2.99)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}\right) \quad (2.100)$$

where $\boldsymbol{\mu}$, $\mathbf{A}$, and $\mathbf{b}$ are parameters governing the means, and $\boldsymbol{\Lambda}$ and $\mathbf{L}$ are precision matrices. If $\mathbf{x}$ has dimensionality M and $\mathbf{y}$ has dimensionality D, then the matrix $\mathbf{A}$ has size $D \times M$.

First we find an expression for the joint distribution over $\mathbf{x}$ and $\mathbf{y}$. To do this, we define

$$\mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \quad (2.101)$$

and then consider the log of the joint distribution

$$\begin{aligned} \ln p(\mathbf{z}) &= \ln p(\mathbf{x}) + \ln p(\mathbf{y}|\mathbf{x}) \\ &= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Lambda}(\mathbf{x}-\boldsymbol{\mu}) \\ &\quad -\frac{1}{2}(\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{b})^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{b}) + \text{const} \qquad (2.102) \end{aligned}$$

where 'const' denotes terms independent of $\mathbf{x}$ and $\mathbf{y}$. As before, we see that this is a quadratic function of the components of $\mathbf{z}$, and hence $p(\mathbf{z})$ is Gaussian distribution. To find the precision of this Gaussian, we consider the second order terms in (2.102), which can be written as

$$\begin{aligned} &-\frac{1}{2}\mathbf{x}^{\mathrm{T}}(\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})\mathbf{x} - \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{L}\mathbf{y} + \frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{L}\mathbf{A}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{y} \\ &= -\frac{1}{2}\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A} & -\mathbf{A}^{\mathrm{T}}\mathbf{L} \\ -\mathbf{L}\mathbf{A} & \mathbf{L} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = -\frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{R}\mathbf{z} \qquad (2.103) \end{aligned}$$

and so the Gaussian distribution over $\mathbf{z}$ has precision (inverse covariance) matrix given by

$$\mathbf{R} = \begin{pmatrix} \boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A} & -\mathbf{A}^{\mathrm{T}}\mathbf{L} \\ -\mathbf{L}\mathbf{A} & \mathbf{L} \end{pmatrix}. \quad (2.104)$$

Manipulating Gaussians

More specifically (from Bishop p. 91), if

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \Lambda^{-1}\right)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\mathbf{A}\,\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}\right)$$

then

$$p\left(\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}\right) = \mathcal{N}\left(\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \middle| \begin{matrix} \boldsymbol{\mu} \\ \mathbf{A}\,\boldsymbol{\mu}+\mathbf{b} \end{matrix}, \mathbf{R}^{-1}\right)$$

where

$$\mathbf{R}^{-1} = \begin{pmatrix} \mathbf{A}+\mathbf{A}^{T}\mathbf{L}\mathbf{A} & -\mathbf{A}^{T}\mathbf{L} \\ -\mathbf{L}\mathbf{A} & \mathbf{L} \end{pmatrix}$$

Kalman filtering/smoothing

Using these rules, you can easily convince yourself that the joint probability of all the LDS variables is one big Gaussian.

The filtering/smoothing messages (alpha/beta) enable fast computation of the marginals.

We will do the alphas briefly

Kalman filtering

In analogy with the rescaled version of the alpha/beta algorithm:

$$c_n\hat{\alpha}(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n)\int\hat{\alpha}(\mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1})d\mathbf{z}_{n-1}$$

Denote $\hat{\alpha}(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \mathbf{V}_n)$ to get

$$\begin{aligned} c_n\mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \mathbf{V}_n) &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \Sigma)\int\mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{z}_{n-1})\mathcal{N}(\mathbf{z}_{n-1}|\boldsymbol{\mu}_{n-1}, \mathbf{V}_{n-1})d\mathbf{z}_{n-1} \\ &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \Sigma)\mathcal{N}(\mathbf{z}_n|\mathbf{A}\boldsymbol{\mu}_{n-1}, \mathbf{P}_{n-1}) \end{aligned}$$

where $\mathbf{P}_{n-1} = \mathbf{A}\mathbf{V}_{n-1}\mathbf{A}^{T} + \Gamma$, using rules about manipulating Gaussians.

Kalman filtering

Applying Gaussian manipulations and Matrix inversion formulas (see Bishop page 696):

$$
\begin{aligned}
\boldsymbol{\mu}_n &= \mathbf{A}\boldsymbol{\mu}_{n-1} + \mathbf{K}_n\left(\mathbf{x}_n - \mathbf{CA}\boldsymbol{\mu}_{n-1}\right)\\
\mathbf{V}_n &= \left(\mathbf{I} - \mathbf{K}_n\mathbf{C}\right)\mathbf{P}_{n-1}\\
c_n &= \mathcal{N}\left(x_n \middle| \mathbf{CA}\boldsymbol{\mu}_{n-1}, \mathbf{CP}_{n-1}\mathbf{C}^T + \Sigma\right)
\end{aligned}
$$

$$\text{where} \quad \mathbf{K}_n = \mathbf{P}_{n-1}\mathbf{C}^T\left(\mathbf{CP}_{n-1}\mathbf{C}^T + \Sigma\right)^{-1}$$

(This is the Kalman gain matrix)

Kalman filtering

For completeness, the initial values are:

$$
\begin{aligned}
\boldsymbol{\mu}_1 &= \mathbf{A}\boldsymbol{\mu}_0 + \mathbf{K}_1\left(\mathbf{x}_1 - \mathbf{C}\boldsymbol{\mu}_0\right)\\
\mathbf{V}_1 &= \left(\mathbf{I} - \mathbf{K}_1\mathbf{C}\right)\mathbf{V}_0\\
c_1 &= \mathcal{N}\left(x_1 \middle| \mathbf{C}\boldsymbol{\mu}_0, \mathbf{CV}_0\mathbf{C}^T + \Sigma\right)
\end{aligned}
$$

$$\text{where} \quad \mathbf{K}_1 = \mathbf{V}_0\mathbf{C}^T\left(\mathbf{CV}_0\mathbf{C}^T + \Sigma\right)^{-1}$$

Kalman filtering

Despite the tedious details, the result is somewhat intuitive. Consider the update of the mean:

$$\boldsymbol{\mu}_n = \underbrace{\mathbf{A}\boldsymbol{\mu}_{n-1}}_{\substack{\text{Believing the model,}\\ \text{ignoring the observation}}} + \mathbf{K}_n\left(\mathbf{x}_n - \underbrace{\mathbf{CA}\boldsymbol{\mu}_{n-1}}_{\substack{\text{Where we think we}\\ \text{should see } \mathbf{x}_n}}\right)$$

The new mean is the propagated previous one, with a correction using the new evidence.

The Kalman gain matrix is a factor of the relation between state variables and observations (C), and the variances. It can be computed in advance of data.

Kalman filtering/smoothing

Kalman filtering by itself makes sense if you are tracking an object on-line and in real time.

$$\boldsymbol{\mu}_n = \underbrace{\mathbf{A}\boldsymbol{\mu}_{n-1}}_{\substack{\text{Believing the model,}\\ \text{ignoring the observation}}} + \mathbf{K}_n\left(\mathbf{x}_n - \underbrace{\mathbf{CA}\boldsymbol{\mu}_{n-1}}_{\substack{\text{Where we think we}\\ \text{should see } \mathbf{x}_n}}\right)$$

However, the future observations can improve the estimates made by only considering the past.

The second pass computes the posterior as function of both.

This is the “smoother” which is analogous to the beta pass.

For details see the rest of Bishop 13.3.1.