A road map to more complex dynamic models

- **Discrete**
  - Y
  - X
  - Mixture model (e.g., mixture of multinomials)
  - HMM (for discrete sequential data, e.g., text)
  - Factorial HMM

- **Continuous**
  - Y
  - X
  - Mixture model (e.g., mixture of Gaussians)
  - HMM (for continuous sequential data, e.g., speech signal)
  - Factor analysis
  - State space model
  - Switching SSM
**Review:**

**A primer to multivariate Gaussian**

- Multivariate Gaussian density:
  \[
  p(\mathbf{x} | \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu)\right)
  \]

- A joint Gaussian:
  \[
  p\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} | \mu, \Sigma\right) = \mathcal{N}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mid \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)
  \]

- How to write down \( p(x_1), p(x_1 | x_2) \) or \( p(x_2 | x_1) \) using the block elements in \( \mu \) and \( \Sigma \)?
  - Formulas to remember:
    \[
    \begin{align*}
    p(x_2) &= \mathcal{N}(x_2 | m_2^m, V_2^m) \\
    p(x_1 | x_2) &= \mathcal{N}(x_1 | m_{12}, V_{12}) \\
    m_2^m &= \mu_2 \\
    V_2^m &= \Sigma_{22} \\
    m_{12} &= \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (x_2 - \mu_2) \\
    V_{12} &= \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}
    \end{align*}
    \]

---

**Review:**

**The matrix inverse lemma**

- Consider a block-partitioned matrix: \( M = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \)
  - First we diagonalize \( M \)
    \[
    \begin{bmatrix} I & -F \Sigma^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} \begin{bmatrix} I & 0 \\ -H \Sigma^{-1} G & I \end{bmatrix} = \begin{bmatrix} E - F \Sigma^{-1} H \Sigma^{-1} G & 0 \\ 0 & H \end{bmatrix}
    \]
  - Schur complement: \( MH = E - F \Sigma^{-1} G \)
  - Then we inverse, using this formula: \( XYZ = W \Rightarrow Y^{-1} = Z W^{-1} X \)
    \[
    M^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -H \Sigma^{-1} G & I \end{bmatrix} \begin{bmatrix} (M/H)^{-1} & -M \Sigma^{-1} H \Sigma^{-1} G \\ -H \Sigma^{-1} G (M/H)^{-1} H^{-1} & (M/H)^{-1} \end{bmatrix} = \begin{bmatrix} E^{-1} + E^{-1} F (M/E)^{-1} G E^{-1} & -E^{-1} F (M/E)^{-1} \\ -M \Sigma^{-1} H \Sigma^{-1} G (M/H)^{-1} H^{-1} & (M/E)^{-1} \end{bmatrix}
    \]
  - Matrix inverse lemma
    \[
    \left( E - F \Sigma^{-1} G \right)^{-1} = E^{-1} + E^{-1} F \left( H - G \Sigma^{-1} F \right)^{-1} \Sigma^{-1} G E^{-1}
    \]

Eric Xing
Review:
Some matrix algebra

- **Trace and derivatives**
  \[ \text{tr}[A] \overset{\text{def}}{=} \sum_i a_{ii} \]

- **Cyclical permutations**
  \[ \text{tr}[ABC] = \text{tr}[CAB] = \text{tr}[BCA] \]

- **Derivatives**
  \[ \frac{\partial}{\partial A} \text{tr}[BA] = B^T \]
  \[ \frac{\partial}{\partial A} \text{tr}[x^T A x] = \frac{\partial}{\partial A} \text{tr}[x x^T A] = xx^T \]

- **Determinants and derivatives**
  \[ \frac{\partial}{\partial A} \log|A| = A^T \]

Factor analysis

- An unsupervised linear regression model
  \[ p(x) = \mathcal{N}(x; \mathbf{0}, I) \]
  \[ p(y|x) = \mathcal{N}(y; \mu + \Lambda x, \Psi) \]

  where \( \Lambda \) is called a factor loading matrix, and \( \Psi \) is diagonal.

- Geometric interpretation

  To generate data, first generate a point within the manifold then add noise. Coordinates of point are components of latent variable.
Marginal data distribution

- A marginal Gaussian (e.g., \( p(x) \)) times a conditional Gaussian (e.g., \( p(y|x) \)) is a joint Gaussian.
- Any marginal (e.g., \( p(y) \)) of a joint Gaussian (e.g., \( p(x,y) \)) is also a Gaussian.
  - Since the marginal is Gaussian, we can determine it by just computing its mean and variance. (Assume noise uncorrelated with data.)

\[
E[y] = E[y]\mu + \Lambda x + \Psi \quad \text{where } W \sim \mathcal{N}(0, \Psi)
\]

\[
\mu = E[y]\mu + \Lambda x + \Psi \mu
\]

\[
V[y] = E[(y - \mu)(y - \mu)^T] = E[(\Lambda x + \Psi)(\Lambda x + \Psi)^T] = \Lambda \Lambda^T + \Psi
\]

FA = Constrained-Covariance Gaussian

- Marginal density for factor analysis (\( y \) is \( p \)-dim, \( x \) is \( k \)-dim):
  \[
p(y|\theta) = \mathcal{N}(y; \mu, \Lambda \Lambda^T + \Psi)
  \]
- So the effective covariance is the low-rank outer product of two long skinny matrices plus a diagonal matrix:

\[
\text{Cov}[y] = \Lambda \Lambda^T + \Psi
\]
- In other words, factor analysis is just a constrained Gaussian model. (If \( \Psi \) were not diagonal then we could model any Gaussian and it would be pointless.)
FA joint distribution

- Model
  \[ p(x) = \mathcal{N}(x; \mathbf{0}, I) \]
  \[ p(y|x) = \mathcal{N}(y; \mu + \Lambda x, \Psi) \]

- Covariance between \( x \) and \( y \)
  \[ \text{Cov}[X,Y] = E[XX'] - E[X]E[X'] = E[X(X + \Lambda X + W - \mu Y)] \]
  \[ = E[XX']I + XX' \Psi + WW' + \mu \mu' - \mu \mu' \]
  \[ = \Lambda I + \Psi \]

- Hence the joint distribution of \( x \) and \( y \):
  \[ p\begin{bmatrix} x \\ y \end{bmatrix} = \mathcal{N}\begin{bmatrix} x \\ y \end{bmatrix} \mid \begin{bmatrix} 0 \\ \mu \\ I \\ \Lambda \Lambda^T + \Psi \end{bmatrix} \]

- Assume noise is uncorrelated with data or latent variables.

Inference in Factor Analysis

- Apply the Gaussian conditioning formulas to the joint distribution we derived above, where
  \[ \Sigma_{11} = I \]
  \[ \Sigma_{12} = \Sigma_{12}^T = \Lambda I \]
  \[ \Sigma_{22} = (\Lambda \Lambda^T + \Psi) \]

we can now derive the posterior of the latent variable \( x \) given observation \( y \), \( p(x|y) = \mathcal{N}(x \mid m_{12}, V_{12}) \), where

\[ m_{12} = \mu_1 + \Sigma_{12}^{-1}(y - \mu_2) \]
\[ V_{12} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{12} \]
\[ = \Lambda \Sigma_{22}^{-1} \Lambda^T (y - \mu) \]
\[ = I - \Lambda \Sigma_{22}^{-1} \Lambda + \Psi \]

Applying the matrix inversion lemma

\[ (EFG'E^T)^{-1} = E^{-1} + E^{-1}F[HGE^{-1}F]^TGE^{-1} \]

\[ \Rightarrow \]
\[ V_{12} = (I + \Lambda \Sigma_{22}^{-1} \Lambda)^{-1} \]
\[ m_{12} = V_{12} \Lambda \Sigma_{22}^{-1} (y - \mu) \]

Here we only need to invert a matrix of size \(|x| \times |x|\), instead of \(|y| \times |y|\).
Geometric interpretation: inference is linear projection

- The posterior is:
  \[ p(x|y) = \mathcal{N}(x; m_{12}, V_{12}) \]
  \[ V_{12} = (I + \Lambda^T \Psi^{-1} \Lambda)^{-1} \]
  \[ m_{12} = V_{12} \Lambda^T \Psi^{-1} (y - \mu) \]

- Posterior covariance does not depend on observed data \( y \)!
- Computing the posterior mean is just a linear operation:

\[
\mu_{12} = V_{12} \Lambda^T \Psi^{-1} (y - \mu)
\]

EM for Factor Analysis

- Incomplete data log likelihood function (marginal density of \( y \))
  \[
  \ell(\theta, \mathcal{D}) = -\frac{N}{2} \log \det \Lambda^T + \Psi - \frac{1}{2} \sum_n (y_n - \mu)^T \left( \Lambda \Lambda^T + \Psi \right)^{-1} (y_n - \mu)
  \]
  \[
  = -\frac{N}{2} \log \det \Lambda^T + \Psi - \frac{1}{2} tr \left[ \left( \Lambda \Lambda^T + \Psi \right)^{-1} S \right]
  \]
  where \( S = \sum_n (y_n - \mu) (y_n - \mu)^T \)

  - Estimating \( m \) is trivial: \( \hat{\mu} = \frac{1}{N} \sum_n y_n \)
  - Parameters \( \Lambda \) and \( \Psi \) are coupled nonlinearly in log-likelihood

- Complete log likelihood
  \[
  \ell_c(\theta, \mathcal{D}) = \sum_n \log p(x_n, y_n) = \sum_n \log p(x_n) + \log p(y_n | x_n)
  \]
  \[
  = -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n x_n^T \Psi^{-1} x_n - \frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n (y_n - \Lambda x_n)^T \Psi^{-1} (y_n - \Lambda x_n)
  \]
  \[
  = -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n tr [x_n x_n^T] - \frac{N}{2} \log |\Psi| - \frac{1}{2} tr [S \Psi^{-1}]
  \]
  where \( S = \frac{1}{N} \sum_n (y_n - \Lambda x_n) (y_n - \Lambda x_n)^T \)
E-step for Factor Analysis

- Compute $\{z(\theta, D)\}_{k=1}^K$

$$
\{z(\theta, D)\} = -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_{n=1}^{N} \left[ X_nX_n^T \right] - \frac{N}{2} tr[S]^{-1}
$$

$$
\langle S \rangle = \frac{1}{N} \sum_{n=1}^{N} \left( y_nX_n^T - y_n \langle X_n^T \rangle \Lambda^T - \Lambda \langle X_n^T \rangle y_n^T + \Lambda \langle X_n \rangle \Lambda^T \right)
$$

$$
\langle X_n \rangle = E[X_n | y_n]
$$

$$
\langle X_n, X_n' \rangle = \text{Var}[X_n | y_n] + E[X_n | y_n]E[X_n | y_n]'
$$

- Recall that we have derived:

$$
V_{12} = \left(I + \Lambda^T \Psi^{-1} \Lambda \right)^{-1}
$$

$$
m_{12} = V_{12} \Lambda^T \Psi^{-1} \left( y - \mu \right)
$$

$$
\Rightarrow \langle X_n \rangle = m_{x_n|y_n} = V_{12} \Lambda^T \Psi^{-1} \left( y_n - \mu \right)
$$

$$
\langle X_n, X_n' \rangle = V_{12} + m_{x_n|y_n} m_{x_n'|y_n}
$$

M-step for Factor Analysis

- Take the derivatives of the expected complete log likelihood wrt. parameters.

  - Using the trace and determinant derivative rules:

$$
\frac{\partial}{\partial \psi^{-1}} \langle \zeta \rangle = \frac{\partial}{\partial \psi^{-1}} \left( -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_{n=1}^{N} \left[ X_nX_n^T \right] - \frac{N}{2} tr[S]^{-1} \right)
$$

$$
= -\frac{N}{2} \psi^{-1} \langle S \rangle
$$

$$
\Rightarrow \psi^{-1} = \langle S \rangle
$$

$$
\frac{\partial}{\partial \Lambda} \langle \zeta \rangle = \frac{\partial}{\partial \Lambda} \left( -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_{n=1}^{N} \left[ X_nX_n^T \right] - \frac{N}{2} tr[S]^{-1} \right)
$$

$$
= -\frac{N}{2} \psi^{-1} \frac{\partial}{\partial \Lambda} \langle S \rangle
$$

$$
= -\frac{N}{2} \psi^{-1} \left( \sum_{n=1}^{N} \langle y_nX_n^T \rangle - \sum_{n=1}^{N} \langle X_n \rangle \Lambda^T - \Lambda \sum_{n=1}^{N} \langle X_n^T \rangle y_n^T + \Lambda \sum_{n=1}^{N} \langle X_n \rangle \Lambda^T \right)
$$

$$
= \psi^{-1} \sum_{n=1}^{N} \langle y_nX_n^T \rangle - \psi^{-1} \Lambda \sum_{n=1}^{N} \langle X_n \rangle
$$

$$
\Rightarrow \Lambda^{-1} = \left( \sum_{n=1}^{N} \langle X_n \rangle \right)^{-1}
$$
Model Invariance and Identifiability

- There is degeneracy in the FA model.
- Since $\Lambda$ only appears as outer product $\Lambda \Lambda^T$, the model is invariant to rotation and axis flips of the latent space.
- We can replace $\Lambda$ with $\Lambda Q$ 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 of the parameters. An infinite number of parameters all give the ML score!
- Such models are called un-identifiable since two people both fitting ML parameters to the identical data will not be guaranteed to identify the same parameters.
Independent Components Analysis (ICA)

- ICA is similar to FA, except it assumes the latent source has non-Gaussian density.
- Hence ICA can extract higher order moments (not just second order).
- It is commonly used to solve blind source separation (cocktail party problem).
- Independent Factor Analysis (IFA) is an approximation to ICA where we model the source using a mixture of Gaussians.

A road map to more complex dynamic models

- Mixture model e.g., mixture of multinomials
  - HMM (for discrete sequential data, e.g., text)
  - Factorial HMM
- Mixture model e.g., mixture of Gaussians
  - HMM (for continuous sequential data, e.g., speech signal)
- Factor analysis
  - State space model
  - Switching SSM
State space models (SSM)

- A sequential FA or a continuous state HMM

\[
\begin{align*}
X_t & \rightarrow X_{t-1} \rightarrow \ldots \\
Y_t & \rightarrow Y_{t-1} \rightarrow \ldots \\
Y_N & \rightarrow \ldots \\
X_0 & \sim \mathcal{N}(0; \Sigma_x) \\
y_t & = Cx_{t-1} + \nu_t \\
w_t & \sim \mathcal{N}(0; Q), \quad \nu_t \sim \mathcal{N}(0, R) \\
x_t & = Ax_{t-1} + Gw_t
\end{align*}
\]

This is a linear dynamic system.

- In general,

\[
\begin{align*}
x_t &= f(x_{t-1}) + Gw_t \\
y_t &= g(x_{t-1}) + \nu_t
\end{align*}
\]

where \( f \) is an (arbitrary) dynamic model, and \( g \) is an (arbitrary) observation model.

The inference problem

- Filtering \( \Rightarrow \) given \( y_1, \ldots, y_t \), estimate \( x_t \)

  The Kalman filter is a way to perform exact online inference (sequential Bayesian updating) in an LDS. It is the Gaussian analog of the forwards algorithm for HMMs:

\[
p(X_t = i \mid y_t) \propto p(y_t \mid X_t = i) \sum_j p(X_t = i \mid X_{t-1} = j) \alpha_{t-1}^j
\]

- Smoothing \( \Rightarrow \) given \( y_1, \ldots, y_T \), estimate \( x_t (t<T) \)

  The Rauch-Tung-Strievel smoother is a way to perform exact off-line inference in an LDS. It is the Gaussian analog of the forwards-backwards algorithm:

\[
p(X_t = i \mid y_{1:T}) \propto \alpha_t^i \beta_t^i
\]
Online vs offline inference

- filtering: \( P(X(0) | y(1:t)) \)
- Viterbi: \( \text{argmax } P(x(1:t) | y(1:t)) \)
- prediction: \( P(X(t+\Delta t) | y(1:t)) \)
- fixed-lag smoothing: \( P(X(t-\tau) | y(1:t)) \)
- fixed interval smoothing (offline): \( P(X(t) | y(1:T)) \)

LDS for 2D tracking

- Dynamics: new position = old position + \( \Delta \times \text{velocity} + \text{noise} \)
  (constant velocity model, Gaussian noise)

\[
\begin{bmatrix}
    x^1_t \\
    x^2_t \\
    x^1_{t-1} \\
    x^2_{t-1}
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 1 & 0 \\
    0 & 1 & 0 & 1 \\
    0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x^1_t \\
    x^2_t \\
    x^1_{t-1} \\
    x^2_{t-1}
\end{bmatrix} + \text{noise}
\]

- Observation: project out first two components (we observe Cartesian position of object - linear!)

\[
\begin{bmatrix}
    y^1_t \\
    y^2_t
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
    x^1_t \\
    x^2_t \\
    x^1_{t-1} \\
    x^2_{t-1}
\end{bmatrix} + \text{noise}
\]
2D tracking

2D filtering

2D smoothing

Kalman filtering in the brain?
Kalman filtering derivation

- Since all CPDs are linear Gaussian, the system defines a large multivariate Gaussian.
  - Hence all marginals are Gaussian.
  - Hence we can represent the belief state $p(X_t|y_1:t)$ as a Gaussian with mean $\tilde{x}_{t|t} = E(X_t | y_1, \ldots, y_t)$ and covariance $P_{t|t} = E(X_tX_t^T | y_1, \ldots, y_t)$.
  - It is common to work with the inverse covariance (precision) matrix $P_{t|t}^{-1}$; this is called information form.

- Kalman filtering is a recursive procedure to update the belief state:
  - Predict step: compute $p(X_{t+1}|y_1:t)$ from prior belief $p(X_t|y_1:t)$ and dynamical model $p(X_{t+1}|X_t)$ --- time update
  - Update step: compute new belief $p(X_{t+1}|y_1:t+1)$ from prediction $p(X_{t+1}|y_1:t)$, observation $y_{t+1}$ and observation model $p(y_{t+1}|X_{t+1})$ --- measurement update

Predict step

- Dynamical Model: $x_{t+1} = Ax_t + Gw_t$, $w_t \sim \mathcal{N}(0, Q)$
  - One step ahead prediction of state:
    $$\tilde{x}_{t+1|t} = E(X_{t+1} | y_1, \ldots, y_t) = A\tilde{x}_{t|t}$$
    $$P_{t+1|t} = E((X_{t+1} - \tilde{x}_{t+1|t})(X_{t+1} - \tilde{x}_{t+1|t})^T | y_1, \ldots, y_t)$$
    $$= E(AX_t + Gw_t - \tilde{x}_{t+1|t})(AX_t + Gw_t - \tilde{x}_{t+1|t})^T | y_1, \ldots, y_t)$$
    $$= AP_{t|t}A^T + GQG^T$$

- Observation model: $y_t = Cx_t + v_t$, $v_t \sim \mathcal{N}(0, R)$
  - One step ahead prediction of observation:
    $$E(Y_{t+1} | y_1, \ldots, y_t) = E(CX_{t+1} + v_{t+1} | y_1, \ldots, y_t) = C\tilde{x}_{t+1|t}$$
    $$E((Y_{t+1} - \tilde{Y}_{t+1})(Y_{t+1} - \tilde{Y}_{t+1})^T | y_1, \ldots, y_t) = CP_{t+1|t}C^T + R$$
    $$E((y_{t+1} - \tilde{Y}_{t+1})(X_{t+1} - \tilde{x}_{t+1|t})^T | y_1, \ldots, y_t) = CP_{t+1|t}$$
Update step

- Summarizing results from previous slide, we have
  \[ p(X_{t+1}, Y_{t+1} | Y_{1:t}) \sim N(m_{t+1}, V_{t+1}) \], where
  \[
  m_{t+1} = \begin{pmatrix}
  \hat{x}_{t+1 | t} \\
  C\hat{x}_{t+1 | t}
  \end{pmatrix}, \quad
  V_{t+1} = \begin{pmatrix}
  P_{t+1 | t} & P_{t+1 | t}^{T} \\
  CP_{t+1 | t} & CP_{t+1 | t}^{T} + R
  \end{pmatrix}
  \]
- Remember the formulas for conditional Gaussian distributions:
  \[
  p(x_1 | x_2, \mu, \Sigma) = \mathcal{N}(x_1 | \mu_1 + \Sigma_{12} \Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21})
  \]
  \[
  m_n = \mu_2, \quad V_n = \Sigma_{22}
  \]
  \[
  m_n = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1}(x_2 - \mu_2), \quad V_n = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}
  \]

Kalman Filter

- Measurement updates:
  \[
  \hat{x}_{t+1 | t} = \hat{x}_{t+1 | t} + K_{t+1}(y_{t+1} - C\hat{x}_{t+1 | t})
  \]
  \[
  P_{t+1 | t+1} = P_{t+1 | t} - KCP_{t+1 | t}
  \]
  where \( K_{t+1} \) is the Kalman gain matrix
  \[
  K_{t+1} = P_{t+1 | t} C^T (CP_{t+1 | t} C^T + R)^{-1}
  \]
- Time updates:
  \[
  \hat{x}_{t+1 | t} = A\hat{x}_{t | t}
  \]
  \[
  P_{t+1 | t} = A P_{t | t} A + GQG^T
  \]
- \( K_t \) can be pre-computed (since it is independent of the data).
Example of KF in 1D

- Consider noisy observations of a 1D particle doing a random walk:
  \[ x_{t+1} = x_t + w, \quad w \sim \mathcal{N}(0, \sigma_w) \]
  \[ z_t = x_t + v, \quad v \sim \mathcal{N}(0, \sigma_z) \]

- KF equations:
  \[ P_{t+1} = AP_tA^T + GQG^T = \sigma_x + \sigma_z \]
  \[ \hat{x}_{t+1} = A\hat{x}_t + K(z_t - C\hat{x}_t) \]
  \[ K = P_{t+1}C^T(CP_{t+1}C^T + R)^{-1} = \frac{(\sigma_z + \sigma_x)\sigma_z}{\sigma_x + \sigma_z + \sigma_z} \]
  \[ P_{t+1} = P_{t+1} - KCP_{t+1} \]

KF intuition

- The KF update of the mean is
  \[ \hat{x}_{t+1} = \hat{x}_t + K_t(z_t - C\hat{x}_t) \]
  
- the term \((z_t - C\hat{x}_t)\) is called the innovation

- New belief is convex combination of updates from prior and observation, weighted by Kalman Gain matrix:
  \[ K_t = P_{t+1}C^T(CP_{t+1}C^T + R)^{-1} \]

- If the observation is unreliable, \(\sigma_z\) (i.e., \(R\)) is large so \(K_{t+1}\) is small, so we pay more attention to the prediction.

- If the old prior is unreliable (large \(\sigma_x\)) or the process is very unpredictable (large \(\sigma_x\)), we pay more attention to the observation.
KF, RLS and LMS

- The KF update of the mean is
  \[ \hat{x}_{t+1} = A\hat{x}_t + K_{t+1}(y_{t+1} - C\hat{x}_t) \]
- Consider the special case where the hidden state is a constant, \( x_t = \theta \), but the “observation matrix” \( C \) is a time-varying vector, \( C = x_T^T \).
  - Hence the observation model at each time slide, \( y_t = x_T^T \theta + \nu_t \), is a linear regression
- We can estimate recursively using the Kalman filter:
  \[ \hat{\theta}_{t+1} = \hat{\theta}_t + P_{t+1}^{-1}(y_{t+1} - x_T^T \hat{\theta}_t)x_T \]
  This is called the recursive least squares (RLS) algorithm.
- We can approximate \( P_{t+1}^{-1} \approx \eta_{t+1} \) by a scalar constant. This is called the least mean squares (LMS) algorithm.
- We can adapt \( \eta_t \) online using stochastic approximation theory.

Complexity of one KF step

- Let \( X_t \in \mathbb{R}^{N_x} \) and \( Y_t \in \mathbb{R}^{N_y} \).
- Computing \( P_{t+1} = AP_t A + GQG^T \) takes \( O(N_x^2) \) time, assuming dense \( P \) and dense \( A \).
- Computing \( K_{t+1} = P_{t+1} C^T (CP_{t+1} C^T + R)^{-1} \) takes \( O(N_y^3) \) time.
- So overall time is, in general, \( \max\{N_x^2, N_y^3\} \).
Rauch-Tung-Strievel smoother

\[ \hat{x}_{t|T} = \hat{x}_{t|T} + L_t (\hat{x}_{t+1|T} - \hat{x}_{t|T}) \]
\[ P_{t|T} = P_{t|T} + L_t (P_{t+1|T} - P_{t|T}) L_t^T \]

- General structure: KF results + the difference of the "smoothed" and predicted results of the next step
- Backward computation: Pretend to know things at t+1 — such conditioning makes things simple and we can remove this condition finally

The difficulty:

The trick:

\[
\begin{align*}
\hat{x}_{t+1|T} &= E[\hat{X}_{t+1} | Y_1, ..., Y_T] \\
\hat{V}_{t+1|T} &= E[V_{t+1} | Y_1, ..., Y_T]
\end{align*}
\]

Same for \( P_{t+1|T} \)

RTS derivation

- Following the results from previous slide, we need to derive
  \( p(X_{t+1}, X_t | Y_{1:t}) \sim \mathcal{N}(m, V) \), where
  \[
  m = \begin{pmatrix} \hat{x}_{t|T} \\ \hat{x}_{t+1|T} \end{pmatrix}, \quad V = \begin{pmatrix} P_{t|T} & P_{t+1|T} A^T \\ A P_{t|T} & P_{t+1|T} \end{pmatrix}
  \]
  - all the quantities here are available after a forward KF pass
- Remember the formulas for conditional Gaussian distributions:
  \[
  p(x_{1:2} | \mu_1, \Sigma_1) = \mathcal{N}(\mu_1, \Sigma_1), \quad p(x_{1:3} | \mu_1, \Sigma_1) = \mathcal{N}(\mu_1, \Sigma_1 + \Sigma_2)
  \]

  - The RTS smoother
  \[
  \hat{x}_{t+1|T} = E[\hat{X}_{t+1} | X_{t+1}, Y_1, ..., Y_T] \\
  = \hat{x}_{t+1|T} + L_t (\hat{x}_{t+1|T} - \hat{x}_{t|T})
  \]
  \[
  P_{t+1|T} = E[\hat{V}_{t+1} | X_{t+1}, Y_1, ..., Y_T] + E[\hat{V}_{t+1} | X_{t+1}, Y_1, ..., Y_T]
  = P_{t+1|T} + L_t (P_{t+1|T} - P_{t|T}) L_t^T
  \]
Learning SSMs

- Complete log likelihood
  \[ \ell_\theta(D) = \sum_t \log p(x_t, y_t) = \sum_t \log p(x_t) + \sum_t \sum_{x_{t-1}} \log p(x_t | x_{t-1}) + \sum_t \sum_{y_{t+1}} \log p(y_t | x_t) \]
  \[ = \ell_{\theta}(X; \Sigma_0) + f_t[\{X_t, X_{t-1}, X_{t+1}, Y_T : \forall T \}, A, Q, G] + f_t[\{X_t, X_{t-1}, X_{t+1}, Y_T : \forall T \}, C, R] \]

- EM
  - E-step: compute \( \langle X_t, X_{t-1}^*, \langle X_s, X_T^* \rangle, \langle X_t \rangle | x_{t-1}, \ldots, y_T \rangle \)
    these quantities can be inferred via KF and RTS filters, etc.,
    \(e.g., \langle X_t, X_{t-1} \rangle = \text{var}(X_t, X_{t-1}) + \mathbb{E}(X_t)^2 = \tilde{p}_{x_t} + x_{t-1}^2\)
  - M-step: MLE using
    \[ \langle \ell(\theta, D) \rangle = f_t[\{x_t, \Sigma_0\}] + f_t[\{X_t, X_{t-1}^*, \langle X_s, X_T^* \rangle, \langle X_t \rangle : \forall T \}, A, Q, G] + f_t[\{X_t, X_{t-1}^*, \langle X_t \rangle : \forall T \}, C, R] \]
    c.f., M-step in factor analysis

Nonlinear systems

- In robotics and other problems, the motion model and the observation model are often nonlinear:
  \[ x_t = f(x_{t-1}) + w_t , \quad y_t = g(x_t) + v_t \]
- An optimal closed form solution to the filtering problem is no longer possible.
- The nonlinear functions \( f \) and \( g \) are sometimes represented by neural networks (multi-layer perceptrons or radial basis function networks).
- The parameters of \( f \) and \( g \) may be learned offline using EM, where we do gradient descent (back propagation) in the M step, c.f. learning a MRF/CRF with hidden nodes.
- Or we may learn the parameters online by adding them to the state space: \( x_t' = (x_t, \theta) \). This makes the problem even more nonlinear.
Extended Kalman Filter (EKF)

- The basic idea of the EKF is to linearize $f$ and $g$ using a second order Taylor expansion, and then apply the standard KF.
  - i.e., we approximate a stationary nonlinear system with a non-stationary linear system.
    
    \[
    \begin{align*}
    \dot{x}_t &= f(\hat{x}_{t-1}) + A_{x_{t-1}}(x_{t-1} - \hat{x}_{t-1}) + w_t \\
    y_t &= g(\hat{x}_{t-1}) + C_{x_{t-1}}(x_{t-1} - \hat{x}_{t-1}) + v_t \\
    \end{align*}
    \]

    where $\hat{x}_{t-1} = f(\hat{x}_{t-1})$ and $A_\dot{x} = \frac{\partial f}{\partial x}\bigg|_{\hat{x}}$ and $C_\dot{x} = \frac{\partial g}{\partial x}\bigg|_{\hat{x}}$

- The noise covariance ($Q$ and $R$) is not changed, i.e., the additional error due to linearization is not modeled.