A road map to more complex dynamic models

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

- **Factorial HMM**

Reading: MJ-Chap. 13, 14, 15
Review:
A primer to multivariate Gaussian

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

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

- How to write down \( p(x_1), p(x_1|x_2) \) or \( p(x_1|x_2) \) using the block elements in \( \mu \) and \( \Sigma \)?
  - Formulas to remember:
    \[
p(x_2) = \mathcal{N}(x_2 \mid m_2, V_2) \quad p(x_1|x_2) = \mathcal{N}(x_1 \mid m_{12}, V_{12})
    \]
    \[
m_2 = \mu_2 \quad V_2 = \Sigma_{22}
    \]

Review:
The matrix inverse lemma

- Consider a block-partitioned matrix: \( M = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \)

- First we diagonalize \( M \)
  \[
  \begin{bmatrix}
  I & -FH^{-1} \\
  0 & I
  \end{bmatrix}
  \begin{bmatrix}
  E & F \\
  G & H
  \end{bmatrix}
  \begin{bmatrix}
  I & 0 \\
  0 & H^{-1}
  \end{bmatrix}
  =
  \begin{bmatrix}
  E - FH^{-1}G & 0 \\
  0 & H
  \end{bmatrix}
  \]
  - Schur complement: \( M/H = E - FH^{-1}G \)

- Then we inverse, using this formula: \( XZ = W \Rightarrow Y^{-1} = ZW^{-1}X \)
  \[
  M^{-1} = \begin{bmatrix}
  E & F \\
  G & H
  \end{bmatrix}^{-1} = \begin{bmatrix}
  I & 0 \\
  0 & H^{-1}
  \end{bmatrix}^{-1} \begin{bmatrix}
  E - FH^{-1}G & 0 \\
  0 & H
  \end{bmatrix}^{-1}
  = \begin{bmatrix}
  (M/H)^{-1} & - (M/H)^{-1}FH^{-1} \\
  -H^{-1}G(M/H)^{-1} & H^{-1} + H^{-1}G(M/H)^{-1}FH^{-1}
  \end{bmatrix}
  = \begin{bmatrix}
  E + E^{-1}F(M/E)^{-1}GE^{-1} & -E^{-1}F(M/E)^{-1} \\
  -(M/E)^{-1}GE^{-1} & (M/E)^{-1}
  \end{bmatrix}
  \]
  - Matrix inverse lemma
    \[
    \left( E - FH^{-1}G \right)^{-1} = E^{-1} + E^{-1}F(H - GE^{-1}F)^{-1}GE^{-1}
    \]
Review: Some matrix algebra

- Trace and derivatives
  \[ \text{tr}[A] \overset{\text{def}}{=} \sum_{\nu} a_{\nu} \]

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

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

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

Factor analysis

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

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

$$
= \mu + \Lambda E[\mu] + E[W]
$$

$$
= \mu + 0 + 0 = \mu
$$

$$
Var[Y] = E[(Y - \mu)(Y - \mu)^T]
$$

$$
= E[(\mu + \Lambda \mu + W - \mu)(\mu + \Lambda \mu + W - \mu)^T]
$$

$$
= E[\Lambda \mu W + \Lambda W^T]
$$

$$
= \Lambda E[WW^T]
$$

$$
= \Lambda \Lambda^T + \Psi
$$

FA = Constrained-Covariance Gaussian

- Marginal density for factor analysis ($y$ is $p$-dim, $x$ is $k$-dim):

$$
p(y \mid \theta) = \mathcal{N}(y; \mu, \Lambda \Lambda^T + \Psi)
$$

- 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 were not diagonal then we could model any Gaussian and it would be pointless.)
**FA joint distribution**

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

- **Covariance between x and y**
  \[
  \text{Cov}(x, y) = E[(x - \mu)(y - \mu)^T] = E[x(\mu + \Lambda x + w - \mu)^T] \\
  = E[XX'\Lambda + \Lambda w'] \\
  = \Lambda
  \]

- **Hence the joint distribution of x and y:**
  \[
  p(x, y) = \mathcal{N}(x; \mu, \Lambda) \cdot \mathcal{N}(y; \mu + \Lambda x, \Psi)
  \]

- 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**
  \[
  \Sigma_{11} = I \\
  \Sigma_{12} = \Sigma_{12}^T = \Lambda^T \\
  \Sigma_{22} = (\Lambda \Lambda^T + \Psi)
  \]
  we can now derive the posterior of the latent variable x given observation y, \( p(x|y) = \mathcal{N}(x; m_{12}, V_{12}) \), where

  \[
  m_{12} = \mu_1 + \Sigma_{12}^{-1} \Sigma_{22}^{-1} (y - \mu_2) \\
  V_{12} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21} \\
  = \Lambda^T (\Lambda \Lambda^T + \Psi)^{-1} (y - \mu) \\
  = I - \Lambda^T (\Lambda \Lambda^T + \Psi)^{-1} \Lambda
  \]

  Applying the matrix inversion lemma (\( E^{-1}F \)) = \( E^{-1} + E^{-1}F[H - GE^{-1}F]^TGE^{-1} \)

  \[ \Rightarrow \]

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

- 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:

\[ \mu = \sum_{n} y_n \]

\[ \Lambda, \Psi \] are coupled nonlinearly in log-likelihood

EM for Factor Analysis

- Incomplete data log likelihood function (marginal density of \( y \))
  \[ \ell(\theta, D) = -\frac{N}{2} \log|\Lambda \Lambda^T + \Psi| - \frac{1}{2} \sum_{n} (y_n - \mu)^T (\Lambda \Lambda^T + \Psi)^{-1} (y_n - \mu) \]
  \[ = -\frac{N}{2} \log|\Lambda \Lambda^T + \Psi| - \frac{1}{2} \text{tr}[(\Lambda \Lambda^T + \Psi)^{-1} S], \quad \text{where } S = \sum_{n} (y_n - \mu)(y_n - \mu)^T \]

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

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

- Compute \( \langle \zeta(\theta,D) \rangle_{\mu,y} \)

\[
\langle \zeta(\theta,D) \rangle = -\frac{N}{2} \log|\Psi| - \frac{1}{2} \sum_{n} \langle x_n^T x_n \rangle - \frac{N}{2} \text{tr}[S]\]

\[
\langle S \rangle = \frac{1}{N} \sum_{n} (y_n x_n^T - y_n \langle X_n \rangle \Lambda^T - \Lambda \langle X_n \rangle y_n^T + \Lambda \langle X_n \rangle^T \Lambda^T)
\]

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

\[
\langle X_n X_n^\top \rangle = \text{Var}[X_n | y_n] + E[X_n | y_n] E[X_n | y_n]^\top
\]

- Recall that we have derived:

\[
V_{12} = (I + \Lambda^T \Psi^{-1} \Lambda)^{-1} \quad \text{and} \quad m_{12} = V_{12} \Lambda^T \Psi^{-1} (y - \mu)
\]

\[
\Rightarrow \langle X_n \rangle = m_{12, y_n} = V_{12} \Lambda^T \Psi^{-1} (y_n - \mu) \quad \text{and} \quad \langle X_n X_n^\top \rangle = V_{12} + m_{12, y_n} m_{12, y_n}^\top
\]

M-step for Factor Analysis

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

- Using the trace and determinant derivative rules:

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

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

\[
= -\frac{N}{2} \Psi^{-1} \frac{\partial}{\partial \Lambda} \left( \sum_{n} (y_n x_n^T - y_n \langle X_n \rangle \Lambda^T - \Lambda \langle X_n \rangle y_n^T + \Lambda \langle X_n \rangle^T \Lambda^T) \right)
\]

\[
= \Psi^{-1} \sum_{n} y_n \langle X_n \rangle^T - \Psi^{-1} \Lambda \sum_{n} \langle X_n \rangle \Lambda^T \Rightarrow \Lambda^{-1} = \left( \sum_{n} y_n \langle X_n \rangle \right) \left( \sum_{n} \langle X_n X_n^T \rangle \right)^{-1}
\]
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: $(\Lambda Q)(\Lambda Q)^T = \Lambda(Q Q^T)\Lambda^T = \Lambda \Lambda^T$.
- 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
- Switching SSM
- Continuous model (e.g., mixture of Gaussians)
- HMM (for continuous sequential data, e.g., speech signal)
- Factor analysis
- State space model

State space models (SSM)

- A sequential FA or a continuous state HMM
  
  \[
  x_t = Ax_{t-1} + Gw_t, \quad y_t = Cx_{t-1} + v_t, \quad w_t \sim \mathcal{N}(0; Q), \quad v_t \sim \mathcal{N}(0; R), \quad x_0 \sim \mathcal{N}(0; \Sigma_0),
  \]

  This is a linear dynamic system.

- In general,
  
  \[
  x_t = f(x_{t-1}) + Gw_t, \quad y_t = f(x_{t-1}) + v_t,
  \]

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

- 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
LDS for 2D tracking

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

\[
\begin{pmatrix}
    x_t^1 \\
    x_t^2 \\
    x_t^1 \\
    x_t^2
\end{pmatrix} = \Delta 
\begin{pmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 1 \\
    0 & 0 & 0 & 1 \\
    0 & 0 & 0 & 1
\end{pmatrix} 
\begin{pmatrix}
    x_{t-1}^1 \\
    x_{t-1}^2 \\
    x_{t-1}^1 \\
    x_{t-1}^2
\end{pmatrix} + \text{noise}
\]

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

\[
\begin{pmatrix}
    y_t^1 \\
    y_t^2
\end{pmatrix} = \begin{pmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0
\end{pmatrix} 
\begin{pmatrix}
    x_t^1 \\
    x_t^2 \\
    x_t^1 \\
    x_t^2
\end{pmatrix} + \text{noise}
\]

2D tracking

2D filtering

2D smoothing
Kalman filtering in the brain?

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 $\bar{x}_t = E(X_t | y_1, \ldots, y_t)$ and covariance $P_{tt} = E(X_t X_t^T | y_1, \ldots, y_t)$.
- It is common to work with the inverse covariance (precision) matrix $P_{tt}^{-1}$; 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 $p(y_{t+1}|X_t)$ --- measurement update
Predict step

- **Dynamical Model:** \( x_{t+1} = Ax_t + Gw_t, \quad w_t \sim \mathcal{N}(0, Q) \)
  - One step ahead prediction of state:
    \[
    \hat{x}_{t+1} = E(X_{t+1} | y_1, \ldots, y_t) = A\hat{x}_t
    \]
    \[
    P_{t+1} = E((X_{t+1} - \hat{x}_{t+1})(X_{t+1} - \hat{x}_{t+1})^T | y_1, \ldots, y_t) = AP_tA + GQG^T
    \]
- **Observation model:** \( y_t = Cx_{t+1} + \nu_t, \quad \nu_t \sim \mathcal{N}(0, R) \)
  - One step ahead prediction of observation:
    \[
    E(Y_{t+1} | y_1, \ldots, y_t) = E(CX_{t+1} + \nu_{t+1} | y_1, \ldots, y_t) = C\hat{x}_{t+1}
    \]
    \[
    E(Y_t - \hat{y}_{t+1})(Y_t - \hat{y}_{t+1})^T | y_1, \ldots, y_t) = CP_{t+1}C^T + R
    \]
    \[
    E(Y_t - \hat{y}_{t+1})(X_t - \hat{x}_{t+1})^T | y_1, \ldots, y_t) = C\dot{P}_{t+1}
    \]

Update step

- Summarizing results from previous slide, we have \( p(X_{t+1}, Y_{t+1} | y_{1:t}) \sim \mathcal{N}(m_{t+1}, V_{t+1}) \), where
  \[
  m_{t+1} = \left[ \begin{array}{c} \hat{x}_{t+1} \\ \hat{x}_{t+1} \\ C\hat{x}_{t+1} \end{array} \right], \quad V_{t+1} = \left[ \begin{array}{ccc} P_{t+1} & P_{t+1}C^T \\ CP_{t+1} & CP_{t+1}C^T + R \end{array} \right]
  \]
- Remember the formulas for conditional Gaussian distributions:
  \[
  p(x_1 | x_2) = \mathcal{N}(x_1 | \mu_1, \Sigma_{12}) \quad p(x_2 | x_1) = \mathcal{N}(x_2 | \mu_2, \Sigma_{21})
  \]
  \[
  m_{21} = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (x_2 - \mu_2)
  \]
  \[
  V_{21} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}
  \]
  \[
  m_{12} = \mu_1 + \Sigma_{12}^{-1} \Sigma_{22} (x_2 - \mu_2)
  \]
  \[
  V_{12} = \Sigma_{11} - \Sigma_{12}^{-1} \Sigma_{22} \Sigma_{21}
  \]
  \[
  m_2 = \mu_2
  \]
  \[
  V_2 = \Sigma_{22}
  \]
  \[
  m_{12} = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (x_2 - \mu_2)
  \]
Kalman Filter

- Measurement updates:
  \[ \hat{x}_{t|t-1} = \hat{x}_{t|t} + K_{t|t} (y_{t|t} - C \hat{x}_{t|t}) \]
  \[ P_{t|t-1} = P_{t|t} - KCP_{t|t} \]
  where \( K_{t|t} \) is the Kalman gain matrix
  \[ K_{t|t} = P_{t|t} C^T (C P_{t|t} C^T + R)^{-1} \]

- Time updates:
  \[ \hat{x}_{t|t} = A \hat{x}_{t|t} \]
  \[ P_{t|t} = AP_{t|t} A + GQG^T \]
  \( K_t \) 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) \]
  \( z_t = x_t + v, \quad v \sim \mathcal{N}(0, \sigma_v) \)

- KF equations:
  \[ P_{t|t} = AP_{t|t-1} A + GQG^T = \sigma_x^2 + \sigma_z^2 \]
  \( \hat{x}_{t|t-1} = A \hat{x}_{t|t-1} \)
  \[ \hat{x}_{t+1|t} = \hat{x}_{t+1|t-1} + K_{t+1|t} (z_{t+1} - C \hat{x}_{t+1|t}) \]
  \[ K_{t+1|t} = P_{t+1|t} C^T (C P_{t+1|t} C^T + R)^{-1} \]
  \( \sigma_x^2 + \sigma_v^2 + \sigma_z^2 \)
  \[ P_{t+1|t} = P_{t+1|t} - KCP_{t+1|t} = \frac{(\sigma_z^2 + \sigma_v^2) \sigma_x^2}{\sigma_x^2 + \sigma_v^2 + \sigma_z^2} \]
KF intuition

- The KF update of the mean is
  \[ \hat{x}_{t+1|t} = \hat{x}_{t|t-1} + K_{t|t-1}(z_{t+1} - C\hat{x}_{t|t-1}) = \frac{(\sigma_x + \sigma_z)z_t + \sigma_z\hat{x}_{t|t}}{\sigma_x + \sigma_z + \sigma_z} \]
  - the term \((z_{t+1} - C\hat{x}_{t|t-1})\) is called the **innovation**
- New belief is convex combination of updates from prior and observation, weighted by Kalman Gain matrix:
  \[ K_{t|t-1} = P_{t|t-1}C^T(CP_{t|t-1}C^T + R)^{-1} \]
  - 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
  \[ \hat{x}_{t+1|t} = A\hat{x}_{t|t} + K_{t+1|t}(y_{t+1} - C\hat{x}_{t+1|t-1}) \]
- 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, \(y_t = x_t^T\theta + v_t\), is a linear regression
- We can estimate recursively using the Kalman filter:
  \[ \hat{\theta}_{t+1} = \hat{\theta}_t + P_{t+1}R^{-1}(y_{t+1} - x_{t+1}\hat{\theta}_t)x_{t+1} \]
  This is called the recursive least squares (RLS) algorithm.
- We can approximate \(P_{t+1}R^{-1} \approx \eta_{t+1}\) by a scalar constant. This is called the least mean squares (LMS) algorithm.
- We can adapt \(\eta\) 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 \( P_{t+1|t} = AP_{t|t}A^T + GG^T \) takes \( O(N_{x}^2) \) time, assuming dense \( P \) and dense \( A \).

- Computing \( K_{t+1} = P_{t+1|t}C^T(CP_{t+1|t}C^T + R)^{-1} \) takes \( O(N_{y}^3) \) time.

- So overall time is, in general, \( \max \{ N_{x}^2, N_{y}^3 \} \).

Rauch-Tung-Strievel smoother

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

- General structure: KF results + the difference of the "smooth" 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: \( X_{t} \mid Y_{1}, \ldots, Y_{T} \)
- The trick:

\[
\begin{align*}
\hat{x}_{t|T} &\overset{\text{def}}{=} E[X_{t} \mid Y_{1}, \ldots, Y_{T}] \\
\Var[X_{t} \mid Z] &\overset{\text{def}}{=} E[\Var[X_{t} \mid Y, Z] \mid Z] + \Var[E[X_{t} \mid Y, Z] \mid Z] \\
\hat{x}_{t|T} &\overset{\text{def}}{=} E[X_{t} \mid Y_{1}, \ldots, Y_{T}] = E[E[X_{t} \mid X_{t+1}, Y_{1}, \ldots, Y_{T}] \mid Y_{1}, \ldots, Y_{T}] \\
&= E[E[X_{t} \mid X_{t+1}, Y_{1}, \ldots, Y_{T}] \mid Y_{1}, \ldots, Y_{T}] \\
&= E[X_{t} \mid X_{t+1}, Y_{1}, \ldots, Y_{T}]
\end{align*}
\]

Same for \( P_{t|T} \).
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 all the quantities here are available after a forward KF pass.

Remember the formulas for conditional Gaussian distributions:
\[
\begin{align*}
\rho(x, y) &= \mathcal{N}(m, V) \\
m &= \mu \\
V &= \Sigma \\
\rho(x, y) &= \mathcal{N}(m + \Sigma x, V) \\
m &= \mu + \Sigma x \\
V &= \Sigma + \Sigma V \Sigma \\
\rho(x, y) &= \mathcal{N}(m, V) \\
m &= \mu \\
V &= \Sigma \\
\rho(x, y) &= \mathcal{N}(m + \Sigma x, V) \\
m &= \mu + \Sigma x \\
V &= \Sigma + \Sigma V \Sigma \\
\rho(x, y) &= \mathcal{N}(m + \Sigma x, V) \\
m &= \mu + \Sigma x \\
V &= \Sigma + \Sigma V \Sigma \\
\rho(x, y) &= \mathcal{N}(m + \Sigma x, V) \\
m &= \mu + \Sigma x \\
V &= \Sigma + \Sigma V \Sigma \\
\end{align*}
\]

The RTS smoother
\[
\tilde{x}_{t|T} = E[X_{t|T} | X_{t+1}, Y_1 \ldots Y_T] = \hat{s}_{t|T} + L_t (\hat{s}_{t+1|T} - \hat{s}_{t|T})
\]
\[
P_{t|T} = Var[\tilde{x}_{t|T} | Y_{1:T}] + E[Var[X_t | X_{t+1}, Y_{1:T}] | Y_{1:T}] = P_{t|T} + L_t (P_{t+1|T} - P_{t|T}) \tilde{P}_{t|T}
\]

Learning SSMs

Complete log likelihood
\[
\zeta(\theta, D) = \sum_x \log p(x_t, y_t) = \sum_x \log p(x_t) + \sum_x \sum_{x_{t-1}} \log p(x_t | x_{t-1}) + \sum_{y_t} \sum_x \log p(y_t | x_t)
\]
\[
= f_\theta(X_t; \Sigma_0) + f_\theta(\langle x_t, x^{*}_{t+1}, \ldots, x^{*}_T, x_t : \forall t \rangle; A, Q, G) + f_\theta(\langle x_t, x^{*}_t, \ldots, x^{*}_T, x_t : \forall t \rangle; C, R)
\]

EM

E-step: compute \( \langle X_t, X^{*}_{t+1}, \ldots, X^{*}_T, x_t : \forall t \rangle | y_T \)

these quantities can be inferred via KF and RTS filters, etc., e.g., \( \langle x_t, x^{*}_t \rangle = \text{var}(x_t, x^{*}_t) + E(x_t) = \rho_{ft} + x^{*}_{ft} \)

M-step: MLE using
\[
\zeta(\theta, D) = f_\theta(\langle x_t \rangle; \Sigma_0) + f_\theta(\langle x_t, x^{*}_{t+1}, \ldots, x^{*}_T, x_t : \forall t \rangle; A, Q, G) + f_\theta(\langle x_t, x^{*}_t, \ldots, x^{*}_T, x_t : \forall t \rangle; C, R)
\]
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:
    \[ x_t = f(\hat{x}_{t-1}) + A_{\theta}(x_{t-1} - \hat{x}_{t-1}) + w_t \]
    \[ y_t = g(\hat{x}_{t-1}) + C_{\theta}(x_t - \hat{x}_{t-1}) + v_t \]
    where \( \hat{x}_{t-1} = f(\hat{x}_{t-1}) \) and \( A_{\theta} = \frac{\partial f}{\partial x}\bigg|_{\hat{x}} \) and \( C_{\theta} = \frac{\partial g}{\partial x}\bigg|_{\hat{x}} \)
- The noise covariance (\( Q \) and \( R \)) is not changed, i.e., the additional error due to linearization is not modeled.