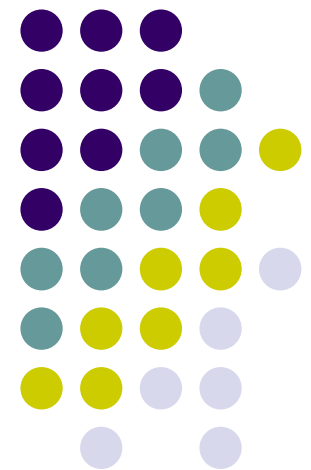




Probabilistic Graphical Models

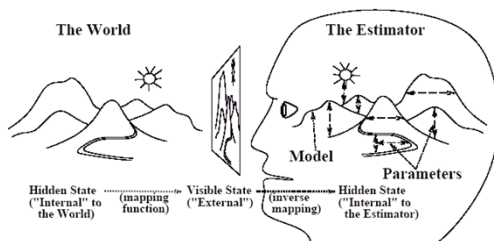
Factor Analysis and State Space Models



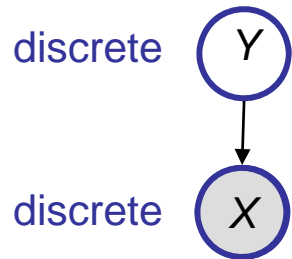
Eric Xing

Lecture 11, February 18, 2015

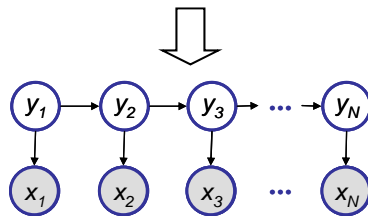
Reading: See class website



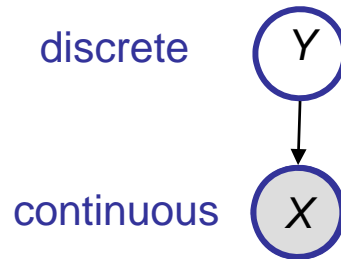
A road map to more complex dynamic models



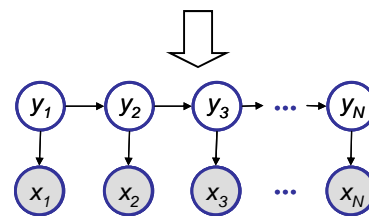
Mixture model
e.g., mixture of multinomials



HMM
(for discrete sequential data, e.g., text)



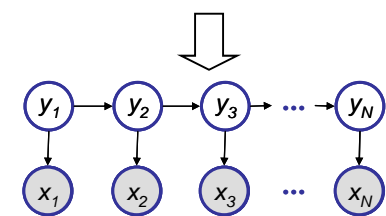
Mixture model
e.g., mixture of Gaussians



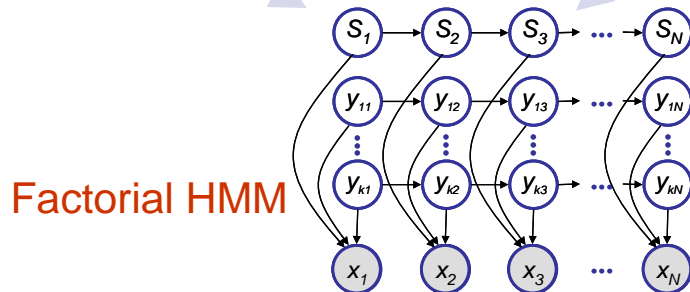
HMM
(for continuous sequential data, e.g., speech signal)



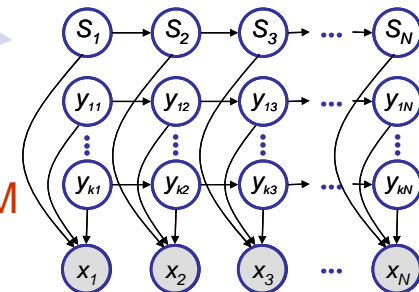
Factor analysis



State space model



Switching SSM





Recall multivariate Gaussian

- Multivariate Gaussian density:

$$p(\mathbf{x} | \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right\}$$

- A joint Gaussian:

$$p\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} | \mu, \Sigma\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \middle| \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

- How to write down $p(\mathbf{x}_1)$, $p(\mathbf{x}_1|\mathbf{x}_2)$ or $p(\mathbf{x}_2|\mathbf{x}_1)$ using the block elements in μ and Σ ?

- Formulas to remember:

$$p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 | \mathbf{m}_2^m, \mathbf{V}_2^m)$$

$$\mathbf{m}_2^m = \mu_2$$

$$\mathbf{V}_2^m = \Sigma_{22}$$

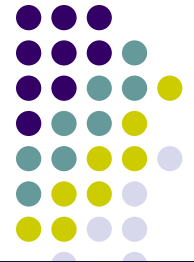
$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1 | \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{m}_{1|2} = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (\mathbf{x}_2 - \mu_2)$$

$$\mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}$$

Review:

The matrix inverse lemma



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

- First we diagonalize M

$$\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} \begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix} = \begin{bmatrix} E-FH^{-1}G & 0 \\ 0 & H \end{bmatrix}$$

- Schur complement: $M/H = E-FH^{-1}G$

- Then we inverse, using this formula: $XYZ = W \Rightarrow Y^{-1} = ZW^{-1}X$

$$\begin{aligned} M^{-1} &= \begin{bmatrix} E & F \\ G & H \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix} \begin{bmatrix} (M/H)^{-1} & 0 \\ 0 & H^{-1} \end{bmatrix} \begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} (M/H)^{-1} & -(M/H)^{-1}FH^{-1} \\ -H^{-1}G(M/H)^{-1} & H^{-1} + H^{-1}G(M/H)^{-1}FH^{-1} \end{bmatrix} = \begin{bmatrix} E^{-1} + E^{-1}F(M/E)^{-1}GE^{-1} & -E^{-1}F(M/E)^{-1} \\ -(M/E)^{-1}GE^{-1} & (M/E)^{-1} \end{bmatrix} \end{aligned}$$

- Matrix inverse lemma

$$(E-FH^{-1}G)^{-1} = E^{-1} + E^{-1}F(H-GE^{-1}F)^{-1}GE^{-1}$$

Review:

Some matrix algebra



- Trace and derivatives
 - Cyclical permutations

$$\text{tr}[A] \stackrel{\text{def}}{=} \sum_i a_{ii}$$

- Derivatives

$$\text{tr}[ABC] = \text{tr}[CAB] = \text{tr}[BCA]$$

$$\frac{\partial}{\partial A} \text{tr}[BA] = B^T$$

$$\frac{\partial}{\partial A} \text{tr}[x^T Ax] = \frac{\partial}{\partial A} \text{tr}[xx^T A] = xx^T$$

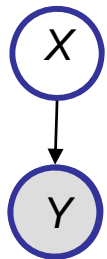
- Determinants and derivatives

$$\frac{\partial}{\partial A} \log|A| = A^{-1}$$



Factor analysis

- An unsupervised linear regression model

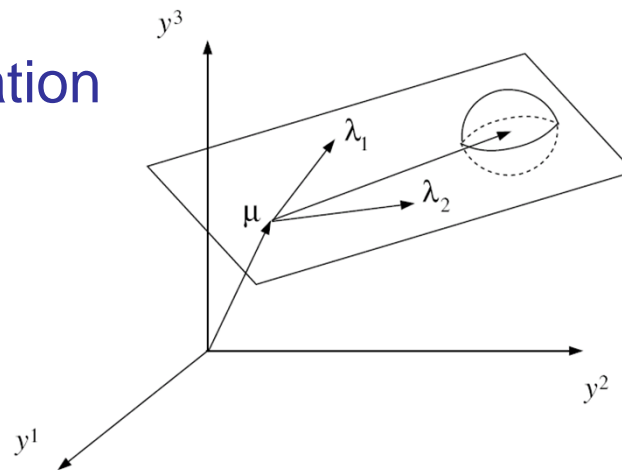


$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, I)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu} + \Lambda\mathbf{x}, \Psi)$$

where Λ is called a factor loading matrix, and Ψ is diagonal.

- Geometric interpretation

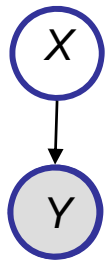


- To generate data, first generate a point within the manifold then add noise. Coordinates of point are components of latent variable.



Marginal data distribution

- A marginal Gaussian (e.g., $p(\mathbf{x})$) times a conditional Gaussian (e.g., $p(\mathbf{y}|\mathbf{x})$) is a joint Gaussian
- Any marginal (e.g., $p(\mathbf{y})$) of a joint Gaussian (e.g., $p(\mathbf{x},\mathbf{y})$) 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.)



$$\begin{aligned} E[\mathbf{Y}] &= E[\boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{X} + \mathbf{W}] && \text{where } \mathbf{W} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}) \\ &= \boldsymbol{\mu} + \boldsymbol{\Lambda}E[\mathbf{X}] + E[\mathbf{W}] \\ &= \boldsymbol{\mu} + \mathbf{0} + \mathbf{0} = \boldsymbol{\mu} \\ \text{Var}[\mathbf{Y}] &= E[(\mathbf{Y} - \boldsymbol{\mu})(\mathbf{Y} - \boldsymbol{\mu})^T] \\ &= E[(\boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{X} + \mathbf{W} - \boldsymbol{\mu})(\boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{X} + \mathbf{W} - \boldsymbol{\mu})^T] \\ &= E[(\boldsymbol{\Lambda}\mathbf{X} + \mathbf{W})(\boldsymbol{\Lambda}\mathbf{X} + \mathbf{W})^T] \\ &= \boldsymbol{\Lambda}E[\mathbf{X}\mathbf{X}^T]\boldsymbol{\Lambda}^T + E[\mathbf{W}\mathbf{W}^T] \\ &= \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi} \end{aligned}$$

FA = Constrained-Covariance Gaussian

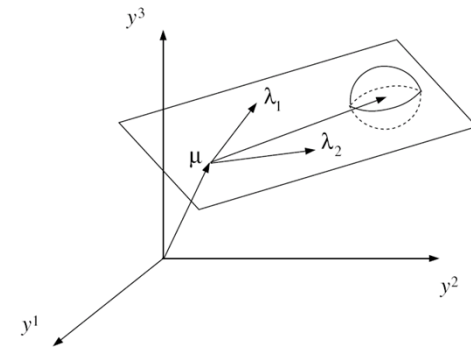


- Marginal density for factor analysis (\mathbf{y} is p -dim, \mathbf{x} is k -dim):

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

- So the effective covariance is the low-rank outer product of two long skinny matrices plus a diagonal matrix:

$$\text{Cov}[\mathbf{y}] = \Lambda \Lambda^T + \Psi$$



- In other words, factor analysis is just a constrained Gaussian model. (If Ψ were not diagonal then we could model any Gaussian and it would be pointless.)



FA joint distribution

- Model

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, I)$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mu + \Lambda\mathbf{x}, \Psi)$$

- Covariance between \mathbf{x} and \mathbf{y}

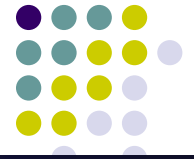
$$\begin{aligned} \text{Cov}[\mathbf{X}, \mathbf{Y}] &= E[(\mathbf{X} - \mathbf{0})(\mathbf{Y} - \mu)^T] = E[\mathbf{X}(\mu + \Lambda\mathbf{X} + \mathbf{W} - \mu)^T] \\ &= E[\mathbf{X}\mathbf{X}^T \Lambda^T + \mathbf{X}\mathbf{W}^T] \\ &= \Lambda^T \end{aligned}$$

- Hence the joint distribution of \mathbf{x} and \mathbf{y} :

$$p\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ \mu \end{bmatrix}, \begin{bmatrix} I & \Lambda^T \\ \Lambda & \Lambda\Lambda^T + \Psi \end{bmatrix}\right)$$

- Assume noise is uncorrelated with data or latent variables.

$$\begin{bmatrix} (M/H)^{-1} & -(M/H)^{-1}FH^{-1} \\ -H^{-1}G(M/H)^{-1} & H^{-1}+H^{-1}G(M/H)^{-1}FH^{-1} \end{bmatrix} = \begin{bmatrix} E^{-1}+E^{-1}F(M/E)^{-1}GE^{-1} & -E^{-1}F(M/E)^{-1} \\ -(M/E)^{-1}GE^{-1} & (M/E)^{-1} \end{bmatrix}$$



Inference in Factor Analysis

- Apply the Gaussian conditioning formulas to the joint distribution we derived above, where

$$\begin{aligned} \Sigma_{11} &= I \\ \Sigma_{12} &= \Sigma_{12}^T = \Lambda^T \\ \Sigma_{22} &= (\Lambda\Lambda^T + \Psi) \end{aligned}$$

we can now derive the posterior of the latent variable \mathbf{x} given observation \mathbf{y} , $p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x} | \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$, where

$$\begin{aligned} \mathbf{m}_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y} - \mu_2) & \mathbf{V}_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\ &= \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}(\mathbf{y} - \mu) & &= I - \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\Lambda \end{aligned}$$

Applying the matrix inversion lemma

$$(E - FH^{-1}G)^{-1} = E^{-1} + E^{-1}F(H - GE^{-1}F)^{-1}GE^{-1}$$

$$\Rightarrow \mathbf{V}_{1|2} = (I + \Lambda^T\Psi^{-1}\Lambda)^{-1} \quad \mathbf{m}_{1|2} = \mathbf{V}_{1|2}\Lambda^T\Psi^{-1}(\mathbf{y} - \mu)$$

- Here we only need to invert a matrix of size $|\mathbf{x}| \times |\mathbf{x}|$, instead of $|\mathbf{y}| \times |\mathbf{y}|$.

Geometric interpretation: inference is linear projection



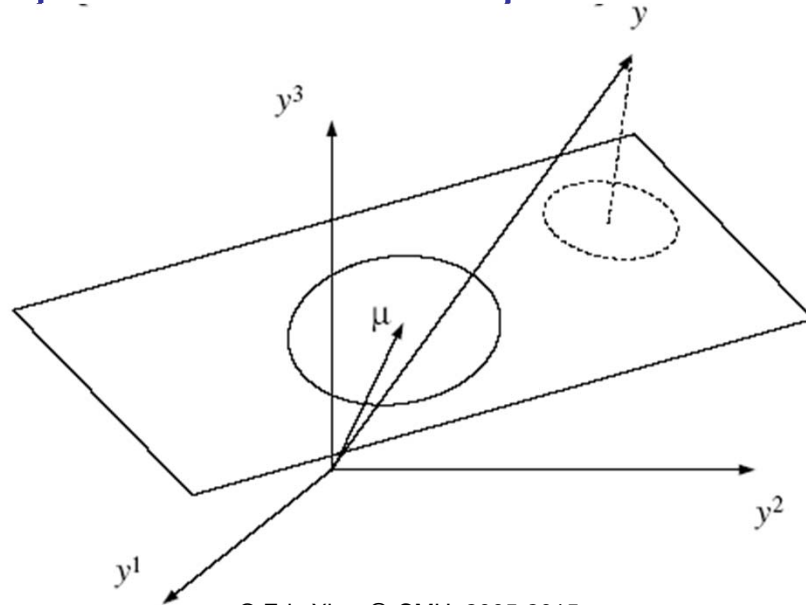
- The posterior is:

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{V}_{1|2} = (\mathbf{I} + \Lambda^T \Psi^{-1} \Lambda)^{-1}$$

$$\mathbf{m}_{1|2} = \mathbf{V}_{1|2} \Lambda^T \Psi^{-1} (\mathbf{y} - \boldsymbol{\mu})$$

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



Learning FA



- Now, assume that we are given $\{y_n\}$ (the observation on high-dimensional data) only
- We have derived how to estimate x_n from $P(X|Y)$
- How can we learning the model?
 - Loading matrix Λ
 - Manifold center μ
 - Variance Ψ



EM for Factor Analysis

- Incomplete data log likelihood function (marginal density of y)

$$\begin{aligned}\ell(\theta, \mathcal{D}) &= -\frac{N}{2} \log |\Lambda \Lambda^T + \Psi| - \frac{1}{2} \sum_n (\mathbf{y}_n - \boldsymbol{\mu})^T (\Lambda \Lambda^T + \Psi)^{-1} (\mathbf{y}_n - \boldsymbol{\mu}) \\ &= -\frac{N}{2} \log |\Lambda \Lambda^T + \Psi| - \frac{1}{2} \text{tr} [(\Lambda \Lambda^T + \Psi)^{-1} \mathbf{S}], \quad \text{where } \mathbf{S} = \sum_n (\mathbf{y}_n - \boldsymbol{\mu})(\mathbf{y}_n - \boldsymbol{\mu})^T\end{aligned}$$

- Estimating $\boldsymbol{\mu}$ is trivial: $\hat{\boldsymbol{\mu}}^{ML} = \frac{1}{N} \sum_n \mathbf{y}_n$
- Parameters Λ and Ψ are coupled nonlinearly in log-likelihood

- Complete log likelihood

$$\begin{aligned}\ell_c(\theta, \mathcal{D}) &= \sum_n \log p(\mathbf{x}_n, \mathbf{y}_n) = \sum_n \log p(\mathbf{x}_n) + \log p(\mathbf{y}_n | \mathbf{x}_n) \\ &= -\frac{N}{2} \log |I| - \frac{1}{2} \sum_n \mathbf{x}_n^T \mathbf{x}_n - \frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n (\mathbf{y}_n - \Lambda \mathbf{x}_n)^T \Psi^{-1} (\mathbf{y}_n - \Lambda \mathbf{x}_n) \\ &= -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n \text{tr} [\mathbf{x}_n \mathbf{x}_n^T] - \frac{N}{2} \text{tr} [\mathbf{S} \Psi^{-1}], \quad \text{where } \mathbf{S} = \frac{1}{N} \sum_n (\mathbf{y}_n - \Lambda \mathbf{x}_n)(\mathbf{y}_n - \Lambda \mathbf{x}_n)^T\end{aligned}$$



E-step for Factor Analysis

- Compute $\langle \ell_c(\theta, \mathcal{D}) \rangle_{p(\mathbf{x}|\mathbf{y})}$

$$\langle \ell_c(\theta, \mathcal{D}) \rangle = -\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n \text{tr}[\langle \mathbf{X}_n \mathbf{X}_n^T \rangle] - \frac{N}{2} \text{tr}[\langle \mathbf{S} \rangle \Psi^{-1}]$$

$$\langle \mathbf{S} \rangle = \frac{1}{N} \sum_n (\mathbf{y}_n \mathbf{y}_n^T - \mathbf{y}_n \langle \mathbf{X}_n^T \rangle \Lambda^T - \Lambda \langle \mathbf{X}_n^T \rangle \mathbf{y}_n^T + \Lambda \langle \mathbf{X}_n \mathbf{X}_n^T \rangle \Lambda^T)$$

$$\langle \mathbf{X}_n \rangle = E[\mathbf{X}_n | \mathbf{y}_n]$$

$$\langle \mathbf{X}_n \mathbf{X}_n^T \rangle = \text{Var}[\mathbf{X}_n | \mathbf{y}_n] + E[\mathbf{X}_n | \mathbf{y}_n] E[\mathbf{X}_n | \mathbf{y}_n]^T$$

- Recall that we have derived:

$$\mathbf{V}_{1|2} = (\mathbf{I} + \Lambda^T \Psi^{-1} \Lambda)^{-1} \quad \mathbf{m}_{1|2} = \mathbf{V}_{1|2} \Lambda^T \Psi^{-1} (\mathbf{y} - \boldsymbol{\mu})$$

$$\Rightarrow \langle \mathbf{X}_n \rangle = \mathbf{m}_{\mathbf{x}_n | \mathbf{y}_n} = \mathbf{V}_{1|2} \Lambda^T \Psi^{-1} (\mathbf{y}_n - \boldsymbol{\mu}) \quad \text{and} \quad \langle \mathbf{X}_n \mathbf{X}_n^T \rangle = \mathbf{V}_{1|2} + \mathbf{m}_{\mathbf{x}_n | \mathbf{y}_n} \mathbf{m}_{\mathbf{x}_n | \mathbf{y}_n}^T$$



M-step for Factor Analysis

- Take the derivatives of the expected complete log likelihood wrt. parameters.
 - Using the trace and determinant derivative rules:

$$\begin{aligned}\frac{\partial}{\partial \Psi^{-1}} \langle \ell_c \rangle &= \frac{\partial}{\partial \Psi^{-1}} \left(-\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n \text{tr}[\langle \mathbf{X}_n \mathbf{X}_n^T \rangle] - \frac{N}{2} \text{tr}[\langle \mathbf{S} \rangle \Psi^{-1}] \right) \\ &= \frac{N}{2} \Psi - \frac{N}{2} \langle \mathbf{S} \rangle \quad \Rightarrow \quad \Psi^{t+1} = \langle \mathbf{S} \rangle\end{aligned}$$

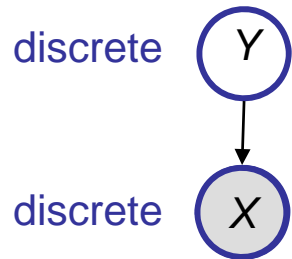
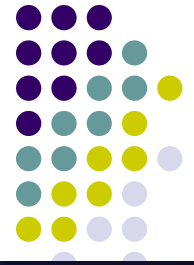
$$\begin{aligned}\frac{\partial}{\partial \Lambda} \langle \ell_c \rangle &= \frac{\partial}{\partial \Lambda} \left(-\frac{N}{2} \log |\Psi| - \frac{1}{2} \sum_n \text{tr}[\langle \mathbf{X}_n \mathbf{X}_n^T \rangle] - \frac{N}{2} \text{tr}[\langle \mathbf{S} \rangle \Psi^{-1}] \right) = -\frac{N}{2} \Psi^{-1} \frac{\partial}{\partial \Lambda} \langle \mathbf{S} \rangle \\ &= -\frac{N}{2} \Psi^{-1} \frac{\partial}{\partial \Lambda} \left(\frac{1}{N} \sum_n (\mathbf{y}_n \mathbf{y}_n^T - \mathbf{y}_n \langle \mathbf{X}_n^T \rangle \Lambda^T - \Lambda \langle \mathbf{X}_n^T \rangle \mathbf{y}_n^T + \Lambda \langle \mathbf{X}_n \mathbf{X}_n^T \rangle \Lambda^T) \right) \\ &= \Psi^{-1} \sum_n \mathbf{y}_n \langle \mathbf{X}_n^T \rangle - \Psi^{-1} \Lambda \sum_n \langle \mathbf{X}_n \mathbf{X}_n^T \rangle \quad \Rightarrow \quad \Lambda^{t+1} = \left(\sum_n \mathbf{y}_n \langle \mathbf{X}_n^T \rangle \right) \left(\sum_n \langle \mathbf{X}_n \mathbf{X}_n^T \rangle \right)^{-1}\end{aligned}$$

Model Invariance and Identifiability

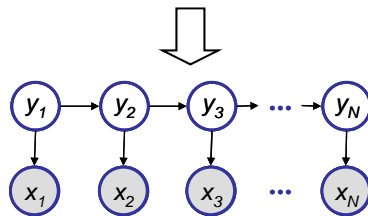


- There is *degeneracy* in the FA model.
- Since Λ only appears as outer product $\Lambda\Lambda^T$, the model is invariant to rotation and axis flips of the latent space.
- We can replace Λ with ΛQ for any orthonormal matrix Q and the model remains the same: $(\Lambda Q)(\Lambda Q)^T = \Lambda(QQ^T)\Lambda^T = \Lambda\Lambda^T$.
- 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.

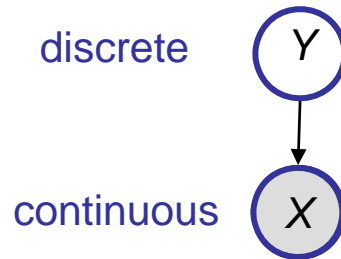
A road map to more complex dynamic models



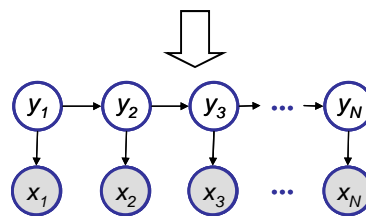
Mixture model
e.g., mixture of multinomials



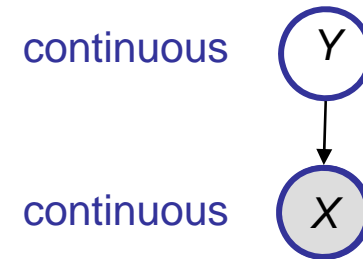
HMM
(for discrete sequential data, e.g., text)



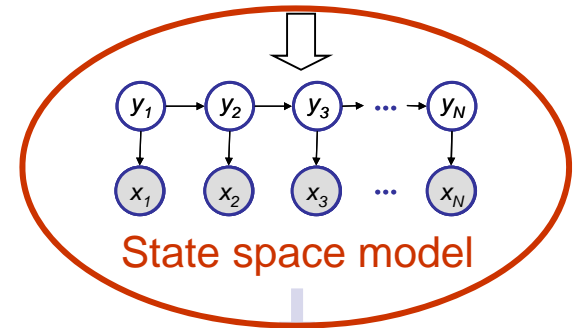
Mixture model
e.g., mixture of Gaussians



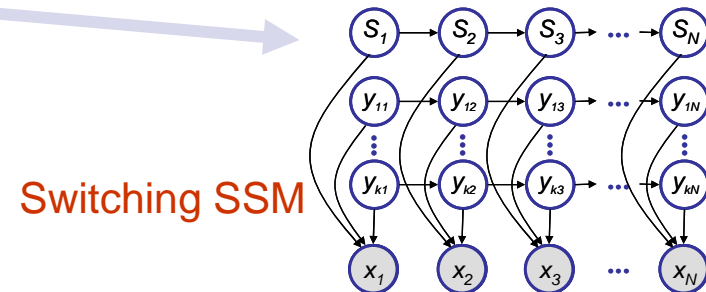
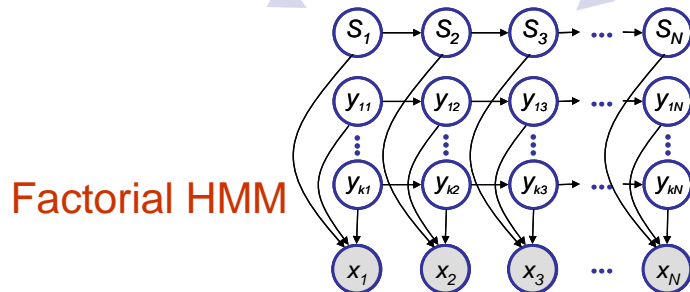
HMM
(for continuous sequential data, e.g., speech signal)



Factor analysis



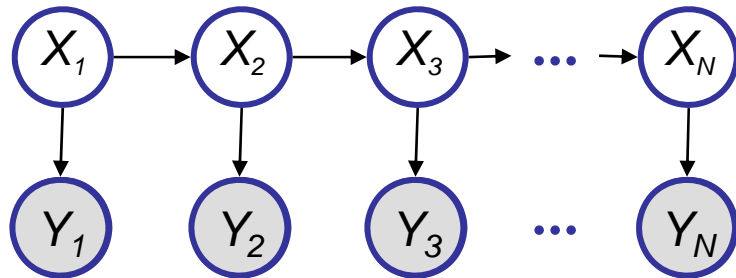
State space model





State space models (SSM)

- A sequential FA or a continuous state HMM



$$\mathbf{x}_t = A\mathbf{x}_{t-1} + G\mathbf{w}_t$$

$$\mathbf{y}_t = C\mathbf{x}_{t-1} + \mathbf{v}_t$$

$$\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}; Q), \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}; R)$$

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}; \Sigma_0),$$

This is a linear dynamic system.

- In general,

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + G\mathbf{w}_t$$

$$\mathbf{y}_t = g(\mathbf{x}_{t-1}) + \mathbf{v}_t$$

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



LDS for 2D tracking

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

$$\begin{pmatrix} x_t^1 \\ x_t^2 \\ \dot{x}_t^1 \\ \dot{x}_t^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1}^1 \\ x_{t-1}^2 \\ \dot{x}_{t-1}^1 \\ \dot{x}_{t-1}^2 \end{pmatrix} + \text{noise}$$

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

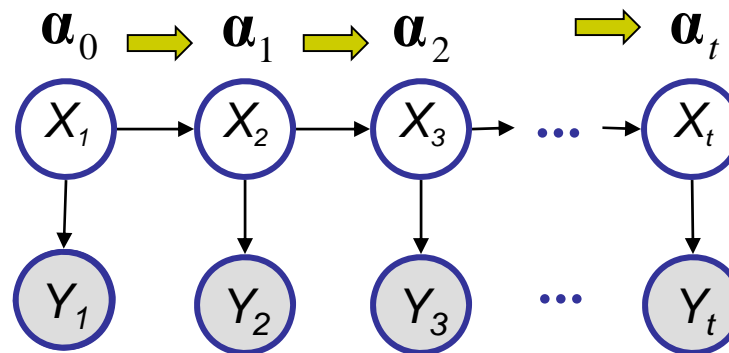
$$\begin{pmatrix} y_t^1 \\ y_t^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_t^1 \\ x_t^2 \\ \dot{x}_t^1 \\ \dot{x}_t^2 \end{pmatrix} + \text{noise}$$



The inference problem 1

- Filtering \rightarrow given y_1, \dots, y_t , estimate x_t : $P(x_t | \mathbf{y}_{1:t})$
 - The **Kalman filter** is a way to perform exact **online inference** (sequential Bayesian updating) in an LDS.
 - It is the Gaussian analog of the **forward algorithm** for HMMs:

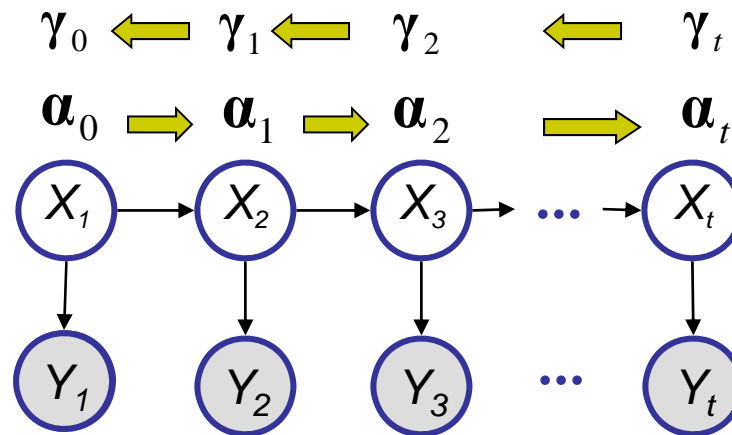
$$p(X_t = i | \mathbf{y}_{1:t}) = \alpha_t^i \propto p(y_t | X_t = i) \sum_j p(X_t = i | X_{t-1} = j) \alpha_{t-1}^j$$





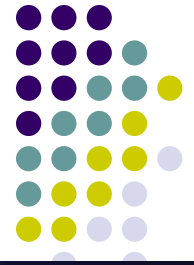
The inference problem 2

- Smoothing \rightarrow given y_1, \dots, y_T , estimate x_t ($t < T$)
 - The Rauch-Tung-Striavel smoother is a way to perform exact off-line inference in an LDS. It is the Gaussian analog of the forwards-backwards (alpha-gamma) algorithm:

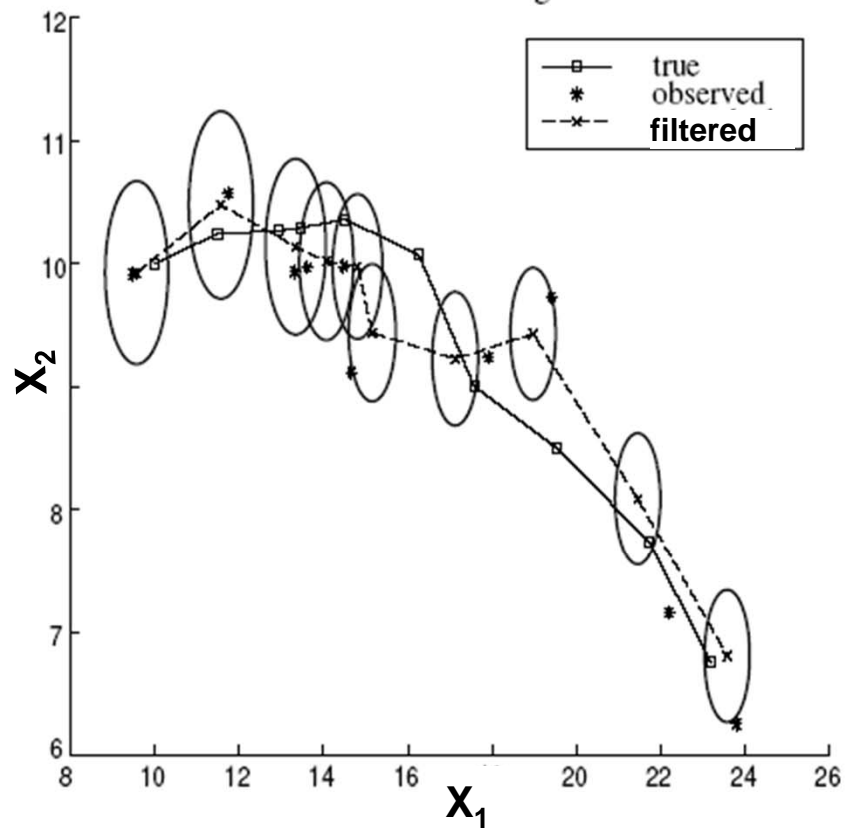


$$p(X_t = i | y_{1:T}) = \gamma_t^i \propto \sum_j \alpha_t^j P(X_{t+1}^j | X_t^i) \gamma_{t+1}^j$$

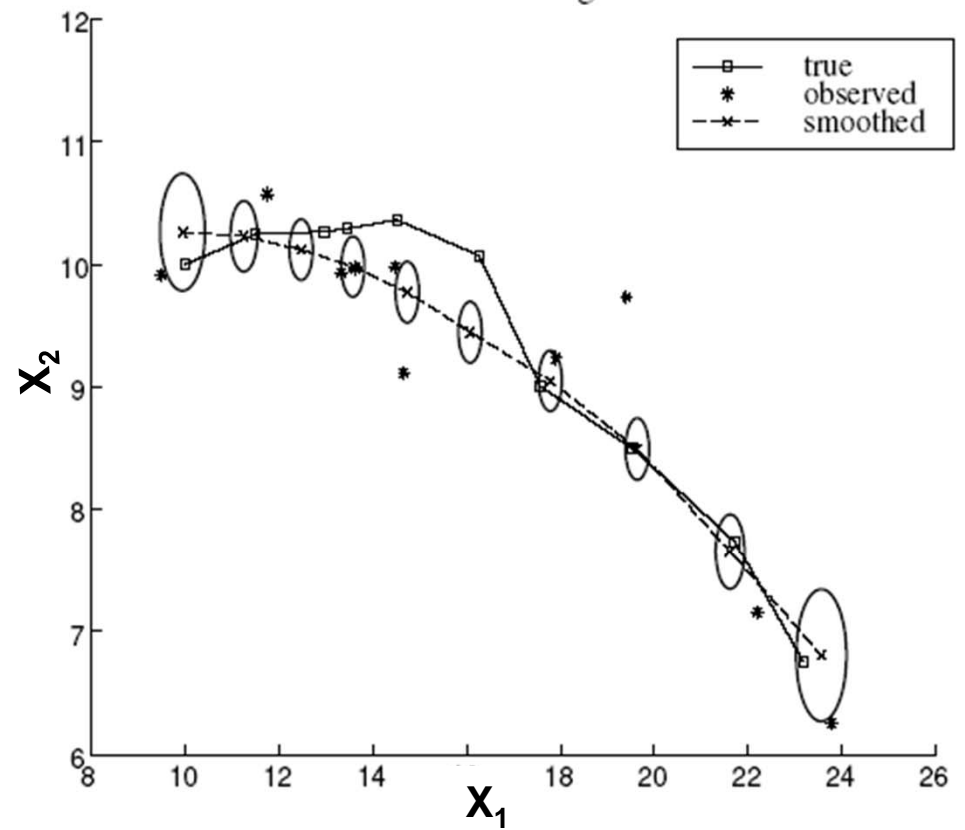
2D tracking



2D filtering

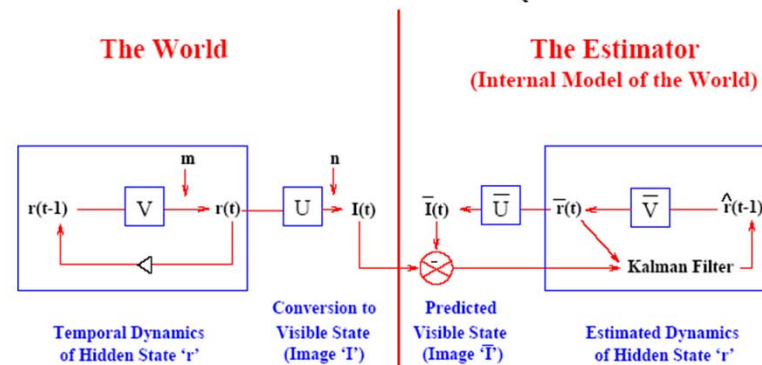
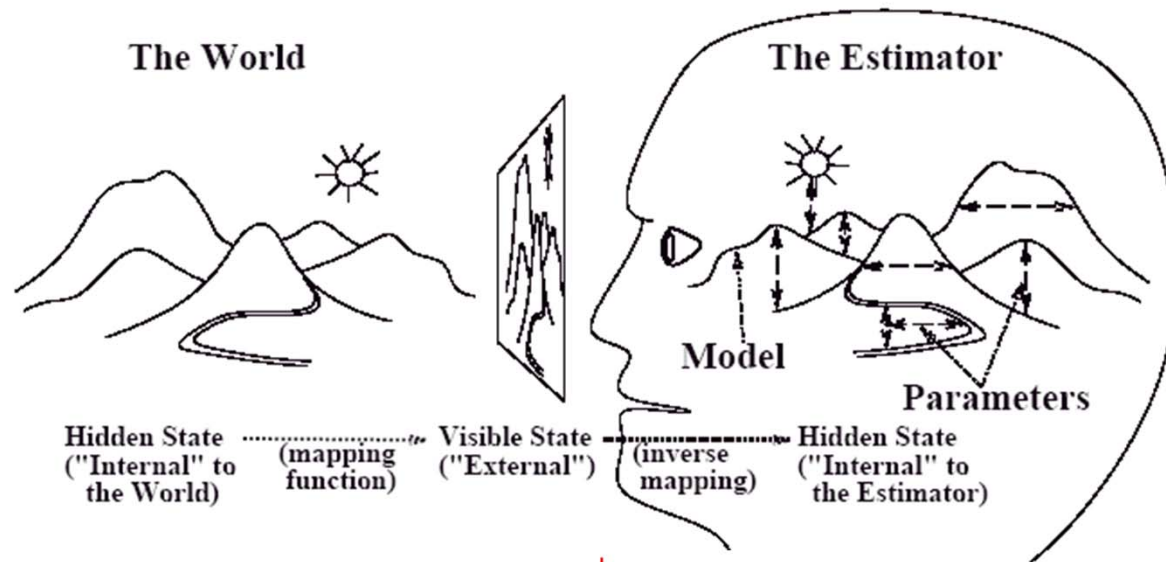


2D smoothing





Kalman filtering in the brain?



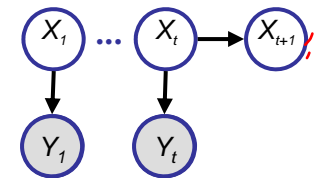


Kalman filtering derivation

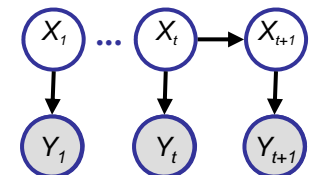
- Since all CPDs are linear Gaussian, the system defines a large multivariate Gaussian.
 - Hence all marginals are Gaussian.
 - Hence we can represent the belief state $p(\mathbf{X}_t | \mathbf{y}_{1:t})$ as a Gaussian with mean μ_t and covariance Σ_t .
 - It is common to work with the inverse covariance (precision) matrix Λ_t ; this is called information form.

- Kalman filtering is a recursive procedure to update the belief state:

- Predict step: compute $p(\mathbf{X}_{t+1} | \mathbf{y}_{1:t})$ from prior belief $p(\mathbf{X}_t | \mathbf{y}_{1:t})$ and dynamical model $p(\mathbf{X}_{t+1} | \mathbf{X}_t)$ --- **time update**



- Update step: compute new belief $p(\mathbf{X}_{t+1} | \mathbf{y}_{1:t+1})$ from prediction $p(\mathbf{X}_{t+1} | \mathbf{y}_{1:t})$, observation \mathbf{y}_{t+1} and observation model $p(\mathbf{y}_{t+1} | \mathbf{X}_{t+1})$ --- **measurement update**

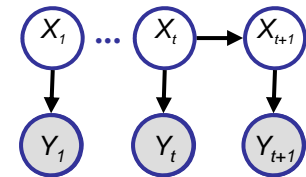
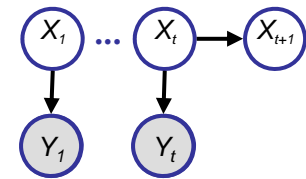


Kalman filtering derivation

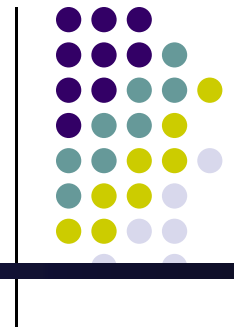


- Kalman filtering is a recursive procedure to update the belief state:

- Predict step: compute $p(\mathbf{X}_{t+1}|\mathbf{y}_{1:t})$ from prior belief $p(\mathbf{X}_t|\mathbf{y}_{1:t})$ and dynamical model $p(\mathbf{X}_{t+1}|\mathbf{X}_t)$ --- **time update**
- Update step: compute new belief $p(\mathbf{X}_{t+1}|\mathbf{y}_{1:t+1})$ from prediction $p(\mathbf{X}_{t+1}|\mathbf{y}_{1:t})$, observation \mathbf{y}_{t+1} and observation model $p(\mathbf{y}_{t+1}|\mathbf{X}_{t+1})$ --- **measurement update**

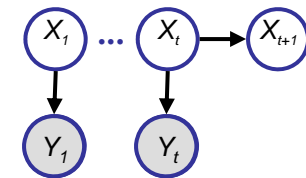


Predict step



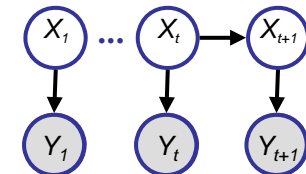
- Dynamical Model: $\mathbf{x}_{t+1} = A\mathbf{x}_t + G\mathbf{w}_t$, $\mathbf{w}_t \sim \mathcal{N}(0; Q)$

- One step ahead prediction of state:



- Observation model: $\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t$, $\mathbf{v}_t \sim \mathcal{N}(0; R)$

- One step ahead prediction of observation:



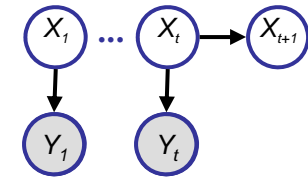
Predict step



- Dynamical Model: $\mathbf{x}_{t+1} = A\mathbf{x}_t + G\mathbf{w}_t$, $\mathbf{w}_t \sim \mathcal{N}(0; Q)$

- One step ahead prediction of state:

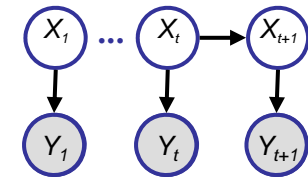
$$\begin{aligned}\hat{\mathbf{x}}_{t+1|t} &= E(\mathbf{X}_{t+1} | \mathbf{y}_1, \dots, \mathbf{y}_t) = A\hat{\mathbf{x}}_{t|t} \\ P_{t+1|t} &= E(\mathbf{X}_{t+1} - \hat{\mathbf{x}}_{t+1|t})(\mathbf{X}_{t+1} - \hat{\mathbf{x}}_{t+1|t})^T | \mathbf{y}_1, \dots, \mathbf{y}_t) \\ &= E(A\mathbf{X}_t + G\mathbf{w}_t - \hat{\mathbf{x}}_{t+1|t})(A\mathbf{X}_t + G\mathbf{w}_t - \hat{\mathbf{x}}_{t+1|t})^T | \mathbf{y}_1, \dots, \mathbf{y}_t) \\ &= AP_{t|t}A + GQG^T\end{aligned}$$



- Observation model: $\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t$, $\mathbf{v}_t \sim \mathcal{N}(0; R)$

- One step ahead prediction of observation:

$$\begin{aligned}E(\mathbf{Y}_{t+1} | \mathbf{y}_1, \dots, \mathbf{y}_t) &= E(C\mathbf{X}_{t+1} + \mathbf{v}_{t+1} | \mathbf{y}_1, \dots, \mathbf{y}_t) = C\hat{\mathbf{x}}_{t+1|t} \\ E(\mathbf{Y}_{t+1} - \hat{\mathbf{y}}_{t+1|t})(\mathbf{Y}_{t+1} - \hat{\mathbf{y}}_{t+1|t})^T | \mathbf{y}_1, \dots, \mathbf{y}_t) &= CP_{t+1|t}C^T + R \\ E(\mathbf{Y}_{t+1} - \hat{\mathbf{y}}_{t+1|t})(\mathbf{X}_{t+1} - \hat{\mathbf{x}}_{t+1|t})^T | \mathbf{y}_1, \dots, \mathbf{y}_t) &= CP_{t+1|t}\end{aligned}$$





Update step

- Summarizing results from previous slide, we have $p(\mathbf{X}_{t+1}, \mathbf{Y}_{t+1} | \mathbf{y}_{1:t}) \sim \mathcal{N}(m_{t+1}, V_{t+1})$, where

$$m_{t+1} = \begin{pmatrix} \hat{x}_{t+1|t} \\ \mathbf{C}\hat{x}_{t+1|t} \end{pmatrix}, \quad V_{t+1} = \begin{pmatrix} P_{t+1|t} & P_{t+1|t}\mathbf{C}^T \\ \mathbf{C}P_{t+1|t} & \mathbf{C}P_{t+1|t}\mathbf{C}^T + R \end{pmatrix},$$

- Remember the formulas for conditional Gaussian distributions:

$$p\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \mid \mu, \Sigma\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \mid \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

$$p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 \mid \mathbf{m}_2^m, \mathbf{V}_2^m)$$

$$\mathbf{m}_2^m = \mu_2$$

$$\mathbf{V}_2^m = \Sigma_{22}$$

$$p(\mathbf{x}_1 | \mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1 \mid \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{m}_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2)$$

$$\mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$



Kalman Filter

- Measurement updates:

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + K_{t+1}(y_{t+1} - C\hat{\mathbf{x}}_{t+1|t})$$

$$P_{t+1|t+1} = P_{t+1|t} - KCP_{t+1|t}$$

- 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:

$$\hat{\mathbf{x}}_{t+1|t} = A\hat{\mathbf{x}}_{t|t}$$

$$P_{t+1|t} = AP_{t|t}A + GQG^T$$

- 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:

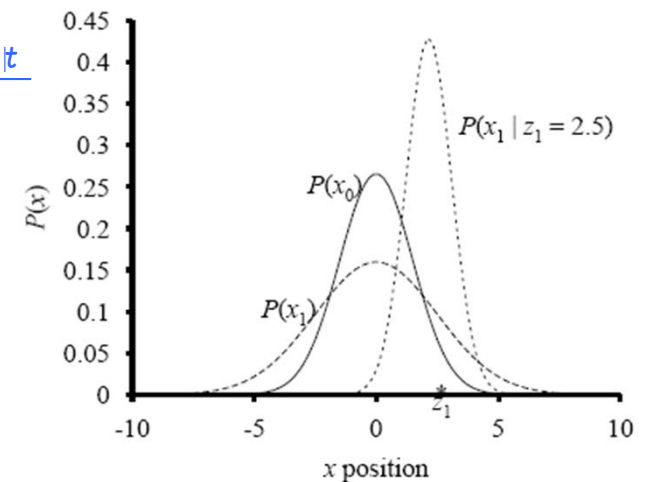
$$x_{t|t-1} = x_{t-1} + w, \quad w \sim \mathcal{N}(0, \sigma_x) \quad z_t = x_t + v, \quad v \sim \mathcal{N}(0, \sigma_z)$$

- KF equations: $P_{t+1|t} = AP_{t|t}A + GQG^T = \sigma_t + \sigma_x$, $\hat{x}_{t+1|t} = A\hat{x}_{t|t} = \hat{x}_{t|t}$

$$K_{t+1} = P_{t+1|t} C^T (CP_{t+1|t} C^T + R)^{-1} = (\sigma_t + \sigma_x)(\sigma_t + \sigma_x + \sigma_z)$$

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_{t+1}(z_{t+1} - C\hat{x}_{t+1|t}) = \frac{(\sigma_t + \sigma_x)z_{t+1} + \sigma_z \hat{x}_{t+1|t}}{\sigma_t + \sigma_x + \sigma_z}$$

$$P_{t+1|t+1} = P_{t+1|t} - KCP_{t+1|t} = \frac{(\sigma_t + \sigma_x)\sigma_z}{\sigma_t + \sigma_x + \sigma_z}$$





KF intuition

- The KF update of the mean is

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_{t+1}(\mathbf{z}_{t+1} - C\hat{x}_{t+1|t}) = \frac{(\sigma_t + \sigma_x)\mathbf{z}_t + \sigma_z\hat{x}_{t|t}}{\sigma_t + \sigma_x + \sigma_z}$$

- the term $(\mathbf{z}_{t+1} - C\hat{x}_{t+1|t})$ is called the *innovation*
- New belief is convex combination of updates from prior and observation, weighted by Kalman Gain matrix:

$$K_{t+1} = P_{t+1|t} C^T (C P_{t+1|t} C^T + R)^{-1}$$

- If the observation is unreliable, σ_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 σ_t) or the process is very unpredictable (large σ_x), we pay more attention to the observation.



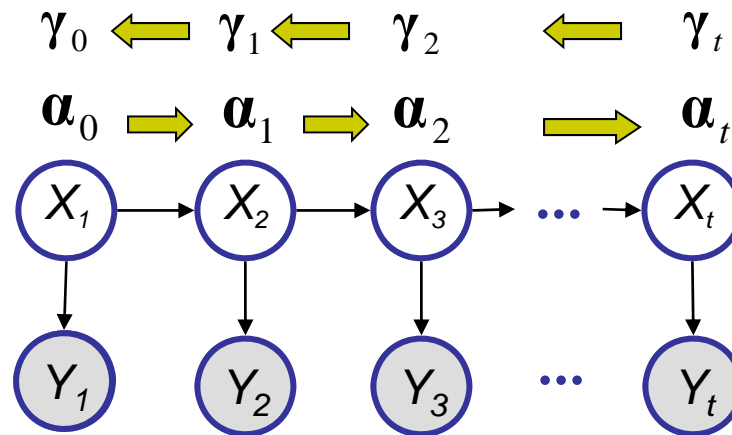
Complexity of one KF step

- Let $X_t \in \mathbb{R}^{N_x}$ and $y_t \in \mathbb{R}^{N_y}$,
- Computing $P_{t+1|t} = AP_{t|t}A + GQG^T$ 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, $\max\{N_x^2, N_y^3\}$



The inference problem 2

- Smoothing \rightarrow given y_1, \dots, y_T , estimate x_t ($t < T$)
 - The Rauch-Tung-Striavel smoother is a way to perform exact off-line inference in an LDS. It is the Gaussian analog of the forwards-backwards (alpha-gamma) algorithm:



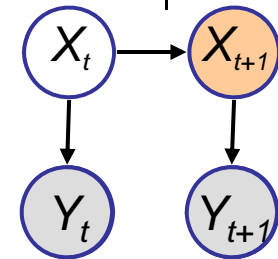
$$p(X_t = i | y_{1:T}) = \gamma_t^i \propto \sum_j \alpha_t^j P(X_{t+1}^j | X_t^i) \gamma_{t+1}^j$$

Rauch-Tung-Strievel smoother



$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + L_t (\hat{\mathbf{x}}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t})$$

$$P_{t|T} = P_{t|t} + L_t (P_{t+1|T} - P_{t+1|t}) L_t^T \quad L_t = P_{t|t} A^T P_{t+1|t}^{-1}$$



- 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: $X_t | y_1, \dots, y_T$

- The trick: $E[X | Z] = E[E[X | Y, Z] | Z]$ (Hw!)

$$\text{Var}[X | Z] = \text{Var}[E[X | Y, Z] | Z] + E[\text{Var}[X | Y, Z] | Z]$$

$$\begin{aligned} \hat{x}_{t|T} &\stackrel{\text{def}}{=} E[X_t | y_1, \dots, y_T] = E[E[X_t | X_{t+1}, y_1, \dots, y_T] | y_1, \dots, y_T] \\ &= E[E[X_t | X_{t+1}, y_1, \dots, y_t] | y_1, \dots, y_T] \\ &= E[X_t | X_{t+1}, y_1, \dots, y_t] \end{aligned}$$

Same for $P_{t|T}$



RTS derivation

- Following the results from previous slide, we need to derive $p(\mathbf{X}_{t+1}, \mathbf{X}_t | \mathbf{y}_{1:t}) \sim \mathcal{N}(m, V)$, where

$$m = \begin{pmatrix} \hat{\mathbf{x}}_{t|t} \\ \hat{\mathbf{x}}_{t+1|t} \end{pmatrix}, \quad V = \begin{pmatrix} P_{t|t} & P_{t|t} \mathbf{A}^T \\ \mathbf{A} P_{t|t} & P_{t+1|t} \end{pmatrix},$$

- all the quantities here are available after a forward KF pass
- Remember the formulas for conditional Gaussian distributions:

$$p\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \mid \mu, \Sigma\right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \mid \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right),$$

$$p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 \mid \mathbf{m}_2^m, \mathbf{V}_2^m)$$

$$\mathbf{m}_2^m = \mu_2$$

$$\mathbf{V}_2^m = \Sigma_{22}$$

$$p(\mathbf{x}_1 | \mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1 \mid \mathbf{m}_{1|2}, \mathbf{V}_{1|2})$$

$$\mathbf{m}_{1|2} = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (\mathbf{x}_2 - \mu_2)$$

$$\mathbf{V}_{1|2} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}$$

- The RTS smoother

$$\begin{aligned} \hat{\mathbf{x}}_{t|T} &= E[\mathbf{X}_t \mid \mathbf{X}_{t+1}, \mathbf{y}_1, \dots, \mathbf{y}_T] \\ &= \hat{\mathbf{x}}_{t|t} + L_t (\hat{\mathbf{x}}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \end{aligned}$$

$$\begin{aligned} P_{t|T} &\stackrel{\text{def}}{=} \text{Var}[\hat{\mathbf{x}}_{t|T} \mid \mathbf{y}_{1:T}] + E[\text{Var}[\mathbf{X}_t \mid \mathbf{X}_{t+1}, \mathbf{y}_{1:t}] \mid \mathbf{y}_{1:T}] \\ &= P_{t|t} + L_t (P_{t+1|T} - P_{t+1|t}) L_t^T \end{aligned}$$



Learning SSMs

- Complete log likelihood

$$\begin{aligned}\ell_c(\theta, \mathcal{D}) &= \sum_n \log p(\mathbf{x}_n, \mathbf{y}_n) = \sum_n \log p(\mathbf{x}_1) + \sum_n \sum_t \log p(\mathbf{x}_{n,t} | \mathbf{x}_{n,t-1}) + \sum_n \sum_t \log p(\mathbf{y}_{n,t} | \mathbf{x}_{n,t}) \\ &= f_1(\mathbf{X}_1; \Sigma_0) + f_2(\{\langle \mathbf{X}_t \mathbf{X}_{t-1}^T \rangle, \langle \mathbf{X}_t \mathbf{X}_t^T \rangle, \langle \mathbf{X}_t \rangle : \forall t\}; A, Q, G) + f_3(\{\langle \mathbf{X}_t \mathbf{X}_t^T \rangle, \langle \mathbf{X}_t \rangle : \forall t\}; C, R)\end{aligned}$$

- EM

- E-step: compute $\langle \mathbf{X}_t \mathbf{X}_{t-1}^T \rangle, \langle \mathbf{X}_t \mathbf{X}_t^T \rangle, \langle \mathbf{X}_t \rangle \mid \mathbf{y}_1, \dots, \mathbf{y}_T$

these quantities can be inferred via KF and RTS filters, etc.,

e.g., $\langle \mathbf{X}_t \mathbf{X}_t^T \rangle \equiv \text{var}(\mathbf{X}_t \mathbf{X}_t^T) + E(\mathbf{X}_t)^2 = P_{t|T} + \hat{\mathbf{x}}_{t|T}^2$

- M-step: MLE using

$$\langle \ell_c(\theta, \mathcal{D}) \rangle = f_1(\langle \mathbf{X}_1 \rangle; \Sigma_0) + f_2(\{\langle \mathbf{X}_t \mathbf{X}_{t-1}^T \rangle, \langle \mathbf{X}_t \mathbf{X}_t^T \rangle, \langle \mathbf{X}_t \rangle : \forall t\}; A, Q, G) + f_3(\{\langle \mathbf{X}_t \mathbf{X}_t^T \rangle, \langle \mathbf{X}_t \rangle : \forall t\}; C, R)$$

c.f., M-step in factor analysis



Nonlinear systems

- In robotics and other problems, the motion model and the observation model are often nonlinear:

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}) + \mathbf{w}_t, \quad \mathbf{y}_t = \mathbf{g}(\mathbf{x}_t) + \mathbf{v}_t$$

- 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: $\mathbf{x}_t' = (\mathbf{x}_t, \theta)$. 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.

$$\mathbf{x}_t = \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1}) + A_{\hat{\mathbf{x}}_{t-1|t-1}}(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1|t-1}) + \mathbf{w}_t$$

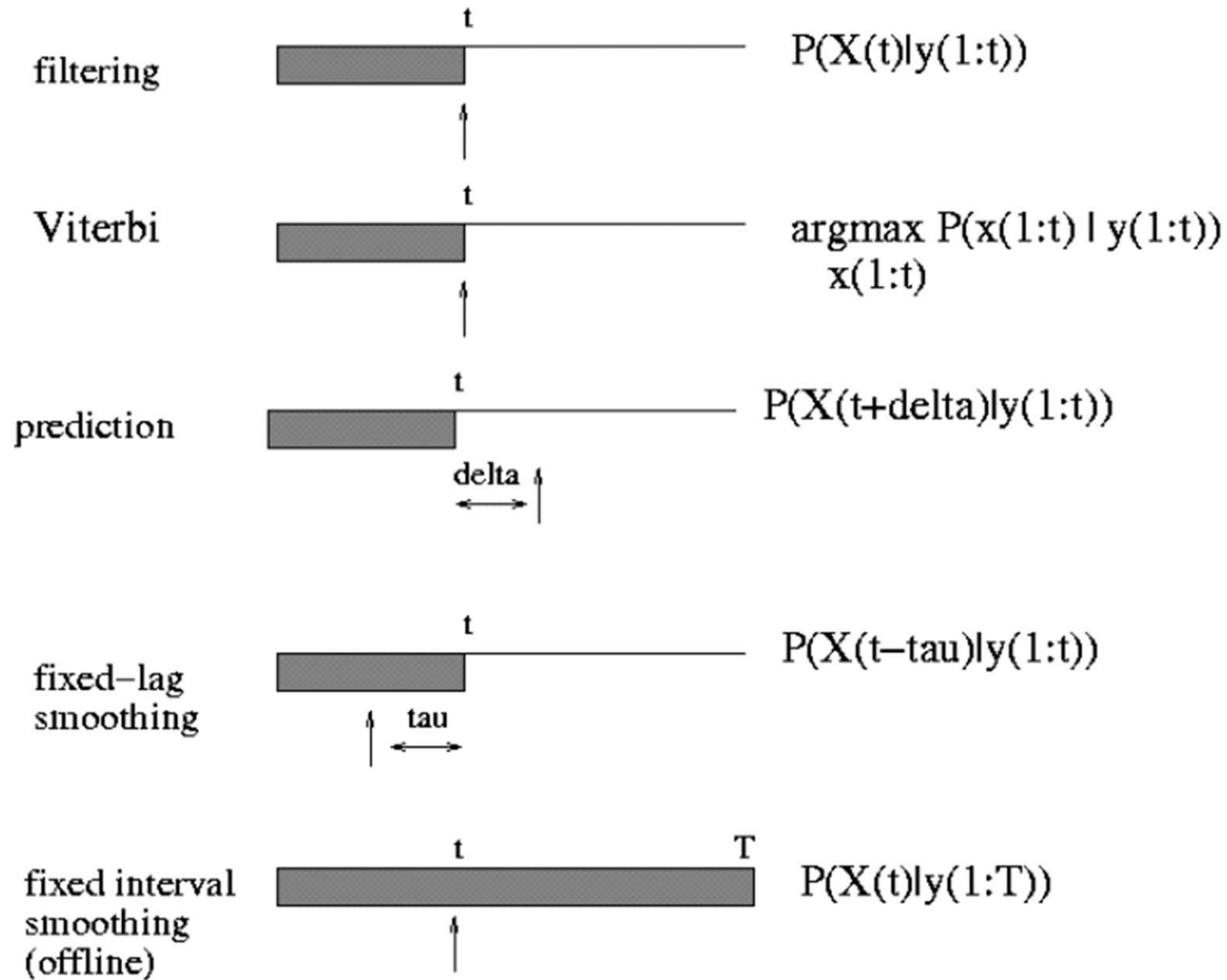
$$\mathbf{y}_t = \mathbf{g}(\hat{\mathbf{x}}_{t|t-1}) + C_{\hat{\mathbf{x}}_{t|t-1}}(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1}) + \mathbf{v}_t$$

where $\hat{\mathbf{x}}_{t|t-1} = \mathbf{f}(\hat{\mathbf{x}}_{t-1|t-1})$ and $A_{\hat{\mathbf{x}}} \stackrel{\text{def}}{=} \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}}$ and $C_{\hat{\mathbf{x}}} \stackrel{\text{def}}{=} \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}}$

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



Online vs offline inference





KF, RLS and LMS

- The KF update of the mean is

$$\hat{\mathbf{x}}_{t+1|t+1} = A\hat{\mathbf{x}}_{t|t} + K_{t+1}(\mathbf{y}_{t+1} - C\hat{\mathbf{x}}_{t+1|t})$$

- Consider the special case where the hidden state is a constant, $\mathbf{x}_t = \theta$, but the “observation matrix” C is a time-varying vector, $C = \mathbf{x}_t^\top$.
 - Hence the observation model at each time slide, $\mathbf{y}_t = \mathbf{x}_t^\top \theta + v_t$, is a linear regression

- We can estimate θ recursively using the Kalman filter:

$$\hat{\theta}_{t+1} = \hat{\theta}_t + P_{t+1}R^{-1}(\mathbf{y}_{t+1} - \mathbf{x}_t^\top \hat{\theta}_t)\mathbf{x}_t$$

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 η_t online using stochastic approximation theory.