Probabilistic Graphical Models

Brown University CSCI 2950-P, Spring 2013
Prof. Erik Sudderth

Lecture 13: Learning in Gaussian Graphical Models, Non-Gaussian Inference, Monte Carlo Methods

Some figures courtesy Michael Jordan's draft textbook,
An Introduction to Probabilistic Graphical Models

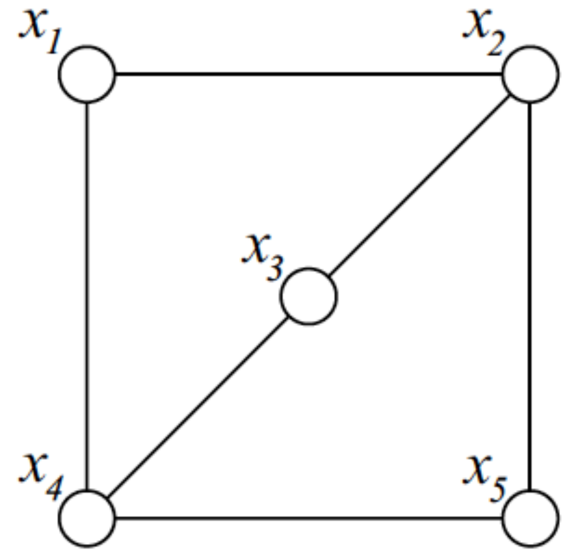
Undirected Gaussian Graphical Models

$$\mathcal{N}(x \mid 0, \Sigma) = \frac{1}{Z} \prod_{(s,t)\in\mathcal{E}} \psi_{st}(x_s, x_t)$$

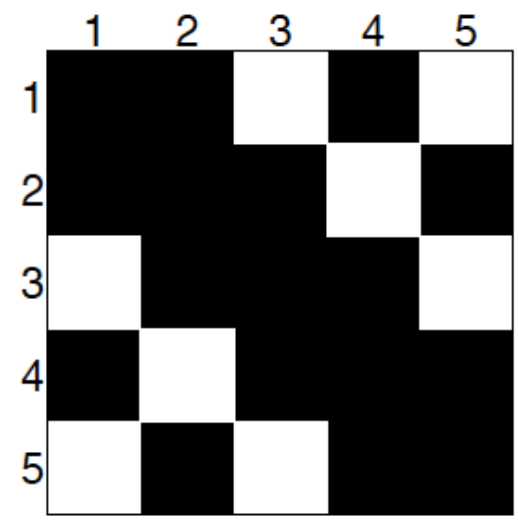
$$Z = \left((2\pi)^N |\Sigma|\right)^{1/2}$$

$$\psi_{s,t}(x_s, x_t) = \exp\left\{ -\frac{1}{2} \begin{bmatrix} x_s^T & x_t^T \end{bmatrix} \begin{bmatrix} J_{s(t)} & J_{s,t} \\ J_{t,s} & J_{t(s)} \end{bmatrix} \begin{bmatrix} x_s \\ x_t \end{bmatrix} \right\}$$

$$\sum_{t \in N(s)} J_{s(t)} = J_{s,s}$$



$$J = \Sigma^{-1}$$



- Undirected Markov properties correspond to *sparse inverse covariance matrices*
- For connected Gaussian MRFs, covariance is usually *dense* (all pairs correlated)
- Number of parameters, and thus learning complexity, reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$

Duality in Gaussian Distributions

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$$\boldsymbol{\mu} = \begin{pmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{pmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix}\boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12}\\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22}\end{pmatrix}$$

Marginals:

$$\begin{aligned} p(\mathbf{x}_1) &= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_{11}) \\ p(\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_{22}) \end{aligned}$$

$$\Sigma_{11}^{-1} = \Lambda_{11} - \Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}$$

Conditionals:

$$p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_{1|2},\boldsymbol{\Sigma}_{1|2})$$

$$\begin{aligned} \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2-\boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\mathbf{x}_2-\boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2}\left(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\mathbf{x}_2-\boldsymbol{\mu}_2)\right) \end{aligned}$$

$$\boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1}$$

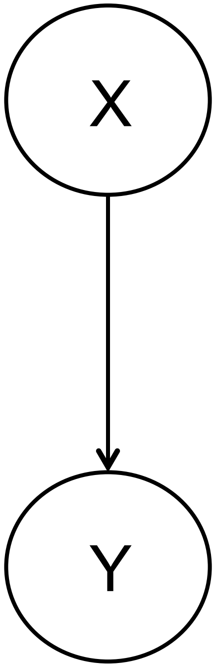
- *Moment parameters:* trivial marginalization, conditioning requires computation
- *Canonical parameters:* trivial conditioning, marginalization requires computation

Linear Gaussian Systems

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \qquad p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \boldsymbol{\Sigma}_y)$$

Marginal Likelihood:

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}_x + \mathbf{b}, \boldsymbol{\Sigma}_y + \mathbf{A}\boldsymbol{\Sigma}_x\mathbf{A}^T)$$



Posterior Distribution:

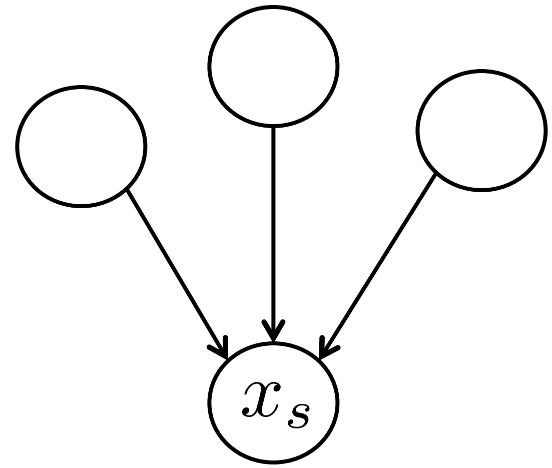
$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y})$$

$$\boldsymbol{\Sigma}_{x|y}^{-1} = \boldsymbol{\Sigma}_x^{-1} + \mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{A}$$

$$\boldsymbol{\mu}_{x|y} = \boldsymbol{\Sigma}_{x|y}[\mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x]$$

Directed Gaussian Graphical Models

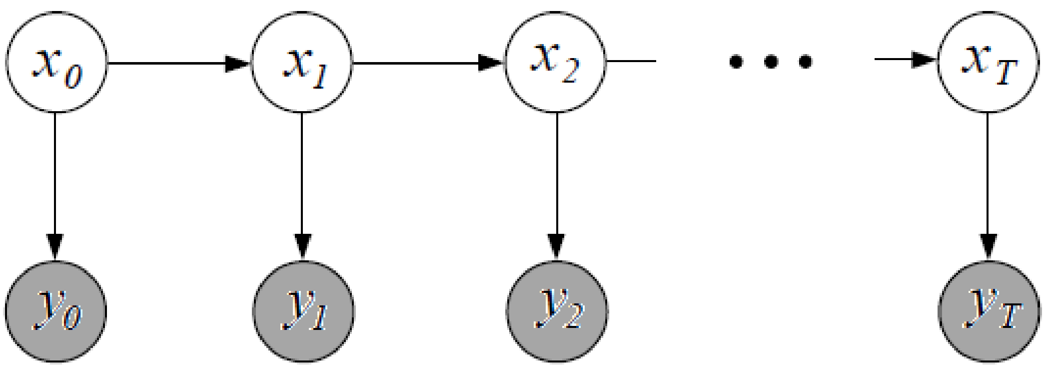
$$\mathcal{N}(x \mid 0, \Sigma) = \prod_{s \in \mathcal{V}} \mathcal{N}(x_s \mid A_s x_{\Gamma(s)}, R_s)$$



- Sequence of locally normalized conditional distributions of each D-dimensional node:

$\boxed{A_s}$ $\boxed{R_s}$

- Linear state space model is a widely used special case:



$$x_{t+1} \sim \mathcal{N}(A_t x_t, Q_t)$$

$$y_t \sim \mathcal{N}(C_t x_t, R_t)$$

- Dimensionality reduction: State smaller than observed vectors
- Rich temporal dynamics: State larger than observed vectors

Probabilistic PCA & Factor Analysis

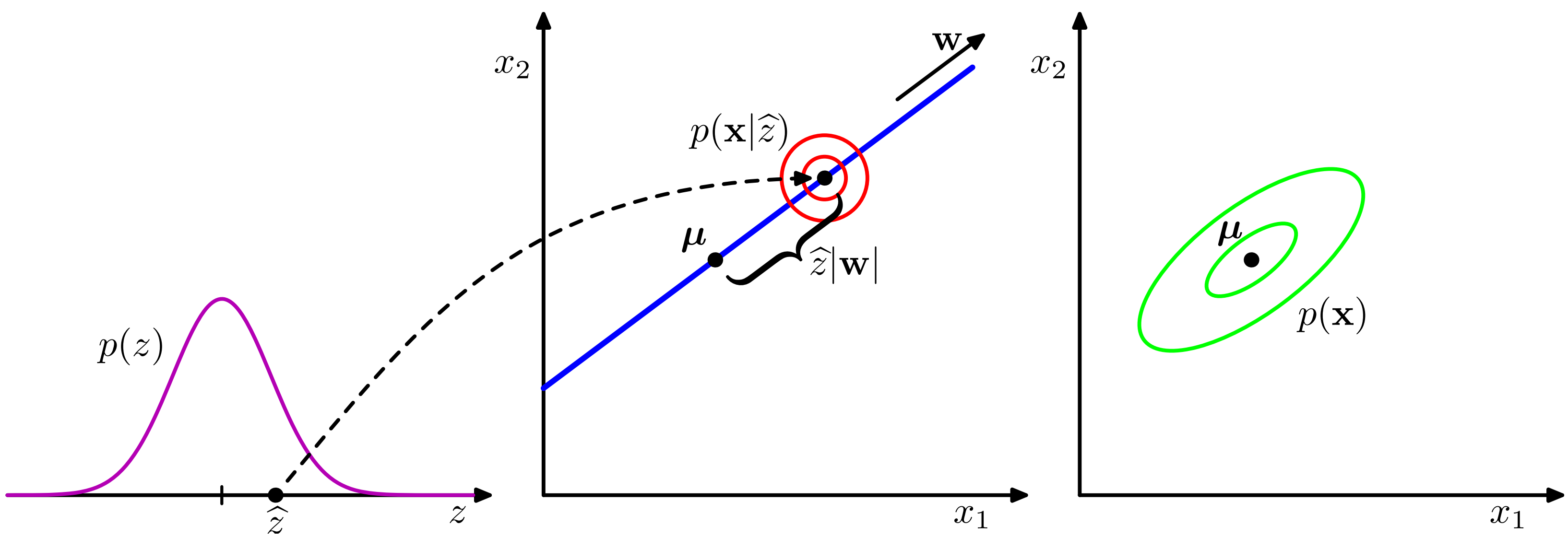
- **Both Models:** Data is a linear function of low-dimensional latent coordinates, plus Gaussian noise

$$p(x_i \mid z_i, \theta) = \mathcal{N}(x_i \mid W z_i + \mu, \Psi) \qquad p(z_i \mid \theta) = \mathcal{N}(z_i \mid 0, I)$$

$$p(x_i \mid \theta) = \mathcal{N}(x_i \mid \mu, WW^T + \Psi)$$

low rank covariance parameterization

- **Factor analysis:** Ψ is a general diagonal matrix
- **Probabilistic PCA:** $\Psi = \sigma^2 I$ is a multiple of identity matrix



C. Bishop, Pattern Recognition & Machine Learning

Expectation Maximization (EM)

$$\ln p(x \mid \theta) = \ln \left(\int_z p(x, z \mid \theta)\, dz \right)$$

$$\ln p(x \mid \theta) \geq \int_z q(z) \ln p(x, z \mid \theta)\, dz - \int_z q(z) \ln q(z)\, dz \triangleq \mathcal{L}(q, \theta)$$

- **Initialization:** Randomly select starting parameters $\theta^{(0)}$
- **E-Step:** Given parameters, find posterior of hidden data

$$q^{(t)} = \arg\max_q \mathcal{L}(q, \theta^{(t-1)})$$

- **M-Step:** Given posterior distributions, find likely parameters

$$\theta^{(t)} = \arg\max_\theta \mathcal{L}(q^{(t)}, \theta)$$

- **Iteration:** Alternate E-step & M-step until convergence

EM: Expectation Step

$$\ln p(x \mid \theta) \geq \int_z q(z) \ln p(x, z \mid \theta)\; dz - \int_z q(z) \ln q(z)\; dz \triangleq \mathcal{L}(q, \theta)$$

$$q^{(t)} = \arg\max_q \mathcal{L}(q, \theta^{(t-1)})$$

- General solution, for any probabilistic model:

$$q^{(t)}(z) = p(z \mid x, \theta^{(t-1)})$$

posterior distribution
given current parameters

- For factor analysis and probabilistic PCA these are Gaussian:

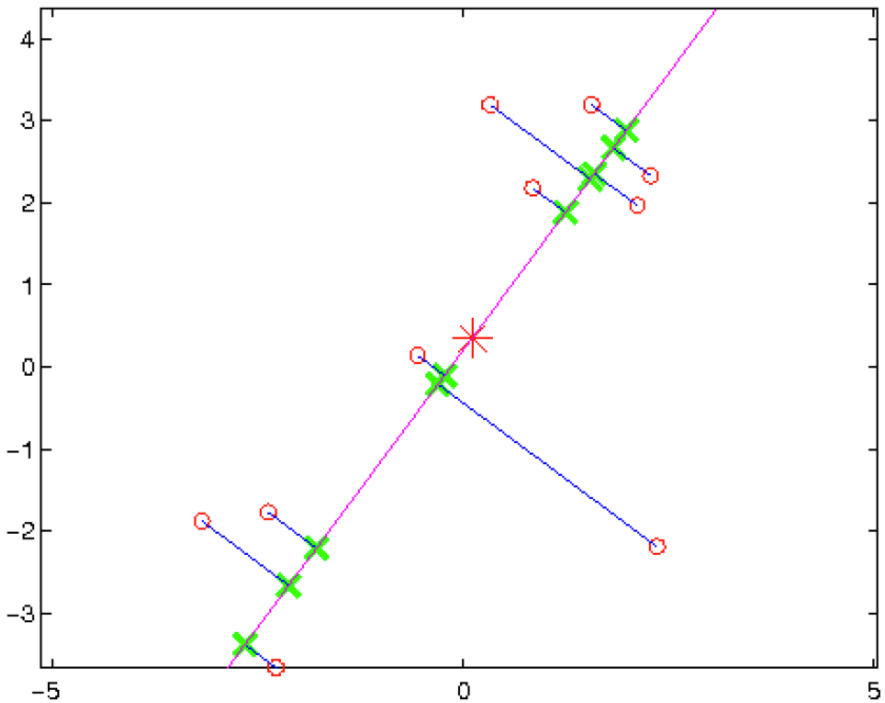
$$p(z \mid x, \theta) = \prod_{i=1}^{N} p(z_i \mid x_i, \theta) \qquad \theta = \{W, \mu, \Psi\}$$

$$p(z_i \mid x_i, W, \mu, \Psi) = \mathcal{N}(z_i \mid \Sigma_i W^T \Psi^{-1}(x_i - \mu), \Sigma_i)$$

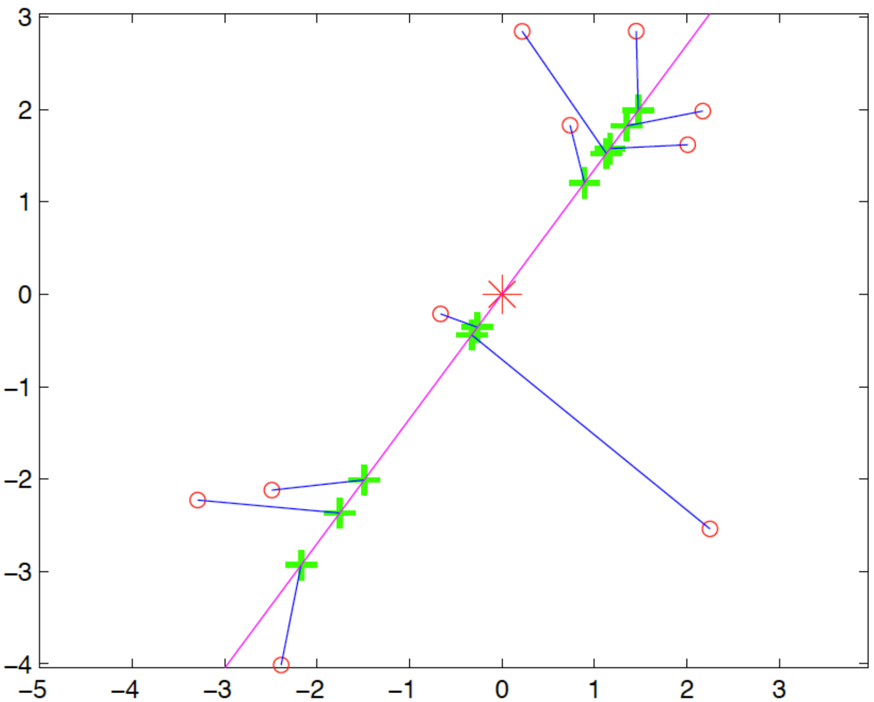
$$\Sigma_i^{-1} = I + W^T \Psi^{-1} W$$

PCA versus Probabilistic PCA

$$p(z_i \mid x_i, W, \mu, \Psi) = \mathcal{N}(z_i \mid \Sigma_i W^T \Psi^{-1}(x_i - \mu), \Sigma_i) \qquad \Sigma_i^{-1} = I + W^T \Psi^{-1} W$$



Standard PCA
(orthogonal projection)



Probabilistic PCA
(shrunk towards mean)

- Maximum likelihood estimates of probabilistic PCA parameters are equal to the classic PCA eigenvector solution
- For classical PCA, optimal embedding is orthogonal projection
- For PPCA, latent coordinates are biased towards mean (zero)

EM: Maximization Step

$$\ln p(x \mid \theta) \geq \int_z q(z) \ln p(x, z \mid \theta)\ dz - \int_z q(z) \ln q(z)\ dz \triangleq \mathcal{L}(q, \theta)$$

$$\theta^{(t)} = \arg\max_\theta \mathcal{L}(q^{(t)}, \theta) = \arg\max_\theta \int_z q(z) \ln p(x, z \mid \theta)\ dz$$

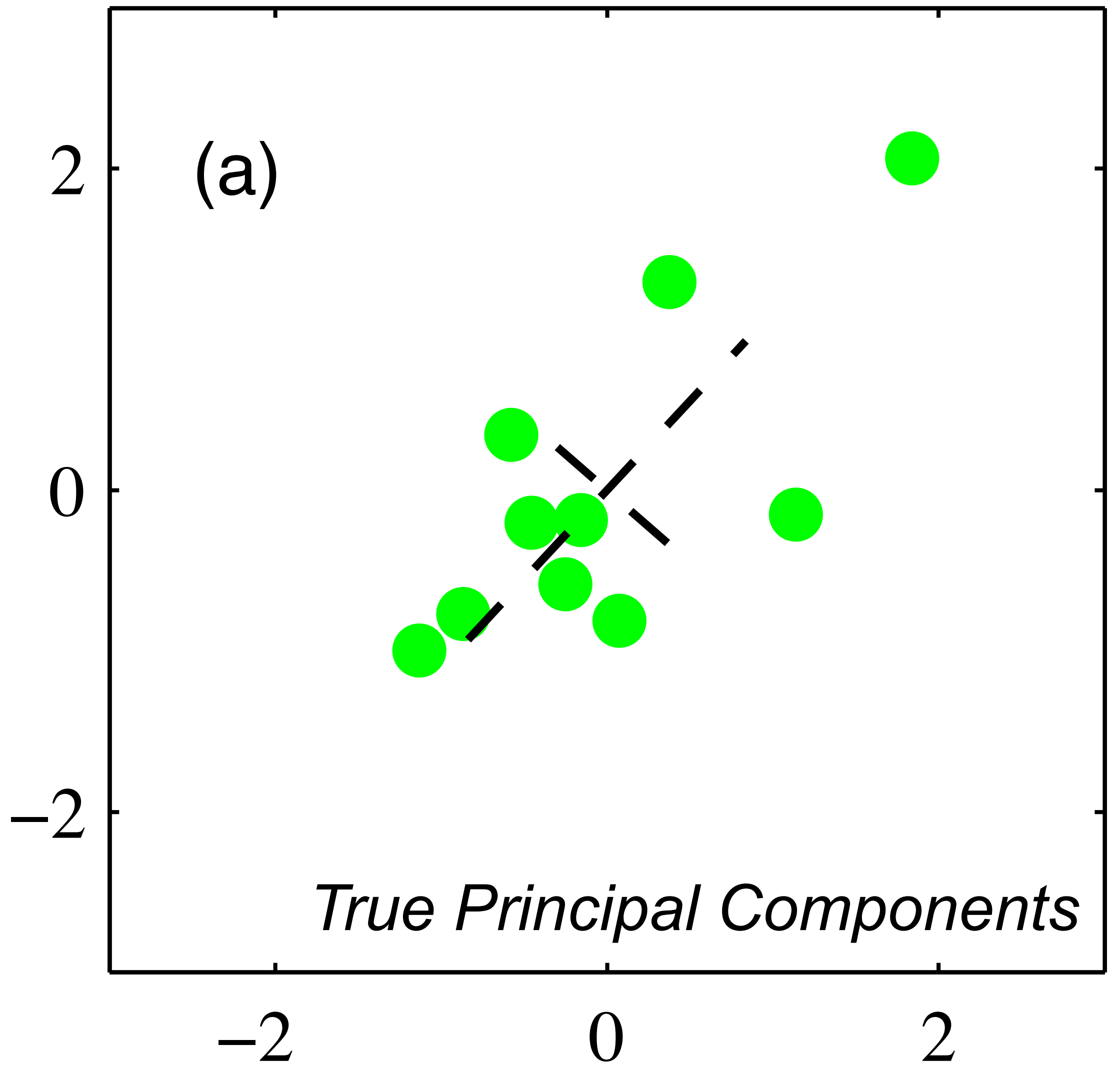
- Unlike E-step, no simplified general solution
- For factor analysis and probabilistic PCA, these reduce to *weighted linear regression* problems

$$-\ln p(x, z \mid \theta) = C + \frac{1}{2} \sum_{i=1}^{N} \left[||z_i||^2 + D \log \sigma^2 + \sigma^{-2} ||x_i - W z_i - \mu||^2 \right]$$

$$\Psi = \sigma^2 I$$

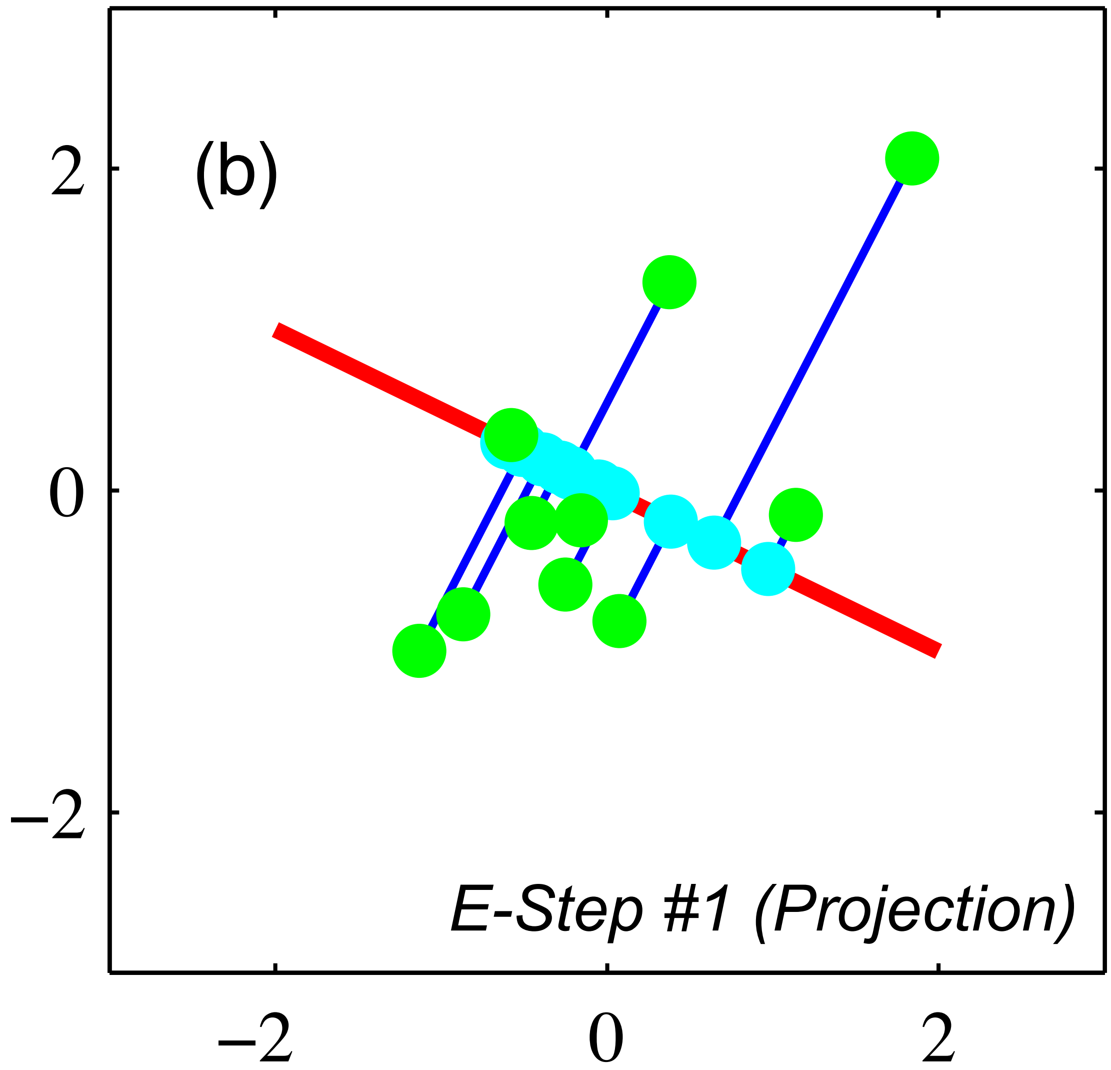
$$\min_{W, \mu, \Psi} \frac{1}{2} \sum_{i=1}^{N} \left[D \log \sigma^2 + \sigma^{-2} \mathbb{E}_q [||x_i - W z_i - \mu||^2] \right]$$

EM Algorithm for Probabilistic PCA



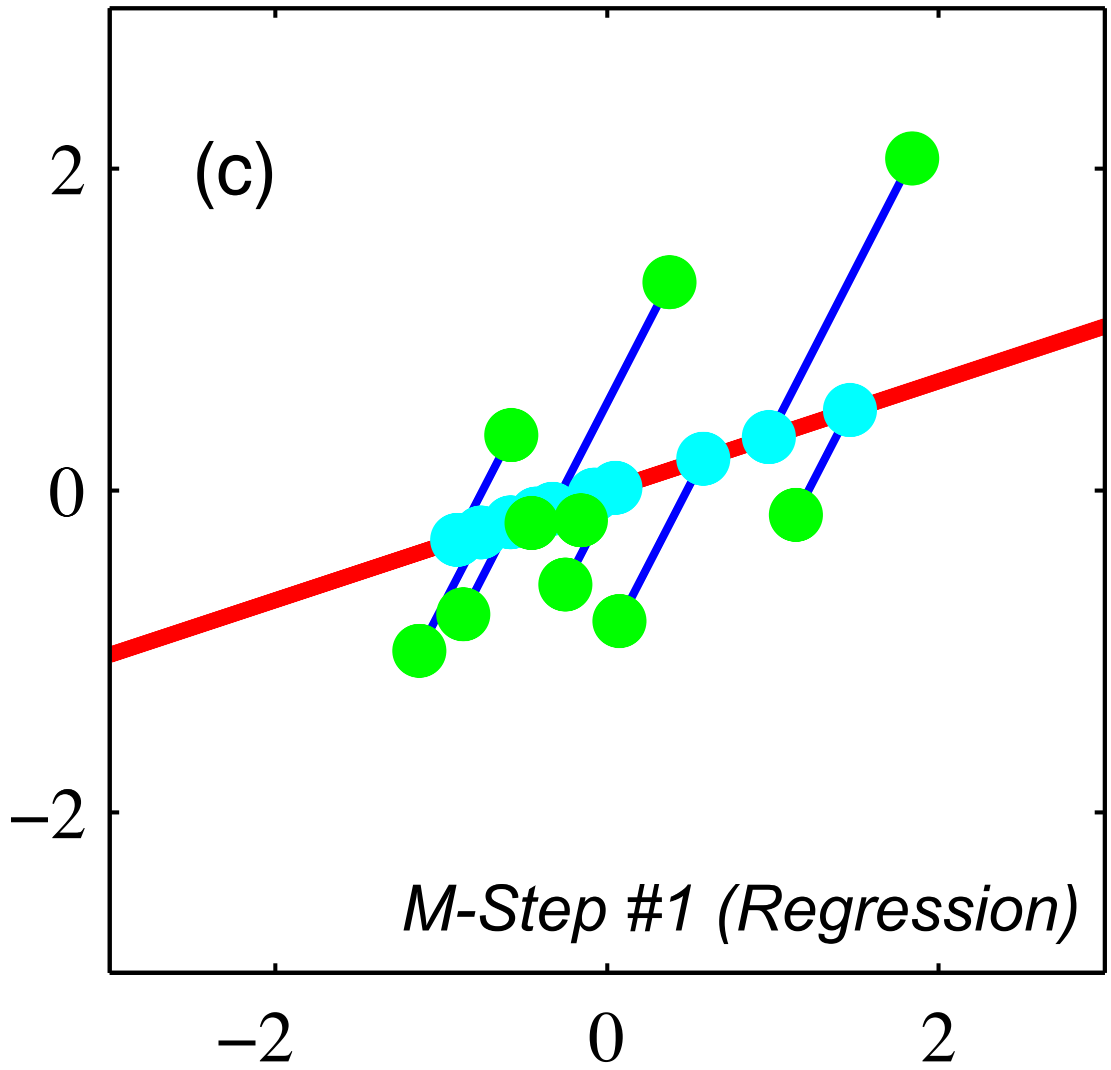
C. Bishop, Pattern Recognition & Machine Learning

EM Algorithm for Probabilistic PCA



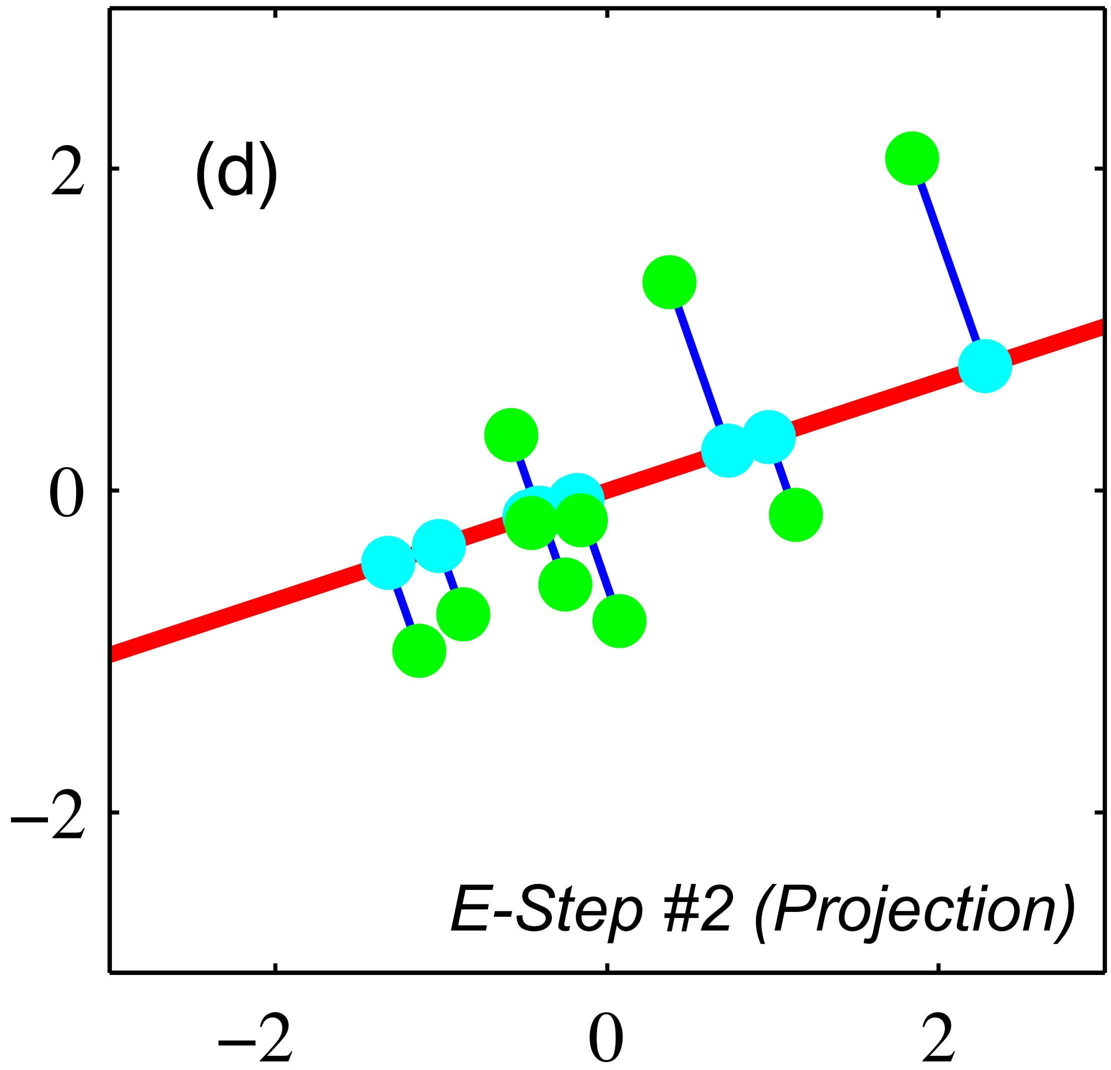
C. Bishop, Pattern Recognition & Machine Learning

EM Algorithm for Probabilistic PCA



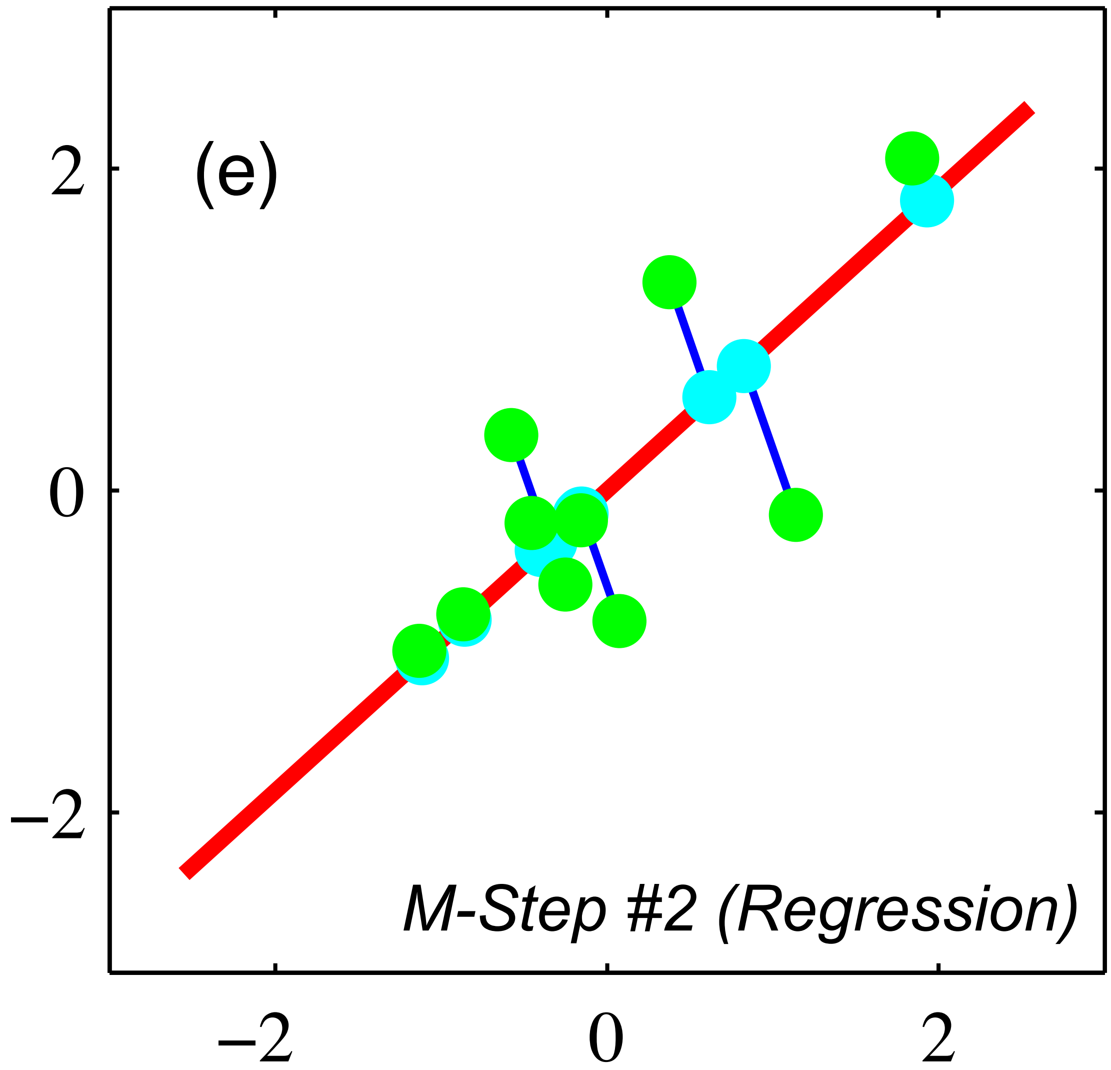
C. Bishop, Pattern Recognition & Machine Learning

EM Algorithm for Probabilistic PCA



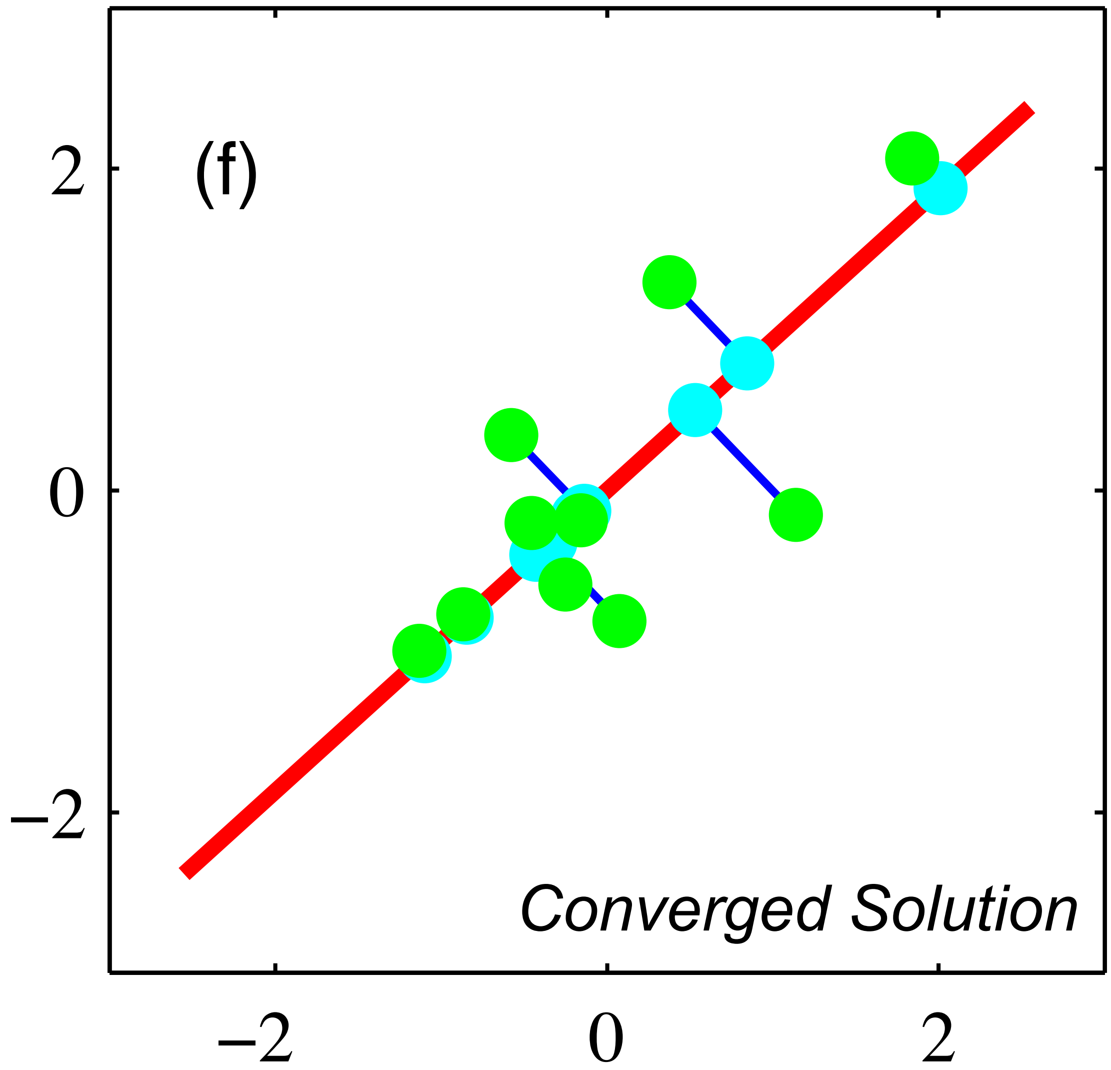
C. Bishop, Pattern Recognition & Machine Learning

EM Algorithm for Probabilistic PCA



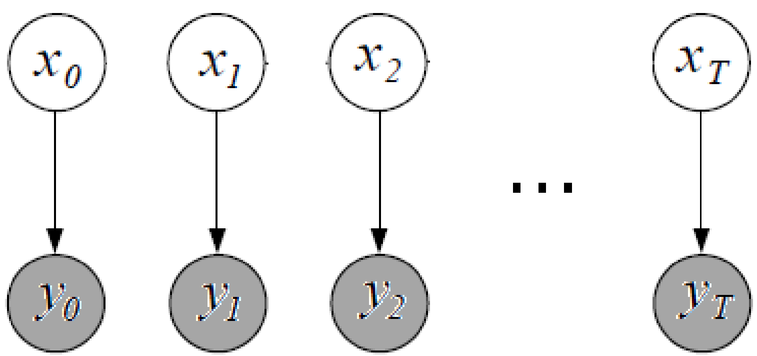
C. Bishop, Pattern Recognition & Machine Learning

EM Algorithm for Probabilistic PCA



C. Bishop, Pattern Recognition & Machine Learning

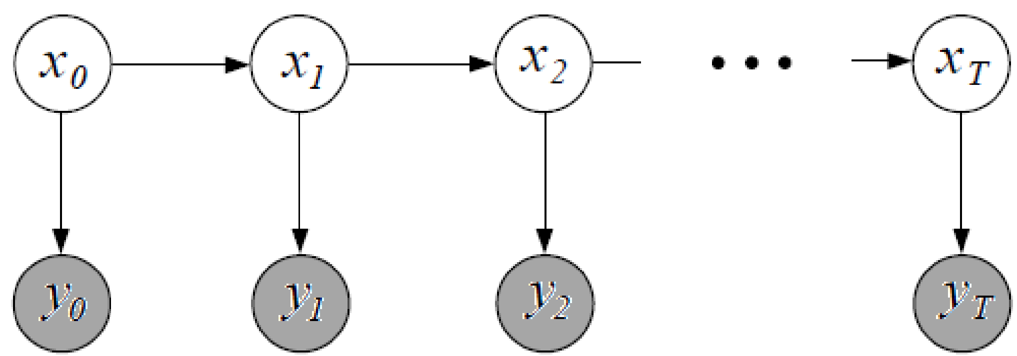
EM for Linear State Space Models



Factor Analysis or PPCA

$$x_t \sim N(0, I)$$

$$y_t \mid x_t \sim N(F x_t, \Phi)$$



Linear-Gaussian State Space Model

$$x_t \mid x_{t-1} \sim N(A x_{t-1}, G Q G^T)$$

$$y_t \mid x_t \sim N(C x_t, R)$$

E-Step:

- Independently find Gaussian posteriors for each observation

M-Step:

- Weighted linear regression to map embeddings to observations

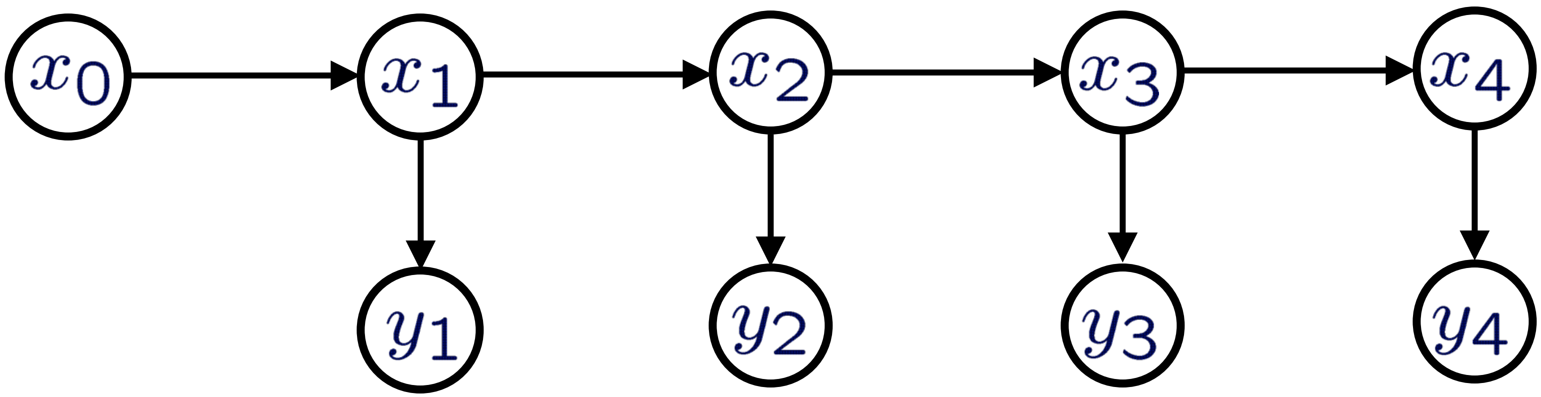
E-Step:

- Determine posterior marginals via Gaussian BP (Kalman smoother)

M-Step:

- Observation regression identical to factor analysis M-step
- Separate dynamics regression

Linear State Space Models



$x_t \in \mathbb{R}^d$

$y_t \in \mathbb{R}^k$

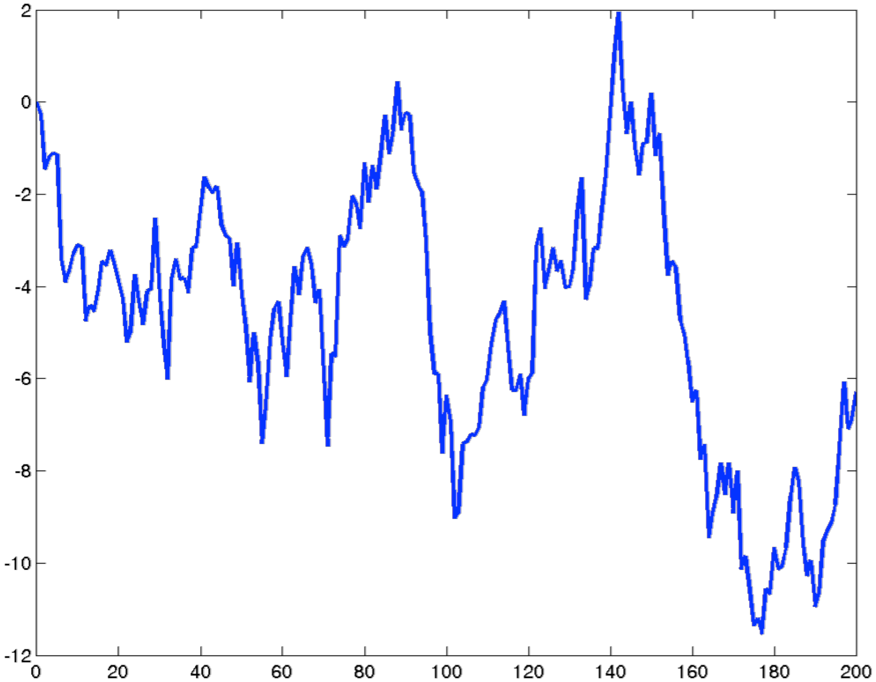
$$x_{t+1} = Ax_t + w_t \qquad w_t \sim \mathcal{N}(0, Q)$$

$$y_t = Cx_t + v_t \qquad v_t \sim \mathcal{N}(0, R)$$

- States & observations jointly Gaussian:
  - All marginals & conditionals Gaussian
  - Linear transformations remain Gaussian

Simple Linear Dynamics

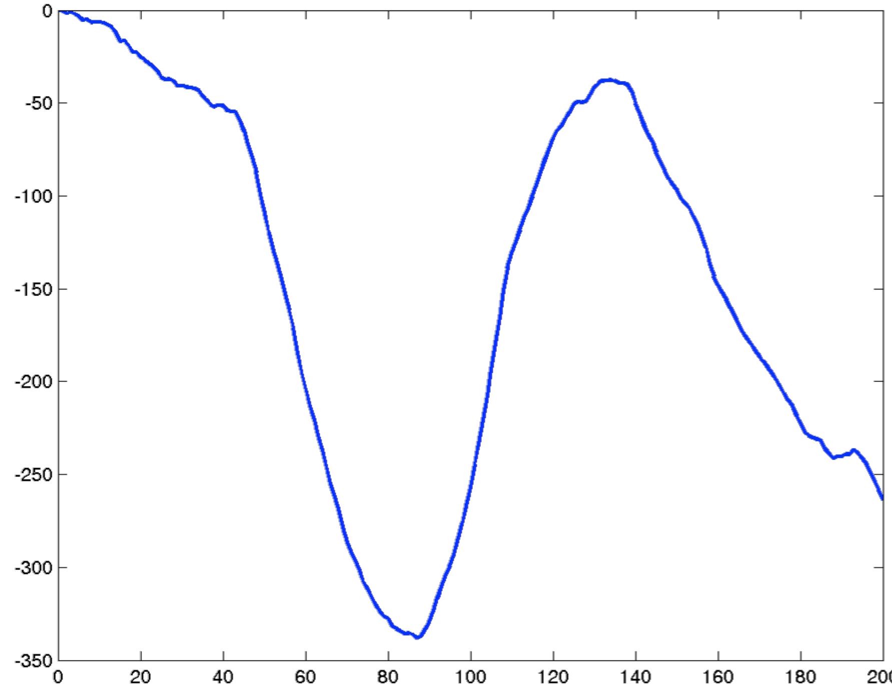
Brownian Motion



Time

$$x_{t+1} = x_t + w_t$$

Constant Velocity

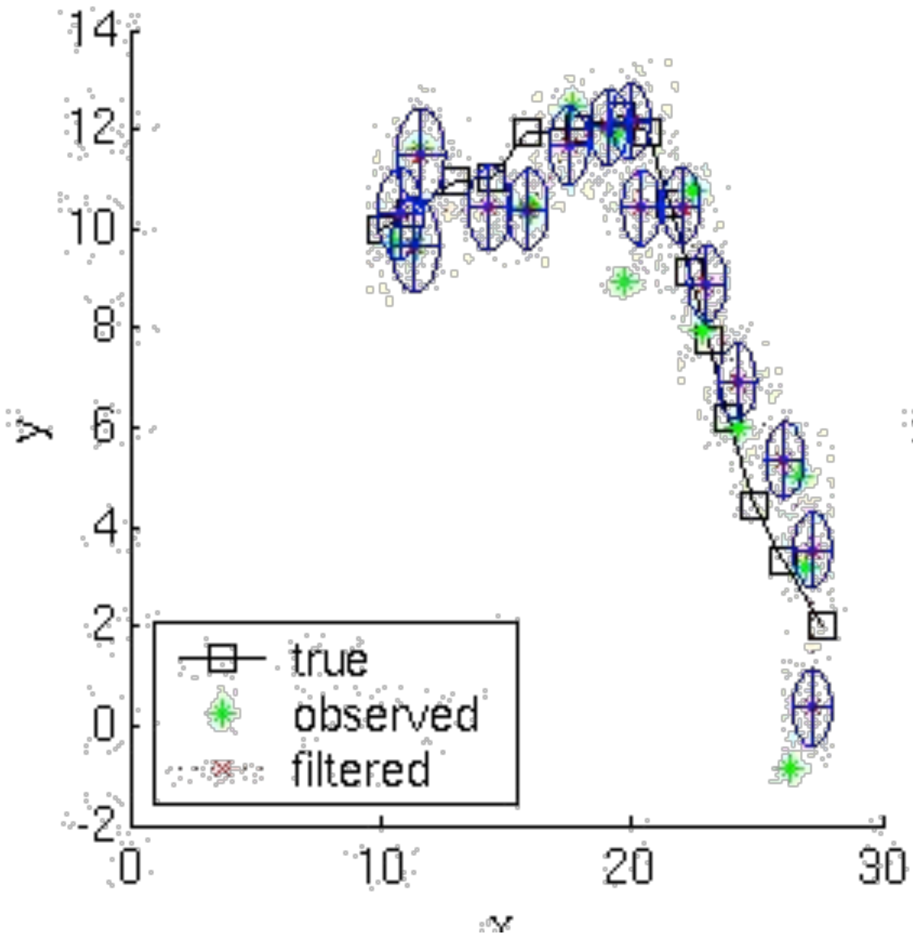


Time

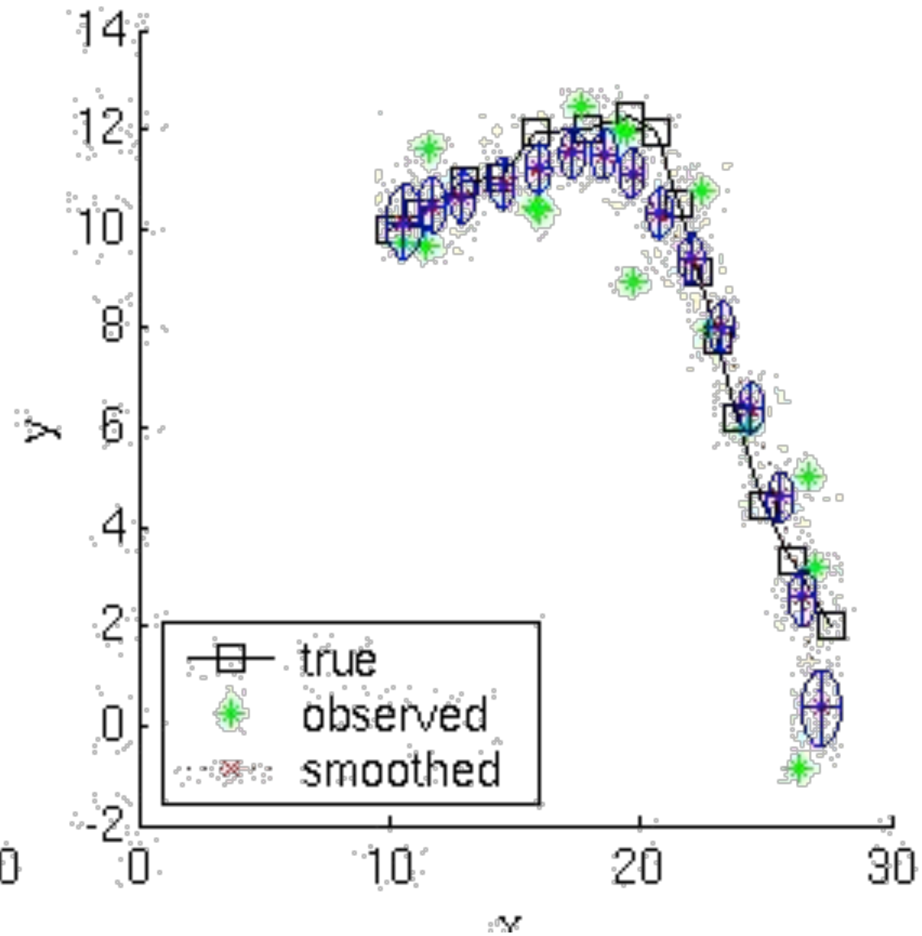
$$\begin{bmatrix} x_{t+1} \\ \delta_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ \delta_t \end{bmatrix} + w_t$$

Constant Velocity Tracking

Kalman Filter

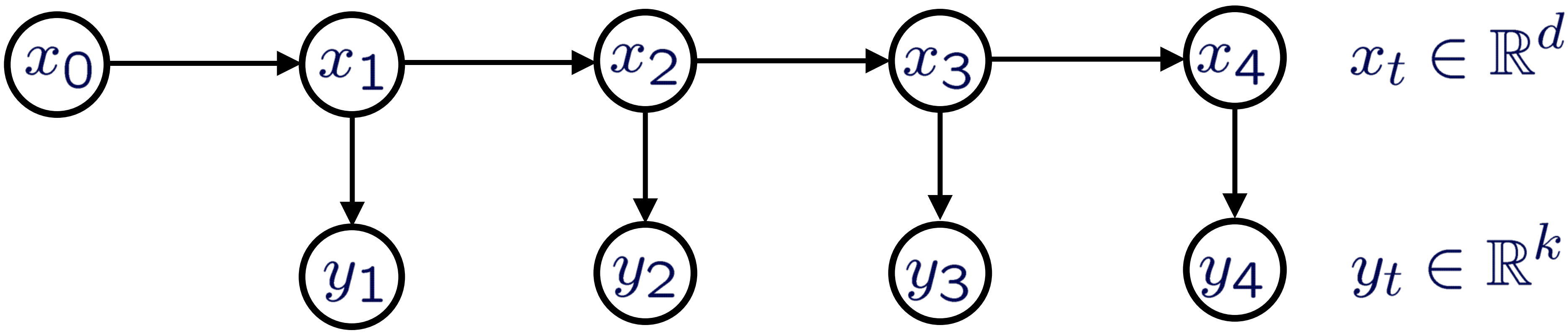


Kalman Smoother



(K. Murphy, 1998)

Nonlinear State Space Models

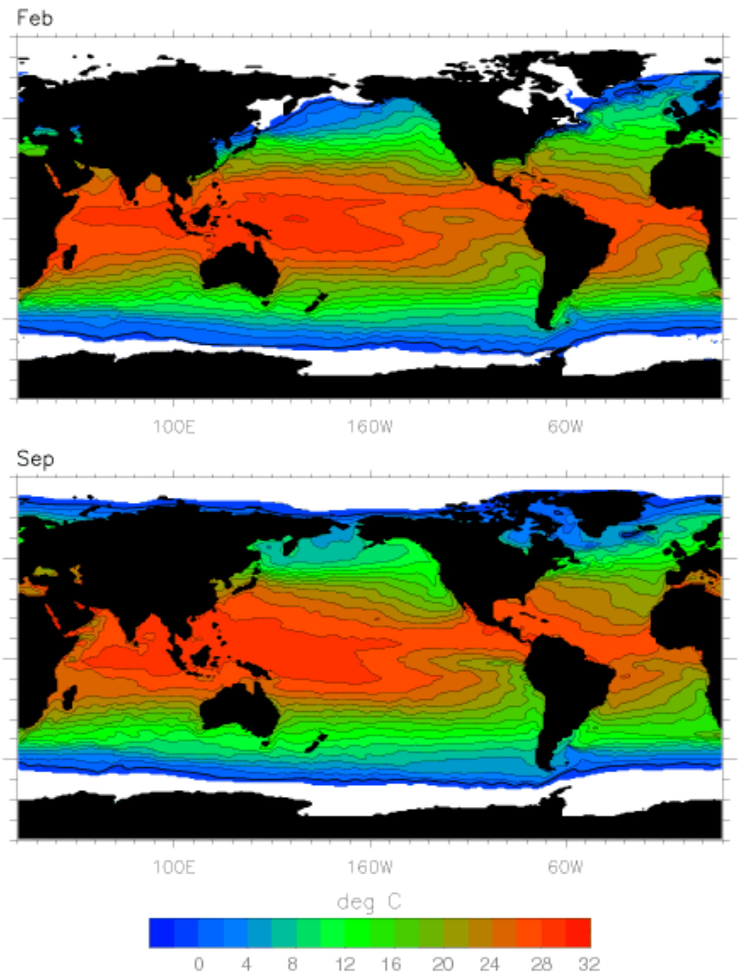


$$x_{t+1} = f(x_t, w_t) \qquad w_t \sim \mathcal{F}$$

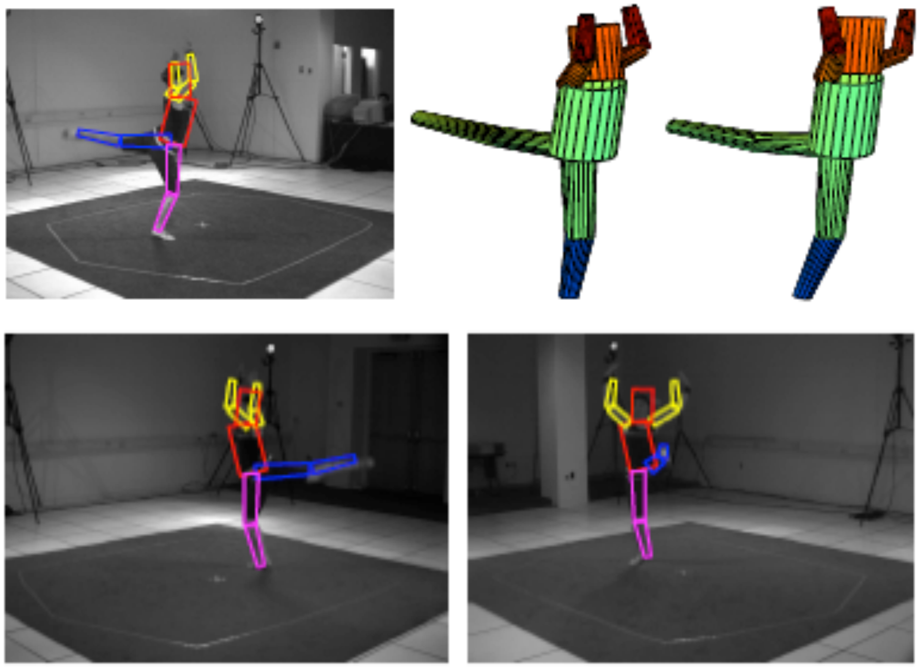
$$y_t = g(x_t, v_t) \qquad v_t \sim \mathcal{G}$$

- State dynamics and measurements given by potentially complex *nonlinear functions*
- Noise sampled from *non-Gaussian* distributions

Examples of Nonlinear Models

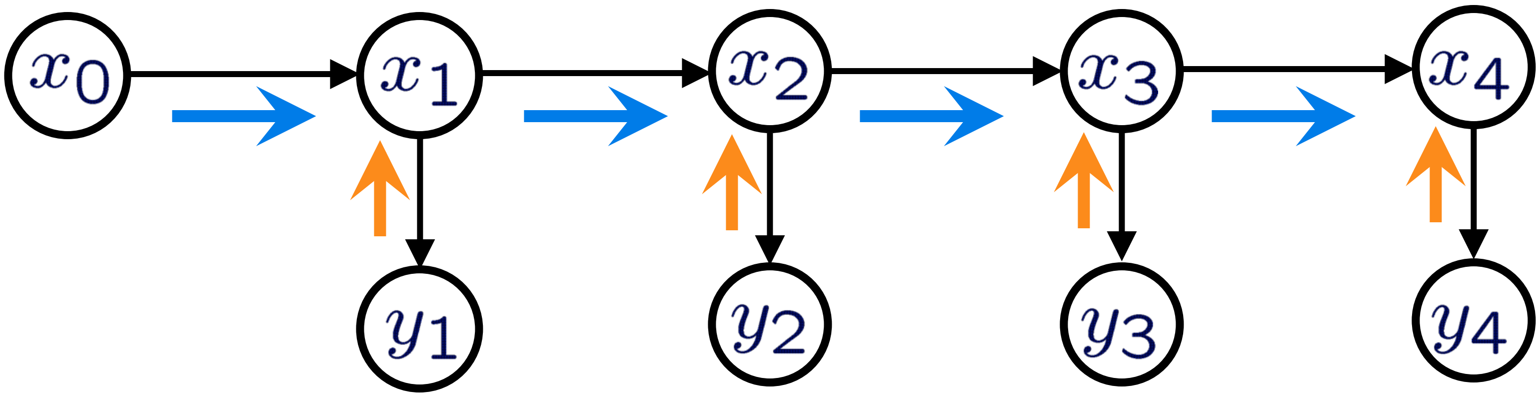


Dynamics implicitly determined by geophysical simulations



Observed image is a complex function of the 3D pose, other nearby objects & clutter, lighting conditions, camera calibration, etc.

Nonlinear Filtering



$$p(x_t \mid y_1, \ldots, y_{t-1}) = \tilde{q}_t(x_t)$$

$$p(x_t \mid y_1, \ldots, y_t) = q_t(x_t)$$

Prediction:

$$\tilde{q}_t(x_t) = \int p(x_t \mid x_{t-1}) q_{t-1}(x_{t-1}) \, dx_{t-1}$$

Update:

$$q_t(x_t) = \frac{1}{Z_t} \tilde{q}_t(x_t) p(y_t \mid x_t)$$

Approximate Nonlinear Filters

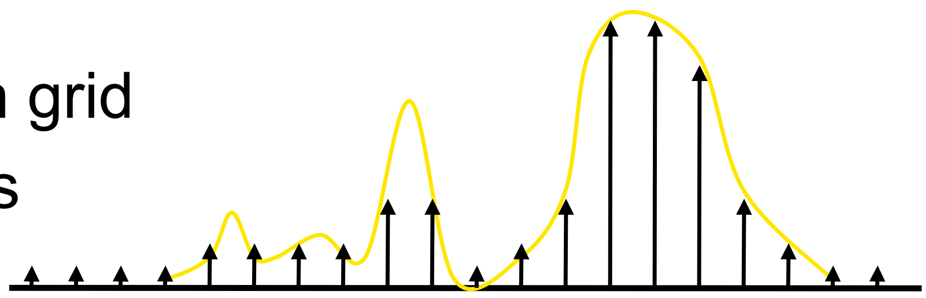
$$q_t(x_t) \propto p(y_t \mid x_t) \cdot \int p(x_t \mid x_{t-1}) q_{t-1}(x_{t-1}) \, dx_{t-1}$$

- No direct *representation* of continuous functions, or closed form for the prediction *integral*
- Big literature on approximate filtering:
  - Histogram filters
  - Extended & unscented Kalman filters
  - Particle filters
  - …

Nonlinear Filtering Taxonomy

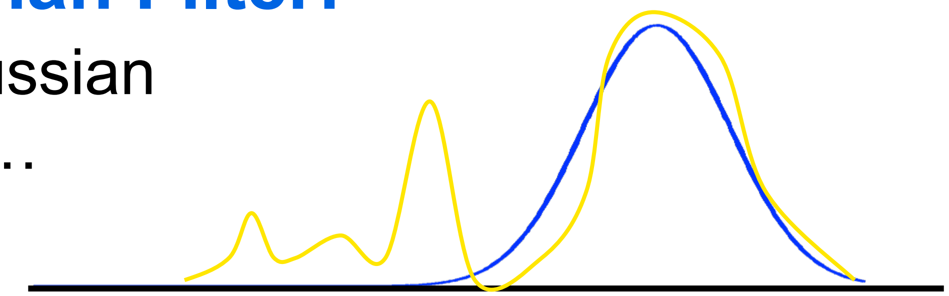
Histogram Filter:

- Evaluate on fixed discretization grid
- Only feasible in low dimensions
- Expensive or inaccurate



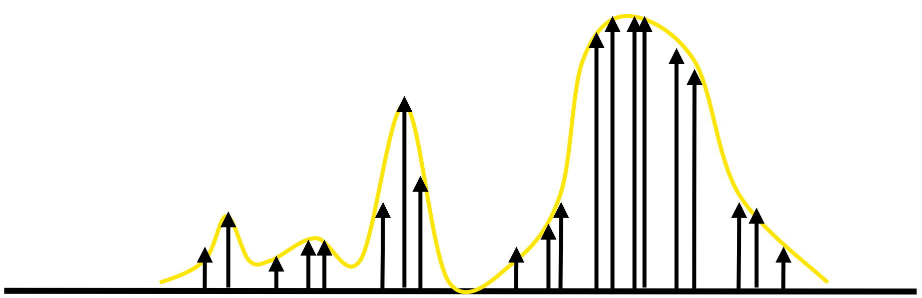
Extended/Unscented Kalman Filter:

- Approximate posterior as Gaussian via linearization, quadrature, …
- Inaccurate for multimodal posterior distributions



Particle Filter:

- Dynamically evaluate states with highest probability
- Monte Carlo approximation



Monte Carlo Methods

$$\mathbb{E}[f] = \int f(z)p(z)\, dz \approx \frac{1}{L}\sum_{\ell=1}^{L} f(z^{(\ell)}) \qquad z^{(\ell)} \sim p(z)$$

Estimation of expected model properties via simulation

Provably good if *L* sufficiently large:

- Unbiased for any sample size
- Variance inversely proportional to sample size (and independent of dimension of space)
- Weak law of large numbers
- Strong law of large numbers
- **Problem:** Drawing samples from complex distributions…

Alternatives for hard problems:

- Importance sampling
- Markov chain Monte Carlo (MCMC)