Classic HMM tutorial – see class website:
*L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. of the IEEE, Vol.77, No.2, pp.257--286, 1989.

Time series, HMMs, Kalman Filters

Machine Learning – 10701/15781

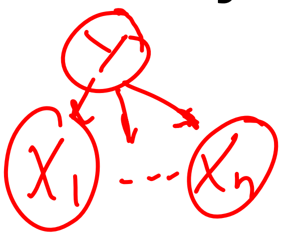
Carlos Guestrin
Carnegie Mellon University

March 28th, 2005

Adventures of our BN hero

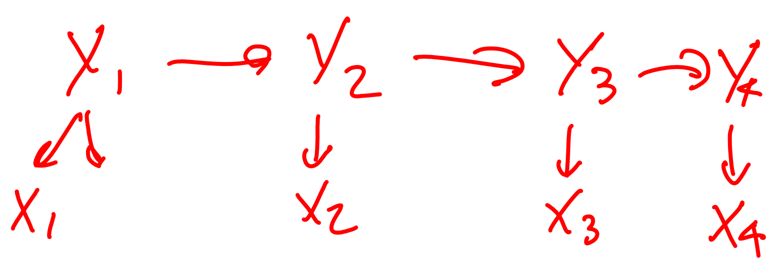
- Compact representation for probability distributions
- Fast inference
- Fast learning

1. Naïve Bayes

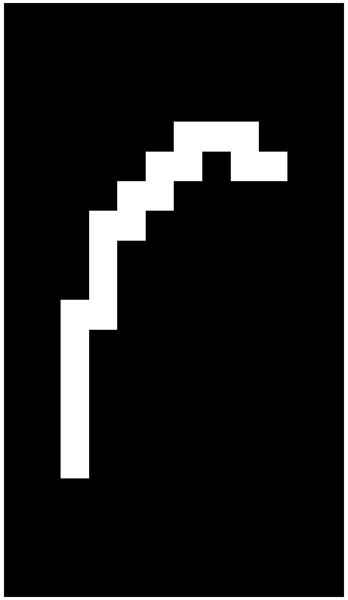


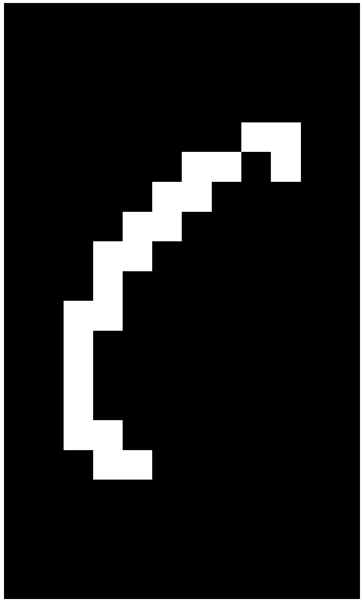
- But… Who are the most popular kids?

2 and 3.
Hidden Markov models (HMMs)
Kalman Filters

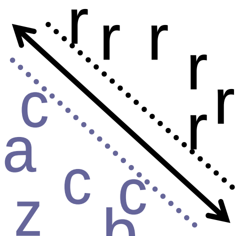


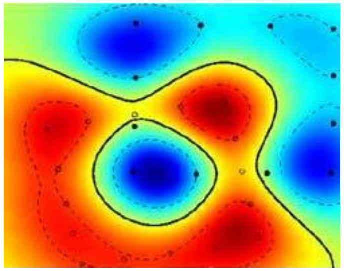
Handwriting recognition





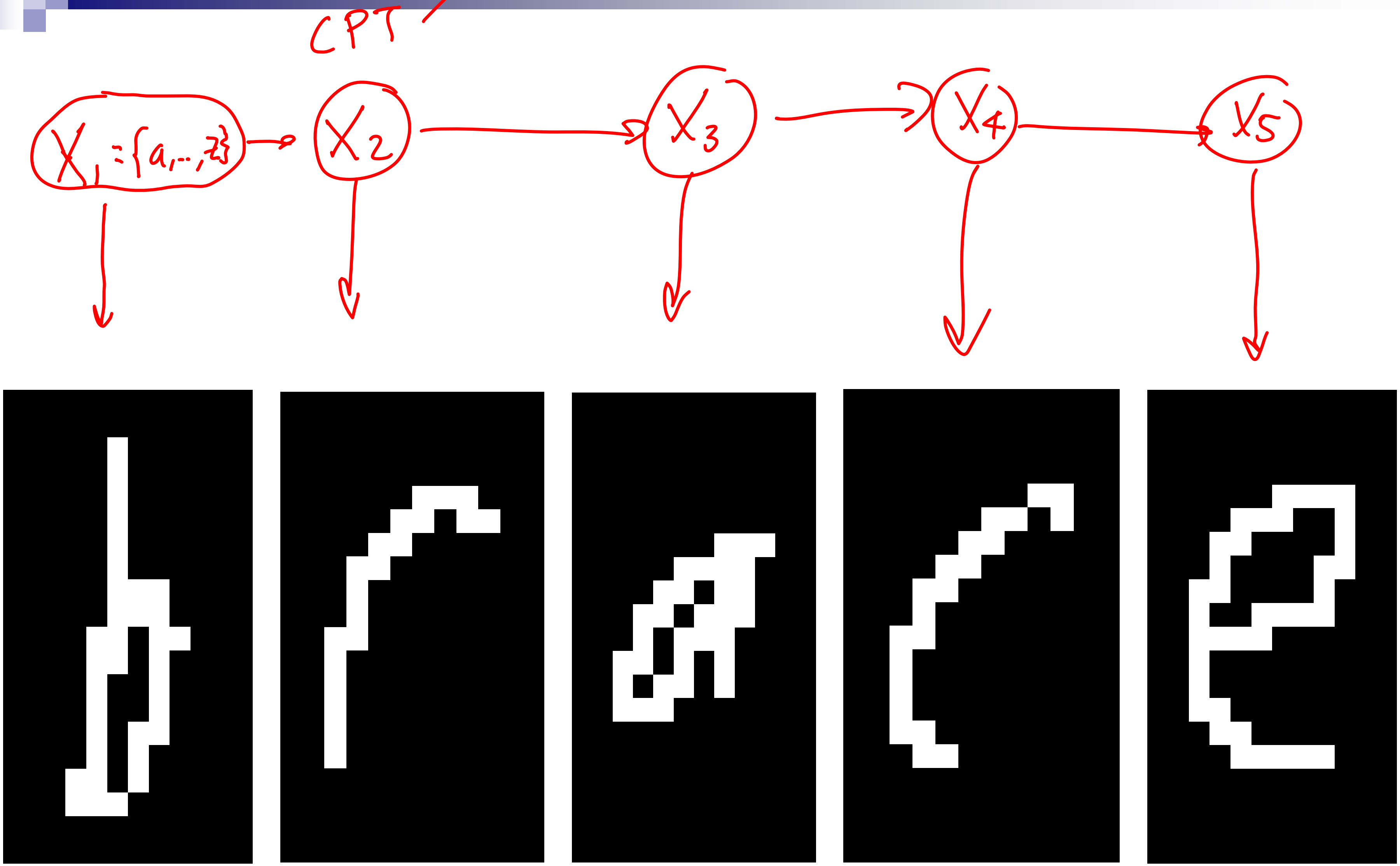
Character recognition, e.g., kernel SVMs





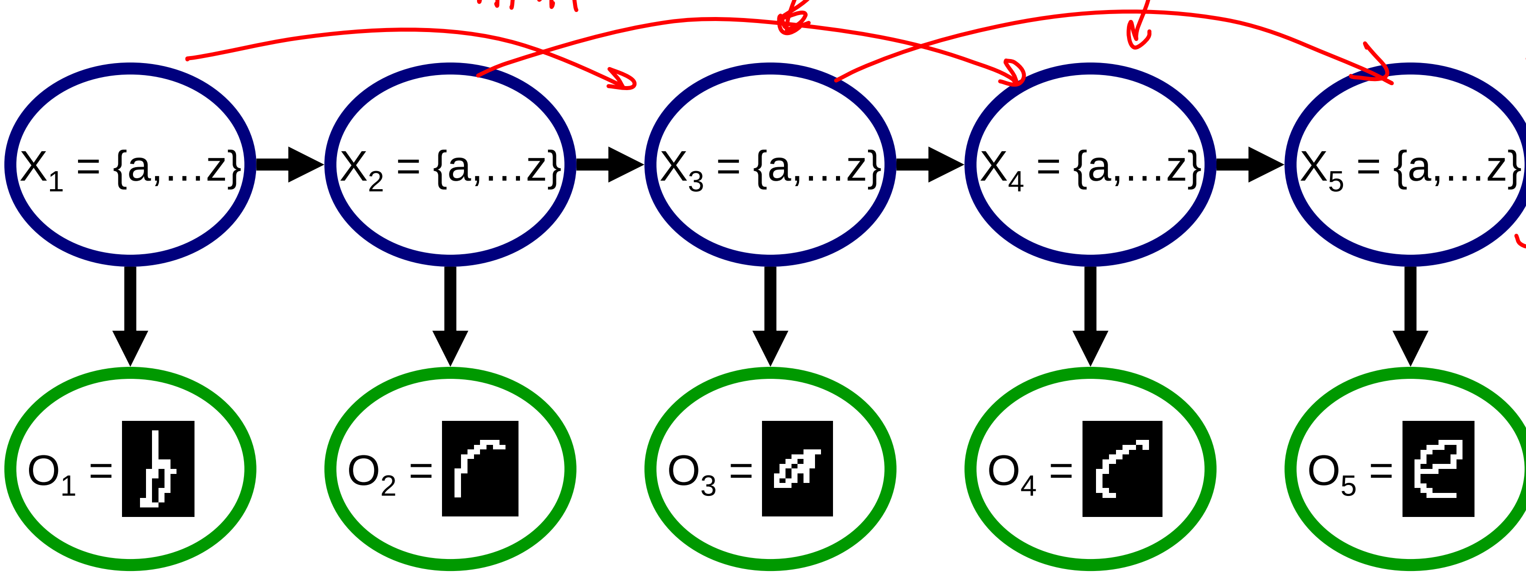
Example of a hidden Markov model (HMM)

$P(X_2 | X_1)$



Understanding the HMM Semantics

second order HMM



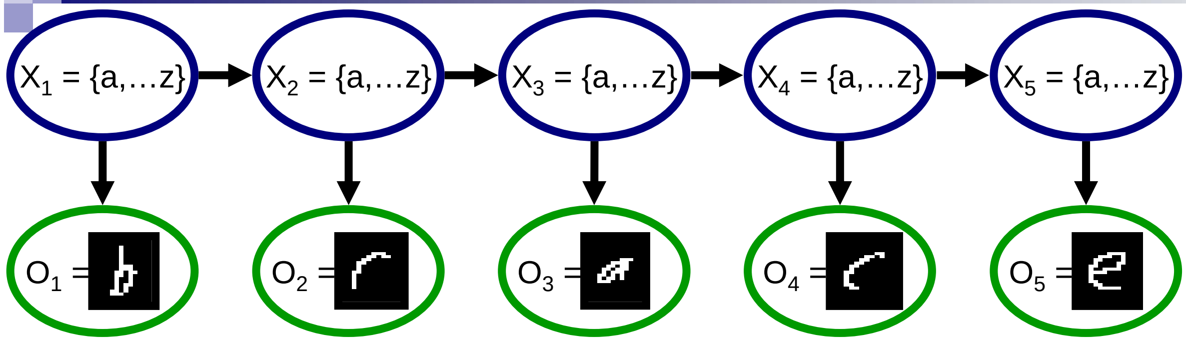
hidden vars

obs.

Independences ① ~~$I(O_i, \text{everyone} \mid X_i)$~~ $(O_i \perp \text{everyone} \mid X_i)$

② $(X_i \perp \text{all previous time vars} \mid X_{i-1})$ – Markov assumpt. HMMs

HMMs semantics: Details



Just 3 distributions:

$$P(X_1)$$ ← starting state dist.

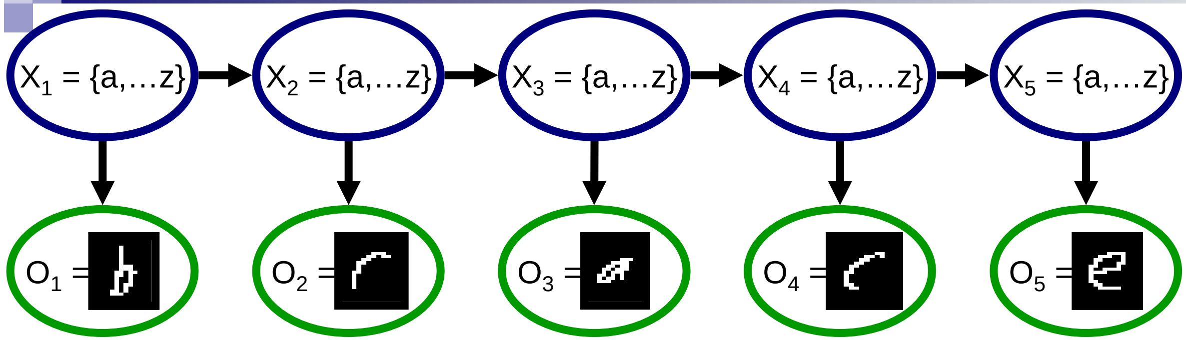
$$P(X_i \mid X_{i-1})$$ ← transition probabilities

usually, same $P(X_i \mid X_{i-1})$ $\forall i > 1$

$$P(O_i \mid X_i)$$ ← observation model

usually same for all i

HMMs semantics: Joint distribution

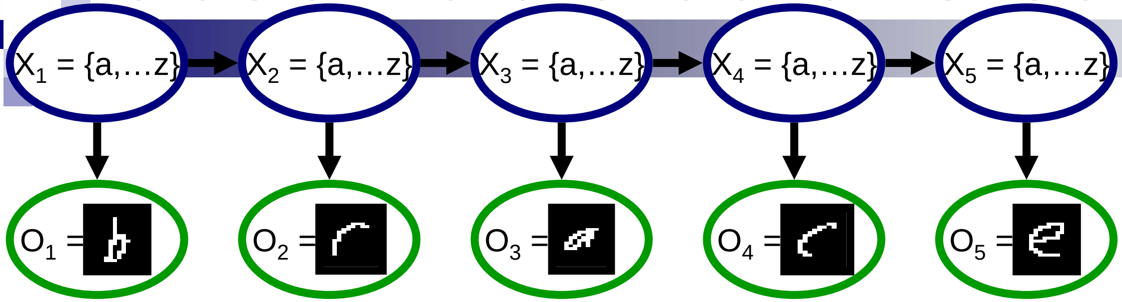


$$P(X_1)$$
$$P(X_i \mid X_{i-1})$$
$$P(O_i \mid X_i)$$

$P(X_1, X_2, X_3, X_4, X_5, O_1, O_2, O_3, O_4, O_5) =$
$P(X_1)\, P(O_1 \mid X_1)\, P(X_2 \mid X_1)\, P(O_2 \mid X_2) \cdots P(X_5 \mid X_4)\, P(O_5 \mid X_5)$

$$P(X_1, \ldots, X_n \mid o_1, \ldots, o_n) = P(X_{1:n} \mid o_{1:n})$$
$$\propto P(X_1) P(o_1 \mid X_1) \prod_{i=2}^{n} P(X_i \mid X_{i-1}) P(o_i \mid X_i)$$

Learning HMMs from fully observable data is easy



Data

$\langle x_1^{(j)}, o_1^{(j)}, x_2^{(j)}, o_2^{(j)}, \ldots, x_n^{(j)}, o_n^{(j)} \rangle$

Learn 3 distributions:

$$P(X_1 = x_1) = \frac{\text{Count}(X_1 = x_1)}{m}$$

use all i's in counts

each data "point", contributes n elements to count

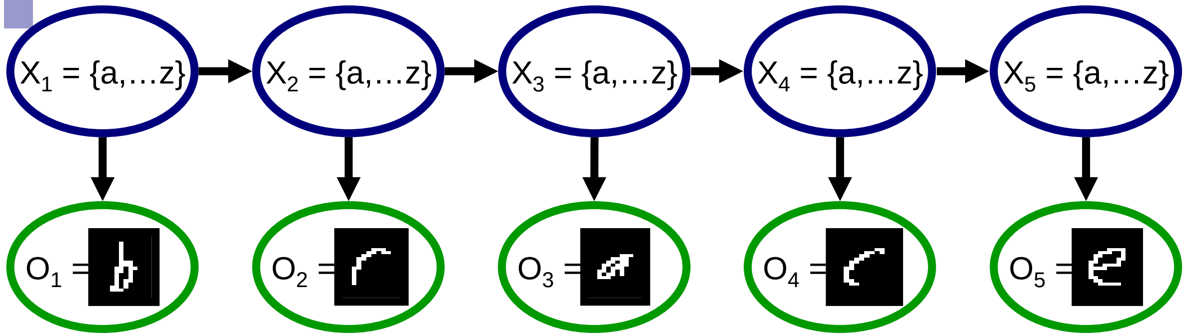
$$P(O_i = o_i \mid X_i = x_i) \propto \frac{\text{Count}(O_i = o_i, X_i = x_i)}{\text{Count}(X_i = x_i)}$$

each j contributes n−1

$$P(X_i = x_i \mid X_{i-1} = x_{i-1}) \propto \frac{\text{Count}(X_{i-1} = x_{i-1}, X_i = x_i)}{\text{Count}(X_{i-1} = x_{i-1})}$$

Parameter sharing / tieing

Possible inference tasks in an HMM



Marginal probability of a hidden variable:

$$P(X_3 \mid o_1 = \boxed{b}, o_2 = \boxed{r}, o_3 = \boxed{a}, o_4 = \boxed{c}, o_5 = \boxed{e})$$

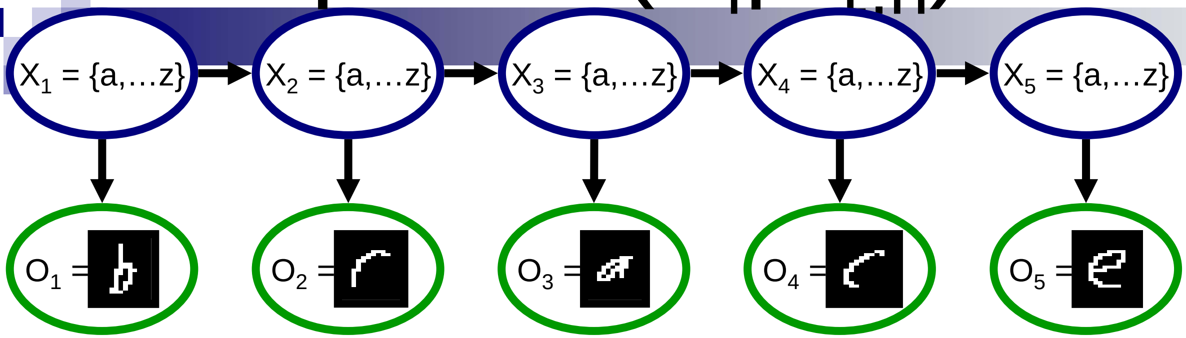
Viterbi decoding – most likely trajectory for hidden vars:

$$\max_{x_1,x_2,x_3,x_4,x_5} P(x_1, x_2, x_3, x_4, x_5 \mid o_1, o_2, o_3, o_4, o_5)$$

Using variable elimination to compute $P(X_i|o_{1:n})$

$A \rightarrow B \rightarrow C$

$A \rightarrow B \leftarrow C$



Compute:

$$P(X_i \mid o_{1:n})$$

$$\sum_{X_{1:i-1}} \sum_{X_{i+1:n}} P(X_1 \ldots X_n \mid o_{1:n})$$

Variable elimination order?

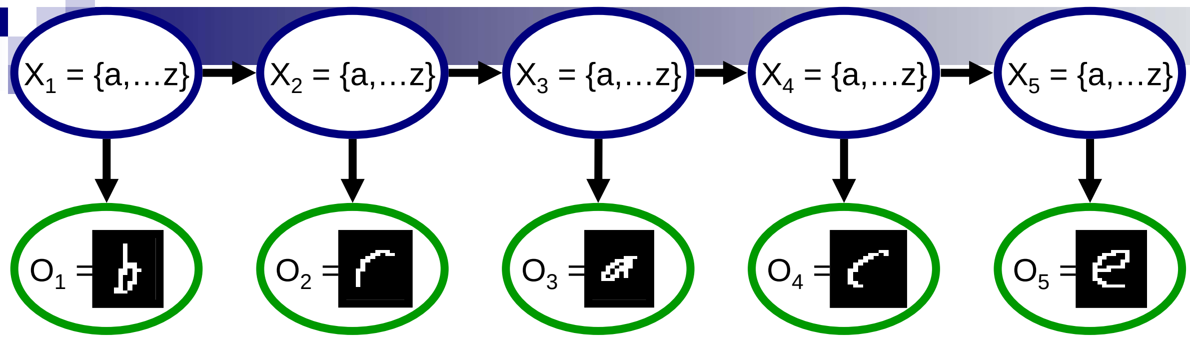
$n, n-1 \ldots, i+1, 1, \ldots, i-1$

Example: $\sum_{X_{1:i-1}} \sum_{X_{i+1:n}} P(X_{1:n} \mid o_{1:n}) \leftarrow P(X_3 \mid o_{1:5})$

$$\sum_{X_1} \sum_{X_2} \sum_{X_4} \sum_{X_5} P(X_1)\, P(o_1|X_1)\, P(X_2|X_1)\, P(o_2|X_2) \ldots$$

eliminate X_5: $\sum_{X_1} \sum_{X_2} \sum_{X_4} P(X_1)\, P(o_1|X_1)\, P(X|X_1) \ldots \underbrace{\sum_{X_5} P(X_5|X_4)\, P(o_5|X_5)}_{f_1(X_4)}$

What if I want to compute $P(X_i|o_{1:n})$ for each i?



Compute:

$$P(X_i \mid o_{1..n})$$

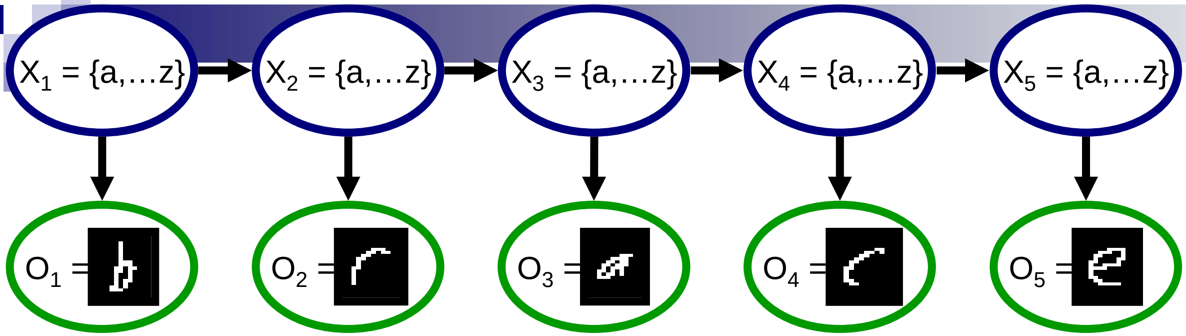
Variable elimination for each i?

Run VE n times

$\underbrace{\text{each time cost me } O(n)}_{\text{total} = O(n^2)}$

Variable elimination for each i, what's the complexity?

Reusing computation



Compute:

$$P(X_i \mid o_{1..n})$$

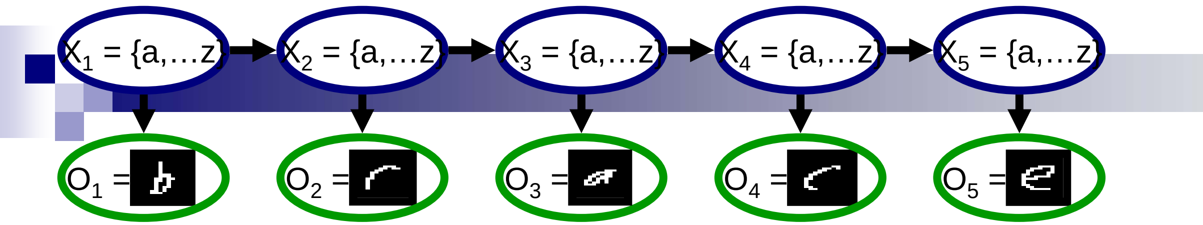
$$P(X_2 \mid o_{1:n}) = \sum_{x_1}\sum_{x_3}\sum_{x_4}\sum_{x_5} P(x_{1:5} \mid o_{1:5})$$

eliminate x_5, generate:

$$\underbrace{\sum_{x_5} P(x_5 \mid x_4)\, P(o_5 \mid x_5)}_{f_1(x_4)}$$

Same factor from $P(X_3 \mid o_{1:n})$

The forwards-backwards algorithm



$$P(X_i \mid o_{1..n})$$

Complexity $O(n)$

- Initialization: $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$
- For i = 2 to n
  - Generate a forwards factor by eliminating X_{i-1}

$\forall x_i \in X_i$

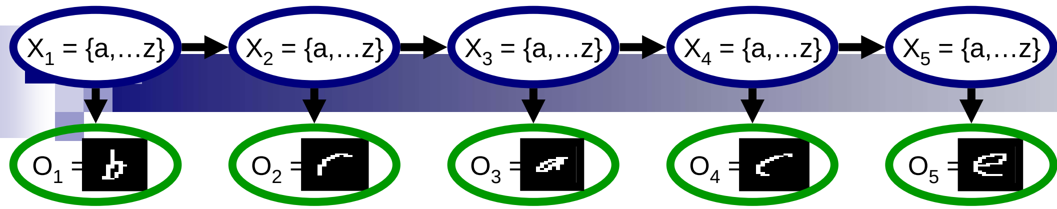
$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i) P(X_i \mid X_{i-1} = x_{i-1}) \alpha_{i-1}(x_{i-1})$$

- Initialization: $\beta_n(X_n) = 1$
- For i = n-1 to 1
  - Generate a backwards factor by eliminating X_{i+1}

$$\beta_i(X_i) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1}) P(x_{i+1} \mid X_i) \beta_{i+1}(x_{i+1})$$

- ∀ i, probability is: $\boxed{P(X_i \mid o_{1..n}) = \alpha_i(X_i)\beta_i(X_i)}$

Most likely explanation



Compute:

$$\max_{x_{1:n}} P(x_{1:n} \mid o_{1:n})$$

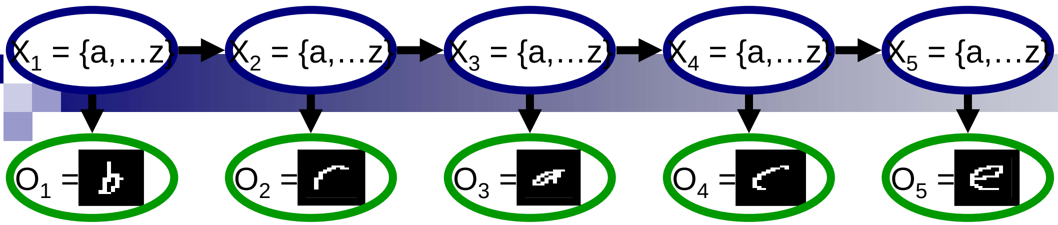
Variable elimination order?

1, 2, 3, ..., n

Example:

$$\max_{x_{2:5}} P(X_3|X_2)\,P(O_3|X_3)\,P(X_4|X_3)\,P(O_4|X_4)\dots \underbrace{\max_{x_1} P(x_1)\,P(o_1|x_1)\,P(x_2|x_1)}_{f_1(x_1)}$$

The Viterbi algorithm



- Initialization: $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$
- For i = 2 to n
  - Generate a forwards factor by eliminating X_{i-1}

$$\alpha_i(X_i) = \max_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$$

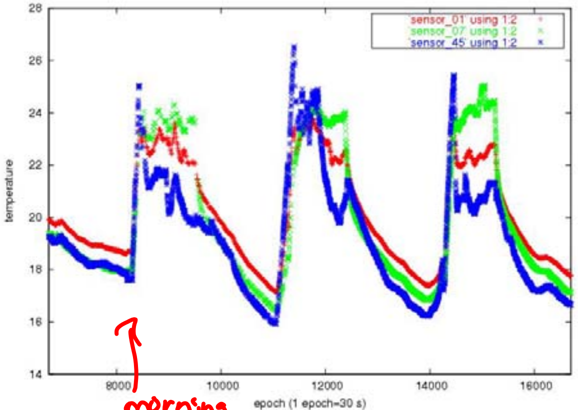
- Computing best explanation: $x_n^* = \operatorname{argmax}_{x_n} \alpha_n(x_n)$
- For i = n-1 to 1
  - Use argmax to get explanation:

$$x_i^* = \operatorname{argmax}_{x_i} P(x_{i+1}^* \mid x_i)\alpha_i(x_i)$$

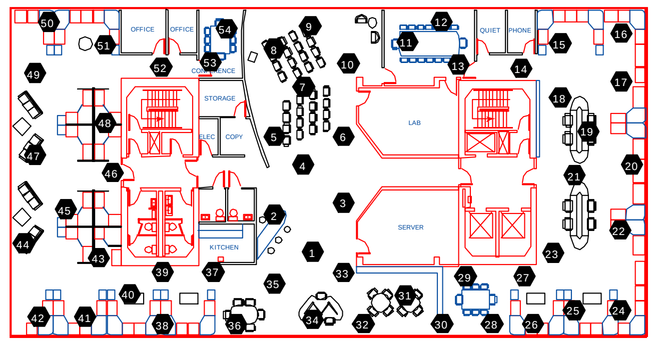
What about continuous variables?

- In general, very hard!
  - Must represent complex distributions
- A special case is very doable
  - When everything is Gaussian
  - Called a Kalman filter
  - One of the most used algorithms in the history of probabilities!

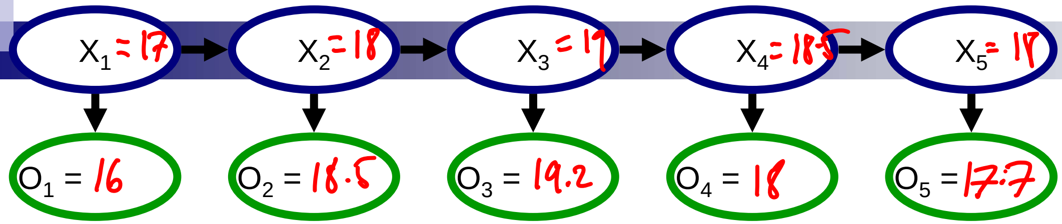
Time series data example: Temperatures from sensor network

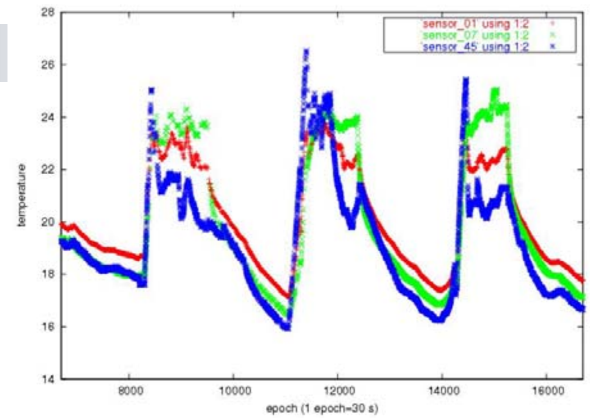


morning



Operations in Kalman filter

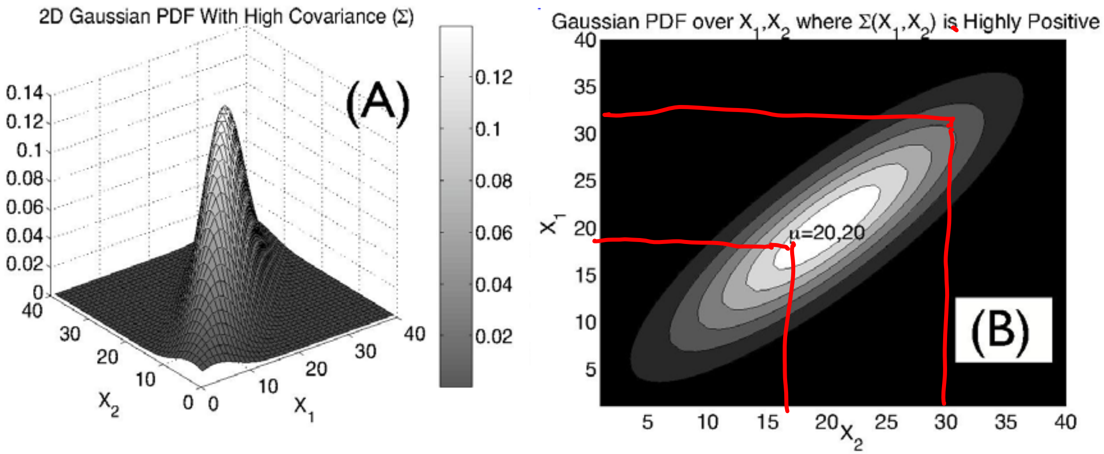




- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1}) p(o_t \mid X_t)$$
  - **Roll-up** (marginalize previous time step)

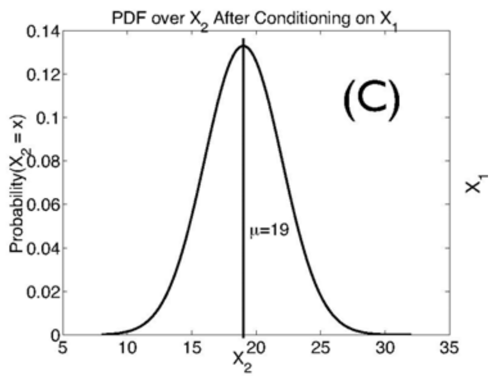
$$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} \underbrace{P(X_{t+1} \mid x_t) p(x_t \mid o_{1:t})}_{P(X_{t+1}, x_t \mid o_{1:t})} dx_t$$

Detour: Understanding Multivariate Gaussians



Observe attributes
Example: Observe $X_1=18$

$P(X_2|X_1=18)$



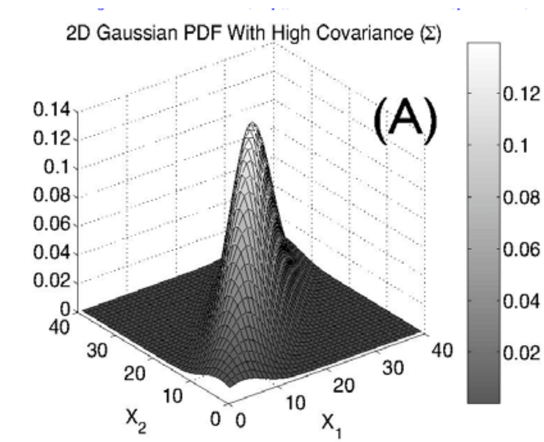
Characterizing a multivariate Gaussian

for 1d case $n=1$ $= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\underbrace{(x-\mu)(\sigma^2)^{-1}(x-\mu)}_{\frac{(x-\mu)^2}{\sigma^2}}\right\}$

$$p(X_1,\ldots,X_n) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

Mean vector:

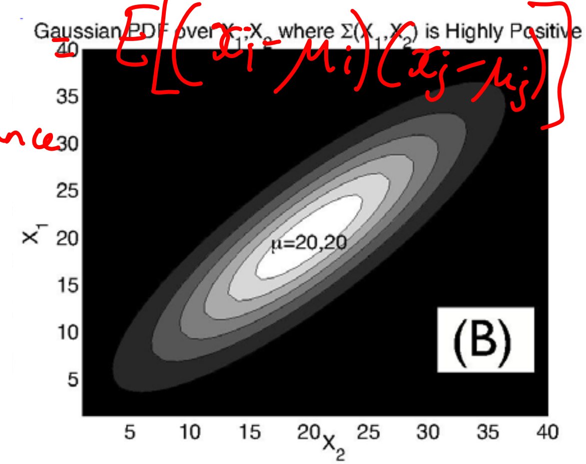
$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}$$



Covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & & \vdots \\ \vdots & \vdots & \ddots & \sigma_n^2 \end{pmatrix}$$

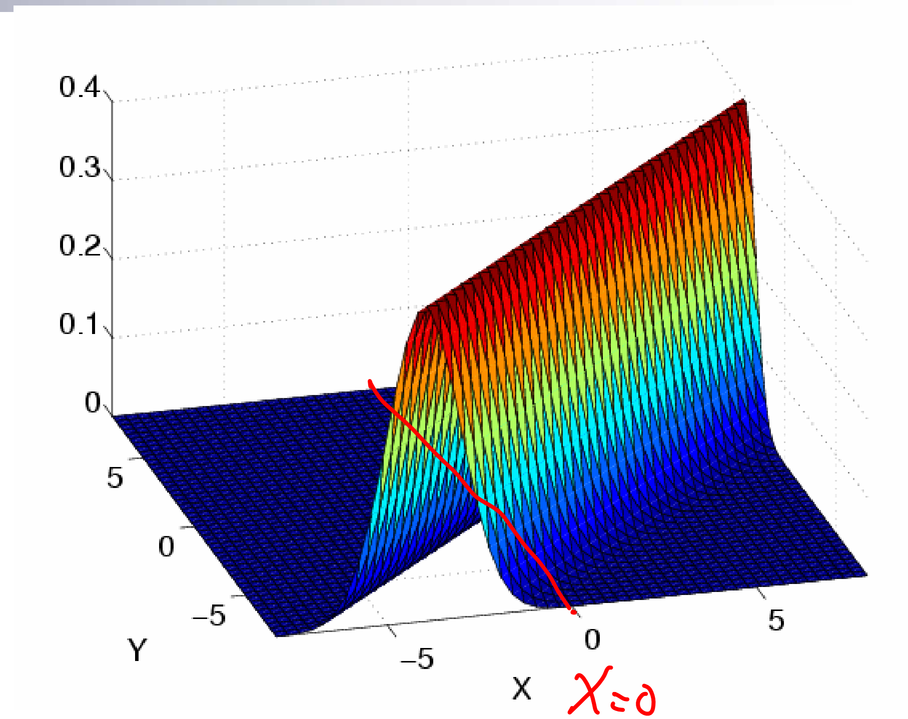
σ_{ij} Covariance $= E[(x_i-\mu_i)(x_j-\mu_j)]$



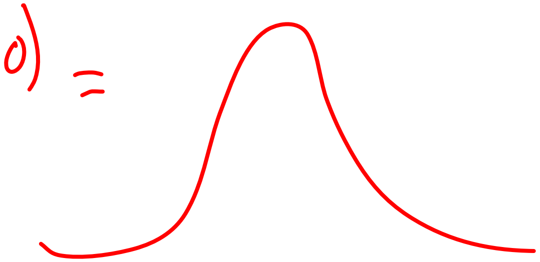
Conditional Gaussians

- Conditional probabilities
  - P(Y|X)

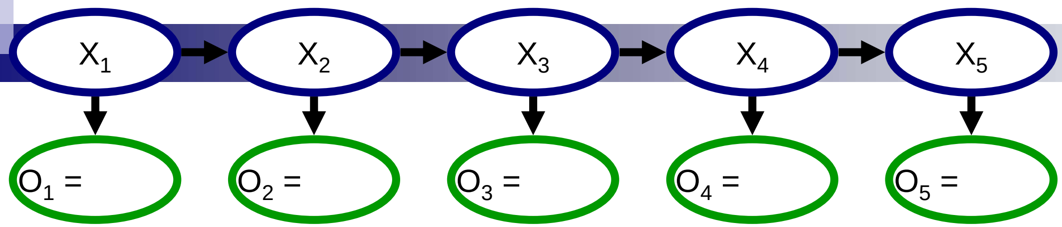
for each x, get Gaussian over Y

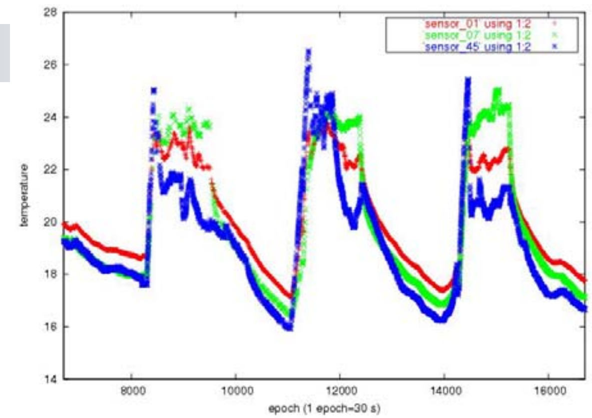


P(Y|X=0) =



Kalman filter with Gaussians





$P(X_1)$ ← Gaussian

$P(O_i \mid X_i)$ ← Conditional Gaussians

$P(X_i \mid X_{i-1})$

- Equivalent to a linear system

$$X_i = a + b X_{i-1} + \varepsilon$$

Gaussian noise

Detour2: Canonical form

$$\begin{aligned} p(X_1, \ldots, X_n) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right\} \\ &= K \exp\left\{\eta^T \mathbf{x} - \frac{1}{2}\mathbf{x}^T \Lambda^{\cancel{-1}} \mathbf{x}\right\} \end{aligned}$$

precision vector precision matrix

- Standard form and canonical forms are related:

$$\begin{aligned} \mu &= \Lambda^{-1}\eta \\ \Sigma &= \Lambda^{-1} \end{aligned}$$

- Conditioning is easy in canonical form
- Marginalization easy in standard form

Conditioning in canonical form

$$p(X_t \mid o_{1:t}) \propto \underbrace{p(X_t \mid o_{1:t-1})}_{\text{Gaussian}}\underbrace{p(o_t \mid X_t)}_{\text{Gaussian}}$$

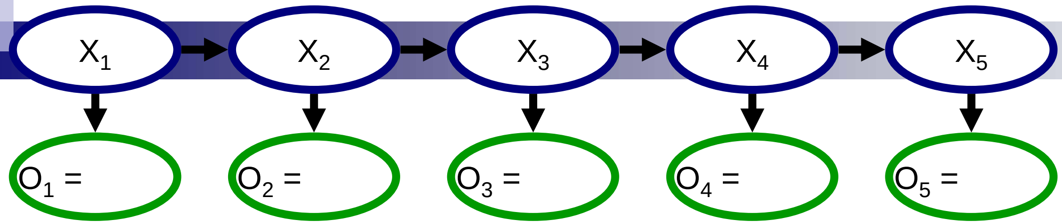
- First multiply: $p(A, B) = p(A)p(B \mid A)$

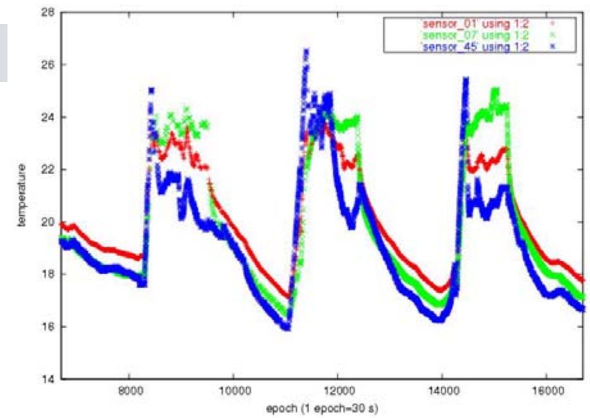
$$p(A): \quad \eta_1, \ \Lambda_1$$
$$p(B \mid A): \quad \eta_2, \ \Lambda_2$$
$$p(A, B): \quad \eta_3 = \eta_1 + \eta_2, \ \Lambda_3 = \Lambda_1 + \Lambda_2$$

- Then, condition on value B = y $\quad p(A \mid B = y)$

$$\eta_{A|B=y} = \eta_A - \Lambda_{AB}.y$$
$$\Lambda_{AA|B=y} = \Lambda_{AA}$$

Operations in Kalman filter





- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1})p(o_t \mid X_t)$$
  - **Roll-up** (marginalize previous time step)

$$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} P(X_{t+1} \mid x_t)p(x_t \mid o_{1:t})dx_t$$

Roll-up in canonical form

$$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} P(X_{t+1} \mid x_t) p(x_t \mid o_{1:t}) dx_t$$

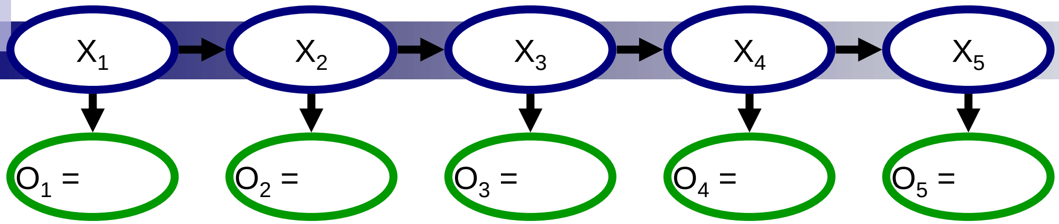
- First multiply: $p(A, B) = p(A) p(B \mid A)$

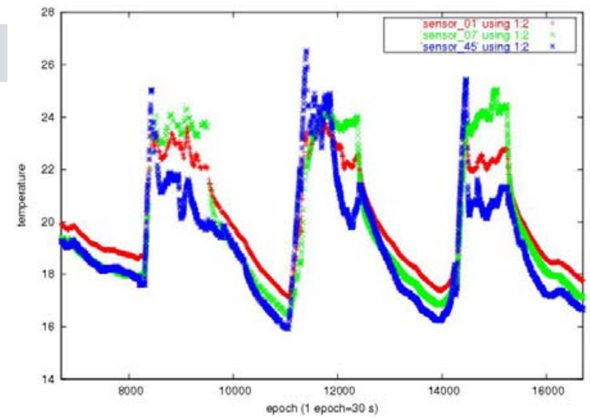
  add η's, Λ's

- Then, marginalize X_t: $p(A) = \int_B P(A, b) db$

$$\begin{aligned} \eta_A^m &= \eta_A - \Lambda_{AB} \Lambda_{BB}^{-1} \eta_B \\ \Lambda_{AA}^m &= \Lambda_{AA} - \Lambda_{AB} \Lambda_{BB}^{-1} \Lambda_{BA} \end{aligned}$$

Operations in Kalman filter





- Compute $p(X_t \mid O_{1:t} = o_{1:t})$
- Start with $p(X_0)$
- At each time step *t*:
  - **Condition** on observation
    $$p(X_t \mid o_{1:t}) \propto p(X_t \mid o_{1:t-1})p(o_t \mid X_t)$$
  - **Roll-up** (marginalize previous time step)

$$p(X_{t+1} \mid o_{1:t}) = \int_{X_t} P(X_{t+1} \mid x_t)p(x_t \mid o_{1:t})dx_t$$

Learning a Kalman filter

- Must learn: $P(X_1)$

$$P(O_i \mid X_i) = \frac{P(O_i, X_i)}{P(O_i)}$$

$$P(X_i \mid X_{i-1}) = \frac{P(X_i, X_{i-1})}{P(X_{i-1})}$$

- Learn joint, and use division rule:

$$p(A): \quad \eta_1, \ \Lambda_1$$

$$p(A,B): \quad \eta_2, \ \Lambda_2$$

$$p(B \mid A) = \frac{p(A,B)}{p(A)}: \quad \eta_3 = \eta_2 - \eta_1, \ \Lambda_3 = \Lambda_2 - \Lambda_1$$

Maximum likelihood learning of a multivariate Gaussian

$$\mu = \Lambda^{-1}\eta$$
$$\Sigma = \Lambda^{-1}$$

- Data: $< x_1^{(j)}, \ldots, x_n^{(j)} >$

- Means are just empirical means:

$$\widehat{\mu}_i = \frac{\sum_{j=1}^{m} x_i^{(j)}}{m}$$

- Empirical covariances:

$$\widehat{\Sigma}_{ik} = \frac{\sum_{j=1}^{m}(x_i^{(j)} - \widehat{\mu}_i)(x_k^{(j)} - \widehat{\mu}_k)}{m}$$

What you need to know

- Hidden Markov models (HMMs)
  - Very useful, very powerful!
  - Speech, OCR,…
  - Parameter sharing, only learn 3 distributions
  - Trick reduces inference from $O(n^2)$ to $O(n)$
  - Special case of BN
- Kalman filter
  - Continuous vars version of HMMs
  - Assumes Gaussian distributions
  - Equivalent to linear system
  - Simple matrix operations for computations