Probabilistic Graphical Models

Lecture 12 – Dynamical Models

CS/CNS/EE 155
Andreas Krause

Announcements

- Homework 3 out tonight
  - Start early!!
- Project milestones due today
  - Please email to TAs

Parameter learning for log-linear models

- Feature functions $\phi_i(C_i)$ defined over cliques
- Log linear model over undirected graph G
  - Feature functions $\phi_1(C_1),\ldots,\phi_k(C_k)$
  - Domains C_i can overlap
- Joint distribution

$$P(X_1, \ldots, X_n) = \frac{1}{Z} \exp\Big(\sum_i w_i^T \phi_i(C_i)\Big)$$

- How do we get weights w_i?

Log-linear conditional random field

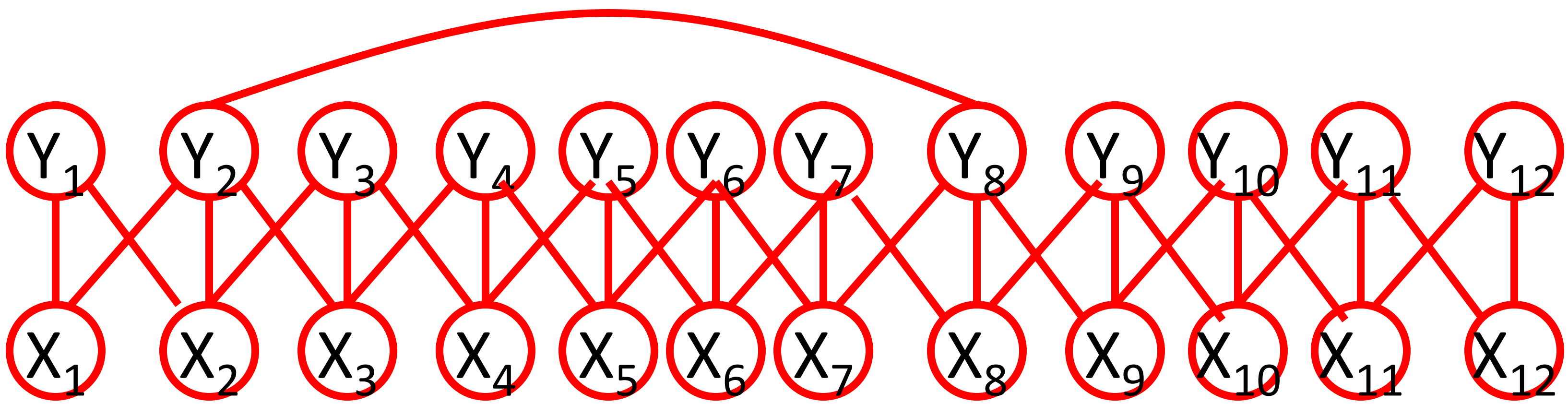
- Define log-linear model over outputs Y
  - No assumptions about inputs X
- Feature functions $\phi_i(C_i,x)$ defined over cliques and inputs

  $C_i \subseteq Y$

- Joint distribution

$$P(Y_1, \ldots, Y_n \mid x) = \frac{1}{Z(x,w)} \exp\Big(\sum_i w_i^T \phi_i(C_i, x)\Big)$$

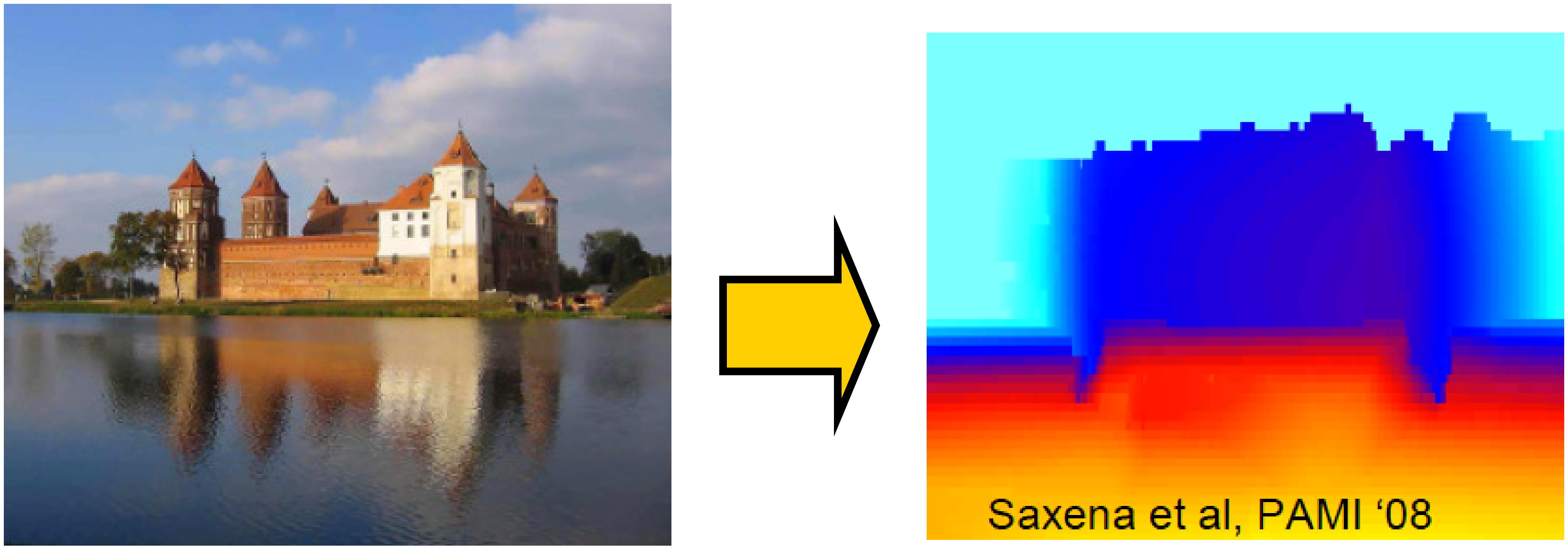
Example: CRFs in NLP



Mrs. Greene spoke today in New York. Green chairs the finance committee

- Classify into Person, Location or Other

Example: CRFs in vision



Gradient of conditional log-likelihood

- Partial derivative

$$\frac{\partial \log P(\mathcal{D}_Y \mid w, \mathcal{D}_X)}{\partial w_i} = \sum_j \left[\phi_i(\mathbf{c}_i^{(j)}, x^{(j)}) + \underbrace{\sum_{\mathbf{c}_i} P(\mathbf{c}_i \mid w, x^{(j)}) \phi_i(\mathbf{c}_i, x^{(j)})}_{\text{Req. Inference}} \right]$$

- Requires one inference per feature and per data point

  Can be very expensive

  Can do "pseudo" likelihood est. (approximate)

- Can optimize using conjugate gradient

Exponential Family Distributions

- Distributions for log-linear models

$$P(X_1, \ldots, X_n) = \frac{1}{Z} \exp\Big(\sum_i w_i^T \phi_i(C_i)\Big)$$

- More generally: **Exponential family distributions**

$$P(x) = h(x) \exp\big(w^T \phi(x) - A(w)\big)$$

  - h(x): Base measure ← Often constant
  - w: natural parameters
  - ϕ(x): Sufficient statistics
  - A(w): log-partition function ← $A(w) = \log Z(w)$
- Hereby x can be continuous (defined over any set)

Examples

- Exp. Family:

$$P(x) = h(x) \exp\big(w^T \phi(x) - A(w)\big)$$

h(x): Base measure
w: natural parameters
ϕ(x): Sufficient statistics
A(w): log-partition function

- Gaussian distribution

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$-\frac{x^2}{2\sigma^2} + \frac{x\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2}$$

$$h(x) = \frac{1}{\sqrt{2\pi}}$$

$$\phi(x) = \left[-\frac{x^2}{2}, x\right], \quad A(w) = \frac{\mu^2}{2\sigma^2} - \log \sigma$$

$$w = \left[\frac{1}{\sigma^2}, \frac{\mu}{\sigma^2}\right],$$

- Other examples: Multinomial, Poisson, Exponential, Gamma, Weibull, chi-square, Dirichlet, Geometric, …

Moments and gradients

$$P(x) = h(x)\exp\big(w^T\phi(x) - A(w)\big)$$

- Correspondence between moments and log-partition function (just like in log-linear models)

$$\frac{\partial A(w)}{\partial w_i} = \int p(x \mid w)\phi_i(x)dx = \mathbb{E}[\phi_i \mid w]$$

$$\frac{\partial^2 A(w)}{\partial w_i \partial w_j} = Cov(\phi_i, \phi_j \mid w)$$

- Can compute moments from derivatives, and derivatives from moments!
- MLE ⇔ moment matching

Conjugate priors in Exponential Family

$$P(x \mid w) = h(x) \exp\big(w^T \phi(x) - A(w)\big)$$

Any exponential family likelihood has a conjugate prior

$$P(w|\alpha,\beta) = \exp\big(\underline{\alpha}^T w - \beta A(w) - \underline{B(\alpha,\beta)}\big)$$

$$P(x|w)\,P(w|\alpha,\beta) \propto \exp\big(w^T(\phi(x)+\alpha) - (\beta+1)\,A(w)\big)$$

Exponential family graphical models

- So far, only defined graphical models over discrete variables.
- Can define GMs over continuous distributions!
- For exponential family distributions:

$$p(X_1,\ldots,X_n) = \prod_i h_i(C_i)\exp\Big(\sum_i w_i^T\phi_i(C_i) - A(w)\Big)$$

$$\exp A(w) = \int\int\ldots\int \prod_i h_i(C_i)\exp\Big(\sum_i w_i^T\phi_i(C_i) - A(w)\Big)dx_1\ldots dx_n$$

- Can do much of what we discussed (VE, JT, parameter learning, etc.) for such exponential family models
- Important example: **Gaussian Networks**

Multivariate Gaussian distribution

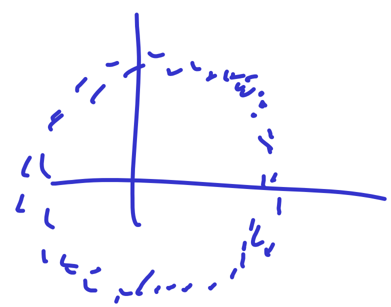
$$\mathcal{N}(x;\Sigma,\mu) = \frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \vdots & & & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix}$$

$\Sigma^T = \Sigma$

pos. def: $x^T \Sigma x \geq 0 \ \forall x$

- Joint distribution over n random variables $P(X_1, \dots X_n)$
- $\sigma_{jk} = E[\,(X_j - \mu_j)\,(X_k - \mu_k)\,]$ $= Cov(X_j, X_k)$
- X_j and X_k independent $\Leftrightarrow$ $\sigma_{jk}=0$



Marginalization

- Suppose $(X_1,\dots,X_n) \sim N(\mu, \Sigma)$
- What is $P(X_1)$??

$$\Sigma = \begin{pmatrix} \boxed{\begin{matrix} \sigma_1^2 & \sigma_{12} \\ \vdots & \end{matrix}} & \dots & \sigma_{1n} \\ & & \vdots \\ \sigma_{n1} \quad \sigma_{n2} & \dots & \sigma_n^2 \end{pmatrix} \quad \mu = \begin{pmatrix} \boxed{\begin{matrix} \mu_1 \\ \mu_2 \end{matrix}} \\ \vdots \\ \mu_n \end{pmatrix}$$

Σ_{AA} μ_A

$P(X_1) = \int\int\int\int\int \dots \int p(x_1 \dots x_n) dx_2 \dots dx_n$

$P(x_1) = \mathcal{N}(x_1; \mu_1, \sigma_1^2)$

- More generally: Let $A=\{i_1,\dots,i_k\} \subseteq \{1,\dots,N\}$
- Write $X_A = (X_{i_1},\dots,X_{i_k})$

- $X_A \sim N(\mu_A, \Sigma_{AA})$

$$\Sigma_{AA} = \begin{pmatrix} \sigma_{i_1}^2 & \sigma_{i_1 i_2} & \dots & \sigma_{i_1 i_k} \\ \vdots & & & \vdots \\ \sigma_{i_k i_1} & \sigma_{i_k i_2} & \dots & \sigma_{i_k}^2 \end{pmatrix} \qquad \mu_A = \begin{pmatrix} \mu_{i_1} \\ \mu_{i_2} \\ \vdots \\ \mu_{i_k} \end{pmatrix}$$

Conditioning

- Suppose $(X_1,\ldots,X_n) \sim N(\mu, \Sigma)$
- Decompose as (X_A, X_B)
- What is $P(X_A \mid X_B)$??

$$\Sigma = \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix}$$

- $P(X_A = x_A \mid X_B = x_B) = N(x_A; \mu_{A|B}, \Sigma_{A|B})$ where

$$\boxed{\begin{aligned} \mu_{A|B=x_B} &= \underline{\mu_A} + \underbrace{\Sigma_{AB}\Sigma_{BB}^{-1}}_{W}(x_B - \mu_B) \\ \Sigma_{A|B} &= \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA} \end{aligned}}$$

Does not depend on x_B

- Computable using linear algebra!

Conditional linear Gaussians

$$\mu_{A|B} = \mu_A + \underbrace{\Sigma_{AB}\Sigma_{BB}^{-1}}_{W}(x_B - \mu_B)$$

$$\Sigma_{A|B} = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$$

$$x_{A|B} = \mu_A + W x_B - W \mu_B + \varepsilon, \quad \underline{\varepsilon \sim \mathcal{N}(0, \Sigma_{A|B})}$$

$$= \underline{W x_B + b + \varepsilon} \leftarrow \text{noise}$$

$$W = \Sigma_{AB}\Sigma_{BB}^{-1}$$

$$b = \mu_A - W\mu_B$$

Canonical Representation

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

$$\propto \exp(\eta^T x - \frac{1}{2}x^T \Lambda x) = \exp\left(\sum_i \eta_i x_i - \frac{1}{2}\sum_{i,j} x_i x_j \Lambda_{ij}\right)$$

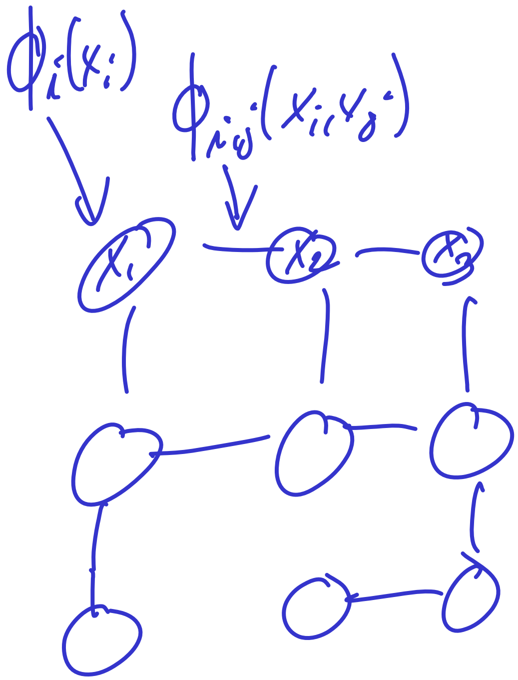
- Multivariate Gaussians in exponential family!
- Standard vs canonical form:

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

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

Gaussian Networks

$$p(X_1, \ldots, X_n) \propto \exp\Big(-\frac{1}{2}\sum_{i,j}\lambda_{i,j}x_i x_j + \sum_i \eta_i x_i\Big)$$



$$= \exp\Big(\sum_{i,j}\lambda_{ij}\phi_{ij}(x_i, x_j) + \sum_i \eta_i \phi_i(x_i)\Big)$$

$$\phi_{ij}(x_i, x_j) = -\frac{1}{2}x_i x_j \qquad \phi_i(x_i) = x_i$$

No edge between X_i, X_j

$$\Longleftrightarrow$$

$$\lambda_{ij} = 0$$

Zeros in precision matrix Λ indicate missing edges in log-linear model!

Inference in Gaussian Networks

- Can compute marginal distributions in $O(n^3)$!
- For large numbers n of variables, still intractable
- If Gaussian Network has low treewidth, can use variable elimination / JT inference!

- Need to be able to multiply and marginalize factors!

$$g = \int_{x_i} \prod_j f_j$$

Multiplying factors in Gaussians

$|A| = k, \; |B| = m$

$\Lambda_1 \in \mathbb{R}^{k \times k}$

$$P(X_A) = \mathcal{N}(x_A; \Lambda_1, \eta_1)$$

$$P(X_B \mid X_A) \propto \mathcal{N}(x_B, x_A; \Lambda_2, \eta_2)$$

CLG representation

$\Lambda_2 \in \mathbb{R}^{m \times m}$

$$P(X_A, X_B) = P(X_A)\, P(X_B \mid X_A)$$

$$\propto \exp(x_A^T \Lambda_1 x_A + \eta_1^T x_A)\, \exp\left((x_A, x_B)^T \Lambda_2 (x_A, x_B) + \eta_2^T (x_A, x_B)\right)$$

$$= \mathcal{N}(x_A, x_B; \Lambda, \eta)$$

$$\Lambda = \Lambda_1 + \Lambda_2 = \begin{pmatrix} \boxed{\Lambda_1} & \\ & 0 \end{pmatrix} + \begin{pmatrix} \Lambda_2 \end{pmatrix}$$

$$\eta = \eta_1 + \eta_2$$

Conditioning in canonical form

- Joint distribution $(X_A, X_B) \sim N(\eta_{AB}, \Lambda_{AB})$

$$\begin{bmatrix} \Lambda_{AA} & \Lambda_{AB} \\ \Lambda_{BA} & \Lambda_{BB} \end{bmatrix}$$

- Conditioning: $P(X_A \mid X_B = x_B) = N(x_A; \eta_{A|B=x_B}, \Lambda_{A|B=x_B})$

$$\eta_{A|B=x_B} = \eta_A - \Lambda_{AB} x_B$$

$$\Lambda_{A|B=x_B} = \Lambda_{AA}$$

Marginalizing in canonical form

- Recall conversion formulas
  - $\mu = \Lambda^{-1}\,\eta$
  - $\Sigma = \Lambda^{-1}$
- Marginal distribution

$$\eta_A^m = \eta_A - \Lambda_{AB}\Lambda_{BB}^{-1}\eta_B$$

$$\Lambda_{AA}^m = \Lambda_{AA} - \Lambda_{AB}\Lambda_{BB}^{-1}\Lambda_{BA}$$

Standard vs. canonical form

	Standard form	Canonical form
Marginalization	$\mu_A^m = \mu_A$ $\Sigma_{AA}^m = \Sigma_{AA}$	$\eta_A^m = \eta_A - \Lambda_{AB}\Lambda_{BB}^{-1}\eta_B$ $\Lambda_{AA}^m = \Lambda_{AA} - \Lambda_{AB}\Lambda_{BB}^{-1}\Lambda_{BA}$
Conditioning	$\mu_{A\|B=x_B} = \mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B)$ $\Sigma_{A\|B=x_B} = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$	$\eta_{A\|B=x_B} = \eta_A - \Lambda_{AB}x_B$ $\Lambda_{A\|B=x_B} = \Lambda_{AA}$

- In standard form, marginalization is easy
- In canonical form, conditioning is easy!

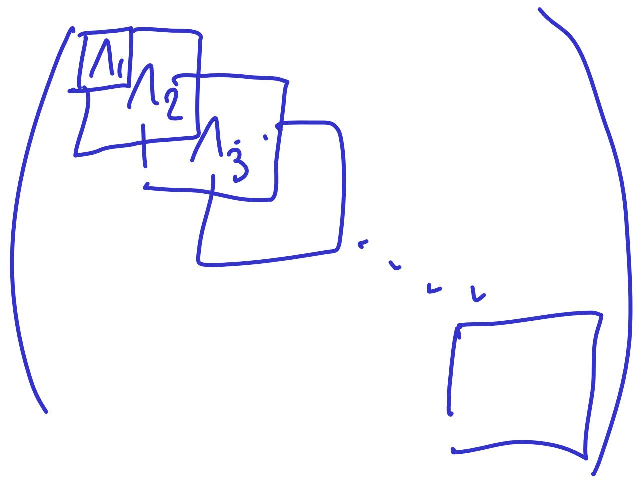
Variable elimination

$$P(X_1, \ldots, X_n) = \underbrace{P(X_1)}_{\Lambda_1} \underbrace{P(X_2 \mid X_1)}_{\Lambda_2} \cdots \underbrace{P(X_n \mid X_{n-1})}_{\Lambda_n}$$

$$X_1 \to X_2 \to X_3 \to X_4$$

$$= \mathcal{N}(\,\cdot\,; \Lambda, \eta)$$

$\Lambda =$



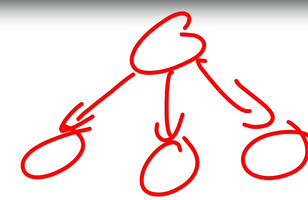
block diagonal

- In Gaussian Markov Networks, Variable elimination = Gaussian elimination (fast for low bandwidth = low treewidth matrices)

Dynamical models

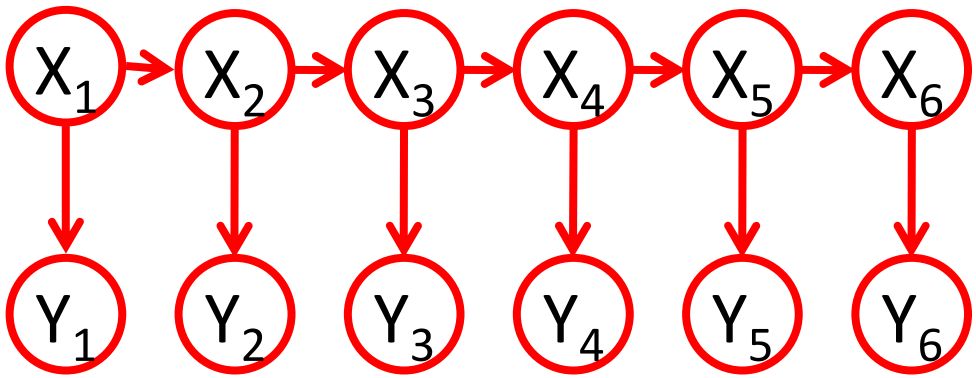
HMMs / Kalman Filters

- Most famous Graphical models:
  - Naïve Bayes model
  - Hidden Markov model
  - Kalman Filter



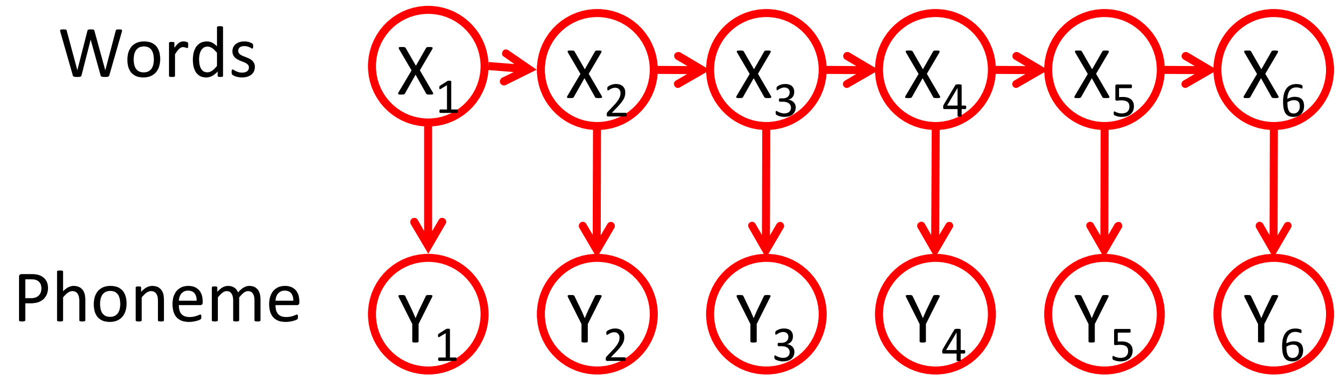
- Hidden Markov models
  - Speech recognition
  - Sequence analysis in comp. bio
- Kalman Filters control
  - Cruise control in cars
  - GPS navigation devices
  - Tracking missiles..
- Very simple models but very powerful!!

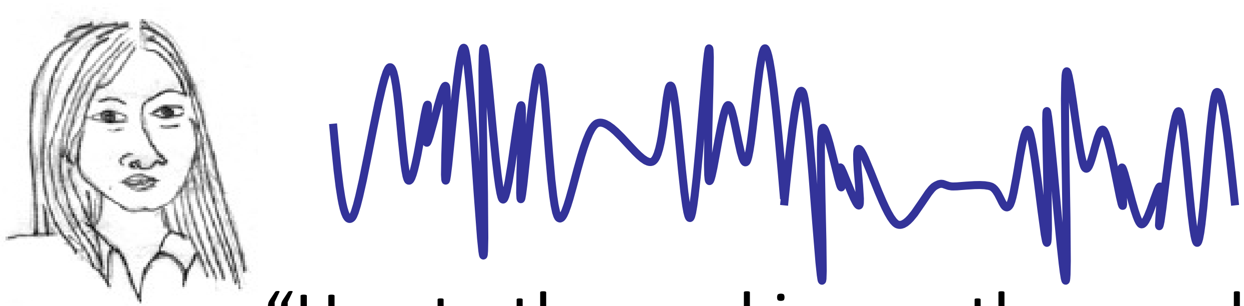
HMMs / Kalman Filters



- $X_1,\ldots,X_T$: Unobserved (hidden) variables
- $Y_1,\ldots,Y_T$: Observations
- HMMs: X_i Multinomial, Y_i arbitrary
- Kalman Filters: X_i, Y_i Gaussian distributions
  - Non-linear KF: X_i Gaussian, Y_i arbitrary

HMMs for speech recognition

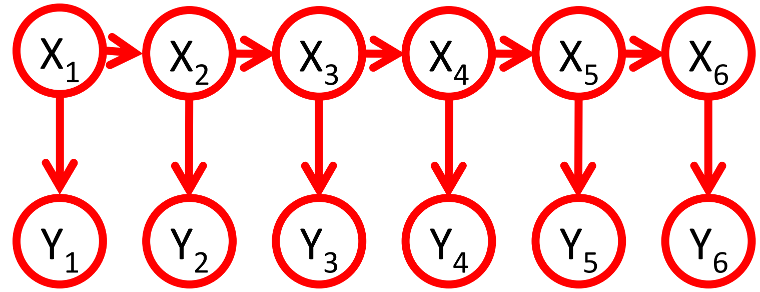




"He ate the cookies on the couch"

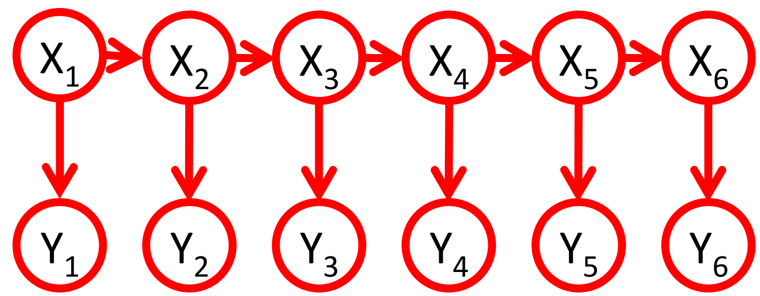
- Infer spoken words from audio signals

Hidden Markov Models



- Inference:
  - In principle, can use VE, JT etc.
  - New variables X_t, Y_t at each time step ➔ need to rerun
- Bayesian Filtering:
  - Suppose we already have computed $P(X_t \mid y_{1,\ldots,t})$
  - Want to efficiently compute $P(X_{t+1} \mid y_{1,\ldots,t+1})$

Bayesian filtering



- Start with $P(X_1)$
- At time t
  - Assume we have $P(X_t \mid y_{1\ldots t-1})$
  - Condition: $P(X_t \mid y_{1\ldots t})$

$$P(X_t \mid y_{1\ldots t}) \propto P(X_t \mid y_{1..t-1}) \underbrace{P(Y_t \mid X_t, y_{1..t-1})}_{\text{cond. ind. } P(Y_t \mid X_t)}$$

  - Prediction: $P(X_{t+1}, X_t \mid y_{1\ldots t})$

$$P(X_{t+1}, X_t \mid y_{1..t}) = P(X_t \mid y_{1..t}) \cdot \underbrace{P(X_{t+1} \mid X_t, y_{1..t})}_{= P(X_{t+1} \mid X_t)}$$

"Roll up"

  - Marginalization: $P(X_{t+1} \mid y_{1\ldots t})$

$$P(X_{t+1} \mid y_{1..t}) = \sum_{x_t} P(X_{t+1}, x_t \mid y_{1..t})$$

Parameter learning in HMMs

- Assume we have labels for hidden variables
- Assume stationarity
  - $P(X_{t+1} \mid X_t)$ is same over all time steps
  - $P(Y_t \mid X_t)$ is same over all time steps
  - Violates parameter independence (➔ parameter "sharing")
- Example: compute parameters for $P(X_{t+1}=x \mid X_t=x')$

$$\log P(x_1, \ldots x_T, y_1, \ldots y_T \mid \theta) = \log P(X_1) \prod_{t=2}^{T} P(X_t \mid X_{t-1}, \theta_1) \prod_{t=1}^{T} P(y_t \mid X_t, \theta_2)$$

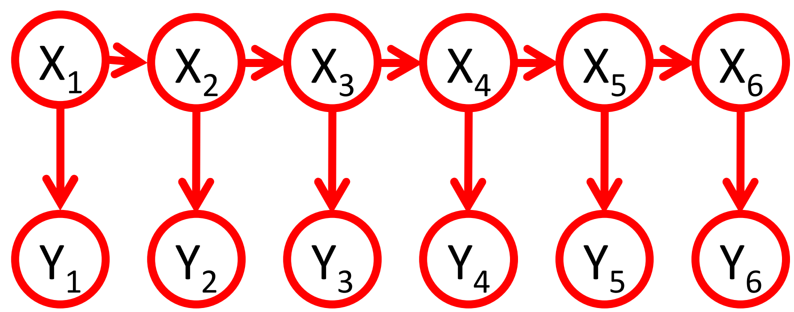
$$= \log P(X_1 \mid \theta_3) + \sum_t \log P(X_t \mid X_{t-1}, \theta_1) + \sum_t \log P(y_t \mid X_t, \theta_2)$$

$$\theta_{X_t = x \mid X_{t-1} = x'} = \frac{\mathrm{Count}(x', x)}{T-1}$$

- What if we don't have labels for hidden vars?
  - ➔ Use EM (later this course)

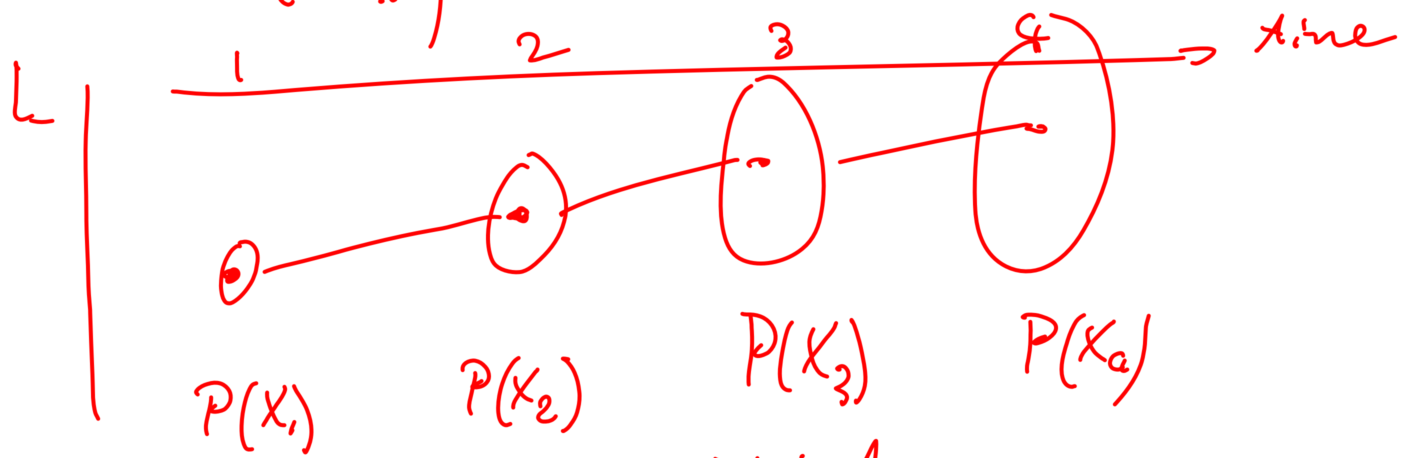
Kalman Filters (Gaussian HMMs)

- $X_1,\ldots,X_T$: Location of object being tracked
- $Y_1,\ldots,Y_T$: Observations
- $P(X_1)$: Prior belief about location at time 1
- $P(X_{t+1}|X_t)$: "Motion model"
  - How do I expect my target to move in the environment?
  - Represented as CLG: $X_{t+1} = A\,X_t + N(0, \Sigma_M)$
- $P(Y_t \mid X_t)$: "Sensor model"
  - What do I observe if target is at location X_t?
  - Represented as CLG: $Y_t = H\,X_t + N(0, \Sigma_O)$



Understanding Motion model

$$X_t = \begin{pmatrix} L_t \\ V_t \end{pmatrix} \qquad X_{t+1} = A X_t + \text{noise} \qquad A = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$$



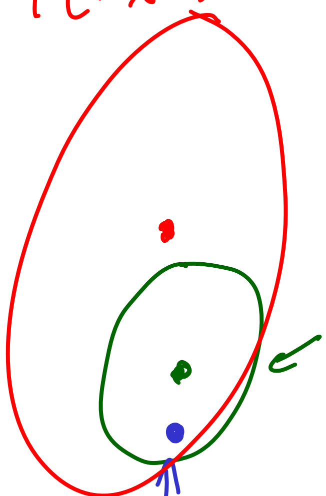
$$= \int P(X_2 | X_1)\, P(X_1)\, dx_1$$

Understanding sensor model

$$X_t = \begin{pmatrix} L_t \\ V_t \end{pmatrix} \qquad Y_t = \underbrace{(1, 0)}_{H} X_t + \text{noise}$$

only obs. location

$P(X_t | y_1 \ldots y_{t-1})$

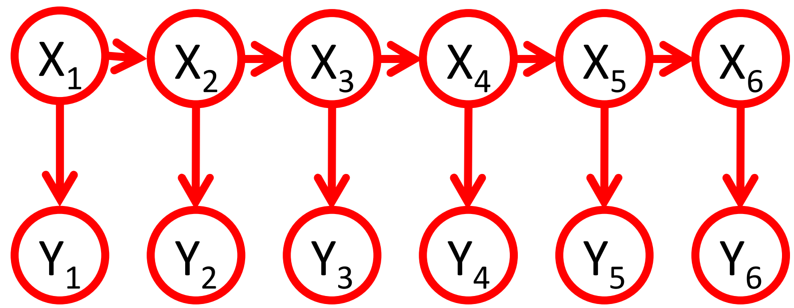


Posterior $P(X_t | y_1 \ldots y_t)$

observation Y_t

Bayesian Filtering for KFs

- Can use Gaussian elimination to perform inference in "unrolled" model



$\mathcal{N}(x_t ; \eta_t , \Lambda_t)$

- Start with prior belief $P(X_1)$
- At every timestep have belief $P(X_t \mid y_{1:t-1})$
  - Condition on observation: $P(X_t \mid y_{1:t})$ ← Multiply likelihood "sensor model"
  - Predict (multiply motion model): $P(X_{t+1},X_t \mid y_{1:t})$ ← Multiply motion model
  - "Roll-up" (marginalize prev. time): $P(X_{t+1} \mid y_{1:t})$

Implementation

- Current belief: $P(x_t \mid y_{1:t-1}) = N(x_t; \eta_{X_t}, \Lambda_{X_t})$
- Multiply sensor and motion model

Motion model
$P(X_{t+1} \mid X_t)$ ← Represented as CLG, canonical params Λ_M, η_M

$$P(X_{t+1}, X_t \mid y_{1..t}) = P(X_t \mid y_{1:t-1})\, P(X_{t+1} \mid X_t)$$
$$= \mathcal{N}(\cdot\,; \eta_{X_t} + \eta_M, \Lambda_{X_t} + \Lambda_M)$$

- Marginalize $P(X_{t+1} \mid y_{1..t}) = \mathcal{N}(\cdot\,; \eta_A^m, \Lambda_{AA}^m)$

$$\eta_A^m = \eta_A - \Lambda_{AB}\Lambda_{BB}^{-1}\eta_B$$

$$\Lambda_{AA}^m = \Lambda_{AA} - \Lambda_{AB}\Lambda_{BB}^{-1}\Lambda_{BA}$$

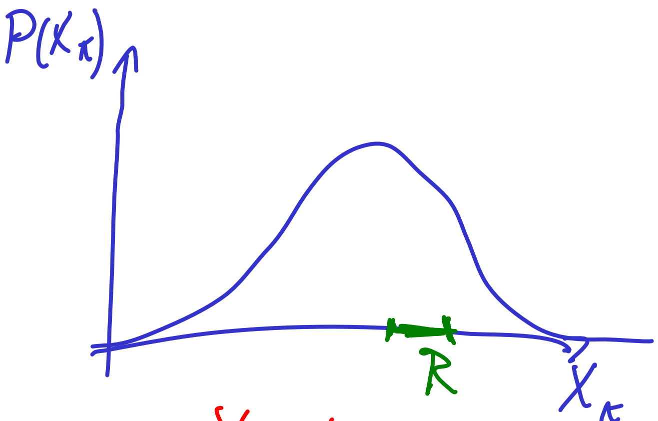
$A = X_{t+1} \mid y_{1..}y_t$
$B = X_t \mid y_1 \ldots y_t$

What if observations not "linear"?

- Linear observations:
  - $Y_t = H X_t + \text{noise}$
- Nonlinear observations:

"Motion detector": $Y_t = 1$ if $X_t \in R$

$= 0$ otherwise



If observe $Y_t = 1$

Incorporating Non-gaussian observations

- Nonlinear observation ➔ $P(Y_t \mid X_t)$ not Gaussian
- First approach: Approximate $P(Y_t \mid X_t)$ as CLG
  - Linearize $P(Y_t \mid X_t)$ around current estimate $E[X_t \mid y_{1..t-1}]$
  - Known as Extended Kalman Filter (EKF)
  - Can perform poorly if $P(Y_t \mid X_t)$ highly nonlinear

- Second approach: Approximate $P(Y_t, X_t)$ as Gaussian
  - Takes correlation in X_t into account
  - After obtaining approximation, condition on $Y_t=y_t$
    (now a "linear" observation)

Finding Gaussian approximations

- Need to find Gaussian approximation of $P(X_t,Y_t)$
- How?
  - Gaussians in Exponential Family ➔ Moment matching!!

- $E[Y_t] = \int y\, P(y_t)\, dy_t = \int y\, \underbrace{P(y_t|x_t)}_{\text{nonlinear}}\, \underbrace{P(x_t)}_{\text{Gaussian}}\, dx_t\, dy_t$

- $E[Y_t^2] = \int y^2\, P(y_t|x_t)\, P(x_t)\, dx_t\, dy_t$

- $E[X_t\, Y_t] = \int x\, y\, P(y_t|x_t)\, P(x_t)\, dx_t\, dy_t$

Linearization by integration

- Need to integrate product of Gaussian with arbitrary function
- Can do that by numerical integration
  - Approximate integral as weighted sum of evaluation points

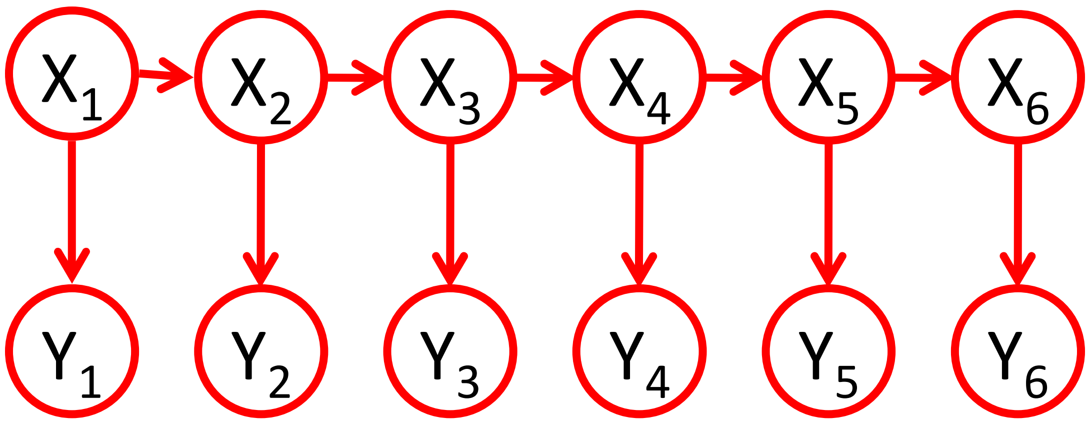
$$\int f(x,y)\,p(x)\,dx\,dy \approx \sum_i w_i\, f(x^{(i)}, y^{(i)})$$

How should we choose ?

  - Gaussian quadrature defines locations and weights of points
  - For 1 dim: **Exact** for polynomials of degree D if choosing 2D points using Gaussian quadrature
  - For higher dimensions: Need exponentially many points to achieve exact evaluation for polynomials
- Application of this is known as “Unscented” Kalman Filter (UKF)

Factored dynamical models

- So far: HMMs and Kalman filters

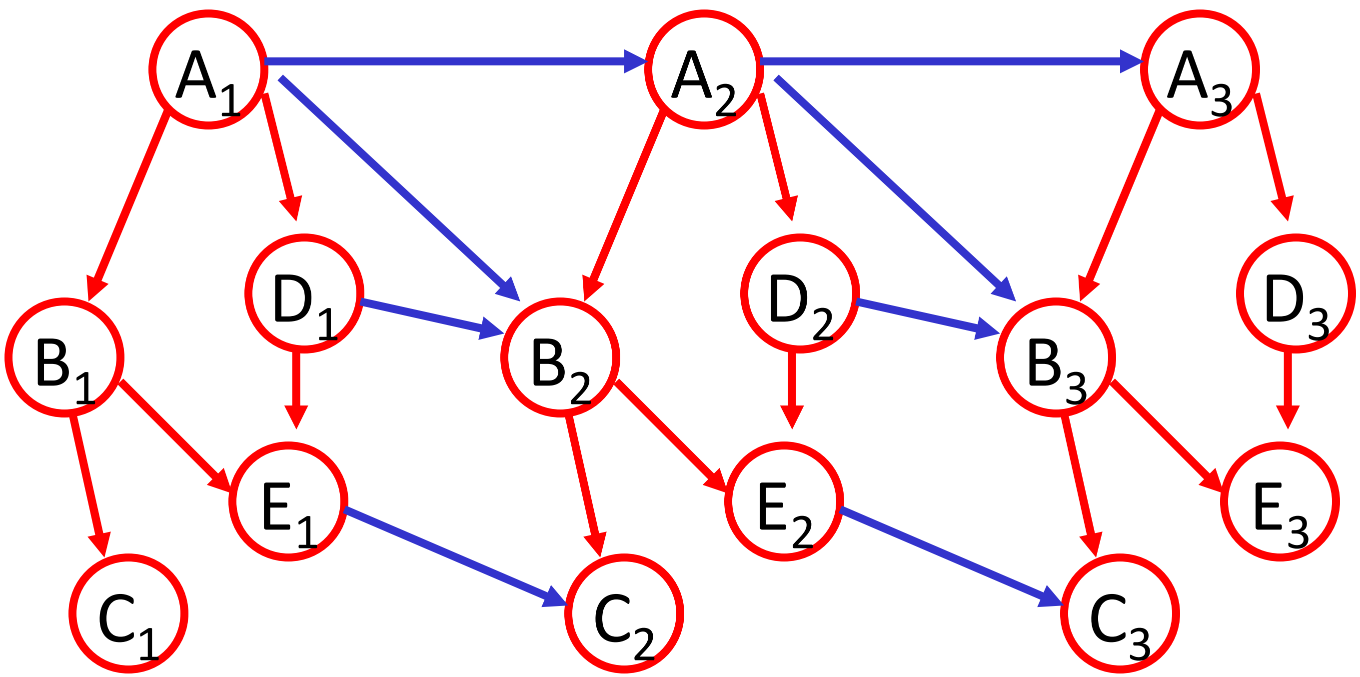


- What if we have more than one variable at each time step?
  - E.g., temperature at different locations, or road conditions in a road network?

  → Spatio-temporal models

Dynamic Bayesian Networks

- At every timestep have a Bayesian Network



- Variables at each time step t called a "slice" $\mathbf{S_t}$
- "Temporal" edges connecting S_{t+1} with S_t

Tasks

- Read Koller & Friedman Chapters 6.2.3, 15.1