

LECTURE 21 (LAST ONE!):

REVIEW

Kevin Murphy  
1 December 2004

KOLLER & FRIEDMAN CHAPTERS

Chap.	Handout	Title
2	y	Foundations (math review)
3	y	The BN representation (Bayes ball, I-maps)
4	y	Local probabilistic models (CPDs, CSI)
5	y	Undirected GMs (BN $\leftrightarrow$ MN)
6	y	Inference with GMs (overview)
7	y	Variable elimination
8	y	Clique trees
9	y	Particle based approximations
10	n	Inference as optimization (unfinished)
11	n	Inference in hybrid networks
12	y	Learning: introduction
13	y	Parameter estimation (fully obs. BNs)
14	y	Structure learning in BNs
15	y	Partially observed data (EM for BNs)

JORDAN CHAPTERS

Chap.	Handout	Title
2	n	Cond. Indep and factorization
3	n	The elimination algorithm
4	n	Prob. propagation and factor graphs
5	y	Statistical concepts
6	n	Linear regression and LMS
7	n	Linear classification
8	y	Exponential family and GLIMs
9	y	Completely observed GMs (IPF, etc)
10	y	Mixtures and conditional mixtures
11	y	The EM algorithm
12	y	HMMs
13	y	The multivariate Gaussian
14	y	Factor analysis
15	y	Kalman filtering and smoothing
16	n	Markov properties of graphs

JORDAN CHAPTERS CONT'D

Chap.	Handout	Title
17	n	The junction tree algorithm
18	n	HMM and state space models revisited
19	y	Features, maxent and duality
20	y	Iterative scaling algorithms
21	n	Sampling methods
22	n	Decision graphs
23	n	Bio-informatics

## WHAT WE COVERED 1

---

- 1 node models
  - Coins/dice (Dirichlet priors), Gaussians, exponential family
  - Bayesian vs frequentist (ML/MAP) estimation
  - Bayesian model selection (Occam's razor)
- 2 node BNs
  - Linear regression
  - Linear classification (logistic regression)
  - Generalized linear models (GLIMs)
  - Mixture models: MoG, K-means, EM
  - Latent variable models: PCA, FA
- 3 node BNs
  - Mixtures of FA
  - Mixtures of experts

## WHAT WE COVERED 3

---

- General graphs: representation
  - Independence properties (Bayes Ball, I-maps)
  - Directed vs undirected graphs, chordal graphs
- General graphs: exact inference
  - Variable elimination
  - Junction tree
- General graphs: parameter learning
  - Bayesian param. est. for fully observed BNs
  - ML for latent BNs (EM)
  - ML for fully observed UGs (IPF)
  - ML for fully observed CRFs (conjugate gradient)

## WHAT WE COVERED 2

---

- Chains
  - HMMs, forwards-backwards algorithm, EM
  - LDS, Kalman filter, EM
  - EKF, UKF, particle filtering, RB PF
- Trees
  - Belief propagation
  - Structure learning (max spanning tree)

## WHAT WE COVERED 4

---

- General BNs: structure learning
  - Search and score
  - Partial observability (structural EM, variational Bayes EM)
- General GMs: stochastic approximations
  - Likelihood weighting, Gibbs sampling, Metropolis Hastings
- General GMs: variational approximations
  - Mean field, structured, loopy belief propagation
- Applications
  - SLAM, tracking, image labeling (CRFs), language modeling (HMMs)

- Swendsen-Wang sampling, perfect sampling, details of MCMC
  - Generalized BP, theory of BP, cluster variational methods
  - Details of expectation propagation (EP)
  - Forwards propagation/ backwards sampling
  - Non-parametric Bayes (Dirichlet process, Gaussian process)
- 
- Quickscore/ QMR-DT and other speedup tricks (e.g., lazy Jtree)
  - Decision making (influence diagrams, LIMIDS, POMDPs etc)
  - First order probabilistic inference (FOPI)
- 
- Causality
  - Frequentist hypothesis testing
  - Conditional Gaussian models (mixed/ hybrid GMs)
  - Applications to error correcting codes, biology, vision, speech

## COINS (BERNOULLI TRIALS)

- We observe  $M$  iid coin flips:  $\mathcal{D} = H, H, T, H, \dots$
- Model:  $p(H) = \theta$   $p(T) = (1 - \theta)$
- We want to estimate  $\theta$  from  $D$ .
- Frequentist (maximum likelihood) approach (point estimate):

$$\hat{\theta}_{ML} = \operatorname{argmax}_{\theta} \ell(\theta; \mathcal{D})$$

where

$$\ell(\theta; D) = \log p(D|\theta) = \sum_m \log p(x^m|\theta)$$

- Bayesian approach

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}$$

or

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

- Jordan ch 5, 8, 13; Mackay ch 3, 23, 37
- Coins/dice, Gaussians, exponential family
- Bayesian vs frequentist (ML/MAP) estimation
- Bayesian vs classical hypothesis testing

## MLE FOR BERNOULLI TRIALS (L10)

- Likelihood:

$$\begin{aligned} \ell(\theta; \mathcal{D}) &= \log p(\mathcal{D}|\theta) = \log \prod_m \theta^{x^m} (1 - \theta)^{1-x^m} \\ &= \log \theta \sum_m x^m + \log(1 - \theta) \sum_m (1 - x^m) \\ &= \log \theta N_H + \log(1 - \theta) N_T \end{aligned}$$

- Take derivatives and set to zero:

$$\begin{aligned} \frac{\partial \ell}{\partial \theta} &= \frac{N_H}{\theta} - \frac{N_T}{1 - \theta} \\ \Rightarrow \theta_{ML}^* &= \frac{N_H}{N_H + N_T} \end{aligned}$$

- The counts  $N_H = \sum_m x^m$  and  $N_T = \sum_m (1 - x^m)$  are sufficient statistics of the data  $D$ .

## BAYESIAN ESTIMATION FOR BERNOULLI TRIALS (L11)

- Likelihood

$$P(D|\theta) = \theta^{N_H}(1 - \theta)^{N_T}$$

- Conjugate Beta Prior

$$P(\theta|\alpha) = \mathcal{B}(\theta; \alpha_h, \alpha_t) \stackrel{\text{def}}{=} \frac{1}{Z(\alpha_h, \alpha_t)} \theta^{\alpha_h-1} (1 - \theta)^{\alpha_t-1}$$

- Posterior

$$\begin{aligned} P(\theta|D, \alpha) &= \frac{P(\theta|\alpha)P(D|\theta)}{P(D|\alpha)} \\ &= \frac{1}{Z(\alpha_h, \alpha_t)P(D|\alpha)} \theta^{\alpha_h-1} (1 - \theta)^{\alpha_t-1} (1 - \theta)^{N_t} \\ &= \mathcal{B}(\theta; \alpha_h + N_h, \alpha_t + N_t) \end{aligned}$$

- Posterior mean  $E\theta = \frac{\alpha_h}{\alpha_h + \alpha_t}$ .

## BAYESIAN HYPOTHESIS TESTING

- We want to compute the posterior ratio of the 2 hypotheses:

$$\frac{P(H_1|D)}{P(H_0|D)} = \frac{P(D|H_1)P(H_1)}{P(D|H_0)P(H_0)}$$

- Let us assume a uniform prior  $P(H_0) = P(H_1) = 0.5$ .
- Then we just focus on the ratio of the marginal likelihoods:

$$P(D|H_1) = \int_0^1 d\theta P(D|\theta, H_1)P(\theta|H_1)$$

- For  $H_0$ , there is no free parameter, so

$$P(D|H_0) = 0.5^N$$

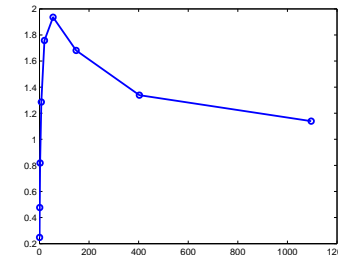
where  $N$  is the number of coin tosses in  $D$ .

## EXAMPLE OF CLASSICAL HYPOTHESIS TESTING (L15)

- When spun on edge  $N = 250$  times, a Belgian one-euro coin came up heads  $Y = 140$  times and tails 110.
- We would like to distinguish two models, or hypotheses:  $H_0$  means the coin is unbiased (so  $p = 0.5$ );  $H_1$  means the coin is biased (has probability of heads  $p \neq 0.5$ ).
- p-value is “less than 7%”:  $p = P(Y \geq 140) + P(Y \leq 110) = 0.066$ :  
 $n=250$ ;  $p = 0.5$ ;  $y = 140$ ;  
 $p = (1 - \text{binocdf}(y-1, n, p)) + \text{binocdf}(n-y, n, p)$
- If  $Y = 141$ , we get  $p = 0.0497$ , so we can reject the null hypothesis at significance level 0.05.
- But is the coin really biased?

## SO, IS THE COIN BIASED OR NOT?

- We plot the Bayes factor vs hyperparameter  $\alpha$ :



- For a uniform prior,  $\frac{P(H_1|D)}{P(H_0|D)} = 0.48$ , (weakly) favoring the fair coin hypothesis  $H_0$ !
- At best, for  $\alpha = 50$ , we can make the biased hypothesis twice as likely.
- Not as dramatic as saying “we reject the null hypothesis (fair coin) with significance 6.6%”.

## FROM COINS TO DICE

---

- Likelihood: binomial  $\rightarrow$  multinomial

$$P(D|\vec{\theta}) = \prod_i \theta_i^{N_i}$$

- Prior: beta  $\rightarrow$  Dirichlet

$$P(\vec{\theta}|\vec{\alpha}) = \frac{1}{Z(\vec{\alpha})} \prod_i \theta_i^{\alpha_i - 1}$$

where

$$Z(\vec{\alpha}) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}$$

- Posterior: beta  $\rightarrow$  Dirichlet

$$P(\vec{\theta}|D) = \text{Dir}(\vec{\alpha} + \vec{N})$$

- Evidence (marginal likelihood)

$$P(D|\vec{\alpha}) = \frac{Z(\vec{\alpha} + \vec{N})}{Z(\vec{\alpha})} = \frac{\prod_i \Gamma(\alpha_i + N_i)}{\prod_i \Gamma(\alpha_i)} \frac{\Gamma(\sum_i \alpha_i)}{\Gamma(\sum_i \alpha_i + N_i)}$$

## FUN WITH GAUSSIANS

---

- Bayesian estimation of 1D Gaussian (homework 5)
- MLE for multivariate Gaussian (Jordan ch 13)
- Bayesian estimation for multivariate Gaussian (Minka TR)
- Inference with multivariate Gaussians (Jordan ch 13)
- Moment vs canonical parameters (Jordan ch 13)

## MLE FOR UNIVARIATE NORMAL (L10)

---

- We observe  $M$  iid real samples:  $\mathcal{D} = 1.18, -0.25, 0.78, \dots$
- Model:  $p(x) = (2\pi\sigma^2)^{-1/2} \exp\{-(x - \mu)^2/2\sigma^2\}$
- Log likelihood:

$$\begin{aligned} \ell(\theta; \mathcal{D}) &= \log p(\mathcal{D}|\theta) \\ &= -\frac{M}{2} \log(2\pi\sigma^2) - \frac{1}{2} \sum_m \frac{(x^m - \mu)^2}{\sigma^2} \end{aligned}$$

- Take derivatives and set to zero:

$$\begin{aligned} \frac{\partial \ell}{\partial \mu} &= (1/\sigma^2) \sum_m (x_m - \mu) \\ \frac{\partial \ell}{\partial \sigma^2} &= -\frac{M}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_m (x_m - \mu)^2 \\ \Rightarrow \mu_{\text{ML}} &= (1/M) \sum_m x_m \\ \sigma_{\text{ML}}^2 &= (1/M) \sum_m (x_m - \mu_{\text{ML}})^2 \end{aligned}$$

## EXPONENTIAL FAMILY (L4, L10)

---

- For a numeric random variable  $\mathbf{x}$

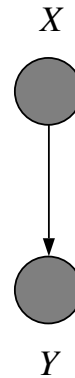
$$\begin{aligned} p(\mathbf{x}|\eta) &= h(\mathbf{x}) \exp\{\eta^\top T(\mathbf{x}) - A(\eta)\} \\ &= \frac{1}{Z(\eta)} h(\mathbf{x}) \exp\{\eta^\top T(\mathbf{x})\} \end{aligned}$$

is an exponential family distribution with *natural (canonical) parameter*  $\eta$ .

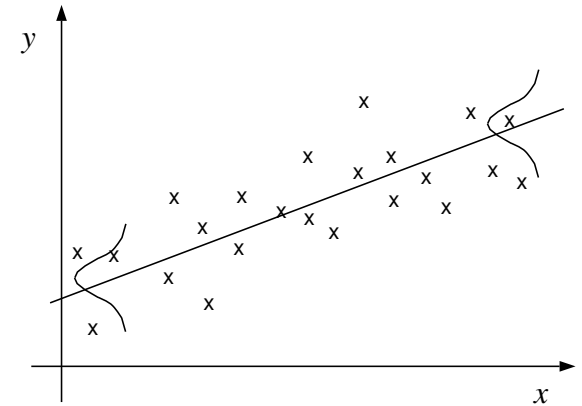
- Function  $T(\mathbf{x})$  is a *sufficient statistic*.
- Function  $A(\eta) = \log Z(\eta)$  is the log normalizer.
- Examples: Bernoulli, multinomial, Gaussian, Poisson, gamma,...
- A distribution  $p(x)$  has finite sufficient statistics (independent of number of data cases) iff it is in the exponential family.
- See Jordan ch 8

## 2 NODE BAYES NETS

- Linear regression (Jordan ch 6)
- Linear classification (logistic regression; Jordan ch 7)
- Generalized linear models (GLIMs; Jordan ch 8)
- Mixture models: MoG, K-means, EM (Jordan ch 10)
- Latent variable models: PCA, FA (Jordan ch 14)



## LINEAR REGRESSION (L11)



## MLE FOR LINEAR REGRESSION

- For vector outputs,

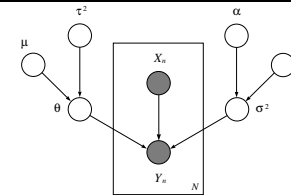
$$A = S_{YX'} S_{XX'}^{-1}$$

where  $S_{YX'} = \sum_m y_m x_m^T$  and  $S_{XX'} = \sum_m x_m x_m^T$ .

- In the special case of scalar outputs, let  $A = \theta^T$ , and the design matrix  $X = [x_m^T]$  stacked as rows and  $Y = [y_m]$  a column vector. Then we get the normal equations

$$\theta = (X^T X)^{-1} X^T Y$$

## BAYESIAN 1D LINEAR REGRESSION



- For scalar (1D) output

$$p(y_n | x_n, \theta, \sigma^2) p(\theta | \mu, \tau^2) p(\sigma^2 | \alpha, \beta)$$

Gaussian  $\times$  Gaussian  $\times$  Gamma

- For vector output

$$p(y_n | x_n, A, \Sigma) p(A | \mu, \tau^2) p(\Sigma | \alpha, \beta)$$

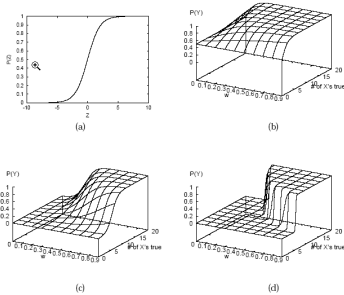
Gaussian  $\times$  matrix-Gaussian  $\times$  Wishart

- See Tom Minka tutorial

## LOGISTIC REGRESSION (L4)

$$P(Y = 1 | X_1, \dots, X_n) = \sigma(w_0 + \sum_{i=1}^n w_i X_i)$$

$P(Y = 1)$  vs number of  $X$ 's that are on vs  $w$



- a: 1D sigmoid
- b:  $w_0 = 0$
- c:  $w_0 = -5$
- d:  $w$  and  $w_0$  are multiplied by 10

## GENERALIZED LINEAR MODELS

Canonical CPDs for  $X \rightarrow Y$  (L4)

$X$	$Y$	$p(Y X)$
$\mathbb{R}^n$	$\mathbb{R}^m$	Gauss( $Y; WX + \mu, \Sigma$ )
$\mathbb{R}^n$	$\{0, 1\}$	Bernoulli( $Y; p = \frac{1}{1+e^{-\theta^T x}}$ )
$\{0, 1\}^n$	$\{0, 1\}$	Bernoulli( $Y; p = \frac{1}{1+e^{-\theta^T x}}$ )
$\mathbb{R}^n$	$\{1, \dots, K\}$	Multinomial( $Y; p_i = \text{softmax}(x, \theta)$ )

Learn using IRLS or conjugate gradient (L11)

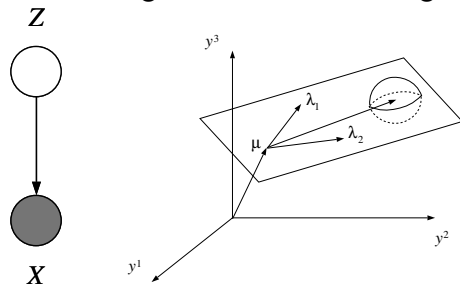
## FACTOR ANALYSIS (L17)

- Unsupervised linear regression is called factor analysis.

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

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

where  $\Lambda$  is the factor loading matrix and  $\Psi$  is diagonal.



- To generate data, first generate a point within the manifold then add noise. Coordinates of point are components of latent variable.
- PCA (Karhunen-Loeve Transform) is zero noise limit of FA.

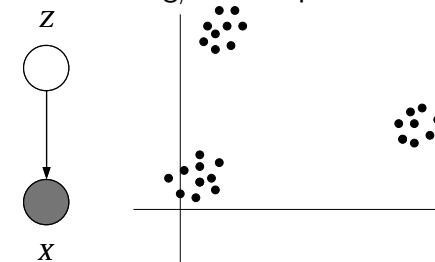
## MIXTURES OF GAUSSIANS (L12)

- Mixture of Gaussians:

$$P(Z = i) = \theta_i$$

$$p(X = x | Z = i) = \mathcal{N}(x; \mu_i, \Sigma_i)$$

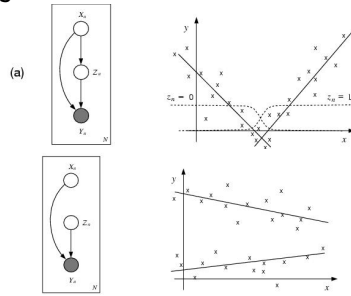
- This can be used for classification (supervised) and clustering/vector quantization (unsupervised).



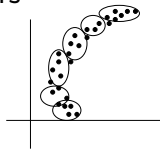
- We can find MLE/MAP estimates of the parameters using EM.
- K-means is a deterministic approximation (vector quantization).

### 3 NODE BAYES NETS (L12)

- Mixtures of experts



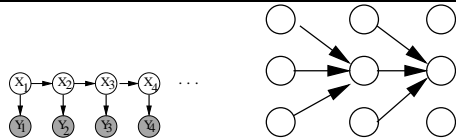
- Mixtures of factor analysers



### CHAINS

- HMMs (Jordan ch 12, Rabiner tutorial)
- LDS (Jordan ch 15, handouts on web)
- Nonlinear state space models (my DBN tutorial)

### FORWARDS-BACKWARDS ALGORITHM (L8)



$$\alpha_t(j) = \sum_i \alpha_{t-1}(i) A(i, j) B_t(j)$$

$$\alpha_t = (A^T \alpha_{t-1}) * B_t$$

$$\beta_t(i) = \sum_j \beta_{t+1}(j) A(i, j) B_{t+1}(j)$$

$$\beta_t = A(\beta_{t+1} * B_{t+1})$$

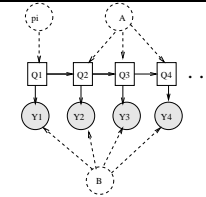
$$\xi_t(i, j) = \alpha_t(i) \beta_{t+1}(j) A(i, j) B_{t+1}(j)$$

$$\xi_t = (\alpha_t (\beta_{t+1} * B_{t+1})^T) * A$$

$$\gamma_t(i) \propto \alpha_t(i) \beta_t(j)$$

$$\gamma_t \propto \alpha_t * \beta_t$$

### LEARNING AN HMM (L10, L12)



- Consider a time-invariant hidden Markov model (HMM)
  - State transition matrix  $A(i, j) \stackrel{\text{def}}{=} P(X_t = j | X_{t-1} = i)$ ,
  - Discrete observation matrix  $B(i, j) \stackrel{\text{def}}{=} P(Y_t = j | X_t = i)$
  - State prior  $\pi(i) \stackrel{\text{def}}{=} P(X_1 = i)$ .
- If all nodes are observed, we can find the globally optimal MLE.
- Otherwise using EM (aka Baum Welch).

## KALMAN FILTER (L17, L18)

• LDS model:  $x_t = Ax_{t-1} + v_t$ ,  $y_t = Cx_t + w_t$

• Time update (prediction step):

$$x_{t|t-1} = Ax_{t-1|t-1}, \quad P_{t|t-1} = AP_{t-1|t-1}A^T + Q, \quad y_{t|t-1} = Cx_{t|t-1}$$

• Measurement update (correction step):

$$\tilde{y}_t = y_t - \hat{y}_{t|t-1} \text{ (error/ innovation)}$$

$$P_{\tilde{y}_t} = CP_{t|t-1}C^T + R \text{ (covariance of error)}$$

$$P_{x_t y_t} = P_{t|t-1}C^T \text{ (cross covariance)}$$

$$K_t = P_{x_t y_t} P_{\tilde{y}_t}^{-1} \text{ (Kalman gain matrix)}$$

$$x_{t|t} = x_{t|t-1} + K_t(y_t - \hat{y}_{t|t-1})$$

$$P_{t|t} = P_{t|t-1} - K_t P_{x_t y_t}^T$$

## KF FOR SLAM (L18)

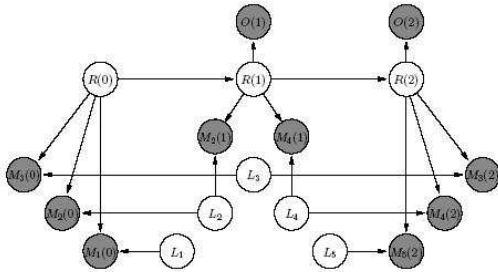
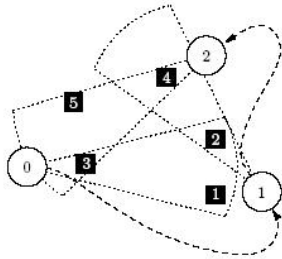
• State is location of robot and landmarks

$$X_t = (R_t, L_t^{1:N})$$

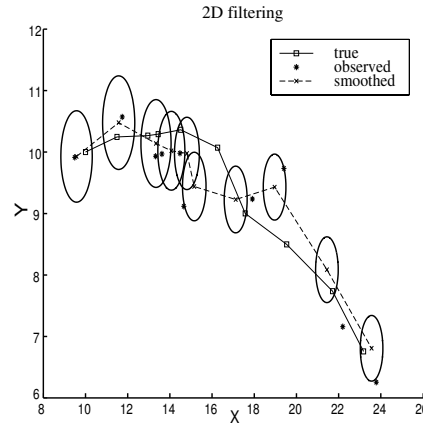
• Measure location of subset of landmarks at each time step.

• Assume everything is linear Gaussian.

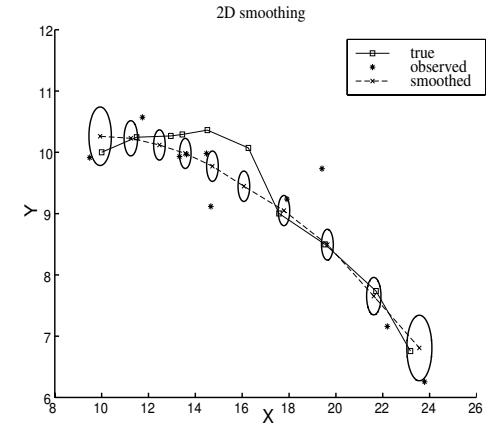
• Use Kalman filter to solve optimally.



## KF FOR 2D TRACKING (L17)



(a)



(b)

## APPROXIMATE DETERMINISTIC FILTERING (L18)

• Extended Kalman filter (EKF)

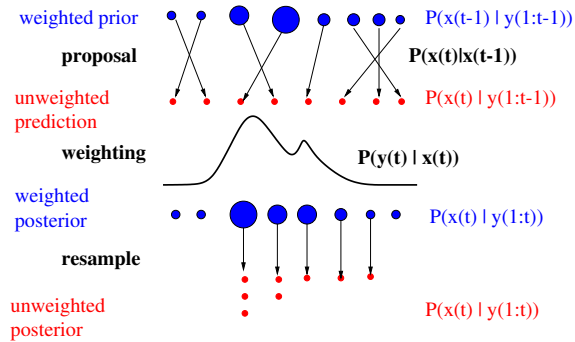
• Unscented Kalman filter (UKF)

• Assumed density filter (ADF)

$$\begin{array}{ccc} & \hat{\alpha}_t & \hat{\alpha}_{t+1} & \text{exact} \\ U \nearrow & P \downarrow & U \nearrow & \\ \tilde{\alpha}_{t-1} & \tilde{\alpha}_t & \tilde{\alpha}_{t+1} & \text{approx} \end{array}$$

## PARTICLE FILTERING (SEQUENTIAL MONTE CARLO) (L19)

- PF is sequential importance sampling with resampling (SISR).
- Goal is to estimate  $P(x_{1:t}|y_{1:t})$  recursively (online) for a state-space model for which Kalman filter/ HMM filter is inapplicable.

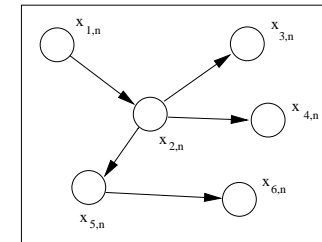


## GENERAL GRAPHS

- Representation: Markov properties, CPDs, log linear models
- Exact inference: var elim, Jtree
- Fully observed param learning
- Fully observed structure learning
- Partially observed param learning
- Approximate inference

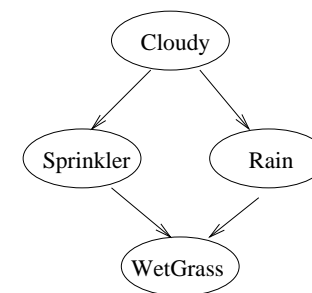
## TREES

- Inference (belief propagation): L9, Yedidia tutorial
- Structure learning (max weight spanning tree): L16
- Application: KF trees for multiscale image analysis (skipped)



## EXAMPLE BN: WATER SPRINKLER (L1)

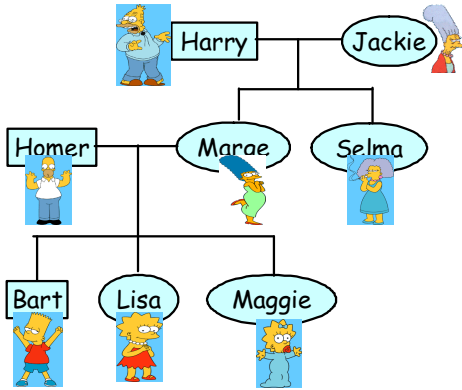
$$P(X_{1:N}) = \prod_{i=1}^N P(X_i | \text{Pa}(X_i))$$



$$P(C, S, R, W) = P(C)P(S|C)P(R|C)P(W|S, R)$$

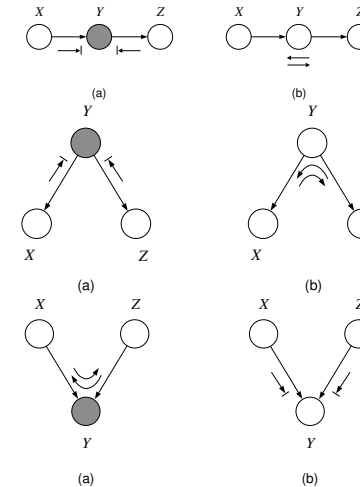
## BAYES NET FOR GENETIC PEDIGREE ANALYSIS (L1)

- $G_i \in \{a, b, o\} \times \{a, b, o\} =$  genotype (allele) of person  $i$
- $B_i \in \{a, b, o, ab\} =$  phenotype (blood type) of person  $i$



## GLOBAL MARKOV PROPERTIES FOR DGs: BAYES-BALL (L2)

$A$  is d-separated from  $B$  given  $C$  if we cannot send a ball from any node in  $A$  to any node in  $B$  according to the rules below, where shaded nodes are in  $C$ .



## MARKOV PROPERTIES FOR UGs (L3)

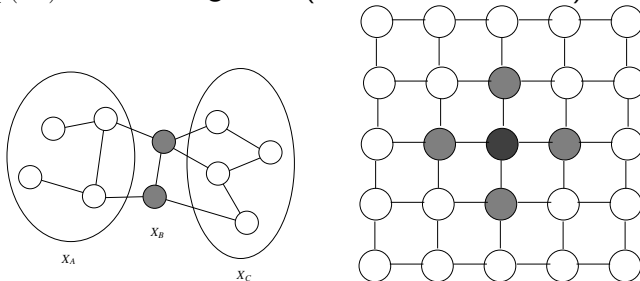
- Defn: the global Markov properties of a UG  $H$  are

$$I(H) = \{(X \perp Y|Z) : sep_H(X; Y|Z)\}$$

- Defn: The local Markov independencies are

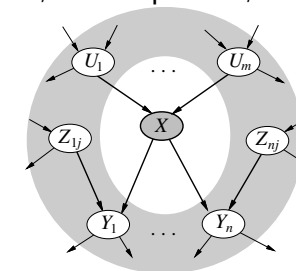
$$I_l(H) = \{(X \perp V \setminus \{X\} \setminus N_H(X) | N_H(X)) : X \in V\}$$

where  $N_H(X)$  are the neighbors (Markov blanket).



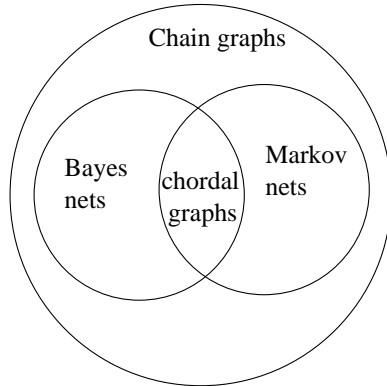
## CONVERTING BAYES NETS TO MARKOV NETS (L3)

- Defn: A Markov net  $H$  is an I-map for a Bayes net  $G$  if  $I(H) \subseteq I(G)$ .
- We can construct a minimal I-map for a BN by finding the minimal Markov blanket for each node.
- We need to block all active paths coming into node  $X$ , from parents, children, and co-parents; so connect them all to  $X$ .

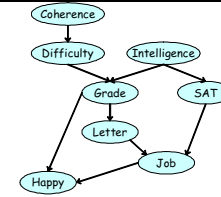


## CHORDAL GRAPHS (L4)

- Chordal graphs encode independencies that can be exactly represented by either directed or undirected graphs.
- Chain graphs combine directed and undirected graphs and represent a larger set of distributions.



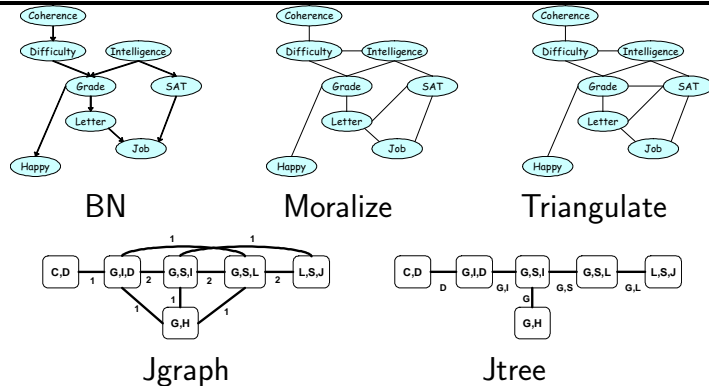
## VARIABLE ELIMINATION ALGORITHM (L7)



- Key idea 1: push sum inside products.
- Key idea 2: use (non-serial) dynamic programming to cache shared subexpressions.

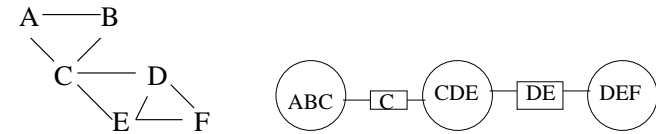
$$\begin{aligned}
 P(J) &= \sum_L \sum_S \sum_G \sum_H \sum_I \sum_D \sum_C P(C, D, I, G, S, L, J, H) \\
 &= \sum_L \sum_S \sum_G \sum_H \sum_I \sum_D \sum_C P(C)P(D|C)P(I)P(G|I, D)P(S|I)P(L|G)P(J|L, S)P(H|G, J) \\
 &= \sum_L \sum_S \sum_G \sum_H \sum_I \sum_D \sum_C \phi_C(C)\phi_D(D, C)\phi_I(I)\phi_G(G, I, D)\phi_S(S, I)\phi_L(L, G)\phi_J(J, L, S)\phi_H(H, G, J) \\
 &= \sum_L \sum_S \phi_J(J, L, S) \sum_G \phi_L(L, G) \sum_H \phi_H(H, G, J) \sum_I \phi_S(S, I)\phi_I(I) \sum_D \phi_G(G, I, D) \sum_C \phi_C(C)\phi_D(D, C)
 \end{aligned}$$

## FROM BAYES NET TO JTREE (L8)



## MESSAGE PASSING ON JTREES (L8, L9)

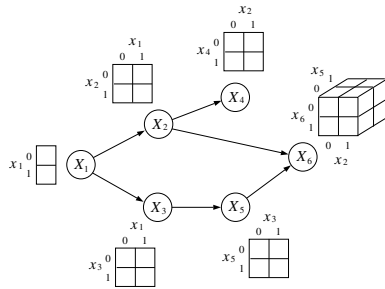
- Hugin vs Shafer Shenoy



## MLE FOR FULLY OBSERVED BNs (L10)

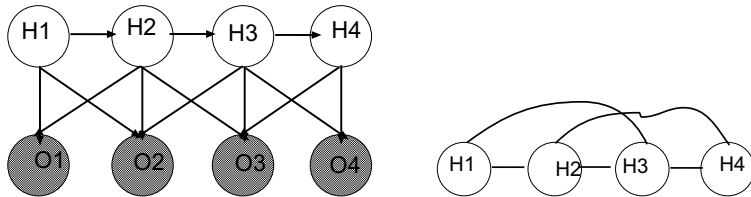
- If we assume the parameters for each CPD are globally independent, then the log-likelihood function decomposes into a sum of local terms, one per node:

$$\log p(\mathcal{D}|\theta) = \log \prod_m \prod_i p(\mathbf{x}_i^m | x_{\pi_i}, \theta_i) = \sum_i \sum_m \log p(\mathbf{x}_i^m | x_{\pi_i}, \theta_i)$$



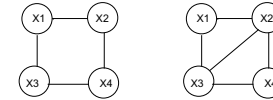
## LEARNING CRFs (L14)

- Conditional random fields are discriminative models.
- Assuming fully observed training data, learning can be done using conjugate gradient descent, just as in a regular MRF with non-maximal cliques.
- Gradient requires computing the partition function, which is (in general) only tractable for low treewidth models (eg chains).



## MLE FOR FULLY OBSERVED UGM (L13)

- Is the graph *decomposable* (triangulated)?
- Are all the clique potentials defined on maximal cliques (not sub-cliques)? e.g.,  $\psi_{123}, \psi_{234}$  not  $\psi_{12}, \psi_{23}, \dots$

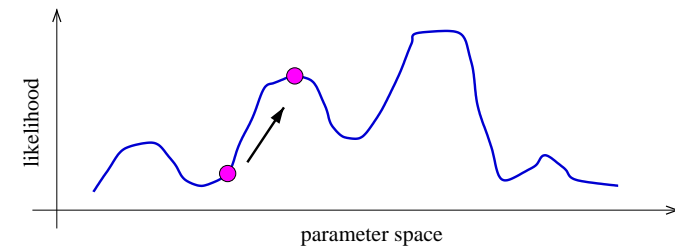


- Are the clique potentials full tables (or Gaussians), or parameterized more compactly, e.g.,  $\psi_c(x_c) = \exp(\sum_k w_k f_k(x_c))$ ?

Decomposable?	Max. Cliques	Tabular	Method
Yes	Yes	Yes	Direct
-	-	Yes	IPF
-	-	-	Gradient ascent
-	-	-	Iterative scaling

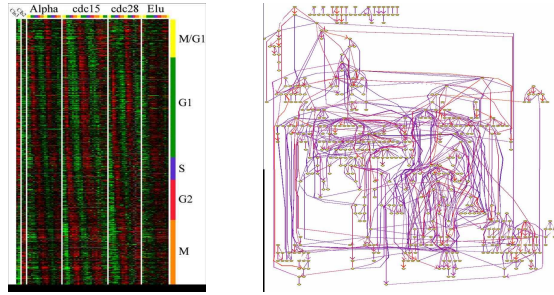
## MLE FOR PARTIALLY OBSERVED BNs (L12)

- Use (conjugate) gradient or EM
- M-step is what we did for the 1 node/2 node BNs



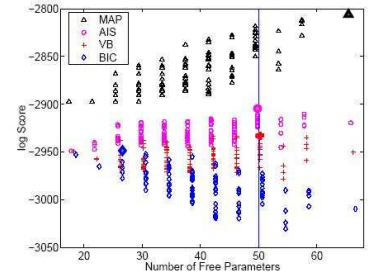
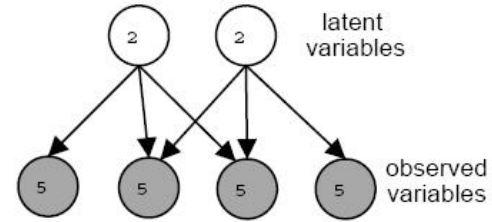
## LEARNING STRUCTURE OF FULLY OBSERVED BNs (L15, L16)

- Search + score (local search + Occam's razor)



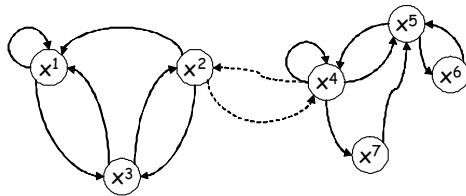
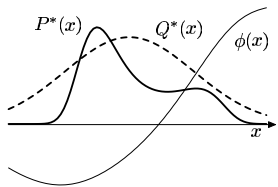
## LEARNING STRUCTURE OF PARTIALLY OBSERVED BNs (L16)

- Search = local search
- Score = expected BIC (structural EM)
- Score = variational Bayes (VB-EM)



## MONTE CARLO METHODS (L19)

- Importance sampling
- Particle filtering
- RBPf
- MCMC: Gibbs sampling and Metropolis Hastings



## VARIATIONAL METHODS (L20)

- Iterative Conditional Modes (ICM)
- Mean field
- Structured variational methods
- Loopy belief propagation

