

# Graphical Models

## Inference and Learning

### Lecture 6

MVA

2020 – 2021

<http://thoth.inrialpes.fr/~alahari/disinfllearn>

# Recall

- Graphical Models
  - Directed vs Undirected
  - Representation and Modeling
- Problem formulation
  - Energy/cost function
- MAP estimation
  - Belief propagation, TRW, graph cuts, LP relaxation, primal-dual, dual decomposition
- Learning
  - Maximum likelihood, max-margin learning

# Recall

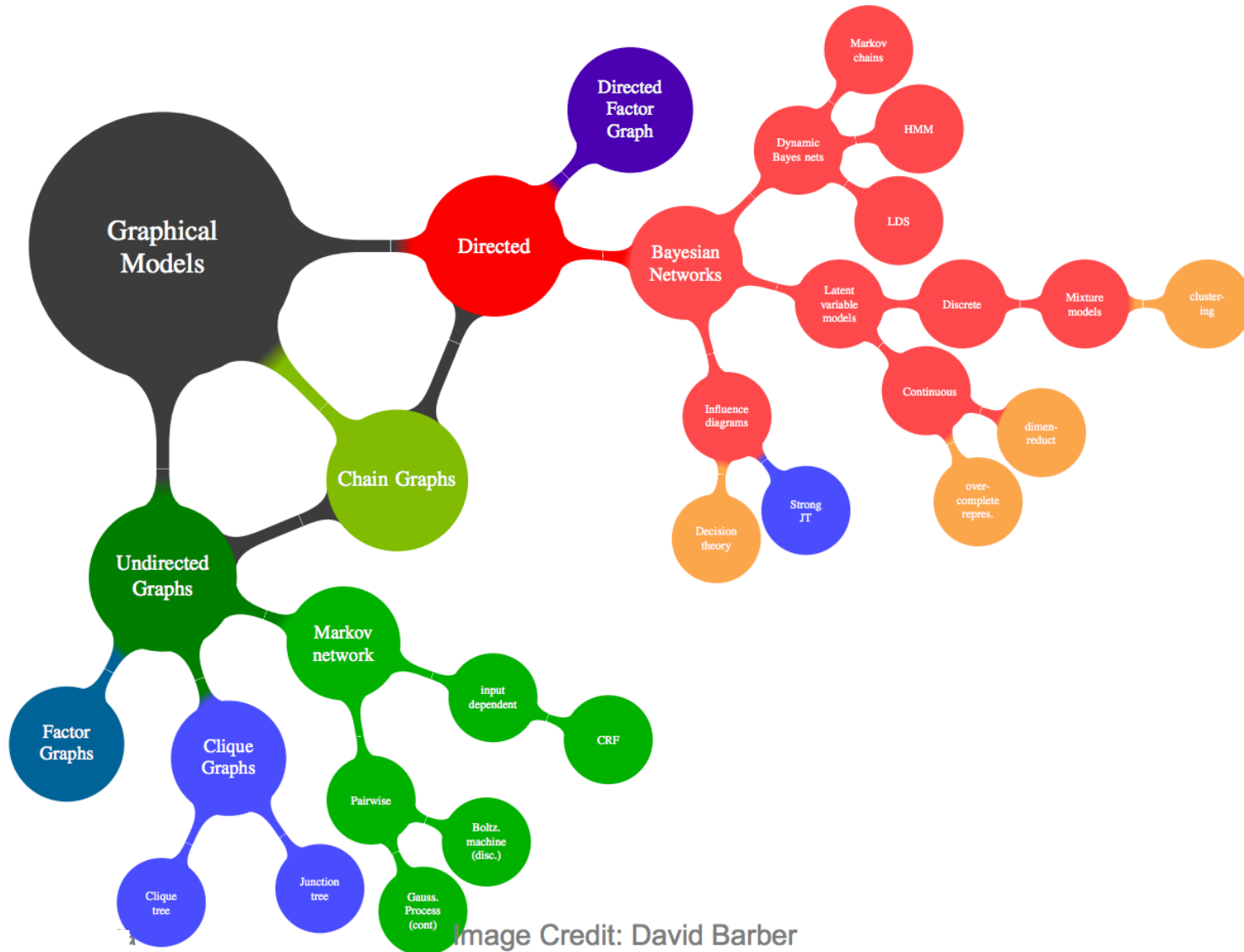


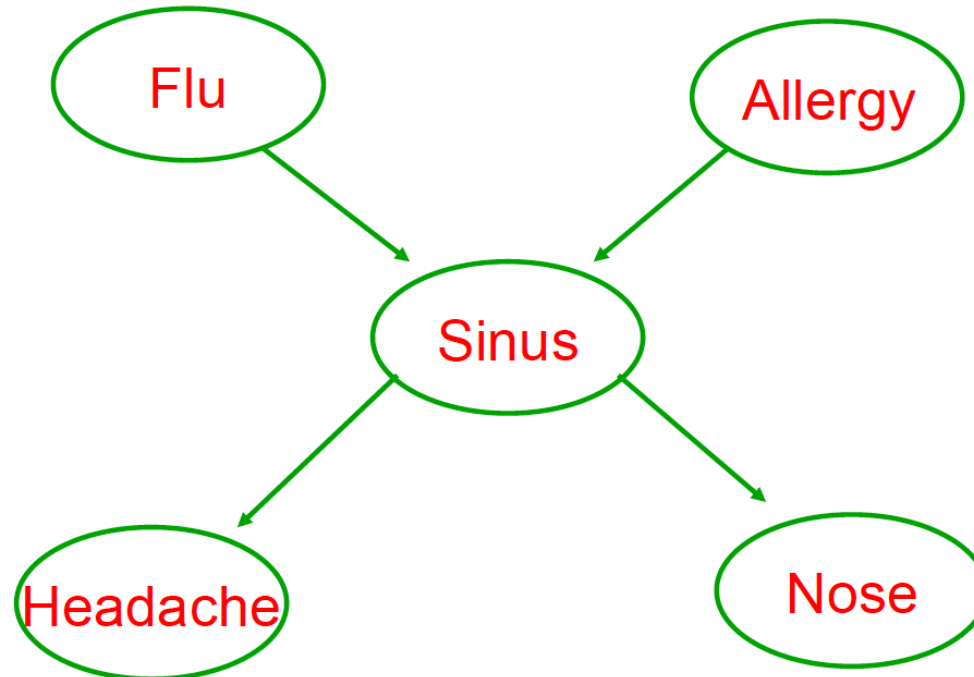
Image Credit: David Barber

# This class

- Bayesian Networks
  - Parameter Learning
  - Structure Learning
  - Inference

# Bayesian Networks

- Example



# Independencies in problem

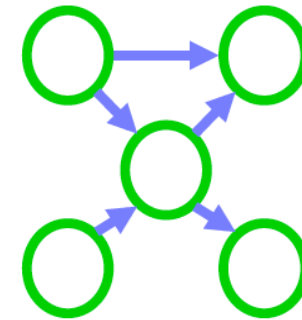
**World, Data, reality:**



True distribution  $P$   
contains independence  
assertions

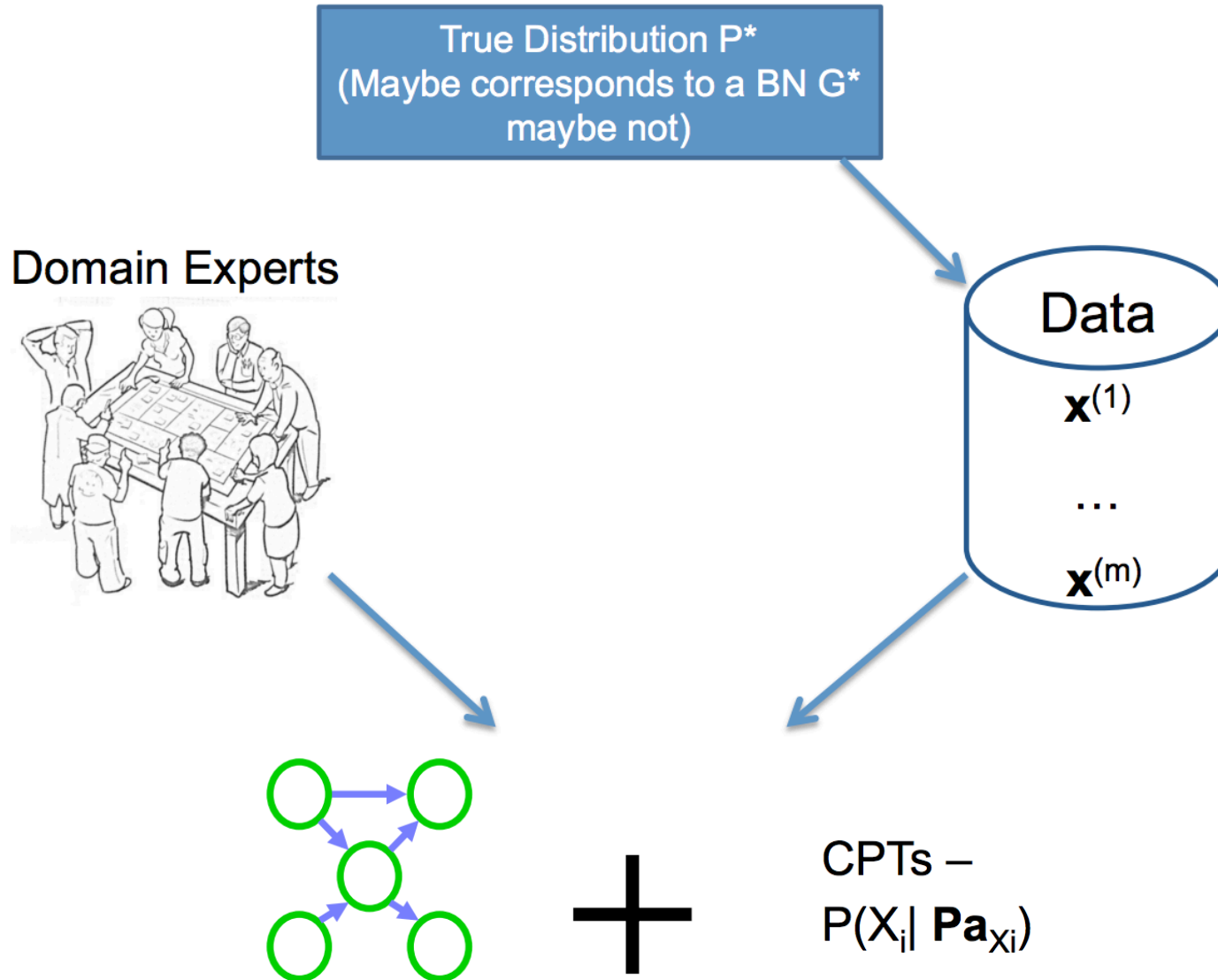


**BN:**



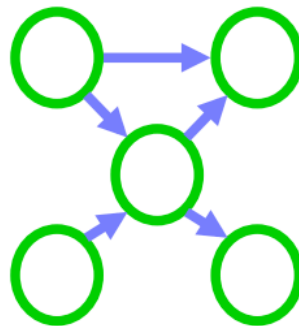
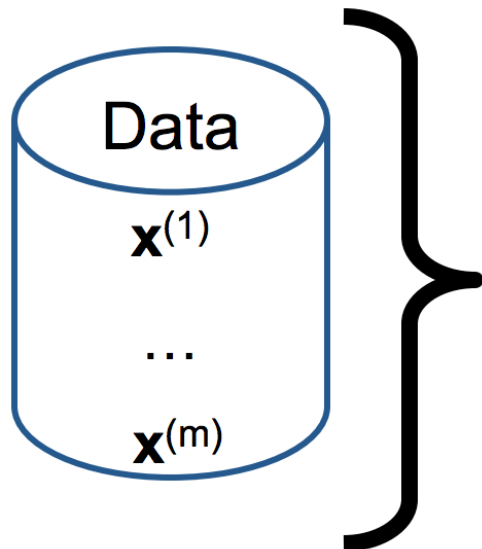
Graph  $G$   
encodes local  
independence  
assumptions

# Learning Bayesian Nets



# Learning Bayesian Nets

	Known structure	Unknown structure
Fully observable data	Very easy	Hard
Missing data	Somewhat easy (EM)	Very very hard



+

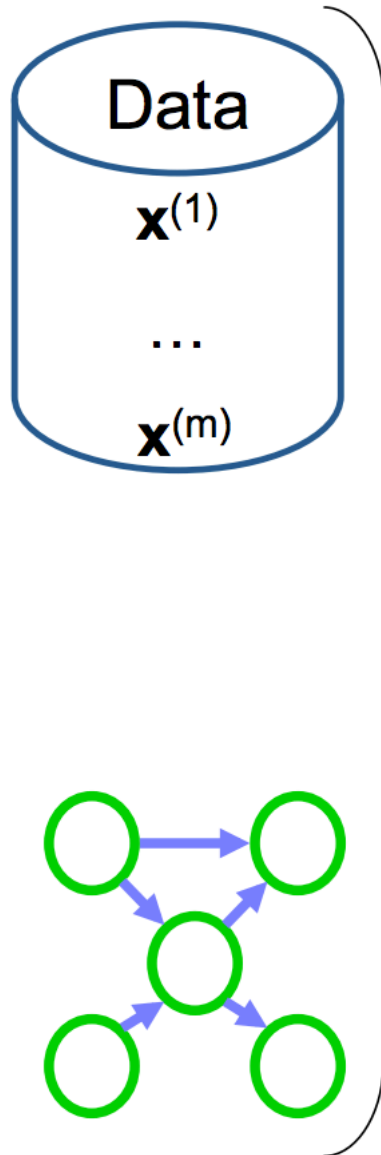
CPTs –  
 $P(X_i | \mathbf{Pa}_{X_i})$



# Maximum Likelihood Estimation

- Goal: Find a good  $\theta$
- What is a good  $\theta$ ?
  - One that makes it likely for us to have seen this data
  - Quality of  $\theta = \text{Likelihood}(\theta; D) = P(D | \theta)$
- Why MLE?
  - $\text{Log-likelihood}(\theta) = \text{entropy}(P^*) - \text{KL}(P^*, P(D | \theta))$
  - i.e., maximizing LL = minimizing KL

# MLE: Learning the CPTs




For each discrete variable  $X_i$

$$\hat{P}_{MLE}(X_i = a \mid \text{Pa}_{X_i} = b) = \frac{\text{Count}(X_i = a, \text{Pa}_{X_i} = b)}{\text{Count}(\text{Pa}_{X_i} = b)}$$

# Bayesian Estimation

- Exploit priors
  - Priors: Beliefs before experiments are conducted
  - Help deal with unseen data
  - Bias us towards “simpler” models
- Beta prior distribution

$$P(\theta) = \frac{\theta^{\alpha_H - 1} (1 - \theta)^{\alpha_T - 1}}{B(\alpha_H, \alpha_T)} \sim \text{Beta}(\alpha_H, \alpha_T)$$

constant 

# Bayesian Estimation

- Posterior

$$P(\mathcal{D} | \theta) = \theta^{m_H} (1 - \theta)^{m_T}$$

$$P(\theta) = \frac{\theta^{\alpha_H - 1} (1 - \theta)^{\alpha_T - 1}}{B(\alpha_H, \alpha_T)} \sim \text{Beta}(\alpha_H, \alpha_T)$$

$$P(\theta | \mathcal{D}) \propto P(\mathcal{D} | \theta) P(\theta)$$

$$P(\theta | \mathcal{D}) \sim \text{Beta}(m_H + \alpha_H, m_T + \alpha_T)$$

# Bayesian Estimation

- MAP: use most likely parameter

$$\hat{\theta} = \arg \max_{\theta} P(\theta | \mathcal{D})$$

$$P(\theta | \mathcal{D}) \sim \text{Beta}(m_H + \alpha_H, m_T + \alpha_T)$$

- Beta prior equiv. to extra H/T
- As  $m \rightarrow \infty$ , prior is “forgotten”
- But, for small sample size, prior is important !

# Bayesian Estimation

- What about the multinomial case?
- Use a Dirichlet for the prior

$$\theta \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_k) \sim \prod_i \theta_i^{\alpha_i - 1}$$

# Meta BN: Bayesian view of BN

- Show parameters explicitly as variables
- Two examples (on board)

# Global parameter independence

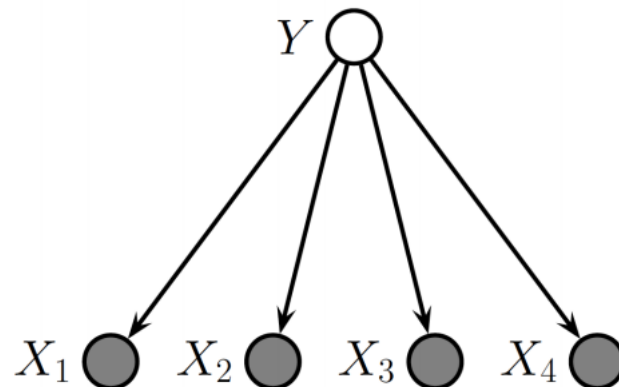
- All CPT parameters are independent
  - Common assumption
- Prior over parameters is product of prior over CPTs, i.e.,

$$P(\theta \mid \mathcal{D}) = \prod_i P(\theta_{X_i \mid \text{Pa}_{X_i}} \mid \mathcal{D})$$



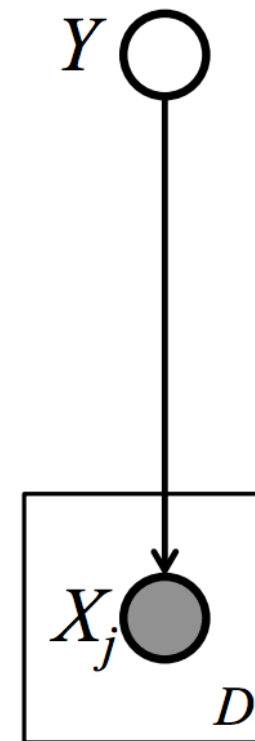
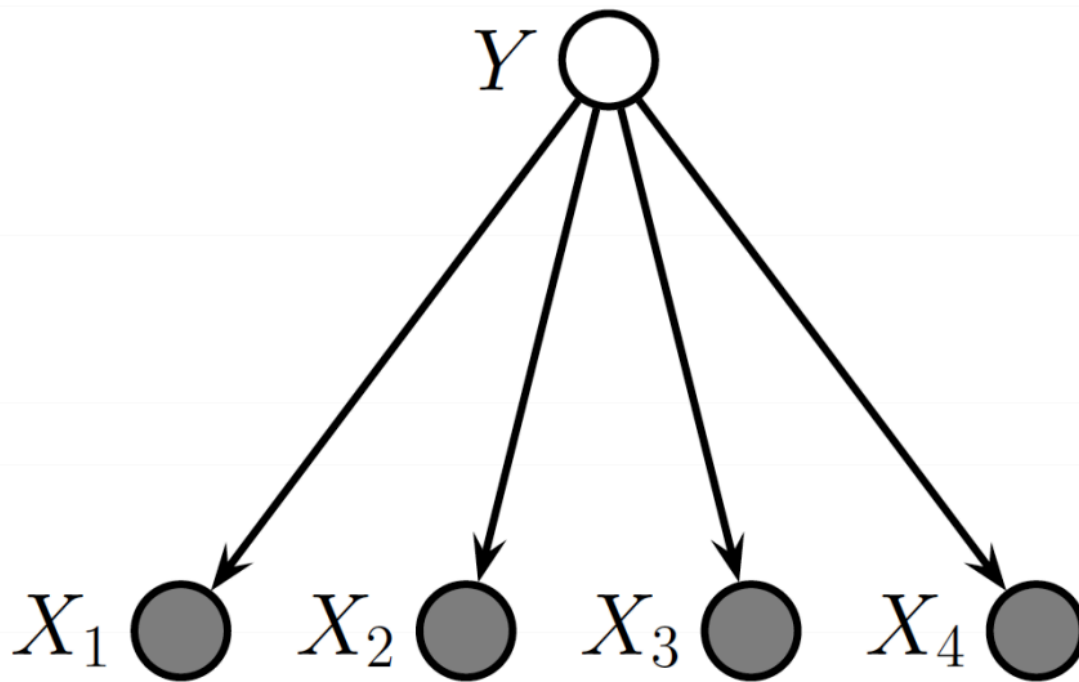
# Parameter Sharing

- Consider the scenario, where  $n$  random variables  $X_1, X_2, \dots, X_n$  represent coin tosses of the **same** coin.
- What is the corresponding BN?



# Parameter Sharing

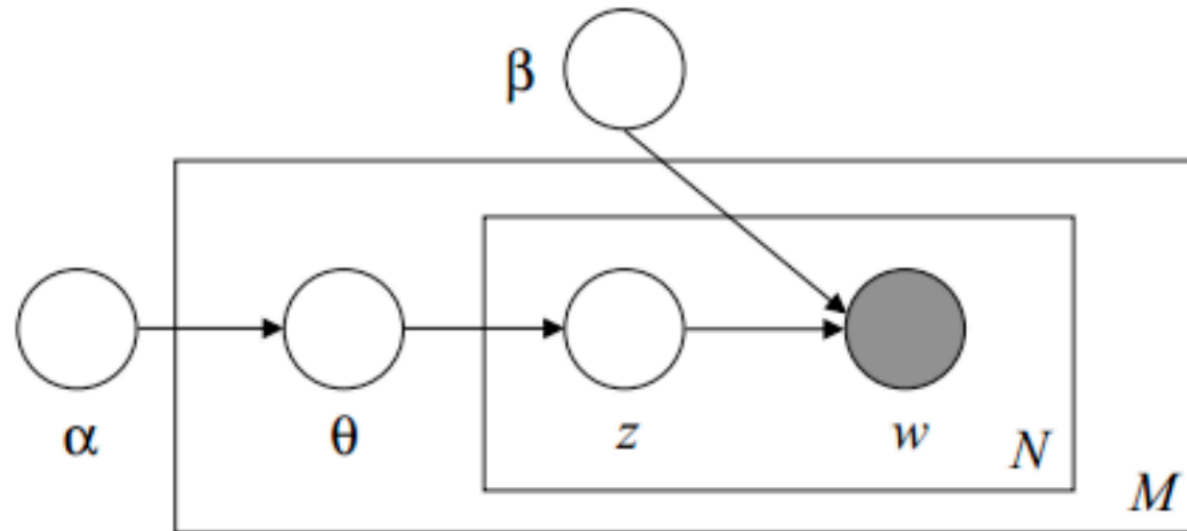
- Plate notation



*Plates denote replication of random variables*

# Hierarchical Bayesian Models

- Why stop with a single prior?

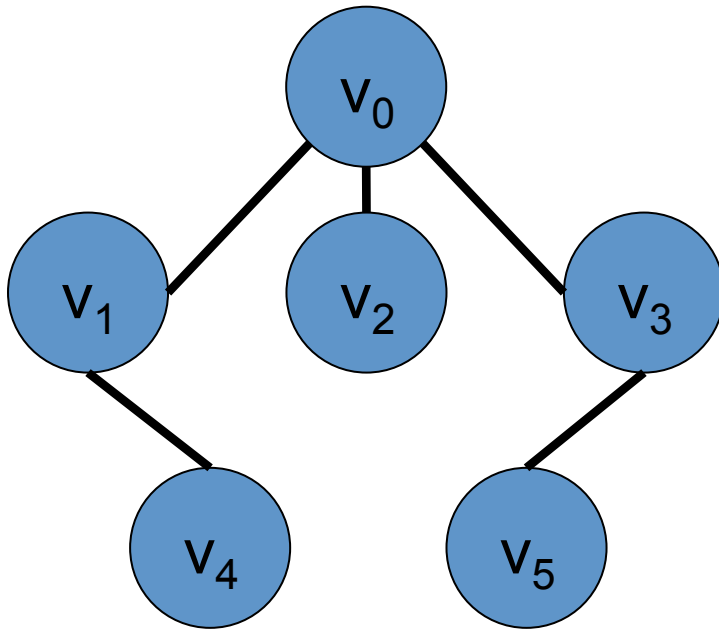


Graphical model representation of LDA. The boxes are “plates” representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

# Summary: Learning BN

- MLE
  - Decomposes; results in counting procedure
- Bayesian estimation
  - Priors = regularization (smoothing)
  - Hierarchical priors
- Plate notation
- Shared parameters

# Known Tree Structure



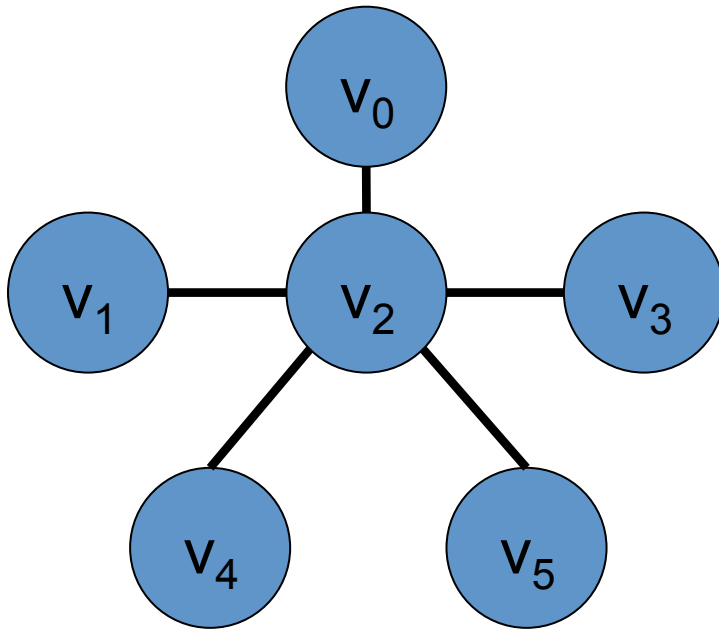
Distribution  $P_T(x)$

$v_{p(a)}$  = “parent” of  $v_a$

$$P_T(x_5|x_3)P_T(x_4|x_1)P_T(x_3|x_0)P_T(x_2|x_0)P_T(x_1|x_0)P_T(x_0)$$

$$\text{Estimate } P_T(x_a|x_{p(a)}) = P(x_a|x_{p(a)})$$

# Known Tree Structure



Distribution  $P_T(x)$

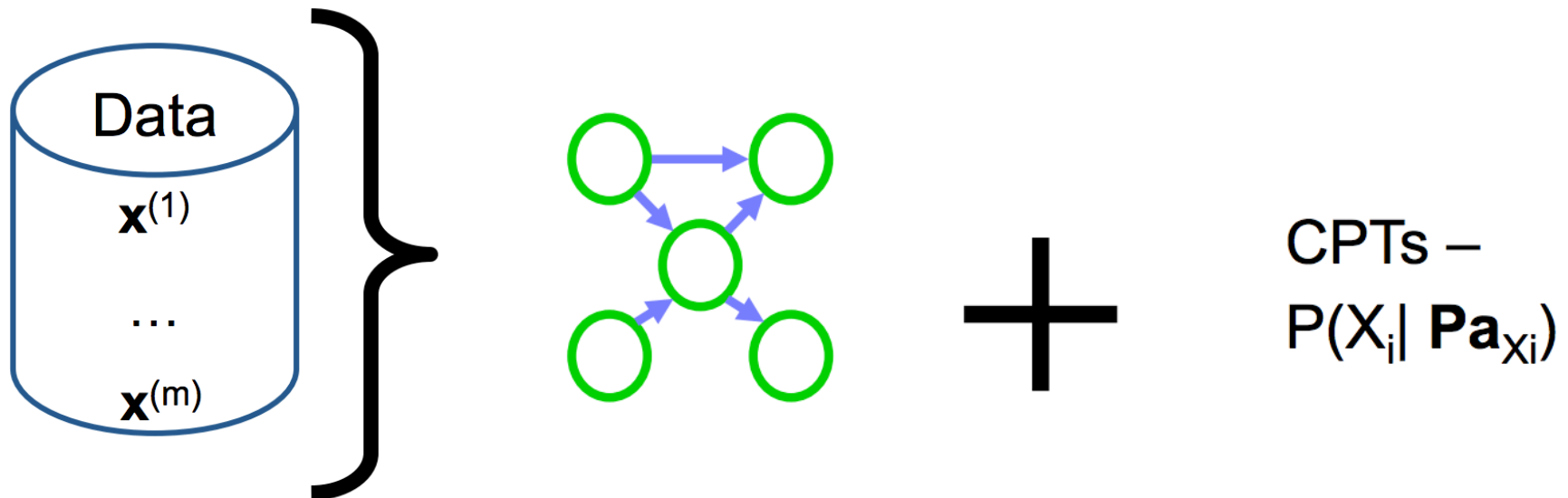
$v_{p(a)}$  = “parent” of  $v_a$

$$P_T(x_5|x_2)P_T(x_4|x_2)P_T(x_3|x_2)P_T(x_2|x_0)P_T(x_1|x_2)P_T(x_0)$$

Estimate  $P_T(x_a|x_{p(a)}) = P(x_a|x_{p(a)})$  **Which tree?**

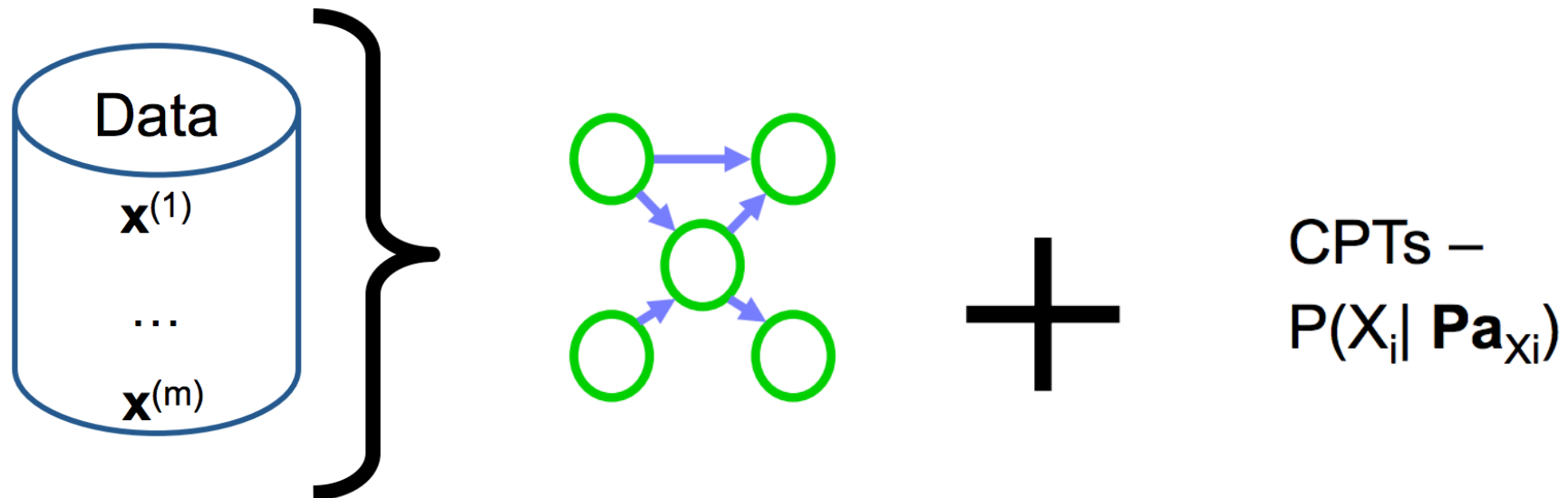
# Learning Bayesian Nets

	Known structure	Unknown structure
Fully observable data	Very easy	Hard
Missing data	Somewhat easy (EM)	Very very hard



# Learning Bayesian Nets: Structure

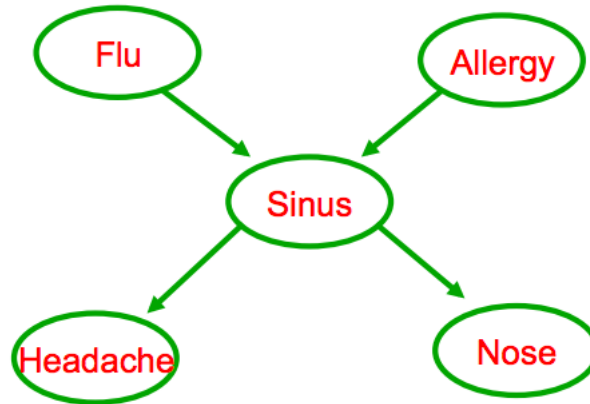
- Prediction: Care about a good structure => good prediction
- Discovery: Understand some system



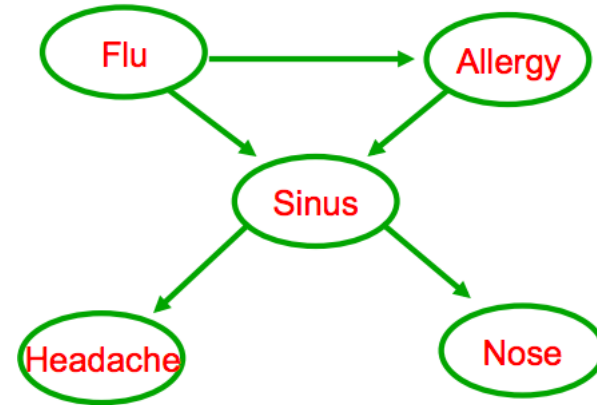
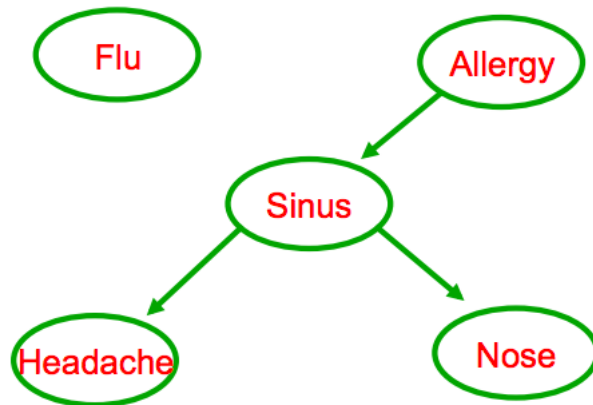


# Learning Bayesian Nets: Structure

- Truth



- Recovered



# Learning Bayesian Nets: Structure

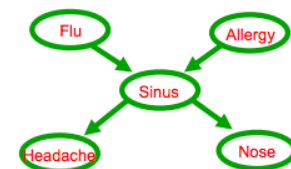
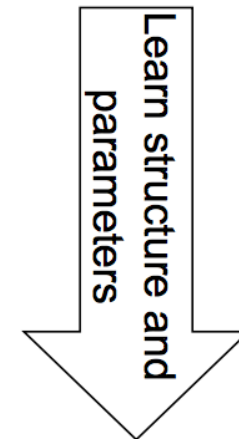
- Constraint-based approach
  - Test conditional independencies in data
  - Find an I-map
- Score-based approach
  - Finding structure and parameters => density estimation task
  - Evaluate model, similar to parameter estimation
    - MLE
    - Bayesian estimation



$\langle x_1^{(1)}, \dots, x_n^{(1)} \rangle$

...

$\langle x_1^{(m)}, \dots, x_n^{(m)} \rangle$



Slide courtesy: Dhruv Batra

# Score-based Approach

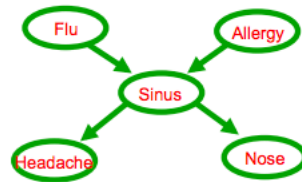


$\langle x_1^{(1)}, \dots, x_n^{(1)} \rangle$

...

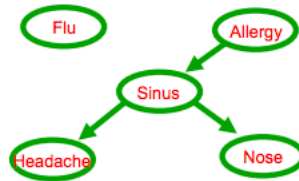
$\langle x_1^{(m)}, \dots, x_n^{(m)} \rangle$

## Possible structures



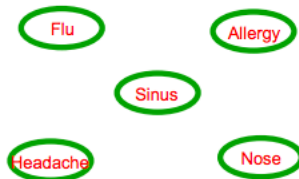
Learn parameters

**Score structure  
-52**



Learn parameters

**Score structure  
-60**



Learn parameters

**Score structure  
-500**

# Score-based Approach

- Say there are  $N$  vertices?
- How many (undirected) graphs in the search space?
- How many (undirected) trees?

# Score-based Approach

- What is a good score?
- How about log-likelihood?
  - $\text{Score}(G) = \log\text{-likelihood}(G: D, \theta_{\text{MLE}}) = \log P(D | G, \theta_{\text{MLE}})$
- How do we interpret this Max Likelihood score?
  - Consider a two-node graph (on board)

# Kullback-Leibler Divergence

$$KL(P_1 || P_2) = - \sum_x P_1(x) \log P_2(x) + \sum_x P_1(x) \log P_1(x)$$

Constant

$$KL(P_1 || P_2) \geq 0$$

$$KL(P_1 || P_1) = 0$$

Substitute  $P_1 = P$  and  $P_2 = P_T$ . Minimize  $KL(P || P_T)$

# Estimating the Tree Structure

$$\min - \sum_x P(x) \log P_T(x)$$

# Estimating the Tree Structure

$$\min - \sum_x P(x) \log \prod_a P_T(x_a | x_{p(a)})$$



# Estimating the Tree Structure

$$\min - \sum_x P(x) \sum_a \log P_T(x_a | x_{p(a)})$$

# Estimating the Tree Structure

$$\min - \sum_x P(x) \sum_a \log \frac{P_T(x_a, x_{p(a)})}{P_T(x_{p(a)})}$$

# Estimating the Tree Structure

$$\min - \sum_x P(x) \sum_a \log \frac{P_T(x_a, x_{p(a)}) P(x_a)}{P_T(x_{p(a)}) P(x_a)}$$

# Estimating the Tree Structure

$$\min - \sum_x P(x) \sum_a \log \frac{P_T(x_a, x_{p(a)})}{P_T(x_{p(a)})P(x_a)}$$

$$- \sum_x P(x) \sum_a \log P(x_a)$$

Independent of the tree structure

# Estimating the Tree Structure

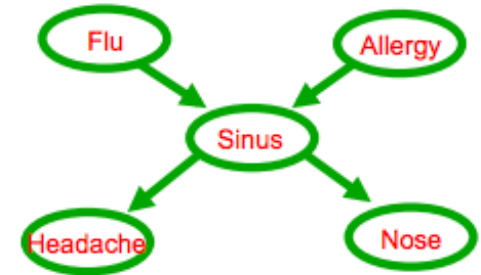
$$\min - \sum_a \sum_{x_a} \sum_{x_{p(a)}} P(x_a, x_{p(a)}) \log \frac{P_T(x_a, x_{p(a)})}{P_T(x_{p(a)})P(x_a)}$$

$$\min - \sum_a I(x_a, x_{p(a)})$$

Mutual Information

# Score-based Approach

- For a general graph  $G$ ,



$$\log \hat{P}(\mathcal{D} | \theta, \mathcal{G}) = m \sum_i \sum_{x_i, \mathbf{Pa}_{x_i, \mathcal{G}}} \hat{P}(x_i, \mathbf{Pa}_{x_i, \mathcal{G}}) \log \hat{P}(x_i | \mathbf{Pa}_{x_i, \mathcal{G}})$$

$$\log \hat{P}(\mathcal{D} | \theta, \mathcal{G}) = m \sum_i \hat{I}(X_i, \mathbf{Pa}_{X_i}) - m \sum_i \hat{H}(X_i)$$

# Score-based Approach

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \hat{I}(X_i, \mathbf{Pa}_{X_i}) - m \sum_i \hat{H}(X_i)$$

- Implications
  - Intuitive: higher mutual info  $\rightarrow$  higher score
  - Decomposes over families (nodes and its parents)
  - Information never hurts!
  - But....

# Score-based Approach

- Adding an edge only improves score!
  - Thus, MLE = complete graph
- Two fixes
  - Restrict space of graphs
    - Say only  $d$  parents allowed
  - Put priors on graphs
    - Prefer sparser graphs



# Chow-Liu Tree Learning - I

- For each pair of variables  $X_i, X_j$ 
  - Compute the empirical distribution

$$\hat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$$

- Compute mutual information

$$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{P}(x_i, x_j) \log \frac{\hat{P}(x_i, x_j)}{\hat{P}(x_i) \hat{P}(x_j)}$$

- Define graph
  - Nodes  $X_1, X_2, \dots, X_n$
  - Edge (i,j) gets weight  $\hat{I}(X_i, X_j)$

# Chow-Liu Tree Learning - II

- Optimal tree BN
  - Compute maximum weight spanning tree
  - Directions:
    - Pick any node as root
    - Direct edges from root (breadth-first search for example)

# Score-based Approach

- Bayesian score
  - => Prior distributions
    - Over structures
    - Over parameters of a structure
- Posterior over structures (given data)

$$\log P(\mathcal{G} | D) \propto \log P(\mathcal{G}) + \log \int_{\theta_{\mathcal{G}}} P(D | \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$

# Bayesian Score: Structure Prior

$$\log P(\mathcal{G} | D) \propto \log P(\mathcal{G}) + \log \int_{\theta_{\mathcal{G}}} P(D | \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$

- Common choices
  - Uniform:  $P(\mathcal{G}) \propto c$
  - Sparsity prior:  $P(\mathcal{G}) \propto c^{|\mathcal{G}|}$
  - Prior penalizing number of parameters
  - $P(\mathcal{G})$  should decompose like the family score

# Bayesian Score: Parameter Prior & Integrals

$$\log P(\mathcal{G} | D) \propto \log P(\mathcal{G}) + \log \int_{\theta_{\mathcal{G}}} P(D | \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$

- If  $P(\theta_{\mathcal{G}} | \mathcal{G})$  is Dirichlet, then the integral has closed form!
- And, it factorizes according to families in  $\mathcal{G}$

# Bayesian Score: Parameter Prior & Integrals

$$\log P(\mathcal{G} | D) \propto \log P(\mathcal{G}) + \log \int_{\theta_{\mathcal{G}}} P(D | \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$

- How should we choose Dirichlet hyperparameters?
  - K2 prior: Fix an  $\alpha$ ,  $P(\theta_{X_i | \text{Pa}_{X_i}}) = \text{Dirichlet}(\alpha, \dots, \alpha)$
  - BDe Prior: Pick a “prior” BN
  - Compute  $P(X_i, \text{Pa}_{X_i})$  under this prior BN

# Learning Bayesian Nets: Structure

- **Question:** Are these score-based approaches really Bayesian?
- So far, we have selected only one structure
- We must average over structures
  - Similar to averaging over parameters

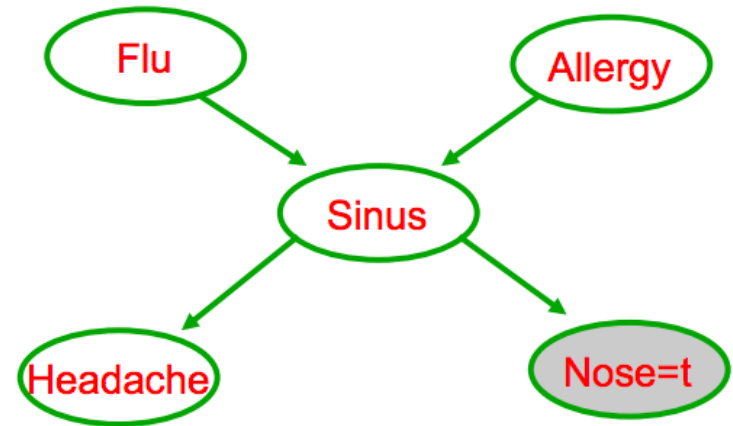
# This class

- Bayesian Networks
  - Parameter Learning
  - Structure Learning
  - **Inference**



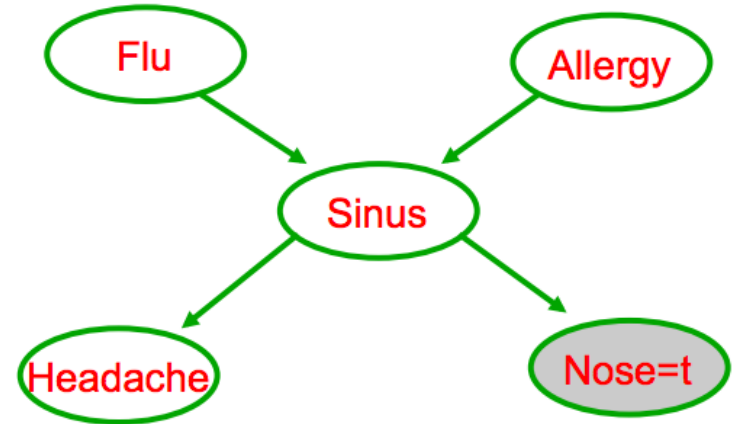
# BNs: Inference

- Evidence  $\mathbf{E}=\mathbf{e}$  (e.g.,  $N=t$ )
- Query variables of interest  $Y$
- Conditional probability:  $P(Y \mid \mathbf{E}=\mathbf{e})$ 
  - e.g.,  $P(F,A \mid N=t)$
  - Special case: Marginals  $P(F)$
- Maximum a posteriori:  $\operatorname{argmax} P(\text{all var} \mid \mathbf{E}=\mathbf{e})$ 
  - $\operatorname{argmax}_{\{f,a,s,h\}} P(f,a,s,h \mid N=t)$



# BNs: Inference

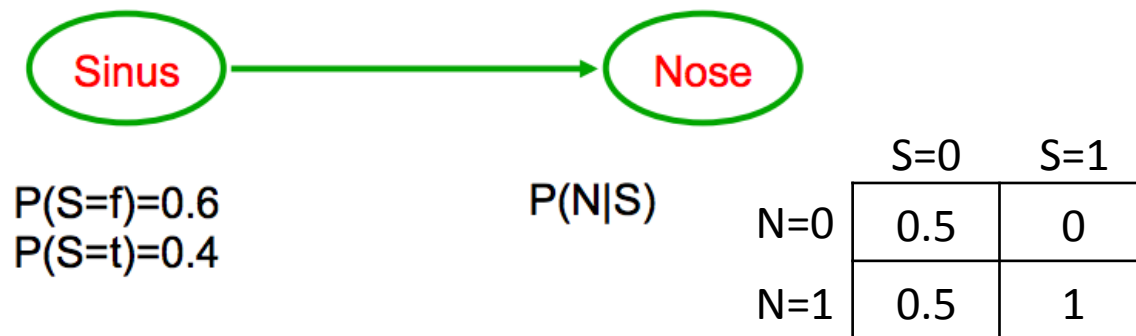
- Evidence  $\mathbf{E}=\mathbf{e}$  (e.g.,  $N=t$ )
- Query variables of interest  $Y$



- Marginal-MAP:  $\operatorname{argmax}_y P(Y=y \mid \mathbf{E}=\mathbf{e})$ 
  - $\operatorname{argmax}_y \sum_{\mathbf{o}} P(Y=y, \mathbf{O}=\mathbf{o} \mid \mathbf{E}=\mathbf{e})$

# BNs: Inference

- Are MAP and max of marginals consistent?
- Verify with this example:



# BNs: Inference

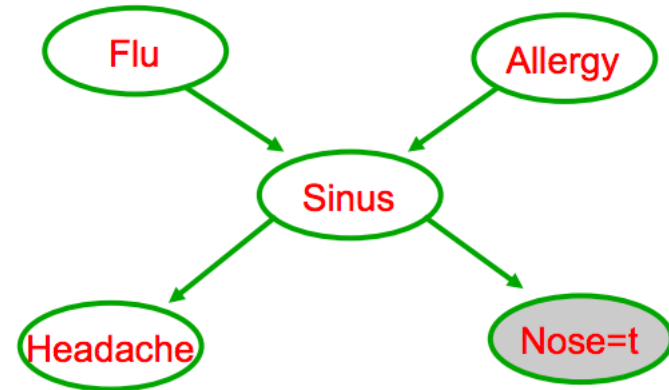
- In general, (at least) NP-hard
- In practice,
  - Exploit structure
  - Many effective approximate algorithms
- We will look at
  - Exact and approximate inference

# BNs: Inference

- Variable Elimination
- Sum-product belief propagation
- Sampling: MCMC
  
- Integer programming (LP relaxation)
- Combinatorial optimization (e.g., graphcuts)

# Marginal Inference

- Consider the example
  - Evidence:  $N=t$
  - Compute:  $P(F \mid N=t)$



- (On board, if time permits)

# Variable Elimination

- Given a BN and a query  $P(\mathbf{Y}|\mathbf{e}) \approx P(\mathbf{Y},\mathbf{e})$ , IMPORTANT!!!
- Choose an ordering on variables, e.g.,  $X_1, \dots, X_n$

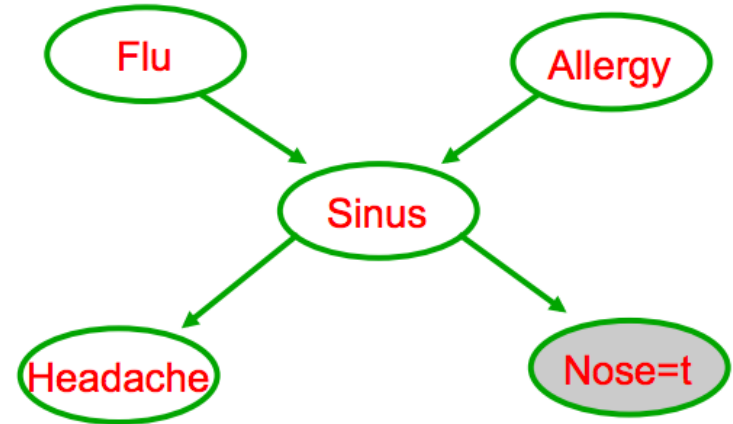
- For  $i=1 \dots n$ , if  $X_i \notin \{\mathbf{Y}, \mathbf{E}\}$ 
  - Collect factors  $f_1 \dots f_k$  that include  $X_i$
  - Generate a new factor by eliminating  $X_i$  from them

$$g = \sum_{X_i} \prod_{j=1}^k f_j$$

- Normalize  $P(\mathbf{Y},\mathbf{e})$  to obtain  $P(\mathbf{Y}|\mathbf{e})$

# MAP Inference

- Evidence  $\mathbf{E}=\mathbf{e}$  (e.g.,  $N=t$ )
- Query variables of interest  $Y$



- Maximum a posteriori:  $\operatorname{argmax} P(\text{all var} \mid \mathbf{E}=\mathbf{e})$ 
  - $\operatorname{argmax}_{\{f,a,s,h\}} P(f,a,s,h \mid N=t)$



# Variable Elimination for MAP Inference

- Given a BN and a query  $\max_{x_1 \dots x_n} P(x_1 \dots x_n, \mathbf{e})$ ,
- Choose an ordering on variables, e.g.,  $X_1, \dots, X_n$
- For  $i=1 \dots n$ , if  $X_i \notin \mathbf{E}$ 
  - Collect factors  $f_1 \dots f_k$  that include  $X_i$
  - Generate a new factor by eliminating  $X_i$  from them

$$g = \max_{x_i} \prod_{j=1}^k f_j$$

- (This completes the forward pass)

# Variable Elimination for MAP Inference

- $\{x_1^* \dots x_n^*\}$  will store the maximizing assignment
- For  $i=n \dots 1$ , if  $X_i \notin \mathbf{E}$ 
  - Take factors  $f_1 \dots f_k$  used when  $X_i$  was eliminated
  - Instantiate  $f_1 \dots f_k$  with  $\{x_{i+1}^* \dots x_n^*\}$
  - Generate maximizing assignment for  $X_i$ :

$$x_i^* \in \operatorname{argmax}_{x_i} \prod_{j=1}^k f_j$$

- (This completes the backward pass)