Graphical Models
Inference and Learning
Lecture 6

MVA

2021 – 2022

http://thoth.inrialpes.fr/~alahari/disinflearn

Outline

- Preliminaries
- Maximum Flow
- Algorithms
- **Energy minimization with max flow/min cut**
  - Multi-Label Energy Functions

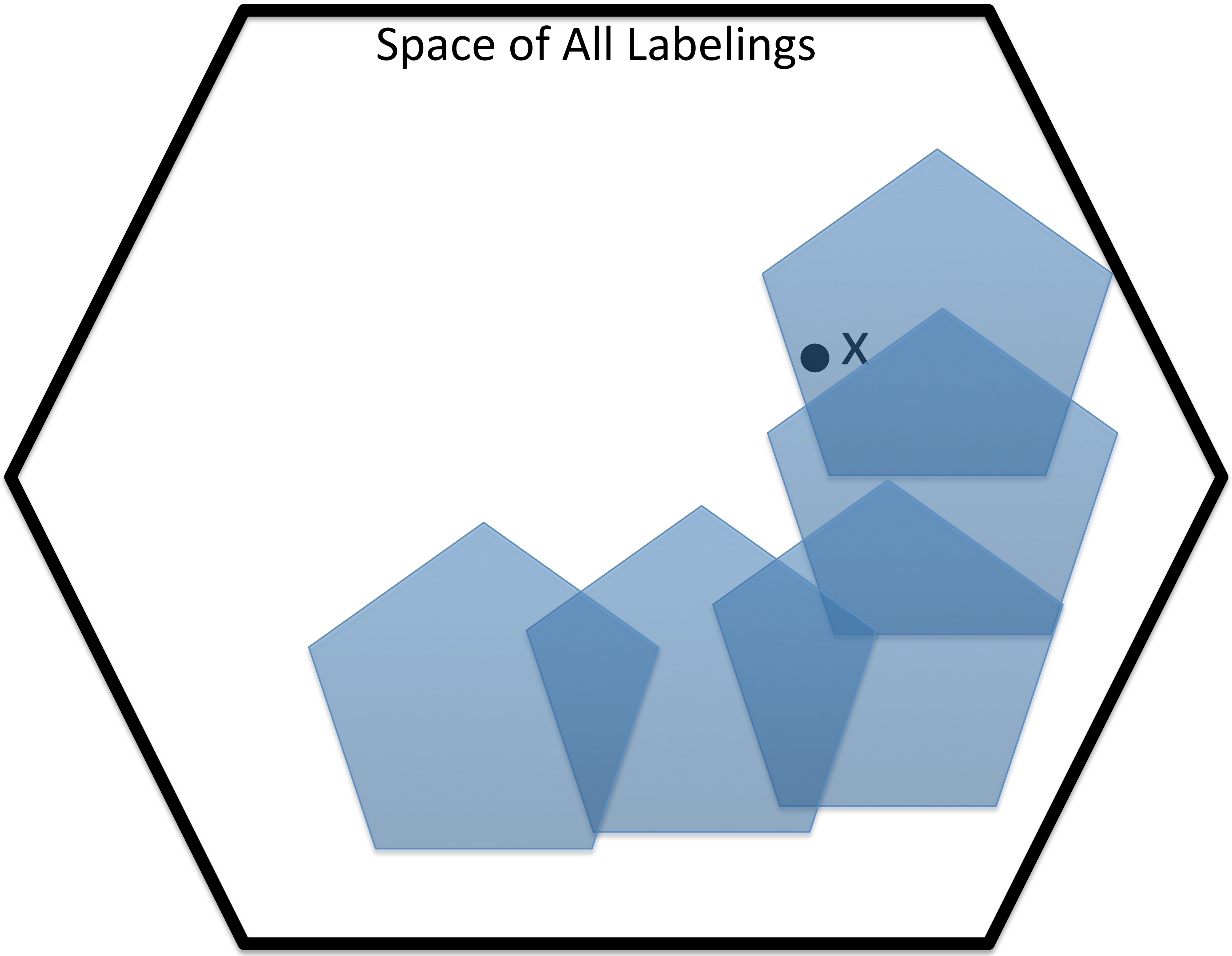
St-mincut based Move algorithms

$$E(\mathbf{x}) = \sum_i \theta_i(x_i) + \sum_{i,j} \theta_{ij}(x_i, x_j)$$

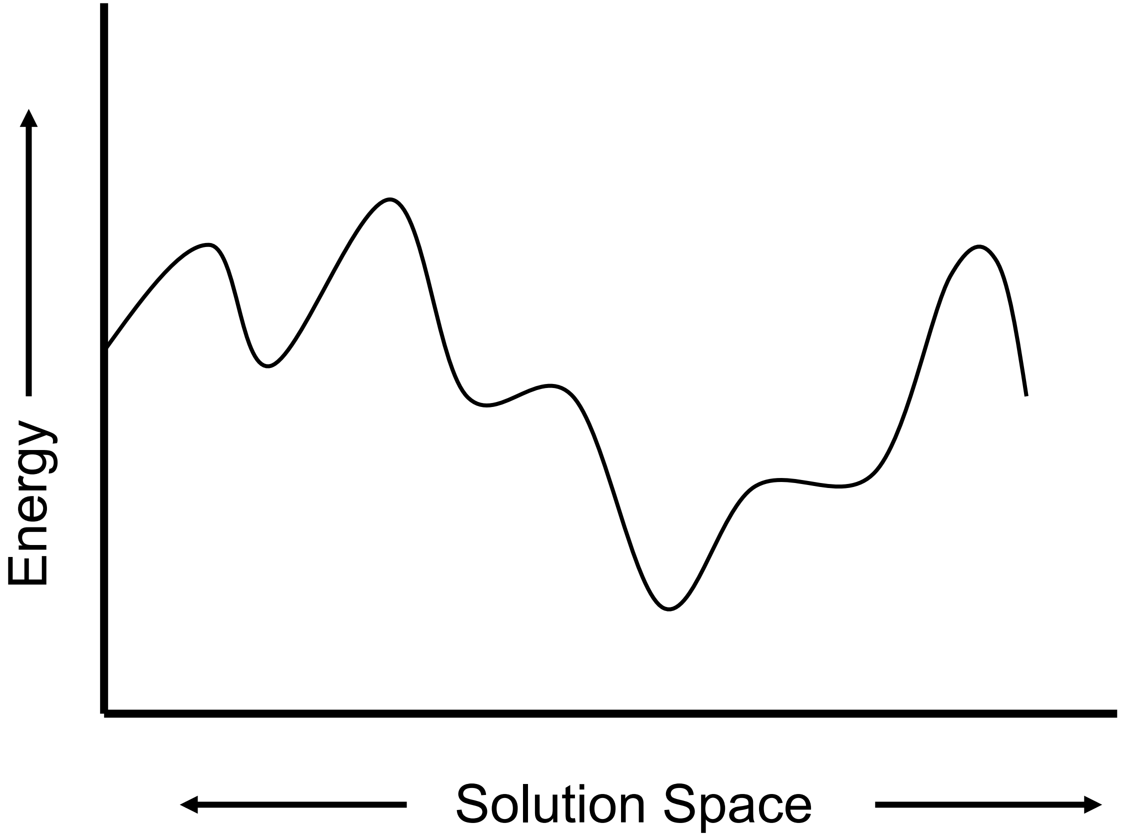
$x \in$ Labels $L = \{l_1, l_2, \ldots, l_k\}$

- Commonly used for solving **non-submodular** multi-label problems
- Extremely efficient and produce good solutions
- Not Exact: Produce local optima

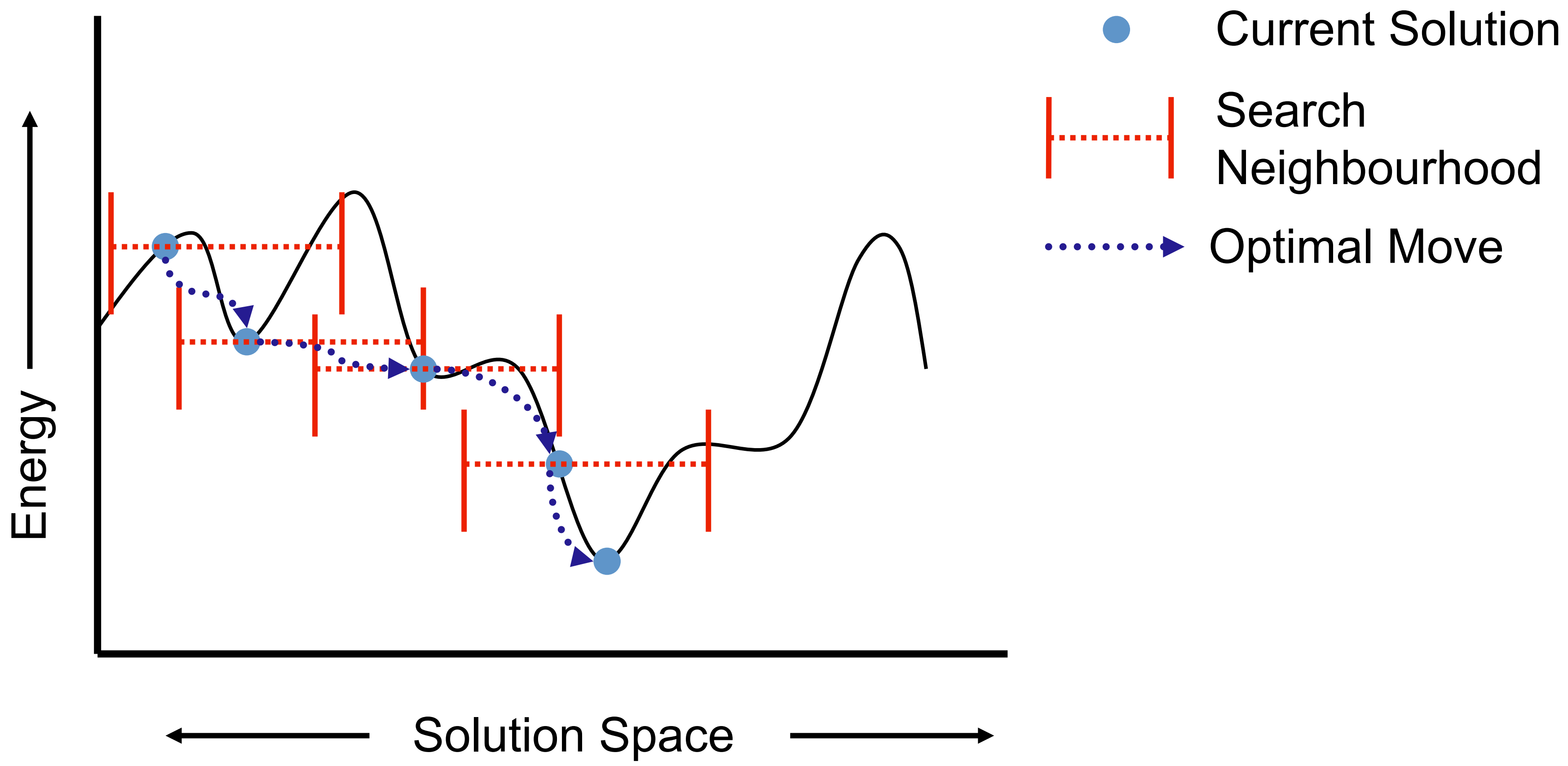
Move-Making Algorithms



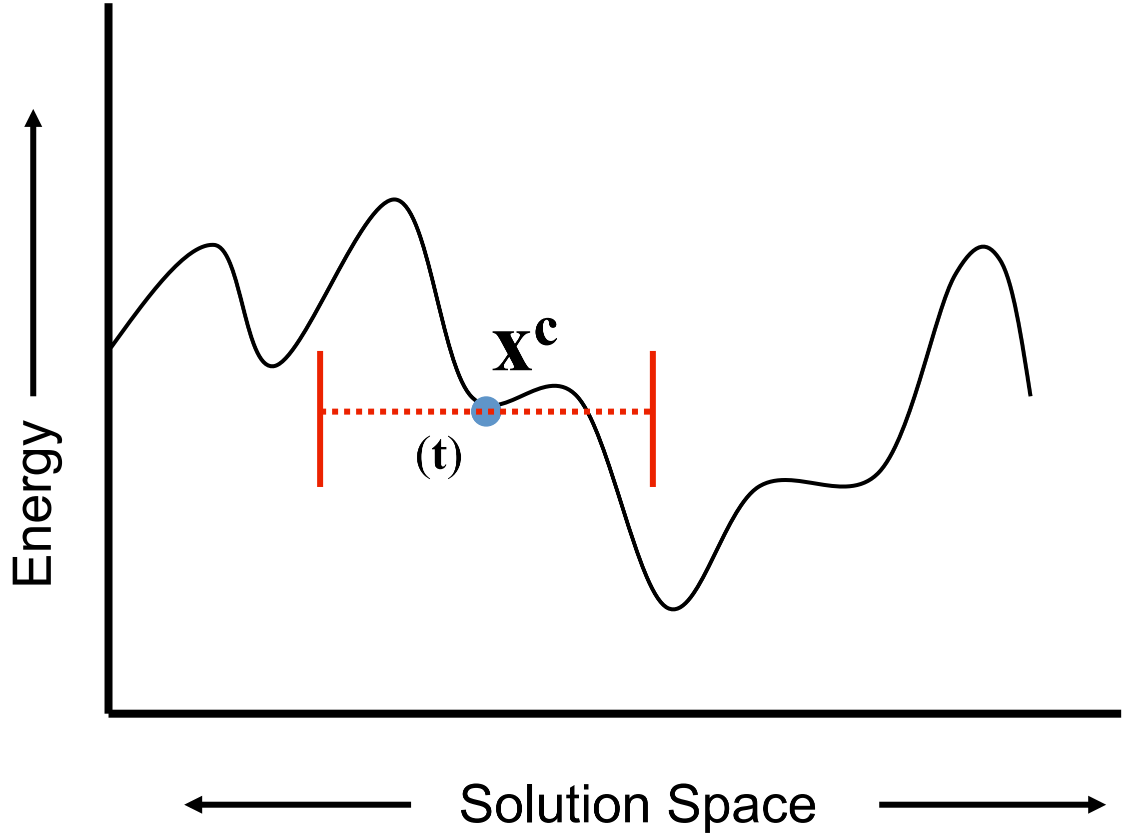
Move Making Algorithms



Move Making Algorithms



Computing the Optimal Move



Current Solution

Search Neighbourhood

Optimal Move

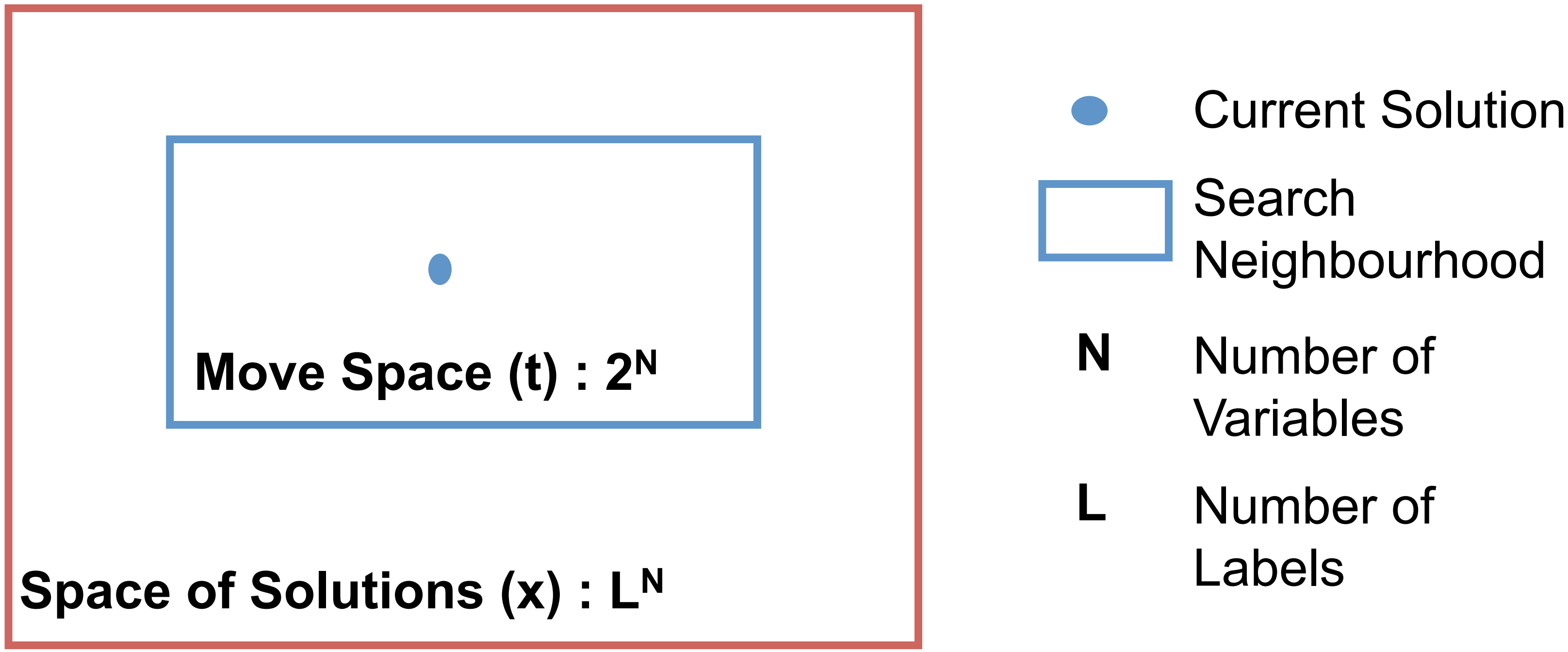
Key Property
Move Space

Bigger move space →
- **Better solutions**
- **Finding the optimal move hard**

Moves using Graph Cuts

Expansion and Swap move algorithms
[Boykov Veksler and Zabih, PAMI 2001]

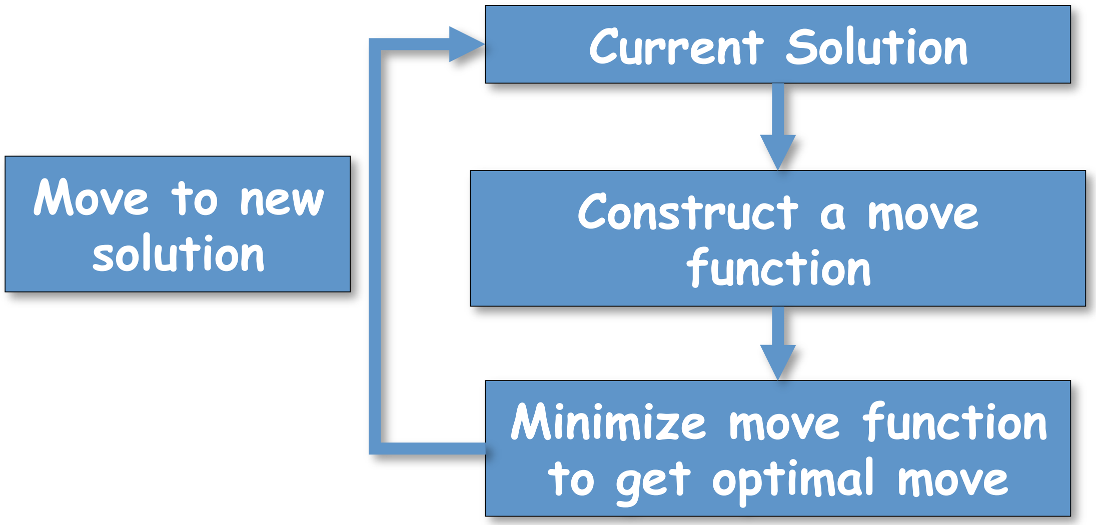
- **Makes a series of changes to the solution (moves)**
- **Each move results in a solution with smaller energy**



Moves using Graph Cuts

Expansion and Swap move algorithms
[Boykov Veksler and Zabih, PAMI 2001]

- **Makes a series of changes to the solution (moves)**
- **Each move results in a solution with smaller energy**



How to minimize move functions?

Expansion Move

- **Variables take label α or retain current label**

[Boykov, Veksler, Zabih]

Expansion Move

- **Variables take label *α* or retain current label**

Tree
Ground
House
Sky

Status: **Initialize with Tree**

[Boykov, Veksler, Zabih]

Expansion Move

- **Variables take label *α* or retain current label**

Tree
Ground
House
Sky

Status: Expand Ground

[Boykov, Veksler, Zabih]

Expansion Move

- **Variables take label *α* or retain current label**

Tree
Ground
House
Sky

Status: Expand House

[Boykov, Veksler, Zabih]

Expansion Move

- **Variables take label *α* or retain current label**

Tree
Ground
House
Sky

Status: **Expand Sky**

[Boykov, Veksler, Zabih]

Expansion Algorithm

Initialize labeling $\mathbf{x} = \mathbf{x}^0$ (say $x^0{}_p = 0$, for all X_p)

For $\alpha = 1, 2, \ldots, h\text{-}1$

$\mathbf{x}^\alpha = \text{argmin}_{\mathbf{x}'} E(\mathbf{x}')$

s.t. $x'_p \in \{x_p\} \cup \{l_\alpha\}$

Update $\mathbf{x} = \mathbf{x}^\alpha$

End

Repeat
until
convergence

Boykov, Veksler and Zabih, 2001

Expansion Move

- **Variables take label *α* or retain current label**

- **Move energy is submodular if:**
  - **Unary Potentials: Arbitrary**
  - **Pairwise potentials: Metric**

$$\theta_{ij}(l_a, l_b) \geq 0$$

$$\theta_{ij}(l_a, l_b) = 0 \quad \text{iff} \quad a = b$$

Semi metric

Examples: Potts model, Truncated linear

Cannot solve truncated quadratic

[Boykov, Veksler, Zabih]

Expansion Move

- Variables take label *α* or retain current label

- **Move energy is submodular if:**
  - **Unary Potentials: Arbitrary**
  - **Pairwise potentials: Metric**

$$\theta_{ij}(l_a,l_b) + \theta_{ij}(l_b,l_c) \geq \theta_{ij}(l_a,l_c)$$

Triangle Inequality

Examples: Potts model, Truncated linear

Cannot solve truncated quadratic

[Boykov, Veksler, Zabih]

Swap Move

- **Variables labeled *α*, *β* can swap their labels**

[Boykov, Veksler, Zabih]

Swap Move

- **Variables labeled α, β can swap their labels**

Swap Sky, House

Tree
Ground
House
Sky

[Boykov, Veksler, Zabih]

Swap Move

- **Variables labeled α, β can swap their labels**

Tree
Ground
House
Sky

Swap Sky, House

[Boykov, Veksler, Zabih]

Swap Move

- **Variables labeled α, β can swap their labels**

- **Move energy is submodular if:**
  - **Unary Potentials: Arbitrary**
  - **Pairwise potentials: Semimetric**

$$\theta_{ij}(l_a, l_b) \geq 0$$

$$\theta_{ij}(l_a, l_b) = 0 \longleftrightarrow a = b$$

Examples: Potts model, Truncated Convex

[Boykov, Veksler, Zabih]

General Binary Moves

$$x = t\, x^1 + (1-t)\, x^2$$

New solution — **First solution** — **Second solution**

Minimize over move variables t

Move Type	First Solution	Second Solution	Guarantee
Expansion	Old solution	All alpha	Metric
Fusion	Any solution	Any solution	✘

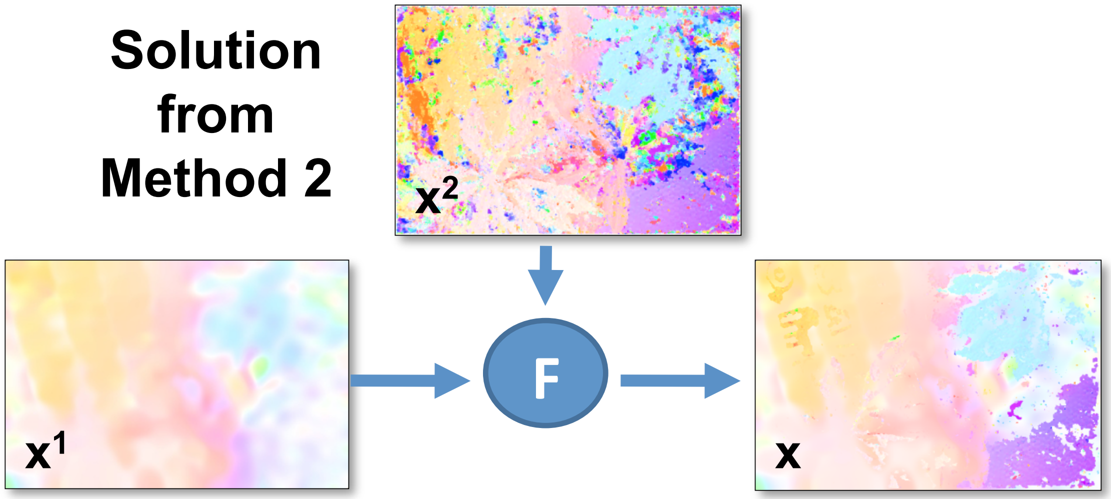
Solving Continuous Problems using Fusion Move

$$\mathbf{x} = t\,\mathbf{x}^1 + (1-t)\,\mathbf{x}^2$$

Optical Flow Example

Solution from Method 2

Solution from Method 1

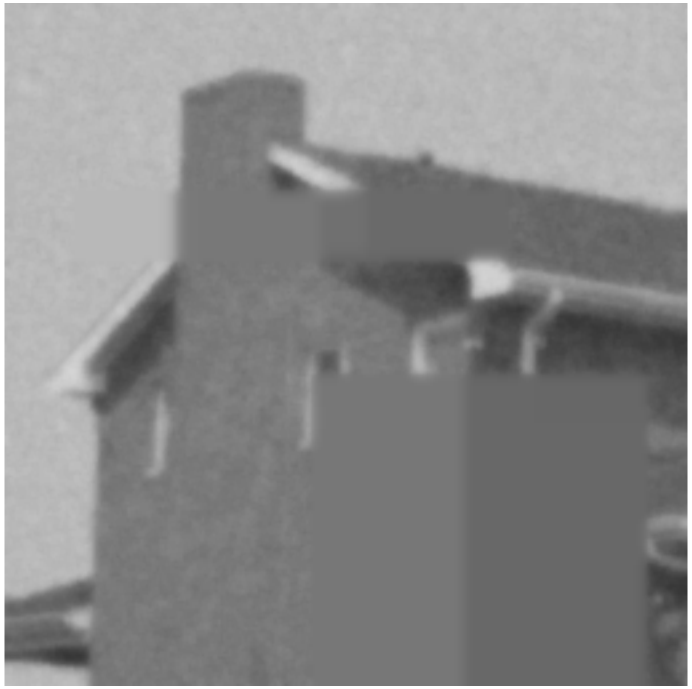


Final Solution

(Lempitsky et al. CVPR08, Woodford et al. CVPR08)

Results – Denoising + Inpainting

Results – Denoising + Inpainting



Results – Denoising + Inpainting

Results – Denoising + Inpainting

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

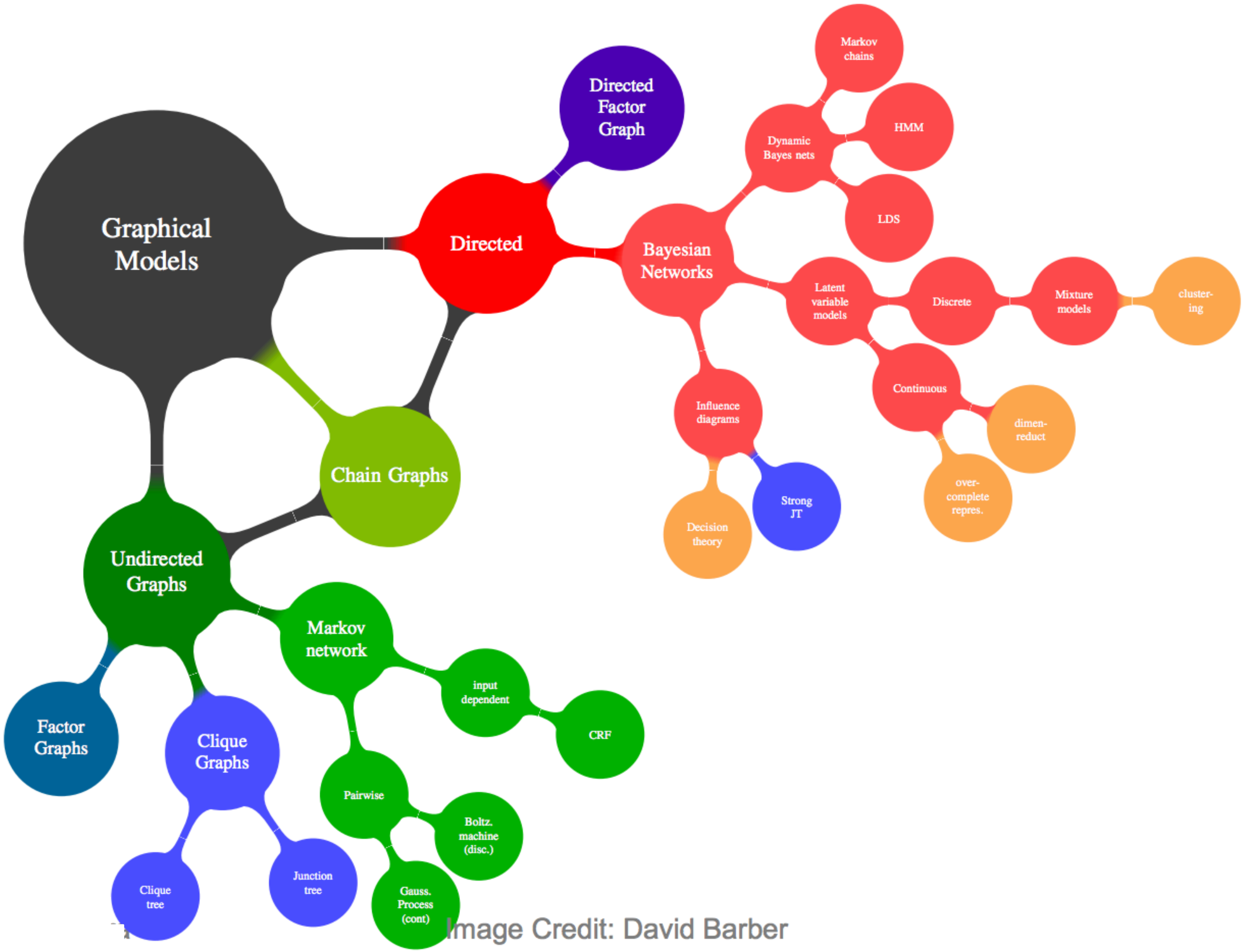


Image Credit: David Barber

Remainder of this class

- Bayesian Networks
  - Parameter Learning
  - Structure Learning
  - Inference

But first…

A quiz !

1. How would you parameterize

$$\mathrm{MRF}_G(\mathbf{x}; \mathbf{u}^k, \mathbf{h}^k) = \sum_p u_p^k(x_p) + \sum_c h_c^k(\mathbf{x}_c)$$

   for learning?

2. How are max-flow and min-cut related?

3. What are two methods for performing max-flow and their drawbacks?

Remainder of this class

- Bayesian Networks
  - Parameter Learning
  - Structure Learning
  - Inference

Bayesian Networks

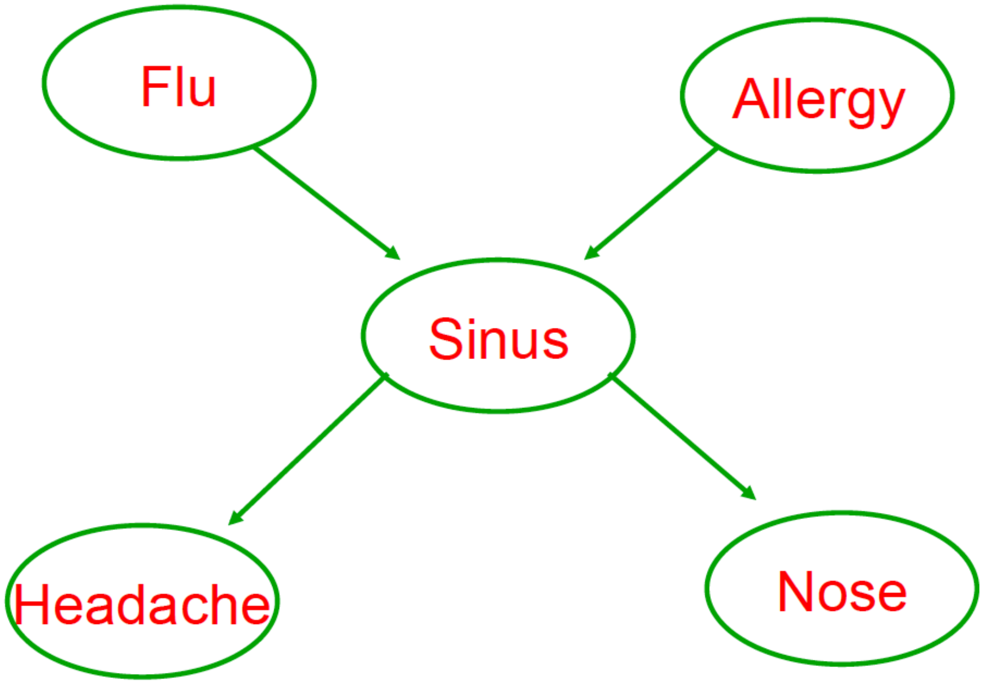
- A general Bayes net
  - Set of random variables
  - DAG: encodes independence assumptions
  - Conditional probability trees
  - Joint distribution

$$P(Y_1,\ldots,Y_n) = \prod_{i=1}^{n} P(Y_i \mid \mathrm{Pa}_{Y_i})$$

Slide courtesy: Dhruv Batra

Bayesian Networks

- Example



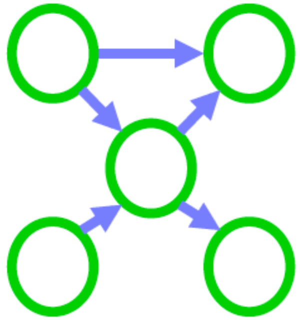
Slide courtesy: Dhruv Batra

Independencies in problem

World, Data, reality:

True distribution *P* contains independence assertions

BN:



Graph *G* encodes local independence assumptions

Slide courtesy: Carlos Guestrin, Dhruv Batra

Learning Bayesian Nets

True Distribution P*
(Maybe corresponds to a BN G*
maybe not)

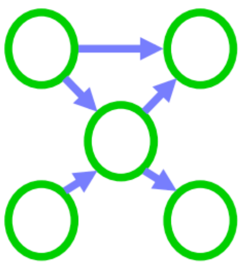
Domain Experts

Data

$\mathbf{x}^{(1)}$

…

$\mathbf{x}^{(m)}$



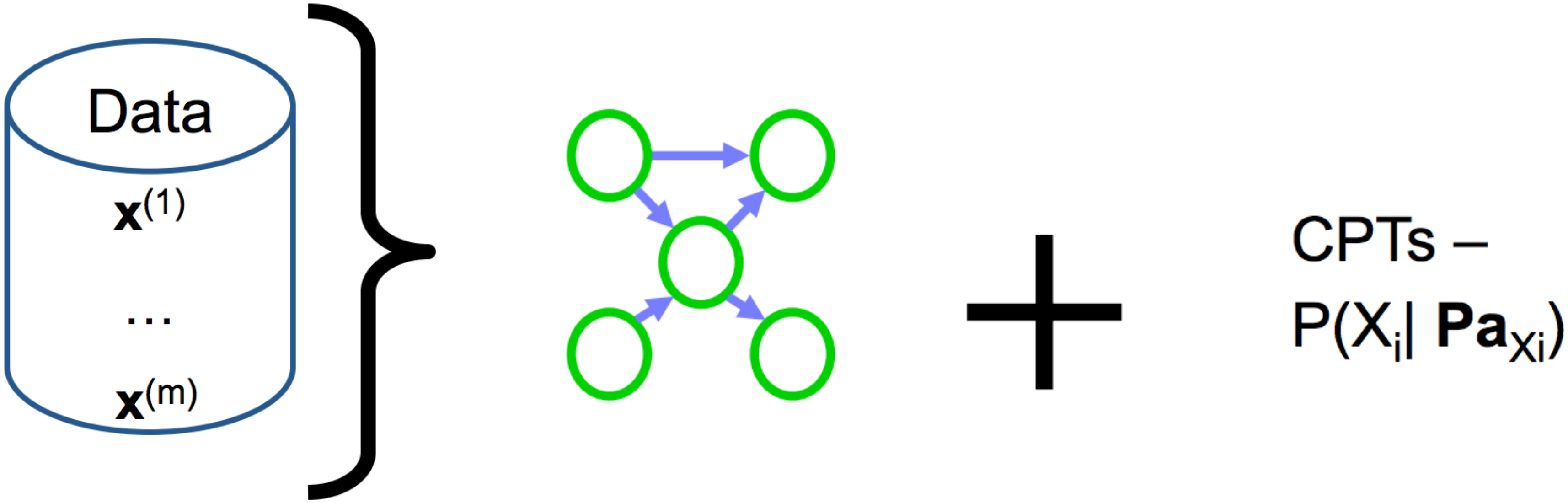
+

CPTs –
$P(X_i | \mathbf{Pa}_{X_i})$

Slide courtesy: Dhruv Batra

Learning Bayesian Nets

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



Slide credit: Carlos Guestrin, Dhruv Batra

Maximum Likelihood Estimation

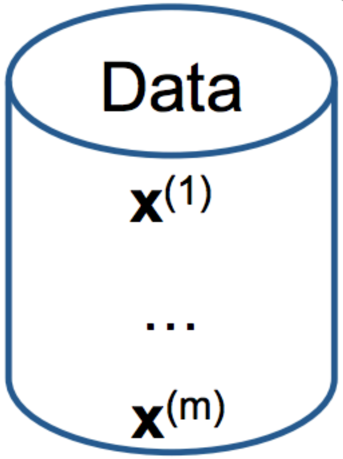
- Goal: Find a good θ

- What is a good θ?
  - One that makes it likely for us to have seen this data
  - Quality of θ = Likelihood(θ; D) = $P(D|\theta)$

- Why MLE?
  - Log-likelihood(θ) = entropy(P^*) – KL(P^*, $P(D|\theta)$)
  - i.e., maximizing LL = minimizing KL

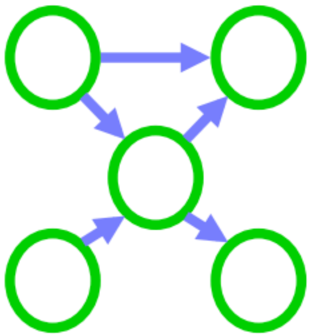
Slide courtesy: Dhruv Batra

MLE: Learning the CPTs



For each discrete variable X_i

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



Slide credit: Carlos Guestrin, Dhruv Batra

Bayesian Estimation

- Exploit priors
  - Priors: Beliefs before experiments are conducted
  - Help deal with unseen data
  - Bias us towards "simpler" models

- Beta prior distribution

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

constant

Slide courtesy: Dhruv Batra

Bayesian Estimation

- Posterior

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

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

$$P(\theta \mid \mathcal{D}) \;\; \propto \;\; P(\mathcal{D} \mid \theta)P(\theta)$$

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

Slide courtesy: Dhruv Batra

Bayesian Estimation

- MAP: use most likely parameter

$$\widehat{\theta} = \arg\max_{\theta} P(\theta \mid \mathcal{D})$$

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

- Beta prior equiv. to extra H/T
- As m → inf, prior is "forgotten"
- But, for small sample size, prior is important !

Slide courtesy: Dhruv Batra

Bayesian Estimation

- What about the multinomial case?

- Use a Dirichlet for the prior

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

Meta BN: Bayesian view of BN

- Show parameters explicitly as variables

- Two examples (on board)

Slide courtesy: Dhruv Batra

Global parameter independence

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

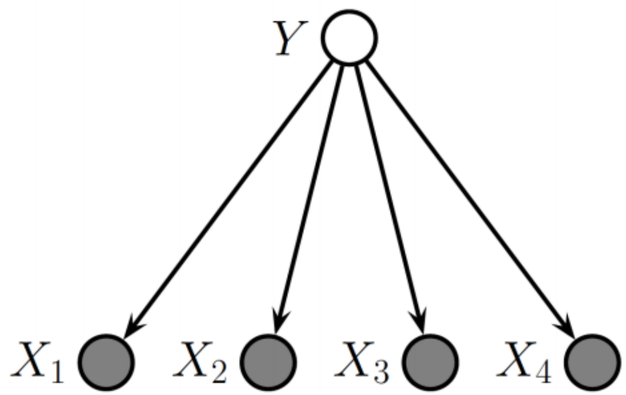
$$P(\theta \mid \mathcal{D}) = \prod_i P(\theta_{X_i|\mathbf{Pa}_{X_i}} \mid \mathcal{D})$$

Slide courtesy: Dhruv Batra

Parameter Sharing

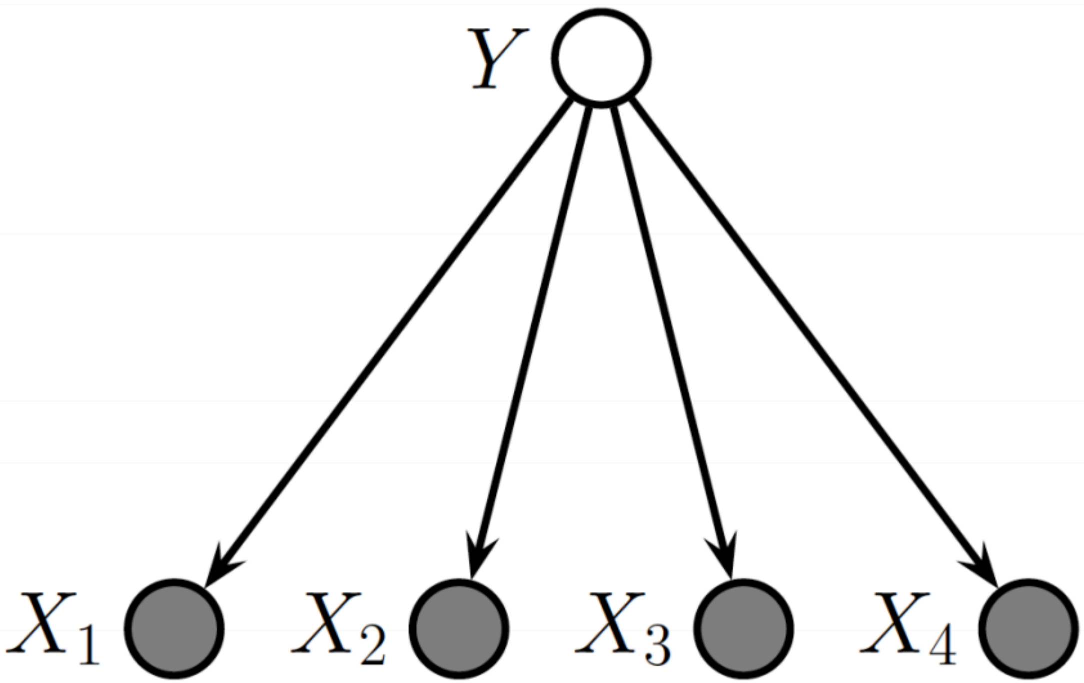
- Consider the scenario, where n random variables X_1, X_2, … X_n represent coin tosses of the **same** coin.

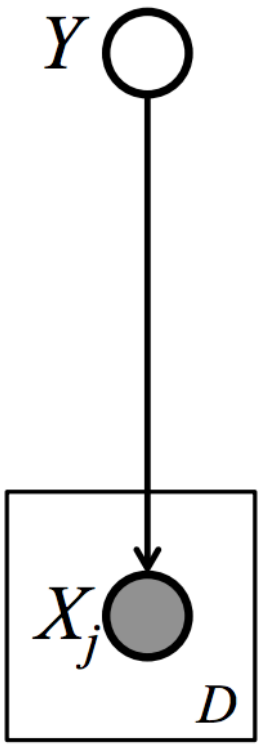
- What is the corresponding BN?



Parameter Sharing

- Plate notation



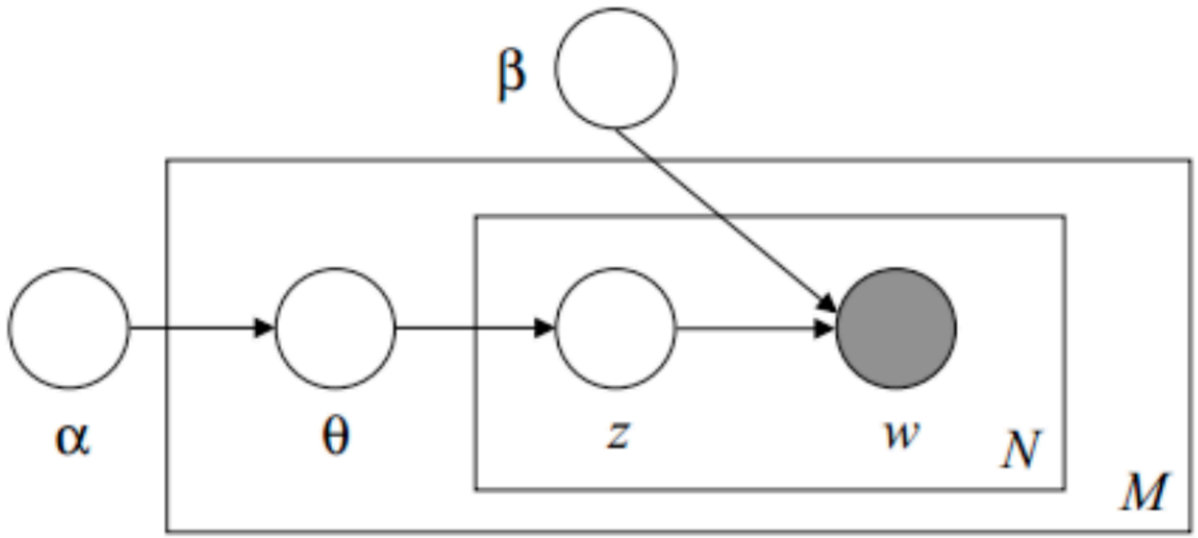


Plates denote replication of random variables

Slide courtesy: Dhruv Batra

Hierarchical Bayesian Models

- Why stop with a single prior?



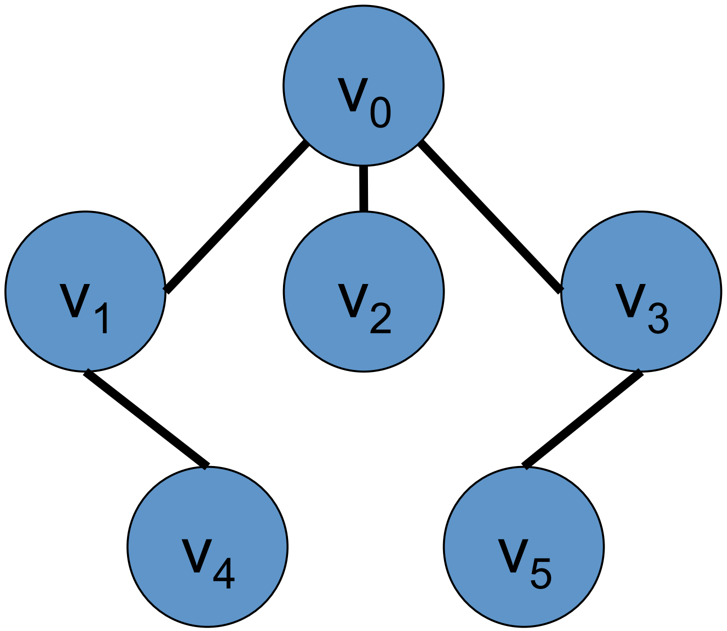
Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

Slide courtesy: Dhruv Batra

Summary: Learning BN

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

Known Tree Structure



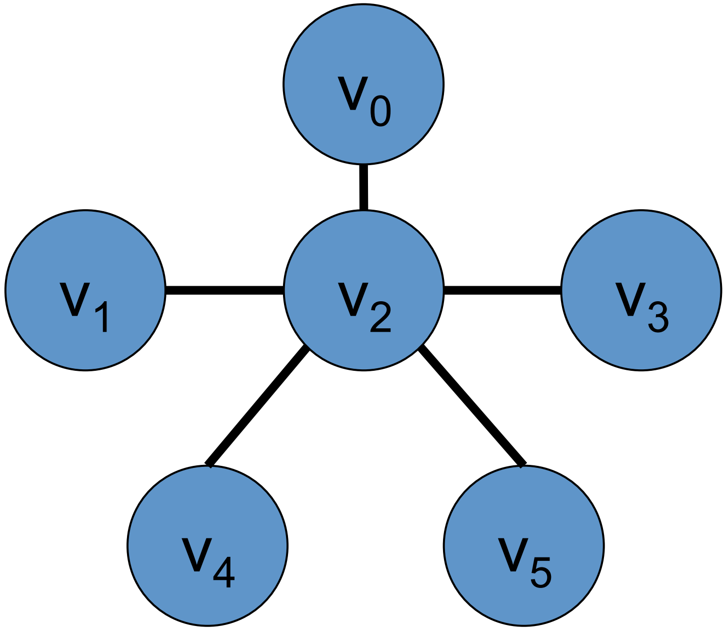
Distribution $P_T(x)$

$v_{p(a)}$ = "parent" of v_a

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

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

Known Tree Structure



Distribution $P_T(x)$

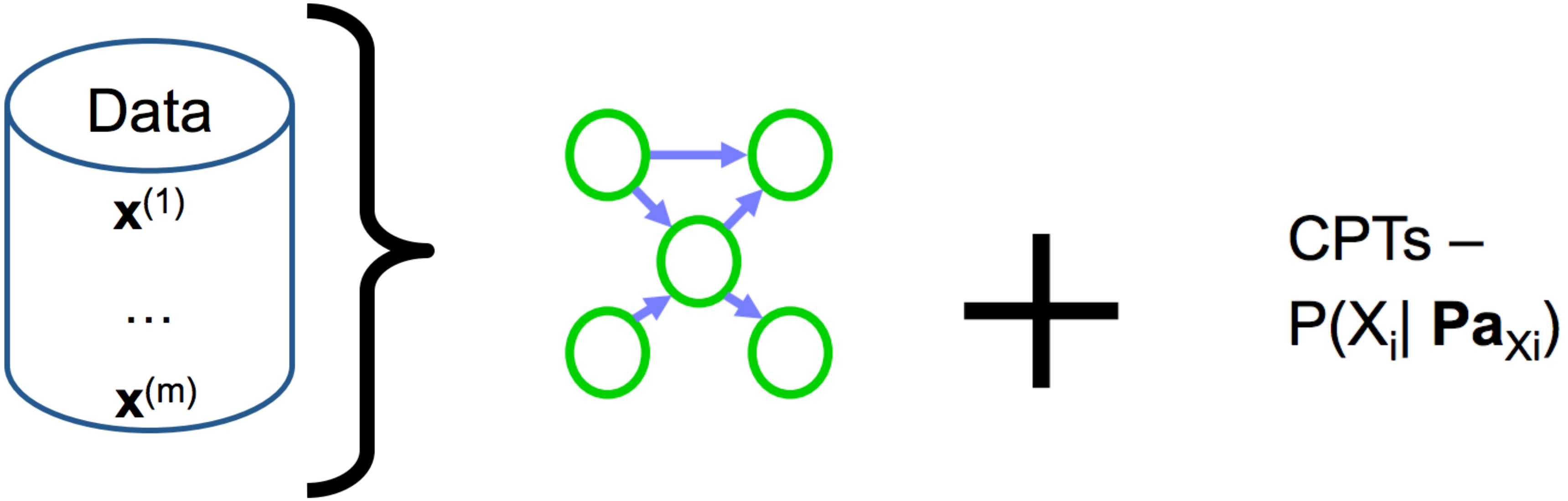
$v_{p(a)}$ = "parent" of v_a

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

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

Learning Bayesian Nets

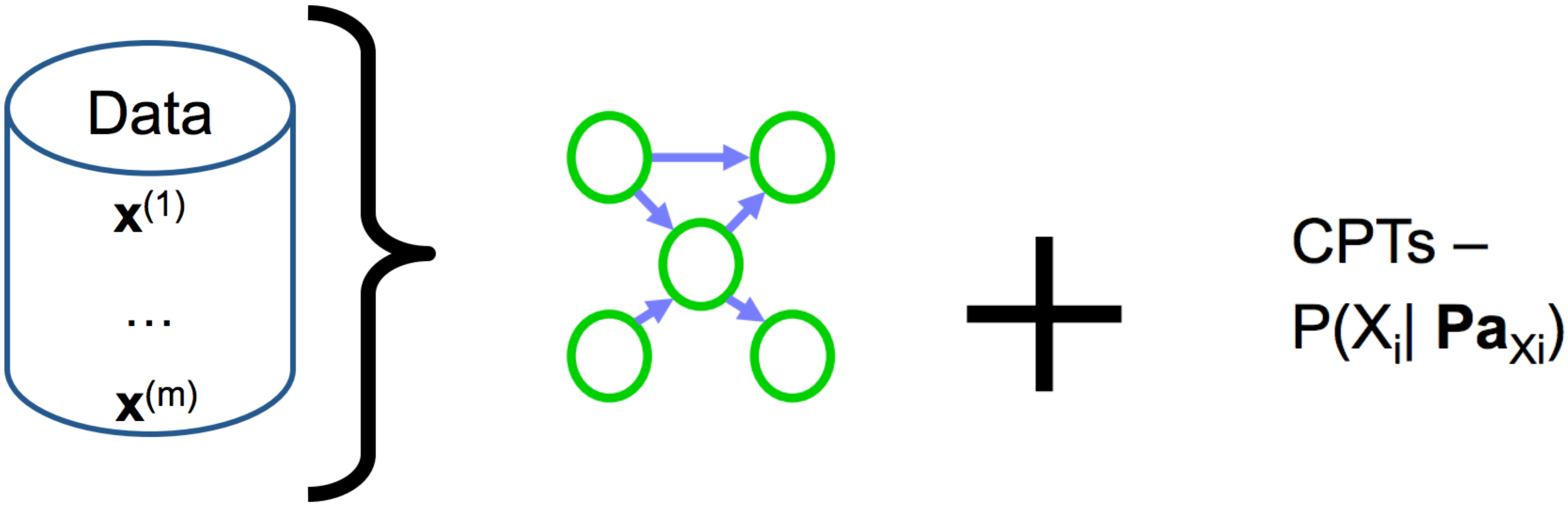
	Known structure	**Unknown structure**
Fully observable data	Very easy	Hard
Missing data	Somewhat easy (EM)	Very very hard



Slide credit: Carlos Guestrin, Dhruv Batra

Learning Bayesian Nets: Structure

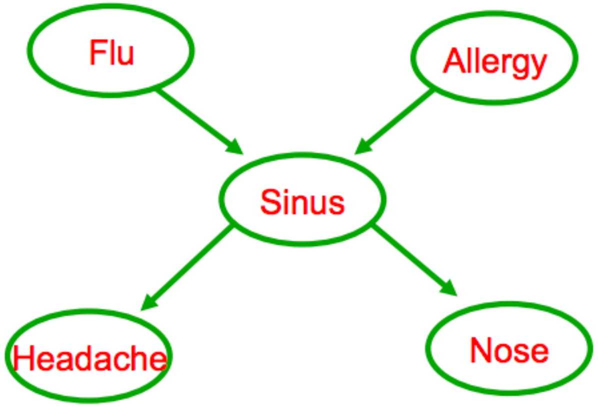
- Prediction: Care about a good structure => good prediction
- Discovery: Understand some system



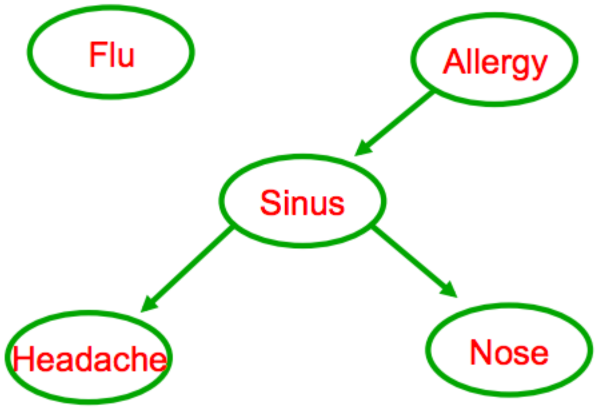
Slide credit: Carlos Guestrin, Dhruv Batra

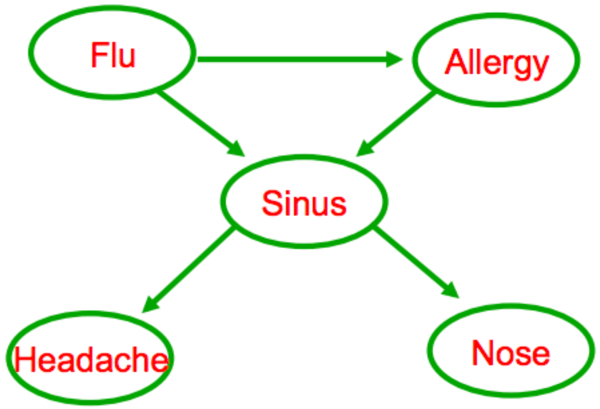
Learning Bayesian Nets: Structure

- Truth



- Recovered

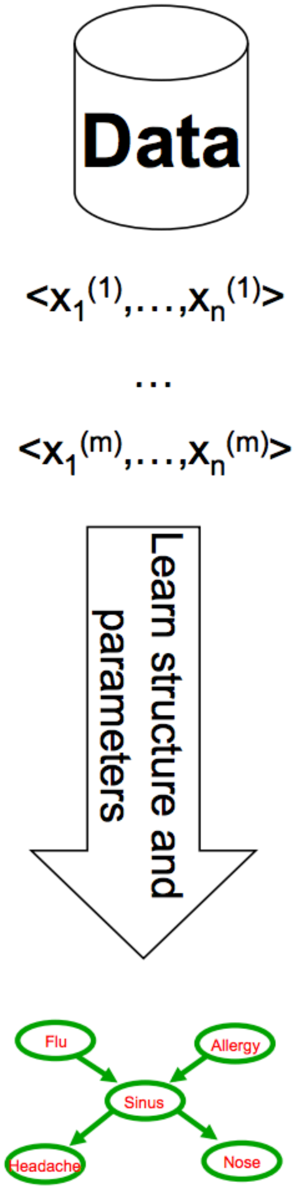




Slide courtesy: Dhruv Batra

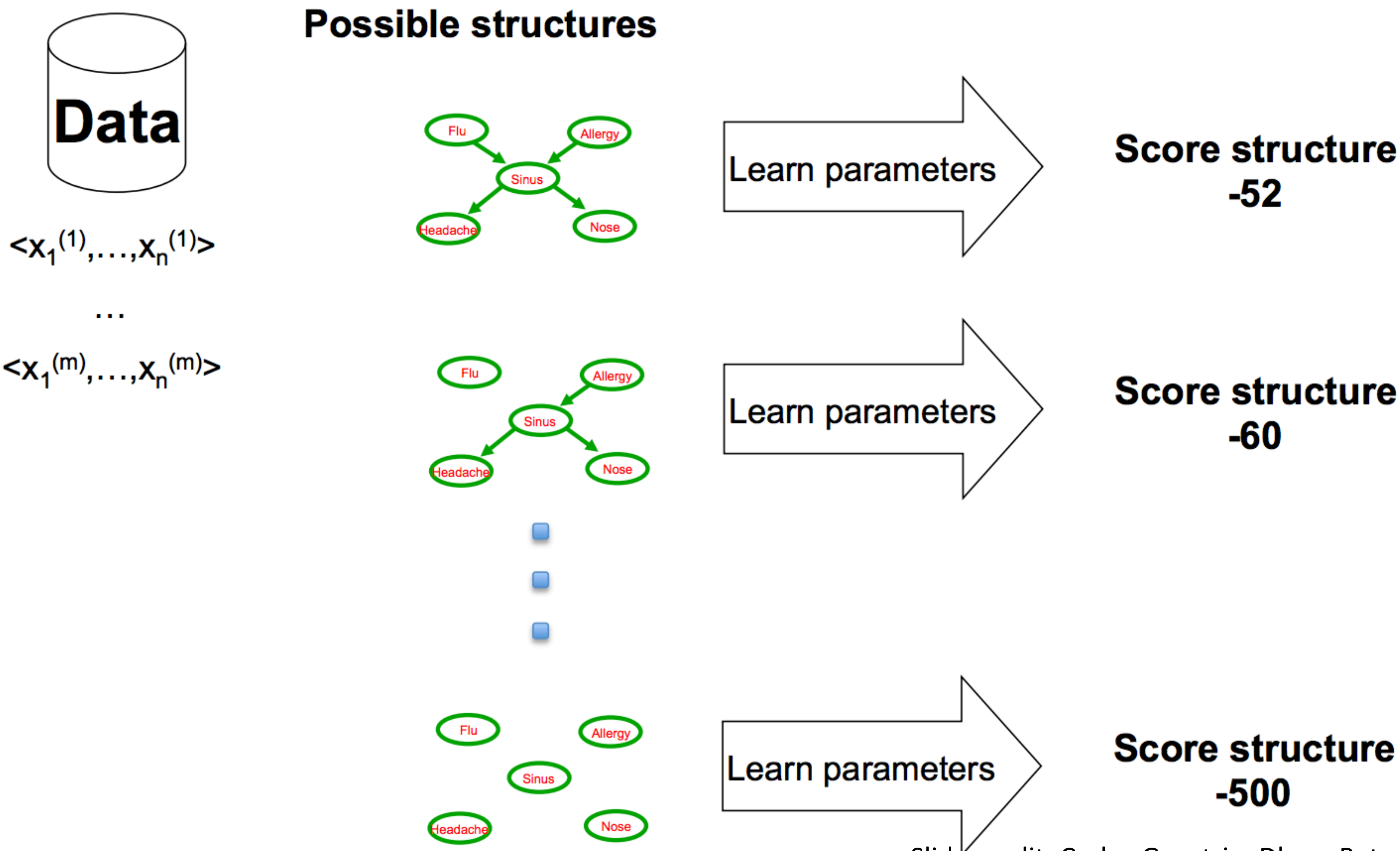
Learning Bayesian Nets: Structure

- 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



Slide courtesy: Dhruv Batra

Score-based Approach



Slide credit: Carlos Guestrin, Dhruv Batra

Score-based Approach

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

Slide courtesy: Dhruv Batra

Score-based Approach

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

Slide courtesy: Dhruv Batra

Kullback-Leibler Divergence

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

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

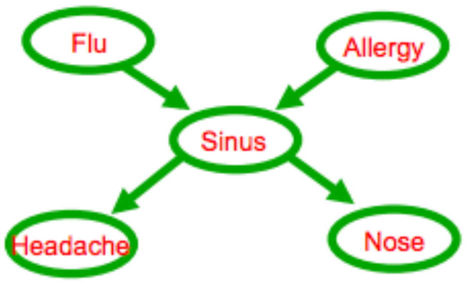
$$min - \sum_a \sum_{x_a} \sum_{x_{p(a)}} P(x_a, x_{p(a)}) \log \frac{P_T(x_a, x_{p(a)})}{P_T(x_{p(a)}) P(x_a)}$$

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

Mutual Information

Score-based Approach

- For a general graph G,



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

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

Slide credit: Carlos Guestrin, Dhruv Batra

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 → higher score
  - Decomposes over families (nodes and its parents)
  - Information never hurts!
  - But….

Slide courtesy: Dhruv Batra

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

Slide courtesy: Dhruv Batra

Chow-Liu Tree Learning - I

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

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

  - Compute mutual information

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

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

Slide credit: Carlos Guestrin, Dhruv Batra

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)

Slide credit: Carlos Guestrin, Dhruv Batra

Score-based Approach

- Bayesian score
  - => Prior distributions
  - – Over structures
  - – Over parameters of a structure

- Posterior over structures (given data)

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

Slide courtesy: Dhruv Batra

Bayesian Score: Structure Prior

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

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

Slide courtesy: Dhruv Batra

Bayesian Score: Parameter Prior & Integrals

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

- If $P(\theta_{\mathcal{G}} | \mathcal{G})$ is Dirichlet, then the integral has closed form!

- And, it factorizes according to families in G

Slide courtesy: Dhruv Batra

Bayesian Score: Parameter Prior & Integrals

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

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

Slide courtesy: Dhruv Batra

Learning Bayesian Nets: Structure

- **Question**: Are these score-based approaches really Bayesian?

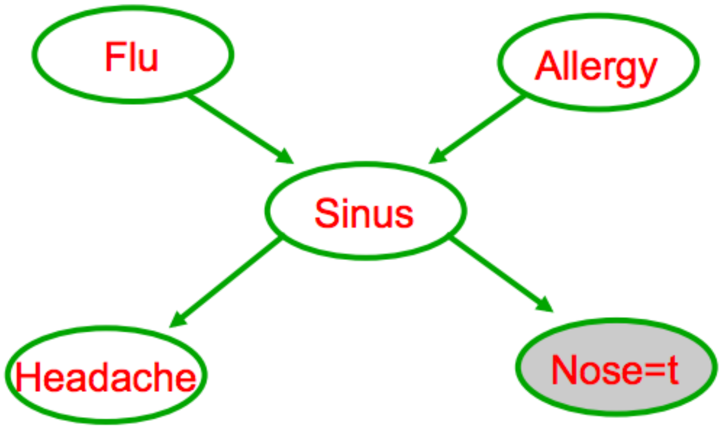
- So far, we have selected only one structure

- We must average over structures
  - Similar to averaging over parameters

This class

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

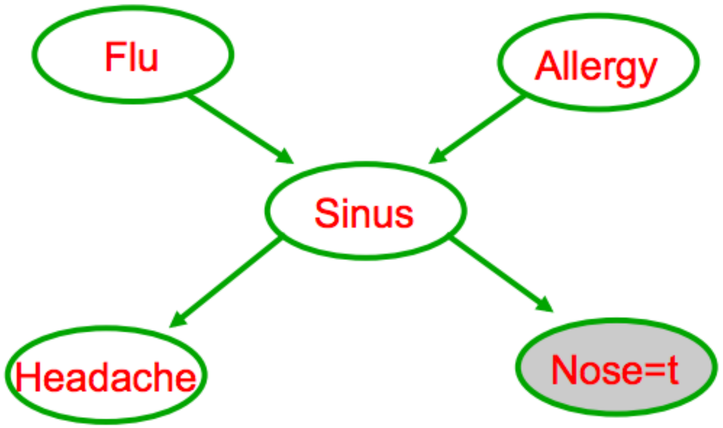
BNs: Inference



- Evidence **E**=**e** (e.g., N=t)
- Query variables of interest Y
- Conditional probability: P(Y | **E**=**e**)
  - e.g., P(F,A | N=t)
  - Special case: Marginals P(F)
- Maximum a posteriori: argmax P(all var | **E**=**e**)
  - argmax_{f,a,s,h} P(f,a,s,h | N=t)

Slide courtesy: Dhruv Batra

BNs: Inference



- Evidence **E**=**e** (e.g., N=t)
- Query variables of interest Y

- Marginal-MAP: argmax_y P(Y | **E**=**e**)
  – argmax_y Σ_o P(**Y**=**y**, **O**=**o** | **E**=**e**)

Slide courtesy: Dhruv Batra

BNs: Inference

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

P(S=f)=0.6
P(S=t)=0.4

P(N|S)

	S=0	S=1
N=0	0.5	0
N=1	0.5	1

Slide courtesy: Dhruv Batra

BNs: Inference

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

Slide courtesy: Dhruv Batra

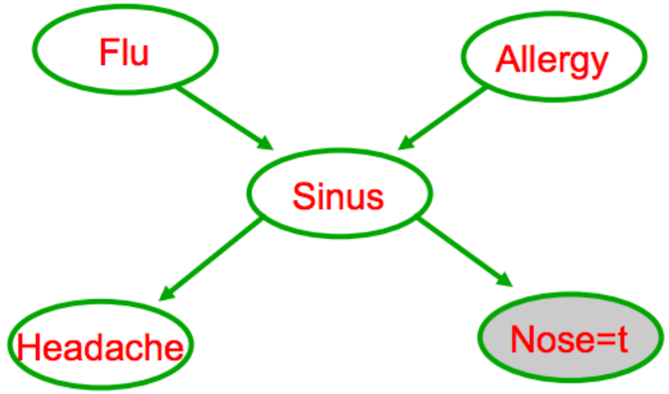
BNs: Inference

- Variable Elimination
- Sum-product belief propagation
- Sampling: MCMC

- Integer programing (LP relaxation)
- Combinatorial optimization (e.g., graphcuts)

Marginal Inference

- Consider the example
  - Evidence: N=t
  - Compute: P(F | 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., $X_1, \ldots X_n$

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

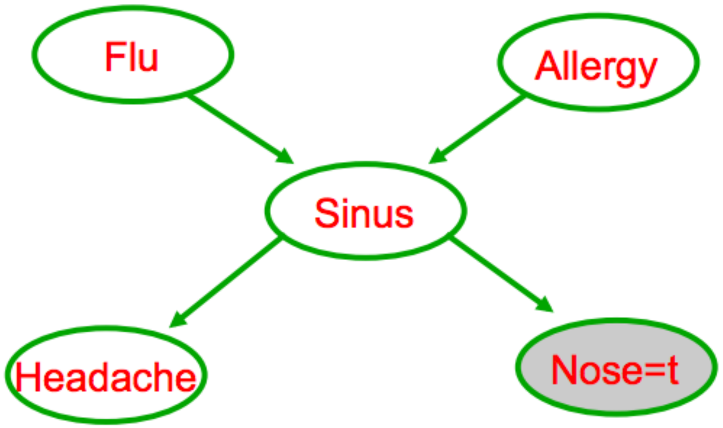
$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

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

Slide courtesy: Dhruv Batra

MAP Inference

- Evidence **E**=**e** (e.g., N=t)
- Query variables of interest Y



- Maximum a posteriori: argmax P(all var | **E**=**e**)
  – argmax_{f,a,s,h} P(f,a,s,h | N=t)

Slide courtesy: Dhruv Batra

Variable Elimination for MAP Inference

- Given a BN and a query $\max_{x1\ldots xn} P(x_1\ldots x_n,\mathbf{e})$,
- Choose an ordering on variables, e.g., $X_1,\ldots X_n$

- For i=1...n, if $X_i \notin \mathbf{E}$
  - Collect factors $f_1\ldots f_k$ 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)

Slide courtesy: Dhruv Batra

Variable Elimination for MAP Inference

- $\{x_1^* \ldots x_n^*\}$ will store the maximizing assignment

- For $i=n\ldots1$, if $X_i \notin \mathbf{E}$
  - Take factors $f_1 \ldots f_k$ used when X_i was eliminated
  - Instantiate $f_1 \ldots f_k$ with $\{x_{i+1}^* \ldots x_n^*\}$
  - Generate maximizing assignment for X_i:

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

- (This completes the backward pass)

Slide courtesy: Dhruv Batra