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
This class

• Bayesian Networks
  – Parameter Learning
  – Structure Learning
  – Inference
But first...
A quiz!

1. How would you parameterize
\[
MRF_G(x; u^k, h^k) = \sum_p u^k_p(x_p) + \sum_c h^k_c(x_c)
\]
for learning?

2. Name two parameter learning approaches for MRFs.

3. What loss functions would you use for these approaches?
   \[
   \min_w R(w) + \sum_{k=1}^{K} L_G(x^k, z^k; w)
   \]
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

\[
P(Y_1, \ldots, Y_n) = \prod_{i=1}^{n} P(Y_i \mid Pa_{Y_i})
\]
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

$x^{(1)}$
...
$x^{(m)}$

CPTs –
$P(X_i|Pa_{X_i})$
Learning Bayesian Nets

<table>
<thead>
<tr>
<th>Fully observable data</th>
<th>Known structure</th>
<th>Unknown structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Very easy</td>
<td>Hard</td>
</tr>
<tr>
<td>Missing data</td>
<td>Somewhat easy</td>
<td>Very very hard</td>
</tr>
</tbody>
</table>

\[\text{Data} \quad \{\begin{array}{c} x^{(1)} \\ \vdots \\ x^{(m)} \end{array}\} \quad + \quad \text{CPTs} - P(X_i | Pa_{X_i})\]

Slide credit: Carlos Guestrin, Dhruv Batra
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?
  – Log-likelihood($\theta$) = 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 $X_i$

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

• Exploit priors
  – Priors: Beliefs before experiments are conducted
  – Help deal with unseen data
  – Bias us towards “simpler” models

• Beta prior distribution

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

\[
\hat{\theta} = \arg \max_{\theta} P(\theta \mid D)
\]

\[
P(\theta \mid D) \sim Beta(m_H + \alpha_H, m_T + \alpha_T)
\]

• Beta prior equiv. to extra H/T
• As \( m \to \infty \), 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

\[ \theta \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k) \sim \prod_i \theta_i^{\alpha_i - 1} \]
Meta BN: Bayesian view of BN

- Show parameters explicitly as variables
- Two examples (on board)
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 $X_1, X_2, \ldots X_n$ represent coin tosses of the same coin.

• What is the corresponding BN?
Parameter Sharing

• Plate notation

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.
Summary: Learning BN

• MLE
  – Decomposes; results in counting procedure
• Bayesian estimation
  – Priors = regularization (smoothing)
  – Hierarchical priors
• Plate notation
• Shared parameters
Known Tree Structure

Distribution $P_T(x)$

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

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

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

Fully observable data
Very easy

Unknown structure

Missing data
Somewhat easy (EM)

Hard

Very very hard

Data
$x^{(1)}$

$\ldots$

$x^{(m)}$

CPTs – $P(X_i| Pa_{Xi})$
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
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

Possible structures

Data

\(<x_1^{(1)}, \ldots, x_n^{(1)}\>

\[
\ldots
\]

\(<x_1^{(m)}, \ldots, x_n^{(m)}\>

Learn parameters

Score structure -52

Learn parameters

Score structure -60

Learn parameters

Score structure -500

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?
  – \( \text{Score}(G) = \text{log-likelihood}(G: D, \theta_{\text{MLE}}) = \log P(D|G, \theta_{\text{MLE}}) \)

• How do we interpret this Max Likelihood score?
  – Consider a two-node graph (on board)
Kullback-Leibler Divergence

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

\[ KL(P_1 \| P_2) \geq 0 \]

\[ KL(P_1 \| P_1) = 0 \]

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

\[ \min - \sum_x P(x) \sum_a \log P_T(x_a | x_{p(a)}) \]
Estimating the Tree Structure

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

$$\log \hat{P}(D \mid \theta, G) = m \sum_i \sum_{x_i, Pa_{x_i}, G} \hat{P}(x_i, Pa_{x_i}, G) \log \hat{P}(x_i \mid Pa_{x_i}, G)$$

$$\log \hat{P}(D \mid \theta, G) = m \sum_i \hat{I}(X_i, Pa_{X_i}) - m \sum_i \hat{H}(X_i)$$
Score-based Approach

\[ \log \hat{P}(D | \theta, G) = m \sum_i \hat{I}(X_i, Pa_{X_i}) - m \sum_i \hat{H}(X_i) \]

• 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 $X_1, X_2, \ldots, X_n$
  – Edge $(i,j)$ gets weight $\hat{I}(X_i, X_j)$

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)
Score-based Approach

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

• Posterior over structures (given data)

\[
\log P(G \mid D) \propto \log P(G) + \log \int_{\theta_G} P(D \mid G, \theta_G) P(\theta_G \mid G) d\theta_G
\]
Bayesian Score: Structure Prior

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

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

Slide courtesy: Dhruv Batra
Bayesian Score: Parameter Prior & Integrals

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

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

• And, it factorizes according to families in \( \mathcal{G} \)
Bayesian Score: Parameter Prior & Integrals

\[
\log P(G \mid D) \propto \log P(G) + \log \int_{\theta_G} P(D \mid G, \theta_G) P(\theta_G \mid G) d\theta_G
\]

- How should we choose Dirichlet hyperparameters?
  - K2 prior: Fix an \( \alpha \), \( P(\theta_{X_i \mid Pa_{Xi}}) = \text{Dirichlet}(\alpha, \ldots, \alpha) \)
  
  - BDe Prior: Pick a "prior" BN
    - Compute \( P(X_i, Pa_{Xi}) \) 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: $P(Y \mid E=e)$
  – e.g., $P(F,A \mid N=t)$
  – Special case: Marginals $P(F)$

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

Slide courtesy: Dhruv Batra
BNs: Inference

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

• Marginal-MAP: $\text{argmax}_y P(Y \mid E=e)$
  $\quad \text{argmax}_y \sum_o P(Y=y, O=o \mid E=e)$
BNs: Inference

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

<table>
<thead>
<tr>
<th></th>
<th>S=0</th>
<th>S=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>N=0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>N=1</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

$P(S=f) = 0.6$
$P(S=t) = 0.4$
• 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 programing (LP relaxation)
• Combinatorial optimization (e.g., graphcuts)
Marginal Inference

• Consider the example
  – Evidence: N=t
  – Compute: P(F | N=t)

• (On board)
Variable Elimination

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

- For $i=1,...,n$, if $X_i \notin \{Y,E\}$
  - Collect factors $f_1...f_k$ 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 $P(Y,e)$ to obtain $P(Y|e)$

Slide courtesy: Dhruv Batra
MAP Inference

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

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

Slide courtesy: Dhruv Batra
Variable Elimination for MAP Inference

• Given a BN and a query \( \max_{x_1...x_n} P(x_1...x_n, e) \),

• Choose an ordering on variables, e.g., \( X_1,...X_n \)

• For \( i=1...n \), if \( X_i \not\in \mathbf{E} \)
  – Collect factors \( f_1...f_k \) 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^*...x_n^*\} will store the maximizing assignment

• For i=n...1, if X_i \not\in E
  – Take factors f_1...f_k used when X_i was eliminated
  – Instantiate f_1...f_k with \{x_{i+1}^*...x_n^*\}
  – Generate maximizing assignment for X_i:

\[
x_{i}^* \in \arg\max_{x_i} \prod_{j=1}^{k} f_{j}
\]

• (This completes the backward pass)