<table>
<thead>
<tr>
<th>Chap.</th>
<th>Handout</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>n</td>
<td>Cond. Indep and factorization</td>
</tr>
<tr>
<td>3</td>
<td>n</td>
<td>The elimination algorithm</td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td>Prob. propagation and factor graphs</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>Statistical concepts</td>
</tr>
<tr>
<td>6</td>
<td>n</td>
<td>Linear regression and LMS</td>
</tr>
<tr>
<td>7</td>
<td>n</td>
<td>Linear classification</td>
</tr>
<tr>
<td>8</td>
<td>y</td>
<td>Exponential family and GLIMs</td>
</tr>
<tr>
<td>9</td>
<td>y</td>
<td>Completely observed GMs (IPF, etc)</td>
</tr>
<tr>
<td>10</td>
<td>y</td>
<td>Mixtures and conditional mixtures</td>
</tr>
<tr>
<td>11</td>
<td>y</td>
<td>The EM algorithm</td>
</tr>
<tr>
<td>12</td>
<td>y</td>
<td>HMMs</td>
</tr>
<tr>
<td>13</td>
<td>y</td>
<td>The multivariate Gaussian</td>
</tr>
<tr>
<td>14</td>
<td>y</td>
<td>Factor analysis</td>
</tr>
<tr>
<td>15</td>
<td>y</td>
<td>Kalman filtering and smoothing</td>
</tr>
<tr>
<td>16</td>
<td>n</td>
<td>Markov properties of graphs</td>
</tr>
<tr>
<td>17</td>
<td>n</td>
<td>The junction tree algorithm</td>
</tr>
<tr>
<td>18</td>
<td>n</td>
<td>HMM and state space models revisited</td>
</tr>
<tr>
<td>19</td>
<td>y</td>
<td>Features, maxent and duality</td>
</tr>
<tr>
<td>20</td>
<td>y</td>
<td>Iterative scaling algorithms</td>
</tr>
<tr>
<td>21</td>
<td>n</td>
<td>Sampling methods</td>
</tr>
<tr>
<td>22</td>
<td>n</td>
<td>Decision graphs</td>
</tr>
<tr>
<td>23</td>
<td>n</td>
<td>Bio-informatics</td>
</tr>
</tbody>
</table>

**Koller & Friedman chapters**

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

**Jordan chapters**

<table>
<thead>
<tr>
<th>Chap.</th>
<th>Handout</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>n</td>
<td>Cond. Indep and factorization</td>
</tr>
<tr>
<td>3</td>
<td>n</td>
<td>The elimination algorithm</td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td>Prob. propagation and factor graphs</td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>Statistical concepts</td>
</tr>
<tr>
<td>6</td>
<td>n</td>
<td>Linear regression and LMS</td>
</tr>
<tr>
<td>7</td>
<td>n</td>
<td>Linear classification</td>
</tr>
<tr>
<td>8</td>
<td>y</td>
<td>Exponential family and GLIMs</td>
</tr>
<tr>
<td>9</td>
<td>y</td>
<td>Completely observed GMs (IPF, etc)</td>
</tr>
<tr>
<td>10</td>
<td>y</td>
<td>Mixtures and conditional mixtures</td>
</tr>
<tr>
<td>11</td>
<td>y</td>
<td>The EM algorithm</td>
</tr>
<tr>
<td>12</td>
<td>y</td>
<td>HMMs</td>
</tr>
<tr>
<td>13</td>
<td>y</td>
<td>The multivariate Gaussian</td>
</tr>
<tr>
<td>14</td>
<td>y</td>
<td>Factor analysis</td>
</tr>
<tr>
<td>15</td>
<td>y</td>
<td>Kalman filtering and smoothing</td>
</tr>
<tr>
<td>16</td>
<td>n</td>
<td>Markov properties of graphs</td>
</tr>
<tr>
<td>17</td>
<td>n</td>
<td>The junction tree algorithm</td>
</tr>
<tr>
<td>18</td>
<td>n</td>
<td>HMM and state space models revisited</td>
</tr>
<tr>
<td>19</td>
<td>y</td>
<td>Features, maxent and duality</td>
</tr>
<tr>
<td>20</td>
<td>y</td>
<td>Iterative scaling algorithms</td>
</tr>
<tr>
<td>21</td>
<td>n</td>
<td>Sampling methods</td>
</tr>
<tr>
<td>22</td>
<td>n</td>
<td>Decision graphs</td>
</tr>
<tr>
<td>23</td>
<td>n</td>
<td>Bio-informatics</td>
</tr>
</tbody>
</table>
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 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 3

- General graphs: representation
  - Independence properties (Bayes Ball, l-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 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: $D = H, H, T, H, \ldots$
- Model: $p(H) = \theta$ $p(T) = (1 - \theta)$
- We want to estimate $\theta$ from $D$.
- Frequentist (maximum likelihood) approach (point estimate):
  $$\hat{\theta}_{ML} = \arg\max_\theta \ell(\theta; D)$$
  where
  $$\ell(\theta; D) = \log p(D|\theta) = \sum_m \log p(x^m|\theta)$$
- Bayesian approach
  $$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$
  or
  $$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

**MLE FOR BERNOULLI TRIALS (L10)**
- Likelihood:
  $$\ell(\theta; D) = \log p(D|\theta) = \log \prod_m \theta^{x^m}(1 - \theta)^{1-x^m}$$
  $$= \log \theta \sum_m x^m + \log(1 - \theta) \sum_m (1 - x^m)$$
  $$= \log \theta N_H + \log(1 - \theta) N_T$$
- Take derivatives and set to zero:
  $$\frac{\partial \ell}{\partial \theta} = \frac{N_H}{\theta} - \frac{N_T}{1 - \theta}$$
  $$\Rightarrow \theta_{ML} = \frac{N_H}{N_H + N_T}$$
- 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
  \[ P(D|\theta) = \theta^N H (1-\theta)^T \]

- Conjugate Beta Prior
  \[ P(\theta | \alpha) = B(\theta; \alpha_h, \alpha_t) \defeq \frac{1}{Z(\alpha_h, \alpha_t)} \theta^{\alpha_h - 1} (1-\theta)^{\alpha_t - 1} \]

- Posterior
  \[ P(\theta | D, \alpha) = \frac{P(\theta | \alpha) P(D|\theta)}{P(D|\alpha)} = \frac{1}{Z(\alpha_h, \alpha_t) P(D|\alpha)} \theta^{\alpha_h - 1} \theta^N H (1-\theta)^{\alpha_t - 1} (1-\theta)^T = B(\theta; \alpha_h + N_h, \alpha_t + N_t) \]

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

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%”: \( p = P(Y \geq 140) + P(Y \leq 110) = 0.066 \):

\[ n=250; p = 0.5; y = 140; \]

\[ p = (1-binocdf(y-1,n,p)) + binocdf(n-y,n,p) \]

- 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?

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 \( D \).

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 → multinomial
  \[ P(D|\theta) = \prod_i \theta_i^{N_i} \]

- Prior: beta → Dirichlet
  \[ P(\tilde{\theta}|\tilde{\alpha}) = \frac{1}{Z(\tilde{\alpha})} \prod_i \theta_i^{\alpha_i-1} \]
  where
  \[ Z(\tilde{\alpha}) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)} \]

- Posterior: beta → Dirichlet
  \[ P(\tilde{\theta}|D) = Dir(\tilde{\alpha} + \tilde{\beta}) \]

- Evidence (marginal likelihood)
  \[ P(D|\tilde{\alpha}) = \frac{Z(\tilde{\alpha})}{Z(\tilde{\alpha})} = \frac{\prod_i \Gamma(\alpha_i + N_i)}{\prod_i \Gamma(\alpha_i)} \frac{\Gamma(\sum_i \alpha_i)}{\Gamma(\sum_i \alpha_i + N_i)} \]

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: \( D=1.18,-.25,.78,\ldots \)
- Model: \( p(x) = (2\pi\sigma^2)^{-1/2} \exp\{-\frac{(x - \mu)^2}{2\sigma^2}\} \)
- Log likelihood:
  \[ \ell(\theta; D) = \log p(D|\theta) = -\frac{M}{2} \log(2\pi\sigma^2) - \frac{1}{2} \sum_m \frac{(x^m - \mu)^2}{\sigma^2} \]
- Take derivatives and set to zero:
  \[ \frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2} \sum_m (x^m - \mu) \]
  \[ \frac{\partial \ell}{\partial \sigma^2} = -\frac{M}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_m (x^m - \mu)^2 \]
  \[ \Rightarrow \mu_{ML} = (1/M) \sum_m x^m \]
  \[ \sigma^2_{ML} = (1/M) \sum_m (x^m - \mu_{ML})^2 \]

Exponential Family (L4, L10)

- For a numeric random variable \( x \)
  \[ p(x|\eta) = h(x) \exp\{\eta^\top T(x) - A(\eta)\} \]
  \[ = \frac{1}{Z(\eta)} h(x) \exp\{\eta^\top T(x)\} \]
  is an exponential family distribution with natural (canonical) parameter \( \eta \).
- Function \( T(x) \) 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
- 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)

**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)

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

\[ P(Y = 1) \] 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

| X \( \in \mathbb{R}^n \) | Y \( \in \mathbb{R}^m \) | \( p(Y|X) \) |
|----------------|----------------|-------------|
| Gauss \( Y;WX + \mu, \Sigma \) |
| \( \{0, 1\} \) \( \in \{0, 1\} \) | Bernoulli \( Y; p = \frac{1}{1+e^{-\theta^T x}} \) |
| \( \{0, 1\} \) \( \in \{0, 1\} \) | Bernoulli \( Y; p = \frac{1}{1+e^{-\theta^T x}} \) |
| \( \mathbb{R}^n \) \( \{1, \ldots, K\} \) | 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)

Learning an HMM (L10, L12)

- Consider a time-invariant hidden Markov model (HMM)
  - State transition matrix $A(i, j) \overset{\text{def}}{=} P(X_t = j | X_{t-1} = i)$,
  - Discrete observation matrix $B(i, j) \overset{\text{def}}{=} P(Y_t = j | X_t = i)$
  - State prior $\pi(i) \overset{\text{def}}{=} P(X_1 = i)$.
- If all nodes are observed, we can find the globally optimal MLE.
- Otherwise using EM (aka Baum Welch).
Kalman filter (L17, L18)

- LDS model: \( x_t = Ax_{t-1} + v_t, \quad y_t = Cx_t + w_t \)
- 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):
  \[
  \hat{y}_t = y_t - \hat{y}_{t|t-1} \quad \text{(error/innovation)}
  \]
  \[
  P_{\hat{y}t} = CP_{t|t-1}C^T + R \quad \text{(covariance of error)}
  \]
  \[
  P_{x\hat{y}t} = P_{t|t-1}C^T \quad \text{(cross covariance)}
  \]
  \[
  K_t = P_{x\hat{y}t}P_{\hat{y}t}^{-1} \quad \text{(Kalman gain matrix)}
  \]
  \[
  x_{t|t} = x_{t|t-1} + K_t(y_t - y_{t|t-1})
  \]
  \[
  P_{t|t} = P_{t|t-1} - K_tP_{x\hat{y}t}
  \]

KF for 2D tracking (L17)

- State is location of robot and landmarks
  \[
  X_t = (R_t, L_1^{1:N})
  \]
- Measure location of subset of landmarks at each time step.
- Assume everything is linear Gaussian.
- Use Kalman filter to solve optimally.

Approximate determinsitic filtering (L18)

- Extended Kalman filter (EKF)
- Unscented Kalman filter (UKF)
- Assumed density filter (ADF)

\[
\begin{align*}
\hat{\alpha}_t & \quad U \quad \hat{\alpha}_t \quad \hat{\alpha}_{t+1} & \quad \text{exact} \\
\tilde{\alpha}_{t-1} & \quad \tilde{\alpha}_t & \quad \tilde{\alpha}_{t+1} & \quad \text{approx}
\end{align*}
\]
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.

![Particle filtering diagram]

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

Example BN: Water sprinkler (L1)

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

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

Trees

- Inference (belief propagation): L9, Yedidia tutorial
- Structure learning (max weight spanning tree): L16
- Application: KF trees for multiscale image analysis (skipped)
Bayes net for genetic pedigree analysis (L1)

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

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
  \[ I(H) = \{(X \perp Y | Z) : \text{sep}_H(X; Y | Z)\} \]
- Defn: The local markov independencies are
  \[ I_l(H) = \{(X \perp V \setminus \{X\} \mid N_H(X) \setminus N_H(X)) : X \in V\} \]
  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.

$$P(J) = \sum_L \sum_S \sum_G \sum_H \sum_I \sum_D \sum_C P(C, D, I, G, S, L, J, H)$$

From Bayes net to jtree (L8)

Hugin vs Shafer Shenoy

Message passing on jtrees (L8, L9)
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:

\[
\log p(D|\theta) = \log \prod_m \prod_i p(x^m_i|x_{\pi(i)}, \theta_i) = \sum_i \sum_m \log p(x^m_i|x_{\pi(i)}, \theta_i)
\]

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 \(\psi_{12}, \psi_{23}, \ldots\).

<table>
<thead>
<tr>
<th>Decomposable?</th>
<th>Max. Cliques</th>
<th>Tabular</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Direct</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>IPF</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Gradient ascent</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Iterative scaling</td>
</tr>
</tbody>
</table>

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 (e.g., chains).

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
A Generative Model for Generative Models

- Mixture of Factor Analyzers (VQ)
- Mixture of Gaussians
- Cooperative Vector Quantization
- SBN, Boltzmann Machines
- Factorial HMM
- Mixture of HMMs
- Switching State-space Models
- Nonlinear Gaussian Belief Nets
- Mixture of LDSs
- Nonlinear Dynamical Systems
- ICA
- Independent Component Analysis

mix: mixture
red-dim: reduced dimension
dyn: dynamics
hier: hierarchical
nonlin: nonlinear
switch: switching