Computer Science

CSC535: Probabilistic Graphical Models

Midterm Review

Prof. Jason Pacheco

Administrivia

- **Vote** Tomorrow 10/27 is the Arizona deadline to mail ballots

- Midterm
  - 8 Questions, 10pts per question, 80pts total
  - In D2L Assignments
  - Due Fri 10/30 @ 11:59pm
  - Send any questions as **private** messages to me on Piazza
  - Do not discuss exam questions with others

Topics

- Probability and Statistics
- Probabilistic Graphical Models
- Message Passing Inference
- Parameter Learning

Topics

- Probability and Statistics
- Probabilistic Graphical Models
- Message Passing Inference
- Parameter Learning

Probability and Random Events

- A random process is modeled by a probability space $(\Omega, \mathcal{F}, P)$ where:
  - **Sample space** Ω is the set of all possible outcomes
  - **Event space** $\mathcal{F}$ is the set of **events**, each being a subset of Ω
  - **Probability function** P assigns a probability in $[0, 1]$ to each event

- Axioms of probability

1. For any event E, $0 \leq P(E) \leq 1$
2. $P(\Omega) = 1 \text{ and } P(\emptyset) = 0$
3. For any *finite* or *countably infinite* sequence of <u>pairwise mutually disjoint events</u> $E_1, E_2, E_3, \ldots$

$$P\left(\bigcup_{i \geq 1} E_i\right) = \sum_{i \geq 1} P(E_i)$$

- An event space must contain $\{\Omega, \emptyset\}$
- Must be closed under:
  - Complements
  - Countable unions
  - Countable intersections

Probability and Random Events

- A **random variable** is a function of samples to real values: $X : \Omega \rightarrow \mathbb{R}$
- $X = x$ Is an event with probability: $p(X = x) = \sum_{\omega \in \Omega \,:\, X(\omega)=x} P(\omega)$
- Some fundamental rules of probability:
  - Conditional: $p(X \mid Y) = \frac{p(X,Y)}{p(Y)} = \frac{p(X,Y)}{\sum_x p(X=x,Y)}$
  - Law of total probability: $p(Y) = \sum_x p(Y, X = x)$
  - Probability chain rule: $p(X, Y) = p(Y)p(X \mid Y)$
- Independence of RVs:
  - Two RVs X & Y are independent iff: $p(X \mid Y) = p(X)$
  - Equivalently: $p(X, Y) = p(X)p(Y)$
  - X & Y are conditionally independent given Z iff: $p(X \mid Y, Z) = p(X \mid Z)$
  - Equivalently: $p(X, Y \mid Z) = p(X \mid Z)p(Y \mid Z)$

Useful Discrete Distributions

Bernoulli *A.k.a. the* ***coinflip*** *distribution on* *<u>binary</u>* *RVs* $X \in \{0, 1\}$

$$p(X) = \pi^X (1-\pi)^{(1-X)}$$

Suppose we flip N <u>independent</u> coins $X_1, X_2, \ldots, X_N$, what is the distribution over their sum $Y = \sum_{i=1}^N X_i$

Binomial Dist.

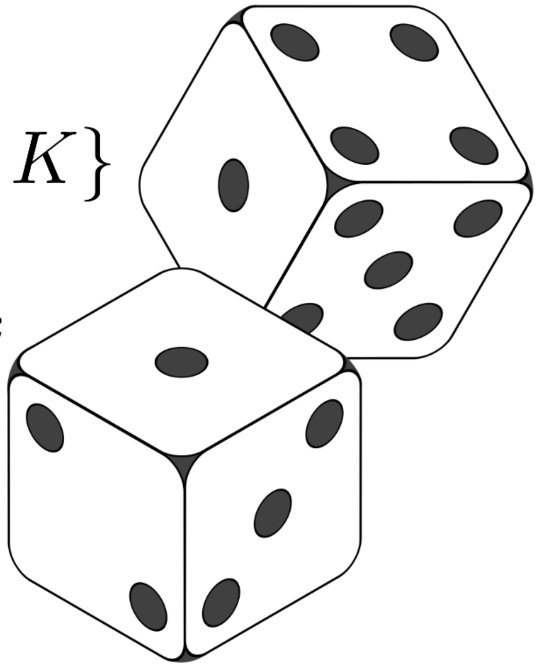
$$p(\underset{\text{Num. "successes" out of N trials}}{Y = k}) = \underset{\text{Num. ways to obtain k successes out of N}}{\binom{N}{k}} \pi^k (1-\pi)^{N-k}$$

Geometric Dist. *on number of independent draws of* $X \sim \text{Bernoulli}(\pi)$ *until success:*

$$p(Y = n) = (1-\pi)^{n-1}\pi \qquad \mathbf{E}[Y] = \frac{1}{\pi}$$

e.g. must be n-1 failures (tails) before a success (heads).

Useful Discrete Distributions



Categorical *Distribution on integer-valued RV* $X \in \{1, \ldots, K\}$

$p(X) = \prod_{k=1}^{K} \pi_k^{\mathbf{I}(X=k)}$ or $p(X) = \sum_{k=1}^{K} \mathbf{I}(X = k) \cdot \pi_k$

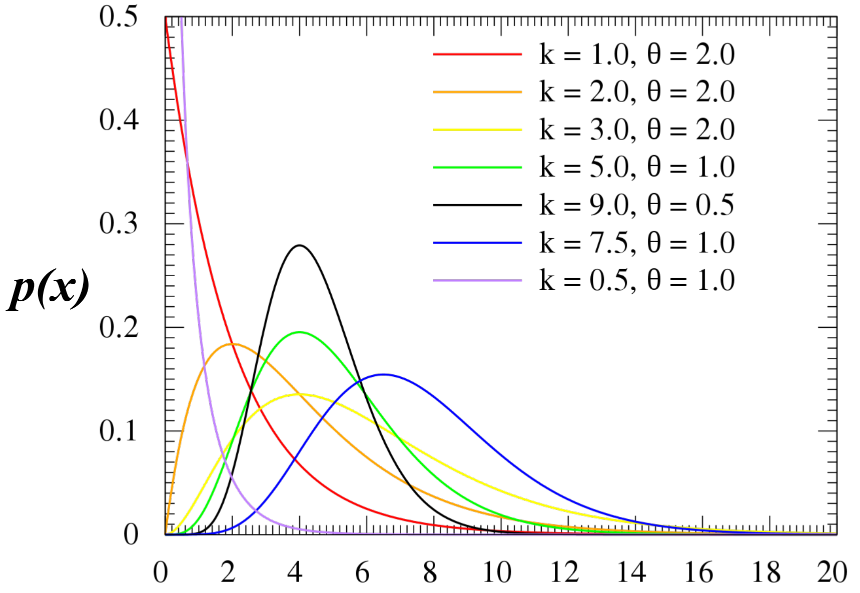
with parameter $p(X = k) = \pi_k$ *and Kronecker delta I()*

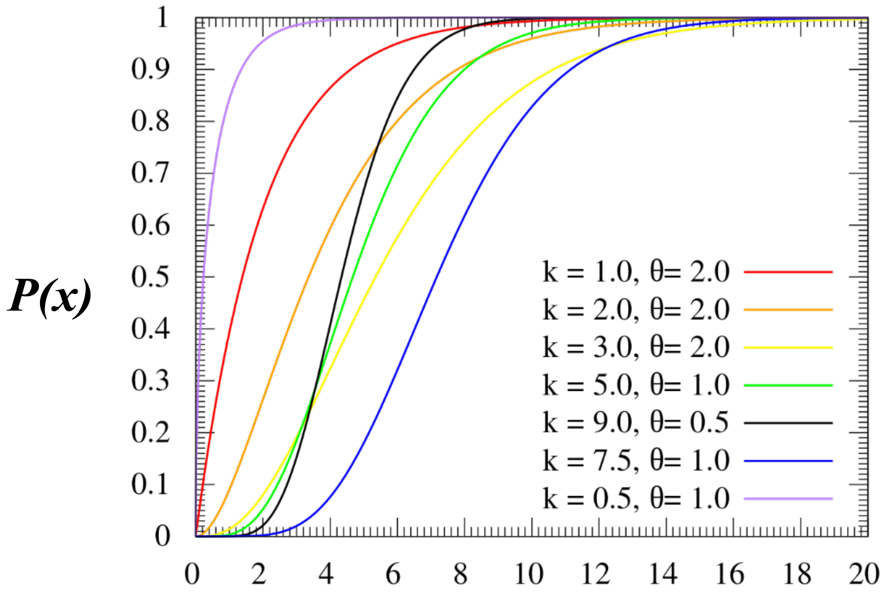
Extension to *N* independent trials…

Multinomial *Distribution on K-vector* $X \in \{0, N\}^K$ *of counts of N repeated trials* $\sum_{k=1}^{K} X_k = N$ *with PMF:*

$$p(x_1, \ldots, x_K) = \binom{n}{x_1 x_2 \ldots x_K} \prod_{k=1}^{K} \pi_k^{x_k}$$

Useful Continuous Distributions





Gamma distribution for $x \geq 0$

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \qquad P(X \leq x) = \frac{1}{\Gamma(\alpha)} \gamma(\alpha, \beta x)$$

For “shape” $\alpha \geq 0$ and “scale” $\beta \geq 0$

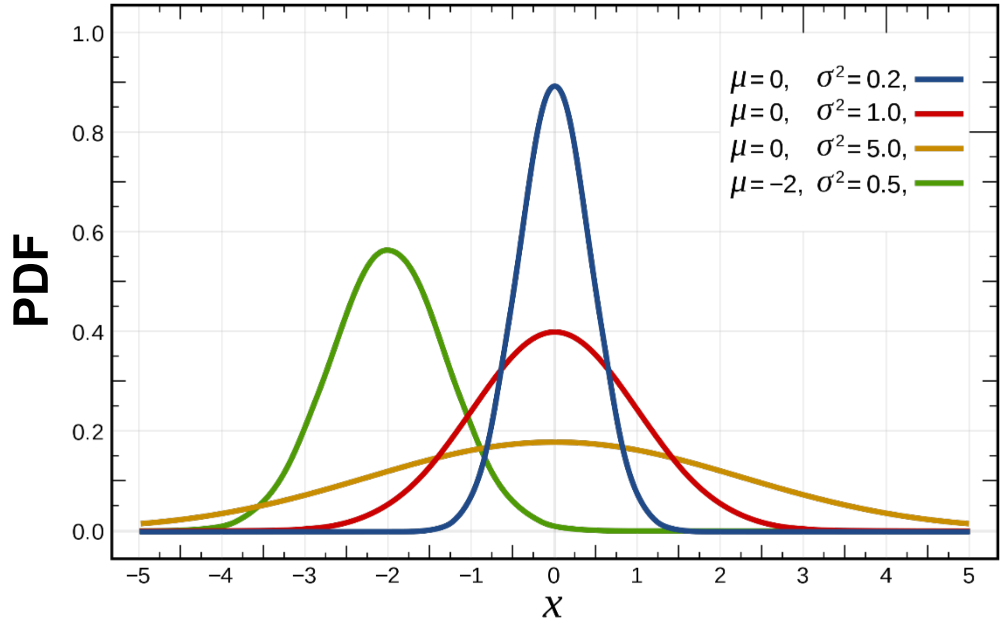
$$\mathbf{E}[X] = \frac{\alpha}{\beta} \qquad \mathbf{Var}[X] = \frac{\alpha}{\beta^2}$$

Useful Continuous Distributions

Gaussian (a.k.a. Normal) distribution with mean mean (location) μ and variance (scale) σ^2 parameters,

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{1}{2}(x-\mu)^2/\sigma^2$$

We say $X \sim \mathcal{N}(\mu, \sigma^2)$.



Useful Properties

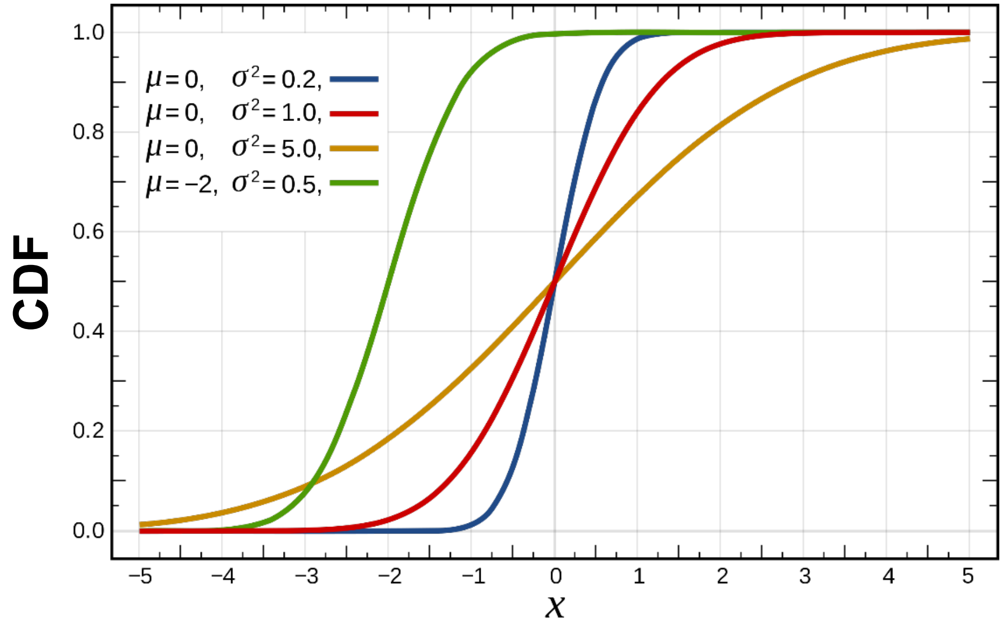
- Closed under additivity:

$$X \sim \mathcal{N}(\mu_x, \sigma_x^2) \qquad Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$$

$$X + Y \sim \mathcal{N}(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2)$$

- Closed under linear functions (a and b constant):

$$aX + b \sim \mathcal{N}(a\mu_x + b, a^2\sigma_x^2)$$

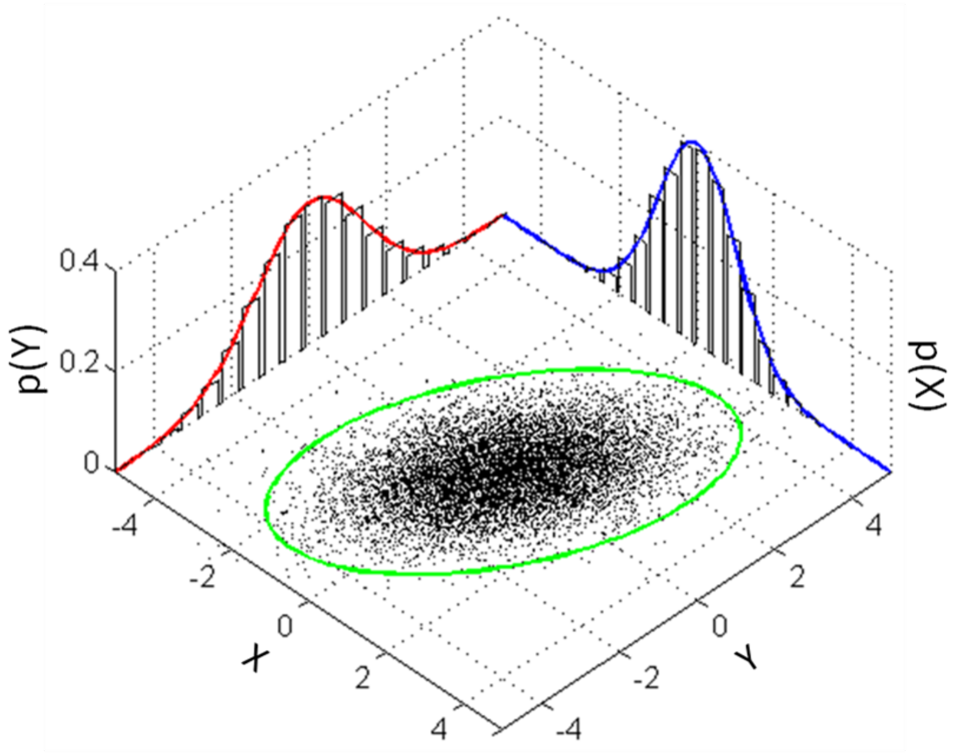


Useful Continuous Distributions

Multivariate Gaussian On RV $X \in \mathcal{R}^d$ with mean $\mu \in \mathcal{R}^d$ and positive semidefinite covariance matrix $\Sigma \in \mathcal{R}^{d \times d}$,

$$p(x) = |2\pi\Sigma|^{-1/2} \exp -\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)$$

Moments given by parameters directly.



Useful Properties

- Closed under additivity (same as univariate case)
- Closed under linear functions,

$$AX + b \sim \mathcal{N}(A\mu_x + b, A\Sigma A^T)$$

  Where $A \in \mathcal{R}^{m \times d}$ and $b \in \mathcal{R}^m$ (output dimensions may change)

- Closed under conditioning and marginalization (See Bishop Sec. 2.3)

Will discuss Gaussians a lot more when we cover exponential families

Bayesian Inference

Posterior distribution is complete representation of uncertainty

Posterior computed by **Bayes' rule:**

Prior Belief

Likelihood

$$p(\theta \mid y) = \frac{p(\theta)p(y \mid \theta)}{p(y)}$$

Marginal Likelihood (more on this later)

- Must specify a <u>prior belief</u> $p(\theta)$ about coin bias
- Coin bias θ is a <u>random quantity</u>
- Interval $p(l(y) < \theta < u(y) \mid y) = 0.95$ can be reported in lieu of full posterior, and takes intuitive interpretation for a <u>single trial</u>

Interval Interpretation: For <u>this trial</u> there is a 95% chance that θ lies in the interval

Marginal Likelihood

Posterior calculation requires the **marginal likelihood**,

$$p(\theta \mid y) = \frac{p(\theta)p(y \mid \theta)}{p(y)} \qquad\qquad p(y) = \int p(\theta)p(y \mid \theta)\,d\theta$$

- Also called the **partition function** or **evidence**
- Key quantity for model learning and selection
- NP-hard to compute in general (actually #P)

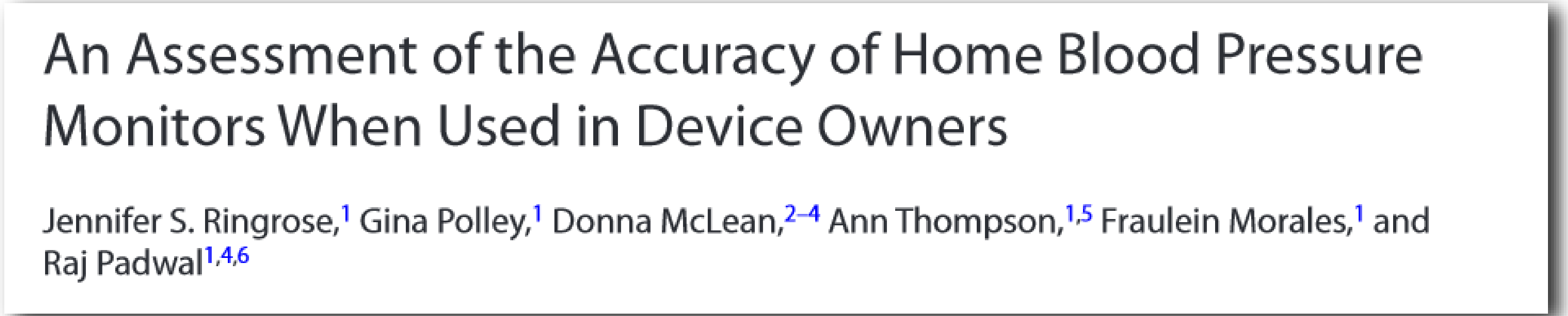
Example: Consider the vector $\theta = (\theta_1, \ldots, \theta_d)^T$ with binary $\theta_i \in \{0, 1\}$,

$$p(y) = \underbrace{\sum_{\theta_1=0}^{1} \sum_{\theta_2=0}^{1} \cdots \sum_{\theta_d=0}^{1}}_{\mathcal{O}(2^d)} p(\theta)p(y \mid \theta)$$

Bayesian Inference Example

About 29% of American adults have high blood pressure (BP). Home test has 30% false positive rate and no false negative error.

A recent home test states that you have high BP. Should you start medication?



An Assessment of the Accuracy of Home Blood Pressure Monitors When Used in Device Owners

Jennifer S. Ringrose,[1] Gina Polley,[1] Donna McLean,[2–4] Ann Thompson,[1,5] Fraulein Morales,[1] and Raj Padwal[1,4,6]

Bayesian Inference Example

About 29% of American adults have high blood pressure (BP). Home test has 30% false positive rate and no false negative error.

- Latent quantity of interest is hypertension: $\theta \in \{true, false\}$
- Measurement of hypertension: $y \in \{true, false\}$
- Prior: $p(\theta = true) = 0.29$
- Likelihood: $p(y = true \mid \theta = false) = 0.30$

$$p(y = true \mid \theta = true) = 1.00$$

Bayesian Inference Example

About 29% of American adults have high blood pressure (BP). Home test has 30% false positive rate and no false negative error.

Suppose we get a positive measurement, then posterior is:

$$p(\theta = true \mid y = true) = \frac{p(\theta = true)p(y = true \mid \theta = true)}{p(y = true)}$$

$$= \frac{0.29 * 1.00}{0.29 * 1.00 + 0.71 * 0.30} \approx 0.58$$

What conclusions can be drawn from this calculation?

Prediction

Can make predictions of unobserved $\tilde{y}$ before seeing any data,

$$p(\tilde{y}) = \int p(\theta)p(\tilde{y} \mid \theta)\, d\theta$$

Similar calculation to marginal likelihood

This is the ***prior predictive*** *distribution*

When we observe y we can predict future observations $\tilde{y}$,

$$p(\tilde{y} \mid y) = \int p(\theta \mid y)p(\tilde{y} \mid \theta)\, d\theta$$

This is the ***posterior predictive*** *distribution*

Prediction Example

About 29% of American adults have high blood pressure (BP). Home test has 30% false positive rate and no false negative error.

What is the likelihood of *another* positive measurement?

$$p(\tilde{y} = true \mid y = true) = \sum_{\theta \in \{true, false\}} p(\theta \mid y = true) p(\tilde{y} = true \mid \theta)$$

$$= 0.42 * 0.30 + 0.58 * 1.00 \approx 0.71$$

What conclusions can be drawn from this calculation?

Bayesian Estimation

Task: *produce an estimate* $\hat{\theta}$ *of* θ *after observing data* y

Bayes estimators minimize expected **loss function**:

$$\mathbb{E}[L(\theta,\hat{\theta}) \mid y] = \int p(\theta \mid y) L(\theta,\hat{\theta})\, d\theta$$

Example: Minimum mean squared error (MMSE):

$$\hat{\theta}^{\text{MMSE}} = \arg\min \mathbb{E}[(\hat{\theta}-\theta)^2 \mid y] = E[\theta \mid y]$$

Posterior mean always minimizes squared error.

Topics

- Probability and Statistics
- Probabilistic Graphical Models
- Message Passing Inference
- Parameter Learning

Directed Graphical Models

- Distribution factorized as product of conditionals via chain rule

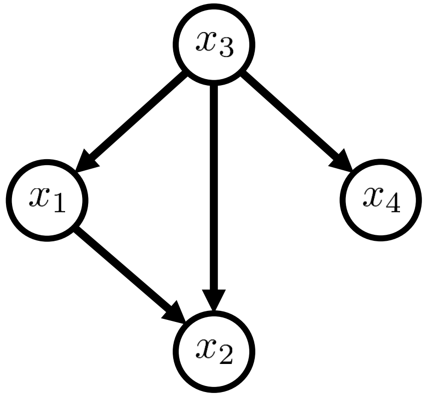
$$p(x_1, x_2, x_3, x_4) = p(x_3)p(x_1 \mid x_3)p(x_4 \mid x_1, x_3)p(x_2 \mid x_1, x_3, x_4)$$

- Choose ordering where terms simplify due to conditional independence

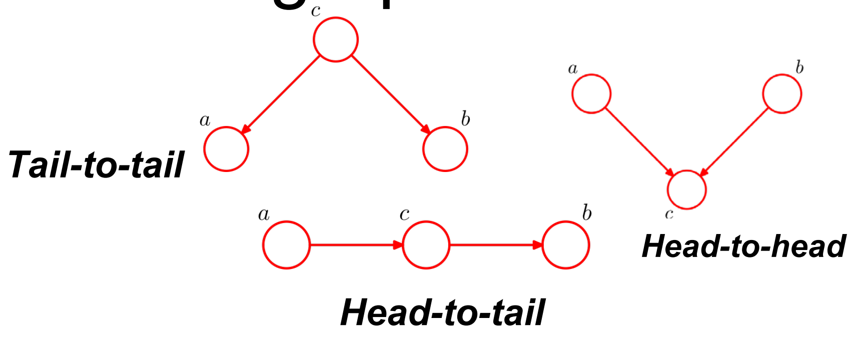
Eg. Suppose $x_4 \perp x_1 \mid x_3$ and $x_2 \perp x_4 \mid x_1$ then:

$$p(x) = p(x_3)p(x_1 \mid x_3)p(x_4 \mid x_3)p(x_2 \mid x_1, x_3)$$

- Directed graph encodes factorized distribution via conditional independence properties



- Test independence using canonical subgraphs:
- Straightforward simulation via **ancestral sampling**



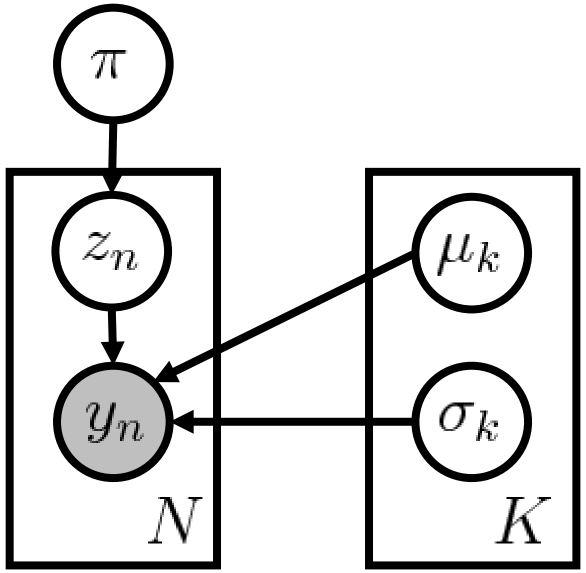
Example: Gaussian Mixture Model

Bayes nets are easily simulated via ancestral sampling…

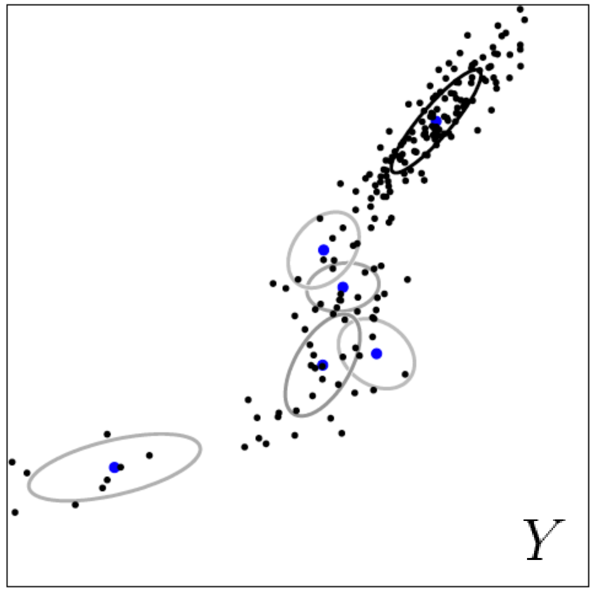
Probability Model

$$\pi \sim \text{Dirichlet}(\cdot)$$
$$\mu_k \sim \mathcal{N}(\cdot)$$
$$\sigma_k \sim \text{Inv-Gamma}(\cdot)$$
$$z_n \mid \pi \sim \text{Cat}(\pi)$$
$$y_n \mid z_n, \mu_{z_n}, \sigma_{z_n} \sim \mathcal{N}(\mu_{z_n}, \sigma_{z_n})$$

Bayes Net



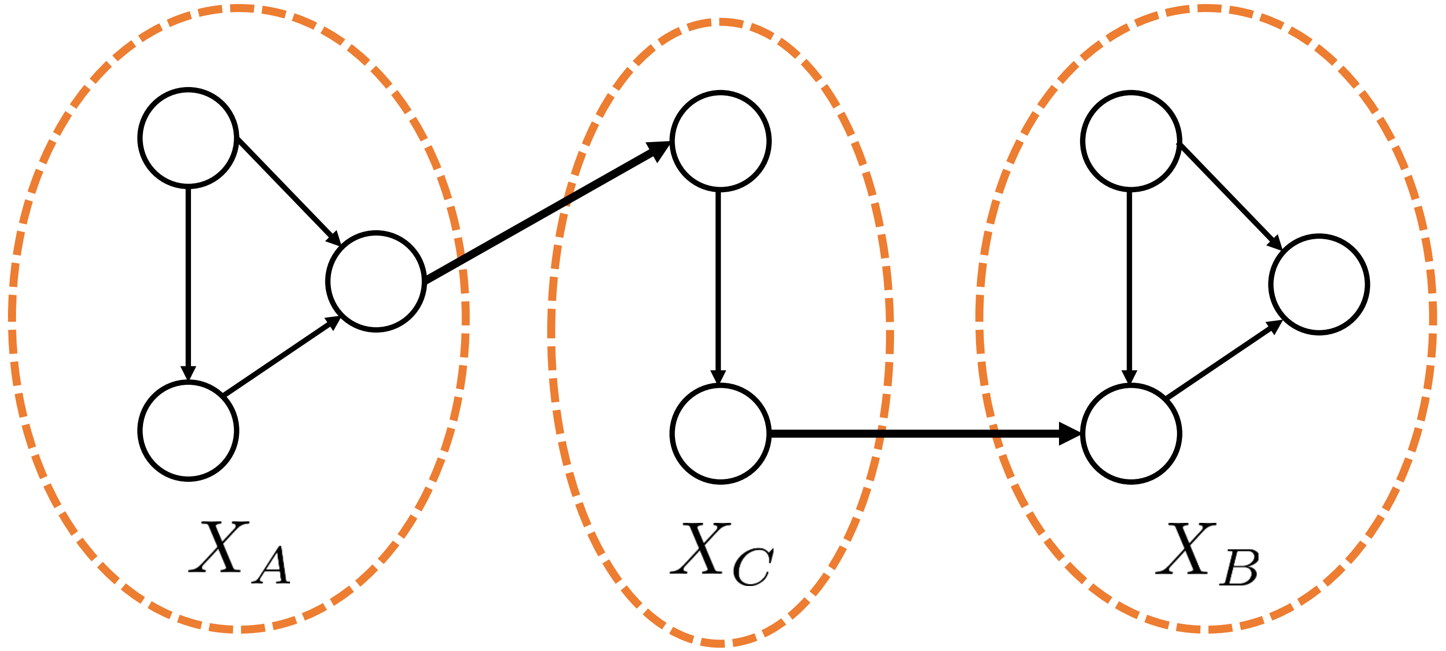
Joint Sample



Sample all nodes with no parents, then children, etc., to terminals. Can sample nodes at same level in parallel.

Bayes Ball Algorithm

To test if $X_A \perp X_B \mid X_C$ roll ball from *every node in* X_A …



If *any* ball reaches *any node* in X_B then

$$X_A \not\perp X_B \mid X_C$$

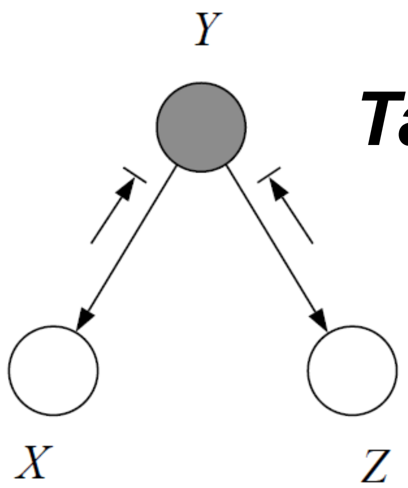
Otherwise:

$$X_A \perp X_B \mid X_C$$

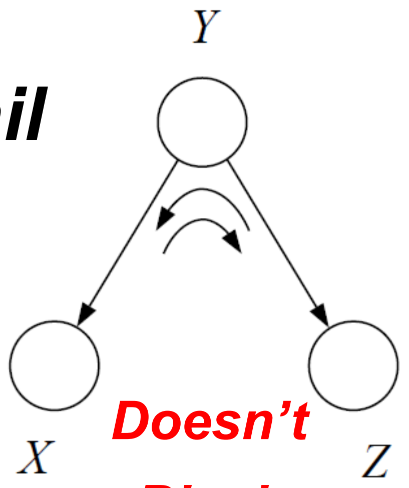
Tests for property of ***directed separation*** **(d-separation)**: if X_C separates / blocks X_A from X_B then $X_A \perp X_B \mid X_C$

Bayes Ball Algorithm

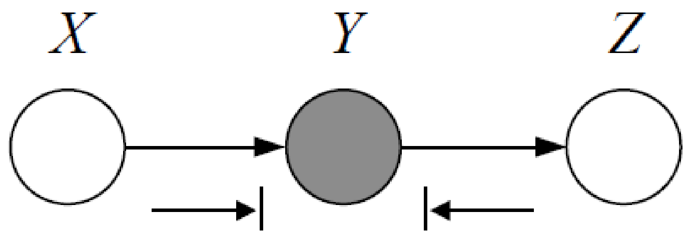
Tail-to-Tail



Blocks

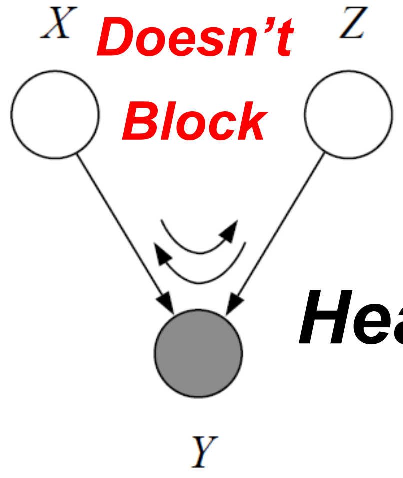


Doesn't Block



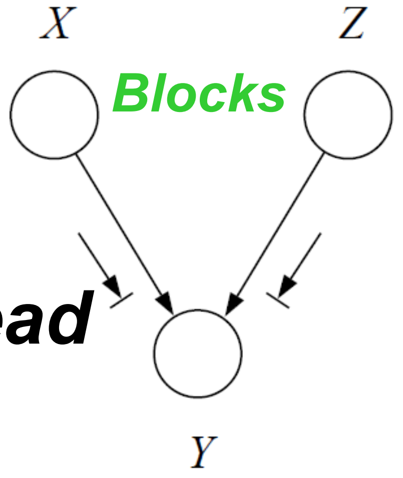
Blocks

Head-to-Tail

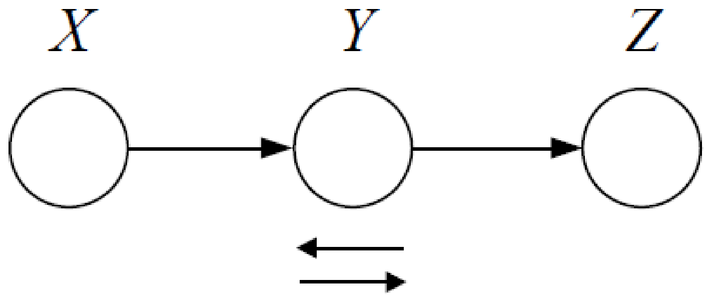


Doesn't Block

Head-to-Head



Blocks



Doesn't Block

Markov Blanket

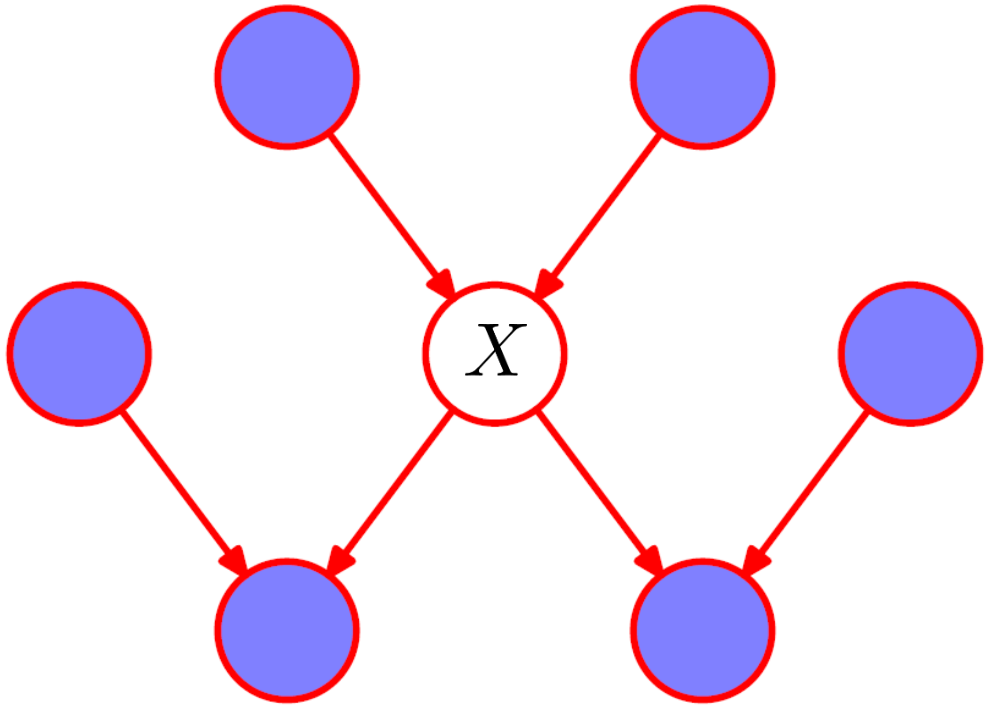
X *conditionally independent of all other nodes, given its Markov blanket*

Definition *A RV X with distribution p(x) that is Markov w.r.t. graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *has a* **Markov blanket** *given by:*

$$\mathrm{Mb}(X) = \mathrm{Pa}(X) \cup \mathrm{Ch}(X) \cup \mathrm{CoPa}(X)$$

For any $Y \notin \mathrm{Mb}(X)$:

$$X \perp Y \mid \mathrm{Mb}(X)$$



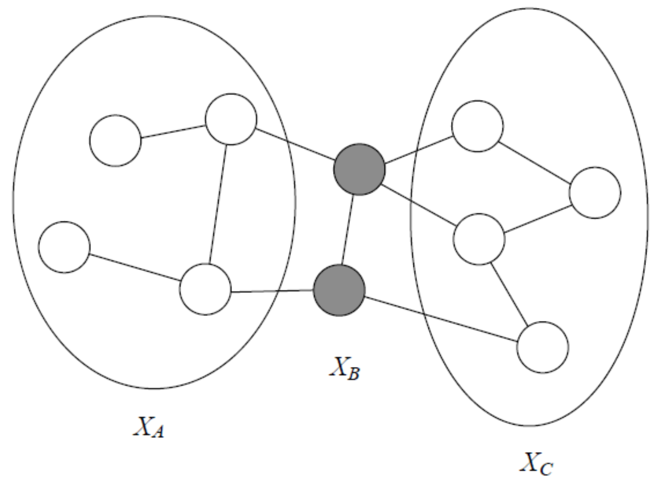
Markov blanket used to simplify inference and distribute computation (e.g. Gibbs sampler, variational inference, etc.)

Undirected Graphical Models

- Joint factorization as nonnegative factors (potentials) over subsets:

$$p(x) \propto \prod_{f \in \mathcal{F}} \psi_f(x_f)$$

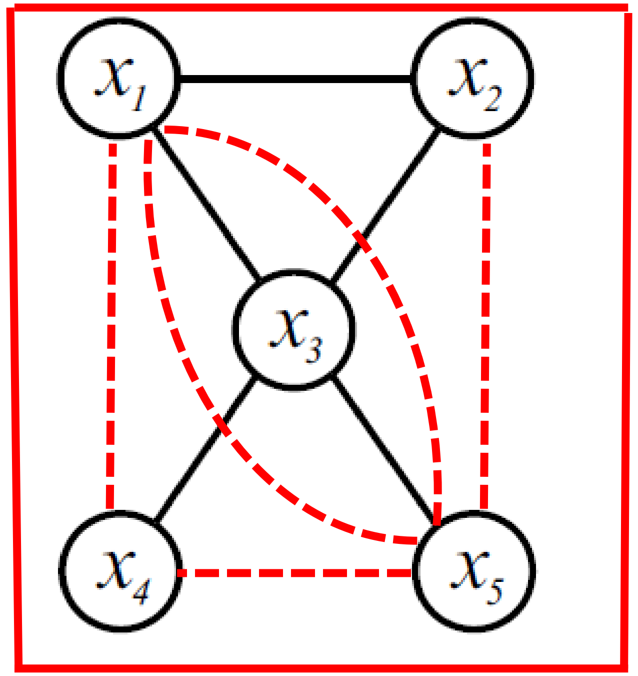
- Easier to specify models compared to Bayes nets since:
  - Factors do not need to be normalized conditional probabilities
  - May specify up to unknown normalization constant
- Easier to verify Markov independence via *separating sets*
- Factorization ambiguous in MRFs, but explicit in factor graphs (factor graphs generally preferred)
- Simulation is not easy in general. Can't do ancestral sampling.



Markov Random Fields (MRFs)

A factor $\psi_c(x_c)$ corresponds to a clique $c \in \mathcal{C}$ (fully connected subgraph) in the MRF

Complete Graph

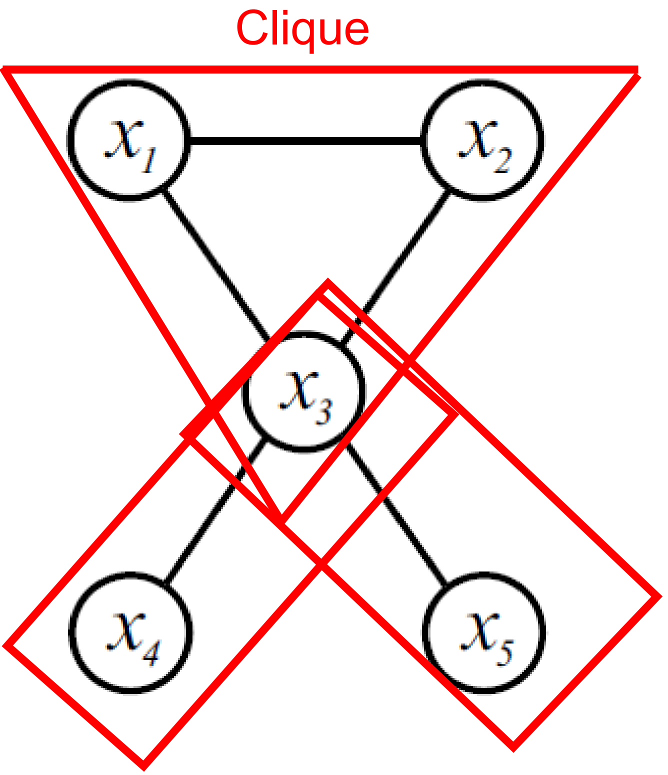


An MRF does not imply a unique factorization, for example all the following are "*valid*":

$$\psi(x_1, x_2, x_3, x_4, x_5)$$

Markov Random Fields (MRFs)

A factor $\psi_c(x_c)$ corresponds to a clique $c \in \mathcal{C}$ (fully connected subgraph) in the MRF



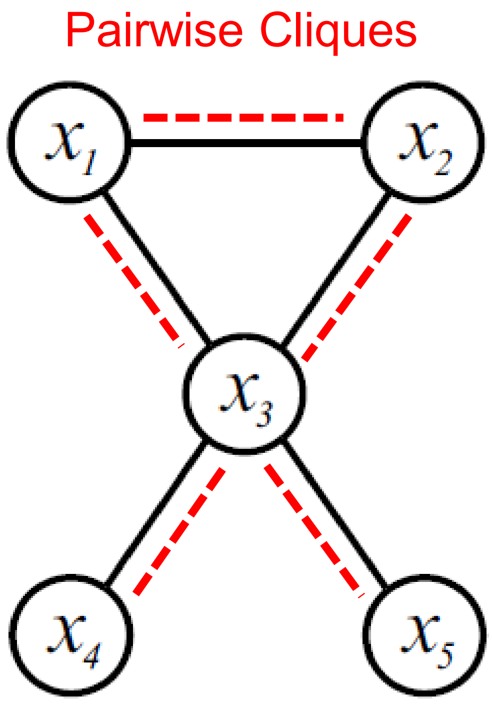
An MRF does not imply a unique factorization, for example all the following are "*valid*":

$$\psi(x_1, x_2, x_3, x_4, x_5)$$

$$\psi(x_1, x_2, x_3)\psi(x_3, x_4)\psi(x_3, x_5)$$

Markov Random Fields (MRFs)

A factor $\psi_c(x_c)$ corresponds to a clique $c \in \mathcal{C}$ (fully connected subgraph) in the MRF



An MRF does not imply a unique factorization, for example all the following are "*valid*":

$$\psi(x_1, x_2, x_3, x_4, x_5)$$

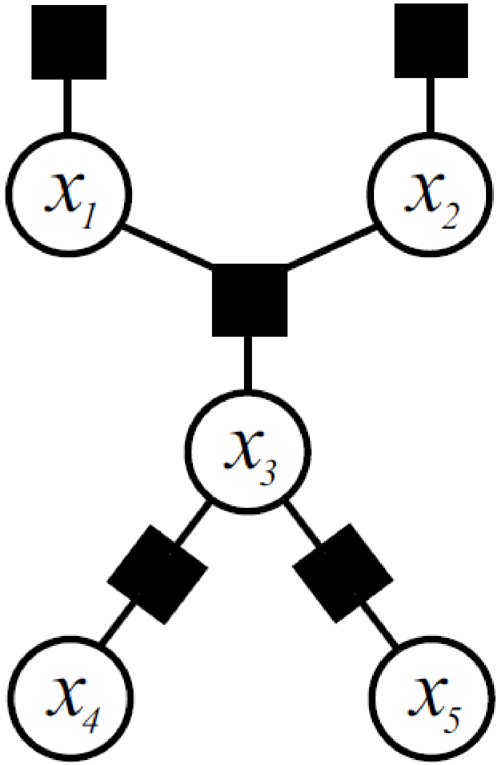
$$\psi(x_1, x_2, x_3)\psi(x_3, x_4)\psi(x_3, x_5)$$

$$\psi(x_1, x_2)\psi(x_2, x_3)\psi(x_1, x_3)\psi(x_3, x_4)\psi(x_3, x_5)$$

A factorization is *valid* if it satisfies the *Global Markov property*, defined by conditional independencies

Factor Graphs

A *hypergraph* $\mathcal{H} = (\mathcal{V}, \mathcal{F})$ where a *hyperedge* $f \in \mathcal{F}$ is a subset of vertices $f \subset \mathcal{V}$.



Factor node for each product term in the joint factorization:

Graphical model makes factorization explicit

$$p(x) \propto \prod_{f \in \mathcal{F}} \psi_f(x_f)$$

where $x_f = \{x_i : i \in f\}$ the set of variables in factor *f*. For example:

$$\psi(x_1)\psi(x_2)\psi(x_1, x_2, x_3)\psi(x_3, x_4)\psi(x_3, x_5)$$

Conditional Independence (Undirected)

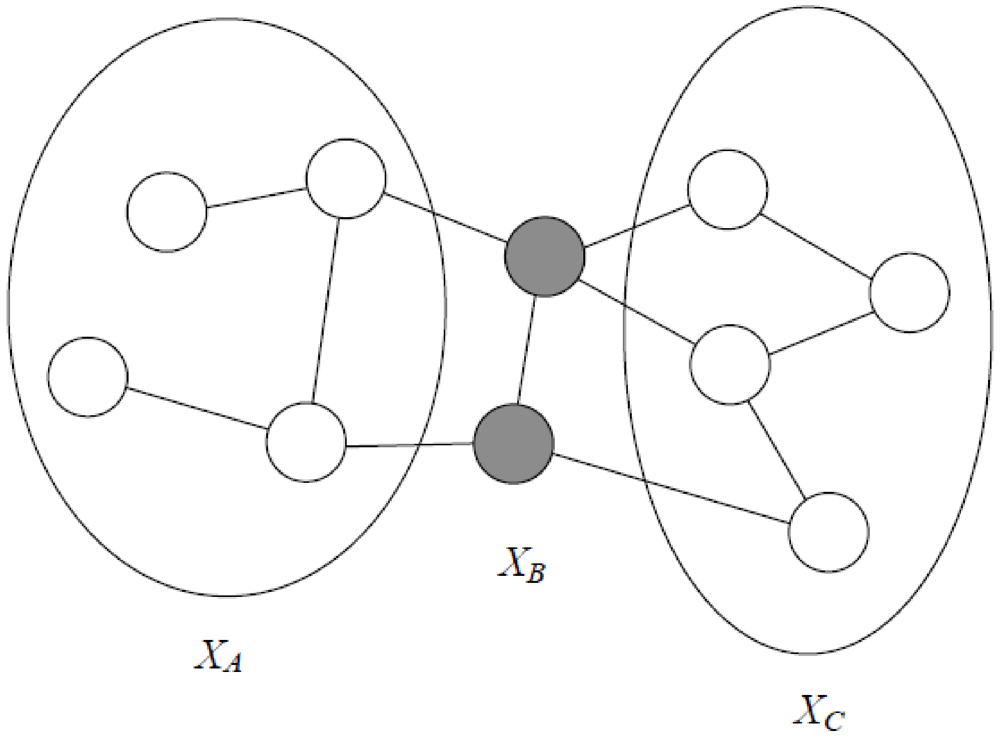
We say x_A and x_C are *independent* or $x_A \perp\!\!\!\perp x_C$ if:

$$p(x_A, x_C) = p(x_A)p(x_C)$$

We say they are *conditionally independent* or $x_A \perp\!\!\!\perp x_C \mid x_B$ if:

$$p(x_A, x_C \mid x_B) = p(x_A \mid x_B)p(x_C \mid x_B)$$

Def. We say $p(x)$ is *globally Markov* w.r.t. $\mathcal{G}$ if $x_A \perp\!\!\!\perp x_C \mid x_B$ in any separating set of $\mathcal{G}$.



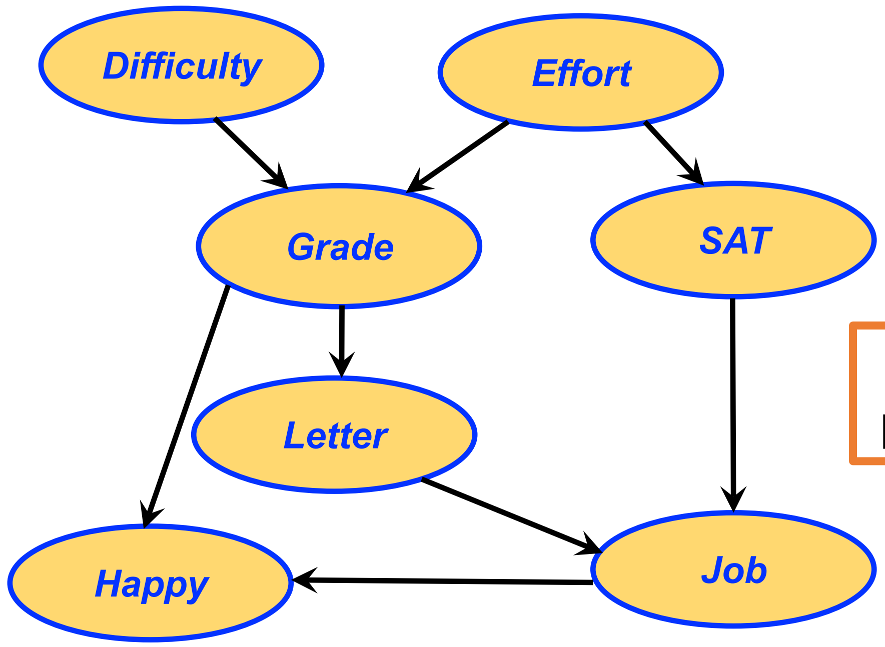
Conditional independence in undirected graphical models is defined by separating sets

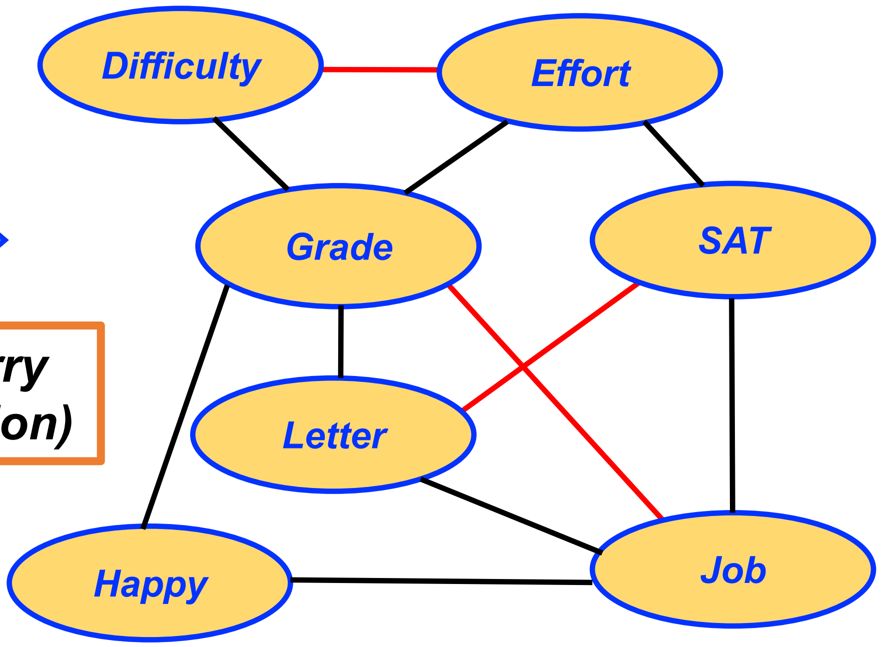
[Source: Michael I. Jordan]

Topics

- Probability and Statistics
- Probabilistic Graphical Models
- Message Passing Inference
- Parameter Learning

Bayes Net → MRF





Added edges *marry* parents (*moralization*)

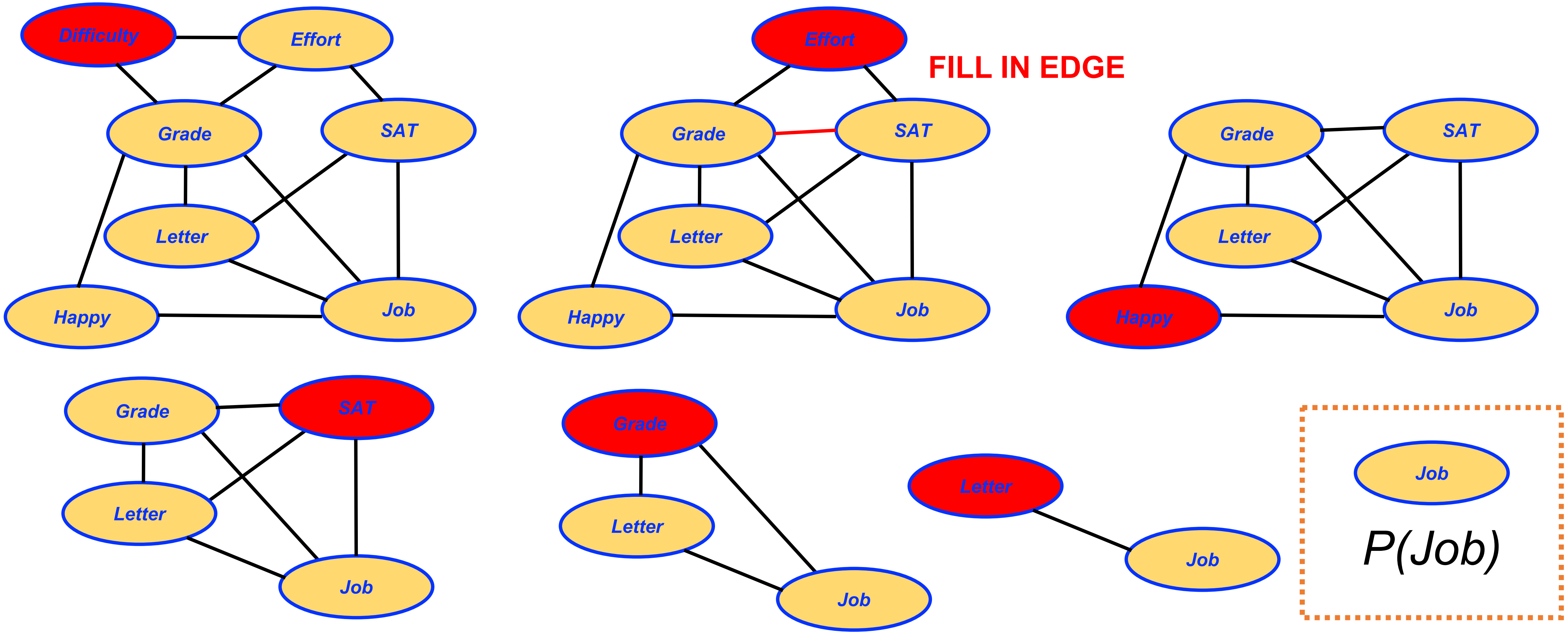
$$P(\cdot) = P(D)P(E)P(G \mid D, E)P(S \mid E)P(L \mid G)P(J \mid S, L)P(H \mid J, G)$$

Drop local normalization

$$P(\cdot) \propto \psi(D)\psi(E)\psi(G, D, E)\psi(S, E)\psi(L, G)\psi(J, S, L)\psi(H, J, G)$$

Variable Elimination

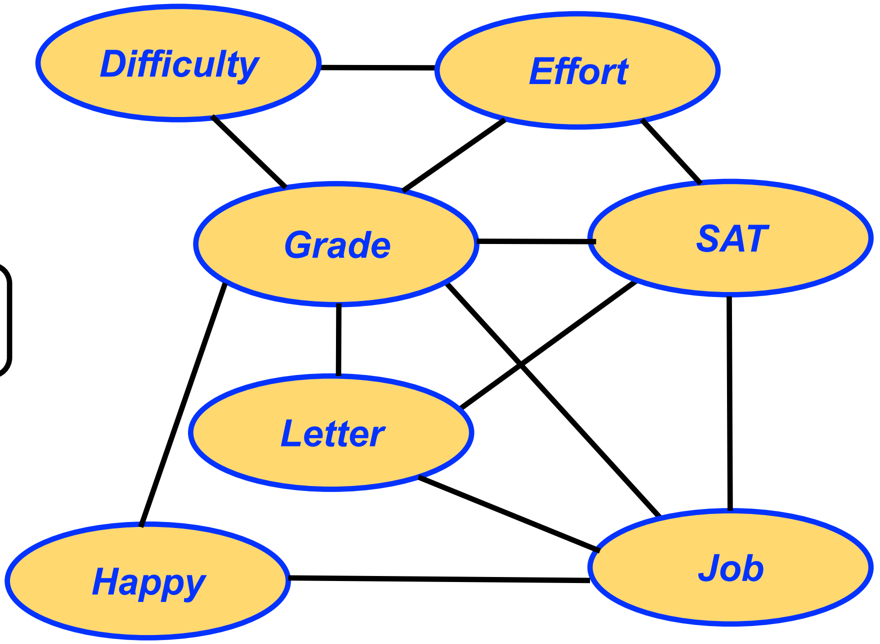
Recall variable elimination sequentially marginalizes out variables…



Computational Complexity

Clique Tree

$$\begin{tikzcd}\boxed{D,E,G} \arrow[r, dash, "{E,G}"] & \boxed{E,G,S} \arrow[r, dash, "{G,S}"] & \boxed{G,S,L,J} \arrow[r, dash, "{G,J}"] & \boxed{H,G,J}\end{tikzcd}$$



Elimination order $\prec$ induces graph with maximal cliques $\mathcal{C}(\prec)$ and *width:*

$$w(\prec) = \max_{c \in \mathcal{C}(\prec)} |c| - 1$$

- Complexity of variable elimination is $\mathcal{O}(K^{w(\prec)+1})$
- Lowest complexity given by the *treewidth:*

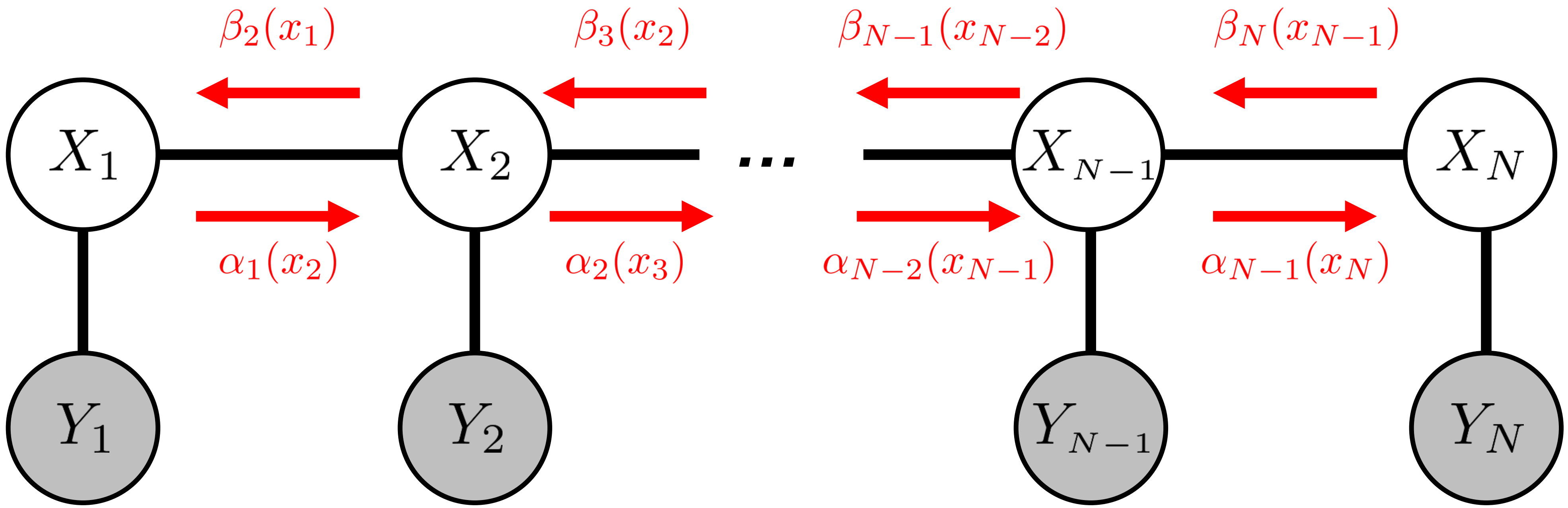
$$w^* = \min_{\prec} \max_{c \in \mathcal{C}(\prec)} |c| - 1$$

It is NP-hard to compute treewidth, and therefore an optimal elimination order (of course…)

Variable Elimination Summary

- Variable elimination allows computation of marginals / conditionals
- Algorithm is valid for **any graphical model**
- Suffices to show variable elimination for MRFs, since Bayes nets → MRFs by *moralization*
- Worst-case complexity is dependent on elimination order, and is **exponential** in number of variables
- Optimal ordering = treewidth, is NP-hard to compute

Forward-Backward Algorithm



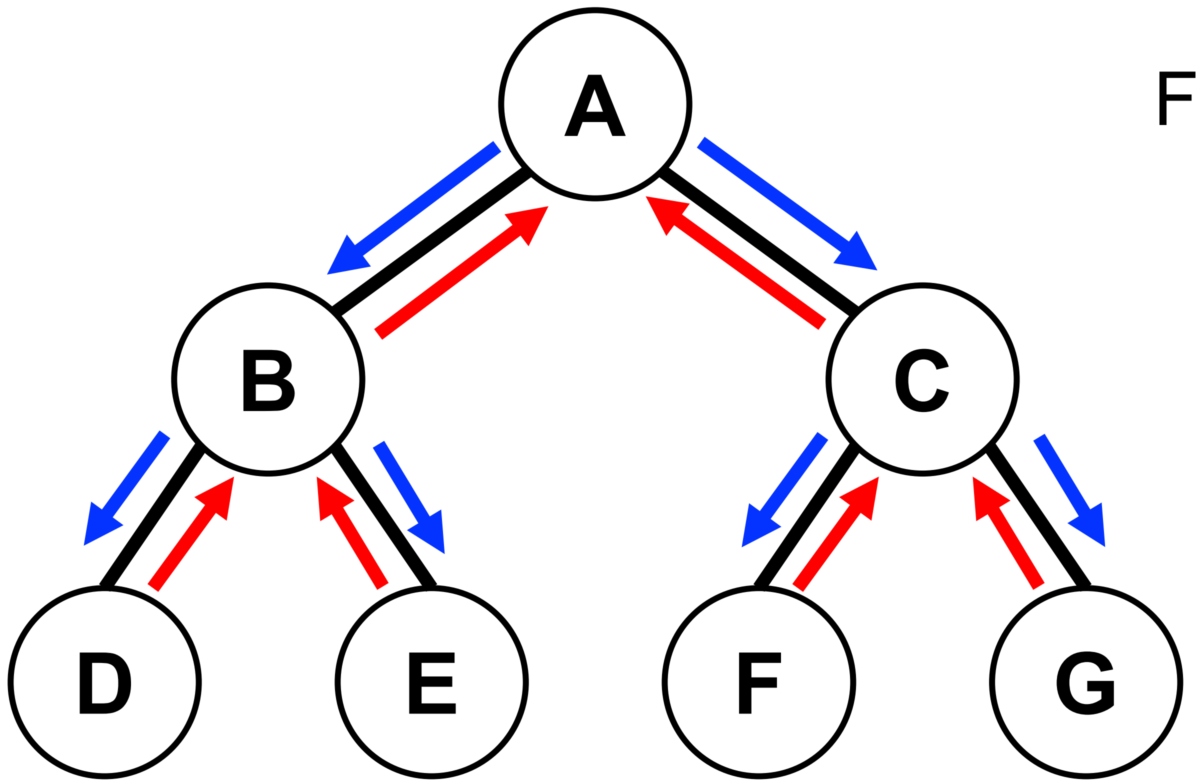
Forward message gives the *filtered posterior*:

$$\alpha_{n-1}(x_n) \propto p(y_1, \ldots, y_n, x_n) \propto p(x_n \mid y_1, \ldots, y_n)$$

Smoothed posterior incorporates all observations:

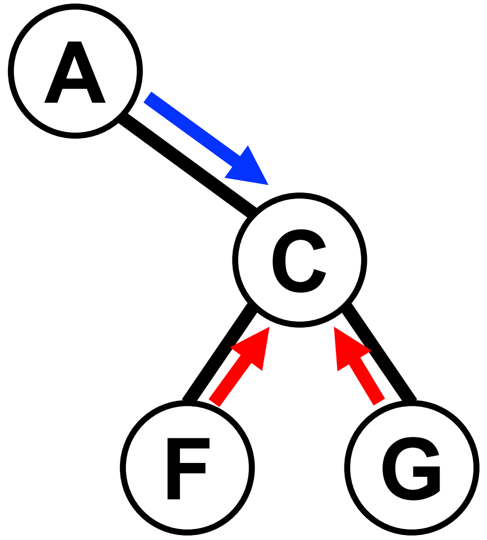
$$\begin{aligned} p(x_n \mid y_1, \ldots, y_N) &\propto p(x_n \mid y_1, \ldots, y_n) p(y_{n+1}, \ldots, y_N \mid x_n) \\ &\propto \alpha_{n-1}(x_n) \beta_{n+1}(x_n) \end{aligned}$$

Sum-Product Belief Propagation



Forward-Backward extends to *any tree-structured pairwise MRF*

Marginal given by *incoming* messages (e.g. node C):

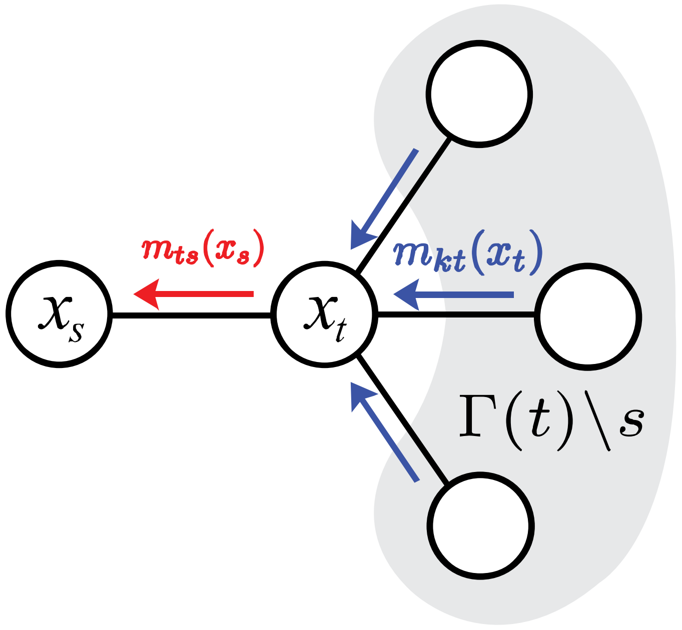


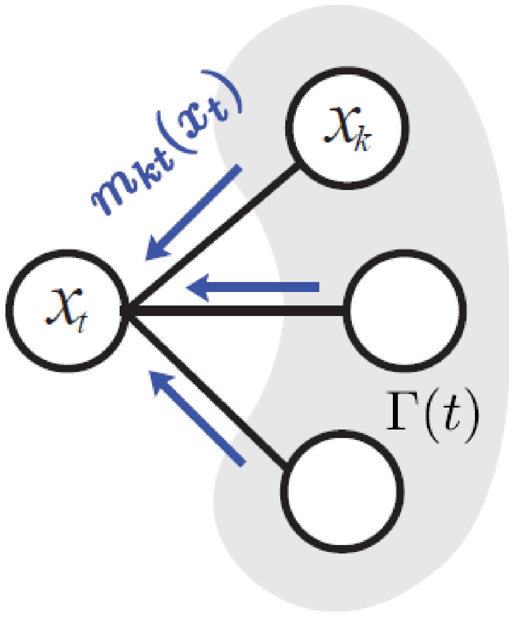
Pass messages from leaves-to-root, then root-to-leaves

$$p(C) \propto \psi(C) m_A(C) m_F(C) m_G(C)$$

Pairwise MRF Sum-Product Belief Propagation

Message updates depend only on Markov blanket…



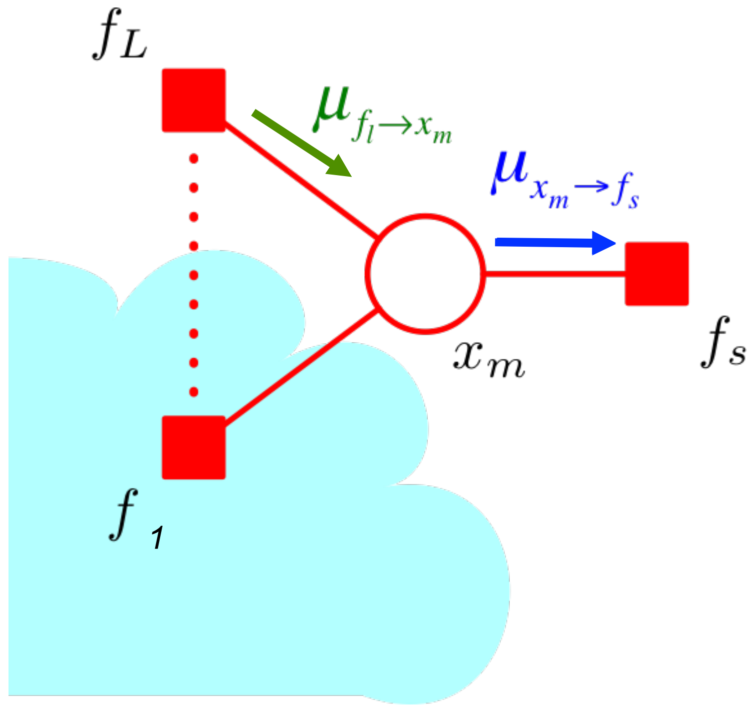


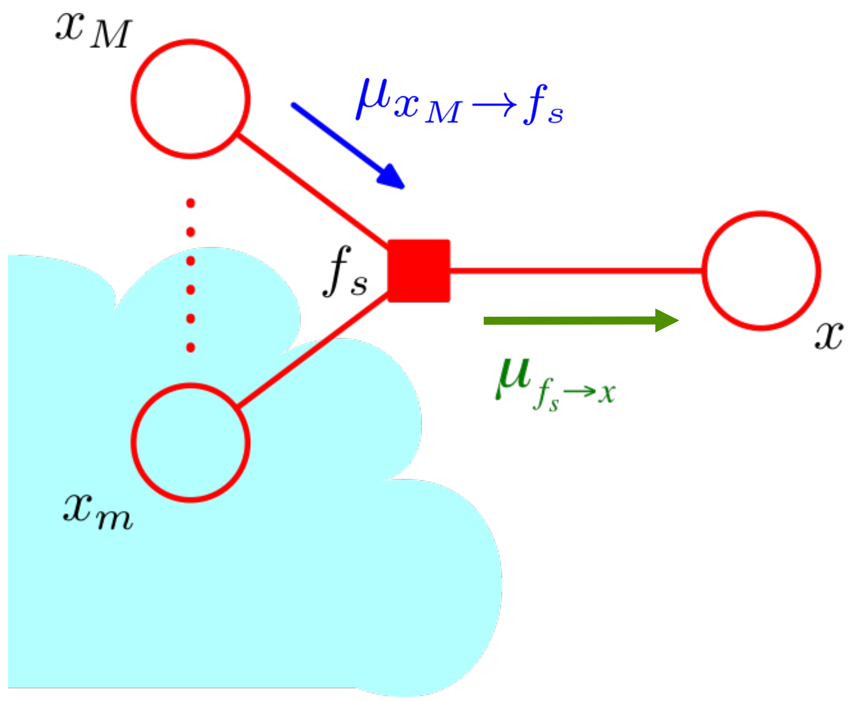
Message $m_{ts}(x_s) = \sum_{x_t} \psi_{st}(x_s, x_t)\psi_t(x_t) \prod_{k \in \Gamma(t) \backslash s} m_{kt}(x_t)$

Marginal $p(x_t) \propto \psi_t(x_t) \prod_{k \in \Gamma(t)} m_{kt}(x_t)$

Messages involve a **sum** over **products**, hence the name “sum-product algorithm”

Factor Graph Sum-Product Belief Propagation





Variable node x_m gathers messages, $\mu_{f_l \to x_m}$, and sends

$$\mu_{x_m \to f_s}(x_m) = \prod_{l \ni f_l \in n(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

Factor f_s gathers messages $\mu_{x_m \to f_s}(x_m)$, and sends

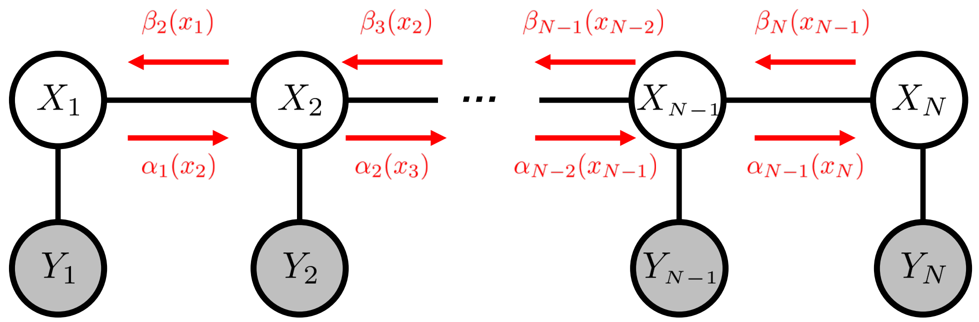
$$\mu_{f_s \to x}(x) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} f_s(x, x_1, x_2, \ldots, x_M) \prod_{m \in ne(f_s) \setminus x} \mu_{x_m \to f_s}(x_m)$$

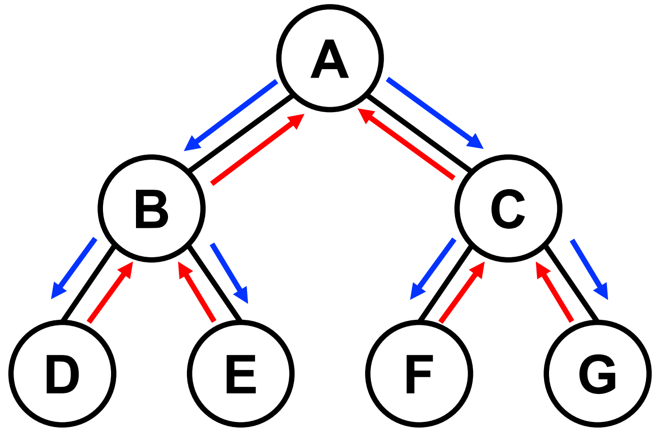
Marginal is product of incoming factor-to-variable messages:

$$p(x_m) \propto \prod_{f_l \in ne(x_m)} \mu_{f_l \to x_m}(x_m)$$

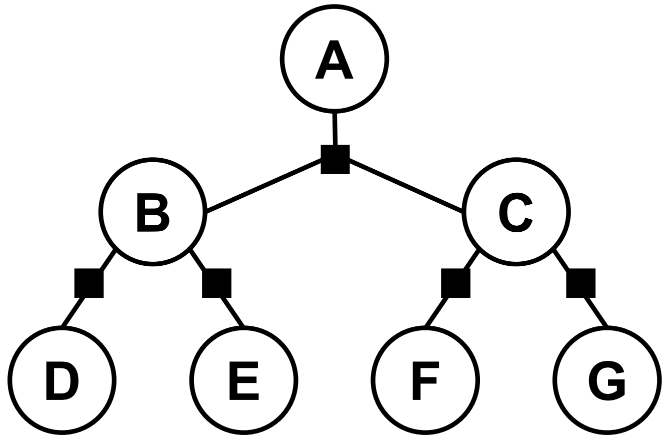
Message Passing Inference Summary

Forward-backward algorithm yields efficient marginal inference on HMM graph





Sum-product belief propagation generalizes marginal inference to tree-structured MRFs



Max-product / max-sum yields maximum a posteriori (MAP) inference in any tree-structured model

Viterbi decoder is special case for HMM

And factor graphs

Variable Elimination

Two major limitations of variable elimination:

1. Computation **exponential** in size of the largest intermediate factor (equivalently, largest clique in clique tree)

2. Computation is not reused for computing a series of marginals

E.g. Suppose we use variable elimination to compute a marginal on an **HMM** with T nodes, each being K-valued

- It takes $\mathcal{O}(TK^2)$ time to compute a single marginal
- It takes $\mathcal{O}(T^2K^2)$ time to compute **all marginals**
- We know forward-backward computes all marginals in $\mathcal{O}(TK^2)$

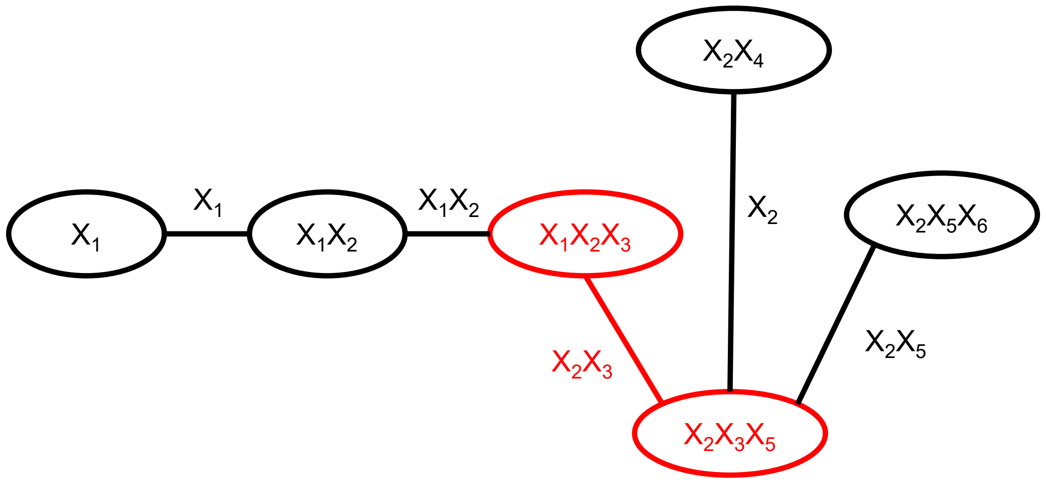
Marginal Inference Algorithms

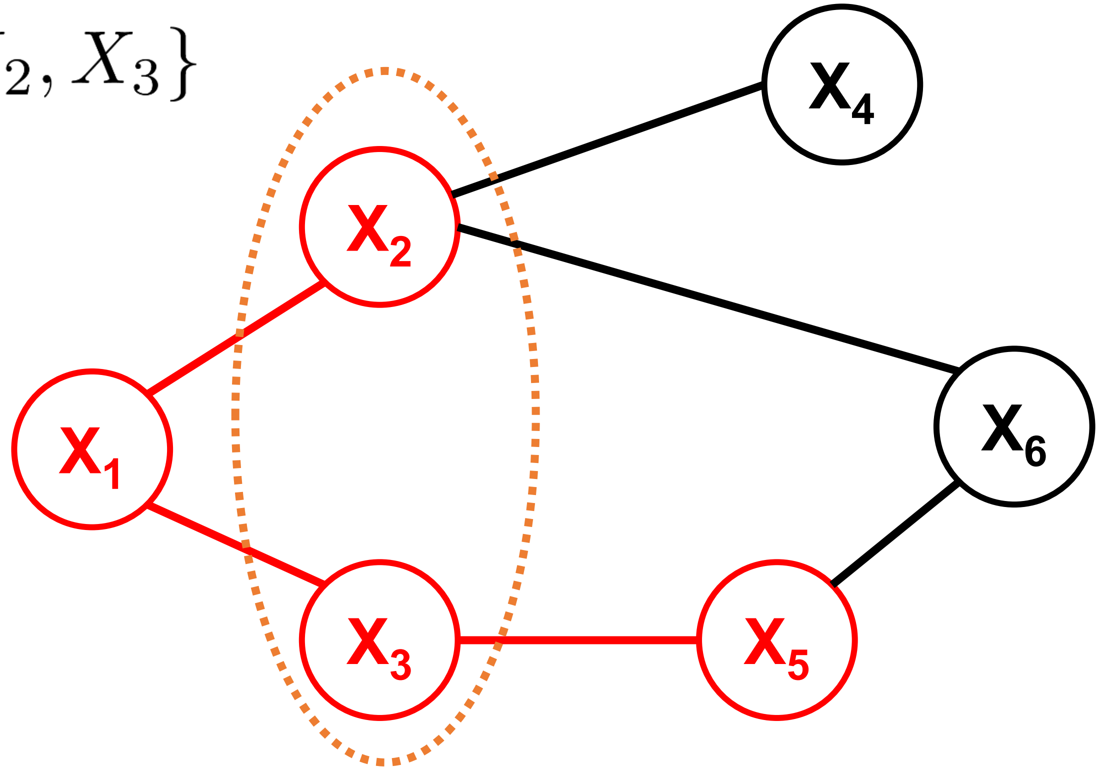
	One Marginal	*All Marginals*
Tree	Elimination applied to leaves of tree	Belief Propagation (BP) or sum-product algorithm
Graph	Variable Elimination	Junction Tree Algorithm BP on a junction tree (special clique tree)

Junction Tree

Clique tree edges are separator sets in original MRF…so clique tree encodes conditional independencies

$$X_1 \perp X_5 \mid \{X_2, X_3\}$$



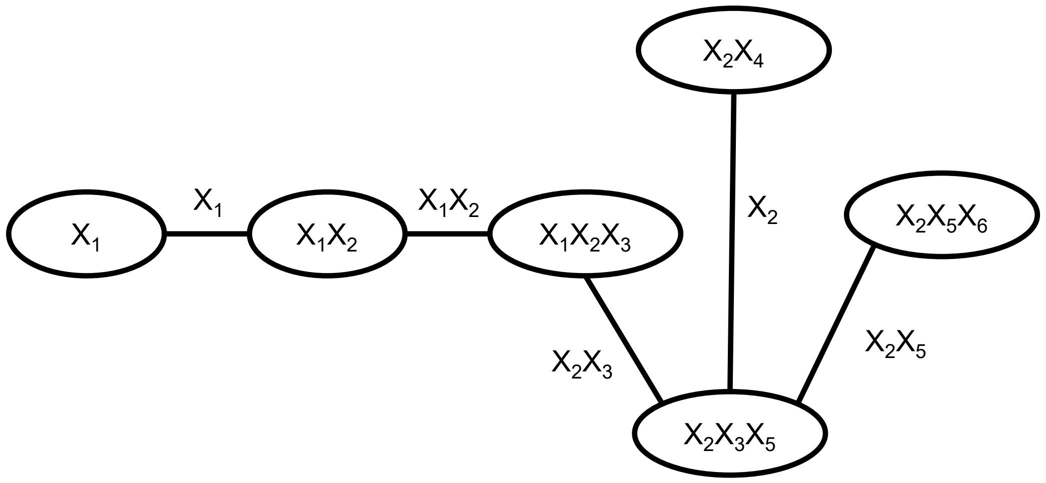


Theorem A clique tree resulting from variable elimination **satisfies the running intersection property** and is thus **a junction tree**

Junction Tree

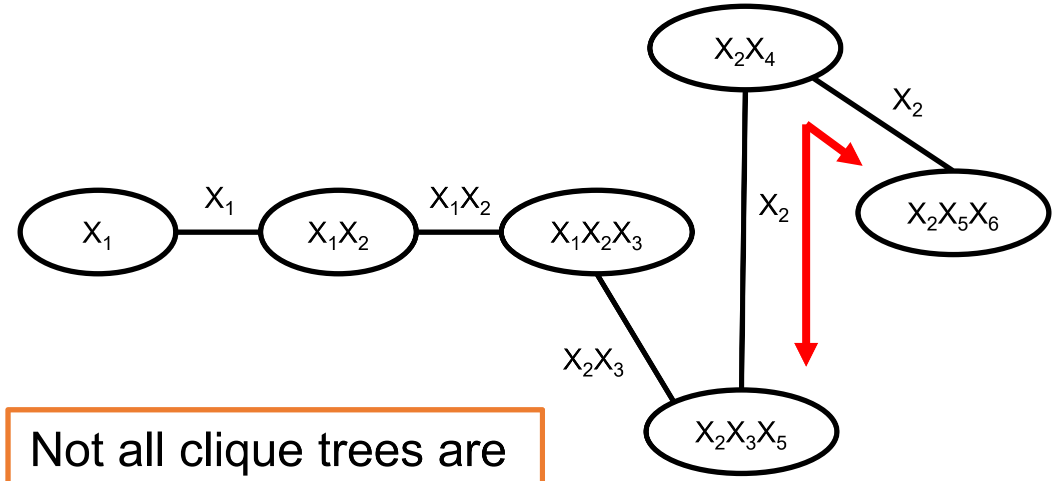
Definition (Running intersection) For any pair of clique nodes V,W all cliques on the *unique path* between V and W contain shared variables

Junction Tree



Not A Junction Tree

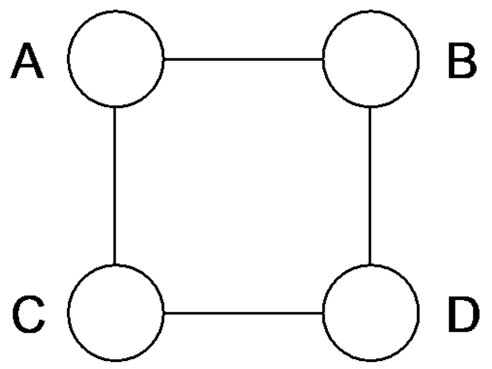
$$\{X_2, X_3, X_5\} \cap \{X_2, X_5, X_6\} = \{X_2, X_5\}$$

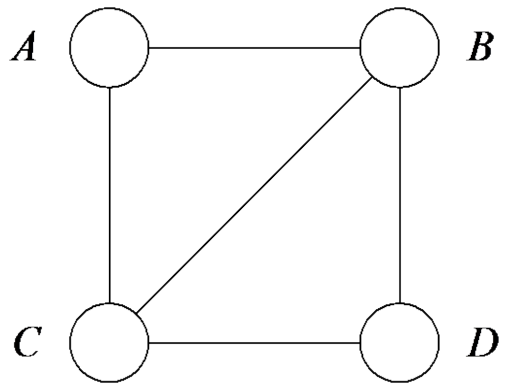


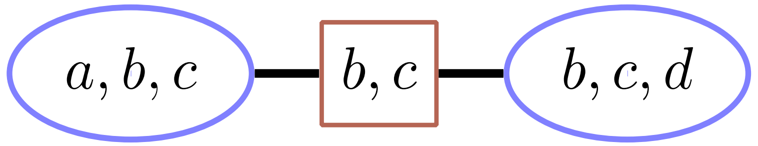
Not all clique trees are junction trees

Theorem A clique tree resulting from variable elimination **satisfies the running intersection property** and is thus **a junction tree**

Junction Trees and Triangulation







- A *chord* is an edge connecting two non-adjacent nodes in some *cycle*
- A cycle is *chordless* if it contains no chords
- A graph is *triangulated (chordal)* if it contains no chordless cycles of length 4 or more

Theorem: The maximal cliques of a graph have a corresponding junction tree *if and only if* that undirected graph is triangulated

Lemma: *For a non-complete triangulated graph with at least 3 nodes, there is a decomposition of the nodes into disjoint sets A, B, S such that S separates A from B, and S is complete.*

- ➢ Key induction argument in constructing junction tree from triangulation
- ➢ Implies existence of *elimination ordering which introduces no new edges*

Induced Graph

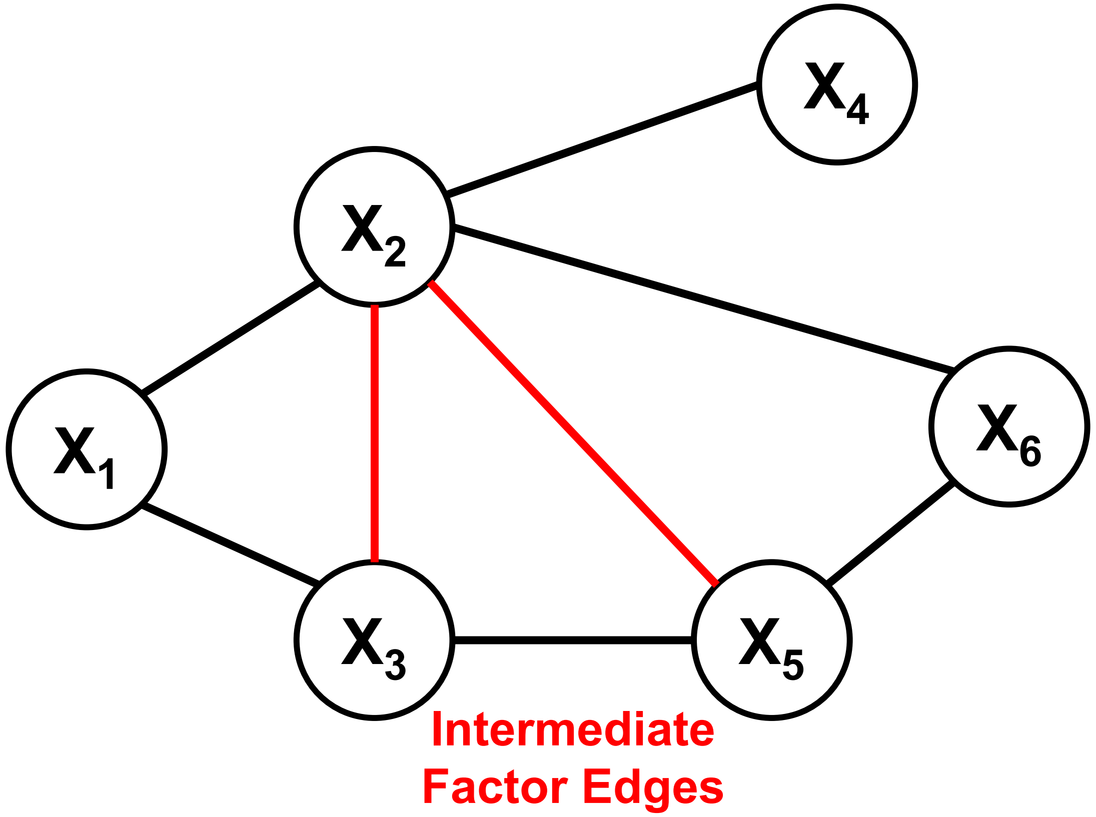
Recall the **induced graph** is the union over intermediate graphs from running variable elimination

The induced graph **is chordal** thus:

- Maximal cliques of the induced graph form a junction tree
- It admits an elimination ordering that introduces *no new edges*

Logic of junction tree algorithm:

1. Triangulate the graph
    a. Implies a junction tree
    b. Induces an elimination order
2. Run sum-product BP on junction tree to compute **all clique marginals**



Loopy Belief Propagation (sum-product)

Initialize Messages

Constant: $m^0_{st}(x_t) = \text{const.}$

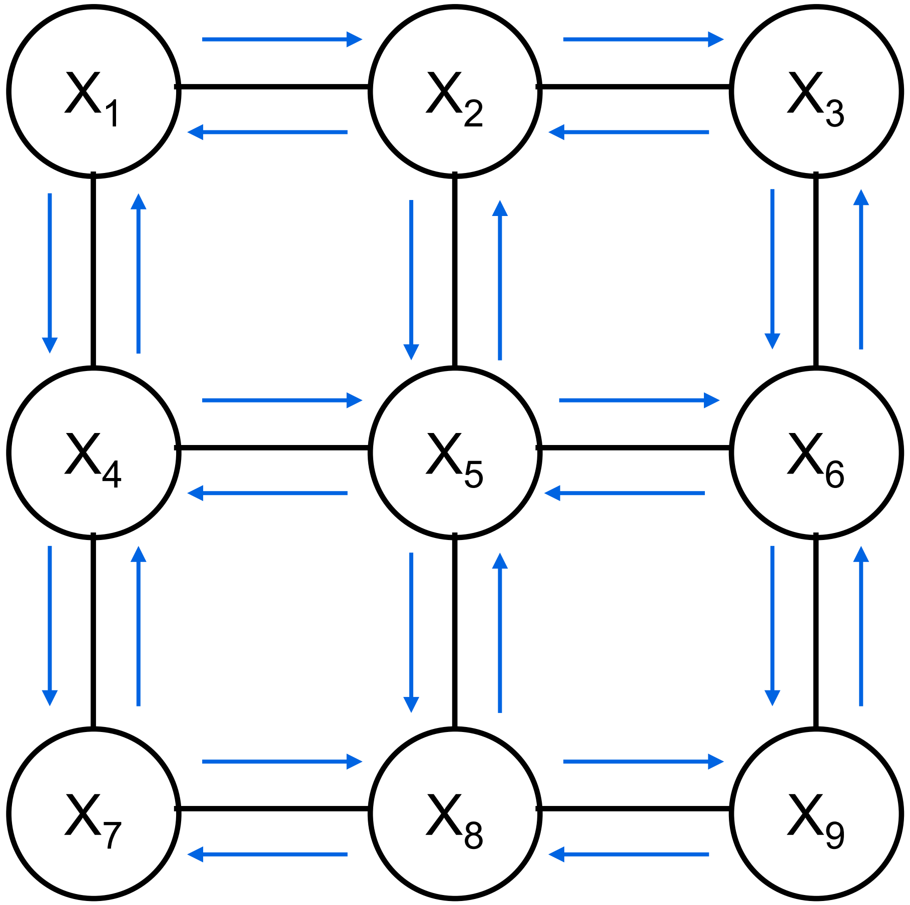
Random: $m^0_{st}(x_t) \sim U([0,1])$

Parallel (Synchronous) Updates

At iteration i update *all messages in parallel* using current messages m^{i-1} from previous iteration:

$$m^i_{st}(x_t) = \sum_{x_s} \psi_{st}(x_s, x_t) \prod_{k \in \Gamma(s) \setminus t} m^{i-1}_{ks}(x_s)$$

- Store, both, the *previous* messages (from iteration i-1) and *current* messages (from iteration i)
- Many convergence results assume parallel updates



Loopy Belief Propagation (sum-product)

Initialize Messages

Constant: $m^0_{st}(x_t) = \text{const.}$

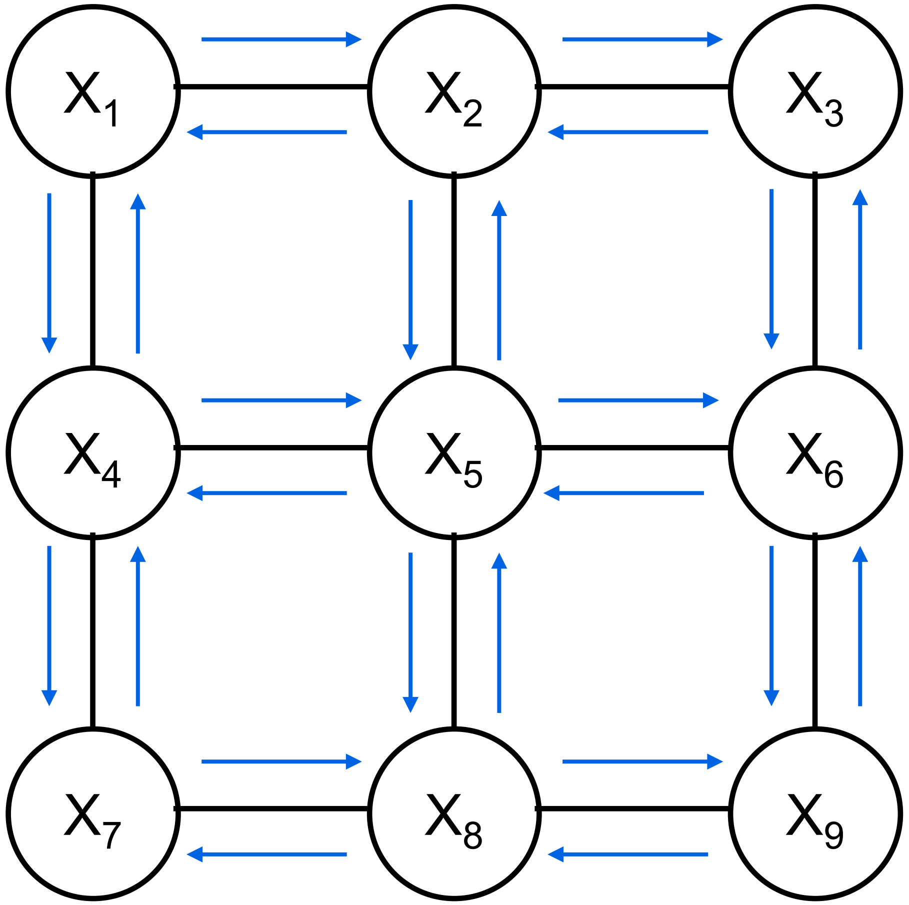
Random: $m^0_{st}(x_t) \sim U([0,1])$

Asynchronous (Sequential) Updates

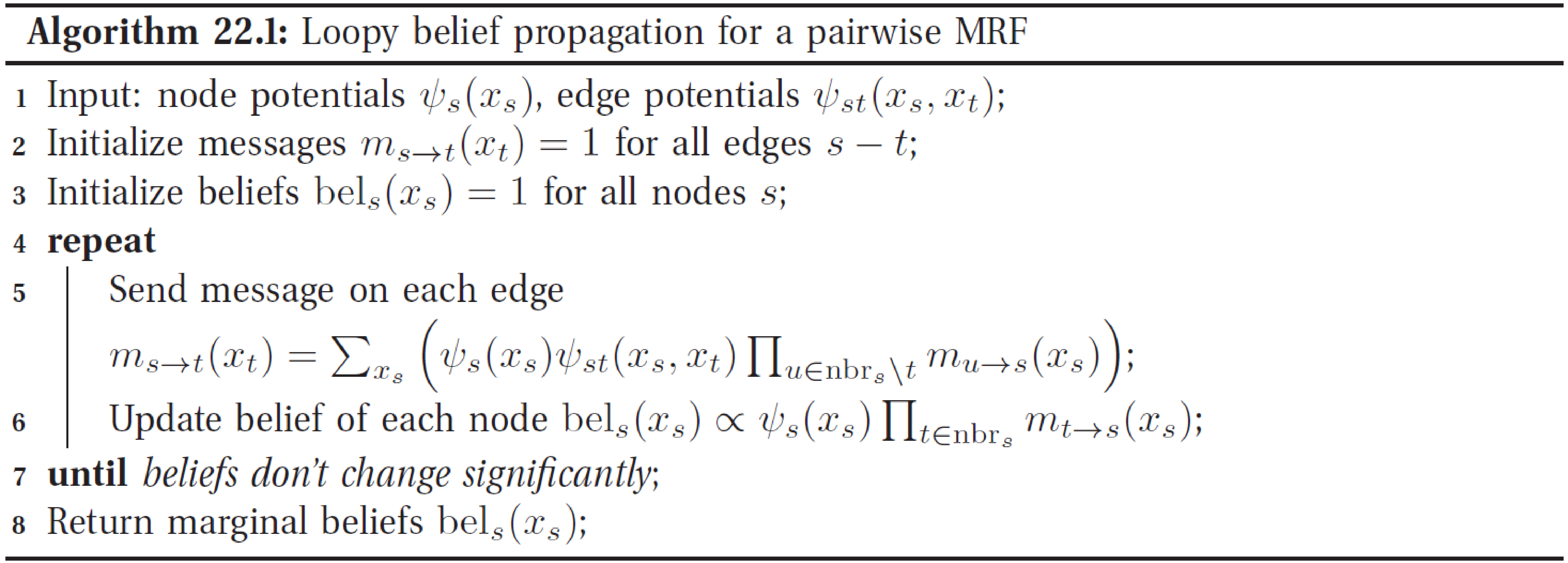
Choose an ordering of nodes and update using the latest available messages:

$$m_{st}(x_t) = \sum_{x_s} \psi_{st}(x_s, x_t) \prod_{k \in \Gamma(s) \setminus t} m_{ks}(x_s)$$

- Simplifies updates since only need to keep track of one copy of messages
- Makes parallel processing trickier



Pseudocode from Murphy’s Textbook



Algorithm 22.1: Loopy belief propagation for a pairwise MRF

```
Input: node potentials $\psi_s(x_s)$, edge potentials $\psi_{st}(x_s, x_t)$;
Initialize messages $m_{s \to t}(x_t) = 1$ for all edges $s - t$;
Initialize beliefs $\mathrm{bel}_s(x_s) = 1$ for all nodes $s$;
repeat
    Send message on each edge
    $m_{s \to t}(x_t) = \sum_{x_s} \left( \psi_s(x_s) \psi_{st}(x_s, x_t) \prod_{u \in \mathrm{nbr}_s \setminus t} m_{u \to s}(x_s) \right)$;
    Update belief of each node $\mathrm{bel}_s(x_s) \propto \psi_s(x_s) \prod_{t \in \mathrm{nbr}_s} m_{t \to s}(x_s)$;
until beliefs don't change significantly;
Return marginal beliefs $\mathrm{bel}_s(x_s)$;
```

Loopy BP Convergence

Loopy BP works well empirically, but there are no guarantees:

- Not guaranteed to converge in general graphs
- BP marginal *beliefs* are **approximations**
- Empirically, when LBP converges it does so quickly and with good approximations

Convergence based on change in messages / marginal approximations:

$$\rho(m^{\text{old}}, m^{\text{current}}) < \epsilon \qquad \text{or} \qquad \rho(\text{bel}^{\text{old}}, \text{bel}^{\text{current}}) < \epsilon$$

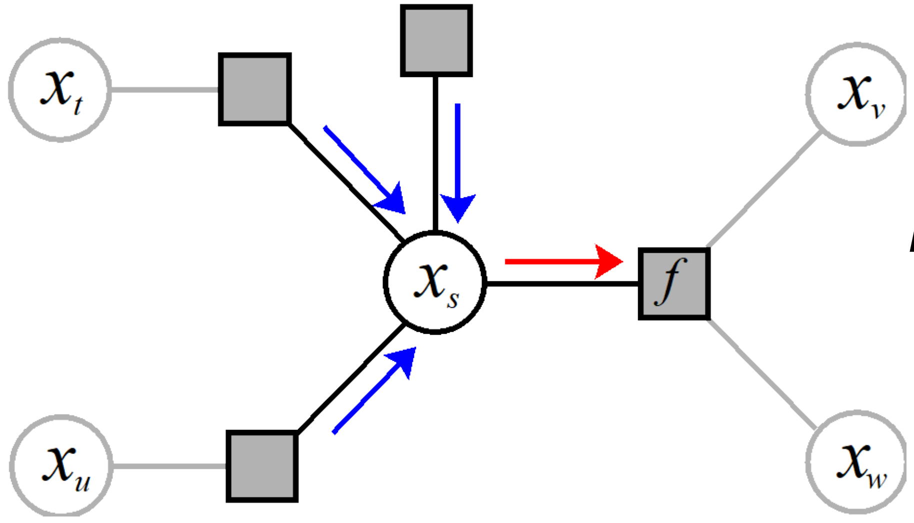
Typical convergence measures are:

Max change: $\rho(m^{\text{old}}, m^{\text{current}}) = \max \{|m^{\text{old}} - m^{\text{current}}|\}$

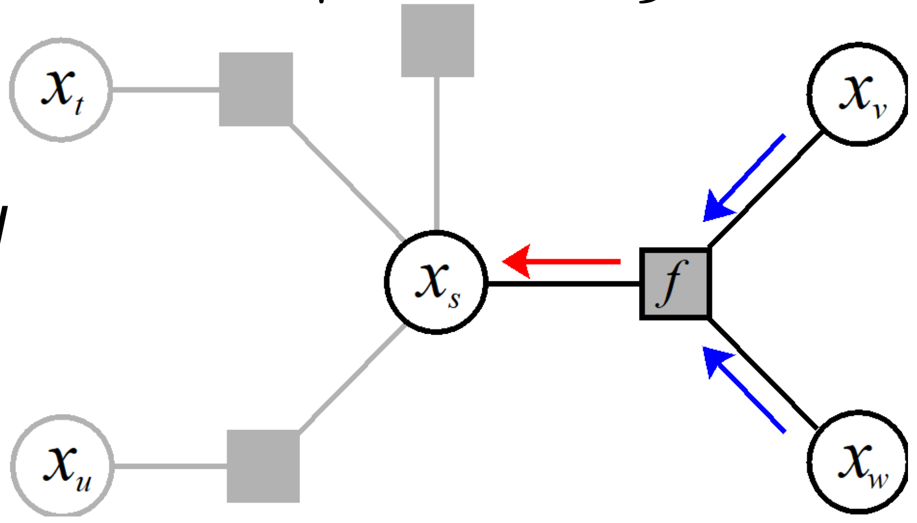
Total change: $\rho(m^{\text{old}}, m^{\text{current}}) = \sum |m^{\text{old}} - m^{\text{current}}|$

Loopy BP on Factor Graphs

Set of *neighbors* of node *s*: $\Gamma(s) = \{f \in \mathcal{F} \mid s \in f\}$



Loopy BP:
Message updates can be iteratively computed on graphs with cycles.
But marginals not guaranteed correct!



$$\bar{m}_{sf}(x_s) = \prod_{g \in \Gamma(s) \setminus f} m_{gs}(x_s) \propto \frac{p_s(x_s)}{m_{fs}(x_s)}$$

$$m_{fs}(x_s) = \sum_{x_{f \setminus s}} \psi_f(x_f) \prod_{t \in f \setminus s} \bar{m}_{tf}(x_t)$$

Marginal Distribution of Each Variable:

$$p_s(x_s) \propto \prod_{f \in \Gamma(s)} m_{fs}(x_s)$$

Marginal Distribution of Each Factor:
Clique of variables linked by factor.

$$p_f(x_f) \propto \psi_f(x_f) \prod_{s \in f} \bar{m}_{sf}(x_s)$$

Message Passing Inference Summary

- Brute-force enumeration exponential regardless of graph
- Sum-Product BP
  - Exact inference in tree-structure graphs in $O(TK^2)$ time for T nodes, each taking K states
  - Reduces to Forward-Backward in HMMs
  - Same for Max-Product BP (reduces to Viterbi in HMMs)
- Variable elimination
  - Exact marginals in general graphs
  - Worst-case complexity exponential in size of largest clique
  - Need to rerun from scratch for each marginal
  - Complexity dependent on elimination order (NP-hard to optimize)

Message Passing Inference Summary

- Junction Tree Algorithm
  - Exact marginals in general graphs
  - Caches messages to compute all marginals
  - Worst-case complexity exponential in size of largest clique
  - Optimizing Jtree is NP-hard (corresponds to finding treewidth)

- Loopy BP
  - BP updates only depend on tree-structured Markov blanket
  - **Approximate** inference in loopy graphs
  - No guarantees, but works well empirically in many instances
  - Some techniques to improve convergence
    - Message damping
    - Asynchronous message update schedules

Topics

- Probability and Statistics
- Probabilistic Graphical Models
- Message Passing Inference
- Parameter Learning

Maximum Likelihood Estimation

$$\theta^{\text{MLE}} = \arg\max_{\theta} p(\mathcal{Y} \mid \theta) = \arg\max_{\theta} \log p(\mathcal{Y} \mid \theta)$$

If concave then just solve for zero-gradient solution,

Log-Likelihood Function doesn't change argmax since log is monotonic

$$\mathcal{L}(\theta) \equiv \log p(\mathcal{Y} \mid \theta) \qquad \nabla_{\theta} \mathcal{L}(\theta^{\text{MLE}}) = 0$$

Logarithm serves a couple of practical purposes:

1) Simplifies derivatives for conditionally independent data

$$\nabla_{\theta} \mathcal{L}(\theta) = \sum_{i=1}^{N} \nabla_{\theta} \log p(y_i \mid \theta)$$

2) Avoids numerical under/overflow

MLE of Gaussian Mean

Assume data are i.i.d. univariate Gaussian,

Variance is known

$$p(\mathcal{Y} \mid \theta) = \prod_{i=1}^{N} \mathcal{N}(y_i \mid \theta, \sigma^2)$$

Log-likelihood function:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}(y_i - \theta)^2 \sigma^{-2}\right)\right)$$

Constant doesn't depend on mean

$$= \text{const.} - \frac{1}{2}\sum_{i=1}^{N}\left((y_i - \theta)^2 \sigma^{-2}\right)$$

MLE doesn't change when we:
1) Drop constant terms (in θ)
2) Minimize negative log-likelihood

MLE estimate is *least squares estimator*:

$$\theta^{\text{MLE}} = -\frac{1}{2\sigma^2} \arg\max_{\theta} \sum_{i=1}^{N} (y_i - \theta)^2 = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - \theta)^2$$

Regularized Maximum Likelihood

Penalty term R minimizes effect of outliers on estimator,

$$\theta^{\text{MLE}} = \arg\max_{\theta} \mathcal{L}(\theta) - \underbrace{\lambda}_{\text{Regularization weight}} \underbrace{R(\theta)}_{\text{Regularizer}}$$

Example L2-regularized Least-Squares,

$$\theta^{\text{MLE}} = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - \theta)^2 + \frac{\lambda}{2}\theta^2$$

In regression setting these have various names: ridge regression, LASSO

Example L1-regularized Least-Squares,

L1 is not differentiable, and so care must be taken in optimizer

$$\theta^{\text{MLE}} = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - \theta)^2 + \lambda|\theta|$$

Regularized Maximum Likelihood

Penalty term R minimizes effect of outliers on estimator,

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta) - \underbrace{\lambda}_{\text{Regularization weight}} \underbrace{R(\theta)}_{\text{Regularizer}}$$

Example L2-regularized Least-Squares,

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2} \sum_{i=1}^{N} (y_i - \theta)^2 + \frac{\lambda}{2} \theta^2$$

In regression setting known as ridge regression

$$\frac{1}{2} \sum_{i=1}^{N} \frac{d}{d\theta} (y_i - \theta)^2 + \frac{d}{d\theta} \frac{\lambda}{2} \theta^2 \quad = - \left(\sum_{i=1}^{N} y_i \right) + N\theta + \lambda\theta = 0$$

$$\hat{\theta} = \frac{1}{N + \lambda} \sum_{i} y_i$$

λ acts as *pseudocount*

MLE More Generally

MLE has a closed-form in Gaussian models because they are convex:

$$\theta^{\mathrm{MLE}} = \arg\max_{\theta} \underbrace{\log p(\mathcal{Y} \mid \theta) \equiv \mathcal{L}(\theta)}_{\textbf{Quadratic in Gaussian MLE}}$$

Log-likelihood is typically non-convex, so we use numerical methods such as Gradient descent:

$$\theta^{k+1} = \theta^k + \beta \nabla_\theta \mathcal{L}(\theta^k)$$

In this setting we cannot generally guarantee optimal MLE estimators

Maximum A Posteriori (MAP) Estimation

Recall the MAP estimator maximizes posterior probability,

$$
\begin{aligned}
\theta^{\mathrm{MAP}} &= \arg\max_{\theta} p(\theta \mid \mathcal{Y}) \\
&= \arg\max_{\theta} p(\theta, \mathcal{Y}) && \textbf{(Bayes' rule)} \\
&= \arg\max_{\theta} p(\mathcal{Y} \mid \theta) p(\theta) && \textbf{(Probability Chain Rule)} \\
&= \arg\max_{\theta} \log p(\mathcal{Y} \mid \theta) + \log p(\theta) && \textbf{(Monotonicity of Logarithm)}
\end{aligned}
$$

Prior serves as regularizer in regularized MLE:

$$\theta^{\mathrm{MLE}} = \arg\max_{\theta} \mathcal{L}(\theta) - \lambda R(\theta)$$

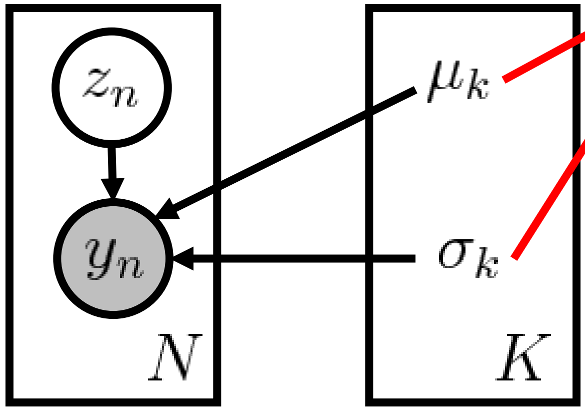
So conceptually, defining a regularizer in MLE imposes prior beliefs

MLE Summary

- Recall the trick of maximizing the p.d.f. by minimizing the negative log
- The Gaussian form for the likelihood led to a least-squares problem
- Least-squares solutions are tightly connected to assuming Gaussian distribution for the random effects (noise)
- If the random part is not Gaussian, then squared error may not make sense
- Squared error and Gaussian assumptions are mathematically very convenient but they are **very sensitive to outliers** (this motivates *robust estimators*)
- The least-squares solution leads to the average as being the “best” way to characterize a group of independent numbers, but there are other answers:
  - Minimum absolute value for error
  - Median
  - Minimum risk / maximal gain

Marginal Likelihood Calculation

Recall the Gaussian Mixture Model…

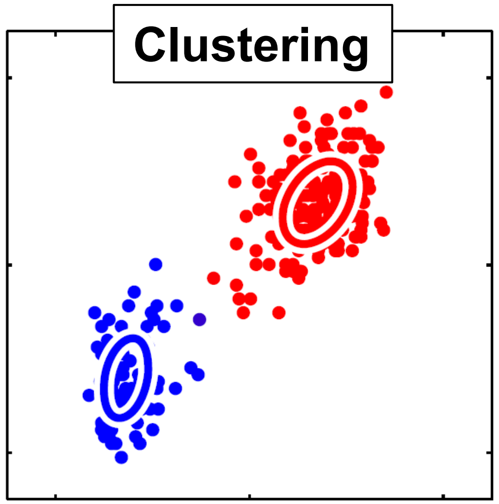


$$\theta = \{\mu_1, \sigma_1, \ldots, \mu_K, \sigma_K\}$$

Marginal Likelihood (likelihood function):

$$p(\mathcal{Y} \mid \theta) = \underbrace{\sum_{z_1} \cdots \sum_{z_N}} p(z_1, \ldots, z_N, \mathcal{Y} \mid \theta)$$

Sum over all possible K^N assignments, which we cannot compute



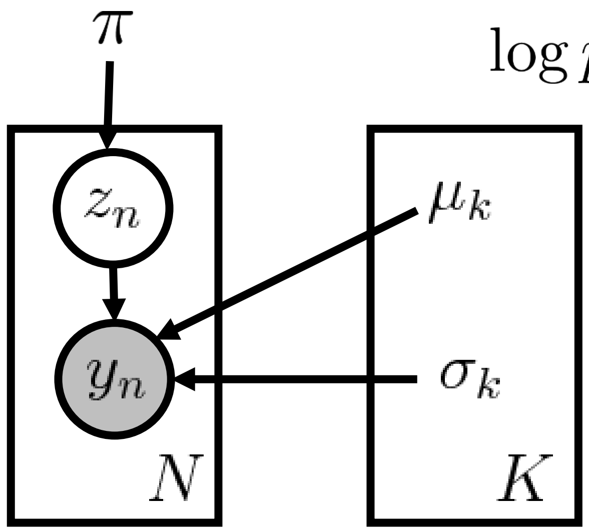
Motivation Approximate MLE / MAP when we cannot compute the marginal likelihood in closed-form

Expectation Maximization

$$\max_{\theta} \log p(\mathcal{Y} \mid \theta) \geq \max_{q,\theta} \mathbf{E}_q \left[\overbrace{\log \frac{p(z, \mathcal{Y} \mid \theta)}{q(z)}}^{\text{Complete Data Log-Likelihood}} \right] \equiv \mathcal{L}(q, \theta)$$

```
Initialize Parameters: $\theta^{(0)}$
At iteration t do:
    E-Step:    $q^{(t)}(z) = p(z \mid y, \theta^{(t-1)})$
    M-Step:    $\theta^{(t)} = \arg\max_\theta \mathcal{L}(q^{(t)}, \theta)$
Until convergence
```

Example: Gaussian Mixture Model



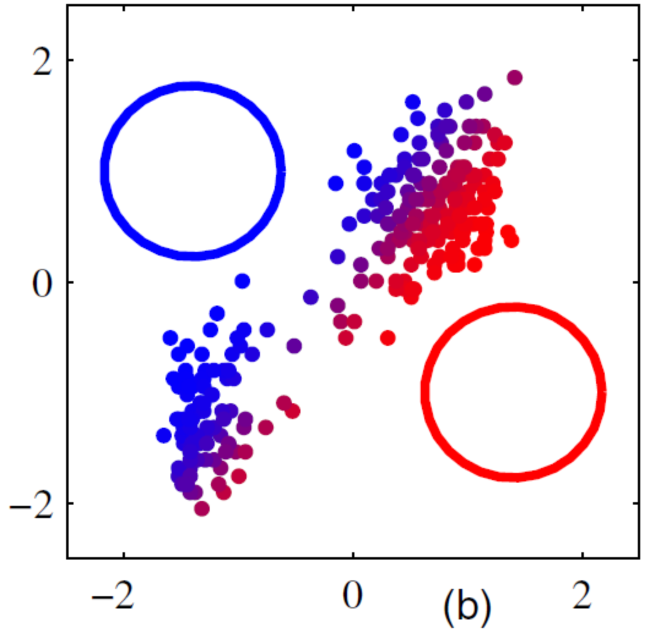
$$\log p(\mathcal{Y} \mid \pi, \mu, \Sigma) \geq \sum_{n=1}^{N} \sum_{k=1}^{K} q(z_n = k) \log \{\pi_k \mathcal{N}(y_n \mid \mu_k, \Sigma_k)\} = \mathcal{L}(q, \theta)$$

E-Step: $q^{\text{new}} = \arg\max_q \mathcal{L}(q, \theta^{\text{old}})$

$$q^{\text{new}}(z_n = k) = p(z_n = k \mid \mathcal{Y}, \mu^{\text{old}}, \Sigma^{\text{old}}, \pi^{\text{old}})$$

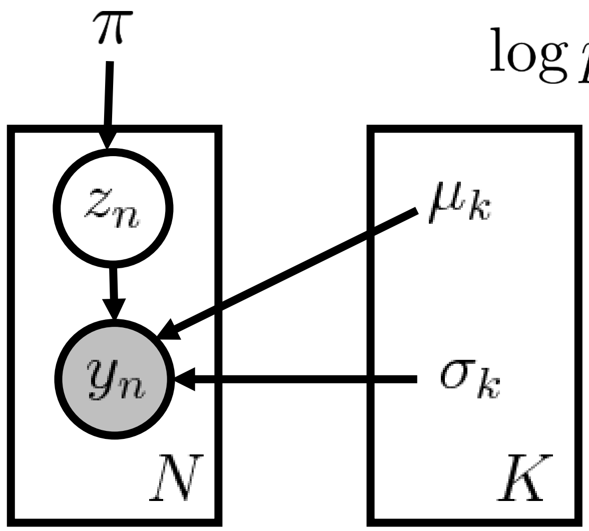
$$= \frac{p(z_n = k, \mathcal{Y} \mid \mu^{\text{old}}, \Sigma^{\text{old}}, \pi^{\text{old}})}{\sum_{j=1}^{K} p(z_n = j, \mathcal{Y} \mid \mu^{\text{old}}, \Sigma^{\text{old}}, \pi^{\text{old}})}$$

$$= \frac{\pi_k \mathcal{N}(y_n \mid \mu_k^{\text{old}}, \Sigma_k^{\text{old}})}{\sum_{j=1}^{K} \pi_j \mathcal{N}(y_n \mid \mu_j^{\text{old}}, \Sigma_j^{\text{old}})}$$



Commonly refer to $q(z_n)$ as *responsibility*

Example: Gaussian Mixture Model



$$\log p(\mathcal{Y} \mid \pi, \mu, \Sigma) \geq \sum_{n=1}^{N} \sum_{k=1}^{K} q(z_n = k) \log \{\pi_k \mathcal{N}(y_n \mid \mu_k, \Sigma_k)\} = \mathcal{L}(q, \theta)$$

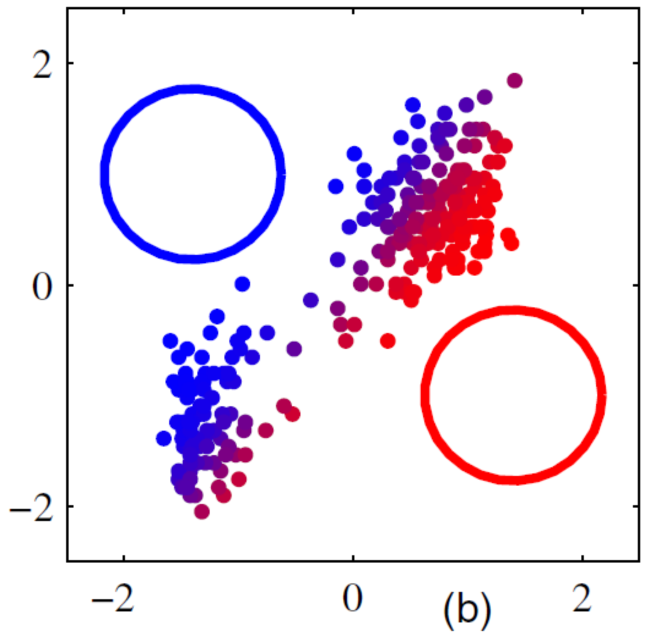
M-Step: $\theta^{\text{new}} = \arg\max_{\theta} \mathcal{L}(q^{\text{new}}, \theta)$

Start with mean parameter μ_k,

$$0 = \nabla_{\mu_k} \mathcal{L}(q^{\text{new}}, \theta)$$

$$0 = \sum_{n=1}^{N} \nabla_{\mu_k} \mathbf{E}_{z_n \sim q^{\text{new}}} \left[\log \mathcal{N}(y_n \mid \mu_{z_n}, \Sigma_{z_n})\right]$$

$$0 = -\sum_{n=1}^{N} q^{\text{new}}(z_n = k) \Sigma_k (y_n - \mu_k)$$



(b)

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} q^{\text{new}}(z_n = k) y_n \quad \text{where} \quad N_k = \sum_{n=1}^{N} q(z_n = k)$$

EM Lower Bound

$$\mathbf{E}_q\left[\log \frac{p(z,y\mid\theta)}{q(z)}\right] = \mathbf{E}_q\left[\log \frac{p(z,y\mid\theta)}{q(z)}\frac{p(y\mid\theta)}{p(y\mid\theta)}\right]$$

(Multiply by 1)

$$= \log p(y\mid\theta) - \mathrm{KL}(q(z)\|p(z\mid y,\theta))$$

(Definition of KL)

Bound gap is the Kullback-Leibler divergence KL(q||p),

$$\mathrm{KL}(q(z)\|p(z\mid y,\theta)) = \sum_z q(z)\log\frac{q(z)}{p(z\mid y,\theta)}$$

- Similar to a “distance” between q and p

$$\mathrm{KL}(q\,||\,p) \geq 0 \text{ and } \mathrm{KL}(q\,||\,p) = 0 \text{ if and only if } q = p$$

- This is why solution to E-step is $q(z) = p(z\mid y,\theta)$

Properties of Expectation Maximization Algorithm

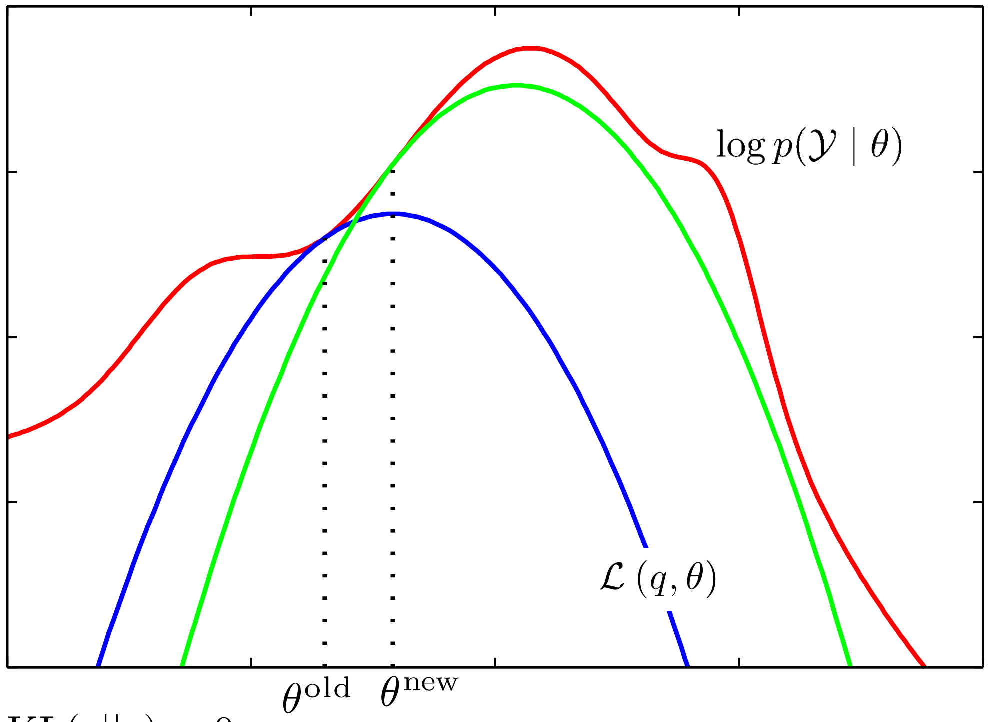
Sequence of bounds is monotonic,

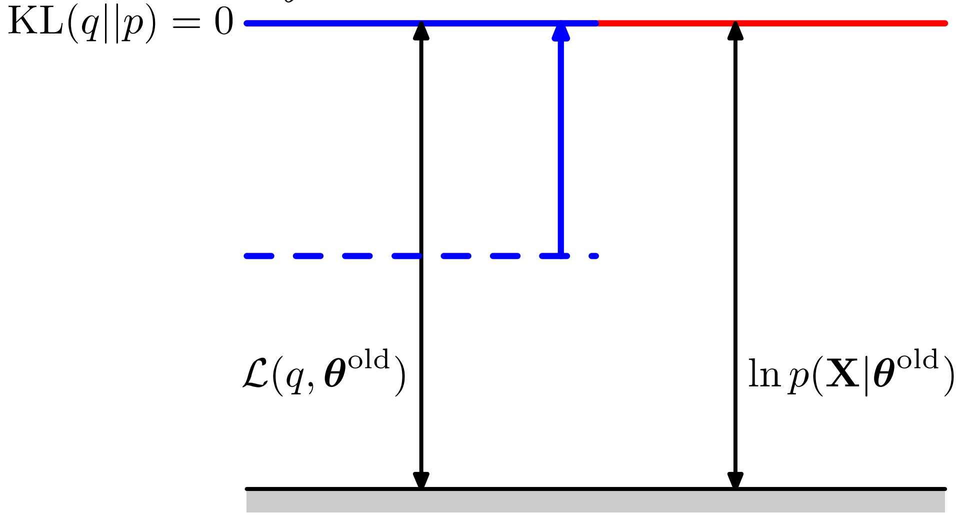
$$\mathcal{L}(q^{(1)}, \theta^{(1)}) \leq \mathcal{L}(q^{(2)}, \theta^{(2)}) \leq \ldots \leq \mathcal{L}(q^{(T)}, \theta^{(T)})$$

Guaranteed to converge
(**Pf.** Monotonic sequence bounded above.)

Converges to a local maximum of the marginal likelihood

After each E-step bound is tight at θ^{old} so likelihood calculation is exact (for those parameters)





$\mathrm{KL}(q||p) = 0$
$\mathcal{L}(q, \boldsymbol{\theta}^{\text{old}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\text{old}})$

MAP EM

Easily extends to (approximate) MAP estimation,

$$\max_{\theta} \log p(\theta \mid \mathcal{Y}) \geq \max_{q,\theta} \mathbf{E}_q \left[\log \frac{p(z, \mathcal{Y} \mid \theta)}{q(z)} \right] + \log p(\theta) + \text{const.}$$

E-step unchanged / Slightly modifies M-step,

E-Step

$$\begin{aligned} q^{\text{new}} &= \arg\max_q \mathcal{L}(q, \theta^{\text{old}}) \\ &= p(z \mid \mathcal{Y}, \theta^{\text{old}}) \end{aligned}$$

M-Step

$$\theta^{\text{new}} = \arg\max_{\theta} \mathcal{L}(q^{\text{new}}, \theta) + \log p(\theta)$$

Properties of both MLE / MAP EM

- Monotonic in $\mathcal{L}(q, \theta)$ or $\mathcal{L}(q, \theta) + \log p(\theta)$ (for MAP)
- Provably converge to local optima (hence approximate estimation)

Learning Summary

Maximum a posteriori (MAP) maximizes posterior probability,

$$\theta^{\text{MAP}} = \arg\max_{\theta} \log p(\theta \mid \mathcal{Y}) = \arg\max_{\theta} \mathcal{L}(\theta) + \log p(\theta)$$

Parameters are *random* quantities with prior $p(\theta)$.

Corresponds to regularized MLE for specific prior/regularizer pair,

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta) - \lambda \mathcal{R}(\theta)$$

Gaussian prior=L2, Laplacian prior=L1

Straightforward sequential updating, e.g. Bayesian linear regression

Learning Summary

- Most models will not yield closed-form MLE/MAP estimates
- Gradient-based methods optimize log-likelihood function

$$\theta^{k+1} = \theta^k + \beta \nabla_\theta \mathcal{L}(\theta^k)$$

- Expectation Maximization (EM) alternative to gradient methods
- Both approaches approximate for non-convex models