ECE 6504: Advanced Topics in Machine Learning

Probabilistic Graphical Models and Large-Scale Learning

Topics
- Markov Random Fields: Inference
  - Exact: VE
  - Exact+Approximate: BP

Readings: Barber 5

Dhruv Batra
Virginia Tech

Administrativia

- HW3
  - Out 2 days ago
  - Due: Apr 4, 11:55pm
  - Implementation: Loopy Belief Propagation in MRFs

Recap of Last Time

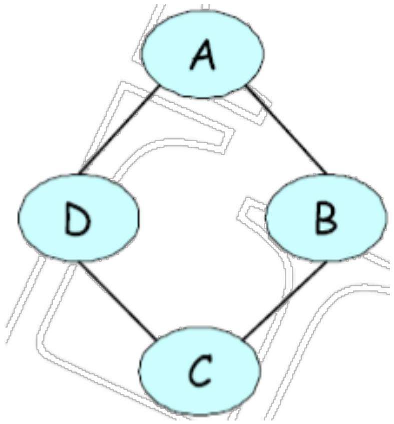
Markov Nets

- Set of random variables
- Undirected graph
  - Encodes independence assumptions
- Unnormalized Factor Tables
- Joint distribution:
  - Product of Factors

Pairwise MRFs

- Pairwise Factors
  - A function of 2 variables
    - Often unary terms are also allowed (although strictly speaking unnecessary)
  - On board

Pairwise MRF: Example



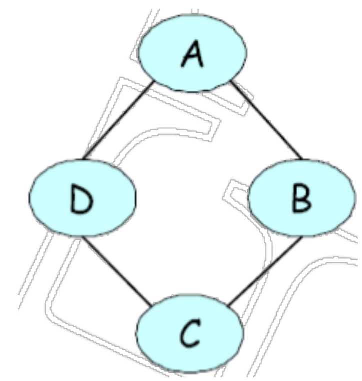
$\phi_1[A,B]$			$\phi_2[B,C]$			$\phi_3[C,D]$			$\phi_4[D,A]$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100

Normalization for computing probabilities

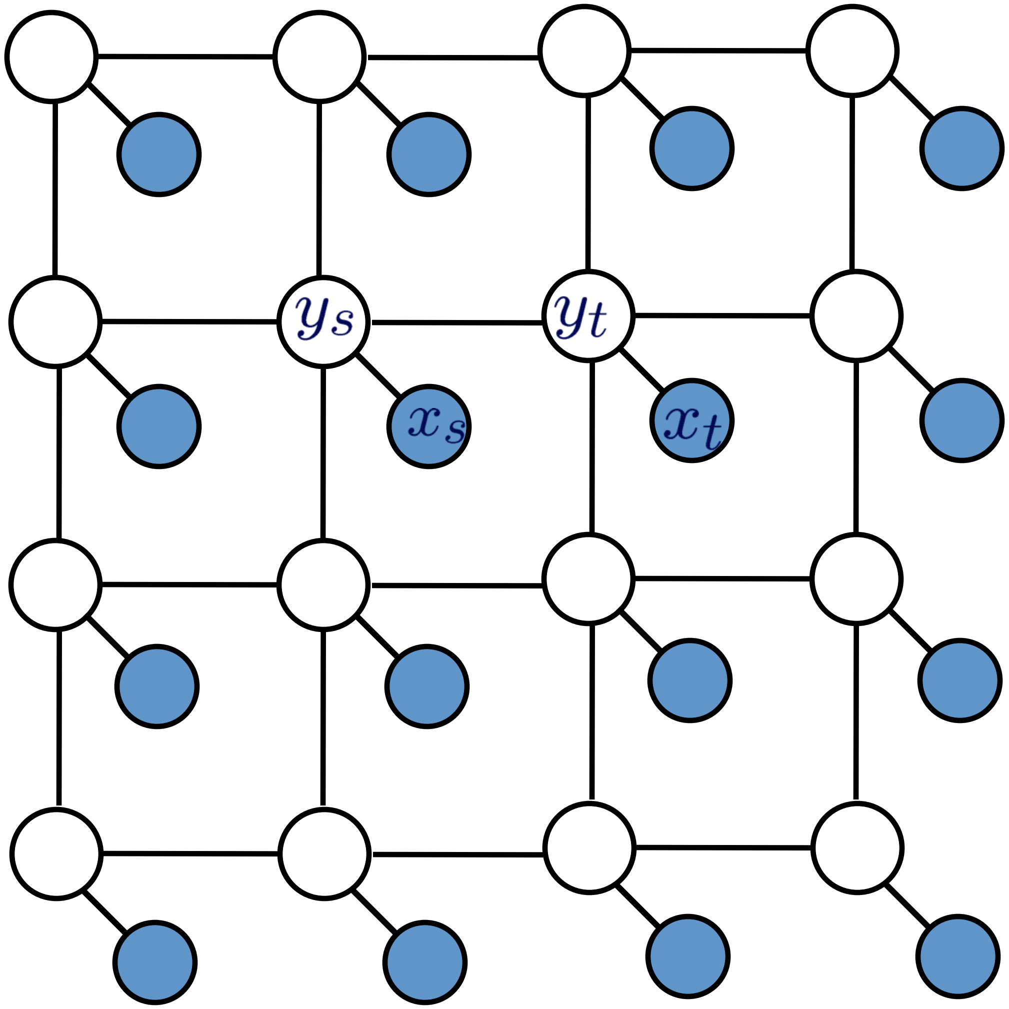
- To compute actual probabilities, must compute normalization constant (also called partition function)

Assignment				Unnormalized	Normalized
a^0	b^0	c^0	d^0	300000	0.04
a^0	b^0	c^0	d^1	300000	0.04
a^0	b^0	c^1	d^0	300000	0.04
a^0	b^0	c^1	d^1	30	$4.1 \cdot 10^{-6}$
a^0	b^1	c^0	d^0	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^0	d^1	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^1	d^0	5000000	0.69
a^0	b^1	c^1	d^1	500	$6.9 \cdot 10^{-5}$
a^1	b^0	c^0	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^0	d^1	1000000	0.14
a^1	b^0	c^1	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^1	d^1	100	$1.4 \cdot 10^{-5}$
a^1	b^1	c^0	d^0	10	$1.4 \cdot 10^{-6}$
a^1	b^1	c^0	d^1	100000	0.014
a^1	b^1	c^1	d^0	100000	0.014
a^1	b^1	c^1	d^1	100000	0.014

- Computing partition function is hard! Must sum over all possible assignments



Nearest-Neighbor Grids



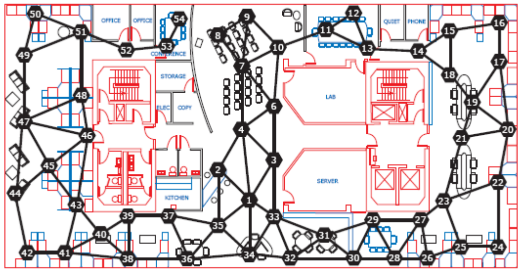
Low Level Vision

- Image denoising
- Stereo
- Optical flow
- Shape from shading
- Superresolution
- Segmentation

$y_s \longrightarrow$ unobserved or hidden variable

$x_s \longrightarrow$ local observation

Factorization in Markov networks



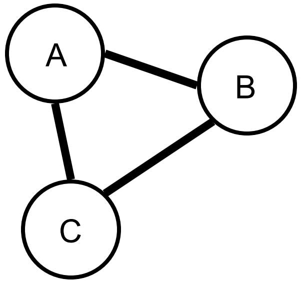
- Given an undirected graph *H* over variables $\mathbf{X}=\{X_1,\ldots,X_n\}$
- A distribution *P* **factorizes** over *H* if there exist
  - subsets of variables $\mathbf{D}_1 \subseteq \mathbf{X}, \ldots, \mathbf{D}_m \subseteq \mathbf{X}$, such that $\mathbf{D}_i$ are *fully connected* in *H*
  - *non-negative potentials* (or factors) $\phi_1(\mathbf{D}_1), \ldots, \phi_m(\mathbf{D}_m)$
    - also known as clique potentials
  - such that

$$P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\phi_i(\mathbf{D}_i)$$

- Also called Markov random field *H,* or Gibbs distribution over *H*

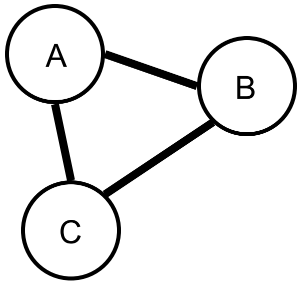
Structure in cliques

- Possible potentials for this graph:



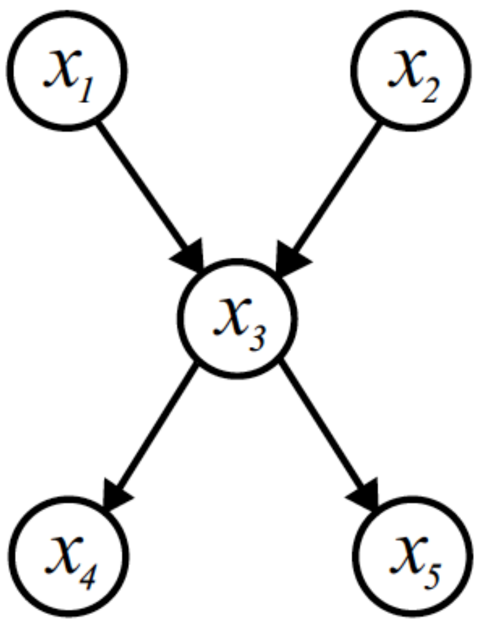
Factor graphs

- Bipartite graph:
  - variable nodes (ovals) for $X_1,\ldots,X_n$
  - factor nodes (squares) for $\phi_1,\ldots,\phi_m$
  - edge $X_i - \phi_j$ if $X_i \in$ Scope[ϕ_j]

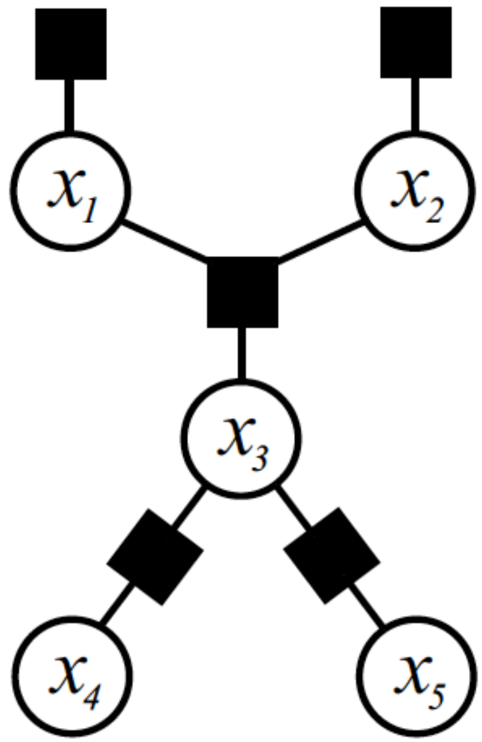


- Very useful for approximate inference
  - Make factor dependency explicit

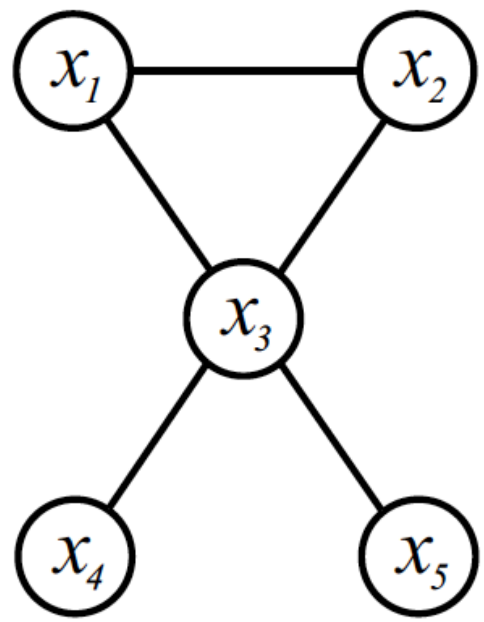
Types of Graphical Models



Directed



Factor



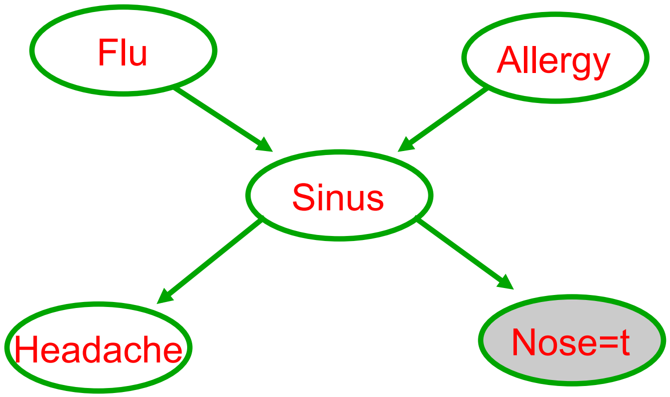
Undirected

Plan for today

- MRF Inference
  - Exact Inference
    - Variable Elimination
  - Exact+Approximate Inference
    - (General) Belief Propagation
    - Cluster Graphs
      - Family Preserving Property
      - Running Intersection Property
    - Message-Passing
  - Approximate Inference
    - Bethe Cluster Graph
    - Loopy BP
  - Exact Inference
    - Junction Tree
    - BP on Junction Trees

Marginal Inference Example

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



- Conditional Probability: P(**Y** | **E**=**e**)
  – P(F | N=t)

Variable Elimination algorithm

- Given a BN and a query P(**Y**|**e**) ≈ P(**Y**,**e**)
  - "Instantiate Evidence"

IMPORTANT!!!

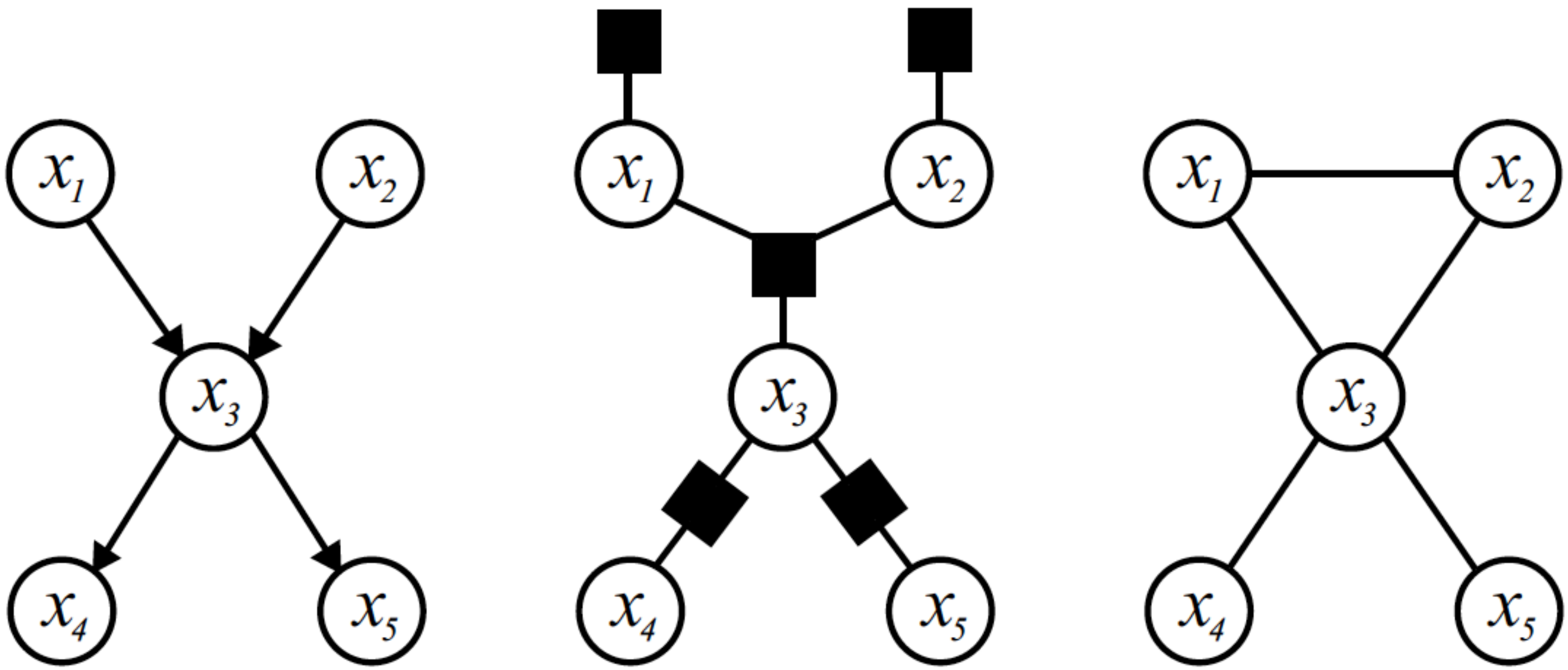
- Choose an ordering on variables, e.g., X_1, …, X_n
- For i = 1 to n, If $X_i \notin \{\mathbf{Y},\mathbf{E}\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

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

  - Variable X_i has been eliminated!
- Normalize P(Y,e) to obtain P(Y|e)

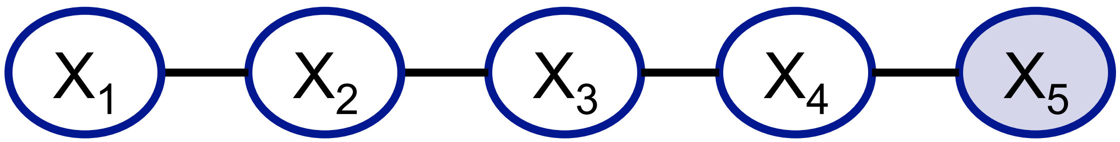
VE for MRF

- Exactly the same algorithm works!
  - Factors are no longer CPTs
  - But VE doesn't care



Example

- Chain MRF



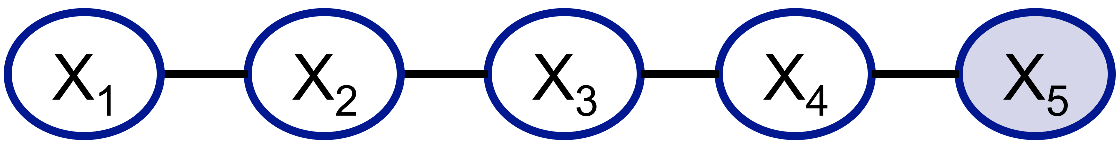
Compute:

$$P(X_1 \mid X_5 = x_5)$$

- VE steps on board

Example

- Chain MRF



Compute:

$$P(X_i \mid X_5 = x_5)$$
$$\forall i \in \{1, 2, 3, 4\}$$

Variable elimination for every i, what's the complexity?

Can we do better by caching intermediate results?

Yes! via Junction-Trees
But let's look at BP first

New Topic: Belief Propagation

What is BP?

- Technique invented by Judea Pearl in 1982
  - Initially to compute marginals in BNs

- Later generalized
  - to MRFs, Factor Graphs
  - To MAP inference; to Marginal-MAP inference

- Lots of analysis
  - Under some cases EXACT
    - Tree graphs
      - In this setting, BP equivalent to VE on Junction-Trees
    - Submodular potentials
      - In this setting, BP equivalent to Graph-Cuts! [Tarlow et al. UAI11]
  - In general Approximate

Message Passing

- Variables/Factors "talk" to each other via messages:

"I (variable X_3) think that you (variable X_2):

belong to state 1 with confidence 0.4
belong to state 2 with confidence 10
belong to state 3 with confidence 1.5"

Image Credit: James Coughlan

Overview of BP

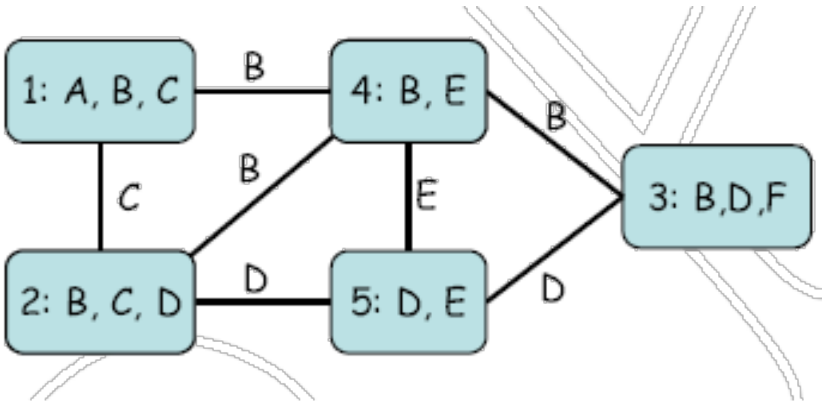
- Pick a graph to pass messages on
  - Cluster Graph

- Pick an ordering of edges
  - Round-robin
  - Leaves-Root-Leaves on a tree
  - Asynchonous

- Till convergence or exhaustion:
  - Pass messages on edges

- At vertices on graph compute *psuedo-marginals*

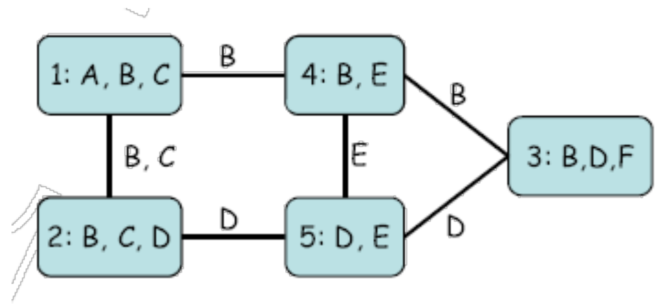
Cluster graph



- **Cluster Graph**:
  For set of factors *F*
  - Undirected graph
  - Each node i associated with a cluster $\mathbf{C}_i$
  - Each edge i – j is associated with a *separator set* of variables $\mathbf{S}_{ij} \subseteq \mathbf{C}_i \cap \mathbf{C}_j$

Generalized BP

- Initialization:
  - Assign each factor ϕ to a cluster $\alpha(\phi)$, Scope[ϕ]$\subseteq \mathbf{C}_{\alpha(\phi)}$
  - Initialize cluster: $\psi_i^0(\mathbf{C}_i) \propto \prod_{\phi:\alpha(\phi)=i} \phi$
  - Initialize messages: $\delta_{j\to i} = 1$

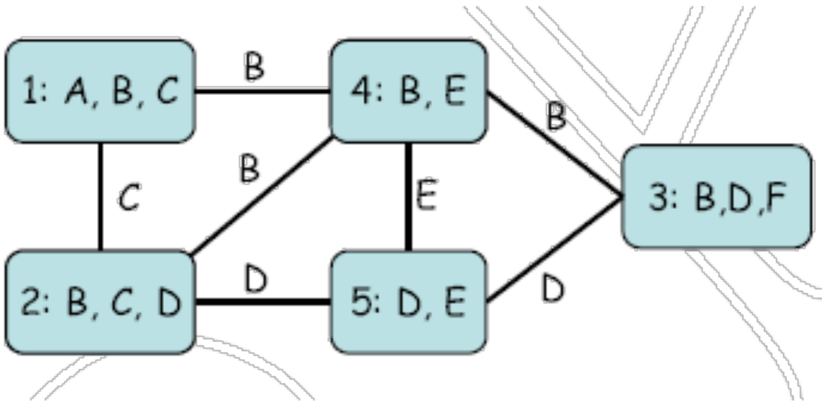


- While not converged, send messages:

$$\delta_{i\to j}(\mathbf{S}_{ij}) \propto \sum_{\mathbf{C}_i - \mathbf{S}_{ij}} \psi_i^0(\mathbf{C}_i) \prod_{k\in\mathcal{N}(i)-j} \delta_{k\to i}(\mathbf{S}_{ik})$$

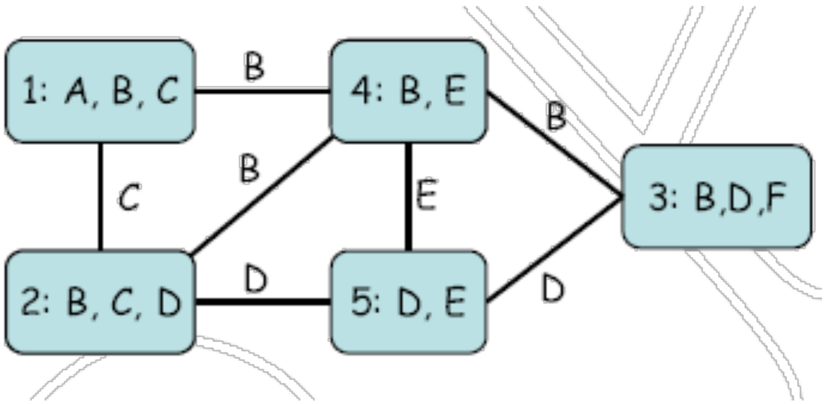
- Belief:
  - On Board

Properties of Cluster Graphs



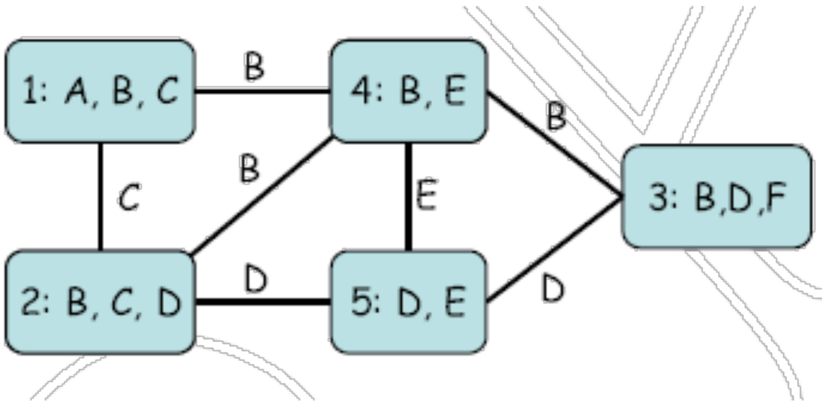
- **Family preserving**:
  For set of factors F
  - for each factor $f_j \in F$, $\exists$ node i such that scope$[f_i] \subseteq \mathbf{C}_i$

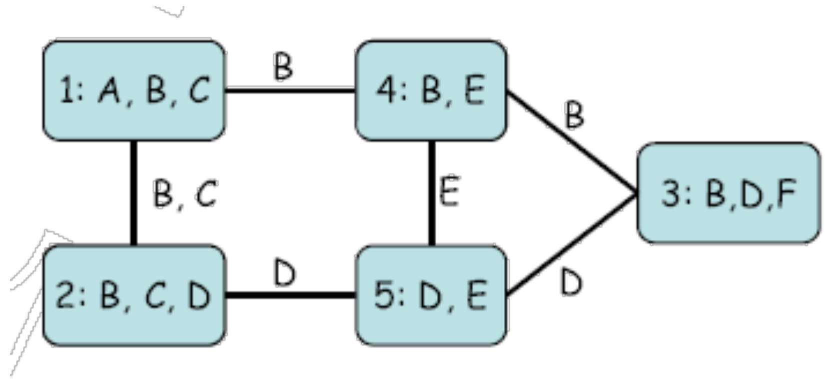
Properties of Cluster Graphs



- **Running intersection property (RIP)**
  - If $X \in \mathbf{C}_i$ and $X \in \mathbf{C}_j$ then $\exists$ *one and only one path* from $\mathbf{C}_i$ to $\mathbf{C}_j$ where $X \in \mathbf{S}_{uv}$ for every edge (u,v) in the path

Two cluster graph satisfying RIP with different edge sets





Overview of BP

- Pick a graph to pass messages on
  - Cluster Graph

- Pick an ordering of edges
  - Round-robin
  - Leaves-Root-Leaves on a tree
  - Asynchonous

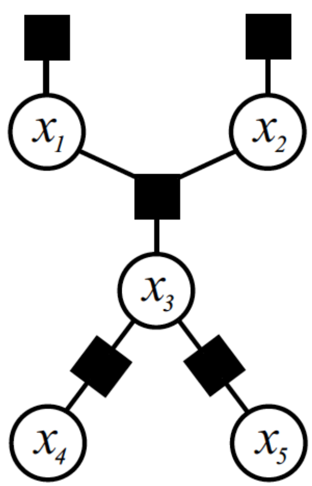
- Till convergence or exhaustion:
  - Pass messages on edges

- At vertices on graph compute *psuedo-marginals*

Cluster Graph for Loopy BP

- Bethe Cluster Graph
  - Set of Clusters = Factors F $\cup$ $\{X_i\}$
  - Sometimes also called "Running BP on Factor Graphs"
  - Example on board

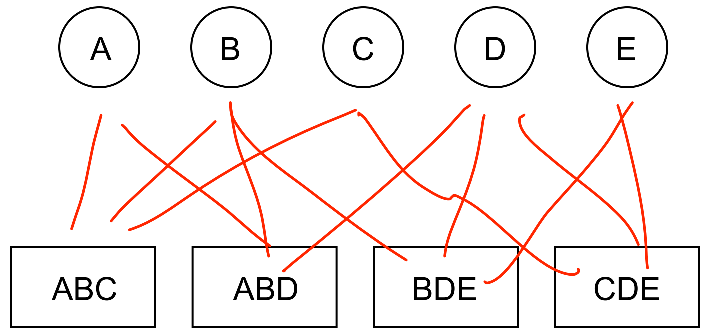
- Does the Bethe Cluster Graph satisfy properties?



Loopy BP in Factor graphs

- From node *i* to factor *j*:
  - *F*(i) factors whose scope includes X_i

$$\delta_{i\to j}(X_i) \propto \prod_{k\in\mathcal{F}(i)-j} \delta_{k\to i}(X_i)$$



- From factor *j* to node *i*:
  - Scope[ϕ_j] = **Y** $\cup$ {X_i}

$$\delta_{j\to i}(X_i) \propto \sum_{\mathbf{y}} \phi_j(X_i,\mathbf{y}) \prod_{X_k\in \text{Scope}[\phi_j]-X_i} \delta_{k\to j}(x_k)$$

- Belief:
  - Node:
  - Factor:

Loopy BP on Pairwise Markov Nets

$$\overrightarrow{\delta_{i\rightarrow j}}(y_j) = \sum_{y_i} \phi_i(y_i)\phi_{ij}(y_i, y_j) \prod_{k \in \mathcal{N}(i)-j} \overrightarrow{\delta_{k\rightarrow i}}(y_i)$$