Graphical Models

Lecture 14:

Message Passing in Loopy Graphs

Andrew McCallum
mccallum@cs.umass.edu

Thanks to Noah Smith and Carlos Guestrin for some slide materials.

Admin

- Repeat announcement: class mailing list 691gm-all@cs
  - Not yet subscribed? Send me request.
- HW#3 now due Friday April 1.

Sum-Product Message Passing

- Each clique tree vertex $\mathbf{C}_i$ passes messages to each of its neighbors once it's ready to do so.

$$\delta_{i \to j} = \sum_{\boldsymbol{C}_i \setminus \boldsymbol{S}_{i,j}} \nu_i \prod_{k \in \text{Neighbors}_i \setminus \{j\}} \delta_{k \to i}$$

  - This is asynchronous; might want to be careful about *scheduling*.
  - One option: two passes (upstream to some root, then downstream).
- At the end, for all $\mathbf{C}_i$:

$$\beta_i = \nu_i \prod_{k \in \text{Neighbors}_i} \delta_{k \to i}$$

  - This is the unnormalized marginal for $\mathbf{C}_i$.

Lecture 11

Calibrated Clique Tree as a Graphical Model

- Original (unnormalized) factor model and calibrated clique tree represent the same (unnormalized) measure:

$$\prod_{\phi \in \Phi} \phi = \frac{\prod_{C \in \text{Vertices}(\mathcal{T})} \beta_C}{\prod_{S \in \text{Edges}(\mathcal{T})} \mu_S}$$

clique **beliefs**

sepset **beliefs**

unnormalized Gibbs distribution from original factors **Φ**

Lecture 12

Belief Update Message Passing (Also Known as Sum-Product-Divide)

- Maintain beliefs at each vertex (β) and edge (μ).
- Initialize each β_i to ν_i.
- Initialize each $\mu_{i,j}$ to **1**.
- Pass belief update messages.

$$\sigma_{i \to j} \leftarrow \sum_{\boldsymbol{C}_i \setminus \boldsymbol{S}_{i,j}} \beta_i$$

$$\beta_j \leftarrow \beta_j \times \frac{\sigma_{i \to j}}{\mu_{i,j}}$$

$$\mu_{i,j} \leftarrow \sigma_{i \to j}$$

Lecture 12

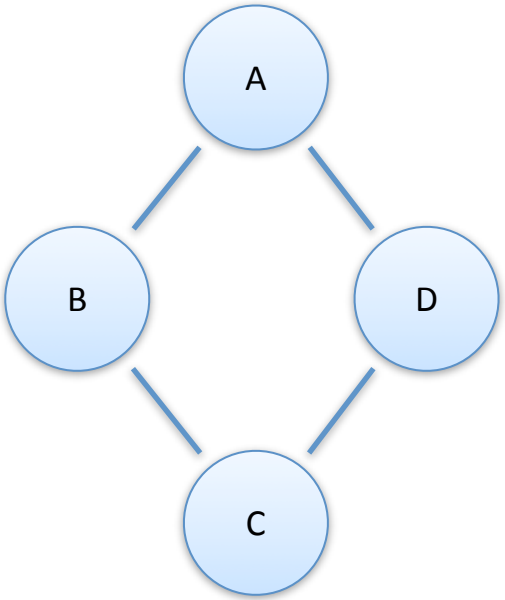
Message Passing

- Result is the same for both versions: calibrated clique tree.

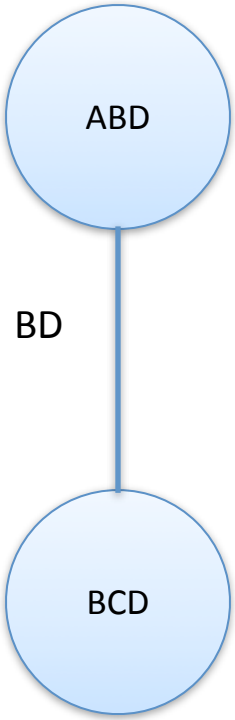
Clique Trees, Generalized

- Clique trees for exact inference:
  - groups of random variables on nodes
  - edges form a tree
  - running intersection property
    (implies sepsets are intersections, in trees)
- **Cluster graph**: generalization!
  - graph can have loops – not necessarily a tree
  - (but we will still want a variant of the
    running intersection property… coming soon)

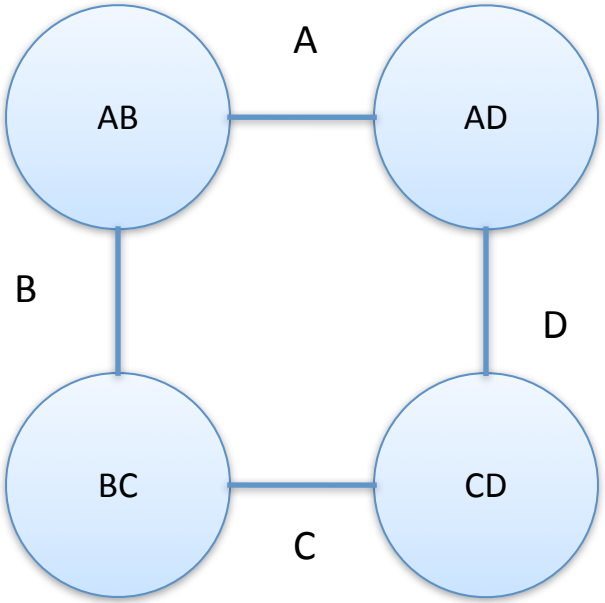
Example



Markov network



clique tree



(loopy) cluster graph

Effects

- Fewer random variables per node.
  - If we were to pass messages, they would be faster to compute.
- Sum-product and sum-product-divide did not hinge on having a tree.
  - We can still run these algorithms.
  - Two-pass convergence guarantee is gone.
  - Indeed, it is not clear that we have *any* convergence guarantee.
  - Node beliefs at the end may not equate to marginals.

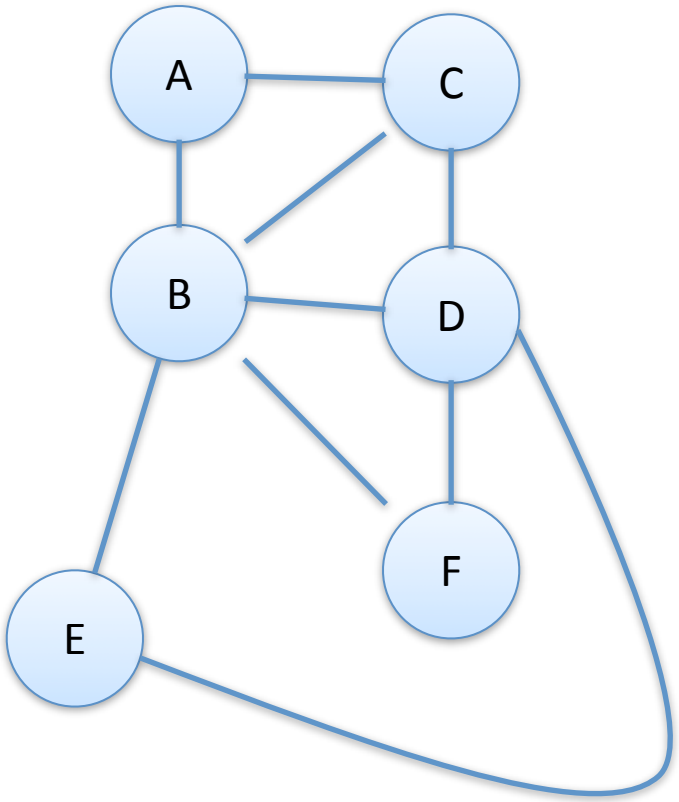
Loopy Graph Message Passing

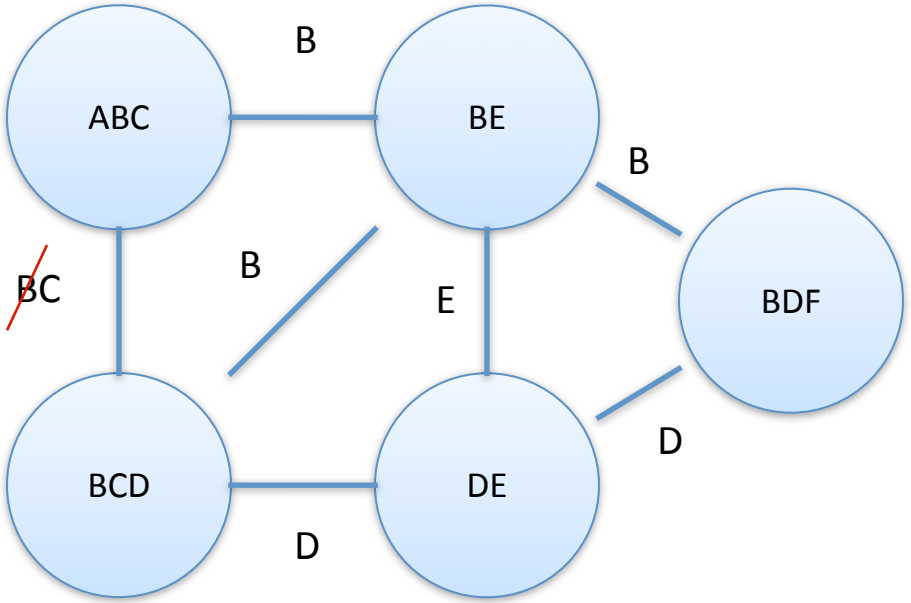
- Will it converge?
- If so, to what?

Running Intersection Property "variant" in Cluster Graphs

- Given any variable X and any two nodes it is a member of, $\mathbf{C}_i$ and $\mathbf{C}_j$, there is a single path between $\mathbf{C}_i$ and $\mathbf{C}_j$ such that X is on every edge.
  - There might be other paths that connect the nodes.
  - Unlike in clique trees, this does *not* imply that $\mathbf{S}_{i,j} = \mathbf{C}_i \cap \mathbf{C}_j$.
  - Instead, $\mathbf{S}_{i,j} \subseteq \mathbf{C}_i \cap \mathbf{C}_j$.

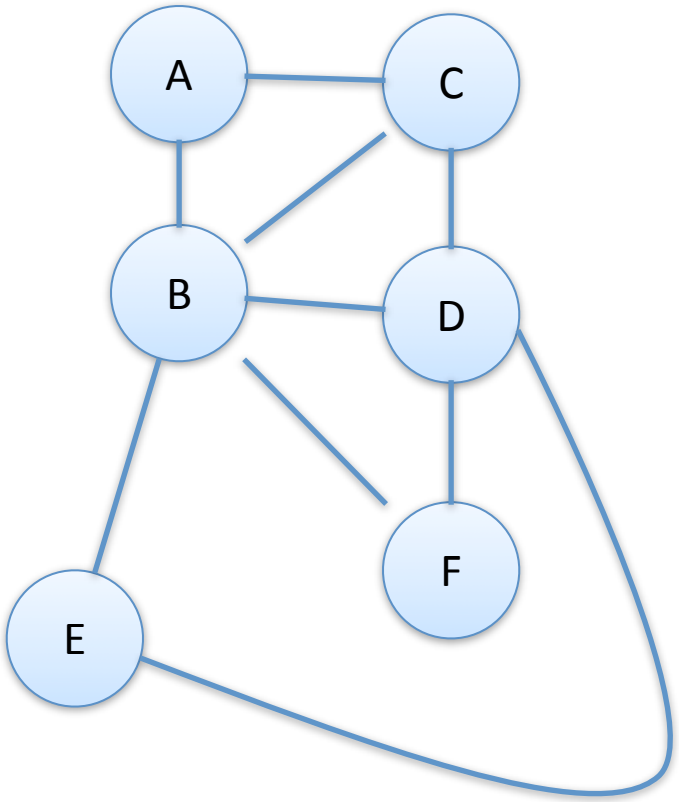
Example

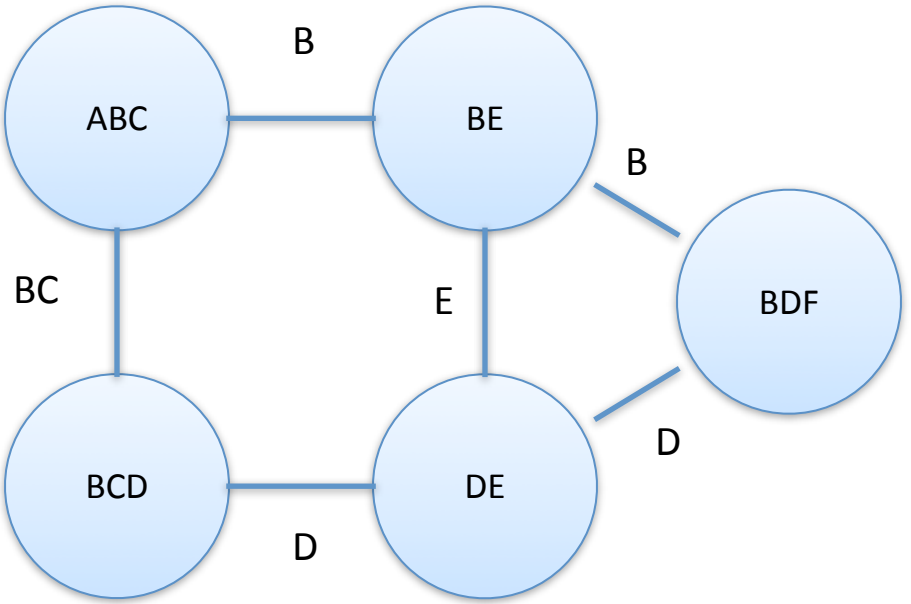




"B" removed to make "B-labeled" edges form a tree.

Example





To form a “clique tree” remove entire edges to make a tree.
To form a “cluster graph obeying running intersection variant” remove variables from messages on edges, such that there are no cycles in subgraphs containing only edges labeled with that variable.

Calibration in Cluster Graphs

- Adjacent nodes' beliefs show agreement on the sepset (not the full intersection).
- For graphs with the running intersection property, a variable X's marginal is identical in all nodes that contain X.

Cluster Graph Belief Propagation

- Both sum-product and sum-product-divide variants.
- Sum-product: how to start if no node has all incoming information yet?
  - Start with all messages = **1**.
- Keep sending messages until calibration.

Claims

- At convergence, we will have a calibrated cluster graph.

$$\begin{aligned} \sum_{\boldsymbol{C}_i \setminus \boldsymbol{S}_{i,j}} \beta_i &= \sum_{\boldsymbol{C}_j \setminus \boldsymbol{S}_{i,j}} \beta_j \\ &= \mu_{i,j}(\boldsymbol{S}_{i,j}) \end{aligned}$$

- Invariant: throughout the algorithm:

$$\prod_{\phi \in \Phi} \phi = \prod_{\boldsymbol{C} \in \text{Vertices}(\mathcal{T})} \nu_{\boldsymbol{C}} = \frac{\prod_{\boldsymbol{C} \in \text{Vertices}(\mathcal{T})} \beta_{\boldsymbol{C}}}{\prod_{\boldsymbol{S} \in \text{Edges}(\mathcal{T})} \mu_{\boldsymbol{S}}}$$

Cluster Graph Invariant

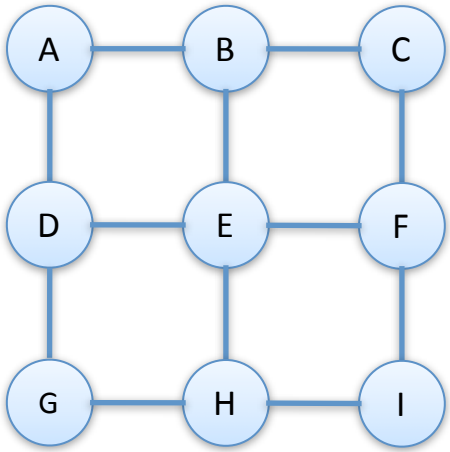
- Exactly like before.
- No information about the original distribution is lost.
- We are simply transforming the original factors into a “more useful” form.

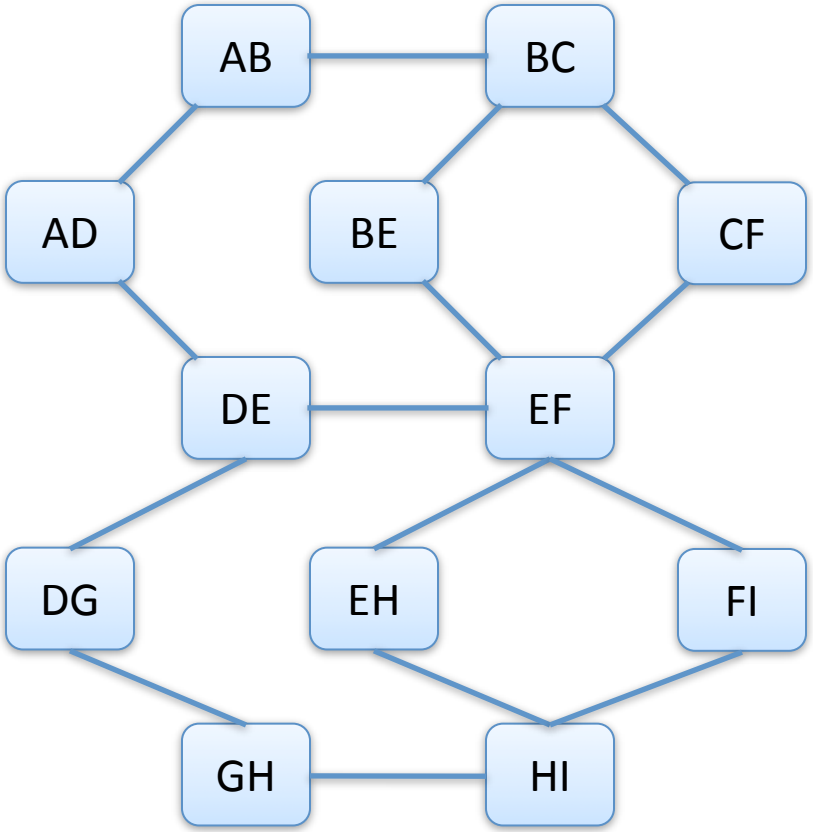
Cluster Graph Trade-Offs

- Intuitively, fewer clusters and bigger sepsets lead to better preservation of information.
- But breaking the graph into smaller parts leads to lower cost.

Bethe Cluster Graphs

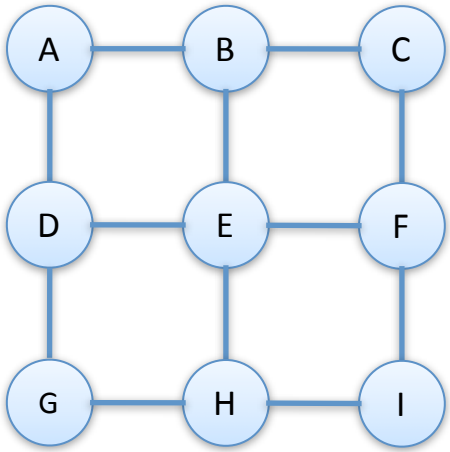
Example

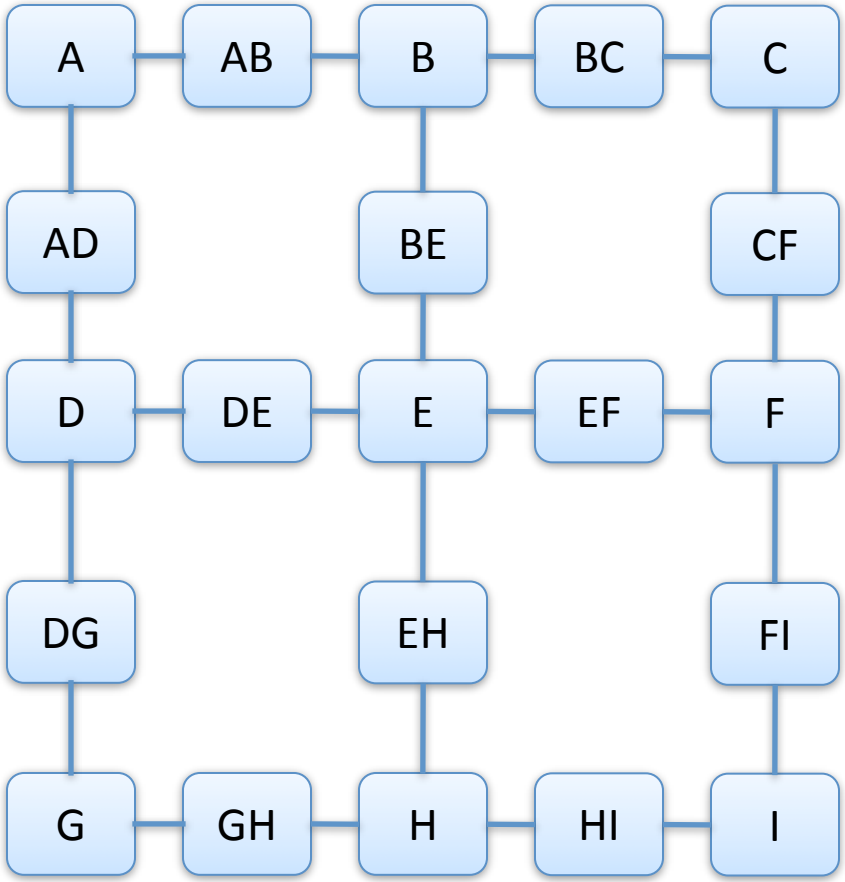




A cluster graph, but not Bethe

Example



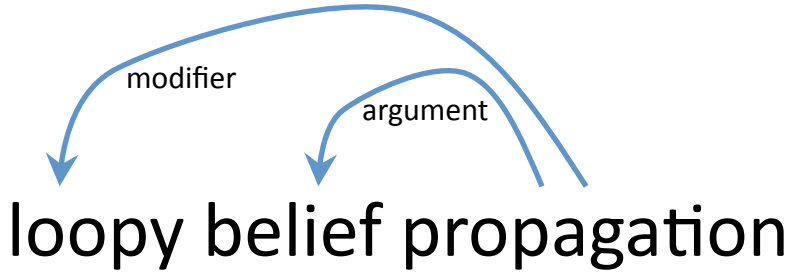


A Bethe cluster graph

Pairwise Markov Network Cluster Graphs

- Technically speaking, this is **loopy belief propagation**.
  - I've been using the term more broadly.

(Primer in NLP)

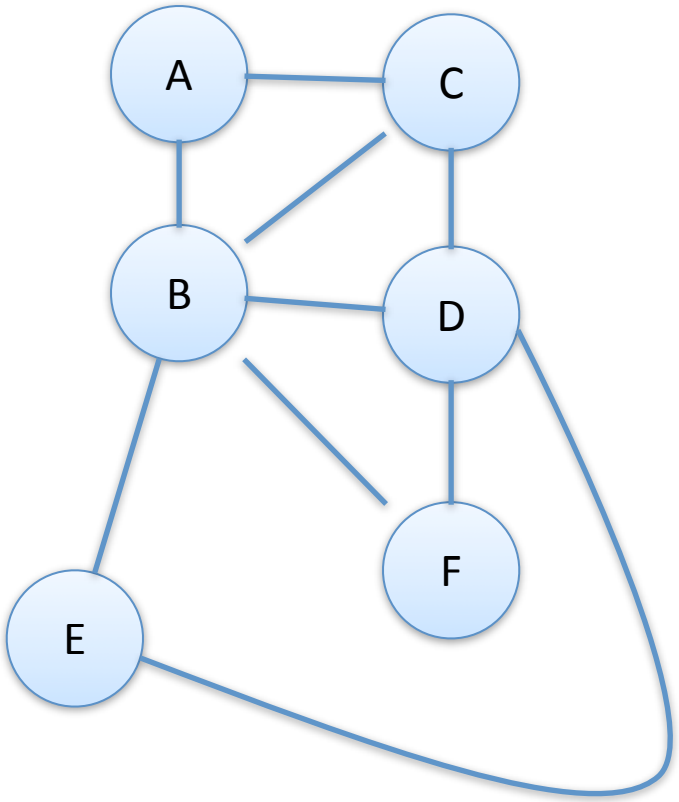


- The propagation is loopy, and it is beliefs that are propagated.
- The beliefs are not loopy!

Factor Graph Cluster Graphs

- Let each factor have a node, and each random variable have a node.
  - Called a **Bethe** cluster graph.

Example



{A, B, C} {B, C, D} {B, D, F} {B, E}, {D, E}

Factor Graph Cluster Graphs

- Let each factor have a node, and each random variable have a node.
  - Called a **Bethe** cluster graph.
- Information about variable interactions is lost during propagation.
  - Correct by merging some pairs?
  - May then have to adjust the sepsets to ensure the running intersection property…

Bad News

- Cluster graph belief propagation does not necessarily converge.
  - Oscillation!
  - Techniques like “dampening” the messages can help with convergence, maybe worse beliefs.
  - This problem tends to be worse for “peakier” or more deterministic models.
  - Lots of little loops are bad; a single loop is okay.
  - Many variations on the algorithm (see book).

Variational Analysis

- Recall the problem of maximizing the energy functional:

$$\max_{Q \in \mathcal{Q}} H_Q + \sum_{\phi \in \mathbf{\Phi}} \mathbb{E}_Q[\log \phi]$$

$$\text{such that} \quad \forall \boldsymbol{x}, \quad Q(\boldsymbol{x}) = \prod_i Q_i(x_i)$$

$$\forall i \quad \sum_{x_i} Q_i(x_i) = 1$$

- (Mean field: approximation by choosing an "easy" class $\mathcal{Q}$.)

Factored Energy Functional

$$F(Q) = H_Q + \sum_{\phi \in \mathbf{\Phi}} \mathbb{E}_Q[\log \phi]$$

$$\tilde{F}(Q) = \left(\sum_{i \in \mathcal{V}} H_{\beta_i} - \sum_{(i,j) \in \mathcal{E}} H_{\mu_{i,j}} \right) + \mathbb{E}_Q[\log \phi]$$

- For trees, they can be shown to be equivalent.
- Form of Q:

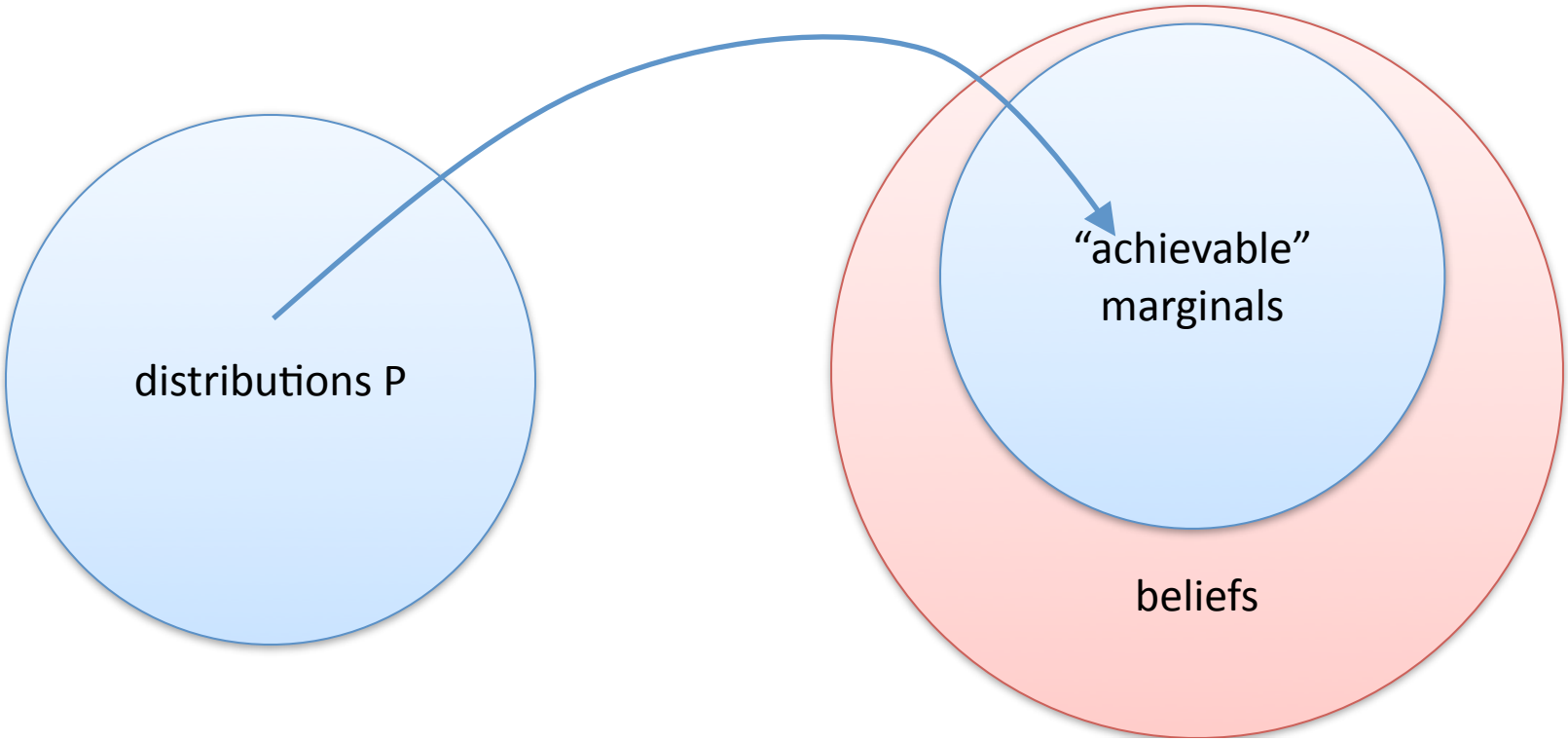
$$Q(\boldsymbol{X}) = \frac{\prod_{i \in \mathcal{V}} \beta_i}{\prod_{(i,j) \in \mathcal{E}} \mu_{i,j}}$$

Factored Energy Functional

- For cluster graphs, this is an approximation.
  - It is not a bound.
  - This is the first way that cluster graph belief propagation falls short.
- Second problem: constraints on the beliefs.
  - Not every setting of the beliefs corresponds to a coherent distribution over **X**.

Marginal Polytope

- It's possible to have a calibrated cluster graph whose beliefs are not *globally* consistent.



Marginal Polytope

- The set of achievable marginals actually forms a **polytope**, called the marginal polytope.
- Bad news:
  - The polytope doesn't generally have a compact representation.
  - It is NP hard in general to determine whether a set of beliefs is in that polytope.
  - *Optimizing* over the polytope is as hard as inference.

polygon = polytope in 2 dimensions

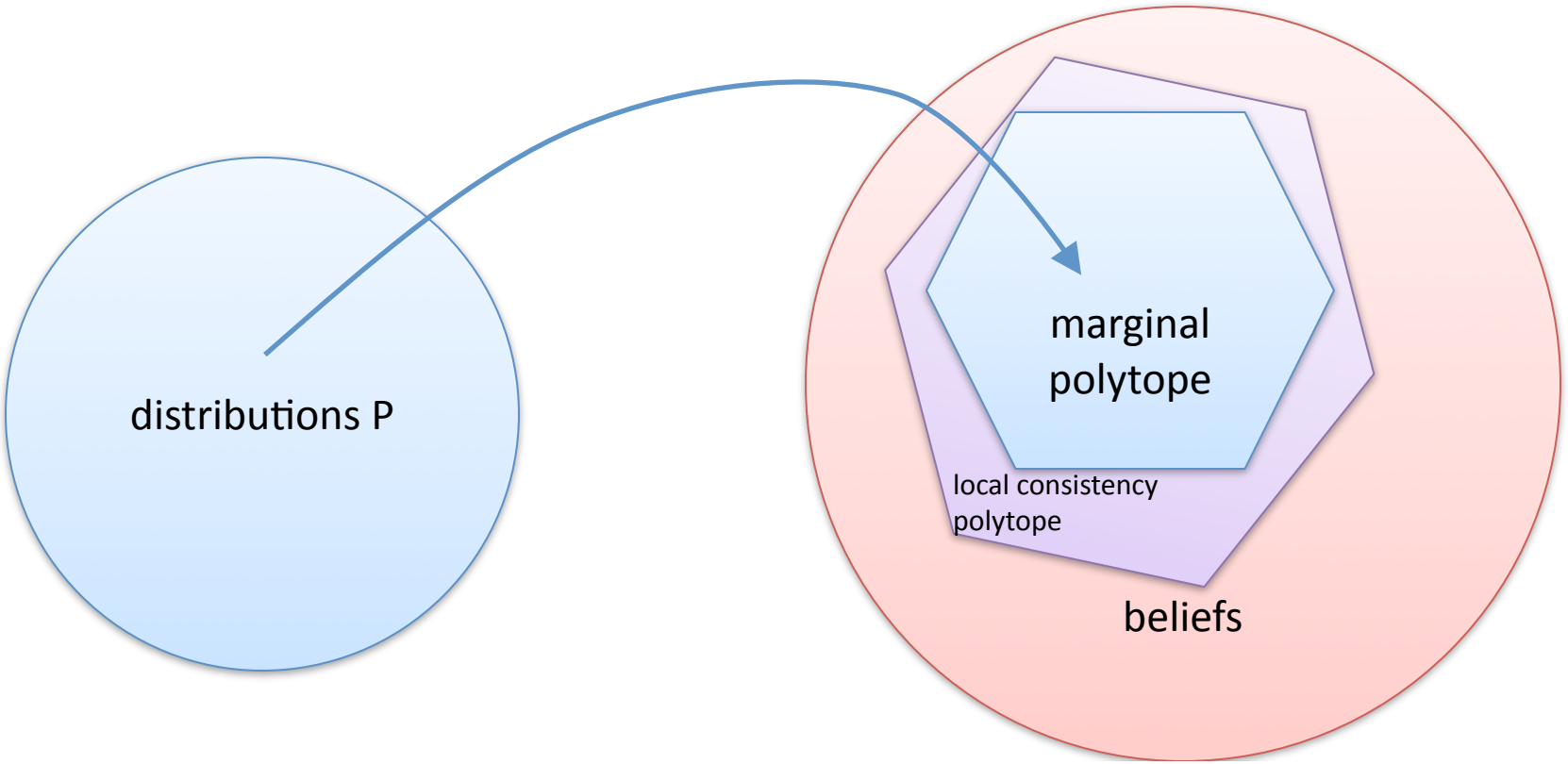
Approximating the Marginal Polytope

- Local consistency constraints:

$$\begin{aligned} \mu_{i,j} &= \sum_{\boldsymbol{C}_i \setminus \boldsymbol{S}_{i,j}} \beta_i \\ \sum_{\beta_i} &= 1 \\ \beta_i &\geq 0 \end{aligned}$$

- This can be understood as a *relaxation* of the marginal polytope.
- Points correspond to **pseudo marginals**.

Local Consistency Polytope



Equivalence

- A convergence point of cluster graph belief propagation equates to a stationary point of the factored energy functional over the local consistency polytope.
- Two approximations:
  - *factored* energy functional
  - *local consistency* polytope (not marginal polytope)
- Compare with mean field …

Caveats

- Not a bound on log Z.
- Might not be a local max:
  - boundary of the polytope
  - saddle point or local minimum
- Cluster graph belief propagation steps may not improve the objective.
  - Oscillation!
- The declarative view may be helpful for understanding better methods. See text.

Final Warnings

- Cluster graph belief propagation methods are a general purpose way to do inference in “hard” graphical models.
- May not converge.
- When it does converge, there may be different convergence points.