ECE 6504: Advanced Topics in Machine Learning

Probabilistic Graphical Models and Large-Scale Learning

Topics

- Markov Random Fields: Inference
  - Exact: Junction Trees
  - Approximate: Variational Inference

Readings: KF 10.1-10.4, Barber 5

Dhruv Batra

Virginia Tech

Recap of Last Time

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

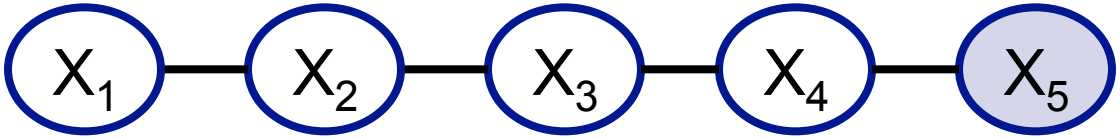
- For i = 1 to n, If $X_i \notin \{\mathbf{Y},\mathbf{E}\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

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

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

Example

- Chain MRF



Compute:

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

- VE steps on board

New Topic: Belief Propagation

Message Passing

- Variables/Factors “talk” to each other via messages:

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

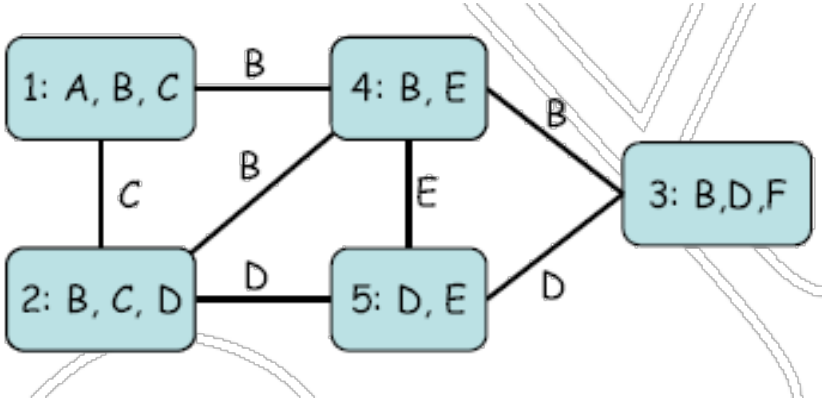
belong to state 1 with confidence 0.4
belong to state 2 with confidence 10
belong to state 3 with confidence 1.5”

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*

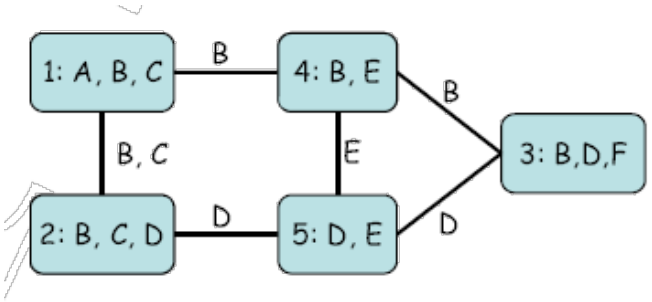
Cluster graph



- **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: $\delta_{j\to i} = 1$



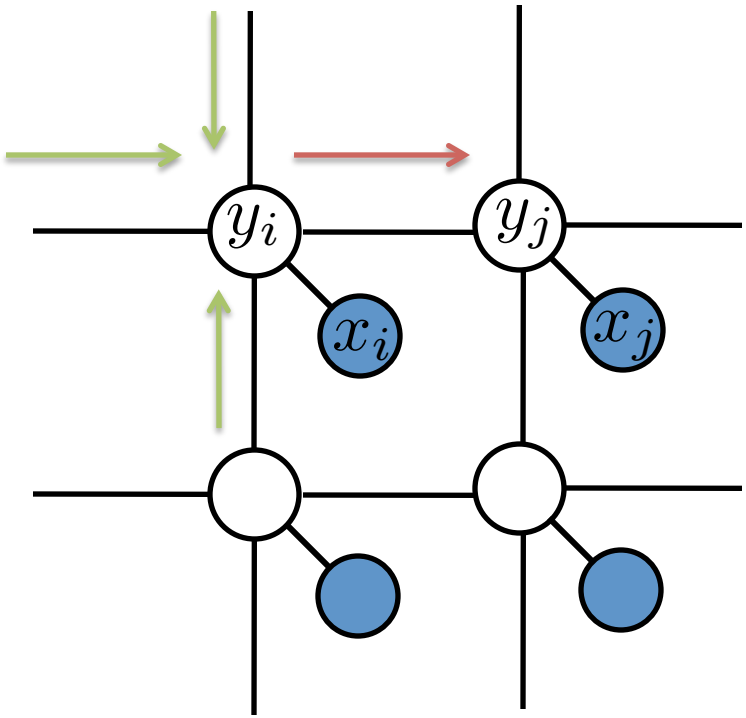
- While not converged, send messages:

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

- Belief:
  - On board

Loopy BP on Pairwise Markov Nets

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



Calibration

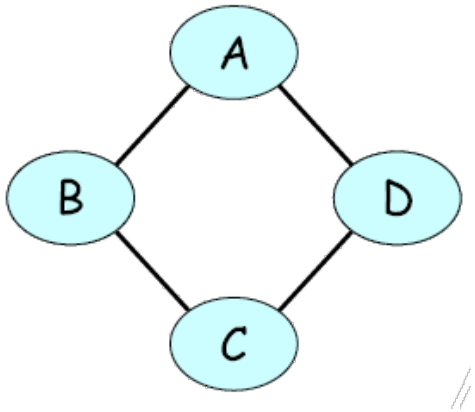
- Cluster Graphs are *calibrated*
  - when adjacent clusters agree in beliefs about sep-sets

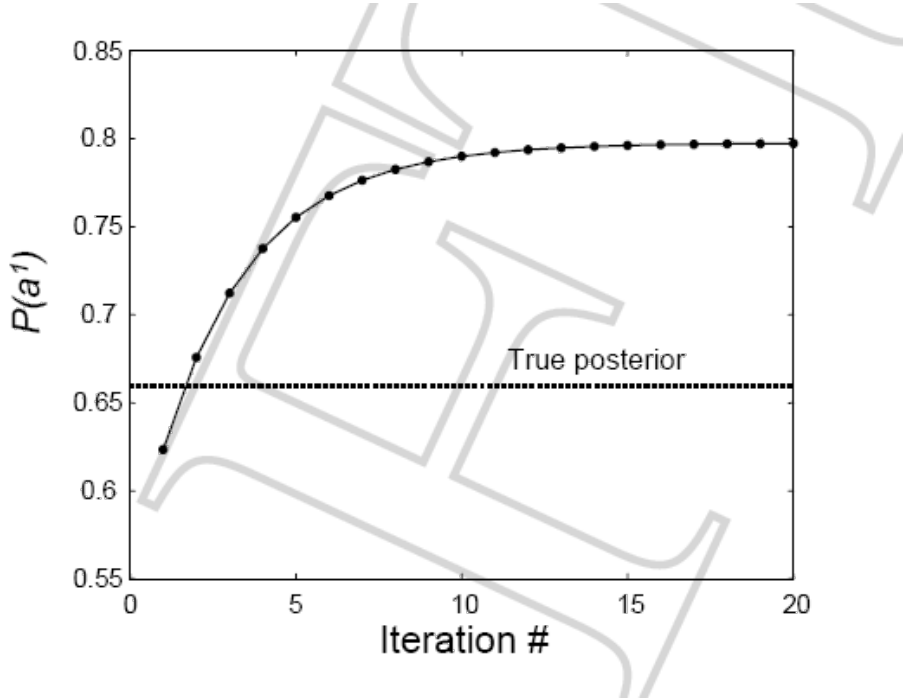
Convergence

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

- If you tried to send all messages, and messages haven't changed (in practice by much) → converged
- Convergence of BP => Calibration of Cluster Graph
- Note, this doesn't mean pseudo-marginals are correct!

An example of running loopy BP





Loopy BP

$$\delta_{i\to j}(X_j) = \sum_{x_i} \phi_i(x_i)\phi_{ij}(x_i, X_j) \prod_{k\in\mathcal{N}(i)-j} \delta_{k\to i}(x_i)$$

- What happened?
  - evidence goes around the loops multiple times
  - may not converge
  - if it converges, usually overconfident about probability values

- But often gives you reasonable, or at least useful answers
  - especially if you just care about the argmax rather than the actual probabilities

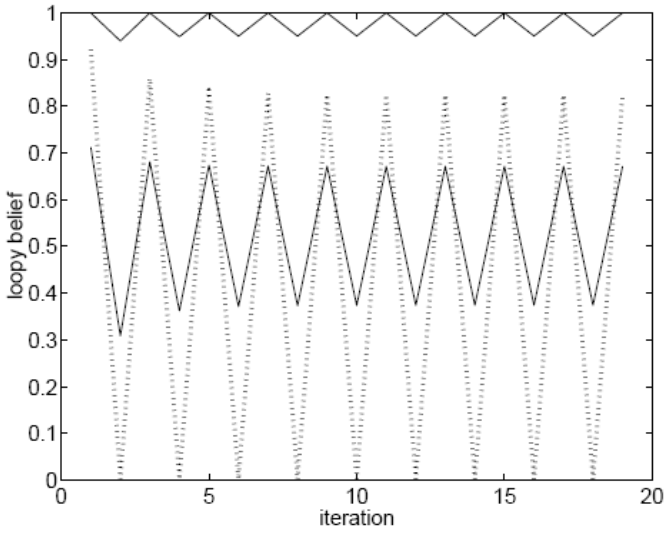
(Non-)Convergence of Loopy BP

- **Loopy BP can oscillate!!!**
  - oscillations can small
  - oscillations can be really bad!

- Typically,
  - if factors are closer to uniform, loopy does well (converges)
  - if factors are closer to deterministic, loopy doesn't behave well

- One approach to help: damping messages
  - new message is average of old message and new one:

  - often better convergence
    - but, when damping is required to get convergence, result often bad



graph from Murphy et al. ' 99

How to pass messages?

- Synchronous
  - All messages at once
  - Good for parallelization
  - Bad for convergence

- Asynchronous
  - Sequential according to some priority
  - Bad for parallelization
  - Good for convergence

Plan for today

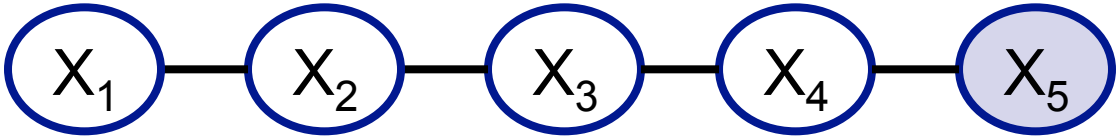
- MRF Inference
  - Exact Inference
    - Junction Tree
    - BP on Junction Trees
  - Message-Passing as Variational Inference
    - Mean Field
    - Structured Mean Field

New Topic

- Making BP Exact
  - Connecting BP to VE on Junction Trees

Example

- Chain MRF

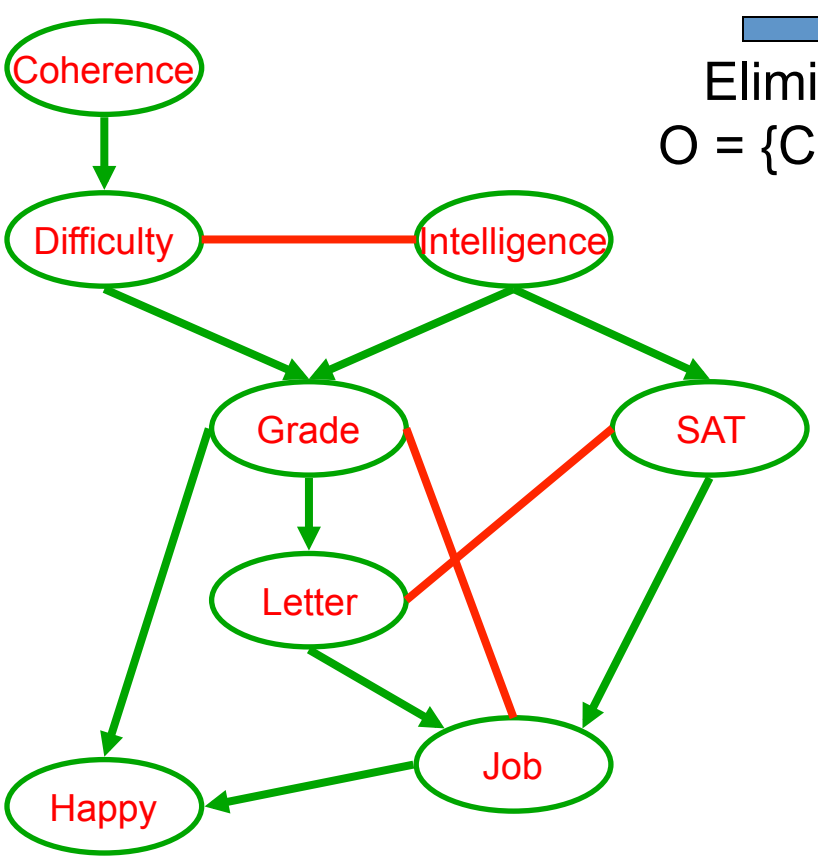


Compute:

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

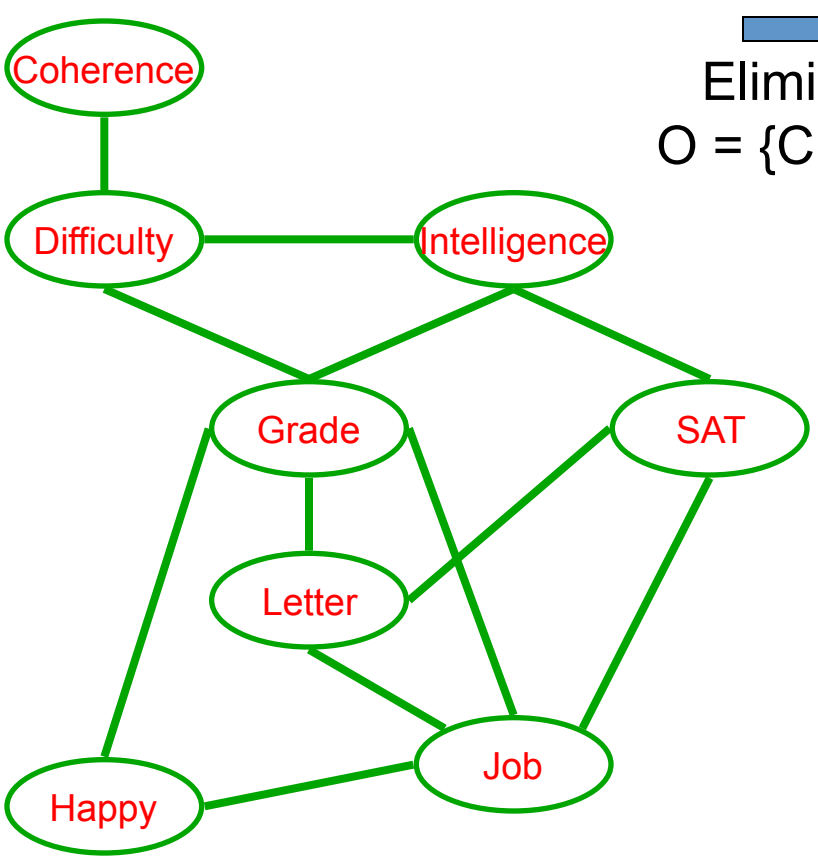
- VE steps on board

Variable Elimination

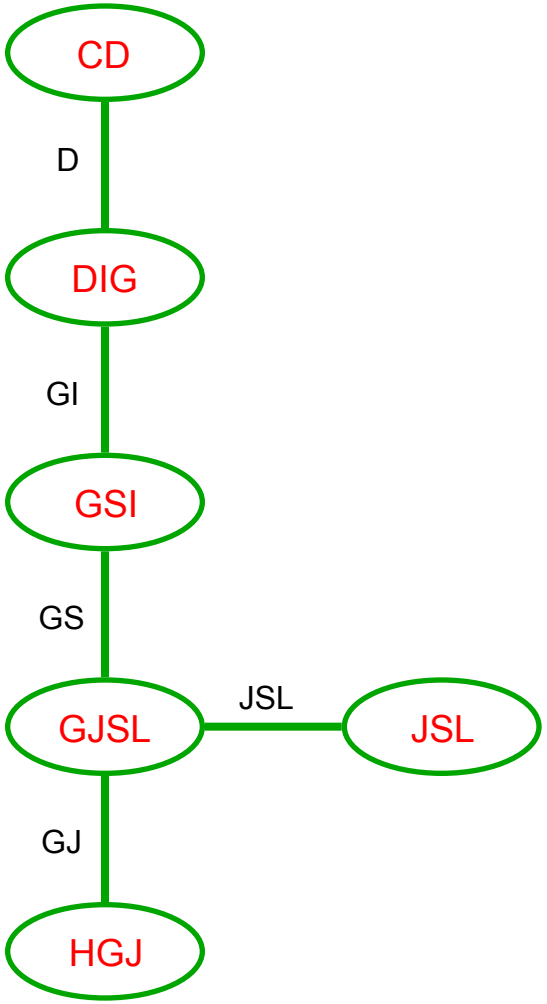


Elimination order:
O = {C,D,I,H,G,S,L,J}

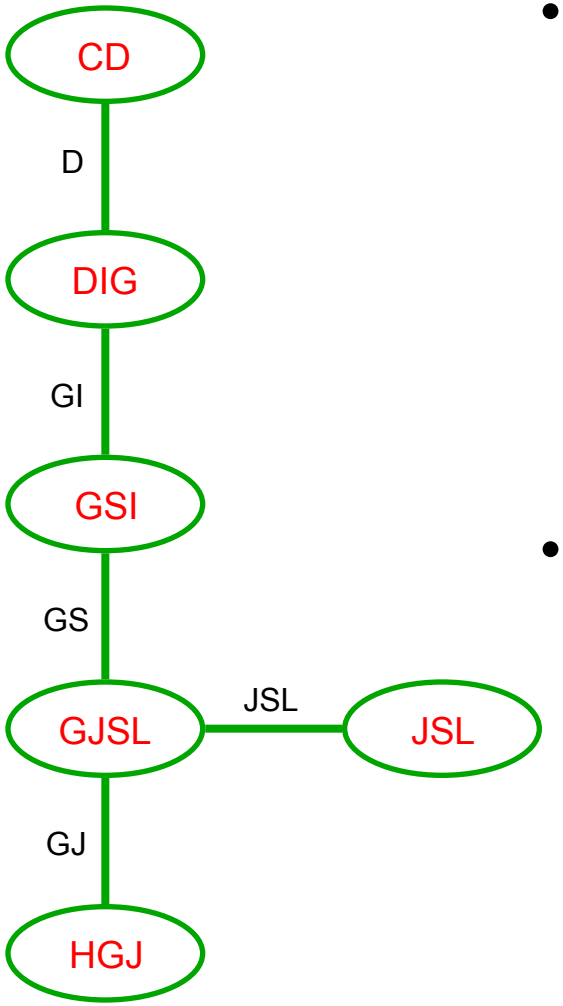
Factors Generated



Elimination order:
O = {C,D,I,H,G,S,L,J}

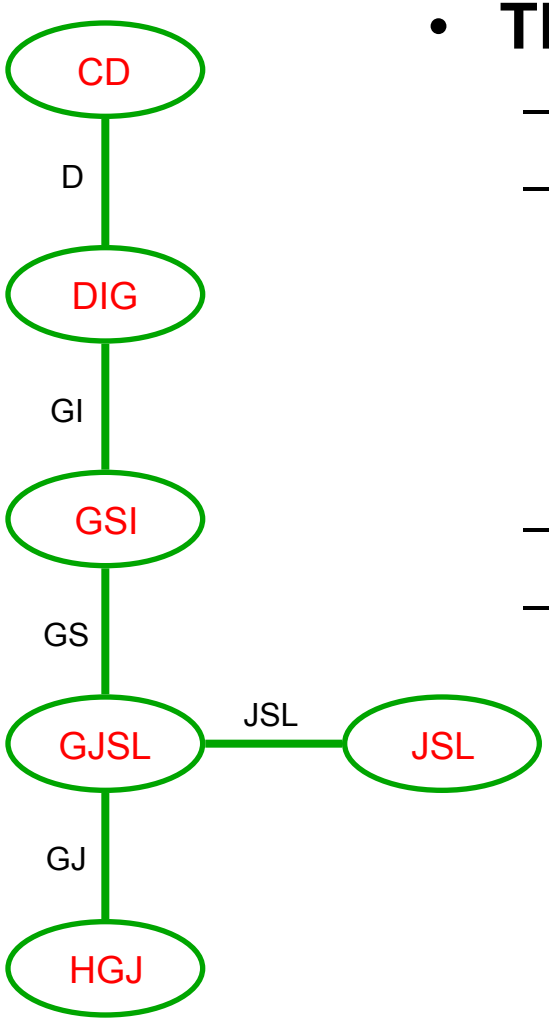


Cluster graph for VE



- **VE generates cluster tree!**
  (Also called Clique Tree or Junction Tree)
  - One cluster for each factor used/generated
  - Edge i – j, if f_i used to generate f_j
  - "Message" from i to j generated when marginalizing a variable from f_i
  - Tree because factors only used once
- **Proposition**:
  - "Message" δ_{ij} from i to j
  - Scope[δ_{ij}] $\subseteq \mathbf{S}_{ij}$

Clique tree & Independencies



- **Theorem**:
  - Given some BN/MN with structure *G* and factors *F*
  - For a clique tree *T* for *F* consider $\mathbf{C}_i$ – $\mathbf{C}_j$ with separator $\mathbf{S}_{ij}$:
    - **X** – any set of vars in $\mathbf{C}_i$ side of the tree
    - **Y** – any set of vars in $\mathbf{C}_i$ side of the tree
  - Then, $(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{S}_{ij})$ in BN/MN
  - Furthermore, $I(T) \subseteq I(G)$

MRF: Exact Inference: What you need to know

- Types of queries
  - Conditional probabilities / Marginals
  - maximum a posteriori (MAP)
  - Marginal-MAP
- Hardness of inference
- Variable elimination algorithm
  - Essentially the same algorithm as for BN
  - Eliminate a variable:
    - Combine factors that include this var into single factor
    - Marginalize/Maximize var from new factor
- VE can be viewed as one pass of BP on Clique Tree
  - Can solve marginal queries for all variables in only twice the cost of query for one variable
- Clique Tree / Cluster Tree / Junction Tree
  - Cliques correspond to maximal cliques in induced graph
  - Constructing clique tree for a BN/MN from elimination order
- Clique tree invariant
  - We are only reparameterizing clique potentials
- Running time (only) exponential in size of largest clique
  - Solve **exactly** problems with thousands (or millions, or more) of variables, and cliques with tens of nodes (or less)

Approximate Inference

- So far: Exact Inference
  - VE & Junction Trees
  - Exponential in tree-width
- There are many many approximate inference algorithms for PGMs
  - You have already seen BP
- Next
  - Variational Inference
  - Connections to BP / Message-Passing

What is Variational Inference?

- A class of methods for approximate inference
  - And parameter learning
  - And approximating integrals basically..

- Key idea
  - Reality is complex
  - Instead of performing approximate computation in something complex
  - Can we perform exact computation in something “simple”?
  - Just need to make sure the simple thing is “close” to the complex thing.

- Key Problems
  - What is close?
  - How do we measure closeness when we can’t perform operations on the complex thing?

KL divergence: Distance between distributions

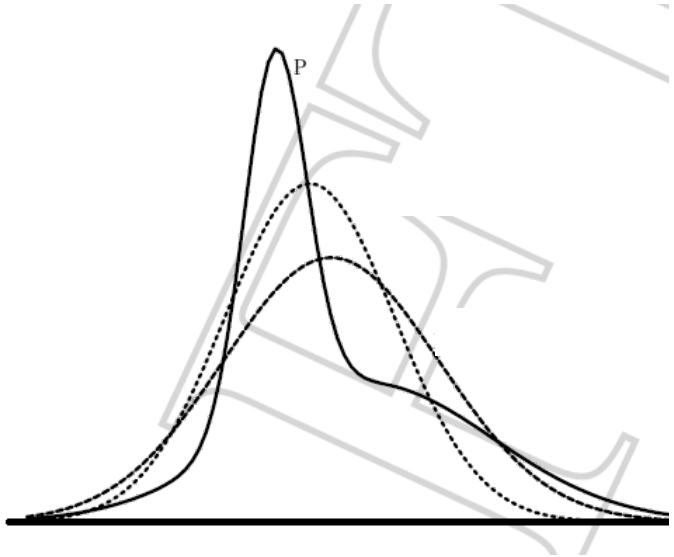
- Given two distributions *p* and *q* KL divergence:

- D(p||q) = 0 iff p=q

- Not symmetric – p determines where difference is important

Find simple approximate distribution

- Suppose *p* is intractable posterior
- Want to find simple *q* that approximates *p*
- KL divergence not symmetric

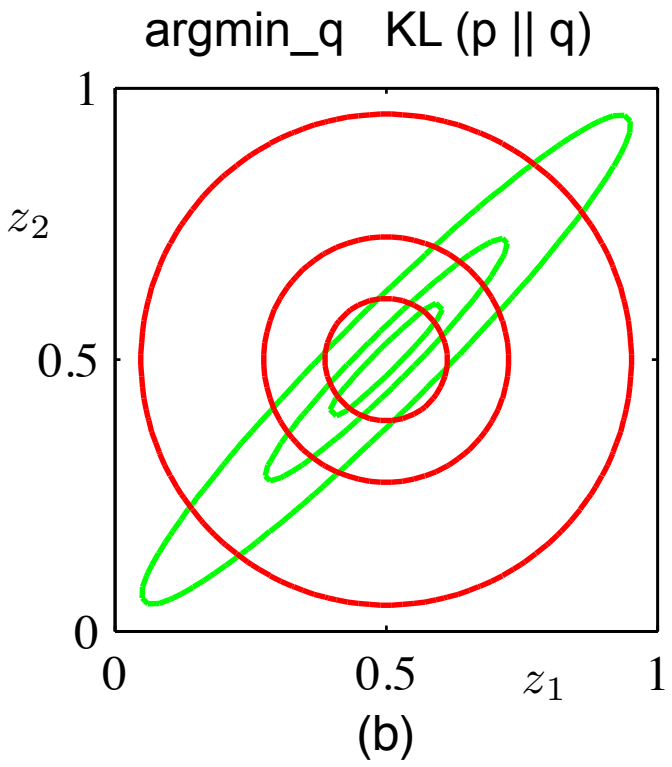


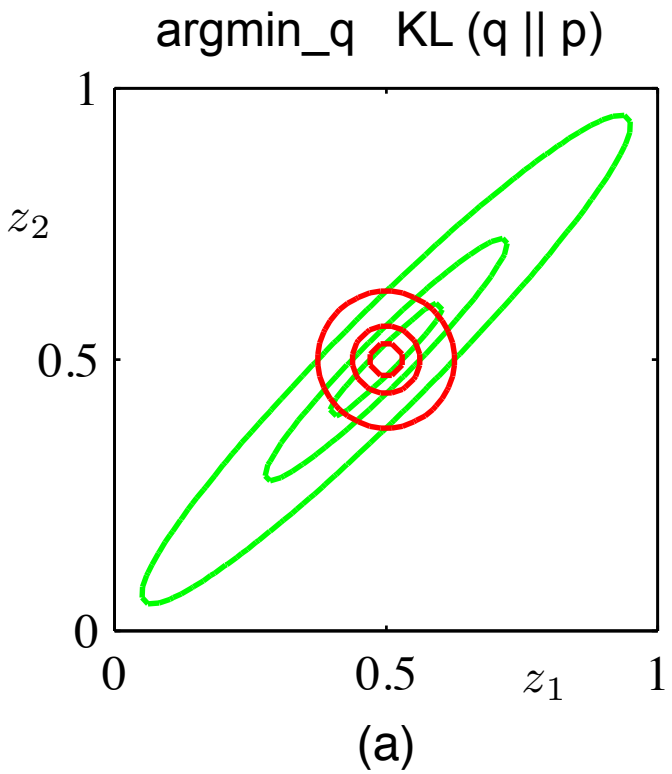
- D(p||q)
  - true distribution p defines support of diff.
  - the “correct” direction
  - will be intractable to compute

- D(q||p)
  - approximate distribution defines support
  - tends to give overconfident results
  - will be tractable

Example 1

- p = 2D Gaussian with arbitrary co-variance
- q = 2D Gaussian with diagonal co-variance



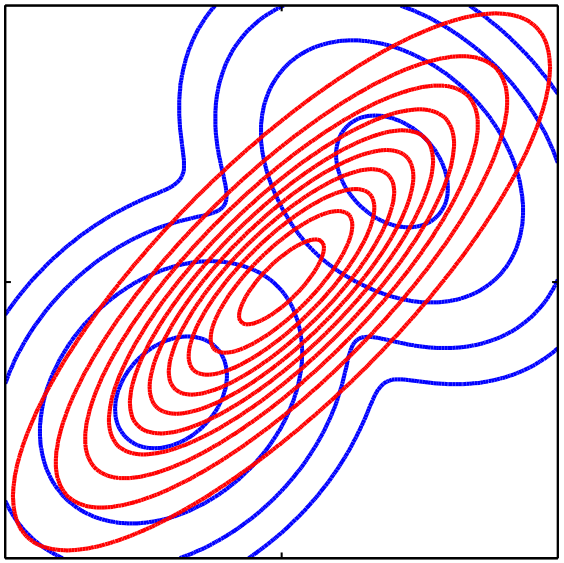


p = Green; q = Red

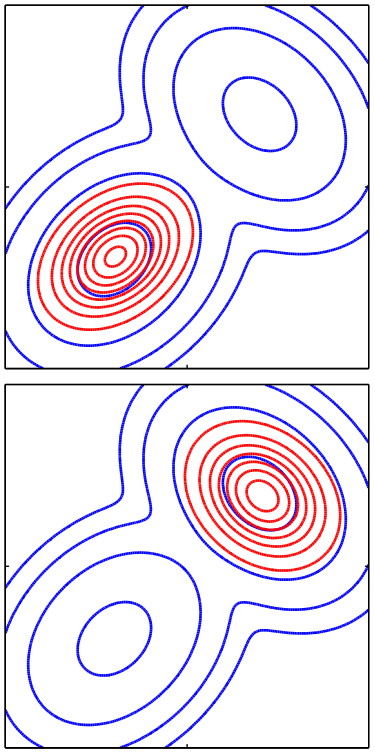
Example 2

- p = Mixture of Two Gaussians
- q = Single Gaussian

argmin_q KL (p || q)



argmin_q KL (q || p)



p = Blue; q = Red

Back to graphical models

- Inference in a graphical model:
  - P(**x**) =
  - want to compute $P(X_i)$
  - our *p*:

- What is the simplest *q*?
  - every variable is independent:
  - mean field approximation
  - can compute any prob. very efficiently

Variational Approximate Inference

$$p(x) = \frac{1}{Z} \prod_{(s,t)\in\mathcal{E}} \psi_{st}(x_s, x_t) \prod_{s\in\mathcal{V}} \psi_s(x_s)$$

- Choose a family of approximating distributions which is tractable. The simplest [Mean Field] Approximation:

$$q(x) = \prod_{s\in\mathcal{V}} q_s(x_s)$$

- Measure the quality of approximations. Two possibilities:

$$D(p \,||\, q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \qquad D(q \,||\, p) = \sum_x q(x) \log \frac{q(x)}{p(x)}$$

- Find the approximation minimizing this distance

D(p||q) for mean field – KL the right way

- D(p||q)=

- Trivially minimized by setting $q_i(x_i) = p_i(x_i)$
- Doesn't provide a computational method…

D(q||p) for mean field – KL the reverse direction

- D(q||p)=

$$D(q||p) = \sum_x q(x) \log q(x) - \sum_x q(x) \log p(x)$$