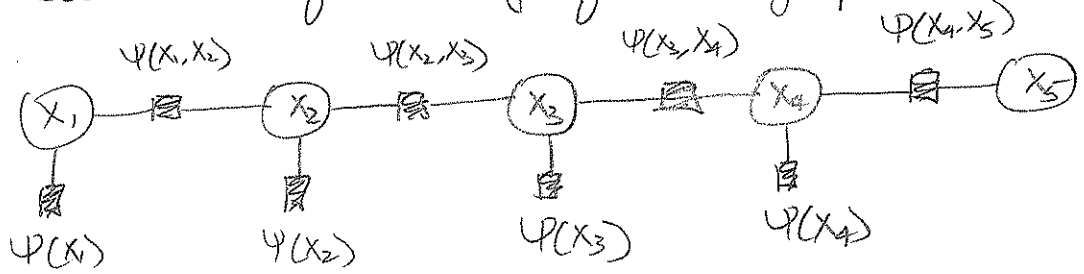
4/1/14



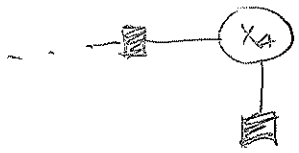
# MRF INFERENCE: CONNECTING BP & VE

① Consider the following factor graph:




Query: Find $P(X_1 \mid X_5 = x_5) \propto P(X_1, X_5 = x_5)$

Step 1: Instantiate Evidence:




$\hat{\psi}(X_4) = \psi(X_4) \cdot \psi(X_4, X_5 = x_5)$

new node potential

Step 2: Running VE → with Elim Ordering $\{X_4, X_3, X_2\}$

& BP → with message schedule

$\{(4,3), (3,2), (2,1), (1,2), (2,3), (3,4)\}$

| Idx | VE | | BP |
|---|---|---|---|
| 1 | $\sum_{X_4} \hat{\psi}(X_4)\psi(X_3, X_4)$ | $\equiv$ | $\delta_{4\to 3}(X_3)$ |
| 2 | $\sum_{X_3} \psi(X_3)\psi(X_2, X_3)\delta_{4\to 3}(X_3)$ | $\equiv$ | $\delta_{3\to 2}(X_2)$ |
| 3 | $\sum_{X_2} \psi(X_2)\psi(X_2, X_1)\delta_{3\to 2}(X_2)$ | $\equiv$ | $\delta_{2\to 1}(X_1)$ |

Note 1: Up to this point, VE & BP operations are IDENTICAL
VE creates intermediate factors. } same
BP calls them messages.

Note 2: At this point, VE stops & proceeds to compute

$$P(X_1) \propto \psi(X_1)\, \delta_{2\to 1}(X_1)$$

But BP goes on in a "backward pass".

Note 3: After BP finishes backward pass of messages, we claim it has converged & no further message-passing is required. Why? Try working out $\delta_{4\to 3}(X_3)$. What do you find?

Key idea: leaves in trees only need to send messages ONCE. Why? They don't multiply any incoming messages. So always done!

Note 4: After VE is done, it can only answer

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

After BP is done, it can answer

$$P(X_i \mid X_5 = x_5) \quad \boxed{\forall i} \quad !$$ Cool!

$$\propto \psi(X_i) \cdot \prod_{j \in N(i)} \delta_{j\to i}(X_i)$$

At only 2x cost of VE, we get so much more.

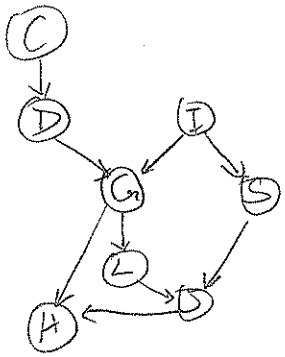
## (2) Generalizing Connection of VE & BP beyond chains.

→ Key Idea: As VE is running on an arbitrary BN/MN, it creates something called a Clique Tree / Junction Tree.

This JT is in fact a valid cluster graph that BP can be run on.

Similar to chains, only need to run BP for 2 passes (Leaves → Root → Leaves)

Example:

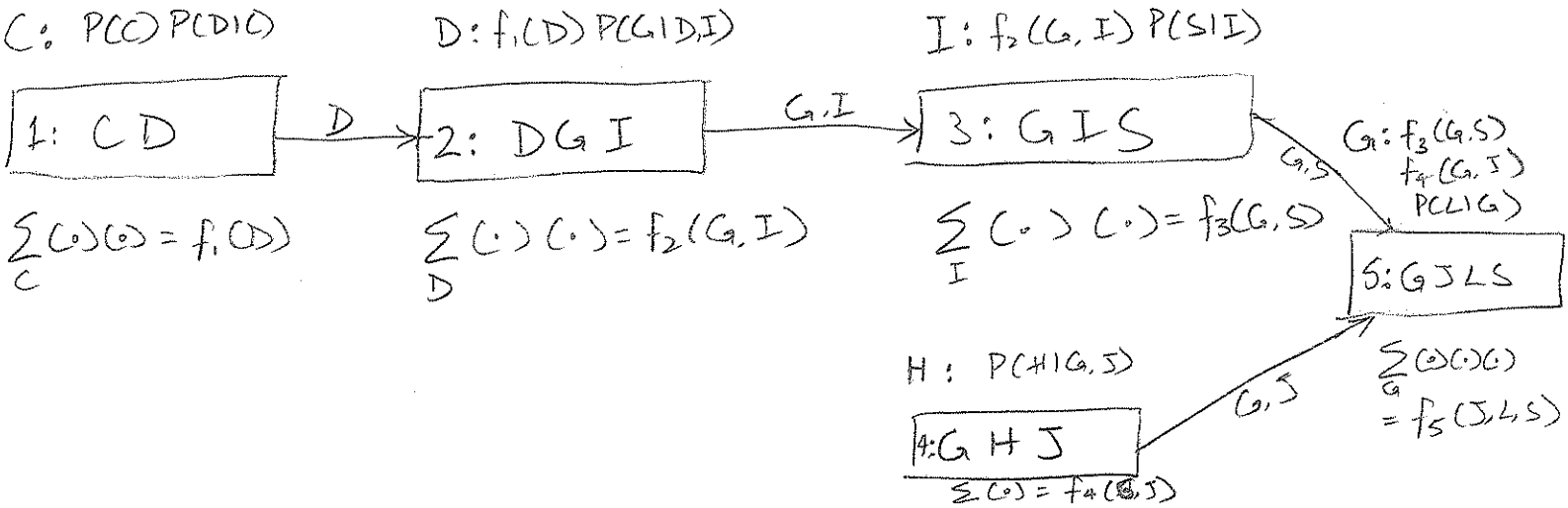



Elim Ordering: {C, D, I, H, G, S, L, J}

→ Let's eliminate

→ Create clusters corresponding to factor multiplication

→ Add edges whenever an intermediate factor is "consumed" in another elimination



Note 1: Claim: This Junction Tree is a Cluster Graph
∵ it satisfies (a) Fam Preserving Prop
(b) RIP

Note 2: If we run BP with a Leaves → Root → Leaves schedule (as shown by directionality of edges) the first part Leaves → Root will be IDENTICAL to VE.

Note 3: Again, after backward pass BP will be converged.
⇒ Cluster graph calibrated
⇒ ready to answer queries $P(X_i)$

---

## (3) Variational Inference:

$P(\vec{X})$: Some complex distribution
→ maybe BN, maybe MN
→ maybe "reality"

→ Want to answer $P(X_i)$. Too hard.

Key Idea:
→ How about we approximate $P(X)$ with a "simple" distribution $Q(X)$ that is "close" to $P(\vec{X})$, but allows easy computation of $Q(X_i)$.

Specifically,
Our goal

$$P(\vec{X}) = \frac{1}{Z} \prod_c \Psi_c(X_c) \qquad \text{MRF}$$

↑ Notation means variables involved in c'th clique

$$\text{e.g.} = \frac{1}{Z} \prod_i \Psi_i(X_i) \prod_{(i,j) \in E} \Psi_{ij}(X_i, X_j)$$

[Pairwise MRF]

P is too complex!

VERY VERY Simple

How about we approximate P with something

Q: a fully factorized distribution

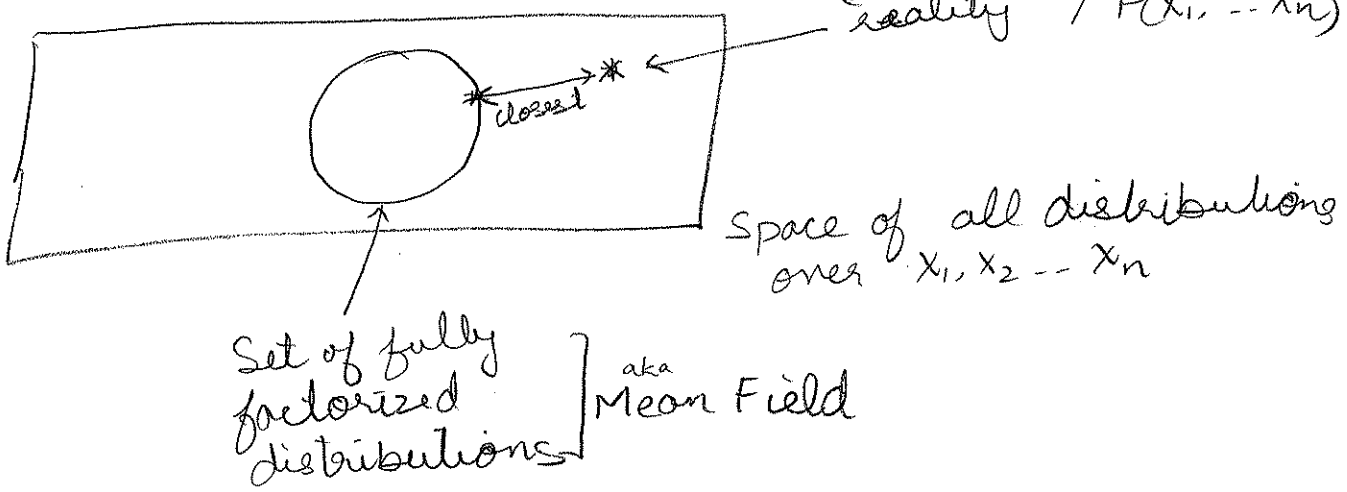
$$= \prod_i Q_i(X_i)$$

Once we find the Q closest to P, it is trivial to find marginals. Just read $Q_i(X_i)$

Want pairwise marginal?

No problem! $Q_{ij}(X_i, X_j) = Q_i(X_i) Q_j(X_j)$

OK. So how do we find a good Q?




Minimize a measure of "distance" between P & Q

KL-divergence!

Two possibilities → $KL(p||q) = \sum_{\vec{x}} p(\vec{x}) \log \frac{p(\vec{x})}{q(\vec{x})}$

∴ KL is not symetric → $KL(q||p) = \sum_{\vec{x}} q(\vec{x}) \log \frac{q(\vec{x})}{p(\vec{x})}$

Note 1: $\min_q KL(p||q)$ is the "correct" direction.

Why? ∵ it weights $\log \frac{p(x)}{q(x)}$ with $p(x)$ — the "true" prob of this event $\vec{x}$.

The other direction gets to ignore certain events by setting $q(x) <<< 1$ where $q$ is a bad approximation of $p$.

See examples in slides.

## ④ Back to Graphical Models:

### Variational Inference: the correct direction

$$p(\vec{x}) = \frac{1}{Z} \prod_c \psi_c(x_c) \qquad q(\vec{x}) = \prod_i q_i(x_i)$$

Goal: $\underset{q}{\operatorname{argmin}} \; KL(p \| q) = \underbrace{\sum_{\vec{x}} p(x) \log p(x)}_{\text{constant w.r.t } q} - \sum_{\vec{x}} p(x) \log q(x)$

$$= \underset{q}{\operatorname{argmax}} \sum_{\vec{x}} p(x) \log q(x)$$

$$= \sum_{\vec{x}} p(x) \log \prod_i q_i(x_i)$$

$$= \sum_{\vec{x}} p(\vec{x}) \sum_i \log q_i(x_i)$$

$$= \sum_i \sum_{\vec{x}} p(\vec{x}) \log q_i(x_i)$$

$$= \sum_i \sum_{x_i} p_i(x_i) \log q_i(x_i)$$

Thus optimization problem

$$\underset{\{q_i\}}{\operatorname{argmax}} \sum_i \sum_{x_i} p_i(x_i) \log q_i(x_i)$$

$$\sum_{x_i} q_i(x_i) = 1 \qquad \forall i$$

$$q_i(x_i) \geq 0$$

Write Lagrangian $L(\{q_i\}, \{\lambda_i\}) = \sum_i \sum_{x_i} p_i(x_i) \log q_i(x_i) - \sum_i \lambda_i \left[ \sum_{x_i} q_i(x_i) - 1 \right]$

Now set $\frac{\partial \mathcal{L}}{\partial q_i(x_i)} = 0$

$$\frac{p_i(x_i)}{q_i(x_i)} - \lambda_i = 0$$

[subtle point: assuming $q_i(x_i) \neq 0$]

$$\Rightarrow \quad q_i(x_i) = \frac{p_i(x_i)}{\lambda_i}$$

$$\sum q_i(x_i) = 1 = \sum p_i(x_i)$$

$$\Rightarrow \lambda_i^* = 1$$

Thus $\boxed{q_i^*(x_i) = p_i(x_i)}$

Doesn't quite help!

Need to know marginals to compute Q.

But we were computing Q to get marginals!