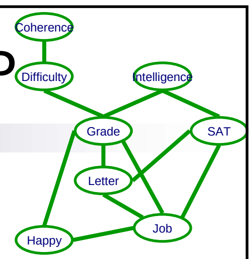
Readings:
K&F: 11.3, 11.5
Yedidia et al. paper from the class website
Chapter 9 - Jordan

Loopy Belief Propagation
Generalized Belief Propagation
Unifying Variational and GBP
Learning Parameters of MNs

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

November 10th, 2006

More details on Loopy BP



- Numerical problem:
  - messages < 1 get multiplied together as we go around the loops
  - numbers can go to zero
  - normalize messages to one:

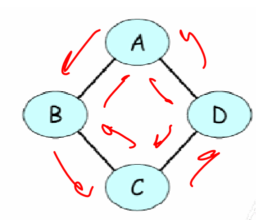
$$\delta_{i\to j}(X_j) = \frac{1}{Z_{i\to j}} \sum_{x_i} \phi_i(x_i)\phi_{ij}(x_i, X_j) \prod_{k\in\mathcal{N}(i)-j} \delta_{k\to i}(x_i)$$

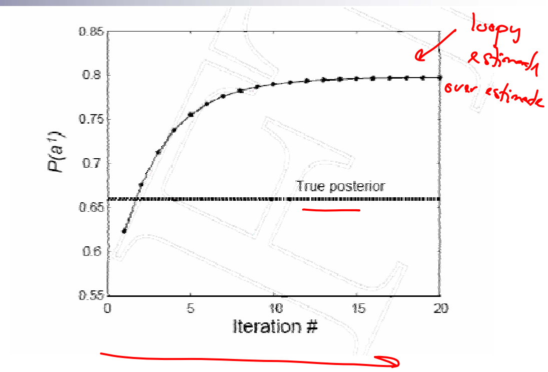
  - $Z_{i\to j}$ doesn't depend on X_j, so doesn't change the answer

- Computing node "beliefs" (estimates of probs.):

$$\hat{P}(X_i) = \frac{1}{Z_i}\phi_i(X_i) \prod_{k\in\mathcal{N}(i)} \delta_{k\to i}(X_i)$$

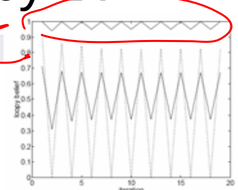
An example of running loopy BP





usually over confident...

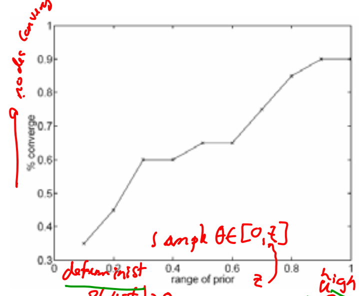
(Non-)Convergence of Loopy BP



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

beliefs in BP

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

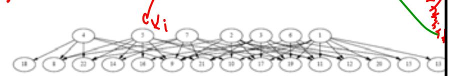


momentum

- One approach to help: damping messages
  - new message is average of old message and new one: $\lambda \in [0,1)$

$$\delta_{i\to j} = \lambda\, \underbrace{\delta_{i\to j}}_{\text{last iteration}} + (1-\lambda)\, \underbrace{\delta_{i\to j}}_{\text{computed}}$$

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



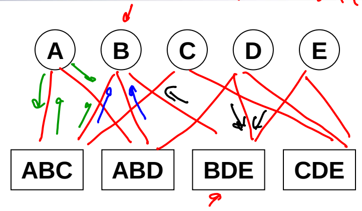
graphs from Murphy et al. '99

Loopy BP in Factor graphs

$P(x) = \frac{1}{Z}\,\psi(ABC)\cdot\psi(ABD)\cdot\psi(BDE)\cdot\psi(CDE)$

- What if we don't have pairwise Markov nets?
  1. Transform to a pairwise MN
  2. Use Loopy BP on a factor graph

ψ(BDE) never needs to be turned into a table



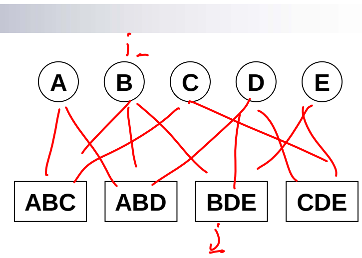
- Message example:
  - from node to factor: $\delta_{B\to BDE}(B) = \delta_{ABC\to B}(B)\;\delta_{ABD\to B}(B)$
  - from factor to node: $\delta_{BDE\to B}(B) = \frac{1}{Z}\sum_{DE}\psi(BDE)\;\delta_{D\to BDE}(D)\;\delta_{E\to BDE}(E)$

10-708 – ©Carlos Guestrin 2006 5

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**∪{X_i}

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

- Estimate of belief?

$\hat{P}(X_i) = \frac{1}{Z}\prod_{\phi_j : X_i\in Scope[\phi_j]} \delta_{j\to i}(X_i)$

Scope[ϕ_j] = C_j

$\hat{P}(C_j) = \frac{1}{Z}\,\phi_j(C_j)\cdot\prod_{X_i\in Scope[\phi_j]}\delta_{i\to j}(X_i)$

10-708 – ©Carlos Guestrin 2006 6

What you need to know about loopy BP

- Application of belief propagation in loopy graphs
- Doesn't always converge
  - damping can help
  - good message schedules can help (see book)
- If converges, often to incorrect, but useful results
- Generalizes from pairwise Markov networks by using factor graphs

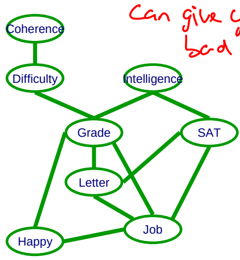
Announcements

- Monday's special recitation
  - Pradeep Ravikumar on exciting new approximate inference algorithms

Loopy BP v. Clique trees: Two ends of a spectrum

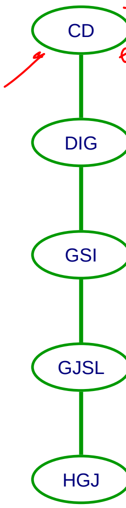
Loopy BP ⇒ messages small: exponential in 2 or size of factors, can give you bad answers

Clique trees ⇒ Exact sol, exp. clique size



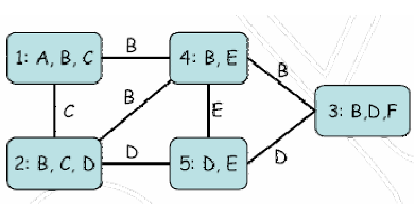
smaller running time than better solutions

?



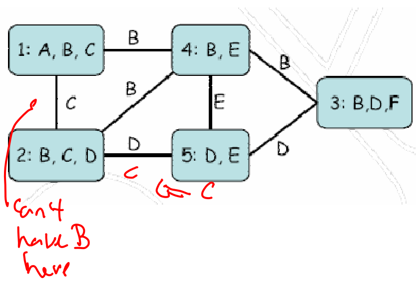
Generalize cluster graph

$C_1 \cap C_2 = \{B, C\}$



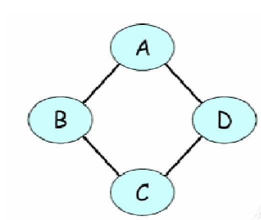
- **Generalized cluster graph**: For set of factors *F*
  - Undirected graph
  - Each node i associated with a cluster $\mathbf{C}_i$
  - *Family preserving*: for each factor $f_j \in F$, $\exists$ node i such that scope$[f_j] \subseteq \mathbf{C}_i$
  - Each edge i – j is associated with a set of variables $\mathbf{S}_{ij} \subseteq \mathbf{C}_i \cap \mathbf{C}_j$

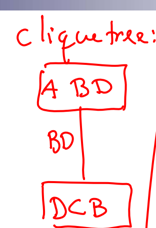
Running intersection property

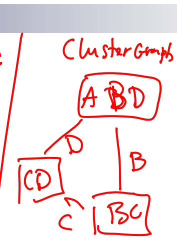


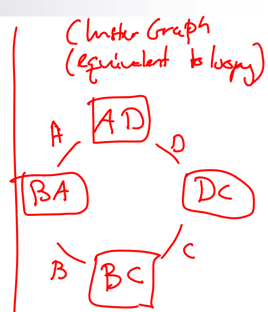
- **(Generalized) Running intersection property (RIP)**
  - Cluster graph satisfies RIP if whenever $X \in \mathbf{C}_i$ and $X \in \mathbf{C}_j$ then $\exists$ *one and only one path* from $\mathbf{C}_i$ to $\mathbf{C}_j$ where $X \in \mathbf{S}_{uv}$ for every edge (u,v) in the path

Examples of cluster graphs









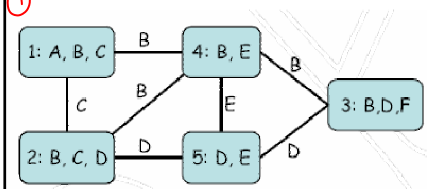
$$P(ABCD) = \frac{1}{Z}\,\varphi(AB)\,\varphi(AD)\,\psi(BC)\,\psi(CD)$$

Cluster-Graph from Loopy always satisfies generalized RIP

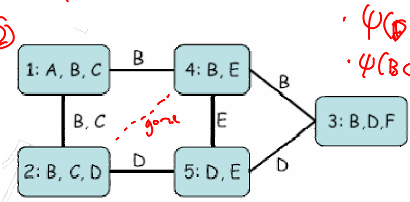
Two cluster graph satisfying RIP with different edge sets

$P(X) = \frac{1}{Z}\, \psi(AB)\psi(CD)\psi(AC)\psi(BE)\psi(DE)\cdot\psi(BD) \cdot \psi(DF) \cdot \psi(BC)$

①



②

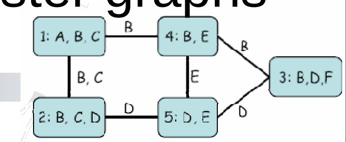


initialization: $\pi_2^0(BCD) = \frac{1}{Z}\, \psi(CD)\psi(BD)\cdot\psi(BC)$; $\delta_{i\to j} = 1$

messages: ① $\delta_{2\to 1}(C) = \frac{1}{Z} \sum_{BD} \pi_2^0(BCD)\, \delta_{4\to 2}(B)\, \delta_{5\to 2}(D)$

② $\delta_{2\to 1}(BC) = \frac{1}{Z} \sum_{D} \pi_2^0(BCD)\, \delta_{5\to 2}(D)$

Generalized BP on cluster graphs satisfying RIP



- Initialization:
  - Assign each factor ϕ to a clique $\alpha(\phi)$, Scope$[\phi] \subseteq \mathbf{C}_{\alpha(\phi)}$
  - Initialize cliques: $\pi_i^0(\mathbf{C}_i) \propto \prod_{\phi:\alpha(\phi)=i} \phi$
  - Initialize messages: $\delta_{j\to i} = 1$
- While not converged, send messages:

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

sum out vars. not in edge

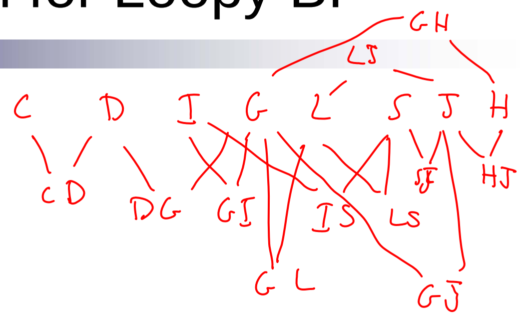
- Belief: $\hat{P}(C_i) = \frac{1}{Z}\, \pi_i^0(C_i) \cdot \prod_{k \in N(i)} \delta_{k\to i}(S_{ik})$

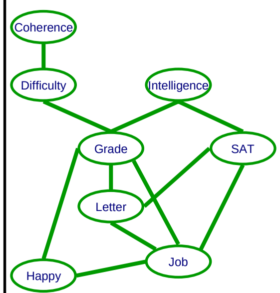
$\hat{P}(S_{ij}) = \frac{1}{Z}\, \delta_{i\to j} \cdot \delta_{j\to i}$

if converged

$\hat{P}(S_{ij}) = \sum_{C_i - S_{ij}} \hat{P}(C_i)$

Cluster graph for Loopy BP





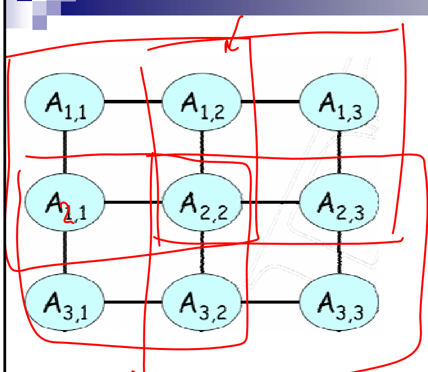
$$\delta_{GH \to HJ}(H) =$$

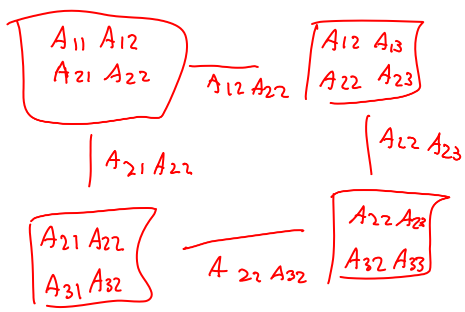
$$\frac{1}{Z} \sum_{G} \pi^{0}_{GH}(G,H) \; \delta_{GL \to GH}(G)$$

$$\uparrow \psi(H) \cdot \psi(G\,H)$$

What if the cluster graph doesn't satisfy RIP

natural Graph:





doesn't satisfy RIP

alg. from 2 slides ago only for RIP cluster graphs

Region graphs to the rescue

- Can address generalized cluster graphs that don't satisfy RIP using *region graphs*:
  - Yedidia et al. from class website
- Example in your homework! ☺
- Hint – From Yedidia et al.: 2005 (2001 paper easier to read)
  - Section 7 – defines region graphs
  - Section 9 – message passing on region graphs
  - Section 10 – An example that will help you a lot!!! ☺

Revisiting Mean-Fields

$$\ln Z = F[P_{\mathcal{F}}, Q] + D(Q||P_{\mathcal{F}}) \quad F[P_{\mathcal{F}}, Q] = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + H_Q(\mathcal{X})$$

constant

decompose

- Choice of Q: $Q(X) = \prod_i Q_i(x_i)$
- Optimization problem:

$$\max_{Q_i} \sum_{\phi \in F} E_Q[\ln \phi] + \sum_i H_{Q_i}(X_i)$$

$$\text{s.t. } \sum_{x_i} Q_i(x_i) = 1$$

$$Q_i(x_i) \geq 0$$

$$\max_Q F[P_{\mathcal{F}}, Q] = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + \sum_j H_{Q_j}(X_j), \quad \forall i, \sum_{x_i} Q_i(x_i) = 1$$

Interpretation of energy functional

- Energy functional: $F[P_{\mathcal{F}},Q] = \sum_{\phi\in\mathcal{F}} E_Q[\ln\phi] + H_Q(\mathcal{X})$
- Exact if P=Q: $\ln Z = F[P_{\mathcal{F}},Q] + D(Q||P_{\mathcal{F}})$

  $Q \in P$ maximized $\rightarrow 0$

- View problem as an approximation of entropy term:

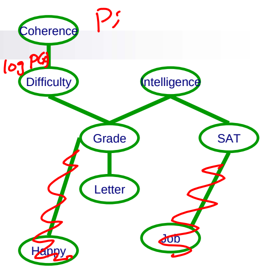
estimate expectation w.r.t. Q

approximating $H_Q(X) \approx H_P(X)$

interpretation: $H_p(X) \approx \sum_i H_{Q_i}(X_i)$

Entropy of a tree distribution

- Entropy term: $H_P(X) = -E_P[\log P(X)] = -\sum_x p(x)\log p(x)$
- Joint distribution: $P(x) = \frac{1}{Z}\prod_{ij}\varphi_{ij}(x_i,x_j)$



P_i

- Decomposing entropy term:

$$P(X) = \frac{P(CD)\cdot P(DG)\cdot P(GI)\cdot P(JS)\, P(GL)}{P(D)\; P(G)\; P(G)\cdot P(J)}$$

CD
ID
DG – GI (G)
IG
GL
IS, SJ

$H_P(X) = -E_P[\log P(x)] =$

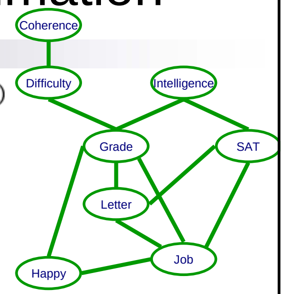
clique

$-[E_P[\log P(CD)] + E_P[\log P(GD)] + \dots]$

$-[E_P[\log P(D)] + E_P[\log P(G)] + E_P[\log P(G)] \dots]$

- More generally: $H_P(X) = \sum_{(i,j)\in E} H(X_i,X_j) - \sum_i (d_i - 1)H(X_i)$
  - d_i number neighbors of X_i

Loopy BP & Bethe approximation

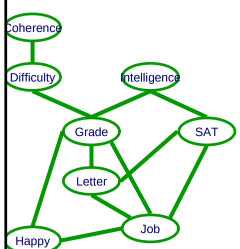


- Energy functional: $F[P_{\mathcal{F}}, Q] = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + H_Q(\mathcal{X})$
- **Bethe approximation of Free Energy**:
  - use entropy for trees, but loopy graphs:

$$\tilde{F}[P_{\mathcal{F}}, Q] = \sum_{(i,j)\in E} E_{\pi_{ij}}[\ln \phi_{ij}] + \sum_{(i,j)\in E} H_{\pi_{ij}}(X_i, X_j) - \sum_i (d_i - 1) H_{\pi_i}(X_i)$$

- **Theorem**: If Loopy BP converges, resulting π_{ij} & π_i are stationary point (usually local maxima) of Bethe Free energy!

GBP & Kikuchi approximation



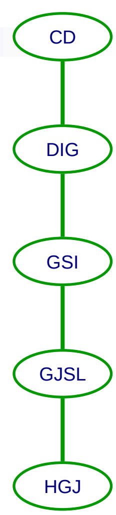
- Exact Free energy: Junction Tree

$$F[P_{\mathcal{F}}, Q] = \sum_{(i,j)\in E} E_{\pi_{ij}}[\ln \phi_{ij}] + \sum_i H_{\pi_{\mathbf{C}_i}}(\mathbf{C}_i) - \sum_{(i,j)\in \mathcal{T}} H_{\pi_{\mathbf{S}_{ij}}}(\mathbf{S}_{ij})$$

- Bethe Free energy:

$$\tilde{F}[P_{\mathcal{F}}, Q] = \sum_{(i,j)\in E} E_{\pi_{ij}}[\ln \phi_{ij}] + \sum_{(i,j)\in E} H_{\pi_{ij}}(X_i, X_j) - \sum_i (d_i - 1) H_{\pi_i}(X_i)$$

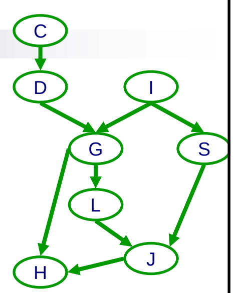
- Kikuchi approximation: Generalized cluster graph
  - spectrum from Bethe to exact
  - entropy terms weighted by counting numbers
  - see Yedidia et al.
- **Theorem**: If GBP converges, resulting $\pi_{\mathbf{C}i}$ are stationary point (usually local maxima) of Kikuchi Free energy!



What you need to know about GBP

- Spectrum between Loopy BP & Junction Trees:
  - More computation, but typically better answers
- If satisfies RIP, equations are very simple
- General setting, slightly trickier equations, but not hard
- Relates to variational methods: Corresponds to local optima of approximate version of energy functional

Learning Parameters of a BN



- Log likelihood decomposes:

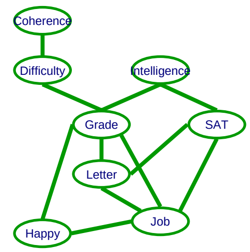
$$\ell(\mathcal{D}:\theta) = \log P(\mathcal{D} \mid \theta) = m \sum_i \sum_{x_i, \mathbf{Pa}_{x_i}} \hat{P}(x_i, \mathbf{Pa}_{x_i}) \log P(x_i \mid \mathbf{Pa}_{x_i})$$

- Learn each CPT independently
- Use counts

$$\hat{P}(\mathbf{u}) = \frac{\text{Count}(\mathbf{U}=\mathbf{u})}{m}$$

Log Likelihood for MN

- Log likelihood of the data:

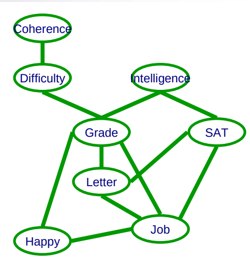


Log Likelihood doesn't decompose for MNs

$$\hat{P}(\mathbf{u}) = \frac{\text{Count}(\mathbf{U}=\mathbf{u})}{m}$$

- Log likelihood:

$$\ell(\mathcal{D}:\theta) = \log P(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i) \log \psi_i(\mathbf{c}_i) - m \log Z$$

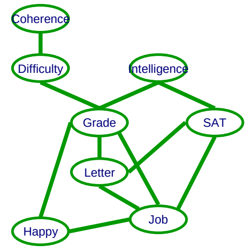


- A convex problem
  - Can find global optimum!!
- Term log Z doesn't decompose!!

Derivative of Log Likelihood for MNs

$$\hat{P}(\mathbf{u}) = \frac{\text{Count}(\mathbf{U}=\mathbf{u})}{m}$$

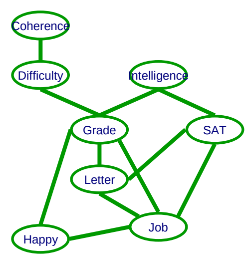
$$\ell(\mathcal{D}:\theta) = \log P(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i) \log \psi_i(\mathbf{c}_i) - m \log Z$$



Derivative of Log Likelihood for MNs

$$\hat{P}(\mathbf{u}) = \frac{\text{Count}(\mathbf{U}=\mathbf{u})}{m}$$

$$\ell(\mathcal{D}:\theta) = \log P(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i) \log \psi_i(\mathbf{c}_i) - m \log Z$$



- Derivative:

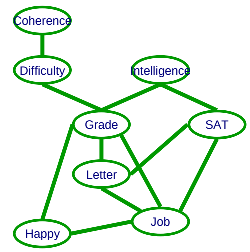
$$\frac{\partial \ell}{\partial \psi_i(\mathbf{c}_i)} = \frac{m\hat{P}(\mathbf{c}_i)}{\psi_i(\mathbf{c}_i)} - \frac{mP_{\mathcal{F}}(\mathbf{c}_i)}{\psi_i(\mathbf{c}_i)}$$

- Setting derivative to zero
- Can optimize using gradient ascent
  - Let's look at a simpler solution

Iterative Proportional Fitting (IPF)

$$\hat{P}(\mathbf{u}) = \frac{\text{Count}(\mathbf{U}=\mathbf{u})}{m}$$

$$\frac{\partial \ell}{\partial \psi_i(\mathbf{c}_i)} = \frac{m\hat{P}(\mathbf{c}_i)}{\psi_i(\mathbf{c}_i)} - \frac{mP_{\mathcal{F}}(\mathbf{c}_i)}{\psi_i(\mathbf{c}_i)}$$



- Setting derivative to zero:
- Fixed point equation:
- Iterate and converge to optimal parameters
  - Each iteration, must compute:

What you need to know about learning MN parameters?

- BN parameter learning easy
- MN parameter learning doesn't decompose!
- Learning requires inference!
- Apply gradient ascent or IPF iterations to obtain optimal parameters