Readings:
K&F: 10.1, 10.5

Mean Field and Variational Methods
finishing off

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

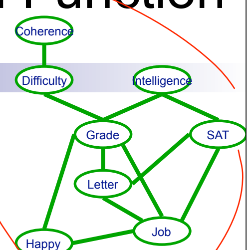
November 5th, 2008

What you need to know so far

- Goal:
  - Find an efficient distribution that is close to posterior
- Distance:
  - measure distance in terms of KL divergence
- Asymmetry of KL:
  - D(p||q) ≠ D(q||p)
- Computing right KL is intractable, so we use the reverse KL

Reverse KL & The Partition Function

Back to the general case



- Consider again the defn. of D(q||p):
  - p is Markov net $P_{\mathcal{F}}$
- **Theorem**: $\ln Z = F[P_{\mathcal{F}}, Q] + D(Q||P_{\mathcal{F}})$
- where energy functional:

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

Understanding Reverse KL, Energy Function & The Partition Function

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

(constant)

- Maximizing Energy Functional ⇔ Minimizing Reverse KL

$D(q||p) \geq 0$

- **Theorem**: Energy Function is lower bound on partition function

$F(P_F, Q) + D(Q||P_F) = \log Z$ (with $D \geq 0$)

$\log Z \geq F[P_F, Q]$ ← what we maximize

  - Maximizing energy functional corresponds to search for tight lower bound on partition function

don't know how to compute Z, so we will try to find a lower bound

Structured Variational Approximate Inference

$\ln Z = F[P_{\mathcal{F}}, Q] + D(Q||P_{\mathcal{F}})$

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

- Pick a family of distributions Q that allow for exact inference
  - e.g., fully factorized (mean field) $q(x) = \prod_j Q_j(x_j)$
- Find Q∈Q that maximizes $F[P_{\mathcal{F}}, Q] \leq \log Z$
- For mean field

$\max_{Q_j} F[P_F, \{Q_1, \dots, Q_n\}]$

subject to $Q_j(x_j) \geq 0$

$\sum_{x_j} Q_j(x_j) = 1$

Optimization for mean field

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

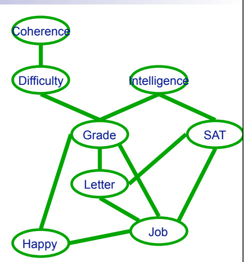
$$\forall i, \sum_{x_i} Q_i(x_i) = 1$$

- Constrained optimization, solved via Lagrangian multiplier
  - ∃ λ, such that optimization equivalent to:
  - Take derivative, set to zero
- **Theorem**: Q is a stationary point of mean field approximation iff for each *i*:

$$Q_i(x_i) = \frac{1}{Z_i} \exp\left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi \mid x_i] \right\}$$

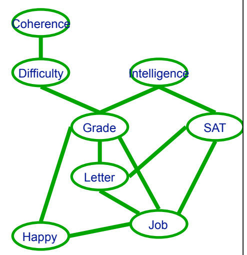
Understanding fixed point equation

$$Q_i(x_i) = \frac{1}{Z_i} \exp\left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi \mid x_i] \right\}$$



Simplifying fixed point equation

$$Q_i(x_i) = \frac{1}{Z_i} \exp\left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi \mid x_i] \right\}$$



Q_i only needs to consider factors that intersect X_i

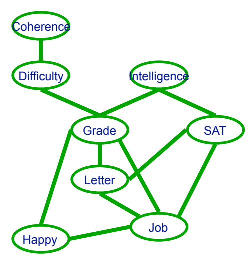
- **Theorem**: The fixed point:

$$Q_i(x_i) = \frac{1}{Z_i} \exp\left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi \mid x_i] \right\}$$

is equivalent to:

$$Q_i(x_i) = \frac{1}{Z_i} \exp\left\{ \sum_{\phi_j : X_i \in \text{Scope}[\phi_j]} E_Q[\ln \phi_j(\mathbf{U}_j, x_i)] \right\}$$

  - where the Scope[ϕ_j] = $\mathbf{U}_j \cup \{X_i\}$



There are many stationary points!

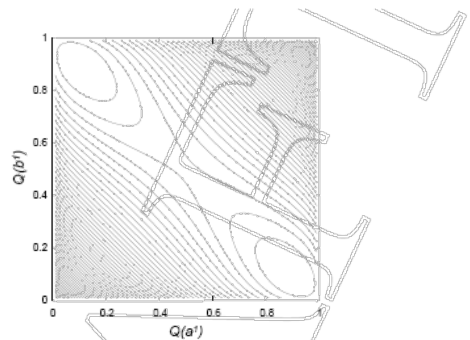
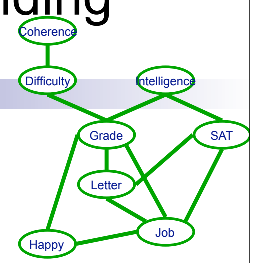


Figure 11.18 An example of a multi-modal mean field energy functional landscape. In this network, $P(a,b) = 0.25 - \epsilon$ if $a \neq b$ and ϵ if $a = b$. The axes correspond to the mean field marginal for A and B and the contours show equi-values of the energy functional.

Very simple approach for finding one stationary point

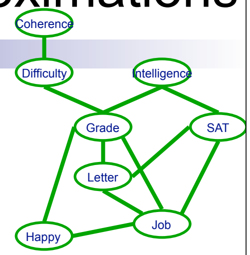


- Initialize Q (e.g., randomly or smartly)
- Set all vars to unprocessed
- Pick unprocessed var X_i
  - update Q_i:

$$Q_i(x_i) = \frac{1}{Z_i} \exp \left\{ \sum_{\phi_j : X_i \in \text{Scope}[\phi_j]} E_Q[\ln \phi_j(\mathbf{U}_j, x_i)] \right\}$$

  - set var i as processed
  - if Q_i changed
    - set neighbors of X_i to unprocessed
- Guaranteed to converge

More general structured approximations

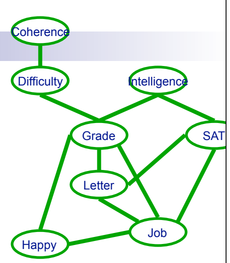


- Mean field very naïve approximation
- Consider more general form for Q
  - assumption: exact inference doable over Q
- **Theorem**: stationary point of energy functional:

$$\psi_j(\mathbf{c_j}) \propto \exp\left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi \mid \mathbf{c_j}] - \sum_{\psi \in \mathcal{Q} \setminus \{\psi_j\}} E_Q[\ln \psi \mid \mathbf{c_j}] \right\}$$

- Very similar update rule

Computing update rule for general case



$$\psi_j(\mathbf{c_j}) \propto \exp\left\{ \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi \mid \mathbf{c_j}] - \sum_{\psi \in \mathcal{Q} \setminus \{\psi_j\}} E_Q[\ln \psi \mid \mathbf{c_j}] \right\}$$

- Consider one ϕ:

Structured Variational update requires inference

$$\psi_j(\mathbf{c_j}) \propto \exp\left\{ \sum_{\phi\in\mathcal{F}} E_Q[\ln \phi \mid \mathbf{c_j}] - \sum_{\psi\in\mathcal{Q}\setminus\{\psi_j\}} E_Q[\ln \psi \mid \mathbf{c_j}] \right\}$$

- Compute marginals wrt Q of cliques in original graph and cliques in new graph, for all cliques
- What is a good way of computing all these marginals?
- Potential updates:
  - sequential: compute marginals, update ψ_j, recompute marginals
  - parallel: compute marginals, update all ψ's, recompute marginals

What you need to know about variational methods

- Structured Variational method:
  - select a form for approximate distribution
  - minimize reverse KL
- Equivalent to maximizing energy functional
  - searching for a tight lower bound on the partition function
- Many possible models for *Q*:
  - independent (mean field)
  - structured as a Markov net
  - cluster variational
- Several subtleties outlined in the book

Readings:
K&F: 10.2, 10.3

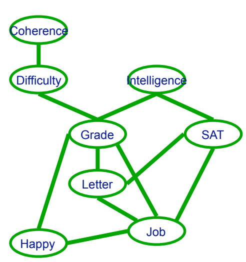
Loopy Belief Propagation

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

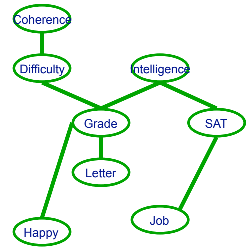
November 5th, 2008

Recall message passing over junction trees

- Exact inference:
  - generate a junction tree
  - message passing over neighbors
  - inference exponential in size of clique



Belief Propagation on Tree Pairwise Markov Nets

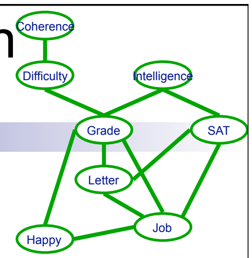


- Tree pairwise Markov net is a tree!!! ☺
  - no need to create a junction tree
- Message passing:
- More general equation:
  - *N*(i) – neighbors of *i* in pairwise MN

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

- **Theorem**: Converges to true probabilities:

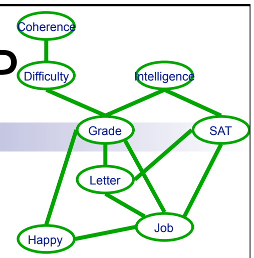
Loopy Belief Propagation on Pairwise Markov Nets



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

- What if we apply BP in a graph with loops?
  - send messages between pairs of nodes in graph, and hope for the best
- What happens?
  - 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 MPE rather than the actual probabilities

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:

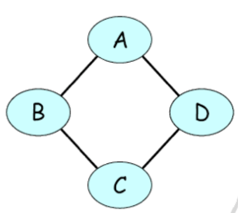
$$\delta_{i\to j}(X_j) = \frac{1}{Z_{i\to j}} \sum_{x_i} \phi_i(x_i)\phi_{ij}(x_i, X_j) \prod_{k\in\mathcal{N}(i)-j} \delta_{k\to i}(x_i)$$

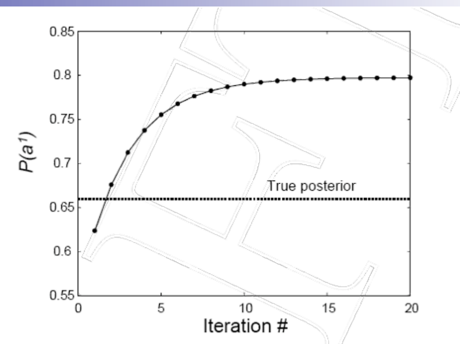
  - $Z_{i\to j}$ doesn't depend on X_j, so doesn't change the answer

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

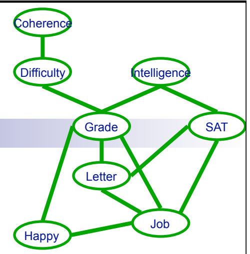
$$\widehat{P}(X_i) = \frac{1}{Z_i}\phi_i(X_i) \prod_{k\in\mathcal{N}(i)} \delta_{k\to i}(X_i)$$

An example of running loopy BP





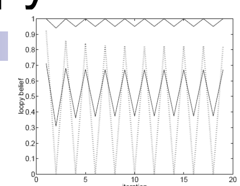
Convergence



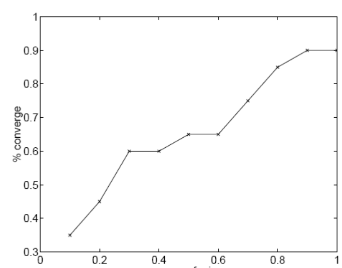
$$\widehat{P}(X_i) = \frac{1}{Z_i}\phi_i(X_i) \prod_{k \in \mathcal{N}(i)} \delta_{k \to i}(X_i)$$

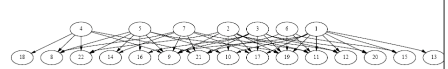
- If you tried to send all messages, and beliefs haven't changed (by much) $\rightarrow$ converged

(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

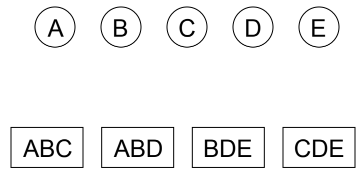




graphs from Murphy et al. '99

Loopy BP in Factor graphs

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

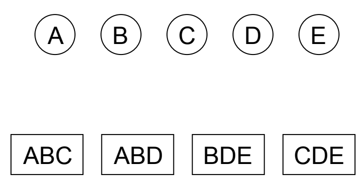


- Message example:
  - from node to factor:
  - from factor to node:

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}

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

- Belief:
  - Node:
  - Factor:

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