Mean Field and Variational Methods
finishing off

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

November 5th, 2008

Readings:
K&F: 10.1, 10.5

© Carlos Guestrin 2006-2008
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:**
  - \( D(p||q) \neq D(q||p) \)

- 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 \( P_F \)

\[
p(x) = \frac{1}{Z} e^{-\sum_{\phi \in \mathcal{F}} \phi(C_{\phi})}
\]

- **Theorem:**
  \[
  \ln Z = F[P_F, Q] + D(Q||P_F)
  \]

- where energy functional:
  \[
  F[P_F, Q] = \sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + H_Q(X)
  \]

- Want to maximize

- Want to minimize

- \( \ln Z \) has only \( Q \), want to compute
Understanding Reverse KL, Energy Function & The Partition Function

\[ \ln Z = F[P_F, Q] + D(Q \parallel P_F) \]

\[ F[P_F, Q] = \sum_{\phi \in F} E_Q[\ln \phi] + H_Q(x) \]

- Maximizing Energy Functional \iff\ Minimizing Reverse KL

\[ D(Q \parallel P) > 0 \]

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

\[ F(P_F, Q) + D(Q \parallel P_F) = \log Z \]

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

\[ \text{don't know how to compute } Z, \text{ so we will try to find a lower bound} \]

Structured Variational Approximate Inference

\[ \ln Z = F[P_F, Q] + D(Q \parallel P_F) \]

\[ F[P_F, Q] = \sum_{\phi \in F} E_Q[\ln \phi] + H_Q(x) \]

- Pick a family of distributions \( Q \) that allow for exact inference
  - e.g., fully factorized (mean field)
- Find \( Q \in Q \) that maximizes \( F[P_F, Q] \)

- For mean field

\[ \max_{Q_x} F[P_F, \{Q_1, \ldots, Q_n\}] \]

\[ \text{subject to } Q_j(x) \geq 0 \]

\[ \sum_{x_j} Q_j(x_j) = 1 \]
Optimization for mean field

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

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

- Constrained optimization, solved via Lagrangian multiplier
  - Find \( \lambda \), such that optimization equivalent to:
    $$\sum_{\phi \in \mathcal{F}} E_Q[\ln \phi] + \sum_j H_{Q_j}(x_j) + \lambda \left( \sum_j (Q_j(x_j) - 1) \right)$$
  - 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

$$Q_0(D = h) \approx P(D \mid J = \phi)$$

$$Q_W = \prod_i Q_i(x_i)$$

$$Q_0(D) = \sum_{\phi \in \phi} \frac{E_Q[\log \phi(D) \mid D = h]}{Z_0} + \frac{E_Q[\log \phi(D) \mid D = 1]}{Z_{\phi}} + \frac{\sum_{\phi \in \phi} E_Q[\log \phi(\phi(D) \mid D = h)]}{Z_{\phi}}$$

$$Z_0 = \frac{1}{Z_0} \exp \left\{ \sum_i Q_i(0) \log \phi(\phi(D = 1)) + E_Q[\log \phi(D \mid D = h)] \right\}$$

$$Z_{\phi} = \frac{1}{Z_0} \exp \left\{ \sum_i Q_i(1) \log \phi(\phi(D = 1)) + E_Q[\log \phi(D \mid D = h)] \right\}$$

$$Q_0(D) \approx P(D \mid J = \phi)$$
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:

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

where the \( \text{Scope}[\phi_j] = U_j \cup \{X_i\} \)

There are many stationary points!

Figure 11.19: An example of a multi-modal mean-field energy functional landscape. In this network, \( P(a, b) = \delta_{a, b} \) if \( a \neq b \) and \( 0 \) 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 $Q_i$:
    \[ Q_i(x_i) \leftarrow \frac{1}{Z_i} \exp \left\{ \sum_{\phi_j : X_i \in \text{Scope}([\phi_j])} E_Q[-\ln \phi_j(U_j, x_i)] \right\} \]
  - 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
  \[ Q(C) = \prod_j Q_j(c_j) \]
  - assumption: exact inference doable over $Q$
- **Theorem**: stationary point of energy functional:
  \[ \psi_j(c_j) \propto \exp \left\{ \sum_{\phi \in \mathcal{F}} E_Q[-\ln \phi(c_j)] - \sum_{\psi \in \mathcal{Q} \setminus \{\psi_j\}} E_Q[-\ln \psi(c_j)] \right\} \]
  - Very similar update rule
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 $\log Z$

- 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
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:
  \[
  \delta_{i \rightarrow j}(x_j) = \sum_{x_i} \phi_i(x_i) \phi_{ij}(x_i, X_j) \prod_{k \in N(i) \setminus j} \delta_{k \rightarrow i}(x_i)
  \]

- More general equation:
  - \( N(i) \) – neighbors of \( i \) in pairwise MN
  \[
  \delta_{i \rightarrow j}(X_j) = \sum_{x_i} \phi_i(x_i) \phi_{ij}(x_i, X_j) \prod_{k \in N(i) \setminus j} \delta_{k \rightarrow i}(x_i)
  \]

- **Theorem**: Converges to true probabilities:
  \[
  b_j(x_j) = p(X_j) \alpha \phi_j(X_j) \prod_{k \in N(j)} \delta_{k \rightarrow j}(x_j)
  \]
Loopy Belief Propagation on Pairwise Markov Nets

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:

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

- \( Z_{i \rightarrow j} \) doesn’t depend on \( X_j \), so doesn’t change the answer

Computing node “beliefs” (estimates of probs.):

\[ b_j(x_j) = \hat{p}(X_j) = \frac{1}{Z_i} \phi_i(X_i) \prod_{k \in N(i)} \delta_{k \rightarrow i}(X_i) \]

Sometimes important to compute in log space

\[ \log \hat{p}(x_j) = \log \hat{p}(x_j) \]
An example of running loopy BP

Convergence

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

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

- normalized messages haven't changed by much
(Non-)Convergence of Loopy BP

- Loopy BP can oscillate!!!
  - oscillations can be 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:
    \[ \phi_{i\rightarrow j}^{(t+1)} = (1-d) \phi_{i\rightarrow j}^{(t)} + d \phi_{i\rightarrow j}^{(t+1)} \]
  - often better convergence
    - but, when damping is required to get convergence, result often bad

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:
    \[ \phi_{\text{factor}}(x) \propto \phi_{B}(x) \phi_{A\rightarrow B}(x) \phi_{D\rightarrow B}(x) \phi_{E\rightarrow B}(x) \]
  - from factor to node:
    \[ \phi_{\text{factor}}(x) \propto \sum_{x} \phi_{A}(x) \phi_{B}(x) \phi_{B}(x) \phi_{D\rightarrow x}(x) \phi_{E\rightarrow x}(x) \]
Loopy BP in Factor graphs

From node $i$ to factor $j$:
- $F(i)$ factors whose scope includes $X_i$
- $\delta_{i\rightarrow j}(X_i) \propto \prod_{k \in F(i) \setminus j} \delta_{k \rightarrow i}(X_i)$

From factor $j$ to node $i$:
- $\text{Scope}[\phi_j] = \{X_i\}$
- $\delta_{j \rightarrow i}(X_i) \propto \sum_{y} \phi_j(X_i, y) \prod_{X_k \in \text{Scope}[\phi_j] \setminus X_i} \delta_{k \rightarrow j}(x_k)$

Belief:
- Node: $P(X_i) \propto b_i(X_i) \propto \prod_{\phi_j \in \text{Scope}[\phi_j]} \phi_j(X_i) \prod_{X_k \in \text{Scope}[\phi_j] \setminus X_i} \delta_{k \rightarrow j}(x_k)$
- Factor: $\phi_j(Y) \propto b_j(Y) \propto \phi_j(Y) \prod_{X_i \in \text{Scope}[\phi_j]} \delta_{j \rightarrow i}(x_i)$

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)
  - One good way: always send message that changed the most since previous message
- If converges, often to incorrect, but useful results
- Generalizes from pairwise Markov networks by using factor graphs