Review: Metropolis-Hastings Algorithm

- Metropolis-Hastings algorithms
  - You decide the transition probability $T^Q$ – based on the proposal distribution $Q$
  - Acceptance probability “corrects” for the discrepancy between $Q$ and $P$
    \[
    A(x \rightarrow x') = \min \left[ 1, \frac{P(x') T^Q(x' \rightarrow x)}{P(x) T^Q(x \rightarrow x')} \right]
    \]
  - Advantage: more “global” move from one state to another (compared to Gibbs sampling)
  - The convergence of the M-H algorithm depends crucially on the proposal distribution $Q$
    - We need a proposal strategy that leads to a rapidly mixing Markov chains (i.e. one that converges quickly to the stationary distribution)
    - Let’s see a toy example from Dellaert et al.*

Revisit: Toy Model for Data Association

- Blue dots: variables, $X_i$ (i=1,2,3,4)
- Red dots: observations (values that we assign to variables)

\[
P(X) \propto \exp\left( -\sum_{i=1}^{4} \frac{||x_i - x_i(value)||^2}{\sigma} \right) \quad \text{if every } x_i \text{ has a different value}
\]
\[
P(X) = 0 \quad \text{otherwise}
\]

- Two modes

- We want to estimate $E_p(f)$ – Let’s use M-H algorithm with three proposal distributions

Proposal distributions for M-H

- Proposal distribution 1 (flip proposal)
  - Simplest way of taking larger steps in moving over the state spaces (compared to Gibbs sampling)
  - Randomly pick two variables, flip their assignments

  - Attractive from a computational point of view, it has the severe disadvantage of leading to slowly mixing chains in many instances...

Proposal distributions for M-H

 Proposal distribution 2 (augmenting path)
- Suggest a move that is more likely to be accepted: recursively resolving the conflict
  - Improving the convergence properties of the chain:
    - 1. randomly pick one variable
    - 2. sample it pretending that all observations are available
    - 3. pick the variable X whose assignment was taken (conflict), goto step 2
    - 4. loop until step 2 creates no conflict

Proposal distributions for M-H

 Proposal distribution 3 ("smart" augmenting path)
- More aggressive way of moving to different states
  - Same as the previous one except for the highlighted
  - 1. randomly pick one variable
  - 2. sample it pretending that all observations are available (excluding the current one)
  - 3. pick the variable whose assignment was taken (conflict), goto step 2
  - 4. loop until step 2 creates no conflict

* F. Dellaert, SM. Seitz, CE. Thorpe and S. Thrun.
Let’s “See” How They Work

- Which proposal strategy is the most “aggressive” in moving over the states??
  - Converges the fastest to the stationary distribution

- Run the following Matlab scripts:
  
  ```matlab
  VisualMCMC2(10000, 0.7, 0.05);
  % live animation of sampling
  % parameters: num of samples, sigma, pause time after each sample
  Plot2;
  % the first few lines of Plot2.m contain the parameters you may want to play around with

  How to evaluate the convergence performance?
  - Compare between multiple Markov chains, in terms of $E_p(f)$, $P(Y=y)$, etc

Plots generated by “Plot2”

- The convergence of the M-H algorithm depends crucially on the proposal distribution $Q$
  - We need a proposal strategy that leads to a rapidly mixing Markov chains
Review: Particle-based Inference

- **General framework:**
  - Estimate $E_p(f)$ from particles $x[1],...,x[M]$ from $P$ (target distribution) or $Q$ (proposal distribution)

- **Full particle methods**
  - **Sampling methods**
    - Forward sampling, Likelihood weighting
    - (Un-normalized/normalized) Importance sampling
    - Markov chain Monte Carlo
      - Gibbs sampling
      - Metropolis-Hastings algorithm
  - Deterministic particle generation
    - Upper/lower bounds of $E_p(f)$

- **Distributional (Collapsed) particles**

---

Let’s now talk about a different kind of approximate inference algorithm that views inference as optimization...

**GLOBAL APPROXIMATE INFERENCE**
General Approximate Inference

- Again, in many real-life applications using large and dense networks, exact inference is infeasible...

- **Strategy**
  - Define a class of simpler distributions $Q$
  - Search for a particular instance in $Q$ that is “close” to $P$
    - All methods we will discuss optimize the same target function for measuring the similarity between $Q$ and $P$
  - Answer queries using inference in $Q$ rather than $P$

- Before considering approximate inference methods, let’s revisit exact inference based on message passing algorithms

Cluster Graph

- A **cluster graph** $K$ for factors $F$ is an undirected graph
  - Nodes are associated with a subset of variables $C_i \subseteq U$
  - The graph is **family preserving**: each factor $\phi \in F$ is associated with one node $C_i$ such that $\text{Scope}[\phi] \subseteq C_i$
  - Each edge $C_i - C_j$ is associated with a **sepset** $S_{ij} = C_i \cap C_j$

- **Clique tree**: a cluster graph over factors $F$ that forms a tree and satisfies the **running intersection property**
Clique Tree Inference

Verify:
- Tree and family preserving
- Running intersection property

Message Passing: Belief Propagation

- Initialize the clique tree
  - For each clique $C_i$ set $\pi_i \leftarrow \prod_{\phi \in \phi_{i-1}} \phi$
  - For each edge $C_i \rightarrow C_j$ set $\mu_{i,j} \leftarrow 1$

- While unset cliques exist (clique tree is calibrated)
  - Select $C_i \rightarrow C_j$
  - Send message from $C_i$ to $C_j$
    - Marginalize the clique over the sepset $\sigma_{i \rightarrow j} \leftarrow \sum_{C_i - S_{i,j}} \pi_i$
    - Update the belief at $C_j$ $\pi_j \leftarrow \pi_j \frac{\sigma_{i \rightarrow j}}{\mu_{i,j}}$
    - Update the sepset at $C_i \rightarrow C_j$ $\mu_{i,j} \leftarrow \sigma_{i \rightarrow j}$
Clique Tree Invariant

- Belief propagation can be viewed as reparameterizing the joint distribution
  - Upon calibration we showed
    \[ p(U) = \frac{\prod_{c \in C} \pi[C]}{\prod_{i \in C, j \in \partial_i} \mu_{i,j}(S_{i,j})} \]
  - Initially this invariant holds since
    \[ \frac{\prod_{c \in C} \pi[C]}{\prod_{i \in C, j \in \partial_i} \mu_{i,j}(S_{i,j})} = \frac{1}{P(U)} \]
  - At each update step invariant is also maintained
    - Message only changes \( \pi_i \) and \( \mu_{i,j} \) so most terms remain unchanged
    - We need to show \( \frac{\pi_i'}{\mu_{i,j}'} = \frac{\pi_i}{\mu_{i,j}} \)
    - But this is exactly the message passing step \( \pi_i' = \frac{\mu_{i,j}' \pi_i}{\mu_{i,j}} \)

\[ \rightarrow \text{Belief propagation re-parameterizes } P \text{ at each step} \]

Global Approximate Inference

- Inference as optimization
- Generalized Belief Propagation (GBP)
  - Define algorithm
  - Constructing cluster graphs
  - Analyze approximation guarantees
  - GBP as optimization
- Propagation with approximate messages (EP)
  - Factorized messages
  - Approximate message propagation
- Structured variational approximations
The Energy Functional

- Suppose we want to approximate $P$ with $Q$
  - Represent $P$ by factors $F$
    $$P_f(U) = \frac{1}{Z} \prod_{\phi \in F} \phi(U)$$
  - Distance metric? - Many ways, but let’s use relative entropy (aka KL-divergence)
    $$D(Q \parallel P_f) = \mathbb{E}_Q[\ln \frac{Q}{P_f}]$$
  - Define the energy functional $F[P_f', Q] = \sum_{\phi \in F} E_Q[\ln \phi] + H_Q(U)$

- Then, we can show that $D(Q \parallel P_f) = \ln Z - F[P_f', Q]$
  - Proof in K&F (page 385)
    - Minimizing $D(Q \parallel P_f)$ is equivalent to maximizing $F[P_f', Q]$
    - $\ln Z \geq F[P_f', Q]$ (since $D(Q \parallel P_f) \geq 0$)

Inference as Optimization

- Basic idea: We can show that inference can be viewed as maximizing the energy functional $F[P_f', Q]$
  - Define a distribution $Q$ over clique potentials
  - Transform $F[P_f', Q]$ to an equivalent factored form $F'[P_f', Q]$

- Show that if $Q$ maximizes $F'[P_f', Q]$ subject to constraints in which $Q$ represents calibrated potentials, then there exists factors (messages) that satisfy the inference message passing equations
  - Equivalent to belief propagation!
Defining Q

- Recall that throughout BP, \( P(U) = \frac{\prod_{c \in \mathcal{T}} \pi_c [C_i]}{\prod_{(c_i \neq C_j) \in \mathcal{T}} \mu_{i,j}(S_{i,j})} \)

- Define Q as re-parameterization of P such that

\[
Q = \{\pi_i\} \cup \{\mu_{i,j} : (C_i - C_j) \in \text{clique tree } T\}
\]

\[
Q_i(U) = \frac{\prod_{c \in \mathcal{T}} \pi_c [C_i]}{\prod_{(c_i \neq C_j) \in \mathcal{T}} \mu_{i,j}(S_{i,j})}
\]

- If T is calibrated, \( D(Q || P_F) = 0 \) and so \( F[P_F', Q] \) is maximized.

Factored Energy Functional

- Recall that the energy functional is defined as

\[
F[P_F', Q] = \sum_{\phi \in F} E_{\phi} [\ln \phi] + H_{\phi}(U)
\]

- Define the factored energy functional as

\[
F'[P_F', Q] = \sum_{c \in \mathcal{T}} E_{\pi_i} [\ln \pi_i^0] + \sum_{(c_i \neq C_j) \in \mathcal{T}} H_{\pi_i}(C_i) - \sum_{(c_i \neq C_j) \in \mathcal{T}} H_{\mu_{i,j}}(S_{i,j})
\]

- Theorem: if \( Q \) is a set of calibrated potentials for \( T \), then \( F[P_F', Q] = F'[P_F', Q] \) (K&F page 387)
Inference as Optimization

- **Optimization task**
  - Find $Q$ that maximizes $F[P, Q]$ subject to
    \[
    \mu_{i,j} = \sum_{C_i \sim S_{i,j}} \pi_i \quad \forall (C_i, C_j) \in \text{clique tree } T
    \]
    \[
    \sum_{C_i} \pi_i = 1 \quad \forall C_i \in T
    \]

- **The solution of the above optimization problem satisfies (if exists)**
  \[
  \delta_{i \to j} \propto \sum_{C_i \sim S_{i,j}} \pi_i^0 \left( \prod_{k \in N_i \setminus \{j\}} \delta_{k \to i} \right)
  \]
  \[
  \pi_i \propto \pi_i^0 \left( \prod_{j \in N_i} \delta_{j \to i} \right)
  \]
  \[
  \mu_{i,j} = \delta_{i \to j} \times \delta_{j \to i}
  \]

- **Suggests iterative procedure**
- **Identical to belief propagation!**

---

Global Approximate Inference

- **Inference as optimization**
- **Generalized Belief Propagation**
  - Define algorithm
  - Constructing cluster graphs
  - Analyze approximation guarantees
  - GBP as optimization

- **Propagation with approximate messages**
  - Factorized messages
  - Approximate message propagation
- **Structured variational approximations**
Revisit: Clique Tree Inference

Verify:
- Tree and family preserving
- Running intersection property

Modify

Generalized Belief Propagation

Strategy:
Perform belief propagation in a cluster graph with loops
Generalized Belief Propagation

Strategy:

Perform belief propagation in a cluster graph with loops

- Inference may be incorrect: double counting evidence

Unlike in BP on trees:
- Convergence is not guaranteed
- Potentials in calibrated tree are not guaranteed to be marginals in $P$
Generalized Cluster Graph

- A cluster graph $K$ for factors $F$ is an undirected graph
  - Nodes are associated with a subset of variables $C_i \subseteq U$
  - The graph is family preserving: each factor $\phi \in F$ is associated with one node $C_i$ such that $\text{Scope}[\phi] \subseteq C_i$
  - Each edge $C_i - C_j$ is associated with a sepset $S_{i,j} = C_i \cap C_j$

- A generalized cluster graph $K$ for factors $F$ is an undirected graph
  - Nodes are associated with a subset of variables $C_i \subseteq U$
  - The graph is family preserving: each factor $\phi \in F$ is associated with one node $C_i$ such that $\text{Scope}[\phi] \subseteq C_i$
  - Each edge $C_i - C_j$ is associated with a subset $S_{i,j} \subseteq C_i \cap C_j$

A generalized cluster graph $K$ for factors $F$ is an undirected graph

- Nodes are associated with a subset of variables $C_i \subseteq U$
- The graph is family preserving: each factor $\phi \in F$ is associated with one node $C_i$ such that $\text{Scope}[\phi] \subseteq C_i$
- Each edge $C_i - C_j$ is associated with a subset $S_{i,j} \subseteq C_i \cap C_j$

A generalized cluster graph $K$ for factors $F$ is an undirected graph

- Nodes are associated with a subset of variables $C_i \subseteq U$
- The graph is family preserving: each factor $\phi \in F$ is associated with one node $C_i$ such that $\text{Scope}[\phi] \subseteq C_i$
- Each edge $C_i - C_j$ is associated with a subset $S_{i,j} \subseteq C_i \cap C_j$

A generalized cluster graph $K$ for factors $F$ is an undirected graph

- Nodes are associated with a subset of variables $C_i \subseteq U$
- The graph is family preserving: each factor $\phi \in F$ is associated with one node $C_i$ such that $\text{Scope}[\phi] \subseteq C_i$
- Each edge $C_i - C_j$ is associated with a subset $S_{i,j} \subseteq C_i \cap C_j$

A generalized cluster graph $K$ for factors $F$ is an undirected graph

- Nodes are associated with a subset of variables $C_i \subseteq U$
- The graph is family preserving: each factor $\phi \in F$ is associated with one node $C_i$ such that $\text{Scope}[\phi] \subseteq C_i$
- Each edge $C_i - C_j$ is associated with a subset $S_{i,j} \subseteq C_i \cap C_j$

Generalized Cluster Graph

- A generalized cluster graph obeys the running intersection property if for each $X \in C_i$ and $X \in C_j$, there is exactly one path between $C_i$ and $C_j$ for which $X \in S$ for each subset $S$ along the path

- All edges associated with $X$ form a tree that spans all the clusters that contain $X$
  - Note: some of these clusters may be connected with more than one path
Calibrated Cluster Graph

- A generalized cluster graph is calibrated if for each edge $C_i - C_j$ we have:

\[
\sum_{C_i \cap S_{i,j}} \pi_i[C_i] = \sum_{C_j \cap S_{i,j}} \pi_j[C_j]
\]

- Weaker than in clique trees, since $S_{i,j}$ is a subset of the intersection between $C_i$ and $C_j$

- If a cluster graph satisfies the running intersection property, then the marginal on any variable $X$ is the same in every cluster that contains $X$

GBP is Efficient

Markov grid network

Note: clique tree in a $n \times n$ grid is exponential in $n$

Round of GBP is $O(n)$

Cluster graph
Global Approximate Inference

- Inference as optimization
- Generalized Belief Propagation
  - Define algorithm
  - Constructing cluster graphs
  - Analyze approximation guarantees
  - GBP as optimization
- Propagation with approximate messages
  - Factorized messages
  - Approximate message propagation
- Structured variational approximations

Constructing Cluster Graphs

- When constructing clique trees, all constructions give the same result, but differ in computational complexity

- In GBP, different cluster graphs can vary in both computational complexity and approximation quality (accuracy)
Transforming Pairwise MNs

- A pairwise Markov network over a graph $H$ has:
  - A set of node potentials $\{\pi[X_i] : i=1,...,n\}$
  - A set of edge potentials $\{\pi[X_i,X_j] : X_i,X_j \in H\}$
- Example:

![Graph representation of a pairwise Markov network]

Transforming Bayesian Networks

- Example:

![Graph representation of a Bayesian network]

- “Large” cluster per each CPD
- Single nodes for each variable
- Connect node and large cluster if node in CPD
- Graph obeys running intersection property
Global Approximate Inference

- Inference as optimization
- Generalized Belief Propagation
  - Define algorithm
  - Constructing cluster graphs
  - Analyze approximation guarantees
- Propagation with approximate messages
  - Factorized messages
  - Approximate message propagation
- Structured variational approximations

Generalized Belief Propagation

- GBP maintains distribution invariance

\[
P_f(U) = \frac{\prod_{C \in \mathbb{K}} \pi_i[C]}{\prod_{(C_1 \otimes C_2) \in \mathbb{K}} \mu_{i,j}(S_{i,j})}
\]

- (since message passing maintains invariance)
Generalized Belief Propagation

- If GBP converges (K is calibrated)
  - Each subtree $T$ is calibrated with edge potentials corresponding to marginals of $P_T(U)$
    \[
    P_T(U) = \frac{\prod_{C \in T} \pi_i(C)}{\prod_{(C_j + C_k) \in T} \mu_{i,j}(S_{i,j})}
    \]
  - (since $P_T(U)$ is a calibrated tree)

Generalized Belief Propagation

- $\rightarrow$ Calibrated graph potentials are not $P_F(U)$ marginals

\[
\begin{align*}
\pi_1[A,B] \pi_2[B,C] \pi_3[C,D] \\
\mu_{1,2}[B] \mu_{2,3}[C]
\end{align*}
\]

\[
\begin{align*}
\pi_4[A,D] \neq \mu_{3,4}[D] \mu_{4,1}[A]
\end{align*}
\]

\[
P_f \neq P_f \rightarrow \pi_1[A,B] \neq P_f(A,B)
\]
**Inference as Optimization**

- **Optimization task**
  
  Find $Q$ that maximizes $F[P_F, Q]$ subject to
  
  $\mu_{i,j} = \sum_{C_i \in S_{i,j}} \pi_i \forall (C_i - C_j) \in \text{clique tree } T$
  
  $\sum_{C_i} \pi_i = 1 \forall C_i \in T$
  
- The solution of the above optimization problem satisfies
  
  $\delta_{i \to j} \propto \sum_{C_i \cap S_{i,j}} \pi_i \left( \prod_{j \in N_i \cap C_i \cap C_j} \delta_{j \to i} \right)$
  
  $\pi_i \propto \pi_i \left( \prod_{j \in N_i \cap C_i \cap C_j} \delta_{j \to i} \right)$
  
  $\mu_{i,j} = \delta_{i \to j} \times \delta_{j \to i}$
  
  Suggests iterative procedure
  
  Identical to belief propagation!

**GBP as Optimization**

- **Optimization task**
  
  Find $Q$ that maximizes $F[P_F, Q]$ subject to
  
  $\mu_{i,j} = \sum_{C_i \in S_{i,j}} \pi_i \forall (C_i - C_j) \in K$
  
  $\sum_{C_i} \pi_i = 1 \forall C_i \in K$
  
- The solution of the above optimization problem satisfies
  
  $\delta_{i \to j} \propto \sum_{C_i \cap S_{i,j}} \pi_i \left( \prod_{j \in N_i \cap C_i \cap C_j} \delta_{j \to i} \right)$
  
  $\pi_i \propto \pi_i \left( \prod_{j \in N_i \cap C_i \cap C_j} \delta_{j \to i} \right)$
  
  $\mu_{i,j} = \delta_{i \to j} \times \delta_{j \to i}$
  
  Note: $S_{i,j}$ is only a subset of intersection between $C_i$ and $C_j$
  
  Iterative optimization procedure is GBP
GBP as Optimization

- Clique trees
  - $F[P_F,Q] = F'[P_F,Q]$
  - Iterative procedure (BP) guaranteed to converge
  - Convergence point represents marginal distributions of $P_F$

- Cluster graphs
  - $F[P_F,Q] = F'[P_F,Q]$ does not hold!
  - Iterative procedure (GBP) not guaranteed to converge
  - Convergence point does not represent marginal distributions of $P_F$

GBP in Practice

- Dealing with non-convergence
  - Often small portions of the network do not converge
    - $\rightarrow$ stop inference and use current beliefs
  - Use intelligent message passing scheduling
    - Tree reparameterization (TRP) selects entire trees, and calibrates them while keeping all other beliefs fixed
    - Focus attention on uncalibrated regions of the graph
Global Approximate Inference

- Inference as optimization
- Generalized Belief Propagation
  - Define algorithm
  - Constructing cluster graphs
  - Analyze approximation guarantees
- Propagation with approximate messages
  - Factorized messages
  - Approximate message propagation
  - Structured variational approximations

Propagation w. Approximate Msgs

- General idea
  - Perform BP (or GBP) as before, but propagate messages that are only approximate
  - Modular approach
    - General inference scheme remains the same
    - Can plug in many different approximate message computations
Factorized Messages

- Keep internal structure of the clique tree cliques
- Calibration involves sending messages that are joint over three variables
- Idea: simplify messages using factored representation
  - Example: \[ \delta_{1 \rightarrow 2}[X_{11}, X_{21}, X_{31}] = \delta_{1 \rightarrow 2}[X_{11}] \delta_{1 \rightarrow 2}[X_{21}] \delta_{1 \rightarrow 2}[X_{31}] \]

Markov network

<table>
<thead>
<tr>
<th>X_{11}</th>
<th>X_{12}</th>
<th>X_{13}</th>
</tr>
</thead>
<tbody>
<tr>
<td>X_{21}</td>
<td>X_{22}</td>
<td>X_{23}</td>
</tr>
<tr>
<td>X_{31}</td>
<td>X_{32}</td>
<td>X_{33}</td>
</tr>
</tbody>
</table>

Clique tree

<table>
<thead>
<tr>
<th>X_{11}</th>
<th>X_{12}</th>
<th>X_{13}</th>
</tr>
</thead>
<tbody>
<tr>
<td>X_{21}</td>
<td>X_{22}</td>
<td>X_{23}</td>
</tr>
<tr>
<td>X_{31}</td>
<td>X_{32}</td>
<td>X_{33}</td>
</tr>
</tbody>
</table>

Acknowledgement

- These lecture notes were generated based on the slides from Prof Eran Segal.