Inference in MRFs

Instructor: Pranjal Awasthi
Announcement

- HW4 out today, due on April 11.
Last Class

• $y = f^*(\vec{X})$
  
  – Use train set $(\vec{X}_1, y_1), (\vec{X}_2, y_2), \ldots, (\vec{X}_m, y_m)$ to build a predictor

• What if want to predict multiple $y'_i$s?
  
  – $y_1 = f_1^*(\vec{X}), y_2 = f_2^*(\vec{X}), \ldots, y_k = f_k^*(\vec{X})$
  
  – $y'_i$s are correlated
**Graphical Models (MRFs)**

• **Markov Constraints**
  • For each $X_i$, let $N(X_i) = \{X_j: X_i \sim X_j \text{ in } G\}$
  • Then $X_i$ is independent of the rest of the variables given $N(X_i)$
  • $P(X_i \mid X_1, X_2, \ldots, X_{i-1}, X_{i+1}, X_n) = P(X_i \mid N(X_i))$
Markov Random Fields (MRFs)

Theorem [Hammersley, Clifford’72]:
If \( P + G \) is an MRF then \( P(\vec{X}) \propto e^{\sum_c \theta_c \phi_c(\vec{X})} \), where the sum is over cliques of \( G \), \( \theta_c \in \mathbb{R} \) and \( \phi_c(\vec{X}) \) is a function defined over variables in \( C \).

\[
P(\vec{X}) = e^{\sum_c \theta_c \phi_c(\vec{X})} - A(\theta)
= \prod_c \psi_c(\vec{X})
\]
Graphical Models

• Learning problems
  – Given $\tilde{X}_1, \tilde{X}_2, \ldots \tilde{X}_m$ learn the graph structure $G$
  – Given $\tilde{X}_1, \tilde{X}_2, \ldots \tilde{X}_m$ and graph $G$, learn the parameters

• Inference
  – Given $G + P(\tilde{X})$
    • Compute $P(X_i)$
    • Compute $P(X_i \mid X_j)$
    • Generate sample from $P(\tilde{X})$
    • Compute $\text{argmax}_{\tilde{X}} P(\tilde{X})$

\[ (\Theta, \phi) \quad \text{(potential)} \]
Inference in Graphical Models

• Techniques
  – Junction Tree Algorithm
  – Markov Chain Monte Carlo Methods
  – Variational Methods
Junction Tree Algorithm

• Given $G + P(\tilde{X})$
  • Compute $P(X_i)$

  $p_{marginals}(X_i) = \sum_{x \neq i} p(x_1, x_2, ..., x_n)$

• Junction Tree Algorithm
  – Provides exact solution
  – Not always computationally efficient
  – Polynomial time for trees (Belief Propagation)
Belief Propagation Algorithm

- Special Case -- $G$ is a tree
- $X_i \in \{-1, 1\}$,
- Goal: Compute $P(X_i = \pm 1)$ for all $i$.

\[
P(\vec{X}) \propto e^{\sum_{i=1}^{n} \theta_i X_i + \sum_{i\sim j} \theta_{ij} X_i X_j}
\]

- Let’s compute $P(X_5 = a)$
Belief Propagation

\[
P(\hat{X}) \propto e^{\sum_{i=1}^{n} \theta_i X_i + \sum_{i\sim j} \theta_{ij} X_i X_j}
\]
Belief Propagation

\[ P(\vec{X}) \propto e^{\sum_{i=1}^{n} \theta_i X_i + \sum_{i \sim j} \theta_{ij} X_i X_j} \]

\[ P(X_5 = a) \propto e^{\theta_{35} \cdot m_{35} (a) m_{35} (a)} \]

\[ m_{35} (a) = \frac{e^{\theta_3 X_3 \theta_{35} X_5}}{\sum_{X_3=\pm 1} m_{83}(X_3)m_{93}(X_3)} \]

\[ m_{35} (a) = \frac{e^{\theta_3 X_3 \theta_{35} X_5}}{m_{83}(X_3)m_{93}(X_3)} \]
Belief Propagation

\[ P(\vec{X}) \propto e^{\sum_{i=1}^{n} \theta_i X_i + \sum_{i \sim j} \theta_{ij} X_i X_j} \]

Node \( i \) has degree \( d_i \)

time to compute \( m_{ij}() = O(d_i) \)

total time \( = \sum_{i=1}^{n} O(d_i) = O(n) \)

\[ P(X_j = a) \propto e^{\theta_j a} \prod_{i \in N(X_j)} m_{i,j}(a) \]

\[ m_{i,j}(a) = \sum_{X_i = \pm 1} e^{\theta_i X_i e^{\theta_i,j X_i \cdot a}} \prod_{j' \in N(X_i) \setminus \{X_j\}} m_{j',i}(X_i) \]
Belief Propagation

\[
P(\vec{X}) \propto e^{\sum_{i=1}^{n} \theta_i X_i + \sum_{i \sim j} \theta_{ij} X_i X_j}
\]

\[
P(X_i = a) \propto e^{\theta_j a} \prod_{i \in N(X_j)} m_{i,j}(a)
\]

\[
m_{i,j}(a) = \sum_{X_i = \pm 1} e^{\theta_i X_i} e^{\theta_{ij} X_i \cdot a} \prod_{j' \in N(X_i) \setminus \{X_j\}} m_{j',i}(X_i)
\]
Belief Propagation

• Given $G + P(\tilde{X})$
  • Compute $P(X_i)$

• Polynomial time for trees and provides exact solution.
• Polynomial time for graphs with correlation decay.
  
  (no long range correlations in the graph)
  (A locally behaves like a tree)
Correlation Decay

\[ |\mathbb{P}(X_i = +1 \mid X_j = 1) - \mathbb{P}(X_i = 1 \mid X_j = 1)| \leq \exp(-d(X_i; S)) \]

BP is optimal for graphs with correlation decay.
Junction Tree Algorithm

- Exact Algorithm for arbitrary graphs
- $X_i \in \{-1, 1\}$
Junction Tree Algorithm

- Exact Algorithm for arbitrary graphs
- \( X_i \in \{-1, 1\} \)

- Step 1: Build junction tree
- Step 2: Run BP on junction tree
Junction Tree Algorithm

- Exact Algorithm for arbitrary graphs
- $X_i \in \{-1, 1\}$

- Step 1: Build junction tree
- Step 2: Run BP on junction tree
Junction Tree Algorithm

- Exact Algorithm for arbitrary graphs
- $X_i \in \{-1,1\}$

Step 1: Build junction tree
Step 2: Run BP on junction tree

Thm 1: A consistent tree is possible if $G$ is triangulated.
Junction Tree Algorithm

- Exact Algorithm for arbitrary graphs
- $X_i \in \{-1, 1\}$

- Step 1: Triangulate the graph
- Step 2: Compute Clique graph
- Step 3: Build junction tree
- Step 4: Run BP on junction tree
Junction Tree Algorithm

• Exact Algorithm for arbitrary graphs
• $X_i \in \{-1, 1\}$

• Step 1: Triangulate the graph
Junction Tree Algorithm

• Exact Algorithm for arbitrary graphs
• \( X_i \in \{-1, 1\} \)

• Step 2: Compute Clique graph
Junction Tree Algorithm

- Exact Algorithm for arbitrary graphs
- $X_i \in \{-1, 1\}$

- Step 3: Build junction tree
Junction Tree Algorithm

• Exact Algorithm for arbitrary graphs
• $X_i \in \{-1, 1\}$

• Step 3: Run BP on junction tree
Junction Tree Algorithm

• Given $G + P(\hat{X})$
  • Compute $P(X_i)$

• Provides exact solution
• Runs in Polynomial time for trees
• Runs in polynomial time on graphs with low treewidth