CS 6347

Lecture 5

Exact Inference in MRFs
Inference

\[ p(x_A, x_B, x_C, x_D) = \frac{1}{Z} \psi_{AB}(x_A, x_B) \psi_{BC}(x_B, x_C) \psi_{CD}(x_C, x_D) \]

\[ Z = \sum_{x_A', x_B', x_C', x_D'} \psi_{AB}(x_A', x_B') \psi_{BC}(x_B', x_C') \psi_{CD}(x_C', x_D') \]
Inference

\[ Z = \sum_{x'_A, x'_B, x'_C, x'_D} \psi_{AB}(x'_A, x'_B) \psi_{BC}(x'_B, x'_C) \psi_{CD}(x'_C, x'_D) \]

\[ = \sum_{x'_A} \sum_{x'_B} \sum_{x'_C} \sum_{x'_D} \psi_{AB}(x'_A, x'_B) \psi_{BC}(x'_B, x'_C) \psi_{CD}(x'_C, x'_D) \]

\[ = \sum_{x'_A} \sum_{x'_B} \psi_{AB}(x'_A, x'_B) \sum_{x'_C} \psi_{BC}(x'_B, x'_C) \sum_{x'_D} \psi_{CD}(x'_C, x'_D) \]
Inference

\[ Z = \sum_{x'_A, x'_B, x'_C, x'_D} \psi_{AB}(x'_A, x'_B)\psi_{BC}(x'_B, x'_C)\psi_{CD}(x'_C, x'_D) \]

\[ = \sum_{x'_A} \sum_{x'_B} \sum_{x'_C} \sum_{x'_D} \psi_{AB}(x'_A, x'_B)\psi_{BC}(x'_B, x'_C)\psi_{CD}(x'_C, x'_D) \]

\[ = \sum_{x'_A} \sum_{x'_B} \sum_{x'_C} \sum_{x'_D} \psi_{AB}(x'_A, x'_B)\psi_{BC}(x'_B, x'_C)\psi_{CD}(x'_C, x'_D) \]

\[ \phi_C(x'_C) \]
\[ Z = \sum_{x_A', x_B', x_C', x_D'} \psi_{AB}(x_A', x_B') \psi_{BC}(x_B', x_C') \psi_{CD}(x_C', x_D') \]

\[ = \sum_{x_A'} \sum_{x_B'} \sum_{x_C'} \sum_{x_D'} \psi_{AB}(x_A', x_B') \psi_{BC}(x_B', x_C') \psi_{CD}(x_C', x_D') \]

\[ = \sum_{x_A'} \sum_{x_B'} \psi_{AB}(x_A', x_B') \sum_{x_C'} \psi_{BC}(x_B', x_C') \phi_C(x_C') \]
Inference

\[ Z = \sum_{x'_A, x'_B, x'_C, x'_D} \psi_{AB}(x'_A, x'_B) \psi_{BC}(x'_B, x'_C) \psi_{CD}(x'_C, x'_D) \]

\[ Z = \sum_{x'_A} \sum_{x'_B} \sum_{x'_C} \sum_{x'_D} \psi_{AB}(x'_A, x'_B) \psi_{BC}(x'_B, x'_C) \psi_{CD}(x'_C, x'_D) \]

\[ = \sum_{x'_A} \sum_{x'_B} \sum_{x'_C} \psi_{AB}(x'_A, x'_B) \psi_{BC}(x'_B, x'_C) \phi_C(x'_C) \]

\[ \phi_B(x'_B) \]

\[ = \sum_{x'_A} \sum_{x'_B} \psi_{AB}(x'_A, x'_B) \sum_{x'_C} \psi_{BC}(x'_B, x'_C) \phi_C(x'_C) \]
Inference

\[ Z = \sum_{x'_A, x'_B, x'_C, x'_D} \psi_{AB}(x'_A, x'_B) \psi_{BC}(x'_B, x'_C) \psi_{CD}(x'_C, x'_D) \]

\[ = \sum_{x'_A} \sum_{x'_B} \sum_{x'_C} \sum_{x'_D} \psi_{AB}(x'_A, x'_B) \psi_{BC}(x'_B, x'_C) \psi_{CD}(x'_C, x'_D) \]

\[ = \sum_{x'_A} \sum_{x'_B} \psi_{AB}(x'_A, x'_B) \phi_B(x'_B) \]
Inference

\[ Z = \sum_{x_A', x_B', x_C', x_D'} \psi_{AB}(x_A', x_B') \psi_{BC}(x_B', x_C') \psi_{CD}(x_C', x_D') \]

\[ = \sum_{x_A'} \sum_{x_B'} \sum_{x_C'} \sum_{x_D'} \psi_{AB}(x_A', x_B') \psi_{BC}(x_B', x_C') \psi_{CD}(x_C', x_D') \]

\[ = \sum_{x_A'} \sum_{x_B'} \psi_{AB}(x_A', x_B') \phi_A(x_A') \]

\[ = \sum_{x_A'} \sum_{x_B'} \psi_{AB}(x_A', x_B') \phi_B(x_B') \]
Inference

\[ Z = \sum_{x_A', x_B', x_C', x_D'} \psi_{AB}(x_A', x_B') \psi_{BC}(x_B', x_C') \psi_{CD}(x_C', x_D') \]

\[ = \sum_{x_A'} \sum_{x_B'} \sum_{x_C'} \sum_{x_D'} \psi_{AB}(x_A', x_B') \psi_{BC}(x_B', x_C') \psi_{CD}(x_C', x_D') \]

\[ = \sum_{x_A'} \phi_A(x_A') \]
Variable Elimination

• Choose an ordering of the random variables

• Sum the joint distribution over the variables one at a time in the specified order exploiting the factorization where possible

  • Each time a variable is eliminated, it creates a new potential that is multiplied back in after removing the sum that generated this potential
Variable Elimination

• What is the cost of the optimal variable elimination on the chain?
Variable Elimination

- What is the cost of the optimal variable elimination on the chain?

\[ \text{length of the chain} \times (\text{size of state space})^2 \]
Another Example

Elimination order: C, B, D, F, E, A

(worked out on the board)
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
Another Example

Elimination order: C, B, D, F, E, A
• The treewidth of a graph is equal to the size of the largest clique created in any optimal elimination ordering

• Tree width of a tree: ?
• The treewidth of a graph is equal to the size of the largest clique created in any optimal elimination ordering

• Tree width of a tree: 1 (as long as it has at least one edge)

• The complexity of variable elimination is upper bounded by

\[ n \cdot \text{(size of the state space)}^{\text{treewidth}+1} \]
What is the Treewidth of this Graph?
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A
What is the Treewidth of this Graph?

Elimination order: D, C, F, E, B, A

Largest clique created had size two
(this is the best that we can do)
Finding the optimal elimination ordering is NP-hard!

Heuristic methods are often used in practice

- Min-degree: the cost of a vertex is the number of neighbors it has in the current graph

- Min-fill: the cost of a vertex is the number of new edges that need to be added to the graph due to its elimination
Belief Propagation

- Efficient method for inference on a tree

- Represent the variable elimination process as a collection of messages passed between nodes in the tree
  
  - The messages keep track of the potential functions produced throughout the elimination process

- Optimal elimination order on a tree always eliminates leaves of the current tree (i.e., always eliminate degree 1 vertices)
Belief Propagation

- \[ p(x_1, ..., x_n) = \frac{1}{Z} \prod_{i \in V} \phi_i(x_i) \prod_{(i, j) \in E} \psi_{ij}(x_i, x_j) \]

\[ m_{i \rightarrow j}(x_j) = \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}(x_i) \]

where \( N(i) \) is the set of neighbors of node \( i \) in the graph

- Messages are passed in two phases: from the leaves up to the root and then from the root down to the leaves
Belief Propagation

• As an added bonus, BP allows you to efficiently compute the marginal probability over each single variable as well as the partition function
  • Multiply the singleton potentials with all of the incoming messages
  • Computing the normalization constant for this function gives the partition function of the model
• A similar strategy when the factor graph is a tree
  • Two types of messages: factor-to-variable and variable-to-factor
Belief Propagation

• What is the complexity of belief propagation on a tree with state space $D$?
Belief Propagation

• What is the complexity of belief propagation on a tree with state space $D$?

$$O(n|D|^2)$$

• What if we want to compute the MAP assignment instead of the partition function?