Inference and Representation

Rachel Hodos

New York University

Lab 5, September 30, 2015

Outline

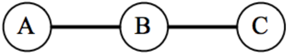
1. Graph Separation in MRFs
2. Revisiting Conditional Random Fields
3. Treewidth and Belief Propagation
4. Pruning “barren nodes”

Graph separation in MRFs

Given an undirected graph G, any distribution that can be represented by G (i.e. written as a product over clique potentials) must satisfy *independence through separation*.

Proof of graph separation in MRFs

- We will show that $A \perp C \mid B$ for the following distribution:



$$p(a,b,c) = \frac{1}{Z}\phi_{AB}(a,b)\phi_{BC}(b,c)$$

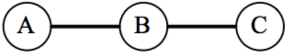
- First, we show that $p(a \mid b)$ can be computed using only $\phi_{AB}(a,b)$:

$$\begin{aligned} p(a \mid b) &= \frac{p(a,b)}{p(b)} \\ &= \frac{\frac{1}{Z}\sum_{\hat{c}} \phi_{AB}(a,b)\phi_{BC}(b,\hat{c})}{\frac{1}{Z}\sum_{\hat{a},\hat{c}} \phi_{AB}(\hat{a},b)\phi_{BC}(b,\hat{c})} \\ &= \frac{\phi_{AB}(a,b)\sum_{\hat{c}} \phi_{BC}(b,\hat{c})}{\sum_{\hat{a}} \phi_{AB}(\hat{a},b)\sum_{\hat{c}} \phi_{BC}(b,\hat{c})} = \frac{\phi_{AB}(a,b)}{\sum_{\hat{a}} \phi_{AB}(\hat{a},b)}. \end{aligned}$$

- More generally, the probability of a variable conditioned on its Markov blanket depends *only* on potentials involving that node

Proof of graph separation in MRFs

- We will show that $A \perp C \mid B$ for the following distribution:



$$p(a, b, c) = \frac{1}{Z}\phi_{AB}(a, b)\phi_{BC}(b, c)$$

Proof.

$$\begin{aligned} p(a, c \mid b) = \frac{p(a, c, b)}{\sum_{\hat{a},\hat{c}} p(\hat{a}, b, \hat{c})} &= \frac{\phi_{AB}(a, b)\phi_{BC}(b, c)}{\sum_{\hat{a},\hat{c}} \phi_{AB}(\hat{a}, b)\phi_{BC}(b, \hat{c})} \\ &= \frac{\phi_{AB}(a, b)\phi_{BC}(b, c)}{\sum_{\hat{a}} \phi_{AB}(\hat{a}, b) \sum_{\hat{c}} \phi_{BC}(b, \hat{c})} \\ &= p(a \mid b)p(c \mid b) \end{aligned}$$

$\square$

Intuition

- Information can only flow between variables along paths

Intuition

- Information can only flow between variables along paths
- Paths can be broken into sub-paths of length 3

Intuition

- Information can only flow between variables along paths
- Paths can be broken into sub-paths of length 3
- We showed that conditioning on the middle variable of a path makes that path inactive

Intuition

- Information can only flow between variables along paths
- Paths can be broken into sub-paths of length 3
- We showed that conditioning on the middle variable of a path makes that path inactive
- Since MRFs are undirected, there is only one type of length-3 path

Formal definition of a CRF

- A CRF is a Markov network on variables $\mathbf{X} \cup \mathbf{Y}$, which specifies the conditional distribution

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \phi_c(\mathbf{x}_c, \mathbf{y}_c)$$

  with partition function

$$Z(\mathbf{x}) = \sum_{\hat{\mathbf{y}}} \prod_{c \in C} \phi_c(\mathbf{x}_c, \hat{\mathbf{y}}_c).$$

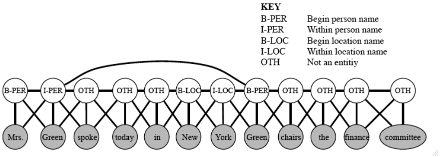
- As before, two variables in the graph are connected with an undirected edge if they appear together in the scope of some factor
- The only difference with a normal Markov network is the normalization term
- Common applications: NLP, computer vision

Example #1 (NLP): named-entity recognition

- Given a sentence, determine the people and organizations involved and the relevant locations:
  “Mrs. Green spoke today in New York. Green chairs the finance committee.”
- Entities sometimes span multiple words. Entity of a word not obvious without considering its *context*
- CRF has one variable X_i for each word, which encodes the possible labels of that word
- The labels are, for example, “B-person, I-person, B-location, I-location, B-organization, I-organization”
  - Having beginning (B) and within (I) allows the model to segment adjacent entities

Example #1 (NLP): named-entity recognition

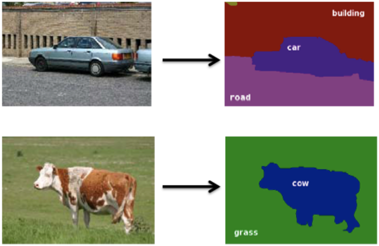
The graphical model looks like (called a *skip-chain CRF*):



There are three types of potentials:

- $\phi^1(Y_t, Y_{t+1})$ represents dependencies between neighboring target variables [analogous to transition distribution in a HMM]
- $\phi^2(Y_t, Y_{t'})$ for all pairs t, t' such that $x_t = x_{t'}$, because if a word appears twice, it is likely to be the same entity
- $\phi^3(Y_t, X_1, \cdots, X_T)$ for dependencies between an entity and the word sequence [e.g., may have features taking into consideration capitalization]

Example #2 (vision): Image segmentation



- Problem: Given an image $\mathbf{X} \in \mathbb{R}^{m \times n \times 3}$, produce a labeling $\mathbf{Y} \in \{1, \ldots, k\}^{m \times n}$.
- The labels $1, \ldots, k$ could correspond to e.g. {*grass, sky, tree*}.

Example #2 (vision): Image segmentation

- Approach: Define a grid-structured CRF to model $P(\mathbf{Y}|\mathbf{X})$, where potentials are based on the intuition that neighboring pixels with similar colors should probably have the same label.
- Pairwise potentials over labels for neighboring pixels $i, i+1$:

$$\phi_{i,i+1}(y_i, y_{i+1}) = \exp\left(\mathbb{1}_{y_i=y_{i+1}}\|x_i - x_{i+1}\| - \mathbb{1}_{y_i\neq y_{i+1}}\|x_i - x_{i+1}\|\right)$$

- x_i represents the 3-dimensional RGB for pixel i
- Then find the MAP solution for Y:

$$Y^* = \text{argmax}_Y P(\mathbf{Y}|\mathbf{X})$$

Treewidth

- The **width** of an induced graph is #nodes in largest clique - 1
- We define the **induced width** $w_{\mathcal{G},\prec}$ to be the width of the graph $\mathcal{I}_{\mathcal{G},\prec}$ induced by applying VE to $\mathcal{G}$ using ordering $\prec$
- The **treewidth**, or "minimal induced width" of graph $\mathcal{G}$ is

$$w_{\mathcal{G}}^{*} = \min_{\prec} w_{\mathcal{G},\prec}$$

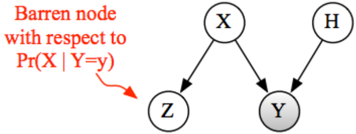
- The treewidth provides a bound on the best running time achievable by VE on a distribution that factorizes over $\mathcal{G}$: $O(mk^{w_{\mathcal{G}}^{*}+1})$,
- Unfortunately, finding the **best** elimination ordering (equivalently, computing the treewidth) for a graph is NP-hard
- In practice, heuristics are used to find a good elimination ordering

Belief Propagation

(Presented on board)

Pruning nodes in Bayesian networks

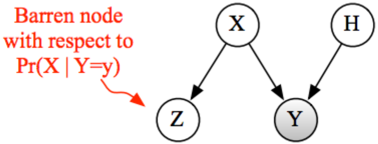
- A node in a Bayesian network $\mathcal{G}$ is a *leaf* if it has no children.
- **Def:** A node is *barren* w.r.t. a query $p_{\mathcal{G}}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y})$ if it is a leaf and it is not in $\mathbf{X} \cup \mathbf{Y}$



- To *remove* a node v from a Bayesian network $\mathcal{G} = (V, E)$ means:
  1. Removing v from V, and removing from E all edges to/from v
  2. Leave the CPDs for the rest of the variables the same
- **Theorem:** Let $\mathcal{G}'$ be the Bayesian network obtained from $\mathcal{G}$ by removing v. If v is barren w.r.t. the query $p_{\mathcal{G}}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y})$, then

$$p_{\mathcal{G}}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y}) = p_{\mathcal{G}'}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y}).$$

Pruning nodes in Bayesian networks



$$
\begin{aligned}
p_{\mathcal{G}}(X = x \mid Y = y) &= \frac{\sum_{h,z} p_{\mathcal{G}}(z, x, y, h)}{\sum_{\hat{x},h,z} p_{\mathcal{G}}(z, \hat{x}, y, h)} \\
&= \frac{\sum_{h,z} \theta_x \theta_h \theta_{z|x} \theta_{y|x,h}}{\sum_{\hat{x},h,z} \theta_{\hat{x}} \theta_h \theta_{z|\hat{x}} \theta_{y|\hat{x},h}} \\
&= \frac{\sum_{h} \theta_x \theta_h \theta_{y|x,h} \sum_z \theta_{z|x}}{\sum_{\hat{x},h} \theta_{\hat{x}} \theta_h \theta_{y|\hat{x},h} \sum_z \theta_{z|\hat{x}}} \\
&= p_{\mathcal{G}'}(X = x \mid Y = y),
\end{aligned}
$$

where $\mathcal{G}'$ is the Bayesian network with Z removed.

Pruning nodes in Bayesian networks

- **Def:** A node is *barren* w.r.t. a query $p_{\mathcal{G}}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y})$ if it is a leaf and it is not in $\mathbf{X} \cup \mathbf{Y}$
- **Theorem:** Let $\mathcal{G}'$ be the Bayesian network obtained from $\mathcal{G}$ by removing v. If v is barren w.r.t. the query $p_{\mathcal{G}}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y})$, then

$$p_{\mathcal{G}}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y}) = p_{\mathcal{G}'}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y}).$$

- Let $An(\mathbf{X} \cup \mathbf{Y})$ be the *ancestral set* of $\mathbf{X} \cup \mathbf{Y}$, i.e. the set including $\mathbf{X} \cup \mathbf{Y}$ and all of their ancestors
- **Corollary:** All the nodes outside of $An(\mathbf{X} \cup \mathbf{Y})$ are irrelevant to the query $p_{\mathcal{G}}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y})$ and can be removed
- **Theorem:** Let $\mathcal{G}'$ be the Bayesian network obtained from $\mathcal{G}$ by removing all nodes that are d-separated from X by Y. Then

$$p_{\mathcal{G}}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y}) = p_{\mathcal{G}'}(\mathbf{X} \mid \mathbf{Y} = \mathbf{y}).$$