Graphical Models - Part II

Bishop PRML Ch. 8

Alireza Ghane
Outline

Probabilistic Models

Bayesian Networks

Markov Random Fields

Inference
Outline

Probabilistic Models

Bayesian Networks

Markov Random Fields

Inference
Joint Distribution

• In general, to answer a query on random variables $Q = Q_1, \ldots, Q_N$ given evidence $E = e, E = E_1, \ldots, E_M, e = e_1, \ldots, e_M$:

$$p(Q|E = e) = \frac{p(Q, E = e)}{p(E = e)}$$

$$= \frac{\sum_h p(Q, E = e, H = h)}{\sum_{q,h} p(Q = q, E = e, H = h)}$$
Problems

- The joint distribution is large
  - e. g. with $K$ boolean random variables, $2^K$ entries
- Inference is slow, previous summations take $O(2^K)$ time
- Learning is difficult, data for $2^K$ parameters
- Analogous problems for continuous random variables
Reminder - Independence

- \( A \) and \( B \) are independent iff
  \[ p(A|B) = p(A) \quad \text{or} \quad p(B|A) = p(B) \quad \text{or} \quad p(A, B) = p(A)p(B) \]

- \( p(\text{Toothache}, \text{Catch}, \text{Cavity}, \text{Weather}) = p(\text{Toothache}, \text{Catch}, \text{Cavity})p(\text{Weather}) \)
  - 32 entries reduced to 12 (\( \text{Weather} \) takes one of 4 values)

- Absolute independence powerful but rare

- Dentistry is a large field with hundreds of variables, none of which are independent. What to do?
Graphical Models

• Graphical Models provide a visual depiction of probabilistic model
• Conditional indepence assumptions can be seen in graph
• Inference and learning algorithms can be expressed in terms of graph operations
• We will look at 2 types of graph (can be combined)
  • Directed graphs: Bayesian networks
  • Undirected graphs: Markov Random Fields
  • Factor graphs (won’t cover)
Outline

Probabilistic Models

Bayesian Networks

Markov Random Fields

Inference
Bayesian Networks

• A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

• Syntax:
  • a set of nodes, one per variable
  • a directed, acyclic graph (link ≈ “directly influences”)
  • a conditional distribution for each node given its parents:
    \[ p(X_i|pa(X_i)) \]

• In the simplest case, conditional distribution represented as a **conditional probability table (CPT)** giving the distribution over \( X_i \) for each combination of parent values
Example

• I’m at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn’t call. Sometimes it’s set off by minor earthquakes. Is there a burglar?

• Variables: Burglar, Earthquake, Alarm, JohnCalls, MaryCalls

• Network topology reflects “causal” knowledge:
  • A burglar can set the alarm off
  • An earthquake can set the alarm off
  • The alarm can cause Mary to call
  • The alarm can cause John to call
Example contd.

| B | E | P(A|B,E) |
|---|---|---------|
| T | T | .95     |
| T | F | .94     |
| F | T | .29     |
| F | F | .001    |

| A   | P(J|A) |
|-----|-------|
| T   | .90   |
| F   | .05   |

| A   | P(M|A) |
|-----|-------|
| T   | .70   |
| F   | .01   |
Global Semantics

- **Global semantics** defines the full joint distribution as the product of the local conditional distributions:

\[
P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|pa(X_i))
\]

e.g., \(P(j \land m \land a \land \neg b \land \neg e) = \)

\[
P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e)
\]

\[
= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998
\]

\[
\approx 0.00063
\]
Constructing Bayesian Networks

- Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics

1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to $n$
   - add $X_i$ to the network
   - select parents from $X_1, \ldots, X_{i-1}$ such that
   $$p(X_i | pa(X_i)) = p(X_i | X_1, \ldots, X_{i-1})$$

- This choice of parents guarantees the global semantics:

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i | X_1, \ldots, X_{i-1}) \quad \text{(chain rule)}$$

$$= \prod_{i=1}^{n} p(X_i | pa(X_i)) \quad \text{(by construction)}$$
Constructing Bayesian Networks

- Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics

1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to $n$
   - add $X_i$ to the network
   - select parents from $X_1, \ldots, X_{i-1}$ such that
     \[ p(X_i|pa(X_i)) = p(X_i|X_1, \ldots, X_{i-1}) \]

- This choice of parents guarantees the global semantics:

\[
p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|X_1, \ldots, X_{i-1}) \quad \text{(chain rule)}
\]

\[
= \prod_{i=1}^{n} p(X_i|pa(X_i)) \quad \text{(by construction)}
\]
Example

Suppose we choose the ordering $M, J, A, B, E$

$P(J|M) = P(J)$?
Example

Suppose we choose the ordering $M, J, A, B, E$

\[
P(J|M) = P(J)\? \quad \text{No}
\]
\[
\]
Example

Suppose we choose the ordering $M, J, A, B, E$

Graphical Models

$P(J|M) = P(J)$? No
$P(B|A, J, M) = P(B|A)$?
$P(B|A, J, M) = P(B)$?
Example

Suppose we choose the ordering $M, J, A, B, E$

$$P(J|M) = P(J)\quad \text{No}$$
$$P(A|J, M) = P(A|J)\quad P(A|J, M) = P(A)\quad \text{No}$$
$$P(B|A, J, M) = P(B|A)\quad \text{Yes}$$
$$P(B|A, J, M) = P(B)\quad \text{No}$$
$$P(E|B, A, J, M) = P(E|A)?$$
$$P(E|B, A, J, M) = P(E|A, B)?$$
Example

Suppose we choose the ordering $M, J, A, B, E$

\[
P(J|M) = P(J)\? \quad \text{No}
\]
\[
P(A|J, M) = P(A|J)\? \; P(A|J, M) = P(A)\? \quad \text{No}
\]
\[
P(B|A, J, M) = P(B|A)\? \quad \text{Yes}
\]
\[
P(B|A, J, M) = P(B)\? \quad \text{No}
\]
\[
P(E|B, A, J, M) = P(E|A)\? \quad \text{No}
\]
\[
P(E|B, A, J, M) = P(E|A, B)\? \quad \text{Yes}
\]
Example contd.

- Deciding conditional independence is hard in noncausal directions
  - (Causal models and conditional independence seem hardwired for humans!)
- Assessing conditional probabilities is hard in noncausal directions
- Network is less compact: \(1 + 2 + 4 + 2 + 4 = 13\) numbers needed
Example - Car Insurance

http://aispace.org/bayes
Example - Polynomial Regression

- Bayesian polynomial regression model
- Observations $t = (t_1, \ldots, t_N)$
- Vector of coefficients $w$
- Inputs $x$ and noise variance $\sigma^2$ were assumed fixed, not stochastic and hence not in model
- Joint distribution:

$$p(t, w) = p(w) \prod_{n=1}^{N} p(t_n | w)$$
Plates

- A shorthand for writing repeated nodes such as the $t_n$ uses plates
Deterministic Model Parameters

- Can also include deterministic parameters (not stochastic) as small nodes
- Bayesian polynomial regression model:

\[ p(t, w | x, \alpha, \sigma^2) = p(w | \alpha) \prod_{n=1}^{N} p(t_n | w, x_n, \sigma^2) \]
• In polynomial regression, we assumed we had a training set of \( N \) pairs \((x_n, t_n)\)
• Convention is to use shaded nodes for observed random variables
• Suppose we wished to predict the value $\hat{t}$ for a new input $\hat{x}$
• The Bayesian network used for this inference task would be this one
Specifying Distributions - Discrete Variables

• Earlier we saw the use of conditional probability tables (CPT) for specifying a distribution over discrete random variables with discrete-valued parents

• For a variable with no parents, with \( K \) possible states:

\[
p(\mathbf{x}|\mathbf{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}
\]

• e.g. \( p(B) = 0.001^B_1 0.999^B_2 \), 1-of-\( K \) representation

<table>
<thead>
<tr>
<th>Burglary</th>
<th>Earthquake</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(B)</td>
<td>P(E)</td>
</tr>
<tr>
<td>.001</td>
<td>.002</td>
</tr>
</tbody>
</table>

\[
\begin{array}{c|cc}
B & E & P(A|B,E) \\
\hline
T & T & .95 \\
T & F & .94 \\
F & T & .29 \\
F & F & .001 \\
\end{array}
\]

\[
\begin{array}{c|cc}
A & P(J|A) \\
\hline
T & .90 \\
F & .05 \\
\end{array}
\]

\[
\begin{array}{c|cc}
A & P(M|A) \\
\hline
T & .70 \\
F & .01 \\
\end{array}
\]
Specifying Distributions - Discrete Variables cont.

- With two variables $x_1, x_2$ can have two cases

  - Dependent

    \[
    p(x_1, x_2 | \mu) = p(x_1 | \mu)p(x_2 | x_1, \mu) = \left( \prod_{k=1}^{K} \mu_{k1}^{x_{1k}} \right) \left( \prod_{k=1}^{K} \prod_{j=1}^{K} \mu_{kj2}^{x_{1k}x_{2j}} \right)
    \]

    - $K^2 - 1$ free parameters in $\mu$

  - Independent

    \[
    p(x_1, x_2 | \mu) = p(x_1 | \mu)p(x_2 | \mu) = \left( \prod_{k=1}^{K} \mu_{k1}^{x_{1k}} \right) \left( \prod_{k=1}^{K} \mu_{k2}^{x_{2k}} \right)
    \]

    - $2(K - 1)$ free parameters in $\mu$
Chains of Nodes

- With $M$ nodes, could form a chain as shown above.
- Number of parameters is:

$$\underbrace{(K - 1)}_{x_1} + \underbrace{(M - 1) K (K - 1)}_{\text{others}}$$

- Compare to:
  - $K^M - 1$ for fully connected graph
  - $M(K - 1)$ for graph with no edges (all independent)
• Another way to reduce number of parameters is sharing parameters (a. k. a. tying of parameters)
• Lower graph reuses same $\mu$ for nodes $2-M$
  • $\mu$ is a random variable in this network, could also be deterministic
• $(K-1) + K(K-1)$ parameters
Specifying Distributions - Continuous Variables

- One common type of conditional distribution for continuous variables is the linear-Gaussian

\[
p(x_i | pa_i) = \mathcal{N} \left( x_i; \sum_{j \in pa_i} w_{ij} x_j + b_i, \nu_i \right)
\]
Specifying Distributions - Continuous Variables

- One common type of conditional distribution for continuous variables is the linear-Gaussian

\[ p(x_i|pa_i) = \mathcal{N}(x_i; \sum_{j \in pa_i} w_{ij}x_j + b_i, v_i) \]

- e.g. With one parent Harvest:

\[ p(c|h) = \mathcal{N}(c; -0.5h + 5, 1) \]

- For harvest \( h \), mean cost is \( -0.5h + 5 \), variance is 1
**Linear Gaussian**

- Interesting fact: if all nodes in a Bayesian Network are linear Gaussian, joint distribution is a multivariate Gaussian

\[
p(x_i | pa_i) = \mathcal{N} \left( x_i; \sum_{j \in pa_i} w_{ij} x_j + b_i, v_i \right)
\]

\[
p(x_1, \ldots, x_N) = \prod_{i=1}^{N} \mathcal{N} \left( x_i; \sum_{j \in pa_i} w_{ij} x_j + b_i, v_i \right)
\]

- Each factor looks like \( \exp \left( (x_i - (w_i^T x_{pa_i})^2) \right) \), this product will be another quadratic form
- With no links in graph, end up with diagonal covariance matrix
- With fully connected graph, end up with full covariance matrix
Conditional Independence in Bayesian Networks

• Recall again that $a$ and $b$ are conditionally independent given $c$ ($a \perp \perp b|c$) if
  • $p(a|b, c) = p(a|c)$ or equivalently
  • $p(a, b|c) = p(a|c)p(b|c)$

• Before we stated that links in a graph are $\approx$ “directly influences”
• We now develop a correct notion of links, in terms of the conditional independences they represent
  • This will be useful for general-purpose inference methods
A Tale of Three Graphs - Part 1

• The graph above means

\[ p(a, b, c) = p(a|c)p(b|c)p(c) \]
\[ p(a, b) = \sum_c p(a|c)p(b|c)p(c) \]
\[ \neq p(a)p(b) \text{ in general} \]

• So \( a \) and \( b \) not independent
A Tale of Three Graphs - Part 1

- However, conditioned on $c$

\[
p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a|c)p(b|c)p(c)}{p(c)} = p(a|c)p(b|c)
\]

- So $a \perp\!\!\!\!\!\!\!\!\!\!\!\perp b|c$
A Tale of Three Graphs - Part 1

- Note the path from $a$ to $b$ in the graph
  - When $c$ is not observed, path is open, $a$ and $b$ not independent
  - When $c$ is observed, path is blocked, $a$ and $b$ independent
- In this case $c$ is tail-to-tail with respect to this path
A Tale of Three Graphs - Part 2

- The graph above means

\[ p(a, b, c) = p(a)p(b|c)p(c|a) \]

- Again, \( a \) and \( b \) not independent
• However, conditioned on $c$

$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a)p(b|c)}{p(c)}p(c|a)$$

$$= \frac{p(a)p(b|c)}{p(c)} \frac{p(a|c)p(c)}{p(a)}$$

Bayes’ Rule

$$= p(a|c)p(b|c)$$

• So $a \perp\!\!\!\!\!\!\!\!\!\perp b|c$
A Tale of Three Graphs - Part 2

- As before, the path from $a$ to $b$ in the graph
  - When $c$ is not observed, path is open, $a$ and $b$ not independent
  - When $c$ is observed, path is blocked, $a$ and $b$ independent
- In this case $c$ is head-to-tail with respect to this path
A Tale of Three Graphs - Part 3

The graph above means

\[ p(a, b, c) = p(a)p(b)p(c|a, b) \]
\[ p(a, b) = \sum_c p(a)p(b)p(c|a, b) \]
\[ = p(a)p(b) \]

This time \( a \) and \( b \) are independent
A Tale of Three Graphs - Part 3

- However, conditioned on $c$

\[
p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a)p(b)p(c|a, b)}{p(c)} \\
\neq p(a|c)p(b|c) \text{ in general}
\]

- So $a \indep b|c$
A Tale of Three Graphs - Part 3

- Frustratingly, the behaviour here is different
  - When \( c \) is not observed, path is blocked, \( a \) and \( b \) independent
  - When \( c \) is observed, path is unblocked, \( a \) and \( b \) not independent
- In this case \( c \) is head-to-head with respect to this path
- Situation is in fact more complex, path is unblocked if any descendent of \( c \) is observed
Part 3 - Intuition

- Binary random variables $B$ (battery charged), $F$ (fuel tank full), $G$ (fuel gauge reads full)
- $B$ and $F$ independent
- But if we observe $G = 0$ (false) things change
  - e.g. $p(F = 0 | G = 0, B = 0)$ could be less than $p(F = 0 | G = 0)$, as $B = 0$ explains away the fact that the gauge reads empty
  - Recall that $p(F | G, B) = p(F | G)$ is another $F \perp \perp B | G$
A general statement of conditional independence

For sets of nodes $A, B, C$, check all paths from $A$ to $B$ in graph

If all paths are blocked, then $A \perp \!\!\!\!\!\!\!\!\!\!\perp B | C$

Path is blocked if:

- Arrows meet head-to-tail or tail-to-tail at a node in $C$
- Arrows meet head-to-head at a node, and neither node nor any descendent is in $C'$
Naive Bayes

- Commonly used naive Bayes classification model
- Class label $z$, features $x_1, \ldots, x_D$
- Model assumes features independent given class label
  - Tail-to-tail at $z$, blocks path between features
Markov Blanket

- What is the minimal set of nodes which makes a node $x_i$ conditionally independent from the rest of the graph?
  - $x_i$’s parents, children, and children’s parents (co-parents)
- Define this set $MB$, and consider:

$$p(x_i | x\{j\neq i\}) = \frac{p(x_1, \ldots, x_D)}{\int p(x_1, \ldots, x_D) dx_i} = \frac{\prod_k p(x_k | pa_k)}{\int \prod_k p(x_k | pa_k) dx_i}$$

- All factors other than those for which $x_i$ is $x_k$ or in $pa_k$ cancel
Learning Parameters

- When all random variables are observed in training data, relatively straight-forward
  - Distribution factors, all factors observed
  - e.g. Maximum likelihood used to set parameters of each distribution $p(x_i|pa_i)$ separately
- When some random variables not observed, it’s tricky
  - This is a common case
  - Expectation-maximization(EM) is a method for this
<table>
<thead>
<tr>
<th>Probabilistic Models</th>
<th>Bayesian Networks</th>
<th>Markov Random Fields</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline</td>
<td>Probabilistic Models</td>
<td>Bayesian Networks</td>
<td>Markov Random Fields</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conditional Independence in Graphs

- Recall that for Bayesian Networks, conditional independence was a bit complicated
  - \textit{d-separation} with head-to-head links
- We would like to construct a graphical representation such that conditional independence is straight-forward path checking
Markov Random Fields

- **Markov random fields** (MRFs) contain one node per variable
- Undirected graph over these nodes
- Conditional independence will be given by simple separation, blockage by observing a node on a path
  - e.g. in above graph, \( A \perp \perp B \mid C \)
With this simple check for conditional independence, **Markov blanket** is also simple

- Recall Markov blanket $MB$ of $x_i$ is set of nodes such that $x_i$ conditionally independent from rest of graph given $MB$
- Markov blanket is neighbours
MRF Factorization

• Remember that graphical models define a factorization of the joint distribution

• What should be the factorization so that we end up with the simple conditional independence check?

• For $x_i$ and $x_j$ not connected by an edge in graph:

  $$x_i \perp \perp x_j \mid \mathcal{X} \setminus \{i, j\}$$

• So there should not be any factor $\psi(x_i, x_j)$ in the factorized form of the joint
• A **clique** in a graph is a subset of nodes such that there is a link between every pair of nodes in the subset

• A **maximal clique** is a clique for which one cannot add another node and have the set remain a clique
MRF Joint Distribution

- Note that nodes in a clique cannot be made conditionally independent from each other
  - So defining factors $\psi(\cdot)$ on nodes in a clique is “safe”
- The joint distribution for a Markov random field is:

$$p(x_1, \ldots, x_K) = \frac{1}{Z} \prod_C \psi_C(x_C)$$

where $x_C$ is the set of nodes in clique $C$, and the product runs over all maximal cliques
- Each $\psi_C(x_C) \geq 0$
- $Z$ is a normalization constant
The joint distribution for a Markov random field is:

\[
p(x_1, \ldots, x_4) = \frac{1}{Z} \prod_C \psi_C(x_C)
\]

\[
= \frac{1}{Z} \psi_{123}(x_1, x_2, x_3)\psi_{234}(x_2, x_3, x_4)
\]

Note that maximal cliques subsume smaller ones: \(\psi_{123}(x_1, x_2, x_3)\) could include \(\psi_{12}(x_1, x_2)\), though sometimes smaller cliques are explicitly used for clarity.
MRF Joint - Terminology

• The joint distribution for a Markov random field is:

\[ p(x_1, \ldots, x_K) = \frac{1}{Z} \prod_C \psi_C(x_C) \]

• Each \( \psi_C(x_C) \) is called a potential function

• \( Z \), the normalization constant, is called the partition function:

\[ Z = \sum_x \prod_C \psi_C(x_C) \]

• \( Z \) is very costly to compute, since it is a sum/integral over all possible states for all variables in \( x \)

• Don’t always need to evaluate it though, will cancel for computing conditional probabilities
Hammersley-Clifford

- The definition of the joint:

\[ p(x_1, \ldots, x_K) = \frac{1}{Z} \prod_C \psi_C(x_C) \]

- Note that we started with particular conditional independences
- We then formulated the factorization based on clique potentials
  - This formulation resulted in the right conditional independences
- The converse is true as well, any distribution with the conditional independences given by the undirected graph can be represented using a product of clique potentials
- This is the Hammersley-Clifford theorem
Energy Functions

- Often use exponential, which is non-negative, to define potential functions:
  \[ \psi_C(x_C) = \exp\{-E_C(x_C)\} \]

- Minus sign — by convention
- \( E_C(x_C) \) is called an energy function
  - From physics, low energy = high probability
- This exponential representation is known as the Boltzmann distribution
Energy Functions - Intuition

- Joint distribution nicely rearranges as

\[ p(x_1, \ldots, x_K) = \frac{1}{Z} \prod_C \psi_C(x_C) \]

\[ = \frac{1}{Z} \exp\left\{ - \sum_C E_C(x_C) \right\} \]

- Intuition about potential functions: \( \psi_C \) are describing good (low energy) sets of states for adjacent nodes
- An example of this is next
Image Denoising

- Consider the problem of trying to correct (denoise) an image that has been corrupted
- Assume image is binary
- Observed (noisy) pixel values $y_i \in \{-1, +1\}$
- Unobserved true pixel values $x_i \in \{-1, +1\}$
Image Denoising - Graphical Model

- Cliques containing each true pixel value $x_i \in \{-1, +1\}$ and observed value $y_i \in \{-1, +1\}$
- Observed pixel value is usually the same as true pixel value
- Energy function $-\eta x_i y_i$, $\eta > 0$, lower energy (better) if $x_i = y_i$
- Cliques containing adjacent true pixel values $x_i, x_j$
- Nearby pixel values are usually the same
- Energy function $-\beta x_i x_j$, $\beta > 0$, lower energy (better) if $x_i = x_j$
Image Denoising - Graphical Model

- Cliques containing each true pixel value $x_i \in \{-1, +1\}$ and observed value $y_i \in \{-1, +1\}$
  - Observed pixel value is usually same as true pixel value
  - Energy function $-\eta x_i y_i$, $\eta > 0$, lower energy (better) if $x_i = y_i$
• Cliques containing each true pixel value $x_i \in \{-1, +1\}$ and observed value $y_i \in \{-1, +1\}$
  • Observed pixel value is usually same as true pixel value
  • Energy function $-\eta x_i y_i$, $\eta > 0$, lower energy (better) if $x_i = y_i$

• Cliques containing adjacent true pixel values $x_i, x_j$
  • Nearby pixel values are usually the same
  • Energy function $-\beta x_i x_j$, $\beta > 0$, lower energy (better) if $x_i = x_j$
Image Denoising - Graphical Model

- Complete energy function:

\[ E(x, y) = -\beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i \]

- Joint distribution:

\[ p(x, y) = \frac{1}{Z} \exp\{-E(x, y)\} \]

- Or, as potential functions\( \psi_n(x_i, x_j) = \exp(\beta x_i x_j) \),\( \psi_p(x_i, y_i) = \exp(\eta x_i y_i) \):

\[ p(x, y) = \frac{1}{Z} \prod_{i,j} \psi_n(x_i, x_j) \prod_i \psi_p(x_i, y_i) \]
Image Denoising - Inference

- The denoising query is $\arg \max_{x} p(x | y)$
- Two approaches:
  - **Iterated conditional modes** (ICM): hill climbing in $x$, one variable $x_i$ at a time
    - Simple to compute, Markov blanket is just observation plus neighbouring pixels
  - **Graph cuts**: formulate as max-flow/min-cut problem, exact inference (for this graph)
Converting Directed Graphs into Undirected Graphs

- Consider a simple directed chain graph:

\[ p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_2) \ldots p(x_N|x_{N-1}) \]

- Can convert to undirected graph
Converting Directed Graphs into Undirected Graphs

- Consider a simple directed chain graph:

\[ p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_2) \ldots p(x_N|x_{N-1}) \]

- Can convert to undirected graph:

\[ p(\mathbf{x}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3) \ldots \psi_{N-1,N}(x_{N-1}, x_N) \]

where \( \psi_{1,2} = p(x_1)p(x_2|x_1) \), all other \( \psi_{k-1,k} = p(x_k|x_{k-1}) \),
\( Z = 1 \)
Converting Directed Graphs into Undirected Graphs

• The chain was straight-forward because for each conditional \( p(x_i | pa_i) \), nodes \( x_i \cup pa_i \) were contained in one clique
  • Hence we could define that clique potential to include that conditional

• For a general undirected graph we can force this to occur by “marrying” the parents
  • Add links between all parents in \( pa_i \)
  • This process known as moralization, creating a moral graph
Strong Morals

- Start with directed graph on left
- Add undirected edges between all parents of each node
- Remove directionality from original edges
• Initialize all potential functions to be 1
• With moral graph, for each $p(x_i|p_{a_i})$, there is at least one clique which contains all of $x_i \cup p_{a_i}$
  • Multiply $p(x_i|p_{a_i})$ into potential function for one of these cliques
• $Z = 1$ again since:

$$p(\mathbf{x}) = \prod_C \psi_C(\mathbf{x}_C) = \prod_i p(x_i|p_{a_i})$$

which is already normalized
Equivalence Between Graph Types

- Note that the moralized undirected graph loses some of the conditional independence statements of the directed graph.
- Further, there are certain conditional independence assumptions which can be represented by directed graphs which cannot be represented by undirected graphs, and vice versa.
Equivalence Between Graph Types

- Note that the moralized undirected graph loses some of the conditional independence statements of the directed graph.
- Further, there are certain conditional independence assumptions which can be represented by directed graphs which cannot be represented by undirected graphs, and vice versa.
Equivalence Between Graph Types

• Note that the moralized undirected graph loses some of the conditional independence statements of the directed graph
• Further, there are certain conditional independence assumptions which can be represented by directed graphs which cannot be represented by indirected graphs, and vice versa
• Directed graph: $A \perp B \mid \emptyset$, $A \parallel B \mid C$, cannot be represented using undirected graph
Equivalence Between Graph Types

- Note that the moralized undirected graph loses some of the conditional independence statements of the directed graph.
- Further, there are certain conditional independence assumptions which can be represented by directed graphs which cannot be represented by indirected graphs, and vice versa.
- Directed graph: $A \perp B|\emptyset$, $A \not
dependent B|C$, cannot be represented using undirected graph.
Equivalence Between Graph Types

- Note that the moralized undirected graph loses some of the conditional independence statements of the directed graph.
- Further, there are certain conditional independence assumptions which can be represented by directed graphs which cannot be represented by undirected graphs, and vice versa.
- Directed graph: $A \perp \!\!\!\!\perp B|\emptyset$, $A \perp \!\!\!\!\perp B|C$, cannot be represented using an undirected graph.
- Undirected graph: $A \perp \!\!\!\!\perp B|\emptyset$, $A \perp \!\!\!\!\perp B|C \cup D$, $C \perp \!\!\!\!\perp D|A \cup B$ cannot be represented using a directed graph.
Outline

Probabilistic Models

Bayesian Networks

Markov Random Fields

Inference
• **Inference** is the process of answering queries such as
  \[ p(x_n | x_e = e) \]
  
• We will focus on computing **marginal posterior distributions** over single variables \( x_n \) using
  \[ p(x_n | x_e = e) \propto p(x_n, x_e = e) \]

• The proportionality constant can be obtained by enforcing
  \[ \sum_{x_n} p(x_n | x_e = e) = 1 \]
Inference on a Chain

• Consider a simple undirected chain
• For inference, we want to compute $p(x_n, x_e = e)$
• First, we’ll show how to compute $p(x_n)$
  • $p(x_n, x_e = e)$ will be a simple modification of this
Inference on a Chain

- The naive method of computing the marginal $p(x_n)$ is to write down the factored form of the joint, and marginalize (sum out) all other variables:

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} p(x)$$

$$= \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} \frac{1}{Z} \prod_C \psi_C(x_C)$$

- This would be slow – $O(K^N)$ work if each variable could take $K$ values
Inference on a Chain

- However, due to the factorization terms in this summation can be rearranged nicely
- This will lead to efficient algorithms
Simple Algebra

- This efficiency comes from a very simple distributivity

\[ ab + ac = a(b + c) \]

- Or more complicated version

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} a_i b_j = a_1 b_1 + a_1 b_2 + \ldots + a_n b_n \\
= (a_1 + \ldots + a_n)(b_1 + \ldots + b_n)
\]

- Much faster to do right hand side \((2(n - 1)\) additions, 1 multiplication) than left hand side \((n^2\) multiplications, \(n^2 - 1\) additions)
A Simple Chain

First consider a chain with 3 nodes, and computing \( p(x_3) \):

\[
p(x_3) = \sum_{x_1} \sum_{x_2} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3)
\]

\[
= \sum_{x_2} \psi_{23}(x_2, x_3) \left[ \sum_{x_1} \psi_{12}(x_1, x_2) \right]
\]
Performing the sums

\[ p(x_3) = \sum_{x_2} \psi_{23}(x_2, x_3) \left[ \sum_{x_1} \psi_{12}(x_1, x_2) \right] \]

- For example, if \( x_i \) are binary:

\[
\psi_{12}(x_1, x_2) = x_1 \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \psi_{23}(x_2, x_3) = x_2 \begin{bmatrix} s & t \\ u & v \end{bmatrix}
\]
Performing the sums

\[ p(x_3) = \sum_{x_2} \psi_{23}(x_2, x_3) \left[ \sum_{x_1} \psi_{12}(x_1, x_2) \right] \]

- For example, if \( x_i \) are binary:

\[ \psi_{12}(x_1, x_2) = x_1 \begin{bmatrix} a & b \\ c & d \end{bmatrix} \]

\[ \psi_{23}(x_2, x_3) = x_2 \begin{bmatrix} s & t \\ u & v \end{bmatrix} \]

\[ \sum_{x_1} \psi_{12}(x_1, x_2) = \begin{bmatrix} a + c & b + d \end{bmatrix} \equiv \mu_{12}(x_2) \]
Performing the sums

\[ p(x_3) = \sum_{x_2} \psi_{23}(x_2, x_3) \left[ \sum_{x_1} \psi_{12}(x_1, x_2) \right] \]

- For example, if \( x_i \) are binary:

\[ \psi_{12}(x_1, x_2) = x_1 \begin{bmatrix} a & b \\ c & d \end{bmatrix}_{x_2} \]

\[ \psi_{23}(x_2, x_3) = x_2 \begin{bmatrix} s & t \\ u & v \end{bmatrix}_{x_3} \]

\[ \sum_{x_1} \psi_{12}(x_1, x_2) = \begin{bmatrix} a + c & b + d \end{bmatrix}_{x_2} \equiv \mu_{12}(x_2) \]

\[ \psi_{23}(x_2, x_3) \times \mu_{12}(x_2) = x_2 \begin{bmatrix} s(a + c) & t(a + c) \\ u(b + d) & v(b + d) \end{bmatrix}_{x_3} \]
Performing the sums

\[ p(x_3) = \sum_{x_2} \psi_{23}(x_2, x_3) \left[ \sum_{x_1} \psi_{12}(x_1, x_2) \right] \]

- For example, if \( x_i \) are binary:

\[
\psi_{12}(x_1, x_2) = x_1 \begin{bmatrix} a & b \\ c & d \end{bmatrix}_{x_2} \\
\psi_{23}(x_2, x_3) = x_2 \begin{bmatrix} s & t \\ u & v \end{bmatrix}_{x_3}
\]

\[
\sum_{x_1} \psi_{12}(x_1, x_2) = \begin{bmatrix} a + c & b + d \end{bmatrix}_{x_2} \equiv \mu_{12}(x_2)
\]

\[
\psi_{23}(x_2, x_3) \times \mu_{12}(x_2) = x_2 \begin{bmatrix} s(a + c) & t(a + c) \\ u(b + d) & v(b + d) \end{bmatrix}_{x_3}
\]

\[
p(x_3) = \begin{bmatrix} s(a + c) + u(b + d) & t(a + c) + v(b + d) \end{bmatrix}_{x_3}
\]
Complexity of Inference

- There were two types of operations
  - Summation
    \[ \sum_{x_1} \psi_{12}(x_1, x_2) \]
    \( K \times K \) numbers in \( \psi_{12} \), takes \( O(K^2) \) time
  - Multiplication
    \[ \psi_{23}(x_2, x_3) \times \mu_{12}(x_2) \]
    Again \( O(K^2) \) work
- For a chain of length \( N \), we repeat these operations \( N - 1 \) times each
  - \( O(NK^2) \) work, versus \( O(N^K) \) for naive evaluation
More complicated chain

- Now consider a 5 node chain, again asking for $p(x_3)$

$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5)$$
More complicated chain

• Now consider a 5 node chain, again asking for $p(x_3)$

\[
p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5)
\]

\[
= \sum_{x_2} \sum_{x_1} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \sum_{x_4} \sum_{x_5} \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5)
\]
More complicated chain

- Now consider a 5 node chain, again asking for $p(x_3)$

$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5)$$

$$= \sum_{x_2} \sum_{x_1} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \sum_{x_4} \sum_{x_5} \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5)$$

$$= \left[ \sum_{x_2} \sum_{x_1} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \right] \left[ \sum_{x_4} \sum_{x_5} \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5) \right]$$
More complicated chain

- Now consider a 5 node chain, again asking for $p(x_3)$

$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5)$$

$$= \sum_{x_2} \sum_{x_1} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \sum_{x_4} \sum_{x_5} \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5)$$

$$= \left[ \sum_{x_2} \sum_{x_1} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \right] \left[ \sum_{x_4} \sum_{x_5} \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5) \right]$$

- Each of these factors resembles the previous, and can be computed efficiently
  - Again $O(NK^2)$ work
The factors can be thought of as messages being passed between nodes in the graph

\[ \mu_{12}(x_2) \equiv \sum_{x_1} \psi_{12}(x_1, x_2) \]

is a message passed from node \( x_1 \) to node \( x_2 \) containing all information in node \( x_1 \n\)

In general,

\[ \mu_{k-1,k}(x_k) = \sum_{x_{k-1}} \psi_{k-1,k}(x_{k-1}, x_k) \mu_{k-2,k-1}(x_{k-1}) \]

Possible to do so because of conditional independence
Computing All Marginals

- Computing one marginal $p(x_n)$ takes $O(NK^2)$ time
- Naively running same algorithms for all nodes in a chain would take $O(N^2K^2)$ time
- But this isn’t necessary, same messages can be reused at all nodes in the chain
- Pass all messages from one end of the chain to the other, pass all messages in the other direction too
- Can compute marginal at any node by multiplying the two messages delivered to the node
  - $2(N - 1)K^2$ work, twice as much as for just one node
Including Evidence

- If a node $x_{k-1} = e$ is observed, simply clamp to observed value rather than summing:

$$
\mu_{k-1,k}(x_k) = \sum_{x_{k-1}} \psi_{k-1,k}(x_{k-1}, x_k) \mu_{k-2,k-1}(x_{k-1})
$$

becomes

$$
\mu_{k-1,k}(x_k) = \psi_{k-1,k}(x_{k-1} = e, x_k) \mu_{k-2,k-1}(x_{k-1} = e)
$$
• The algorithm for a tree-structured graph is very similar to that for chains
• Formulation in PRML uses factor graphs, we’ll just give the intuition here
• Consider calculating the marginal \( p(x_n) \) for the center node of the graph at right
• Treat \( x_n \) as root of tree, pass messages from leaf nodes up to root
Trees

- Message passing similar to that in chains, but possibly multiple messages reaching a node
- With multiple messages, multiply them together
- As before, sum out variables

\[ \mu_{k-1,k}(x_k) = \sum_{x_{k-1}} \psi_{k-1,k}(x_{k-1}, x_k) \mu_{k-2,k-1}(x_{k-1}) \]

- Known as sum-product algorithm
- Complexity still \( O(NK^2) \)
Most Likely Configuration

- A similar algorithm exists for finding

$$\arg \max_{x_1, \ldots, x_N} p(x_1, \ldots, x_N)$$

- Replace summation operations with maximize operations
- Maximum of products at each node
- Known as max-sum, since often take log-probability to avoid underflow errors
General Graphs

- **Junction tree algorithm** is an exact inference method for arbitrary graphs
  - A particular tree structure defined over cliques of variables
  - Inference ends up being exponential in maximum clique size
  - Therefore slow in many cases
- **Approximate inference techniques**
  - **Loopy belief propagation**: run message passing scheme (sum-product) for a while
    - Sometimes works
    - Not guaranteed to converge
  - **Variational methods**: approximate desired distribution using analytically simple forms, find parameters to make these forms similar to actual desired distribution (Ch. 10, we won’t cover)
  - **Sampling methods**: represent desired distribution with a set of samples, as more samples are used, obtain more accurate representation (Ch. 11, we will cover)
Conclusion

• Readings: Ch. 8
• Graphical models depict conditional independence assumptions
• Directed graphs (Bayesian networks)
  • Factorization of joint distribution as conditional on node given parents
• Undirected graphs (Markov random fields)
  • Factorization of joint distribution as clique potential functions
• Inference algorithm sum-product, based on local message passing
  • Works for tree-structured graphs
  • Non-tree-structured graphs, either slow exact or approximate inference