Probabilistic Graphical Models (I)

Hongxin Zhang
zhx@cad.zju.edu.cn

State Key Lab of CAD&CG, ZJU
2015-03-31
Probabilistic Graphical Models

- Modeling many real-world problems => a large number of random variables
  - Dependences among variables may be used to reduce the size to encode the model (PCA ?), or
  - They may be the goal by themselves, that is, the idea is to understand the correlations among variables.
Modeling the domain

- Discrete random variables
  - Take 5 random binary variables \((A, B, C, D, E)\)
  - i.i.d. data from a multinomial distribution

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>~b</td>
<td>~c</td>
<td>~d</td>
<td>~e</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>b</td>
<td>~c</td>
<td>d</td>
<td>~e</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>~b</td>
<td>c</td>
<td>d</td>
<td>~e</td>
</tr>
</tbody>
</table>
Goals

- (Parameter) Learning: using training data, estimate the joint distribution
  - Which are the values $P(A, B, C, D, E)$?
  - ... and if there were one hundred binary variables? (Size of model certainly greater than number of atoms on Earth!)

- Inference: Given the distribution $P(A, B, C, D, E)$,
  - Belief updating: compute the probability of an event
    - What is the probability of $A=a$ given $E=e$?
  - Maximum a posteriori: compute the states of variables that maximize their probability.
    - Which state of $A$ maximizes $P(A | E=e)$? Is it $a$ or $\sim a$?
The unstructured approach

- To specify the joint distribution, there is an exponential number of values:

  \[ p(a, b, c, d, e), p(a, b, c, d, \neg e), p(a, b, c, \neg d, e), \]
  \[ p(a, b, c, \neg d, \neg e), p(a, b, \neg c, d, e), p(a, b, \neg c, d, \neg e), \]
  \[ p(a, b, \neg c, \neg d, e), p(a, b, \neg c, \neg d, \neg e), \ldots \]

- We can compute the probability of events by:

  \[ p(a) = \sum_{B,C,D,E} p(a, B, C, D, E) \]
  \[ p(a|d, \neg e) = \frac{p(a, d, \neg e)}{p(d, \neg e)} = \frac{\sum_{B,C} p(a, B, C, d, \neg e)}{\sum_{A,B,C} p(A, B, C, d, \neg e)} \]

- There are exponentially many terms in the summations...
The naïve Bayesian approach

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

- Application: Email spanning
Bayesian Networks

- An arbitrary **joint distribution** $p(a, b, c)$ over three variables $a$, $b$, and $c$
  - the product rule of probability:
    \[
    p(a, b, c) = p(c \mid a, b) p(a, b) \\
    = p(c \mid a, b) p(b \mid a) p(a)
    \]

- General case: $p(x_1, x_2, \ldots, x_K)$
  \[
  p(x_1, \ldots, x_K) = p(x_K \mid x_1, \ldots, x_{K-1}) \cdots p(x_2 \mid x_1) p(x_1)
  \]
Not fully connected graph

- Joint distribution: \( p(x_1, x_2, \ldots, x_7) \)

\[
p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)
\]
General form

- For a graph with $K$ nodes, the joint distribution is given by:

$$p(x) = \prod_{k=1}^{K} p(x_k|\text{pa}_k)$$

- where $\text{pa}_k$ denotes the set of parents of $x_k$, and $x = \{x_1, \ldots, x_K\}$
Definitions

- A set of variables associated with nodes of a Directed Acyclic Graph (DAG).
- Markov condition (w.r.t. the DAG): each variable is independent of its non-descendants given its parents.
- For each variable (node), local probability distributions:
  - $P(A)$, $P(B|A=a)$, $P(B|A=a)$, $P(C|A=a)$, $P(C|A=\sim a)$, $P(D|b, c)$, $P(D|b, c)$, $P(D|b, c)$, $P(D|b, c)$; $P(D|b, c)$, $P(E|c)$, $P(E|c)$,
- All these values are precise.
Regression revisit: Polynomial Curve Fitting

\[ t(x, w) = w_0 + w_1 h_1(x) + w_2 h_2(x) + \ldots + w_N h_N(x) = \sum_{j=0}^{N} w_j h_j(x) \]

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

\[ t = h(x) \cdot w \]

\[ w = (H^T H)^{-1} H^T t \]

Normal equation
Example: Polynomial regression

\[ t(x, w) = w_0 + w_1 h_1(x) + w_2 h_2(x) + \ldots + w_N h_N(x) = \sum_{j=0}^{N} w_j h_j(x) \]

\[ p(t, w) = p(w) \prod_{n=1}^{N} p(t_n|w) \]
Example: Polynomial regression

\[ t(x, \mathbf{w}) = w_0 + w_1 h_1(x) + w_2 h_2(x) + \ldots + w_N h_N(x) = \sum_{j=0}^{N} w_j h_j(x) \]

\[ p(t, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^{N} p(t_n|\mathbf{w}) \]
Example: Polynomial regression

\[ t(x, \mathbf{w}) = w_0 + w_1 h_1(x) + w_2 h_2(x) + \ldots + w_N h_N(x) = \sum_{j=0}^{N} w_j h_j(x) \]

\[ p(t, \mathbf{w} | \mathbf{x}, \alpha, \sigma^2) = p(\mathbf{w} | \alpha) \prod_{n=1}^{N} p(t_n | \mathbf{w}, x_n, \sigma^2) \]

the noise variance \( \sigma^2 \), and the hyperparameter \( \alpha \) representing the precision of the Gaussian prior over \( \mathbf{w} \)
Linear-Gaussian models

- Consider an arbitrary DAG over $D$ variables in which node $i$ represents a single continuous random variable $x_i$ having a Gaussian distribution.

- The mean of this distribution is taken to be a linear combination of the states of its parent nodes $pa_i$ of node $i$.

$$p(x_i|pa_i) = \mathcal{N} \left( x_i \left| \sum_{j \in pa_i} w_{ij} x_j + b_i, v_i \right. \right)$$
Linear-Gaussian models

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

\[
\ln p(x) = \sum_{i=1}^{D} \ln p(x_i | \text{pa}_i)
\]

\[ = - \sum_{i=1}^{D} \frac{1}{2v_i} \left( x_i - \sum_{j \in \text{pa}_i} w_{ij}x_j - b_i \right)^2 + \text{const} \]
Linear-Gaussian models

\[ p(x_i|\text{pa}_i) = \mathcal{N}\left(x_i \left| \sum_{j \in \text{pa}_i} w_{ij} x_j + b_i, \nu_i \right. \right) \]

\[ x_i = \sum_{j \in \text{pa}_i} w_{ij} x_j + b_i + \sqrt{\nu_i} \epsilon_i \]

\[ \mathbb{E}[x_i] = \sum_{j \in \text{pa}_i} w_{ij} \mathbb{E}[x_j] + b_i \]

\[ \text{cov}[x_i, x_j] = \mathbb{E}\left[ (x_i - \mathbb{E}[x_i])(x_j - \mathbb{E}[x_j]) \right] \]

\[ = \mathbb{E}\left[ (x_i - \mathbb{E}[x_i]) \left\{ \sum_{k \in \text{pa}_j} w_{jk} (x_k - \mathbb{E}[x_k]) + \sqrt{\nu_j} \epsilon_j \right\} \right] \]

\[ = \sum_{k \in \text{pa}_j} w_{jk} \text{cov}[x_i, x_k] + I_{ij} \nu_j \]
Linear-Gaussian models

- **Case 1: no links in the graph**
  - The joint distribution:
    - $2D$ parameters and represents
    - $D$ independent univariate Gaussian distributions.

- **Case 2: fully connected graph**
  - $D(D-1)/2 + D$ independent parameters

- **Case 3:**

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

\[
\mu = (b_1, b_2 + w_{21}b_1, b_3 + w_{32}b_2 + w_{32}w_{21}b_1)^T
\]

\[
\Sigma = \begin{pmatrix}
  v_1 & w_{21}v_1 & w_{32}w_{21}v_1 \\
  w_{21}v_1 & v_2 + w_{21}^2v_1 & w_{32}(v_2 + w_{21}^2v_1) \\
  w_{32}w_{21}v_1 & w_{32}(v_2 + w_{21}^2v_1) & v_3 + w_{32}^2(v_2 + w_{21}^2v_1)
\end{pmatrix}
\]
Conditional independence

- Three random variables: a, b, and c
  - a is conditionally independent of b given c iff
    \[ P(a \mid b, c) = P(a \mid c) \]
    \[ a \perp b \mid c \]
  - This can be re-written in following way
    \[ P(a, b \mid c) = P(a \mid b, c) P(b \mid c) \]
    \[ = P(a \mid c) P(b \mid c) \]
    The joint distribution of a and b factorizes into the product of the marginal distribution of a and ~b.
Simple example (1)

- **Joint distribution:**
  \[ P(a, b, c) = P(a \mid c) P(b \mid c) P(c) \]

- **Condition on c:**
  \[ P(a, b \mid c) = P(a, b, c) / P(c) = P(a \mid c) P(b \mid c) \]
  \[ \implies a \independent b \mid c \]
Simple example (2)

- **Joint distribution:**
  \[ P(a, b, c) = P(a) \cdot P(c | a) \cdot P(b | c) \]

- **Factorization:**
  \[
P(a, b) = \sum_c P(a, b, c) = P(a) \sum_c P(c | a)P(b | c)
  \]
  \[
  = P(a)P(b | a)
  \]

- **Condition on c:**
  \[
P(a, b | c) = \frac{P(a, b, c)}{P(c)} = \frac{P(a)P(c | a)P(b | c)}{P(c)}
  \]
  \[
  = P(a | c)P(b | c)
  \]

Bayesian Theorem
Simple example (3)

- **Joint distribution:**
  
  \[ P(a, b, c) = P(a)P(b)P(c | a, b) \]

- **Factorization:**
  
  \[
  P(a, b) = \sum_c P(a, b, c) = P(a)P(b)\sum_c P(c | a, b) \\
  = P(a)P(b)
  \]

- **Condition on c:**
  
  \[
  P(a, b | c) = \frac{P(a, b, c)}{P(c)} = \frac{P(a)P(b)P(c | a, b)}{P(c)} \\
  \neq P(a | c)P(b | c)
  \]
Conditional independence

- Tail-to-Tail: yes
- Head-to-Tail: yes
- Head-to-Head: no
Markov condition

- We say that node $y$ is a *descendant* of node $x$ if there is a path from $x$ to $y$ in which each step of the path follows the directions of the arrows.
- If each variable is independent of its non-descendants given its parents, then:

$$
\begin{align*}
B & \perp (C, E) | A, \\
D & \perp (A, E) | (B, C), \\
E & \perp (A, B, D) | C.
\end{align*}
$$
D-separation

- All possible paths from any node in $A$ to any node in $B$. Any such path is said to be *blocked* if it includes a node such that either
  - the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set $C$, or
  - the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in the set $C$
- If all paths are blocked, then $A$ is said to be *d-separated* from $B$ by $C$. 
D-separation

- In graph (a), the path from \( a \) to \( b \) is not blocked by node \( c \)
- In graph (b), the path from \( a \) to \( b \) is blocked by node \( f \) and \( e \)
D-separation

- A particular directed graph represents a specific decomposition of a joint probability distribution into a product of conditional probabilities
- A directed graph is a filter
Markov blanket

- Joint distribution $p(x_1, \ldots, x_D)$ represented by a directed graph having $D$ nodes

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

- The set of nodes comprising the parents, the children and the co-parents is called the Markov blanket
Markov Random Fields

- Also known as a *Markov network* or an *undirected graphical model*

- Conditional independence properties:

  Conditional dependence exists if there exists a path that connects any node in $A$ to any node in $B$.

  If there are no such paths, then the conditional independence property must hold.
Clique

- A subset of the nodes in a graph such that there exists a link between all pairs of nodes in the subset
  - In other words, the set of nodes in a clique is fully connected
  - Maximal clique …
  - A four-node undirected graph showing a clique (outlined in green) and a maximal clique (outlined in blue)
Potential function

- \( x_C \): the set of variables in that clique \( C \)
- The joint distribution is written as a product of potential functions \( \psi_C(x_C) \) over the maximal cliques of the graph

\[
p(x) = \frac{1}{Z} \prod_C \psi_C(x_C)
\]

- The quantity \( Z \), called the partition function, is a normalization constant

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

- Potential functions \( \psi_C(x_C) \) are strictly positive. Possible choice

\[
\psi_C(x_C) = \exp \{-E(x_C)\}
\]
Image de-noising

Bayes' Theorem
Bayes' Theorem
Bayes' Theorem
Bayes' Theorem
Relation to directed graphs

- **Joint distribution:**
  - Directed:
    \[
p(x) = p(x_1)p(x_2|x_1)p(x_3|x_2) \cdots p(x_N|x_{N-1})\]
  - Undirected:
    \[
p(x) = \frac{1}{Z} \psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N)\]
Relation to directed graphs

(a)

\[ \psi_{1,2}(x_1, x_2) = p(x_1)p(x_2|x_1) \]
\[ \psi_{2,3}(x_2, x_3) = p(x_3|x_2) \]
\[ \vdots \]
\[ \psi_{N-1,N}(x_{N-1}, x_N) = p(x_N|x_{N-1}) \]
Relation to directed graphs

- this process of ‘marrying the parents’ has become known as **moralization**, and the resulting undirected graph, after dropping the arrows, is called the **moral graph**.

\[ p(x) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \]
Inference in Graphical Models

\[ p(x, y) = p(x)p(y|x) \]

\[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} \]
Inference on a chain

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

\[ p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} p(x) \]
Inference on a chain

\[ p(x_n) = \frac{1}{Z} \]

\[
\begin{bmatrix}
\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \\
\sum_{x_2} \psi_{2,3}(x_2, x_3) \\
\sum_{x_1} \psi_{1,2}(x_1, x_2)
\end{bmatrix} \ldots
\begin{bmatrix}
\sum_{x_{N-1}} \psi_{N-1,N}(x_{N-1}, x_N)
\end{bmatrix}
\]

\[ \mu_{\alpha}(x_n) \]

\[ \mu_{\beta}(x_n) \]

(8.52)
Inference on a chain

Passing of local *messages* around on the graph

\[
p(x_n) = \frac{1}{Z} \left( \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[ \sum_{x_2} \psi_{2,3}(x_2, x_3) \left[ \sum_{x_1} \psi_{1,2}(x_1, x_2) \right] \cdots \right] \right)
\]

\[
\mu_\alpha(x_n)
\]

\[
\sum_{x_{n+1}} \psi_{n+1,n}(x_n, x_{n+1}) \cdots \left[ \sum_{x_{N-1}} \psi_{N-1,N}(x_{N-1}, x_N) \right] \cdots.
\]

\[
(8.52)
\]

\[
p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n)
\]
Inference on a chain

Passing of local messages around on the graph

\[ p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n) = \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \mu_\alpha(x_{n-1}). \]

\[ \mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \]
Inference on a chain

Passing of local *messages* around on the graph

\[
\mu_\beta(x_n) = \sum_{x_{n+1}} \psi_{n+1,n}(x_{n+1}, x_n) \left[ \sum_{x_{n+2}} \cdots \right]
\]

\[
p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n) = \sum_{x_{n+1}} \psi_{n+1,n}(x_{n+1}, x_n) \mu_\beta(x_{n+1}).
\]
Inference on a chain

Passing of local *messages* around on the graph

\[
p(x_{n-1}, x_n) = \frac{1}{Z} \mu_\alpha(x_{n-1}) \psi_{n-1,n}(x_{n-1}, x_n) \mu_\beta(x_n)
\]
Tree

(a)  

(b)  

(c)
Factor graph

- the joint distribution over a set of variables in the form of a product of factors

\[ p(x) = \prod_{s} f_s(x_s) \]

- where \( x_s \) denotes a subset of the variables
Factor graph

\[ p(x) = f_a(x_1, x_2) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3) \]
Factor graph

- an undirected graph => a factor graph
  - create variable nodes corresponding to the nodes in the original undirected graph
  - create additional factor nodes corresponding to the maximal cliques $x_s$
  - Multiple choices of $f_g$
(a) An undirected graph with a single clique potential $\psi(x_1, x_2, x_3)$.

(b) A factor graph with factor $f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$ representing the same distribution as the undirected graph.

(c) A different factor graph representing the same distribution, whose factors satisfy $f_a(x_1, x_2, x_3)f_b(x_1, x_2) = \psi(x_1, x_2, x_3)$. 
The sum-product algorithm

- The problem of finding the marginal $p(x)$ for particular variable node $x$

\[ p(x) = \sum_{x \setminus x} p(x) \]

\[ p(x) = \prod_{s \in \text{ne}(x)} F_s(x, X_s) \]
The sum-product algorithm

- The problem of finding the marginal $p(x)$ for particular variable node $x$

\[
p(x) = \prod_{s \in \text{ne}(x)} \left[ \sum_{X_s} F_s(x, X_s) \right] \\
= \prod_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x).
\]

\[
\mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} F_s(x, X_s)
\]
The sum-product algorithm

- The problem of finding the marginal $p(x)$ for particular variable node $x$

\[ \mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} F_s(x, X_s) \]
The sum-product algorithm

- The problem of finding the marginal $p(x)$ for particular variable node $x$

\[ F_s(x, X_s) = f_s(x, x_1, \ldots, x_M)G_1(x_1, X_{s1}) \ldots G_M(x_M, X_{sM}) \]

\[ \mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} F_s(x, X_s) \]
The sum-product algorithm

- The problem of finding the marginal $p(x)$ for particular variable node $x$

\[
\mu_{f_s \rightarrow x}(x) = \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \left[ \sum_{X_{x_m}} G_m(x_m, X_{sm}) \right]
\]

\[
= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f_s}(x_m) \quad (8.66)
\]

\[
\mu_{x_m \rightarrow f_s}(x_m) \equiv \sum_{X_{sm}} G_m(x_m, X_{sm})
\]
Junction tree algorithm

- deal with graphs having loops

Algorithm:
- directed graph $\Rightarrow$ undirected graph (moralization)
- The graph is triangulated
- join tree
- Junction tree
- a two-stage message passing algorithm, essentially equivalent to the sum-product algorithm
Graph inference example

- Computer-Generated Residential Building Layouts [SIG ASIA 2010]
The End

新浪微博: @浙大张宏鑫

微信公众号:  

![QR Code]