Chapter 15
Graphical Models

supplementary slides to
Machine Learning Fundamentals
© Hui Jiang 2020
published by Cambridge University Press

August 2020
Outline

1. Concepts of Graphical Models
2. Bayesian Networks
3. Markov Random Fields
Graphical Models

- graphical models: a graphical representation for generative models

- a graphical model essentially represents a joint distribution of some random variables
  - a node for a random variable
  - an arc for relationship between random variables

- two different types of graphical models:
  1. directed graphical models, a.k.a. Bayesian networks, use directed arcs, representing conditional distributions
  2. undirected graphical models, a.k.a. Markov random fields, use undirected arcs
Bayesian Networks

- each directed arc represents a conditional distribution among nodes
- a Bayesian network (BN) represents a way to factorize a joint distribution of all underlying random variable

\[
p(x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | \text{pa}(x_i))
\]

- e.g. a joint distribution of 5 R.V.’s:

\[
p(x_1, x_2, x_3, x_4, x_5) = p(x_1) \cdot p(x_2 | x_1) \cdot p(x_3 | x_1, x_2) \cdot p(x_4 | x_1, x_2, x_3) \cdot p(x_5 | x_1, x_2, x_3, x_4)
\]
Bayesian Networks: A Sparse Example

- why use graphical representations for joint distributions?

- a sparse graph indicates some conditional independence among variables

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

- if each conditional distribution is chosen from e-family, a Bayesian network represents another distribution in e-family
Bayesian Networks for Discrete Random Variables

- All nodes represent discrete random variables.
- An $M$-value random variable $x$ is encoded as a 1-of-$M$ vector $x = [x_1 \ x_2 \ \cdots \ x_M]^T$.
- Conditional probabilities are stored as tables:
  \[ \mu_{ij} \triangleright= \Pr(x = i \mid y = j) = \Pr(x_i = 1 \mid y_j = 1) \]
- Notations for conditional distributions:
  \[
p(x \mid y) = p(x \mid y) = \prod_{i=1}^{M} \prod_{j=1}^{N} \mu_{ij}^{x_i y_j}
  \]
  \[
p(x \mid y, z) = p(x \mid y, z) = \prod_{i=1}^{M} \prod_{j=1}^{N} \prod_{k=1}^{K} \mu_{ijk}^{x_i y_j z_k}
  \]
Bayesian Networks: Outline

1. Conditional Independence
2. Represent Generative Models as Bayesian Networks
3. Learning Bayesian Networks
4. Inference Algorithms
5. Case Study (I): Naive Bayes Classifier
6. Case Study (II): Latent Dirichlet Allocation
Conditional Independence

- two random variables are independent

\[ x \perp y \iff p(x, y) = p(x)p(y) \]

- two random variables are conditionally independent

\[ x \perp y \mid z \iff p(x, y \mid z) = p(x \mid z)p(y \mid z) \]

- a missing link in Bayesian networks normally indicates some conditional independence among variables

- conditional independence: useful for interpreting data and simplifying computation

- how to identify conditional independence in Bayesian networks?
Confounding

- confounding: a fork junction pattern $x \leftarrow z \rightarrow y$, where $z$ is called a confounder
  
  \[ p(x, y, z) = p(z) \cdot p(x|z) \cdot p(y|z) \]

- unconditionally dependent: a confounder introduces spurious association
  
  \[ x \not\perp y \iff p(x, y) \neq p(x)p(y) \]

- conditionally independent:
  
  \[ x \perp y \mid z \iff p(x, y \mid z) = p(x \mid z)p(y \mid z) \]

- the ice-cream example
  
  - eating ice-cream $\implies$ drowning in pool?
  - confounding $\neq$ causation
a *chain* junction pattern $x \rightarrow z \rightarrow y$, where $z$ is called a *mediator*

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

unconditionally dependent: a mediator introduces spurious association

\[ x \not\perp y \iff p(x, y) \neq p(x)p(y) \]

conditionally independent:

\[ x \perp y \mid z \iff p(x, y \mid z) = p(x \mid z)p(y \mid z) \]

the cooking example

- hungry $\implies$ burning fingers?
- mediating $\neq$ causation
Colliding

- a *colliding* junction pattern $x \rightarrow z \leftarrow y$, where $z$ is called a *collider*

\[
p(x, y, z) = p(x) \cdot p(y) \cdot p(z \mid x, y)
\]

- unconditionally independent:

\[
x \perp y \iff p(x, y) = p(x)p(y)
\]

- conditionally dependent:

\[
x \not\perp y \mid z \iff p(x, y \mid z) \neq p(x \mid z)p(y \mid z)
\]

- the explain-away phenomenon
Colliding Causes Explain-away

- there exist two independent causes for a common effect
- observing one will explain away another
- the wet driveway example
  - rain $\implies$ wet driveway
  - leaking pipe $\implies$ wet driveway
  - after observing the wet driveway ($W = 1$), we have

$$\Pr(R = 1 \mid W = 1) = 0.3048$$

$$\Pr(R = 1 \mid W = 1, L = 1) = 0.1667$$

$\Pr(R = 1) = 0.1$

$\Pr(L = 1) = 0.01$

$\Pr(W = 1 \mid R = 1, L = 1) = 0.90$

$\Pr(W = 1 \mid R = 1, L = 0) = 0.80$

$\Pr(W = 1 \mid R = 0, L = 1) = 0.50$

$\Pr(W = 1 \mid R = 0, L = 0) = 0.20$
Conditional Independence: d-separation Rule

- for any three disjoint subsets $A$, $B$ and $C$, $A \perp B \mid C \iff$ any path between $A$ and $B$ is blocked by $C$
- a.k.a. $A$ and $B$ are d-separated by $C$
- a path is blocked by $C$ if both hold:
  1. all confounders and mediators along the path belong to $C$
  2. neither any collider nor any of its descendants belongs to $C$
- e.g. we can verify:
  
  $a \not\perp f \mid c \quad a \not\perp b \mid c$
  
  $a \not\perp c \mid f \quad a \perp b \mid f \quad e \perp b \mid f$
**Bayesian Networks: Representing Generative Models (I)**

- **Gaussian models:**
  
  \[ p(x_i) = \mathcal{N}(x_i \mid \mu, \Sigma) \]

- **Bayesian learning of Gaussian models:**
  
  \[ p(\mu, x_1, \cdots x_N) = p(\mu) \prod_{i=1}^{N} p(x_i \mid \mu) \]
Gaussian mixture models (GMMs):

- latent variable: the index is encoded as a 1-of-M vector $\mathbf{z}_i = [z_{i1} \ z_{i2} \ \cdots \ z_{iM}]$
- $p(\mathbf{z}_i) = \prod_{m=1}^{M} (w_m)^{z_{im}}$
- $p(\mathbf{x}_i | \mathbf{z}_i) = \prod_{m=1}^{M} \left( \mathcal{N}(\mathbf{x}_i | \mu_m, \Sigma_m) \right)^{z_{im}}$

joint distribution:

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_1, \cdots, \mathbf{z}_N) = \prod_{i=1}^{N} p(\mathbf{z}_i) p(\mathbf{x}_i | \mathbf{z}_i)$$

marginal distribution:

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N) = \prod_{i=1}^{N} \left( \sum_{\mathbf{z}_i} p(\mathbf{z}_i) p(\mathbf{x}_i | \mathbf{z}_i) \right)$$
Bayesian Networks: Representing Generative Models (III)

- Bayesian learning of Gaussian mixture models (GMMs):

\[
p(w) = \text{Dir}(w \mid \alpha^{(0)})
\]

\[
p(z_i \mid w) = \prod_{m=1}^{M} (w_m)^{z_{im}} \quad \forall i = 1, 2, \cdots N
\]

\[
p(\Sigma_m) = \mathcal{W}^{-1}(\Sigma_m \mid \Phi_m^{(0)}, \nu_m^{(0)}) \quad \forall m = 1, 2, \cdots M
\]

\[
p(\mu_m \mid \Sigma_m) = \mathcal{N}\left(\mu_m \mid \nu_m^{(0)}, \frac{1}{\lambda_m^{(0)}} \Sigma_m\right) \quad \forall m = 1, 2, \cdots M
\]

\[
p(x_i \mid z_i, \{\mu_m, \Sigma_m\}) = \prod_{m=1}^{M} \left(\mathcal{N}(x_i \mid \mu_m, \Sigma_m)\right)^{z_{im}} \quad \forall i = 1, 2, \cdots N
\]
Bayesian Networks: Representing Generative Models (IV)

- Markov chain models
  - 1st-order: \( p(x_i | x_{i-1}) \)
  - 2nd-order: \( p(x_i | x_{i-1}, x_{i-2}) \)

- hidden Markov models (HMMs)
  
  \[
  p(s_1, \ldots, s_T, x_1, \ldots, x_T) = p(s_1)p(x_1 | s_1) \prod_{t=2}^{T} p(s_t | s_{t-1})p(x_t | s_t)
  \]

  \[
  p(x_1, \ldots, x_T) = \sum_{s_1, \ldots, s_T} p(s_1, \ldots, s_T, x_1, \ldots, x_T)
  \]
Learning of Bayesian Networks

- structure learning: an unsolved open problem
- parameter estimation
  - a Bayesian network: \( p_\theta(x_1, x_2, x_3, \cdots) \)
  - MLE using full data observations: simple

\[
\{ (x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \cdots), (x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \cdots), \cdots (x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \cdots), \cdots \}
\]

\[
l(\theta) = \sum_i \ln p_\theta(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \cdots)
\]

- MLE using partially observed data: requires EM method

\[
\{ (x_1^{(1)}, *, x_3^{(1)}, \cdots), (x_1^{(2)}, *, x_3^{(2)}, \cdots), \cdots, (x_1^{(i)}, *, x_3^{(i)}, \cdots), \cdots \}
\]

\[
l(\theta) = \sum_i \ln \sum_{x_2} p_\theta(x_1^{(1)}, x_2, x_3^{(1)}, \cdots)
\]
Bayesian Networks: Inference Problem

- Inference problem: infer any conditional distribution using BNs

\[ p(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, \cdots) \]
\[
\text{observed } x \quad \text{interested } y \quad \text{missing } z
\]

\[ p(y \mid x) = \frac{p(x, y)}{p(x)} = \frac{\sum_z p(x, y, z)}{\sum_{y,z} p(x, y, z)} \]

- The key is how to compute any summation efficiently.

- Efficiency depends on the network structure:
  - Sparser networks \( \implies \) more efficient inference methods.
  - Densely structured networks are generally hard to infer.
### Bayesian Networks: Inference Algorithms

<table>
<thead>
<tr>
<th>inference algorithm</th>
<th>applicable graphs</th>
<th>complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>brute-force</td>
<td>all</td>
<td>$O(K^T)$</td>
</tr>
<tr>
<td>Forward-Backward</td>
<td>chain</td>
<td>$O(T \cdot K^2)$</td>
</tr>
<tr>
<td>Sum-Product (Belief Propagation)</td>
<td>tree</td>
<td>$O(T \cdot K^2)$</td>
</tr>
<tr>
<td>Max-Sum</td>
<td>tree</td>
<td>$O(T \cdot K^2)$</td>
</tr>
<tr>
<td>Junction Tree</td>
<td>all</td>
<td>$O(K^p)$</td>
</tr>
</tbody>
</table>

#### Exact Inference

- Loopy Belief Propagation: $O(T \cdot K^2)$
- Variational Inference: $O(T \cdot K^2)$
- Expectation Propagation: $O(T \cdot K^2)$
- Monte Carlo Sampling: $O(T \cdot K^2)$

#### Approximate Inference

- Loopy Belief Propagation: $O(T \cdot K^2)$
- Variational Inference: $O(T \cdot K^2)$
- Expectation Propagation: $O(T \cdot K^2)$
- Monte Carlo Sampling: $O(T \cdot K^2)$
Forward-Backward: Message-Passing on a Chain (I)

\[ p(x_1, x_2, \cdots, x_T) = p(x_1)p(x_2|x_1) \cdots p(x_n|x_{n-1}) \cdots p(x_T|x_{T-1}) \]

consider any marginal distribution \( p(x_n) \)

\[
p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_T} p(x_1)p(x_2|x_1)p(x_3|x_2) \cdots p(x_T|x_{T-1})
\]

\[
= \left( \sum_{x_1 \cdots x_{n-1}} p(x_1) \cdots p(x_n|x_{n-1}) \right) \left( \sum_{x_{n+1} \cdots x_T} p(x_{n+1}|x_n) \cdots p(x_T|x_{T-1}) \right)
\]
Forward-Backward: Message-Passing on a Chain (II)

\[ p(x_n) = \left( \sum_{x_{n-1}} p(x_n|x_{n-1}) \right) \cdots \left( \sum_{x_2} p(x_3|x_2) \left( \sum_{x_1} p(x_1)p(x_2|x_1) \right) \right) \]

\[ \left( \sum_{x_{n+1}} p(x_{n+1}|x_n) \right) \cdots \left( \sum_{x_{T-1}} p(x_T|x_{T-1}) \left( \sum_{x_T} p(x_T|x_{T-1}) \right) \right) \]
summations are computed recursively \(\implies\) message passing

extended to any other marginal distributions:

- for any unobserved variable \(x_{t-1}\) or \(x_{t+1}\)
  \[
  \alpha_t(x_t) = \sum_{x_{t-1}} p(x_t|x_{t-1}) \alpha_{t-1}(x_{t-1}) \\
  \beta_t(x_t) = \sum_{x_{t+1}} p(x_{t+1}|x_t) \beta_{t+1}(x_{t+1})
  \]

- for an observed variable \(x_t\)
  \[
  \alpha_{t+1}(x_{t+1}) = p(x_{t+1}|x_t) \alpha_t(x_t) \bigg| x_t = \omega_k \\
  \beta_{t-1}(x_{t-1}) = p(x_t|x_{t-1}) \beta_t(x_t) \bigg| x_t = \omega_k
  \]
Monte Carlo sampling for $p(x_6, x_7 \mid \hat{x}_1, \hat{x}_3, \hat{x}_5)$

1. $\mathcal{D} = \emptyset$; $n = 0$
2. \textbf{while} $n < N$ \textbf{do}
   1. sampling $\hat{x}_2^{(n)} \sim p(x_2)$
   2. sampling $\hat{x}_4^{(n)} \sim p(x_4 \mid \hat{x}_1, \hat{x}_2^{(n)}, \hat{x}_3)$
   3. sampling $\hat{x}_6^{(n)} \sim p(x_6 \mid \hat{x}_4^{(n)})$
   4. sampling $\hat{x}_7^{(n)} \sim p(x_7 \mid \hat{x}_4^{(n)}, \hat{x}_5)$
   5. $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\hat{x}_6^{(n)}, \hat{x}_7^{(n)})\}$
   6. $n = n + 1$
\textbf{end while}
Case Study (I): Naive Bayes Classifier

- naive Bayes assumption: all features are conditionally independent given the class label
- naive Bayes classifiers: the simplest BN structure

\[ p(y, x_1, x_2, \ldots, x_d) = p(y)p(x_1|y)p(x_2|y) \cdots p(x_d|y) \]
\[ = p(y) \prod_{i=1}^{d} p(x_i|y) \]

- train each \( p(x_i|y) \) separately
- inference is also simple

\[ y^* = \arg \max_y p(y|x_1, x_2, \ldots, x_d) = \arg \max_y p(y) \prod_{i=1}^{d} p(x_i|y) \]
Case Study (II): Latent Dirichlet Allocation (1)

- topic modeling for text documents
  - a document mentions a few topics
  - each topic is described by a unique distribution of words

- latent Dirichlet allocation (LDA):
  - for each document, sample a topic distribution (multinomial)
    \[ \theta_i \sim p(\theta) = \text{Dir}(\theta | \alpha) \]
  - for \( j \)-th location in \( i \)-th document
    1. sample a topic \( z_{ij} \):
      \[ z_{ij} \sim p(z | \theta_i) = \text{Mult}(z | \theta_i) \]
    2. sample a word from
      \[ w_{ij} \sim \prod_{k=1}^{K} \left( \text{Mult}(w_{ij} | \beta_k) \right)^{z_{ijk}} \]

words labelled by the same color come from the same topic
Case Study (II): Latent Dirichlet Allocation (2)

- latent Dirichlet allocation (LDA):
  - topic distributions as \( \Theta = \{ \theta_i \mid 1 \leq i \leq M \} \)
  - all words in all documents as \( W = \{ w_{ij} \mid 1 \leq i \leq M; 1 \leq j \leq N_i \} \)
  - all sampled topics as \( Z = \{ z_{ij} \mid 1 \leq i \leq M; 1 \leq j \leq N_i \} \)

\[
p(\Theta, Z, W) = \prod_{i=1}^{M} p(\theta_i) \prod_{j=1}^{N_i} p(z_{ij} \mid \theta_i) p(w_{ij} \mid z_{ij})
\]

\[
p(\theta_i) = \text{Dir}(\theta_i \mid \alpha)
\]

\[
p(z_{ij} \mid \theta_i) = \text{Mult}(z_{ij} \mid \theta_i)
\]

\[
p(w_{ij} \mid z_{ij}) = \prod_{k=1}^{K} \left( \text{Mult}(w_{ij} \mid \beta_k) \right)^{z_{ijk}}
\]

LDA parameters:
- \( \alpha \in \mathbb{R}^K \)
- \( \beta \in \mathbb{R}^{K \times V} \)
Case Study (II): Latent Dirichlet Allocation (3)

- training problem: maximize the likelihood function

\[
p(W; \alpha, \beta) = \int \int \int_{\theta_1 \ldots \theta_M} \prod_{i=1}^{M} p(\theta_i) \prod_{j=1}^{N_i} \sum_{z_{ij}} p(z_{ij} | \theta_i) p(w_{ij} | z_{ij}) \, d\theta_1 \ldots d\theta_M
\]

- inference problem:

\[
p(\Theta, Z | W) = \frac{p(\Theta, Z, W)}{p(W)} = \frac{p(\Theta, Z, W)}{\int \int \int_{\Theta} \sum_{Z} p(\Theta, Z, W) \, d\Theta}
\]

- both are computationally intractable
- approximation by a variational distribution

\[
p(\Theta, Z | W) \approx q(\Theta, Z) = \prod_{i=1}^{M} q(\theta_i | \gamma) \prod_{j=1}^{N_i} q(z_{ij} | \phi_{ij})
\]

- learn \(\alpha\) and \(\beta\) by maximizing a variational lower-bound of

\[
p(W; \alpha, \beta)
\]
Markov Random Fields: Maximum Cliques

- **Markov random fields**: use undirected graphs
  - a node $\Rightarrow$ a random variable
  - a undirected link $\neq$ a conditional distribution

- **Conditional independence** $\iff$ topological connection
  - e.g. $A \perp B \mid C$

- **Cliques**: any set of fully-connected nodes
  - $\{x_1, x_2\}$, $\{x_1, x_2, x_3\}$, $\{x_2, x_4\}$, etc.

- **Maximum cliques**: not contained by another clique
  - $\{x_1, x_2, x_3\}$, $\{x_2, x_4\}$, $\{x_4, x_5\}$, $\{x_6, x_7\}$
Markov Random Fields: Potential and Partition Functions

- potential function $\psi(\cdot)$: any non-negative function defined over all variables in a maximum clique
  - use the exponential function: $\psi_c(x_c) = \exp(-E(x_c))$
  - $E(x_c)$ is called the energy function
  - exponential potential functions $\Rightarrow$ a Boltzmann distribution

- joint distribution of an MRF: a product of the potential functions of all maximum cliques, divided by a normalization term

$$p(x) = \frac{1}{Z} \prod_c \psi_c(x_c)$$

- $Z$ is called the partition function: $Z = \sum_x \prod_c \psi_c(x_c)$
- MRFs are hard to learn due to the intractable partition function
- all BN inference algorithms are equally applicable to MRFs
Case Study (III): Conditional Random Fields

- Conditional random fields (CRFs) define a conditional distribution between two sets of random variables

\[
p(Y \mid X) = \frac{\prod_c \psi_C(Y_c, X)}{\sum_Y \prod_c \psi_C(Y_c, X)}
\]

- Each potential function is defined on \(X\) and a maximum clique of \(Y\)

- Linear-chain CRFs: all \(Y\) nodes form a chain

\[
p(Y \mid X) = \frac{\prod_{t=1}^{T-1} \psi(y_t, y_{t+1}, X)}{\sum_Y \prod_{t=1}^{T-1} \psi(y_t, y_{t+1}, X)}
\]

\[
\psi(y_t, y_{t+1}, X) = \exp \left( \sum_{k=1}^{K} w_k \cdot f_k(y_t, y_{t+1}, X) \right)
\]
Case Study (IV): Restricted Boltzmann Machines (1)

- restricted Boltzmann machines (RBMs) define a joint distribution of some bipartite random variables
  - visible variables: \( v_i \in \{0, 1\} \ (1 \leq i \leq I) \)
  - hidden variables: \( h_j \in \{0, 1\} \ (1 \leq j \leq J) \)
- maximum cliques: any \( \{v_i, h_j\} \ (\forall i, j) \)
- potential functions:
  \[
  \psi(v_i, h_j) = \exp(a_i v_i + b_j h_j + w_{ij} v_i h_j)
  \]
- the joint distribution:
  \[
  p(v_1, \cdots, v_I, h_1, \cdots, h_J) = \frac{1}{Z} \exp\left( \sum_{i=1}^{I} a_i v_i + \sum_{j=1}^{J} b_j h_j + \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ij} v_i h_j \right)
  \]
Case Study (IV): Restricted Boltzmann Machines (2)

\[
\begin{align*}
    &a = \begin{bmatrix} a_1 \\ \vdots \\ a_I \end{bmatrix} & b = \begin{bmatrix} b_1 \\ \vdots \\ b_J \end{bmatrix} & v = \begin{bmatrix} v_1 \\ \vdots \\ v_I \end{bmatrix} & h = \begin{bmatrix} h_1 \\ \vdots \\ h_J \end{bmatrix} & W = \begin{bmatrix} w_{ij} \end{bmatrix}_{I \times J}
\end{align*}
\]

- RBM in matrix form: \( p(v, h) = \frac{1}{Z} \exp \left( a^\top v + b^\top h + v^\top Wh \right) \)
- Conditional independence in RBMs:
  \[
  p(h \mid v) = \prod_{j=1}^{J} p(h_j \mid v) \quad p(v \mid h) = \prod_{i=1}^{I} p(v_i \mid h)
  \]
  where \( \Pr(h_j = 1 \mid v) \) and \( \Pr(v_i = 1 \mid h) \) are sigmoids.
- Learning RBMs is not trivial due to the partition function; needs sampling methods

\[
\arg \max_{a, b, W} \prod_{v \in \mathcal{D}} p(v) = \arg \max_{a, b, W} \prod_{v \in \mathcal{D}} \frac{p(v, h)}{\sum_h p(v, h)}
\]