Chapter 15
Graphical Models

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

August 2020

CAMBRIDGE UNIVERSITY PRESS

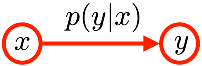
Outline

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

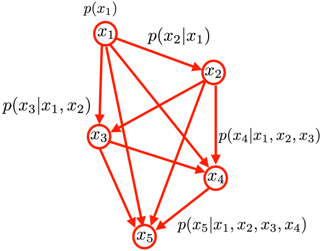


- a Bayesian network (BN) represents a way to factorize a joint distribution of all underlying random variable

$$p(x_1, x_2, \cdots, x_N) = \prod_{i=1}^{N} p\big(x_i \,|\, \textbf{pa}(x_i)\big)$$

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

$$\begin{aligned} & p(x_1, x_2, x_3, x_4, x_5) \\ = \; & p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdot \\ & p(x_4|x_1, x_2, x_3) \cdot p(x_5|x_1, x_2, x_3, x_4) \end{aligned}$$

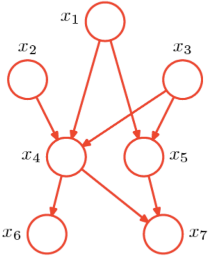


Bayesian Networks: A Sparse Example

- why use graphical representations for joint distributions?
- a sparse graph indicates some conditional independence among variables

$$
\begin{aligned}
& p(x_1, x_2, x_3, x_4, x_5, x_6, x_7) \\
= \; & p(x_1)\, p(x_2)\, p(x_3)\, p(x_4|x_1, x_2, x_3) \\
& p(x_5|x_1, x_3)\, p(x_6|x_4)\, p(x_7|x_4, x_5)
\end{aligned}
$$

- if each conditional distribution is chosen from e-family, a Bayesian network represents another distribution in e-family



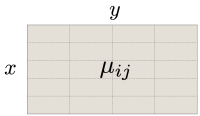
Bayesian Networks for Discrete Random Variables

- all nodes represent discrete random variables
- an M-value random variable x is encoded as a 1-of-M vector $\mathbf{x} = \begin{bmatrix} x_1 \; x_2 \; \cdots \; x_M \end{bmatrix}^\intercal$

$$\begin{bmatrix} 1 \; 0 \; \cdots \; 0 \end{bmatrix}, \begin{bmatrix} 0 \; 1 \; \cdots \; 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \; 0 \; \cdots \; 1 \end{bmatrix}$$

- conditional probabilities are stored as tables

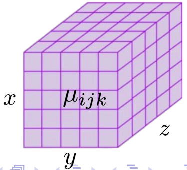
$$\mu_{ij} \triangleq \Pr(x = i \mid y = j) = \Pr(x_i = 1 \mid y_j = 1)$$



- notations for conditional distributions

$$p(x \mid y) = p(\mathbf{x} \mid \mathbf{y}) = \prod_{i=1}^{M} \prod_{j=1}^{N} \mu_{ij}^{x_i y_j}$$

$$p(x \mid y, z) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = \prod_{i=1}^{M} \prod_{j=1}^{N} \prod_{k=1}^{K} \mu_{ijk}^{x_i y_j z_k}$$



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*

$$x \perp y \,\Big|\, z \iff p(x,y\,|\,z) = p(x|z)\,p(y|z)$$

- 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*

$$p(x, y, z) = p(z) \cdot p(x|z) \cdot p(y|z)$$

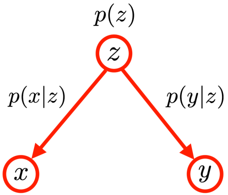
- unconditionally dependent: a confounder introduces spurious association

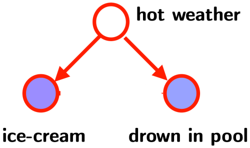
$$x \not\perp y \quad \Longleftrightarrow \quad p(x, y) \neq p(x)\, p(y)$$

- conditionally independent:

$$x \perp y \,\big|\, z \quad \Longleftrightarrow \quad p(x, y \,|\, z) = p(x|z)\, p(y|z)$$

- the ice-cream example
  - eating ice-cream $\Longrightarrow$ drowning in pool?
  - confounding $\neq$ causation





Chain

- a *chain* junction pattern $x \to z \to y$, where z is called a *mediator*

$$p(x,y,z) = p(x) \cdot p(z|x) \cdot p(y|z)$$

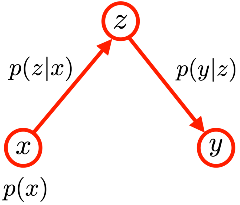
- unconditionally dependent: a mediator introduces spurious association

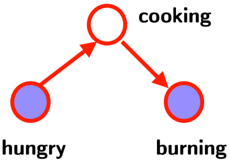
$$x \not\perp y \quad \Longleftrightarrow \quad p(x,y) \neq p(x)\,p(y)$$

- conditionally independent:

$$x \perp y \,\big|\, z \quad \Longleftrightarrow \quad p(x,y\,|\,z) = p(x|z)\,p(y|z)$$

- the cooking example
  - hungry $\Longrightarrow$ burning fingers?
  - mediating $\neq$ causation





Colliding

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

$$p(x,y,z) = p(x) \cdot p(y) \cdot p(z \,|\, x, y)$$

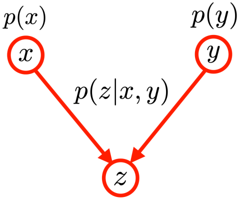
- unconditionally independent:

$$x \perp y \quad \Longleftrightarrow \quad p(x,y) = p(x)p(y)$$

- conditionally dependent:

$$x \not\perp y \,\big|\, z \quad \Longleftrightarrow \quad p(x,y\,|\,z) \neq p(x|z)p(y|z)$$

- the explain-away phenomenon

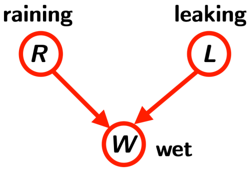


Colliding Causes Explain-away

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

$$\Pr(R = 1 \,|\, W = 1) = \mathbf{0.3048}$$

$$\Pr(R = 1 \,|\, W = 1, L = 1) = \mathbf{0.1667}$$



$$\Pr(R = 1) = 0.1$$

$$\Pr(L = 1) = 0.01$$

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

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

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

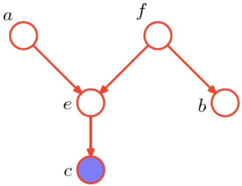
$$\Pr(W = 1 \,|\, R = 0, L = 0) = 0.20$$

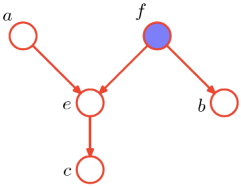
Conditional Independence: d-separation Rule

- 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:

$$a \not\perp f \mid c \qquad a \not\perp b \mid c$$

$$a \not\perp c \mid f \qquad a \perp b \mid f \qquad e \perp b \mid f$$

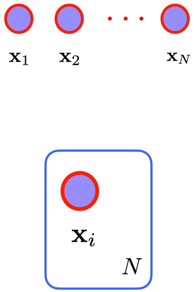




Bayesian Networks: Representing Generative Models (I)

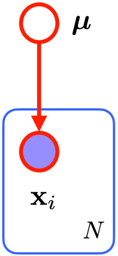
- Gaussian models:

$$p(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i \,|\, \boldsymbol{\mu}, \Sigma)$$



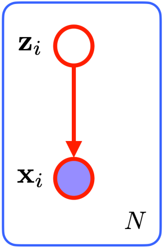
- Bayesian learning of Gaussian models:

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



Bayesian Networks: Representing Generative Models (II)

- Gaussian mixture models (GMMs):
  - latent variable: the index is encoded as a 1-of-M vector $\mathbf{z}_i = \begin{bmatrix} z_{i1} & z_{i2} & \cdots & z_{iM} \end{bmatrix}$
  - $p(\mathbf{z}_i) = \prod_{m=1}^{M} \left(w_m\right)^{z_{im}}$
  - $p(\mathbf{x}_i \,|\, \mathbf{z}_i) = \prod_{m=1}^{M} \Big(\mathcal{N}(\mathbf{x}_i \,|\, \boldsymbol{\mu}_m, \Sigma_m)\Big)^{z_{im}}$



- 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:

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

Bayesian Networks: Representing Generative Models (III)

- Bayesian learning of Gaussian mixture models (GMMs):

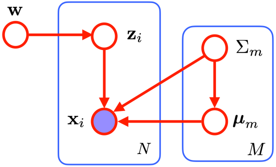
$$p(\mathbf{w}) = \mathsf{Dir}\big(\mathbf{w} \,\big|\, \boldsymbol{\alpha}^{(0)}\big)$$

$$p(\mathbf{z}_i \,|\, \mathbf{w}) = \prod_{m=1}^{M} \big(w_m\big)^{z_{im}} \quad \forall i = 1, 2, \cdots N$$

$$p\big(\Sigma_m\big) = \mathcal{W}^{-1}\big(\Sigma_m \,\big|\, \Phi_m^{(0)}, \nu_m^{(0)}\big) \quad \forall m = 1, 2, \cdots M$$

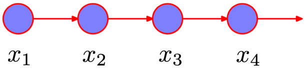
$$p(\boldsymbol{\mu}_m \,|\, \Sigma_m) = \mathcal{N}\Big(\boldsymbol{\mu}_m \,\Big|\, \boldsymbol{\nu}_m^{(0)}, \frac{1}{\lambda_m^{(0)}}\Sigma_m\Big) \quad \forall m = 1, 2, \cdots M$$

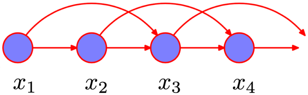
$$p\big(\mathbf{x}_i \,\big|\, \mathbf{z}_i, \{\boldsymbol{\mu}_m, \Sigma_m\}\big) = \prod_{m=1}^{M} \Big(\mathcal{N}(\mathbf{x}_i \,|\, \boldsymbol{\mu}_m, \Sigma_m)\Big)^{z_{im}} \quad \forall i = 1, 2, \cdots N$$



Bayesian Networks: Representing Generative Models (IV)

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



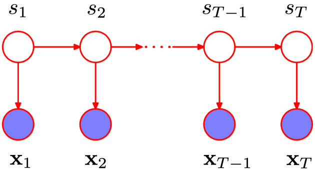


- hidden Markov models (HMMs)

$$p(s_1, \cdots, s_T, \mathbf{x}_1, \cdots, \mathbf{x}_T)$$

$$= p(s_1)p(\mathbf{x}_1|s_1) \prod_{t=2}^{T} p(s_t|s_{t-1})p(\mathbf{x}_t|s_t)$$

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_T) = \sum_{s_1, \cdots, s_T} p(s_1, \cdots, s_T, \mathbf{x}_1, \cdots, \mathbf{x}_T)$$



Learning of Bayesian Networks

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

$$\left\{\left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \cdots\right), \left(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \cdots\right), \cdots \left(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \cdots\right), \cdots\right\}$$

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

  - MLE using partially observed data: requires EM method

$$\left\{\left(x_1^{(1)}, *, x_3^{(1)}, \cdots\right), \left(x_1^{(2)}, *, x_3^{(2)}, \cdots\right), \cdots, \left(x_1^{(i)}, *, x_3^{(i)}, \cdots\right), \cdots\right\}$$

$$l(\boldsymbol{\theta}) = \sum_i \ln \sum_{x_2} p_{\boldsymbol{\theta}}\left(x_1^{(1)}, x_2, x_3^{(1)}, \cdots\right)$$

Bayesian Networks: Inference Problem

- inference problem: infer any conditional distribution using BNs

$$p\big(\underbrace{x_1, x_2, x_3}_{\text{observed } \mathbf{x}}, \underbrace{x_4, x_5, x_6}_{\text{interested } \mathbf{y}}, \underbrace{x_7, x_8, \cdots}_{\text{missing } \mathbf{z}}\big)$$

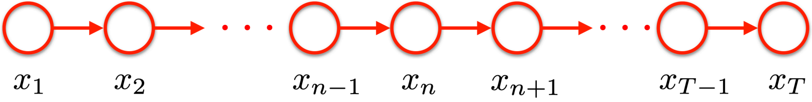
$$p(\mathbf{y} \,|\, \mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})} = \frac{\sum_{\mathbf{z}} \, p(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\sum_{\mathbf{y},\mathbf{z}} \, p(\mathbf{x}, \mathbf{y}, \mathbf{z})}$$

- the key is how to compute any summation efficiently
- efficiency depends on the network structure
  - sparser networks $\Longrightarrow$ more efficient inference methods
  - densely structured networks are generally hard to infer

Bayesian Networks: Inference Algorithms

	inference algorithm	applicable graphs	complexity
exact inference	brute-force	all	$O(K^T)$
	Forward-Backward	chain	$O(T \cdot K^2)$
	Sum-Product (Belief Propagation)	tree	$O(T \cdot K^2)$
	Max-Sum	tree	$O(T \cdot K^2)$
	Junction Tree	all	$O(K^p)$
approximate inference	Loopy Belief Propagation	all	-
	Variational Inference	all	-
	Expectation Propagation	all	-
	Monte Carlo Sampling	all	-

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 $p(x_n)$

$$\begin{aligned} p(x_n) &= \sum_{x_1}\cdots\sum_{x_{n-1}}\sum_{x_{n+1}}\cdots\sum_{x_T} p(x_1)p(x_2|x_1)p(x_3|x_2)\cdots p(x_T|x_{T-1}) \\ &= \Bigg(\sum_{x_1\cdots x_{n-1}} p(x_1)\cdots p(x_n|x_{n-1})\Bigg)\Bigg(\sum_{x_{n+1}\cdots x_T} p(x_{n+1}|x_n)\cdots p(x_T|x_{T-1})\Bigg) \end{aligned}$$

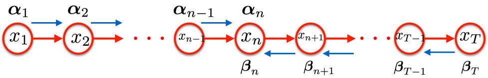
Forward-Backward: Message-Passing on a Chain (II)

$$p(x_n) = \overbrace{\left(\sum_{x_{n-1}} p(x_n|x_{n-1}) \cdots \overbrace{\left(\sum_{x_2} p(x_3|x_2) \overbrace{\left(\sum_{x_1} \overbrace{p(x_1)}^{\alpha_1(x_1)} p(x_2|x_1)\right)}^{\alpha_2(x_2)}\right)}^{\alpha_3(x_3)}\right)}^{\alpha_n(x_n)}$$

$$\underbrace{\left(\sum_{x_{n+1}} p(x_{n+1}|x_n) \cdots \underbrace{\left(\sum_{x_{T-1}} p(x_{T-1}|x_{T-2}) \underbrace{\left(\sum_{x_T} p(x_T|x_{T-1})\right)}_{\beta_{T-1}(x_{T-1})}\right)}_{\beta_{T-2}(x_{T-2})}\right)}_{\beta_n(x_n)}$$

Forward-Backward: Message-Passing on a Chain (III)

summations are computed recursively $\Longrightarrow$ message passing



extended to any other marginal distributions:

- for any unobserved variable x_{t-1} or x_{t+1}

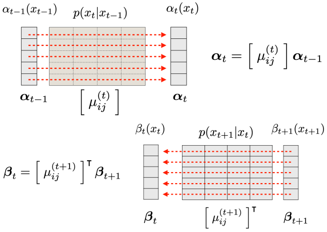
$$\alpha_t(x_t) = \sum_{x_{t-1}} p(x_t|x_{t-1})\,\alpha_{t-1}(x_{t-1})$$

$$\beta_t(x_t) = \sum_{x_{t+1}} p(x_{t+1}|x_t)\,\beta_{t+1}(x_{t+1})$$

- for an observed variable x_t

$$\alpha_{t+1}(x_{t+1}) = p(x_{t+1}|x_t)\alpha_t(x_t)\Big|_{x_t=\omega_k}$$

$$\beta_{t-1}(x_{t-1}) = p(x_t|x_{t-1})\beta_t(x_t)\Big|_{x_t=\omega_k}$$



Monte Carlo Sampling

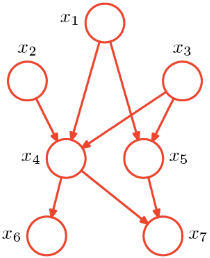
Monte Carlo sampling for $p(x_6, x_7 \,|\, \hat{x}_1, \hat{x}_3, \hat{x}_5)$

$\mathcal{D} = \emptyset$; $n = 0$

while $n < N$ **do**

1. sampling $\hat{x}_2^{(n)} \sim p(x_2)$
2. sampling $\hat{x}_4^{(n)} \sim p(x_4 \,|\, \hat{x}_1, \hat{x}_2^{(n)}, \hat{x}_3)$
3. sampling $\hat{x}_6^{(n)} \sim p(x_6 \,|\, \hat{x}_4^{(n)})$
4. sampling $\hat{x}_7^{(n)} \sim p(x_7 \,|\, \hat{x}_4^{(n)}, \hat{x}_5)$
5. $\mathcal{D} \Leftarrow \mathcal{D} \cup \{(\hat{x}_6^{(n)}, \hat{x}_7^{(n)})\}$
6. $n = n + 1$

end while

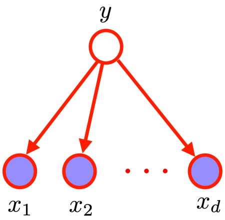


$p(x_1)$, $p(x_2)$, $p(x_3)$
$p(x_4|x_1, x_2, x_3)$
$p(x_5|x_1, x_3)$, $p(x_6|x_4)$
$p(x_7|x_4, x_5)$

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

$$\begin{aligned} p(y, x_1, x_2, \cdots, x_d) &= p(y)p(x_1|y)p(x_2|y)\cdots p(x_d|y) \\ &= p(y)\prod_{i=1}^{d} p(x_i|y) \end{aligned}$$



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

$$y^* = \arg\max_y \; p(y|x_1, x_2, \cdots, x_d) = \arg\max_y \; p(y)\prod_{i=1}^{d} p(x_i|y)$$

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)

$$\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}) = \mathrm{Dir}(\boldsymbol{\theta} \,|\, \boldsymbol{\alpha})$$

  - for j-th location in i-th document
    1. sample a topic $\mathbf{z}_{ij}$:

$$\mathbf{z}_{ij} \sim p(\mathbf{z} \,|\, \boldsymbol{\theta}_i) = \mathrm{Mult}(\mathbf{z} \,|\, \boldsymbol{\theta}_i)$$

    2. sample a word from

$$\mathbf{w}_{ij} \sim \prod_{k=1}^{K} \Big(\mathrm{Mult}\big(\mathbf{w}_{ij} \,|\, \boldsymbol{\beta}_k\big)\Big)^{z_{ijk}}$$

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

words labelled by the same color come from the same topic

Case Study (II): Latent Dirichlet Allocation (2)

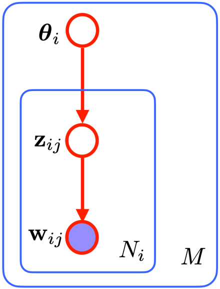
- latent Dirichlet allocation (LDA):
  - topic distributions as $\mathbf{\Theta} = \{\boldsymbol{\theta}_i \,|\, 1 \leq i \leq M\}$
  - all words in all documents as $\mathbf{W} = \{\mathbf{w}_{ij} \,|\, 1 \leq i \leq M; 1 \leq j \leq N_i\}$
  - all sampled topics as $\mathbf{Z} = \{\mathbf{z}_{ij} \,|\, 1 \leq i \leq M; 1 \leq j \leq N_i\}$

$$p(\mathbf{\Theta}, \mathbf{Z}, \mathbf{W}) = \prod_{i=1}^{M} p(\boldsymbol{\theta}_i) \prod_{j=1}^{N_i} p(\mathbf{z}_{ij} \,|\, \boldsymbol{\theta}_i)\, p(\mathbf{w}_{ij} \,|\, \mathbf{z}_{ij})$$

$$p(\boldsymbol{\theta}_i) = \mathsf{Dir}(\boldsymbol{\theta}_i \,|\, \boldsymbol{\alpha})$$

$$p(\mathbf{z}_{ij} \,|\, \boldsymbol{\theta}_i) = \mathsf{Mult}(\mathbf{z}_{ij} \,|\, \boldsymbol{\theta}_i)$$

$$p(\mathbf{w}_{ij} \,|\, \mathbf{z}_{ij}) = \prod_{k=1}^{K} \Big(\mathsf{Mult}\big(\mathbf{w}_{ij} \,|\, \boldsymbol{\beta}_k\big)\Big)^{z_{ijk}}$$



LDA parameters:

- $\boldsymbol{\alpha} \in \mathbb{R}^K$
- $\boldsymbol{\beta} \in \mathbb{R}^{K \times V}$

Case Study (II): Latent Dirichlet Allocation (3)

- training problem: maximize the likelihood function

$$p(\mathbf{W}\,;\,\boldsymbol{\alpha},\boldsymbol{\beta}) = \iiint_{\boldsymbol{\theta}_1\cdots\boldsymbol{\theta}_M} \prod_{i=1}^{M} p(\boldsymbol{\theta}_i) \prod_{j=1}^{N_i} \sum_{\mathbf{z}_{ij}} p(\mathbf{z}_{ij}\,|\,\boldsymbol{\theta}_i)\, p(\mathbf{w}_{ij}\,|\,\mathbf{z}_{ij})\, d\boldsymbol{\theta}_1 \cdots d\boldsymbol{\theta}_M$$

- inference problem:

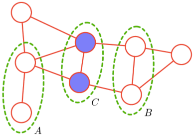
$$p(\boldsymbol{\Theta},\mathbf{Z}\,|\,\mathbf{W}) = \frac{p(\boldsymbol{\Theta},\mathbf{Z},\mathbf{W})}{p(\mathbf{W})} = \frac{p(\boldsymbol{\Theta},\mathbf{Z},\mathbf{W})}{\iiint_{\boldsymbol{\Theta}} \sum_{\mathbf{Z}} p(\boldsymbol{\Theta},\mathbf{Z},\mathbf{W})\, d\boldsymbol{\Theta}}$$

- both are computationally intractable
- approximation by a variational distribution

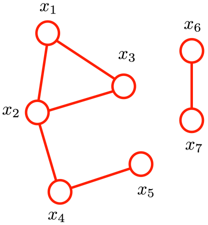
$$p(\boldsymbol{\Theta},\mathbf{Z}\,|\,\mathbf{W}) \approx q(\boldsymbol{\Theta},\mathbf{Z}) = \prod_{i=1}^{M} q(\boldsymbol{\theta}_i\,|\,\boldsymbol{\gamma}) \prod_{j=1}^{N_i} q(\mathbf{z}_{ij}\,|\,\boldsymbol{\phi}_{ij})$$

- learn $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by maximizing a variational lower-bound of $p(\mathbf{W}\,;\,\boldsymbol{\alpha},\boldsymbol{\beta})$

Markov Random Fields: Maximum Cliques



- Markov random fields: use undirected graphs
  - a node $\Longrightarrow$ a random variable
  - a undirected link $\neq$ a conditional distribution
- conditional independence $\Longleftrightarrow$ topological connection
  - e.g. $A \perp B \,\big|\, C$
- 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 variable in a maximum clique
  - use the exponential function: $\psi_c(\mathbf{x}_c) = \exp\big(-E(\mathbf{x}_c)\big)$
  - $E(\mathbf{x}_c)$ is called the energy function
  - exponential potential functions $\implies$ a Boltzmann distribution
- joint distribution of an MRF: a product of the potential functions of all maximum cliques, divided by a normalization term

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

- Z is called the partition function: $Z = \sum_{\mathbf{x}} \prod_c \psi_c(\mathbf{x}_c)$
- 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

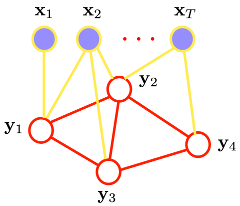
$$p(\mathbf{Y} \,|\, \mathbf{X}) = \frac{\prod_c \psi_C(\mathbf{Y}_c, \mathbf{X})}{\sum_{\mathbf{Y}} \prod_c \psi_C(\mathbf{Y}_c, \mathbf{X})}$$

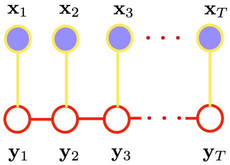
- each potential function is defined on $\mathbf{X}$ and a maximum clique of $\mathbf{Y}$

- linear-chain CRFs: all $\mathbf{Y}$ nodes form a chain

$$p(\mathbf{Y} \,|\, \mathbf{X}) = \frac{\prod_{t=1}^{T-1} \psi(\mathbf{y}_t, \mathbf{y}_{t+1}, \mathbf{X})}{\sum_{\mathbf{Y}} \prod_{t=1}^{T-1} \psi(\mathbf{y}_t, \mathbf{y}_{t+1}, \mathbf{X})}$$

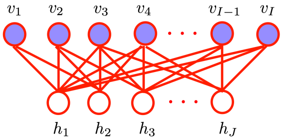
$$\psi(\mathbf{y}_t, \mathbf{y}_{t+1}, \mathbf{X}) = \exp\left(\sum_{k=1}^{K} w_k \cdot f_k(\mathbf{y}_t, \mathbf{y}_{t+1}, \mathbf{X})\right)$$





Case Study (IV): Restricted Boltzmann Machines (1)

- restricted Boltzman 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:

$$\psi(v_i, h_j) = \exp\left(a_i v_i + b_j h_j + w_{ij} v_i h_j\right)$$

- 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)

$$\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_I \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_J \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_I \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} h_1 \\ \vdots \\ h_J \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} w_{ij} \end{bmatrix}_{I \times J}$$

- RBM in matrix form: $p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\left(\mathbf{a}^\intercal \mathbf{v} + \mathbf{b}^\intercal \mathbf{h} + \mathbf{v}^\intercal \mathbf{W} \mathbf{h}\right)$
- conditional independence in RBMs:

$$p(\mathbf{h} \,|\, \mathbf{v}) = \prod_{j=1}^{J} p(h_j \,|\, \mathbf{v}) \qquad p(\mathbf{v} \,|\, \mathbf{h}) = \prod_{i=1}^{I} p(v_i \,|\, \mathbf{h})$$

  where $\Pr(h_j = 1 \,|\, \mathbf{v})$ and $\Pr(v_i = 1 \,|\, \mathbf{h})$ are sigmoids.
- learning RBMs is not trivial due to the partition function; needs sampling methods

$$\arg\max_{\mathbf{a},\mathbf{b},\mathbf{W}} \prod_{\mathbf{v} \in \mathcal{D}} p(\mathbf{v}) = \arg\max_{\mathbf{a},\mathbf{b},\mathbf{W}} \prod_{\mathbf{v} \in \mathcal{D}} \frac{p(\mathbf{v}, \mathbf{h})}{\sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h})}$$