Graphical Models

Shan-Hung Wu
shwu@cs.nthu.edu.tw

Department of Computer Science,
National Tsing Hua University, Taiwan

NetDB-ML, Spring 2015

Introduction

- In ***graphical models***, we model a problem using a graph where
  - Each node represents a random variable
  - Each link expresses a probabilistic relationship between two nodes
    - Directed link: conditional dependency (forming a ***Bayesian network***)
    - Undirected: correlation (forming a ***Markov random field***, or ***Markov network***)
- Graphical models offer the following advantages:
  - Visualization of the probabilistic models and motivating new models
  - Insight into the probabilistic properties (e.g., conditional independence between any two groups of nodes)
  - Complex computation (required to perform inference/learning) that can be carried along the graph

Outline

1. Bayesian Networks
   - Definitions
   - Conditional Independence and D-Separation
   - Modeling Problems as Graphs
   - Common Tasks
2. Evaluating Continuous Marginals
3. Bayesian Estimation
4. Evaluating Discrete Marginals
   - Belief Propagation
   - Sampling
5. Latent Dirichlet Allocation
6. Markov Random Fields**

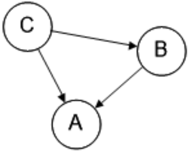
Outline

1. Bayesian Networks
   - Definitions
   - Conditional Independence and D-Separation
   - Modeling Problems as Graphs
   - Common Tasks
2. Evaluating Continuous Marginals
3. Bayesian Estimation
4. Evaluating Discrete Marginals
   - Belief Propagation
   - Sampling
5. Latent Dirichlet Allocation
6. Markov Random Fields**

Outline

Definitions (1/3)

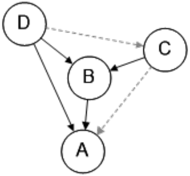
- Consider the joint probability $P(A = a, B = b, C = c)$ (or $P(A, B, C)$ ***for short***) of three random variables A, B, and C
- It can be factorized into, for example, $P(A|B, C)P(B|C)P(C)$
    - Holds for any distribution
- We can draw the factorization as a graph:



-
    - Each node is a random variable
    - A link denotes conditional dependency
- The graph must be a ***Directed Acyclic Graph*** (***DAG***) [Proof: by induction on the number of nodes]

Definitions (2/3)

- Given $P(X_1, X_2, \cdots, X_M)$ of M random variables, we have
  - Some factorization, e.g., $P(X_1, X_2, \cdots, X_M) = P(X_1|X_2, \cdots, X_M) \cdots P(X_M)$
  - A fully connected graph
- It is the ***missing links*** that convey interesting information



  - A missing link from D to C implies independence between D and C
    - $P(C|D) = P(C)$, denoted by $\{C\} \perp\!\!\!\perp \{D\}$ or $\{C\} \perp\!\!\!\perp \{D\} \,|\, \emptyset$
  - A missing link from C to A implies ***conditional independence*** between C and A given B and D
    - $P(A|B, C, D) = P(A|B, D)$, denoted by $\{A\} \perp\!\!\!\perp \{C\} \,|\, \{B, D\}$

Definitions (3/3)

- A graph visualizes a factorization:

$$P(X_1, X_2, \cdots, X_M) = \prod_{i=1}^{M} P(X_i | parent(X_i)),$$

  where $parent(X_i)$ is the values of the parent nodes of X_i
- One graph for each factorization
  - Given a set of variables, we may construct different graphs based on different factorizations

Extensions (1/2)

- Values of some random variables may be observed in our problem
  - E.g., we may only care about $P(B, C, \cdots | A)$ given an observed variable $A = a$
  - Denoted as solid nodes in the graph
- There can be deterministic variables
  - E.g., we may assume parameters (e.g., $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in classification, and $\boldsymbol{w}$ in regression) and hyperparameters (e.g., α and β in regression) to simplify calculation of a specific term in the factorization
  - Denoted by small dots in the graph
- Repeating subgraphs can be collapsed into a plate marked by multiplicity

Extensions (2/2)

- Observed variable $X = x$ vs deterministic variable α?

Extensions (2/2)

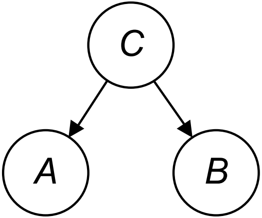
- Observed variable $X = x$ vs deterministic variable α?
- Even X is observed, $P(X = x) \neq 1$ if X has a nontrivial distribution
  - Can be in the consequent of a conditional probability
- $P(\alpha)$ is undefined
  - Can only be a parameter in a conditional probability
  - α cannot have parents
  - Must be observed
  - If α parametrizes $P(Y)$ (denoted by $P(Y) = P(Y;\alpha)$), then $P(X|Y;\alpha) = P(X|Y)$
    - Note, however, that $P(X;\alpha = c) \neq P(X;\alpha = c')$

Outline

Independence and Conditional Independence

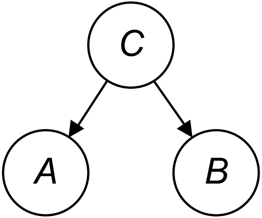
- $\{A\} \perp\!\!\!\perp \{B\} \mid \{C\}$ denotes conditional independence
  - $P(A|B,C) = P(A|C)$
  - Or equivalently, $P(A,B|C) = P(A|B,C)P(B|C) = P(A|C)P(B|C)$
- Many tasks are solved by the aid of conditional independence between nodes
- But checking conditional independence involving more than three nodes is usually cumbersome
- A graph visualizes the conditional independence and provides an easy way for checking
  - Given three sets of nodes P, Q, and R, you should be able to tell whether $P \perp\!\!\!\perp Q \mid R$ by directly looking at the graph

Canonical Cases (1/3)



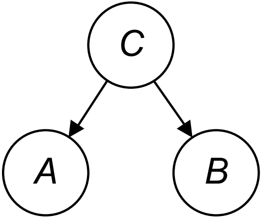
- Consider a tail-to-tail path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \emptyset$?

Canonical Cases (1/3)



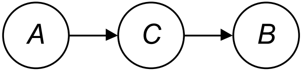
- Consider a tail-to-tail path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \emptyset$? No
  - $p(A,B) = \int p(A,B,C)dC = \int p(A|C)p(B|C)p(C)dC$, which does not equal to $p(A)p(B)$ for all distributions
- If C is observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \{C\}$?

Canonical Cases (1/3)



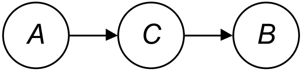
- Consider a tail-to-tail path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \emptyset$? No
  - $p(A,B) = \int p(A,B,C)dC = \int p(A|C)p(B|C)p(C)dC$, which does not equal to $p(A)p(B)$ for all distributions
- If C is observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \{C\}$? Yes
  - $p(A,B|C) = \frac{p(A,B,C)}{p(C)} = \frac{p(A|C)p(B|C)p(C)}{p(C)} = p(A|C)p(B|C)$
  - We say the path from A to B is ***blocked*** by C if C is observed

Canonical Cases (2/3)



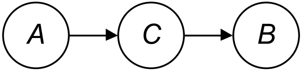
- Consider a head-to-tail path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \emptyset$?

Canonical Cases (2/3)



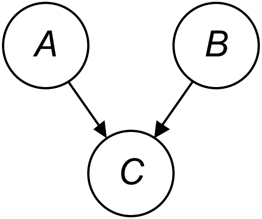
- Consider a head-to-tail path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \emptyset$? No
- If C is observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \{C\}$?

Canonical Cases (2/3)



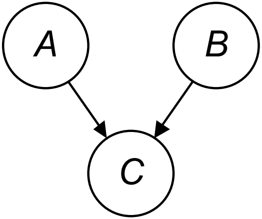
- Consider a head-to-tail path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \,|\, \emptyset$? No
- If C is observed
  - $\{A\} \perp\!\!\!\perp \{B\} \,|\, \{C\}$? Yes
  - $p(A,B|C) = \frac{p(A,B,C)}{p(C)} = \frac{p(B|C)p(C|A)p(A)}{p(C)} = p(B|C)p(A|C)$
  - The path from A to B is blocked by C if C is observed

Canonical Cases (3/3)



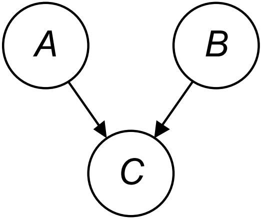
- Consider a head-to-head path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \emptyset$?

Canonical Cases (3/3)



- Consider a head-to-head path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \emptyset$? ***Yes***
  - $p(A,B) = \int p(A,B,C)dC = \int p(C|A,B)p(A)p(B)dC = p(A)p(B)\int p(C|A,B)dC = p(A)p(B)$
  - The path from A to B is blocked by C if C is not observed
- If C is observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \{C\}$?

Canonical Cases (3/3)

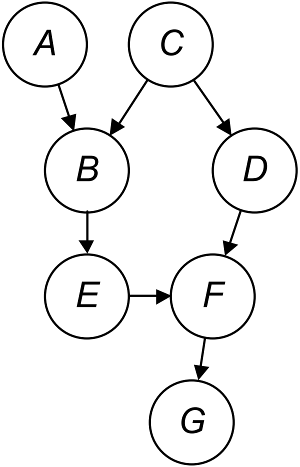


- Consider a head-to-head path at C
- If C is not observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \emptyset$? ***Yes***
  - $p(A,B) = \int p(A,B,C)dC = \int p(C|A,B)p(A)p(B)dC = p(A)p(B)\int p(C|A,B)dC = p(A)p(B)$
  - The path from A to B is blocked by C if C is not observed
- If C is observed
  - $\{A\} \perp\!\!\!\perp \{B\} \mid \{C\}$? ***No***
  - Actually, if C has descendents, A and B become dependent if any of the descendents is observed [Homework]

D-Separation (1/2)

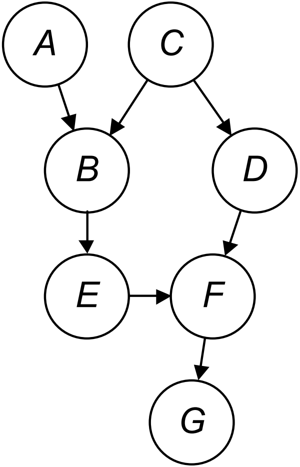
- Given three sets of non-intersecting random variables P, Q, and R, we say P is ***d-separated*** ("d" means "direct") from Q given R, denoted as $P \perp\!\!\!\perp Q \mid R$, iff all paths from P to Q are blocked
- A path (of arbitrary length) is blocked if either
  - There are two links meet head-to-tail or tail-to-tail at a node, and that node is in R, or
  - There are two links meet head-to-head at a node, and neither the node, nor its descendents, is in R
- Deterministic parameters play no role in d-separation
  - A parameter α must be observed and have no parent
  - Path passing through α must be tail-to-tail, so is blocked

D-Separation (2/2)



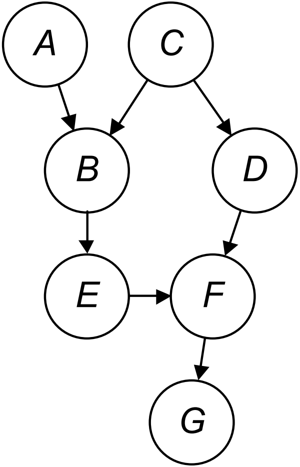
- $\{A\} \perp\!\!\!\perp \{C\} \mid \emptyset$?

D-Separation (2/2)



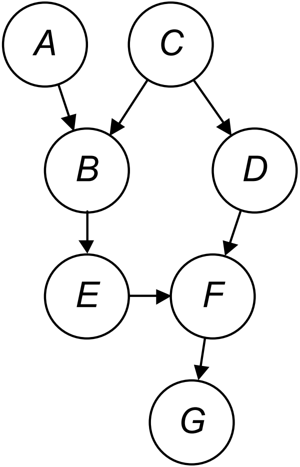
- $\{A\} \perp\!\!\!\perp \{C\} \mid \emptyset$? Yes
- $\{B\} \perp\!\!\!\perp \{D\} \mid \{C\}$?

D-Separation (2/2)



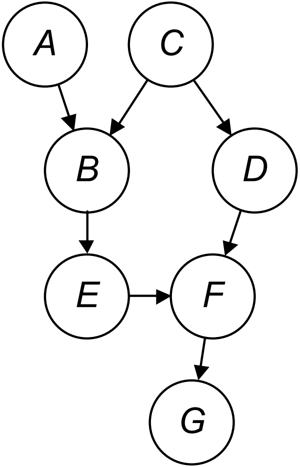
- $\{A\} \perp\!\!\!\perp \{C\} \mid \emptyset$? Yes
- $\{B\} \perp\!\!\!\perp \{D\} \mid \{C\}$? Yes
- $\{B\} \perp\!\!\!\perp \{D, F\} \mid \{C, E\}$?

D-Separation (2/2)



- $\{A\} \perp\!\!\!\perp \{C\} \mid \emptyset$? Yes
- $\{B\} \perp\!\!\!\perp \{D\} \mid \{C\}$? Yes
- $\{B\} \perp\!\!\!\perp \{D, F\} \mid \{C, E\}$? Yes
- $\{D\} \perp\!\!\!\perp \{E\} \mid \{C, G\}$?

D-Separation (2/2)



- $\{A\} \perp\!\!\!\perp \{C\} \mid \emptyset$? Yes
- $\{B\} \perp\!\!\!\perp \{D\} \mid \{C\}$? Yes
- $\{B\} \perp\!\!\!\perp \{D, F\} \mid \{C, E\}$? Yes
- $\{D\} \perp\!\!\!\perp \{E\} \mid \{C, G\}$? No

Outline

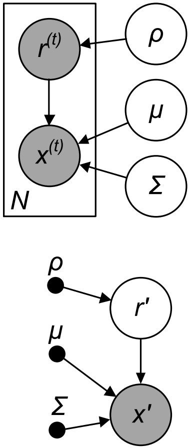
Modeling a Problem

- How to model a problem as a graph right (or, how determine the right factorization)?

Modeling a Problem

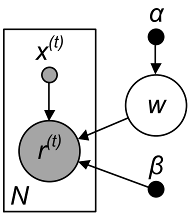
- How to model a problem as a graph right (or, how determine the right factorization)?
  1. Identify nodes
  2. For each node X, draw links from others $Y_1, Y_2, \cdots$ to X based on your assumptions of dependency
  3. Make sure
     - The network is connected
     - You did not add too many links that prevents the graph from being a DAG
- You ***should not*** invert the direction of a link just because you know how to use Bayes' rule

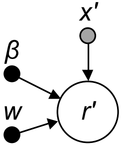
Example: Classification



- Model parameters $\boldsymbol{\rho} = \{\rho_i\}_{i=1}^K$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_i\}_{i=1}^K$, and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_i\}_{i=1}^K$ are deterministic variables
- Here we assume a ***generative model*** where an observation ($\boldsymbol{x}$) is the cause of some reasons ($\boldsymbol{r}$) that may not be observable
- Training: $(\rho, \boldsymbol{\mu}, \boldsymbol{\Sigma})_{MAP} = \arg_{\rho,\boldsymbol{\mu},\boldsymbol{\Sigma}} \max p(\rho, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathcal{X})$
- Prediction: $y' = \arg_y \max P(y | \boldsymbol{x}'; \boldsymbol{\rho}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$

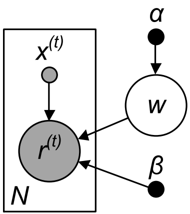
Example: Linear Regression (1/2)

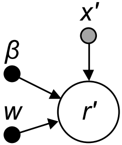




- Why don't we draw links from $r^{(t)}/r'$ to $\boldsymbol{x}^{(t)}/\boldsymbol{x}'$?

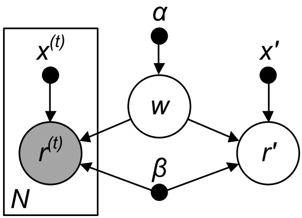
Example: Linear Regression (1/2)





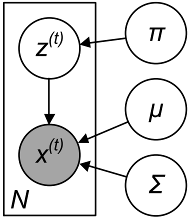
- Why don't we draw links from $r^{(t)}/r'$ to $\boldsymbol{x}^{(t)}/\boldsymbol{x}'$?
- Regression is ***not*** a generative model
  - We don't know how to evaluate $P(\boldsymbol{x}'|r',\cdots)$ given our assumptions
- Training: $\boldsymbol{w}_{MAP} = \arg_{\boldsymbol{w}} \max p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta)$
  - Recall that we may assume $p(\boldsymbol{w}) \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\boldsymbol{I})$
- Prediction: $y' = \arg_y \max p(y|\boldsymbol{x}', \boldsymbol{w}, \beta)$

Example: Linear Regression (2/2)



- $\boldsymbol{w}$ is a random variable in Bayesian estimation for r'
- Prediction:
  $y' = \arg_y \max p(y|\boldsymbol{x}', \mathcal{X}, \alpha, \beta) = \arg_y \max \int p(y, \boldsymbol{w}|\boldsymbol{x}', \mathcal{X}) d\boldsymbol{w}$
- There is no separate training phase

Example: Clustering



- $\boldsymbol{\pi} = \{\pi_i\}_{i=1}^K$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_i\}_{i=1}^K$, $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_i\}_{i=1}^K$
- Target: $(\{z^{(t)}\}_t, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})_{MAP} = \arg_{\{\boldsymbol{z}^{(t)}\}_t, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}} \max p(\{z^{(t)}\}_t, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathcal{X})$
  - $p(\{\boldsymbol{z}^{(t)}\}_t, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathcal{X}) = p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \{\boldsymbol{z}^{(t)}\}_t, \mathcal{X}) p(\{\boldsymbol{z}^{(t)}\}_t | \mathcal{X}) \propto p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \{\boldsymbol{z}^{(t)}\}_t, \mathcal{X}) p(\mathcal{X} | \{\boldsymbol{z}^{(t)}\}_t) p(\{\boldsymbol{z}^{(t)}\}_t)$
  - Can be simplified to $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \{\boldsymbol{z}^{(t)}\}_t, \mathcal{X}) p(\mathcal{X} | \{\boldsymbol{z}^{(t)}\}_t)$ is we have no preference on a particular $\{\boldsymbol{z}^{(t)}\}_t$ set
  - The problem is, we cannot evaluate $p(\mathcal{X} | \{\boldsymbol{z}^{(t)}\}_t)$ without knowing $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$
- E-step: treat $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ as parameters and estimate $\{z^{(t)}\}_t$
- M-step: treat $\{z^{(t)}\}_t$ as parameter and estimate $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$

Outline

1. Bayesian Networks
   - Definitions
   - Conditional Independence and D-Separation
   - Modeling Problems as Graphs
   - Common Tasks
2. Evaluating Continuous Marginals
3. Bayesian Estimation
4. Evaluating Discrete Marginals
   - Belief Propagation
   - Sampling
5. Latent Dirichlet Allocation
6. Markov Random Fields**

Common Tasks

- Tasks given a graph, evidence E, and optionally parameters:
- Inference: solve $\arg_z \max P(Z = z|E)$
  - E.g., training a classifier/regressor, making predictions, clustering, etc.
  - Based on ML/MAP estimators, or full ***Bayesian estimation***
- More?

Common Tasks

- Tasks given a graph, evidence E, and optionally parameters:
- Inference: solve $\arg_z \max P(Z = z|E)$
  - E.g., training a classifier/regressor, making predictions, clustering, etc.
  - Based on ML/MAP estimators, or full ***Bayesian estimation***
- More?
- Evaluating the marginals $P(Z|E)$ in some complicate models
  - E.g., ***Latent Dirichlet Allocation*** (LDA), etc.
- Learning the structure of a graph**
  - E.g., association rules, other advanced topics

Outline

Conjugate Prior of the Likelihood (1)

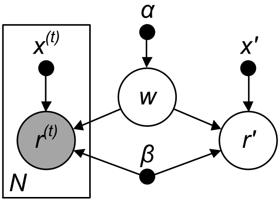
- In many cases, we want to write down $P(Z|E)$ in closed form
  - By Bayes' rule, we have $P(Z|E) = \frac{P(E|Z)P(Z)}{P(E)}$
- If we assume some distribution of the likelihood $P(E|Z)$, then we face a problem: how to pick the distribution of the prior $P(Z)$ such that the posterior $P(Z|E)$ is tractable?
- It is known that for certain likelihood distribution, some prior distribution will lead to the posterior distribution that is in the same family as prior distribution
  - Prior of such distribution is called the ***conjugate prior*** of the likelihood

Linear Gaussian Model

- For each node X_i, we assume $p(X_i|parent(X_i))$ follows some (parametrized) distribution
- A common choice is to form a ***linear Gaussian model***, where each node X_i resembles a linear combination of its parents $Y \in parent(X_i)$
  - $p(x_i|y_1,\cdots,y_p) = \mathcal{N}(x_i|\sum_{j=1}^{p} w_{i,j}y_j + b_i, \sigma_i^2)$, or $p(\boldsymbol{x}_i|\boldsymbol{y}_1,\cdots,\boldsymbol{y}_p) = \mathcal{N}(\boldsymbol{x}_i|\sum_{j=1}^{p} \boldsymbol{W}_{i,j}\boldsymbol{y}_j + \boldsymbol{b}_i, \boldsymbol{\Sigma}_i)$
  - And $p(\boldsymbol{y}_1,\cdots,\boldsymbol{y}_p)$ is Gaussian
- For two nodes X and Y, if $p(X_i|Y)$ and $p(Y)$ follow the linear Gaussian model, then $p(Y|X_i)$ and $p(X_i)$ are both normal distribution
  - $p(X_i)$ is called the ***conjugate prior*** of the likelihood $p(Y|X_i)$ of X_i

Outline

Bayesian Estimation for Linear Regression (1/3)



- Assuming hyperparameters α and β, we have
  $\int p(y, \boldsymbol{w}|\boldsymbol{x}', \mathcal{X}, \alpha, \beta) d\boldsymbol{w} =$
  $\int p(y|\boldsymbol{x}', \boldsymbol{w}, \mathcal{X}, \alpha, \beta) p(\boldsymbol{w}|\boldsymbol{x}', \mathcal{X}, \alpha, \beta) d\boldsymbol{w} =$
  $\int p(y|\boldsymbol{x}', \boldsymbol{w}, \mathcal{X}, \beta) p(\boldsymbol{w}|\boldsymbol{x}', \mathcal{X}, \alpha, \beta) d\boldsymbol{w} =$
  $\int p(y|\boldsymbol{x}', \boldsymbol{w}, \beta) p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta) d\boldsymbol{w}$
  - $\{y\} \perp\!\!\!\perp \{\mathcal{X}\} \,|\, \{\boldsymbol{x}', \boldsymbol{w}, \beta\}$
    - $\{\boldsymbol{w}\} \perp\!\!\!\perp \{\boldsymbol{x}'\} \,|\, \{\mathcal{X}, \alpha, \beta\}$

Bayesian Estimation for Linear Regression (2/3)

- $y' = \arg_y \max \int p(y|\boldsymbol{x}', \boldsymbol{w}, \beta) p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta) d\boldsymbol{w}$
  - $p(y|\boldsymbol{x}', \boldsymbol{w}, \beta) = \mathcal{N}(y|\boldsymbol{w}^\top \boldsymbol{x}', \beta^{-1})$
  - $p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta) = p(\{r^{(t)}\}_t | \{\boldsymbol{x}^{(t)}\}_t, \boldsymbol{w}, \alpha, \beta) p(\boldsymbol{w}|\{\boldsymbol{x}^{(t)}\}_t, \alpha, \beta) = p(\{r^{(t)}\}_t | \{\boldsymbol{x}^{(t)}\}_t, \boldsymbol{w}, \beta) p(\boldsymbol{w}|\alpha)$
  - Let $\boldsymbol{r} = [r^{(1)}, \cdots, r^{(N)}]^\top$ and $\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}]^\top \in \mathbb{R}^{N \times d}$, we have
    - $p(\{r^{(t)}\}_t | \{\boldsymbol{x}^{(t)}\}_t, \boldsymbol{w}, \beta) = p(\boldsymbol{r}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \mathcal{N}(\boldsymbol{r}|\boldsymbol{X}\boldsymbol{w}, \beta^{-1}\boldsymbol{I})$
    - $p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$
  - Notice that $p(\boldsymbol{r}|\boldsymbol{X}, \boldsymbol{w}, \beta)$ and $p(\boldsymbol{w}|\alpha)$ form a linear Gaussian model
    - $\boldsymbol{w}$ is the parent of $\boldsymbol{r}$ and the mean of $p(\boldsymbol{r}|\boldsymbol{X}, \boldsymbol{w}, \beta)$ is a linear combination of $\boldsymbol{w}$
  - Therefore, $p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta) = p(\boldsymbol{w}|\boldsymbol{r}, \boldsymbol{X}, \alpha, \beta) = \mathcal{N}(\boldsymbol{w}|\beta\boldsymbol{\Sigma}\boldsymbol{X}^\top \boldsymbol{r}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = (\alpha\boldsymbol{I} + \beta\boldsymbol{X}^\top\boldsymbol{X})^{-1}$

Bayesian Estimation for Linear Regression (3/3)

- $y' = \arg_y \max \int p(y|\boldsymbol{x}', \boldsymbol{w}, \beta) p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta) d\boldsymbol{w}$, where $p(y|\boldsymbol{x}', \boldsymbol{w}, \beta) = \mathcal{N}(y|(\boldsymbol{x}')^\top \boldsymbol{w}, \beta^{-1})$ and $p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta) = \mathcal{N}(\boldsymbol{w}|\beta\boldsymbol{\Sigma}\boldsymbol{X}^\top \boldsymbol{r}, \boldsymbol{\Sigma})$
  - Again, $p(y|\boldsymbol{x}', \boldsymbol{w}, \beta)$ and $p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta)$ form a linear Gaussian model
    - $\boldsymbol{w}$ is the parent of y and the mean of $p(y|\boldsymbol{x}', \boldsymbol{w}, \beta)$ is a linear combination of $\boldsymbol{w}$
  - We have $\int p(y|\boldsymbol{x}', \boldsymbol{w}, \beta) p(\boldsymbol{w}|\mathcal{X}, \alpha, \beta) d\boldsymbol{w} = \mathcal{N}(y|\beta(\boldsymbol{x}')^\top \boldsymbol{\Sigma}\boldsymbol{X}^\top \boldsymbol{r}, \frac{1}{\beta} + (\boldsymbol{x}')^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x}')$
  - Finally, $y' = \beta(\boldsymbol{x}')^\top \boldsymbol{\Sigma}\boldsymbol{X}^\top \boldsymbol{r} = (\beta\boldsymbol{\Sigma}\boldsymbol{X}^\top \boldsymbol{r})^\top \boldsymbol{x}'$, where $\boldsymbol{\Sigma} = (\alpha \boldsymbol{I} + \beta \boldsymbol{X}^\top \boldsymbol{X})^{-1}$

Why Bayesian Estimation?

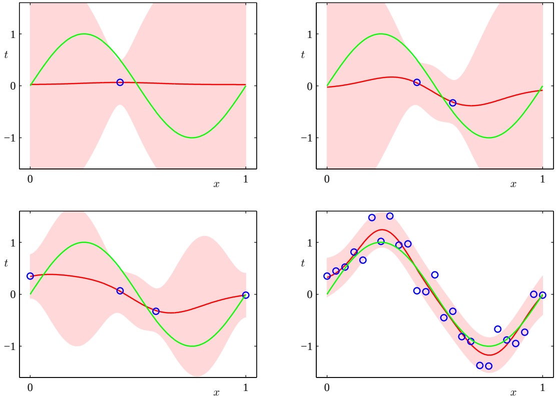
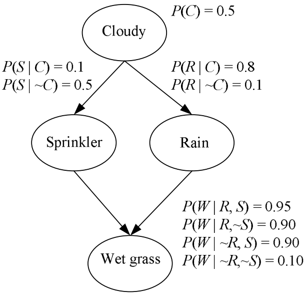


Figure : The prediction made by Bayesian estimation regressor is the red line; where the predictions made by MAP- (or ML-) estimated regressor could be any line in the shaded area.

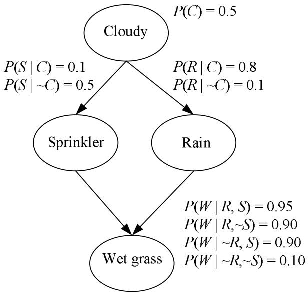
Outline

Space Complexity



- For each node X_i, we need to evaluate/store all possible values of $P(X_i | parent(X_i))$
- Suppose each node has K states and there are totally M nodes, what's the space complexity?

Space Complexity



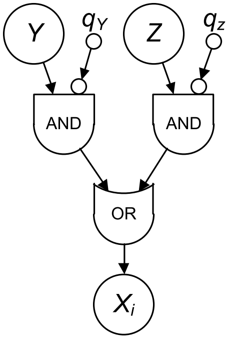
- For each node X_i, we need to evaluate/store all possible values of $P(X_i|parent(X_i))$
- Suppose each node has K states and there are totally M nodes, what's the space complexity?
  - Chain: $(K-1)+(M-1)K(K-1) = O(MK^2)$
  - Fully connected graph: $\sum_i (K-1)K^{|parent(X_i)|} = K^M - 1 = O(K^M)$

Reducing Space Complexity

- How?

Reducing Space Complexity

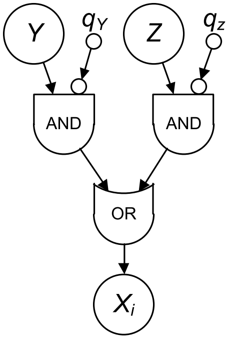
- How?
- Tying: sharing parameters between combinations of parent values



- E.g., modeling the dependency between binary variables as ***noisy OR gates***
- Inhibitors are independent with each other and happens with probabilities q_i
- $P(X_i = 1 | Y = 1, Z = 0) = 1 - q_Y$
- $P(X_i = 1 | Y = 1, Z = 1) = 1 - q_Y q_Z$
- $P(X_i | parent(X_i)) = 1 - \prod_{Y \in parent(X_i), Y=1} q_Y$
- Space complexity?

Reducing Space Complexity

- How?
- Tying: sharing parameters between combinations of parent values



- E.g., modeling the dependency between binary variables as ***noisy OR gates***
- Inhibitors are independent with each other and happens with probabilities q_i
- $P(X_i = 1 | Y = 1, Z = 0) = 1 - q_Y$
- $P(X_i = 1 | Y = 1, Z = 1) = 1 - q_Y q_Z$
- $P(X_i | parent(X_i)) = 1 - \prod_{Y \in parent(X_i), Y=1} q_Y$
- Space complexity? $O(M^2)$ ($(O(M))$ for each node)

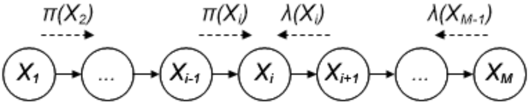
Outline

Evaluating Marginals of All Nodes

- Sometimes, we want to evaluate the marginals of all nodes (given some evidence)
- ***Belief propagation*** allows some components of these marginals to be shared and evaluated just once
  - Reduces time complexity significantly

Evaluating $P(X_i)$'s in a Chain (1/2)

- Problem: to evaluate $P(X_i)$ of ***every*** node X_i in a chain:



- We can evaluate $P(X_i)$ one-by-one
- No problem if nodes are continuous and $p(X_i) = \int_{X_{j,j\neq i}} p(X_1, \cdots, X_i, \cdots, X_M)$ can be written as a closed form (e.g., by assuming a linear Gaussian model)
- Time consuming for discrete variables though, since $P(X_i) = \sum_{\{X_j:, j\neq i\}} P(X_1)P(X_2|X_1), \cdots, P(X_i|X_{i-1}), P(X_{i+1}|X_i), \cdots, P(X_M|X_{M-1})$
  - Assuming that each node has K states, we have time complexity: $O(K^{M-1})$ for each node, $O(MK^{M-1})$ in total

Evaluating $P(X_i)$s in a Chain (2/2)

- Speed up?

Evaluating $P(X_i)$s in a Chain (2/2)

- Speed up?
  - Observer that when computing $P(X_i)$ and $P(X_j)$, $i \neq j$, most conditional probabilities $P(X_{k+1}|X_k)$, $1 \leqslant k \leqslant M-1$, are computed twice
    - It is plausible that we can "reuse" these conditional probabilities to reduce time complexity
- How?

Evaluating $P(X_i)$s in a Chain (2/2)

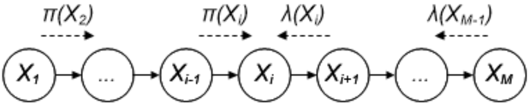
- Speed up?
  - Observer that when computing $P(X_i)$ and $P(X_j)$, $i \neq j$, most conditional probabilities $P(X_{k+1}|X_k)$, $1 \leqslant k \leqslant M-1$, are computed twice
    - It is plausible that we can "reuse" these conditional probabilities to reduce time complexity
- How?
  - One way is to precompute all $P(X_{k+1}|X_k)$s, $1 \leqslant k \leqslant M-1$, and then look up these results to obtain $P(X_i)$s
  - Still exponential to M in time complexity

Belief Propagation along a Chain (1/3)

- Notice that $P(X_i) = \sum_{X_1,X_M} P(X_1, X_i, X_M) = \sum_{X_1,X_M} P(X_1, X_M|X_i)P(X_i) = \sum_{X_1,X_M} P(X_1|X_i)P(X_M|X_i)P(X_i) = \sum_{X_1,X_M} \frac{P(X_i|X_1)P(X_1)}{P(X_i)} P(X_M|X_i)P(X_i) = \sum_{X_1,X_M} \alpha(X_1)\pi(X_i)\lambda(X_i)$
  - $\pi(X_i) = P(X_i|X_1)$ if $i > 1$, and $\pi(X_1) = P(X_1)$
    - $\lambda(X_i) = P(X_M|X_i)$ if $i < M$, and $\lambda(X_M) = 1$
    - $\alpha(X_1) = P(X_1) = \pi(X_1)$ is independent with X_i
  - In addition, $\pi(X_i) = P(X_i|X_1) = \sum_{X_{i-1}} P(X_i, X_{i-1}|X_1) = \sum_{X_{i-1}} P(X_i|X_{i-1}, X_1)P(X_{i-1}|X_1) = \sum_{X_{i-1}} P(X_i|X_{i-1})P(X_{i-1}|X_1) = \sum_{X_{i-1}} P(X_i|X_{i-1})\pi(X_{i-1})$
  - $\lambda(X_i) = P(X_M|X_i) = \sum_{X_{i+1}} P(X_M|X_{i+1}, X_i)P(X_{i+1}|X_i) = \sum_{X_{i+1}} P(X_M|X_{i+1})P(X_{i+1}|X_i) = \sum_{X_{i+1}} P(X_{i+1}|X_i)\lambda(X_{i+1})$

Belief Propagation along a Chain (2/3)

- $P(X_i) = \sum_{X_1, X_M} \alpha(X_1, X_M)\pi(X_i)\lambda(X_i)$
  - $\pi(X_{i+1}) = \sum_{X_i} P(X_{i+1}|X_i)\pi(X_i)$ for $1 \leqslant i \leqslant M-1$
    - $\lambda(X_{i-1}) = \sum_{X_i} P(X_i|X_{i-1})\lambda(X_i)$ for $2 \leqslant i \leqslant M$



- Starting from X_1 till X_{M-1}, each node X_i can forward all $\pi(X_{i+1})$s downward along to chain upon receiving $\pi(X_i)$s from its parent
- Starting from X_M till X_2, each node X_i forwards all its $\lambda(X_{i-1})$s upward along to chain upon upon receiving $\lambda(X_i)$s from its child
- After receiving both $\pi(X_i)$s and $\lambda(X_i)$s from its parent and child respectively, each node X_i can compute $P(X_i)$
  - Note that $\alpha(X_1)$s can be broadcasted to all nodes by X_1 parallel to the above propagations

Belief Propagation along a Chain (3/3)

- The task of evaluating all $P(X_i)$s is now divided into local computation of πs and λs and exchange of these local results
  - We call the inference using this message-passing style as ***belief propagation***
  - Time complexity?

Belief Propagation along a Chain (3/3)

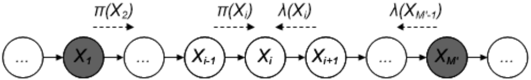
- The task of evaluating all $P(X_i)$s is now divided into local computation of πs and λs and exchange of these local results
  - We call the inference using this message-passing style as ***belief propagation***
  - Time complexity?
    - $O(MK^2 + K^2)$ for each node ($O(MK^2)$ for message exchange and $O(K^2)$ for computing $P(X_i)$)
    - $O(MK^2 + MK^2)$ ***in total***, provided that each node X_i stores its intermediate messages (i.e., $\pi(X_i)$s and $\lambda(X_i)$s)
  - Space complexity?

Belief Propagation along a Chain (3/3)

- The task of evaluating all $P(X_i)$s is now divided into local computation of πs and λs and exchange of these local results
  - We call the inference using this message-passing style as ***belief propagation***
  - Time complexity?
    - $O(MK^2 + K^2)$ for each node ($O(MK^2)$ for message exchange and $O(K^2)$ for computing $P(X_i)$)
    - $O(MK^2 + MK^2)$ ***in total***, provided that each node X_i stores its intermediate messages (i.e., $\pi(X_i)$s and $\lambda(X_i)$s)
  - Space complexity?
    - $O(K^2)$ on each node X_i (for $P(X_{i+1}|X_i)$s, $P(X_i|X_{i-1})$s, $\pi(X_i)$s, and $\lambda(X_i)$s)
    - $O(MK^2)$ totally

Evidences (1/2)

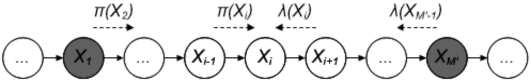
- What if we are given an evidence E?
  - Without loss of generality, let's consider a chain from X_1 to $X_{M'}$, where $\{X_1, X_{M'}\} \subseteq E$, as below:



  - Problem: to evaluate $P(X_i)$ for $2 \leqslant i \leqslant M' - 1$
  - $P(X_i|E) = P(X_i|X_1, X_{M'}) = \frac{P(X_i, X_1, X_{M'})}{P(X_1, X_{M'})} = \alpha(X_1, X_{M'})\pi(X_i)\lambda(X_i)$

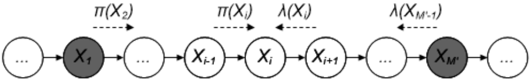
    [Proof]
    - $\pi(X_i) = P(X_i|X_1)$ if $i > 1$, and $\pi(X_1) = P(X_1)$
    - $\lambda(X_i) = P(X_{M'}|X_i)$ if $i < M'$, and $\lambda(X_{M'}) = 1$
    - $\alpha(X_1, X_{M'}) = \frac{P(X_1)}{P(X_1, X_{M'})} = \frac{P(X_1)}{P(X_{M'}|X_1)P(X_1)} = \frac{1}{\pi(X_{M'})} = \frac{1}{\lambda(X_1)}$ is independent with X_i

Evidences (2/2)



- Belief propagation is still applicable except that there is only one $\pi(X_{M'})$ and one $\lambda(X_1)$
  - $\alpha(X_1, X_{M'})$ can be broadcasted to all nodes by $X_{M'-1}$ once it computes $\pi(X_{M'})$ (or by X_2 once it computes $\lambda(X_1)$)
- Time/space complexity?

Evidences (2/2)



- Belief propagation is still applicable except that there is only one $\pi(X_{M'})$ and one $\lambda(X_1)$
  - $\alpha(X_1, X_{M'})$ can be broadcasted to all nodes by $X_{M'-1}$ once it computes $\pi(X_{M'})$ (or by X_2 once it computes $\lambda(X_1)$)
- Time/space complexity? Still $O(M'K^2)$ in both time and space
- If either X_1 or $X_{M'}$ is unobserved, we have either K $\lambda(X_1)$ or K $\pi(X_{M'})$ messages respectively

Outline

Why Sampling?

- To evaluate discrete $P(X|E)$ in a Bayesian network, we produce n samples of it and have the estimate:

$$P(X = x|E = e) = \frac{1}{n}(\#\text{ samples having } X = x \text{ given } E = e)$$

- More generally, to evaluate the expected value of some function f defined over X and E:

$$\mathrm{E}[f|E = e] = \sum_x f(x, e) P(X = x|E = e)$$

we can produce n samples $x^{(t)}$, where $X^{(t)} \sim P(X)$, then estimate

$$\mathrm{E}[f|E = e] = \frac{1}{n}\sum_{t=1}^{n} f\left(x^{(t)}, e\right).$$

Ancestral Sampling

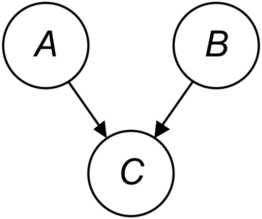
- Given M random variables $X_1, X_2, \cdots, X_M$, we want samples of these variables following their joint distribution $P(X_1, X_2, \cdots, X_M)$
  - How?

Ancestral Sampling

- Given M random variables $X_1, X_2, \cdots, X_M$, we want samples of these variables following their joint distribution $P(X_1, X_2, \cdots, X_M)$
  - How?
- If we have a graph, we can draw sets of samples $\{x_1, x_2, \cdots, x_M\}$ one-by-one, each by:
  1. Sample nodes X's having no parent by following the corresponding $P(X)$
  2. Repeat: sample each child node X whose parents are all sampled by following $P(X|parent(X))$ with parents set to their sampled values
- We call this ***ancestral sampling***

Evidence

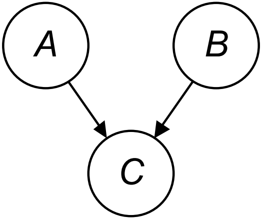
- If is a node without parent, simple fix the value to evidence
- Now suppose $P(A,B,C) = P(A)P(B)P(C|A,B)$



- If $C = c$ is observed, how to make sure the sample value c follows $P(C|A,B)$?

Evidence

- If is a node without parent, simple fix the value to evidence
- Now suppose $P(A, B, C) = P(A)P(B)P(C|A, B)$



- If $C = c$ is observed, how to make sure the sample value c follows $P(C|A, B)$?
- Sample and discard inconsistent ones
  - ***Start over from roots***
- Very in-efficient

Gibbs Sampling (1/2)

- ***Gibbs sampling*** is a Markov Chain Monte Carlo (MCMC) algorithm for obtaining a sequence of observations which are ***approximately*** from from the joint probability distribution of two or more random variables), when direct sampling is difficult
  - Monte Carlo vs. Las Vegas randomized algorithms?
- Suppose we want to obtain M samples of $\boldsymbol{X} = \{X_1, \cdots, X_N\}$ from a joint distribution $P(X_1, \cdots, X_N)$
- Denote the t-th sample by $\boldsymbol{x}^{(t)} = \left\{x_1^{(t)}, \ldots, x_N^{(t)}\right\}$

Gibbs Sampling (2/2)

Input: M, a Bayesian network of $X_1, \cdots, X_N$, and W burn-in samples to discard

Output: $\boldsymbol{x}^{(t)}$'s for $t = 1, \cdots, M$

Initiate $\boldsymbol{x}^{(0)}$;

for $t \leftarrow 1$ **to** $W + M$ **do**

  for $i \leftarrow 1$ **to** N **do**

    $x_i^{(t)} \leftarrow$ a value sampled from $P\left(X_i | X_1^{(t)}, \ldots, X_{i-1}^{(t)}, X_{i+1}^{(t-1)}, \ldots, X_N^{(t-1)}\right)$;

  end

  if $t > W$ **then** Output $x_i^{(t)}$;

end

Algorithm 1: Gibbs sampling algorithm.

Gibbs Sampling (2/2)

Input: M, a Bayesian network of $X_1, \cdots, X_N$, and W burn-in samples to discard

Output: $\boldsymbol{x}^{(t)}$'s for $t = 1, \cdots, M$

Initiate $\boldsymbol{x}^{(0)}$;

for $t \leftarrow 1$ **to** $W + M$ **do**

  for $i \leftarrow 1$ **to** N **do**

    $x_i^{(t)} \leftarrow$ a value sampled from $P\left(X_i | X_1^{(t)}, \ldots, X_{i-1}^{(t)}, X_{i+1}^{(t-1)}, \ldots, X_N^{(t-1)}\right)$;

  end

  if $t > W$ **then** Output $x_i^{(t)}$;

end

Algorithm 2: Gibbs sampling algorithm.

- Why does it work?
- Why discard early (burn-in) samples?

Markov Chain for Inference of $P(X)$

- Set the state space $\mathcal{S}$ of a Markov chain to the range of $\boldsymbol{X}$ ($\mathcal{S}$ may be astronomically large)
- Find a tpm (transition prob. matrix) $\boldsymbol{P}$ such that $P(\boldsymbol{X}) \sim \boldsymbol{\pi_P}$, the steady state distribution
- Then, we can have samples by simply running a random walk:
  1. Pick $\boldsymbol{x}^{(0)}$ somehow;
  2. For $t = 1, \ldots, W + N$, sample $\boldsymbol{x}^{(t)}$ from $P\left(\boldsymbol{X}^{(t)} | \boldsymbol{X}^{(t-1)} = \boldsymbol{x}^{(t-1)}\right)$;
  3. Discard the first W burn-in samples, and output remaining samples;

Why Does the Gibbs Sampling Work?

- The tpm of the Gibbs sampler for $P(\boldsymbol{X})$ where $\boldsymbol{X} = \{X_1, \dots, X_N\}$ is $\boldsymbol{P} = \prod_{i=1}^{N} \boldsymbol{P}^{(i)}$, where

$$\boldsymbol{P}^{(i)}_{\boldsymbol{x}',\boldsymbol{x}} = \begin{cases} 0 & \text{if } \boldsymbol{x}'_{-i} \neq \boldsymbol{x}_{-i} \\ P(X_i = x'_i | \boldsymbol{X}_{-i} = \boldsymbol{x}_{-i}) & \text{if } \boldsymbol{x}'_{-i} = \boldsymbol{x}_{-i} \end{cases}$$

  and the subscript $-i$ denotes all but the i-th element
- Informally, the Gibbs sampler cycles through each of the variables X_i, replacing the current value x_i with a sample from $P(X_i | \boldsymbol{X}_{-i} = \boldsymbol{x}_{-i})$
- If $\boldsymbol{x}$ is a sample from $P(\boldsymbol{X})$, then so is $\boldsymbol{x}'$, since $\boldsymbol{x}'$ differs from $\boldsymbol{x}$ only by replacing x_i with a sample from $P(X_i | \boldsymbol{X}_{-i} = \boldsymbol{x}_{-i})$
- Since $\boldsymbol{P}^{(i)}$ maps samples from $P(\boldsymbol{X})$ to samples from $P(\boldsymbol{X})$, so does $\boldsymbol{P}$. Thus, $P(\boldsymbol{X})$ is a stationary distribution for $\boldsymbol{P}$
- There is another explanation using detailed balance equations [Proof]

Outline

Topic Model

- Topic modeling is a method for analyzing large quantities of unlabeled data.
  - For our purposes, a *topic* is a probability distribution over a collection of words and a *topic model* is a formal statistical relationship between a group of observed and latent (unknown) random variables that specifies a probabilistic procedure to generate the topics—a generative model.
    - The central goal of a topic is to provide a "thematic summary" of a collection of documents.

An Example

- Given 2 documents D_1, D_2 with words
  - $D_1 = \{\text{cat, dog, bird, fish}\}$
    - $D_2 = \{\text{car, bike, bus}\}$
    - We can discover the "topics" (pet, vehicle, ...).
    - A document may have one or more topics in practice.

Latent Dirichlet Allocation

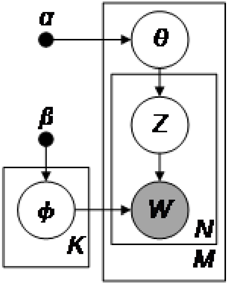
- *Latent Dirichlet allocation* (LDA) is the most common topic model currently in use, allowing documents to have a mixture of topics.
  - LDA provides a generative model that describes how the documents in a corpus were created.

Notation and Terminology

- A *word* is the basic unit of discrete data, defined to be an item from a vocabulary $\{w^1, \ldots, w^V\}$.
    - A *document* D_i is a sequence of N words denoted by $\mathbf{w}_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,N})$, where $w_{i,n}$ is the nth word in the sequence.
        - A *corpus* is a collection of M documents denoted by $D = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$.

The Generative Process

- Assume we know K topic distributions for our corpus, meaning K categoricals containing V elements each.



1. Choose the topic distribution $\theta_i \sim \mathrm{Dir}(\alpha)$ for each document D_i where $i \in \{1, \ldots, M\}$ (θ_i is a categorical of length K).
2. Choose the word distribution $\phi_k \sim \mathrm{Dirichlet}(\beta)$ for each topic where $k \in \{1, \ldots, K\}$ (ϕ_k is a vector of length V).
   - β is a V-dimension vector of positive reals.

   a. For each of the words $w_{i,n}$ where $n \in \{1, \ldots, N\}$:
      1. Choose a topic $z_{i,n} \sim \mathrm{Categorical}(\theta_i)$.
      2. Choose a word $w_{i,n} \sim \mathrm{Categorical}(\phi_{z_{i,n}})$.

Our Goal

- Given α, β, and document D_i with word sequence $\mathbf{w}_i$, what are the most probable values for θ_i?

Our Goal

- Given α, β, and document D_i with word sequence $\mathbf{w}_i$, what are the most probable values for θ_i?

$$
\begin{aligned}
P(\theta_i|\mathbf{w}_i, \alpha, \beta) &= \int \sum_{\boldsymbol{z}_i} P(\theta_i, \boldsymbol{z}_i, \phi|\mathbf{w}_i, \alpha, \beta)\, d\phi \\
&\propto \int \sum_{\boldsymbol{z}_i} P(\mathbf{w}_i|\theta_i, \mathbf{z}_i, \phi, \beta)\, P(\theta_i, \mathbf{z}_i, \phi|\alpha)\, d\phi
\end{aligned}
$$

- The close form of the posterior is intractable (due to the unknown $\boldsymbol{z}_i$)

Gibbs Sampling for LDA (1/3)

- In LDA, the distribution of the topics $\mathbf{Z}$ for words $\mathbf{W}$ is unknown and $\mathbf{Z}$ is multivariate.
- Hence, the Gibbs sampling procedure boils down to estimate

$$P\left(Z_{i,n} = t | \mathbf{z}_{-i,n}, \mathbf{w}\right).$$

- Here, θ, ϕ are integrated out. If we know the exact $\mathbf{Z}_i$ for each document D_i, it's trivial to estimate θ_i and ϕ_i.
- We have

$$\begin{aligned} P\left(Z_{i,n} = t | \mathbf{z}_{-i,n}, \mathbf{w}, \alpha, \beta\right) \\ & \propto P\left(Z_{i,n} = t, w_{i,n} | \mathbf{z}_{-i,n}, \mathbf{w}_{-i,n}, \alpha, \beta\right) \\ & = P\left(w_{i,n} | Z_{i,n} = t, \mathbf{z}_{-i,n}, \mathbf{w}_{-i,n}, \beta\right) P\left(Z_{i,n} = t | \mathbf{z}_{-i,n}, \mathbf{w}_{-i,n}, \alpha\right) \\ & = P\left(w_{i,n} | Z_{i,n} = t, \mathbf{z}_{-i,n}, \mathbf{w}_{-i,n}, \beta\right) P\left(Z_{i,n} = t | \mathbf{z}_{-i,n}, \alpha\right) \end{aligned}$$

Gibbs Sampling for LDA (2/3)

- For the first term, we have

$$P(w_{i,n}|Z_{i,n}=t,\mathbf{z}_{-i,n},\mathbf{w}_{-i,n},\beta) = \int P(w_{i,n}|Z_{i,n}=t,\phi_t)\,P(\phi_t|\mathbf{z}_{-i,n},\mathbf{w}_{-i,n},\beta)\,d\phi_t$$

$$P(\phi_t|\mathbf{z}_{-i,n},\mathbf{w}_{-i,n},\beta) = \frac{P(\mathbf{w}_{-i,n}|\phi_t,\mathbf{z}_{-i,n})\,P(\phi_t|\beta)}{P(\mathbf{w}_{-i,n}|\mathbf{z}_{-i,n},\beta)} \sim \text{Dirichlet}\left(\beta+\mathbf{N}_t^{-i,n(w)}\right)$$

- Here, $\mathbf{N}_t^{-i,n(w)}$ is a V-dimension vector and $\mathbf{N}_{t,v}^{-i,n(w)}$ is the number of instances of the v-th word in the vocabulary assigned to topic t in document D_i, excluding the instance $w_{i,n}$. Recall that the Dirichlet is the conjugate prior for the multinomial. Thus, the posterior is also Dirichlet.
- Using the property of Dirichlet-multinomial distribution, we have

$$P(w_{i,n}|Z_{i,n}=t,\mathbf{z}_{-i,n},\mathbf{w}_{-i,n},\beta) = \frac{\Gamma\left(\sum_v\left(\beta_v+\mathbf{N}_{t,v}^{-i,n(w)}\right)\right)}{\Gamma\left(1+\sum_v\left(\beta_v+\mathbf{N}_{t,v}^{-i,n(w)}\right)\right)}\left(\frac{\Gamma\left(\mathbf{N}_{t,w_{i,n}}^{-i,n(w)}+\beta_{w_{i,n}}+1\right)}{\Gamma\left(\mathbf{N}_{t,w_{i,n}}^{-i,n(w)}+\beta_{w_{i,n}}\right)}\right) = \frac{\mathbf{N}_{t,w_{i,n}}^{-i,n(w)}+\beta_{w_{i,n}}}{\sum_v\left(\mathbf{N}_{t,v}^{-i,n(w)}+\beta_v\right)}.$$

Gibbs Sampling for LDA (3/3)

- Similarly, for the second term, we have

$$P\left(Z_{i,n}=t|\mathbf{z}_{-i,n},\boldsymbol{\alpha}\right)=\int P\left(Z_{i,n}=t|\boldsymbol{\theta}_i\right)P\left(\boldsymbol{\theta}_i|\mathbf{z}_{-i,n},\boldsymbol{\alpha}\right)d\boldsymbol{\theta}_i$$

$$\begin{aligned}P\left(\boldsymbol{\theta}_i|\mathbf{z}_{-i,n},\boldsymbol{\alpha}\right)&\propto P\left(\mathbf{z}_{-i,n}|\boldsymbol{\theta}_i\right)P\left(\boldsymbol{\theta}_i|\boldsymbol{\alpha}\right)\\&\sim\mathrm{Dirichlet}\left(\boldsymbol{\alpha}+\mathbf{N}^{-i,n(z)}\right)\end{aligned}$$

where $\mathbf{N}^{-i,n(z)}$ is a K-dimension vector and $\mathbf{N}_k^{-i,n(z)}$ is the number of words assigned to topic k in document D_i, excluding the instance $z_{i,n}$.

- Then, we have

$$P\left(Z_{i,n}=t|\mathbf{z}_{-i,n},\boldsymbol{\alpha}\right)=\frac{\mathbf{N}_t^{-i,n(z)}+\alpha_t}{\sum_k\left(\mathbf{N}_k^{-i,n(z)}+\alpha_k\right)}.$$

- Thus,

$$P\left(Z_{i,n}=t|\mathbf{z}_{-i,n},\mathbf{w},\boldsymbol{\alpha},\boldsymbol{\beta}\right)\propto\frac{\mathbf{N}_{t,w_{i,n}}^{-i,n(w)}+\beta_{w_{i,n}}}{\sum_v\left(\mathbf{N}_{t,v}^{-i,n(w)}+\beta_v\right)}\times\frac{\mathbf{N}_t^{-i,n(z)}+\alpha_t}{\sum_k\left(\mathbf{N}_k^{-i,n(z)}+\alpha_k\right)}.$$

Estimate ϕ and θ

- To obtain ϕ and θ, we can simply calculate

$$\phi_{k,v} = \frac{n_v^{(k)} + \beta_v}{\sum_{j=1}^{V} \left(n_j^{(k)} + \beta_j \right)}$$

$$\theta_{i,k} = \frac{n_k^{(i)} + \alpha_k}{\sum_{t=1}^{K} \left(n_t^{(i)} + \alpha_t \right)}$$

  where $n_j^{(k)}$ is the frequency of word w^j in the vocabulary assigned to topic k, and $n_t^{(i)}$ is the number of words assigned to topic t in document D_i.

Outline