Markov Chain Monte Carlo

Richard Xu

January 3, 2023

1 Topic Summary

In many data science scenarios, we need to model a latent variable of interest $\mathbf{z}$ given data $\mathbf{x}$, for example, we may need to compute the following quantities:

$$\mathbb{E}_{p(\mathbf{z}|\mathbf{x})}[\mathbf{z}] \tag{1}$$

this can be approximated by:

$$\frac{1}{N}\sum_{i=1}^{N}\mathbf{z}^{(i)} \quad \mathbf{z}^{(i)} \sim p(\mathbf{z}|\mathbf{x}) \tag{2}$$

Alternatively, you might be interested in computing $\arg\max_{\mathbf{z}}\{p(\mathbf{z}|\mathbf{x})\}$. This can be approximated as:

$$\arg\max\{p(\mathbf{z}^{(i)})\} \quad \mathbf{z}^{(i)} \sim p(\mathbf{z}|\mathbf{x}) \tag{3}$$

So you can see that both approximations require sampling $\mathbf{z} \sim p(\mathbf{z}|\mathbf{x})$. However, sampling may not always be straightforward to implement. Therefore, in this topic, we will discuss a sampling method called, Markov Chain Monte Carlo (MCMC).

1.1 Examples

1.1.1 LDA example

For example, in the LDA example:

$$\begin{aligned}\mathbf{x} &\equiv \{w_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}\\ \mathbf{z} &\equiv \{\{\beta_j\}_{j=1}^{K}, \{\theta_d\}_{d=1}^{D}, \{z_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}\}\end{aligned} \tag{4}$$

Therefore, in LDA, by obtaining samples from:

$$p\Big(\underbrace{\{\beta_j\}_{j=1}^{K}, \{\theta_d\}_{d=1}^{D}, \{z_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}}_{\mathbf{z}} \,\Big|\, \underbrace{\{w_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}}_{\mathbf{x}}\Big) \tag{5}$$

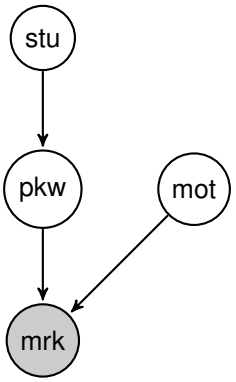
it allows us to approximate:

$$\mathbb{E}\Big[\Big\{\{\beta_j\}_{j=1}^{K}, \{\theta_d\}_{d=1}^{D}, \{z_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}\Big\}\Big] \tag{6}$$

1.1.2 Generic "mark" example

Check out the toy example of the graph model from the previous lesson:

1. "months of studies" (stu)
2. "prior knowledge" (pkw)
3. "motivation" (mot)
4. "mark obtained" (mrk)



We have three latent variables, (stu), (pkw), (mot) and one observation (mrk), then if we want to perform posterior inference, i.e.,:

$$\Pr\Big(\underbrace{\text{stu}, \text{pkw}, \text{mot}}_{\text{latent}} \Big| \underbrace{\text{mrk}}_{\text{observation}}\Big) \tag{7}$$

which allows us to compute things such as:

$$\mathbb{E}_{\Pr\left(\text{stu},\text{pkw},\text{mot}|\text{mrk}\right)}[\text{stu}, \text{pkw}, \text{mot}] \tag{8}$$

This can be approximated by Monte-Carlo, i.e., Eq.(2), and it requires us to first be able to sample from $\Pr\big(\text{stu}, \text{pkw}, \text{mot}|\text{mrk}\big)$! Of course, in this lecture, we study how the MCMC algorithm can help us achieving this.

2 Explain using finite dimensionality

2.1 Stochastic matrices

Right stochastic matrix (or row stochastic matrix) is a real square matrix, with **each row** summing to 1.

$$\begin{bmatrix} K_{1\to 1} & \dots & K_{1\to n} \\ \dots & \dots & \dots \\ K_{d\to 1} & \dots & K_{d\to n} \\ \dots & \dots & \dots \\ K_{n\to 1} & \dots & K_{n\to n} \end{bmatrix} \tag{9}$$

for example:

$$\begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.05 & 0.9 & 0.05 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \tag{10}$$

Left stochastic matrix (or column stochastic matrix) is a real square matrix, with **each column** summing to 1

$$\begin{bmatrix} K_{1\to 1} & \dots & K_{n\to 1} \\ \dots & \dots & \dots \\ K_{1\to d} & \dots & K_{n\to d} \\ \dots & \dots & \dots \\ K_{1\to n} & \dots & K_{n\to n} \end{bmatrix} \tag{11}$$

for example:

$$\begin{bmatrix} 0.3 & 0.05 & 0.7 \\ 0.2 & 0.9 & 0.2 \\ 0.5 & 0.05 & 0.1 \end{bmatrix} \tag{12}$$

doubly stochastic matrices: is a real square matrix, where both **each column** and **each row** summing to 1.

2.2 Product of two stochastic matrix is still stochastic

We choose both A and B to be **right stochastic matrix**, where each entry in the product $C = AB$ is a dot product of a row from A and a column from B:

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj} \tag{13}$$

let's look at what if we sum over a row of AB, i.e., $C_{i,:}$

$$\begin{aligned} \sum_{j=1}^{n} C_{ij} &= \sum_{j=1}^{n} \sum_{k=1}^{n} A_{ik} B_{kj} \\ &= \sum_{k=1}^{n} (A_{ik} \sum_{j=1}^{n} B_{kj}) \end{aligned} \tag{14}$$

1. Because B is right stochastic, $\sum_{j=1}^{n} B_{kj} = 1$
2. Because A is right stochastic, $\sum_{k=1}^{n} A_{ik} = 1$

therefore, once again, the new product matrix $C = AB$ is a **right stochastic matrix**.

2.2.1 left stochastic matrix

We can show the similar result when A and B are **left stochastic matrix**:

$$\begin{aligned}\sum_{i=1}^{n} C_{ij} &= \sum_{i=1}^{n}\sum_{k=1}^{n} A_{ik}B_{kj} \\ &= \sum_{k=1}^{n}(B_{kj}\sum_{i=1}^{n} A_{ik}) \\ &= \sum_{k=1}^{n} B_{kj} \qquad \because \sum_{i=1}^{n} A_{ik} = 1 \\ &= 1\end{aligned} \tag{15}$$

2.3 Perron-Frobenius Theorem:

Theorem 1 *If K is a left stochastic matrix, then, 1 is an eigenvalue of multiplicity one. (meaning 1 is the largest eigenvalue: all the other eigenvalues have absolute value smaller than 1), and the eigenvectors corresponding to the eigenvalue 1 have either only positive entries or only negative entries.*

3 Power Method Convergence Theorem

Let K be a positive, **left** (i.e., each column add to one) stochastic $n \times n$ matrix, and π^* be its **target** probabilistic eigenvector corresponding to the eigenvalue 1. then:

$$\begin{bmatrix} K_{1\to 1} & \dots & K_{n\to 1} \\ \dots & \dots & \dots \\ K_{1\to n} & \dots & K_{n\to n} \end{bmatrix} \begin{bmatrix} \pi_1^* \\ \dots \\ \pi_n^* \end{bmatrix} = \begin{bmatrix} \pi_1^* \\ \dots \\ \pi_n^* \end{bmatrix}$$
$$\implies \pi_d^* = \sum_{i=1}^{n} \pi_i^* K_{i\to d} \tag{16}$$

we can verify $\sum_{d=1}^{n} \pi_d^* = 1$:

$$\begin{aligned}\sum_{d=1}^{n} \pi_d^* &= \sum_{d=1}^{n}\sum_{i=1}^{n} \pi_i^* K_{i\to d} \\ &= \sum_{i=1}^{n} \pi_i^* \sum_{d=1}^{n} K_{i\to d} \\ &= 1\end{aligned} \tag{17}$$

therefore, only when **left** stochastic matrix multiplies a **column** stochastic vector, results to another stochastic vector.

Let $\pi^{(1)}$ be the column vector with all entries equal to some arbitrary stochastic vector. Then sequence:

$$\{\pi^{(1)}, K\pi^{(1)}, K^2\pi^{(1)}, \ldots, K^t\pi^{(1)} \ldots, K^\infty\pi^{(1)}\} \tag{18}$$

converges to the vector π^*

$$\lim_{t\to\infty} K^t = K^\infty \implies \lim_{t\to\infty} K^t\pi^{(1)} = \pi^* \tag{19}$$

Exercise Generate some random matrix in MATLAB and to show an example of the above.

3.1 Power method

```
import numpy as np
# create left stochastic matrix
d = 4
K = np.random.rand(d,d)
print(K.sum(axis=0)[None,:])
K = K/(K.sum(axis=0)[None,:])
print(K)
print("")

# K_\infty = K^100
Kinf = np.linalg.matrix_power(K, 100)
print(Kinf)

# try a random initialization
print("random initialization")
for i in range(10):
    pi = np.random.rand(d,1)
    pi = pi/sum(pi)
    print("")
    print(Kinf@pi)
```

We will see the same power method in Page Ranking algorithm again. The initial probability vector: $\pi^{(1)}$ can be expressed as a linear combination of eigenvectors of K:

$$\pi^{(1)} = c_1 \times \pi^* + c_2 v_2 + \ldots c_n v_n \tag{20}$$

Then:

$$
\begin{aligned}
K\pi^{(1)} &= K(\pi^* + c_2 v_2 + \dots c_n v_n) \\
&= c_1 \underbrace{\lambda_1}_{=1} \pi^* + c_2 \lambda_2 v_2 + \dots c_n \lambda_n v_n \qquad \text{definition of eigven value/vector} \\
&= c_1 \pi^* + c_2 \lambda_2 v_2 + \dots c_n \lambda_n v_n \\
\implies K^2 x &= c_1 \pi^* + c_2 \lambda_2^2 v_2 + \dots c_n \lambda_n^2 v_n \\
\implies K^t x &= c_1 \pi^* + c_t \lambda_2^t v_2 + \dots c_n \lambda_n^t v_n
\end{aligned} \tag{21}
$$

therefore:

$$
\begin{aligned}
&\lim_{t \to \infty} \lambda_j^k \to 0 \\
\implies &\lim_{t \to \infty} K^t x \to c_1 \pi^*
\end{aligned} \tag{22}
$$

and since $K^t x$ will result a stochastic vector, so $c_1 \to 1$

3.2 Extend to continous case

in the **discrete** case:

$$
\begin{bmatrix}
K_{1\to1} & K_{2\to1} & \dots & K_{n\to1} \\
\dots & \dots & \dots & \dots \\
K_{1\to d} & K_{2\to d} & \dots & K_{n\to d} \\
\dots & \dots & \dots & \dots \\
K_{1\to n} & K_{2\to n} & \dots & K_{n\to n}
\end{bmatrix}
\begin{bmatrix}
\pi_1^* \\ \dots \\ \pi_d^* \\ \dots \\ \pi_n^*
\end{bmatrix}
=
\begin{bmatrix}
\pi_1^* \\ \dots \\ \pi_d^* \\ \dots \\ \pi_n^*
\end{bmatrix} \tag{23}
$$

then let's see what it may be in the **continous** case, let $\pi(x)$ be the target distribution:

$$
\pi(x^{(n+1)}) = \int_{x_n} \pi\big(x^{(n)}\big) K\big(x^{(n)} \to x^{(n+1)}\big) \tag{24}
$$

A transition kernel K contains element-wise entries:

$$
\{K\big(x^{(n)} \to x^{(n+1)}\big)\} \qquad \forall x^{(n)}, x^{(n+1)} \tag{25}
$$

Sometimes we prefer to write $\big(x^{(n)}$ as $x\big)$ and $\big(x^{(n+1)}$ as $x^*\big)$

- $K(x \to x^*)$ is the probability a process at state x moves to state x^* in a **one step**
- $K^n(x \to x^*)$ is the probability a process at state x moves to state x^* in **n steps**

3.2.1 Power Method Convergence in continuous case

One may have first sample $x^{(1)}$ distributed from an arbitrary distribution:

$$x^{(1)} \sim \pi^{(1)} \tag{26}$$

by applying K function, to obtain $x^{(2)}$ given $x^{(1)}$ with probability:

$$\begin{aligned} \pi^{(2)}(x^{(2)}) &= \int_{x^{(1)}} \boldsymbol{\pi}(x^{(1)}, x^{(2)}) \mathrm{d}x^{(1)} \\ &= \int_{x^{(1)}} \pi^{(1)}(x^{(1)}) K\big(x^{(1)} \to x^{(2)}\big) \mathrm{d}x^{(1)} \end{aligned} \tag{27}$$

by applying K function again, to obtain $x^{(3)}$ with probability:

$$\begin{aligned} \pi^{(3)}(x^{(3)}) &= \int_{x^{(1)}} \int_{x^{(2)}} \boldsymbol{\pi}\big(x^{(1)}, x^{(2)}, x^{(3)}\big) \mathrm{d}x^{(1)} \mathrm{d}x^{(2)} \\ &= \int_{x^{(1)}} \int_{x^{(2)}} \pi^{(1)}(x^{(1)}) K\big(x^{(1)} \to x^{(2)}\big) K\big(x^{(2)} \to x^{(3)}\big) \mathrm{d}x^{(1)} \mathrm{d}x^{(2)} \quad \text{Markov} \\ &= \int_{x^{(1)}} \pi^{(1)}(x^{(1)}) \underbrace{\int_{x^{(2)}} K\big(x^{(1)} \to x^{(2)}\big) K\big(x^{(2)} \to x^{(3)}\big) \mathrm{d}x^{(2)}}\, \mathrm{d}x^{(1)} \\ &= \int_{x^{(1)}} \pi^{(1)}(x^{(1)}) K^2\big(x^{(1)} \to x^{(3)}\big) \mathrm{d}x^{(1)} \\ &\vdots \\ \pi^{(t)}(x^{(t)}) &= \int_{x^{(1)}} \pi^{(1)}(x^{(1)}) K^{t-1}\big(x^{(1)} \to x^{(t)}\big) \mathrm{d}x^{(1)} \end{aligned} \tag{28}$$

The last line is simply the continuous (or function equivalent) of $\pi^{(t)} = K\pi^{(t-1)}$. Then:

$$\lim_{t \to \infty} \pi^{(t)}\big(x^{(t)}\big) \to \pi(x^{(t)}) \tag{29}$$

3.3 MCMC algorithm

Given the above formula, it seems that we can (using the correct/corresponding kernel $K(x \to x^{*})$ to construct a series of distributions $\pi^{(1)}, \pi^{(2)}, \ldots$, and eventually, it will converge to a stationary distributions π. Along the way, we can sample from these distributions. However, this does not help at all! All the construction of auxiliary distributions $\pi^{(1)}, \pi^{(2)}, \ldots$ are of no interest to us at all!

Instead, however, we can conditionally transit (or informally "move") one sample using conditional density/transition kernel $K(x \to x^{*})$, and as Eq.(28) shows that at time t^{th} of the "movement", the samples will be targeting $\pi^{(t)}$. Therefore, as long as we keep "move" them enough times, they eventually will be targeting the stationary distribution π, and thereafter, applying $K(x \to x^{*})$ again won't change its distribution.

3.4 Burn in samples

We know,

$$\lim_{t\to\infty} \pi^{(t)}\big(x^{(t)}\big) \to \pi(x^{(t)}) \tag{30}$$

But, in practice,

$$\lim_{t\to B} \pi^{(t)}\big(x^{(t)}\big) \not\to \pi(x^{(t)}) \tag{31}$$

$\{x^{(1)}, \ldots, x^{(B)}\}$ are the **burn-in** samples, which we discard.

3.5 What is MCMC research is all about

- **equilibrium equation**:

$$\pi(x^*) = \int_x \pi(x) K\big(x \to x^*\big) \mathrm{d}x \tag{32}$$

- In machine learning, we always know the expression of stationary distribution $\pi(x)$,
- Our task is therefore, given target distribution $\pi(x)$, find the **corresponding** $K(x \to x^*)$ to generate samples in a Markov fashion.

3.6 Detailed Balance

At equilibrium, that stationary distribution satisfies:

$$\pi(x^*) = \int_x \pi(x) K\big(x \to x^*\big) \mathrm{d}x \qquad \textbf{equilibrium equation} \tag{33}$$

Proving **equilibrium equation** may be difficult in some cases, therefore, we instead prove **detail balance**:

$$\pi(x) K\big(x \to x^*\big) = \pi\big(x^*\big) K\big(x^* \to x\big) \tag{34}$$

detailed balance implies **equilibrium equation**:

$$\begin{aligned} \int_x \pi(x) K\big(x \to x^*\big) \mathrm{d}x &= \int_x \pi(x^*) K\big(x^* \to x\big) \mathrm{d}x \\ &= \pi(x^*) \int_x K(x^* \to x) \mathrm{d}x \\ &= \pi(x^*) \qquad \textbf{equilibrium equation} \end{aligned} \tag{35}$$

the reverse is not always true.

3.7 Extend target distribution with auxiliary variables

- At equilibrium, that stationary distribution satisfies:

$$\pi(x^*) = \int_x \pi(x) K\big(x \to x^*\big) \mathrm{d}x \tag{36}$$

- under many scenarios, we may have an extended joint density (x, u):

$$\pi(x|u)\pi(u)K\big(u, x \to u^*, x^*\big) = \pi\big(x^*|u^*\big)\pi(u^*)K\big(x^*, u^* \to x, u\big) \tag{37}$$

- u is auxiliary variables help samping
- one needs to ensure that:

$$\int_u \pi(x, u)\mathrm{d}u = \pi(x) \tag{38}$$

3.8 Alternative Use of Stochastic Matrix

- Before dive deep into MCMC algorithms, let's have a look at alternative use of stochastic matrix
- PageRank algorithm is different to MCMC, in PageRank algorithm: K is known
- PageRank algorithm then computes π which is the **invariant distribution**, tells the importance of each web page.

4 PageRank algorithm (Optional)

This the opposite problem to monte-carlo markov chain. In MCMC, we do know target distribution π, but we need to discover the transition Kernel K, so that we can put it in a conditional density algorithm. However, for Page rank algorithm, we do know transition Kernel K, but we do not know what is the target distribution π

Imagine we have the following four web pages and their links, we can then compute the probability of navigating from i^{th} page (discrete state) to j^{th} page (discrete state)

- Page 1 links to pages $\{2, 3\}$

$$\implies K_{1\to1} = 0, K_{1\to2} = \frac{1}{2}, K_{1\to3} = \frac{1}{2}, K_{1\to4} = 0 \tag{39}$$

- Page 2 has links to pages $\{1, 3, 4\}$

$$\implies K_{2\to1} = \frac{1}{3}, K_{2\to2} = 0, K_{2\to3} = \frac{1}{3}, K_{2\to4} = \frac{1}{3} \tag{40}$$

- Page 3 has links to pages $\{1, 3\}$

$$\implies K_{3\to1} = \frac{1}{2}, K_{3\to2} = 0, K_{3\to3} = \frac{1}{2}, K_{3\to4} = 0 \tag{41}$$

- Page 4 has links to pages $\{2, 3\}$

$$\implies K_{4\to1} = 0, K_{4\to2} = \frac{1}{2}, K_{4\to3} = \frac{1}{2}, K_{4\to4} = 0 \tag{42}$$

4.1 Stochastic matrix K

- From the preceding example, **Left stochastic matrix** is:

$$\begin{bmatrix} K_{1\to1} & K_{2\to1} & K_{3\to1} & K_{4\to1} \\ K_{1\to2} & K_{2\to2} & K_{3\to2} & K_{4\to2} \\ K_{1\to3} & K_{2\to3} & K_{3\to3} & K_{4\to3} \\ K_{1\to4} & K_{2\to4} & K_{3\to4} & K_{4\to4} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{3} & 0 & 0 \end{bmatrix} \tag{43}$$

- From Power Method Convergence Theorem, we know:
  - sequence $\{\pi^{(1)}, K\pi^{(1)}, K^2\pi^{(1)}, \ldots, K^t\pi^{(1)} \ldots, K^\infty\pi^{(1)}\}$ converges to the vector π^*

$$\lim_{t\to\infty} K^t\pi^{(1)} = \pi^* \tag{44}$$

    where π^* is a **probabilistic eigenvector** of K corresponding to the eigenvalue 1.

- **Exercise** What is the usefulness of π^* in the setting of web pages?

4.2 Usefulness of π^* in the setting of web pages

The **answer** to usefulness of π^* in the setting of web pages is:

- Shows how **important** each webpage is
- i.e., regardless of the probabilities of the initial webpage visit: $\pi^{(1)}$,
- $\pi^{(1)} \to \pi^*$, where $\pi^*(i)$ is the target distribution i.e, the probability that the visit will end up at a web page i.
- Note that this is a **reverse problem** of MCMC

4.3 Dangling nodes

- What happens when you have the following K:

$$\begin{bmatrix} K_{1\to1} & K_{2\to1} & K_{3\to1} & K_{4\to1} \\ K_{1\to2} & K_{2\to2} & K_{3\to2} & K_{4\to2} \\ K_{1\to3} & K_{2\to3} & K_{3\to3} & K_{4\to3} \\ K_{1\to4} & K_{2\to4} & K_{3\to4} & K_{4\to4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 & 0 \end{bmatrix} \tag{45}$$

- Note that 4^{th} has no out-going node
- **Exercise** check eigenvector correspond to eigenvalue of 1
- What is the eigenvector correspond to eigenvalue of 1, if we change K into:

$$\begin{bmatrix} K_{1\to1} & K_{2\to1} & K_{3\to1} & K_{4\to1} \\ K_{1\to2} & K_{2\to2} & K_{3\to2} & K_{4\to2} \\ K_{1\to3} & K_{2\to3} & K_{3\to3} & K_{4\to3} \\ K_{1\to4} & K_{2\to4} & K_{3\to4} & K_{4\to4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 & \mathbf{1} \end{bmatrix} \tag{46}$$

  Exercise give reason to why this is so?
- **Exercise** How can we solve this?

4.4 Dangling nodes: what may be the solution?

$$\begin{bmatrix} K_{1\to1} & K_{2\to1} & K_{3\to1} & K_{4\to1} \\ K_{1\to2} & K_{2\to2} & K_{3\to2} & K_{4\to2} \\ K_{1\to3} & K_{2\to3} & K_{3\to3} & K_{4\to3} \\ K_{1\to4} & K_{2\to4} & K_{3\to4} & K_{4\to4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 & 0 \end{bmatrix} \tag{47}$$

- One simply solution is:

$$\begin{bmatrix} K_{1\to1} & K_{2\to1} & K_{3\to1} & K_{4\to1} \\ K_{1\to2} & K_{2\to2} & K_{3\to2} & K_{4\to2} \\ K_{1\to3} & K_{2\to3} & K_{3\to3} & K_{4\to3} \\ K_{1\to4} & K_{2\to4} & K_{3\to4} & K_{4\to4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{2} & 1 & 0 \end{bmatrix} \tag{48}$$

- in words, it means any page doesn't have out-link, we assume it has equal probability of visiting entire web.
- Of course, **data mining** researchers may argue certain web page (having certain properties) may attract higher weights etc.

4.5 Disconnected sub-graphs

- What happens when you have the following K:

$$\begin{bmatrix} K_{1\to1} & K_{2\to1} & K_{3\to1} & K_{4\to1} \\ K_{1\to2} & K_{2\to2} & K_{3\to2} & K_{4\to2} \\ K_{1\to3} & K_{2\to3} & K_{3\to3} & K_{4\to3} \\ K_{1\to4} & K_{2\to4} & K_{3\to4} & K_{4\to4} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \end{bmatrix} \tag{49}$$

- node $\{1, 2\}$ and $\{3, 4\}$ each form a sub-graph.
- **Exercise** check eigenvector correspond to eigenvalue of 1, also multiplicity of eigenvalue 1
- **Exercise** How can we solve this?

4.6 Disconnected sub-graphs: what may be the solution?

$$\begin{bmatrix} K_{1\to1} & K_{2\to1} & K_{3\to1} & K_{4\to1} \\ K_{1\to2} & K_{2\to2} & K_{3\to2} & K_{4\to2} \\ K_{1\to3} & K_{2\to3} & K_{3\to3} & K_{4\to3} \\ K_{1\to4} & K_{2\to4} & K_{3\to4} & K_{4\to4} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{2} \end{bmatrix} \tag{50}$$

- One solution is to use a convex combination between K and a square matrix having identical elements $\frac{1}{n}$:

$$K' = (1-p)K + p\left(\frac{1}{n}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right) \tag{51}$$

- in words, it means most of the time $1-p$, a surfer will follow links to navigate a page
- but with probability p, it will arbitrarily close the current page and go to the new one
- **Exercise** Prove K remains a left stochastic matrix

4.7 How to compute the one hundblue billion dimension eigenvector?

- starting from the vector (not probabilistic eigenvector), x:

$$x = [1 \quad 1 \quad \dots \quad 1]^\top \tag{52}$$

- generate the sequence: $\{x, Kx, K^2x \dots K^tx\}$ until convergence then its is the eigenvectors of K correspond to eigenvalue of 1, up to a normalisation constant c
- This is solved using **power method**

4.8 Power method

power method is used to finding an eigenvector of a square matrix corresponding to the **largest** eigenvalue (in terms of absolute value). We already saw it in the section (3.1):

- for stochastic matrix K has eigenvalues:

$$1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n| \tag{53}$$

- the initial vector: x as a linear combination of eigenvectors of K:

$$x = c_1 v_1 + c_2 v_2 + \ldots c_n v_n \tag{54}$$

Then,

$$\begin{aligned} Kx &= K(c_1 v_1 + c_2 v_2 + \ldots c_n v_n) \\ &= c_1 \underbrace{\lambda_1}_{=1} v_1 + c_2 \lambda_2 v_2 + \ldots c_n \lambda_n v_n \qquad \text{definition of eigven value/vector} \\ &= c_1 v_1 + c_2 \lambda_2 v_2 + \ldots c_n \lambda_n v_n \\ \implies K^2 x &= c_1 v_1 + c_2 \lambda_2^2 v_2 + \ldots c_n \lambda_n^2 v_n \\ \implies K^t x &= c_1 v_1 + c_t \lambda_2^t v_2 + \ldots c_n \lambda_n^t v_n \end{aligned} \tag{55}$$

$$\lambda_j^k \to 0 \text{ when } j \geq 2 \implies K^t x \to c_1 v_1 \tag{56}$$

4.9 Few notes

- The second largest eigen value determines the convergence
- **Exercise** Perform the following simulations:
  - generate lots of K and choose one which has **large** second eigen values in absolute value
  - also generate a K which has **small** second eigen values in absolute value
  - in both cases, try to compute the sequence $\{x, Kx, K^2 x \ldots K^t x\}$, using an arbitrary vector x
- **Exercise** Generate K from known eigen value/vectors are called Inverse Eigenvalue Problems. Use IEP to generate stochastic matrices above
- Try something like, "Doubly Stochastic Matrices with Prescribed Positive Spectrum"

5 Metropolis Hasting Algorithm

Le't take a look at the M.H. Algorithm:

1. initialize $x^{(0)}$
2. run:

$$
\begin{aligned}
&\textbf{for } i = 0 \text{ to } N-1 \\
&\quad u \sim U(0,1) \\
&\quad x^* \sim q(x^*|x^{(i)}) \\
&\quad \textbf{if} \quad u < \min\left(1, \frac{\pi(x^*)q(x|x^*)}{\pi(x)q(x^*|x)}\right) \\
&\qquad x^{(i+1)} = x^* \\
&\quad \textbf{else} \\
&\qquad x^{(i+1)} = x^{(i)}
\end{aligned}
\tag{57}
$$

The key message here is that it does not "drop" samples like rejection sampling. It just "duplicates" the sample. If the same sample is repeated too many times, it has **bad mixing**

see demo for an example.

5.1 Metropolis Hasting - Why it work?

$K(x \to x^*)$ includes the joint density of the following:

1. Propose x^* from $q(x^*|x)$,
2. then accept x^* with ratio:

$$
\alpha(x^*, x) = \min\left(1, \frac{\pi(x^*)q(x|x^*)}{\pi(x)q(x^*|x)}\right) \tag{58}
$$

very easily verify it satisfy **detailed balance**:

$$
\begin{aligned}
\pi(x)q(x^*|x)\alpha(x^*, x) &= \pi(x)q(x^*|x)\min\left(1, \frac{\pi(x^*)q(x|x^*)}{\pi(x)q(x^*|x)}\right) \\
&= \min\left(\pi(x)q(x^*|x), \pi(x^*)q(x|x^*)\right) \\
&= \pi(x^*)q(x|x^*)\min\left(1, \frac{\pi(x)q(x^*|x)}{\pi(x^*)q(x|x^*)}\right) \\
&= \pi(x^*)q(x|x^*)\alpha(x, x^*)
\end{aligned}
\tag{59}
$$

note that $\alpha(x^*, x) \neq \alpha(x, x^*)$!

Exercise wait a second, are we missing anything here?

5.2 Metropolis Hasting - Missing the self transition part

1. Case 1: $x^* \neq x$

$$K(x \to x^*) = \alpha(x^*, x) q(x^*|x) \tag{60}$$

detailed balance was already seen in Eq.(59)

2. Case 2: $x^* = x$

$$K(x \to x) = \alpha(x, x) q(x|x) + \int_{x'} q(x'|x)\big(1 - \alpha(x', x)\big)\mathrm{d}x' \tag{61}$$

detailed balance is **trivially** seen, as this is totally symmetrical

5.2.1 something to think about

let $\pi(x) \propto L(x)\pi_{\text{prior}}(x)$:

$$\begin{aligned} & \alpha(x^*, x) = \min\left(1, \frac{\pi(x^*)q(x|x^*)}{\pi(x)q(x^*|x)}\right) \\ \implies & \alpha(x^*, x) = \min\left(1, \underbrace{\frac{\pi_{\text{prior}}(x^*)q(x|x^*)}{\pi_{\text{prior}}(x)q(x^*|x)}}_{\text{may be cheaper to compute}}\right) \min\left(1, \frac{L(x^*)}{L(x)}\right) \end{aligned} \tag{62}$$

what if $q(x^*|x)$ is symmetrical, like Gaussian, i.e., $q(x^*|x) = q(x|x^*)$

5.3 Gibbs sampling

you would like to use Gibbs sampling algorithm to sample:

$$\left\{(x_1, y_1, z_1)^\top, (x_2, y_2, z_2)^\top, (x_3, y_3, z_3)^\top, \ldots, (x_N, y_N, z_N)^\top\right\} \sim P(x, y, z) \tag{63}$$

the Gibbs sampling algorithm is:

1. starting with a sample $(x_1, y_1, z_1)^\top$
2. the algorithm is:

$$\begin{aligned} x_2 &\sim P(x|y_1, z_1) \\ y_2 &\sim P(y|x_2, z_1) \\ z_2 &\sim P(z|x_2, y_2) \\ &\text{---------} \\ x_3 &\sim P(x|y_2, z_2) \\ y_3 &\sim P(y|x_3, z_2) \\ z_3 &\sim P(z|x_3, y_3) \\ &\vdots \end{aligned} \tag{64}$$

3. diagrammatically, it means:

$$
\begin{array}{ll}
\underbrace{\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}}_{\text{sample 1}} \quad \underbrace{\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}}_{\text{sample 2}} & x_2 \sim P(x|y_1, z_1) \\
\underbrace{\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}}_{\text{sample 1}} \quad \underbrace{\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}}_{\text{sample 2}} & y_2 \sim P(y|x_2, z_1) \\
\underbrace{\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}}_{\text{sample 1}} \quad \underbrace{\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}}_{\text{sample 2}} & y_2 \sim P(z|x_2, y_2)
\end{array}
\tag{65}
$$

you can **not** perform $x_2 \sim P(x|y_1, z_1)$ followed by $y_2 \sim P(y|x_1, z_1)$. This is wrong, as it no longer conforms to Eq.(71), i.e., it is **no longer** a special case of M-H, and detailed balance cannot therefore be implied automatically.

5.4 Gibbs sampling Toy Example

In this toy example, let's sample:

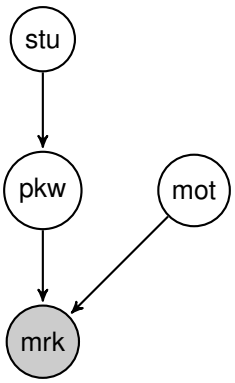
$$
\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right) \tag{66}
$$

$$
\begin{aligned}
x_1|x_2 &\sim \mathcal{N}\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}\left(x_2 - \mu_2\right), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right) \\
x_2|x_1 &\sim \mathcal{N}\left(\mu_2 + \Sigma_{12}\Sigma_{11}^{-1}\left(x_1 - \mu_1\right), \Sigma_{22} - \Sigma_{12}\Sigma_{11}^{-1}\Sigma_{12}\right)
\end{aligned}
\tag{67}
$$

You can find Gaussian conditional easily,

5.5 Gibbs sampling toy example 2

looking at the toy "mark" example again:



Since the posterior is:

$$\Pr\left(\text{stu}, \text{pkw}, \text{mot} | \text{mrk}\right) \tag{68}$$

then the Gibbs Sampling algorithm should be:

$$\begin{aligned} \text{stu}^{(t)} &\sim \Pr(\cdot \mid \text{pkw}^{(t-1)}) \\ \text{pkw}^{(t)} &\sim \Pr(\cdot \mid \text{stu}^{(t)}, \text{mrk}, \text{mot}^{(t-1)}) \\ \text{mot}^{(t)} &\sim \Pr(\cdot \mid \text{pkw}^{(t)}, \text{mrk}) \end{aligned} \tag{69}$$

5.6 Gibbs is a special case of M-H

Why Gibbs sampling achieves detailed balance as per Eq.(35)? Instead of proving it, we can just show it's a special case of M-H. Since we already proved that M-H achieved detailed balance in section 5.1.

Now look at the M-H acceptance ratio Let $\mathbf{x} = x_1, \ldots, x_D$.

When sampling k^{th} component:

$$\begin{aligned} q_k(\mathbf{x}^*|\mathbf{x}) &= \pi(x_k^*|\mathbf{x}_{-k}) \\ \mathbf{x}_{-k}^* &= \mathbf{x}_{-k} \end{aligned} \tag{70}$$

let's take a look at the $\alpha(\mathbf{x}^*, \mathbf{x})$:

$$\frac{\pi(\mathbf{x}^*)q(\mathbf{x}|\mathbf{x}^*)}{\pi(\mathbf{x})q(\mathbf{x}^*|\mathbf{x})} = \frac{\pi(\mathbf{x}^*)\pi(x_k|\mathbf{x}^*{}_{-k})}{\pi(\mathbf{x})\pi(x_k^*|\mathbf{x}_{-k})} = \frac{\pi(x_k^*|\mathbf{x}_{-k}^*)\pi(x_k|\mathbf{x}^*{}_{-k})}{\pi(x_k|\mathbf{x}_{-k})\pi(x_k^*|\mathbf{x}_{-k})} = 1 \tag{71}$$

5.7 Collapsed Gibbs sampling

Treats (x, y) as a single variable

$$\begin{aligned} &(x_2, y_2) \sim P(x, y|z_1) \quad \Longrightarrow\ x_2 \sim p(x|z_1)\ \ y_2 \sim p(y|x_2, z_1) \\ &\quad z_2 \sim P(z|x_2, y_2) \\ &\hline \\ &(x_3, y_3) \sim P(x, y|z_2) \quad \Longrightarrow\ x_3 \sim p(x|z_2)\ \ y_3 \sim p(y|x_3, z_2) \\ &\quad z_3 \sim P(z|x_3, y_3) \\ &\quad \ldots \end{aligned} \tag{72}$$

However, we need to know how to compute:

$$P(x|z) = \int_y P(x, y|z)\mathrm{d}y \tag{73}$$

The algorithm blueuces **auto-correction**.

5.8 What is auto-correction

- lag-k **autocovariance** of the functional $g(X1), g(X2)$

$$\gamma(k) = \text{cov}\big(g(X_i), g(X_{i+k})\big) \tag{74}$$

- lag-k **autocorrelation** of the functional $g(X1), g(X2)$

$$\frac{\gamma(k)}{\gamma(0)} \tag{75}$$

- need to perform **thinning** to make samples more like drawn using i.i.d
- Let's look at an autocorrelation **demo** for computing multivariate Gaussian distribution of having 2-D, ... 5-D.
- **Exercise** what would be an appropriate $g(\cdot)$ used here?
- **Exercise** you need to write a similar code

5.9 Parallel Gibbs sampling

- You can see the algorithm won't "parallelise".
- However, under some models (and clever work-around) machine learning researcher able to parallelise some Gibbs sampling scheme for various models, typically, using

$$p(x_1, x_2, \ldots, x_n) = \int_u p(x_1, x_2, \ldots x_n|u)p(u)\mathrm{d}u \tag{76}$$

and also have the property that:

$$p(x_1, x_2, \ldots x_n|u) = \prod_{i=1}^{n} p(x_i|u) \tag{77}$$

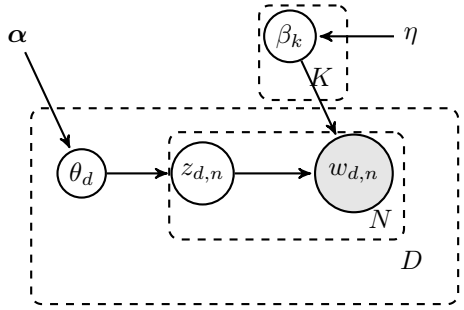
- Well, make sense to perform inference to **Big data** with CUDA, multiple processors.

5.10 Convergence Diagnostics

- The question is when to stop sampling.
- **word of caution**: individual sample do not converge. It's the entire distribution.
- sample will generally be correlated with each other, slowing the algorithm in its attempt to sample from the entire stationary distribution
- run **convergence diagnostics**: *Cowles, M.K.; Carlin, B.P. (1996). "Markov chain Monte Carlo convergence diagnostics: a comparative review". Journal of the American Statistical Association. 91: 883 - 904.*
- or using R Package "coda"

6 Gibbs sampling for Latent Dirichlet Allocation

Remember the graphical model for Latent Dirichlet Allocation [1]?



$$
\begin{aligned}
&\beta_k \sim \text{Dir}(\eta, \dots \eta) \quad \text{for } k \in \{1, ..., K\} \\
&\text{for each doc } d: \\
&\quad \theta \sim \text{Dir}(\alpha, \dots, \alpha) \\
&\quad \text{for each word} \quad w \in \{1, ..., d_N\}: \\
&\qquad z_{dn} \sim \text{Mult}(\theta_d) \\
&\qquad w_{dn} \sim \text{Mult}(\beta_{z_{dn}})
\end{aligned}
\tag{78}
$$

our goal is to sample:

$$
p\left(\{\beta_j\}_{j=1}^K, \{\theta_d\}_{d=1}^D, \{z_{d\in\{1\dots D\}, n\in\{1\dots N\}}\} \middle| \{w_{d\in\{1\dots D\}, n\in\{1\dots N\}}\}\right) \tag{79}
$$

let's see how this can be done by Gibbs sampling

6.1 Basic tools: Multinomial-Dirichlet

Posterior

$$
\begin{aligned}
&P(p_1, \dots, p_k | n_1, \dots, n_k) \\
&\propto \underbrace{\frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^k p_i^{\alpha_i - 1}}_{\text{Dir}(p_1,\dots,p_k|\alpha_1,\dots,\alpha_k)} \underbrace{\frac{n!}{n_1! \dots n_k!} \prod_{i=1}^k p_i^{n_i}}_{\text{Mult}(n_1,\dots,n_k|p_1,\dots p_k)} \\
&\propto \prod_{i=1}^k p_i^{\alpha_i - 1} \prod_{i=1}^k p_i^{n_i} = \prod_{i=1}^k p_i^{\alpha_i - 1 + n_i} \\
&= \text{Dir}(p_1, \dots p_k | \alpha_i + n_i, \dots \alpha_k + n_k)
\end{aligned}
\tag{80}
$$

Marginal

$$
\begin{aligned}
p(n_1, \dots n_k) &= \int_{p_1,\dots,p_k} P(p_1, \dots, p_k, n_1, \dots, n_k) \\
&= \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \frac{n!}{n_1! \dots n_k!} \int_{p_1,\dots,p_k} \prod_{i=1}^k p_i^{\alpha_i - 1 + n_i} \\
&= \frac{N!}{n_1! \dots n_k!} \times \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \times \frac{\prod_{i=1}^k \Gamma(\alpha_i + n_i)}{\Gamma\left(N + \sum_{i=1}^k \alpha_i\right)}
\end{aligned} \tag{81}
$$

6.2 Gibbs sampling for Latent Dirichlet Allocation

The parameters of the model include:

- $\{\beta_k\}_{k=1}^K$ each β_k has dimension V (vocab)
- $\{\theta_d\}_{d=1}^D$
- $\{z_{d\in\{1,\dots,D\}, n\in\{1,\dots,d_N\}}\}$

since everything **conjugate**, posterior inference is easy:

1. for each topic k:

$$
\beta_k \sim \text{Dir}\left(\eta + N_1^{(k)}, \dots, \eta + N_v^{(k)}, \eta + N_V^{(k)}\right) \qquad k \in \{1, \dots, K\} \tag{82}
$$

where $N_v^{(k)} = \#\left(\{w_{dn} = v \text{ AND } z_{dn} = k\}\right)$; In words, this means the number of times the v^{th} word is assigned to the topic k in the entire document corpus:

Let's illustrate this with a scenario where the vocabulary size is $V = 3$ and the number of topics $K = 3$:

d_1	d_2
$(w_{1,1} = 1, z_{1,1} = 1)$	$(w_{2,1} = 2, z_{2,1} = 2)$
$(w_{1,2} = 2, z_{1,2} = 2)$	$(w_{2,2} = 2, z_{2,2} = 2)$
$(w_{1,3} = 2, z_{1,3} = 3)$	$(w_{2,3} = 3, z_{2,3} = 1)$
$(w_{1,4} = 1, z_{1,4} = 1)$	$(w_{2,4} = 3, z_{2,4} = 1)$
$(w_{1,5} = 3, z_{1,5} = 3)$	$(w_{2,5} = 1, z_{2,5} = 3)$
$(w_{1,6} = 2, z_{1,6} = 2)$	$(w_{2,6} = 2, z_{2,6} = 3)$
$(w_{1,7} = 1, z_{1,7} = 2)$	$(w_{2,7} = 1, z_{2,7} = 1)$

let's illustrate just with one example of how to compute $N_2^{(3)}$:

(a) firstly, let's find elements where $v = 2$, i.e, those entries with $w_{i,j} = 2$:

d_1	d_2
$(w_{1,1} = 1, z_{1,1} = 1)$	$(w_{2,1} = 2, z_{2,1} = 2)$
$(w_{1,2} = 2, z_{1,2} = 2)$	$(w_{2,2} = 2, z_{2,2} = 2)$
$(w_{1,3} = 2, z_{1,3} = 3)$	$(w_{2,3} = 3, z_{2,3} = 1)$
$(w_{1,4} = 1, z_{1,4} = 1)$	$(w_{2,4} = 3, z_{2,4} = 1)$
$(w_{1,5} = 3, z_{1,5} = 3)$	$(w_{2,5} = 1, z_{2,5} = 3)$
$(w_{1,6} = 2, z_{1,6} = 2)$	$(w_{2,6} = 2, z_{2,6} = 3)$
$(w_{1,7} = 1, z_{1,7} = 2)$	$(w_{2,7} = 1, z_{2,7} = 1)$

(b) then, you see that there are 2 times for $z_{i,j} = 3$:

d_1	d_2
$(w_{1,1} = 1, z_{1,1} = 1)$	$(w_{2,1} = 2, z_{2,1} = 2)$
$(w_{1,2} = 2, z_{1,2} = 2)$	$(w_{2,2} = 2, z_{2,2} = 2)$
$(w_{1,3} = 2, z_{1,3} = 3)$	$(w_{2,3} = 3, z_{2,3} = 1)$
$(w_{1,4} = 1, z_{1,4} = 1)$	$(w_{2,4} = 3, z_{2,4} = 1)$
$(w_{1,5} = 3, z_{1,5} = 3)$	$(w_{2,5} = 1, z_{2,5} = 3)$
$(w_{1,6} = 2, z_{1,6} = 2)$	$(w_{2,6} = 2, z_{2,6} = 3)$
$(w_{1,7} = 1, z_{1,7} = 2)$	$(w_{2,7} = 1, z_{2,7} = 1)$

then we have:

$$N_2^{(3)} = 2 \tag{83}$$

2. For each document d:

$$\theta_d \sim \text{Dir}(\alpha + M_1^{(d)}, \ldots, \alpha + M_K^{(d)}) \tag{84}$$

where $M_k^{(d)} = \#\{z_{d,n} = k\}$
this can be illustrated using the following table of updating θ_1 from d_1:

d_1
$(w_{1,1} = 1, z_{1,1} = 1)$
$(w_{1,2} = 2, z_{1,2} = 2)$
$(w_{1,3} = 2, z_{1,3} = 3)$
$(w_{1,4} = 1, z_{1,4} = 1)$
$(w_{1,5} = 3, z_{1,5} = 3)$
$(w_{1,6} = 2, z_{1,6} = 2)$
$(w_{1,7} = 1, z_{1,7} = 2)$

we can tell that $M_1^{(1)} = 2$, $M_2^{(1)} = 3$, $M_3^{(1)} = 1$

For each word $z_{dn} \in \{1, ..., d_N\} \quad \forall d \in \{1, \ldots D\}$:

$$\begin{aligned} \Pr(z_{dn} = k) &= \Pr(w_{dn} | z_{dn} = k, \beta_k) p(z_{dn} = k | \theta_d) \\ &\propto \beta_{k, w_{dn}} \theta_{d,k} \end{aligned} \tag{85}$$

6.2.1 looking at Gibbs for LDA more closely

For the Gibbs sampling of each of the set of variables:

1. the first line is condition on ALL rest of the variables
2. the second line is condition only on the Markov Blanket. You need to spend some time to think about why this is so in each case
3. the last line is the actual formula

$$
\begin{aligned}
&p\left(\beta_k \middle| \{\beta_j\}_{j=1,j\neq k}^{K}, \{\theta_d\}_{d=1}^{D}, \{z_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}, \{w_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}\right) \\
&\quad = p\left(\beta_k | \{\theta_d\}_{d=1}^{D}, \{z_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}, \{w_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}\right) \\
&\quad = \text{Dir}(\eta + N_1^{(k)}, \ldots, \eta + N_V^{(k)})
\end{aligned} \tag{86}
$$

$$
\begin{aligned}
&p\left(\theta_d \middle| \{\beta_j\}_{j=1}^{K}, \{\theta_j\}_{j=1,j\neq d}^{D}, \{z_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}, \{w_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}\right) \\
&\quad = p\left(\theta_d | \{z_{d,n\in\{1\ldots N\}}\}\right) \\
&\quad = \text{Dir}(\alpha + M_1^{(d)}, \ldots, \alpha + M_K^{(d)})
\end{aligned} \tag{87}
$$

$$
\begin{aligned}
&p\left(z_{dn} = k \middle| \{\beta_j\}_{j=1}^{K}, \{\theta_d\}_{d=1}^{D}, \{z_{d\in\{1\ldots D\},j\in\{1\ldots N\}j\neq n}\}, \{w_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}\right) \\
&\quad = p\left(z_{dn} = k \middle| \theta_d, w_{d,n}\right) \\
&\quad \propto \beta_{k,w_{dn}} \theta_{d,k}
\end{aligned} \tag{88}
$$

6.3 Collapsed sampling for LDA (Optional)

- we may only interested in sampling $\{z_{d\in\{1\ldots D\},n\in\{1\ldots N\}}\}$
- we could collapse both $\{\beta_j\}_{j=1,j\neq k}^{K}$ and $\{\theta_d\}_{d=1}^{D}$,

$$
p\left(z_{dn} | \mathbf{z}_{-dn}, \mathbf{w}\right) \tag{89}
$$

where $\mathbf{z}_{-dn}$ are all the $\mathbf{z}$ except z_{dn}

$$
\begin{aligned}
&\Pr\left(z_{dn} | \mathbf{z}_{-dn}, \mathbf{w}\right) \\
&\quad \propto \Pr\left(z_{dn}, \mathbf{z}_{-dn}, w_{dn}, \mathbf{w}_{-dn}\right) \\
&\quad = \Pr\left(w_{dn} | z_{dn}, \mathbf{z}_{-dn}, \mathbf{w}_{-dn}\right) \Pr\left(z_{dn}, \mathbf{z}_{-dn}, \mathbf{w}_{-dn}\right) \\
&\quad = \Pr\left(w_{dn} | z_{dn}, \mathbf{z}_{-dn}, \mathbf{w}_{-dn}\right) \Pr\left(z_{dn} | \mathbf{z}_{-dn}, \mathbf{w}_{-dn}\right) \Pr\left(\mathbf{z}_{-dn}, \mathbf{w}_{-dn}\right) \\
&\quad \propto \Pr\left(w_{dn} | z_{dn}, \mathbf{z}_{-dn}, \mathbf{w}_{-dn}\right) \underbrace{\Pr\left(z_{dn} | \mathbf{z}_{-dn}\right)}_{\text{there is no } \mathbf{w}\text{,prior}}
\end{aligned} \tag{90}
$$

- note that, previously, $\Pr\left(w_{dn} | z_{dn}, \mathbf{z}_{-dn}, \mathbf{w}_{-dn}, \beta\right) = \Pr\left(w_{dn} | z_{dn}, \beta_k\right) = \beta_{z_{dn},w_{dn}}$

6.3.1 look at: $p\left(z_{dn} = i | \mathbf{z}_{-dn}\right)$

$$
\Pr\left(z_{dn} | \mathbf{z}_{-dn}, \mathbf{w}\right) \propto p\left(w_{dn} | z_{dn}, \mathbf{z}_{-dn}, \mathbf{w}_{-dn}\right) \underbrace{\Pr\left(z_{dn} | \mathbf{z}_{-dn}\right)} \tag{91}
$$

- Looking at $\Pr\left(z_{dn} = i | \mathbf{z}_{-dn}\right)$ using i instead of loop index k:

$$
\begin{aligned}
&\Pr\left(z_{dn}=i|\mathbf{z}_{-dn}\right)=\int_{\theta_d} p\left(z_{dn}=i,\theta_d|\mathbf{z}_{-dn}\right)\mathrm{d}\theta_d \\
&=\int_{\theta_d}\Pr(z_{dn}=i|\theta_d)p(\theta_d|\mathbf{z}_{-dn})\mathrm{d}\theta_d \\
&\propto\int_{\theta_d}\Pr(z_{dn}=i|\theta_d)\underbrace{\Pr(\mathbf{z}_{-dn}|\theta_d)p(\theta_d)}\,\mathrm{d}\theta_d \\
&=\int_{\theta_d}\mathsf{Mult}(z_{dn}=i|\theta_d)\underbrace{\mathsf{Dir}(\alpha+N_1^{(d)},\ldots,\alpha+N_K^{(d)})}\,\mathrm{d}\theta_d \\
&=\frac{\Gamma\left(\sum_{k=1}^K(\alpha+N_k^{(d)})\right)}{\prod_{k=1}^K\Gamma(\alpha+N_k^{(d)})}\times\frac{\Gamma((\alpha+N_i^{(d)})+1)\left(\prod_{k=1,k\neq i}^K\Gamma((\alpha+N_k^{(d)}))\right)}{\Gamma\left(1+\sum_{k=1}^K(\alpha+N_k^{(d)})\right)} \\
&=\frac{\alpha+N_i^{(d)}}{\sum_{k=1}^K(\alpha+N_k^{(d)})}=\frac{\alpha+N_i^{(d)}}{K\alpha+N^{(d)}}
\end{aligned}
\tag{92}
$$

- $N_i^{(d)}, N^{(d)}$ are counted without z_{dn}, i.e, $N_k^{(\mathrm{d})}=\#\left(\{z_{\widetilde{dn}\neq dn}=i\}\right)$

6.3.2 look at: $p\left(w_{dn}|z_{dn}=i,\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)$

$$
\Pr\left(z_{dn}|\mathbf{z}_{-dn},\mathbf{w}\right)\propto\underbrace{p\left(w_{dn}|z_{dn}=i,\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)}\Pr\left(z_{dn}|\mathbf{z}_{-dn}\right) \tag{93}
$$

- Looking at $\Pr\left(w_{dn}|z_{dn}=i,\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)$ using i instead of loop index k:

$$
\begin{aligned}
&\Pr\left(w_{dn}|z_{dn}=i,\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)=\int_{\beta}\Pr\left(w_{dn},\beta|z_{dn}=i,\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)\mathrm{d}\beta \\
&=\int_{\beta}\Pr\left(w_{dn}|\beta,z_{dn}=i,\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)\underbrace{p\left(\beta,z_{dn}=i|\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)}p\left(\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)\mathrm{d}\beta \\
&\propto\int_{\beta_i}\Pr\left(w_{dn}|\beta,z_{dn}=i\right)\underbrace{p\left(\beta_i|\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)}\mathrm{d}\beta_i \\
&=\int_{\beta_i}\beta_{i,w_{dn}}\underbrace{\mathsf{Dir}(\eta+N_1^{(\mathrm{v})},\ldots,\eta+N_V^{(\mathrm{v})})}\,\mathrm{d}\beta_i
\end{aligned}
\tag{94}
$$

this is just the expectation of $\beta_{i,w_{dn}}$, i.e., the w_{dn}^{th} component of vector β_i

- using expectation of Dirichlet distribution:

$$
\Pr\left(w_{dn}|z_{dn}=i,\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)=\frac{\eta+N_{w_{dn}}^{(\mathrm{v})}}{\sum_{v\in\{1,\ldots V\}}\eta+N_{w_{dn}}^{(\mathrm{v})}}=\frac{\eta+N_{w_{dn}}^{(\mathrm{v})}}{V\eta+N^{(\mathrm{v})}} \tag{95}
$$

where $N_v^{(\mathrm{v})}=\#\left(\{w_{\widetilde{dn}\neq dn}=v\text{ AND }z_{\widetilde{dn}\neq dn}=i\}\right)$

6.3.3 Putting things together

$$\begin{aligned}\Pr\left(z_{dn}=i|\mathbf{z}_{-dn},\mathbf{w}\right) &\propto \Pr\left(w_{dn}|z_{dn}=i,\mathbf{z}_{-dn},\mathbf{w}_{-dn}\right)\Pr\left(z_{dn}=i|\mathbf{z}_{-dn}\right)\\ &= \frac{\eta+N^{(\mathrm{v})}_{w_{dn}}}{V\eta+N^{(\mathrm{v})}}\frac{\alpha+N^{(d)}_{i}}{K\alpha+N^{(d)}}\end{aligned} \tag{96}$$

where:

- $N^{(\mathrm{d})}_{k}=\#\left(\{z_{\widetilde{dn}\neq dn}=i\}\right)$
- $N^{(\mathrm{v})}_{v}=\#\left(\{w_{\widetilde{dn}\neq dn}=v \text{ AND } z_{\widetilde{dn}\neq dn}=i\}\right)$

6.3.4 What about β and θ_d

- **Exercise** think about what you are going to do for β and θ_d when $\mathbf{z}$ are available

References

[1] David M Blei, Andrew Y Ng, and Michael I Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.