Markov Chain Monte Carlo

A/Prof Richard Yi Da Xu
Yida.Xu@uts.edu.au
Wechat: aubedata
https://github.com/roboticcam/machine-learning-notes

University of Technology Sydney (UTS)

June 4, 2018

Metropolis Hasting Algorithm

```
1. initialise $x^{(0)}$
2. for i = 0 to $N-1$
       $u \sim U(0,1)$
       $x^* \sim q(x^*|x^{(i)})$
       if $u < \min\left(1, \frac{\pi(x^*)q(x|x^*)}{\pi(x)q(x^*|x)}\right)$
           $x^{(i+1)} = x^*$
       else
           $x^{(i+1)} = x^{(i)}$
```

- The take-home message here, is that it does not "disgard" samples like rejection sampling. It simply "repeats" samples.
- If the same sample repeats too many times, it has **bad mixing**
- see demo for an example.

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 $\alpha(x^*, x) = \min\left(1, \frac{\pi(x^*)q(x|x^*)}{\pi(x)q(x^*|x)}\right)$
- very easily verify it satisfy **detailed balance**:

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

- note that $\alpha(x^*, x) \neq \alpha(x, x^*)$
- **Exercise** wait a second, are we missing anything here?

Metropolis Hasting - Missing the self transition part

- when $x*$ is accepted, it's accepted on a specific value $\sim q(\cdot)$
- when $x*$ is discarded for a x repeat, $x*$ can be **a range of values** $\sim q(\cdot)$:

$$\begin{cases} p(x^* \neq x) & = \alpha(x) \\ p(x^* = x) & = 1 - \alpha(x) \end{cases}$$

$$\begin{aligned} p(x^* \neq x) = \alpha(x) &= \int_{x^*} p(x^* \neq x | x^*, x) q(x^* | x) \mathrm{d}x^* \\ &= \int_{x^*} \alpha(x^*, x) q(x^* | x) \mathrm{d}x^* \end{aligned}$$

$$p(x^* = x) = 1 - \alpha(x) = 1 - \int_{x^*} \alpha(x^*, x) q(x^* | x) \mathrm{d}x^*$$

$$K(x \to x^*) = q(x^* | x)\alpha(x^*, x) + \underbrace{\underbrace{\delta_x(x^*)}_{=0:\text{when } x \neq x^*} \quad (1 - \alpha(x))}_{=\delta_{x^*}(x)(1-\alpha(x^*))}$$

Two stage acceptance rule

let $\pi(x) \propto L(x)\pi^p(x)$:

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

Hamiltonian Metropolis Hasting (HMC)

- Let Hamiltonian to be $H(q,p)$
- where q is the position and p is the momentum, for each dimension i:

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial q_i}$$

- For HMC, we usually use Hamiltonian function:

$$H(q,p) = U(q) + K(p)$$

- As an example:

$$H(q,p) = U(q) + K(p) = \frac{q^2}{2} + \frac{p^2}{2}$$

- the solution is:

$$q(t) = r\cos(a+t) \qquad p(t) = -r\sin(a+t)$$

Reversibility of Hamiltonian dynamics

- Let a mapping function T_s and its inverse T_{-s}:

$$\begin{aligned} T_s\big(q(t), p(t)\big) &= \big(q(t+s), p(t+s)\big) \\ T_{-s}\big(q(t+s), p(t+s)\big) &= \big(q(t), p(t)\big) \end{aligned}$$

- are invariant under a reversal of the direction of time $t \rightarrow -t$, when q_i and p_i are changed to:

$$q_i \rightarrow q_i \qquad\qquad p_i = \frac{\mathrm{d}q_i}{\mathrm{d}t} \rightarrow \frac{\mathrm{d}q_i}{\mathrm{d}(-t)} = -p_i$$

- this implies:

$$\frac{\mathrm{d}q_i}{\mathrm{d}(-t)} = -\frac{\partial H}{\partial p_i} \qquad\qquad \frac{\mathrm{d}(-p_i)}{\mathrm{d}(-t)} = -\frac{\partial H}{\partial q_i}$$

- or:

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i} \qquad\qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial q_i}$$

- form of equations does not change: rate of change is the reversed.
- at t, evolution is stopped, sign of velocity is reversed
- system is allowed to evolve once again for another time interval t; return to its original starting point

Gibbs sampling

Gibbs sampling algorithm:

- given a starting sample $(x_1, y_1, z_1)^\top$
- you want to sample

$$\left\{(x_2, y_2, z_2)^\top, (x_3, y_3, z_3)^\top, \dots, (x_N, y_N, z_N)^\top\right\} \sim P(x, y, z)$$

- Then the algorithm goes:

$$\begin{aligned} x_2 &\sim P(x|y_1, z_1) \\ y_2 &\sim P(y|x_2, z_1) \\ z_2 &\sim P(z|x_2, y_2) \end{aligned}$$

$$\overline{\hphantom{x_2 \sim P(x|y_1, z_1)}}$$

$$\begin{aligned} x_3 &\sim P(x|y_2, z_2) \\ y_3 &\sim P(y|x_3, z_2) \\ z_3 &\sim P(z|x_3, y_3) \end{aligned}$$

$$\dots$$

Gibbs sampling Toy Example

In this toy example, let's sample:

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

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

A special case of M-H

Looking at the M-H accpetance ratio

- Let $\mathbf{x} = x_1, \ldots, x_D$.
- When sampling k^{th} component, $q_k(\mathbf{x}^*|\mathbf{x}) = \pi(x_k^*|\mathbf{x}_{-k})$
- When sampling k^{th} component, $\mathbf{x}^*_{-k} = \mathbf{x}_{-k}$

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

Collapsed Gibbs sampling

- Treats (x, y) as a single variable

$$\begin{aligned}
(x_2, y_2) &\sim P(x, y|z_1) \qquad \Longrightarrow \; x_2 \sim p(x|z_1) \;\; y_2 \sim p(y|x_2, z_1) \\
z_2 &\sim P(z|x_2, y_2)
\end{aligned}$$

$$\begin{aligned}
(x_3, y_3) &\sim P(x, y|z_2) \qquad \Longrightarrow \; x_3 \sim p(x|z_2) \;\; y_3 \sim p(y|x_3, z_2) \\
z_3 &\sim P(z|x_3, y_3) \\
&\ldots
\end{aligned}$$

- However, we need to know how to compute: $P(x|z) = \int_y P(x, y|z) \mathrm{d}y$
- The algorithm reduces **auto-correction**.

What is auto-correction

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

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

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

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

- 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?
- **Homework** you need to write a similar code

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

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

  and also have the property that:

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

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

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'

Swendesen-Wang

- Potts Model:

$$M(\Pi) \propto \exp\left(\sum_{i<j} \beta_{ij} \mathbf{1}_{z_i = z_j}\right)$$

- Swendsen-Wang algorithm: The joint density between **nodes** Π and edges $r_{ij} \in \{0, 1\}$:

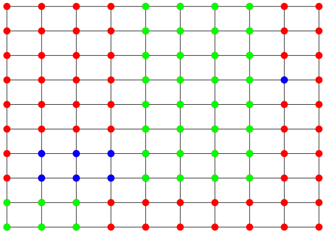
$$P(\Pi, \mathbf{r}) = P(\Pi) p(\mathbf{r}|\Pi)$$

- each edges can be sampled independently:

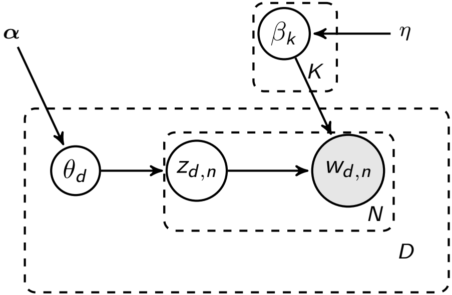
$$P(r_{ij} = 0|\Pi) = \exp(-\beta_{ij} \mathbf{1}_{z_i = z_j}) = q_{ij}$$
$$P(\mathbf{r}|\Pi) = \prod_{1 \leq i < j \leq n} P(r_{ij}|\Pi)$$

- the **trick** is to sample the nodes condition on the edges:

$$P(\Pi|\mathbf{r}) = \prod_{1 \leq i < j \leq n} \left[\exp(\beta_{ij} \mathbf{1}_{z_i = z_j}) - 1\right]^{r_{ij}}$$



A real sampling example: Latent Dirichlet Allocation



- $\beta_k \sim \text{Dir}(\eta, \ldots \eta)$ for $k \in \{1, ..., K\}$.
- For each document d:
  $\theta \sim \text{Dir}(\alpha, \ldots, \alpha)$
  For each word $w \in \{1, ..., N\}$:
    $z_{dn} \sim \text{Mult}(\theta_d)$
    $w_{dn} \sim \text{Mult}(\beta_{z_{dn}})$

Basic tools: Multinomial-Dirichlet

Posterior

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

Marginal

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

Gibbs sampling for Latent Dirichlet Allocation

The parameters of the model include:

- $\{\boldsymbol{\beta}_k\}_{k=1}^K$ each $\boldsymbol{\beta}_k$ has dimension V (vocab)
- $\{\theta_d\}_{d=1}^D$
- $\{z_{d\in\{1\dots D\},n\in\{1\dots N\}}\}$

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

- start with random initial values to all the variables
- $\boldsymbol{\beta}_k \sim \mathrm{Dir}(\eta + N_1^{(\mathrm{v})}, \dots \eta + N_K^{(\mathrm{v})})$ for $k \in \{1, ..., K\}$
  where $N_v^{(\mathrm{v})} = \#(\{w_{dn} = v \text{ AND } z_{dn} = k\})$
- For each document d:
  $\theta_d \sim \mathrm{Dir}(\alpha + N_1^{(d)}, \dots, \alpha + N_K^{(d)})$
  where $N_k^{(\mathrm{d})} = \#(\{z_{dn} = k\})$
  For each word $w \in \{1, ..., N\}$:

$$
\begin{aligned}
\Pr(z_{dn} = k) &= \Pr(w_{dn}|z_{dn}, \boldsymbol{\beta}_k)p(z_{dn}|\theta_d) \\
&\propto \boldsymbol{\beta}_{k,w_{dn}}\theta_d
\end{aligned}
$$

- $\beta_k \sim \mathrm{Dir}(\eta, \dots \eta)$ for $k \in \{1, ..., K\}$.
- For each document d:
  $\theta \sim \mathrm{Dir}(\alpha, \dots, \alpha)$
  For each word $w \in \{1, ..., N\}$:
  $z_{dn} \sim \mathrm{Mult}(\theta_d)$
  $w_{dn} \sim \mathrm{Mult}(\beta_{z_{dn}})$

Exercise and Homework for LDA

- **Exercise** For the Gibbs sampling of each of the set of variables, verify they are true, i.e.,

$$
\begin{aligned}
& p\left(\boldsymbol{\beta}_k | \{\boldsymbol{\beta}_j\}_{j=1, j\neq k}^{K}, \{\theta_d\}_{d=1}^{D}, \{z_{d\in\{1\ldots D\}, n\in\{1\ldots N\}}\}\right) \\
& \qquad = \mathrm{Dir}(\eta + N_1^{(\mathrm{v})}, \ldots \eta + N_V^{(\mathrm{v})}) \\
& p\left(\theta_d | \{\boldsymbol{\beta}_j\}_{j=1, j\neq k}^{K}, \{\theta_j\}_{j=1, j\neq d}^{D}, \{z_{d\in\{1\ldots D\}, n\in\{1\ldots N\}}\}\right) \\
& \qquad = \mathrm{Dir}(\alpha + N_1^{(d)}, \ldots, \alpha + N_K^{(d)}) \\
& p\left(z_{dn} = k | \{\boldsymbol{\beta}_j\}_{j=1, j\neq k}^{K}, \{\theta_d\}_{d=1}^{D}, \{z_{d\in\{1\ldots D\}, j\in\{1\ldots N\} j\neq n}\}\right) \\
& \qquad \propto \boldsymbol{\beta}_{k, w_{dn}} \theta_d
\end{aligned}
$$

- **Homework** generate a set of synthetic values for ale variables and $\{w_{dn}\}$
  - think about the structure for each of the variables
  - caution: MATLAB has no dirichlet generator, what is the alternative?

Collapsed sampling for LDA

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

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

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

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

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

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

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

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

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

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

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

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

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

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

this is just the expectation of $\boldsymbol{\beta}_{i,w_{dn}}$, i.e., the w_{dn}^{th} component of vector $\boldsymbol{\beta}_i$

- using expectation of Dirichlet distribution:

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

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

Putting things together

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

where:

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

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

Slice Sampling - joint density with auxiliary variable and marginal

- given some un-normalised function $f(x)$, where

$$Z = \int f(x) dx \qquad \pi(x) = \frac{f(x)}{Z}$$

- Introduce auxiliary variable u a joint distribution over (x, u) is defined as:

$$\pi(x, u) = \begin{cases} 1/Z & \text{if } 0 < u < f(x) \\ 0 & \text{otherwise} \end{cases}$$

- The joint density of (x, u) is uniform over the region

$$\{(x, u) : 0 < u < f(x)\}$$

- Marginal distribution over x is:

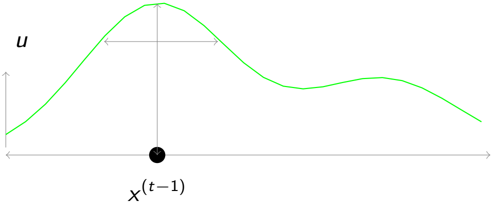
$$\pi(x) = \int_0^{f(x)} \frac{1}{Z} \mathrm{d}u = \frac{f(x)}{Z} = \pi(x)$$

- if $\pi(x) \propto L(x)\pi^p(x)$:

$$u \sim U(0, L(x^{(t-1)})) \qquad \begin{aligned} & x^{(t)} \sim U\left(\mathbf{1}[\pi(x) > u\pi^p(x)]\right) \\ \implies & x^{(t)} \sim U\left(\mathbf{1}[L(x) > u]\right) \end{aligned}$$

Slice Sampling - conditional

- Top-down view of the $\pi(u, x)$ joint density:



- using gibbs sampling:

$$u \sim U(0, \pi(x^{(t-1)})$$

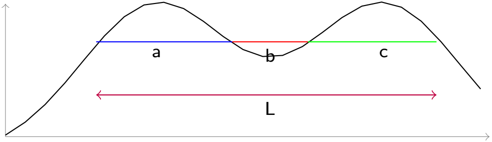
$$x^{(t)} \sim U\left(\mathbf{1}[\pi(.) > u]\right)$$

- Very powerful technique, been working with it in a number of non-parametric Bayes settings.

Slice Sampling - Shrink algorithm

- Usually, its **not** simple to sample $x^{(t)} \sim \mathbf{1}[p(x) > u]$, given $p(x)$ is usually not a concave function.
- We need to use "shrinkage algoirthm" or "expansion algorithm"
- to see why "shrinkage" targets the right distribution:

Shrinkage algorithm



$$\Pr(X^{(t)} \in a \to X^{(t+1)} \in a) = \frac{a+b}{L}$$
$$\Pr(X^{(t)} \in a \to X^{(t+1)} \in c) = \frac{c}{L}$$
$$\Pr(X^{(t)} \in c \to X^{(t+1)} \in a) = \frac{a}{L}$$
$$\Pr(X^{(t)} \in c \to X^{(t+1)} \in c) = \frac{b+c}{L}$$

Therefore,

$$\begin{aligned}\Pr(X^{(t+1)} \in a) =& \Pr(X^{(t)} \in a \to X^{(t+1)} \in a)\Pr(X^{(t)} \in a) + \Pr(X^{(t)} \in c \to X^{(t+1)} \in a)\Pr(X^{(t)} \in c)\\ =& \frac{a+b}{L}\frac{a}{L} + \frac{a}{L}\frac{c}{L} = \frac{a^2+ab+ac}{L^2} = \frac{a}{L} = \Pr(X^{(t)} \in a)\end{aligned}$$

$$\begin{aligned}\Pr(X^{(t+1)} \in c) =& \Pr(X^{(t)} \in a \to X^{(t+1)} \in c)\Pr(X^{(t)} \in a) + \Pr(X^{(t)} \in c \to X^{(t+1)} \in c)\Pr(X^{(t)} \in c)\\ =& \frac{c}{L}\frac{a}{L} + \frac{b+c}{L}\frac{c}{L} = \frac{ac+bc+c^2}{L^2} = \frac{c}{L} = \Pr(X^{(t)} \in c)\end{aligned}$$

$$\Pr(X^{(t+1)}) = \Pr(X^{(t)})$$

Elliptical Slice Sampling

Murray, Iain, and Ryan P. Adams. "Slice sampling covariance hyperparameters of latent Gaussian models.", NIPS 2010

```
1.  choose ellipse: $v \sim \mathcal{N}(0, \Sigma)$
2.  log-likelihood threshold:

        $$u \sim U(0,1)$$
        $$\log(y) = \log(L(x)) + \log(u) \implies y = uL(x)$$

3.  draw an initial proposal, and defining a bracket:

        $$\theta \sim U(0, 2\pi)$$
        $$[\theta_{\min}, \theta_{\max}] = [\theta - 2\pi, \theta]$$

4.  $x^* = x\cos(\theta) + v\sin(\theta)$
5.  **if** $\log(L(x^*)) > \log(y)$, i.e., $L(x^*) > uL(x)$ (this is similar to slice sampling)
6.          **accept**: return $x^*$
7.  **else**
            shrink the bracket - the following procedure only "shrink" one-side:
8.          **if** $\theta < 0$ **then:** $\theta_{\min} = \theta$ **else**: $\theta_{\max} = \theta$
                think about what happens when: $(\theta_{\max} > 0, \theta_{\min} > 0)$, $(\theta_{\max} < 0, \theta_{\min} < 0)$ and
               $(\theta_{\max} > 0, \theta_{\min} < 0)$
            try a new point:
9.          $\theta \sim U(\theta_{\min}, \theta_{\max})$
10.         **Goto** step 4
```

detailed balance

- let's look at again $K(x \to x^*)$ extended to variables u and u:

$$\pi(x)K(x \to x^*)$$

$$= \underbrace{\pi(x)}_{L(x)\mathcal{N}(0,\Sigma)} \underbrace{p(\text{height}|x)}_{\pi(u|x)} \underbrace{p(\text{shape})}_{\pi(v)} \pi(x^*|\text{height}, \text{shape})\mathbf{1}(\mathcal{E}(x,u,v), \mathcal{E}(x^*,u^*,v^*))$$

$$= L(x)\mathcal{N}(x|0,\Sigma) \frac{1}{\underbrace{L(x)}_{\text{height - }\textbf{not }\pi(x)}} \underbrace{\mathcal{N}(v|0,\Sigma)}_{\text{shape}} \underbrace{p(\{\theta_k\}, x^*|\mathcal{E}(x,u,v), x)}_{\text{shrink ellipse to accept}x^*} \mathbf{1}(\mathcal{E}(x,u,v), \mathcal{E}(x^*,u^*,v^*))$$

$$= \mathcal{N}(x|0,\Sigma)\mathcal{N}(v|0,\Sigma)p(\{\theta_k\}, x^*|\mathcal{E}(x,u,v), x)\mathbf{1}(\mathcal{E}(x,u,v), \mathcal{E}(x^*,u^*,v^*))$$

- In order to prove **reversibility**:

$$\pi(x)K(x \to x^*) = \pi(x^*)K(x^* \to x) \implies$$

$$\mathcal{N}(x|0,\Sigma)\mathcal{N}(v|0,\Sigma)p(\{\theta_k\}, x^*|\underbrace{\mathcal{E}(x,u,v)}_{\text{same ellipse}}, x)$$

$$= \mathcal{N}(x^*|0,\Sigma)\mathcal{N}(v^*|0,\Sigma)p(\{\theta'_k\}, x|\underbrace{\mathcal{E}(x^*,u^*,v^*)}_{\text{same ellipse}}, x^*)$$

- here comes a central thing **clever idea**, can we always find a v' such that:
  firstly ellipse(x,u,v) = ellipse (x^*,u^*,v^*)
  secondly $\mathcal{N}(x|0,\Sigma)\mathcal{N}(v|0,\Sigma) = \mathcal{N}(x^*|0,\Sigma)\mathcal{N}(v^*|0,\Sigma)$

“Same-dimension” rotation factor of Gaussian

Let $\begin{bmatrix} x'_1 \\ v'_1 \end{bmatrix} = \mathcal{R}(\theta) \begin{bmatrix} x_1 \\ v_1 \end{bmatrix}$ and $\begin{bmatrix} x'_2 \\ v'_2 \end{bmatrix} = \mathcal{R}(\theta) \begin{bmatrix} x_2 \\ v_2 \end{bmatrix}$

Collectively, we write the above as: $\begin{bmatrix} \mathbf{x}' \\ \mathbf{v}' \end{bmatrix} = \mathcal{R}(\theta) \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix} \implies \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \end{bmatrix} = \mathcal{R}(-\theta) \begin{bmatrix} \mathbf{x}' \\ \mathbf{v}' \end{bmatrix}$

$$\begin{aligned} & \mathcal{N}(\mathbf{x}'|0, \Sigma)\mathcal{N}(\mathbf{v}'|0, \Sigma) \\ = & \exp\left[\frac{-1}{2} \left(\Sigma_{1,1}(x'^2_1 + v'^2_1) + 2\Sigma_{1,2}(x'_1 x'_2 + v'_1 v'_2) + \Sigma_{2,2}(x'^2_2 + v'^2_2) \right) \right] \end{aligned}$$

We know that,

$$\begin{aligned} x'_1 x'_2 + v'_1 v'_2 &= [\cos(\theta)x_1 - \sin(\theta)v_1][\cos(\theta)x_2 - \sin(\theta)v_2] + [\sin(\theta)x_1 + \cos(\theta)v_1][\sin(\theta)x_2 + \cos(\theta)v_2] \\ &= \cos^2(\theta)x_1x_2 - \cos(\theta)\sin(\theta)x_1v_2 - \sin(\theta)\cos(\theta)v_1x_2 + \sin^2(\theta)v_1v_2 \\ &\qquad + \sin^2(\theta)x_1x_2 + \cos(\theta)v_1x_2 + \sin(\theta)x_1v_2 + \cos^2(\theta)v_1v_2 \\ &= x_1x_2 + v_1v_2 \end{aligned}$$

It is then obvious that,

$$\mathcal{N}(\mathbf{x}'|0, \Sigma)\mathcal{N}(\mathbf{v}'|0, \Sigma) = \mathcal{N}(\mathbf{x}|0, \Sigma)\mathcal{N}(\mathbf{v}|0, \Sigma)$$