

# **Personal notes – Bayesian machine learning**

Tuan Anh Le

September 26, 2014

# Contents

<b>1</b>	<b>Notation</b>	<b>5</b>
<b>2</b>	<b>Basics</b>	<b>6</b>
2.1	Probability distributions . . . . .	6
2.1.1	Uniform distribution . . . . .	6
2.1.2	Beta distribution . . . . .	6
2.1.3	Bernoulli distribution . . . . .	6
2.1.4	Binomial distribution . . . . .	6
2.1.5	Beta-binomial distribution . . . . .	6
2.1.6	Categorical distribution . . . . .	6
2.1.7	Dirichlet distribution . . . . .	6
2.1.8	Multinomial distribution . . . . .	6
2.1.9	Pareto distribution . . . . .	6
2.1.10	Gaussian distribution . . . . .	6
2.2	Other . . . . .	7
2.2.1	Kullback-Leibler divergence . . . . .	7
<b>3</b>	<b>Bayesian parameter estimation</b>	<b>8</b>
3.1	Beta-Bernoulli model . . . . .	8
3.1.1	Summary . . . . .	8
3.1.2	Derivations . . . . .	8
3.2	Beta-binomial model . . . . .	8
3.2.1	Summary . . . . .	8
3.2.2	Derivations . . . . .	9
3.3	Dirichlet-categorical model . . . . .	9
3.3.1	Summary . . . . .	9
3.3.2	Derivations . . . . .	10
3.4	Dirichlet-multinomial model . . . . .	10
3.4.1	Summary . . . . .	10
3.4.2	Derivations . . . . .	11
3.5	Poisson-gamma model . . . . .	11
3.5.1	Summary . . . . .	11
3.5.2	Derivations . . . . .	11
<b>4</b>	<b>Advanced models</b>	<b>12</b>
4.1	Mixture models . . . . .	12
4.1.1	EM algorithm . . . . .	13

4.1.2	Gaussian mixture model . . . . .	15
4.2	Hidden Markov model . . . . .	16
4.3	Linear regression . . . . .	16
4.4	Logistic regression . . . . .	16
4.5	Latent Dirichlet allocation . . . . .	16
4.6	Linear dynamical systems . . . . .	16
4.7	Principal components analysis . . . . .	16
4.7.1	Classical PCA . . . . .	16
4.7.2	Probabilistic PCA . . . . .	17
4.8	Factor analysis . . . . .	19
4.9	Independent components analysis . . . . .	19
<b>5</b>	<b>Sampling algorithms</b>	<b>20</b>
5.1	Introduction . . . . .	20
5.2	Rejection sampling . . . . .	20
5.2.1	Why it works? . . . . .	20
5.3	Importance sampling . . . . .	21
5.3.1	Convergence of estimator as $R$ increases . . . . .	22
5.3.2	Optimal proposal distribution . . . . .	22
5.4	Sampling importance resampling . . . . .	23
5.4.1	Why it works? . . . . .	23
5.5	Particle filtering . . . . .	24
5.5.1	Sequential importance sampling (SIS) . . . . .	24
5.5.2	The degeneracy problem . . . . .	26
5.5.3	The resampling step . . . . .	27
5.5.4	Particle filter animation . . . . .	29
5.5.5	The proposal distribution . . . . .	30
5.6	Sequential Monte Carlo . . . . .	31
5.7	Markov chain Monte Carlo methods . . . . .	32
5.7.1	Definitions . . . . .	32
5.7.2	Metropolis Hastings algorithm . . . . .	35
5.7.3	Gibbs sampling . . . . .	36
5.8	Particle Markov Chain Monte Carlo . . . . .	36
5.8.1	Particle independent Metropolis Hastings (PIMH) sampler . . . . .	36
5.8.2	Particle marginal Metropolis Hastings (PMMH) sampler . . . . .	37
5.8.3	Particle Gibbs (PG) sampler . . . . .	39
<b>6</b>	<b>Nonparametric Bayesian models</b>	<b>41</b>
6.1	Gaussian process . . . . .	41
6.2	Dirichlet process . . . . .	41
6.3	Chinese restaurant process . . . . .	41
6.4	Hierarchical Dirichlet process . . . . .	41
6.5	Hierarchical Dirichlet process . . . . .	41
6.6	Indian buffet process . . . . .	41

6.7	Dirichlet diffusion trees . . . . .	41
6.8	Pitman-Yor process . . . . .	41

# 1 Notation

$\{a_n\}$	Same as $\{a_n\}_{n=1}^N$ and $\{a_1, \dots, a_N\}$ – denotes a set of sequence
$\mathbf{x} \in R^D$	$D$ -dimensional real-valued vector
$\sum_k f(\cdot)$	Shorthand for $\sum_{k=1}^K f(\cdot)$ (for an arbitrary index letter)
$\prod_k f(\cdot)$	Shorthand for $\prod_{k=1}^K f(\cdot)$ (for an arbitrary index letter)
$\text{diag}(x_1, \dots, x_N)$	Diagonal matrix formed from the elements $x_1, \dots, x_N$ .

## 2 Basics

### 2.1 Probability distributions

#### 2.1.1 Uniform distribution

#### 2.1.2 Beta distribution

#### 2.1.3 Bernoulli distribution

#### 2.1.4 Binomial distribution

#### 2.1.5 Beta-binomial distribution

#### 2.1.6 Categorical distribution

#### 2.1.7 Dirichlet distribution

#### 2.1.8 Multinomial distribution

#### 2.1.9 Pareto distribution

#### 2.1.10 Gaussian distribution

#### Linear Gaussian model

Given the marginal and conditional distributions to be

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad (2.1)$$

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{Ax} + \mathbf{b}, \mathbf{L}^{-1}) \quad (2.2)$$

the marginal distribution of  $\mathbf{y}$  and the conditional distribution of  $\mathbf{x}$  given  $\mathbf{y}$  are given by

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^T) \quad (2.3)$$

$$p(\mathbf{x} \mid \mathbf{y}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\Sigma} \{ \mathbf{A}^T \mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu} \}, \boldsymbol{\Sigma}) \quad (2.4)$$

where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^T \mathbf{L} \mathbf{A})^{-1} \quad (2.5)$$

#### Why it works

## 2.2 Other

### 2.2.1 Kullback-Leibler divergence

A.k.a. *KL divergence*, or *relative entropy*. KL divergence between the distributions  $p(\mathbf{x})$  and  $q(\mathbf{x})$  for  $\mathbf{x} \in \mathcal{X}$ , denoted  $\text{KL}(p \parallel q)$  or  $\text{KL}(p, q)$ , is a measure of similarity between  $p$  and  $q$  and is given by

$$\begin{aligned}\text{KL}(p \parallel q) &= - \int_{\mathcal{X}} p(\mathbf{x}) \ln q(\mathbf{x}) \, d\mathbf{x} - \left( - \int_{\mathcal{X}} p(\mathbf{x}) \ln p(\mathbf{x}) \, d\mathbf{x} \right) \\ &= - \int_{\mathcal{X}} p(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} \, d\mathbf{x}\end{aligned}\tag{2.6}$$

Note that  $\text{KL}(p \parallel q) \neq \text{KL}(q \parallel p)$ .

**Claim 2.2.1.**  $\text{KL}(p \parallel q) \geq 0$  with equality if and only if  $p(\mathbf{x}) = q(\mathbf{x})$ .

*Proof.* asdf

□

## 3 Bayesian parameter estimation

### 3.1 Beta-Bernoulli model

#### 3.1.1 Summary

The model

$$X_i \sim \text{Ber}(\theta), \text{ for } i \in \{1, \dots, N\} \quad (3.1)$$

$$\mathcal{D} = \{x_1, \dots, x_N\} \quad (3.2)$$

$$N_1 = \sum_{i=1}^N \mathbb{I}(x_i = 1) \quad (3.3)$$

$$N_0 = \sum_{i=1}^N \mathbb{I}(x_i = 0) \quad (3.4)$$

Likelihood

$$p(\mathcal{D}|\theta) = \theta^{N_1} (1 - \theta)^{N_0} \quad (3.5)$$

Prior

$$p(\theta) = \text{Beta}(\theta|a, b) \quad (3.6)$$

Posterior

$$p(\theta|\mathcal{D}) = \text{Beta}(\theta|a' = N_1 + a, b' = N_0 + b) \quad (3.7)$$

Posterior predictive

$$p(\tilde{x} = 1|\mathcal{D}) = \frac{a'}{a' + b'} \quad (3.8)$$

Evidence

#### 3.1.2 Derivations

### 3.2 Beta-binomial model

#### 3.2.1 Summary

The model

$$N_1 \sim \text{Bin}(N, \theta) \quad (3.9)$$



$$\mathcal{D} = \{N_1, N\} \quad (3.10)$$

$$N_1 = \text{number of successes} \quad (3.11)$$

$$N = \text{total number of trials} \quad (3.12)$$

$$\tilde{\mathcal{D}} = \{\tilde{N}_1, \tilde{N}\} \quad (3.13)$$

$$\tilde{N}_1 = \text{number of successes in a new batch of data} \quad (3.14)$$

$$\tilde{N} = \text{total number of trials in a new batch of data} \quad (3.15)$$

#### Likelihood

$$p(\mathcal{D}|\theta) = \text{Bin}(N_1|N, \theta) \quad (3.16)$$

#### Prior

$$p(\theta) = \text{Beta}(\theta|a, b) \quad (3.17)$$

#### Posterior

$$p(\theta|\mathcal{D}) = \text{Beta}(\theta|a' = N_1 + a, b' = N_0 + b) \quad (3.18)$$

#### Posterior predictive

$$p(\tilde{\mathcal{D}}|\mathcal{D}) = \text{Bb}(\tilde{N}_1; a', b', \tilde{N}) \quad (3.19)$$

#### Evidence

### 3.2.2 Derivations

## 3.3 Dirichlet-categorical model

### 3.3.1 Summary

#### The model

$$X_i \sim \text{Cat}(\boldsymbol{\theta} = (\theta_1, \dots, \theta_K)^T), \text{ for } i \in \{1, \dots, N\} \quad (3.20)$$

$$\mathcal{D} = \{x_1, \dots, x_N\} \quad (3.21)$$

$$n_k = \sum_{i=1}^N \mathbb{I}(x_i = k) \quad (3.22)$$

#### Likelihood

$$p(\mathcal{D}|\theta) = \prod_{k=1}^K \theta_k^{n_k} \quad (3.23)$$

#### Prior

$$p(\theta) = \text{Dir}(\boldsymbol{\theta}; \boldsymbol{\alpha}) \quad (3.24)$$

### Posterior

$$p(\theta|\mathcal{D}) = \text{Dir}(\boldsymbol{\theta}; \boldsymbol{\alpha}' = \boldsymbol{\alpha} + (n_1, \dots, n_K)^T) \quad (3.25)$$

### Posterior predictive

$$p(\tilde{X} = j|\mathcal{D}) = \frac{\alpha'_j}{\sum_{k=1}^K \alpha'_k} \quad (3.26)$$

$$= \frac{\alpha_j + n_j}{\alpha_0 + N} \quad (3.27)$$

$$\text{where } \alpha_0 = \sum_{k=1}^K \alpha_k \quad (3.28)$$

### Evidence

#### 3.3.2 Derivations

## 3.4 Dirichlet-multinomial model

### 3.4.1 Summary

#### The model

$$\mathbf{N} \sim \text{Mult}(N, \boldsymbol{\theta}) \in \mathbb{R}^K \quad (3.29)$$

$$\mathcal{D} = \{\mathbf{n} = \text{vector of counts of successes}\} \quad (3.30)$$

$$N = \sum_{i=1}^K n_i \quad (3.31)$$

$$\tilde{\mathcal{D}} = \{\tilde{\mathbf{n}} = \text{vector of counts of successes in a new batch of data}\} \quad (3.32)$$

$$\tilde{N} = \sum_{i=1}^K \tilde{n}_i \quad (3.33)$$

#### Likelihood

$$p(\mathcal{D}|\theta) = \text{Mult}(\mathbf{n}; N, \boldsymbol{\theta}) \quad (3.34)$$

#### Prior

$$p(\theta) = \text{Dir}(\boldsymbol{\theta}; \boldsymbol{\alpha}) \quad (3.35)$$

#### Posterior

$$p(\theta|\mathcal{D}) = \text{Dir}(\boldsymbol{\theta}; \boldsymbol{\alpha}' = \boldsymbol{\alpha} + (n_1, \dots, n_K)^T) \quad (3.36)$$

### Posterior predictive

$$p(\tilde{\mathcal{D}}|\mathcal{D}) = \frac{\Gamma(\alpha_0 + N)}{\Gamma(\alpha_0 + N + \tilde{N})} \prod_{k=1}^K \frac{\Gamma(\alpha_k + n_k + \tilde{n}_k)}{\Gamma(\alpha_k + n_k)} \quad (3.37)$$

$$\text{where } \alpha_0 = \sum_{k=1}^K \alpha_k \quad (3.38)$$

### Evidence

$$p(\mathcal{D}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + N)} \prod_{k=1}^K \frac{\Gamma(\alpha_k + n_k)}{\Gamma(\alpha_k)} \quad (3.39)$$

### 3.4.2 Derivations

## 3.5 Poisson-gamma model

### 3.5.1 Summary

#### The model

$$x \sim \text{Poi}(\lambda) \quad (3.40)$$

$$\mathcal{D} = \{x_1, \dots, x_N\} \quad (3.41)$$

#### Likelihood

$$p(\mathcal{D}|\lambda) = \prod_{i=1}^N \frac{\lambda^{x_i}}{x_i!} \exp(-\lambda) \quad (3.42)$$

#### Prior

$$p(\lambda) = \text{Gamma}(\lambda; a, b) \quad (3.43)$$

#### Posterior

$$p(\lambda|\mathcal{D}) = \text{Gamma}\left(\lambda; a' = a + \sum_{i=1}^N x_i, b' = b + N\right) \quad (3.44)$$

#### Posterior predictive

$$p(\tilde{x}|\mathcal{D}) = \text{NB}\left(\tilde{x}|a', \frac{1}{1+b'}\right) \quad (3.45)$$

#### Evidence

$$p(\mathcal{D}) = \quad (3.46)$$

### 3.5.2 Derivations

## 4 Advanced models

### 4.1 Mixture models

In mixture models, we have discrete latent states  $\mathbf{Z} = \{z_n, z_n \in \{1, \dots, K\}\}, n = 1, \dots, N$  and observed states  $\mathbf{X} = \{\mathbf{x}_n, \mathbf{x}_n \in \mathbb{R}^D\}, n = 1, \dots, N$ . We set the priors and the class conditional likelihoods to be  $p(z_n) = \text{Cat}(\boldsymbol{\pi}), \boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$  and  $p(\mathbf{x}_n | z_n = k; \boldsymbol{\theta}) = p_k(\mathbf{x}_n | \boldsymbol{\theta})$ . We can thus express the likelihood of the observed variables to be:

$$\begin{aligned} p(\mathbf{x}_n | \boldsymbol{\theta}) &= \sum_{k=1}^K p(\mathbf{x}_n, z_n = k; \boldsymbol{\theta}) \\ &= \sum_{k=1}^K p(\mathbf{x}_n | z_n = k; \boldsymbol{\theta}) p(z_n = k | \boldsymbol{\theta}) \\ &= \sum_{k=1}^K \pi_k p_k(\mathbf{x}_n | \boldsymbol{\theta}) \end{aligned} \quad (4.1)$$

We can also express the posterior probability that point  $n$  belongs to cluster  $k$ , or the *responsibility*  $r_{nk}$  of cluster  $k$  for point  $n$  to be:

$$\begin{aligned} r_{nk} &\triangleq p(z_n = k | \mathbf{x}_n; \boldsymbol{\theta}) \\ &= \frac{p(\mathbf{x}_n | z_n = k; \boldsymbol{\theta}) p(z_n = k | \boldsymbol{\theta})}{\sum_{k'=1}^K p(\mathbf{x}_n | z_n = k'; \boldsymbol{\theta}) p(z_n = k' | \boldsymbol{\theta})} \end{aligned} \quad (4.2)$$

Evaluating the above is called *soft clustering*. *Hard clustering* finds the MAP estimate as follows:

$$\begin{aligned} z_n^* &= \arg \max_k r_{nk} \\ &= \arg \max_k \{\log p(\mathbf{x}_n | z_n = k; \boldsymbol{\theta}) + \log p(z_n = k | \boldsymbol{\theta})\} \end{aligned} \quad (4.3)$$

*Unidentifiability* refers to the fact that the posterior distribution for the parameter  $p(\boldsymbol{\theta} | \mathcal{D})$  can be multimodal (with equal peaks) and hence can't find a unique ML/MAP estimate.

We distinguish between two log likelihoods – log likelihood for the observed data, denoted by  $\ell(\boldsymbol{\theta})$  and log likelihood for complete data, denoted by  $\ell_c(\boldsymbol{\theta})$ . These two quantities can be expressed as:

$$\ell(\boldsymbol{\theta}) \triangleq \log p(\mathcal{D} | \boldsymbol{\theta})$$

$$\begin{aligned}
&= \log \prod_{n=1}^N p(\mathbf{x}_n \mid \boldsymbol{\theta}) \\
&= \log \left\{ \prod_{n=1}^N \sum_{k=1}^K p(\mathbf{x}_n, z_n = k \mid \boldsymbol{\theta}) \right\} \\
&= \sum_{n=1}^N \log \sum_{k=1}^K p(\mathbf{x}_n, z_n = k \mid \boldsymbol{\theta}) \tag{4.4} \\
\ell_c(\boldsymbol{\theta}) &\triangleq \log p(\{\mathbf{x}_n, z_n\} \mid \boldsymbol{\theta}) \\
&= \log \prod_n p(\mathbf{x}_n, z_n \mid \boldsymbol{\theta}) \\
&= \sum_n \log p(\mathbf{x}_n, z_n \mid \boldsymbol{\theta}) \tag{4.5}
\end{aligned}$$

The log likelihood for observed data,  $\ell(\boldsymbol{\theta})$  can't be guaranteed to be convex so it might be intractable to find ML/MAP estimates. Alternatively, we just express these terms as  $\ell(\boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\theta})$  and  $\ell_c(\boldsymbol{\theta}) = \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$ .

#### 4.1.1 EM algorithm

##### Maximise the likelihood

Goal is to maximise

$$p(\mathbf{X} \mid \boldsymbol{\theta})$$

Assume it's easy to maximise

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) \tag{4.6}$$

w.r.t.  $\boldsymbol{\theta}$ .

We can express  $\ln p(\mathbf{X} \mid \boldsymbol{\theta})$  as

$$\ln p(\mathbf{X} \mid \boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \text{KL}(q \parallel p) \tag{4.7}$$

where

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{q(\mathbf{Z})} \tag{4.8}$$

$$\text{KL}(q \parallel p) = - \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \tag{4.9}$$

because

$$\text{RHS} = \mathcal{L}(q, \boldsymbol{\theta}) + \text{KL}(q \parallel p)$$

$$\begin{aligned}
&= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})}{q(\mathbf{Z})} - \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \\
&= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})}{p(\mathbf{Z} | \mathbf{X}; \boldsymbol{\theta})} \\
&= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln p(\mathbf{X} | \boldsymbol{\theta}) \\
&= \ln p(\mathbf{X} | \boldsymbol{\theta}) \\
&= \text{LHS}
\end{aligned}$$

The actual algorithm is as follows

---

**Algorithm 1** EM algorithm for maximising the likelihood

---

- 1: Initialise  $\boldsymbol{\theta}^{\text{new}}$ .
  - 2: **repeat**
  - 3:    $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$
  - 4:   E step: Set  $q(\mathbf{Z}) = p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{\text{old}})$ .
  - 5:   M step: Hold  $q(\mathbf{Z})$  fixed and maximise  $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$  w.r.t.  $\boldsymbol{\theta}$  to get  $\boldsymbol{\theta}^{\text{new}}$ .
  - 6: **until** convergence.
- 

**E step.** Hold  $\boldsymbol{\theta}^{\text{old}}$ , maximise  $\mathcal{L}(q, \boldsymbol{\theta}^{\text{old}})$  w.r.t.  $q$ . Since the quantity  $\ln p(\mathbf{X} | \boldsymbol{\theta})$  in (4.7) is constant w.r.t.  $q$ , we can maximise  $\mathcal{L}(q, \boldsymbol{\theta}^{\text{old}})$  by minimising  $\text{KL}(q \| p)$ . This can be done by setting the KL to 0 by setting  $q(\mathbf{Z}) = p(\mathbf{Z} | \mathbf{X}; \boldsymbol{\theta}^{\text{old}})$ .

**M step.** Hold  $q(\mathbf{Z}) = p(\mathbf{Z} | \mathbf{X}; \boldsymbol{\theta}^{\text{old}})$  fixed, maximise  $\mathcal{L}(q, \boldsymbol{\theta})$  w.r.t.  $\boldsymbol{\theta}$  to get  $\boldsymbol{\theta}^{\text{new}}$ . We can rewrite  $\mathcal{L}(q, \boldsymbol{\theta})$  as

$$\begin{aligned}
\mathcal{L}(q, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})}{q(\mathbf{Z})} \\
&= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) - \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln q(\mathbf{Z}) \\
&= \sum_{\mathbf{Z}} p(\mathbf{Z} | \mathbf{X}; \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z} | \mathbf{X}; \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{Z} | \mathbf{X}; \boldsymbol{\theta}^{\text{old}}) \\
&= \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \text{constant w.r.t. } \boldsymbol{\theta}
\end{aligned}$$

from which we can see that we should maximise  $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$ . In both steps, the value of  $\mathcal{L}(q, \boldsymbol{\theta})$  increases.

### Maximising the posterior

Goal is to maximise

$$p(\boldsymbol{\theta} | \mathbf{X})$$

Assume it's easy to maximise

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \ln p(\boldsymbol{\theta}) \quad (4.10)$$

w.r.t.  $\boldsymbol{\theta}$ .

We can express  $\ln p(\boldsymbol{\theta} \mid \mathbf{X})$  as

$$\begin{aligned} \ln p(\boldsymbol{\theta} \mid \mathbf{X}) &= \ln p(\mathbf{X} \mid \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) - \ln p(\mathbf{X}) \\ &= \mathcal{L}(q, \boldsymbol{\theta}) + \text{KL}(q \parallel p) + \ln p(\boldsymbol{\theta}) - \ln p(\mathbf{X}) \end{aligned} \quad (4.11)$$

**E step.** Here, we perform the same thing as in maximising the likelihood, with the same reasons.

**M step.** Hold  $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}; \boldsymbol{\theta}^{\text{old}})$  fixed, maximise  $\mathcal{L}(q, \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})$  w.r.t.  $\boldsymbol{\theta}$  to get  $\boldsymbol{\theta}^{\text{new}}$ . We can rewrite  $\mathcal{L}(q, \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})$  as

$$\mathcal{L}(q, \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) = \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \ln p(\boldsymbol{\theta}) + \text{constant w.r.t. } \boldsymbol{\theta}$$

from which we can see that we should maximise  $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \ln p(\boldsymbol{\theta})$ . In both steps, the value of  $\mathcal{L}(q, \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})$  increases.

The actual algorithm is as follows

---

**Algorithm 2** EM algorithm for maximising the posterior

---

- 1: Initialise  $\boldsymbol{\theta}^{\text{new}}$ .
  - 2: **repeat**
  - 3:      $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$
  - 4:     E step: Set  $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{\text{old}})$ .
  - 5:     M step: Hold  $q(\mathbf{Z})$  fixed and maximise  $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) + \ln p(\boldsymbol{\theta})$  w.r.t.  $\boldsymbol{\theta}$  to get  $\boldsymbol{\theta}^{\text{new}}$ .
  - 6: **until** convergence.
- 

### 4.1.2 Gaussian mixture model

Gaussian mixture model, a.k.a. GMM, or mixture of Gaussians is a mixture model where

$$p_k(\mathbf{x}_n \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (4.12)$$

where  $\boldsymbol{\theta} = (\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}, k = 1, \dots, K)$ .

## 4.2 Hidden Markov model

## 4.3 Linear regression

## 4.4 Logistic regression

## 4.5 Latent Dirichlet allocation

## 4.6 Linear dynamical systems

## 4.7 Principal components analysis

### 4.7.1 Classical PCA

We have data points  $\{\mathbf{x}_n, \mathbf{x}_n \in \mathbb{R}^D\}, n = 1, \dots, N$ . The goal is to project to a lower dimensional space with dimension  $M, M < D$ , while maximising the variance to get data points in the *principal space*,  $\{\mathbf{z}_n, \mathbf{z}_n \in \mathbb{R}^M\}, n = 1, \dots, N$ . Let the *principal components* be  $\{\mathbf{u}_m, \mathbf{u}_m \in \mathbb{R}^D, \|\mathbf{u}_m\| = 1\}, m = 1, \dots, M$ . The projected data can be expressed as

$$\begin{aligned}\mathbf{z}_n &= \begin{bmatrix} \mathbf{u}_1^T \mathbf{x}_n \\ \vdots \\ \mathbf{u}_M^T \mathbf{x}_n \end{bmatrix} \\ &= \mathbf{U}^T \mathbf{x}_n\end{aligned}$$

for  $n = 1, \dots, N$  where  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_M]$ .

The total variance we are trying to maximise, i.e. the sum of variances along the dimensions  $\{\mathbf{u}_m\}$  is

$$\begin{aligned}V &= \sum_{m=1}^M \text{var}(\text{dimension } m) \\ &= \sum_{m=1}^M \frac{1}{N} \sum_{n=1}^N (z_{nm} - \bar{z}_m)^2 \\ &\quad \left( \text{where } \bar{z}_m = \frac{1}{N} \sum_{n=1}^N z_{nm} \right) \tag{4.13} \\ &= \frac{1}{N} \sum_{m=1}^M \sum_{n=1}^N (z_{nm}^2 - 2z_{nm}\bar{z}_m + \bar{z}_m^2) \\ &= \frac{1}{N} \sum_{m=1}^M \sum_{n=1}^N \left( (\mathbf{u}_m^T \mathbf{x}_n)^2 - 2(\mathbf{u}_m^T \mathbf{x}_n)(\mathbf{u}_m^T \bar{\mathbf{x}}) + (\mathbf{u}_m^T \bar{\mathbf{x}})^2 \right), \text{ where } \bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \\ &= \sum_{m=1}^M \mathbf{u}_m^T \left( \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^T - 2\bar{\mathbf{x}} \bar{\mathbf{x}}^T + \bar{\mathbf{x}} \bar{\mathbf{x}}^T \right) \mathbf{u}_m\end{aligned}$$



$$\begin{aligned}
&= \sum_{m=1}^M \mathbf{u}_m^T \left( \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T \right) \mathbf{u}_m \\
&= \sum_{m=1}^M \mathbf{u}_m^T \mathbf{S} \mathbf{u}_m
\end{aligned} \tag{4.14}$$

$$\left( \text{where } \mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T \right) \tag{4.15}$$

We want to maximise this with the constraint  $\|\mathbf{u}_m\| = 1, m = 1, \dots, M$  which is equivalent to  $\mathbf{u}_m^T \mathbf{u}_m = 1, m = 1, \dots, M$ . We use Lagrange multipliers  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_M)$ . Hence we need to maximise the following Lagrangian

$$\mathcal{L}(\boldsymbol{\lambda}, \mathbf{u}_1, \dots, \mathbf{u}_M) = \sum_{m=1}^M \mathbf{u}_m^T \mathbf{S} \mathbf{u}_m + \boldsymbol{\lambda}^T \begin{bmatrix} 1 - \mathbf{u}_1^T \mathbf{u}_1 \\ \vdots \\ 1 - \mathbf{u}_M^T \mathbf{u}_M \end{bmatrix}$$

We know that  $\mathbf{S}$  is positive semi-definite because it is a covariance matrix for  $\{\mathbf{x}_n\}$ . The term  $\mathbf{u}_m^T \mathbf{S} \mathbf{u}_m$  is convex w.r.t.  $\mathbf{u}_m$  because the Hessian  $2\mathbf{S}$  is positive semi-definite. Hence  $\sum_{m=1}^M \mathbf{u}_m^T \mathbf{S} \mathbf{u}_m$  must be convex w.r.t.  $(\mathbf{u}_1, \dots, \mathbf{u}_M)$ . Also, the second term in the Lagrangian is convex w.r.t. the principal components. Hence, we can maximise the Lagrangian by setting the gradients to zero:

$$\text{grad}_{\boldsymbol{\lambda}} \mathcal{L} = \mathbf{0} \tag{4.16}$$

$$\text{grad}_{\mathbf{u}_m} \mathcal{L} = \mathbf{0}, m = 1, \dots, M \tag{4.17}$$

From (4.16), we obtain  $\mathbf{u}_m^T \mathbf{u}_m = 1, m = 1, \dots, M$ . From (4.17), we obtain

$$\text{grad}_{\mathbf{u}_m} \mathcal{L} = 2\mathbf{S} \mathbf{u}_m - 2\lambda_m \mathbf{u}_m \tag{4.18}$$

$$= 0 \tag{4.19}$$

$$\implies \mathbf{S} \mathbf{u}_m = \lambda_m \mathbf{u}_m \tag{4.20}$$

Thus we can see that  $\{\mathbf{u}_m\}$  should be selected to be the eigenvectors corresponding to the eigenvalues  $\{\lambda_m\}$  of  $\mathbf{S}$ . If we premultiply (4.20) by  $\mathbf{u}_m^T$ , we get  $\lambda_m = \mathbf{u}_m^T \mathbf{S} \mathbf{u}_m$  which can be substituted back to total variance

$$V = \sum_{m=1}^M \lambda_m$$

from which we can see that to maximise, we set  $\{\lambda_m\}$  to be the largest  $M$  eigenvalues of  $\mathbf{S}$ . The principal components  $\{\mathbf{u}_m\}$  are the corresponding eigenvectors.

#### 4.7.2 Probabilistic PCA

Following the mixture model, where  $\mathbf{Z} = \{\mathbf{z}_n, \mathbf{z}_n \in \mathbb{R}^M\}$ ,  $n = 1, \dots, N$  are the latent variables and  $\mathbf{X} = \{\mathbf{x}_n, \mathbf{x}_n \in \mathbb{R}^D\}$ ,  $n = 1, \dots, N$  are the observed variables, probabilistic

PCA assumes  $\mathbb{R}^M$  is the lower-dimensional space we want to project our data in  $\mathbb{R}^D$  to. We have the following assumptions:

$$\begin{aligned} p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}) \\ p(\mathbf{x} | \mathbf{z}) &= \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}) \end{aligned}$$

where  $\mathbf{0}, \mathbf{I}, \mathbf{W}, \boldsymbol{\mu}, \mathbf{I}$  all have the appropriate dimensions. Following Subsection 2.1.10, we can express the remaining marginal and conditional as

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C}) \\ p(\mathbf{z} | \mathbf{x}) &= \mathcal{N}(\mathbf{z}; \mathbf{M}^{-1} \mathbf{W}^T (\mathbf{x} - \boldsymbol{\mu}), \sigma^2 \mathbf{M}^{-1}) \end{aligned}$$

where

$$\begin{aligned} \mathbf{C} &= \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I} \\ \mathbf{M} &= \mathbf{W}^T \mathbf{W} + \sigma^2 \mathbf{I} \end{aligned}$$

### MLE for probabilistic PCA

To find ML estimates for our model, we want to maximise the following likelihood function:

$$\begin{aligned} p(\mathcal{D} | \boldsymbol{\theta}) &= \prod_{n=1}^N p(\mathbf{x}_n | \boldsymbol{\theta}) \\ &= \prod_{n=1}^N \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}, \mathbf{C}) \end{aligned}$$

Maximising this w.r.t. the parameters  $\mathbf{W}$  and  $\sigma^2$ , we get the following MLEs:

$$\begin{aligned} \mathbf{W}_{ML} &= \mathbf{U}_M (\mathbf{L}_M - \sigma^2 \mathbf{I})^{1/2} \mathbf{R} \\ \sigma_{ML}^2 &= \frac{1}{D - M} \sum_{i=M+1}^D \lambda_i \end{aligned}$$

where  $\mathbf{R}, \mathbf{R} \in \mathbb{R}^{M \times M}$ ,  $\mathbf{R}\mathbf{R}^T = \mathbf{I}$  is an arbitrary orthogonal matrix and

$$\begin{aligned} \mathbf{U}_M &= [\mathbf{u}_1, \dots, \mathbf{u}_M] \\ \mathbf{L}_M &= \text{diag}(\lambda_1, \dots, \lambda_M) \end{aligned}$$

where  $\mathbf{u}_1, \dots, \mathbf{u}_D$  and  $\lambda_1, \dots, \lambda_D$  are eigenvectors and eigenvalues of the data covariance matrix  $\mathbf{S}$  (defined below in (4.15)), sorted in descending order.

### Other stuff to note

**Alternative view.** fdsaf a

**Intuitive view.** fsda

**Redundancy in parameterisation.** f ds

**Computational complexity.** fsdaf

**EM algorithm for probabilistic PCA**

**Bayesian PCA**

## **4.8 Factor analysis**

## **4.9 Independent components analysis**

# 5 Sampling algorithms

## 5.1 Introduction

Let  $p$  be a probability distribution with a pdf  $p(\mathbf{x})$ ,  $\mathbf{x} \in \mathcal{X}$  (usually  $\mathcal{X} = \mathbb{R}^D$ ,  $D \in \mathbb{N}$ ), which we assume can be evaluated within a multiplicative factor (i.e. we can only evaluate  $p^*(\mathbf{x}) = Z_p p(\mathbf{x})$ , where  $Z_p = \int_{\mathcal{X}} p^*(\mathbf{x}) d\mathbf{x}$ ). We want to achieve the following:

**Problem 1** Generate samples  $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(R)}\}$ ,  $R \in \mathbb{N}$  (we will use the shorthand notation  $\{\mathbf{x}^{(r)}\}$  from now) from the probability distribution  $p$ .

**Problem 2** Estimate the expectation of an arbitrary function  $f$  given  $\mathbf{x} \sim p$ ,  $E_{\mathbf{x} \sim p} [f(\mathbf{x})]$  (we will use the shorthand notation  $E[f]$  from now).

## 5.2 Rejection sampling

Assume we can sample from a proposal distribution  $q$  with a pdf  $q(\mathbf{x})$ , which can be evaluated within a multiplicative factor (i.e. we can only evaluate  $q^*(\mathbf{x}) = Z_q q(\mathbf{x})$ ). Also assume we know the value of a constant  $c$  such that

$$cq^*(\mathbf{x}) > p^*(\mathbf{x}) \text{ for all } \mathbf{x} \quad (5.1)$$

The procedure that generates a sample  $\mathbf{x} \sim p$  is described in Algorithm 3 below.

---

### Algorithm 3 Rejection sampling

---

- 1: Generate  $\mathbf{x} \sim q$ .
  - 2: Generate  $u \sim \text{Unif}(0, cq^*(\mathbf{x}))$ .
  - 3: If  $u > p^*(\mathbf{x})$  it is rejected, otherwise it is accepted.
- 

### 5.2.1 Why it works?

Assume  $\mathbf{x} \in \mathbb{R}^D$ . Define sets  $\mathcal{X}$  and  $\mathcal{X}'$  to be

$$\mathcal{X} = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^{d+1} : \alpha_{1:d} \in \mathbb{R}^d, \alpha_{d+1} \in [0, cq^*(\boldsymbol{\alpha})] \right\} \quad (5.2)$$

$$\mathcal{X}' = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^{d+1} : \alpha_{1:d} \in \mathbb{R}^d, \alpha_{d+1} \in [0, p^*(\boldsymbol{\alpha})] \right\} \quad (5.3)$$

Note that  $\mathcal{X}' \subseteq \mathcal{X}$ .

By definition,  $\mathcal{X}$  is the support of  $(\mathbf{x}, u)$ . The probability of  $(\mathbf{x}, u)$  can be expressed as

$$\Pr(\mathbf{x}, u) = \Pr(\mathbf{x}) \Pr(u) \quad (5.4)$$

$$= q(\mathbf{x}) \frac{1}{cq^*(\mathbf{x})} \quad (5.5)$$

$$= q(\mathbf{x}) \frac{1}{cZ_q q(\mathbf{x})} \quad (5.6)$$

$$= \frac{1}{cZ_q} \quad (5.7)$$

which is constant w.r.t.  $(\mathbf{x}, u)$ , i.e.

$$(\mathbf{x}, u) \sim \text{Unif}(\mathcal{X}) \quad (5.8)$$

Let  $(\mathbf{x}', u')$  be the value of  $(\mathbf{x}, u)$  that gets accepted. By definition,  $\mathcal{X}'$  is the support of  $(\mathbf{x}', u')$ :

$$(\mathbf{x}', u') = \begin{cases} (\mathbf{x}, u) & \text{if } (\mathbf{x}, u) \in \mathcal{X}' \\ \text{nothing} & \text{otherwise.} \end{cases} \quad (5.9)$$

The probability of  $(\mathbf{x}', u')$  can be expressed as

$$\Pr(\mathbf{x}', u') = \begin{cases} \Pr(\mathbf{x}, u) & \text{if } (\mathbf{x}, u) \in \mathcal{X}' \\ 0 & \text{otherwise.} \end{cases} \quad (5.10)$$

which means

$$(\mathbf{x}', u') \sim \text{Unif}(\mathcal{X}') \quad (5.11)$$

Working backwards

$$\Pr(\mathbf{x}') = \frac{\Pr(\mathbf{x}', u')}{\Pr(u')} \quad (5.12)$$

$$\propto \frac{1}{1/p^*(\mathbf{x}')} \quad (5.13)$$

$$\propto p^*(\mathbf{x}') \quad (5.14)$$

Hence the accepted  $\mathbf{x}, \mathbf{x}'$  is  $\sim p$ .

### 5.3 Importance sampling

Assume we can sample from a proposal distribution  $q$  with a pdf  $q(\mathbf{x})$ , which can be evaluated within a multiplicative factor (i.e. we can only evaluate  $q^*(\mathbf{x}) = Z_q q(\mathbf{x})$ ). To solve problem 2, we follow Algorithm 4 below.

---

**Algorithm 4** Importance sampling

---

- 1: Generate samples from  $q$ ,  $\{\mathbf{x}^{(r)}\}$ .
  - 2: Calculate importance weights  $w_r = \frac{p^*(\mathbf{x}^{(r)})}{q^*(\mathbf{x}^{(r)})}$ .
  - 3:  $\hat{\mathbf{y}} = \frac{\sum_r w_r f(\mathbf{x}^{(r)})}{\sum_r w_r}$  is the estimator of  $E[f]$ .
-

### 5.3.1 Convergence of estimator as $R$ increases

We want to prove that if  $q(\mathbf{x})$  is non-zero for all  $\mathbf{x}$  where  $p(\mathbf{x})$  is non-zero, the estimator  $\hat{\mathbf{y}}$  converges to  $E[f]$ , as  $R$  increases. We consider the the expectations of the numerator and denominator separately:

$$E_q[\text{numer}] = E_q \left[ \sum_r w_r f(\mathbf{x}^{(r)}) \right] \quad (5.15)$$

$$= \sum_r E_q \left[ w_r f(\mathbf{x}^{(r)}) \right] \quad (5.16)$$

$$= \sum_r E_q \left[ \frac{p^*(\mathbf{x}^{(r)})}{q^*(\mathbf{x}^{(r)})} f(\mathbf{x}^{(r)}) \right] \quad (5.17)$$

$$= \sum_r E_q \left[ \frac{Z_p p(\mathbf{x}^{(r)})}{Z_q q(\mathbf{x}^{(r)})} f(\mathbf{x}^{(r)}) \right] \quad (5.18)$$

$$= \frac{Z_p}{Z_q} \sum_r \int_{\mathbf{x}^{(r)}} p(\mathbf{x}^{(r)}) f(\mathbf{x}^{(r)}) d\mathbf{x}^{(r)} \quad (5.19)$$

$$= \frac{Z_p}{Z_q} \sum_r E_p \left[ f(\mathbf{x}^{(r)}) \right] \quad (5.20)$$

$$= \frac{Z_p}{Z_q} R E_p [f(\mathbf{x})] \quad (5.21)$$

$$E_q[\text{denom}] = E_q \left[ \sum_r w_r \right] \quad (5.22)$$

$$= \sum_r E_q \left[ \frac{p^*(\mathbf{x}^{(r)})}{q^*(\mathbf{x}^{(r)})} \right] \quad (5.23)$$

$$= \sum_r E_q \left[ \frac{Z_p p(\mathbf{x}^{(r)})}{Z_q q(\mathbf{x}^{(r)})} \right] \quad (5.24)$$

$$= \frac{Z_p}{Z_q} \sum_r \int_{\mathbf{x}^{(r)}} p(\mathbf{x}^{(r)}) d\mathbf{x}^{(r)} \quad (5.25)$$

$$= \frac{Z_p}{Z_q} R \quad (5.26)$$

Hence  $\hat{\mathbf{y}}$  converges to  $E_p[f]$  as  $R$  increases (but is not necessarily an unbiased estimator because  $E_q[\hat{\mathbf{y}}]$  is not necessarily  $= E_p[f]$ ).

### 5.3.2 Optimal proposal distribution

Assuming we can evaluate  $p(\mathbf{x})$  and  $q(\mathbf{x})$ , we want to find a proposal distribution  $q$  to minimise the variance of the weighted samples

$$\text{var}_q \left[ \frac{p(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x}) \right] = E_q \left[ \frac{p^2(\mathbf{x})}{q^2(\mathbf{x})} f^2(\mathbf{x}) \right] - \left( E_q \left[ \frac{p(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x}) \right] \right)^2 \quad (5.27)$$

$$= \mathbb{E}_q \left[ \frac{p^2(\mathbf{x})}{q^2(\mathbf{x})} f^2(\mathbf{x}) \right] - (\mathbb{E}_p [f(\mathbf{x})])^2 \quad (5.28)$$

The second part is independent of  $q$  so we can ignore it. By Jensen's inequality, we have  $\mathbb{E}[g(u(\mathbf{x}))] \geq g(\mathbb{E}[u(\mathbf{x})])$  for  $u(\mathbf{x}) \geq 0$  where  $g : x \mapsto x^2$ . Setting  $u(\mathbf{x}) = p(\mathbf{x})|f(\mathbf{x})|/q(\mathbf{x})$ , we have the following lower bound:

$$\mathbb{E}_q \left[ \frac{p^2(\mathbf{x})}{q^2(\mathbf{x})} f^2(\mathbf{x}) \right] \geq \left( \mathbb{E}_q \left[ \frac{p(\mathbf{x})}{q(\mathbf{x})} |f(\mathbf{x})| \right] \right)^2 = (\mathbb{E}_p [|f(\mathbf{x})|])^2 \quad (5.29)$$

with the equality when  $u(\mathbf{x}) = \text{const.} \implies q_{\text{optimal}}(\mathbf{x}) \propto |f(\mathbf{x})|p(\mathbf{x})$ . Taking care of normalisation, we get

$$q_{\text{optimal}}(\mathbf{x}) = \frac{|f(\mathbf{x})|p(\mathbf{x})}{\int |f(\mathbf{x}')|p(\mathbf{x}') d\mathbf{x}'} \quad (5.30)$$

## 5.4 Sampling importance resampling

In Sampling importance resampling (SIR), we approximate the pdf of  $p$  as point masses and resample from them to get samples approximately  $\sim p$ . The process is described in Algorithm 5 below.

---

### Algorithm 5 Sampling importance resampling

---

- 1: Generate samples  $\{\mathbf{x}^{(r)}\}$  from  $q$ .
- 2: Calculate importance weights  $\left\{w_r = \frac{p^*(\mathbf{z}^{(r)})}{q^*(\mathbf{z}^{(r)})}\right\}$ .
- 3: Calculate the normalised importance weights  $\left\{\hat{w}_r = \frac{w_r}{\sum_{r'} w_{r'}}\right\}$ . Note that  $\sum_r \hat{w}_r = 1$ .
- 4: We can resample from

$$\hat{p}(d\mathbf{x}) = \sum_r \hat{w}_r \delta_{\mathbf{x}^{(r)}}(d\mathbf{x}) \quad (5.31)$$

to estimate sampling from  $p(\mathbf{x})$ .

---

### 5.4.1 Why it works?

We consider the univariate case (to do: general case) as the number of proposal samples (particles)  $R \rightarrow \infty$ . We can express the number of proposal samples that are in the interval  $\lim_{\delta x \rightarrow 0} [x, x + \delta x]$ ,  $N(x)$ , to be

$$N(x) = \lim_{\delta x \rightarrow 0} Rq(x)\delta x \quad (5.32)$$

We can express the probability of the one final sample,  $x^{(r)}$  being in the interval  $\lim_{\delta x \rightarrow 0} [x, x + \delta x]$  to be

$$\lim_{\delta x \rightarrow 0} \Pr(x \leq x^{(r)} \leq x + \delta x) = N(x)\hat{w}_r \quad (5.33)$$

$$\propto \lim_{\delta x \rightarrow 0} Rq(x)\delta x \frac{p(x)}{q(x)} \quad (5.34)$$

$$\propto \lim_{\delta x \rightarrow 0} p(x) \delta x \quad (5.35)$$

Hence (to do: why exactly does that result in an integral)

$$\Pr(a \leq x^{(r)} \leq b) \propto \int_a^b p(x) dx \quad (5.36)$$

$$\implies x^{(r)} \sim p \quad (5.37)$$

## 5.5 Particle filtering

### 5.5.1 Sequential importance sampling (SIS)

Assume the probabilistic graphical model similar to the one in HMMs, where

- $\mathbf{x}_t, \mathbf{x}_t \in \mathcal{X}^D$  and  $\mathbf{y}_t, \mathbf{y}_t \in \mathcal{Y}^D$  are the hidden and observed random variables at time  $t, t = 1, \dots, T$ .
- The initial state is characterised by  $\mathbf{x}_1 \sim \mu(\cdot | \boldsymbol{\theta})$  for some known parameter  $\boldsymbol{\theta} \in \Theta$ .
- The transitions are characterised by  $\mathbf{x}_t | \mathbf{x}_{t-1} \sim f(\cdot | \mathbf{x}_{t-1}; \boldsymbol{\theta})$ .
- The emissions are characterised by  $\mathbf{y}_t | \mathbf{x}_t \sim g(\cdot | \mathbf{x}_t; \boldsymbol{\theta})$ .

We want to sample from the distribution  $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}; \boldsymbol{\theta})$ . Assume we can sample from the probability distribution with the pdf of the following form

$$q(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}; \boldsymbol{\theta}) = q(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}; \boldsymbol{\theta}) q(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t}; \boldsymbol{\theta}) \quad (5.38)$$

$$= q(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}; \boldsymbol{\theta}) q(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}; \boldsymbol{\theta}) \quad (5.39)$$

$$= q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t; \boldsymbol{\theta}) \quad (5.40)$$

If we express the pdf of  $p$  for  $t = 1, \dots, T$  in the form of (for convenience, we drop the conditional dependency on  $\boldsymbol{\theta}$ ):

$$p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t} | \mathbf{x}_{1:t}) p(\mathbf{x}_{1:t})}{p(\mathbf{y}_{1:t})} \quad (5.41)$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1}) p(\mathbf{y}_{1:t-1} | \mathbf{x}_{1:t}) p(\mathbf{x}_{1:t})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1}) p(\mathbf{y}_{1:t-1})} \quad (5.42)$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1}) p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1})} \quad (5.43)$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1}) p(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}) p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1})} \quad (5.44)$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1})} \quad (5.45)$$

$$\propto p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) \quad (5.46)$$



$$= g(\mathbf{y}_t | \mathbf{x}_t) f(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) \quad (5.47)$$

we can write the weight of the sample  $\mathbf{x}_{1:t}^{(r)}$  from the proposal  $q$  to be

$$w_t^{(r)} \propto \frac{p(\mathbf{x}_{1:t}^{(r)} | \mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t}^{(r)} | \mathbf{y}_{1:t})} \quad (5.48)$$

$$\propto \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(r)}) p(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}) p(\mathbf{x}_{1:t-1}^{(r)} | \mathbf{y}_{1:t-1})}{q(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t) q(\mathbf{x}_{1:t-1}^{(r)} | \mathbf{y}_{1:t-1})} \quad (5.49)$$

$$= w_{t-1}^{(r)} \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(r)}) p(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)})}{q(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t)} \quad (5.50)$$

$$= w_{t-1}^{(r)} \frac{g(\mathbf{y}_t | \mathbf{x}_t^{(r)}) f(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)})}{q(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t)} \quad (5.51)$$

For  $t = 1$

$$w_1^{(r)} \propto \frac{p(\mathbf{x}_1^{(r)} | \mathbf{y}_1)}{q(\mathbf{x}_1^{(r)} | \mathbf{y}_1)} \quad (5.52)$$

$$\propto \frac{p(\mathbf{x}_1^{(r)}, \mathbf{y}_1)}{q(\mathbf{x}_1^{(r)} | \mathbf{y}_1)} \quad (5.53)$$

$$\propto \frac{p(\mathbf{y}_1 | \mathbf{x}_1^{(r)}) p(\mathbf{x}_1^{(r)})}{q(\mathbf{x}_1^{(r)} | \mathbf{y}_1)} \quad (5.54)$$

$$= \frac{g(\mathbf{y}_1 | \mathbf{x}_1^{(r)}) \mu(\mathbf{x}_1^{(r)})}{q(\mathbf{x}_1^{(r)} | \mathbf{y}_1)} \quad (5.55)$$

Note that second line is proportional to the first line with respect to  $p(\mathbf{y}_1)$  which is justifiable because the the constant of proportionality cancels out during the normalisation step. The algorithm for SIS is shown in Algorithm 6 below.

---

**Algorithm 6** Sequential importance sampling

---

1: Sample from proposal ▷ Initialisation

$$\mathbf{x}_1^{(r)} \sim q(\cdot | \mathbf{y}_1^{(r)}, \boldsymbol{\theta}), r = 1, \dots, R \quad (5.56)$$

2: Compute weights

$$w_1^{(r)} \propto \frac{g(\mathbf{y}_1 | \mathbf{x}_1^{(r)}) \mu(\mathbf{x}_1^{(r)})}{q(\mathbf{x}_1^{(r)} | \mathbf{y}_1)}, r = 1, \dots, R \quad (5.57)$$

3: Normalise weights

$$\hat{w}_1^{(r)} = \frac{w_1^{(r)}}{\sum_{r'} w_1^{(r')}}, r = 1, \dots, R \quad (5.58)$$

4: We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_1 \mid \mathbf{y}_1, \boldsymbol{\theta}) = \sum_r \hat{w}_1^{(r)} \delta_{\mathbf{x}_1^{(r)}}(\mathrm{d}\mathbf{x}_1) \quad (5.59)$$

to estimate

$$p(\mathbf{x}_1 \mid \mathbf{y}_1, \boldsymbol{\theta}) \quad (5.60)$$

5: **for**  $t = 2, \dots, T$  **do**

▷ Main loop

6:     Compute weights

$$w_t^{(r)} \propto w_{t-1}^{(r)} \frac{g(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}, \boldsymbol{\theta}) f(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \boldsymbol{\theta})}{q(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \mathbf{y}_t, \boldsymbol{\theta})}, r = 1, \dots, R \quad (5.61)$$

7:     Normalise weights

$$\hat{w}_t^{(r)} = \frac{w_t^{(r)}}{\sum_{r'} w_t^{(r')}}, r = 1, \dots, R \quad (5.62)$$

8:     We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) = \sum_r \hat{w}_t^{(r)} \delta_{\mathbf{x}_{1:t}^{(r)}}(\mathrm{d}\mathbf{x}_{1:t}) \quad (5.63)$$

to estimate

$$p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) \quad (5.64)$$

The reason why it works is the same as in the case of Sampling importance resampling described in section 5.4.

### 5.5.2 The degeneracy problem

Because the support of the pdf we are approximating ( $p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$ ) is growing, the constant number of weights we use ( $R$ ) won't be sufficient after a while. This is because many weights will become very negligible, wasting our resources. An **effective sample size** is used to measure this degeneracy is defined to be and approximated by the following:

$$S_{\text{eff}} \triangleq \frac{S}{1 + \text{var} \left[ w_t^{(r)*} \right]} \quad (5.65)$$

$$\hat{S}_{\text{eff}} \approx \frac{1}{\sum_r \left( w_t^{(r)} \right)^2} \quad (5.66)$$

where  $w_t^{(r)*} = p(\mathbf{x}_t^{(r)} | \mathbf{y}_{1:t}) / q(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t)$  is the “true weight” of particle  $r$ .

There are (among others) two solutions to this problem – introduce the resampling step, and using a good proposal distribution.

### 5.5.3 The resampling step

Whenever the effective sample size drops below some threshold, resample to get new  $R$  samples from the approximation of the pdf. This step is also called **rejuvenation**. The full algorithm for a generic particle filter is shown in Algorithm 7 below in which we resample during every tie step.

---

#### Algorithm 7 Generic particle filter

---

- 1: Sample from proposal ▷ Initialisation

$$\mathbf{x}_1^{(r)} \sim q(\cdot | \mathbf{y}_1, \boldsymbol{\theta}), r = 1, \dots, R \quad (5.67)$$

- 2: Compute weights

$$w_1^{(r)} \propto \frac{p(\mathbf{x}_1^{(r)} | \mathbf{y}_1, \boldsymbol{\theta})}{q(\mathbf{x}_1^{(r)} | \mathbf{y}_1, \boldsymbol{\theta})}, r = 1, \dots, R \quad (5.68)$$

- 3: Normalise weights

$$\hat{w}_1^{(r)} = \frac{w_1^{(r)}}{\sum_{r'} w_1^{(r')}}, r = 1, \dots, R \quad (5.69)$$

- 4: We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_1 | \mathbf{y}_1, \boldsymbol{\theta}) = \sum_r \hat{w}_1^{(r)} \delta_{\mathbf{x}_1^{(r)}}(\mathrm{d}\mathbf{x}_1) \quad (5.70)$$

to estimate

$$p(\mathbf{x}_1 | \mathbf{y}_1, \boldsymbol{\theta}) \quad (5.71)$$

- 5: **for**  $t = 2, \dots, T$  **do** ▷ Main loop

- 6:     Sample parents' indices of  $t^{\text{th}}$  generation

$$A_{t-1}^{(r)} \sim \text{Cat}(\hat{w}_{t-1}), r = 1, \dots, R \quad (5.72)$$

- 7:     Sample  $t^{\text{th}}$  generation using corresponding parents

$$\mathbf{x}_t^{(r)} \sim q(\cdot | \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \mathbf{y}_t, \boldsymbol{\theta}), r = 1, \dots, R \quad (5.73)$$

- 8:     Compute weights

$$w_t^{(r)} \propto w_{t-1}^{(r)} \frac{g(\mathbf{y}_t | \mathbf{x}_t^{(r)}, \boldsymbol{\theta}) f(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \boldsymbol{\theta})}{q(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \mathbf{y}_t, \boldsymbol{\theta})}, r = 1, \dots, R \quad (5.74)$$

9:     Normalise weights

$$\hat{w}_t^{(r)} = \frac{w_t^{(r)}}{\sum_{r'} w_t^{(r')}}, r = 1, \dots, R \quad (5.75)$$

10:    We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) = \sum_r \hat{w}_t^{(r)} \delta_{\mathbf{x}_{1:t}^{(r)}}(\mathrm{d}\mathbf{x}_{1:t}) \quad (5.76)$$

to estimate

$$p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) \quad (5.77)$$

---

#### **5.5.4 Particle filter animation**

### 5.5.5 The proposal distribution

It is common to use the following proposal distribution

$$q(\mathbf{x}_{1:t}^{(r)} | \mathbf{y}_{1:t}) = q(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t) \quad (5.78)$$

$$= p(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}) \quad (5.79)$$

$$= f(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}) \quad (5.80)$$

Hence the weight equation in (5.51) becomes

$$w_t^{(r)} \propto w_{t-1}^{(r)} \frac{g(\mathbf{y}_t | \mathbf{x}_t^{(r)}) f(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)})}{q(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t)} \quad (5.81)$$

$$= w_{t-1}^{(r)} g(\mathbf{y}_t | \mathbf{x}_t^{(r)}) \quad (5.82)$$

This approach can be inefficient because the likelihood,  $p(\mathbf{y}_t | \mathbf{x}_t^{(r)})$ , can be very small at many places meaning many of the particles will be very small.

The optimal proposal distribution has the form

$$q(\mathbf{x}_{1:t}^{(r)} | \mathbf{y}_{1:t}) = q(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t) \quad (5.83)$$

$$= p(\mathbf{x}_t^{(r)} | \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t) \quad (5.84)$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{x}_{t-1}^{(r)}) p(\mathbf{x}_t, \mathbf{x}_{t-1}^{(r)})}{p(\mathbf{x}_{t-1}^{(r)} | \mathbf{y}_t)} \quad (5.85)$$

$$= \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(r)})}{p(\mathbf{y}_t | \mathbf{x}_{t-1}^{(r)})} \quad (5.86)$$

$$= \frac{g(\mathbf{y}_t | \mathbf{x}_t) f(\mathbf{x}_t | \mathbf{x}_{t-1}^{(r)})}{p(\mathbf{y}_t | \mathbf{x}_{t-1}^{(r)})} \quad (5.87)$$

The weight equation in (5.51) becomes

$$w_t^{(r)} \propto w_{t-1}^{(r)} p(\mathbf{y}_t | \mathbf{x}_{t-1}^{(r)}) \quad (5.88)$$

$$= w_{t-1}^{(r)} \int p(\mathbf{y}_t, \mathbf{x}'_t | \mathbf{x}_{t-1}^{(r)}) d\mathbf{x}' \quad (5.89)$$

$$= w_{t-1}^{(r)} \int p(\mathbf{y}_t | \mathbf{x}'_t, \mathbf{x}_{t-1}^{(r)}) p(\mathbf{x}'_t | \mathbf{x}_{t-1}^{(r)}) d\mathbf{x}' \quad (5.90)$$

$$= w_{t-1}^{(r)} \int p(\mathbf{y}_t | \mathbf{x}'_t) p(\mathbf{x}'_t | \mathbf{x}_{t-1}^{(r)}) d\mathbf{x}' \quad (5.91)$$

$$= w_{t-1}^{(r)} \int g(\mathbf{y}_t | \mathbf{x}'_t) f(\mathbf{x}'_t | \mathbf{x}_{t-1}^{(r)}) d\mathbf{x}'_t \quad (5.92)$$

The proposal distribution is optimal because for any fixed  $\mathbf{x}_{t-1}^{(r)}$ , the new weight  $w_t^{(r)}$  takes the same value regardless of the value drawn for  $\mathbf{x}_t^{(r)}$ . Hence, conditional on the old values, the variance of true weights is zero.

## 5.6 Sequential Monte Carlo

(to do: improve to be more rigorous)

Assume that at time  $t$ , we can extend a particle's path using a Markov kernel  $M_t$ :

$$p_t(x_t) = p_{t-1}(x_{t-1})M_t(x_{t-1}, x_t) \quad (5.93)$$

Also assume that

$$\tilde{p}_t(x_{0:t}) = p_t(x_t) \sum_{k=1}^t L_k(x_k, x_{k-1}) \quad (5.94)$$

where  $\{L_k\}$  is a sequence of auxiliary Markov transition kernels.

The generic algorithm for Sequential Monte Carlo (SMC) can be found in Algorithm 8.

---

### Algorithm 8 Generic Sequential Monte Carlo

---

- 1: Initialisation,  $t = 0$ :
  - 2: **for**  $r = 1, \dots, R$  **do** ▷ Sample.
  - 3:     Sample  $\tilde{x}_0^{(r)} \sim q_0(\cdot)$ .
  - 4: **for**  $r = 1, \dots, R$  **do**
  - 5:     Calculate normalised weights  $\hat{w}_0^{(r)} \propto \frac{p_0(\tilde{x}_0^{(r)})}{q_0(\tilde{x}_0^{(r)})}$ , such that  $\sum_r \hat{w}_0^{(r)} = 1$ .
  - 6: Resample from the pmf  $\sum_r \hat{w}_0^{(r)} \delta_{\tilde{x}_0^{(r)}}(\cdot)$  to get  $R$  samples  $\{x_0^{(r)}\}$ . ▷ Resample.
  - 7:
  - 8: Iterate,  $t = 1, \dots, T$ :
  - 9: **for**  $t = 1, \dots, T$  **do**
  - 10:     **for**  $r = 1, \dots, R$  **do** ▷ Sample.
  - 11:         Set  $\tilde{x}_{0:t-1}^{(r)} = x_{0:t-1}^{(r)}$ .
  - 12:         Sample  $\tilde{x}_t^{(r)} \sim M_t(\tilde{x}_{0:t-1}^{(r)}, \cdot)$ .
  - 13:     **for**  $r = 1, \dots, R$  **do**
  - 14:         Calculate normalised weights  $\hat{w}_t^{(r)} \propto \frac{p_t(x_t) L_t(x_t, x_{t-1})}{p_{t-1}(x_{t-1}) M_t(x_{t-1}, x_t)}$ .
  - 15:     Resample from the pmf  $\sum_r \hat{w}_t^{(r)} \delta_{\tilde{x}_t^{(r)}}(\cdot)$  to get  $R$  samples  $\{x_t^{(r)}\}$ . Reset the weights to  $1/R$ . ▷ Resample.
-

## 5.7 Markov chain Monte Carlo methods

### 5.7.1 Definitions

**Definition 5.7.1.** Markov chain (MC) is defined via a state space  $\mathcal{X}$  and a model that defines, for every state  $\mathbf{x} \in \mathcal{X}$  a next-state distribution over  $\mathcal{X}$ . More precisely, the transition model  $\mathcal{T}$  specifies for each pair of state  $\mathbf{x}, \mathbf{x}'$  the probability  $\mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}')$  of going from  $\mathbf{x}$  to  $\mathbf{x}'$ , i.e.  $\mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \Pr(\mathbf{x}' \mid \mathbf{x})$ . This transition probability applies whenever the chain is in state  $\mathbf{x}$ .

If the MCMC generates a sequence of states  $\mathbf{x}_0, \dots, \mathbf{x}_T$ , the state at time  $t$ ,  $\mathbf{x}_t$  can be viewed as a random variable  $\mathbf{X}_t$  for  $t = 1, \dots, T$ .

**Theorem 5.7.1** (Ergodic Theorem for MC (simplified)). If  $(\mathbf{X}_0, \dots, \mathbf{X}_T)$  is an irreducible, time-homogeneous discrete space MC with stationary distribution  $\pi$ , then

$$\frac{1}{T} \sum_{t=1}^T f(\mathbf{X}_t) \xrightarrow[n \rightarrow \infty]{a.s.} \mathbb{E}[f(\mathbf{X})] \quad \text{where } \mathbf{X} \sim \pi \quad (5.95)$$

for any bounded function  $f : \mathcal{X} \mapsto \mathbb{R}$ .

If further, it is aperiodic, then

$$\Pr(\mathbf{X}_T = \mathbf{x} \mid \mathbf{X}_0 = \mathbf{x}_0) \xrightarrow[n \rightarrow \infty]{} \pi(\mathbf{x}) \quad \forall \mathbf{x}, \mathbf{x}_0 \in \mathcal{X}. \quad (5.96)$$

A MC following these conditions is ergodic

**Definition 5.7.2.** A MC  $(\mathbf{X}_t)$  is time-homogeneous if  $\Pr(\mathbf{X}_{t+1} = b \mid \mathbf{X}_t = a) = \mathcal{T}(a \rightarrow b) \forall t \in \{1, \dots, T-1\} \forall a, b \in \mathcal{X}$  for some kernel function  $\mathcal{T}$ .

**Definition 5.7.3.** A pmf  $\pi$  on  $\mathcal{X}$  is a stationary (invariant) distribution (w.r.t.  $\mathcal{T}$ ) if

$$\pi(\mathbf{X} = \mathbf{x}') = \sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{X} = \mathbf{x}) \mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') \quad \forall \mathbf{x}' \quad (5.97)$$

**Definition 5.7.4.** A MC  $(\mathbf{X}_t)$  is irreducible if  $\forall a, b \in \mathcal{X} \exists t \geq 0$  s.t.  $\Pr(\mathbf{X}_t = b \mid \mathbf{X}_0 = a) > 0$ .

**Definition 5.7.5.** An irreducible MC  $(\mathbf{X}_t)$  is aperiodic if  $\forall a \in \mathcal{X}$ ,

$$\gcd\{t : \Pr(\mathbf{X}_t = a \mid \mathbf{X}_0 = a) > 0\} = 1. \quad (5.98)$$

**Definition 5.7.6.** A MC is regular if there exists some number  $k$  such that, for every  $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ , the probability of getting from  $\mathbf{x}$  to  $\mathbf{x}'$  in exactly  $k$  steps is  $> 0$ .

**Theorem 5.7.2.** If a finite state MC described by  $\mathcal{T}$  is regular, then it has a unique stationary distribution.

A MC being *ergodic* is equivalent to it being *regular* [1, p. 510].



**Definition 5.7.7.** A finite state MC described by  $\mathcal{T}$  is reversible if there exists a unique distribution  $\pi$  such that, for all  $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$

$$\pi(\mathbf{x})\mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \pi(\mathbf{x}')\mathcal{T}(\mathbf{x}' \rightarrow \mathbf{x}). \quad (5.99)$$

This equation is called the detailed balance (DB).

**Proposition 5.7.1.** If a finite state MC described by  $\mathcal{T}$  is regular and satisfies the detailed balance equation relative to  $\pi$ , then  $\pi$  is the unique stationary distribution of  $\mathcal{T}$ .

*Proof.* Assuming the DB equation (5.99), we want to prove the stationarity equation (5.97) to ensure  $\pi$  is a stationary distribution of  $\mathcal{T}$ . We have

$$\sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{x})\mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{x}')\mathcal{T}(\mathbf{x}' \rightarrow \mathbf{x}) \quad (5.100)$$

$$= \sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{x}') \Pr(\mathbf{x} \mid \mathbf{x}') \quad (5.101)$$

$$= \pi(\mathbf{x}') \sum_{\mathbf{x} \in \mathcal{X}} \Pr(\mathbf{x} \mid \mathbf{x}') \quad (5.102)$$

$$= \pi(\mathbf{x}') \quad (5.103)$$

which proves the equation (5.97).  $\pi$  is the unique stationary distribution of  $\mathcal{T}$  because of Theorem 5.7.2.  $\square$

**Proposition 5.7.2.** Let  $\mathcal{T}_1, \dots, \mathcal{T}_K$  be a set of kernels each of which satisfies detailed balance w.r.t.  $\pi$ . Let  $p_1, \dots, p_K$  be any distribution over  $\{1, \dots, K\}$ . The mixture MC  $\mathcal{T}$ , which at each step takes a step sampled from  $\mathcal{T}_k$  with probability  $p_k$  also satisfies the detailed balance equation relative to  $\pi$ .

*Proof.* The aggregate kernel can be written as

$$\mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \Pr(\mathbf{x}' \mid \mathbf{x}) \quad (5.104)$$

$$= \sum_k \Pr(\mathbf{x}', k \mid \mathbf{x}) \quad (5.105)$$

$$= \sum_k \Pr(\mathbf{x}' \mid k, \mathbf{x}) \Pr(k \mid \mathbf{x}) \quad (5.106)$$

$$= \sum_k \mathcal{T}_k(\mathbf{x} \rightarrow \mathbf{x}') p_k \quad (5.107)$$

Using this, we can prove the detailed balance as follows

$$\pi(\mathbf{x})\mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \pi(\mathbf{x}) \sum_k \mathcal{T}_k(\mathbf{x} \rightarrow \mathbf{x}') p_k \quad (5.108)$$

$$= \sum_k \pi(\mathbf{x}) \mathcal{T}_k(\mathbf{x} \rightarrow \mathbf{x}') p_k \quad (5.109)$$

$$= \sum_k \pi(\mathbf{x}') \mathcal{T}_k(\mathbf{x}' \rightarrow \mathbf{x}) p_k \quad (5.110)$$

$$= \pi(\mathbf{x}') \sum_k \mathcal{T}_k(\mathbf{x}' \rightarrow \mathbf{x}) p_k \quad (5.111)$$

$$= \pi(\mathbf{x}') \mathcal{T}(\mathbf{x}' \rightarrow \mathbf{x}) \quad (5.112)$$

□

**Proposition 5.7.3.** *Let  $\mathcal{T}_1, \dots, \mathcal{T}_K$  be a set of kernels each of which satisfies detailed balance w.r.t.  $\pi$ . The aggregate MC,  $\mathcal{T}$ , where each step consists of a sequence of  $K$  steps, with step  $k$  being sampled from  $\mathcal{T}_k$  has  $\pi$  as its stationary distribution.*

*Proof.* The aggregate kernel can be written as

$$\mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \Pr(\mathbf{x}' | \mathbf{x}) \quad (5.113)$$

$$= \sum_{\mathbf{x}_{1:K-1}} \Pr(\mathbf{x}', \mathbf{x}_{K-1}, \dots, \mathbf{x}_1 | \mathbf{x}) \quad (5.114)$$

$$= \sum_{\mathbf{x}_{1:K-1}} \Pr(\mathbf{x}_K, \dots, \mathbf{x}_1 | \mathbf{x}_0) \quad (5.115)$$

$$= \sum_{\mathbf{x}_{1:K-1}} \Pr(\mathbf{x}_1 | \mathbf{x}_0) \cdots \Pr(\mathbf{x}_K | \mathbf{x}_{K-1}) \quad (5.116)$$

$$= \sum_{\mathbf{x}_{1:K-1}} \mathcal{T}_1(\mathbf{x}_0 \rightarrow \mathbf{x}_1) \cdots \mathcal{T}_K(\mathbf{x}_{K-1} \rightarrow \mathbf{x}_K) \quad (5.117)$$

where we've used the substitution  $\mathbf{x} = \mathbf{x}_0$  and  $\mathbf{x}' = \mathbf{x}_K$ . Using this, we can prove that  $\pi$  is the stationary distribution as follows

$$\sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{x}) \mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \sum_{\mathbf{x}_0} \pi(\mathbf{x}_0) \sum_{\mathbf{x}_{1:K-1}} \mathcal{T}_1(\mathbf{x}_0 \rightarrow \mathbf{x}_1) \cdots \mathcal{T}_K(\mathbf{x}_{K-1} \rightarrow \mathbf{x}_K) \quad (5.118)$$

$$= \sum_{\mathbf{x}_{0:K-1}} \pi(\mathbf{x}_0) \mathcal{T}_1(\mathbf{x}_0 \rightarrow \mathbf{x}_1) \cdots \mathcal{T}_K(\mathbf{x}_{K-1} \rightarrow \mathbf{x}_K) \quad (5.119)$$

$$= \sum_{\mathbf{x}_{0:K-1}} \mathcal{T}_1(\mathbf{x}_1 \rightarrow \mathbf{x}_0) \pi(\mathbf{x}_1) \cdots \mathcal{T}_K(\mathbf{x}_{K-1} \rightarrow \mathbf{x}_K) \quad (5.120)$$

...

$$= \sum_{\mathbf{x}_{0:K-1}} \mathcal{T}_1(\mathbf{x}_1 \rightarrow \mathbf{x}_0) \cdots \mathcal{T}_K(\mathbf{x}_K \rightarrow \mathbf{x}_{K-1}) \pi(\mathbf{x}_K) \quad (5.121)$$

$$= \pi(\mathbf{x}_K) \sum_{\mathbf{x}_{0:K-1}} \mathcal{T}_K(\mathbf{x}_K \rightarrow \mathbf{x}_{K-1}) \cdots \mathcal{T}_1(\mathbf{x}_1 \rightarrow \mathbf{x}_0) \quad (5.122)$$

$$= \pi(\mathbf{x}_K) \sum_{\mathbf{x}_{0:K-1}} \Pr(\mathbf{x}_{0:K-1} | \mathbf{x}_K) \quad (5.123)$$

$$= \pi(\mathbf{x}_K). \quad (5.124)$$

□

### 5.7.2 Metropolis Hastings algorithm

The Metropolis Hastings (MH) algorithm is a recipe to create a MCMC with a particular stationary distribution. Assume we can sample from a proposal distribution  $q(\cdot | \mathbf{x}) \equiv q(\mathbf{x} \rightarrow \cdot)$ . Let  $p \equiv \pi$  be the required distribution (stationary distribution for this MCMC). Assume we can only evaluate  $q$  and  $\pi$  up to a multiplicative factor (i.e. we can only evaluate  $q^*(\mathbf{x} \rightarrow \mathbf{x}') = Z_q q(\mathbf{x} \rightarrow \mathbf{x}')$  and  $\pi^*(\mathbf{x}) = Z_p \pi(\mathbf{x})$ ). The MH algorithm is outlined in Algorithm 9.

---

#### Algorithm 9 Metropolis Hastings algorithm

---

- 1: Sample  $\mathbf{x}^{(0)}$  from an arbitrary probability distribution over  $\mathcal{X}$ .
- 2: **for**  $t = 1, \dots, T$  **do**
- 3:     **repeat**
- 4:         Sample  $\mathbf{x}^{(t)} \sim q(\mathbf{x}^{(t-1)} \rightarrow \cdot)$ .
- 5:         Accept  $\mathbf{x}^{(t)}$  with the acceptance probability

$$\mathcal{A}(\mathbf{x}^{(t-1)} \rightarrow \mathbf{x}^{(t)}) = \min \left( 1, \frac{\pi^*(\mathbf{x}^{(t)}) q^*(\mathbf{x}^{(t)} \rightarrow \mathbf{x}^{(t-1)})}{\pi^*(\mathbf{x}^{(t-1)}) q^*(\mathbf{x}^{(t-1)} \rightarrow \mathbf{x}^{(t)})} \right) \quad (5.125)$$

- 6:     **until**  $\mathbf{x}^{(t)}$  is accepted.
- 

#### Why it works?

We need to prove that  $\pi$  is the unique stationary distribution of this MCMC.

We can express the aggregate transition model to be

$$\mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \begin{cases} q(\mathbf{x} \rightarrow \mathbf{x}') \mathcal{A}(\mathbf{x} \rightarrow \mathbf{x}') & \text{if } \mathbf{x} \neq \mathbf{x}' \\ q(\mathbf{x} \rightarrow \mathbf{x}) + \sum_{\mathbf{x}', \mathbf{x}' \neq \mathbf{x}} q(\mathbf{x} \rightarrow \mathbf{x}') (1 - \mathcal{A}(\mathbf{x} \rightarrow \mathbf{x}')) & \text{if } \mathbf{x} = \mathbf{x}' \end{cases} \quad (5.126)$$

To prove that  $\pi$  is a stationary distribution of this MCMC, we make sure the DB equation holds.

For  $\mathbf{x} \neq \mathbf{x}'$ , we have

$$\pi(\mathbf{x}) \mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \pi(\mathbf{x}) q(\mathbf{x} \rightarrow \mathbf{x}') \min \left( 1, \frac{\pi(\mathbf{x}') q(\mathbf{x}' \rightarrow \mathbf{x})}{\pi(\mathbf{x}) q(\mathbf{x} \rightarrow \mathbf{x}')} \right) \quad (5.127)$$

$$= \min (\pi(\mathbf{x}) q(\mathbf{x} \rightarrow \mathbf{x}'), \pi(\mathbf{x}') q(\mathbf{x}' \rightarrow \mathbf{x})) \quad (5.128)$$

$$= \pi(\mathbf{x}') q(\mathbf{x}' \rightarrow \mathbf{x}) \min \left( 1, \frac{\pi(\mathbf{x}) q(\mathbf{x} \rightarrow \mathbf{x}')}{\pi(\mathbf{x}') q(\mathbf{x}' \rightarrow \mathbf{x})} \right) \quad (5.129)$$

$$= \pi(\mathbf{x}') \mathcal{T}(\mathbf{x}' \rightarrow \mathbf{x}) \quad (5.130)$$

For  $\mathbf{x} = \mathbf{x}'$ , the DB equation  $\pi(\mathbf{x}) \mathcal{T}(\mathbf{x} \rightarrow \mathbf{x}') = \pi(\mathbf{x}') \mathcal{T}(\mathbf{x}' \rightarrow \mathbf{x})$  obviously holds.

Hence  $\pi$  is a stationary distribution of the MCMC described via  $\mathcal{T}$ . Unfortunately, regularity doesn't hold in general. We need to make sure our created MCMC is regular before we can claim that  $\pi$  is the unique stationary distribution of this MCMC.

### 5.7.3 Gibbs sampling

Assume we want to sample from  $p(\mathbf{x}) = p(x_1, \dots, x_D)$ . We can only sample from the conditionals  $p(x_i \mid \mathbf{x}_{-i})$  where  $\mathbf{x}_{-i}$  denotes  $\mathbf{x}$  with the  $i^{\text{th}}$  component omitted. The Gibbs sampling algorithm (10) is given below.

---

**Algorithm 10** Gibbs sampling algorithm

---

- 1: Sample  $\mathbf{x}^{(0)}$  from an arbitrary probability distribution over  $\mathcal{X}$ .
  - 2: **for**  $t = 1, \dots, T$  **do**
  - 3:     Sample  $x_1^{(t)} \sim p(\cdot \mid x_2^{(t-1)}, x_3^{(t-1)}, \dots, x_D^{(t-1)})$
  - 4:     Sample  $x_2^{(t)} \sim p(\cdot \mid x_1^{(t)}, x_3^{(t-1)}, \dots, x_D^{(t-1)})$
  - 5:      $\vdots$
  - 6:     Sample  $x_D^{(t)} \sim p(\cdot \mid x_1^{(t)}, x_2^{(t)}, \dots, x_{D-1}^{(t)})$
- 

#### Why it works?

Each of the sampling steps can be viewed to be governed by a different kernel with the whole process being governed by the aggregate kernel. We prove that the single kernels follow the DB equation with respect to  $p$ :

$$p(\mathbf{x})\mathcal{T}_i(\mathbf{x} \rightarrow \mathbf{x}') = p(\mathbf{x})p(\mathbf{x}_{-i}, x'_i \mid \mathbf{x}) \quad (5.131)$$

$$= p(\mathbf{x}_{-i}, x'_i, \mathbf{x}) \quad (5.132)$$

$$= p(\mathbf{x}, x'_i, \mathbf{x}_{-i}) \quad (5.133)$$

$$= p(\mathbf{x}')p(\mathbf{x} \mid x'_i, \mathbf{x}_{-i}) \quad (5.134)$$

$$= p(\mathbf{x}')\mathcal{T}_i(\mathbf{x}' \rightarrow \mathbf{x}) \quad (5.135)$$

This is the premise of Proposition 5.7.3, hence the aggregate kernel  $\mathcal{T}$  has  $p$  as its stationary distribution.

We can also view Gibbs sampling as an instance of the MH algorithm. If the proposal of MH  $q_i(\mathbf{x} \rightarrow \mathbf{x}')$  is set to be  $p(\mathbf{x}' \mid \mathbf{x}) = p(x'_i \mid \mathbf{x})$  the acceptance probability is one (shown below) and so it is equivalent to one sampling step in Gibbs sampling.

$$\mathcal{A}(\mathbf{x} \rightarrow \mathbf{x}') = \min \left( 1, \frac{p(\mathbf{x}')p(\mathbf{x} \mid \mathbf{x}')}{p(\mathbf{x})p(\mathbf{x}' \mid \mathbf{x})} \right) \quad (5.136)$$

$$= \min \left( 1, \frac{p(\mathbf{x}', \mathbf{x})}{p(\mathbf{x}', \mathbf{x})} \right) \quad (5.137)$$

$$= 1 \quad (5.138)$$

## 5.8 Particle Markov Chain Monte Carlo

### 5.8.1 Particle independent Metropolis Hastings (PIMH) sampler

We want to sample from  $p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta})$ .

---

**Algorithm 11** Particle independent Metropolis Hastings sampler

---

1: Run SMC targetting

▷ Initial sweep  $s = 0$

$$p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}; \boldsymbol{\theta})$$

2: Sample

$$\mathbf{x}_{1:T}(0) \sim \hat{p}(\cdot \mid \mathbf{y}_{1:T}; \boldsymbol{\theta})$$

3: Let

$$\hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})$$

denote the corresponding marginal likelihood estimate.

4: **for**  $s = 1, \dots, S$  **do**

▷ Main loop

5:   Run SMC targetting

$$p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}; \boldsymbol{\theta})$$

6:   Sample

$$\mathbf{x}_{1:T}^* \sim \hat{p}(\cdot \mid \mathbf{y}_{1:T}; \boldsymbol{\theta})$$

7:   Let

$$\hat{p}(\mathbf{y}_{1:T}; \boldsymbol{\theta})^*$$

denote the corresponding marginal likelihood estimate

8:   Sample from  $\text{Ber}(\cdot)$  with the success probability

$$\min \left( 1, \frac{\hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})^*}{\hat{p}(\mathbf{y}_{1:T}; \boldsymbol{\theta})(s-1)} \right)$$

9:   **if** success **then**

10:     Set

$$\mathbf{x}_{1:T}(s) = \mathbf{x}_{1:T}^*$$

$$\hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})(s) = \hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})^*$$

11:   **else**

12:     Set

$$\mathbf{x}_{1:T}(s) = \mathbf{x}_{1:T}(s-1)$$

$$\hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})(s) = \hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})(s-1)$$

---

**5.8.2 Particle marginal Metropolis Hastings (PMMH) sampler**

---

We want to sample from  $p(\boldsymbol{\theta}, \mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}) \propto p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}; \boldsymbol{\theta})p(\boldsymbol{\theta})$ .

---

**Algorithm 12** Particle marginal Metropolis Hastings sampler

---

1: Set  $\boldsymbol{\theta}(0)$  arbitrarily.

2: Run SMC targetting

▷ Initial sweep  $s = 0$

$$p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}; \boldsymbol{\theta}(0))$$

3: Sample

$$\mathbf{x}_{1:T}(0) \sim \hat{p}(\cdot \mid \mathbf{y}_{1:T}; \boldsymbol{\theta}(0))$$

4: Let

$$\hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}(0))$$

denote the corresponding marginal likelihood estimate.

5: **for**  $s = 1, \dots, S$  **do**

▷ Main loop

6:     Sample

$$\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}(s-1))$$

7:     Run SMC targetting

$$p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}; \boldsymbol{\theta}^*)$$

8:     Sample

$$\mathbf{x}_{1:T}^* \sim \hat{p}(\cdot \mid \mathbf{y}_{1:T}; \boldsymbol{\theta}^*)$$

9:     Let

$$\hat{p}(\mathbf{y}_{1:T}; \boldsymbol{\theta}^*)$$

denote the corresponding marginal likelihood estimate

10:     Sample from  $\text{Ber}(\cdot)$  with the success probability

$$\min \left( 1, \frac{\hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*) q(\boldsymbol{\theta}(s-1) \mid \boldsymbol{\theta}^*)}{\hat{p}(\mathbf{y}_{1:T}; \boldsymbol{\theta}(s-1)) p(\boldsymbol{\theta}(s-1)) q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}(s-1))} \right)$$

11:     **if** success **then**

12:         Set

$$\boldsymbol{\theta}(s) = \boldsymbol{\theta}^*$$

$$\mathbf{x}_{1:T}(s) = \mathbf{x}_{1:T}^*$$

$$\hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})(s) = \hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}^*)$$

13:     **else**

14:         Set

$$\boldsymbol{\theta}(s) = \boldsymbol{\theta}(s-1)$$

$$\mathbf{x}_{1:T}(s) = \mathbf{x}_{1:T}(s-1)$$

$$\hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})(s) = \hat{p}(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})(s-1)$$

### 5.8.3 Particle Gibbs (PG) sampler

#### Conditional SMC update

We want to sample from  $p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}; \boldsymbol{\theta})$ .

---

#### Algorithm 13 Conditional SMC update

---

1: Choose a fixed ancestral lineage  $B_{1:T}$  arbitrarily. ▷ Initialise fixed path

2: Let

$$\mathbf{x}_{1:T} = \left( \mathbf{x}_1^{(B_1)}, \dots, \mathbf{x}_T^{(B_T)} \right)$$

be a path associated with the ancestral lineage  $B_{1:T}$ .

3: For  $r \neq B_1$ , sample

▷ Time  $t = 1$

$$\mathbf{x}_1^{(r)} \sim q(\cdot \mid \mathbf{y}_1, \boldsymbol{\theta})$$

4: Compute weights

$$w_1^{(r)} \propto \frac{p\left(\mathbf{x}_1^{(r)}, \mathbf{y}_1\right)}{q\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1\right)}$$

5: Normalise weights

$$\hat{w}_1^{(r)} = \frac{w_1^{(r)}}{\sum_{r'} w_1^{(r')}}$$

6: We can resample from

$$\hat{p}(d\mathbf{x}_1 \mid \mathbf{y}_1, \boldsymbol{\theta}) = \sum_r \hat{w}_1^{(r)} \delta_{\mathbf{x}_1^{(r)}}(d\mathbf{x}_1)$$

to estimate

$$p(\mathbf{x}_1 \mid \mathbf{y}_1, \boldsymbol{\theta})$$

7: **for**  $t = 2, \dots, T$  **do**

▷ Main loop

8:     For  $r \neq B_t$ , sample

$$A_{t-1}^{(r)} \sim \text{Cat}\left(\hat{w}_{t-1}^{(1)}, \dots, \hat{w}_{t-1}^{(R)}\right)$$

9:     For  $r \neq B_t$ , sample

$$\mathbf{x}_t^{(r)} \sim q\left(\cdot \mid \mathbf{y}_t, \mathbf{x}_{t-1}^{(A_{t-1}^{(r)})}\right)$$

10:    Compute weights

$$w_t^{(r)} = \frac{p\left(\mathbf{x}_{1:t}^{(r)}, \mathbf{y}_{1:t}; \boldsymbol{\theta}\right)}{p\left(\mathbf{x}_{1:t-1}^{(A_{t-1}^{(r)})}, \mathbf{y}_{1:t-1}; \boldsymbol{\theta}\right) q\left(\mathbf{x}_t^{(r)} \mid \mathbf{y}_t, \mathbf{x}_{t-1}^{(A_{t-1}^{(r)})}; \boldsymbol{\theta}\right)}$$

11: Normalise weights

$$\hat{w}_t = \frac{w_t^{(r)}}{\sum_{r'} w_t^{(r')}} w_t^{(r')}$$

12: We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) = \sum_r \hat{w}_t^{(r)} \delta_{\mathbf{x}_{1:t}^{(r)}}(\mathrm{d}\mathbf{x}_{1:t})$$

to estimate

$$p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta})$$

### Particle Gibbs sampler

We want to sample from  $p(\boldsymbol{\theta}, \mathbf{x}_{1:T} \mid \mathbf{y}_{1:T})$ .

#### Algorithm 14 Particle Gibbs sampler

- |  |                           |
|--|---------------------------|
| 1: Set $\theta(0)$ , $\mathbf{x}_{1:T}(0)$ , $B_{1:T}(0)$ arbitrarily. | ▷ Initialisation, $s = 0$ |
| 2: <b>for</b> Sweep $s = 1, \dots, S$ <b>do</b>                        | ▷ Main loop               |
| 3:   Sample parameter  |                           |

$$\boldsymbol{\theta}(s) \sim p(\cdot \mid \mathbf{y}_{1:T}, \mathbf{x}_{1:T}(s-1))$$

- 4:   Run conditional SMC (Algorithm 13) targetting

$$p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}; \boldsymbol{\theta}(s))$$

conditional on

- $\mathbf{x}_{1:T}(s-1)$ , and
- $B_{1:T}(s-1)$ .

- 5:   Sample

$$\mathbf{x}_{1:T}(s) \sim \hat{p}(\cdot \mid \mathbf{y}_{1:T}; \boldsymbol{\theta}(s))$$



## **6 Nonparametric Bayesian models**

**6.1 Gaussian process**

**6.2 Dirichlet process**

**6.3 Chinese restaurant process**

**6.4 Hierarchical Dirichlet process**

**6.5 Hierarchical Dirichlet process**

**6.6 Indian buffet process**

**6.7 Dirichlet diffusion trees**

**6.8 Pitman-Yor process**

# Bibliography

- [1] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.