Personal notes – Bayesian machine learning

Tuan Anh Le

September 7, 2014

Contents

1 Probability distributions

1.1 Uniform distribution

1.2 Beta distribution

1.3 Bernoulli distribution

1.4 Binomial distribution

1.5 Beta-binomial distribution

1.6 Categorical distribution

1.7 Dirichlet distribution

1.8 Multinomial distribution

1.9 Pareto distribution

2 Bayesian parameter estimation

2.1 Beta-Bernoulli model

2.1.1 Summary

The model

$$X_i \sim \text{Ber}(\theta), \text{for } i \in \{1, \ldots, N\} \tag{2.1}$$

$$\mathcal{D} = \{x_1, \ldots, x_N\} \tag{2.2}$$

$$N_1 = \sum_{i=1}^{N} \mathbb{I}(x_i = 1) \tag{2.3}$$

$$N_0 = \sum_{i=1}^{N} \mathbb{I}(x_i = 0) \tag{2.4}$$

Likelihood

$$p(\mathcal{D}|\theta) = \theta^{N_1}(1 - \theta)^{N_0} \tag{2.5}$$

Prior

$$p(\theta) = \text{Beta}(\theta|a, b) \tag{2.6}$$

Posterior

$$p(\theta|\mathcal{D}) = \text{Beta}(\theta|a' = N_1 + a, b' = N_0 + b) \tag{2.7}$$

Posterior predictive

$$p(\tilde{x} = 1|\mathcal{D}) = \frac{a'}{a' + b'} \tag{2.8}$$

Evidence

2.1.2 Derivations

2.2 Beta-binomial model

2.2.1 Summary

The model

$$N_1 \sim \text{Bin}(N, \theta) \tag{2.9}$$

$$\mathcal{D} = \{N_1, N\} \tag{2.10}$$
$$N_1 = \text{number of successes} \tag{2.11}$$
$$N = \text{total number of trials} \tag{2.12}$$
$$\tilde{\mathcal{D}} = \{\tilde{N}_1, \tilde{N}\} \tag{2.13}$$
$$\tilde{N}_1 = \text{number of successes in a new batch of data} \tag{2.14}$$
$$\tilde{N} = \text{total number of trials in a new batch of data} \tag{2.15}$$

Likelihood
$$p(\mathcal{D}|\theta) = \text{Bin}(N_1|N, \theta) \tag{2.16}$$

Prior
$$p(\theta) = \text{Beta}(\theta|a, b) \tag{2.17}$$

Posterior
$$p(\theta|\mathcal{D}) = \text{Beta}(\theta|a' = N_1 + a, b' = N_0 + b) \tag{2.18}$$

Posterior predictive
$$p(\tilde{\mathcal{D}}|\mathcal{D}) = \text{Bb}(\tilde{N}_1; a', b', \tilde{N}) \tag{2.19}$$

Evidence

2.2.2 Derivations

2.3 Dirichlet-categorical model

2.3.1 Summary

The model
$$X_i \sim \text{Cat}\left(\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)^T\right), \text{for } i \in \{1, \ldots, N\} \tag{2.20}$$
$$\mathcal{D} = \{x_1, \ldots, x_N\} \tag{2.21}$$
$$n_k = \sum_{i=1}^{N} \mathbb{I}(x_i = k) \tag{2.22}$$

Likelihood
$$p(\mathcal{D}|\theta) = \prod_{k=1}^{K} \theta_k^{n_k} \tag{2.23}$$

Prior
$$p(\theta) = \text{Dir}(\boldsymbol{\theta}; \boldsymbol{\alpha}) \tag{2.24}$$

Posterior

$$p(\theta|\mathcal{D}) = \text{Dir}\left(\boldsymbol{\theta}; \boldsymbol{\alpha}' = \boldsymbol{\alpha} + (n_1, \ldots, n_K)^T\right) \tag{2.25}$$

Posterior predictive

$$p(\tilde{X} = j|\mathcal{D}) = \frac{\alpha'_j}{\sum_{k=1}^{K} \alpha'_i} \tag{2.26}$$

$$= \frac{\alpha_j + n_j}{\alpha_0 + N} \tag{2.27}$$

$$\text{where } \alpha_0 = \sum_{k=1}^{K} \alpha_k \tag{2.28}$$

Evidence

2.3.2 Derivations

2.4 Dirichlet-multinomial model

2.4.1 Summary

The model

$$\mathbf{N} \sim \text{Mult}(N, \boldsymbol{\theta}) \in \mathbb{R}^K \tag{2.29}$$

$$\mathcal{D} = \{\mathbf{n} = \text{vector of counts of successes}\} \tag{2.30}$$

$$N = \sum_{i=1}^{K} n_i \tag{2.31}$$

$$\tilde{\mathcal{D}} = \{\tilde{\mathbf{n}} = \text{vector of counts of successes in a new batch of data}\} \tag{2.32}$$

$$\tilde{N} = \sum_{i=1}^{K} \tilde{n}_i \tag{2.33}$$

Likelihood

$$p(\mathcal{D}|\theta) = \text{Mult}(\mathbf{n}; N, \boldsymbol{\theta}) \tag{2.34}$$

Prior

$$p(\theta) = \text{Dir}(\boldsymbol{\theta}; \boldsymbol{\alpha}) \tag{2.35}$$

Posterior

$$p(\theta|\mathcal{D}) = \text{Dir}\left(\boldsymbol{\theta}; \boldsymbol{\alpha}' = \boldsymbol{\alpha} + (n_1, \ldots, n_K)^T\right) \tag{2.36}$$

Posterior predictive

$$p(\tilde{\mathcal{D}}|\mathcal{D}) = \frac{\Gamma(\alpha_0 + N)}{\Gamma(\alpha_0 + N + \tilde{N})} \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + n_k + \tilde{n}_k)}{\Gamma(\alpha_k + n_k)} \tag{2.37}$$

$$\text{where } \alpha_0 = \sum_{k=1}^{K} \alpha_k \tag{2.38}$$

Evidence

$$p(\mathcal{D}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + N)} \prod_{k=1}^{K} \frac{\Gamma(\alpha_k + n_k)}{\Gamma(\alpha_k)} \tag{2.39}$$

2.4.2 Derivations

2.5 Poisson-gamma model

2.5.1 Summary

The model

$$x \sim \text{Poi}(\lambda) \tag{2.40}$$
$$\mathcal{D} = \{x_1, \ldots, x_N\} \tag{2.41}$$

Likelihood

$$p(\mathcal{D}|\lambda) = \prod_{i=1}^{N} \frac{\lambda^{x_i}}{x_i!} \exp(-\lambda) \tag{2.42}$$

Prior

$$p(\lambda) = \text{Gamma}(\lambda; a, b) \tag{2.43}$$

Posterior

$$p(\lambda|\mathcal{D}) = \text{Gamma}\left(\lambda; a' = a + \sum_{i=1}^{N} x_i, b' = b + N\right) \tag{2.44}$$

Posterior predictive

$$p(\tilde{x}|\mathcal{D}) = \text{NB}\left(\tilde{x}|a', \frac{1}{1+b'}\right) \tag{2.45}$$

Evidence

$$p(\mathcal{D}) = \tag{2.46}$$

2.5.2 Derivations

3 Sampling algorithms

3.1 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 $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(R)}\}, R \in \mathbb{N}$ (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 $\mathbf{x} \sim p$, $\mathrm{E}_{\mathbf{x} \sim p}[f(\mathbf{x})]$ (we will use the shorthand notation $\mathrm{E}[f]$ from now).

3.2 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

$$cq^*(\mathbf{x}) > p^*(\mathbf{x}) \text{ for all } \mathbf{x} \tag{3.1}$$

The procedure that generates a sample $\mathbf{x} \sim p$ is described in Algorithm 1 below.

Algorithm 1 Rejection sampling

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

3.2.1 Why it works?

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

$$\mathcal{X} = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^{d+1} : \alpha_{1:d} \in \mathbb{R}^d, \alpha_{d+1} \in [0, cq^*(\boldsymbol{\alpha})] \right\} \tag{3.2}$$

$$\mathcal{X}' = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^{d+1} : \alpha_{1:d} \in \mathbb{R}^d, \alpha_{d+1} \in [0, p^*(\boldsymbol{\alpha})] \right\} \tag{3.3}$$

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

$$\Pr(\mathbf{x}, u) = \Pr(\mathbf{x}) \Pr(u) \tag{3.4}$$

$$= q(\mathbf{x}) \frac{1}{c q^*(\mathbf{x})} \tag{3.5}$$

$$= q(\mathbf{x}) \frac{1}{c Z_q q(\mathbf{x})} \tag{3.6}$$

$$= \frac{1}{c Z_q} \tag{3.7}$$

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

$$(\mathbf{x}, u) \sim \text{Unif}(\mathcal{X}) \tag{3.8}$$

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

$$(\mathbf{x}', u') = \begin{cases} (\mathbf{x}, u) & \text{if } (\mathbf{x}, u) \in \mathcal{X}' \\ \text{nothing} & \text{otherwise.} \end{cases} \tag{3.9}$$

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

$$\Pr(\mathbf{x}', u') = \begin{cases} \Pr(\mathbf{x}, u) & \text{if } (\mathbf{x}, u) \in \mathcal{X}' \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

which means

$$(\mathbf{x}', u') \sim \text{Unif}(\mathcal{X}') \tag{3.11}$$

Working backwards

$$\Pr(\mathbf{x}') = \frac{\Pr(\mathbf{x}', u')}{\Pr(u')} \tag{3.12}$$

$$\propto \frac{1}{1/p^*(\mathbf{x}')} \tag{3.13}$$

$$\propto p^*(\mathbf{x}') \tag{3.14}$$

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

3.3 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 2 below.

Algorithm 2 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 $\mathrm{E}[f]$.

3.3.1 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 $\mathrm{E}[f]$, as R increases. We consider the the expectations of the numerator and denominator separately:

$$\mathrm{E}_q[\text{numer}] = \mathrm{E}_q\left[\sum_r w_r f(\mathbf{x}^{(r)})\right] \tag{3.15}$$

$$= \sum_r \mathrm{E}_q\left[w_r f(\mathbf{x}^{(r)})\right] \tag{3.16}$$

$$= \sum_r \mathrm{E}_q\left[\frac{p^*(\mathbf{x}^{(r)})}{q^*(\mathbf{x}^{(r)})} f(\mathbf{x}^{(r)})\right] \tag{3.17}$$

$$= \sum_r \mathrm{E}_q\left[\frac{Z_p p(\mathbf{x}^{(r)})}{Z_q q(\mathbf{x}^{(r)})} f(\mathbf{x}^{(r)})\right] \tag{3.18}$$

$$= \frac{Z_p}{Z_q} \sum_r \int_{\mathbf{x}^{(r)}} p(\mathbf{x}^{(r)}) f(\mathbf{x}^{(r)}) \, \mathrm{d}\mathbf{x}^{(r)} \tag{3.19}$$

$$= \frac{Z_p}{Z_q} \sum_r \mathrm{E}_p\left[f(\mathbf{x}^{(r)})\right] \tag{3.20}$$

$$= \frac{Z_p}{Z_q} R \, \mathrm{E}_p\left[f(\mathbf{x})\right] \tag{3.21}$$

$$\mathrm{E}_q[\text{denom}] = \mathrm{E}_q\left[\sum_r w_r\right] \tag{3.22}$$

$$= \sum_r \mathrm{E}_q\left[\frac{p^*(\mathbf{x}^{(r)})}{q^*(\mathbf{x}^{(r)})}\right] \tag{3.23}$$

$$= \sum_r \mathrm{E}_q\left[\frac{Z_p p(\mathbf{x}^{(r)})}{Z_q q(\mathbf{x}^{(r)})}\right] \tag{3.24}$$

$$= \frac{Z_p}{Z_q} \sum_r \int_{\mathbf{x}^{(r)}} p(\mathbf{x}^{(r)}) \, \mathrm{d}\mathbf{x}^{(r)} \tag{3.25}$$

$$= \frac{Z_p}{Z_q} R \tag{3.26}$$

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

3.3.2 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

$$\text{var}_q \left[\frac{p(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x}) \right] = \text{E}_q \left[\frac{p^2(\mathbf{x})}{q^2(\mathbf{x})} f^2(\mathbf{x}) \right] - \left(\text{E}_q \left[\frac{p(\mathbf{x})}{q(\mathbf{x})} f(\mathbf{x}) \right] \right)^2 \tag{3.27}$$

$$= \text{E}_q \left[\frac{p^2(\mathbf{x})}{q^2(\mathbf{x})} f^2(\mathbf{x}) \right] - \left(\text{E}_p \left[f(\mathbf{x}) \right] \right)^2 \tag{3.28}$$

The second part is independent of q so we can ignore it. By Jensen's inequality, we have $\text{E}\left[g(u(\mathbf{x})) \right] \geq g\left(\text{E}\left[u(\mathbf{x}) \right] \right)$ 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:

$$\text{E}_q \left[\frac{p^2(\mathbf{x})}{q^2(\mathbf{x})} f^2(\mathbf{x}) \right] \geq \left(\text{E}_q \left[\frac{p(\mathbf{x})}{q(\mathbf{x})} |f(\mathbf{x})| \right] \right)^2 = \left(\text{E}_p[|f(\mathbf{x})|] \right)^2 \tag{3.29}$$

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

$$q_{\text{optimal}}(\mathbf{x}) = \frac{|f(\mathbf{x})|p(\mathbf{x})}{\int |f(\mathbf{x}')|p(\mathbf{x}')\, \text{d}\mathbf{x}'} \tag{3.30}$$

3.4 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 3 below.

Algorithm 3 Sampling importance resampling

1: Generate samples $\left\{ \mathbf{x}^{(r)} \right\}$ 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

$$\hat{p}(\text{d}\mathbf{x}) = \sum_r \hat{w}_r \delta_{\mathbf{x}^{(r)}}(\text{d}\mathbf{x}) \tag{3.31}$$

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

3.4.1 Why it works?

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

$$N(x) = \lim_{\delta x \to 0} Rq(x)\delta x \tag{3.32}$$

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

$$\lim_{\delta x \to 0} \Pr(x \leq x^{(r)} \leq x + \delta x) = N(x)\hat{w}_r \tag{3.33}$$

$$\propto \lim_{\delta x \to 0} Rq(x)\delta x \frac{p(x)}{q(x)} \tag{3.34}$$

$$\propto \lim_{\delta x \to 0} p(x)\delta x \tag{3.35}$$

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

$$\Pr(a \leq x^{(r)} \leq b) \propto \int_a^b p(x)\,\mathrm{d}x \tag{3.36}$$

$$\implies x^{(r)} \sim p \tag{3.37}$$

3.5 Particle filtering

3.5.1 Sequential importance sampling (SIS)

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

- $\mathbf{x}_t, \mathbf{x}_t \subset \mathcal{X}^D$ and $\mathbf{y}_t, \mathbf{y}_t \subset \mathcal{Y}^D$ are the hidden and observed random variables at time $t, t = 1, \ldots, T$.

- The initial state is characterised by $\mathbf{x}_1 \sim \mu(\cdot \mid \boldsymbol{\theta})$ for some known parameter $\boldsymbol{\theta} \subset \Theta$.

- The transitions are characterised by $\mathbf{x}_t \mid \mathbf{x}_{t-1} \sim f(\cdot \mid \mathbf{x}_{t-1}; \boldsymbol{\theta})$.

- The emmissions are characterised by $\mathbf{y}_t \mid \mathbf{x}_t \sim g(\cdot \mid \mathbf{x}_t; \boldsymbol{\theta})$.

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

$$q(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}; \boldsymbol{\theta}) = q(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}; \boldsymbol{\theta})q(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t}; \boldsymbol{\theta}) \tag{3.38}$$

$$= q(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}; \boldsymbol{\theta})q(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1}; \boldsymbol{\theta}) \tag{3.39}$$

$$= q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_t; \boldsymbol{\theta}) \tag{3.40}$$

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

$$p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t} \mid \mathbf{x}_{1:t})p(\mathbf{x}_{1:t})}{p(\mathbf{y}_{1:t})} \tag{3.41}$$

$$= \frac{p(\mathbf{y}_t \mid \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1} \mid \mathbf{x}_{1:t})p(\mathbf{x}_{1:t})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})} \tag{3.42}$$

$$= \frac{p(\mathbf{y}_t \mid \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1})p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} \tag{3.43}$$

$$= \frac{p(\mathbf{y}_t \mid \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1}) p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1}) p(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} \quad (3.44)$$

$$= \frac{p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} \quad (3.45)$$

$$\propto p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1}) \quad (3.46)$$

$$= g(\mathbf{y}_t \mid \mathbf{x}_t) f(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1}) \quad (3.47)$$

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

$$w_t^{(r)} \propto \frac{p\left(\mathbf{x}_{1:t}^{(r)} \mid \mathbf{y}_{1:t}\right)}{q\left(\mathbf{x}_{1:t}^{(r)} \mid \mathbf{y}_{1:t}\right)} \quad (3.48)$$

$$\propto \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right) p\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}\right) p\left(\mathbf{x}_{1:t-1}^{(r)} \mid \mathbf{y}_{1:t-1}\right)}{q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right) q\left(\mathbf{x}_{1:t-1}^{(r)} \mid \mathbf{y}_{1:t-1}\right)} \quad (3.49)$$

$$= w_{t-1}^{(r)} \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right) p\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}\right)}{q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right)} \quad (3.50)$$

$$= w_{t-1}^{(r)} \frac{g\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right) f\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}\right)}{q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right)} \quad (3.51)$$

For $t = 1$

$$w_1^{(r)} \propto \frac{p\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1\right)}{q\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1\right)} \quad (3.52)$$

$$\propto \frac{p\left(\mathbf{x}_1^{(r)}, \mathbf{y}_1\right)}{q\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1\right)} \quad (3.53)$$

$$\propto \frac{p\left(\mathbf{y}_1 \mid \mathbf{x}_1^{(r)}\right) p\left(\mathbf{x}_1^{(r)}\right)}{q\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1\right)} \quad (3.54)$$

$$= \frac{g\left(\mathbf{y}_1 \mid \mathbf{x}_1^{(r)}\right) \mu\left(\mathbf{x}_1^{(r)}\right)}{q\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1\right)} \quad (3.55)$$

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 4 below.

The reason why it works is the same as in the case of Sampling importance resampling described in section 3.4.

Algorithm 4 Sequential importance sampling

1: Sample from proposal ▷ Initialisation

$$\mathbf{x}_1^{(r)} \sim q\left(\cdot \mid \mathbf{y}_1^{(r)}, \boldsymbol{\theta}\right), r = 1, \ldots, R \tag{3.56}$$

2: Compute weights

$$w_1^{(r)} \propto \frac{g\left(\mathbf{y}_1 \mid \mathbf{x}_1^{(r)}\right) \mu\left(\mathbf{x}_1^{(r)}\right)}{q\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1\right)}, r = 1, \ldots, R \tag{3.57}$$

3: Normalise weights

$$\hat{w}_1^{(r)} = \frac{w_1^{(r)}}{\sum_{r'} w_1^{(r')}}, r = 1, \ldots, R \tag{3.58}$$

4: We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_1 \mid \mathbf{y}_1, \boldsymbol{\theta}) = \sum_r \hat{w}_1^{(r)} \delta_{\mathbf{x}_1^{(r)}}(\mathrm{d}\mathbf{x}_1) \tag{3.59}$$

to estimate

$$p(\mathbf{x}_1 \mid \mathbf{y}_1, \boldsymbol{\theta}) \tag{3.60}$$

5: **for** $t = 2, \ldots, T$ **do** ▷ Main loop
6:     Compute weights

$$w_t^{(r)} \propto w_{t-1}^{(r)} \frac{g\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}, \boldsymbol{\theta}\right) f\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \boldsymbol{\theta}\right)}{q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \mathbf{y}_t, \boldsymbol{\theta}\right)}, r = 1, \ldots, R \tag{3.61}$$

7:     Normalise weights

$$\hat{w}_t^{(r)} = \frac{w_t^{(r)}}{\sum_{r'} w_t^{(r')}}, r = 1, \ldots, R \tag{3.62}$$

8:     We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) = \sum_r \hat{w}_t^{(r)} \delta_{\mathbf{x}_{1:t}^{(r)}}(\mathrm{d}\mathbf{x}_{1:t}) \tag{3.63}$$

to estimate

$$p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) \tag{3.64}$$

3.5.2 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:

$$S_{\text{eff}} \triangleq \frac{S}{1 + \text{var}\left[w_t^{(r)*}\right]} \tag{3.65}$$

$$\hat{S}_{\text{eff}} \approx \frac{1}{\sum_r \left(w_t^{(r)}\right)^2} \tag{3.66}$$

where $w_t^{(r)*} = p(\mathbf{x}_t^{(r)} \mid \mathbf{y}_{1:t})/q(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t)$ 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.

3.5.3 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 1 below in which we resample during every tie step.

Algorithm 1 Generic particle filter

1: Sample from proposal ▷ Initialisation

$$\mathbf{x}_1^{(r)} \sim q\left(\cdot \mid \mathbf{y}_1^{(r)}, \boldsymbol{\theta}\right), r = 1, \ldots, R \tag{3.67}$$

2: Compute weights

$$w_1^{(r)} \propto \frac{p\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1, \boldsymbol{\theta}\right)}{q\left(\mathbf{x}_1^{(r)} \mid \mathbf{y}_1, \boldsymbol{\theta}\right)}, r = 1, \ldots, R \tag{3.68}$$

3: Normalise weights

$$\hat{w}_1^{(r)} = \frac{w_1^{(r)}}{\sum_{r'} w_1^{(r')}}, r = 1, \ldots, R \tag{3.69}$$

4: We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_1 \mid \mathbf{y}_1, \boldsymbol{\theta}) = \sum_r \hat{w}_1^{(r)} \delta_{\mathbf{x}_1^{(r)}}(\mathrm{d}\mathbf{x}_1) \tag{3.70}$$

to estimate

$$p(\mathbf{x}_1 \mid \mathbf{y}_1, \boldsymbol{\theta}) \tag{3.71}$$

5: **for** $t = 2, \ldots, T$ **do** ▷ Main loop
6:     Sample parents' indices of tth generation

$$A_{t-1}^{(r)} \sim \text{Cat}\left(\hat{w}_{t-1}\right), r = 1, \ldots, R \tag{3.72}$$

7:     Sample t^{th} generation using corresponding parents

$$\mathbf{x}_t^{(r)} \sim q\left(\cdot \mid \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \mathbf{y}_t, \boldsymbol{\theta}\right), r = 1, \ldots, R \tag{3.73}$$

8:     Compute weights

$$w_t^{(r)} \propto w_{t-1}^{(r)} \frac{g\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}, \boldsymbol{\theta}\right) f\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \boldsymbol{\theta}\right)}{q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{A_{t-1}^{(r)}}, \mathbf{y}_t, \boldsymbol{\theta}\right)}, r = 1, \ldots, R \tag{3.74}$$

9:     Normalise weights

$$\hat{w}_t^{(r)} = \frac{w_t^{(r)}}{\sum_{r'} w_t^{(r')}}, r = 1, \ldots, R \tag{3.75}$$

10:     We can resample from

$$\hat{p}(\mathrm{d}\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) = \sum_r \hat{w}_t^{(r)} \delta_{\mathbf{x}_{1:t}^{(r)}}(\mathrm{d}\mathbf{x}_{1:t}) \tag{3.76}$$

    to estimate

$$p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta}) \tag{3.77}$$

3.5.4 Particle filter animation

3.5.5 The proposal distribution

It is common to use the following proposal distribution

$$q\left(\mathbf{x}_{1:t}^{(r)} \mid \mathbf{y}_{1:t}\right) = q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right) \tag{3.78}$$

$$= p\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}\right) \tag{3.79}$$

$$= f\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}\right) \tag{3.80}$$

Hence the weight equation in (3.51) becomes

$$w_t^{(r)} \propto w_{t-1}^{(r)} \frac{g\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right) f\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}\right)}{q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right)} \tag{3.81}$$

$$= w_{t-1}^{(r)} g\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right) \tag{3.82}$$

This approach can be **i**nefficient because the likelihood, $p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right)$, can be very small at many places meaning many of the particles will be very small.

The optimal proposal distribution has the form

$$q\left(\mathbf{x}_{1:t}^{(r)} \mid \mathbf{y}_{1:t}\right) = q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right) \tag{3.83}$$

$$= p\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right) \tag{3.84}$$

$$= \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}^{(r)}\right) p\left(\mathbf{x}_t, \mathbf{x}_{t-1}^{(r)}\right)}{p\left(\mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right)} \tag{3.85}$$

$$= \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t\right) p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(r)}\right)}{p\left(\mathbf{y}_t \mid \mathbf{x}_{t-1}^{(r)}\right)} \tag{3.86}$$

$$= \frac{g\left(\mathbf{y}_t \mid \mathbf{x}_t\right) f\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(r)}\right)}{p\left(\mathbf{y}_t \mid \mathbf{x}_{t-1}^{(r)}\right)} \tag{3.87}$$

The weight equation in (3.51) becomes

$$w_t^{(r)} \propto w_{t-1}^{(r)} p\left(\mathbf{y}_t \mid \mathbf{x}_{t-1}^{(r)}\right) \tag{3.88}$$

$$= w_{t-1}^{(r)} \int p\left(\mathbf{y}_t, \mathbf{x}_t' \mid \mathbf{x}_{t-1}^{(r)}\right) \, \mathrm{d}\mathbf{x}' \tag{3.89}$$

$$= w_{t-1}^{(r)} \int p\left(\mathbf{y}_t \mid \mathbf{x}_t', \mathbf{x}_{t-1}^{(r)}\right) p\left(\mathbf{x}_t' \mid \mathbf{x}_{t-1}^{(r)}\right) \, \mathrm{d}\mathbf{x}' \tag{3.90}$$

$$= w_{t-1}^{(r)} \int p\left(\mathbf{y}_t \mid \mathbf{x}_t'\right) p\left(\mathbf{x}_t' \mid \mathbf{x}_{t-1}^{(r)}\right) \, \mathrm{d}\mathbf{x}' \tag{3.91}$$

$$= w_{t-1}^{(r)} \int g\left(\mathbf{y}_t \mid \mathbf{x}_t'\right) f\left(\mathbf{x}_t' \mid \mathbf{x}_{t-1}^{(r)}\right) \, \mathrm{d}\mathbf{x}' \tag{3.92}$$

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.

3.6 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:

$$p_t(x_t) = p_{t-1}(x_{t-1})M_t(x_{t-1}, x_t) \tag{3.93}$$

Also assume that

$$\tilde{p}_t(x_{0:t}) = p_t(x_t) \sum_{k=1}^{t} L_k(x_k, x_{k-1}) \tag{3.94}$$

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

The generic algorithm for Sequential Monte Carlo (SMC) can be found in Algorithm 5.

Algorithm 5 Generic Sequential Monte Carlo

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

3.7 Markov chain Monte Carlo methods

3.7.1 Definitions

Definition 3.7.1. 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 $\mathcal{T}(\mathbf{x} \to \mathbf{x}')$ of going from $\mathbf{x}$ to $\mathbf{x}'$, i.e. $\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \Pr(\mathbf{x}' \mid \mathbf{x})$. This transition probability applies whenever the chain is in state $\mathbf{x}$.*

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

Theorem 3.7.1 (Ergodic Theorem for MC (simplified)). *If $(\mathbf{X}_0, \ldots, \mathbf{X}_T)$ is an irreducible, time-homogeneous discrete space MC with stationary distribution π, then*

$$\frac{1}{T} \sum_{t=1}^{T} f(\mathbf{X}_t) \xrightarrow[n \to \infty]{a.s.} \mathrm{E}[f(\mathbf{X})] \qquad \text{where } \mathbf{X} \sim \pi \qquad (3.95)$$

for any bounded function $f : \mathcal{X} \mapsto \mathbb{R}$.
If further, it is aperiodic, then

$$\Pr(\mathbf{X}_T = \mathbf{x} \mid \mathbf{X}_0 = \mathbf{x}_0) \xrightarrow[n \to \infty]{} \pi(\mathbf{x}) \qquad \forall \mathbf{x}, \mathbf{x}_0 \in \mathcal{X}. \qquad (3.96)$$

A MC following these conditions is ergodic

Definition 3.7.2. *A MC $(\mathbf{X}_t)$ is time-homogeneous if $\Pr(\mathbf{X}_{t+1} = b \mid \mathbf{X}_t = a) = \mathcal{T}(a \to b) \; \forall\, t \in \{1, \ldots, T-1\} \; \forall\, a, b \in \mathcal{X}$ for some kernel function $\mathcal{T}$.*

Definition 3.7.3. *A pmf π on $\mathcal{X}$ is a stationary (invariant) distribution (w.r.t. $\mathcal{T}$) if*

$$\pi(\mathbf{X} = \mathbf{x}') = \sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{X} = \mathbf{x}) \mathcal{T}(\mathbf{x} \to \mathbf{x}') \qquad \forall \mathbf{x}' \qquad (3.97)$$

Definition 3.7.4. *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 3.7.5. *An irreducible MC $(\mathbf{X}_t)$ is aperiodic if $\forall\, a \in \mathcal{X}$,*

$$\gcd\{t : \Pr(\mathbf{X}_t = a \mid \mathbf{X}_0 = a) > 0\} = 1. \qquad (3.98)$$

Definition 3.7.6. *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 3.7.2. *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 3.7.7. *A finite state MC described by $\mathcal{T}$ is* reversible *if there exists a unique distribution π such that, for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$*

$$\pi(\mathbf{x})\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x}')\mathcal{T}(\mathbf{x}' \to \mathbf{x}). \tag{3.99}$$

This equation is called the detailed balance *(DB).*

Proposition 3.7.1. *If a finite state MC described by $\mathcal{T}$ is* regular *and satisfies the* detailed balance *equation relative to π, then π is the* unique stationary distribution *of $\mathcal{T}$.*

Proof. Assuming the DB equation (3.99), we want to prove the stationarity equation (3.97) to ensure π is a stationary distribution of $\mathcal{T}$. We have

$$\sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{x})\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{x}')\mathcal{T}(\mathbf{x}' \to \mathbf{x}) \tag{3.100}$$

$$= \sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{x}') \Pr(\mathbf{x} \mid \mathbf{x}') \tag{3.101}$$

$$= \pi(\mathbf{x}') \sum_{\mathbf{x} \in \mathcal{X}} \Pr(\mathbf{x} \mid \mathbf{x}') \tag{3.102}$$

$$= \pi(\mathbf{x}') \tag{3.103}$$

which proves the equation (3.97). π is the unique stationary distribution of $\mathcal{T}$ because of Theorem 3.7.2. $\qquad\square$

Proposition 3.7.2. *Let $\mathcal{T}_1, \ldots, \mathcal{T}_K$ be a set of kernels each of which satisfies detailed balance w.r.t. π. Let $p_1, \ldots, p_K$ be any distribution over $\{1, \ldots, K\}$. 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 π.*

Proof. The aggregate kernel can be written as

$$\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \Pr(\mathbf{x}' \mid \mathbf{x}) \tag{3.104}$$

$$= \sum_{k} \Pr(\mathbf{x}', k \mid \mathbf{x}) \tag{3.105}$$

$$= \sum_{k} \Pr(\mathbf{x}' \mid k, \mathbf{x}) \Pr(k \mid \mathbf{x}) \tag{3.106}$$

$$= \sum_{k} \mathcal{T}_k(\mathbf{x} \to \mathbf{x}')p_k \tag{3.107}$$

Using this, we can prove the detailed balance as follows

$$\pi(\mathbf{x})\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x}) \sum_{k} \mathcal{T}_k(\mathbf{x} \to \mathbf{x}')p_k \tag{3.108}$$

$$= \sum_{k} \pi(\mathbf{x})\mathcal{T}_k(\mathbf{x} \to \mathbf{x}')p_k \tag{3.109}$$

$$= \sum_k \pi(\mathbf{x}') \mathcal{T}_k(\mathbf{x}' \to \mathbf{x}) p_k \tag{3.110}$$

$$= \pi(\mathbf{x}') \sum_k \mathcal{T}_k(\mathbf{x}' \to \mathbf{x}) p_k \tag{3.111}$$

$$= \pi(\mathbf{x}') \mathcal{T}(\mathbf{x}' \to \mathbf{x}) \tag{3.112}$$

$\square$

Proposition 3.7.3. *Let $\mathcal{T}_1, \ldots, \mathcal{T}_K$ be a set of kernels each of which satisfies detailed balance w.r.t. π. 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 π as its stationary distribution.*

Proof. The aggregate kernel can be written as

$$\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \Pr(\mathbf{x}' \mid \mathbf{x}) \tag{3.113}$$

$$= \sum_{\mathbf{x}_{1:K-1}} \Pr(\mathbf{x}', \mathbf{x}_{K-1}, \ldots, \mathbf{x}_1 \mid \mathbf{x}) \tag{3.114}$$

$$= \sum_{\mathbf{x}_{1:K-1}} \Pr(\mathbf{x}_K, \ldots, \mathbf{x}_1 \mid \mathbf{x}_0) \tag{3.115}$$

$$= \sum_{\mathbf{x}_{1:K-1}} \Pr(\mathbf{x}_1 \mid \mathbf{x}_0) \cdots \Pr(\mathbf{x}_K \mid \mathbf{x}_{K-1}) \tag{3.116}$$

$$= \sum_{\mathbf{x}_{1:K-1}} \mathcal{T}_1(\mathbf{x}_0 \to \mathbf{x}_1) \cdots \mathcal{T}_K(\mathbf{x}_{K-1} \to \mathbf{x}_K) \tag{3.117}$$

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

$$\sum_{\mathbf{x} \in \mathcal{X}} \pi(\mathbf{x}) \mathcal{T}(\mathbf{x} \to \mathbf{x}') = \sum_{\mathbf{x}_0} \pi(\mathbf{x}_0) \sum_{\mathbf{x}_{1:K-1}} \mathcal{T}_1(\mathbf{x}_0 \to \mathbf{x}_1) \cdots \mathcal{T}_K(\mathbf{x}_{K-1} \to \mathbf{x}_K) \tag{3.118}$$

$$= \sum_{\mathbf{x}_{0:K-1}} \pi(\mathbf{x}_0) \mathcal{T}_1(\mathbf{x}_0 \to \mathbf{x}_1) \cdots \mathcal{T}_K(\mathbf{x}_{K-1} \to \mathbf{x}_K) \tag{3.119}$$

$$= \sum_{\mathbf{x}_{0:K-1}} \mathcal{T}_1(\mathbf{x}_1 \to \mathbf{x}_0) \pi(\mathbf{x}_1) \cdots \mathcal{T}_K(\mathbf{x}_{K-1} \to \mathbf{x}_K) \tag{3.120}$$

$$\cdots$$

$$= \sum_{\mathbf{x}_{0:K-1}} \mathcal{T}_1(\mathbf{x}_1 \to \mathbf{x}_0) \cdots \mathcal{T}_K(\mathbf{x}_K \to \mathbf{x}_{K-1}) \pi(\mathbf{x}_K) \tag{3.121}$$

$$= \pi(\mathbf{x}_K) \sum_{\mathbf{x}_{0:K-1}} \mathcal{T}_K(\mathbf{x}_K \to \mathbf{x}_{K-1}) \cdots \mathcal{T}_1(\mathbf{x}_1 \to \mathbf{x}_0) \tag{3.122}$$

$$= \pi(\mathbf{x}_K) \sum_{\mathbf{x}_{0:K-1}} \Pr(\mathbf{x}_{0:K-1} \mid \mathbf{x}_K) \tag{3.123}$$

$$= \pi(\mathbf{x}_K). \tag{3.124}$$

$\square$

3.7.2 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 $q(\cdot \mid \mathbf{x}) \equiv q(\mathbf{x} \to \cdot)$. Let $p \equiv \pi$ be the required distribution (stationary distribution for this MCMC). Assume we can only evaluate q and π up to a multiplicative factor (i.e. we can only evaluate $q^*(\mathbf{x} \to \mathbf{x}') = Z_q q(\mathbf{x} \to \mathbf{x}')$ and $\pi^*(\mathbf{x}) = Z_p \pi(\mathbf{x})$). The MH algorithm is outlined in Algorithm 6.

Algorithm 6 Metropolis Hastings algorithm

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

$$\mathcal{A}(\mathbf{x}^{(t-1)} \to \mathbf{x}^{(t)}) = \min\left(1, \frac{\pi^*(\mathbf{x}^{(t)}) q^*(\mathbf{x}^{(t)} \to \mathbf{x}^{(t-1)})}{\pi^*(\mathbf{x}^{(t-1)}) q^*(\mathbf{x}^{(t-1)} \to \mathbf{x}^{(t)})}\right) \tag{3.125}$$

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

Why it works?

We need to prove that π is the unique stationary distribution of this MCMC.

We can express the aggregate transition model to be

$$\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \begin{cases} q(\mathbf{x} \to \mathbf{x}')\mathcal{A}(\mathbf{x} \to \mathbf{x}') & \text{if } \mathbf{x} \neq \mathbf{x}' \\ q(\mathbf{x} \to \mathbf{x}) + \displaystyle\sum_{\mathbf{x}', \mathbf{x}' \neq \mathbf{x}} q(\mathbf{x} \to \mathbf{x}')(1 - \mathcal{A}(\mathbf{x} \to \mathbf{x}')) & \text{if } \mathbf{x} = \mathbf{x}' \end{cases} \tag{3.126}$$

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

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

$$\pi(\mathbf{x})\mathcal{T}(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}')\min\left(1, \frac{\pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}')}\right) \tag{3.127}$$

$$= \min\left(\pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}'), \pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x})\right) \tag{3.128}$$

$$= \pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x})\min\left(1, \frac{\pi(\mathbf{x})q(\mathbf{x} \to \mathbf{x}')}{\pi(\mathbf{x}')q(\mathbf{x}' \to \mathbf{x})}\right) \tag{3.129}$$

$$= \pi(\mathbf{x}')\mathcal{T}(\mathbf{x}' \to \mathbf{x}) \tag{3.130}$$

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

Hence π 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 π is the unique stationary distribution of this MCMC.

3.7.3 Gibbs sampling

Assume we want to sample from $p(\mathbf{x}) = p(x_1, \ldots, x_D)$. We can only sample from the conditionals $p(x_i \mid \mathbf{x}_{-i})$ where $\mathbf{x}_{-i}$ denotes $\mathbf{x}$ with the ith component ommited. The Gibbs sampling algorithm (7) is given below.

Algorithm 7 Gibbs sampling algorithm

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

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:

$$p(\mathbf{x})\mathcal{T}_i(\mathbf{x} \to \mathbf{x}') = p(\mathbf{x})p(\mathbf{x}_{-i}, x_i' \mid \mathbf{x}) \tag{3.131}$$

$$= p(\mathbf{x}_{-i}, x_i', \mathbf{x}) \tag{3.132}$$

$$= p(\mathbf{x}, x_i', \mathbf{x}_{-i}) \tag{3.133}$$

$$= p(\mathbf{x}')p(\mathbf{x} \mid x_i', \mathbf{x}_{-i}) \tag{3.134}$$

$$= p(\mathbf{x}')\mathcal{T}_i(\mathbf{x}' \to \mathbf{x}) \tag{3.135}$$

This is the premise of Proposition 3.7.3, 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 $q_i(\mathbf{x} \to \mathbf{x}')$ 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.

$$\mathcal{A}(\mathbf{x} \to \mathbf{x}') = \min\left(1, \frac{p(\mathbf{x}')p(\mathbf{x} \mid \mathbf{x}')}{p(\mathbf{x})p(\mathbf{x}' \mid \mathbf{x})} \right) \tag{3.136}$$

$$= \min\left(1, \frac{p(\mathbf{x}', \mathbf{x})}{p(\mathbf{x}', \mathbf{x})} \right) \tag{3.137}$$

$$= 1 \tag{3.138}$$

3.8 Particle Markov Chain Monte Carlo

3.8.1 Particle independent Metropolis Hastings (PIMH) sampler

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

3.8.2 Particle marginal Metropolis Hastings (PMMH) sampler

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

3.8.3 Particle Gibbs (PG) sampler

Conditional SMC update

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

Particle Gibbs sampler

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

Algorithm 8 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** $s = 1, \ldots, S$ **do** ▷ Main loop

5:   Run SMC targeting

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

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 coresponding marginal likelihood estimate

8:   Sample from $\mathrm{Ber}(\cdot)$ 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

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

Algorithm 9 Particle marginal Metropolis Hastings sampler

1: Set $\boldsymbol{\theta}(0)$ arbitrarily.
2: Run SMC targetting $\triangleright$ 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** $s = 1, \ldots, S$ **do** $\triangleright$ Main loop
6:      Sample
$$\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}(s-1))$$

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

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 coresponding 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

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

Algorithm 10 Conditional SMC update

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

2: Let

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

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

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

to estimate

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

7: **for** $t = 2, \ldots, T$ **do**   ▷ Main loop

8:     For $r \neq B_t$, sample

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

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

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

11:     Normalise weights

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

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

$$\overset{29}{p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}, \boldsymbol{\theta})}$$

Algorithm 11 Particle Gibbs sampler

1: Set $\theta(0)$, $\mathbf{x}_{1:T}(0)$, $B_{1:T}(0)$ arbitrarily. $\qquad\qquad\qquad$ ▷ Initialisation, $s = 0$

2: **for** Sweep $s = 1, \ldots, S$ **do** $\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ Main loop

3: $\qquad$ Sample parameter

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

4: $\qquad$ Run conditional SMC (Algorithm 10) 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: $\qquad$ Sample

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

Bibliography

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