Personal notes – Bayesian machine learning

Tuan Anh Le

August 25, 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

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

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

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

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

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

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

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

Evidence

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

2.5.2 Derivations

3 Sampling algorithms

3.1 Introduction

Let p be a probability distribution with pdf $p(\mathbf{x})$, 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 $Z_p = \int p^*(\mathbf{x})\,\mathrm{d}\mathbf{x}$). We want to achieve the following:

Problem 1 Generate samples $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(R)}\}$ (shorthand notation $\{\mathbf{x}^{(r)}\}$) from the probability distribution p.

Problem 2 Estimate the expectation of an arbitrary function f given that $\mathbf{x} \sim p$, $\mathrm{E}[f]$.

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

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

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

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

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

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

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

$$= \frac{1}{cZ_q} \tag{3.7}$$

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

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

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

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

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.

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:

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

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 $\mathrm{E}[f]$.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 $\left\{\mathbf{x}^{(r)}\right\}$ which are approximately $\sim p$. The process is described in Algorithm 3 below.

Algorithm 3 Sampling importance resampling

1: Generate samples $\left\{\mathbf{z}^{(r)}\right\}$ 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: Resample from a probability distribution with the pmf

$$f(\mathbf{x}) = \sum_r \hat{w}_r \delta_{\mathbf{z}^{(r)}}(\mathbf{x}) \tag{3.31}$$

5: The resulting samples $\left\{\mathbf{x}^{(r)}\right\}$ are approximately $\sim p$.

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, $x^{(x)}$ 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 $\mathbf{x}_t$ and $\mathbf{y}_t$ are the hidden and observed random variables at time t. We want to sample from the distribution $p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$. Assume we can sample from the probability distribution with the pdf of the following form

$$\begin{aligned}
q(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}) &= q(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t})q(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t}) && (3.38)\\
&= q(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t})q(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1}) && (3.39)\\
&= q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_t) && (3.40)
\end{aligned}$$

If we express the pdf of p in the form of

$$\begin{aligned}
p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}) &= \frac{p(\mathbf{y}_{1:t} \mid \mathbf{x}_{1:t})p(\mathbf{x}_{1:t})}{p(\mathbf{y}_{1:t})} && (3.41)\\
&= \frac{p(\mathbf{y}_t \mid \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1} \mid \mathbf{x}_{1:t})p(\mathbf{x}_{1:t})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})} && (3.42)\\
&= \frac{p(\mathbf{y}_t \mid \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1})p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} && (3.43)\\
&= \frac{p(\mathbf{y}_t \mid \mathbf{x}_{1:t}, \mathbf{y}_{1:t-1})p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1})p(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} && (3.44)\\
&= \frac{p(\mathbf{y}_t \mid \mathbf{x}_t)p(\mathbf{x}_t \mid \mathbf{x}_{t-1})p(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})} && (3.45)\\
&\propto p(\mathbf{y}_t \mid \mathbf{x}_t)p(\mathbf{x}_t \mid \mathbf{x}_{t-1})p(\mathbf{x}_{1:t-1} \mid \mathbf{y}_{1:t-1}) && (3.46)
\end{aligned}$$

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

$$w_t^{(r)} \propto \frac{p\left(\mathbf{x}_{1:t}^{(r)} \mid \mathbf{y}_{1:t}\right)}{q\left(\mathbf{x}_{1:t}^{(r)} \mid \mathbf{y}_{1:t}\right)} \tag{3.47}$$

$$\propto \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right) p\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}\right) p\left(\mathbf{x}_{1:t-1}^{(r)} \mid \mathbf{y}_{1:t-1}\right)}{q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right) q\left(\mathbf{x}_{1:t-1}^{(r)} \mid \mathbf{y}_{1:t-1}\right)} \tag{3.48}$$

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

The algorithm for SIS is shown in Algorithm 4 below. The reason why it works is the

Algorithm 4 Sequential importance sampling

1: Initialise weights $\left\{w_0^{(r)} = \frac{1}{R}\right\}$.
2: **for** $t = 1, \ldots, T$ **do**
3: Observe $\mathbf{y}_t$.
4: Sample $\left\{\mathbf{x}_{1:t}^{(r)}\right\}$ from $q\left(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}\right)$.
5: Calculate weights $\left\{w_t^{(r)}\right\}$ according to (3.49).
6: Calculate normalised weights $\left\{\hat{w}_t^{(r)} = \frac{w_t^{(r)}}{\sum_{r'} w_t^{(r')}}\right\}$.
7: ▷ The pmf $\sum_r \hat{w}_t^{(r)} \delta_{\mathbf{x}_{1:t}^{(r)}}(\mathbf{x}_{1:t})$ approximates the pdf $p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$. Hence we can approximate the pdf $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ by $\sum_r \hat{w}_t^{(r)} \delta_{\mathbf{x}_t^{(r)}}(\mathbf{x}_t)$.

same as in the case of Sampling importance resampling described in section 3.4.

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:

$$S_{\text{eff}} \triangleq \frac{S}{1 + \text{var}\left[w_t^{(r)^*}\right]} \tag{3.50}$$

$$\hat{S}_{\text{eff}} \approx \frac{1}{\sum_r \left(w_t^{(r)}\right)^2} \tag{3.51}$$

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 5 below.

Algorithm 5 Generic particle filter

1: Initialise weights $\left\{w_0^{(r)} = \frac{1}{R}\right\}$.
2: **for** $t = 1, \ldots, T$ **do**
3: Observe $\mathbf{y}_t$.
4: Sample $\left\{\mathbf{x}_{1:t}^{(r)}\right\}$ from $q\left(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t}\right)$.
5: Calculate weights $\left\{w_t^{(r)}\right\}$ according to (3.49).
6: Calculate normalised weights $\left\{\hat{w}_t^{(r)} = \frac{w_t^{(r)}}{\sum_{r'} w_t^{(r')}}\right\}$.
7: Calculate the effective sample size, $\hat{S}_{\text{eff}}$, according to (3.51).
8: **if** $\hat{S}_{\text{eff}} < S_{\text{min}}$ **then**
9: Resample R particles, $\left\{\mathbf{x}_t^{(r)}\right\}$from the pmf $\sum_r \hat{w}_t^{(r)} \delta_{\mathbf{x}_t^{(r)}}(\mathbf{x}_t)$.
10: Reassign $w_t^{(r)} = \frac{1}{R}$ for $r = 1, \ldots, R$.

3.5.4 The proposal distribution

It is common to use the following proposal distribution

$$q(\mathbf{x}_{1:t}^{(r)} \mid \mathbf{y}_{1:t}) = q(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t) \tag{3.52}$$
$$= p(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}) \tag{3.53}$$

Hence the weight equation in (3.49) becomes

$$w_t^{(r)} \propto w_{t-1}^{(r)} \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right) p\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}\right)}{q\left(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t\right)} \tag{3.54}$$
$$= w_{t-1}^{(r)} p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(r)}\right) \tag{3.55}$$

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

$$q(\mathbf{x}_{1:t}^{(r)} \mid \mathbf{y}_{1:t}) = q(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t) \tag{3.56}$$
$$= p(\mathbf{x}_t^{(r)} \mid \mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t) \tag{3.57}$$

$$= \frac{p(\mathbf{y}_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}^{(r)}) p(\mathbf{x}_t, \mathbf{x}_{t-1}^{(r)})}{p(\mathbf{x}_{t-1}^{(r)}, \mathbf{y}_t)} \tag{3.58}$$

$$= \frac{p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(r)})}{p(\mathbf{y}_t \mid \mathbf{x}_{t-1}^{(r)})} \tag{3.59}$$

The weight equation in (3.49) becomes

$$w_t^{(r)} \propto w_{t-1}^{(r)} p\left(\mathbf{y}_t \mid \mathbf{x}_{t-1}^{(r)}\right) \tag{3.60}$$

$$= w_{t-1}^{(r)} \int p(\mathbf{y}_t, \mathbf{x}'_t \mid \mathbf{x}_{t-1}^{(r)}) \, \mathrm{d}\mathbf{x}' \tag{3.61}$$

$$= w_{t-1}^{(r)} \int p(\mathbf{y}_t \mid \mathbf{x}'_t, \mathbf{x}_{t-1}^{(r)}) p(\mathbf{x}'_t \mid \mathbf{x}_{t-1}^{(r)}) \, \mathrm{d}\mathbf{x}' \tag{3.62}$$

$$= w_{t-1}^{(r)} \int p(\mathbf{y}_t \mid \mathbf{x}'_t) p(\mathbf{x}'_t \mid \mathbf{x}_{t-1}^{(r)}) \, \mathrm{d}\mathbf{x}' \tag{3.63}$$

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:

$$p_t(x_t) = p_{t-1}(x_{t-1}) M_t(x_{t-1}, x_t) \tag{3.64}$$

Also assume that

$$\tilde{p}_t(x_{0:t}) = p_t(x_t) \sum_{k=1}^{t} L_k(x_k, x_{k-1}) \tag{3.65}$$

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

The generic algorithm for Sequential Monte Carlo (SMC) can be found in Algorithm 6.

Algorithm 6 Generic Sequential Monte Carlo

Initialisation, $t = 0$:
for $r = 1, \ldots, R$ **do** ▷ Sample.
    Sample $\tilde{x}_0^{(r)} \sim q_0(\cdot)$.
for $r = 1, \ldots, R$ **do**
    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 $\sum_r' \hat{w}_0^{(r')} = 1$.
Resample from the pmf $\sum_r \hat{w}_0^{(r)} \delta_{\tilde{x}_0^{(r)}}(\cdot)$ to get R samples $\left\{x_0^{(r)}\right\}$. ▷ Resample.

Iterate, $t = 1, \ldots, T$:
for $t = 1, \ldots, T$ **do**
    for $r = 1, \ldots, R$ **do** ▷ Sample.
        Set $\tilde{x}_{0:t-1}^{(r)} = x_{0:t-1}^{(r)}$.
        Sample $\tilde{x}_t^{(r)} \sim M_t\left(\tilde{x}_{0:t-1}^{(r)}, \cdot\right)$.
    for $r = 1, \ldots, R$ **do**
        Calculate normalised weights $\hat{w}_t^{(r)} \propto \frac{p_t(x_t) L_t(x_t, x_{t-1})}{p_{t-1}(x_{t-1}) M_t(x_{t-1}, x_t)}$.
    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 $1/R$. ▷ Resample.