Sequential Monte Carlo

Consider a sequence of probability distributions

$$\pi_1, \pi_2, \ldots,$$

defined on a sequence of state spaces

$$\mathbb{X}_1, \mathbb{X}_2,$$

where, for each n, $\mathbb{X}_n = \mathbb{X}^n$ for some $\mathbb{X}$.

Suppose that

$$\pi_n(\boldsymbol{x}_n) = \frac{g_n(\boldsymbol{x}_n)}{Z_n}$$

where $\boldsymbol{x}_n = (x_1, \ldots, x_n)$, g_n is an integrable non-negative function, and Z_n is a normalizing constant.

Such a series of distributions could arise in

1. **Dynamic problems:** where data are collected sequentially though time, as in *state-space models*

$$\begin{aligned} y_t &= h(x_t; \theta) + \epsilon_t \\ x_t &= \mu + \phi(x_{t-1} - \mu) + \varepsilon_t \end{aligned}$$

   where the observed data $\{y_1, \ldots, y_n\}$ are related to the unobserved states $\{x_1, \ldots, x_n\}$.

   Here the parameters $\theta, \mu, \phi, \sigma_\epsilon^2, \sigma_\varepsilon^2$ are also of interest.

   Here π_t is the posterior distribution derived from the data $\{y_1, \ldots, y_t)$.

2. **Static problems:** where data are collected together, but used sequentially. For example, in a Bayesian inference problem, we may have
   - π_1 being the posterior computed from x_1
   - π_2 being the posterior computed from x_1, x_2

   and so on.

In both cases, we may wish to perform a Monte Carlo calculation with respect to π_n, such as an expectation of the form

$$\mathbb{E}_{\pi_n}[h(\boldsymbol{X}_n)] = \int h(\boldsymbol{x}_n)\pi_n(\boldsymbol{x}_n)\, d\boldsymbol{x}_n$$

Sequential Monte Carlo

Recall the basic principle of importance sampling:

$$\int h(x)p(x)\,dx = \int h(x)p(x)\frac{p_0(x)}{p_0(x)}\,dx = \int \frac{h(x)p(x)}{p_0(x)}p_0(x)\,dx$$

demonstrates that

$$\mathbb{E}_p[h(X)] = \mathbb{E}_{p_0}\left[\frac{h(X)p(X)}{p_0(X)}\right]$$

so that an estimator of the LHS is

$$\widehat{I}_N^{(p_0)}(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{h(X_i)p(X_i)}{p_0(X_i)}$$

where $X_1, \ldots, X_N \sim p_0(.)$.

Here, write

$$\pi_n(\boldsymbol{x}_n) = \frac{g_n(\boldsymbol{x}_n)}{Z_n} = \frac{g_n(\boldsymbol{x}_n)}{\int g(\boldsymbol{x}_n)\, d\boldsymbol{x}_n}$$

$$= \frac{w_n(\boldsymbol{x}_n) p_n(\boldsymbol{x}_n)}{\int w_n(\boldsymbol{x}_n) p_n(\boldsymbol{x}_n)\, d\boldsymbol{x}_n}$$

where p_n is the importance distribution, and

$$w_n(\boldsymbol{x}_n) = \frac{g_n(\boldsymbol{x}_n)}{p_n(\boldsymbol{x}_n)}$$

is the importance weight.

Sequential Monte Carlo

If a i.i.d. sample of size N

$$\boldsymbol{x}_n^{(1)}, \ldots, \boldsymbol{x}_n^{(N)}$$

can be obtained from p_n, then the empirical distribution

$$\widehat{p}_n(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{x}_n^{(i)}}(\boldsymbol{x})$$

can be used to estimate π_n

$$\widehat{\pi}_n(\boldsymbol{x}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{x}_n^{(i)}}(\boldsymbol{x})$$

where

$$W_n^{(i)} = \frac{w_n(\boldsymbol{x}_n^{(i)})}{N\widehat{Z}_n} \qquad \widehat{Z}_n = \frac{1}{N} \sum_{i=1}^{N} w_n(\boldsymbol{x}_n^{(i)})$$

Recall that we require the variance of the weights to be finite.

In a "static" Bayesian calculation, the posterior from n data points can be written

$$\pi_n(\theta|\boldsymbol{y}_n) = \frac{L(\theta;\boldsymbol{y}_n)\pi(\theta)}{\int L(\varphi;\boldsymbol{y}_n)\pi(\varphi)\,d\varphi}$$

so could consider setting

$$p_n(\theta) = \pi(\theta)$$

for all n. The problem with this is that $L(\theta;\boldsymbol{y}_n)$ is likely to be very peaked compared to $\pi(\theta)$ once n gets moderately large.

Sequential Importance Sampling

Using the chain-rule factorization

$$\pi_n(\boldsymbol{x}_n) = \pi_1(x_1) \prod_{j=2}^{n} \widetilde{\pi}_j(x_j|\boldsymbol{x}_{j-1})$$

where

$$\widetilde{\pi}_j(x_j|\boldsymbol{x}_{j-1}) = \frac{\pi_j(\boldsymbol{x}_j)}{\pi_{j-1}(\boldsymbol{x}_{j-1})}$$

is the conditional distribution of element x_j, given $\boldsymbol{x}_{j-1} = (x_1, \ldots, x_{j-1})$, it is possible to construct a *Sequential Importance Sampling* (SIS) density

$$p_n(\boldsymbol{x}_n) = p_1(x_1) \prod_{j=2}^{n} \widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})$$

where $\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})$ is again a conditional distribution.

Sequential Importance Sampling

The importance sampling weight then takes the form

$$w_n(\boldsymbol{x}_n) = \frac{\pi_n(\boldsymbol{x}_n)}{p_n(\boldsymbol{x}_n)} = \frac{\pi_1(x_1)}{p_1(x_1)} \prod_{j=2}^{n} \frac{\widetilde{\pi}_j(x_j|\boldsymbol{x}_{j-1})}{\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})}$$

so therefore

$$\begin{aligned} w_n(\boldsymbol{x}_n) &= w_{n-1}(\boldsymbol{x}_{n-1}) \frac{\widetilde{\pi}_n(x_n|\boldsymbol{x}_{n-1})}{\widetilde{p}_n(x_n|\boldsymbol{x}_{n-1})} \\ w_{n-1}(\boldsymbol{x}_{n-1}) &= w_{n-2}(\boldsymbol{x}_{n-2}) \frac{\widetilde{\pi}_{n-1}(x_{n-1}|\boldsymbol{x}_{n-2})}{\widetilde{p}_{n-1}(x_{n-1}|\boldsymbol{x}_{n-2})} \end{aligned}$$

and so on, so there is a *recursive* weight calculation,

$$w_j(\boldsymbol{x}_j) = w_{j-1}(\boldsymbol{x}_{j-1}) \frac{\widetilde{\pi}_j(x_j|\boldsymbol{x}_{j-1})}{\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})}.$$

Sequential Importance Sampling

In the numerator of the weights, we have the conditional densities

$$\widetilde{\pi}_j(x_j|\boldsymbol{x}_{j-1}) = \frac{\pi_j(\boldsymbol{x}_j)}{\pi_{j-1}(\boldsymbol{x}_{j-1})}$$

but to compute this we need the marginal distributions π_j and π_{j-1}, which may not be available analytically.

A strategy for the **static** Monte Carlo case involves, for each j, defining the *auxiliary distributions*

$$p_{0j}(\boldsymbol{x}_j) \qquad j = 1, \ldots, n$$

such that $p_{0n}(\boldsymbol{x}_n) \equiv \pi_n(\boldsymbol{x}_n)$, and where p_{0j} resembles π_j, but is **directly available**.

Combining Importance Sampling and MCMC

Recall that

$$\int h(x)\pi(x)\,dx = \int h(x)\frac{\pi(x)}{p_0(x)}p_0(x)\,dx = \int h(x)w(x)p_0(x)\,dx$$

If P is a Markov transition kernel with stationary (invariant) distribution π, that is, for all x

$$\int \pi(z)P(z,x)\,dz = \pi(x)$$

then also

$$\int \frac{\pi(z)}{p_0(z)}P(z,x)p_0(z)\,dz = \pi(x)$$

Combining Importance Sampling and MCMC

Thus

$$\int w(z)P(z,x)p_0(z)\,dz = \pi(x)$$

and hence, from the importance sampling identity

$$\mathbb{E}_\pi[h(X)] = \mathbb{E}_{p_0}[w(X)h(X)] = \int\int h(x)w(z)P(x,z)p_0(x)\,dzdx$$

and the estimator

$$\frac{1}{N}\sum_{i=1}^{N} h(X_i)w(Z_i)$$

can be constructed, where

- $X_1, \ldots, X_N \sim p_0$
- $Z_1, \ldots, Z_N \sim f(z|x)$

where $f(z|x)$ is the conditional density for Z given X implied by $P(.,.)$.

Annealed Importance Sampling

Annealed Importance Sampling (AIS) uses a SIS approach, but with auxiliary densities **each in n dimensions** defined by

$$p_{0j}(\boldsymbol{x}_j) = \{p_0(\boldsymbol{x}_j)\}^{1-\xi_j}\{\pi_n(\boldsymbol{x}_j)\}^{\xi_j}$$

where p_0 is a "diffuse" distribution, and

$$0 = \xi_0 < \xi_1 < \cdots < \xi_n = 1.$$

This defines a "path" from p_0 to π_n. These distributions are utilized by employing a MCMC kernel.

Annealed Importance Sampling

The AIS algorithm proceeds as follows: dropping the dependence on n throughout

1. Sample $\boldsymbol{x}_0$ from p_{00}, set $w_0 = 1$
2. For $j = 1, 2, \ldots, n$,
   (i) Sample $\boldsymbol{x}_j$ from $P_j(\boldsymbol{x}_{j-1}, \boldsymbol{x})$, where P_j is a Markov kernel with invariant distribution p_{0j}.
   (ii) Set
   $$w_j = w_{j-1} \frac{p_{0j}(\boldsymbol{x}_j)}{p_{0\,j-1}(\boldsymbol{x}_{j-1})}$$
3. Return to 1 and repeat N times to produce N samples

   $$\boldsymbol{x}_n^{(1)}, \ldots, \boldsymbol{x}_n^{(N)}$$

   and weights

   $$w_n^{(1)}, \ldots, w_n^{(N)}$$

Annealed Importance Sampling

If P_j^R is the "reverse" kernel, for each instance

$$P_j^R(\boldsymbol{x}, \boldsymbol{x}') = \frac{P_j(\boldsymbol{x}', \boldsymbol{x}) p_{0j}(\boldsymbol{x}')}{p_{0j}(\boldsymbol{x})}$$

and

$$\begin{aligned} \pi^\star(\boldsymbol{x}^\star) &= p_{0n}(\boldsymbol{x}_n) \prod_{j=1}^{n} P_j^R(\boldsymbol{x}_{n-j}, \boldsymbol{x}_{n-j+1}) \\ p_0^\star(\boldsymbol{x}^\star) &= p_{00}(\boldsymbol{x}_0) \prod_{j=1}^{n} P_j(\boldsymbol{x}_{j-1}, \boldsymbol{x}_j) \end{aligned}$$

then

$$w^\star(\boldsymbol{x}^\star) = \frac{\pi^\star(\boldsymbol{x}^\star)}{p_0^\star(\boldsymbol{x}^\star)}$$

is the importance sampling weight for the augmented sample $(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ generated from $p_0^\star$. The marginal distribution from $\pi^\star$ of $\boldsymbol{x}_n$ is the **true target** π_n.

Sequential Monte Carlo for State-Space Models

Recall the state-space model: for $t = 1, \ldots, T$

$$\begin{aligned} y_t &= g(x_t; \theta) + \epsilon_t \\ x_{t+1} &= h(x_t; \theta) + \varepsilon_t \end{aligned}$$

where only the series $y_1, \ldots, y_T$, is observed. Sequences $\{\epsilon_t\}$ and $\{\varepsilon_t\}$ are independent zero mean random variables.

If $\overline{y}_t = (y_1, \ldots, y_t)$, $\overline{x}_t = (x_1, \ldots, x_t)$. we must carry out

- **Filtering:** Compute $p(x_t|\overline{y}_t)$
- **Prediction:** Compute $p(x_{t+1}|\overline{y}_t)$
- **Smoothing:** Compute $p(x_t|\overline{y}_T)$

Filtering: By Bayes theorem

$$\begin{aligned} p(x_t|\overline{y}_t) &= \frac{p(y_t|x_t)p(x_t|\overline{y}_{t-1})}{p(y_t|\overline{y}_{t-1})} \\ &\propto p(y_t|x_t)p(x_t|\overline{y}_{t-1}) \end{aligned}$$

where the function $p(y_t|x_t)$ is usually a straightforward conditional density, but the function $p(x_t|\overline{y}_{t-1})$ is the density computed at the previous time point **prediction** step.

Prediction:

$$p(x_{t+1}|\overline{y}_t) = \int p(x_{t+1}|x_t)p(x_t|\overline{y}_t)\, dx_t$$

where $p(x_{t+1}|x_t)$ is again a conditional density derived from the state equation, and $p(x_t|\overline{y}_t)$ is the posterior density computed at the time point **filtering** step.

We consider simulation-based versions of these calculations.

For example, in the **filtering** step, suppose we have a sample

$$z_t^{(1)}, \ldots, z_t^{(N)}$$

from $p(x_t|\overline{y}_{t-1})$ obtained at the previous prediction step. Then we can obtain a *particle approximation* to $p(x_t|\overline{y}_t)$ as

$$\widehat{p}(x_t|\overline{y}_t) = \frac{1}{N}\sum_{i=1}^{N} p(y_t|z_t^{(i)})\delta_{z_t^{(i)}}$$

Sequential Monte Carlo for State-Space Models

Note also that the denominator $p(y_t|\overline{y}_{t-1})$ can be approximated by the sum of the weights

$$\widehat{p}(y_t|\overline{y}_{t-1}) = \frac{1}{N}\sum_{i=1}^{N} p(y_t|z_t^{(i)})$$

which is useful in approximating the likelihood by

$$\widehat{L}(\overline{y}_t) = \widehat{p}(y_1)\prod_{t=2}^{T}\widehat{p}(y_t|\overline{y}_{t-1})$$

We can also produce a sample from the particle approximation $\widehat{p}(x_t|\overline{y}_t)$ by performing a *weighted resampling* from the collection $(z_t^{(1)}, \ldots, z_t^{(N)})$ with weights proportional to $p(y_t|z_t^{(i)})$. Denote this resampled vector

$$v_t^{(1)}, \ldots, v_t^{(N)}$$

Sequential Monte Carlo for State-Space Models

For the **prediction** step, the particle approximation to $p(x_{t+1}|\overline{y}_t)$ is obtained using a similar Monte Carlo strategy

$$\widehat{p}(x_{t+1}|\overline{y}_t) = \frac{1}{N}\sum_{i=1}^{N} p(x_{t+1}|v_t^{(i)})$$

Also, a sample from this approximation can be obtained by propagating the samples $v_t^{(1)}, \ldots, v_t^{(N)}$ forward by sampling the transition densities $p(x_{t+1}|v_t^{(i)})$ for each i.

This yields the sample

$$z_{t+1}^{(1)}, \ldots, z_{t+1}^{(N)}$$

and the recursion continues.

Sequential Monte Carlo for State-Space Models

Recap: to perform SIS for $\pi_n(\boldsymbol{x}_n) = g_n(\boldsymbol{x}_n)/Z_n$, use the IS density

$$p_n(\boldsymbol{x}_n) = p_1(x_1) \prod_{j=2}^{n} \widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})$$

with weights given by $w_1(\boldsymbol{x}_1) = \pi_1(x_1)/p_1(x_1)$ and

$$w_j(\boldsymbol{x}_j) = w_{j-1}(\boldsymbol{x}_{j-1}) \frac{\widetilde{\pi}_j(x_j|\boldsymbol{x}_{j-1})}{\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})} = w_{j-1}(\boldsymbol{x}_{j-1}) \frac{\pi_j(\boldsymbol{x}_j)}{\pi_{j-1}(\boldsymbol{x}_{j-1})\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})}$$

If the normalizing constants Z_j for $j = 1, \ldots, n$ are unknown, the weights can be replaced by $w_1(\boldsymbol{x}_1) = g_1(x_1)/p_1(x_1)$ and

$$w_j(\boldsymbol{x}_j) = w_{j-1}(\boldsymbol{x}_{j-1}) \frac{g_j(\boldsymbol{x}_j)}{g_{j-1}(\boldsymbol{x}_{j-1})\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})} = w_{j-1}(\boldsymbol{x}_{j-1})\alpha_j(\boldsymbol{x}_j)$$

say.

The Monte Carlo computation is carried out by sampling $x_1^{(1)}, \ldots, x_1^{(N)}$ from p_1, then at step $j \geqslant 2$ sampling

$$x_j^{(1)}, \ldots, x_j^{(N)} \quad \text{from} \quad \tilde{p}_j(.|\boldsymbol{x}_{j-1}^{(i)})$$

for $i = 1, \ldots, N$. The importance sampling weights are $w_1(x_1^{(i)})$ and, for $j \geqslant 2$,

$$w_j(\boldsymbol{x}_j^{(i)}) = w_{j-1}(\boldsymbol{x}_{j-1}^{(i)})\alpha_j(\boldsymbol{x}_j^{(i)})$$

We then have

$$\widehat{\pi}_n(\boldsymbol{x}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{x}_n^{(i)}}(\boldsymbol{x})$$

where

$$W_n^{(i)} = \frac{w_n(\boldsymbol{x}_n^{(i)})}{N\widehat{Z}_n} \qquad \widehat{Z}_n = \frac{1}{N}\sum_{i=1}^{N} w_n(\boldsymbol{x}_n^{(i)})$$

in the usual way.

Sequential Monte Carlo for State-Space Models

In a simple state-space model, write

$$\begin{aligned} \text{Observation equation} &: p_{Y|X}(y_j|x_j) \\ \text{State equation} &: p_X(x_j|x_{j-1}) \end{aligned}$$

with $X_1 \sim p_{X_1}$.

In light of the observed data $\boldsymbol{y}_n = (y_1, \ldots, y_n)$, the posterior of interest is

$$\pi_n(\boldsymbol{x}_n) = p(\boldsymbol{x}_n|\boldsymbol{y}_n) = \frac{p(\boldsymbol{x}_n, \boldsymbol{y}_n)}{p(\boldsymbol{y}_n)} = \frac{g_n(\boldsymbol{x}_n, \boldsymbol{y}_n)}{Z_n(\boldsymbol{y}_n)}$$

We have

$$g_n(\boldsymbol{x}_n, \boldsymbol{y}_n) = p_{X_1}(x_1) \left\{ \prod_{j=2}^{n} p_X(x_j|x_{j-1}) \right\} \left\{ \prod_{j=1}^{n} p_{Y|X}(y_j|x_j) \right\}$$

and

$$Z_n(\boldsymbol{y}_n) = \int g_n(\boldsymbol{x}_n, \boldsymbol{y}_n)\, d\boldsymbol{x}_n$$

Sequential Monte Carlo for State-Space Models

Now, for sequential Monte Carlo, the optimal choice of IS density is

$$\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})$$

is

$$\widetilde{\pi}_j(x_j|\boldsymbol{x}_{j-1})$$

which in the simple state-space model is given by

$$\begin{aligned}\widetilde{\pi}_j(x_j|\boldsymbol{x}_{j-1}) \equiv p(x_j|\boldsymbol{y}_j, \boldsymbol{x}_{j-1}) &= \frac{p_{Y|X}(y_j|x_j)p_X(x_j|x_{j-1})}{p(y_j|x_{j-1})} \\ &= \frac{p_{Y|X}(y_j|x_j)p_X(x_j|x_{j-1})}{\int p_{Y|X}(y_j|x)p_X(x|x_{j-1})\,dx}\end{aligned}$$

In most cases (specifically, outside of the Gaussian case), the optimal choice is not feasible: recall that we need successive samples from the IS densities

$$\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1}) = p(x_j|\boldsymbol{y}_j, \boldsymbol{x}_{j-1}).$$

We might choose instead

$$\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1}) = p(x_j|\boldsymbol{x}_{j-1})$$

which typically is straightforward to sample from, and yields the recursive weight calculation

$$w_j(\boldsymbol{x}_j) = w_{j-1}(\boldsymbol{x}_{j-1}) p_{\mathrm{Y}|\mathrm{X}}(y_j|x_j)$$

Example (Stochastic volatility model)

Suppose

$$\text{Observation equation} \quad : \quad p_{Y|X}(y_j|x_j) \equiv \mathcal{N}(0, \exp\{2x_j\})$$

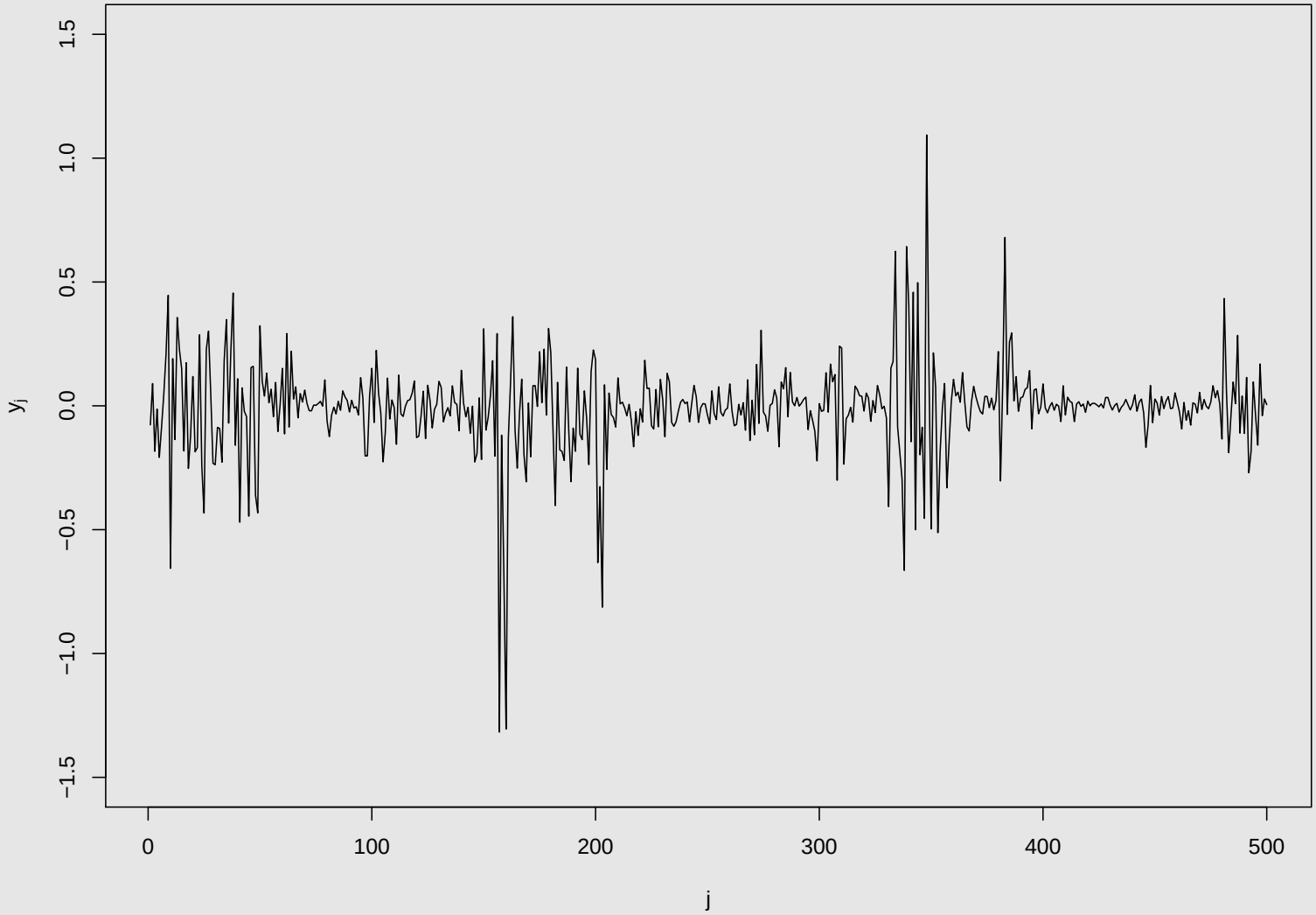
$$\text{State equation} \quad : \quad p_X(x_j|x_{j-1}) \equiv \mathcal{N}(\mu + \phi(x_{j-1} - \mu), \sigma^2)$$

with $p_{X_1}(x) \equiv \mathcal{N}(\mu, \sigma^2/(1-\phi^2))$. We regard μ, ϕ and σ as known constants.

This model is used in the analysis of financial time series: it is a non-linear state-space model, due to the non-linear dependence of the distribution of y_j on x_j.

Example (Stochastic volatility model)

$n = 500$ realizations from the stochastic volatility model.



Example (Stochastic volatility model)

Here, for $j \geqslant 2$,

$$p_{Y|X}(y_j|x_j) = \left(\frac{e^{-2x_j}}{2\pi}\right)^{1/2} \exp\{-e^{-2x_j} y_j^2/2\}$$

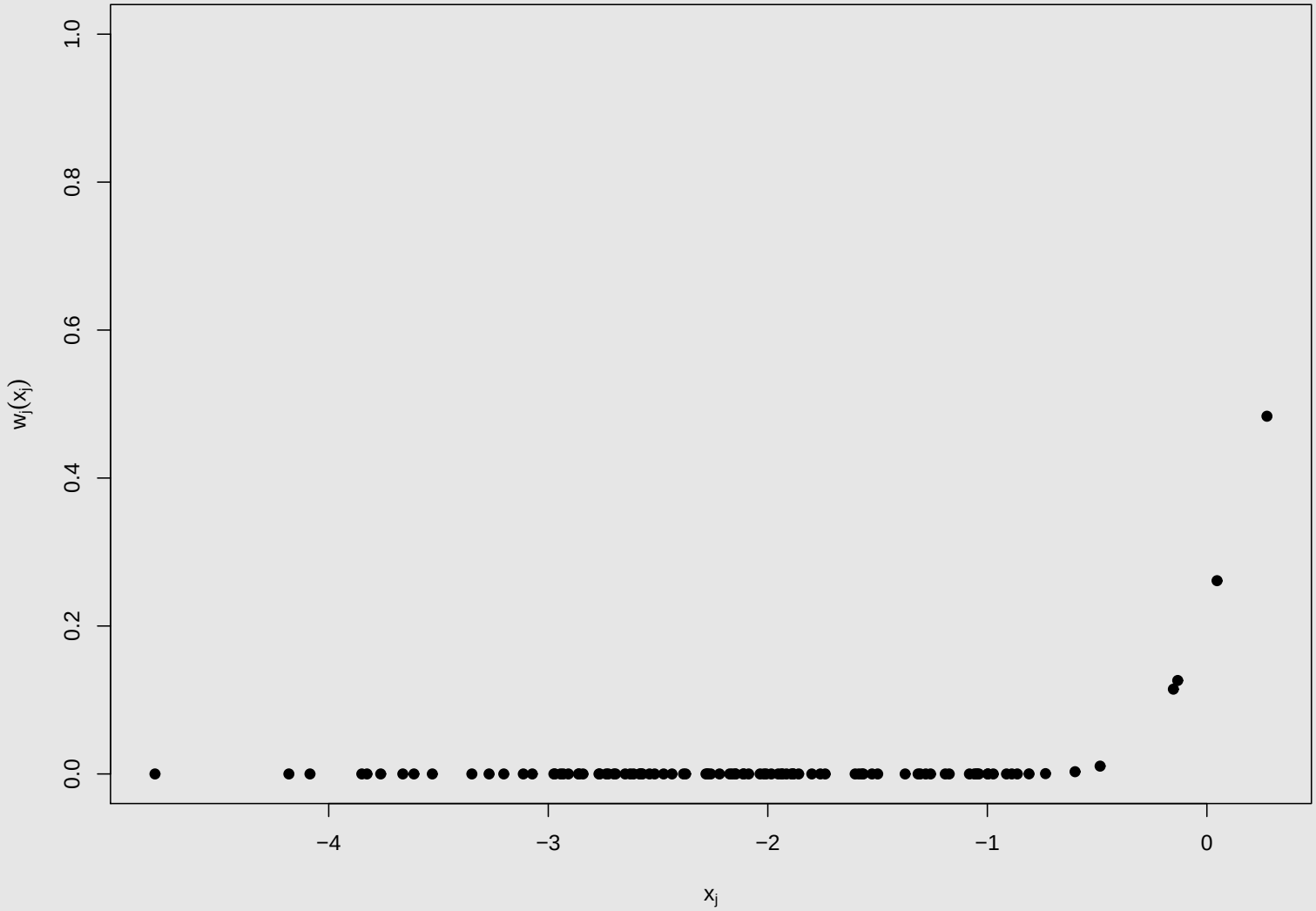
and the recursive weight calculation is based on

$$w_j(\boldsymbol{x}_j) = \prod_{k=1}^{j} p_{Y|X}(y_k|x_k)$$

that is, the likelihood to observation j.

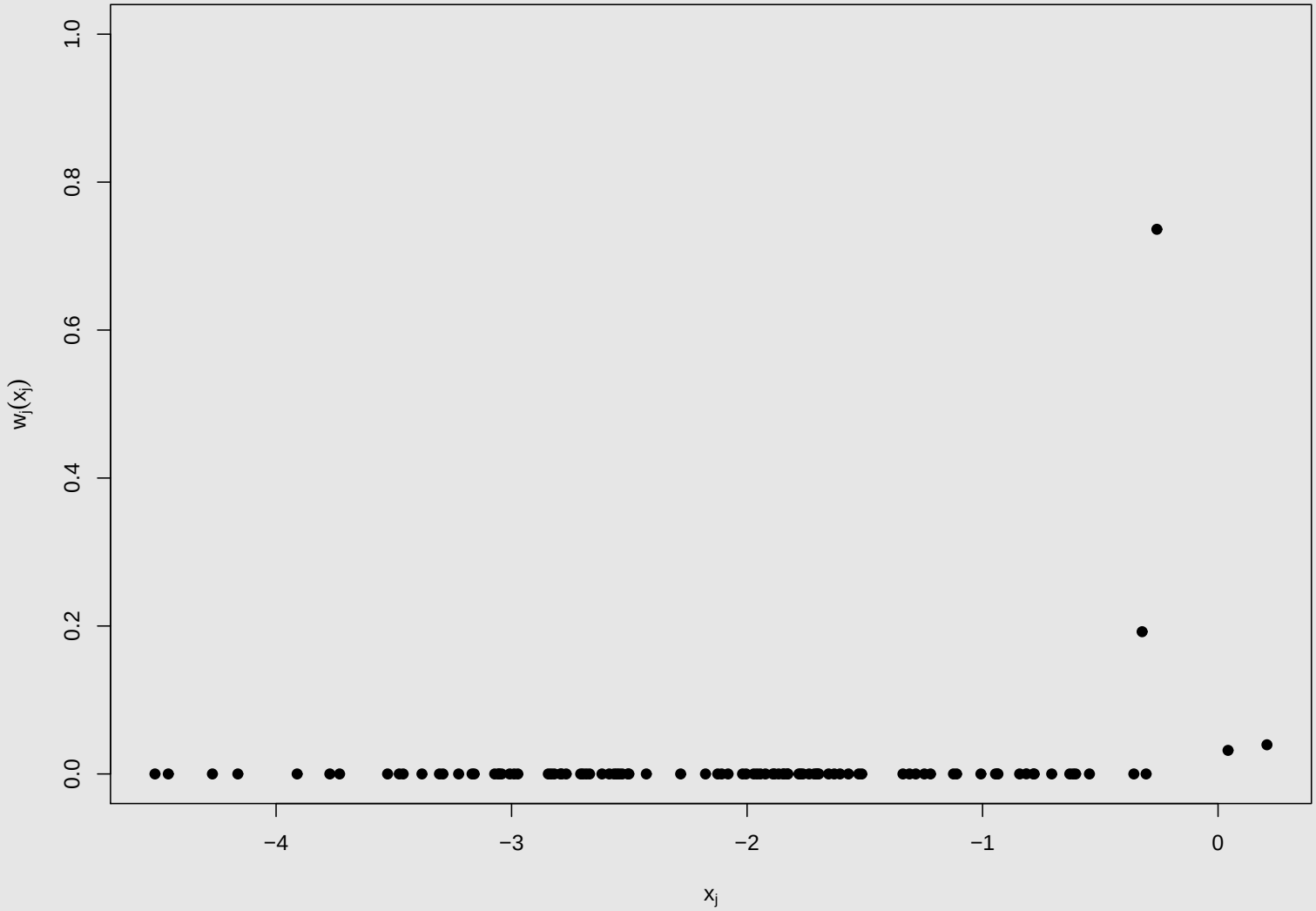
Example (Stochastic volatility model)

$N = 100$ particles and weights at $j = 1$



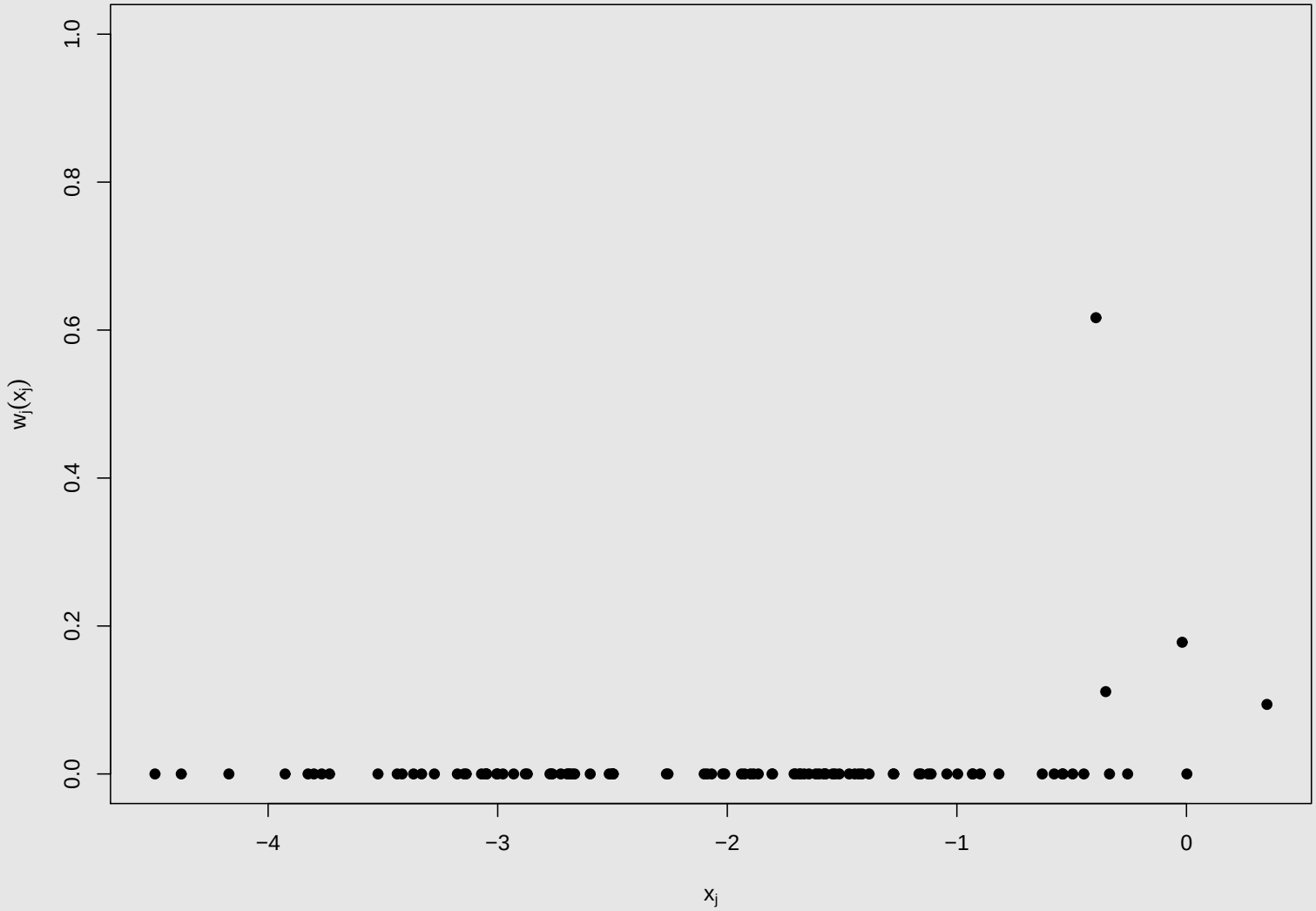
Example (Stochastic volatility model)

$N = 100$ particles and weights at $j = 2$



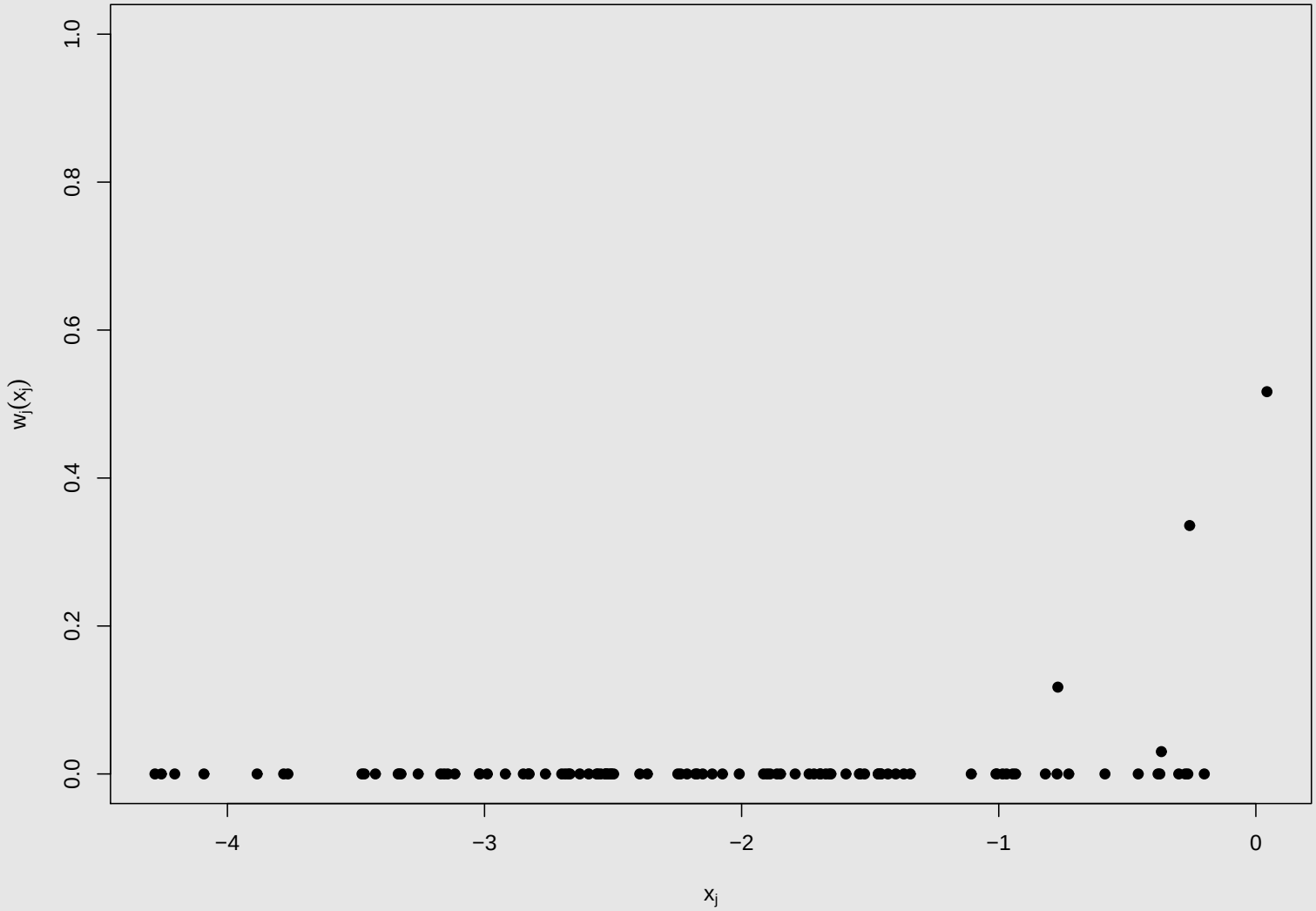
Example (Stochastic volatility model)

$N = 100$ particles and weights at $j = 3$



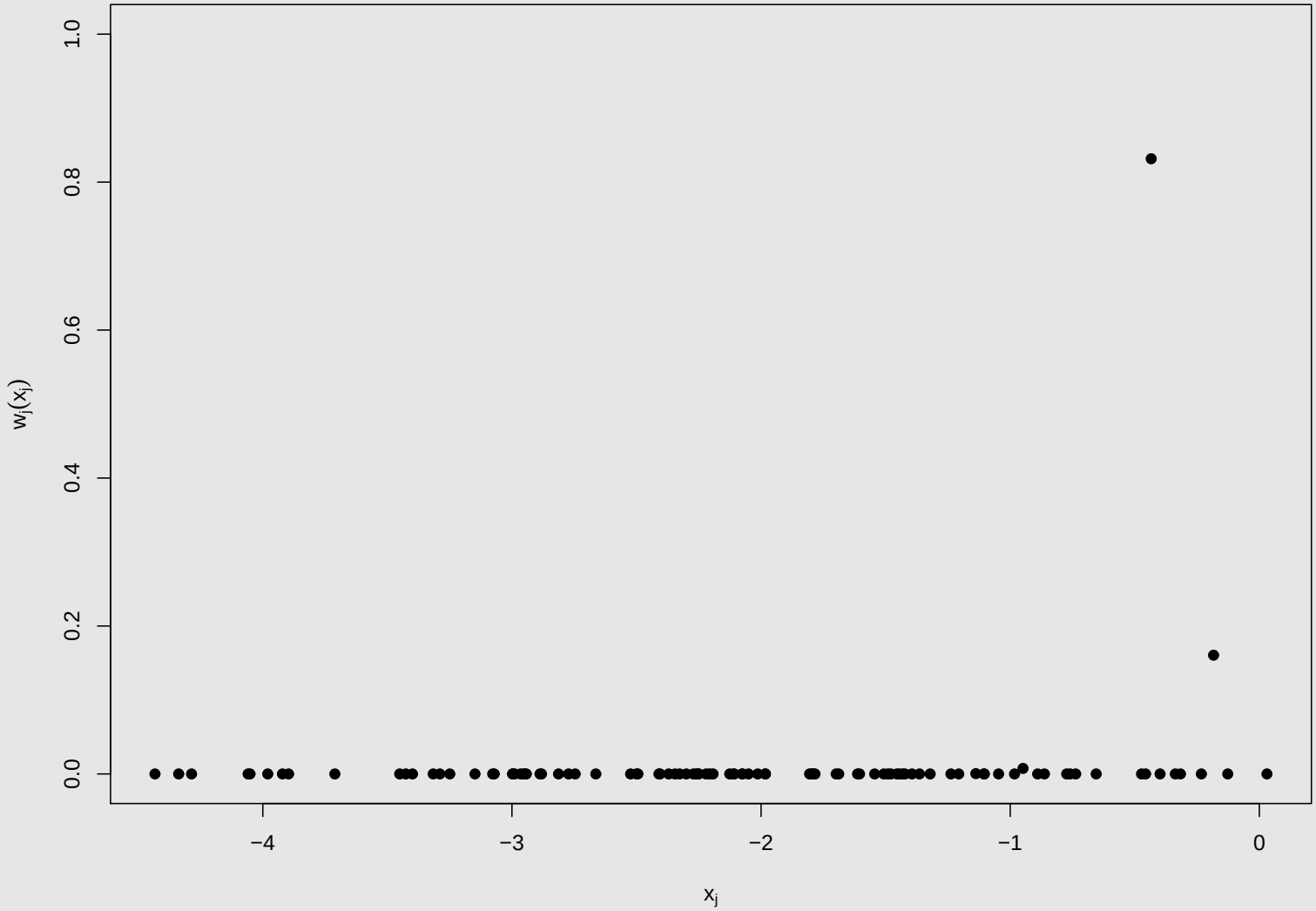
Example (Stochastic volatility model)

$N = 100$ particles and weights at $j = 4$



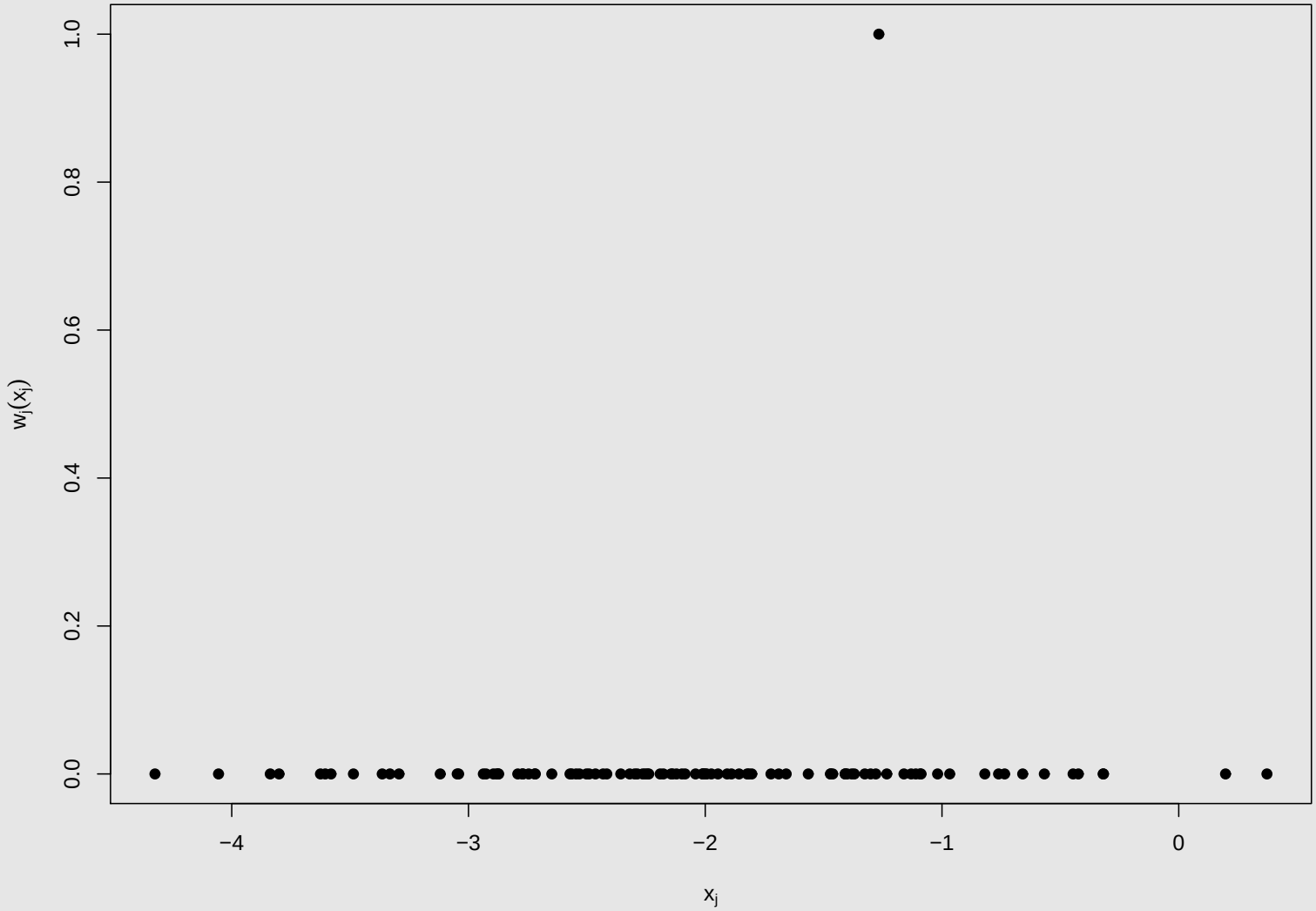
Example (Stochastic volatility model)

$N = 100$ particles and weights at $j = 5$



Example (Stochastic volatility model)

$N = 100$ particles and weights at $j = 20$



Particle Degeneracy

A problem with this strategy is that the weights start to **degenerate** as j increases;

- the weights that define the Monte Carlo estimate $\widehat{\pi}_n(\boldsymbol{x})$ are very small for most i, with only a few large weights.
- the empirical variance of the collection of unnormalized weights $\{w_j(x_j^{(i)})\}$ for $i = 1, \ldots, N$ increases
- this is mitigated, but not resolved by a better choice of $\widetilde{p}_j(x_j|\boldsymbol{x}_{j-1})$
- could try a normal approximation of

$$\begin{aligned} p(x_j|\boldsymbol{y}_j, \boldsymbol{x}_{j-1}) &\propto p_{Y|X}(y_j|x_j)p_X(x_j|x_{j-1}) \\ &= \exp\left\{-x_j - \frac{y_j^2}{2e^{2x_j}} - \frac{1}{2\sigma^2}(x_j - \mu_j)^2\right\} \end{aligned}$$

where $\mu_j = \mu + \phi(x_{j-1} - \mu)$.

Example (Stochastic volatility model)

Here the mode of

$$p(x|y_j, x_{j-1}) = \exp\left\{-x - \frac{y_j^2}{2e^{2x}} - \frac{1}{2\sigma^2}(x-\mu_j)^2\right\}$$

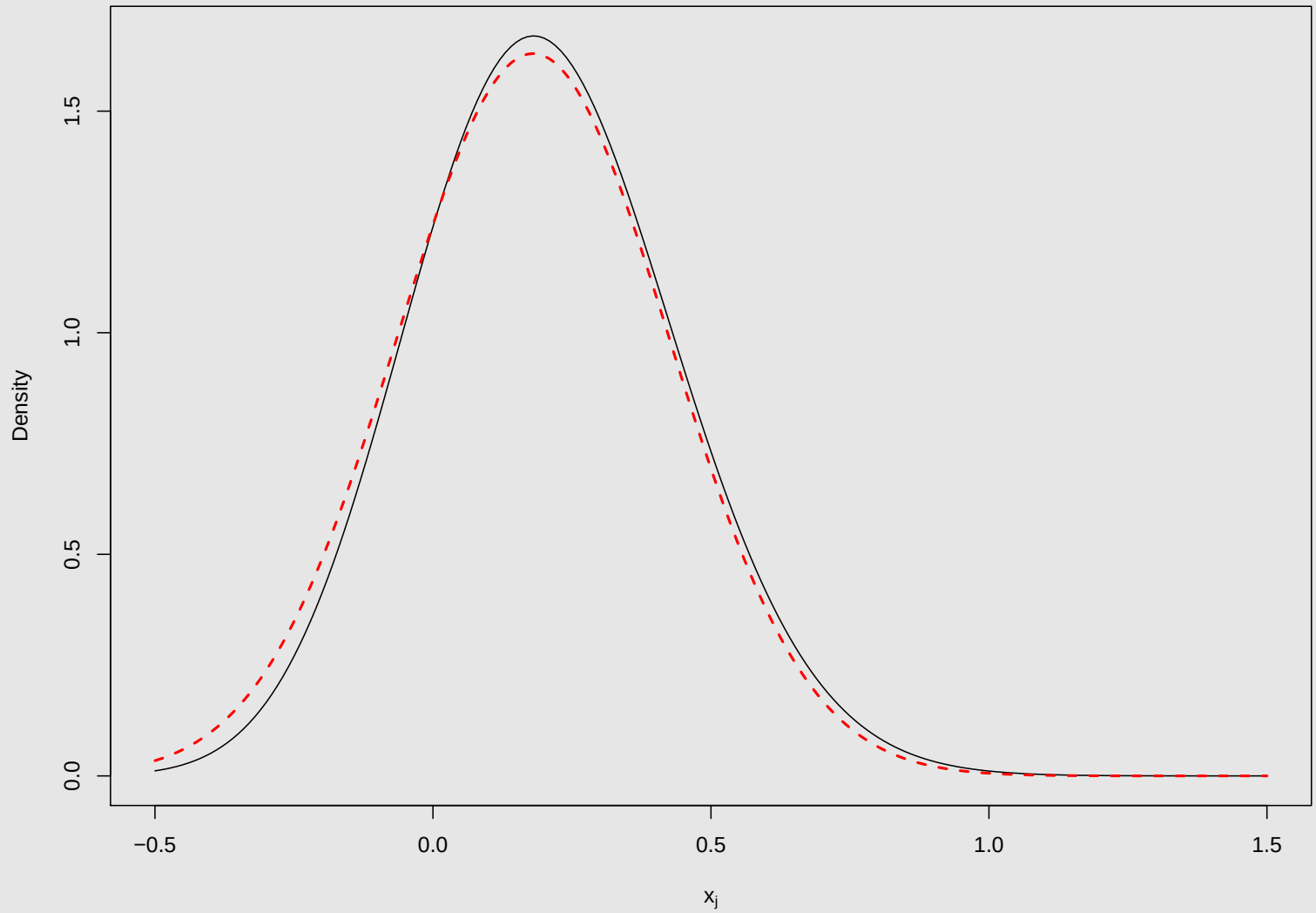
can be computed numerically, and the variance of the normal approximation can be computed by inspecting the curvature at the mode, given by

$$\left[-\frac{\partial^2 \log p(x|y_j, x_{j-1})}{\partial x^2}\right]^{-1} = \left[2y_j^2 e^{-2x} + 1/\sigma^2\right]^{-1}$$

Particle Degeneracy

Example (Stochastic volatility model)

Normal approximation to $p(x_j|y_j, x_{j-1})$ for $x_{j-1} = 0, y_j = 2$:



Particle Degeneracy

The main reason for particle degeneracy is that particles with low weight at step $j-1$ tend to have low weight at step j.

Particle degeneracy can be overcome by *resampling* that removes the particles with low weights. The commonest form of resampling is to use a non-parametric bootstrap, with multinomial sampling.

At step j, the Monte Carlo estimate of π_j is

$$\widehat{\pi}_j(\boldsymbol{x}) = \sum_{i=1}^{N} W_j(\boldsymbol{x}_j^{(i)}) \delta_{\boldsymbol{x}_j^{(i)}}(\boldsymbol{x})$$

This discrete distribution on

$$\boldsymbol{x}_j^{(1)}, \ldots, \boldsymbol{x}_j^{(N)}$$

is resampled N times according to the weights

$$W_j(\boldsymbol{x}_j^{(1)}), \ldots, W_j(\boldsymbol{x}_j^{(N)}).$$

Particle Degeneracy

After resampling, the new resampled values replace the

$$\boldsymbol{x}_j^{(1)}, \ldots, \boldsymbol{x}_j^{(N)}$$

collection, and each resampled point carries with it its normalized resampling weight $W_j(\boldsymbol{x}_j^{(i)})$.

Each original value $\boldsymbol{x}_j^{(i)}$ may appear in the resampled collection more than once.

Particle Degeneracy

Let $n_{j1}, \ldots, n_{jN}$ denote the numbers of times that particles $\boldsymbol{x}_j^{(1)}, \ldots, \boldsymbol{x}_j^{(N)}$ are resampled during multinomial sampling. Let

$$\widetilde{\pi}_j(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{n_{ji}}{N} \delta_{\boldsymbol{x}_j^{(i)}}(\boldsymbol{x})$$

Resampling does not disrupt the particle approximation in expectation (that is, it does not introduce bias)

$$\mathbb{E}[\widetilde{\pi}_j(\boldsymbol{x}) | \widehat{\pi}_j(\boldsymbol{x})] = \widehat{\pi}_j(\boldsymbol{x})$$

but it does introduce "noise" - for any suitable h,

$$\mathbb{V}ar_{\widetilde{\pi}_j}[h(\boldsymbol{X}_j)] \geqslant \mathbb{V}ar_{\widehat{\pi}_j}[h(\boldsymbol{X}_j)]$$

This follows by the law of iterated variance.

Particle Degeneracy

The message from this result is that although you should perform resampling to preserve particle diversity, you should not resample too often, as that introduces variability.

A common strategy is to resample only when the observed variance of the weights exceeds some threshold. A statistic to track is the *effective sample size* (ESS) where

$$\text{ESS} = \left(\sum_{i=1}^{N} \left\{ W_j^{(i)} \right\}^2 \right)^{-1}$$

We have $1 \leqslant \text{ESS} \leqslant N$:

$$\text{ESS} = 1 \quad \text{if} \quad W_j^{(i)} = 1 \quad \text{for some } i$$

$$\text{ESS} = N \quad \text{if} \quad W_j^{(i)} = \frac{1}{N} \quad \text{for each } i$$

Particle Degeneracy

Can also use *entropy* as a measure

$$\text{Entropy} = -\sum_{i=1}^{N} W_j^{(i)} \log W_j^{(i)}$$

where $0 \leqslant \text{Entropy} \leqslant \log_2 N$

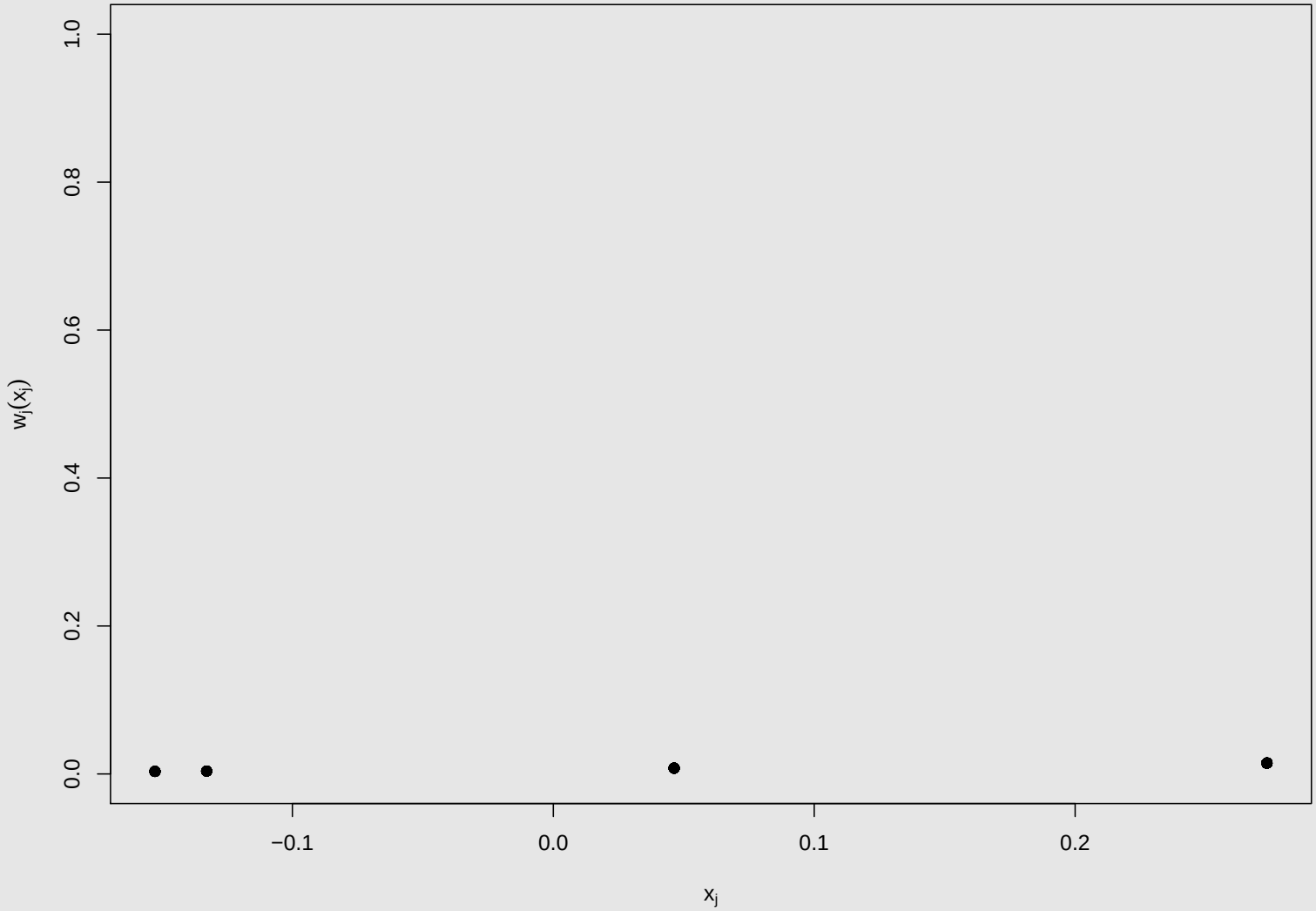
$$\text{Entropy} = 0 \quad \text{if} \quad W_j^{(i)} = 1 \quad \text{for some } i$$

$$\text{Entropy} = \log_2 N \quad \text{if} \quad W_j^{(i)} = \frac{1}{N} \quad \text{for each } i$$

Particle Degeneracy

Example (Stochastic volatility model)

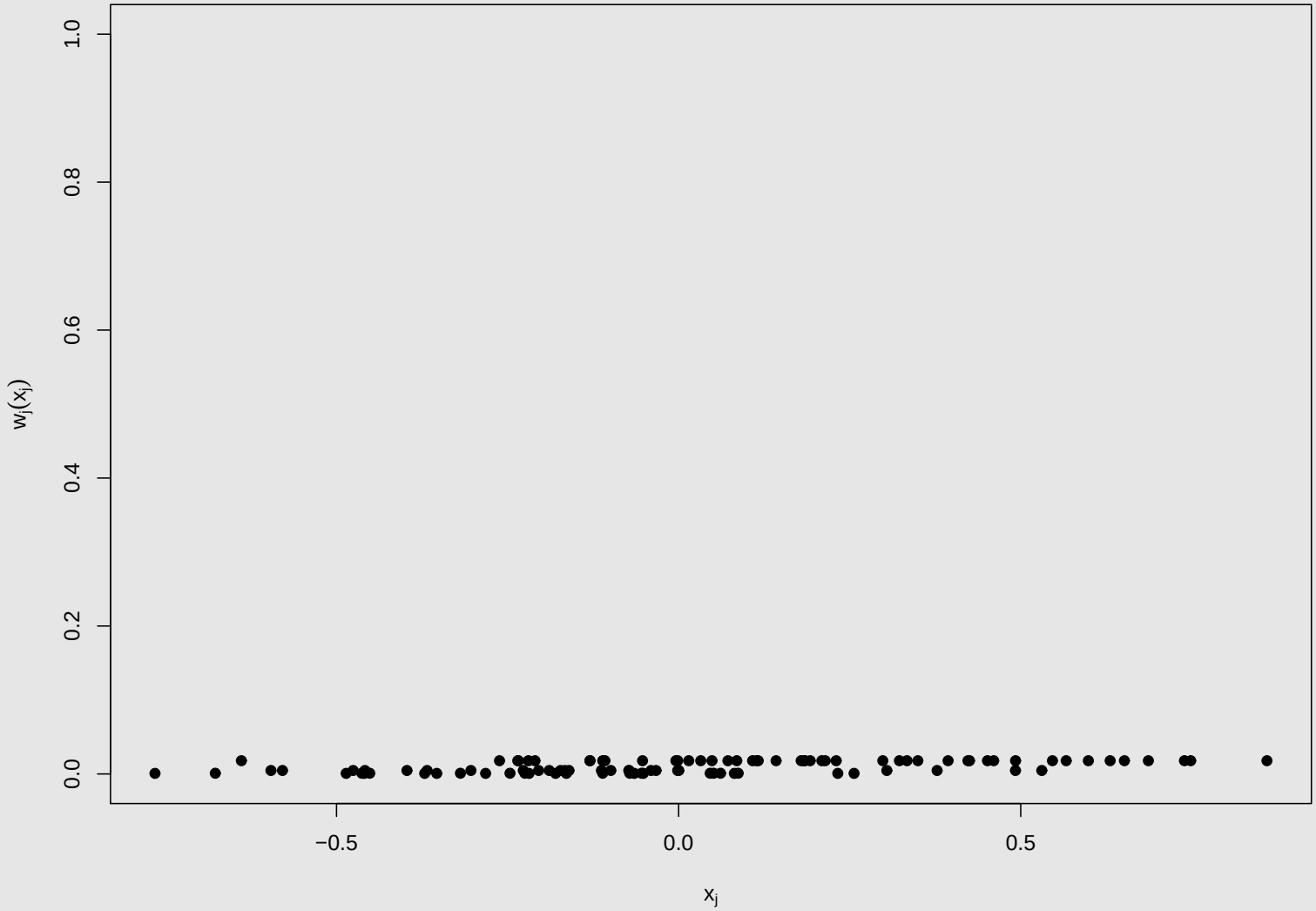
$N = 100$ resampled particles and weights at $j = 1$



Particle Degeneracy

Example (Stochastic volatility model)

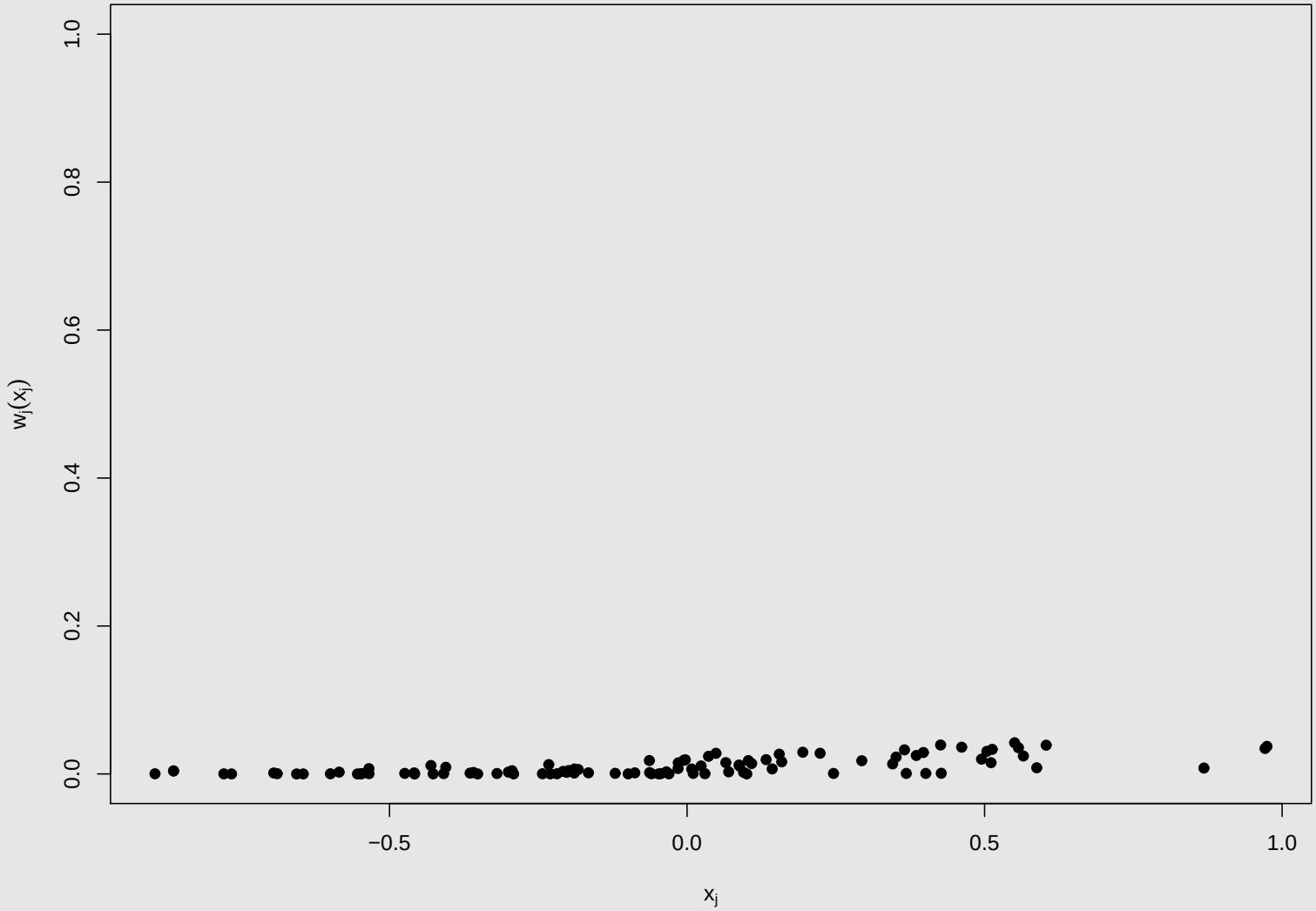
$N = 100$ resampled particles and weights at $j = 2$



Particle Degeneracy

Example (Stochastic volatility model)

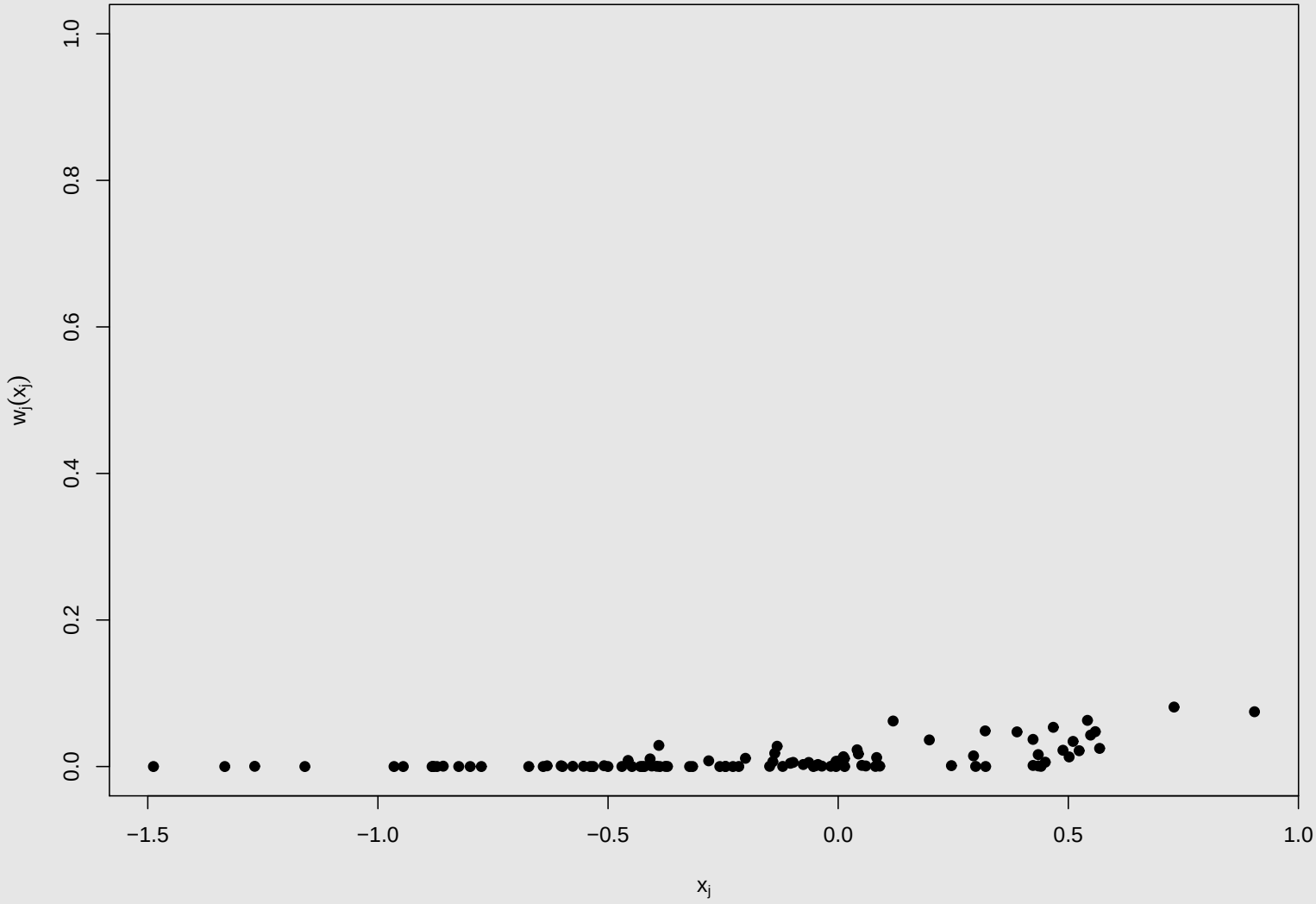
$N = 100$ resampled particles and weights at $j = 3$



Particle Degeneracy

Example (Stochastic volatility model)

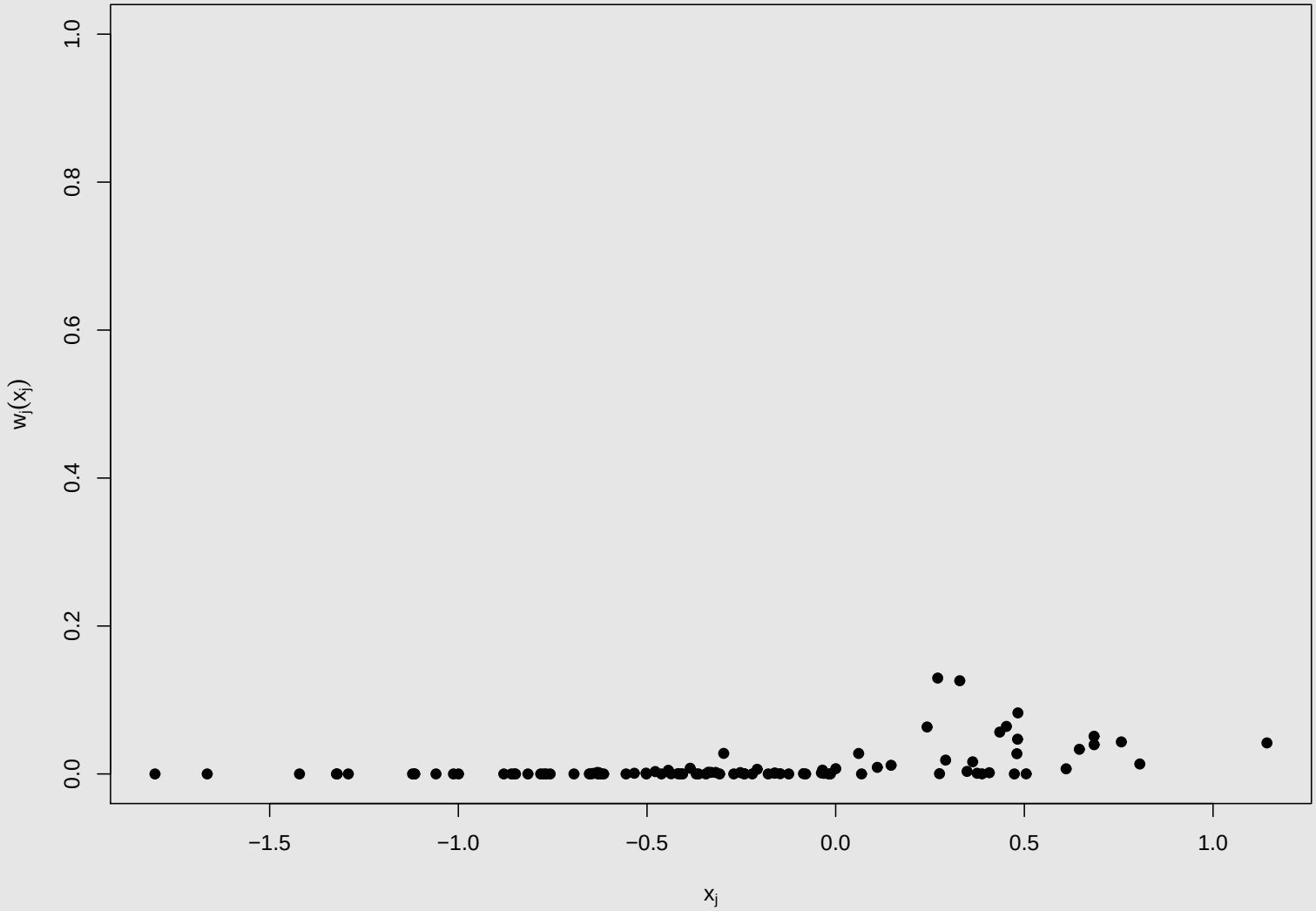
$N = 100$ resampled particles and weights at $j = 4$



Particle Degeneracy

Example (Stochastic volatility model)

$N = 100$ resampled particles and weights at $j = 5$



Particle Degeneracy

Example (Stochastic volatility model)

Effective sample size (ESS) with resampling when ESS < 10

