Particle filters

Recall the setting for Kalman filtering:

Evolution–observation model:

$$\begin{aligned} X_{j+1} &= AX_j + V_{j+1}, \quad j = 0, 1, 2, \ldots \\ Y_j &= BX_j + E_j, \quad j = 1, 2, \ldots . \end{aligned}$$

The first equation is used for *prediction*, the second equation for *correction* of the prediction.

Kalman filtering is based on the assumption that everything is Gaussian.

Normality: Mean and covariance determine the density.

Assumptions of the noise processes and the initial process:

1. Normality:

$$V_j \sim \mathcal{N}(0, \Gamma_j), \quad E_j \sim \mathcal{N}(0, \Sigma_j).$$

2. Independency: Variables V_j, E_j, all mutually independent.

3. Initial density:

$$X_0 \sim \mathcal{N}(x_0, D_0),$$

   and X_0 is independent of the noise processes.

Limitations

The Kalman filtering is not applicable if

- the model is not linear with additive noise
- any of the assumptions 1.–3. fail.

Non-Gaussian densities: exploration by sampling.

Dynamic sampler requires two steps

1. *Propagation* of the sample points, called *particles*.
2. *Resampling* of the particles when the observation data arrives.

General evolution-observation model

We consider the more general model

$$\begin{aligned} X_{j+1} &= F\big(X_j, V_{j+1}\big), \quad j = 0, 1, 2, \dots \\ Y_j &= G\big(X_j, E_j\big), \quad j = 1, 2, \dots. \end{aligned}$$

The functions F and G are assumed known.

For simplicity, it is assumed here that F and G are time invariant. More generally, they could be different at each step.

For simplicity, we assume also that

- V_{j+1} is independent of X_j,
- E_j is independent of X_j.

Initialization

As in Kalman filtering, we assume an *a priori* probability density $\pi(x_0)$ for X_0.

Step 1: Generate a sample

$$S_0 = \left\{x_0^1, x_0^2, \ldots, x_0^N\right\}$$

by drawing from the density $\pi(x_0)$.

Observe: if the initial density is complicated (e.g. non-Gaussian), the generation of the initial sample may require the use of MCMC methods.

PROPAGATION

Suppose that we have a sample

$$S_k = \{x_k^1, x_k^2, \ldots, x_k^N\}$$

of points that are distributed according to the probability density

$$\pi(x_k \mid y_1, y_2, \ldots, y_k).$$

Step 2: Draw a sample of the evolution noise realizations

$$\{v_{j+1}^1, v_{j+1}^2, \ldots, v_{j+1}^N\}$$

from the distribution $\pi(v_{j+1})$ of the random variable V_{j+1}.

Calculate the propagated prediction sample

$$\widetilde{S}_{k+1} = \{\widetilde{x}_{k+1}^1, \widetilde{x}_{k+1}^2, \ldots, \widetilde{x}_{k+1}^N\}$$

by the propagation formula

$$\widetilde{x}_{k+1}^j = F(x_k^j, v_{k+1}^j), \quad 1 \leq j \leq N.$$

Correction

Assume that from the observation model

$$Y_k = G(X_k, E_k),$$

we can calculate the likelihood density,

$$\pi(y_k \mid x_k), \quad k = 1, 2, \ldots$$

up to a multiplicative constant, denoted here by C.

Step 3: With $y_{k+1} = y_{k+1,\mathrm{obs}}$, calculate the *importance* of each propagated particle:

$$\widetilde{w}_{k+1}^j = C\pi(y_{k+1} \mid \widetilde{x}_{k+1}^j), \quad 1 \le j \le N,$$

and further, the *relative importance* by scaling,

$$w_{k+1}^j = \frac{\widetilde{w}_{k+1}^j}{W}, \quad W = \sum_{j=1}^N \widetilde{w}_{k+1}^j.$$

Correction (cont.)

Now we have the predicted sample, with attached relative importance weights,

$$\left\{ (\widetilde{x}_{k+1}^1, w_{k+1}^1), (\widetilde{x}_{k+1}^2, w_{k+1}^2), \ldots, (\widetilde{x}_{k+1}^N, w_{k+1}^N) \right\} .$$

Step 4: Resampling: draw a new sample

$$S_{k+1} = \left\{ x_{k+1}^1, x_{k+1}^2, \ldots, x_{k+1}^N \right\}$$

from the sample $\widetilde{S}_{k+1}$ drawing the particles according to their relative importance w_{k+1}^j.

The algorithm described above is referred to as *Sampling Importance Resampling (SIR).*

Implementation of the resampling step:

- Divide the unit interval in pieces I^j of length w_{k+1}^j. Notice that

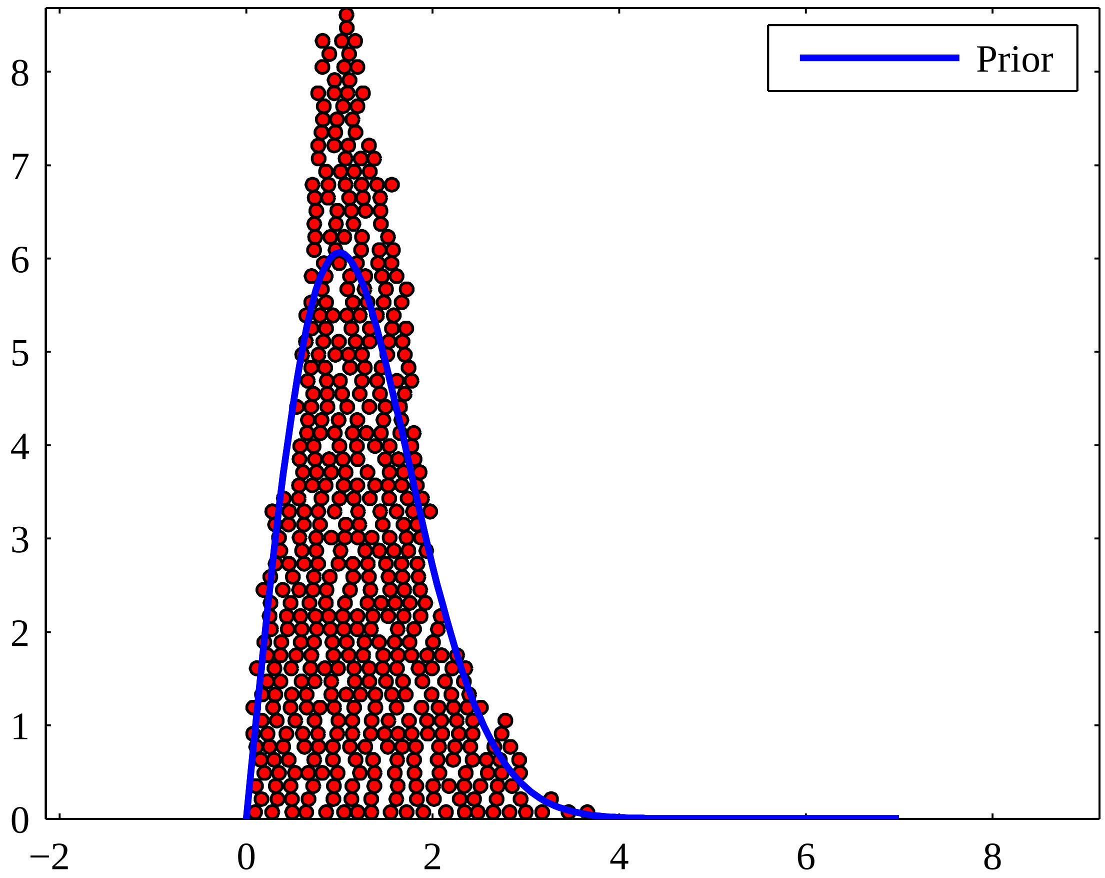
$$\sum_{j=1}^{N} w_{k+1}^j = 1.$$

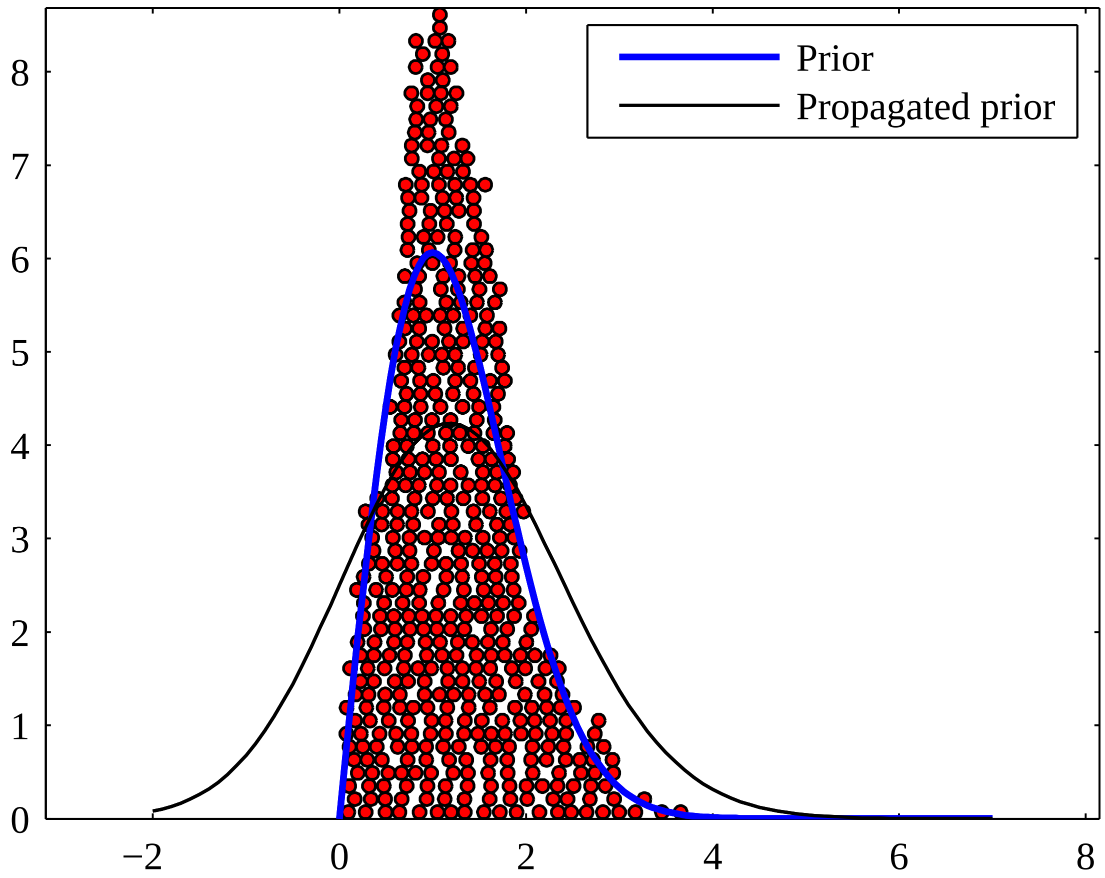
- Repeat for $\ell = 1, \ldots, N$:

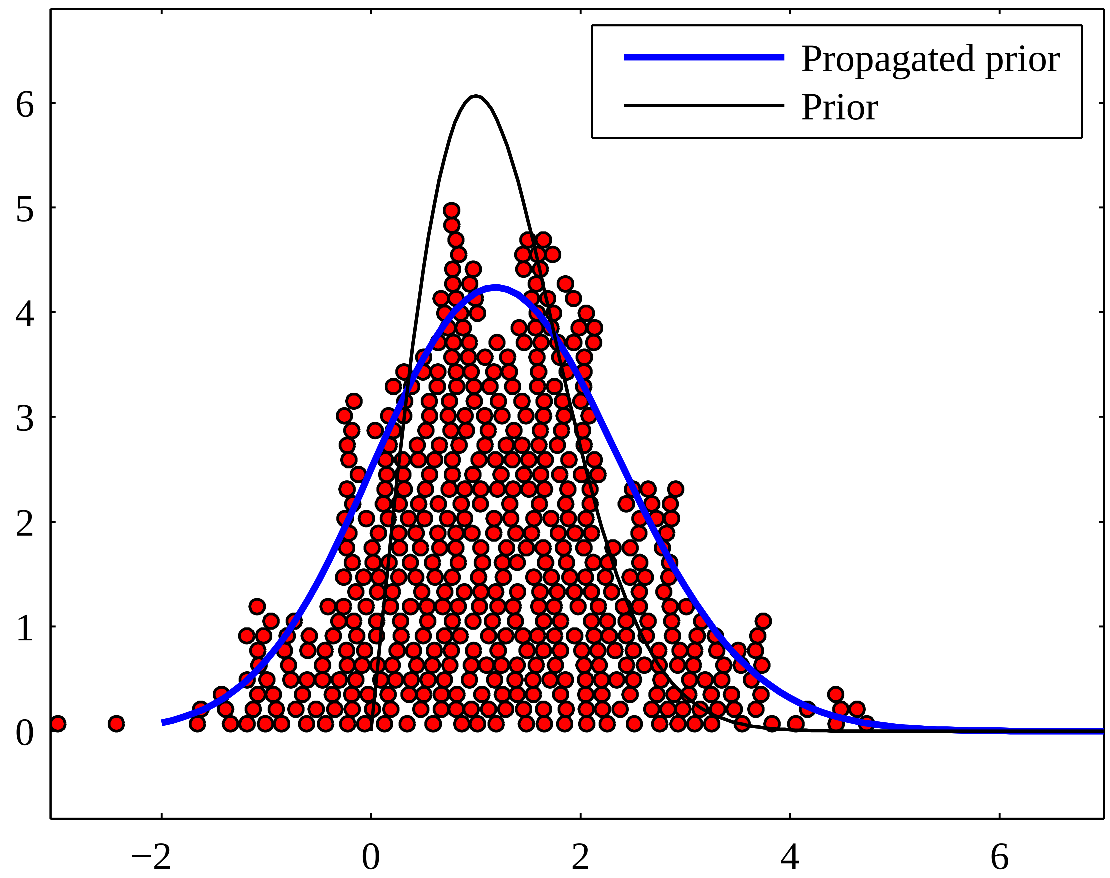
  1. Draw $\xi \sim \text{Uniform}([0,1])$,
  2. If $\xi \in I^j$, set $x_{k+1}^\ell = \widetilde{x}_{k+1}^j$.

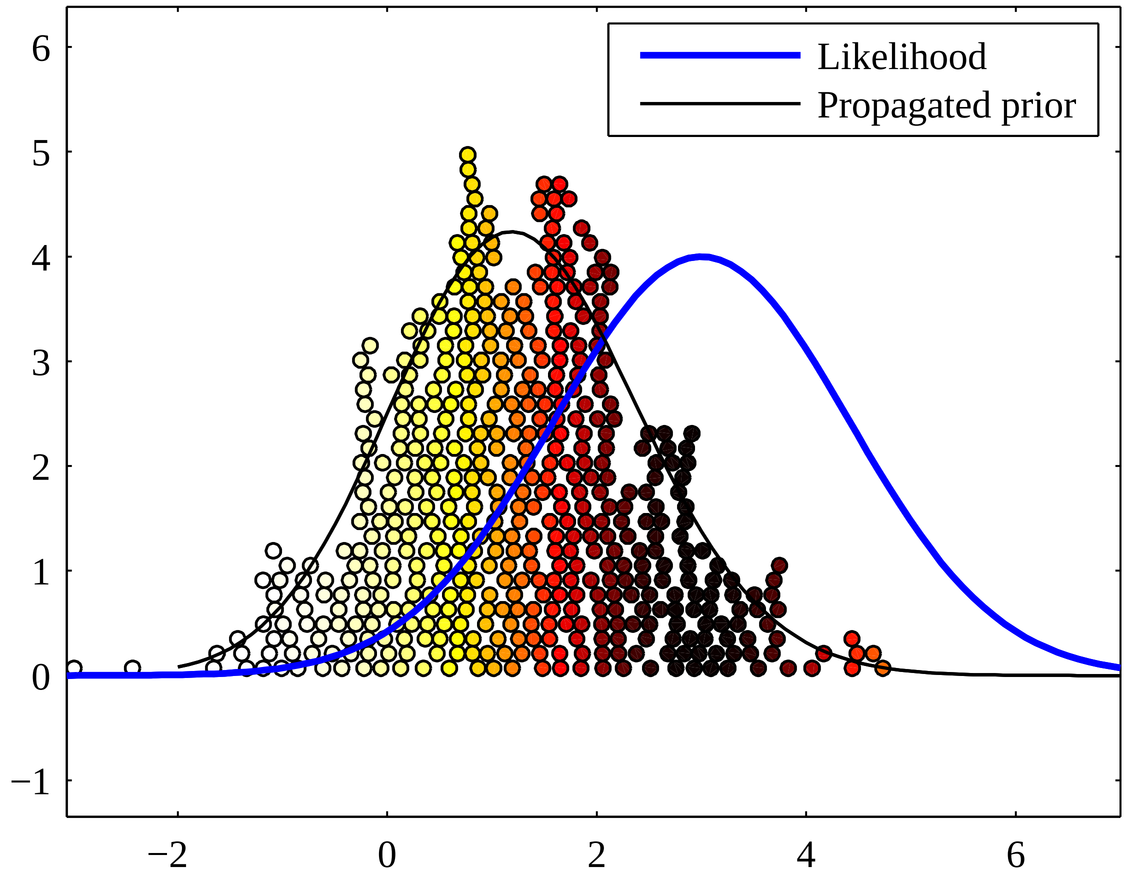
Observations

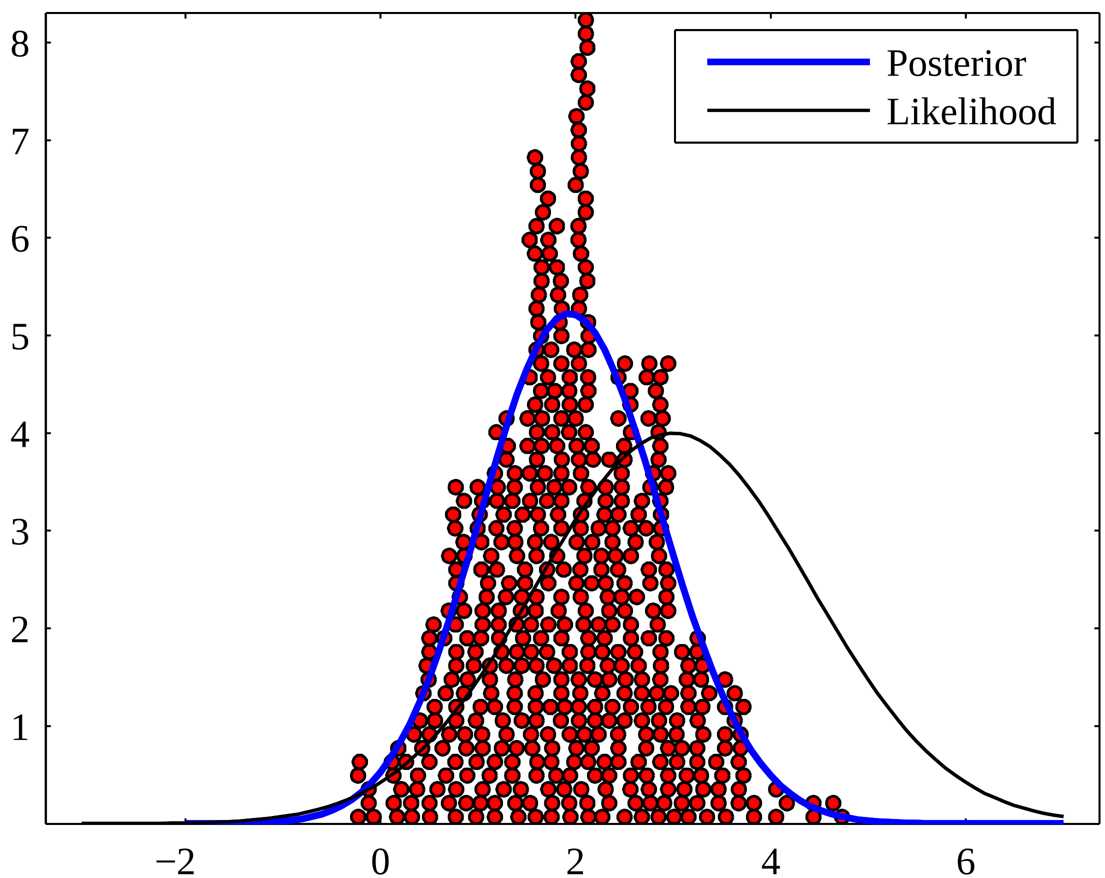
- The same particle $\widetilde{x}^j_{k+1}$ may appear in the final sample several times.
- In fact, if w^j_{k+1} is large, it is likely that the jth particle appears several times.
- The phenomenon that the final sample contains copies of only very few propagated particles is called *data thinning*.
- Data thinning is a typical phenomenon if the likelihood is very narrow.


Prior
8
7
6
5
4
3
2
1
0
−2
0
2
4
6
8


Prior
Propagated prior
8
7
6
5
4
3
2
1
0
−2
0
2
4
6
8


Propagated prior
Prior
6
5
4
3
2
1
0
−2
0
2
4
6


Likelihood
Propagated prior
6
5
4
3
2
1
0
−1
−2
0
2
4
6


Posterior
Likelihood
8
7
6
5
4
3
2
1
−2
0
2
4
6

Biomagnetic inverse problem

A single planar dipole moves in the plane $P = \{p = [p_1; p_2; 0]\}$.

Vertical component of the resulting magnetic field is observed above the plane.

Data:

$$b(x) = \begin{bmatrix} b_1(x) \\ \vdots \\ b_L(x) \end{bmatrix} \in \mathbb{R}^L, \quad b_j(x) = \frac{\mu_0}{4\pi} \frac{e_z \cdot q \times (r_j - p)}{|r_j - p|^3},$$

$$q = \text{dipole moment} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \in \mathbb{R}^2$$

Time dependent dipole:

$$p = p(t), \quad q = q(t)$$

Discrete time, $t = t_k$. Model parameter:

$$x_k = \begin{bmatrix} p_1(t_k) \\ p_2(t_k) \\ q_1(t_k) \\ q_2(t_k) \end{bmatrix} \in \mathbb{R}^4.$$

Random walk model

$$X_{k+1} = X_k + V_{k+1},$$

where the covariance of V_{k+1} is

$$\Gamma = \mathrm{diag}(\lambda^2, \lambda^2, \delta^2, \delta^2) \in \mathbb{R}^{4\times 4},$$

The model corresponds to the Markov transition kernel

$$\pi(x_{x+1} \mid x_x) \propto \exp\left(-\frac{1}{2}(x_{x+1} - x_x)^{\mathrm{T}}\Gamma^{-1}(x_{x+1} - x_x)\right).$$

The observation model:

$$Y_k = b(X_k) + E_k,$$

where E_k is independent of X_j, $j \leq k$ and Gaussian with zero mean and variance Γ_{noise} known.

Likelihood:

$$\pi(y_k \mid x_k) \propto \exp\left(-\frac{1}{2}(y_k - b(x_k))^{\mathrm{T}}\Gamma_{\mathrm{noise}}^{-1}(y_k - b(x_k))\right).$$

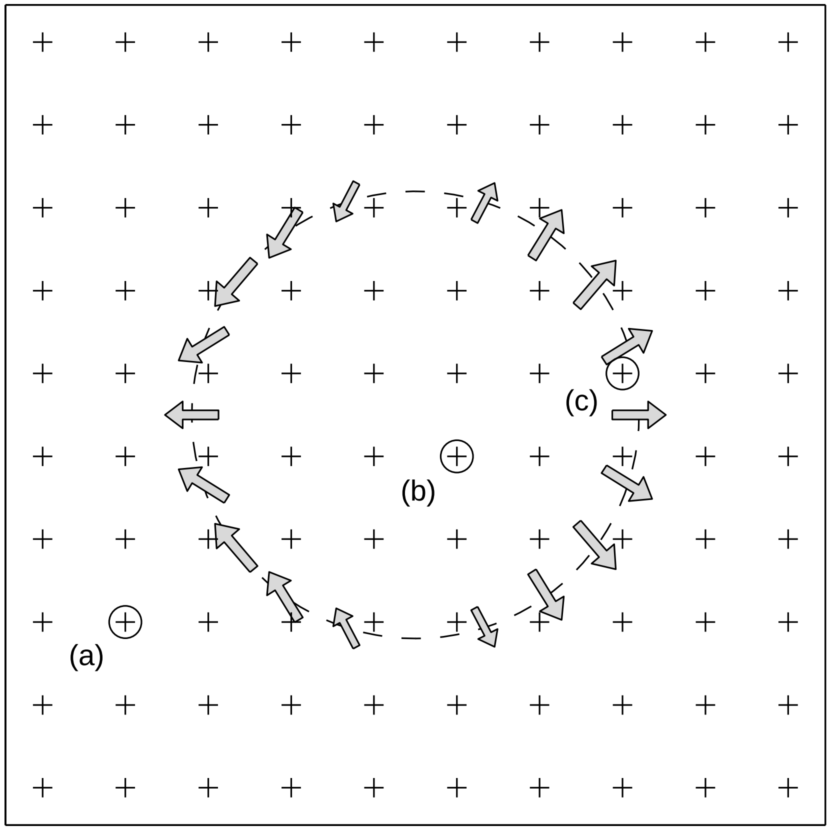
Initial prior probability density of X_0:

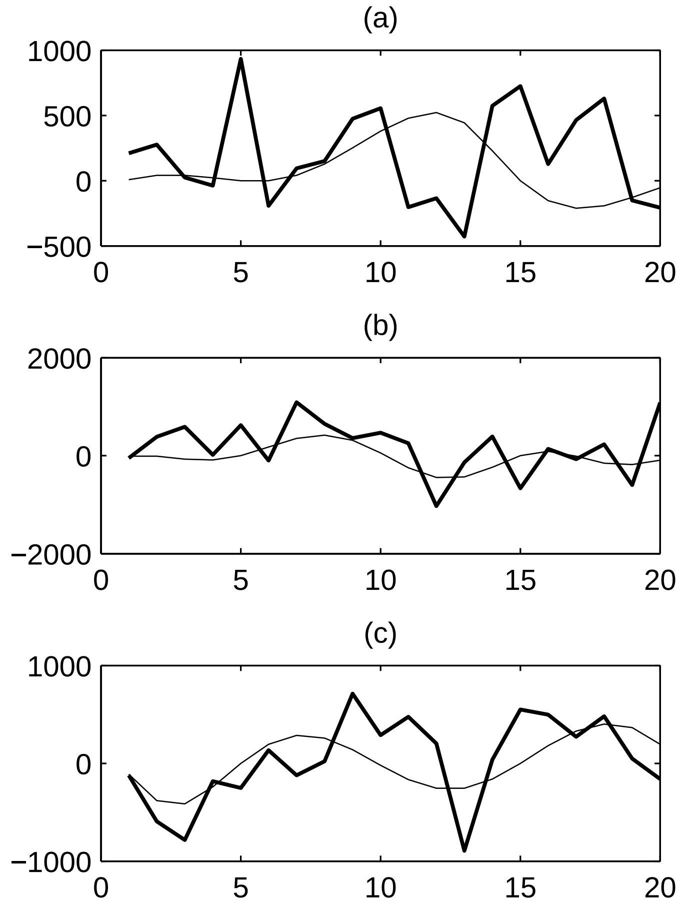
$$X_0 \sim \pi_0(x_0) \propto \exp\left(\frac{1}{2}(x_0 - \overline{x}_0)^{\mathrm{T}}\Gamma_0^{-1}(x_0 - \overline{x}_0)\right).$$

The particle filtering algorithm for single dipole tracking can be summarized as follows.

Choose sample size N, and draw $x_0^1, \ldots, x_0^N \in \mathbb{R}^4$, from π_0 and set $k = 0$.
do
  Draw $v^1, \ldots, v^N \sim \mathcal{N}(0, C)$ and define $z^j = x_k^j + v^j$, $1 \leq j \leq N$
  Calculate the relative likelihoods, $w^j = \pi(y_k \mid z^j)/W$, $W = \sum_{j=1}^N \pi(y_k \mid z^j)$
  Draw x_{k+1}^j, $1 \leq j \leq N$ from $\{z^1, \ldots, z^N\}$, the propability of z^j being w^j.
 $k \leftarrow k + 1$
end

The above loop is repeated as long as new observations y_k keep arriving.



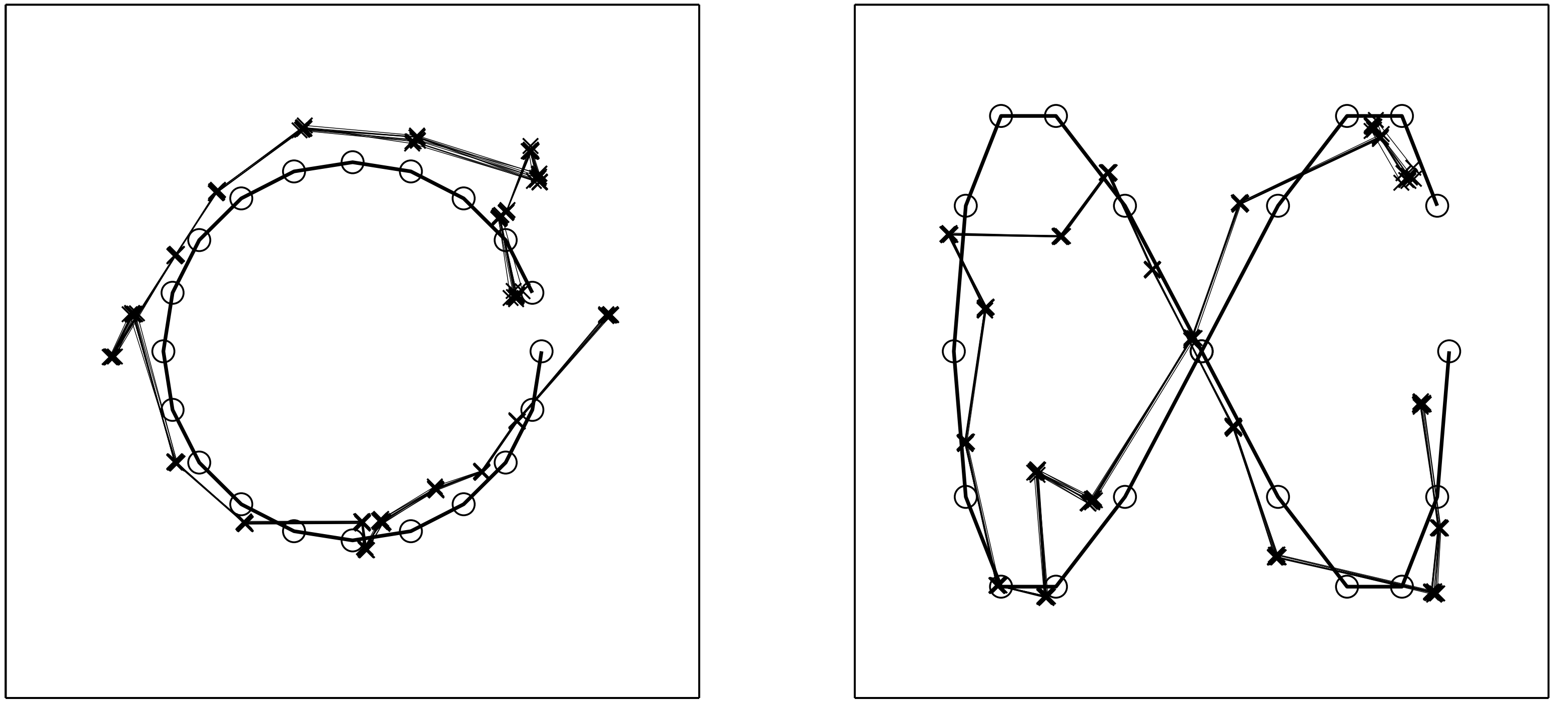


Noise level: STD = 80% of the maximum of the noiseless signal.

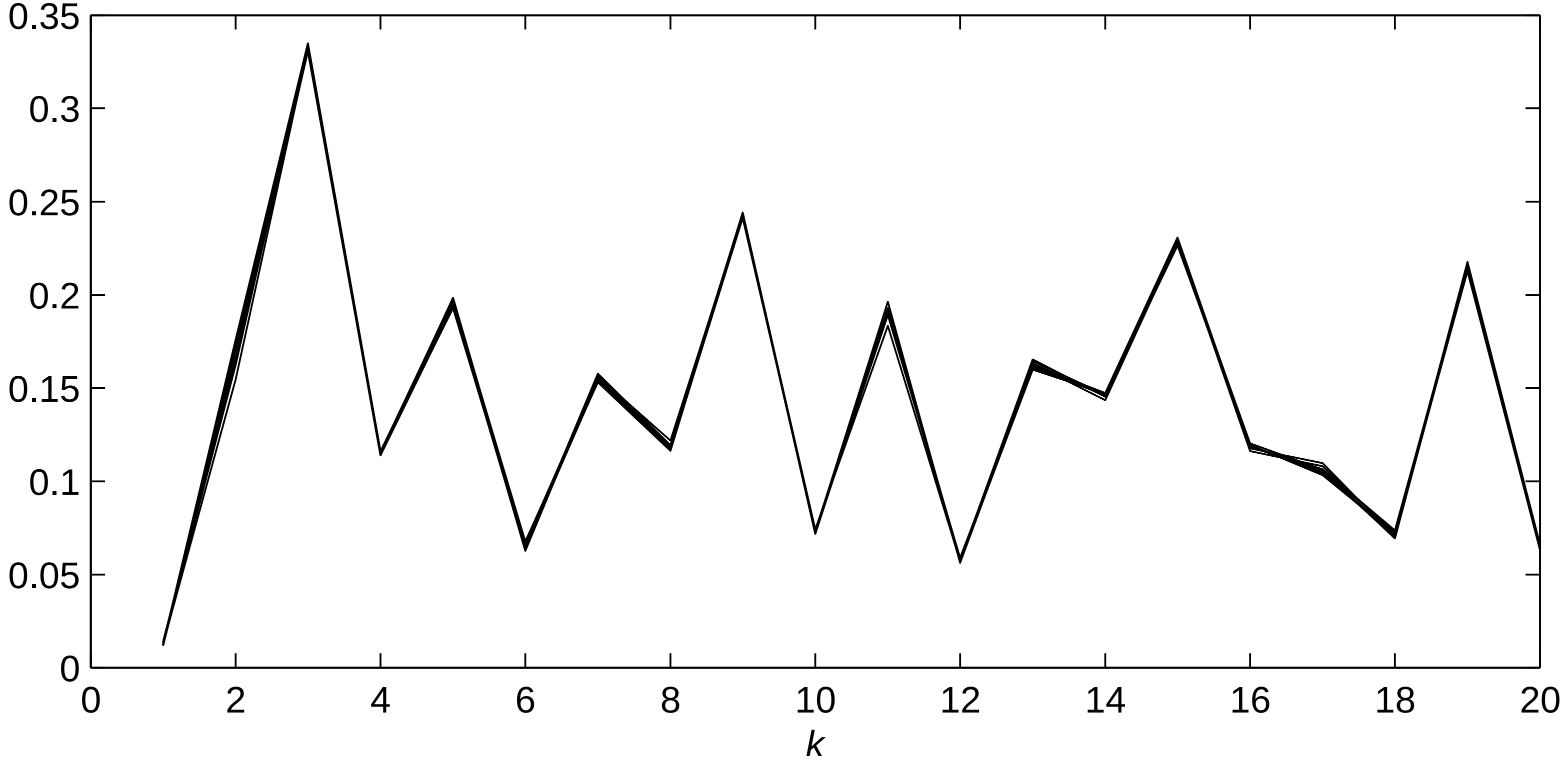
Selection of model parameters

- Number of particles: $N = 200\,000$
- Step length for location : $\lambda = 1$ units (Size of the image = 10 units per direction.)
- Step size for amplitude evolution: $\delta = 0.25$ units, about 20% of maximum dipole value in simulation.

Mean over particle sample



Diagnostics of data thinning



Relative number of particles in the prediction sample that are resampled at least once.