

Nonlinear filtering tutorial cheat sheet

Anna Kutschireiter, PhD

February 12, 2020

1 Kalman Filter

Algorithm 1 Kalman filter (discrete time)

```
1: procedure KALMAN FILTER( $y_{1:T}$ )
2:                                      $\triangleright$  Solves for mean and variance in LDS.
3: Parameters:  $F_{1:T}, Q_{1:T}, H_{1:T}, R_{1:T}, \mu_0, \Sigma_0$ 
4:    $K_1 = \Sigma_0 H_1^\top (H_1 \Sigma_0 H_1^\top + R_1)^{-1}$ 
5:    $\mu_1 = \mu_0 + K_1(y_1 - H_1 \mu_0)$ 
6:    $\Sigma_1 = (\mathbb{I} - K_1 H_1) \Sigma_0$ 
7:   for  $t = 2 \dots T$  do
8:      $P_t = F_t \Sigma_{t-1} F_t^\top + Q_t$   $\triangleright$  Variance of  $p(x_t | y_{1:t-1})$ 
9:      $K_t = P_t H_t^\top (H_t P_t H_t^\top + R_t)^{-1}$   $\triangleright$  Kalman gain
10:     $\mu_t = F_t \mu_{t-1} + K_t(y_t - H_t F_t \mu_{t-1})$   $\triangleright$  Mean of  $p(x_t | y_{1:t})$ 
11:     $\Sigma_t = (\mathbb{I} - K_t H_t) P_t$   $\triangleright$  Variance of  $p(x_t | y_{1:t})$ 
12:   end for
13:   return  $\mu_{1:T}, \Sigma_{1:T}$ 
14: end procedure
```

2 Importance sampling

Importance sampling (Algorithm 2) empirically approximates $p(x) \approx \sum_{i=1}^N w^{(i)} \delta(x - x^{(i)})$ with samples $x^{(i)} \sim q(x)$. Equivalently, it can be used to compute expectations of a function $\phi(x)$ via $\langle \phi(x) \rangle_{p(x)} \approx \sum_{i=1}^N w^{(i)} \phi(x^{(i)})$.

3 Particle filtering

A generic particle filter is given in Algorithm 3. Here, π denotes an arbitrary proposal function, Apart from the restriction that the the proposal has to factorize, i.e. $\pi(x_{0:t} | y_{1:t}) = \pi(x_0) \prod_{s=1}^t \pi(x_s | x_{0:s-1}, y_{1:s})$, we are free to choose any proposal.

Algorithm 2 Importance Sampling

```
1: procedure IMPORTANCE SAMPLING
2:     ▷ Produces weighted samples from  $p(x)$  by sampling from  $q(x)$ .
3: Parameters:  $N$ 
4:   for  $i = 1 \dots N$  do
5:      $x^{(i)} \sim q(x)$                                 ▷ Sample from proposal
6:      $\tilde{w}^{(i)} = \frac{p(x^{(i)})}{q(x^{(i)})}$                 ▷ Unnormalized importance weight
7:   end for
8:   for  $i = 1 \dots N$  do
9:      $w^{(i)} = \frac{\tilde{w}^{(i)}}{\sum_{j=1}^N \tilde{w}^{(j)}}$                 ▷ Normalized importance weight
10:  end for
11:  return  $\{x^{(i)}, w^{(i)}\}_{i=1:N}$ 
12: end procedure
```

More often than not, this algorithm will suffer from particle degeneracy: after a finite number of time steps, all but one sample trajectory will have negligible weights, with the consequence that the whole trajectory is represented by *a single sample*. To avoid this, we can include a resampling procedure (Algorithm 4) into the particle filter whenever the number of particles is low. Note that adding this disrupts particle identity, so it should only be used if we're interested in marginals, such as the filtering distribution $p(x_t|y_{1:t})$ rather than the full posterior path.

The famous Bootstrap particle filter (BPF) is a realization of the generic particle filter with $\pi(x_t|x_{0:t-1}, y_{1:t}) = p(x_t|x_{t-1})$ and a resampling step after every time step. In Algorithm 5, we modify it to only resample if the effective number of particle is low.

4 References and further reading

- Kalman filters:
 - Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4). Berlin, Heidelberg: Springer-Verlag.
Section 13.3 offers a comprehensive derivation of the Kalman filter equations in discrete time.
 - Ghahramani, Z., & Hinton, G. E. (1996). Parameter Estimation for Linear Dynamical Systems. University of Toronto Technical Report. <https://doi.org/10.1080/00207177208932224>.
Uses EM for parameter learning of linear dynamical systems (LDS). This was not covered by this tutorial, but is inevitable when working with real data.
- Particle filters:

Algorithm 3 Particle filter (generic)

```
1: procedure PARTICLEFILTER( $y_{1:T}$ )
2:                                      $\triangleright$  Approximates  $p(x_{0:t}|y_{1:t})$  with weighted samples.
3: Parameters:  $N$ 
4:   for  $i = 1 \dots N$  do                                      $\triangleright$  Initialization
5:      $x_0^{(i)} \sim \pi(x_0)$ 
6:      $\tilde{w}_0^{(i)} = \frac{p(x_0^{(i)})}{\pi(x_0^{(i)})}$ 
7:   end for
8:   for  $i = 1 \dots N$  do
9:      $w_0^{(i)} = \frac{\tilde{w}_0^{(i)}}{\sum_{j=1}^N \tilde{w}_0^{(j)}}$ 
10:  end for
11:  for  $t = 1 \dots T$  do                                      $\triangleright$  Filtering recursion
12:    for  $i = 1 \dots N$  do
13:       $x_t^{(i)} \sim \pi(x_t | x_{1:t-1}^{(i)}, y_{1:t})$ 
14:       $\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t | x_t^{(i)})p(x_t^{(i)} | x_{t-1}^{(i)})}{\pi(x_t^{(i)} | x_{1:t-1}^{(i)}, y_{1:t-1})}$ 
15:    end for
16:    for  $i = 1 \dots N$  do
17:       $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}}$ 
18:    end for
19:  end for
20:  return  $\{x_{0:t}^{(i)}, w_{0:t}^{(i)}\}_{i=1:N}$ 
21: end procedure
```

Algorithm 4 Particle resampling

```
1: procedure RESAMPLE( $x^{(i)}, w^{(i)}$ )
2:                                      $\triangleright$  Resamples particles according to their weight.
3: Parameters:  $N$ 
4:   for  $i = 1 \dots N$  do
5:     draw  $j$  with probability  $w^{(j)}$ 
6:      $\tilde{x}^{(i)} = x^{(j)}$ 
7:   end for
8:   for  $i = 1 \dots N$  do
9:      $w^{(i)} = 1/N$ 
10:  end for
11:  return  $\{\tilde{x}^{(i)}, w^{(i)}\}_{i=1:N}$ 
12: end procedure
```

Algorithm 5 Bootstrap Particle Filter

```
1: procedure BOOTSTRAPPARTICLEFILTER( $y_{1:T}$ )
2:    $\triangleright$  Approximates  $p(x_t|y_{1:t})$  at every time step with weighted samples.
3:   Parameters:  $N, c$ 
4:   for  $i = 1 \dots N$  do  $\triangleright$  Initialization
5:      $x_0^{(i)} \sim p(x_0)$ 
6:      $w_0^{(i)} = 1/N$ 
7:   end for
8:   for  $t = 1 \dots T$  do  $\triangleright$  Filtering recursion
9:     for  $i = 1 \dots N$  do
10:       $x_t^{(i)} \sim p(x_t|x_{t-1}^{(i)})$ 
11:       $\tilde{w}_t^{(i)} = w_{t-1}^{(i)}p(y_t|x_t^{(i)})$ 
12:    end for
13:    for  $i = 1 \dots N$  do
14:       $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}}$ 
15:    end for
16:    if  $N_{eff} = (\sum_i (w_t^{(i)})^2)^{-1} < c \cdot N$  then  $\triangleright$  Resampling
17:      RESAMPLE( $\{x_t^{(i)}, w_t^{(i)}\}_{i=1:N}$ )
18:    end if
19:  end for
20:  return  $\{x_{0:t}^{(i)}, w_{0:t}^{(i)}\}_{i=1:N}$ 
21: end procedure
```

- Speekenbrink, M. (2016). A tutorial on particle filters. *Journal of Mathematical Psychology*. <https://doi.org/10.1016/j.jmp.2016.05.006>.
Very intuitive tutorial on particle filters, starting from importance sampling and slowly increasing complexity. Has lots of examples.
 - Doucet, A., & Johansen, A. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*. http://automatica.dei.unipd.it/tl_files/utenti/lucaschenato/Courses/PSC10_11/Tutorial_PF_doucet_johansen.pdf.
The go-to tutorial for particle filtering, written by one of the particle-filtering pioneers (Doucet). Sometimes explanations are confusion and not as intuitive.
 - Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings F: Radar and Signal Processing*. <https://doi.org/10.1049/ip-f-2.1993.0015>.
The seminal paper that introduced the Bootstrap Particle Filter. Fun fact: The term "Particle filter" was coined much later.
- Nonlinear filtering in general:
 - Bain, A., & Crisan, D. (2009). *Fundamentals of Stochastic Filtering*. <https://doi.org/10.1007/978-0-387-76896-0>.
One of the standard textbooks on nonlinear filtering, but rather math-y and therefore useless for practitioners, who want to get down to business fast. It does provide pretty deep understanding, though.
 - Kutschireiter, A., Surace, S. C., & Pfister, J. P. (2020). The Hitchhiker's guide to nonlinear filtering. *Journal of Mathematical Psychology*. <https://doi.org/10.1016/j.jmp.2019.102307>.
Tries to explain life, the universe and nonlinear filtering from a change of measure perspective.