553.633/433

Homework #13 (the last one—congratulations!)

Due Wednesday 12/6/17

A.2 (c) (textbook). You can take parts (a) and (b) as a given (no need for re-derivation) since you did those parts in HW 12.

A. Consider the following special case of a nonlinear state-space model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k) + \boldsymbol{w}_k\,,$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k\,,$$

where $\boldsymbol{H}_k$ is a matrix and $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are mutually independent zero-mean Gaussian noise processes with covariance matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ (as usual). As discussed in the reading Arulampalam et al. (2002) and in the online class handout `Kalman_particle_MCMC_handout.pdf`, the optimal importance density for particle filtering under the independent noise assumption is $p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}, \boldsymbol{z}_k)$. For the model, above, it is known that that $p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}, \boldsymbol{z}_k)$ corresponds to a $N(\boldsymbol{a}_k, \boldsymbol{\Sigma}_k)$ distribution with $\boldsymbol{a}_k$ being a mean vector (not relevant here) and $\boldsymbol{\Sigma}_k = \boldsymbol{Q}_{k-1} - \boldsymbol{Q}_{k-1}\boldsymbol{H}_k^T\boldsymbol{S}_k^{-1}\boldsymbol{H}_k\boldsymbol{Q}_{k-1}$ and $\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{Q}_{k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k$. Derive this formula for $\boldsymbol{\Sigma}_k$. (Hint: The matrix inversion lemma [sometimes called the Sherman–Morrison–Woodbury formula] may be useful: $(\boldsymbol{C} + \boldsymbol{AUB})^{-1} = \boldsymbol{C}^{-1} - \boldsymbol{C}^{-1}\boldsymbol{A}(\boldsymbol{BC}^{-1}\boldsymbol{A} + \boldsymbol{U}^{-1})^{-1}\boldsymbol{BC}^{-1}$, given that the indicated inverses are assumed to exist.)

Note: The main reading for particle filtering in the class, Arulampalam et al. (2002), almost gives the result to be proved in expression (60) in the reading, but the paper does not show how the expression is derived (simply stating "one obtains…." before (60)). You are to prove the result using the given model form, first principles, and Bayes formula, as needed. You may need to prove the result (58) in Arulampalam et al. (2002) as part of the derivation ((58) is not to be taken as a "given").

B. Consider the following linear state-space model of a three-state dynamical system with scalar measurements:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 \end{bmatrix} \boldsymbol{x}_k + \boldsymbol{w}_k$$

$$z_k = [1\ 1\ 1]\, \boldsymbol{x}_k + v_k$$

where $\boldsymbol{w}_k$ and v_k are mutually independent zero-mean Gaussian noise processes with covariance matrices $\boldsymbol{Q}_k = \boldsymbol{I}_3$ (3×3 identity matrix) and $R_k = 1$ for all k. Further, assume that the initial state has $\boldsymbol{x}_0 = [0\ 0\ 0]^T$ and $\boldsymbol{P}_0 = \boldsymbol{I}_3$. Generate one simulated set of measurements $\{z_1, z_2, \ldots, z_{200}\}$ and use both the Kalman filter and the particle filter *with resampling* to estimate the state vectors corresponding to the 200 measurements. Using N = 10 or 1000 particles, calculate the relative errors for each three components between the two estimates according to the following:

(a) For each of the three components in the state vector, using N = 10 and N = 1000, plot the true state, Kalman filter estimate and the particle filter estimate over the time range [0,200] (i.e., produce three plots with three paths on each plot for each N, so a total of 6 plots). All studies use the same set of 200 measurements.

(b) Compare and comment briefly on the observed accuracy of the particle filter estimates relative to the Kalman filter estimates across the time range [0,200] for the two values of N. That is, for each of the three components, take the (Euclidean) norm of the difference between the 200-dimensional vectors of particle filter estimates and Kalman filter estimates, and divide by the norm of the 200-dimensional vector of Kalman filter estimates.

Note: For problem B, you may use particle filter code posted at the course website. You should use your formerly developed (HW12) Kalman filter code as well.