# Sample-based filters

ISTA 410 / INFO 510: Bayesian Modeling and Inference

U. of Arizona School of Information

December 9, 2020

## Outline

Last time:

- Extended Kalman filter

Today: sampling approaches

- Ensemble Kalman filter
- Unscented Kalman filter (deterministic sampling)
- Particle filter (importance sampling)

## Propagating mean and covariance through time

Basic idea:

- Kalman filter propagates mean ($\hat{x}_k$) and covariance ($P_k$) of estimate through time steps
- When all distributions are Gaussian, KF works fine
- When not, we can encode the mean and covariance in a sample

The three filters today differ in their approach to creating this sample

## Ensemble Kalman filter

## Ensemble Kalman filter

- Ensemble model: run several of the same model in parallel, and estimate with the mean of
- Ensemble Kalman filter: run a Kalman filter with $N$ estimates $\hat{x}_k^{(i)}$, $1 \leq i \leq N$
    - Sample starting points from $N(x_0, P_0)$ given initial state, estimate covariances
    - When projecting the points forward, perturb each $\hat{x}_k^{(i)}$ by a random variable $e_v^{(i)} \sim N(0, Q_k)$
    - When incorporating measurements, perturb each $z_k$ by a random variable $e_w^{(i)} \sim N(0, R_k)$
- Perturbing at each step prevents the ensemble from converging to the same estimate

## Update equations

Then the time update (predict) equation becomes:

$$\hat{x}_k^{(i)-} = f(x_k^{(i-1)}) + e_v^{(i)}$$

and the state estimate (update) equation becomes:

$$\hat{x}_k^{(i)} = \hat{x}_k^{(i)-} + K_k(z_k + e_w^{(i)} - \hat{z}_k^-)$$

At any time point we can obtain an overall estimate of the system state by computing the sample mean and covariance of the vectors $x_k^{(i)}, 1 \leq i \leq N$.

## Covariance and gain

The principle advantage of the Ensemble KF in handling nonlinearity is that the covariances and gain matrices can be calculated without linearizing the observation function $h$.

## Covariance and gain

The principle advantage of the Ensemble KF in handling nonlinearity is that the covariances and gain matrices can be calculated without linearizing the observation function $h$.

Previously:

$$K_k = P_- H^T (H P_- H^T + R)^{-1}$$

which requires linearizing $h$ to get $H$.

## Covariance and gain

The principle advantage of the Ensemble KF in handling nonlinearity is that the covariances and gain matrices can be calculated without linearizing the observation function $h$.

Previously:

$$K_k = P_- H^T (HP_- H^T + R)^{-1}$$

which requires linearizing $h$ to get $H$.

Now:

$$K_k = P_{xz}^- (P_{zz}^- 1)^{-1}$$

where $P_z z^-, P_{xz}^-$ are two covariance/cross-covariance matrices computed from the sample $\hat{x}_k^{(i)}$ and the predicted/actual measurements.

## Unscented Kalman filter

## Sigma points

The unscented filter is based on a deterministic sampling approach called the *unscented transform*.

The idea is, given the mean and variance of a random variable $x$, to compute the mean and variance of $y = f(x)$.

We do this by projecting forward a specially chosen set of points.

## Simple example

Suppose we have a random variable $x \sim \mathrm{Normal}(48, 0.75^2)$, and a nonlinear transformation $y = f(x) = 0.1x\cos(0.01x^2)$; what are the mean and variance of $y$?

Two approaches:

- Linearize around $x = 40$ and project the density forward (the EKF approach)
- Project a sample of values from $x$ forward and compute mean, variance from that (the UKF approach)

## Picking a sample

Instead of randomly sampling, the unscented filter chooses a set of so-called *sigma points*.

There are many potential choices, but here is one option:

$$
\begin{aligned}
x_0 &= \mu_x \\
x_1 &= \mu_x + \sqrt{(1 + 2\alpha^2)\sigma_x^2} \\
x_1 &= \mu_x - \sqrt{(1 + 2\alpha^2)\sigma_x^2}
\end{aligned}
$$

$\alpha$ is a tuning parameter determining the spread of the samples

## Weighting the samples

Define weights:

$$
\begin{aligned}
\omega_0^{\mu} &= \frac{2\alpha^2}{2\alpha^2+1} \\
\omega_{\pm 1}^{\mu} &= \frac{1}{2(2\alpha^2+1)} \\
\omega_0^{\sigma^2} &= \omega_0^{\mu} + 1 - \alpha^2 + \beta \\
\omega_{\pm 1}^{\sigma^2} &= \frac{1}{2(2\alpha^2+1)}
\end{aligned}
$$

($\beta$ depends on the distribution of $x$ – $\beta = 2$ for a Gaussian)

## Recovering transformed $\mu, \sigma$

Then, the mean and variance of $y$ can be recovered as

$$\hat{\mu}_y = \sum \omega_i^{\mu} f(x_i)$$

$$\sigma_y^2 = \sum \omega_i^{\sigma^2} (f(x_i) - \hat{\mu}_y)^2$$

There are straightforward generalizations for vector-valued $x$, with 2 additional sigma points per dimension.

## Results

- linearization: $\mu_y = -2.393$, $\sigma_y = 2.958$
- unscented transform: $\mu_y = -1.806$, $\sigma_y = 2.203$
- true values: $\mu_y = -1.979$, $\sigma_y = 2.472$

So, in this example, the unscented transform gives a better approximation.

## Using this in a filter

The unscented transform is not in and of itself a filtering algorithm, but it can be used in place of linearization to push means and covariances through the time-update and measurement equations

$$x_{k+1} = f(x_k) + w_k$$
$$z_k = h(x_k) + v_k$$

1. Prediction step: propagate sigma points through $f$ and reconstruct the mean estimate $\hat{x}_k^-$ and covariance $P_-$
2. Update step:
   2.1 Sample new sigma points
   2.2 Compute covariances and gain
   2.3 Update in the usual way

## How this generalizes

Like the Ensemble KF, this does not require linearity, because the covariances and gain are computed from the samples, and we don't have to explicitly linearize the observation function $h$.

However, the choice of sigma points as a representative sample of the distribution implies Gaussian (or at least unimodal) distributions.

More general: particle filtering

## Particle filtering

In particle filtering, we maintain $N$ state trajectories, similar to the Ensemble KF, but use a variation of importance sampling to project forward in the next time slice.

This frees us from the assumption of Gaussian distributions, because importance sampling works on much broader classes of distributions, subject to a good choice of proposal.

The simplest form, *bootstrap filtering*, does not require us to explicitly choose a proposal distribution, but uses importance-like weights to resample from our set of state trajectories at each time step.

## Bootstrap filter

In this approach, we presume we have a probabilistic model for the dynamical transitions – that is, we know $p(x_{k+1}|x_k)$. (This includes our Gaussian model, where $p(x_k + 1|x_k)$ is a normal distribution centered at $f(x_k)$, but can include broader classes of models too.)

We also have a data likelihood model $p(z_k|x_k)$.

The term *bootstrap* refers to the fact that at each time step, we choose a sample of the trajectories to continue, but with replacement – so some trajectories may branch while others are dropped.

## Bootstrap filter

At each time step we have $N$ trajectory states (the particles) $\hat{x}_k^{(i)}$, with associated weights $w_i$.

To propagate forward:

- For $j \in 1, \ldots, N$, draw a particle according to its weight, $\tilde{x}_k^{(j})$;
- For each $\tilde{x}_k^j$, propagate forward by sampling $\hat{x}_{k+1}$ from $p(x_{k+1}|\tilde{x}_k^j)$
- Re-compute weights for the $\hat{x}_{k+1}$ proportional to the data likelihood $p(z_{k+1}|\hat{x}_{k+1})$.

## Summary

Several approaches to nonlinear filtering that are more robust than extended KF:

- Ensemble KF
- Unscented KF
- Bootstrap filter

For more details following the same development: Hwang & Brown, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises* (chapter 7.1, available online through UA library); Koller & Friedman, *Probabilistic Graphical Models: Principles and Techniques* (chapter 15.3, library has one hard copy)

## Final logistics

Final: take-home, similar to a homework but broader, due end of day Thursday 12/17; available by midnight tomorrow

## Final logistics

Final: take-home, similar to a homework but broader, due end of day Thursday 12/17; available by midnight tomorrow

Thank you for a great semester!