Non-Gaussian and Non-Parametric (Particle) Filters

Brian Hunt

13 Jun 2013

Statistical Formulation

- Given observed time series: $y_1, y_2, \ldots$
- Want to estimate state vectors $x_1, x_2, \ldots$
- Stat. model: known pdfs $p(y_n|x_n)$, $p(x_n|x_{n-1})$
  - Often assume Gaussian noise:
  - Measurement: $p(y_n|x_n) \sim N(h(x_n), R)$
  - Dynamical: $p(x_n|x_{n-1}) \sim N(m(x_{n-1}), Q)$
  - If $Q = 0$, then $x_n = m(x_{n-1})$
- Problem: describe $p(x_n|y_1, y_2, \ldots, y_n)$, perhaps in terms of a "prior" pdf $p(x_0)$.

Bayesian Data Assimilation

- Data assimilation solves the problem iteratively (and in practice, approximately).
- Exact solution: given $p(x_{n-1}|y_1, \ldots, y_{n-1})$,
  - Forecast step: $p(x_n|y_1, \ldots, y_{n-1}) = \int p(x_n|x_{n-1})p(x_{n-1}|y_1, \ldots, y_{n-1})dx_{n-1}$
  - Analysis step: $p(x_n|y_1, \ldots, y_n) \sim p(y_n|x_n)p(x_n|y_1, \ldots, y_{n-1})$ [Bayes' rule]
- Drawback: forecast step is intractable.

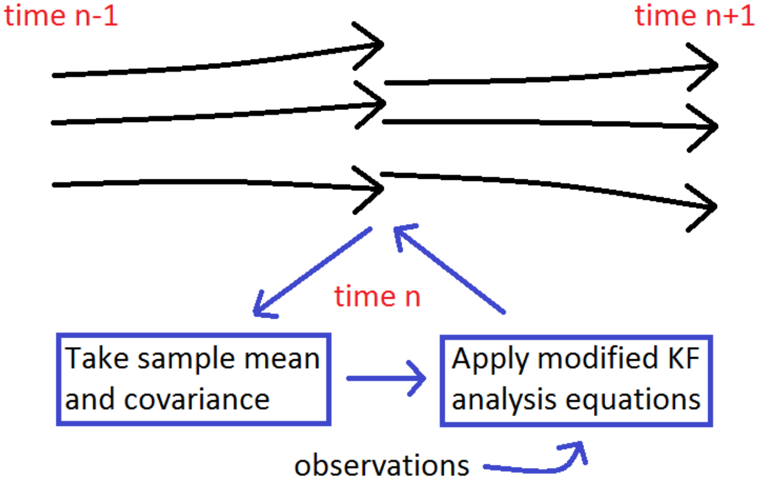
Kalman Filter

- If $p(y_n|x_n)$ and $p(x_n|x_{n-1})$ are Gaussian and linear
$$\begin{aligned} p(y_n|x_n) &\sim N(Hx_n, R) \\ p(x_n|x_{n-1}) &\sim N(Mx_{n-1}, Q) \end{aligned}$$
then all distributions on previous slide are Gaussian.
- For Bayesian data assimilation, need only keep track of mean and covariance.
- Kalman Filter expresses forecast and analysis steps as linear algebra equations.

Ensemble Kalman Filters

- Forecast an ensemble of state vectors according to dynamical model.
- Associate a Gaussian distribution with the ensemble via sample mean and covariance.
- Use Kalman Filter analysis equations to transform sample mean and covariance.
- Choose an ensemble consistent with the output mean and covariance to initialize the next forecast (variety of approaches).

EnKF: Schematic



Non-Gaussian Ensemble Filtering

- Replace boxes on previous slide!
- Nonparametric approach: particle filters.
- Parametric approach:
  - Infer parameters of assumed form for $p(x_n|y_1,\ldots,y_{n-1})$ from forecast ensemble.
  - Given y_n, perform MLE (or suitable approximation) to determine parameters for $p(x_n|y_1,\ldots,y_n)$.
  - Choose ensemble consistent with those parameters to initialize next forecast.

Example 1: MLEF

- Maximum Likelihood Ensemble Filter [Zupanski 2005]: Use Gaussian for $p(x_n|\cdots)$ but allow $p(y_n|x_n)$ to be non-Gaussian.
- An MLE then requires minimizing a nonquadratic function, which must be done numerically.
- The Gaussian distribution is based on a low-rank sample covariance; need only minimize in the space spanned by the ensemble (not the model state space).

Example 2: Non-Gaussian LETKF

- Joint work with John Harlim [2007].
- Uses framework of LETKF [Hunt, Kostelich, Szunyogh 2007; Ott et al. 2004].
- Idea: associate to the ensemble a non-Gaussian distribution whose support is still in the ensemble space.
- For $p(x_n| \cdots)$, we used a distribution with exponential tail (decays slower than Gaussian) – similar to “Huber” cost function.
- Tested with simulated observations generated by adding Gaussian noise to a model trajectory.

Numerical Experiments by Harlim

- Performed experiments with Lorenz models ('63 and '96) and SPEEDY model [Molteni 2003], a simplified (atmospheric) GCM.
- Non-Gaussian filter yielded noticeable improvement for Lorenz models when observation noise and time between observations was sufficiently large.
- For SPEEDY with realistic observation noise, improvement was mainly during initialization. Computation time was only about 30% slower than Gaussian filter.

Summary

- Gaussian data assimilation minimizes a quadratic function algebraically.
- Non-Gaussian data assimilation requires numerical minimization and thus is computationally more expensive.
- NG Ensemble DA still inexpensive if ensemble size is not too large (EnKF successful with 40 ensemble members in multi-million-dimensional model space).
- Can be worth the effort if the application is sufficiently nonlinear and/or non-Gaussian.

Particle Filters

- Main idea: Do approximate Bayesian data assimilation with discrete probability distributions supported on a finite number of model states ("particles").
- Keep track of a reasonably large ensemble of particles $x_1, \ldots, x_k$ and corresponding (scalar) weights $w_1, \ldots, w_k$ whose sum is 1.
- The associated pdf (using Dirac δ) is

$$p(x) = \sum_{i=1}^{k} w_i \delta(x - x_i)$$

Particle Forecast and Adjustment

- Make a (stochastic) model forecast for each particle x_i.
- Applying Bayes' rule changes only the weights $\{w_i\}$, not the particles.
- Advantage: no averaging of model states!
- Disadvantage: If the model is chaotic, then none of the particles will stay close to the truth. Also, the weights tend to concentrate on one particle.

“Importance” Resampling

- After adjusting weights, resample (many stragegies available) the particles so that particles with low weight are eliminated and particles with high weight are replicated.
- If the forecasts are deterministic, this won’t help – eventually all particles will be the same and be decorrelated from the truth.
- With a stochastic forecast, replicated (high-probability) particles spread out and hopefully sample the vicinity of the truth well enough to maintain a high number of particles near the truth.

Curse of Dimensionality

- Ideally, in the limit that the number of particles goes to infinity, a particle filter converges toward exact Bayesian data assimilation with continuous prior.
- For low-dimensional systems, it is plausible to use enough particles to reasonably sample the “correct” prior and posterior distributions.
- It is implausible to use enough particles to reasonably sample a high-dimensional probability distribution; if there are many degrees of freedom, may need a hybrid.