The Local Ensemble Transform Kalman Filter

Eric J. Kostelich

Arizona State University
School of Mathematics and Statistics

MSRI Climate Change Summer School
July 21, 2008

Co-workers:

Istvan Szunyogh, Brian Hunt, Edward Ott,

Eugenia Kalnay, Jim Yorke

and many others!

Thanks to: Dave Kuhl

Papers, preprints, and codes:

http://www.weatherchaos.umd.edu
http://math.asu.edu/~eric

Principal papers

Preprints: www.weatherchaos.umd.edu

Initial papers:

- E. Ott *et al.*, *Tellus A* **56** (2004), 415–428.
- I. Szunyogh *et al.*, *Tellus A* **57** (2005), 528–545.

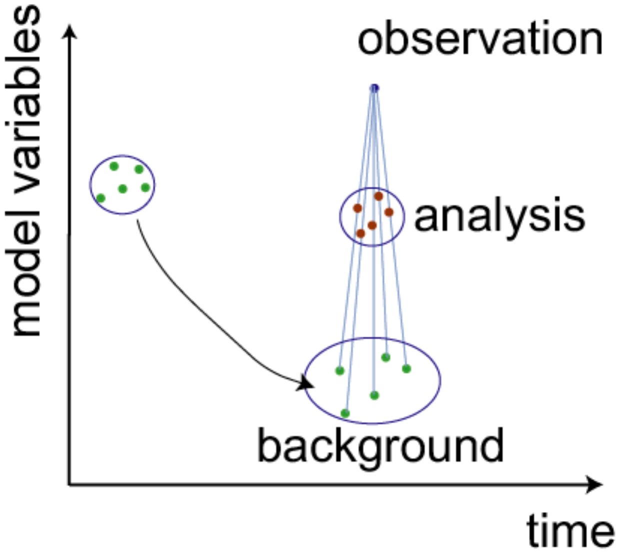
Refined mathematical implementation: B. R. Hunt, E. K., I. Szunyogh, *Physica D* **230** (2007) 112–126.

Results with real data: I. Szunyogh, E.K. *et al.*, *Tellus A* **60** (2008) 113–130.

The estimation problem

- Treat both the observations and the system state as Gaussian random variables
- Find the system trajectory $\mathbf{x}(t)$ that is “most likely” to produce the given set of observations
- The Kalman filter provides an iterative method to find $\mathbf{x}$
- The Kalman estimate of $\mathbf{x}$ has Gaussian error when the underlying model is linear
- It is also unbiased and has minimum variance

Cartoon of the algorithm



Maximum likelihood estimation

- Observations $\mathbf{y}_i = H_i(\mathbf{x}(t_i)) + \varepsilon_i$ at times $t_1 < t_2 < \cdots < t_N$, where $\varepsilon_i \sim N(\mathbf{0}, \mathbf{R}_i)$
- The likelihood of the trajectory $\mathbf{x}(t)$ is proportional to

$$\prod_{i=1}^{N} \exp\left(-\frac{1}{2}[\mathbf{y}_j - H_j(\mathbf{x}(t_n))]^{\mathrm{T}}\mathbf{R}_j^{-1}[\mathbf{y}_j - H_j(\mathbf{x}(t_n))]\right)$$

Maximum likelihood estimation

- The "most likely" trajectory minimizes

$$J(\mathbf{x}(t)) = \sum_{i=1}^{n} [\mathbf{y}_i - H_i(\mathbf{x}(t_i))]^{\mathrm{T}} \mathbf{R}_i^{-1} [\mathbf{y}_i - H_i(\mathbf{x}(t_i))]$$

- Let M_{t,t_i} be the map that propagates a solution of the model from time t to time t_i.
- The minimization problem can be re-expressed as a function of the system state at time t:

$$J(\mathbf{x}) = \sum_{i=1}^{n} [\mathbf{y}_i - H_i(M_{t,t_i}(\mathbf{x}))]^{\mathrm{T}} \mathbf{R}_i^{-1} [\mathbf{y}_i - H_i(M_{t,t_i}(\mathbf{x}))]$$

The Kalman Filter

- If M and H are linear, then J has a unique minimizer
- No such guarantee for a nonlinear map (but weather models are assumed to be "linear enough" over short time intervals to make this useful)
- Minimizing J from scratch at each new time t_i is expensive
- The Kalman filter computes the minimizer by an iterative method

Linear scenario

- Suppose a linear model: $M_{t,t_i}(\mathbf{x}) = \mathbf{M}_{t,t_i}\mathbf{x}$
- And linear observation operator: $H_i(\mathbf{x}) = \mathbf{H}_i\mathbf{x}$
- Suppose we have the most likely system state given the observations up to time t_{n-1}
- Take a forecast from t_{n-1} to t_n
- Combine with observations at time t_n to obtain the most likely state given the observations up to time t_n

Linear scenario: Kalman filter

- $\mathbf{x}_a(t_{n-1})$ is the most likely state at time t_{n-1}
- The associated uncertainty is $\mathbf{P}_a(t_{n-1})$
- The relative likelihood of system states up to t_{n-1} is $N(\mathbf{x}_a, \mathbf{P}_a)$
- Note: matrix $\times$ Gaussian $=$ Gaussian

Linear scenario: Kalman filter

- Algebraically, we assume that

$$\sum_{i=1}^{n-1} [\mathbf{y}_i - H_i(M_{t_{n-1},t_i}(\mathbf{x}))]^{\mathrm{T}} \mathbf{R}_i^{-1} [\mathbf{y}_i - H_i(M_{t_{n-1},t_i}(\mathbf{x}))]$$

$$= [\mathbf{x} - \mathbf{x}_a(t_{n-1})]^{\mathrm{T}} \mathbf{P}_a(t_{n-1})^{-1} [\mathbf{x} - \mathbf{x}_a(t_{n-1})] + C$$

- The analysis $\mathbf{x}_a(t_{n-1})$ "completes the square" so that the part of the quadratic cost function J that depends on the observations up to time t_{n-1} is a single quadratic form (plus a constant)

Linear scenario: Kalman filter

- The Kalman filter determines $\mathbf{x}_a(t_n)$ and $\mathbf{P}_a(t_n)$ so that an analogous relation holds at time t_n
- Forecast (background): $\mathbf{x}_b(t_n) = \mathbf{M}_{t_{n-1},t_n}\mathbf{x}_a(t_{n-1})$
- Forecast covariance: $\mathbf{P}_b(t_n) = \mathbf{M}_{t_{n-1},t_n}\mathbf{P}_a(t_{n-1})\mathbf{M}^{\mathrm{T}}_{t_{n-1},t_n}$
- At $t = t_n$,

$$J_{t_n}(\mathbf{x}) = [\mathbf{x} - \mathbf{x}_b(t_n)]^{\mathrm{T}}\mathbf{P}_b(t_n)^{-1}[\mathbf{x} - \mathbf{x}_b(t_n)]$$

$$+ [\mathbf{y}_n - \mathbf{H}_n\mathbf{x}]^{\mathrm{T}}\mathbf{R}_n^{-1}[\mathbf{y}_n - \mathbf{H}_n\mathbf{x}] + C$$

Kalman filter estimates

- The Kalman filter gives the analysis $\mathbf{x}_a(t_n)$ and associated covariance $\mathbf{P}_a(t_n)$ so that

$$J(\mathbf{x}) = [\mathbf{x} - \mathbf{x}_a(t_n)]^{\mathrm{T}} \mathbf{P}_a(t_n)^{-1} [\mathbf{x} - \mathbf{x}_a(t_n)] + C'$$

- Analysis: $\mathbf{x}_a(t_n) = \mathbf{P}_a(t_n) \left[\mathbf{P}_b(t_n)^{-1} \mathbf{x}_b(t_n) + \mathbf{H}_n^{\mathrm{T}} \mathbf{R}_n^{-1} \mathbf{y}_n\right]$
- Covariance: $\mathbf{P}_a(t_n) = \left[\mathbf{P}_b(t_n)^{-1} + \mathbf{H}_n^{\mathrm{T}} \mathbf{R}_n^{-1} \mathbf{H}_n\right]^{-1}$

Equivalent formulation

- In terms of the innovation:

$$\underbrace{\mathbf{x}_a(t_n)}_{\text{analysis}} = \underbrace{\mathbf{x}_b(t_n)}_{\text{background}} + \underbrace{\mathbf{P}_a(t_n)\mathbf{H}_n^{\mathrm{T}}\mathbf{R}_n^{-1}}_{\text{Kalman gain matrix}} \underbrace{(\mathbf{y}_n - \mathbf{H}_n\mathbf{x}_b(t_n))}_{\text{innovation}}.$$

- Analysis covariance:

$$\mathbf{P}_a(t_n) = \left[\mathbf{I} + \mathbf{P}_b(t_n)\mathbf{H}_n^{\mathrm{T}}\mathbf{R}_n^{-1}\mathbf{H}_n\right]^{-1}\mathbf{P}_b(t_n)$$

- Even noisy observations are useful:

$$\mathbf{P}_a(t_n)^{-1} = \mathbf{P}_b(t_n)^{-1} + \mathbf{H}_n^{\mathrm{T}}\mathbf{R}_n^{-1}\mathbf{H}_n$$

Practical difficulties

- For a weather forecast model, $\mathbf{P}_b(t_n)$ is too complicated to be computed explicitly
- $\mathbf{P}_b(t_n)$ is too huge to be inverted explicitly
- But it's essential for accuracy!
- The LETKF addresses these problems in 3 ways

LETKF approach

- Localization: Look at local regions ($\sim$Texas) instead of the whole globe
- Ensemble forecast replaces

$$\mathbf{P}_b(t_n) = \mathbf{M}_{t_{n-1},t_n}\mathbf{P}_a(t_{n-1})\mathbf{M}^{\mathrm{T}}_{t_{n-1},t_n}$$

with the estimate

$$\widehat{\mathbf{P}}_b = [\mathbf{X}_b - \bar{\mathbf{x}}_b]\,[\mathbf{X}_b - \bar{\mathbf{x}}_b]^{\mathrm{T}}$$

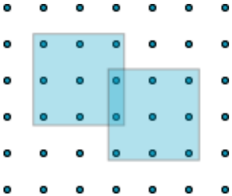
where each column of $\mathbf{X}_b$ is a forecast

- Transform the forecast ensemble $\mathbf{X}_b$ into an analysis ensemble $\mathbf{X}_a$ by accounting for the observations

Use dynamics to reduce the dimensionality

- Key idea: Over typical synoptic regions, the forecast uncertainty evolves in a much lower-dimensional space than the phase space
- This appears to be a general property of many spatiotemporal models (weather, climate, ocean)
- A typical ensemble of $100 - 200$ forecast vectors over a typical synoptic (“Texas-sized”) region spans a ~ 40 dimensional “unstable manifold”
- Do the Kalman filtering in this lower-dimensional subspace to reduce the forecast uncertainty

The LETKF algorithm



- For each local region, apply the ensemble Kalman filter and save the analysis at the center grid point
- There is one local region about each model grid point
- Local regions are updated independently
- The set of observations used for each local analysis varies slowly in space (important for continuity)
- Naturally parallel algorithm

Computational outline

Within each local region:

- Find the mean of the local k-member ensemble and the k differences (“pertubations”) therefrom
- Change coordinates to the resulting subspace, which presumably spans the forecast uncertainty locally
- Find the linear combination of forecast perturbations that minimizes the quadratic cost function in the local region
- The local covariance matrix $\mathbf{P}_b$ is $k \times k$
- Yields one $k \times k$ matrix inversion problem for every model grid point

Details of the local analyses

- Each local region is a cylinder centered about a given model grid point containing ℓ model variables
- Each local region is processed independently
- The analysis at each center grid point is assembled into a global analysis
- Given the set of forecast vectors $\{\mathbf{x}_b^i\}_{i=1}^k$, form their mean $\bar{\mathbf{x}}_b$ and the $\ell \times k$ background perturbation matrix $\mathbf{X}_b$ whose ith column is $\mathbf{x}_b^i - \bar{\mathbf{x}}_b$

Details of the local analyses 2

- $\mathbf{P}_b = (k-1)^{-1}\mathbf{X}_b\mathbf{X}_b^T$ is the $k \times k$ local ensemble covariance matrix (of rank $k-1$)
- If $\mathbf{w}$ is Gaussian with mean $\mathbf{0}$ and covariance $(k-1)^{-1}\mathbf{I}$, then $\mathbf{x} = \bar{\mathbf{x}}_b + \mathbf{X}_b\mathbf{w}$ is Gaussian with mean $\bar{\mathbf{x}}_b$ and covariance $\mathbf{P}_b$

Treatment of observations

- Compute the observation operator for each ensemble solution to form $\{\mathbf{H}(\mathbf{x}_b^i)\}$ and its mean $\bar{\mathbf{y}}_b$
- Suppose the local region contains s observations
- Linearize $\mathbf{H}$ about the ensemble mean as $\mathbf{H}(\bar{\mathbf{x}}_b + \mathbf{X}_b\mathbf{w}) \approx \bar{\mathbf{y}}_b + \mathbf{Y}_b\mathbf{w}$ where $\mathbf{Y}_b$ is the $s \times k$ matrix whose ith column is $\mathbf{H}(\mathbf{x}_b^i) - \bar{\mathbf{y}}_b$
- Only the s components of $\mathbf{H}(\mathbf{x}_b^i)$ belonging to the given local region are used in the computation of $\bar{\mathbf{y}}_b$ and $\overline{\mathbf{Y}}_b$

The minimization procedure

- Goal: find the linear combination of background forecast perturbations that best fits the observations
- Minimize the cost function $\widehat{J}(\mathbf{w}) = (k-1)^{-1}\mathbf{w}^{\mathrm{T}}\mathbf{w} + [\mathbf{y} - \overline{\mathbf{y}}_b - \mathbf{Y}_b\mathbf{w}]^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{y} - \overline{\mathbf{y}}_b - \mathbf{Y}_b\mathbf{w}]$
- The minimizer $\mathbf{w}_a$ of $\widehat{J}$ is perpendicular to the null space of $\mathbf{X}_b$, and $\overline{\mathbf{x}}_a = \overline{\mathbf{x}}_b + \mathbf{X}_b\mathbf{w}_a$ minimizes the original J

Net result

- The minimizer is the k-vector $\mathbf{w}_a = \mathbf{Q}\mathbf{Y}_b^T\mathbf{R}^{-1}(\mathbf{y} - \bar{\mathbf{y}}_b)$ where $\mathbf{Q} = [(k-1)\mathbf{I} + \mathbf{Y}_b^T\mathbf{R}^{-1}\mathbf{Y}_b]^{-1}$ is $k \times k$
- In model space, the analysis mean becomes $\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b + \mathbf{X}_b\mathbf{w}_a$
- The analysis perturbations are given by $\mathbf{X}_a = \mathbf{X}_b\mathbf{W}_a$ where $\mathbf{W}_a = [(k-1)\mathbf{Q}]^{1/2}$ and we take the symmetric square root
- This choice assures that $\mathbf{W}_a$ depends continuously on $\mathbf{Q}$ and that the columns of $\mathbf{X}_a$ sum to zero (for the correct sample mean)

Computational complexity

- In principle, the only free parameters are the ensemble size k and the size of the local regions
- Given a local region with s observations and an ensemble of size k
- The most expensive step ($> 90\%$ of cpu cycles): computing $\mathbf{Y}_b^T\mathbf{R}^{-1}\mathbf{Y}_b$, which is $O(k^2 s)$
- Second most expensive step: computing the symmetric square root, which is $O(k^3)$
- Observation lookup is $O(\log L)$, where L is the size of the total observation set

Efficient parallel implementation

- All data needed to analyze a given model grid point is distributed only once
- Efficient load balancing algorithm provides nearly linear scaling up to 128 processors (at least)
- Assimilating $\sim 250{,}000$ observations into the T62 GFS ($\sim 3{,}000{,}000$ variables) takes 180 seconds on 16 single-core Xeon processors
- Greatest expense is computing the forecasts (also embarrassingly parallel)

Key properties of the LETKF

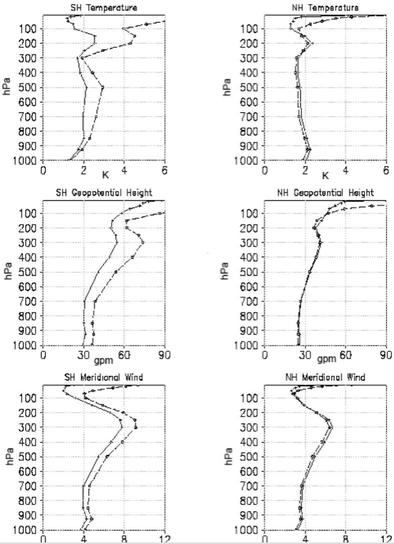
- Estimates of the most likely current state and also its uncertainty
- Assimilates all data at once
- Observations can be nonlinear functions of the state vector
- Can interpolate in time as well as space
- The only free parameters are the ensemble size and size of local regions
- Model independent (no adjoints!)

Validation experiments

- Operational (T254, ~ 50 km) analyses with full observing network are taken as "truth"
- LETKF used with full observing network (minus satellite radiances but including satellite-derived wind estimates) and 60-member ensemble
- Analysis/forecast cycle run from Jan. 1 to March 1, 2004 using operational GFS at T62 resolution
- Benchmark analysis: NCEP operational system at T62 (~ 200 km) with reduced dataset
- LETKF analysis: LETKF using reduced dataset at T62
- Thanks to Zoltan Toth and NCEP

Sample results: 48-hour forecast error

- NCEP operational analyses taken as truth
- Compute $\langle \text{forecast} - \text{truth} \rangle$ using forecasts started from LETKF and NCEP T62 analyses from Jan. 11 to Feb. 27, 2004
- NCEP surface analyses form the boundary conditions



Application to the ECOM model

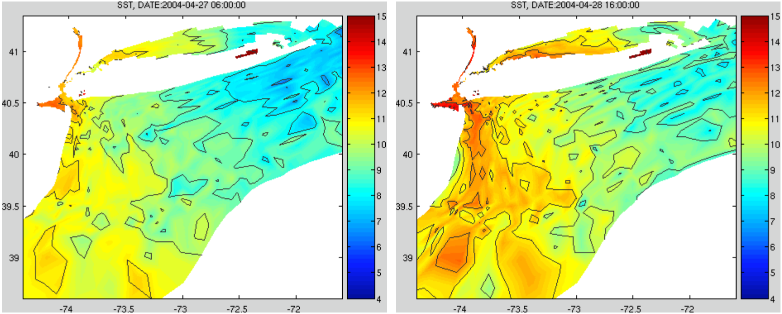
- Estuarine and Coastal Ocean Model
- Model author: Alan Blumberg, Princeton
- Co-workers: Rui Ponte and Ross Hoffman, Atmospheric & Environmental Research, Inc., Lexington, MA
- Istvan Szunyogh, U. Maryland

The ECOM model

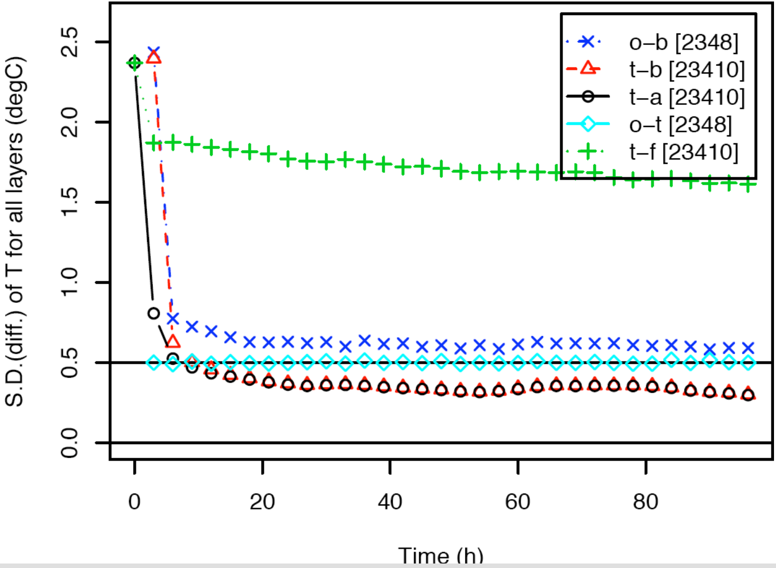
- Offshoot of the Princeton Ocean Model
- Computes water circulation, temperature, salinity, mixing, and transport in rivers, lakes, bays, estuaries, and the coastal open ocean
- LETKF used to assimilate simulated observations of surface height, temperature, horizontal velocity components, and salinity
- Each has natural analogs in the GFS
- GFS/LETKF code used with minimal changes

Dynamic, challenging test case

Large change in warm freshwater plume over 34 hours



Evolution of the temperature errur



Ongoing work

- Estimation of model and observation bias
- High-resolution GFS forecasts
- Other ocean circulation models
- Carbon data assimilation in climate models (with I. Fung and co-workers)
- Planetary atmospheres (Mars Microwave Sounder)

Not all problems are solved!

- What if the model has systematic biases?
- Not all observations give sufficient information about the dynamics (the observability problem)
- If you could take *more* observations, where and when should you do so? (the targeted observation problem)
- Observation error has two components:
  - Instrumentation error
  - Representativeness error (from subgrid-scale dynamics)
- Nonlinearities cause $\widehat{\mathbf{P}}_b$ to underestimate the true uncertainty

Conclusions

- The LETKF merits serious consideration for research and operational applications
- LETKF's advantages are greatest where observations are sparsest
- Ideal for planetary atmosphere DA, if the model errors are sufficiently small
- Model independence makes it adaptable
- One of the most mature, flexible, computationally efficient, and well-tested ensemble-based Kalman filters

Our sponsors

Thanks to:

- National Science Foundation
- Keck Foundation
- McDonnell Foundation
- NASA
- Army Research Office
- National Centers for Environmental Research
- ASU College of Liberal Arts and Sciences