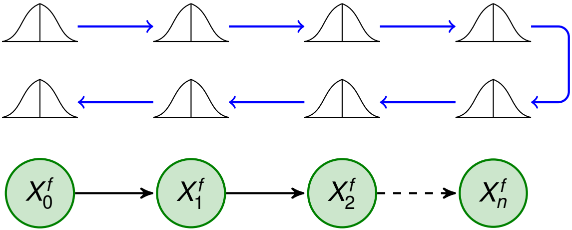
Quantifying Uncertainty

Sai Ravela

M. I. T

2012

Uncertainty Propagates in Time-Depdendent Processes

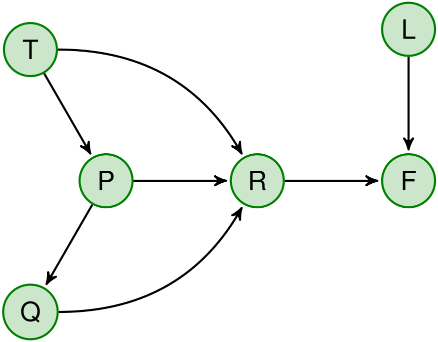


$$M \equiv M(\underline{x}_n; \alpha_n)$$

$$\underline{x}_{n+1} = M(\underline{x}_n; \alpha_n) + \underline{\omega}_n$$

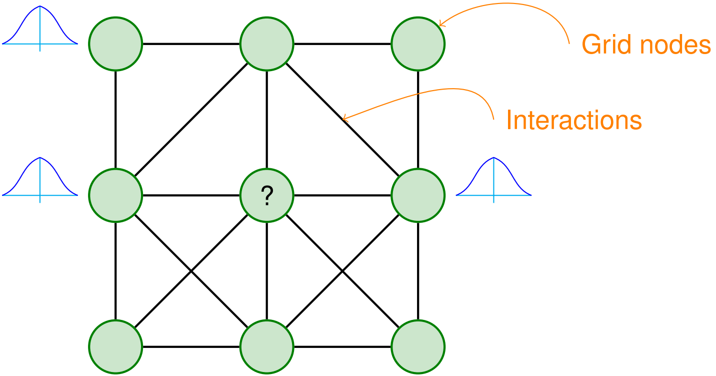
M: -Physical or Statistical Model

Uncertainty Propagates in Bayesian Networks



Found in Hierarchical Bayes, Graphical Models.

Uncertainty Propagates in Spatial Processes



Inference Problems

1. Two-point boundary value problems, incl. uncertainty estimation propagation. Fixed Point Smoother.
2. Recursive Bayesian Estimation for Sequential Filtering and Smoothing.
3. Nonlinearity and Dimensionality and Uncertainty: Ensemble Filter & Smoother.

Inference Problems

1. Two-point boundary value problems, incl. uncertainty estimation propagation. Fixed Point Smoother.
2. Recursive Bayesian Estimation for Sequential Filtering and Smoothing.
3. Nonlinearity and Dimensionality and Uncertainty: Ensemble Filter & Smoother.

Propagating Uncertainty, a first step.

Variational Inference

$$J(\underline{x}_0) := \frac{1}{2}(\underline{x}_0 - \underline{x}_b)^T C_{00}^{-1}(\underline{x}_0 - \underline{x}_b) +$$
$$\sum_{i=1}^{m} \left\{ \frac{1}{2}(\underline{y}_i - H\underline{x}_i)^T R^{-1}(\underline{y}_i - H\underline{x}_i) + \underline{\lambda}_i^T [\underline{x}_i - M(x_{i-1}; \underline{\alpha})] \right\}$$

Cannot deal with stochastic model (i.e. model error). Needs a Bayesian formalism.

Filters and Smoothers

Sequential Filtering:

$$
\begin{aligned}
P(\underline{x}_n|\underline{y}_1 \ldots \underline{y}_n) &\propto P(\underline{y}_n|\underline{x}_n) \underset{x_{n-1}}{\quad\quad} P(\underline{x}_n|\underline{x}_{n-1}) P(\underline{x}_{n-1}|\underline{y}_1 \ldots \underline{y}_{n-1}) \quad (1)\\
&= P(\underline{y}_n|\underline{x}_n) P(\underline{x}_n|\underline{y}_1 \ldots \underline{y}_{n-1}) \quad (2)\\
&= P(\underline{y}_n|\underline{x}_n) P(\underline{x}_n^f) \quad (3)
\end{aligned}
$$

The recursive form is simple when a perfect model is assumed, but

the Kolmogorov-Chapman equation has to be used in the presence of model error. $P(\underline{x}_n|\underline{y}_1 \ldots \underline{y}_{n-1})$ is the forecast distribution or prior distribution also seen as $P(\underline{x}_n^f)$

Write the Objective

Sequential Filtering:

$$\begin{aligned} J(\underline{x}_n) &:= \frac{1}{2}(\underline{x}_n - \underline{x}_n^f)^T P_f^{-1}(\underline{x}_n - \underline{x}_n^f) + \\ & \frac{1}{2}(\underline{y}_n - H\underline{x}_n)^T R^{-1}(\underline{y}_n - H\underline{x}_n) \end{aligned} \tag{4}$$

We have assumed a linear observation operator $\underline{y}_n = H\underline{x}_n + \underline{\eta}$, with $\eta \sim N(0, R)$.

Find the Stationary Point

Sequential Filtering:

$$\begin{aligned}
\hat{\underline{x}}_n &= \underline{x}_n^f + P_f H^T (H P_f H^T + R)^{-1} (\underline{y}_n - H \underline{x}_n^f) && (5)\\
&= \underline{x}^a && (6)\\
P_a &= (H^T R^{-1} H + P_f^{-1})^{-1} && (7)\\
&= P_f - P_f H^T (H P_f H^T + R)^{-1} H P_f && (8)
\end{aligned}$$

Then, launch a new prediction $\underline{x}_{n+1}^f = M(\underline{x}_n)$ and the new uncertainty (predicted) is $P_f = L P_a L^T$, where $L = \frac{\partial M}{\partial \underline{x}_n}$ when the model is nonlinear. Propagating produces the moments of $P(\underline{x}_{n+1} | \underline{y}_1 \ldots \underline{y}_n)$.

Smoother

We are interested in the state estimates at all points in an interval, that is:

$$P(\underline{x}_1 \ldots \underline{x}_n | \underline{y}_1 \ldots \underline{y}_n) \tag{9}$$

The joint distribution can account for model errors, state and parameter errors within its framework.
We break it down via Bayes Rule, Conditional Independence and Markov assumption, and marginalization and perfect modelassumption, leading to a coupled set of equations that are recursively solved.

Uncertainty Propagation is Expensive

Forward(you'll need this in the end)

$$C_{ii} = \frac{\partial M}{\partial x_{i-1}} C_{i-1\,i-1} \frac{\partial M^T}{\partial x_{i-1}} \; 0 < i \leq m$$

Backward via information form:

$$\begin{aligned}
\hat{I}_{mm} &= H^T R^{-1} H \\
\hat{I}_{ii} &= \frac{\partial M^T}{\partial x_i} I_{i+1\,i+1} \frac{\partial M}{\partial x_i} + H^T R^{-1} H \\
\hat{C}_{00} &= \left[C_{00}^{-1} + \frac{\partial M^T}{\partial x_0} \hat{I}_{11} \frac{\partial M}{\partial x_0} \right]^{-1}
\end{aligned}$$

The Dimensionality and Nonlinearity Challenges

Monte-Carlo

- Reduced-rank approximation
- Particle Filter

Domain Decomposition

- Localization, Localized Filters
- Scale-recursive Spatial Inference

Model Reduction & Interpolation

- Snapshots & POD
- Krylov Subspace

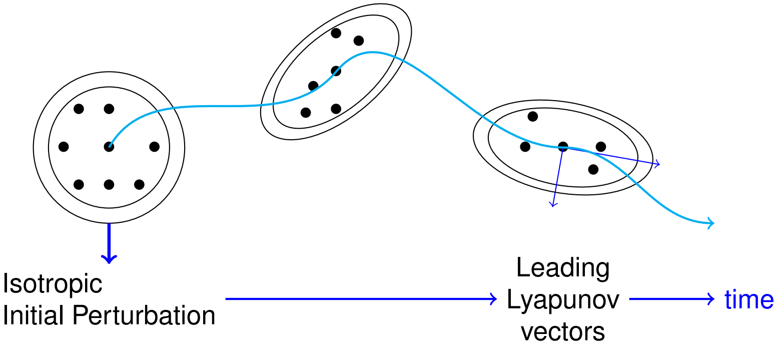
Response Surface Models

- Deterministic Equivalent Modeling Method
- Stochastic Response Surface Methodology

Polynomial Chaos Expansions

- Generalized Polynomial Chaos

Monte-Carlo


Isotropic
Initial Perturbation
Leading
Lyapunov
vectors
time

Filter-Updating

$$\begin{aligned} A^f &= [\underline{x}_1^f \ldots \underline{x}_s^f] \Rightarrow \text{All at time T} \\ \tilde{A}^f &= [\tilde{\underline{x}}_1^f \ldots \tilde{\underline{x}}_s^f] \end{aligned}$$

NOTE THAT

$$P^f = \frac{1}{s-1} \tilde{A}^f \tilde{A}^{fT} \Leftarrow \text{Uncertainty}$$

So, propagate uncertainty through Samples “Integrated” forward.
Model is not linearized.

No Linearization

$$\underline{y} = h(\underline{x}) + \underline{\eta}, \quad \underline{\eta} \sim N(0, R)$$

$$Z = [\underline{y} + \underline{\eta}_1, \ldots, \underline{y} + \underline{\eta}_s] \leftarrow \text{Perturbed Observations}$$

$$R \approx \frac{1}{s-1} \tilde{Z} \cdot \tilde{Z}^T$$

Also, let

$$\Omega^f = h(A^f) = [h(\underline{x}_1^f) \ldots h(\underline{x}_s^f)]$$

$\tilde{\Omega}^f$ defined similarly

Uncorrelated Noise

Note

$$(\tilde{\Omega}^f + \tilde{Z})(\tilde{\Omega}^{fT} + \tilde{Z}^T) = (\tilde{\Omega}^f \tilde{\Omega}^{fT} + \tilde{Z}\tilde{Z}^T)$$

When observation noise is uncorrelated with state $\equiv$ an assumption

Let

$\underline{x}^a$ be the estimate, analysis, 'posterior' rv.
A^a and $\tilde{A}^a$ similarly, defined.

Easy Formulation

$$A^a = A^f + \tilde{A}^f \tilde{\Omega}^{fT} \left[\tilde{\Omega}^f \tilde{\Omega}^{fT} + \tilde{Z}\tilde{Z}^T \right]^{-1} \left[Z - \Omega^f \right]$$

Identical to KF/EKF in linear/linearized case
$\Rightarrow$ No linearization of the model
$\Rightarrow$ No explicit uncertainty (covariance) propagation

$$\begin{aligned} \left[\tilde{\Omega}^f \tilde{\Omega}^{fT} + \tilde{Z}\tilde{Z}^T \right]^{-1} &= \left([\tilde{\Omega}^f + \tilde{Z}][\tilde{\Omega}^{fT} + \tilde{Z}^T] \right)^{-1} \\ &= (CC^T)^{-1} \end{aligned}$$

Solution

Let

$$C = [U \quad S \quad V^T]$$

$$\begin{aligned}
[CC^T]^{-1} &= US^{-2}U^T \\
&= (US^{-1})(US^{-1})^T \\
&= \sqrt{D}\sqrt{D}^T \\
&= D
\end{aligned}$$

Fast Calculation

$$A^a = A^f + \tilde{A}^f \tilde{\Omega}^{fT} [US^{-2}U^T][Z - \Omega^f]$$

$$(n,s) \quad (n,s) \quad (n,s)(s,n)(n,s)(s,s)(s,n)(n,s) \quad (n,s)$$

Return by right to left, multiply; FAST, low-dimensional

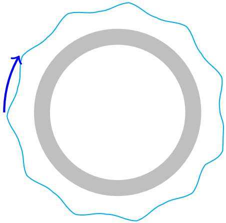
$$\begin{aligned} A^a =& A^f + \tilde{A}^f X_5 \\ =& A^f (I_s + X_4) \\ =& A^f X_5 \end{aligned}$$

A “weakly” nonlinear transformation ($X_5 \equiv X_5(A^f)$)

Time Dependent Example

Lorentz

$$\begin{aligned}\dot{x} &= -x_{i-2}x_{i-1} + x_{i-1}x_{i+1} - x_i + u \\ &= \underbrace{x_{i-1}\left[x_{i+1} - x_{i-2}\right]}_{\textit{Advective}} - \underbrace{x_i}_{\textit{Dissipative}} + \underbrace{u}_{\textit{Forcing}}\end{aligned}$$



Chaotic

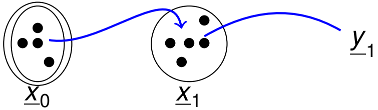
Filter

Need to Get multimedia WORKING

Play ENKFLP.wmv!
Chalk Talk: Method 2.
Demo: Matlab.
Demo: PI Bottle.

Plug and Play

So,

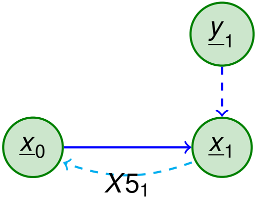


$$
\begin{aligned}
A_0^a &= A_0^f I_s \leftarrow \text{No measurement} \\
A_1^a &= A_1^f X5_1 \leftarrow \text{Filter, same as } X_5 \\
A_1^s &= A_1^a I_s \leftarrow \text{No future measurement} \\
A_0^s &= A_0^a + \tilde{A}_0^a \tilde{\Omega}_1^{fT} [U_1 S_1^{-2} U_1^T][Z_1 - \Omega_1^f] \\
&= A_0^a X5_1
\end{aligned}
$$

Note: $X5$ here is same as X_5 in earlier slide.

Send me a message

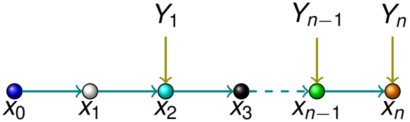
On the graph



Message sent from $\underline{x}_1$ to $\underline{x}_0(X5_1)$
$\underline{x}_0$ smoothed by $\underline{y}_1$ i.e $A_0^s \sim Pr(\underline{x}_0|\underline{y}_1)$

Fixed Interval & Fixed Lag

Fixed Interval

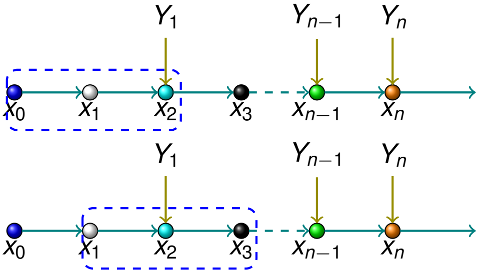


$$\left.\begin{array}{l} P(\underline{x}_0|\underline{y}_1 \cdots \underline{y}_n) \\ P(\underline{x}_1|\underline{y}_1 \cdots Y_n) \\ \vdots \end{array}\right\} \textit{Smoother}$$

$$P(\underline{x}_n|\underline{y}_1 \cdots \underline{y}_n) \leftarrow \textit{Filter}$$

Fixed Interval & Fixed Lag

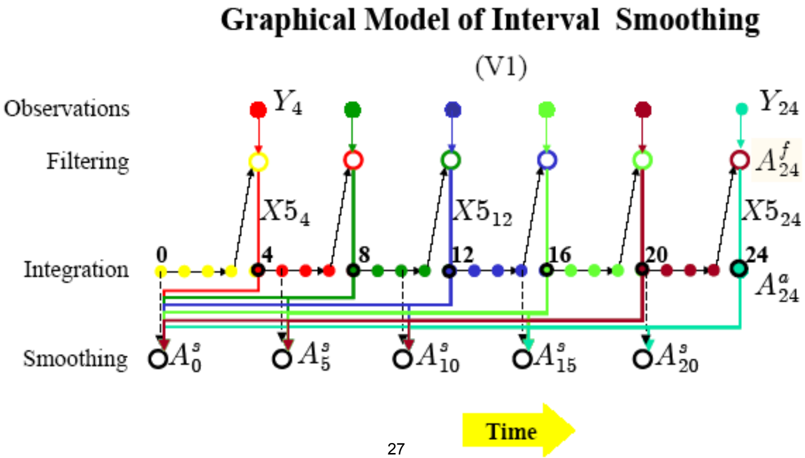
Fixed Lag



$$P(\underline{x}_0|\underline{y}_1 \dots \underline{y}_n) \cong P(\underline{x}_0|\underline{y}_1 \dots \underline{y}_L), \quad (L < n)$$
$$P(\underline{x}_i|Y_0 \dots \underline{y}_{i+L})$$

Smothed up to a "window"

Fixed Interval: The Dumb Way



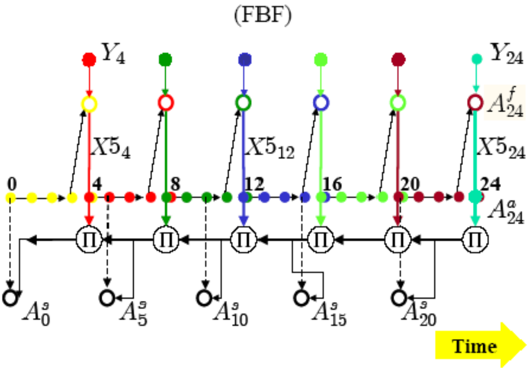
Backward Recursion

Key Assumption: Jointly Gaussian Distributions.

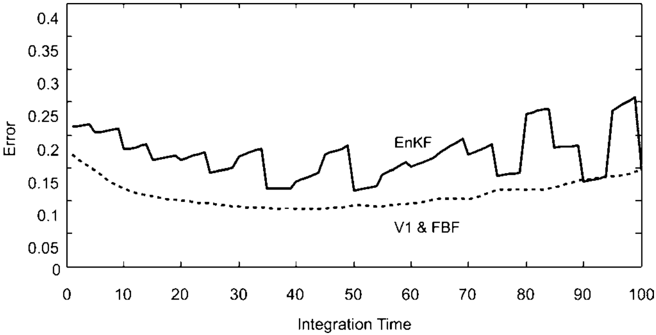
$$A_k^s = A_k^a \prod_{j=k+1}^{N} X5_j$$

$$C_k = \prod_{j=k+1}^{N} X5_j = X5_{k+1} C_{k+1}$$

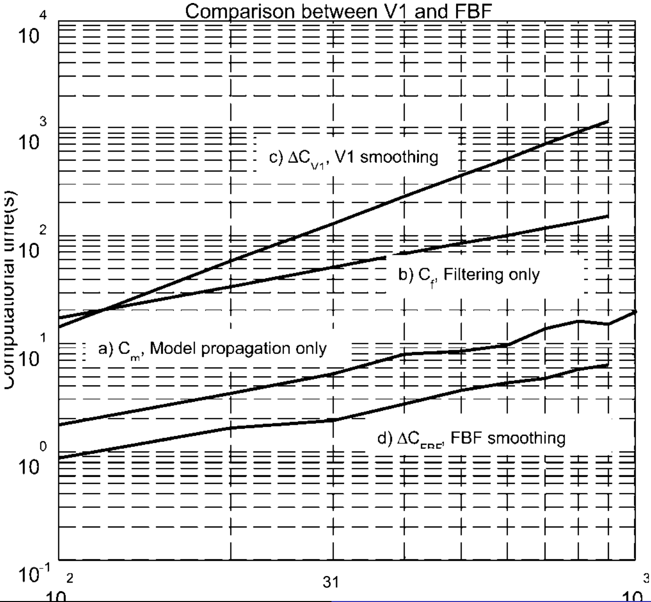
Fixed Interval: The New Normal



Fixed Interval on Lorenz

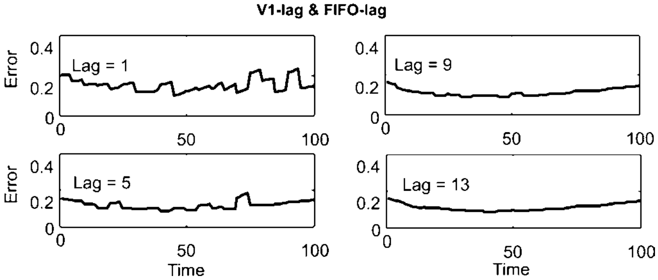


Costs of Inference, Toy Problem

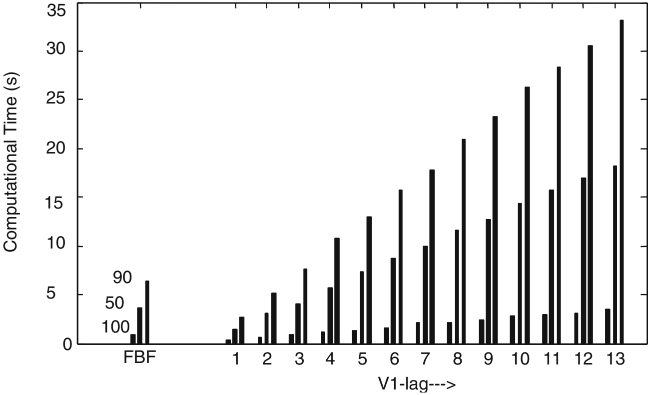


Fixed Lag

Fixed Lag Smoother



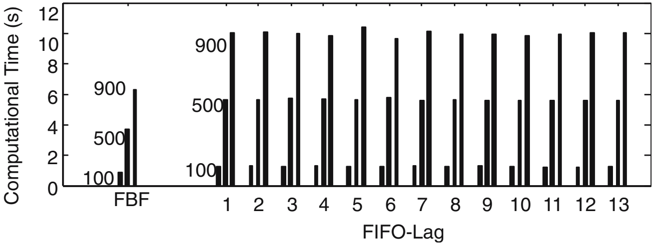
Fixed Lag: The Dumb Way



Fixed Lag is FIFO

$$
\begin{aligned}
A_k^s &= A_k^a \prod_{j=k+1}^{k+w} X5_j \\
&= A_k^a C_k \\
C_k &= X5_k^{-1} C_{k-1} X5_{k+w}
\end{aligned}
$$

Fixed Lag: The New Normal

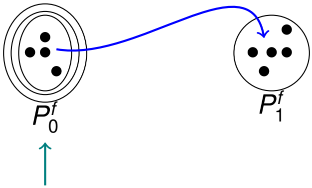


We need to fix the multimedia!

Watch FLKSO.wmv!
Reading: Ravela and McLaughlin, Fast Ensemble Smoothing, Ocean Dynamics, 2007
Schneider 2001: Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values, Journal of Climate

Where does ensemble come from?

singular vectors



Low dimensional
Subspace
Span $\{\underline{u}^{(0)} \ldots \underline{u}^{(N)}\}$

$$\underline{u}_0^{(0)} \rightarrow L_0 \approx \left.\frac{\partial M}{\partial x}\right|_{x=x_0} \rightarrow \underline{u}_1^{(0)}$$

Things get tough... the Tough linearize

Thus

$$
\begin{aligned}
\underline{x}_1 &= M(\underline{x}_0) \\
&= M(\bar{\underline{x}}_0 + \tilde{\underline{x}}_0) \\
&= M(\bar{\underline{x}}_0) + \left.\frac{\partial M}{\partial \underline{x}}\right|_{X=\bar{\underline{x}}_0} \tilde{\underline{x}}_0 \\
\tilde{x}_1 &= \mathcal{L}\tilde{x}_0 \\
\underline{u}_1^{(k)} &= \mathcal{L}\underline{u}_0^{(k)}
\end{aligned}
$$

Eigenvalue Problem

Now,let C_1 be a metric on vector $\underline{u}_1$ and let C_0 be a metric on $\underline{u}_0$

$$\lambda = \frac{< \mathcal{L}\underline{u}_0, C_1 \mathcal{L}\underline{u}_0 >}{< \underline{u}_0, C_0 \underline{u}_0 >} = \frac{< \underline{u}_0, \mathcal{L}^{\#} C_1 \mathcal{L}\underline{u}_0 >}{< \underline{u}_0, C_0 \underline{u}_0 >}$$

Maximize ratio for the k^{th} perturbation: λ_k:

$$\Rightarrow \mathcal{L}^{\#} C_1 \mathcal{L}\underline{u}_0^{(k)} = \lambda_k C_0 \underline{u}_0^{(k)}$$

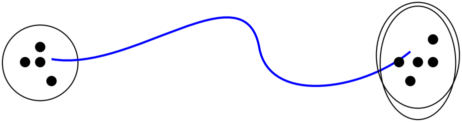
Which is a generalized eigenvalue problem. Note that when $C_1 = I$, and $C_0 = P_0^f$ then $\underline{u}_1^{(k)}$ are leading directions of P_1^f

SV aproach

Notes

- Adjoint & TLM not easy to calculate but robust.
- $\mathcal{L}$ may be really large too! How can we reduce $\mathcal{L}$?
- Sensitivity to norm.

Breeding



Initial
Perturbation

Align with leading
directions of error growth
(Lyapunov vectors)

$$\underbrace{Q_{i+1} R_{i+1}}_{\text{Q R decomposition}} = \underbrace{\mathcal{L}}_{TLM} Q_i$$

$$Q_0 \equiv I \qquad Q_0 \rightarrow Q_i \cdots \underbrace{Q_k}_{forgets Q_0}$$

It's easy to breed

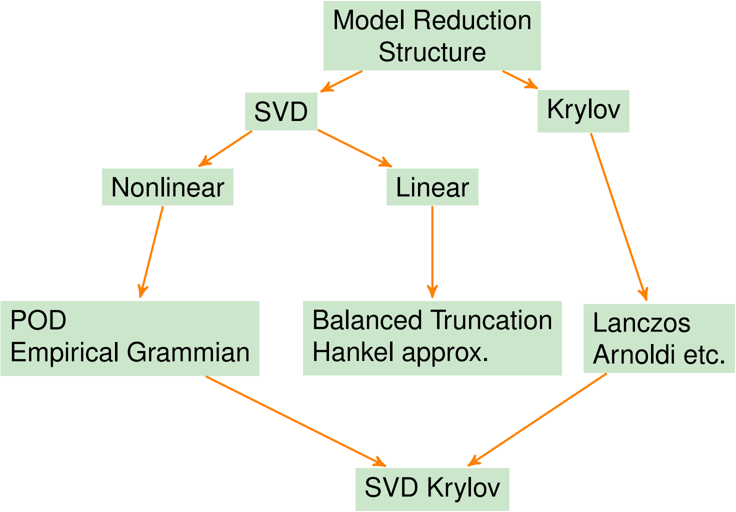
1. Generate "random" initial perturbation
2. Let it grow; renormalize. (i.e propagate it)
3. Repeat

$\Rightarrow$ Breeding vectors
How many bred vectors ?
$\Rightarrow$ Size of L?

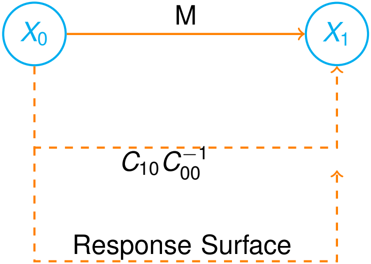
Ways to simplify Models for Uncertainty Propagation

1. Spectral Truncation: Find a few leading directions of Covariance or Model and propagate them. Breed Vectors. Calculate a reduced local linear model from ensemble.
2. Localization: Localize filtering and smoothing, use scale-recursive decomposition.
3. Model Reduction: Reduce order of linearized model, construct a reduced model from snapshots.
4. Sample Input-Output pairs to create a simple auxiliary model.

Model Reduction



Model Bypass – Non-Intrusive Approaches



Response Surface Modeling, Polynomial Chaos.

Extra-Special on Covariance Representations

If we have a large covariance matrix C nonetheless representable by computer and if we know it is a block-circulant matrix, then the Fourier Transform can be used to diagonalize it:

$$D = UCU^T \tag{10}$$

For the unitary transform U, and D is diagonal. So, subsequent processing with covariance is simplified, provided the model and state can be also expressed in fourier domain.

$$\begin{aligned} U\delta x_{n+1} &= U\frac{\partial M}{\partial x}U^T \;\; Ux_n && (11)\\ \delta\xi_{n+1} &= \mathcal{L}_{\mathcal{F}}\delta\xi_n && (12)\\ \delta x_n^T C_n^{-1}\delta x_n &= \delta\xi^T D_n^{-1}\delta\xi && (13) \end{aligned}$$

Spectral truncation to a few wave numbers in U also leads to a reduced order model. Incidentally, similar process for wavelet decomposition.

DO MATLAB EXAMPLE

Iterative calculation

If a covariance C has eigen vectors U and eigenvalues λ, i.e. $CU = U\Lambda$, then we may recursively calucate the leading modes in U because:

$$C = \sum_{k=1}^{N} \underline{u}_k \lambda_{kk} \underline{u}_k^T \tag{14}$$

Where $U = [\underline{u}_1 \ldots \underline{u}_N]$ and $\Lambda = diag(\lambda_{11}, \ldots, \lambda_{NN})$, in decreasing order. Let $C_{11} = C$, and iteratively calculate:

$$\begin{aligned} for\ k &= 1 \ldots N && (15)\\ \{\underline{u}_k, \lambda_{kk}\} &= LeadingEig(C_{kk}) && (16)\\ C_{k+1k+1} &= C_{kk} - \underline{u}_k \lambda_{kk} \underline{u}_k^T && (17)\\ & && (18) \end{aligned}$$

We need a procedure to calculate the leading Eigenvector and Eigenvalue.

Basic approach: Power Iteration

WARNING: There are many advanced methods for calculating eigen vectors and eigen values iteratively and one should use them (e.g. from ARPACK). Here, we provide an intuition for the process.
To calculate the leading vector of C, let us consider a vector in the basis $\underline{z}$, which we may expand as:

$$\underline{z} = \sum_{k=1}^{N} c_k \underline{u}_k \tag{19}$$

Now, we can write for the n^{th} power of C:

$$\begin{aligned} C^n \underline{z} &= \sum_{k=1}^{N} c_k C^n \underline{u}_k & (20) \\ &= \sum_{k=1}^{N} c_k \lambda_{kk}^n \underline{u}_k & (21) \end{aligned}$$

HOW?

Power Iteration Continued

$$C^n \underline{z} \quad = \quad c_1 \lambda_{11}^n \left(\underline{u}_1 + \sum_{k=2}^{N} \frac{c_k}{c_1} \frac{\lambda_{kk}^n}{\lambda_{11}^n} \underline{u}_k \right) \tag{22}$$

Defining $\underline{z}_n = C^n \underline{z}$, we note that

$$n \rightarrow \infty \Rightarrow \frac{\underline{z}_n}{||\underline{z}_n||} \rightarrow \underline{u}_1 \tag{23}$$

Algorithm *PowerIteration*(C):

Initialize $\underline{z}$; $\underline{z} \leftarrow \frac{\underline{z}}{||\underline{z}||}$

Iterate: $t \leftarrow Cz$, $z \leftarrow \frac{\underline{t}}{||\underline{t}||}$

LeadingEig(C)

$$\begin{aligned} \underline{u} &= \mathit{PowerIteration}(C) \qquad (24) \\ \lambda &= u^T C u \qquad (25) \\ \mathit{return}(\underline{u}, \lambda) & \quad \mathit{end} \qquad (26) \end{aligned}$$

But the Covariance is too LARGE!

- The preceding discussion is all fine, but often the dimensionality is such that we have a really large covariance that cannot be represented. Fortunately, many physical problems have only a few modes of interest which we represent through data, e.g. an ensemble.
- So we begin with a skinny matrix X, and assume the covariance is $C = XX^T$. We would like a representation without explicitly calculating C and exploiting the rank-deficiency due to a skinny X.

Alternate form

- Let $X = USV^T$ be the singular value decomposition and here $S_{ii} \geq S_{i+1i+1}$. Then $C = U\Lambda U^T$ where $\Lambda = S^2$
- We will calculate only a few top left and right singular vectors and singular values **iteratively** for a reduced order representation $C_d = U_d \Lambda_d U_d^T$.
- Note that because X is skinny, i.e. it is of size $n \times N$ with $N << n$. We may further only pick d modes, $d \leq N$.
- We would like a representation of C_d without explicitly calculating it.
- Notice that $D = X^T X$ is a small matrix when X is skinny.

Alternate form

- Let $X = USV^T$ be the singular value decomposition and here $S_{ii} \geq S_{i+1i+1}$. Then $D = X^T X = V \Lambda V^T$ where $\Lambda = S^2$, a small matrix.
- We calculate the eigen vectors and eigen values of D recursively. Let $D_1 = D$; and for $k = 1 \ldots d$

$$\begin{aligned} \underline{v}_k &= PowerIteration(D_k) &\quad (27)\\ \lambda_{kk} &= \underline{v}_k^T D_k \underline{v}_k &\quad (28)\\ D_{k+1} &= D_k - \underline{v}_k \lambda_{kk} \underline{v}_k^T &\quad (29) \end{aligned}$$

Noting that $S_d = \sqrt{\Lambda_d}$, we obtain U_d as a skinny *nxd* matrix:

$$U_d = X V_d S_d^{-1} \quad (30)$$

Store U_d and Λ_d and use them to calculate the norm in an application. DEMO IN MATLAB

Applicable to Processes

$$\frac{\partial \theta}{\partial t}(x,t) = F\theta(x,t) \rightarrow \mathit{System}$$

$$R(\theta) = \frac{\partial \theta}{\partial t} - F\theta \rightarrow \mathit{Residual}$$

$$\theta = u\eta(t) \rightarrow \text{KLT (POD or Krylov)}$$

$$u^T R = 0 \rightarrow \text{Galerkin Projection}$$

$$\frac{\partial \eta}{\partial t} = u^T F u \eta \rightarrow \mathit{ROM}$$

12.S990 Quantifying Uncertainty
Fall 2012