Optimal DATA Assimilation

Ronald DeVore

Collaborators: Peter Binev, Albert Cohen,

Wolfgang Dahmen, Guergana Petrova,

Przemek Wojtaszczyk

Dinner with Carl

Visit to China

Young Carl

My # 60

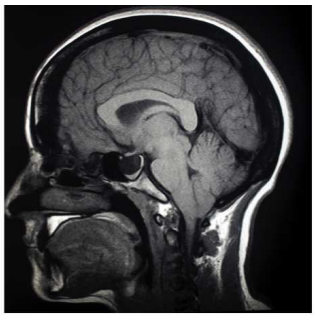
Presidential Medal

The Tuxedo

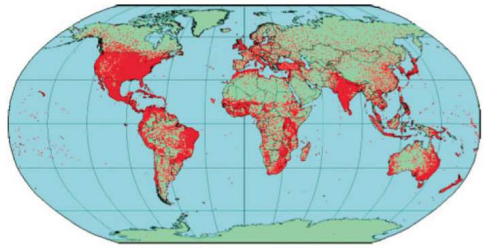
Data Fitting

- Common Scientific Problem: We are given data about some underlying function f (scientific process) and we wish to 'fit the data' to answer some question about f
- This talk will concentrate on
  - Optimality of Algorithms
  - Certifiable Performance
- We put forward general principles can be tailored to any specific application

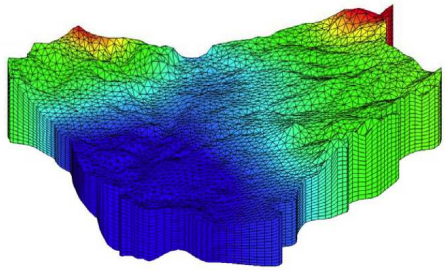
Your Favorite Application



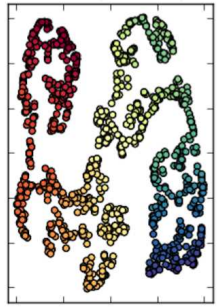
MRI



Global Temperatures



Groundwater Modeling



Manifold Learning

Data Tasks

- Two types of tasks
  - Prediction: Approximate f
  - Quantity of Interest: calculate some narrower quantity
    - maximum/minimum of f
    - average behavior: calculate an integral of f
    - value of f at some designated point
- This talk will concern algorithms which, in some sense, can be proven optimal for recovering f or answering questions of interest about f: optimal and certifiable performance

Mathematical Formulation

- Consider the full approximation problem for f
  - Form of the Data?: We assume $w_j = l_j(f), \quad j = 1, \ldots, m$, where l_j are linear functionals
    - Measurement map $M(f) = w := (w_1, \ldots, w_m)$
  - How to measure performance? We measure distortion by a norm $\|\cdot\|_X$ with X a Banach space
- An algorithm is a mapping $A : I\!R^m \mapsto X$ where $A(M(f))$ is an approximation to $f \in X$ giving error

$$E(f, A)_X := E(f, M, A)_X := \|f - A(M(f))\|_X$$

Model Classes

- With no other information we can say nothing about the error or discuss best algorithms
- To state a meaningful problem we need to have additional information about f
- This additional information is typically given in an assumption that $f \in \mathcal{K} \subset X$
- The set $\mathcal{K}$ is called a model class
- Typical deterministic model classes $\mathcal{K}$ are given by
  - smoothness: Lipschitz, Sobolev, Besov balls
  - spectral conditions: bandlimited;
  - In high dimensions, i.e. when f depends on a lot of variables, model classes are built on sparsity, compressibility, anisotropy, variable reduction, etc.

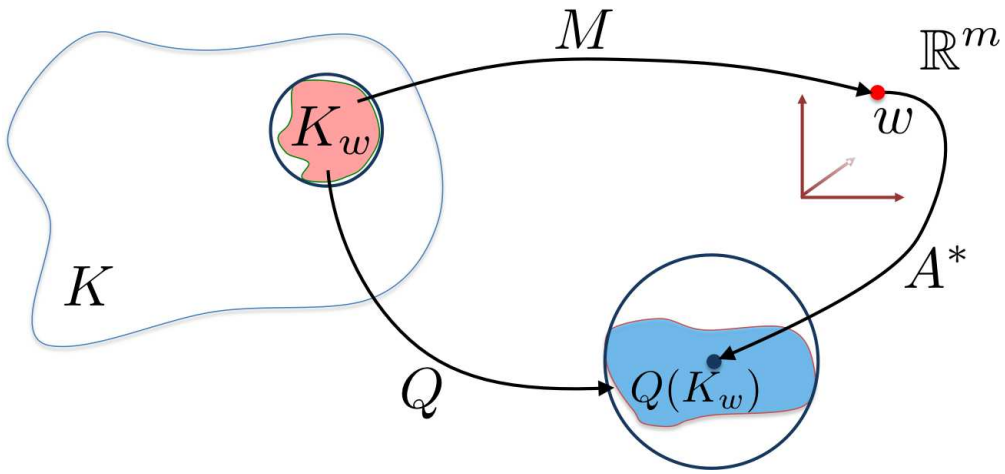
Model Class

- An accurate description of the model class $\mathcal{K}$ is the most important ingredient in data assimilation
- The more info we have on $\mathcal{K}$ the better we can do
- In scientific computation this is extracted by understanding the scientific process: for example, bandlimits for signals, regularity theorems for PDEs
- In other settings this is more nebulous and so one seeks algorithms that are universal, i..e. work simultaneously for a wide range of model classes

Optimal Recovery: Best Algorithms

- $\mathcal{K}$, $\|\cdot\|_X$ fixed and consider any algorithm A
- Define $\mathcal{K}_w := \{f \in \mathcal{K} : M(f) = w\}$
  - Membership in $\mathcal{K}_w$ is all we know about f
- Pointwise error: $E(\mathcal{K}_w, M, A) := \sup_{f \in \mathcal{K}_w} \|f - A(w)\|_X$
- Worst case error:
  $E(\mathcal{K}, M, A) := \sup_{f \in \mathcal{K}} \|f - A(Mf))\|_X = \sup_{w \in I\!R^m} E(\mathcal{K}_w, M, A)$
- Optimal Performance: $\boxed{E^*(\mathcal{K}, M) := \inf_A E(\mathcal{K}, M, A)}$
- Optimal Recovery: The best algorithm A^*
  - Let $B(g_w, R_w)$ be the smallest ball that contains $\mathcal{K}_w$
  - $A^* : w \mapsto g_w$ is an algorithm that is pointwise optimal
    $E(\mathcal{K}_w, M, A^*)_X = E^*(\mathcal{K}_w, M) = R_w$

Graphic for Optimal Recovery



Not so Fast!

- You may think that this is the end of the story
  - But finding the Chebyshev ball is a substantial problem and is only carried out in certain special settings: for certain $\mathcal{K}$ and certain distortion metrics $\|\cdot\|_X$
  - Results where optimal recovery is known are summarized in Micchelli-Rivlin
- The main point of this talk is to point out that there is a general setting where we can determine optimal or near optimal algorithms and we can determine a priori the optimal performance
  - This setting will also expose when one has good data or bad data

Approximation Sets

- Any algorithm will be based on some form of approximation!
- Let $V = V_n$ be the functions used in the approximation: polynomials, neural nets, wavelets, sparse sums, etc.
- Since we have chosen V we think $\mathcal{K}$ is described by the fact it is well approximated by V
- Natural Model class: Approximation set:

$$\mathcal{K} := \mathcal{K}(\epsilon, V) = \{f : \ \mathrm{dist}(f, V)_X \leq \epsilon\}$$

- We shall describe algorithms which are optimal over all ϵ and you do not need to know ϵ

Performance estimates

- Full approximation problem: Performance determined by V and null space $\mathcal{N} := \{f \in X : \ M(f) = 0\}$ via

$$\mu(\mathcal{N}, V) := \mu(\mathcal{N}, V)_X := \sup_{\eta \in \mathcal{N}} \frac{\|\eta\|}{\operatorname{dist}(\eta, V)}$$

- When X is a Hilbert space best performance for an approximation set $\mathcal{K} = \mathcal{K}(\epsilon, V)$ is

$$\boxed{E^*(\mathcal{K}, M) = \mu(\mathcal{N}, V)\epsilon}$$

- When X is a general Banach space best performance $E(\mathcal{K}, M)$ for an approximation set $\mathcal{K} = \mathcal{K}(\epsilon, V)$ satisfies

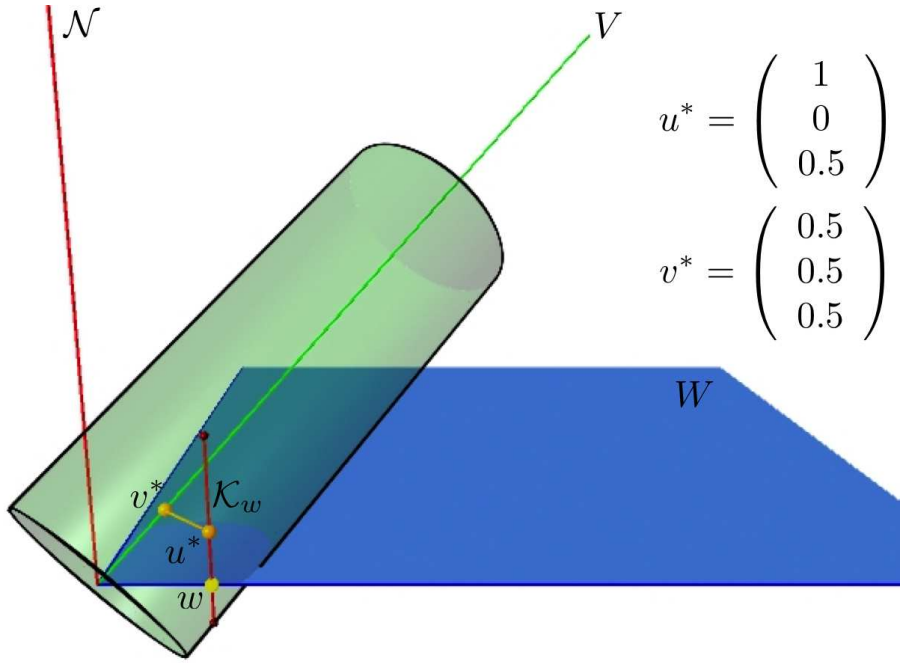
$$\mu(\mathcal{N}, V)\epsilon \le E(\mathcal{K}, M) \le 2\mu(\mathcal{N}, V)\epsilon$$

- Important: μ is easy to compute and (near) best algorithms can be described as will follow

A simple example

- Take X to be a Hilbert space
- If $l_j(f) = \langle f, \omega_j \rangle$, $j = 1, \ldots, m$ with $(\omega_j)_{j=1}^m$ ONS, then
  - $v^*(w) := \underset{v \in V}{\operatorname{Argmin}} \|w - M(v)\|_{\ell_2}$
  - $A : w \mapsto v^*(w)$ is near optimal with constant 2
  - If $u^*(w) \in \mathcal{K}_w$ is the closest element $v^*(w)$ then $A^* : w \mapsto u^*(w)$ is linear and pointwise optimal
- Best algorithm is essentially least squares fit:
- μ can be computed by SVD of cross Grammian
- What is new?: Generally you do not see μ and performance estimates for least squares
- Note: Data is good if μ is small and bad if μ is large

Hilbert space geometry



$$u^* = \begin{pmatrix} 1 \\ 0 \\ 0.5 \end{pmatrix}$$

$$v^* = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}$$

Choosing V

- The above optimal estimates take the form
  $\|f - A(M(f)\|_X \leq C\mu(\mathcal{N}, V)\,\mathrm{dist}(f, V)$
- Here there is a competition between μ and $\mathrm{dist}(f, V)$
  - Increasing the complexity of V improves $\mathrm{dist}(f, V)$ but increases $\mu(\mathcal{N}, V)$
- I want to illustrate this with a (toy) example
- $X = C(D)$ with D a domain in $\mathbb{R}^d$
- $w_j = l_j(f) = f(x_j)$ with $x_j \in D$, $j = 1, \ldots, m$
- $V \subset C(D)$ a linear space of dimension $n \leq m$
- $\mu(\mathcal{N}, V) = 1 + \mu(V, \mathcal{N})$ where

$$\mu(V, \mathcal{N}) = \sup_{v \in V} \frac{\|v\|_{C(D)}}{\max_{1 \leq j \leq m} |v(x_j)|}$$

Point values

- Near best algorithm is $v^* := \mathrm{Argmin}_{v \in V} \|w - M(v)\|_{\ell_\infty}$
- Example $X = C([0,1])$, $\xi_1, \ldots, \xi_m$ equally spaced, $V = \mathcal{P}_{n-1}$ - polynomials of degree $< n$. Then it is known
  - If you choose $n = m$ then $\mu(\mathcal{N}, \mathcal{P}_m) \approx a^N, \ a > 1$
  - If $n = \sqrt{m}$ then $\mu(\mathcal{N}, \mathcal{P}_n) \leq 3$
  - This gives $\|f - A(M(f))\|_C \leq 3 \, \mathrm{dist}(f, \mathcal{P}_{\sqrt{n}})_C$
- This points out the importance of the choice of V
- Do not interpolate!!
- Analogy with statistical learning: Do not overfit data
- computing μ tells you what overfit means

High dimension

- What happens when f depends on many variables/parameters: many features in data
  - The main issue is what is the correct model class $\mathcal{K}$ - what is the correct V to avoid the curse of dimensionality
  - Model classes $\mathcal{K}$ are proposed built on sparsity, anisotropy, variable reduction, feature selection, etc.
  - Typical V are built on highly nonlinear methods such as dictionary appproximation, neural networks
  - To have a quantitative theory (certifiable performance) we need to understand
    - Which functions are approximated well by V - if and only if theorems
    - What is $\mu(\mathcal{N}, V)$ for given data and V
    - Computational complexity of optimal algorithms

Additional Remarks

- The main references for the above are: Binev-Cohen-Dahmen-DeVore-Petrova-Wojtaszczyk (Hilbert space), DeVore -Petrova-Wojtaszczyk (Banach space)
- Closely related work emphasizing more the issue of stable computation is given by Adcock, Hansen, Shadrin, Trefethen, et al

Linear Algorithms

- An interesting question is whether there are optimal or near optimal algorithms that are linear
  - For Hilbert space this is clear from the above
  - For $X = C(\Omega)$ and $l_j(f) = f(x_j)$ this can be proved using generalizations of Kalman's convexity theorem (DeVore-Foucart-Petrova- Wojtaszczyk
  - DFPW There is a linear algorithm $A^*(w) = \sum_{j=1}^m w_j \phi_j(x)$ with $\phi_j \in C(D)$
  - For each $x \in D$, the mapping $w \mapsto \sum_{j=1}^m w_j \phi_j(x)$ is optimal for recovering $\delta_x(f) = f(x)$
  - The proof based on Functional Analysis and is not constructive

Quasi-interpolants

- A very constructive way to find a near optimal algorithm A is through quasi-interpolant operators for V
  - A linear operator $\Lambda : C(D) \mapsto C(D)$ is called a quasi-interpolant if there exist points $\xi_j \in D,\ j = 1, \ldots, N$, and $\psi_j \in C(D)$ such that the operator $\Lambda(f) := \sum_{j=1}^{N} f(\xi_j)\psi_j$ satisfies
  - $\Lambda(v) = v$ for all $v \in V$
  - $\|\Lambda\|_{X \mapsto X} \leq C_0$
- There always exists quasi-interpolants - the issue is how large is N? For algebraic or trigonometric polynomials one can take $N = 2n$ with $C_0 \leq 4$
- Once a quasi-interpolant is known, one can construct a near optimal linear algorithm by solving N constrained ℓ_1 minimization problems

Quantities of Interest

- A similar theory of optimal recovery exists for quantities of interest Q
- Performance now controlled by

$$\mu(\mathcal{N}, V, Q) := \mu(\mathcal{N}, V, Q)_X := \sup_{\eta \in \mathcal{N}} \frac{\|Q(\eta)\|}{\operatorname{dist}(\eta, V)}$$

- For any Banach space X we have the performance bounds

$$\mu(\mathcal{N}, V, Q)\epsilon \leq E(Q, \mathcal{K}(\epsilon, V), M) \leq 2\mu(\mathcal{N}, V, Q)\epsilon$$

Constructive Opt. Linear Algorithm

- When $\mathcal{K}$ is an approximation set and Q is a linear functional then one can find an optimal algorithm that is linear by constrained optimization:
- Let $\mathcal{L}_Q := \{l = \sum_{j=1}^m a_j l_j : \ l(v) = Q(v), \ v \in V\}$ and

$$l^* := \operatorname*{Argmin}_{l \in \mathcal{L}_Q} \|Q - l\|_{X^*} = \sum_{j=1}^m a_j^* l_j$$

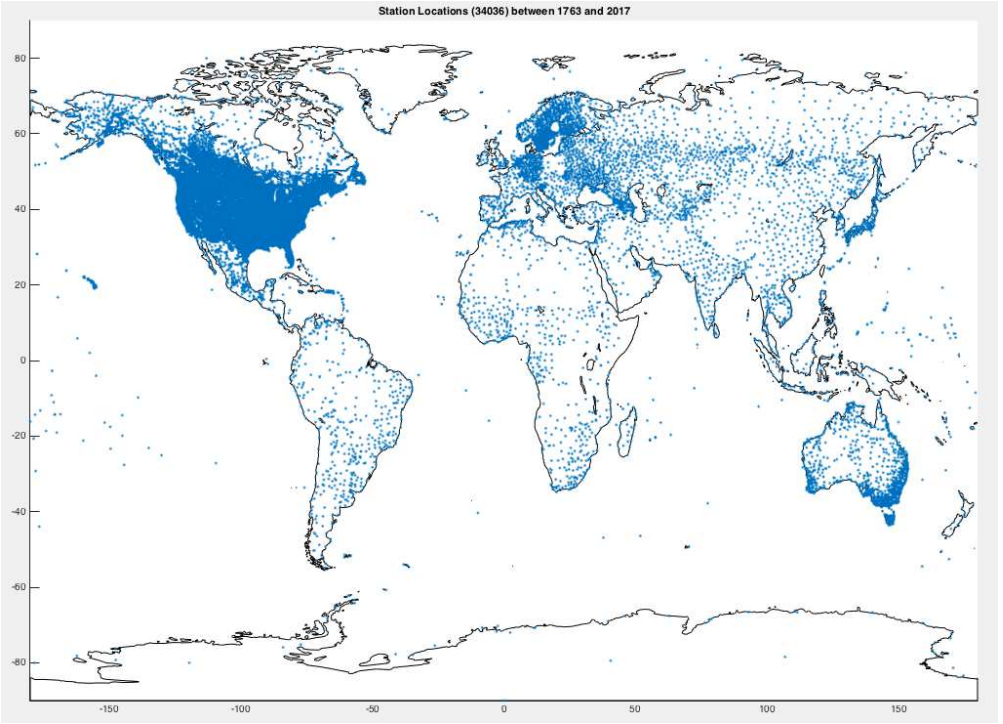
- Then $\boxed{A^* : w \mapsto \sum_{j=1}^m a_j^* w_j}$ is an optimal algorithm
- Note this may be numerically intensive constrained minimization
- Perf: $\boxed{|Q(f) - A^*(Mf)| \le \|Q - l^*\|_{X^*} \operatorname{dist}(f, V)_X}$
- You see $\mu \le \|Q - l^*\|_{X^*}$

Example: Quadrature

- Integration: Option trading, uncertainty quantification, Quasi-Monte Carlo, etc.
- Data are point values $l_j(f) = f(x_j),\ j = 1, \ldots, m$,
- We want to compute $Q(f) = \int_D \omega(x) f(x)\, dx,\ f \in \mathcal{K}(\epsilon, V)$
- The optimal quadrature on $X = C(D)$ using the points $x_j \in D$ is
  - $A^*(f) = \sum_{j=1}^m a_j^* f(x_j)$
  - $(a_j^*) := \text{Argmin}\{\sum_{j=1}^m |a_j| :\ \sum_{j=1}^m a_j v(x_j) = \int_D \omega(x) v(x)\, dx\}$
- This is a constrained ℓ_1 minimization problem
- $\mu(\mathcal{N}, V, Q) = \sum_{j=1}^m |a_j^*|$
- $|\int f - A^*(M(f))| \le \mu(\mathcal{N}, V, Q)\, \text{dist}(f, V)_{C(D)}$

Example: Global Temperature

- Let $T(x,t)$ denote temperature at position x on earth and time t
- Quantity of interest $Q(T) = \int_{Year} \int_{Earth} T(x,t)\, dx\, dt$
- Roughly 14K sites from 1950 till 2017

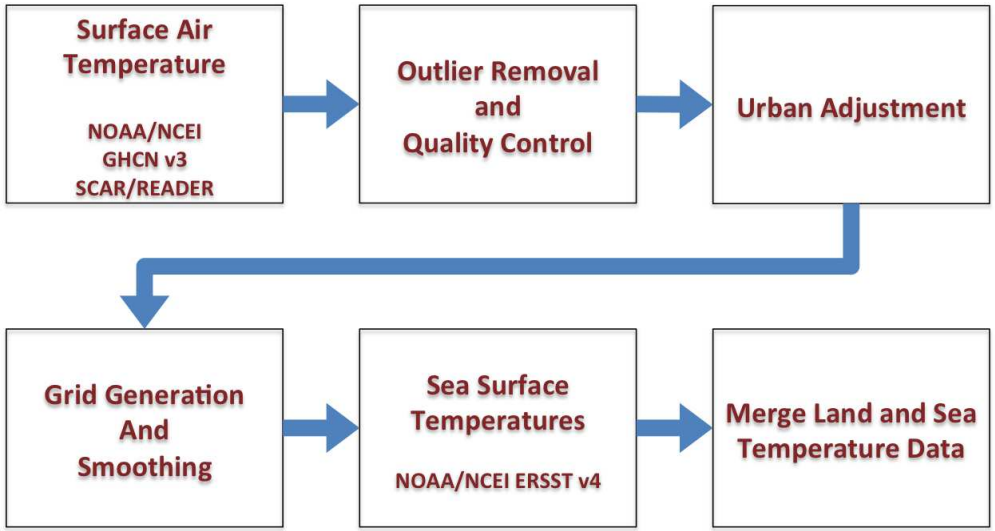


Obstacles to Mathematical Analysis

- Life would be good if
  - We knew the right model class for $T(x,t)$ - the right V
  - if data sites, equipment, and measuring times did not change each year
- Current algorithms use models based on pw polynomials - not clear what space
- We will use spherical harmonics
- We compare Spherical Harmonics versus GISTemp (NASA) on their adjusted data set
- We can compute μ for spherical harmonics but not for GISTemp

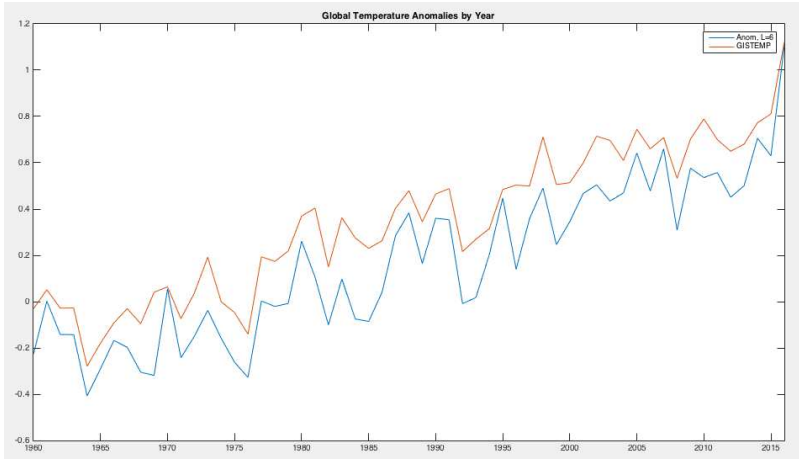
Current Algorithms

- There are many algorithms
- The following flowchart gives the main steps of the NOAA and NASA algorithms using piecewise polynomials on a uniform grid

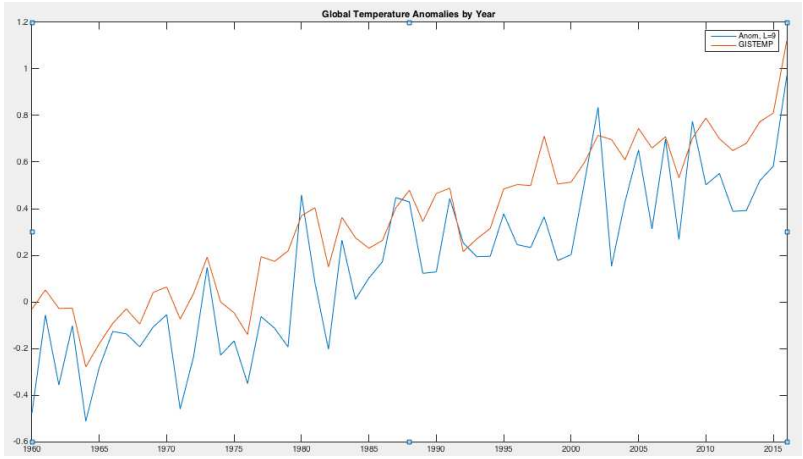


- Impossible to analyze accuracy because of the ad hoc adjustments to the data

Comparison:GISTempvs. SH6



Comparison: GISTemp vs. SH9



Typical Growth of μ

- Are we computing global temperature?
  - This would require proving validity of our model class: would require analysis from physical principles
  - Also depends on behavior of μ

n	3	6	9	12	15	18
μ	1	1.03	2.61	24.13	223.50	2779.85

- We see that even if we justify our model class, we need to restrict the size of n

Summary

- We have given a mathematical view of Data Assimilation
  - This theory require a valid model class for the functions we want to capture
  - If this validity is established then
    - We have given optimal algorithms
    - We have given certified performance of these algorithms
- The challenge in application scenarios is
  - verification of the correct model class - this is especially challenging in high dimensions
  - Feasibility of the computation of an optimal algorithm
  - Understanding μ