Strategies for Capturing High Dimensional Functions

Ronald DeVore

Texas A& M University

Challenging World Problems

- Some of the most pressing scientific problems challenge our computational ability
  - Atmospheric modeling: predicting climate change
  - Monitoring threat activities
  - Contaminant transport
  - Optimal engineering design
  - Medical diagnostics
  - Modeling the internet
  - Option pricing, bond valuation
  - ....

Mathematical/Computational Challenge

- One common characteristic of these problems is they involve processes with many variables or parameters
- Mathematically this means we are faced with numerically approximating a high dimensional function
  - $F : [0,1]^D \to X$
  - X a Banach space (often just $\mathbb{R}$ or $\mathbb{R}^m$)
  - D large and possibly infinite
  - Typical Computational Tasks
    - Create an approximation $\hat{F}$ to F
    - Evaluate some quantity of interest: $Q(F)$
    - Q is some linear or nonlinear functional, for example
      - $Q(F)$ is a high dimensional integral of F
      - $Q(F)$ is the max or min of F

Evaluating Algorithms

- To have a meaningful discussion of the quality of algorithms one needs
  - a norm on functions to measure error $\|\cdot\| = \|\cdot\|_Y$
  - Typically Y is an L_p space or uniform norm ($p = \infty$)
  - the assumptions made on F
- We view the assumptions we make about F as placing F in a model class $\mathcal{K}$ which is a compact subset of Y
  - In numerical analysis of the last century model classes were almost exclusively smoothness spaces - how many derivatives does F have
  - Statistical model classes place restrictions on the regression function or the probability distribution
  - In Signal/Image Processing conditions on the Fourier Transform of F - e.g. band limited

Bad News

- Classical model classes based solely on smoothness of F are not sufficient in high dimensions
  - Suppose the assumption is that F is real valued and has smoothness (of order s)
    - Approximation theory tells us with n computations we can only capture F to accuracy $C(D, s)n^{-s/D}$ where D is the number of variables
    - When D is large than s must also be very large to guarantee any reasonable accuracy
    - But we have no control over s which is inherent in the real world problem
    - So conventional assumptions on F and conventional numerical methods will not work
- Also beware that $C(D, s)$ grows exponentially with D

Example (Novak-Wozniakowski)

- To drive home the debilitating effect of high dimensions consider the following example
$\Omega := [0,1]^D, \quad X = I\!R, \quad \mathcal{K} := \{F : \; \|D^\nu F\|_{L_\infty} \leq 1, \; \forall \nu\}$
- Any algorithm which computes for each $F \in \mathcal{K}$ an approximation $\hat{F}$ to accuracy $1/2$ in L_∞ will need at least $2^{D/2}$ FLOPS
- So if $D = 100$, we would need at least $2^{50} \asymp 10^{15}$ computations to achieve even the coarsest resolution
- This phenomenon is referred to as The Curse of Dimensionality
- The usual definition of the Curse is polynomial in d versus exponential in d growth in computational cost
- Real question is whether an acceptable error tolerance can be reached in alloted computational time

The Remedy

- Conventional thought is that most real world HD functions do not suffer the curse
- Classical smoothness models is not the right model -need new models
  - Sparsity : F is a sum of a small number of functions from a fixed basis/frame/dictionary
  - Anisotropy/Variable Reduction: not all variables are equally important - get rid of the weak ones
  - Tensor structures: variable separability
  - Superposition: F is a composition of functions of few variables - Hilbert's 13-th problem
  - Many new approaches based on these ideas: Manifold Learning; Laplacians on Graphs; Sparse Grids; Sensitivity Analysis; ANOVA Decompositions; Tensor Formats; Discrepancy

Numerical Algorithms

- Let us turn now to constructing numerical algorithms in HD -such algorithms depend on the information we are given about F
- Setting I: Query Algorithms: We can ask questions about F in the form of Queries
  - A query is the application of a linear functional to F
    - Examples: Point evaluation or weighted integrals
  - Given that $F \in \mathcal{K}$ and a query budget n - where should we query to best reconstruct F
- Setting II: Data Assimilation: We cannot ask questions but rather are given data in the form of some information about F?
  - Given that $F \in \mathcal{K}$ and given the data how can we best reconstruct F

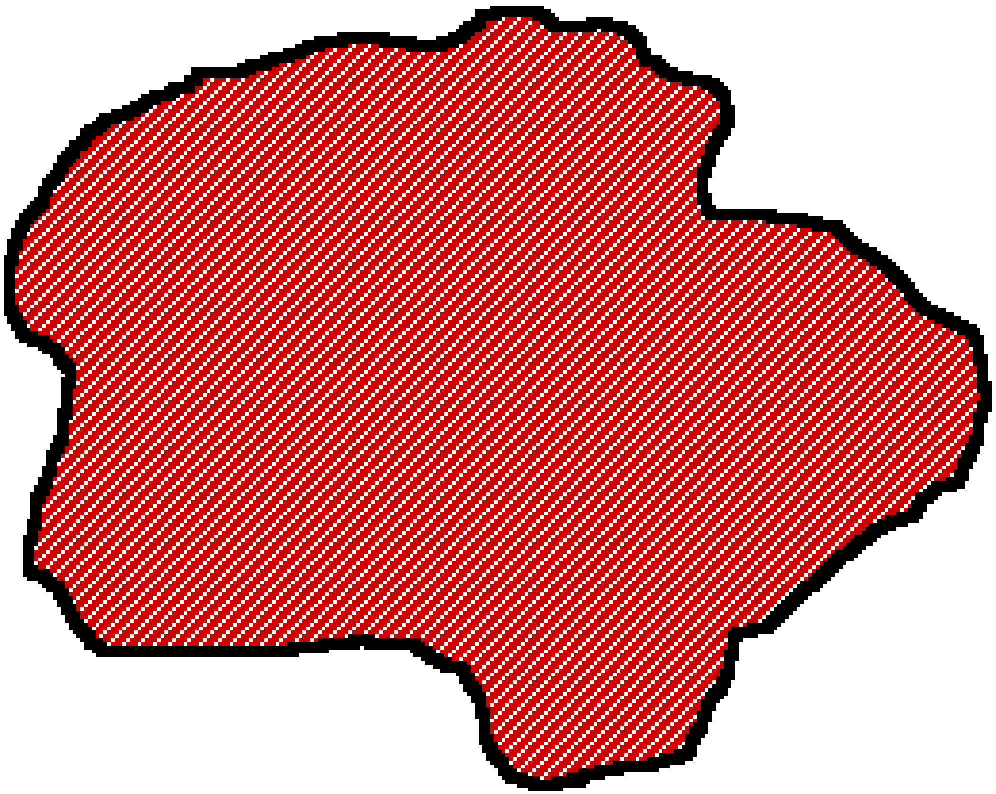
Numerical Goals

- Determine performance limits for the model class
- Does it break the curse of dimensionality?
- Certifiability of the performance of the proposed algorithm
- Rate-distortion guarantees
- Is the proposed algorithm optimal/ near optimal?

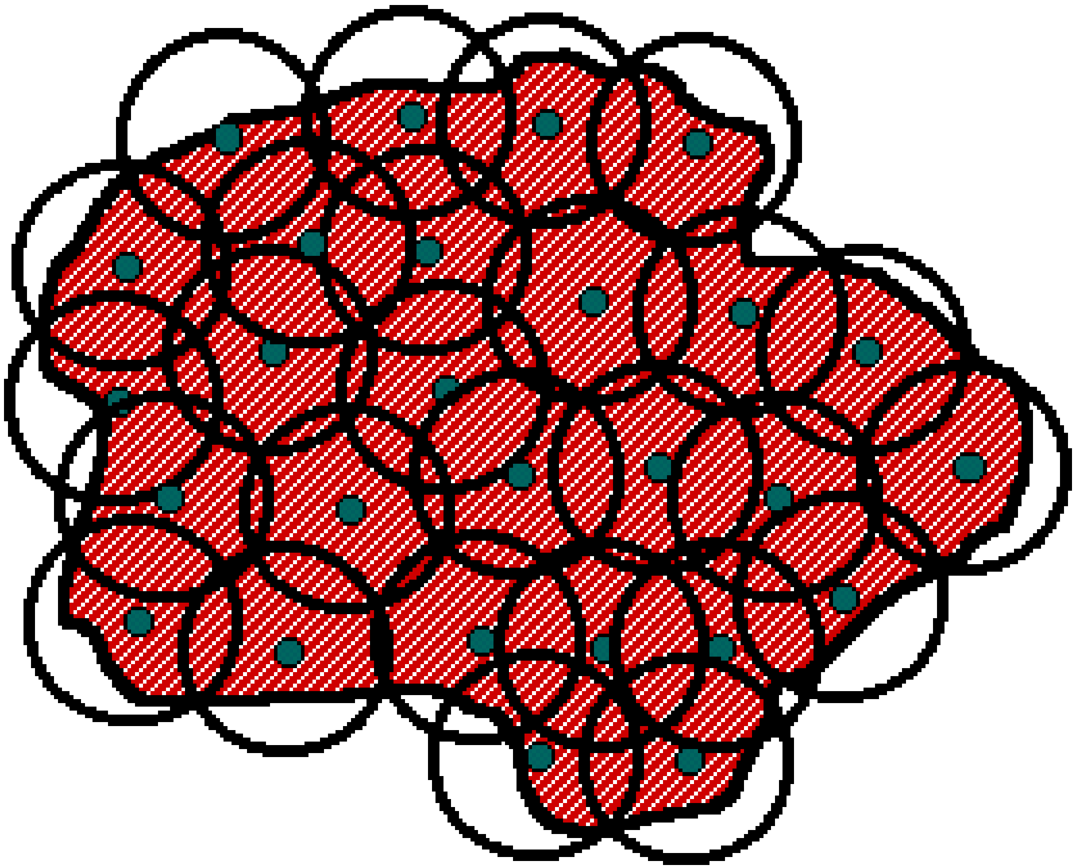
General complexity bound: Entropy

- There is a general criteria to see whether a model class $\mathcal{K}$ is HD friendly for computation
- It is given by the Kolmogorov metric entropy of $\mathcal{K}$
  - Given $\epsilon > 0$: How many balls of radius ϵ in Y do we need to cover $\mathcal{K}$?
  - $N_\epsilon(\mathcal{K})_Y$ denotes the smallest number
  - $H_\epsilon(K)_Y := \log_2 N_\epsilon(K)_Y$ Kolmogorov entropy
  - any numerical method which captures each $F \in \mathcal{K}$ to accuracy ϵ will need at least $H_\epsilon(\mathcal{K})_Y$ computations
  - So if the entropy of $\mathcal{K}$ is not reasonable this is not a useful model class
  - For example: This is how to prove the Novak-Wozniakowski result

Covering



Covering



An Example: Parametric PDEs

- $\Omega \subset I\!R^d$ domain and $\mathcal{A}$ is a collection of diffusion coefficients a that satisfy the Uniform Ellipticity Assumption: $0 < r \le a(x) \le R, \quad x \in \Omega$
- u_a solution to the elliptic problem

$$
\begin{array}{rrcll}
(*) & -\operatorname{div}(a(x)\nabla u_a(x)) & = & f(x), & x \in \Omega, \\
 & u_a(x) & = & 0, & x \in \partial\Omega
\end{array}
$$

- $a(x,y) = \bar{a}(x) + \sum_{j=1}^{\infty} y_j \psi_j(x)$, $y_j \in [-1,1]$, $j = 1, 2, \dots$
- $F(y) = u_{a(y)} \quad F : [-1,1]^{\mathcal{N}} \mapsto X$, $\quad X := H_0^1 \quad D = \infty$
- $\hat{F}$ is an on line method for computing $F(y) = u_{a(y)}$, $\forall y$

Query Algorithms

- A query algorithm extracts information $\ell_1(F), \ldots, \ell_n(F)$ and creates an approximation $A_n(F) \in Y$ to F using only the extracted data and knowledge $F \in \mathcal{K}$
- The minimal distortion in query algorithms is

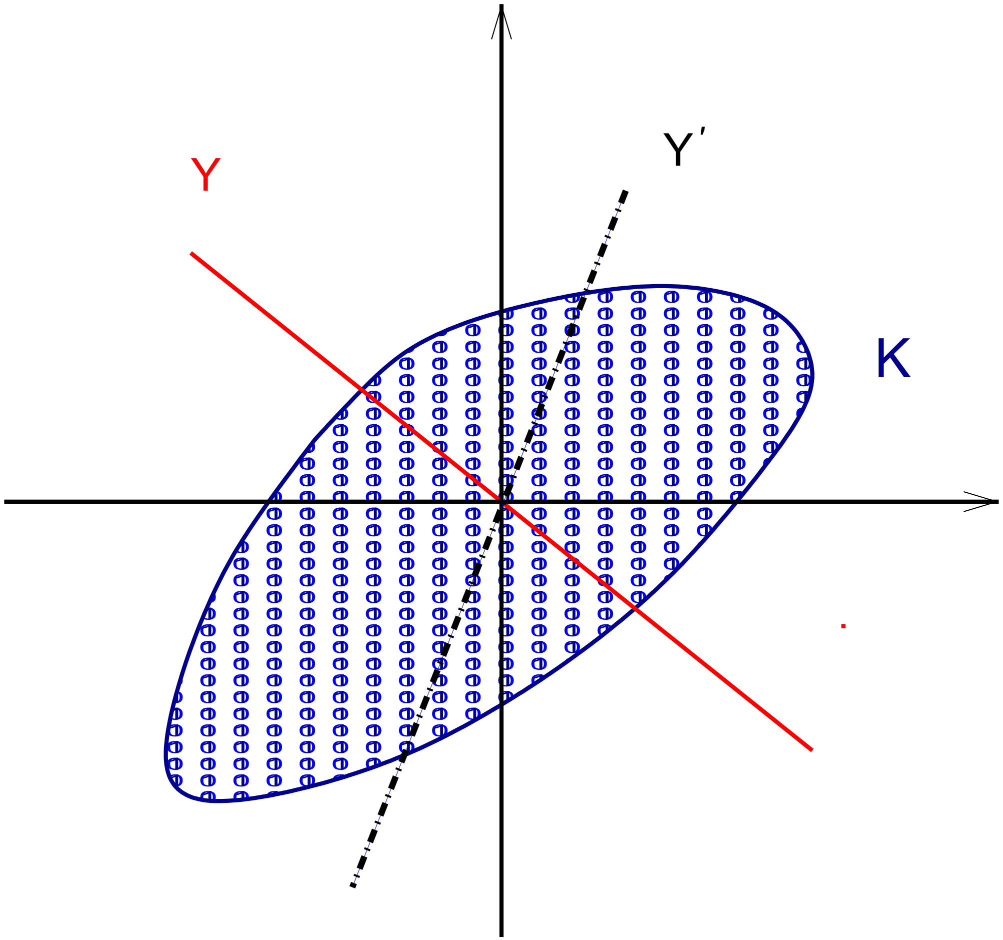
$$\delta_n(\mathcal{K}) := \inf_{A_n} \sup_{F \in \mathcal{K}} \|F - A_n(F)\|_Y$$

- If no restrictions are imposed on the queries the optimal performance is given by the Gelfand width $d^n(\mathcal{K})_Y$

$$\delta_n(\mathcal{K}) \asymp d^n(\mathcal{K})_Y = \inf_{\mathrm{codim}(V)=n} \sup_{f \in \mathcal{K} \cap V} \|f\|_Y$$

- Computing Gelfand widths of a model class could tell us whether the model class is reasonable
  - However, determining the Gelfand width does not constitute an algorithm

Gelfand Widths



Sparsity

- Let $\mathcal{D}$ be a dictionary of functions mapping $[0,1]^D \mapsto X$
- Typical examples: $\mathcal{D}$ is a basis or frame
- Define: $\Sigma_m := \{S : \ S = \sum_{g\in\Lambda} c_g g, \ \Lambda \subset \mathcal{D}, \#(\Lambda) \le m\}$
- The elements in Σ_m are said to be m sparse
- Sparsity is too restrictive to be a good model class and should be replaced by compressibility
  - $\sigma_m(F)_Y := \inf_{S\in\Sigma_m} \|F - S\|_Y$
  - $\mathcal{A}^\alpha := \{F : \ \sigma_m(F)_Y \le Cm^{-\alpha}\}$, $|f|_{\mathcal{A}^\alpha}$ is smallest C
- $\mathcal{A}^\alpha$ model class of compressible functions of order α
- Y Hilbert space, $\mathcal{D} = \{\psi_j\}$ basis $F = \sum_{j=1}^\infty a_j(F)\psi_j$
- $F \in \mathcal{A}^\alpha$ if and only if $|a_j^*(F)| \le Mj^{-\alpha-1/2}$

Compressed Sensing

- Developed for capturing sparse vectors in $x \in R^D$
  - Sparsity: x has at most m nonzero entries $m << D$
  - Sample is inner product $\nu \cdot x$ where $\nu \in \mathbb{R}^D$
  - We can view x as the linear function $F_x(y) := x \cdot y$
  - Then a sample is the point evaluation of F_x
  - The n samples represented by a $n \times D$ matrix Φ
- Two Chapters
  - 1970's: Functional Analysts show that there exists ($n \asymp m \log D$) samples which identify every sparse vector Kashin, Gluskin, Johnson, Lindenstrauss
  - 2000's: It is shown that the sampling measurements can be detangled and the sparse vector identified through ℓ_1 minimization: Donoho, Candes, Tao

Remarks on CS

- Optimal matrices are random, e.g. a $n \times D$ Bernoulli matrix with ± 1 entries with sign selected by coin flips
  - However, there is no easy check whether a given matrix is optimal (sufficient condition is RIP)
  - Optimal Algorithms for Sparse: Random Sampling followed by ℓ_1 minimization decoding - (can also use Orthogonal Matching Pursuit to decode)
  - Optimality proved by Gelfand widths
- Major question: optimal deterministic constructions
  - Projective geometry, number theory, combinatorics:Bourgain+, Calderbank+, D.
- Compressed Sensing generalizes to infinite dimensional settings: Adcock-Hansen + and compressible signals Cohen-Dahmen-D

Sparsity/Compressibility in practice

- Adcock-Bastounis-Hansen-Roman call into question standard sparsity
- How can one be sure in practice?
- Situation is better in PDEs where one can prove regularity of solution
  - Return to the solution map F for parametric elliptic problems
  - Cohen-D-Schwab If $(\|\psi_j\|_{L_\infty(\Omega)}) \in \ell_p$, $p < 1$ then

    $$F(y) = \sum_\nu u_\nu y^\nu$$

    - $(\|u_\nu\|_X) \in \ell_p$
    - $$\sup_{y \in [0,1]^{I\!N}} \|F(y) - \sum_{\nu \in \Lambda} u_\nu y^\nu\|_X \leq C n^{-1/p-1}, \#(\Lambda) \leq n$$
    - Compressibility proven

Fourier +

- Suppose we wish to recover a sparse Fourier polynomial $F = \sum_{j \in \Lambda} c_j e^{ijx}$, $\Lambda \subset \Gamma$, $\#(\Gamma) = D$
- Take x_i, $i = 1, \dots, n$ random with respect to uniform measure
- Long history: Candes, Tao, Vershynin, Rudelson, Rauhut,...
- Best result: Sufficient to have $n \geq Cs(\log s)^2(\log D)$ measurements Chkifa, Webster,...
- Extends to general orthogonal systems ψ_j with $\|\psi_j\|_{L_\infty(\Omega)} \leq M$
- Extends to HD with some care
- Does not extend to wavelets as such (shrinking support)

Variable Reduction Model Classes

- A common assumption in treating high dimensional problems is that not all variables are equally important
- Algorithms identify the important variables and use approximation techniques for low dimension once found
- Simplest example: $F(x_1, \ldots, x_D) = g(x_{j_1}, \ldots, x_{j_d})$, where $g \in C^s$ with s, $j_1, \ldots, j_d$ and d not known.
- The point clouds in Query Algorithms have two tasks:
  - Determine change coordinates $j_1, \ldots, j_d$
  - Give a uniform grid with spacing $h \asymp n^{-1/d}$ for each d dimensional space spanned by a possible $j_1, \ldots, j_d$
- Such point clouds are constructed using Hashing

Hashing

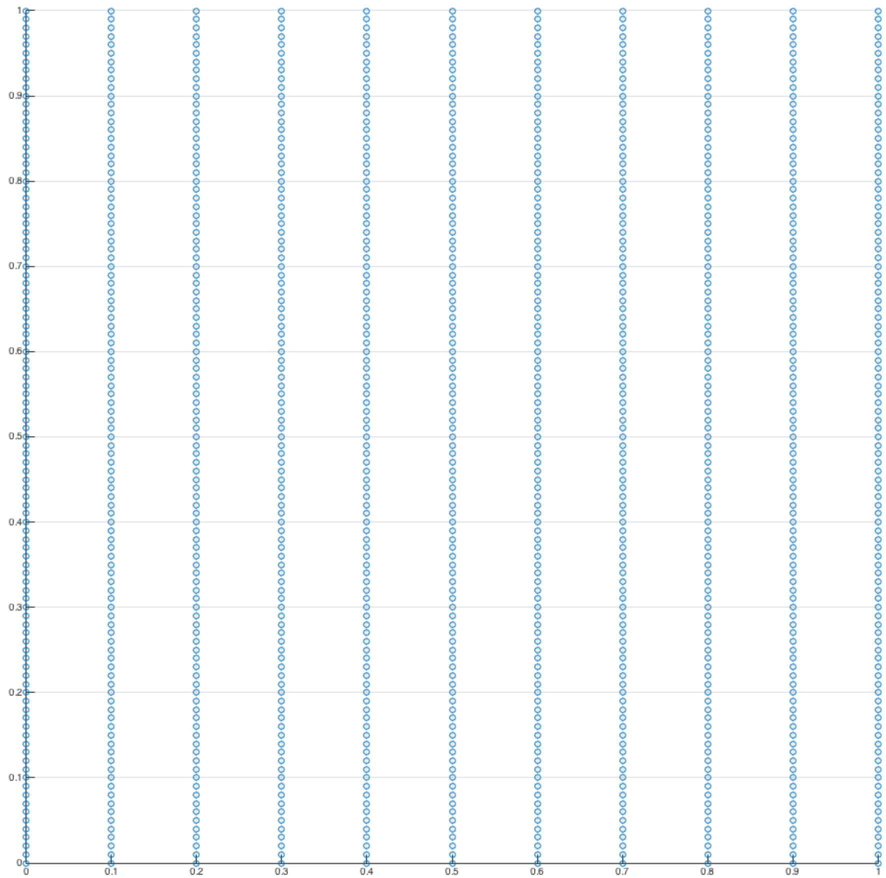
- We create a family $\mathcal{A}$ of partitions $A = (A_1, \ldots, A_d)$ of $\{1, \ldots, D\}$
  - Given any $j_1, \ldots, j_d$ there is one $A \in \mathcal{A}$ such that each j_i appears in exactly one set A_k of A - when $d = 2$ just take binary partitions
- With Hashing we can construct $\mathcal{P} \subset [0,1]^D$ such that

  Projection Property: For any d dimensional coordinate subspace V of $I\!R^D$, the projection of $\mathcal{P}$ onto $V \cap [0,1]^D$ gives a uniform grid of spacing h
- With Hashing we can create point clouds $\mathcal{A}$ to determine the change coordinates $j_1, \ldots, j_d$
- Certifiable Optimal Algorithm (D-Petrova-Wojtaszczyk) With n queries we can appproximate F to accuracy $C(d,s)(\log D)n^{-s/d}$

More General Anisotropy

- Anisotropic smoothness spaces: $\bar{s} = (s_1, \dots, s_d)$
- The space $W^{\bar{s}}(L_p)$ consist of all $F \in L_p[0,1]^D$ such that $\|D_{x_j}^{s_j} F\|_{L_p} \le 1$, $j = 1, \dots, D$
  - $S := g(\bar{s}) := \{\frac{1}{s_1} + \cdots + \frac{1}{s_D}\}^{-1}$
  - With n queries, we can recover all functions in $W^{\bar{s}}(L_p)$ in the $L_p[0,1]^D)$ norm with accuracy Cn^{-S}
  - For example, if $p = \infty$ it is enough to take point sample on an anisotropic grid
  - Example $\bar{s} = (2,1)$, $S = 2/3$

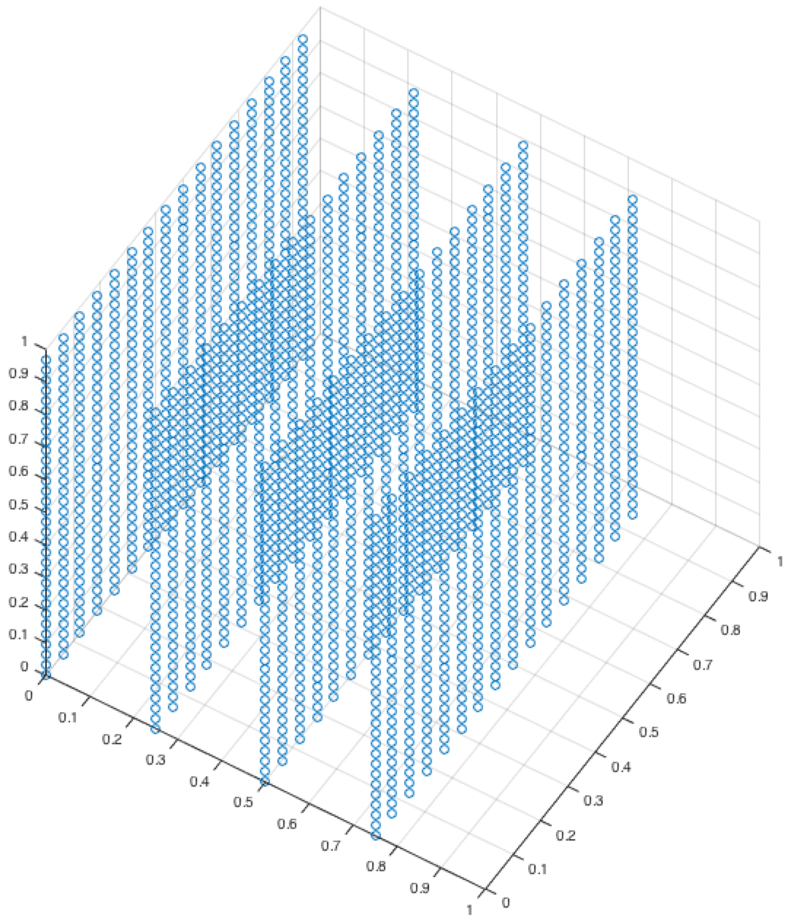
$$\bar{s} = (2, 1), D = 2$$



General Anisotropic Spaces

- For $S > 0$, $W^S(L_p) := \bigcup_{g(\bar{s})=S} W^{\bar{s}}(L_p)$
  - Do not know the coordinates of anisotropy
- Where to query to optimally recover $W^S(L_p[0,1]^D)$?
  - In the case $p = \infty$ one query set is sampling on sparse grids?
    - Given $n = 2^k$, write $k = k_1 + k_2 + \cdots + k_D$
    - Take the with spacing $2^{-k_1} \times \cdots \times 2^{-k_D}$
    - Sparse Grid: union $\asymp n(logn)^{D-1}$ points
  - Sparse grid sampling gives error $C(D,s)(\frac{\log n)^{D-1}}{n})^S$ for the above spaces $W^S(L_\infty[0,1]^D)$
  - Not known if this is optimal (a question of logarithms)
  - Note that the case there are d nonzero s_i and all equal s we arrive at our original example of functions

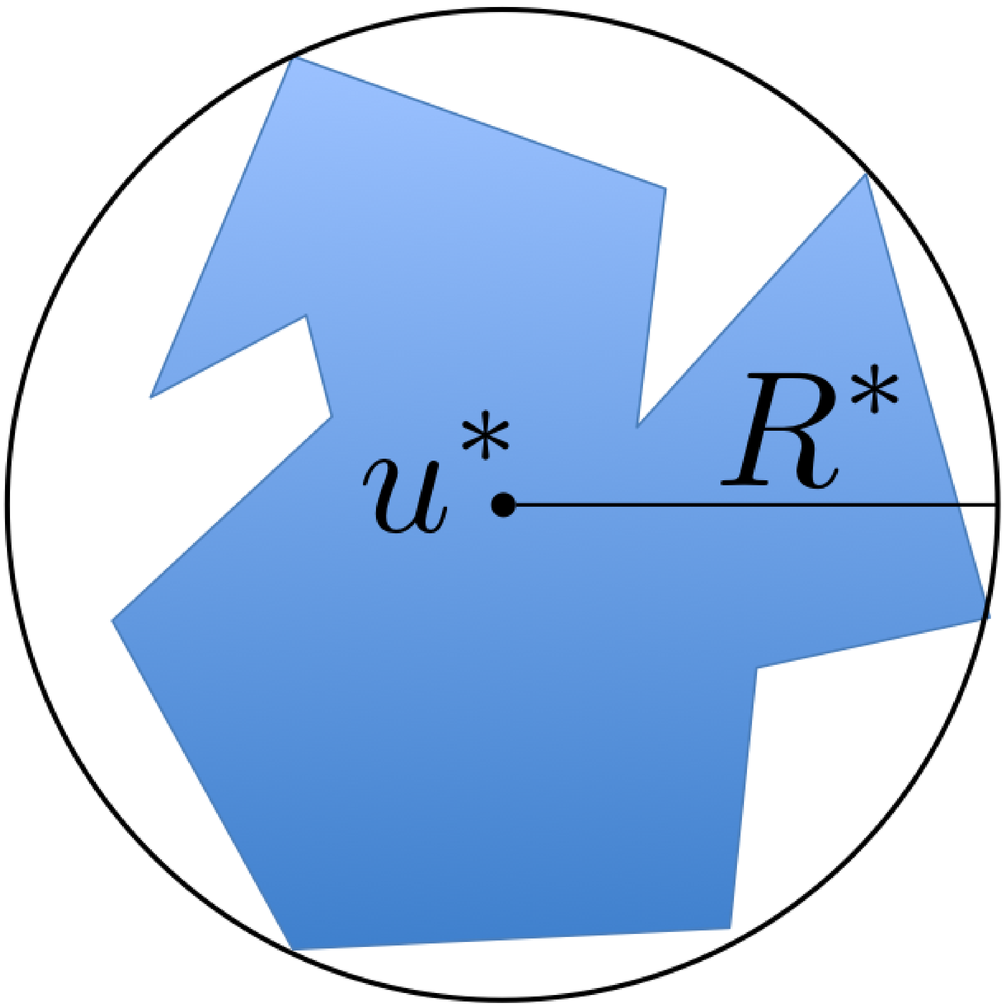
Sparse Grids: $4 \times 16 \times 32$ Grid



Data Assimilation

- The data $w = (w_1, \dots, w_n)$ comes from linear functionals applied to F: $w_i := \ell_i(F),\ i = 1, \dots, n$
- A Data Assimilation Algorithm is a mapping $A_n : w \mapsto A_n(w) \in Y$
- Let $\mathcal{K}_w := \{g \in \mathcal{K} : \ell_i(g) = w_i,\ i = 1, \dots, n\}$
  - Each $g \in \mathcal{K}_w$ is given the same approximant $A_n(w)$
  - Let $B(y(w), R(\mathcal{K}_w))$ be the smallest ball that contains $\mathcal{K}_w$ - the Chebyshev ball
  - The best algorithm: $A_n : w \mapsto y(w)$
    - Best algorithm has distortion $R(w) = R(\mathcal{K}_w)$
- Computing $R(\mathcal{K}_w)$ tells us the best performance
- Finding $y(w)$ is a best algorithm
- Numerically finding an element $\hat{y}(w)$ in $B(y(w), R(\mathcal{K}_w))$ is a near best algorithm

Chebyshev Ball Graphic



Data Assimilation

- Data Assimilation is a problem of Optimal Recovery
- Optimal Recovery results are usually for classical settings (smoothness spaces) and little is known in HD
- I want to put forward one general useful principle
- Usually hard to find Chebyshev ball for model class
- Especially in HD since we do not always have a good analytic description of the model class
- Frequently, all we know is that $\mathcal{K}$ can be approximated by a certain sequence V_m, $m = 1, 2, \ldots$ of m dimensional spaces to accuracy ϵ_m
- This leads us to replace $\mathcal{K}$ by the somewhat larger set $\bar{\mathcal{K}} := \mathcal{K}(\epsilon_m, V_m) := \{f : \ \mathrm{dist}(f, V_m) \le \epsilon_m\}$
- We can determine optimal data assimilation for $\bar{\mathcal{K}}$

Assimilation for Approximation Sets

- To keep life simple assume $Y = \mathcal{H}$ is a Hilbert space
- Maday-Patera-Penn-Yano give the following algorithm A
  - Given $w = (w_1, \ldots, w_n)$, consider $\mathcal{H}_w := \{u \in \mathcal{H} : \ell_j(u) = w_j,\ j = 1, \ldots, n\}$
  - Determine (by least squares) $\bar{u}(w) \in \mathcal{H}_w$, $\bar{v}(w) \in V_m$ closest: $\|\bar{u}(w) - \bar{v}(w)\| = \mathrm{dist}(\mathcal{H}_w, V_m)$
  - Define $A(w) := \bar{u}(w)$
  - Their algorithm is optimal (Binev-Cohen-Dahmen-D-Petrova-Wojtaszczyk)

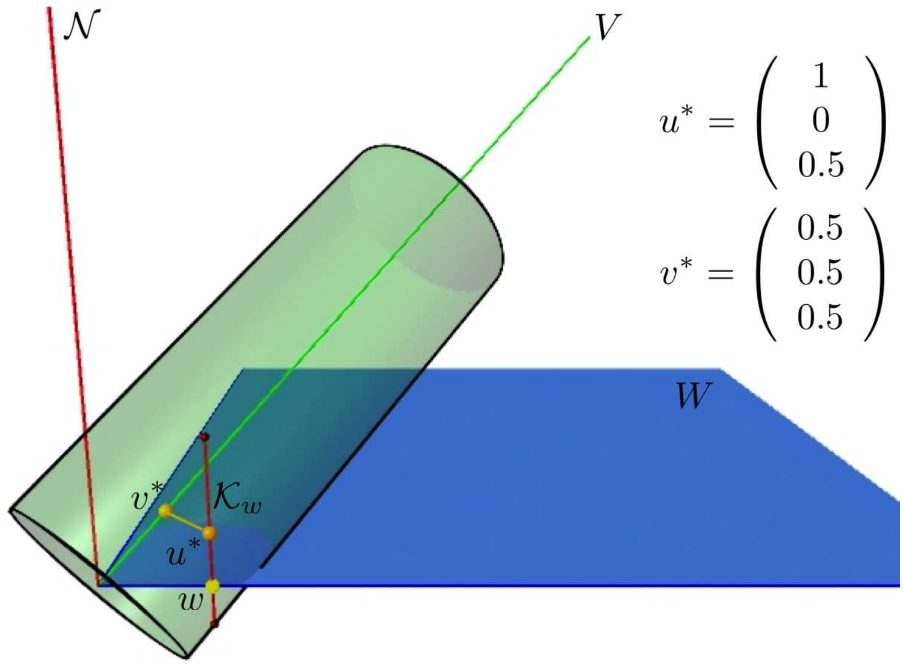
Performance of Algorithm

- The interesting point about this setting is one can determine a priori the performance of the algorithm
- Let $\mathcal{N} \subset \mathcal{H}$ be the null space of the measurements
- Define $\mu(V_m, \mathcal{N}) := \sup_{\eta \in \mathcal{N}} \frac{\|\eta\|}{\mathrm{dist}(\eta, V)}$
- Performance:

$$R(\mathcal{H}_w)^2 = \mu(V_m, \mathcal{N})^2 \{\epsilon_m^2 - \|\bar{u}(w) - \bar{v}(w)\|_{\mathcal{H}}^2\}$$

- Note $\mu(V_m, \mathcal{N}) = \infty$ if $n < m$
- Similar results hold for general Banach spaces - D-Petrova-Wojtaszczyk

Hilbert space geometry



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

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

Computing μ

- The quantity $\mu(V_m, \mathcal{N})$ can usually be computed
- In the Hilbert space case it is the reciprocal of the angle between $\mathcal{N}$ and V_m computed from singular values of a certain cross Grammian
- Here is another interesting example
  - $Y = C(\Omega)$, $\ell_j(f) = f(x_j)$ with $x_j \in \Omega$, $j = 1, \dots, n$
  - $\mu(V_m, \mathcal{N}) = \sup_{v \in V_m} \frac{\|v\|_{C(\Omega)}}{\max_{1 \le i \le n} |v(x_i)|}$
  - So we recover f with these measurements to accuracy $\mu(V_m, \mathcal{N}) \mathrm{dist}(f, V_m)$
  - Data $f(x_i)$, $x_i = i/n$, $i = 1, \dots, n$ $f \in C[0,1]$, $V_m = \mathcal{P}_{m-1}$ $\mu(V_m, W) \ge C\lambda^n$, $\lambda > 1$, $\mu(V_{\sqrt{n}}, W) \le C$
  - Two Errors: $\lambda^n E_n(f),\ m = n, \quad CE_{\sqrt{n}}(f),\ m = \sqrt{n}$
    do not interpolate!

What Time Prevented

- Tensors
  - the Tensor zoo
  - Concentrated on algebraic aspects not query/assimilation Hackbusch, Grasedyck
  - some impressive applications Griebel, Schneider, ...
  - Sparse grids, Smolyak representation, discrepancy theory, quasi-Monte Carlo
- High dimensional polynomial interpolation/approximation
  - Lower sets, Leja points, Smolyak multi-scale
- Stochastic setting
  - Outstanding results for sparsity with undersampling
  - Donoho, Candes, Wainwright, Buhlmann, ...
- More detail in the above settings