Manifold learning with sparse grid methods

Michael Griebel
joint work with Bastian Bohn
INS, Universität Bonn

1. Data and the curse of dimension
2. Principal manifold learning
   2.1. Sparse grids
   2.2. Regression
3. Concluding remarks
High-dimensional data

- **Data** show up in many different areas

Biomedicine

- SNP-interaction  
  (UBonn-MedBio)

Meteorology

- weather radar  
  (DFG-TR32)

Finance

- basket options  
  (ESF-AMaMeF)

Engineering

- crash dynamics  
  (BMBF-Simdata-NL)

- Most often, they are **high-dimensional**, i.e. they can be considered as (a set of) points in huge-dimensional space
Aims

• Typical tasks
  – Density estimation
  – Classification ⇒ Find hidden structures and patterns in the data
  – Regression

• Unified framework via conditional density estimation, the Cameron–Martin theory of stochastic processes and the maximum a posteriori method for Gaussian processes [Bogachev98, Hegland07, G.+Hegland10]

• We may approach these tasks as scattered data approximation problems which is well understood [Wendland04]
  – derive a model which approximates the data points
  – evaluate this model in new data points ⇒ prediction

• So, where is the difficulty ?
Curse of dimension

- \( f : \Omega^{(d)} \rightarrow \mathbb{R}, \; f \in H^r(\Omega^{(d)}), \; r \) isotropic Sobolev smoothness

- Bellmann ’61: curse of dimension
  \[ \| f - f_N \|_{H^s} = C(d) \cdot N^{-r/d} \quad \| f \|_{H^{s+r}} = O(N^{-r/d}) \]

- Find situations where curse can be broken? The curse is there for \( H^r \), so we have to change the setting

- Restrict isotropic smoothness to \( r = O(d) \). Then
  \[ \| f - f_N \| = O(N^{-cd/d}) = O(N^{-c}) \]

  - First example: \( \nabla f \in FL_1 \) where \( FL_1 \) is class of functions with Fourier transform in \( L_1 \). Then, \( \| f - f_N \| = O(N^{-1/2}) \) [Barron93]
  
  - Radial basis schemes, Gaussian bump algebra [Meyer92] corresponds to ball in Besov space [Niyogi+Girosi98] \( B_{1,1}^d(\mathbb{R}^d) \Rightarrow r \geq d \)
  
  - Sobolev embedding: \( r > d/2 \Rightarrow \text{point evaluation continuous, RKHS} \)
  
  - Analyticity helps, smoothness is at least proportional to \( d \), exponential convergence rate compensates the curse
  
  - Stochastics helps, error in expectation or in probability, concentration of measure, stochastic sampling techniques, MC
Curse of dimension

• Restrict to a certain **mixed** Sobolev smoothness,
  – i.e. to bounded mixed $r$-th derivatives
  – to weighted mixed spaces [Sloan+Wozniakowski98]
  – to anisotropic mixed spaces [Temlyakov93]
  – to mixed Besov spaces [Dung+Temlyakov+Ullrich18]

  – Then, for suited and properly adapted **sparse grid/hyperbolic cross** approximations [Korobov59, Babenko60, Smolyak63], the curse appears only in logarithmic terms or completely **disappears** (but still may be present in the order constants)

  – Note that **mixed spaces** depend directly on the **coordinate axes** and involve axiparallel smoothness

• In any case: **some** smoothness changes with $d$ or the **importance** of coordinates **decays** successively
Curse of dimension

- Restrict to a **lower dimensional** (nonlinear) smooth manifold
  - Success of most machine learning algorithms: High-dimensional problems and x-data often live on a manifold with relatively small intrinsic dimension. Unfortunately, neither this manifold nor its (nonlinear) coordinate system is usually known a-priori.

  - Reconstruct it approximately from the data and provide a generative mapping from it to x-space.
  - Also breaks the **curse of dimension**: Algorithms that work in the manifold coordinate system involve cost that only depend (exponentially) on the small intrinsic dimension.

  - The manifold, i.e. the best coordinate system for a problem, is in general not spanned by a collection of linear coordinates (PCA).

=> nonlinear mapping
Low-dimensional non-linear manifold

- A simple 3D-example is here:
The problem

- **Given**: Data points \( \{ x_1, \ldots, x_N \} \subset X \) drawn iid from an unknown underlying probability distribution \( p(x), \ x \in X \) (\( = \mathbb{R}^n \))
- **Define** index set \( T (= \mathbb{R}^d) \), map \( f : T \rightarrow X \), class \( F \) of maps
- **Aim**: Find \( f \) such that
\[
R(f) = \int \min_{t \in T} c(x, f(t)) \ dp(x)
\]
is minimized in \( F \):
\[
\arg \min_{f \in F} R(f)
\]
- Loss function \( c(x, f(t)) \) determines error of reconstruction
- We stick to simple least squares regression
\[
c(x, f(t)) = \| x - f(t) \|^2_2 \]
The problem

• We have many dimension reduction techniques [Lee+Verleysen07] that realize a down-projection $P : X \rightarrow T$
  – local linear embedding, curvilinear component analysis, Laplacian eigenmaps, diffusion maps, ...
• But we want a generative approach which gives mappings in both ways
  – Projection $P : X \rightarrow T$
  – Generative map $f : T \rightarrow X$
• Why?
  – Interpolation on manifold
  – Prediction of parametric surrogate models in new parameter values
  – Quantification of error of embedding by norm in $X$-space and not in $T$-space
• PCA, GTM, PML, generative auto-encoders
The problem

- **Unsolvable** since \( p(x) \) is unknown
- Replace \( p(x) \) by empirical density

\[
p_N(x) := \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i)
\]

- Minimize empirical quantization error

\[
\int_{X} \min_{t \in T} \| x - f(t) \|^2 dp(x) \approx \frac{1}{N} \sum_{i=1}^{N} \min_{t \in T} \| x_i - f(t) \|^2 =: R_{emp}(f)
\]

on the set of all maps \( f \in F \)

\[
\arg \min_{f \in F} R_{emp}(f)
\]
Non-linear maps

- **Principal curves and manifolds** [Hastie84, Hastie+Stützle89, Smola01]
  \[ T := [0,1]^d, \quad f : t \rightarrow f(t), \quad f \in F \] class of continuous \( \mathcal{R}^d \)-valued functions

  \[
  R_{\text{emp}}(f) = \frac{1}{N} \sum_{i=1}^{N} \min_{t \in [0,1]^d} \| x_i - f(t) \|_2^2
  \]

- Find for each \( x_i \) the minimum
  This gives \( t_i \) and
  \[
  R_{\text{emp}}(f) = \min_{t_1, \ldots, t_N} \frac{1}{N} \sum_{i=1}^{N} \| x_i - f(t_i) \|_2^2
  \]

- **Nonlinear model**

- **Ill-posed problem**, unless \( F \) is compact
Regularization and expansion

- **Penalization:**
  \[ R_{\text{reg}}(f) = R_{\text{emp}}(f) + \gamma S(f) \]
  - error term
  - regularization parameter, balances both terms
  - regularization term, enforces certain smoothness on \( f \)

- **Convex, nonnegative, \( G \) (pseudo)-diffop**

- **Expand** \( f \) in terms of a basis \( \{\phi_i(t)\} \) of \( F \)
  \[ f(t) \approx f_M(t, \alpha) = \sum_{j=1}^{M} \alpha_j \phi_j(t) \]

- **Find**
  \[
  \arg\min_{t_1, \ldots, t_N \in T, \alpha_1, \ldots, \alpha_M \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \| x_i - f_M(t_i, \alpha) \|_2^2 + \gamma \| Gf_M(t, \alpha) \|_0^2
  \]

- **Non-linear minimization problem**
EM algorithm

• Chose initial values (f.e. as result of PCA) and iterate:

• Projection step: keep \( \{\alpha_j\} \) fix, minimize w.r.t. \( \{t_i\} \)

\[
\min_{t_i} \left\| x_i - f_M(t_i, \alpha) \right\|_2^2 \quad i = 1, \ldots, N
\]
downhill simplex, Max-Powell

• Adaption step: keep \( \{t_i\} \) fix, minimize w.r.t. \( \{\alpha_j\} \)

\[
\min_{\alpha_1, \ldots, \alpha_M \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^N \left\| x_i - f_M(t_i; \alpha) \right\|_2^2 + \gamma \left\| Gf_M(t, \alpha) \right\|_0
\]

– This is just a vector-valued regression problem with data \( (x_i, t_i) \)
– Differentiation w.r.t. \( \{\alpha_i\} \) results in the linear system

\[
(B^T B + \frac{M\gamma}{2} C) \alpha = B^T x
\]

with \( N \times M \) matrix \( B_{ij} = \phi_j(t_i) \)
and \( M \times M \) matrix \( C_{ij} = \int G\phi_i \cdot G\phi_j dt \)
Our approach

- We use tensor product hierarchical Faber basis/prewavelets
- For regularization term:
  - bounded mixed derivatives \( S(f) = \| f \|_{H_{mix}^1}^2 \) \( C \approx \) product of 1d Laplacian without the \( \| f \|_{L_2}^2 \)-term
  - Relates to length of curve, area of surface, volume of manifold,…

- How to choose the expansion?
  - Uniform full grid:
    \[ f(t) \approx f_k(t) = \sum_{|l|_{\infty} < k} \sum_i \alpha_{l,i} \phi_{1,i}(t) \]
    \[ \text{dof: } O(2^{kd}) \]
    curse of dimension
  - Sparse grid
    \[ f(t) \approx f_k(t) = \sum_{|l|_1 < k} \sum_i \alpha_{l,i} \phi_{1,i}(t) \]
    \[ \text{dof: } M = O(k^{d-1}2^k) \]
    breaks curse of dimension at least somewhat
Regular sparse grids
Sparse grids

- **Cost** for regular **sparse grid** method
  - Projection step: $O(M \cdot N)$ global optimization, $O(k^{d-1} \cdot N)$ local optim.
  - Adaption step: set up of matrix $C$: $O(M)$
    set up of matrix $B$: $O(k^{d-1} \cdot N)$
    solution: $O(M^\beta)$, $\beta = 1$ (multiscale solver), $\beta \approx 2$ (PCG)

=> Scales **linearly** in #data and (nearly) **linearly** in #parameters

- Allows furthermore for
  - **generalized** sparse grids
  - **dimension-adaptive** sparse grids
  - **locally adaptive** sparse grids
Boundary basis

- Important extension: \( l \in \mathbb{N}_{-1} := \mathbb{N}_0 \cup \{-1\} \)
  - The 1D basis is extended to the boundary by constant and linear (and not two linears), two more levels

- After tensorization:
  Trivial embedding into high-dimensional space due to constant

- Close relation to ANOVA expansion

- Analogously for global polynomial basis expansion
The dimension-adaptive algorithm

• Build index set adaptively, greedy-type methods

• Original algorithm for quadrature: [G+Gerstner03]
  – Successively enlarges/adapts the index set $\mathcal{S}^{act}$ according to $bcr$-indicator $\varepsilon(l)$:
    If $\varepsilon(l)$ larger than global threshold $E$, then refine
  – Maintains downward-closedness
  – $d$ successor indices, not $2^d - 1$

• Modifications:
  – Start with regular sparse grid on level 2
  – Compression and refinement steps [Feuersänger10, Bohn+G12]
  – Boundary with constant and linear, $l = -1,0,1...$
  – Compression: For all $l \in \mathcal{S}^{act}$ check if $\varepsilon(k) \leq E$ for all $k : k \geq l$ and $k \in \mathcal{S}^{act}$
    if yes, remove all these $k$ from $\mathcal{S}^{act}$
  – Refinement: For all $l \in \mathcal{S}^{act}$ check if $\varepsilon(l) \geq E$
    if yes, add all $k : k \leq l + e_j, j = 1,...,d$ with $l_j \neq -1$
Example

- Evolution of the algorithm:

  - As any adaptive heuristics: may **terminate** too early

  ![Example](image)
Sparse grid manifold learning

• Recall: We have an iterative EM-type algorithm with
  – Projection step
  – Adaption step

• We now use sparse grids therein [Bohn+Garcke+Griebel16]
  – Regular sparse grid method
  – Dimension-adaptive sparse grid method

• Generalization to vector-valued functions
  – Adaptive method for each component separately
  – Union of active sets for all components
  – Modified error indicator [Bohn+Garcke+Griebel16]
Two lines \( n = 2, \; d = 1, \; S(f) = \| \nabla f \|_0^2 \)

fixed length of curve [Kegl00]

the data

start with 2nd eigenvector of PCA

level 2

level 3

level 4

level 5
Sensitivity on starting values

PCA

nonlinear PML approach

start with 1th eigenvalue of PCA

start with 2nd eigenvalue of PCA

projection of data points onto $T$
3/4 circle \( n = 2, \ d = 1, \ S(f) = \| \nabla f \|_0^2 \)

Start value by PCA, Solution direct on level 5

Multilevel approach: Start value by PCA on coarse level, Successive refinement up to level 5

- Again: Sensitivity on starting values
- Multilevel approach helps
- It is a non-linear method after all
Convergence for helix problem

Use successively more sample points, more grid points and successively smaller values of $\gamma$

$n = 3, \ d = 1, \ S(f) = \| \nabla f \|_0^2$
Oil flow data: Visualization and clustering

1000 samples, 3 classes, \( n = 12, \ d = 2, \ k = 6, \ \gamma = 10^{-2} \)

[Bishop+Svensen+Williams98]

PCA fails completely
no separation of classes

regular sparse grid PML works well
separates the classes much better
Oil flow data

Regular sparse grid PML $n = 12, \ d = 3, \ S(f) = \| \nabla f \|_0^2 \ k = 5, \ \gamma = 10^{-2}$

Again: Sparse grid manifold approach clearly separates the classes

Separation and clustering even clearer and more compact in 3D
**1D-kink**

- **Kink-shaped 1D manifold** \( x(t) := (x_1, x_2, x_3)^T = (t, |t|, t)^T \) on \( T = [-1,1] \)
- **N samples perturbed by** \( \mathcal{N}(0, 0.05 \cdot I_3) \) \( n = 3, \ d = 1 \)
- **Start with regular sparse grid on level 2 and refine for** \( E = 10^{-2} \)

\[
N = 10, \ \gamma = 10^{-3} \quad \text{and} \quad N = 100
\]

- **Overestimated dimension**: Start dim-adaptive method with \( d = 2 \)

Adaptivity (compress) \( \Rightarrow \) intrinsic dimension reduction
S-shaped manifold

\[(t_1, t_2) \in [-\frac{3}{2} \pi, \frac{3}{2} \pi] \times [0,5] \]
\[x(t_1, t_2) := (x_1, x_2, x_3)^T = (\sin(t_1), t_2, \text{sign}(t_1)(\cos(t_1) - 1))^T\]

- **Input data:** Draw 1000 points in \(T\), iid, uniformly, apply \(x(t_1, t_2)\) and add 3D \(\mathcal{N}(0, 0.01 \cdot I_3)\) Gaussian noise

- **Dimension-adaptive algorithm**

\[(E, \gamma) = (0.02, 0.1 \times 10^{-4})\]

needs only 35 points whereas the regular sparse grid needs 339 points

Only sparse grid points on the **boundary** needed

learned manifold (after one final compression step)
Car crash analysis

- Automotive industrie: **FE-simulations** of car crash for new product development with the aim of passenger safety
- Reduces the huge **costs** of real life car crash experiments

- Design process: engineer changes **parameters** like plate thickness or material properties for each new FE-run
- Each simulation is a **point** in huge-dimensional space
- Run-time per simulation $\frac{1}{2}$ day $\Rightarrow$ number $N$ of simulations is quite **small**

- Same mesh configuration and same physical laws $\Rightarrow$ variation of parameters form a nonlinear, low-dimensional structure/manifold in high-dimensional simulation space
Car crash analysis

- Project SIMDATA-NL in BMBF support program
- Example: Frontal crash simulation of Ford Taurus

- Involves 900,000 FE-nodes over 300 time steps, LS-DYNA
- 19 parameters (plate thickness of 19 parts = 15 beams + 4 further attached parts) were varied by up to 5%
Car crash analysis

- 264 crash simulations for training, 10 for test/evaluation
- **Displacement data:** FEM($t=150$) - FEM($t=0$)
  - At time step $t=0$, simulation **starts**, same car speed for all simulations
  - At time step $t=150$ the crash **impact** took already place but car is not yet bouncing back from obstacle

For each simulation:
- FEM-model: $n \sim 3*900.000$ dof in space
- Analyzed substructure consists of **15 parts** with dimensions ranging from 3*934 to 3*4675

- Analysis for each subpart separately and putting together
- **Precomputation:** Dimension reduction by lossless PCA
  \[ \implies n = 264 \]

- Run dimension-adaptive sparse grid method for $d = 1,2,3,..$ and compare to corresponding simple PCA [Bohn+Garcke+G16]
Car crash analysis

PCA

dimension-adaptive PML \((\varepsilon, \gamma) = (10^{-2}, 10^{-3})\)

\[
d = 1
\]

Shown: Error per node, averaged over the 10 test cases, color-coded blue=0mm, red>50mm

\[
d = 2
\]

PML has substantially less error due to its non-linearity than PCA

\[
d = 3
\]
Concluding remarks

• Dimension-adaptive sparse grids for manifold learning
  – Cost linear in data and \( \sim \)linear in dof in contrast to kernel methods
  – Captures nonlinear effects in contrast to PCA
  – Smaller intrinsic dimension

• Intrinsic dimension
  – Must be chosen a-priori for regular sparse grids
  – Can (in principle) determined automatically for adaptive sparse grids
    • Take \( \dim(T) = \dim(X) \) and run dim-adaptive procedure
    • Whitney’s and Taken’s embedding theorem: Take even \( \dim(T) = 2\dim(X) + 1 \) ?
    • Cover’s theorem on linear separability with high probability. Take even \( \dim(T) = N - 1 \) ?

• Loss function
  – Was here \( L^2 \)-norm, least square regression
  – Cross entropy leads to generative topographic mapping [G+Hullmann14]

• Function on manifold \( f(x) \rightarrow f(x(t)) =: g(t) \) lower-dim function

• Concatenation of functions
  – Kernels of kernels, new representer theorem for deep kernel learning [Bohn+G+Rieger17]