Low-rank and sparse methods for high-dimensional approximation and model order reduction

—

Lecture 1

High-dimensional approximation

Outline

1. High dimensional problems
2. High-dimensional approximation and the curse of dimensionality
3. How to beat the curse of dimensionality ?

Outline

1. High dimensional problems
2. High-dimensional approximation and the curse of dimensionality
3. How to beat the curse of dimensionality ?

High-dimensional problems in physics

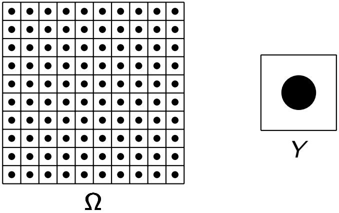
- Navier-Stokes equation in a 3-dimensional domain Ω

$$u(x_1, x_2, x_3, t)$$

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla p + \nu \Delta u$$

- Multiscale problems

$$x \equiv (x, y), \quad u(x,t) \equiv u(x,y,t), \quad x \in \Omega, \; y \in Y$$



High-dimensional problems in physics

- Schrodinger equation

$$\Psi(x_1, \dots, x_d, t)$$

$$i\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar}{2\mu} \Delta \Psi + V\Psi$$

- Boltzmann equation

$$p(x_1, \dots, x_d, t)$$

$$\frac{\partial p}{\partial t} + \sum_{i=1}^{d} v_i \frac{\partial p}{\partial x_i} = H(p, p)$$

- Fokker-Planck equation

$$p(x_1, \dots, x_d, t)$$

$$\frac{\partial p}{\partial t} + \sum_{i=1}^{d} \frac{\partial}{\partial x_i}(a_i p) - \frac{1}{2} \sum_{i,j=1}^{d} \frac{\partial^2}{\partial x_i x_j}(b_{ij} p) = 0$$

- Master equation

$$P(x_1, \dots, x_d, t), \quad (x_1, \dots, x_d) \in \mathcal{X} = \{1, \dots, N\}^d$$

$$\frac{\partial P}{\partial t}(x, t) = \sum_{y \in \mathcal{X}} A(x, y) P(y, t)$$

High-dimensional problems in stochastic analysis

Stochastic differential equations (SDEs)

$$dX_t = a(X_t, t)dt + \sigma(X_t, t)dW_t, \quad X_t \in \mathbb{R}^d$$

- Fokker-Planck equation for probability density function $p(x_1, \dots, x_d, t)$ of X_t

$$\frac{\partial p}{\partial t} = \mathcal{L}p = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}(a_i p) + \frac{1}{2}\sum_{i,j=1}^{d} \frac{\partial^2}{\partial x_i x_j}((\sigma\sigma^T)_{ij} p)$$

- Feynman-Kac formula for

$$u(x,t) = \mathbb{E}^{X_t = x}\left(\int_t^T e^{\int_t^s r(X_r, r)dr} f(X_s, s)ds\right)$$

  yields a high-dimensional PDE

$$\partial_t u + \mathcal{L}^* u + ru + f = 0 \quad \text{in } \mathbb{R}^d \times (0, T), \quad u(x, T) = 0$$

- Functional approach to SDEs using a parametrization of the noise

$$W_t = \sum_{i=1}^{\infty} \xi_i \varphi_i(t), \quad \xi_i \sim N(0, I),$$

$$X_t(\omega) \equiv u(t, \xi_1(\omega), \xi_2(\omega), \dots)$$

High-dimensional problems in uncertainty quantification

Parameter-dependent models

$$\mathcal{M}(u(X); X) = 0$$

where $X = (X_1, \dots, X_d)$ are random variables.

- Forward problem: evaluation of statistics, probability of events, sensitivity indices...

$$\mathbb{E}(f(u(X))) = \int_{\mathbb{R}^d} f(u(x_1, \dots, x_d))p(x_1, \dots, x_d)dx_1 \dots dx_d$$

- Inverse problem: from (partial) observations of u, estimate the density of X

$$p(x_1, \dots, x_d)$$

- Meta-models: approximation of the high-dimensional function

$$u(x_1, \dots, x_d)$$

Outline

1. High dimensional problems
2. High-dimensional approximation and the curse of dimensionality
3. How to beat the curse of dimensionality ?

High-dimensional approximation

The goal of approximation is to replace a function

$$u(x_1, \dots, x_d)$$

by a simpler function (easy to evaluate) depending on a few parameters.

For a certain subset of functions X_n described by n parameters (or $O(n)$ parameters), the error of best approximation of u by elements of X_n is defined by

$$e_n(u) = \inf_{v \in X_n} d(u, v)$$

where d is a distance measuring the quality of an approximation.

A sequence of subsets $(X_n)_{n \geq 1}$ is called an approximation tool. We distinguish linear approximation, where X_n are linear spaces, from nonlinear approximation, where X_n are nonlinear spaces.

High-dimensional approximation

Fundamental problems are

- to determine if and how fast $e_n(u)$ tends to 0 for a certain class of functions and a certain approximation tool,
- to provide algorithms which produce approximations $u_n \in X_n$ such that

  $$\|u - u_n\| \leq C e_n(u)$$

  with C independent of n or $C(n)e_n(u) \to 0$ as $n \to \infty$

The curse of dimensionality

Let consider u in $X = L^p(\mathcal{X})$ with $\mathcal{X} = (0,1)^d$ and the natural distance $d(u,v) = \|u - v\|_{L^p}$ on X. Let X_n be the space of polynomials of partial degree m, with $n = (m+1)^d$ parameters.

If u is in the Sobolev space $W^{k,p}(\mathcal{X})$ for a certain $k \leq m+1$,

$$e_n(u) \leq M n^{-k/d}$$

We observe

- the curse of dimensionality : deterioration of the rate of approximation when d increases. Exponential growth with d of the complexity for reaching a given accuracy.
- the blessing of smoothness : improvement of the rate of approximation when k increases.

We may ask if the curse of dimensionality is due to the particular choice of approximation tool (polynomials) for approximating functions in $W^{k,p}(\mathcal{X})$? We may also ask if the curse of dimensionality is still present if $k = \infty$ (smooth functions) ?

The curse of dimensionality

For a set of functions K in a normed vector space X, the Kolmogorov n-width of K is

$$d_n(K) = \inf_{\dim(X_n)=n} \sup_{u \in K} \inf_{v \in X_n} d(u, v)$$

where the infimum is taken over all linear subspaces of dimension n. $d_n(K)$ measures how well the set of functions K can be approximated by a n-dimensional space. It measures the ideal performance that we can expect from linear approximation methods.

Let $X = L^p(\mathcal{X})$ with $\mathcal{X} = (0,1)^d$.

- For K the unit ball of $W^{k,p}(\mathcal{X})$, we have

$$d_n(K) \sim n^{-k/d}$$

- For $K = \{v \in C^\infty(\mathcal{X}) : \sup_\alpha \|D^\alpha v\|_{L^\infty} < \infty\}$, we have

$$\min\{n : d_n(K) \leq 1/2\} \geq c2^{d/2}$$

  Extra smoothness does not help !

- Similar results are obtained for non-linear widths measuring the ideal performance of nonlinear approximation methods. Nonlinear methods can not help !

Outline

1 High dimensional problems

2 High-dimensional approximation and the curse of dimensionality

3 How to beat the curse of dimensionality ?

How to beat the curse of dimensionality ?

The key is to consider classes of functions with specific low-dimensional structures and to propose approximation formats (models) which exploit these structures (application-dependent).

Approximations are searched in subsets X_n with a number of parameters

$$n = O(d^p)$$

but

- X_n is usually nonlinear, and
- X_n may be non smooth.

This turns approximation problems

$$\min_{v \in X_n} d(u, v)$$

into nonlinear and possibly non smooth optimization problems.

Low-dimensional models for high-dimensional approximation

- Low-order interactions, e.g.
  - No interaction (additive model)

$$u(x_1, \dots, x_d) \approx u_0 + u_1(x_1) + \dots + u_d(x_d)$$

  - First-order interactions

$$u(x_1, \dots, x_d) \approx u_0 + \sum_i u_i(x_i) + \sum_{i \neq j} u_{i,j}(x_i, x_j)$$

- Small number of interactions
  - For a given $\Lambda \subset 2^{\{1,\dots,d\}}$ (set of interaction groups),

$$u(x_1, \dots, x_d) \approx \sum_{\alpha \in \Lambda} u_\alpha(x_\alpha)$$

  - Λ as a parameter

$$u(x_1, \dots, x_d) \approx \sum_{\alpha \in \Lambda} u_\alpha(x_\alpha) \quad \text{with} \quad \#\Lambda = n$$

Low-dimensional models for high-dimensional approximation

- Sparsity relatively to a basis or frame $\{\psi_\alpha\}_{\alpha \in \mathbb{N}}$

$$u(x_1, \ldots, x_d) \approx \sum_{\alpha \in \Lambda} a_\alpha \psi_\alpha(x_1, \ldots, x_d), \quad \#\Lambda = n$$

- Sparsity relatively to a dictionary $\mathcal{D}$

$$u(x_1, \ldots, x_d) \approx \sum_{i=1}^{n} a_i \psi_i(x_1, \ldots, x_d), \quad \psi_i \in \mathcal{D}$$

Low-dimensional models for high-dimensional approximation

- Low rank, e.g.

$$u(x_1, \dots, x_d) \approx u_1(x_1) \dots u_d(x_d)$$

$$u(x_1, \dots, x_d) \approx \sum_{i=1}^{r} u_{1,i}(x_1) \dots u_{d,i}(x_d)$$

$$u(x_1, \dots, x_d) \approx \sum_{i_1=1}^{r_1} \dots \sum_{i_{d-1}=1}^{r_{d-1}} u_{1,i_1}(x_1) u_{i_1,i_2}(x_2) \dots u_{i_{d-1},1}(x_d)$$

$$\dots$$

Low-dimensional models for high-dimensional approximation

Structures possibly discovered with suitable transformations, which may also be considered as additional parameters:

$$u(x_1, \dots, x_d) \approx g(y_1, \dots, y_m), \quad (y_1, \dots, y_m) = h(x_1, \dots, x_d),$$

- One-dimensional model after linear transformation (Generalized Linear Model)

$$u(x_1, \dots, x_d) \approx g(\alpha_1 x_1 + \dots + \alpha_d x_d)$$

- Additive model after linear transformations (Projection Pursuit)

$$u(x_1, \dots, x_d) \approx g_1(y_1) + \dots + g_m(y_m), \quad y_k = \alpha_1^k x_1 + \dots + \alpha_d^k x_d$$

  Neural networks (single hidden layer) as a particular case where functions g_k are equal and fixed.

- ...