Data sampling for surrogate modeling and optimization

Tyler Chang
(and others)

Argonne National Laboratory

ICIAM 2023, Tokyo, Japan
Aug 23, 2023

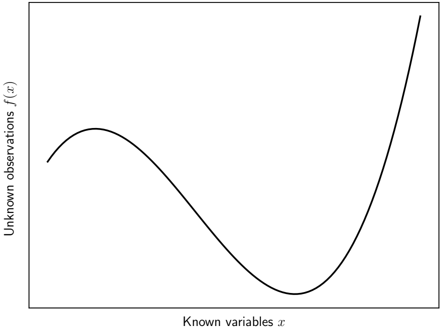
Outlines

Inference problems, the curse of dimensionality, and measure collapse

Modeling for high-dimensional optimization

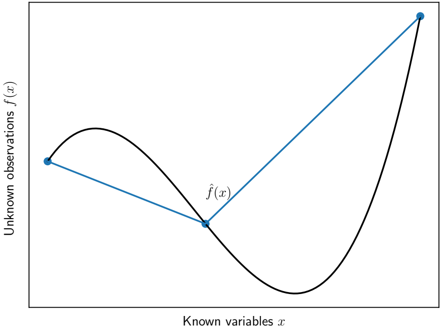
The fundamental machine learning problem

The fundamental machine learning problem



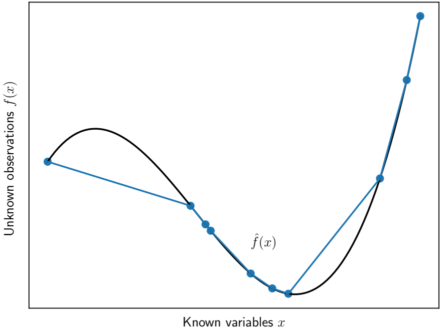
- Want to predict unknown $f(x)$ for observation x

The fundamental machine learning problem



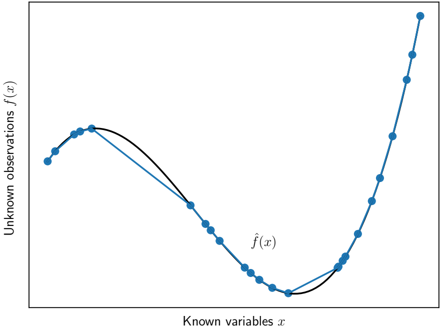
- Want to predict unknown $f(x)$ for observation x
- **ML**: *Learn* approximation $\hat{f} \sim f$ based on *training data* $\mathcal{X}$
- **NA**: fit an interpolant (piecewise-linear) to f on $\mathcal{X}$

The fundamental machine learning problem



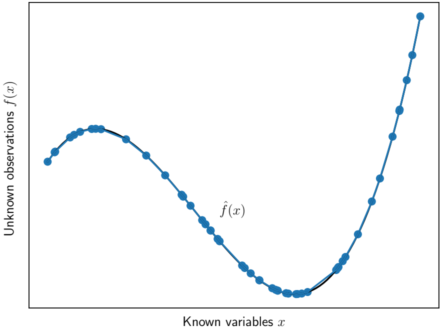
- Want to predict unknown $f(x)$ for observation x
- **ML**: *Learn* approximation $\hat{f} \sim f$ based on *training data* $\mathcal{X}$
- **NA**: fit an interpolant (piecewise-linear) to f on $\mathcal{X}$
- Both cases: more data $\Rightarrow$ better $\hat{f}$

The fundamental machine learning problem



- Want to predict unknown $f(x)$ for observation x
- **ML**: *Learn* approximation $\hat{f} \sim f$ based on *training data* $\mathcal{X}$
- **NA**: fit an interpolant (piecewise-linear) to f on $\mathcal{X}$
- Both cases: more data $\Rightarrow$ better $\hat{f}$
- Real data not perfectly balanced $\Rightarrow$ $\hat{f} \to f$ non-uniformly

The fundamental machine learning problem



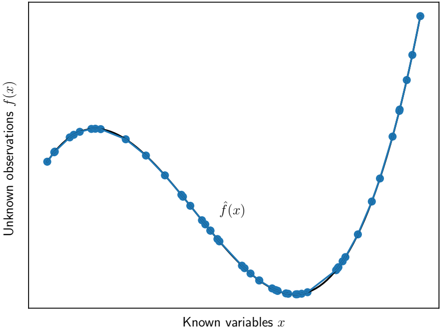
- Want to predict unknown $f(x)$ for observation x
- **ML**: *Learn* approximation $\hat{f} \sim f$ based on *training data* $\mathcal{X}$
- **NA**: fit an interpolant (piecewise-linear) to f on $\mathcal{X}$
- Both cases: more data $\Rightarrow$ better $\hat{f}$
- Real data not perfectly balanced $\Rightarrow$ $\hat{f} \to f$ non-uniformly
- If we have enough data, it doesn't matter

Some basic numerical analysis results

When $\hat{f}$ is a piecewise linear spline:

For h "small enough" – let q be the querry point

$$|f(q) - \hat{f}(q)| \sim \mathcal{O}(h^2)$$



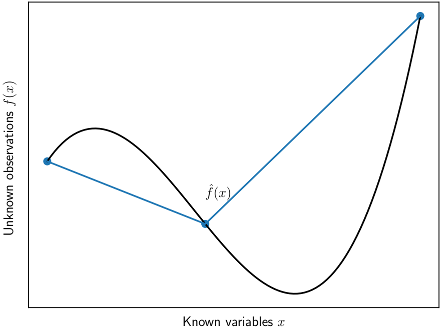
- h is a "mesh fineness" parameter $\sim$ distance between points in $\mathcal{X}$
- For irregular $\mathcal{X}$, h could be the distance from q to the nearest neighbor in $\mathcal{X}$
- Constants proportional to the Lip constant of ∇f

Some basic numerical analysis results

When $\hat{f}$ is a piecewise linear spline:

For h “small enough” – let q be the querry point

$$|f(q) - \hat{f}(q)| \sim \mathcal{O}(h^2)$$



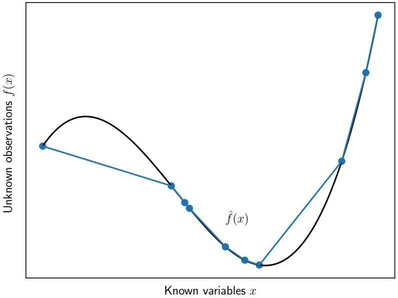
- h is a “mesh fineness” parameter $\sim$ distance between points in $\mathcal{X}$
- For irregular $\mathcal{X}$, h could be the distance from q to the nearest neighbor in $\mathcal{X}$
- Constants proportional to the Lip constant of ∇f

Some basic deep learning

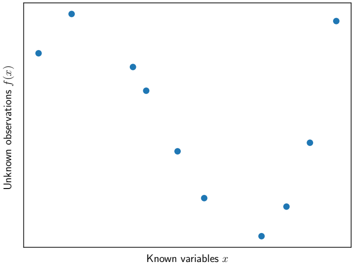
- Train a fully-connected multi-layer perceptron (MLP) using $\mathcal{X}$
- The most popular activation function is ReLU (piecewise linear)
- In modern ML, train as close to zero error as possible (interpolate)

Some basic deep learning

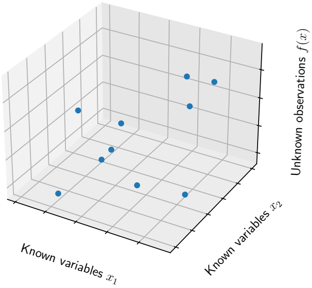
- Train a fully-connected multi-layer perceptron (MLP) using $\mathcal{X}$
- The most popular activation function is ReLU (piecewise linear)
- In modern ML, train as close to zero error as possible (interpolate)



The curse of dimensionality

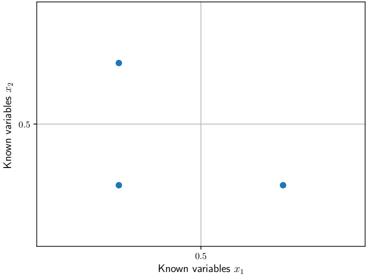


10 training points in 1D



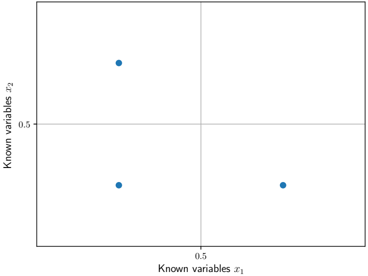
10 training points in 2D

The curse of ~~dimensionality~~ no data



Need data in all quadrants?

The curse of ~~dimensionality~~ no data



Need data in all quadrants?

- Inference in 2D : $2^2 = 4$
- Inference in 10D : $2^{10} \approx 1000$
- Inference in 100D : $2^{100} \approx 10^{30}$ (orders of magnitude bigger than exascale)
- Many ML problems : inference in 1000+ dimensions

Measure collapse

Can we still make good predictions where we **do** have data?

Measure collapse

Can we still make good predictions where we **do** have data?

No, because we have no data anywhere

We measure where we *might* have enough data to make a prediction using the "convex hull" of the training data $CH(\mathcal{X})$

Measure collapse

Can we still make good predictions where we **do** have data?

No, because we have no data anywhere

We measure where we *might* have enough data to make a prediction using the "convex hull" of the training data $CH(\mathcal{X})$

If $\mathcal{X}$ are sampled from *any* distribution, $\mu(CH(\mathcal{X})) \to 0$ *exponentially* as d grows

This is called a *concentration of measure*

Gorban and Tyukin, Stochastic separation theorems. *Neural Networks 94*, pp. 255-259 (2017).

Example

Suppose that we uniformly sample $x = (x_1, x_2, \ldots, x_d)$ from $[0,1]^d$

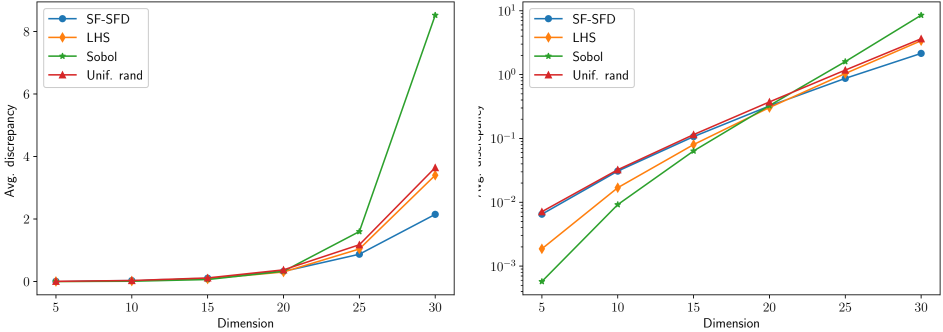
$$\|x - \frac{1}{2}\|_2^2 = \sum_{i=1}^{d} (x_i - \frac{1}{2})^2.$$

$$\mathbb{E}\left[\left(x_i - \frac{1}{2}\right)^2\right] = \int_0^1 \left(u - \frac{1}{2}\right)^2 du = \frac{1}{12}$$

with finite variance ν

By CLT for all $x \in \mathcal{X}$: $\mathbb{E}[\|x - \frac{1}{2}\|_2^2] = \frac{d}{12}$ with variance $\frac{\nu}{d} \to 0$ as $d \to \infty$.

Collapse of some common distributions



Garg, Chang, and Raghavan, Stochastic optimization of Fourier coefficiencts to generate space-filling designs. *To appear in Winter Sim 2023.*

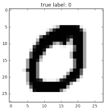
Representation learning solution

“There’s more to machine learning than function approximation”

Representation learning solution

"There's more to machine learning than function approximation"

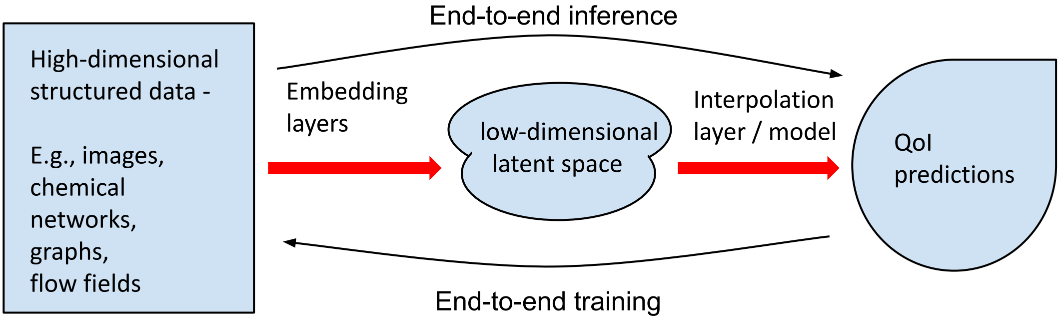
- f is often highly *structured* – MLPs with nothing else are from the 60s



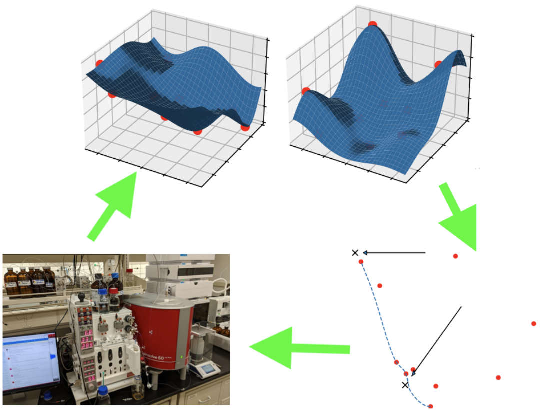
$$\xrightarrow{\quad f \quad} \quad 0$$

28×28 pixels $\neq 784$ dimensions...

Modern deep learning pipeline



Hope in context of optimization



Global modeling is harder than optimization

For optimization, only need model accuracy near the solution...

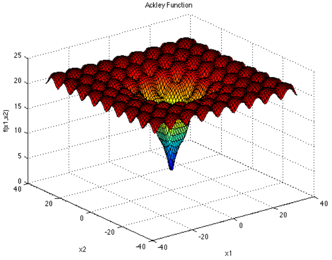
- Global modeling is *significantly harder than optimizing*
- To build a *globally accurate model* over n variables, need $\mathcal{O}(2^n)$ samples
- To build a *locally accurate model* over n variables, need $\mathcal{O}(n)$ samples

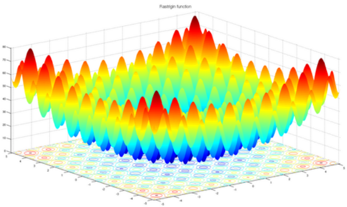
Global optimization

In global optimization literature...

- Balance exploration vs. exploitation
- Drive *global model error* to zero
- Need exponentially many samples to guarantee global convergence

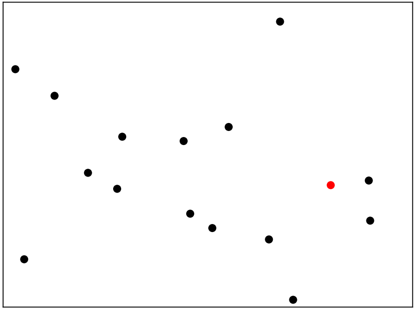
Guarantees convergence for problems with thousands of local minima





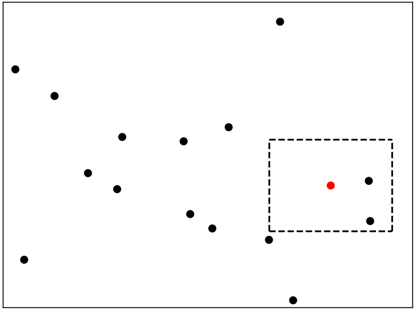
Local optimization

- Only exploit – maybe multi-start or large initial search
- Fit a model that is *locally accurate*
  - Sample requirement grows only *linearly with* dimension
- Modification is as simple as putting a *trust-region* around interesting points



Local optimization

- Only exploit – maybe multi-start or large initial search
- Fit a model that is *locally accurate*
  - Sample requirement grows only *linearly with* dimension
- Modification is as simple as putting a *trust-region* around interesting points



ParMOO

parMOO

Written in Python

Version 0.3.0 is now available on available on `pip`, `conda-forge`, and `GitHub`

`https://github.com/parmoo/parmoo`

`https://parmoo.readthedocs.io`

PyPI

CONDA-FORGE

Chang and Wild. ParMOO: A Python library for parallel multiobjective simulation optimization. *JOSS 8(82):4468 (2023).*