The curse of dimensionality

Tyler Chang

Argonne National Laboratory

CAA&CM Argonne Student Visit
Sep 28, 2023

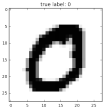
Everything is a function

Everything is a function

$$f(x_1, x_2, x_3, \ldots, x_n) \to y$$

Everything is a function

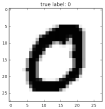
$$f(x_1, x_2, x_3, \ldots, x_n) \rightarrow y$$



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

Everything is a function

$$f(x_1, x_2, x_3, \ldots, x_n) \to y$$



$\xrightarrow{f}$ 0

"The dog jumped over the ..." $\xrightarrow{f}$ "fence"

Everything is a function

$$f(x_1, x_2, x_3, \ldots, x_n) \rightarrow y$$

Everything is a function

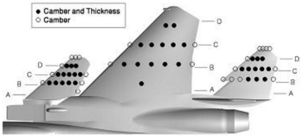
$$f(x_1, x_2, x_3, \ldots, x_n) \rightarrow y$$

Molecular discovery for better battery electrolytes (photo from the Argonne MERF).

Everything is a function

$$f(x_1, x_2, x_3, \ldots, x_n) \to y$$

Molecular discovery for better battery electrolytes (photo from the Argonne MERF).

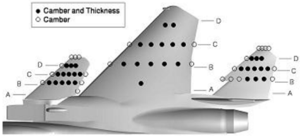


Nonparametric aircraft geometry (photo from NASA Langley).

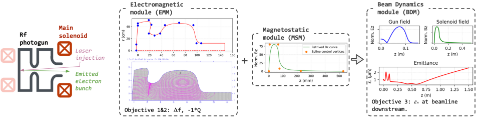
Everything is a function

$$f(x_1, x_2, x_3, \ldots, x_n) \rightarrow y$$

Molecular discovery for better battery electrolytes (photo from the Argonne MERF).



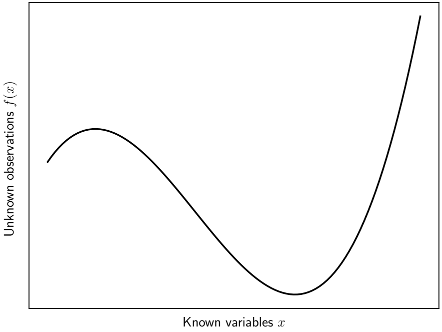
Nonparametric aircraft geometry (photo from NASA Langley).



Particle accelerator designs (photo from simulation run on HPCs at Argonne).

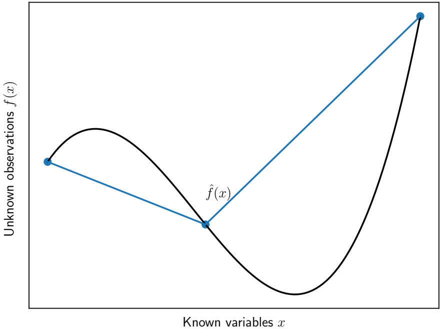
The fundamental machine learning problem

The fundamental machine learning problem



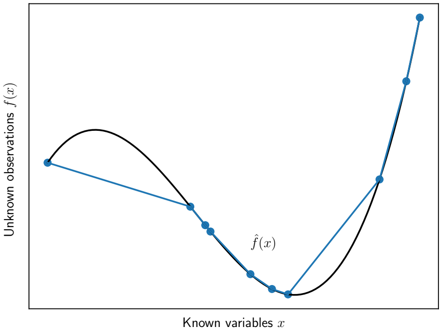
- Want to predict unknown $f(x)$ for observation x

The fundamental machine learning problem



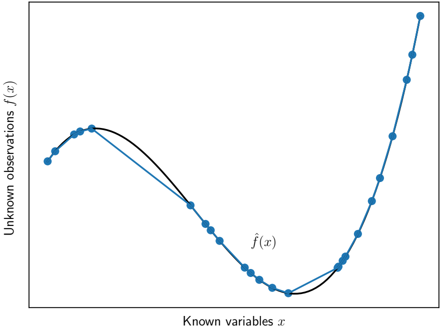
- Want to predict unknown $f(x)$ for observation x
- **ML**: *Learn* approximation $\hat{f} \sim f$ based on *training data* $\mathcal{X}$
- **NA**: fit an interpolant (piecewise-linear) to f on $\mathcal{X}$

The fundamental machine learning problem



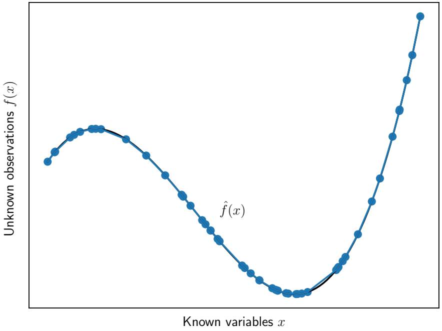
- Want to predict unknown $f(x)$ for observation x
- **ML**: *Learn* approximation $\hat{f} \sim f$ based on *training data* $\mathcal{X}$
- **NA**: fit an interpolant (piecewise-linear) to f on $\mathcal{X}$
- Both cases: more data $\Rightarrow$ better $\hat{f}$

The fundamental machine learning problem



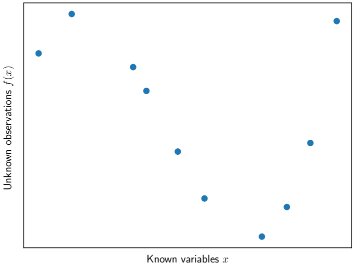
- Want to predict unknown $f(x)$ for observation x
- **ML**: *Learn* approximation $\hat{f} \sim f$ based on *training data* $\mathcal{X}$
- **NA**: fit an interpolant (piecewise-linear) to f on $\mathcal{X}$
- Both cases: more data $\Rightarrow$ better $\hat{f}$
- Real data not perfectly balanced $\Rightarrow$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

The fundamental machine learning problem

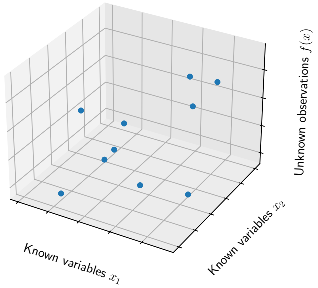


- Want to predict unknown $f(x)$ for observation x
- **ML**: *Learn* approximation $\hat{f} \sim f$ based on *training data* $\mathcal{X}$
- **NA**: fit an interpolant (piecewise-linear) to f on $\mathcal{X}$
- Both cases: more data $\Rightarrow$ better $\hat{f}$
- Real data not perfectly balanced $\Rightarrow$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

The curse of dimensionality

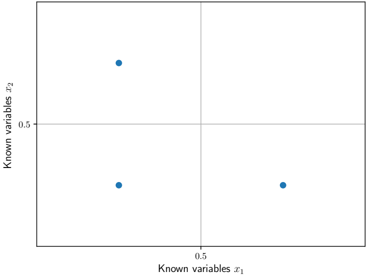


10 training points in 1D



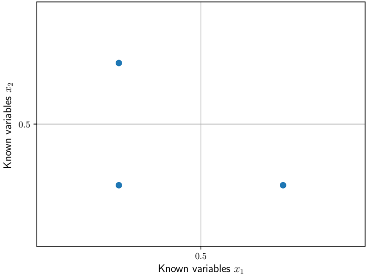
10 training points in 2D

The curse of ~~dimensionality~~ no data



Need data in all quadrants?

The curse of ~~dimensionality~~ no data

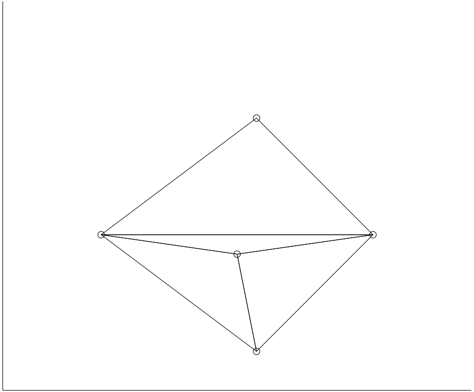


Need data in all quadrants?

- 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

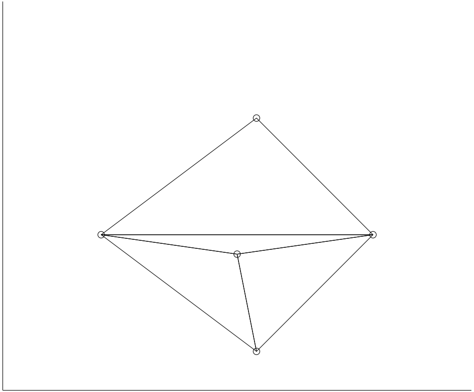
The curse of ~~dimensionality~~ too much data

Classical methods to "connect the dots" in high-dimensions (from applied math literature) rely on meshing:



The curse of ~~dimensionality~~ too much data

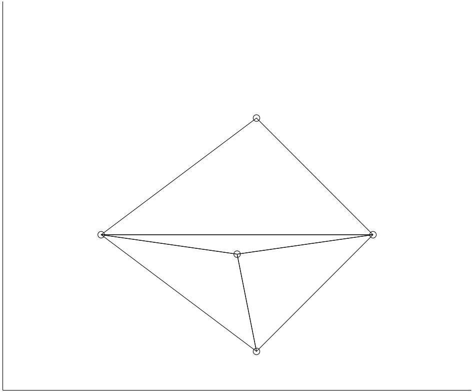
Classical methods to "connect the dots" in high-dimensions (from applied math literature) rely on meshing:



- A mesh of n points in $\mathbb{R}^d$ can have up to $\mathcal{O}(n^{d/2})$ elements

The curse of ~~dimensionality~~ too much data

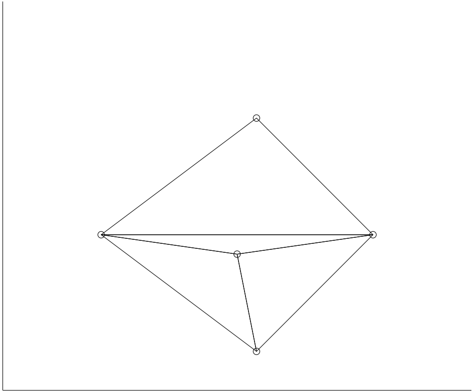
Classical methods to “connect the dots” in high-dimensions (from applied math literature) rely on meshing:



- A mesh of n points in $\mathbb{R}^d$ can have up to $\mathcal{O}(n^{d/2})$ elements
- Takes **at least** $\mathcal{O}(n^{d/2})$ time to compute

The curse of ~~dimensionality~~ too much data

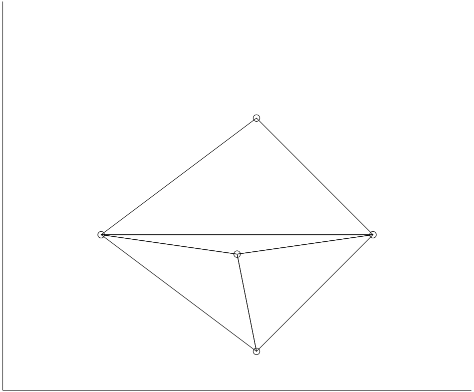
Classical methods to "connect the dots" in high-dimensions (from applied math literature) rely on meshing:



- A mesh of n points in $\mathbb{R}^d$ can have up to $\mathcal{O}(n^{d/2})$ elements
- Takes **at least** $\mathcal{O}(n^{d/2})$ time to compute
- Requires **at least** $\mathcal{O}(n^{d/2})$ storage

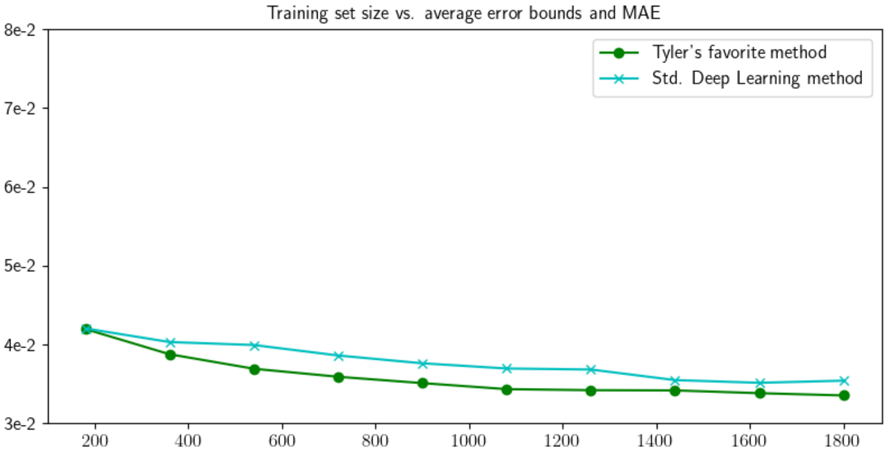
The curse of ~~dimensionality~~ too much data

Classical methods to "connect the dots" in high-dimensions (from applied math literature) rely on meshing:

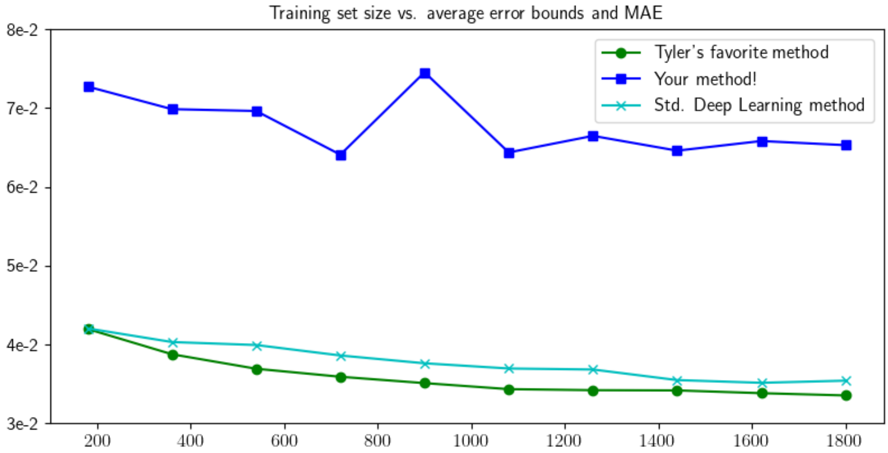


- A mesh of n points in $\mathbb{R}^d$ can have up to $\mathcal{O}(n^{d/2})$ elements
- Takes **at least** $\mathcal{O}(n^{d/2})$ time to compute
- Requires **at least** $\mathcal{O}(n^{d/2})$ storage
- **Impossible for large data sets**

Results for some methods



Results for some methods



Questions

Everything is a function

The fundamental ML problem (multidimensional inference)

The curse of dimensionality

- Not enough data to make accurate predictions
- Too much data for many "classical" methods

Data from some existing methods

Some courses to take

Math:

- Advanced linear algebra
- Numerical analysis
- Functional analysis

CS:

- Data structures & algorithms
- Parallel computing
- Data analysis and/or Machine learning