Problem: "Curse of dimensionality"

Geometric formulation:
\[ x = \{ x_1 \ldots x_m \} \in \mathbb{R}^m \]

Training sample \[ \{ x_i \}_{i=1}^N \in \mathbb{R}^m \]
\[ \{ (y_i, x_i) \}_{i=1}^N \in \mathbb{R}^{m+1} \]

\( f(x) \) is a surface "above" \( \mathbb{R}^m \) that passes near \( \{ y_i, x_i \}_{i=1}^N \).

If \( f(x) \) is arbitrarily complex and (more or less) completely unknown,

\[ \Rightarrow \] need dense sample to learn it well

but, dense samples are hard to get in high dimensions.

"Curse" even for the very largest data sets
(a) Sampling density \( \sim N_{m}^{\frac{1}{m}} \)

let \( N_{1} \) = dense sample in \( \mathbb{R}^{1} \)

then \( N_{m} = N_{1}^{m} \) = sample size for same density in \( \mathbb{R}^{m} \)

(b) Interpoint distances are all large, and about equal in \( \mathbb{R}^{m} \)

\[ \text{dist} \sim (\text{vol})^{\frac{1}{m}} \]

neighborhoods that contain even a few points have large radii.

consider \( \mathbf{x} \sim \mathcal{U}^{m}(0,1) \)

neighborhood (hypercube) containing fraction \( p \) points, has edge length

\[ e_{m}(p) = p^{\frac{1}{m}} \]

\[ e_{10}(0.01) = 0.63, \quad e_{10}(0.1) = 0.80 \]

\[ e_{100}(0.01) = 0.95 \]

\[ d(\text{closest point})/d(1000\text{ points}) = \frac{1}{2}, \quad m = 10 \]
(c) (nearly) all points close to edge (boundary)

\[ x \sim U^m(0, 1) \]

Expected \( L^\infty \) dist. to closest point

\[ r(m, N) = \frac{1}{2} \left( \frac{1}{N} \right)^{\frac{1}{m}} \]

\[ r(10, 1000) \approx 0.5 \]
\[ r(1000, 10^6) \approx 0.45 \]

\[ = \max. \text{ dist. to all edges} \]

So nearly all points are closer to an edge than to another training point. (\( L^\infty \) dist. most favorable)

Prediction much harder near edges (ex: trap. vs. interp.)

Most points on or near convex hull of training sample.
every point sees itself as an outlier with (nearly) all other points to one side clumped near the origin.

Note: no problem for theory

imagine \( N \to \infty \) \( \Rightarrow \) arbitrarily dense samples (asymptopia)

Only a practical problem

(a) how to get and use \( N \to \infty \), or

* (b) how to deal with (relatively) small (sparse) samples
"Curse-of-dimensionality" is a diversion - does not exist.

Problem is complexity rather than dimensionality.

Illustrations:

Kolmogorov's Thm (Hilbert's 13th prob.)

\[ f(x_1, \ldots, x_m) = \sum_{j=1}^{n} g_f \left[ \sum_{i=1}^{m} \lambda_i \phi_j^n(x_i) \right] \]

\[ \{ \lambda_i \}_{i=1}^{n} = \text{"universal constants"} \]
\[ \{ \phi_j^n(x) \}_{i=1}^{2m+1} = \text{universal transformations} \]

\[ g_f(u) = \text{continuous, totally characterizes } f(x_1, \ldots, x_m) \]

\[ \exists \text{ a 1-dim. continuous function } g_f(u) \text{ that characterizes any continuous function of } m \text{-arguments} \]

\[ \Rightarrow \text{ # functions s.t. } \dim > 1 \]
$\Rightarrow \quad \text{no curse-of-dimensionality}$

However, complexity of $f(x)$ has not been reduced.

for $f(x_1 \ldots x_m)$ fairly mild, $x \in \mathbb{R}^m$

$\Rightarrow g_f(u)$ very wild for $u \in \mathbb{R}^1$

Hilbert: "The not true."

Intuition was correct: "bad" functions cannot be represented in a simple way by "good" functions.

G. G. Lorentz (1986): "Kolmogorov's theorem shows only that the number of variables $n$ is not a satisfactory characteristic of badness."


"Beat curse-of-dim." $\Rightarrow$ read fine print
target function $f(x)$

$$y = E[y|x] + (y - E[y|x]) \sim \text{"noise"} \sim$$

"Curse" can be overcome:

Example: $y = f(x) + \varepsilon$, $\text{var}(\varepsilon) = \sigma^2$

suppose $f(x) = \sum_{j=1}^{m} a_j x_j$ (linear fun.)

$$\hat{f}(x) = \sum_{j=1}^{m} \hat{a}_j x_j \text{ (linear approx.)}$$

$$[\hat{a}_j]_1^m = \arg\min \sum_{i=1}^{N} \left[ y_i - \sum_{j=1}^{m} a_j x_{ij} \right]^2$$

(linear least squares fit)

$$E \left[ f(x) - \hat{f}(x) \right]^2 = \frac{m \sigma^2}{N}$$

increases linearly with $N$,

not exponentially as "curse"

would suggest.
More generally, suppose

\[ f(x) = \sum_{m=1}^{M} \lambda_m B_m(x) \]

\[ \hat{f}(x) = \sum_{m=1}^{M} \hat{\lambda}_m B_m(x) \]

\[ \{ \hat{\lambda}_m \}^M_{m=1} = \arg\min \sum_{i=1}^{N} [y_i - \sum_{m=1}^{M} \lambda_m B_m(x_i)]^2 \]

Then

\[ \mathbb{E} \left[ f(x) - \hat{f}(x) \right]^2 = \frac{M \sigma^2}{N} \]

What happened?

\[ f(x) \] happens to be in a very restricted class of functions, and we chose an estimator (method) especially appropriate for that class.

Either:

(1) Memew class in advance.
(2) Got lucky.
Another (final) illustration:

Suppose

\[ f_m (x_1 \ldots x_n) = \sum_{j=1}^{m} a_j x_j \quad (n\text{-dim.}) \]

\[ f_1 (x_1) = \sum_{j=1}^{m} a_j \varphi_j (x_1) \quad (1\text{-dim.}) \]

e.g. \[ \varphi_j (x_1) = x_1^{j-1} \quad \text{n-1 degree polynomial} \]

We take

\[ \hat{f}_m (x_1) = \sum_{j=1}^{m} \hat{a}_j x_j \]

\[ \hat{f}_1 (x_1) = \sum_{j=1}^{m} \hat{a}_j \varphi_j (x_1) \]

Then complexity of \( \hat{f}_m \) and \( \hat{f}_1 \) is the same

\( \hat{f}_m = \text{simple function of n-vars.} \)

\( \hat{f}_1 = \text{complicated 1-var.} \)
The basic reason for the “curse” is that high dimensional functions have the potential to be much more complicated than low dimensional ones, and those complications are harder to discern.

The only way to beat the “curse” (i.e. be successful) is to incorporate knowledge outside the data (assumptions) about f(x), that is correct.

Choosing a method (implicitly or explicitly) does this.

Methods differ on:

1) the particular nature of the knowledge one assumes (impose)
2) the strength of that imposition
3) their robustness to violations of these assumptions.

Need a toolkit of methods
Not just a hammer