MA 5755: Data Analysis & Visualization in R/Python/SQL

Week 2: Structure, Dimension, and Representation in Supervised & Unsupervised Learning

Dr. Rakhi Singh
Department of Mathematics
IIT Madras

Spring 2026

Overview and Learning Goals

- **Geometric View.** View learning methods as shapes in data space; topics: *projection vs neighborhoods*, *bias–variance tradeoff*, methods: least squares, kNN.
- **Decision-Theoretic Foundations.** Understand prediction through loss and risk; topics: *conditional mean vs median*, *squared loss*, *absolute loss*.
- **High Dimensionality.** See why high dimension breaks intuition and local methods; topics: *curse of dimensionality*, *distance concentration*, *empty neighborhoods*.
- **Structure via Restrictions.** Learn how assumptions stabilize function estimation; topics: *smoothness*, *sparsity*, *additivity*, *penalization*, *kernels*, *basis expansions*.
- **Unsupervised Representations.** Discover structure from X alone and how it supports prediction;

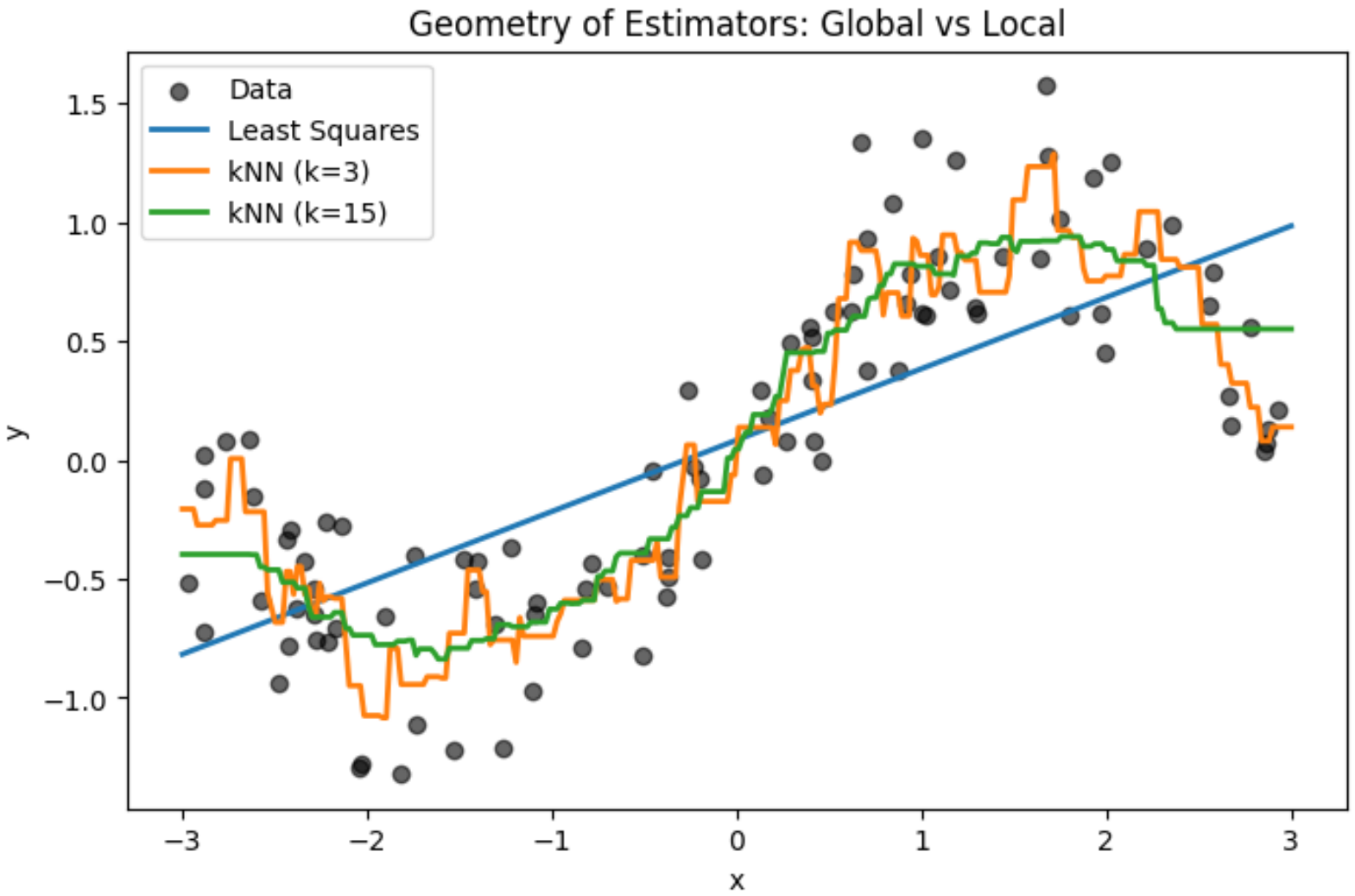
A Geometric View of Learning

Learning is not just algebra; it is geometry in the space of inputs.

- Every method draws a shape through the data cloud; A p-dimensional space for p features. One finds a way to map *response* on top of this space.
- Some methods look at the whole space at once, others only at what is nearby; think about *global vs local structure*.
- Today, we contrast two extremes:
  - a straight global shape: least squares
  - a local, data-driven shape: kNN

Same data. Different models may lead to very different geometries.

A Geometric View of Learning



Least Squares: One Shape for All Data

Think of fitting a single line or plane through a cloud of points.

- The model chooses the best global direction that explains the data; *projection*, *linear subspace*.
- Every point influences the final fit; *global influence*, *stability*.
- This works well when the true pattern is simple and smooth, but struggles when the relationship is curved; *model bias*.

Model equation (global):

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x_1} + \cdots + \beta_p \mathbf{x_p} + \boldsymbol{\epsilon},$$

where ϵ is a random error term.

A single linear function is used everywhere in the input space.

kNN: Let the Neighborhood Decide

Instead of one global rule, kNN asks: *"What do nearby points say?"*

- Prediction is an average of the closest observations; *local averaging*, *distance*.
- Different parts of the space get different predictions; *piecewise structure*, *adaptivity*.
- Small neighborhoods follow data closely, large neighborhoods smooth things out; *bias–variance tradeoff*.

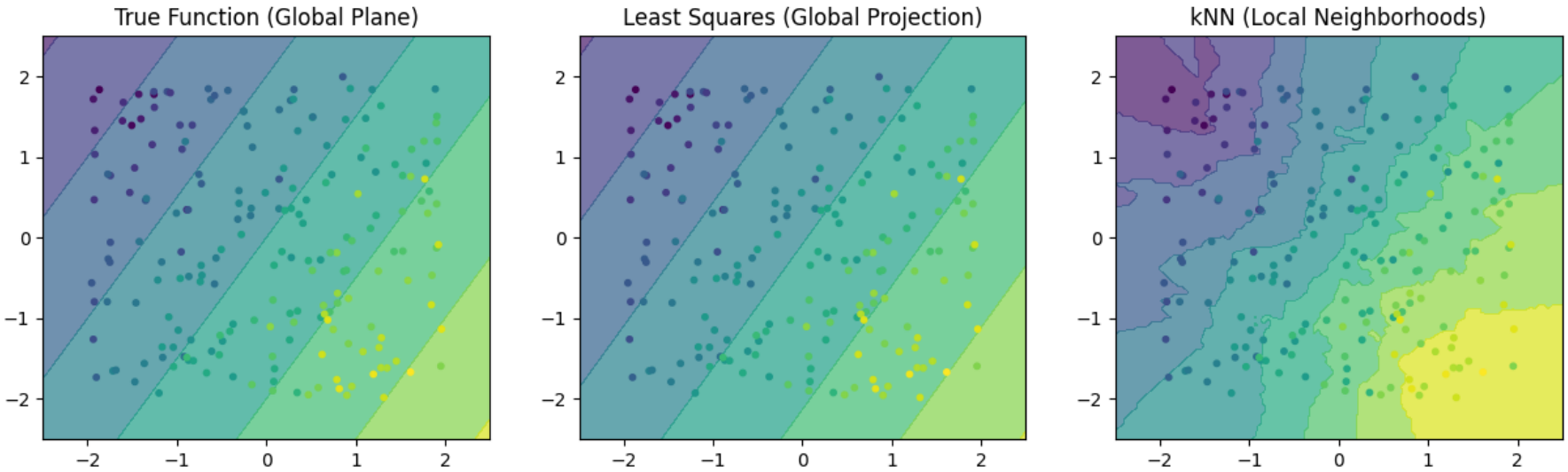
Model equation (local):

$$\hat{y}(\mathbf{x}) = \frac{1}{k} \sum_{i \in N_k(\mathbf{x})} y_i$$

The prediction at x depends only on the responses of its k nearest neighbors. Many small rules instead of one big one!

2D example

```
TRUE function:      y = 3*x1 - 2*x2
Least Squares fit:  y = 2.99*x1 + -1.95*x2 + -0.15
kNN: no explicit equation (local averaging of neighbors)
```



Geometry Explains Bias and Variance

The prediction surfaces and their shapes explain bias and variance.

- High bias, low variance: A smooth, rigid surface (as in least squares) uses one global shape everywhere. It ignores local curvature in the data, but gives stable predictions across the space. It cannot bend to match curved patterns (high bias). It changes little if you resample data (low variance).
- Low bias, high variance: A flexible, patchy surface (as in kNN with small k) bends to follow nearby points. It captures local structure, but changes strongly when the data change. Shape is flexible and follows data closely (low bias). It changes a lot if data change (high variance).
- **The tradeoff:** Learning is choosing how much freedom the surface should have: too stiff misses structure, too flexible follows noise.

Decision-Theoretic View of Prediction

Prediction can be framed as a statistical decision problem.

- For a given input x, the response Y is random with conditional distribution $P(Y \mid X = x)$.
- A predictor chooses a value $\hat{y}$ to minimize the expected loss under this distribution.
- The optimal prediction solves a conditional risk minimization problem:

$$\hat{y}(x) = \arg\min_{a} \; \mathbb{E}\big[L(Y, a) \mid X = x\big]$$

- Different choices of loss function $L(Y, a)$ lead to different optimal predictors.
- The learning problem is therefore defined by both the data model and the loss function.

What we predict depends on how we measure error.

Loss Functions and Risk

A loss function measures how costly a prediction error is.

- The loss compares the true outcome Y with the prediction $\hat{y}$.
- Different losses reflect different priorities: large errors may be punished heavily or mildly.
- The goal of learning is to minimize the average loss over all possible outcomes.

Expected risk:

$$R(\hat{y}) = \mathbb{E}[L(Y, \hat{y}) \mid X = x]$$

We choose the prediction that minimizes expected loss.

Squared Loss and the Conditional Mean

Suppose we measure error using squared loss:

$$L(Y, \hat{y}) = (Y - \hat{y})^2$$

- Large errors are penalized strongly because they are squared.
- The prediction that minimizes expected squared loss is the average outcome at x.
- This means the optimal prediction is the conditional mean of Y given $X = x$.

$$\hat{y}(x) = \mathbb{E}[Y \mid X = x]$$

With squared loss, the best prediction is the mean.

Absolute Loss and the Conditional Median

Now suppose we measure error using absolute loss:

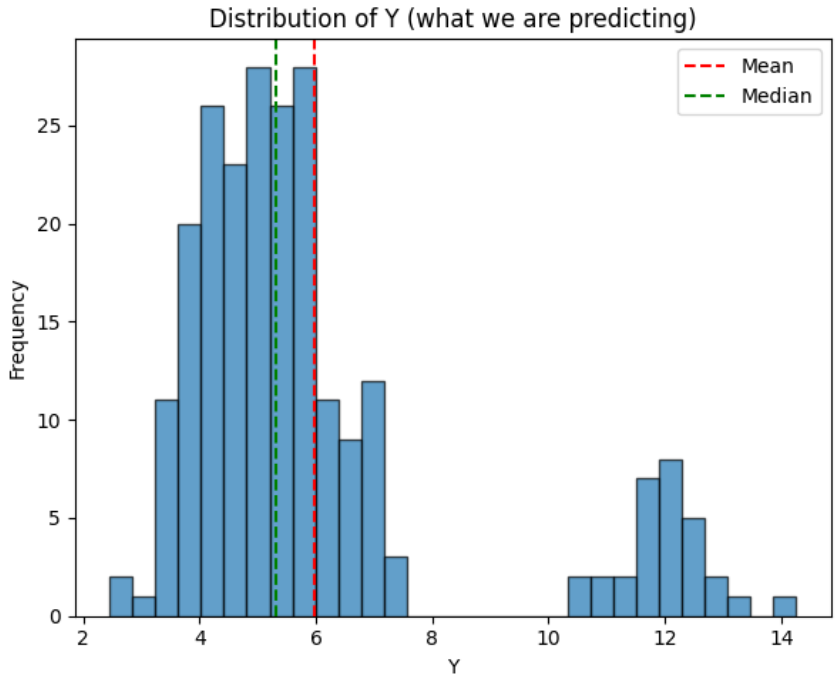
$$L(Y, \hat{y}) = |Y - \hat{y}|$$

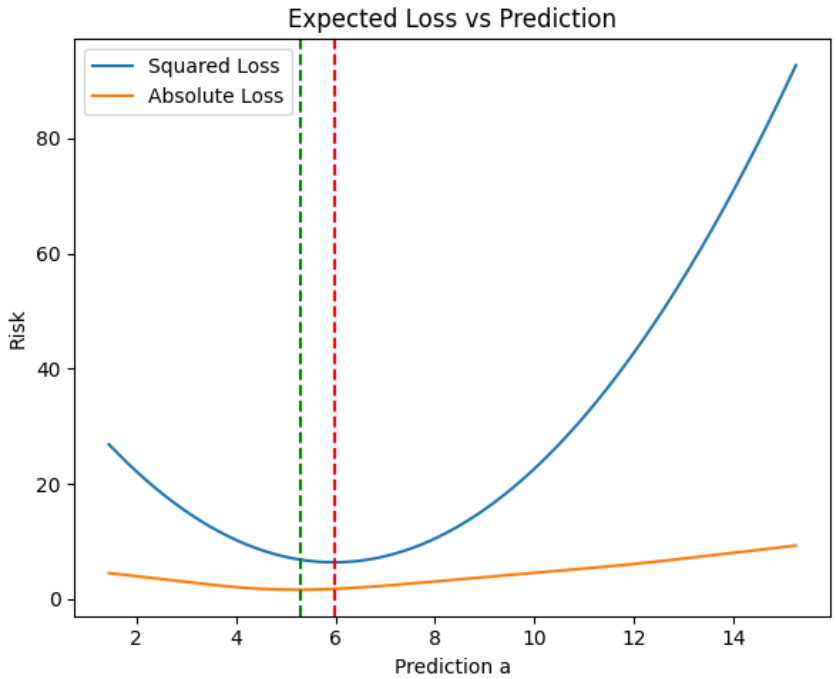
- All errors grow linearly, so outliers are less dominant.
- The prediction that minimizes expected absolute loss splits the outcomes in half.
- This makes the optimal prediction the conditional median of Y given $X = x$.

$$\hat{y}(x) = \text{median}(Y \mid X = x)$$

With absolute loss, the best prediction is the median.

A small demo


Distribution of Y (what we are predicting)
Mean
Median
Frequency
Y
0
5
10
15
20
25
2
4
6
8
10
12
14


Expected Loss vs Prediction
Squared Loss
Absolute Loss
Risk
Prediction a
0
20
40
60
80
2
4
6
8
10
12
14

Categorical Responses and Classification

Now suppose Y takes values in a finite set of classes $\{1, \ldots, K\}$.

0–1 loss:

$$L(Y, a) = \begin{cases} 0 & \text{if } Y = a \\ 1 & \text{if } Y \neq a \end{cases}$$

The conditional risk is minimized by choosing the most probable class:

$$\hat{f}(x) = \arg\max_k P(Y = k \mid X = x)$$

- This rule is known as the Bayes classifier.
- It depends only on the conditional class probabilities.
- Different loss functions would lead to different classification rules.

Classification is also a risk minimization problem.

How Do We Survive High Dimension?

In low dimensions, geometry matches our intuition. In high dimensions, it does not.

- As dimension grows, data spread out in many directions at once.
- Concepts like *near*, *far*, and *dense* begin to lose their usual meaning.
- Methods that rely on local neighborhoods assume that nearby points carry useful information.

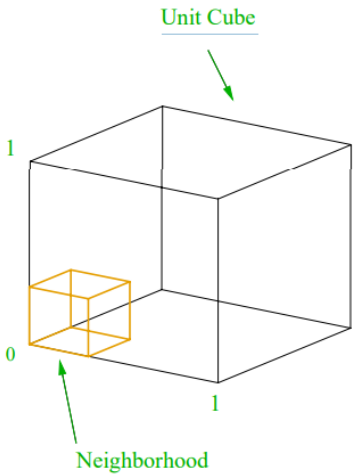
High-dim. problems are only solvable because real data are not arbitrary.

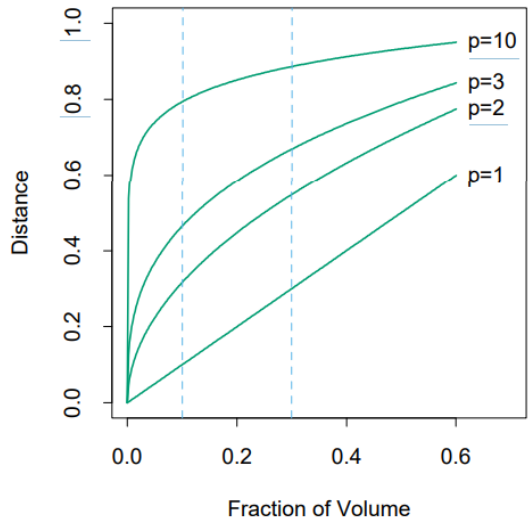
- Useful data lie on low-dimensional structures: manifolds, sparse subspaces, or additive models.
- Learning succeeds by exploiting this hidden structure.
- Dimensionality reduction and regularization are ways of enforcing geometric simplicity.

Local Neighborhoods Stop Being Local

Let $\mathbf{X} \sim$ Uniform(p-dim unit cube). To capture a fraction r of the data, a hypercube neighborhood must have edge length, $e_p(r) = r^{1/p}$.

For $p = 10$, to capture only 10% of the data, we need $e_{10}(0.1) \approx 0.80$, 80% coverage of the range of every coordinate.





Neighborhoods that were local in low dimensions become global in high dimensions.

Nearest Neighbors Drift to the Boundary

Consider N points uniformly distributed in a p-dimensional unit ball.

The median distance from the origin to the closest point is

$$d(p, N) = \left(1 - \frac{1}{2^{1/N}}\right)^{1/p}.$$

For $N = 500$ and $p = 10$:

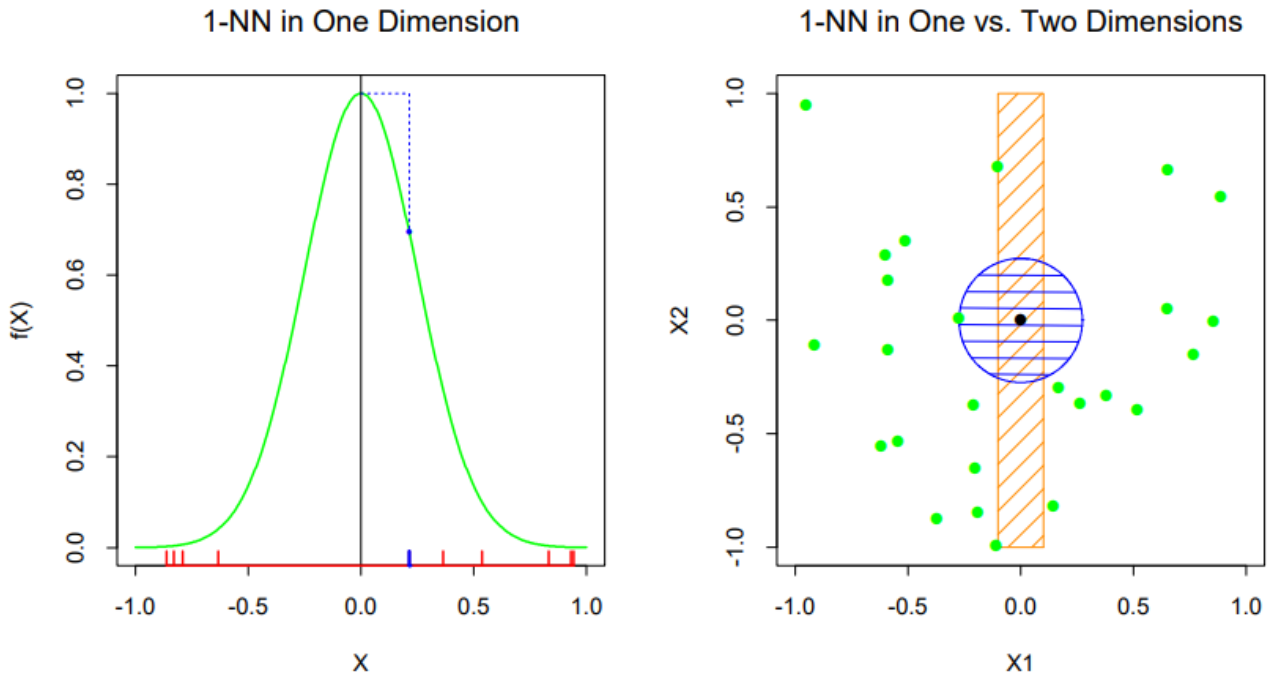
$$d(p, N) \approx 0.52.$$

- Most data points lie closer to the boundary than to each other.
- Typically, we find prediction is difficult at edges than in between.
- The above happens because the prediction then becomes an extrapolation rather than an interpolation.

In high dimensions, the nearest neighbor is not very near.

Bias and Variance of 1-NN in High Dimensions

Let $x_i \sim$ U[-1,1]. The true relationship is $f(X) = e^{-8\|\mathbf{X}\|^2}$ and we predict $f(0)$ using 1-nearest neighbor.



Unless the nearest neighbor is at 0, $\hat{y}_0$ will be smaller than $f(0)$ in this example, and the average estimate will be biased downward.

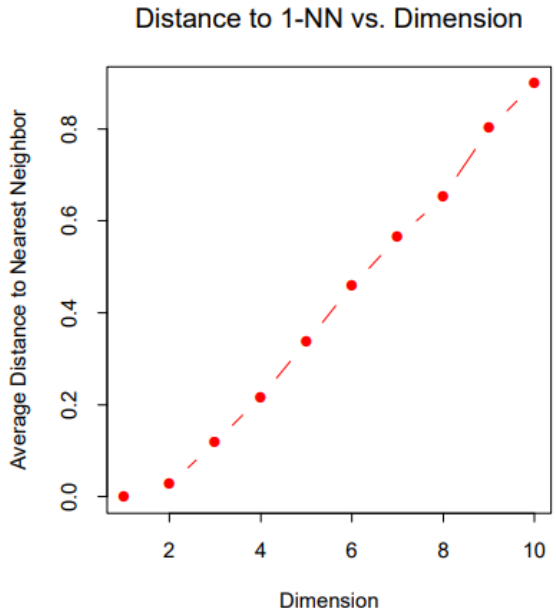
Bias and Variance of 1-NN in High Dimensions

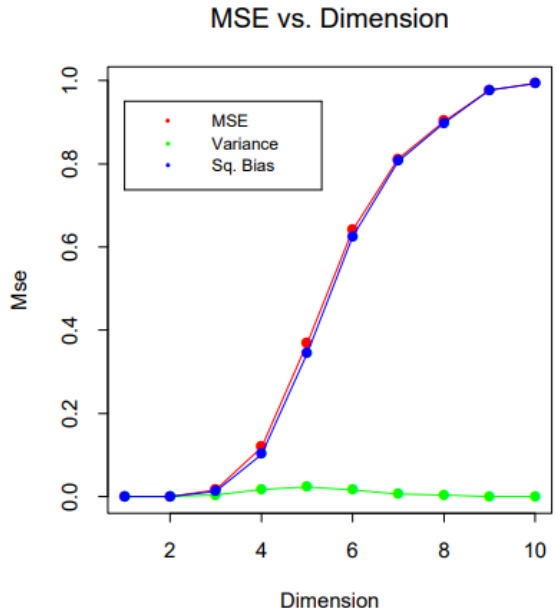
The mean squared error decomposes as:

$$\text{MSE}(x_0) = E_T(\hat{y}_0) = \text{Var}_T(\hat{y}_0) + \text{Bias}^2(\hat{y}_0).$$

The variance here is due to the sampling variance if 1-NN.

In low dimensions and with $N = 1000$, the nearest neighbor is close to x_0, so both bias and variance are small. As p increases, the nearest neighbor moves far from x_0.





Why Local Methods Break Down

Local averaging assumes nearby points share similar responses.

- In high dimension, nearest neighbors are far away in absolute distance.
- Neighborhoods include points that are not truly similar.
- The estimate becomes noisy and unstable.

Formally, the bias–variance tradeoff worsens:

- small neighborhoods $\Rightarrow$ very high variance,
- large neighborhoods $\Rightarrow$ very high bias.

Local methods are no longer local.

Why Linear Models Escape the Curse

Assume the true model is linear:

$$Y = X^T\beta + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2).$$

For least squares, the expected prediction error satisfies:

$$\mathbb{E}[\text{EPE}] \approx \sigma^2 \left(1 + \frac{p}{N}\right).$$

- There is no bias under the correct linear model.
- Variance grows only linearly with dimension.
- Strong structural assumptions prevent exponential complexity.
- But the model assumptions must hold. No guarantee if they don't.

Structure replaces locality as the defense against high dimension.

Local vs Rigid Models in High Dimensions

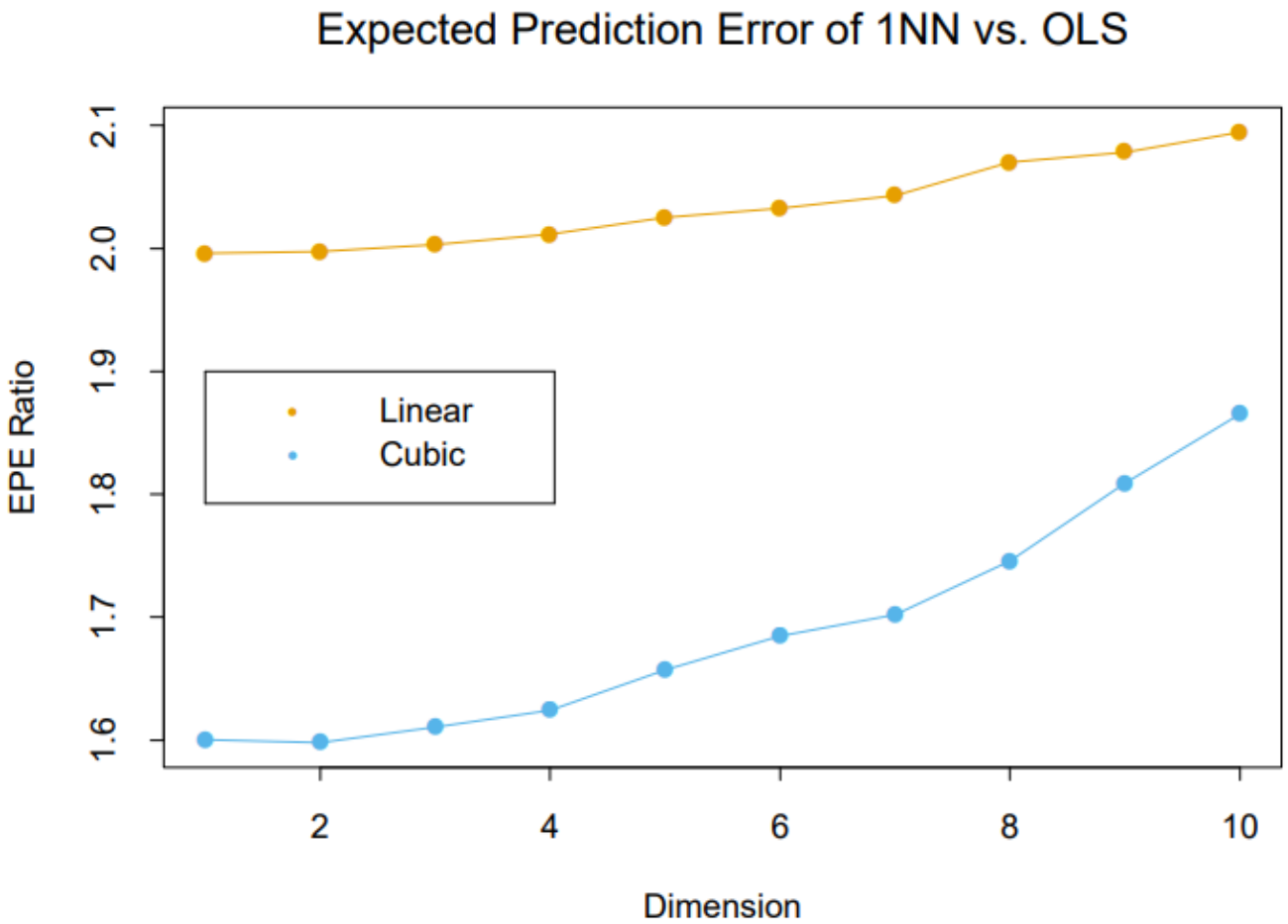


FIGURE 2.9. *The curves show the expected prediction error (at $x_0 = 0$) for 1-nearest neighbor relative to least squares for the model $Y = f(X) + \varepsilon$. For the orange curve, $f(x) = x_1$, while for the blue curve $f(x) = \frac{1}{2}(x_1 + 1)^3$.*

Local vs Rigid Models in High Dimensions

- When f is linear, least squares is unbiased and has low variance.
- 1-nearest neighbor always has variance at least σ^2 and grows worse with dimension.
- When f is nonlinear but low-dimensional, kNN can win.
- If the assumptions are wrong, all bets are usually off.
- No method dominates without assumptions about structure.
- For proof about the *EPEs*, you can go to the textbook, *ESL*.

The curse of dimensionality is avoided by imposing structure, not by more data alone.

Structure via Restrictions: Why Assumptions Help

High-dimensional learning is only possible because we restrict the class of functions we allow.

Core idea. Instead of estimating an arbitrary function $f(x_1, \ldots, x_p)$, we assume f belongs to a simpler family:

$$f \in \mathcal{F}_{\text{restricted}} \subset \mathcal{F}_{\text{all}}.$$

- Smaller function classes reduce variance and sample complexity.
- The price is bias: we may miss the true function if the restriction is wrong.
- All modern learning methods differ mainly in how they restrict $\mathcal{F}$.

Learning is possible because the world is not arbitrary.

Smoothness: Nearby Points Behave Similarly

Smoothness assumption. Small changes in input produce small changes in output:

$$|f(x) - f(x')| \text{ is small when } \|x - x'\| \text{ is small.}$$

- This rules out highly oscillatory or jagged functions.
- It allows estimation by local averaging and interpolation.
- Kernel methods make smoothness explicit by weighting nearby points:

$$\hat{f}(x) = \sum_i K(x, x_i) y_i$$

Smoothness turns noisy data into a continuous surface.

Sparsity and Additivity: Few Variables Matter

Sparsity assumption. Only a small subset of variables influences the response:

$$f(x) = \sum_{j \in S} \beta_j x_j, \quad |S| \ll p.$$

Additivity assumption. Variables contribute independently without high-order interactions:

$$f(x) = \sum_{j=1}^{p} f_j(x_j).$$

- These assumptions avoid exponential interaction complexity.
- They replace p-dimensional estimation with several 1D problems.
- Many real systems exhibit approximate sparsity or additivity.

Structure replaces dimensional explosion with separability.

Penalization and Basis Expansions

Penalization. We control complexity by discouraging large or many coefficients:

$$\min_f \sum_i (y_i - f(x_i))^2 + \lambda \Omega(f)$$

- Ridge: $\Omega(f) = \|f\|_2^2$ encourages small coefficients.
- Lasso: $\Omega(f) = \|f\|_1$ encourages sparsity.

Basis expansions. Represent f using simple building blocks:

$$f(x) = \sum_k \theta_k \phi_k(x)$$

- Polynomials, splines, Fourier bases, wavelets.
- Complexity is controlled by the number and size of coefficients.

Unsupervised Representations: Learning from X Alone

In unsupervised learning, we observe only inputs X, without responses Y.

Core goal. Discover structure in the distribution of X:

$$P(X) \quad \text{rather than} \quad P(Y \mid X).$$

- Basically, a transformation $Z = g(X)$ that makes structure visible.
- Good representations simplify geometry, similarity, or dependence.
- These representations can later be used for prediction or classification.

Examples.

- **Customer and market analysis:** discovering groups of customers and frequent purchase patterns without predefined categories.
- **Feature learning for prediction:** learning representations from raw data before applying supervised models when labeled data are limited.