Stat 577 "Statistical Learning Theory"

T1: Overview on supervised learning

Xiongzhi Chen

Washington State University

Overview and Some Questions

Some questions: I

1. Why the decision boundary of a nearest-neighbor method is nonlinear? Can it be linear?
2. In terms of capturing the domain of the relationship $\mathbf{y} = f(\mathbf{x})$ for the purpose of predicting $\mathbf{y}$, what is the relationship between a linear regression model and k-nearest neighbor method?
3. How does large dimensionality play an unfavorable role in a linear model or a nearest-neighbor method?
4. What is the "curse of dimensionality" for a nearest-neighbor method? What are the causes for it?
5. Can you describe the interaction between dimensionality, underlying probability distribution, sampling scheme, and a method being used?

Some questions: II

1. Why "learning by example" can be framed as "function approximation"? Why is there a need for approximation?
2. What is the statistical and mathematical meaning of a loss function and of the expected prediction error? What are the advantages and disadvantages of the squared loss?
3. Does it make sense to consider

$$\varepsilon_q(f) = \int \|\mathbf{y} - f(\mathbf{x})\|^q \Pr(d\mathbf{x}, d\mathbf{y}) \text{ where } q \in (0, \infty)$$

   and use the minimizer $\hat{f}$ of $\varepsilon_q(f)$, if it exists, as a procedure?
4. Why correctly balancing the bias-variance trade-off is important? Does the bias-variance decomposition formula provided in the Text hold when the loss is not the integrated squared loss?
5. You should fully understand the following figures in Chapter 2: Figures 2.6, 2.7 and 2.9

Overview

What is "learning"?

- Is learning the process of understanding a physical phenomenon?
- Is learning the process of developing knowledge on a physical phenomenon?
- Is the purpose of learning to make future predictions on or capture the truth underlying a physical phenomenon?

Overview

Depending on whether "training samples" are provided, learning can be roughly categorized into:

- Supervised learning, where a learner is provided examples from what is to be learnt
- Unsupervised learning, where a learner is not provided examples from what is to be learnt

Depending on whether uncertainty is quantified, learning can also be roughly categorized into:

- Statistical learning, where uncertainty quantification is provided by a learner
- Machine learning, where uncertainty quantification is not always provided by a learner

Some methods for supervised learning

Two popular methods for supervised learning:

- The "least squares" method and its variants
    1. Regression models and their regularized versions
- The "nearest neighbors" method and its variants

Questions: Can you provide

1. 3 good properties of least squares method/estimate?
2. 3 not-so-good properties of least squares method/estimate?

Linear Models

Settings for classification

Two-class classification:

- N observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \in \mathbb{R}^{p+1}$, where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$
- $\mathbf{x}_i$ records the ith observation for the same p features, and y_i is the class label for $\mathbf{x}_i$, for which $y_i \in \{0, 1\}$ for all i, i.e., 0 and 1 are the class labels
- For a new observation $(\mathbf{x}, y)$ for which $\mathbf{x}$ is known, estimate its label y, i.e., classify $\mathbf{x}$ into one of the 2 classes

Remark: Training set $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ and the new observation $(\mathbf{x}, y) \notin \mathcal{T}$

Linear model

- Predictor $X^T = (X_1, X_2, ..., X_p)$ and response Y
- Model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

- Its extended version

$$\hat{Y} = \tilde{X}^T \hat{\boldsymbol{\beta}}$$

(What are $\tilde{X}$ and $\hat{\boldsymbol{\beta}}$?)

Least squares method

Using training set $\mathcal{T}$, we obtain:

- Residual sums of squares

$$\text{RSS}(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

- Minimizing $\text{RSS}(\boldsymbol{\beta})$ wrt to $\boldsymbol{\beta}$ gives "normal equation"

$$\mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

- If $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ exits, then unique solution

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T\mathbf{y}$$

  Do you see that $\hat{\boldsymbol{\beta}}$ is a "linear estimator"?

Task: write out $\mathbf{X}$, $\boldsymbol{\beta}$ and derive $\hat{\boldsymbol{\beta}}$

Linear classifier: illustration

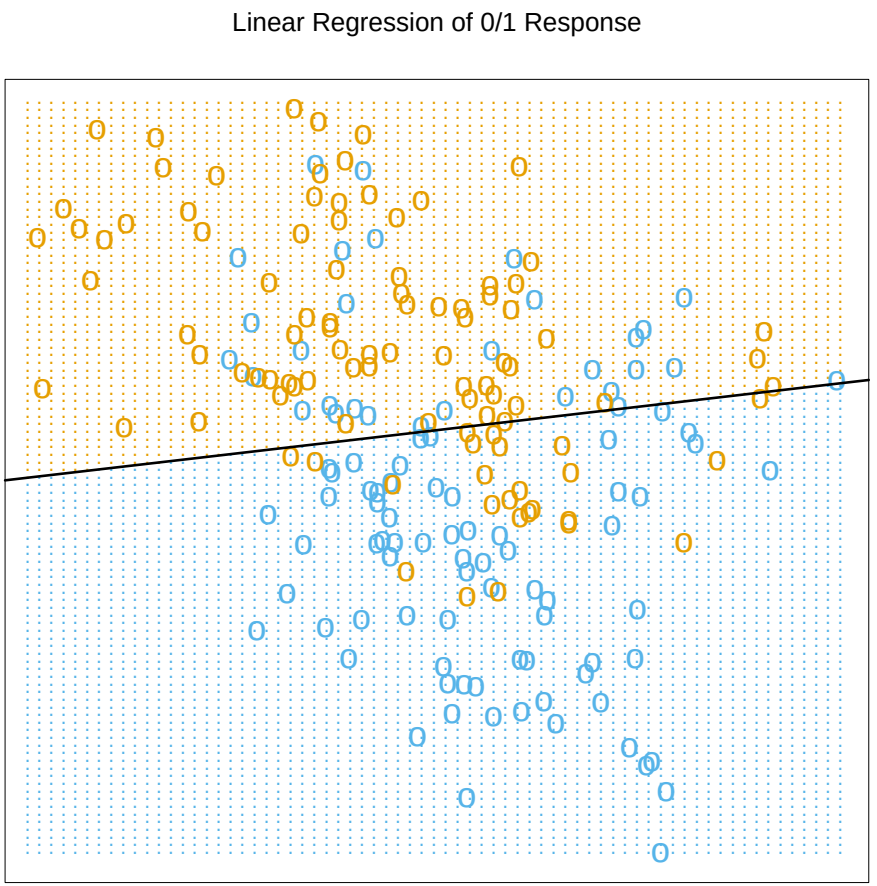


FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (*BLUE $= 0$, ORANGE $= 1$*), and then fit by linear regression. The line is the decision boundary defined by* $x^T \hat{\beta} = 0.5$*. The orange shaded region denotes that part of input space classified as* ORANGE*, while the blue region is classified as* BLUE*.*

Column space

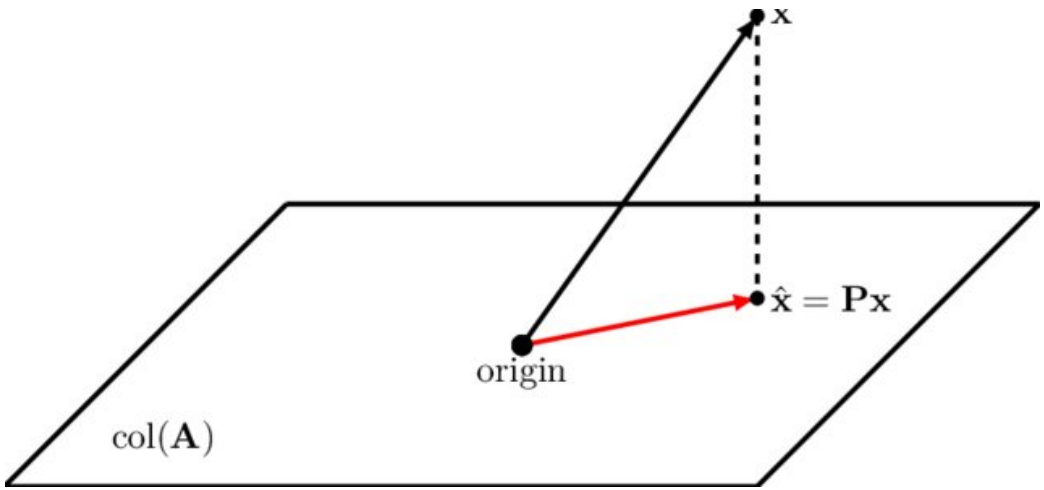
- Let $\mathbf{w}_i, i = 1, ..., c$ be the columns of the matrix $\mathbf{W}_{r\times c}$, then "column space" col $(\mathbf{W})$ of $\mathbf{W}$ is defined as

$$\text{col}\,(\mathbf{W}) = \left\{ \mathbf{w} \in \mathbb{R}^r : \mathbf{w} = \sum_{i=1}^{n} a_i \mathbf{w}_i, a_i \in \mathbb{R} \right\}$$

- col $(\mathbf{W})$ is a vector space itself and a vector subspace of $\mathbb{R}^s$ with $s \geq r$
- col $(\mathbf{W})$ is a closed subset of $\mathbb{R}^s$ with $s \geq r$, where $\mathbb{R}^s$ is endowed with Euclidean distance
- col $(\mathbf{W})$ is a convex subset of $\mathbb{R}^s$ with $s \geq r$, where $\mathbb{R}^s$ is considered a vector space

Geometry behind least squares method

- Orthogonal projection operator: $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$ onto the column space col $(\mathbf{X})$ of $\mathbf{X}$
- $\mathbf{H}$ is idempotent, i.e., $\mathbf{H}^T = \mathbf{H}$ and $\mathbf{HH} = \mathbf{H}$
- Best estimate $\hat{\mathbf{y}} = \mathbf{Hy}$ and solution $\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{y}$, where $\mathbf{A}^-$ denotes Moore–Penrose inverse of matrix $\mathbf{A}$
- Unique orthogonal projection but not always unique solution



"Least squares" in general settings

- Let $\mathbf{V}$ be a closed set in a complete metric space $(\mathbf{M}, d)$ (e.g., a Euclidean space, or some space of functions)
- Let $\mathbf{y}$ be a point in $\mathbf{M}$. Then there exist $\hat{\mathbf{y}} \in \mathbf{V}$ such that

$$\hat{\mathbf{y}} \in \underset{\mathbf{z} \in \mathbf{V}}{\operatorname{argmin}}\, d^2(\mathbf{z}, \mathbf{y}),$$

  and $\hat{\mathbf{y}}$ is unique when $\mathbf{V}$ is also convex

Remark:

1. A vector space is convex
2. $\hat{\mathbf{y}}$ does not need to be a linear function(al) of $\mathbf{y}$
3. Many learning theory and methods can be cast in this general setting!

Nearest Neighbors

Nearest neighbors: algorithm

"K-nearest-neighbor (K-NN)" methods are described as follows:

- Specify a positive integer k as the number of neighboring points for a given observation
- Compute

$$\hat{Y}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} y_i = \text{Ave}\left(y_i | \mathbf{x}_i \in N_k(\mathbf{x})\right),$$

  where $N_k(\mathbf{x})$ is the neighborhood of $\mathbf{x}$ consisting k points $\mathbf{x}_i$ in $\mathcal{T}$ that are closest to $\mathbf{x}$
- Specify a threshold $\beta \in (0,1)$, and classify $\mathbf{x}$ as follows: $\hat{Y}(\mathbf{x}) = 1$ if $\hat{Y}(\mathbf{x}) > \beta$, i.e., $\mathbf{x}$ is classified into class 1 if $\hat{Y}(\mathbf{x}) > \beta$; otherwise $\hat{Y}(\mathbf{x}) = 0$. For example, $\beta = 0.5$ can be used

Nearest neighbors: neighbor size

- How many parameters does k-NN have for a fixed k?
  - It is 1 or N/k, where N is the sample size?
- Let Y be the population label:

$$P(Y = 1) = p \text{ and } P(Y = 0) = 1 - p \text{ with } p \in [0, 1]$$

  - What is $\hat{Y}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} y_i$?
  - What does k-NN estimate?
  - How does k-NN estimate?
- What if different values of k are used?

Vapnik–Chervonenkis dimension

Vapnik–Chervonenkis (VC) dimension:

- Let C be a set and $\mathcal{H} \subseteq \mathcal{P}(C)$, the power set of C
- Definition: C is shattered by $\mathcal{H}$ if

$$|\{H \in \mathcal{H} : H \cap C\}| = 2^{|C|},$$

  i.e., no element of C is not contained in a element of $\mathcal{H}$
- VC dimension D of $\mathcal{H}$ is the largest cardinality of sets shattered by $\mathcal{H}$
  1. Constant classifier f has VC dimension 0
  2. Indicator classifier f on $\mathbb{R}$ has VC dimension 1
- Effective number of parameters for k-NN relates to VC dimension

Nearest neighbors: "closeness" and threshold

- In order to determine "neighborhood" $N_k(\mathbf{x})$ of a point $\mathbf{x}$, a concept of "how close" is needed. How to determine such a concept?
  1. Euclidean distance: $d(\mathbf{u}, \mathbf{v}) = \sqrt{(\mathbf{u}-\mathbf{v})^T (\mathbf{u}-\mathbf{v})}$
  2. Mahalanobis distance: $d(\mathbf{u}, \mathbf{v}) = \sqrt{(\mathbf{u}-\mathbf{v})^T \mathbf{S} (\mathbf{u}-\mathbf{v})}$ with $\mathbf{S} \succ 0$
  3. Hamming distance: $d(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{p} 1_{\{u_i \neq v_i\}} (u_i, v_i)$
- What does "neighborhood" $N_k(\mathbf{x})$ of a point $\mathbf{x}$ look like?
- What if a different threshold $\beta \in (0,1)$ than $\beta = 0.5$ is used, so that $\hat{Y}(\mathbf{x}) = 1$ iff $\hat{Y}(\mathbf{x}) > \beta$?

Illustrations: settings

The following examples are taken from Chapter 2 of the Text:

- Observations are from two classes `ORANGE` or `BLUE`, such that `ORANGE` $= 1$ and `BLUE` $= 0$
- Classifier: KNN for which the class label for $\mathbf{x}$ is `ORANGE` if $\hat{Y}(\mathbf{x}) > 0.5$
- The following can be observed: usually a larger k gives a smoother decision boundary for classification

Illustrations: visualization I

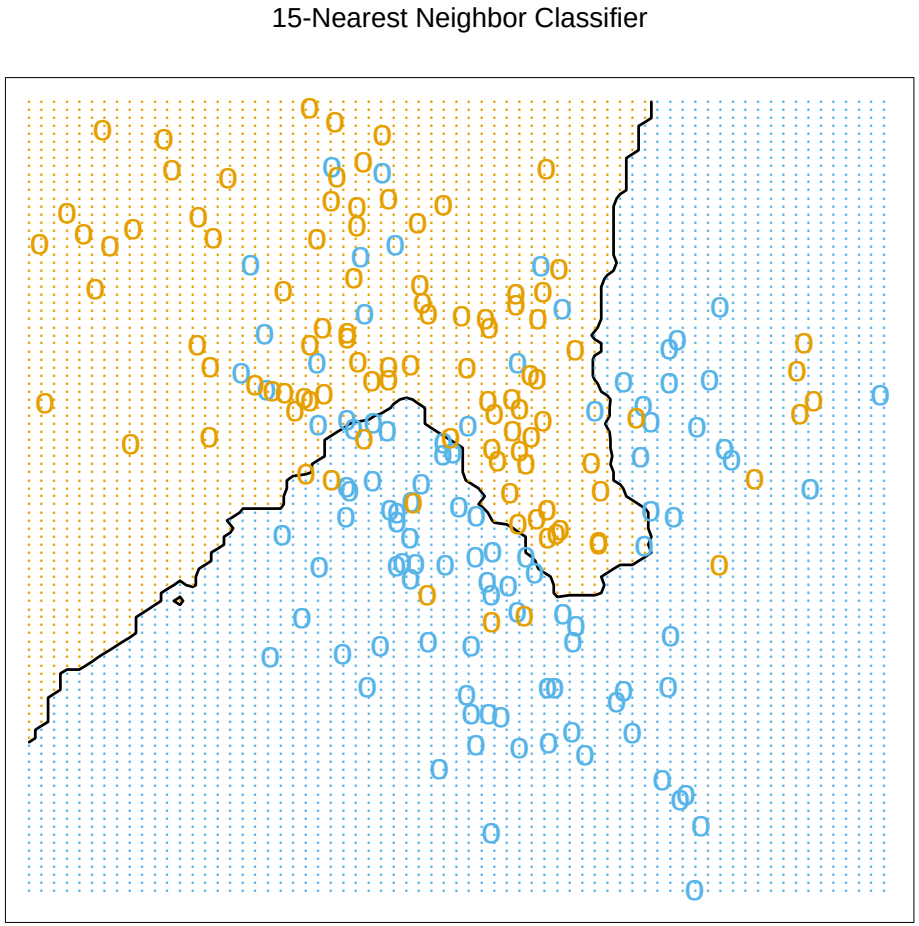


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

Illustrations:: visualization II

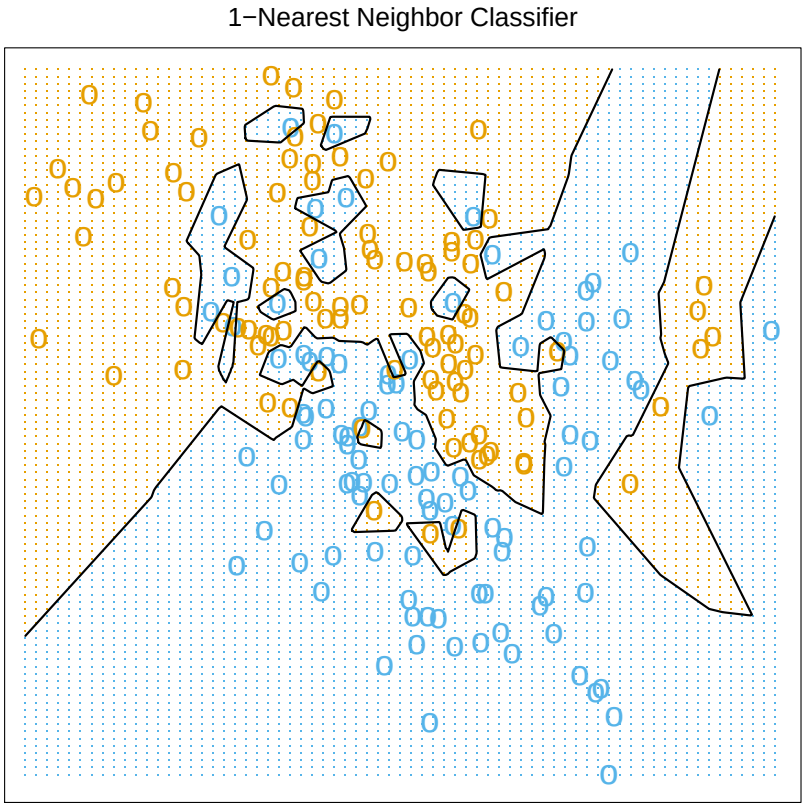


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$)*, and then predicted by* 1*-nearest-neighbor classification.*

Quick comparison

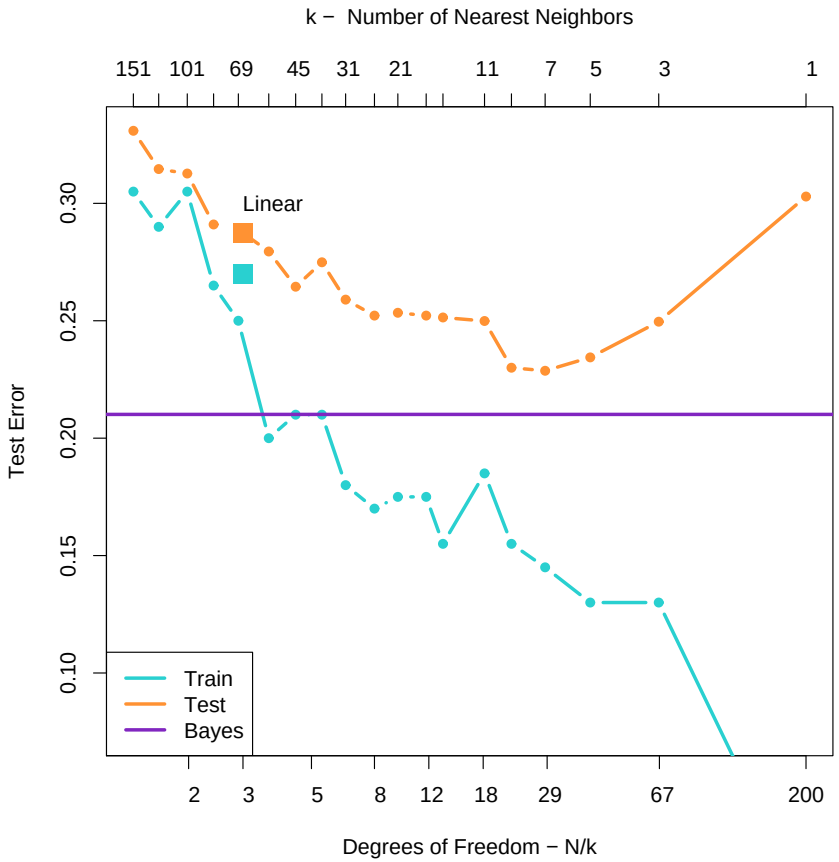


FIGURE 2.4. *Misclassification curves for the simulation example used in Figures 2.1, 2.2 and 2.3. A single training sample of size* 200 *was used, and a test sample of size* $10,000$. *The orange curves are test and the blue are training error for k-nearest-neighbor classification. The results for linear regression are the bigger orange and blue squares at three degrees of freedom. The purple line is the optimal Bayes error rate.*

Curse of Dimensionality

Uniform distribution on hypercube

- ∞-norm: $\|\mathbf{x}\|_\infty = \max_{1\le i\le p} |x_i|$ for $\mathbf{x} = \left(x_1, ..., x_p\right)^T \in \mathbb{R}^p$
- Non-negative hyper-quadrant:

$$\mathbb{R}^p_{\ge 0} = \left\{\mathbf{x} \in \mathbb{R}^p : \min_{1\le i\le p} x_i \ge 0\right\}$$

- p-dimensional unit cube: $C_p = \left\{\mathbf{x} \in \mathbb{R}^p_{\ge 0} : \|\mathbf{x}\|_\infty \le 1\right\}$
- Scaling C_p: for $\lambda \ge 0$, $\lambda C_p = \left\{\mathbf{x} \in \mathbb{R}^p_{\ge 0} : \|\mathbf{x}\|_\infty \le \lambda\right\}$
- Translating λC_p: for any $\mathbf{z} = \left(z_1, ..., z_p\right)^T \in \mathbb{R}^p$,

$$\lambda C_p\left(\mathbf{z}\right) = \left\{\mathbf{x} \in \mathbb{R}^p : |x_i - z_i| \le 2^{-1}\lambda, 1 \le i \le p\right\}$$

- Uniform distribution on hypercube $\lambda C_p\left(\mathbf{z}\right)$:

$$\mu_{\lambda C_p(\mathbf{z})}\left(A\right) = \frac{1}{\lambda^p}\int_A d\mathbf{x} \text{ for all "good" } A \subseteq \lambda C_p\left(\mathbf{z}\right)$$

Hypercube as neighborhood

- The hypercube λC_p has volume λ^p, and if $\lambda = r^{1/p}$, then λC_p covers a fraction r of C_p
- Translating λC_p by any $z \in \mathbb{R}^p$ does not change the volume of λC_p, i.e., volume in Euclidean space is translation-invariant
- Fix $p = 10$, then

r	0.01	0.1
λ	0.63	0.80

- When $\{\mathbf{x}_i\}_{i=1}^N$ follow uniform distribution on C_p, then as dimension p increases, for an $\mathbf{x} \in C_p$ a hypercube neighborhood $N_k(\mathbf{x})$ of size k will have side length $(k/N)^{1/p}$, gradually forcing k-NN to be non-local (since $\lim_{p\to\infty} r^{1/p} = 1$ for any fixed $r \in (0, 1]$) (why?)

Curse of dimensionality on "uniform" unit cube

Uniform distribution of observations + Increasing dimension + Restricted neighborhood shape $\Rightarrow$ Curse of dimensionality for k-NN

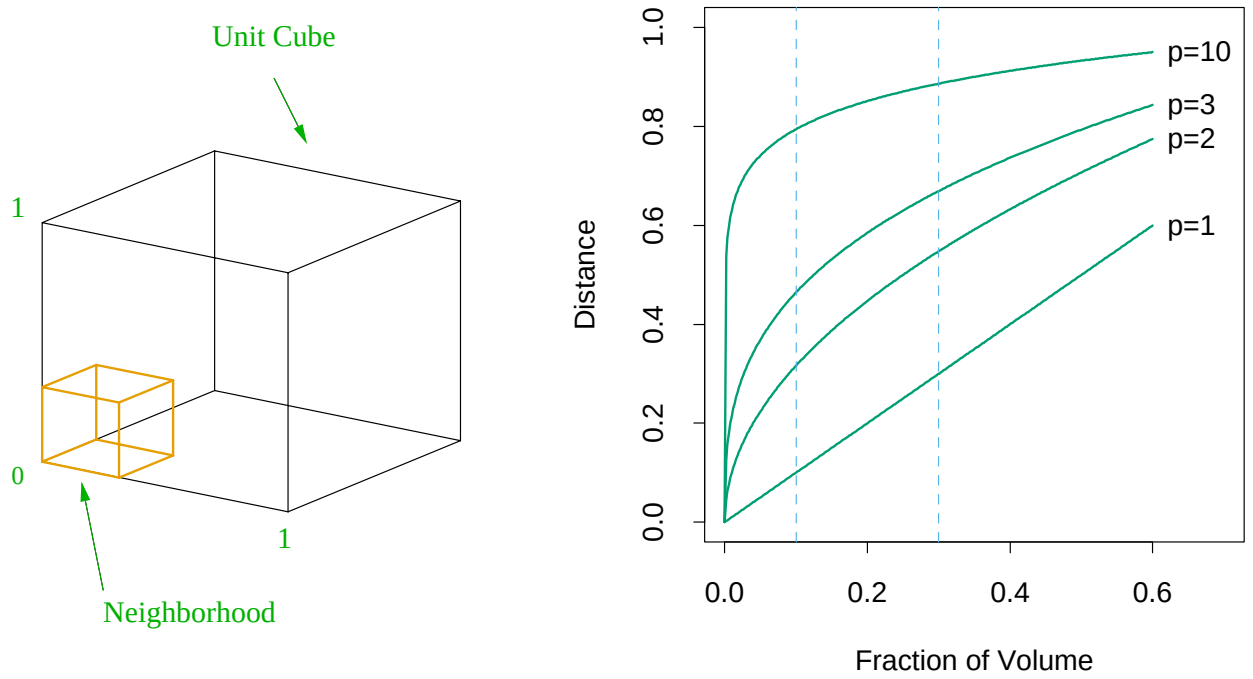


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

Uniform distribution on a ball

- p-dimensional ball of radius r: $B_p(r) = \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x}\| \leq r\}$
- $(p-1)$-dim. sphere of radius r: $S_p(r) = \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x}\| = r\}$
- p-dim. unit ball $B_p = B_p(1)$ and $(p-1)$-dim. unit sphere $S_p = S_p(1)$
- Uniform distribution on $B_p(r)$:

$$\mu_{B_p(r)}(A) = \frac{1}{v_p(r)} \int_A d\mathbf{x} \text{ for all "good" } A \subseteq B_p(r),$$

where volume of $B_p(r)$ is

$$v_p(r) = \frac{\pi^{p/2} r^p}{\Gamma(1 + 2^{-1}p)}$$

Ball as neighborhood

- If data points $\{\mathbf{z}_i\}_{i=1}^N$ are i.i.d. and follow μ_{B_p}, then the median of distance from 0 to the closest data point is

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

- So, $B_p(a)$ with $a \geq 0.5$ likely contains data points, and hence $N_k(0)$ for $\mathbf{x} = 0$ should contain $B_p(0.5)$ in order to contain at least one $\mathbf{z}_i$ in order to be a sensible neighborhood for k-NN
- **Proof sketch** for the identity (1):
  1. $Z \sim \mu_{B_p}$ implies the density f of Z is $f(\mathbf{z}) = 1/v_p(1)$. So,

$$P(d(Z,0) \leq s) = P(\|Z\| \leq s) = v_p(s)/v_p(1) = s^p$$

Median of closest distance in "uniform" unit ball

- 2. Notice $\delta_{\min,N} = \min_{1\le i\le N} \|\mathbf{z}_i\| = \min_{1\le i\le N} d(\mathbf{z}_i, 0)$. For i.i.d. $\mathbf{z}_i \sim Z \sim \mu_{B_p}$, then

$$G(s) = P(\delta_{\min,N} \le s) = 1 - (1 - s^p)^N;$$

set $G(s_0) = 1/2$ and $s_0 = \delta(p, N) = \left(1 - 2^{-1/N}\right)^{1/p}$. (Interpret s_0)

- E.g., $\delta(10, 500) \approx 0.52$. So, 50% of times there is no observation in any $B_p(r)$ for all $0 \le r < s_0$, leading to empty $N_k(\mathbf{x})$ for $\mathbf{x}$ close to the origin (Why?)
- Recall $\mathcal{T}$ as the training set. Then for larger p, prediction near the boundary $\partial\mathcal{T}$ of $\mathcal{T}$ becomes harder (Why?) and extrapolation from neighboring points (rather than interpolation between them) is needed

Example 1

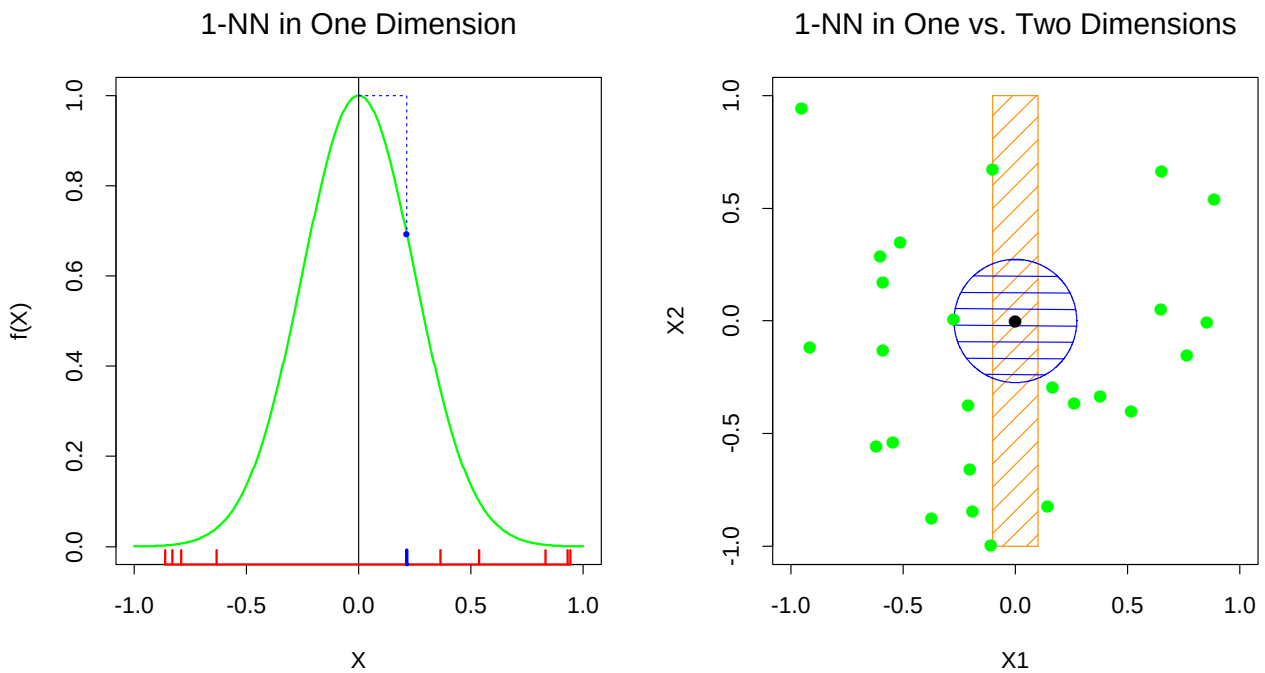
- $N = 1000$ training samples $\mathbf{x}_i$ generated uniformly on $[-1, 1]^p$;
- True relationship between X and Y is $Y = f(X) = e^{-8\|X\|^2}$ without measurement error
- Use 1-NN to predict y_0 at test point $\mathbf{x}_0 = 0$ by $\hat{y}_0$
- Let $\mathcal{T}$ be training set. Mean squared error (MSE) for estimating $f(0)$:

$$\text{MSE}(\mathbf{x}_0) = E_{\mathcal{T}}\left[(f(\mathbf{x}_0) - \hat{y}_0)^2\right] = \text{Var}_{\mathcal{T}}(\hat{y}_0) + \text{Bias}^2_{\mathcal{T}}(\hat{y}_0),$$

the bias-variance decomposition; $\text{Var}_{\mathcal{T}}(\hat{y}_0)$ is due to sampling variance of 1-NN

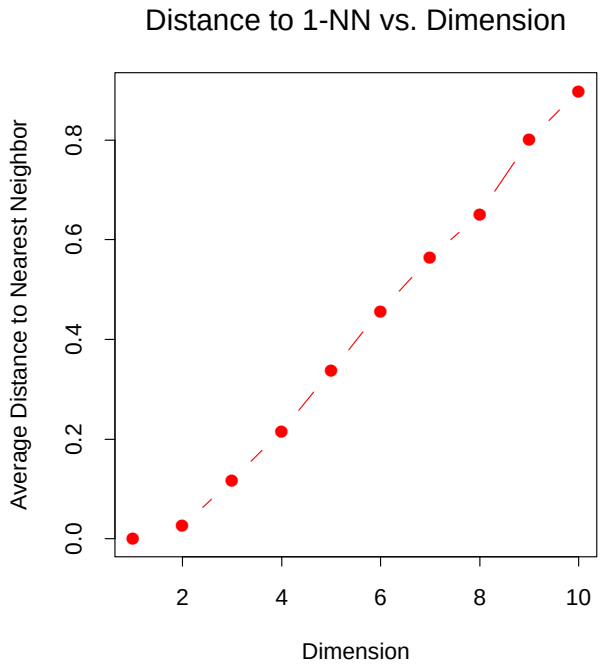
Example 1: visualization

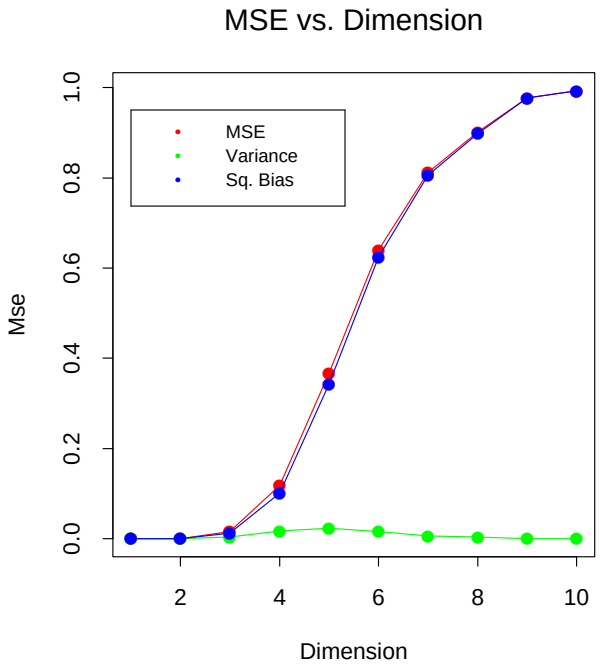
- Left figure: target function; training sample indicated by blue tick mark; 1-NN has downward bias (why?)
- Right figure: for any $\mathbf{x}$, radius of $N_1(\mathbf{x})$ increases with dimension p (intuitively why?); for $p = 10$, for more than 99% of samples the nearest neighborhood is at distance ≥ 0.5 from origin



Example 1: visualization

- Left figure: average radius of 1-NN neighborhoods
- Right figure: MSE, squared bias, and variance curves; squared bias dominates variance (why? Hint: $Y = f(X) = e^{-8\|X\|^2}$)
- As p increases, 1-NN estimate of $f(0)$ tends to 0 more often than not (why?)





Example 2: effective dimension

- Effective dimension

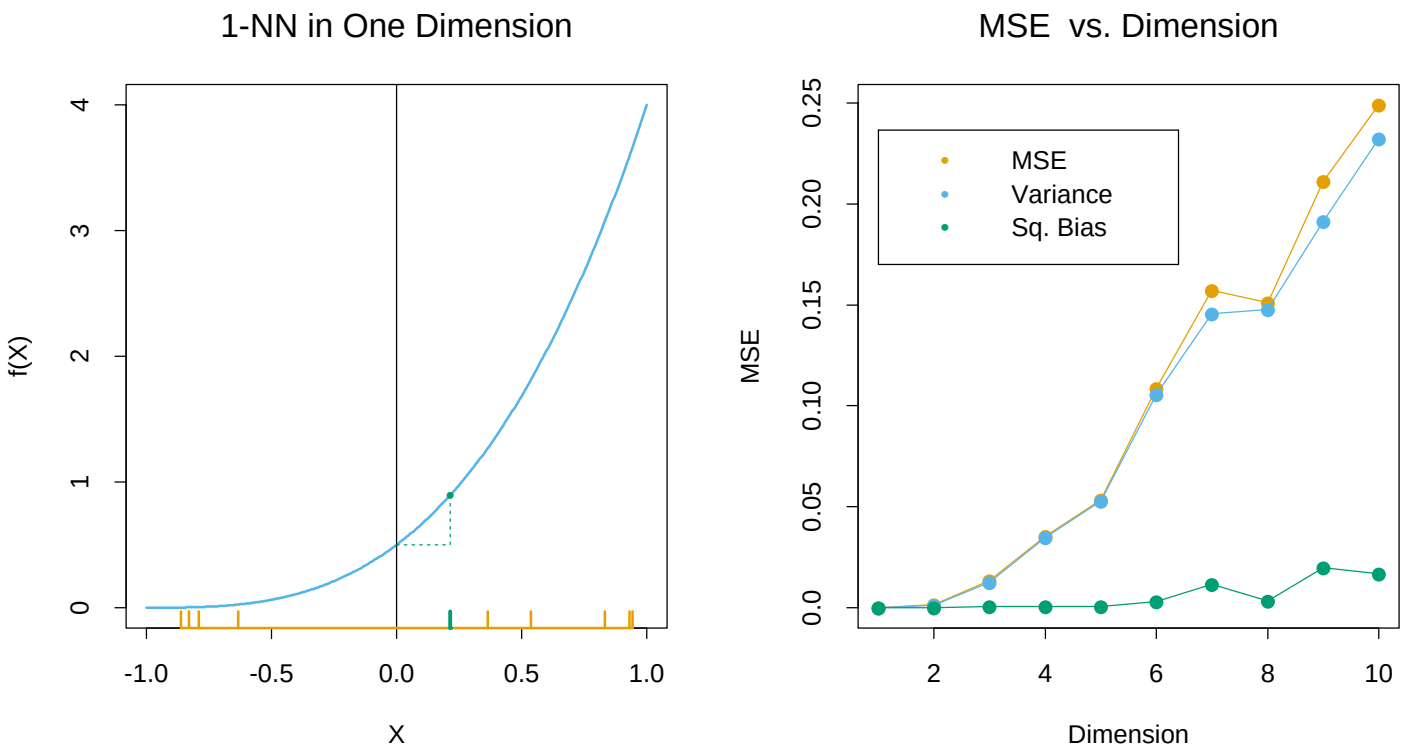


FIGURE 2.8. *A simulation example with the same setup as in Figure 2.7. Here the function is constant in all but one dimension:* $F(X) = \frac{1}{2}(X_1 + 1)^3$. *The variance dominates.*

Example 2

- $N = 1000$ training samples $\mathbf{x}_i$ generated uniformly on $[-1, 1]^p$;
- True relationship between X and Y is $Y = X^T\beta + \varepsilon$ with $\varepsilon \sim N\left(0, \sigma^2\right)$
- Use least squares to obtain $\hat{\beta}$. For any test point $\mathbf{x}_0$, estimator

$$\hat{y}_0 = \mathbf{x}_0^T\hat{\beta} = \mathbf{x}_0^T\beta + \sum_{i=1}^{N} l_i\left(\mathbf{x}_0\right)\varepsilon_i,$$

  where $l_i\left(\mathbf{x}_0\right)$ is the ith element of $\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{x}_0$
- Expected squared prediction error (ESPE):

$$\begin{aligned}
\text{ESPE}\left(\mathbf{x}_0\right) &= E_{y_0|\mathbf{x}_0}\left[\left(y_0 - \hat{y}_0\right)^2\right] \\
&= \text{Var}\left(y_0|\mathbf{x}_0\right) + \text{Var}_{\mathcal{T}}\left(\hat{y}_0\right) + \text{Bias}_{\mathcal{T}}^2\left(\hat{y}_0\right) \\
&= \sigma^2 + \sigma^2 E_{\mathcal{T}}\left(\mathbf{x}_0^T\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{x}_0\right) + 0^2
\end{aligned}$$

Example 2: role of dimensionality

- $\text{ESPE}(\mathbf{x}_0) = \sigma^2 + \sigma^2 E_{\mathcal{T}}\left(\mathbf{x}_0^T \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{x}_0\right) + 0^2$
- If N is large, $\mathcal{T}$ were selected at random, $E(X) = 0$, then $\mathbf{X}^T\mathbf{X} \to N\text{Cov}(X)$ and

$$\begin{aligned} E_{\mathbf{x}_0}\left[\text{ESPE}(\mathbf{x}_0)\right] &\sim \sigma^2 N^{-1} E_{\mathbf{x}_0}\left(\mathbf{x}_0^T \text{Cov}(X)^{-1}\mathbf{x}_0\right) + \sigma^2 \\ &= \sigma^2 N^{-1}\text{trace}\left(\text{Cov}(X)^{-1}\text{Cov}(X)\right) + \sigma^2 \\ &= \frac{p\sigma^2}{N} + \sigma^2 \end{aligned}$$

Example 2: visualization

- $Y = f(X) + \varepsilon$, where X is uniform and $\varepsilon \sim N(0, 1)$; $N = 500$
- y-axis: ESPE of 1-NN over ESPE of OLS

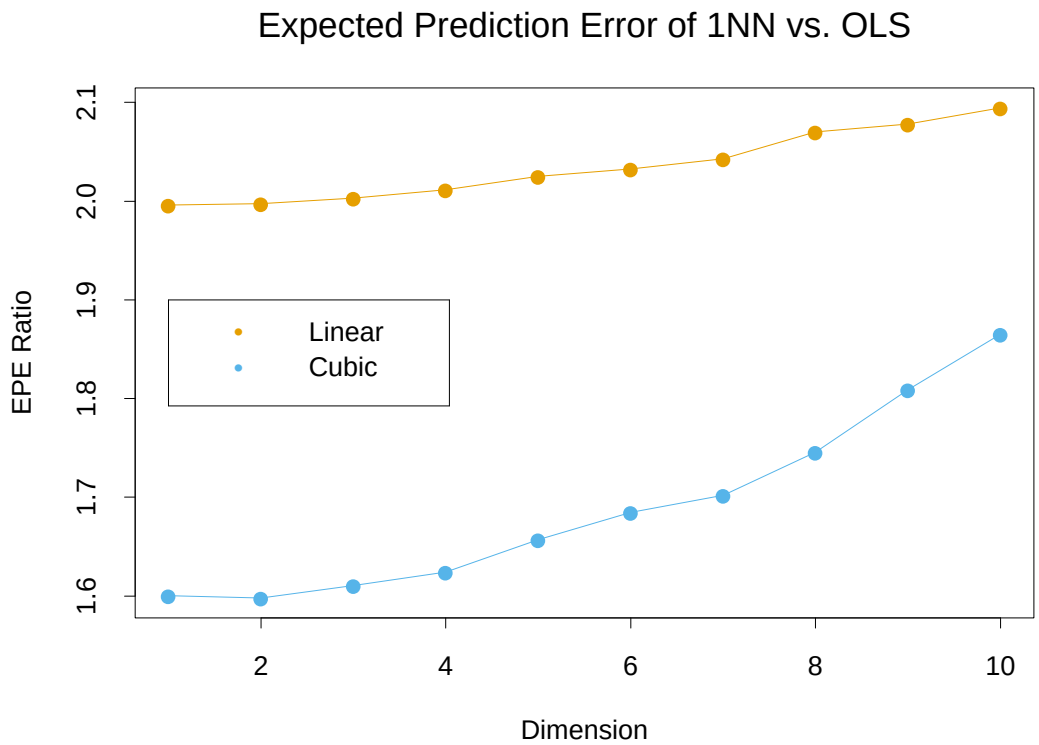


FIGURE 2.9. *The curves show the expected prediction error (at $x_0 = 0$) for 1-nearest neighbor relative to least squares for the model $Y = f(X) + \varepsilon$. For the orange curve, $f(x) = x_1$, while for the blue curve $f(x) = \frac{1}{2}(x_1 + 1)^3$.*

Flexible models in high dimensions

- Rigid assumptions give a linear model unbiasedness and negligible variance, while the ESPE of 1-NN (without any assumptions at all) is substantially larger. However, if assumptions are wrong, 1-NN may dominate linear model
- There is a whole spectrum of models between rigid linear models and extremely flexible 1-NN models, each with their own assumptions and biases
- Relying heavily on specific assumptions, these intermediate models can avoid the exponential growth in complexity of employed modelling functions in high dimensions

Bayes Classifier

Bayes classifier: settings

For two-class classification:

- Population label $Y \sim Bernoulli\,(p)$ with $P\,(Y=1)=p$
- Posterior probability:

$$P\,(Y|X=\mathbf{x})=\frac{P\,(X=\mathbf{x}|Y)\,P\,(Y)}{P\,(X=\mathbf{x})}$$

- Bayes classifier:

$$\hat{Y}\,(\mathbf{x})=\underset{g\in\{0,1\}}{\operatorname{argmax}}\,P\,(Y=g|X=\mathbf{x})$$

Remark: multi-class classification employs multi-nomial distribution but same classification rule

Bayes classifier: illustration

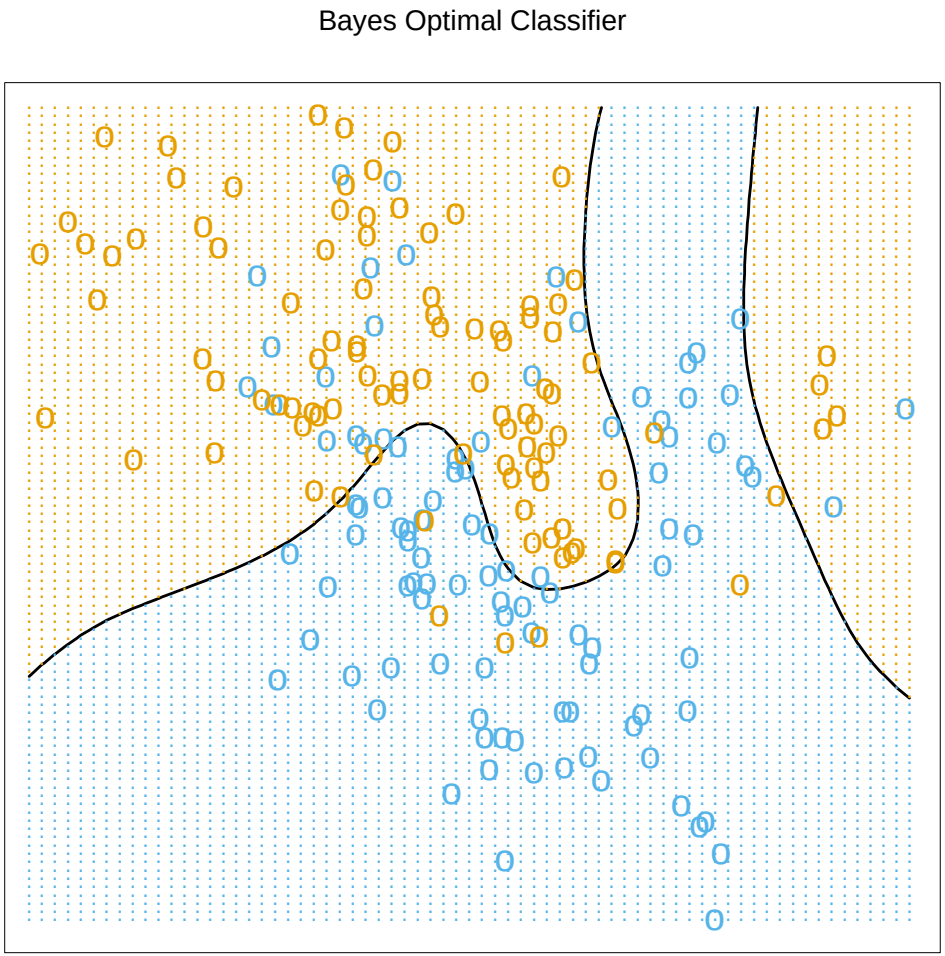


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

Bayes classifier: optimality

Bayes classifier:

- Is Bayes classifier optimal? If so, in what sense?
- In practice can we fully implement Bayes classifier and achieve its optimality?
- How is Bayes classifier related to and different from k-NN classifier and linear-regression classifier?

Formulation of Learning

General formulation via loss

- Covariate vector $X \in \mathbb{R}^p$ and response $Y \in \mathbb{R}$
- Probabilistic setup: (X, Y) has joint distribution $\Pr(X, Y)$
- Learning: capture $\Pr(X, Y)$
- Predictive Learner f: $f(X)$ to predict Y
- Loss L: $L(Y, f(X))$; e.g., squared loss $L(Y, f(X)) = (Y - f(X))^2$
- Expected prediction error (EPE):

$$\mathrm{EPE}(f, L) = E(L(Y, f(X)))$$

and for squared loss

$$\mathrm{EPE}_2(f) = \int (y - f(\mathbf{x}))^2 \Pr(d\mathbf{x}, dy)$$

Loss function

- Loss L is a nonnegative function(al), chosen by user, and defined on general metric space (e.g., Y does not need to be a scalar)
- L^p-loss on Euclidean spaces:

$$L_p\left(Y, f\left(X\right)\right) = \left\|Y - f\left(X\right)\right\|^p, p > 0,$$

  e.g., when $Y \in \mathbb{R}^q$
- L^1-loss is more robust to outliers than L_2-loss
- For classification: L_0-loss or 0-1 loss

$$L\left(y, f\left(\mathbf{x}\right)\right) = \begin{cases} 0 \text{ if } f\left(\mathbf{x}\right) = y \\ 1 \text{ if } f\left(\mathbf{x}\right) \neq y \end{cases} \quad \text{for observation } (\mathbf{x}, y)$$

- Expected prediction error (EPE) is unknown and needs to be estimated

Minimizer under a loss

For expected prediction error (EPE)

$$\mathrm{EPE}_p(f) = \int \|y - f(\mathbf{x})\|^p \Pr(d\mathbf{x}, dy)$$

- Under L_2-loss, the unique minimizer is

$$f_{2,\mathrm{opt}}(\mathbf{x}) = E(Y|X = \mathbf{x})$$

- Under L_1-loss, a minimizer is

$$f_{1,\mathrm{opt}}(\mathbf{x}) = \mathrm{Median}(Y|X = \mathbf{x})$$

Caution: median is not always unique or defined

Minimizer under squared loss

- For EPE under L_2-loss, the unique minimizer is

$$f_{2,\text{opt}}(\mathbf{x}) = E(Y|X = \mathbf{x})$$

- Intuition:
  1. The space $\mathcal{H}$ of random variables with finite variances form a complete metric space under metric

$$d_2(Z_1, Z_2) = \sqrt{E\left(\|Z_1 - Z_2\|^2\right)}$$

  2. Let $\mathcal{H}_X$ be the set of random variables that are (measurable) functions of X. Then $\mathcal{H}_X$ is a linear subspace of $\mathcal{H}$
  3. Since $E(Y|X)$ is the orthogonal projection of Y onto $\mathcal{H}_X$, it minimizes $\text{EPE}_2(f)$

Minimizer under squared loss

Theorem: $\hat{f}(\mathbf{x}) = E(Y|X = \mathbf{x})$ minimizes $\mathrm{EPE}_2(f)$

- Since $\Pr(X, Y) = P(Y|X)\,P(X)$, then

$$\mathrm{EPE}_2(f) = \int [y - f(\mathbf{x})]^2 \Pr(d\mathbf{x}, dy) = E_X(E_{Y|X}[(Y - f(X))^2|X])$$

- However,

$$E_{Y|X}[(Y - f(X))^2|X] \geq E_{Y|X}[(Y - E(Y|X))^2|X]$$

since $E_{Y|X}([Y - E(Y|X)][E(Y|X) - f(X)]|X) = 0$ and

$$E_{Y|X}\left([Y - f(X)]^2|X\right) = E_{Y|X}[(Y - E(Y|X) + E(Y|X) - f(X))^2|X]$$
$$= E_{Y|X}([Y - E(Y|X)]^2|X) + E_{Y|X}([E(Y|X) - f(X)]^2|X)$$

Pythagorean theorem

- Squared length of random vector X as $\|X\|_2 = E\left(\|X\|^2\right)$
- Let $h(X) = E(Y|X)$. Then

$$\|Y - f(X)\|_2^2 = \|Y - h(X)\|_2^2 + \|h(X) - f(X)\|_2^2$$

  for any measurable f of X
- Equivalently, $E\{[Y - h(X)][h(X) - f(X)]\} = 0$ for any measurable f of X, i.e.,

$$Y - h(X) \perp h(X) - f(X)$$

Strict convexity of squared 2-norm

- Fact: squared 2-norm is strictly convex
- Proof: (1) $\|x\|_2 \neq \|y\|_2$ and $\lambda \in (0,1)$
  $\Longrightarrow \lambda(1-\lambda)(\|x\|_2 - \|y\|_2)^2 > 0$. Expand and rearrange terms to obtain

$$(\lambda \|x\|_2 + (1-\lambda)\|y\|_2)^2 < \lambda \|x\|_2^2 + (1-\lambda)\|y\|_2^2$$

- (2) Holder's inequality:

$$\|fg\|_1 \leq \|f\|_p \|g\|_q \quad \text{for } p,q \in [1,\infty] \text{ and } \frac{1}{p} + \frac{1}{q} = 1$$

  implies Minkowski inequality

$$\|f+g\|_p \leq \|f\|_p + \|g\|_p \quad \text{for } \; p \in [1,\infty]$$

- (2) Combine these to get

$$\|\lambda x + (1-\lambda)y\|_2^2 \leq (\lambda \|x\|_2 + (1-\lambda)\|y\|_2)^2 < \lambda \|x\|_2^2 + (1-\lambda)\|y\|_2^2$$

Median not unique

- Let Z have CDF F with median c, then

$$P(Z \geq c) \geq \frac{1}{2} \text{ and } P(Z \leq c) \geq \frac{1}{2}$$

- But $F(c) = P(Z \leq c)$ and

$$F(c) = 1 - P(Z > c) = 1 - P(Z \geq c) + P(Z = c)$$

  implies

$$\frac{1}{2} \leq F(c) \leq \frac{1}{2} + P(Z = c)$$

- So, median is any number that satisfies the above inequality

Minimizer under absolute deviation loss

Theorem: $\hat{f}(\mathbf{x}) = \text{Median}(Y|X = \mathbf{x})$ minimizes $\text{EPE}_1(f)$

- Minimizing $g(X) = E_{Y|X}(|Y - f(X)||X)$ leads to minimizing

$$\text{EPE}_1(f) = \int |Y - f(X)| P(d\mathbf{x}, dy) = E_X[E_{Y|X}(|Y - f(X)||X)]$$

- But $g(\mathbf{x})$ is minimized by $\hat{f}(\mathbf{x}) = \text{Med}(Y|X = \mathbf{x})$ since:
  1. Let $m = \text{Median}(W)$. Then $P(W \leq m) \geq 1/2$ and $P(W \geq m) \geq 1/2$
  2. Let $\delta = |W - c| - |W|$. If $m = 0$, then for $c > 0$

$$\begin{cases} E[\delta 1_{\{W \leq 0\}}(W)] = cP(W \leq 0); \\ E[\delta 1_{\{W > 0\}}(W)] \geq -cP(W > 0); \\ E(\delta) \geq c[P(W \leq 0) - P(W > 0)] = c[2P(W \leq 0) - 1] \geq 0 \end{cases}$$

  and $0 = \text{argmin}_{c \geq 0} E(|W - c|)$; if $m = 0$ and $c < 0$, then consider $-W$ and $-c$ in δ; if $m \neq 0$, then consider $W - m$
  3. Thus,

$$\text{Median}(W) = \underset{c \in \mathbb{R}}{\text{argmin}}\, E(|W - c|)$$

Minimizer under 0-1 loss

- Class label $Y \in \mathcal{G} = \{1, 2, \ldots, K\}$, classifier $\hat{Y}(X)$ and 0-1 loss L
- Class $\mathcal{G}_k$ contains observations whose class label is k
- EPE under L_0-loss is

$$\text{EPE}_0(f) = E\left[L\left(Y, \hat{Y}(X)\right)\right] = E_X\left(\sum_{k=1}^{K} L\left(\mathcal{G}_k, \hat{Y}(X)\right) P\left(\mathcal{G}_k | X\right)\right)$$

- Minimization pointwise:

$$\hat{G}(\mathbf{x}) = \underset{g \in \mathcal{G}}{\text{argmin}} \sum_{k=1}^{K} L\left(\mathcal{G}_k, g\right) P\left(\mathcal{G}_k | X\right) = \underset{g \in \mathcal{G}}{\text{argmin}} \left(1 - P\left(Y = g | X\right)\right)$$

or

$$\hat{f}_{0,\text{opt}}(\mathbf{x}) = \underset{g \in \{1,\ldots,K\}}{\text{argmax}} P\left(Y = g | X = \mathbf{x}\right)$$

Summary on three methods

- Regression methods: $\hat{Y}(\mathbf{x}) = E(Y|X = \mathbf{x})$
  - Simple linear model (global methods): $\hat{Y}(\mathbf{x}) = \mathbf{x}^T\hat{\beta}$ with

$$\hat{\beta} = \left[E\left(XX^T\right)\right]^{-1} E(XY) \quad \text{(data version: } \hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}\text{)}$$

- k-NN methods (local methods):

$$\hat{Y}(\mathbf{x}) = \text{Ave}\left(y_i|\mathbf{x}_i \in N_k(\mathbf{x})\right),$$

  converging to $E(Y|X = \mathbf{x})$ as $N, k \to \infty$ and $k/N \to 0$
- Bayes method: $\hat{Y}(\mathbf{x}) = \text{argmax}_{g\in\{1,\ldots,K\}} P(Y = g|X = \mathbf{x})$

Remark: Essentially all three methods estimate $E(Y|X = \mathbf{x})$. Since we can estimate both $\Pr(Y)$ and $\Pr(X)$, statistical learning essentially estimates $\Pr(X, Y)$

Conceptual Framework of Learning

Supervised learning for prediction

Supervised learning for predictive modelling:

- Training set $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}^q$ (and $(\mathbf{x}_i, y_i) \in \mathbb{R}^{p+q}$)
- Unknown underlying relationship $f : X \mapsto Y$ generates $\mathcal{T}$
- Learner $\hat{f}$ learns f based on $\mathcal{T}$, and predicts y for a test point $\mathbf{x}$ for $(\mathbf{x}, y) \notin \mathcal{T}$
- Three example learners:
  - Simple linear model; k-NN methods; Bayes classifier

  and roughly speaking, they all estimate $g(\mathbf{x}) = E(Y|X = \mathbf{x})$
- Ultimate goal: if we can estimate $\Pr(X, Y)$, then we can make accurate predictions

Balance accuracy and complexity

In general, $\Pr(X, Y)$ may be very complicated ...

- Dealing with full generality may not be realistic; e.g., family of candidate models not amenable for algorithmic implementation or theoretical analysis
- Structural assumptions: if special structure is known to exist, we can reduce both bias and variance of estimates of a learner
  - Additive error (i.e., additive model): $Y = f(X) + \varepsilon$ with $E(\varepsilon) = 0$ for numerical response Y, which implies

$$E(Y|X = \mathbf{x}) = f(\mathbf{x})$$

  - Additive model for numerical response: the squared loss L_2 is a natural performance measure, and accordingly the least squares method for implementation

Balance accuracy and complexity

In general, $\Pr(X, Y)$ may be very complicated ...

- For categorical response Y, additive model

$$Y = f(X) + \varepsilon \text{ with } E(\varepsilon) = 0$$

  is usually insensible, and we model $g(\mathbf{x}) = E(Y|X = \mathbf{x})$
- Regardless, the model

$$g(\mathbf{x}) = E(Y|X = \mathbf{x})$$

  subsumes additive model, and can still be quite complicated
- So, additional structural assumptions are needed on g. This leads to functional approximation, structured regression, and model selection

Function approximation

To estimate $g(\mathbf{x}) = E(Y|X = \mathbf{x})$ (with $\mathbf{x} = (x_1, ..., x_p)^T$):

- Semi-parametric models:

$$g_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{k=1}^{K} h_k(\mathbf{x}) \theta_k, \tag{2}$$

  where $\mathcal{H}_0 = \{h_k \in \mathcal{F} : k \in \mathbb{N}\}$ and $\mathcal{F}$ is a set of suitable functions and $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)^T$
  - Parametric models (simple linear models): $g(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\beta}$ where $\boldsymbol{\beta} = \boldsymbol{\theta}$ and $h_k(\mathbf{x}) = x_k$
  - Parametric models (logistic regression): $h_k(\mathbf{x}) = 1/\left(1 + \exp\left(-\mathbf{x}^T \boldsymbol{\beta}_k\right)\right)$
  - Parametric models (polynomial regression): $h_k(\mathbf{x}) = x_k^k$
- Non-parametric models: NN methods and some kernel density estimation methods

Implementations and algorithms

- For numerical response Y and model (2), least squares can be applied to minimize $\text{RSS}(\boldsymbol{\theta}) = \sum_{i=1}^{N} (y_i - g_{\boldsymbol{\theta}}(\mathbf{x}_i))^2$ to estimate $\boldsymbol{\theta}$
- In general settings where $\{y_i\}_{i=1}^{N}$ are i.i.d. from Y with density $\Pr_{\boldsymbol{\theta}}(\cdot)$, "maximum likelihood principle" can be applied to maximize wrt $\boldsymbol{\theta}$ the joint density/likelihood

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \Pr_{\boldsymbol{\theta}}(y_i),$$

where $\Pr_{\boldsymbol{\theta}}(Y)$ is the density of Y.

1. The above principle is equivalent to the least squares for linear models with Gaussian error
2. The above likelihood is equivalent to cross entropy when Y is multinomial for classification

Structured regression

Estimate $g(\mathbf{x}) = E(Y|X = \mathbf{x})$ (with $\mathbf{x} = (x_1, ..., x_p)^T$):

- There may be infinitely many solutions $\hat{f}$ to

$$\text{minimizing RSS}(f) = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 \quad \text{wrt } f \in \mathcal{F},$$

  none with good predictive performance
- Regularization with penalty, i.e., function(al) J to incorporate structural information
  - Penalized RSS: $\text{PRSS}(f; \lambda, J) = \text{RSS}(f) + \lambda J(f)$
  - Penalized likelihood: $\tilde{L}(\boldsymbol{\theta}; J) = L(\boldsymbol{\theta}) + \lambda J(\boldsymbol{\theta})$
- The penalty/regularizer J needs to incorporate structural information and balance model complexity and predictive performance

Bia-variance Trade-off

Classic paradigm for bias-variance trade-off

- Single descent; usually for small models

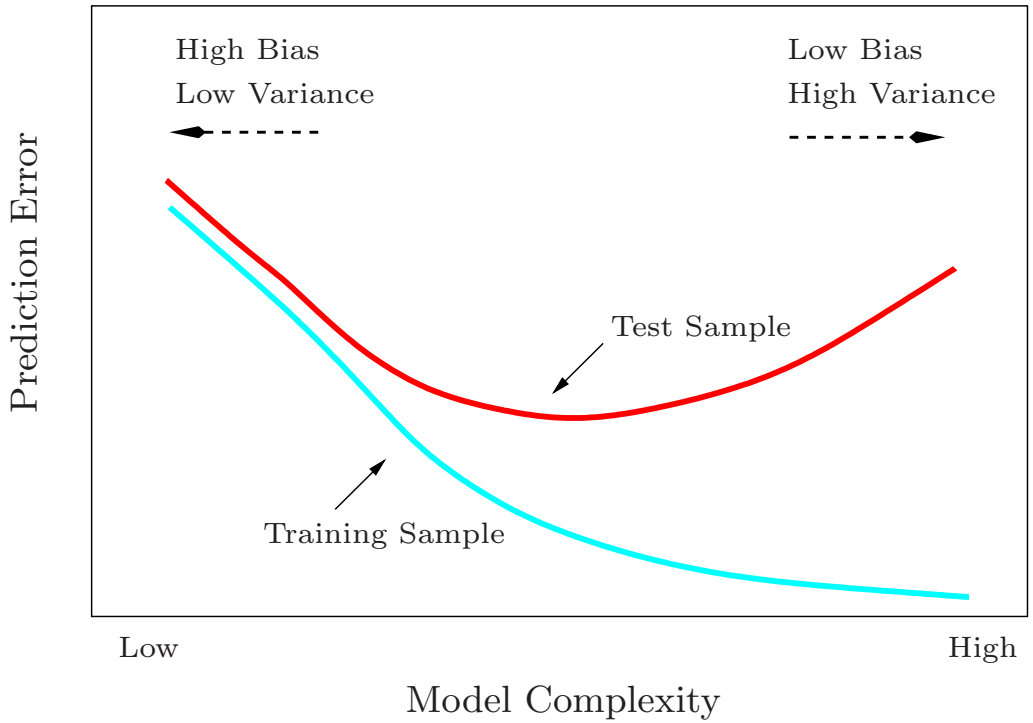
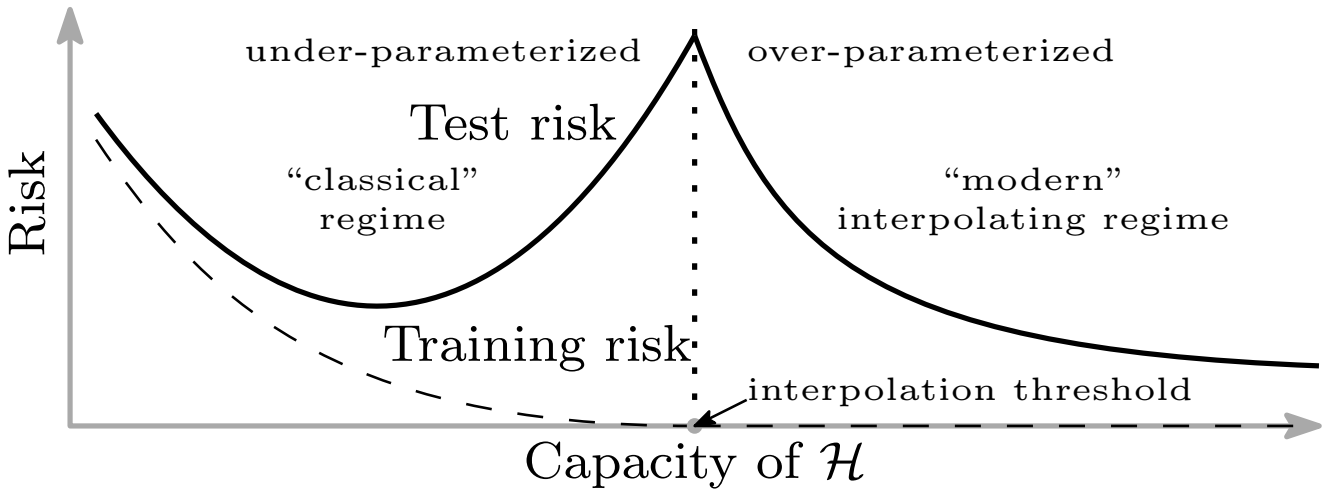


FIGURE 2.11. *Test and training error as a function of model complexity.*

Modern paradigm for bias-variance trade-off

- **Double descent**; usually for big models
- $\mathcal{H}$: family of functions employed by models
- Credit: https://www.pnas.org/content/116/32/15849



Modern paradigm for bias-variance trade-off

- **Triple descent**: usually for big models
- P number of parameters; N sample size; D dimension of inputs
- Credit: https://arxiv.org/abs/2006.03509

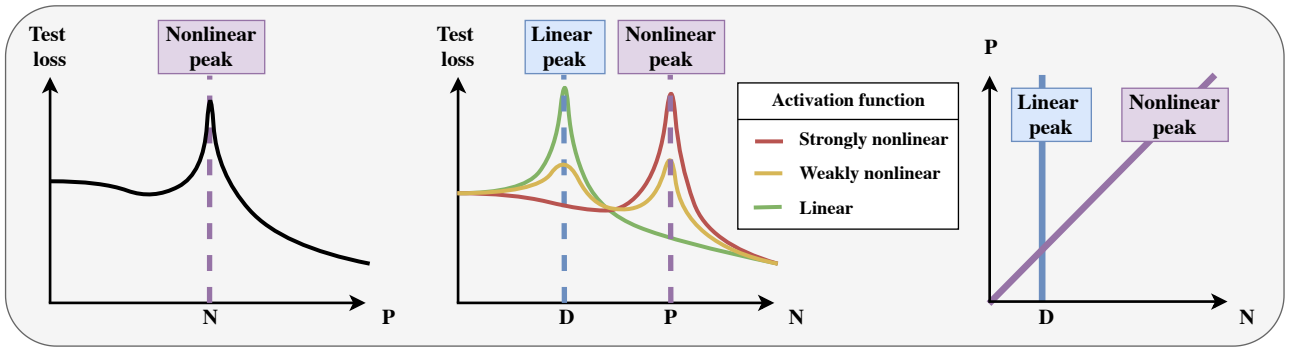


Figure 1: **Left:** The parameter-wise profile of the test loss exhibits double descent, with a peak at $P = N$. **Middle:** The sample-wise profile can, at high noise, exhibit a single peak at $N = P$, a single peak at $N = D$, or a combination of the two (triple descent[2]) depending on the degree of nonlinearity of the activation function. **Right:** Color-coded location of the peaks in the (P, N) phase space.