Optimization and Machine Learning SI151

Lu Sun

School of Information Science and Technology

ShanghaiTech University

March 4, 2020

Today:

- Overview of supervised learning II
    - Statistical decision theory
    - Local methods in high dimensions
    - Statistical models
    - Model selection

Readings:

- The Element of Statistical Learning, Chapters 1 and 2
- Pattern Recognition and Machine Learning, Chapter 1

Overview of Supervised Learning II

--- Statistical Decision Theory

Statistical Decision Theory

- Given:
  - random input vector $X \in \mathbb{R}^p$,
  - random output variable $Y \in \mathbb{R}$,
  - joint distribution $\Pr(X, Y)$,
- Goal: we seek a function $f(X)$ for predicting Y given values of X.
- To penalize prediction errors, we introduce the *loss function* $L\big(Y, f(X)\big)$.
- *Squared error loss*:

$$L\big(Y, f(X)\big) = (Y - f(X))^2.$$

- *Expected prediction error (EPE)*:

$$\mathrm{EPE}(f) = \mathrm{E}(Y - f(X))^2$$

$$= \int (y - f(x))^2 \Pr(dx, dy).$$

- Since $\Pr(X, Y) = \Pr(Y|X)\Pr(X)$, EPE can also be written as

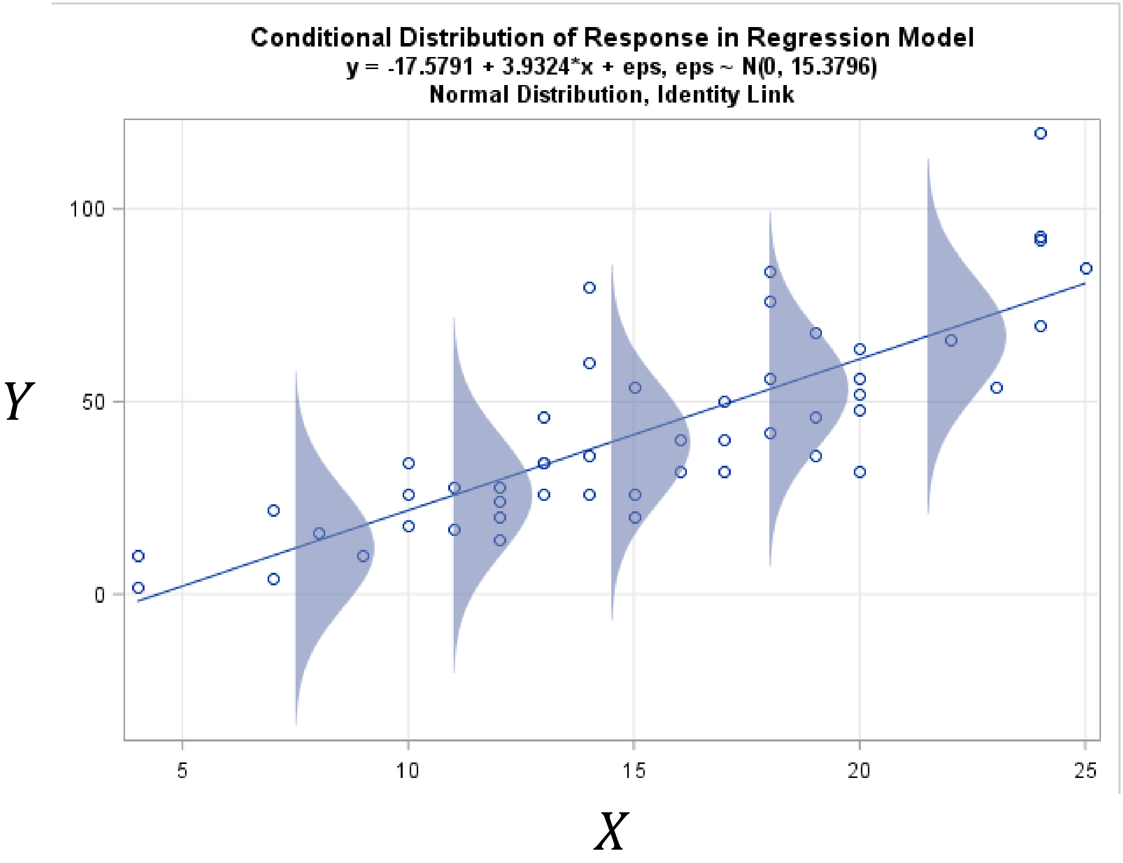
$$\mathrm{EPE}(f) = \mathrm{E}_X \mathrm{E}_{Y|X}([Y - f(X)]^2 | X).$$

- Thus, it suffices to minimize EPE *pointwise*:

$$f(x) = \mathrm{argmin}_c\, \mathrm{E}_{Y|X}\,([Y - c]^2 | X = x)$$

Regression function: $f(x) = E(Y|X = x)$.

Statistical Decision Theory



Regression function: $f(x) = E(Y|X = x)$.

Statistical Decision Theory

- Nearest neighbor methods try to directly implement this recipe
  $$\hat{f}(x) = \text{Ave}(y_i | x_i \in N_k(x)).$$
- Two approximations:
  - expectation is approximated by averaging over sample data;
  - conditioning at a point is relaxed to conditioning on neighborhood.
- As $N, k \to \infty$ and $\frac{k}{N} \to 0$, we have
  $$\hat{f}(x) \to E(Y|X = x).$$
- But usually we do not have very large samples.
- By making assumptions (linearity), we can reduce the required number of observations greatly.
- As increasing the number p of dimensions, the number N of observations required in the training data set increases exponentially.
- Thus the *rate of convergence* to the true estimator (with increasing k) decreases.

Regression function: $f(x) = E(Y|X = x)$.

Statistical Decision Theory

- Linear regression assumes that the regression function is approximately linear
$$f(x) \approx x^T \beta.$$
- This is a model-based approach.
- Plugging this $f(x)$ into EPE,
$$\mathrm{EPE}(f) = E(Y - f(X))^2$$
$$= \mathrm{E}((Y - X^T\beta)^T(Y - X^T\beta))$$
- Differentiating w.r.t. β, leads to
$$\beta = [\mathrm{E}(XX^T)]^{-1}\mathrm{E}(XY)$$

- Again, linear regression replaces the theoretical expectation by averaging over the observed data
$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N}(y_i - x_i^T\beta)^2$$
$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$
- Summary – approximation of $f(X)$
  - Least squares:
    globally linear function
  - Nearest neighbors:
    locally constant function.

Regression function: $f(x) = E(Y|X = x)$.

Statistical Decision Theory

- Additional methods in our course are often model-based but more flexible than the linear model.
- For example, additive models

$$f(X) = \sum_{j=1}^{p} f_j\,(X_j)$$

  - Coordinate function f_j is arbitrary.
  - Approximate *univariate* conditional expectations *simultaneously* for each f_j.
  - Model assumption: additivity.

- What happens if we use another loss function?

$$L_1(Y, f(X)) = E|Y - f(X)|$$

- In this case,

$$\hat{f}(x) = \text{median}(Y|X = x)$$

- More robust than the conditional mean.
- Summary:
  - L_1 criterion not differentiable.
  - Squared error is the most popular.

Statistical Decision Theory

- Procedure for categorical output variable G with values from $\mathcal{G}$.
- Loss function is $K \times K$ matrix $\mathbf{L}$, where $K = \text{card}(\mathcal{G})$
- $\mathbf{L}(k, l)$ is the price paid for misclassifying an observation belonging to class $\mathcal{G}_k$ as class $\mathcal{G}_l$
- $\mathbf{L}$ is zero on the diagonal
- We often use the zero-one loss function

$$\mathbf{L}(k, l) = 1 - \delta_{kl}$$

where $\delta_{kl} = 1$ if $k = l$, otherwise $\delta_{kl} = 0$

- Expected prediction error (EPE)

$$\text{EPE} = \text{E}[L(G, \hat{G}(X))]$$

where expectation taken w.r.t. $\Pr(G, X)$

- Conditioning on X yields

$$\text{EPE} = \text{E}_X \sum_{k=1}^{K} L[\mathcal{G}_k, \hat{G}(X)] \Pr(\mathcal{G}_k | X)$$

- Again, it suffices to pointwise minimization

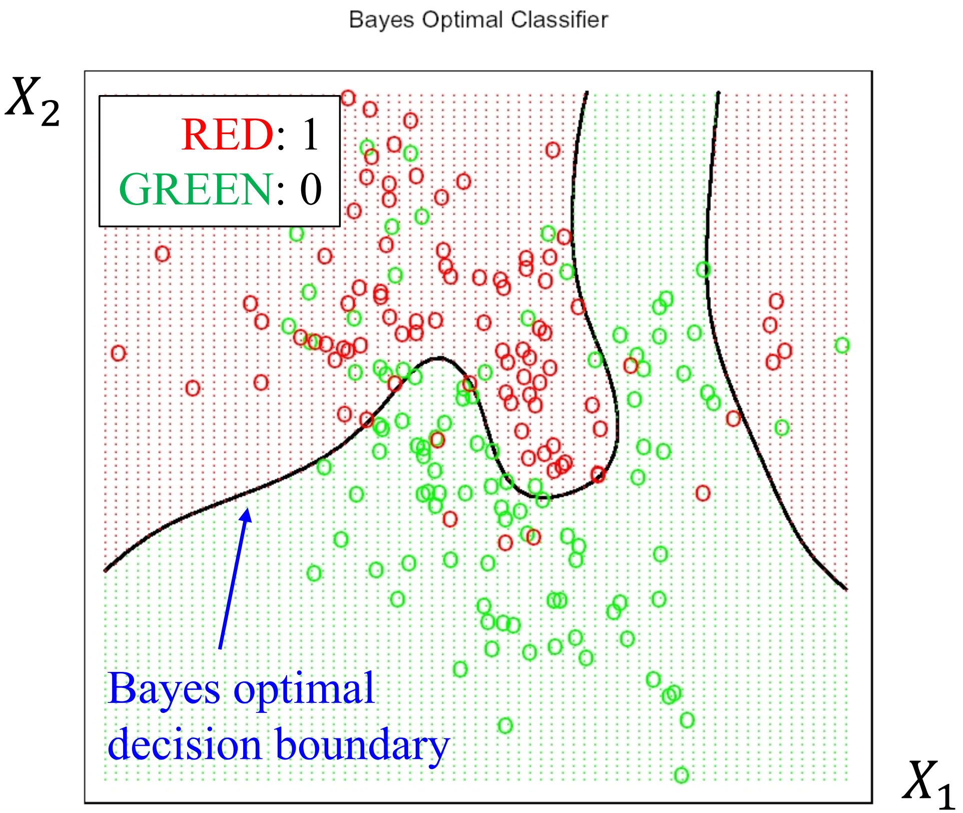
$$\hat{G}(x) = \text{argmin}_{g \in \mathrm{G}} \sum_{k=1}^{K} L(\mathcal{G}_k, g) \Pr(\mathcal{G}_k | X = x)$$

- Or simply

Bayes classifier

$$\hat{G}(x) = \max_{g \in \mathrm{G}} \Pr(g | X = x)$$

Statistical Decision Theory



Since the generating density is known for each class, this boundary can be calculated exactly.

- Expected prediction error (EPE)

$$\mathrm{EPE} = \mathrm{E}[L(G, \hat{G}(X))]$$

where expectation taken w.r.t. $\Pr(G, X)$

- Conditioning on X yields

$$\mathrm{EPE} = \mathrm{E}_X \sum_{k=1}^{K} L[\mathcal{G}_k, \hat{G}(X)] \Pr(\mathcal{G}_k|X)$$

- Again, it suffices to pointwise minimization

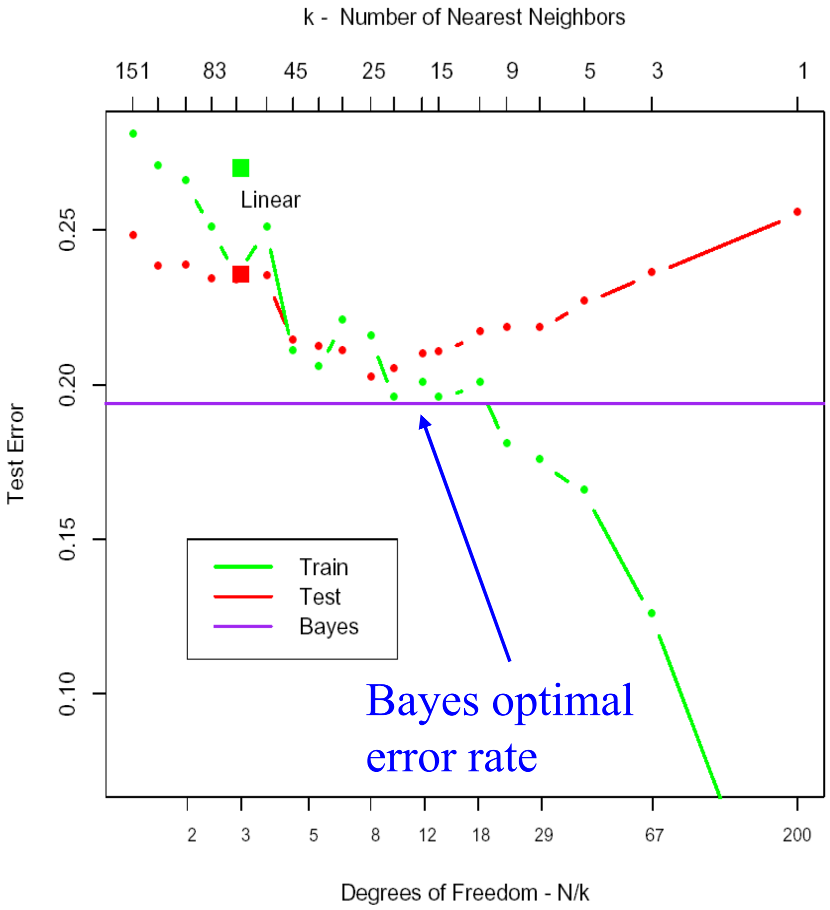
$$\hat{G}(x) = \mathrm{argmin}_{g \in \mathrm{G}} \sum_{k=1}^{K} L(\mathcal{G}_k, g) \Pr(\mathcal{G}_k|X = x)$$

- Or simply

Bayes classifier

$$\hat{G}(x) = \max_{g \in \mathrm{G}} \Pr(g|X = x)$$

Statistical Decision Theory



Bayes optimal error rate

- Expected prediction error (EPE)

$$\mathrm{EPE} = \mathrm{E}[L(G, \hat{G}(X))]$$

where expectation taken w.r.t. $\Pr(G, X)$

- Conditioning on X yields

$$\mathrm{EPE} = \mathrm{E}_X \sum_{k=1}^{K} L[\mathcal{G}_k, \hat{G}(X)] \Pr(\mathcal{G}_k | X)$$

- Again, it suffices to pointwise minimization

$$\hat{G}(x) = \mathrm{argmin}_{g \in \mathrm{G}} \sum_{k=1}^{K} L(\mathcal{G}_k, g) \Pr(\mathcal{G}_k | X = x)$$

- Or simply

Bayes classifier

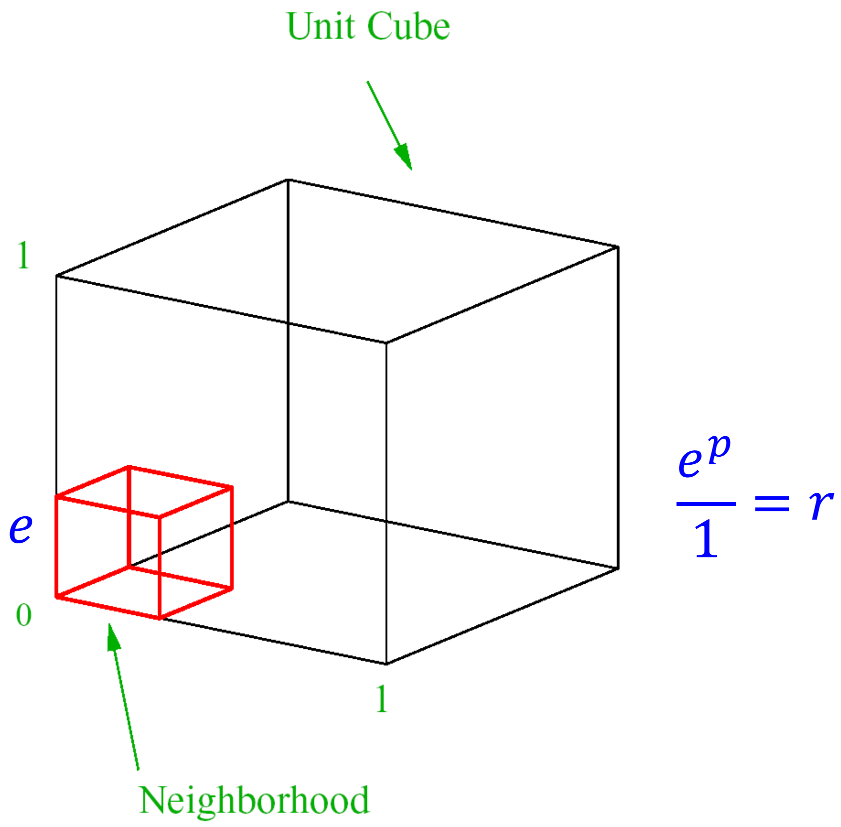
$$\hat{G}(x) = \max_{g \in \mathrm{G}} \Pr(g | X = x)$$

Overview of Supervised Learning II

--- Local Methods in High Dimensions

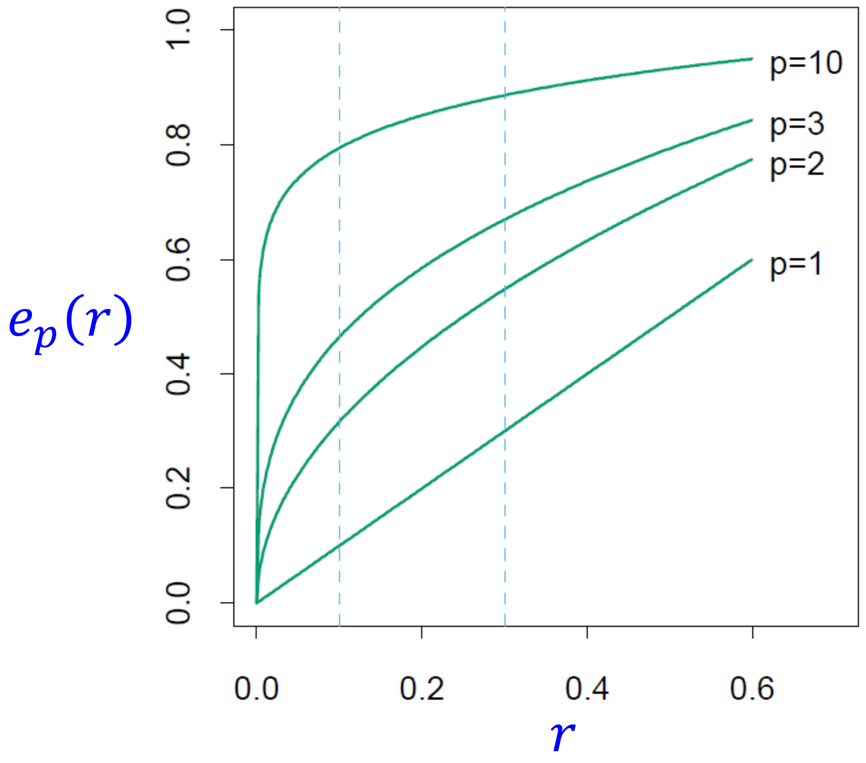
Local Models in High Dimensions

- Curse of Dimensionality: Local neighborhoods become increasingly global, as the number of dimension increases
- Example: Points uniformly distributed in a p-dimensional unit hypercube.
- Hypercubical neighborhood in p dimensions that captures a fraction r of the data
  - edge length: $e_p(r) = r^{\frac{1}{p}}$
  - $e_{10}(0.01) = 0.63$
  - $e_{10}(0.1) \quad = 0.80$

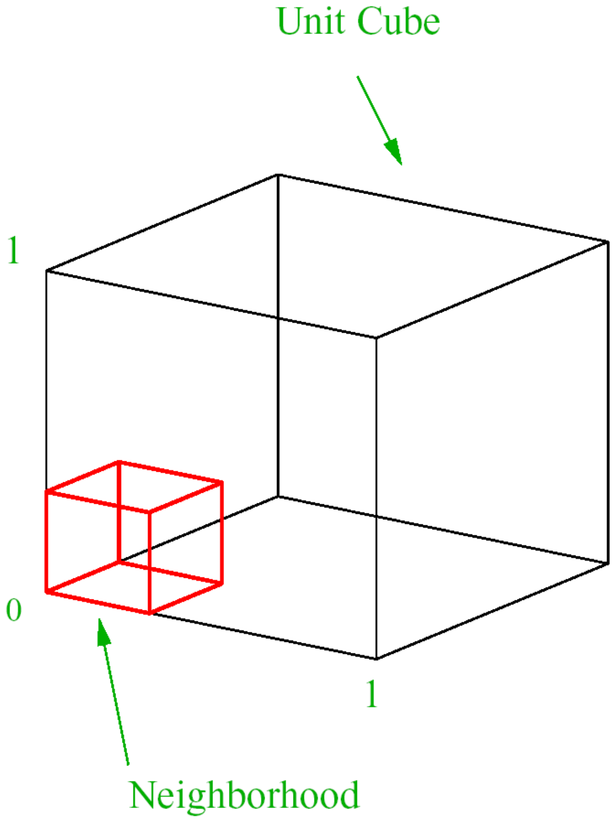


In ten dimensions we need to cover 63% (80%) of the range of each coordinate to capture 1% (10%) of the data.

Local Models in High Dimensions



Reducing r reduces the number of observations and thus the stability.



In ten dimensions we need to cover 63% (80%) of the range of each coordinate to capture 1% (10%) of the data.

Local Models in High Dimensions

- In high dimensions, all sample points are close to the edge of the sample
- N data points uniformly distributed in a p-dimensional unit ball centered at the origin
- Median distance from the closest point to the origin

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

- $d(10{,}500) \approx 0.52$: more than half the way to the boundary

(1) $\prod_{i=1}^{N} \Pr(\|x_i\| > r) = \frac{1}{2}$

(2) $\Pr(\|x_i\| > r) = 1 - \Pr(\|x_i\| \leq r)$
$= 1 - r^p$

(3) $(1 - r^p)^N = \frac{1}{2}$

Volume of a p-ball: $V_p(r) = \frac{\pi^{\frac{p}{2}}}{\Gamma(\frac{p}{2}+1)} r^p$

Local Models in High Dimensions

- In high dimensions, all sample points are close to the edge of the sample
- N data points uniformly distributed in a p-dimensional unit ball centered at the origin
- Median distance from the closest point to the origin

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

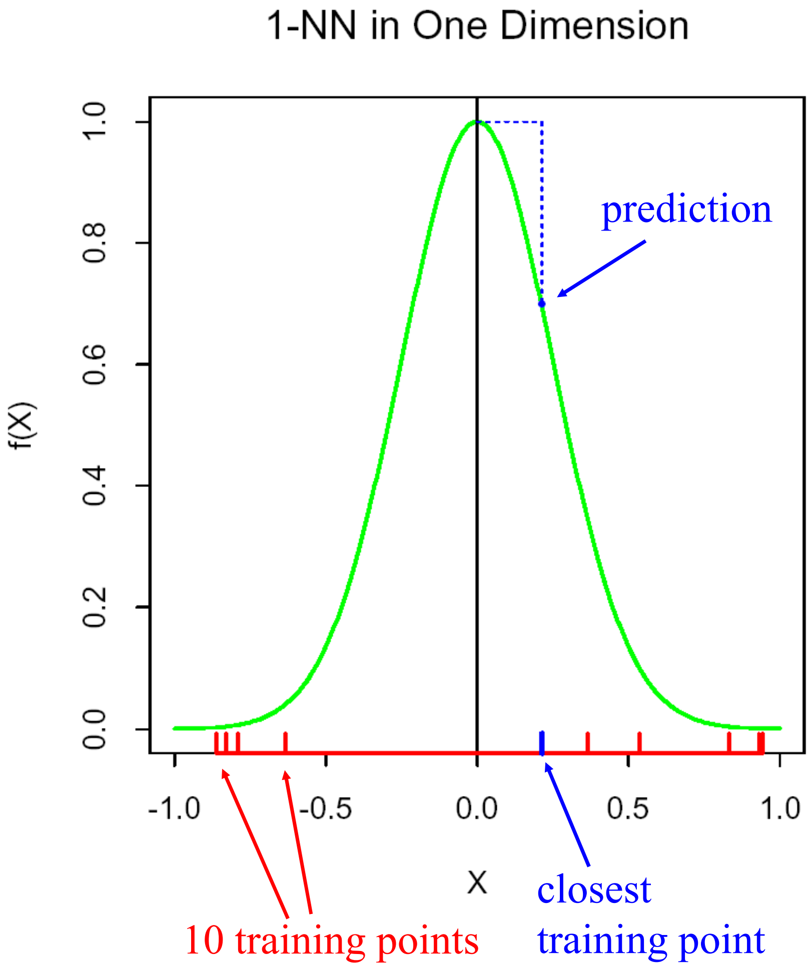
- $d(10{,}500) \approx 0.52$: more than half the way to the boundary
- Sampling density is proportional to $N^{1/p}$
- If $N_1 = 100$ is a dense sample for one input, then $N_{10} = 100^{10}$ is an equally dense sample for 10 inputs.
- Thus in high dimensions all feasible training samples sparsely populate the input space.

Local Models in High Dimensions

- Another example
- $\mathcal{T}$: set of training points x_i generated uniformly in $[-1,1]^p$ (red)
- Functional relationship between X and Y (green)

$$Y = f(X) = e^{-8\|X\|^2}$$

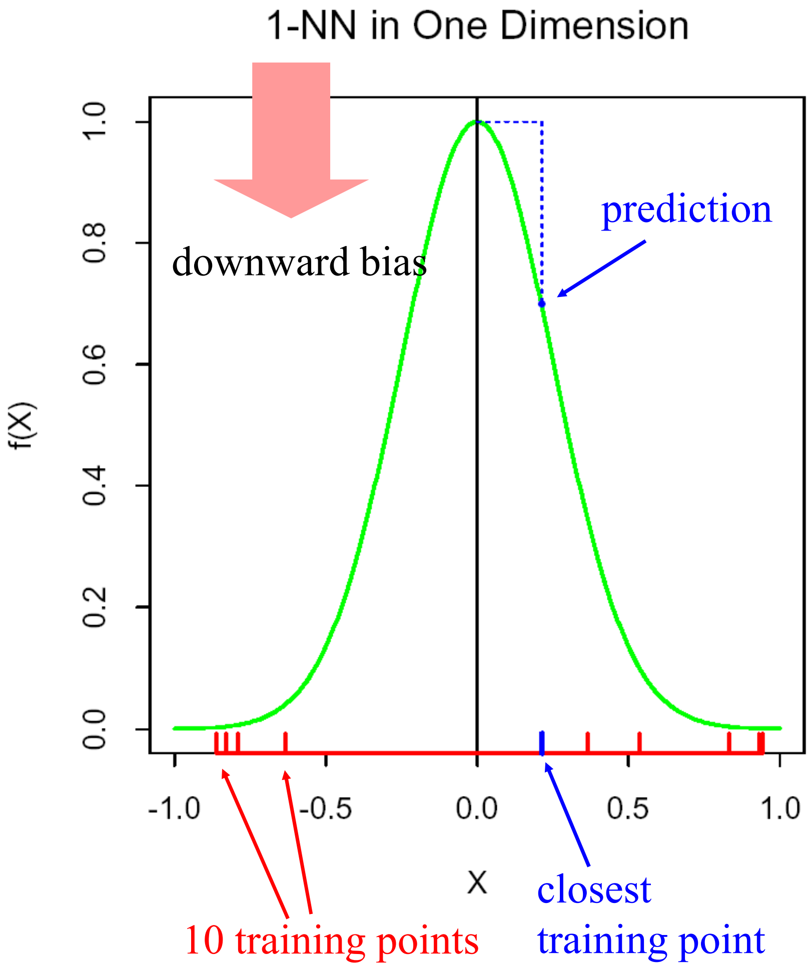
- No measurement error
- Error of a 1-nearest neighbor classifier in estimating $f(0)$ (blue)



Local Models in High Dimensions

- Another example
- Problem deterministic:
  Prediction error is the mean-squared error for estimating $f(0)$

$$
\begin{aligned}
\mathrm{MSE}(x_0) &= \mathrm{E}_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\
&= \mathrm{E}_{\mathcal{T}}[\hat{y}_0 - \mathrm{E}_{\mathcal{T}}(\hat{y}_0)]^2 \\
&\quad + [\mathrm{E}_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\
&= \mathrm{Var}_{\mathcal{T}}(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0)
\end{aligned}
$$



Local Models in High Dimensions

$$E_{\mathcal{T}}\big(\hat{y}_0 - E_{\mathcal{T}}(\hat{y}_0)\big)\big(E_{\mathcal{T}}(\hat{y}_0) - f(x_0)\big) = 0$$

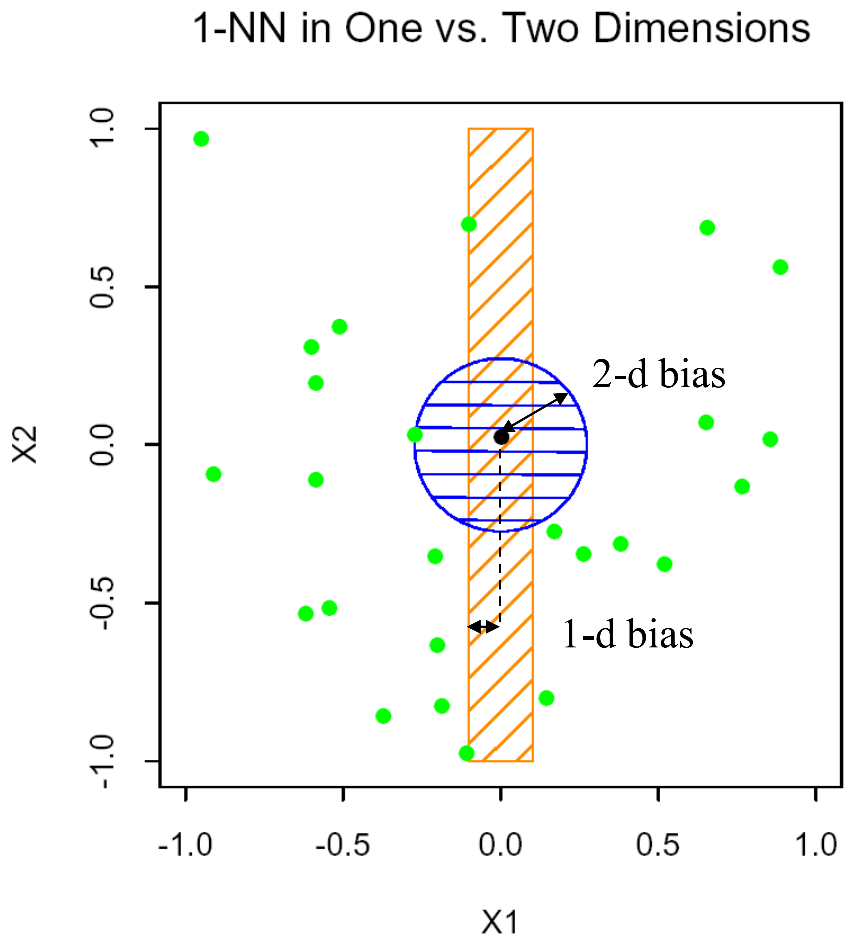
$$
\begin{aligned}
\mathrm{MSE}(x_0) &= E_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\
&= E_{\mathcal{T}}[\hat{y}_0 - E_{\mathcal{T}}(\hat{y}_0) + E_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\
&= E_{\mathcal{T}}\left[\big(\hat{y}_0 - E_{\mathcal{T}}(\hat{y}_0)\big)^2 + 2\underline{\big(\hat{y}_0 - E_{\mathcal{T}}(\hat{y}_0)\big)\big(E_{\mathcal{T}}(\hat{y}_0) - f(x_0)}\big) + \big(E_{\mathcal{T}}(\hat{y}_0) - \underbrace{f(x_0)\big)}_{\text{Constant}}{}^2\right] \\
&= E_{\mathcal{T}}\left[\big(\hat{y}_0 - E_{\mathcal{T}}(\hat{y}_0)\big)^2\right] + \big(E_{\mathcal{T}}(\hat{y}_0) - f(x_0)\big)^2 \\
&= \mathrm{Var}_{\mathcal{T}}(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0)
\end{aligned}
$$

This is known as the bias-variance decomposition.

Local Models in High Dimensions

- Another example
- 1-d (red) vs 2-d (blue)
- As p increases, the bias increases

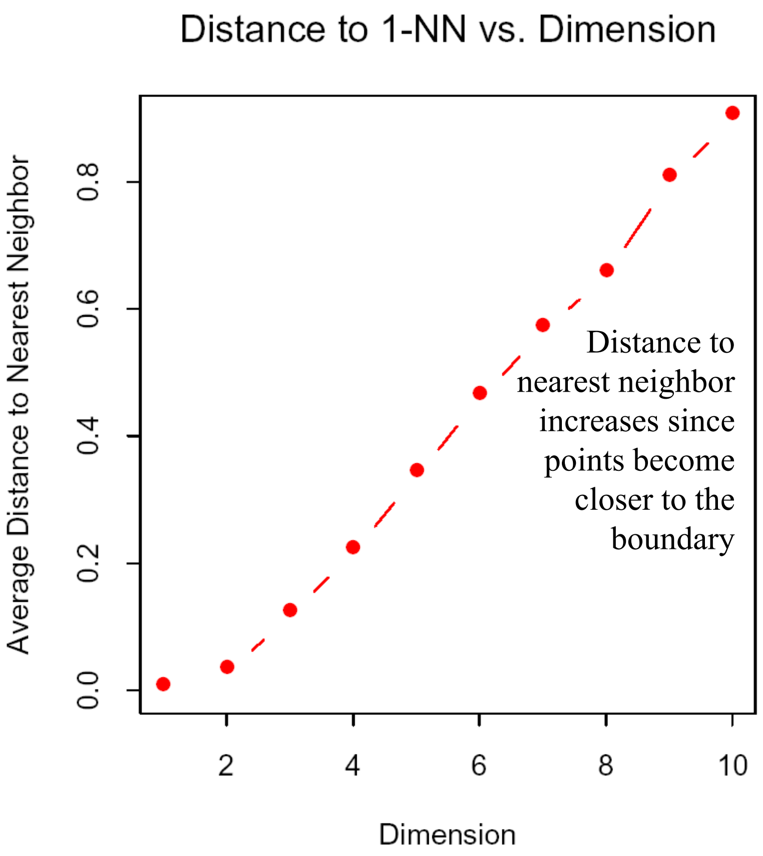
$$
\begin{aligned}
\mathrm{MSE}(x_0) &= \mathrm{E}_{\mathcal{T}}[f(x_0) - \hat{y}_0]^2 \\
&= \mathrm{E}_{\mathcal{T}}[\hat{y}_0 - \mathrm{E}_{\mathcal{T}}(\hat{y}_0)]^2 \\
&\quad + [\mathrm{E}_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\
&= \mathrm{Var}_{\mathcal{T}}(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0)
\end{aligned}
$$



Local Models in High Dimensions

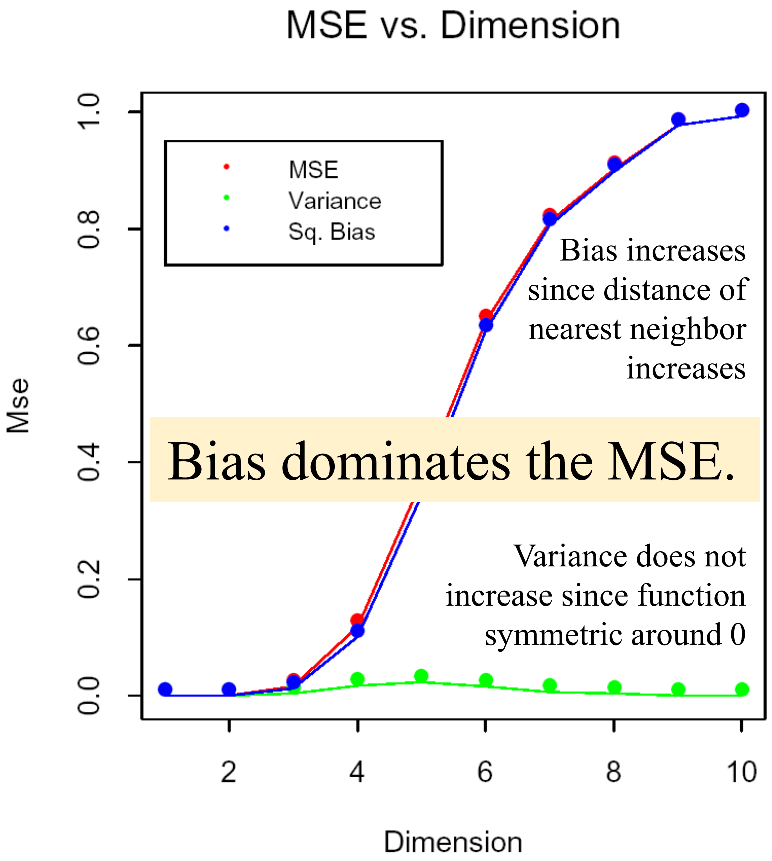
$$Y = f(X) = e^{-8\|X\|^2}$$

- The case on N=1000 training points



Distance to 1-NN vs. Dimension

Distance to nearest neighbor increases since points become closer to the boundary



MSE vs. Dimension

Bias increases since distance of nearest neighbor increases

Bias dominates the MSE.

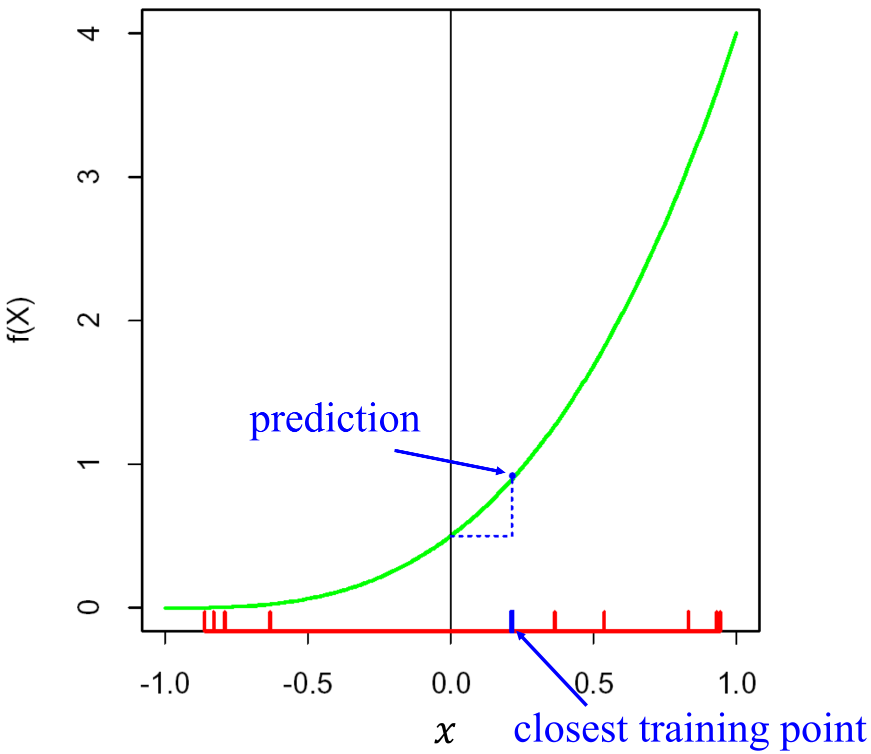
Variance does not increase since function symmetric around 0

Local Models in High Dimensions

$$Y = f(X) = \frac{1}{2}(X_1 + 1)^3$$

- Yet another example

1-NN in One Dimension

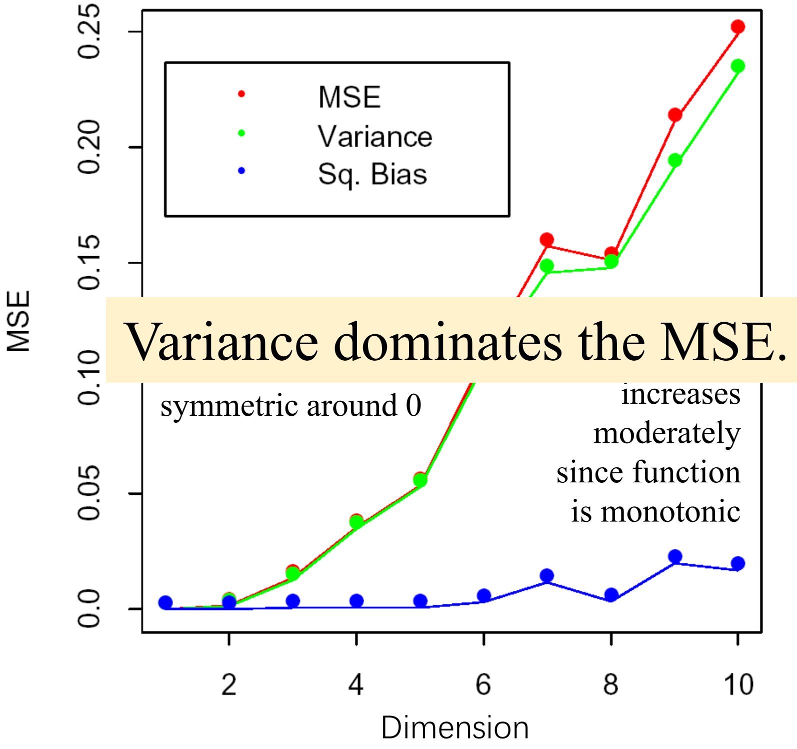


prediction

x

closest training point

MSE vs. Dimension



Variance dominates the MSE.

symmetric around 0

increases moderately since function is monotonic

Local Models in High Dimensions

- Suppose a linear relationship with measurement error

$$Y = X^T\beta + \varepsilon,\ \ \varepsilon \sim \mathcal{N}\,(0, \sigma^2)$$

- We fit the model by least squares.
- For an arbitrary test point x_0,

$$\hat{y}_0 = x_0^T\hat{\beta} = x_0^T\beta + \sum_{i=1}^{N} \ell_i\,(x_0)\varepsilon_i$$

where $\ell_i(x_0)$ is the i-th element of $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}x_0$

Q: How can we get it?

Hint: solution of least squares:

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

Local Models in High Dimensions

- Suppose a linear relationship with measurement error

$$Y = X^T\beta + \varepsilon,\ \ \varepsilon \sim \mathcal{N}\,(0, \sigma^2)$$

- We fit the model by least squares
- For an arbitrary test point x_0,

$$\hat{y}_0 = x_0^T\hat{\beta} = x_0^T\beta + \sum_{i=1}^{N} \ell_i\,(x_0)\varepsilon_i$$

where $\ell_i(x_0)$ is the i-th element of $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}x_0$

- We find that

$$\mathrm{EPE}(x_0) = \ \sigma^2 + \mathrm{E}_{\mathcal{T}}[x_0^T(\mathbf{X}^T\mathbf{X})^{-1}x_0]\sigma^2$$

$$\begin{aligned}\mathrm{EPE}(x_0) &= \mathrm{E}_{y_0|x_0}\mathrm{E}_{\mathcal{T}}(y_0 - \hat{y}_0)^2\\ &= \mathrm{Var}(y_0|x_0) + \mathrm{E}_{\mathcal{T}}[y_0 - \mathrm{E}_{\mathrm{T}}(\hat{y}_0)]^2 + [\mathrm{E}_{\mathcal{T}}(\hat{y}_0) - \mathrm{E}_{\mathcal{T}}(y_0)]^2\\ &= \mathrm{Var}(y_0|x_0) + \mathrm{Var}_{\mathcal{T}}(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0)\\ &= \sigma^2 + \mathrm{E}_{\mathcal{T}}[x_0^T(\mathbf{X}^T\mathbf{X})^{-1}x_0]\sigma^2 + 0^2\end{aligned}$$

Additional variance

Local Models in High Dimensions

- Suppose a linear relationship with measurement error

$$Y = X^T\beta + \varepsilon,\ \ \varepsilon \sim \mathcal{N}\,(0, \sigma^2)$$

- We fit the model by least squares.
- For an arbitrary test point x_0,

$$\hat{y}_0 = x_0^T\hat{\beta} = x_0^T\beta + \sum_{i=1}^{N} \ell_i\,(x_0)\varepsilon_i$$

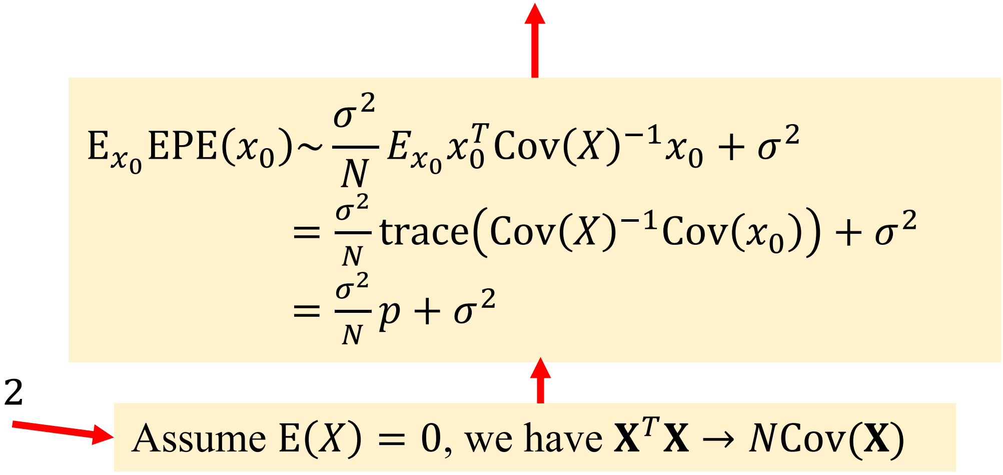
where $\ell_i(x_0)$ is the i-th element of $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}x_0$

- We find that

$$\text{EPE}(x_0) = \ \sigma^2 + \text{E}_{\mathcal{T}}[x_0^T(\mathbf{X}^T\mathbf{X})^{-1}x_0]\sigma^2$$

- Additional variance σ^2 originates from the nondeterministic output
- Variance depends on x_0
- If N is large, we get

$$\text{E}_{x_0}\text{EPE}(x_0) \sim \frac{\sigma^2}{N}p + \sigma^2$$



$$\begin{aligned}\text{E}_{x_0}\text{EPE}(x_0) &\sim \frac{\sigma^2}{N}E_{x_0}x_0^T\text{Cov}(X)^{-1}x_0 + \sigma^2\\ &= \frac{\sigma^2}{N}\text{trace}\big(\text{Cov}(X)^{-1}\text{Cov}(x_0)\big) + \sigma^2\\ &= \frac{\sigma^2}{N}p + \sigma^2\end{aligned}$$

Assume $\text{E}(X) = 0$, we have $\mathbf{X}^T\mathbf{X} \rightarrow N\text{Cov}(\mathbf{X})$

Local Models in High Dimensions

- Suppose a linear relationship with measurement error

$$Y = X^T \beta + \varepsilon, \ \ \varepsilon \sim \mathcal{N}\,(0, \sigma^2)$$

- We fit the model by least squares.
- For an arbitrary test point x_0,

$$\hat{y}_0 = x_0^T \hat{\beta} = x_0^T \beta + \sum_{i=1}^{N} \ell_i\,(x_0)\varepsilon_i$$

  where $\ell_i(x_0)$ is the i-th element of $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}x_0$

- We find that

$$\text{EPE}(x_0) = \ \sigma^2 + \text{E}_{\mathcal{T}}[x_0^T(\mathbf{X}^T\mathbf{X})^{-1}x_0]\sigma^2$$

- Additional variance σ^2 originates from the nondeterministic output
- Variance depends on x_0
- If N is large, we get

$$\text{E}_{x_0}\text{EPE}(x_0) \sim \frac{\sigma^2}{N} p + \sigma^2$$

- As p increases, variance grows negligible for large N or small σ^2
- No bias
- Curse of dimensionality controlled

Local Models in High Dimensions

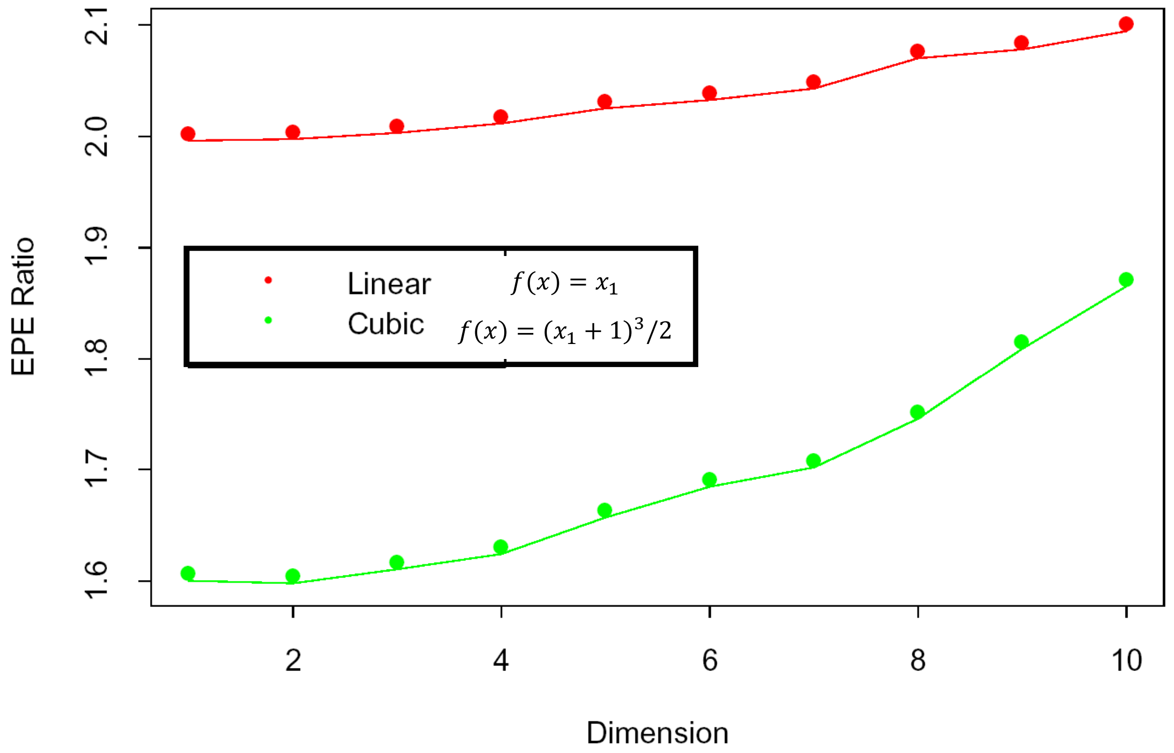
- More generally

$$Y = f(X) + \varepsilon,$$

$$X \text{ uniform}, \ \varepsilon \sim \mathcal{N}\,(0,1)$$

- Sample size: $N = 500$
- Linear case
  - EPE (Least Squares) is slightly above 1 no bias
  - EPE (1-NN) always above 2, grows slowly as nearest training point strays from target

$$\text{EPE ratio} = \frac{\text{EPE (1} - \text{NN)}}{\text{EPE (least squares)}}$$



Local Models in High Dimensions

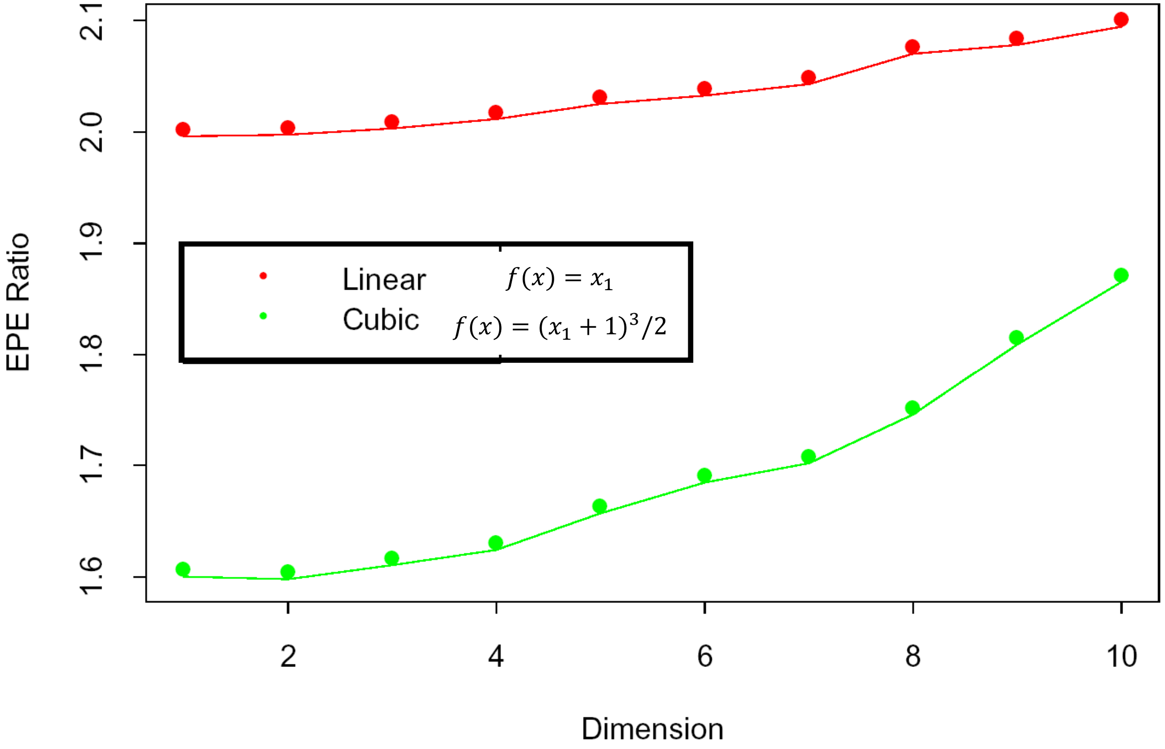
- More generally

$$Y = f(X) + \varepsilon,$$

$$X \text{ uniform}, \ \varepsilon \sim \mathcal{N}(0,1)$$

- Sample size: $N = 500$
- Cubic case
  - EPE (Least Squares) is biased, thus ratio is smaller

$$\text{EPE ratio} = \frac{\text{EPE }(1-\text{NN})}{\text{EPE (least squares)}}$$



Overview of Supervised Learning II

--- Statistical Models

Statistical Models

- NN methods are the direct implementation of

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

- But it can fail in two ways
  - With high dimensions NN need not be close to the target point
  - If special structure exists in the problem, this can be used to reduce variance and bias
- We anticipate using other classes of models for $f(x)$, and discuss a framework for incorporating them into the prediction problem.

Statistical Models – A Statistical Model for $\Pr(X, Y)$

- Assume additive error model

  $Y = f(X) + \varepsilon$

  $\mathrm{E}(\varepsilon) = 0$

  ε independent of X

- Then $\Pr(Y|X)$ depends only on the conditional mean of $f(x)$
- This model is a good approximation in many cases
- In many cases, $f(x)$ is deterministic and error enters through uncertainty in the input. This can often be mapped on uncertainty in the output with deterministic input.

Statistical Models – Function Approximation

- Data: pairs (x_i, y_i) that are points in $(p+1)$-dimensional Euclidean space, we fit $f: \mathbb{R}^p \rightarrow \mathbb{R}$ by

$$y_i = f(x_i) + \varepsilon_i$$

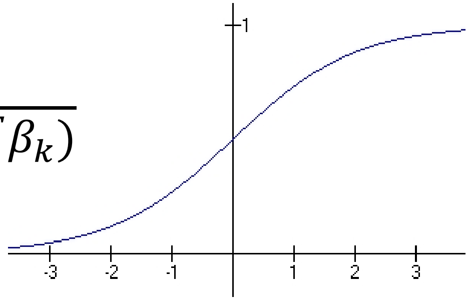
- More general input spaces are possible
- Goal: a good approximation of $f(x)$ in some region of input space, given the training set $\mathcal{T}$
- Many models have certain parameters θ
  - E.g. for the linear model $f(x) = x^T\beta$ and $\theta = \beta$

- Linear basis expansions have the more general form

$$f_\theta(x) = \sum_{k=1}^{K} h_k(x)\theta_k$$

- h_k: a suitable set of functions or transformations of the input vector x.
- Examples:
  - Polynomial expansions: $h_k(x) = x_1 x_2^2$
  - Trigonometric expansions: $h_k(x) = \cos(x_1)$
  - Sigmoid expansion:

$$h_k(x) = \frac{1}{1 + \exp(-x^T\beta_k)}$$



Statistical Models – Function Approximation

- Approximating f_θ by minimizing the residual sum of squares

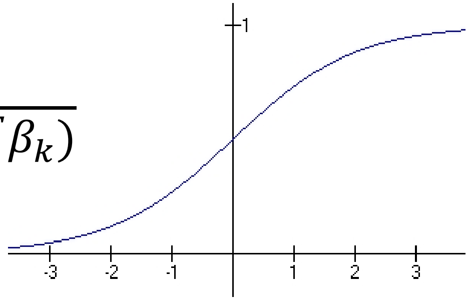
$$\mathrm{RSS}(\theta) = \sum_{i=1}^{N} (y_i - f_\theta(x_i))^2$$

- Linear basis expansions have the more general form

$$f_\theta(x) = \sum_{k=1}^{K} h_k(x)\theta_k$$

- h_k: a suitable set of functions or transformations of the input vector x.
- Examples:
  - Polynomial expansions: $h_k(x) = x_1 x_2^2$
  - Trigonometric expansions: $h_k(x) = \cos(x_1)$
  - Sigmoid expansion:

$$h_k(x) = \frac{1}{1 + \exp(-x^T\beta_k)}$$

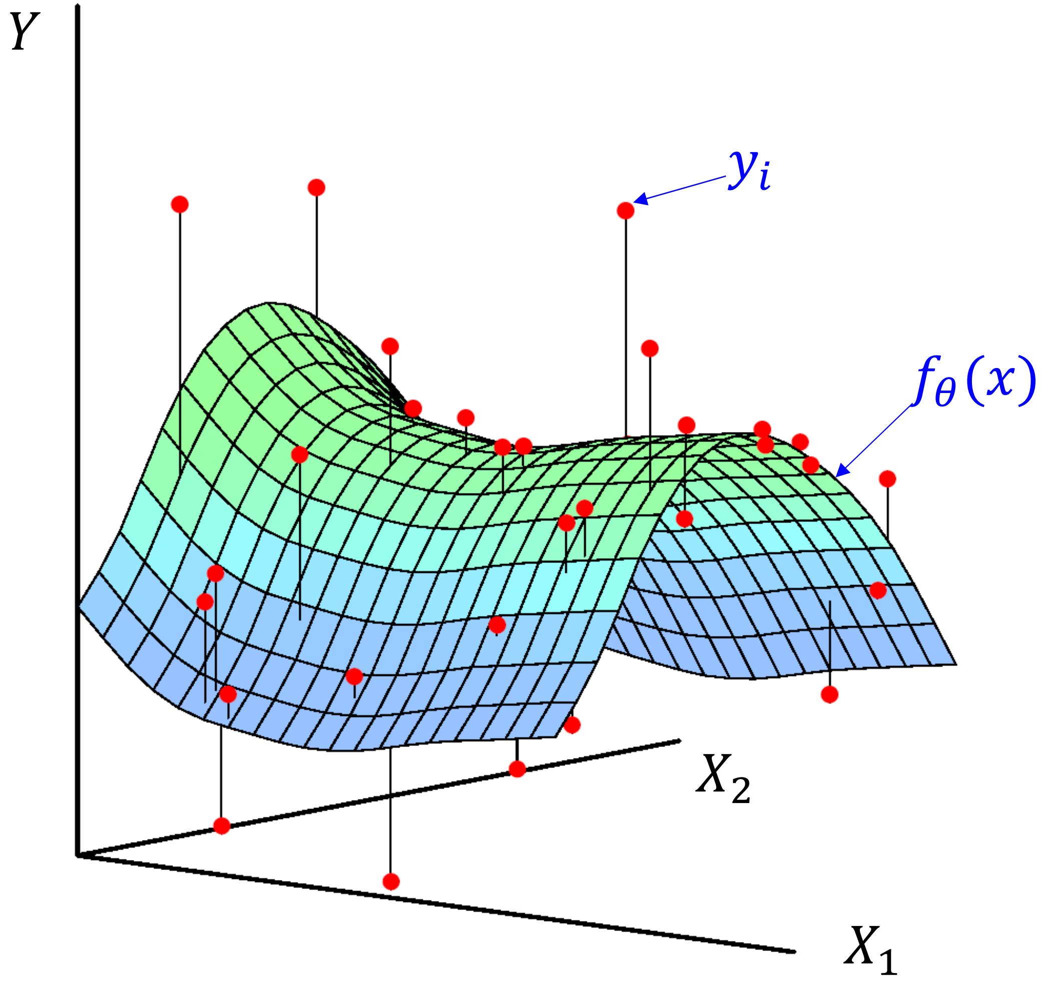


Statistical Models – Function Approximation

- Approximating f_θ by minimizing the residual sum of squares

$$\mathrm{RSS}(\theta) = \sum_{i=1}^{N} (y_i - f_\theta(x_i))^2$$

- Intuition
  - f surface in $(p+1)$ −space
  - Observe noisy realizations
  - Want fitted surface as close to the observed points as possible
  - Distance measured by RSS
- Methods
  - Closed form: if basis function have no hidden parameters
  - Iterative: otherwise



Statistical Models – Function Approximation

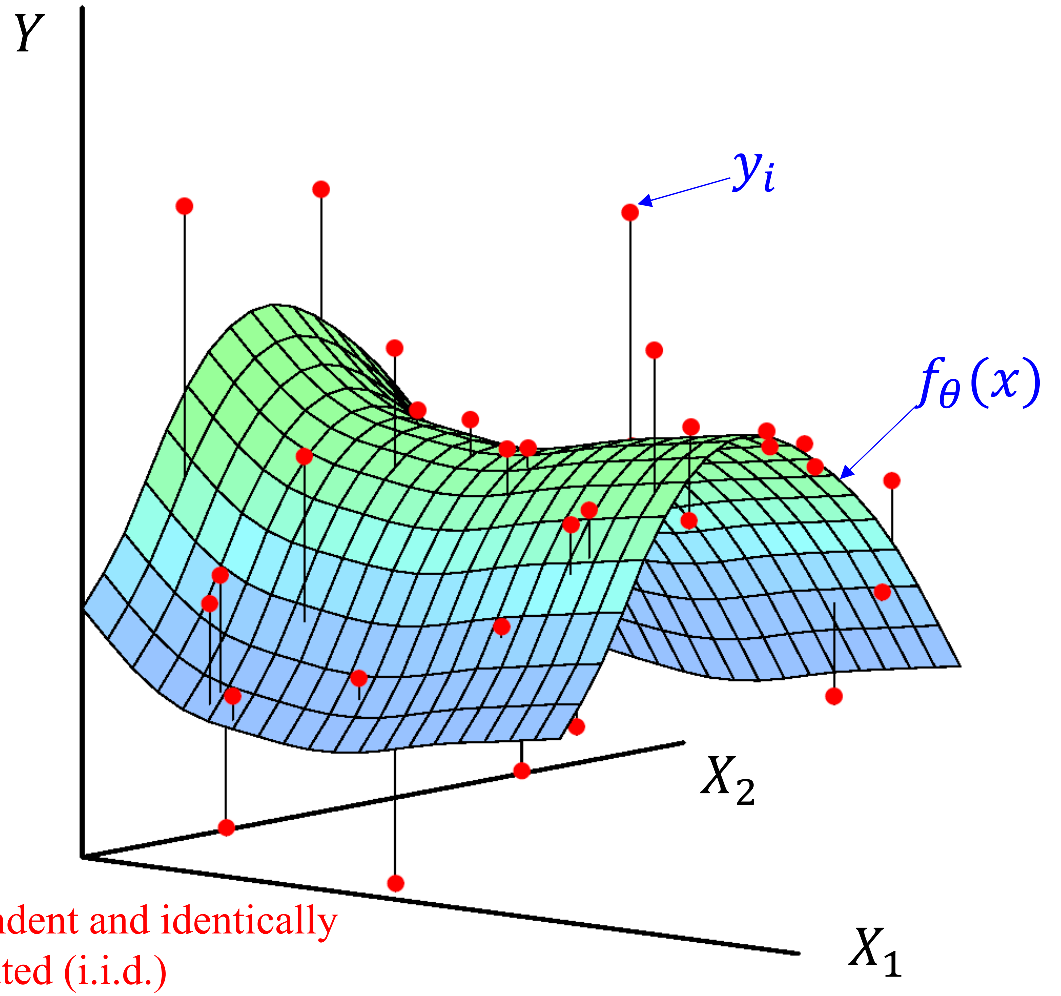
- Approximating f_θ by maximum likelihood estimation (MLE)
- Assume an independently drawn random sample $y_i, i = 1, \dots, N$ from a probability density $\Pr_\theta(y)$.
- The log-probability of observing the sample is

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

$$L(\theta) = \log \Pr_\theta\big(\underline{y_1, y_2, \dots, y_N}\big)$$

$$= \log \prod_{i=1}^{N} Pr_\theta(y_i)$$

independent and identically distributed (i.i.d.)



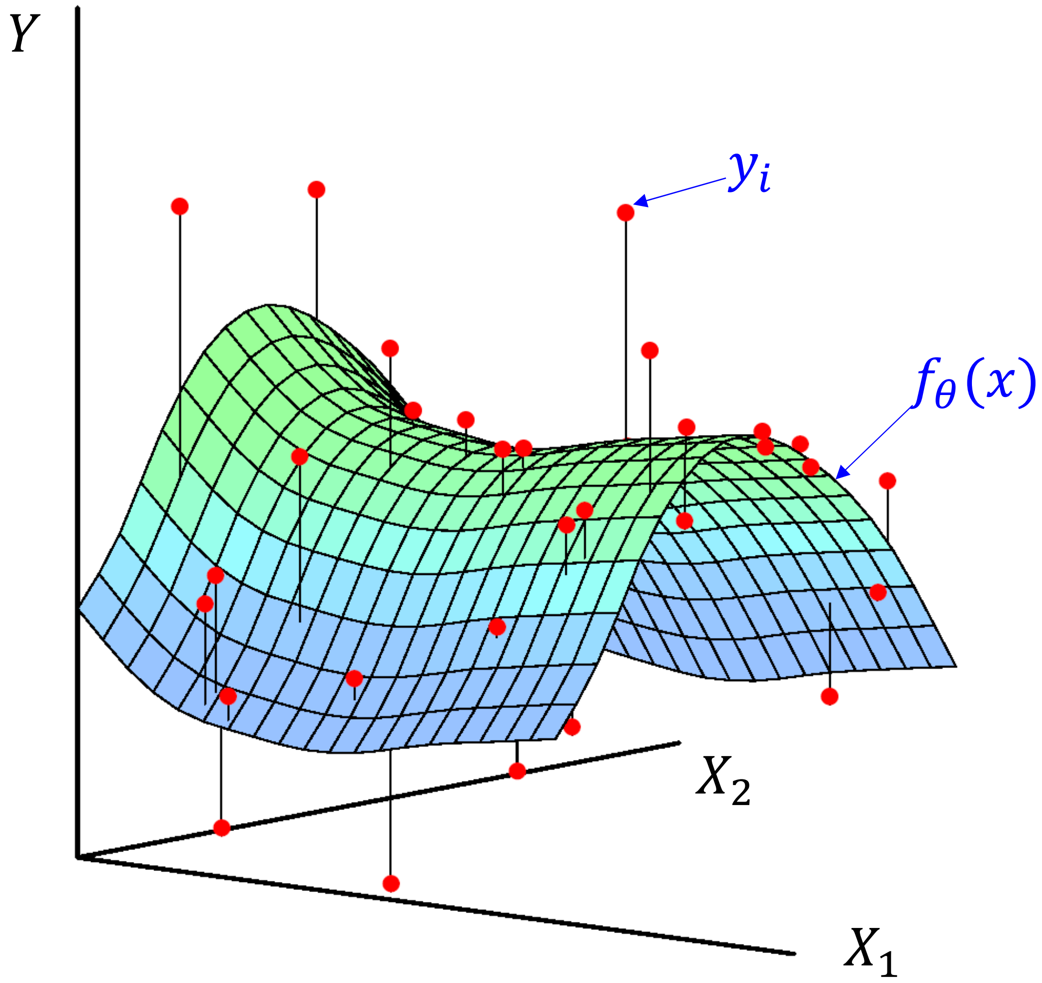
Statistical Models – Function Approximation

- Approximating f_θ by maximum likelihood estimation (MLE)
- Assume an independently drawn random sample $y_i, i = 1, \dots, N$ from a probability density $\mathrm{Pr}_\theta(y)$.
- The log-probability of observing the sample is

$$L(\theta) = \sum_{i=1}^{N} \log \mathrm{Pr}_\theta(y_i)$$

- Set θ to maximize $L(\theta)$

Intuition:
Under the assumed statistical model, the observed data is most probable.



Statistical Models – Function Approximation

- Approximating f_θ by maximum likelihood estimation (MLE)
- Assume an independently drawn random sample $y_i, i = 1, \dots, N$ from a probability density $\Pr_\theta(y)$.
- The log-probability of observing the sample is

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

- Set θ to maximize $L(\theta)$

$$\Pr_\theta(y|X = x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}(\frac{y - f_\theta(x)}{\sigma})^2)$$

- Least squares with additive error model

$$Y = f_\theta(X) + \varepsilon$$
$$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$

is equivalent to maximum likelihood with the conditional likelihood

$$\Pr_\theta(Y|X) = \mathcal{N}(f_\theta(X), \sigma^2)$$

- This is, because in this case the *log-likelihood* of data is

$$L(\theta) = -\frac{N}{2}\log(2\pi) - N\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - f_\theta(x_i))^2$$

Statistical Models – Function Approximation

- Approximating f_θ by maximum likelihood estimation (MLE)
- Assume an independently drawn random sample $y_i, i = 1, \dots, N$ from a probability density $\Pr_\theta(y)$.
- The log-probability of observing the sample is

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

- Set θ to maximize $L(\theta)$

- Least squares with additive error model

$$Y = f_\theta(X) + \varepsilon$$
$$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$

is equivalent to maximum likelihood with the conditional likelihood

$$\Pr_\theta(Y|X) = \mathcal{N}(f_\theta(X), \sigma^2)$$

- This is, because in this case the *log-likelihood* of data is

$$L(\theta) = -\frac{N}{2}\log(2\pi) - N\log\sigma$$
$$-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - f_\theta(x_i))^2$$

Proportional to RSS

$$\operatorname{argmax}_\theta L(\theta) = \operatorname{argmin}_\theta \mathrm{RSS}(\theta) = \operatorname{argmin}_\theta \sum_{i=1}^{N}(y_i - f_\theta(x_i))^2$$

Overview of Supervised Learning II

--- Classes of Restricted Estimators

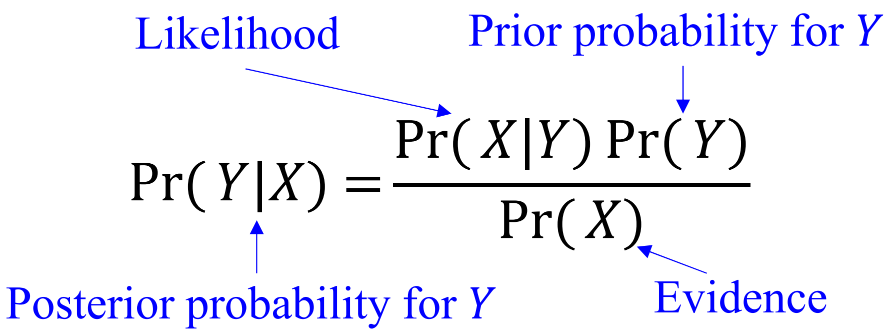
- Bayesian Methods and Roughness Penalty
- Kernel Methods and Local Regression
- Basis Functions

Bayesian Methods and Roughness Penalty

- Bayesian methods
- Formula for joint probabilities

$$\Pr(X,Y) = \Pr(Y|X)\Pr(X) = \Pr(X|Y)\Pr(Y)$$

- Bayes's theorem



Likelihood

Prior probability for Y

$$\Pr(Y|X) = \frac{\Pr(X|Y)\Pr(Y)}{\Pr(X)}$$

Posterior probability for Y

Evidence

Posterior $\propto$ Likelihood $\times$ Prior

- RSS is penalized with a roughness penalty

$$\mathrm{PRSS}(f\,;\lambda) = \mathrm{RSS}(f) + \lambda J(f)$$

- $J(f)$ is large for ragged functions
  - E.g. cubic smoothing spline is the solution for the least-squares problem

$$\mathrm{PRSS}(f\,;\lambda) = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda\int [f''(x)]^2\,dx$$

  - Large second derivative is penalized

Bayesian Methods and Roughness Penalty

- Introducing penalty functions is a type of regularization
  - It works against overfitting
  - It implements beliefs about unseen parts of the problem
- In a Bayesian framework
  - Penalty J is the log-prior (probability distribution)
  - PRSS is the log-posterior (probability distribution)

Posterior $\propto$ Likelihood $\times$ Prior

- RSS is penalized with a roughness penalty

$$\mathrm{PRSS}(f\,;\lambda) = \mathrm{RSS}(f) + \lambda J(f)$$

- $J(f)$ is large for ragged functions
  - E.g. cubic smoothing spline is the solution for the least-squares problem

$$\mathrm{PRSS}(f\,;\lambda) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int [f''(x)]^2\,dx$$

  - Large second derivative is penalized

Kernel Methods and Local Regression

- Kernel functions
  - model the local neighborhoods in NN methods
- Gaussian kernel

$$K_\lambda(x_0, x) = \frac{1}{\lambda}\exp\left[-\frac{\|x - x_0\|^2}{2\lambda}\right]$$

  - assigns weights to points that die exponentially with the square of the distance from the point x_0
  - λ controls the variance of the Gaussian density (neighborhood width)

- Simplest kernel estimate:
  Nadaraya-Watson weighted average

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

- General local regression estimate of $f\ (x_0)$ is $f_{\hat{\theta}}(x_0)$, where $\hat{\theta}$ minimizes

$$\mathrm{RSS}(f_\theta, x_0) = \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - f_\theta(x_i))^2$$

Kernel Methods and Local Regression

- f_θ is a parameterized function, such as a low-order polynomial
  - $f_\theta = \theta_0$
    Nadaraya-Watson estimate
  - $f_\theta = \theta_0 + \theta^T x$
    local linear regression model
- NN methods can be regarded as kernel methods with a special metric
  $K_k(x, x_0)$
  $= I\left(\|x - x_0\| \le \|x_{(k)} - x_0\|\right)$

$x_{(k)}$ training sample ranked k in distance from x_0
I indicator function

- Simplest Kernel estimate:
  Nadaraya-Watson weighted average

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

- General local regression estimate of $f\ (x_0)$ is $f_{\hat{\theta}}(x_0)$, where $\hat{\theta}$ minimizes

$$\mathrm{RSS}(f_\theta, x_0) = \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - f_\theta(x_i))^2$$

Basis Functions

- Include linear and polynomial expansions and more
- General form

$$f_\theta(x) = \sum_{m=1}^{M} \theta_m \, h_m(x)$$

- The term linear refers to the action of the parameters θ
- Cover a wide range of methods

- Radial Basis Functions

$$f_\theta(x) = \sum_{m=1}^{M} K_{\lambda_m}(\mu_m, x)\theta_m$$

  - Parameters are
    - Centroids μ_m
    - Scales λ_m
  - E.g. the Gaussian kernel

$$K_\lambda(\mu, x) = \exp(-\frac{\|x-\mu\|^2}{2\lambda})$$

  - Change the regression problem from a linear problem to a combinatorially nonlinear problem.

Basis Functions

- Include linear and polynomial expansions and more
- General form

$$f_\theta(x) = \sum_{m=1}^{M} \theta_m \, h_m(x)$$

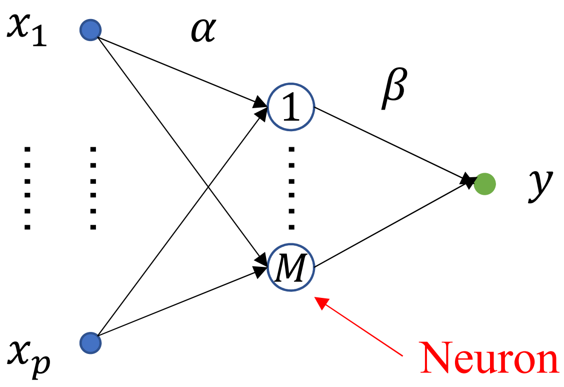
- The term linear refers to the action of the parameters θ
- Cover a wide range of methods

- Neural Networks
  - Single-layer feed-forward model

$$f_\theta(x) = \sum_{m=1}^{M} \beta_m \, \sigma(\alpha_m^T x + b_m)$$

Neuron weight

Neuron output



Neuron

Basis Functions

- Include linear and polynomial expansions and more
- General form

$$f_\theta(x) = \sum_{m=1}^{M} \theta_m \, h_m(x)$$

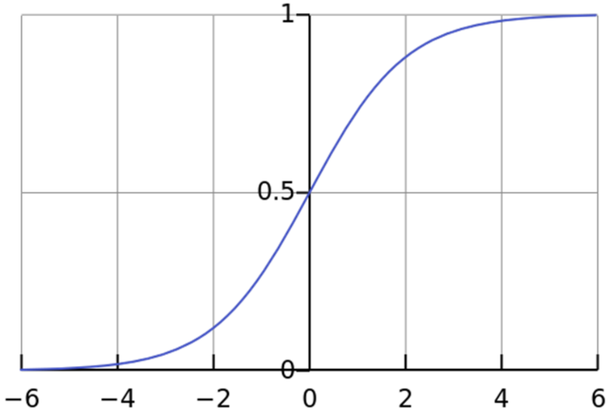
- The term linear refers to the action of the parameters θ
- Cover a wide range of methods

- Neural Networks
  - Single-layer feed-forward model

$$f_\theta(x) = \sum_{m=1}^{M} \beta_m \, \sigma(\alpha_m^T x + b_m)$$

  - $\sigma(x)$ is known as *activation function*

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Overview of Supervised Learning II

--- Model Selection

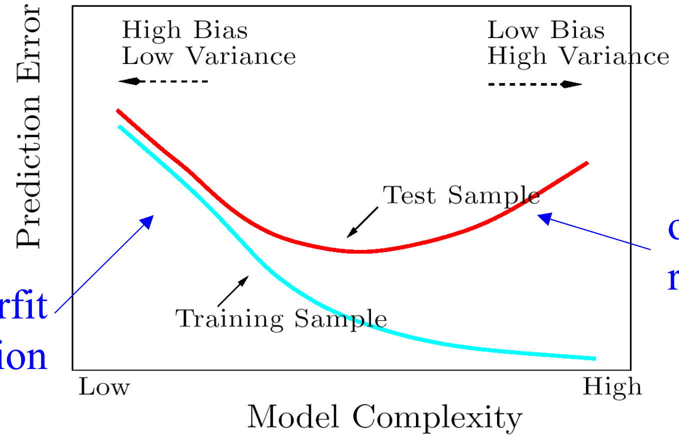
Model Selection

- Smoothing and complexity parameters
  - Coefficient of the penalty term
  - Width of the kernel
  - Number of basis functions
- The setting of the parameters implements a trade-off between bias and variance
- Example: k-NN methods

$$Y = f(X) + \varepsilon$$
$$\mathrm{E}(\varepsilon) = 0$$
$$\mathrm{Var}(\varepsilon) = \sigma^2$$

- Generalization error

$$\begin{aligned}\mathrm{EPE}_k(x_0) &= \mathrm{E}[Y - \hat{f}_k(x_0) | X = x_0] \\ &= \sigma^2 + [\mathrm{Bias}^2(\hat{f}_k(x_0)) + \mathrm{Var}_{\mathcal{T}}(\hat{f}_k(x_0))] \\ &= \underbrace{\sigma^2}_{\text{irreducible error}} + \underbrace{\left[f(x_0) - \frac{1}{k}\sum_{\ell=1}^{k} f(x_{(\ell)})\right]^2 + \frac{\sigma^2}{k}}_{\text{mean-square error}}\end{aligned}$$



underfit region

overfit region