STATS 415: Assessing Model Accuracy Part II

Prof. Liza Levina

Department of Statistics, University of Michigan

Measuring Quality of Fit: MSE

- Recall the regression setting: y_i is the observed value of response for point i; $\hat{y}_i$ is the predicted value of response for point i.
- Can measure accuracy by the mean squared error (MSE), i.e.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Training MSE: plug in training data
- Test MSE: plug in test data

Training vs Test MSE

- Regression methods are designed to make training MSE small
- If you increase flexibility (add new variables in linear regression; add interaction terms; reduce K in KNN regression), the training MSE will never increase, and in most cases will decrease.
- Test error may go up or down: reducing training error may just be overfitting
- To understand this better, we need to look at the bias and variance trade-off

Bias

- Bias refers to systematic error introduced by approximating a real life problem by a model (e.g., a linear model).
- Formally, if $y = f(x) + \varepsilon$, $E\varepsilon = 0$, then

$$\mathrm{bias}(\hat{f}(x)) = E\hat{f}(x) - f(x)$$

- The expectation is taken over the distribution of noise
- A method is called unbiased if $\mathrm{bias}(\mathrm{x}) = 0$ for all x
- In general, the more flexible a method, the lower its bias.

E(y0=F(f(x)+epsilon)=f(x)=0=f(x)

Variance

- Variance refers to random error resulting from sample variability; it measures how much $\hat{f}$ would change if you had a different training sample from the same distribution.
- Formally, if $y = f(x) + \varepsilon$, $E\varepsilon = 0$, then

$$\mathrm{Var}(\hat{f}(x)) = E(\hat{f}(x) - E\hat{f}(x))^2$$

- The expectation is taken over the distribution of noise
- In general, the more flexible a method, the higher its variance.

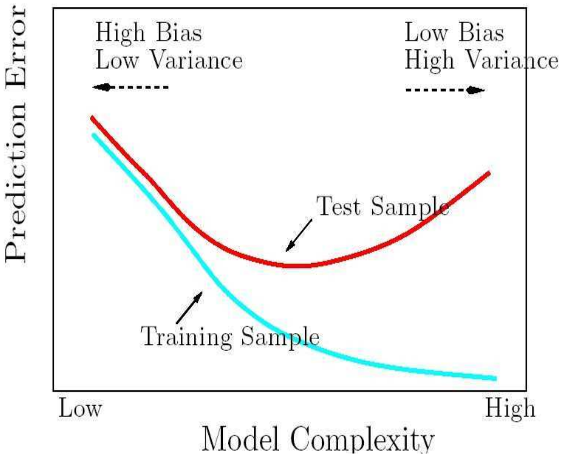
The Bias/Variance Trade-off

- The bias and variance are competing forces; generally reducing one increases the other.
- For a given fixed point x, the expected test MSE for a new y at x is

$$
\begin{aligned}
E(\mathrm{MSE}(x)) &= E(y-\hat{f}(x))^2 \\
&= [E(\hat{f}(x)) - f(x)]^2 + E[\hat{f}(x) - E(\hat{f}(x))]^2 + \mathrm{Var}(\varepsilon) \\
&= [\mathrm{Bias}(\hat{f}(x))]^2 + \mathrm{Var}(\hat{f}(x)) + \sigma^2
\end{aligned}
$$

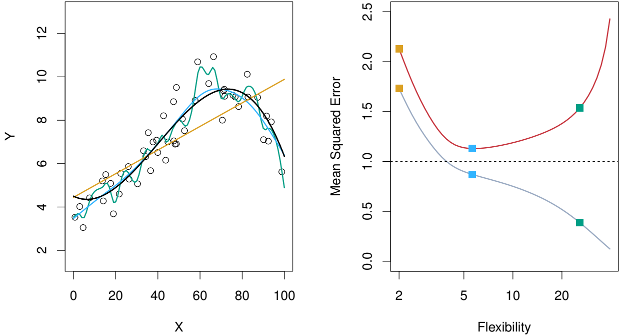
- Thus the expected test MSE may go up or down with increased complexity, depending on which term dominates.
- σ^2 is the irreducible noise; no method can do better than that.

Model complexity trade-off



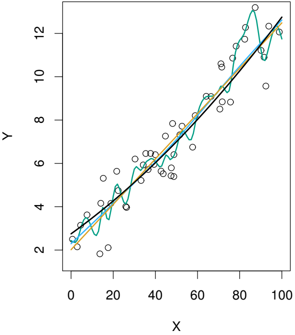
Different Levels of Flexibility: Example I

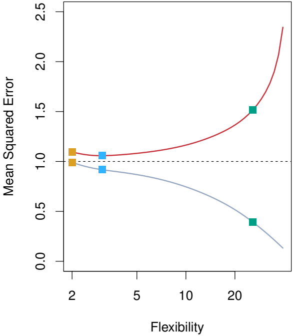
- Left: black (truth); orange (linear estimate); blue (smoothing spline); green (smoothing spline, more flexible)
- Right: red (test MSE); grey (training MSE); dashed (minimum possible test MSE, irreducible error)



Different Levels of Flexibility: Example II

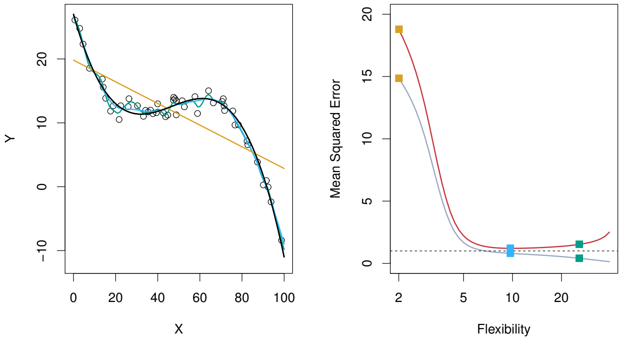
- Left: black (truth); orange (linear estimate); blue (smoothing spline); green (smoothing spline, more flexible)
- Right: red (test MSE); grey (training MSE); dashed (minimum possible test MSE, irreducible error)



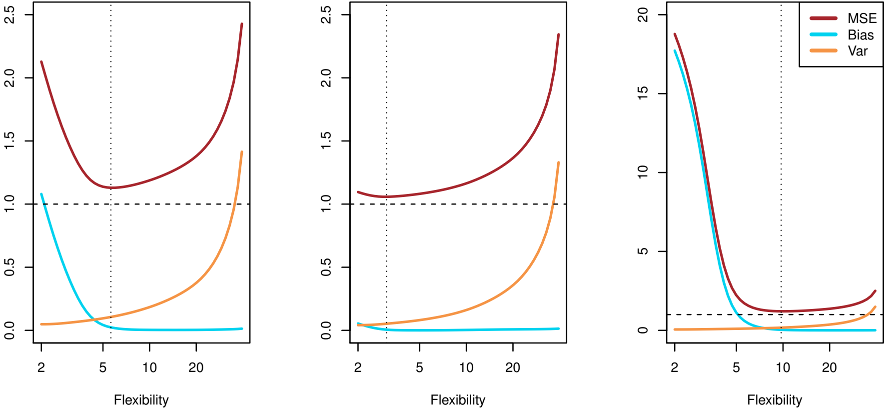


Different Levels of Flexibility: Example III

- Left: black (truth); orange (linear estimate); blue (smoothing spline); green (smoothing spline, more flexible)
- Right: red (test MSE); grey (training MSE); dashed (minimum possible test MSE, irreducible error)



Test MSE, Bias and Variance for Examples I, II, III



KNN regression: MSE

- A flexible non-parametric method, predicting value at x as

$$\hat{f}(x) = \frac{1}{K} \sum_{i: x_i \in N_K(x)} y_i$$

  where $N_K(x)$ is K closest neighbors of x in the training data
- The smaller K, the more complex the model: the number of "parameters" is roughly n/K.
- As long as $n/K > p$, KNN is more "flexible" than a linear model with p predictors.
- Suppose the data arise from a model $y = f(x) + \varepsilon$, with $E(\varepsilon) = 0$ and $\text{Var}(\varepsilon) = \sigma^2$. The general formula is

$$E(\text{MSE}(x)) = E(y - \hat{f}(x))^2 = [\text{Bias}(\hat{f}(x))]^2 + \text{Var}(\hat{f}(x)) + \sigma^2$$

KNN: Bias/Variance Trade-off

- Recall $\hat{y} = \frac{1}{K}\sum_{\ell=1}^{K} y_{(\ell)}$, where the subscript (ℓ) indicates the sequence of nearest neighbors to x.
- For simplicity, assume x_i's in the sample are fixed (nonrandom). Then

$$
\begin{aligned}
Ey_{(\ell)} &= Ef(x_{(\ell)}) + E\varepsilon = f(x_{(\ell)}) \\
\mathrm{Var}(y_{(\ell)}) &= \mathrm{Var}(f(x_{(\ell)})) + \mathrm{Var}(\varepsilon) = \sigma^2 \\
E\hat{f}(x) &= \frac{1}{K}\sum_{\ell=1}^{K} Ey_{(\ell)} = \frac{1}{K}\sum_{\ell=1}^{K} f(x_{(\ell)}) \\
\mathrm{Var}(\hat{f}(x)) &= \frac{1}{K}\sum_{\ell=1}^{K} \mathrm{Var}(y_{(\ell)}) = \frac{\sigma^2}{K} \\
E(\mathrm{MSE}(x)) &= \left(f(x) - \frac{1}{K}\sum_{\ell=1}^{K} f(x_{(\ell)})\right)^2 + \frac{\sigma^2}{K} + \sigma^2
\end{aligned}
$$

$$E(\mathrm{MSE}(x)) = \left(f(x) - \frac{1}{K}\sum_{\ell=1}^{K} f(x_{(\ell)})\right)^2 + \frac{\sigma^2}{K} + \sigma^2$$
$$= \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Irreducible\ Error}$$

- The squared bias term tends to increase with K.
    - For small K, the closest neighbors have values $f(x_{(\ell)})$ similar to $f(x_0)$, at least if f is smooth.
    - For large K, “further away” points are counted as neighbors.
- The variance term decreases when K increases.

The Classification Setting

- The class label y takes values in a finite, unordered set (spam/email, cancer type, etc).
  - Two-class: $y \in \{c_1, c_2\}$
  - Multi-class: $y \in \{c_1, c_2, \dots, c_K\}$
- For a classification problem we can use the error rate, i.e.

$$\text{Error Rate} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i \neq \hat{y}_i)$$

- $\mathbb{I}(y_i \neq \hat{y}_i)$ is an indicator function, = 1 if $(y_i \neq \hat{y}_i)$ and otherwise 0.
- The error rate is the fraction of incorrect classifications, or misclassifications.
- Again, training error always goes down as model complexity increases; the test error can go up or down.

The Optimal Classifier

- (x, y) have a joint probability distribution.
- Want a classifier $\hat{C}(x)$ with a small misclassification error:

$$\mathsf{R}(\hat{C}) = P(\hat{C}(x) \neq y)$$

- Bayes optimal classifier:

$$\begin{aligned} C^*(x_0) &= \arg\min_C \mathsf{R}(C) \\ &= \arg\max_k P(y = c_k | x = x_0) \end{aligned}$$

- In practice, $P(y|x)$ is not known, but it can be estimated; KNN estimates it in a flexible non-parametric way.

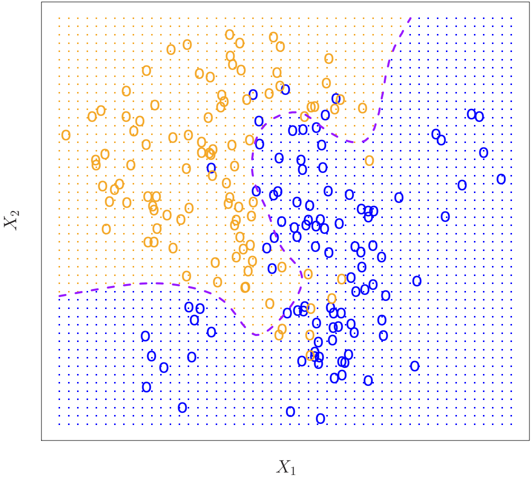
Bayes Error Rate

- The Bayes error rate is the error of the Bayes optimal classifier:

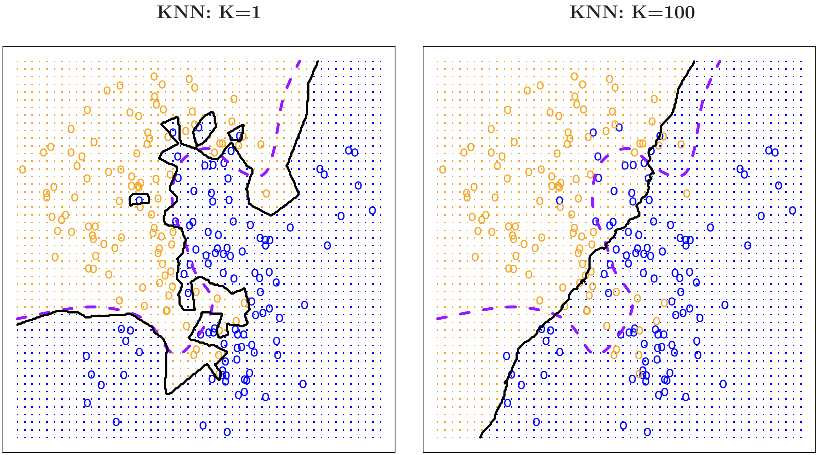
$$P(C^*(x) \neq y)$$

- This is the lowest possible error rate that can only be achieved if we knew exactly the “true” probability distribution of the data.
- No classifier (or statistical learning method) can achieve lower expected test error than the Bayes error rate; but in practice it cannot be calculated.

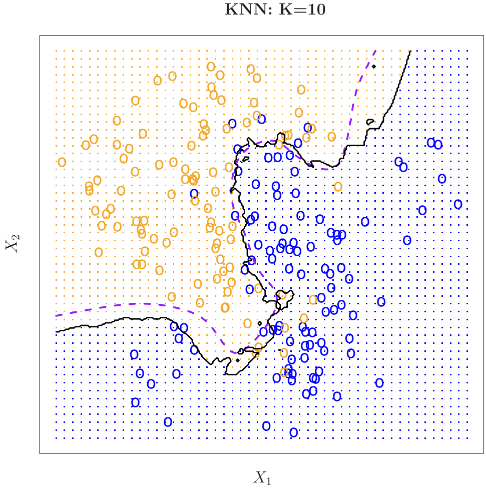
Bayes Optimal Classifier: Simulated Data



Simulated Data: KNN classifier with $K = 1$ and $K = 100$

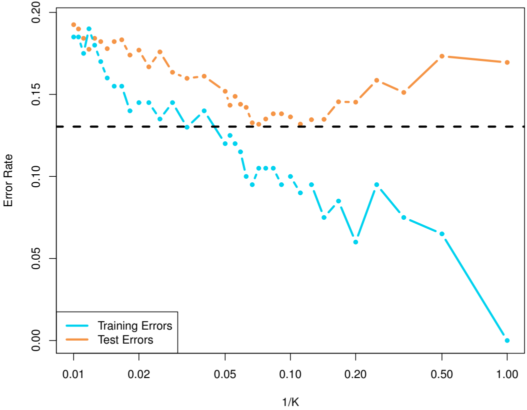


Simulated Data: KNN classifier with $K = 10$

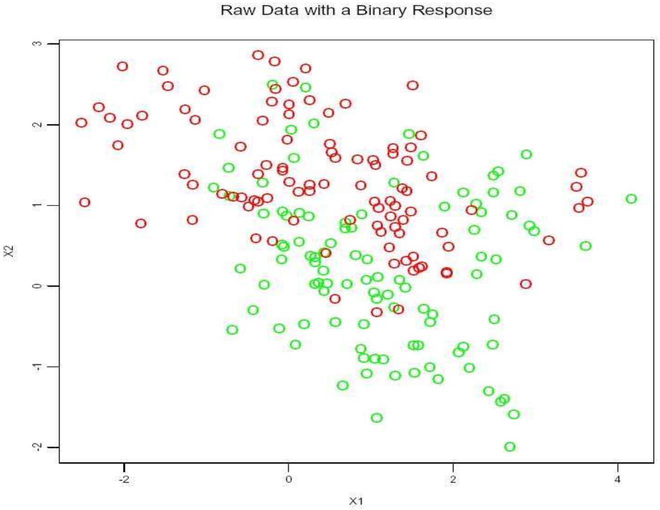


Training vs Test Error Rates on the Simulated Data

- As K increases, the bias goes up; the variance goes down; the optimum test error is somewhere in between.

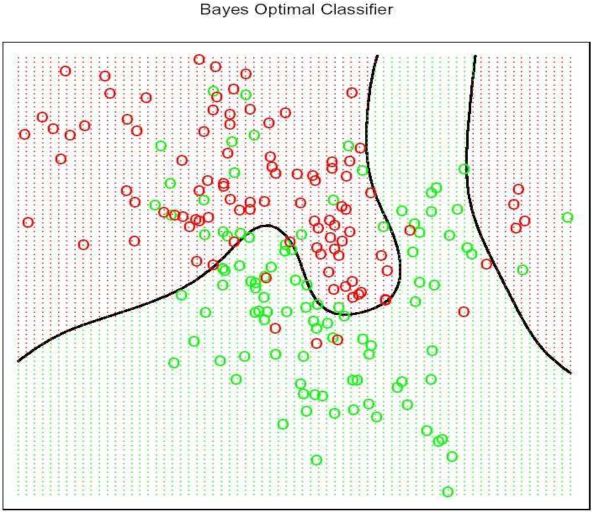


Another binary example: simulated data

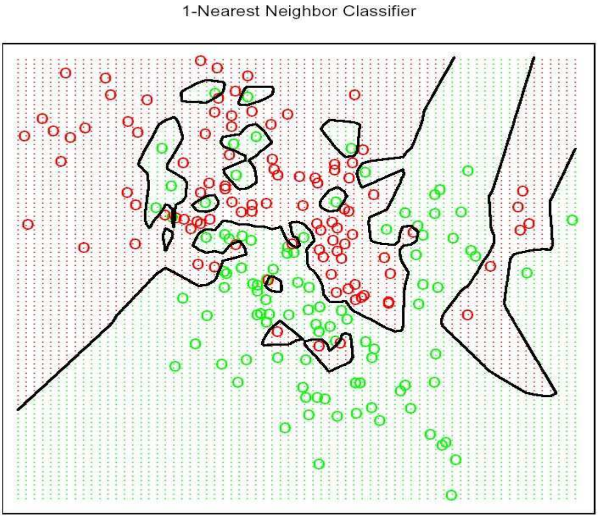


The optimal classifier

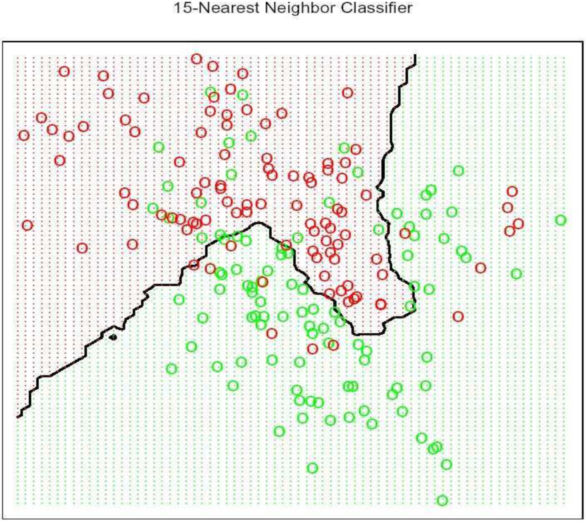
Can only be computed because this is simulated data



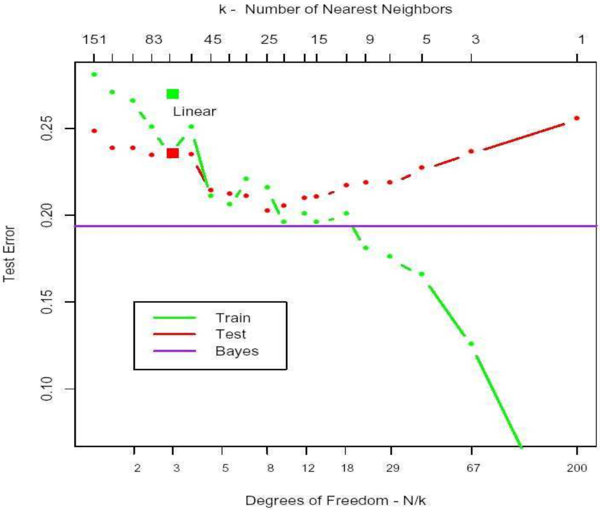
The 1-NN classifier



The 15-NN classifier



Traning vs test error for kNN classifier



Summary: model complexity trade-off