Chapter 2: Overview of statistical learning

Yu-Tzung Chang and Hsuan-Wei Lee

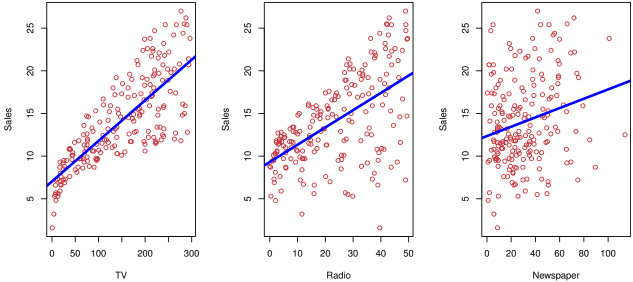
Department of Political Science, National Taiwan University

2018.10.04

Outline

- What is statistical learning?
- Why and how to estimate the model?
- The trade-off between prediction accuracy and model interpretability
- Measuring quality of fit
- The bias-variance trade-off
- The classification setting

Introduction



The data shown are *Sales* vs *TV*, *Radio*, and *Newspaper*, with a blue linear-regression line fit separately to each.
We want to predict *Sales* using the information of the other three variables, that is, we want to find a *model* f such that

$$\mathit{Sales} \approx f(\mathit{TV}, \mathit{Radio}, \mathit{Newspaper}).$$

Notations of the book

- Here *Sales* is a *response* or *target* that one wishes to predict, this is usually denoted as a response variable Y.
- The variables *TV*, *Radio*, and *Newspaper* are *features*, or *inputs*, or *predictors*; we name them as X_1, X_2, and X_3.
- The input vector could be written collectively as

$$x = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}.$$

- The model could be written as

$$Y = f(X) + \epsilon$$

where ϵ captures measurement errors and other discrepancies.

The choices of the models f

There are infinite numbers of models f to choose from. For example,

$$Sales = f(TV, Radio, Newspaper)$$

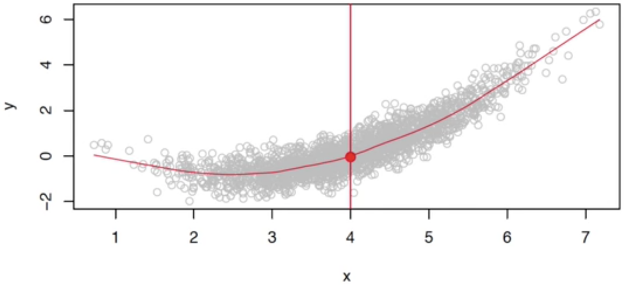
could be

- $Sales = \beta_0 + \beta_1 TV + \beta_2 Radio + \beta_3 Newspaper + \epsilon$
- $Sales = e^{\beta_0 + \beta_1 TV + \beta_2 Radio + \beta_3 Newspaper} + \epsilon$
- $Sales = log(\beta_0 + \beta_1 TV + \beta_2 Radio + \beta_3 Newspaper) + \epsilon$
- $Sales = \beta_0 + \beta_1 TV \times Radio + \beta_2 Newspaper + \epsilon$

What is f(X) good for?

- With a good model f, we can make predictions of Y at new points $X = x$.
- We can understand which components of $X = (X_1, X_2, \ldots, X_p)$ are important in explaining Y, and which are irrelevant. For example, *age* has a huge impact on *height*, but the *zodiac signs* does not.
- Depending on the complexity of the model f, we may be able to understand how each component X_j of X affects Y.

Choosing a possible model f



Is there an ideal $f(X)$? In particular, what is a good value for $f(X)$ at any selected value of X, say $X = 4$? There can be infinite amount of candidates Y values at $X = 4$. A good value is

$$f(4) = E(Y|X = 4)$$

where $E(Y|X = 4)$ means the *expected values* of Y given $X = 4$.This model $f(x) = E(Y|X = x)$ is called the *regression function*.

The regression function $f(x)$

- This can be defined and written in a vector form

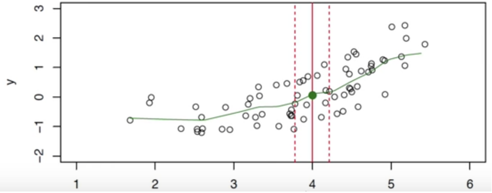
$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

- Is the *ideal* or *optimal* predictor of Y with regard to mean-squared prediction error: $f(x) = E(Y|X = x)$ is the function that minimizes $E[(Y - g(X))^2|X = x]$ over all functions of g at all points $X = x$.
- $\epsilon = Y - f(x)$ is the *irreducible* error, that is, even if we know $f(x)$, we would still make errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values.
- We have

$$E[(Y - \hat{f}(X))^2|X = x] = [f(x) - \hat{f}(x)]^2 + Var(\epsilon)$$

The first term is reducible and the second term in irreducible.

How to estimate f?



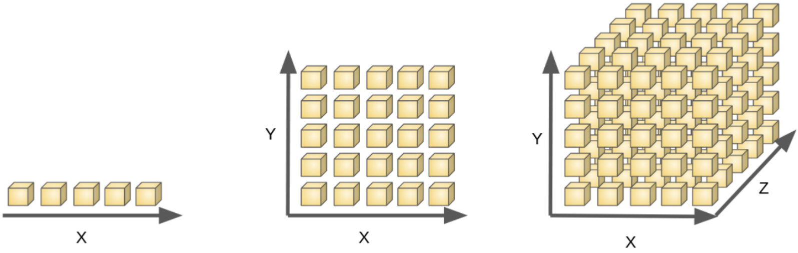
- Typically we have few if any data points with $X = 4$ exactly. Therefore we can't compute $E(Y|X = x)$ directly.
- Relax the definition and let

$$\hat{f}(x) = Ave(Y|x \in N(x))$$

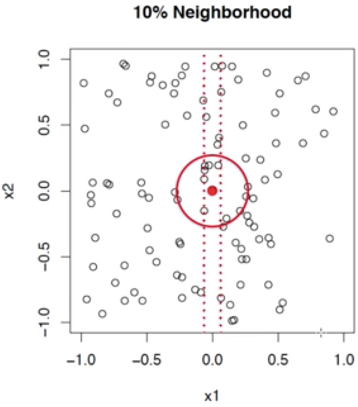
  where $N(x)$ is some *neighborhood* of x.
- Nearest neighbor averaging can be good when the number of independent variables is not too large.
- Other smoothing methods like *kernel* and *spline* would be discussed later.

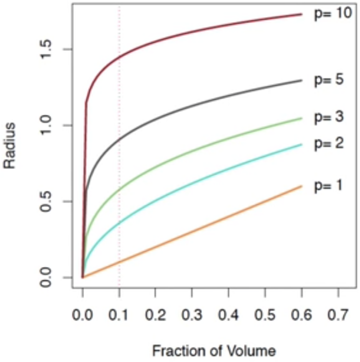
Curse of dimensionality

- Nearest neighbor methods can be bad when the number of independent variables is too large.
- *Curse of dimensionality*: nearest neighbors tend to be far away in high dimensions. Then the method loses it spirit of estimating $E(Y|X=x)$ by local averaging.



Curse of dimensionality





Parametric and structured models

The *linear* model is an important example of a parametric model:

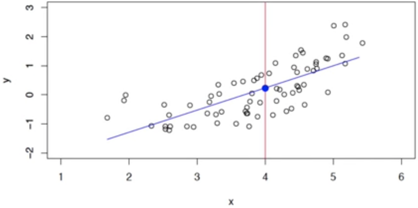
$$F_L(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p.$$

- A linear model is specified in terms of $p + 1$ parameters $\beta_0, \beta_1, \ldots, \beta_p$.
- We estimate the parameters by fitting the model to training data.
- Linear models are almost never correct. However, they are often good for interpretation and sometimes do better than complicated models in predicting.

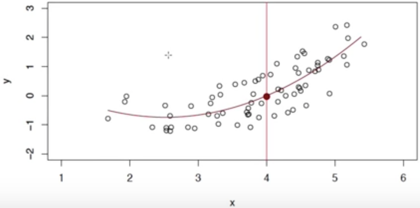
Some choice of models

Always have a *scatter plot* first if you have only one independent variable.

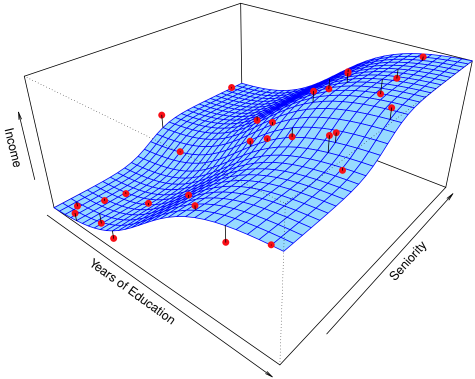
- A linear model $\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ gives a good fit here.



- A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ gives a good fit here.



Fitting the data – is it a good model?

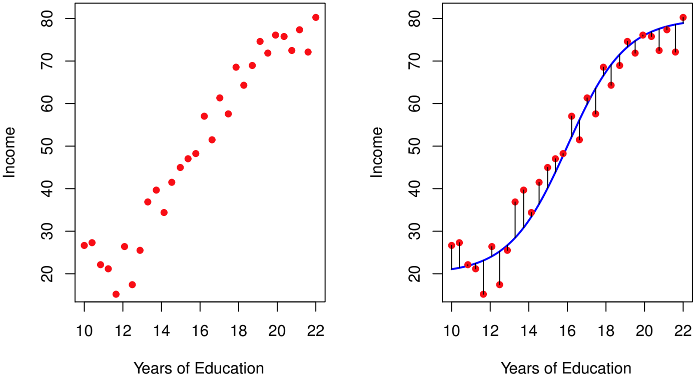


Some simulated example. Red dots are simulated values for *income* from the model

$$\textit{income} = f(\textit{education}, \textit{seniority}) + \epsilon$$

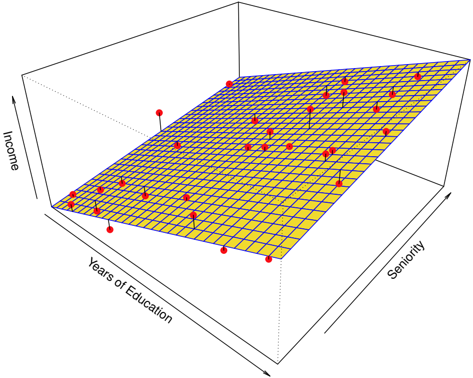
f is the blue surface.

Fitting the data – is it a good model?



One dimensional case. Fix the other independent variable as a constant.

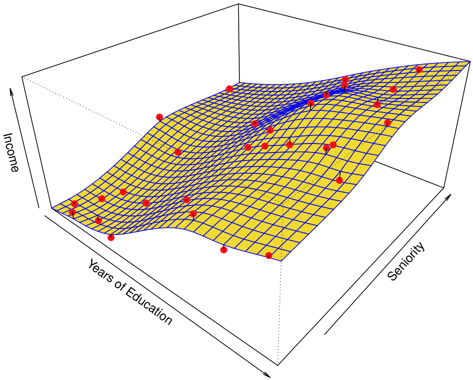
Fitting the data – is it a good model?



Linear regression model fit to the simulated data.

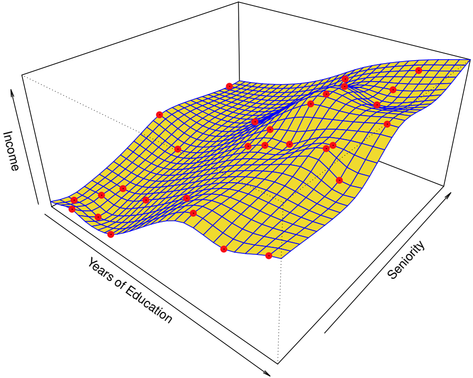
$$\hat{f}_L(\textit{education}, \textit{seniority}) = \hat{\beta}_0 + \hat{\beta}_1 \times \textit{education} + \hat{\beta}_2 \times \textit{seniority}$$

Fitting the data – is it a good model?



More flexible regression model $\hat{f}_S$(*education*, *seniority*) fit to the simulated data. Here the *thin-spline method* is used to fit a flexible surface. The roughness of the fit is also controllable (chapter 7).

Fitting the data – is it a good model?

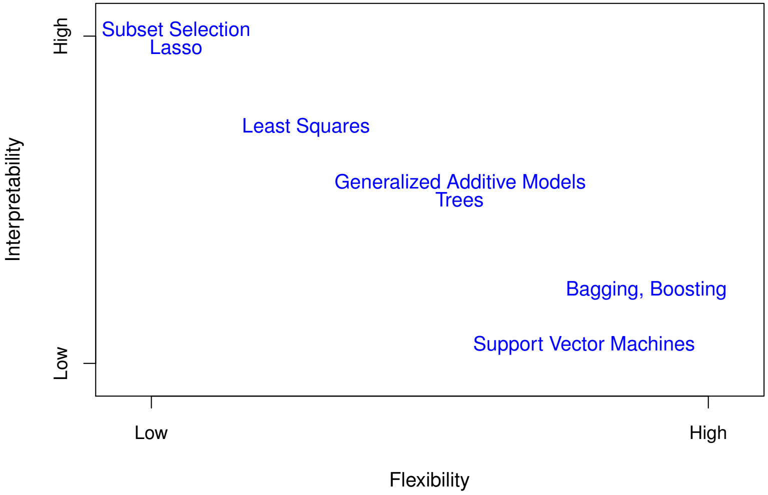


Even more flexible regression model $\hat{f}_S($*education*, *seniority*$)$ fit to the simulated data. Here the fitted model makes no errors on the training data. This is also known as *overfitting*.

Trade-offs

- Prediction accuracy versus interpretability.
  – Linear models are easy to interpret; thin-plate splines are not.
- Good fit versus over-fit or under-fit
  – How do we know when the fit is good enough?
- Parsimony versus black-box
  – We often prefer a simpler model involving fewer variables over a black-box predictor involving them all.

Trade-off of models



Assessing model accuracy

Suppose we fit a model $\hat{f}(x)$ to some training data $Tr = \{x_i, y_i\}_{i=1}^{N}$, and we wish to see how well it performs.

- We could compute the average squared prediction error over Tr:

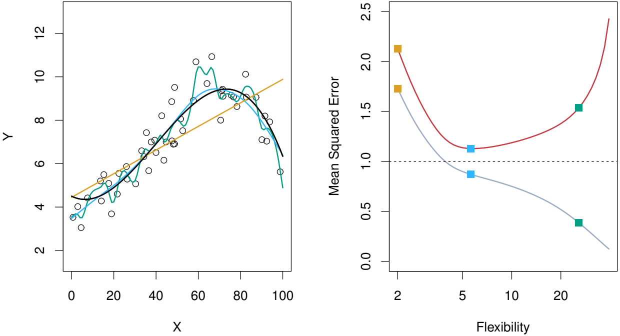
$$MSE_{Tr} = Ave_{i \in Tr}[y_i - \hat{f}(x_i)]^2$$

This may be biased toward more overfit models.

- Instead we should, if possible, compute it using fresh *test data* $Te = \{x_i, y_i\}_{i=1}^{M}$:

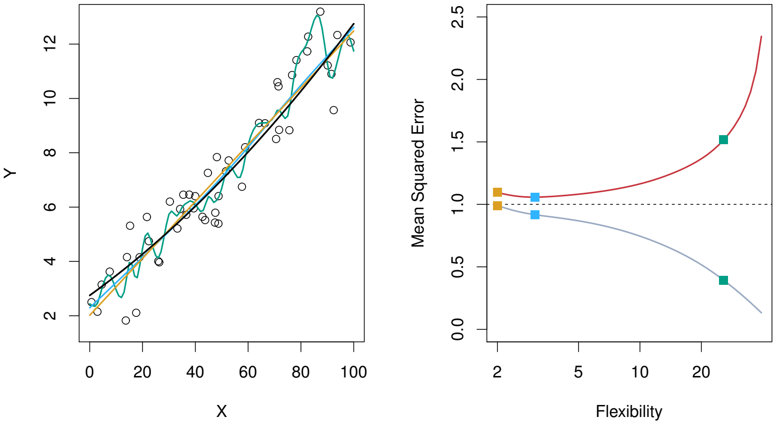
$$MSE_{Te} = Ave_{i \in Te}[y_i - \hat{f}(x_i)]^2$$

MSE and flexibility



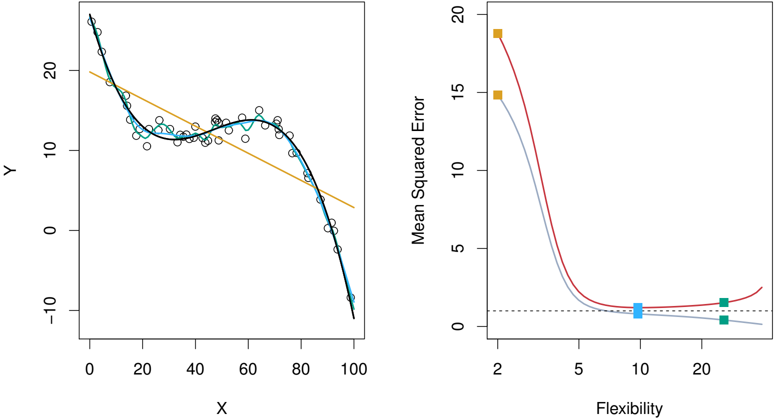
Black curve is truth, the data is simulated from the true model. Red curve on the right is MSE_{Te}, grey curve is MSE_{Tr}. Orange, blue, and green curves/squares correspond to fits of different flexibility.

MSE and flexibility



Here the truth is smoother (close to linear), so the smoother fit and linear model do really well.

MSE and flexibility



Here the truth is wiggly and the noise is low, so the more flexible fits do the best.

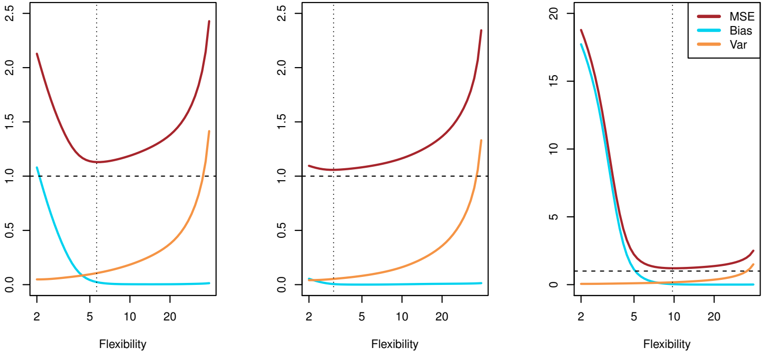
Bias-variance trade-off

- Suppose we have fit a model $\hat{f}(x)$ to some training data Tr, and let (x_0, y_0) be a test observation drawn from the population. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E(Y|X = x)$), then

$$E\big(y_0 - \hat{f}(x_0)\big)^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon).$$

- The expectation averages over the variability of y_0 as well as the variability in Tr. Note that $Bias(\hat{f}(x_0)) = E[\hat{f}(x_0] - f(x_0)$.
- Typically as the *flexibility* of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on average test error amounts to a *bias-variance trade-off*.

Bias-variance trade-off for the three examples

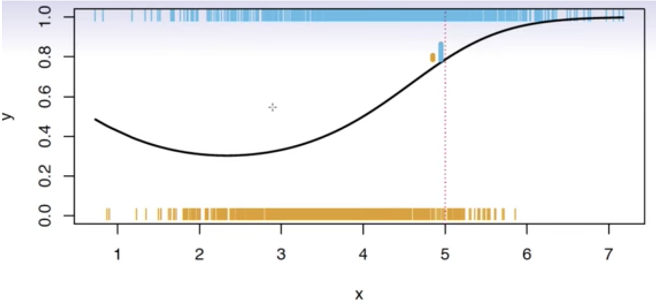


Classification problems

Here the response variable Y is *qualitative* – e.g. email is one of $C = (\textit{spam}, \textit{ham})$ (*ham* = good email), digit class is one of $C = \{0, 1, 2, \ldots, 9\}$. The goal is to:

- Build a classifier $C(X)$ that assigns a class label from C to a future unlabeled observation X.
- Access the uncertainty in each classification.
- Understand the roles of the different predictors among $X = (X_1, X_2, \ldots, X_P)$.

Binary classifier



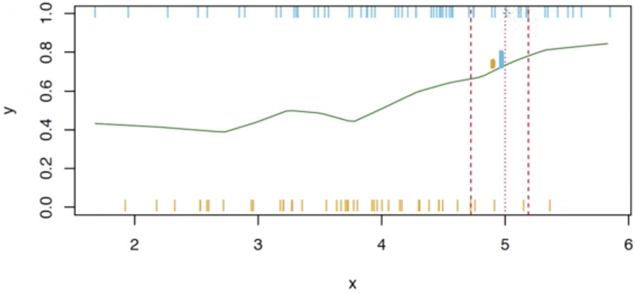
Is there an ideal $C(X)$? Suppose the K elements in C are numbered $1, 2, \ldots, K$. Let

$$p_k(x) = Pr(Y = k|X = x), k = 1, 2, \ldots, K.$$

These are the *conditional class probabilities* at x. Then the *Bayes optimal* classifier at x is

$$C(x) = j \text{ if } p_j(x) = \max\{p_j(x), p_2(x), \ldots, p_K(x)\}.$$

Binary classifier



Nearest-neighbor averaging can be used as before. But this also breaks when the dimension is large. However, the impact on $\hat{C}(x)$ is less than on $\hat{p}_k(x), k = 1, 2, \ldots, K$.

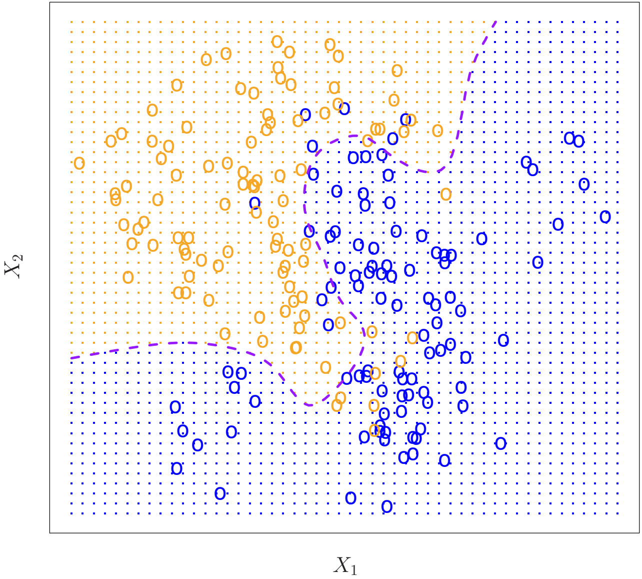
Classifications: some details

- Typically we measure the performance of $\hat{C}(x)$ using the misclassification error rate:

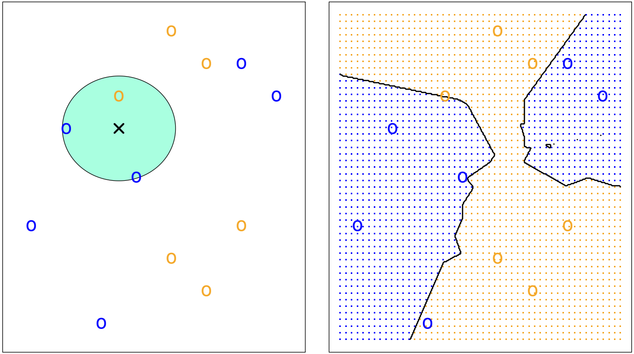
$$Err_{Te} = \ Ave_{i \in Te} I[y_i \neq \hat{C}(x_i)]$$

- The Bayes classifier (using the true $p_k(x)$) has smallest error
- Support vector machines build structured model for $C(x)$
- We will also build structured models for representing the $p_k(x)$. e.g. logistic regression, generalized additive models

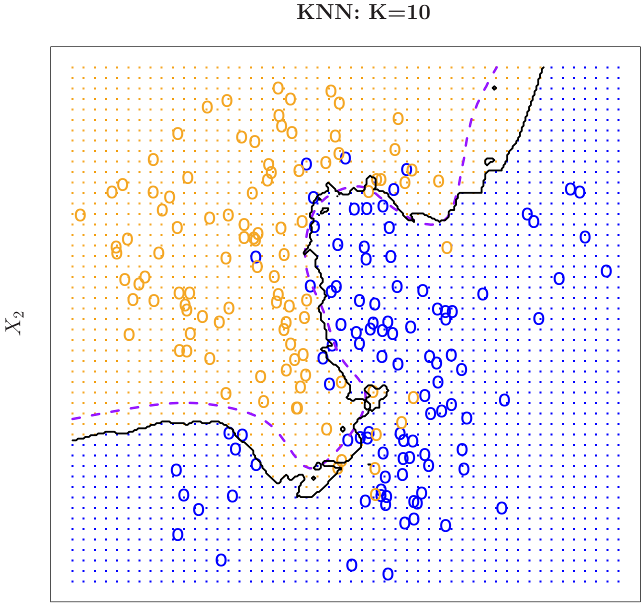
K-nearest neighbors in two dimensions



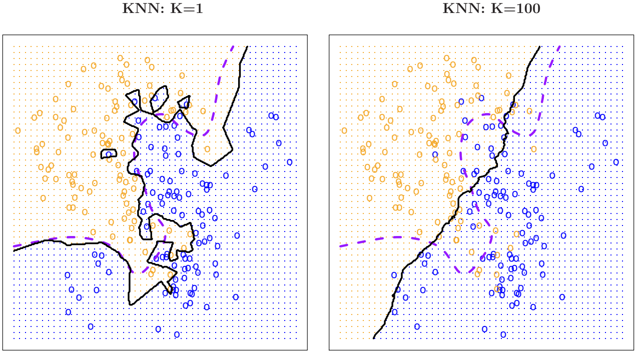
K-nearest neighbors in two dimensions



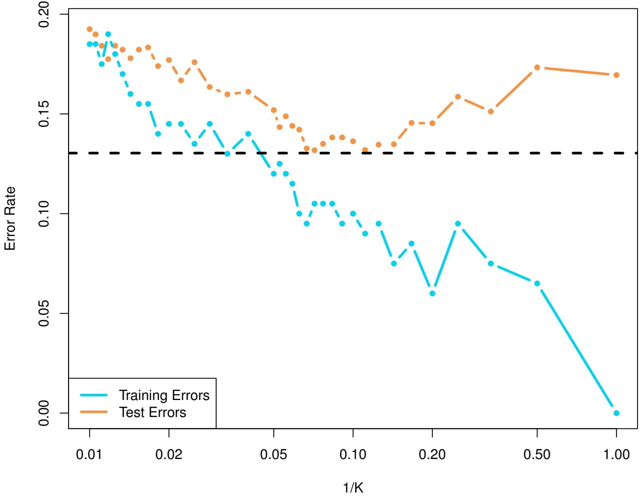
K-nearest neighbors in two dimensions



K-nearest neighbors in two dimensions



Performance of K-nearest neighbors method


Error Rate
1/K
Training Errors
Test Errors