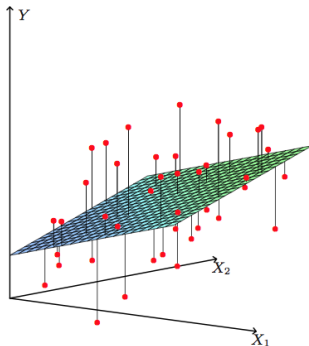


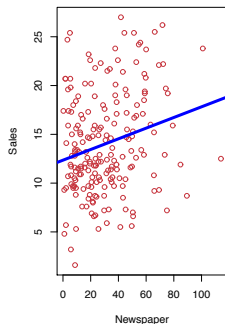
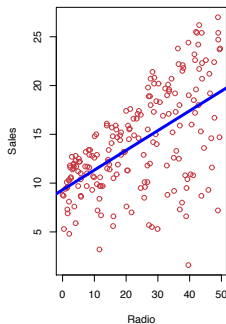
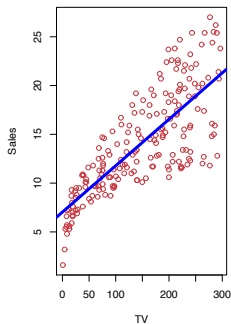
Statistical Methods for Data Mining

Kuangnan Fang

Xiamen University Email: xmufkn@xmu.edu.cn



What is Statistical Learning?



Shown are **Sales** vs **TV**, **Radio** and **Newspaper**, with a blue linear-regression line fit separately to each.

Can we predict **Sales** using these three?

Perhaps we can do better using a model

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

单变量回归存在缺失
变量问题
估计结果不一致。
有偏

Notation

Here **Sales** is a *response* or *target* that we wish to predict. We generically refer to the response as Y .

TV is a *feature*, or *input*, or *predictor*; we name it X_1 .

Likewise name **Radio** as X_2 , and so on.

We can refer to the *input vector* collectively as

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

Now we write our model as

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

where ϵ captures measurement errors and other discrepancies.

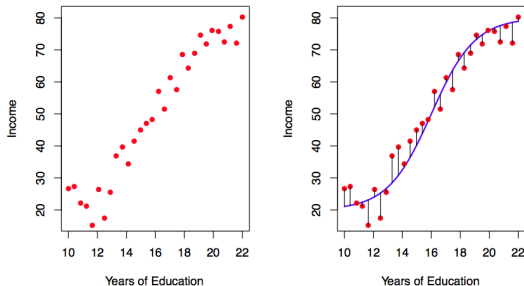


FIGURE 2.2. *The **Income** data set. Left: The red dots are the observed values of **income** (in tens of thousands of dollars) and **years of education** for 30 individuals. Right: The blue curve represents the true underlying relationship between **income** and **years of education**, which is generally unknown (but is known in this case because the data were simulated). The black lines represent the error associated with each observation. Note that some errors are positive (if an observation lies above the blue curve) and some are negative (if an observation lies below the curve). Overall, these errors have approximately mean zero.*

Why Estimate f

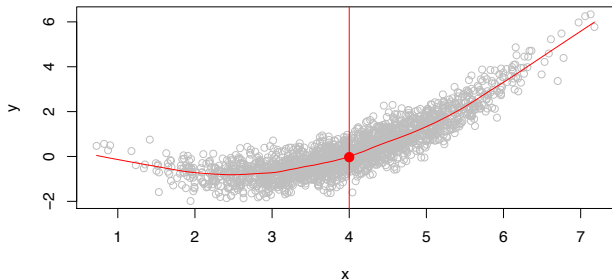
- Two main reasons: *prediction* and *inference*
- *prediction*: In many situations, a set of inputs X are readily available, but the output Y cannot be easily obtained. In this setting, since the error term averages to zero, we can predict Y using

$$\hat{Y} = \hat{f}(X)$$

where \hat{f} represents our estimate for f , and \hat{Y} represents the resulting prediction for Y . In this setting, \hat{f} is often treated as a black box, in the sense that one is not typically concerned with the exact form of f , provided that it yields accurate predictions for Y .

What is $f(X)$ good for?

- With a good f we can make predictions of Y at new points $X = x$.
- We can understand which components of $X = (X_1, X_2, \dots, X_p)$ are important in explaining Y , and which are irrelevant. e.g. **Seniority** and **Years of Education** have a big impact on **Income**, but **Marital Status** typically does not.
- Depending on the complexity of f , we may be able to understand how each component X_j of X affects Y .



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 many Y values at $X = 4$. A good value is

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

$E(Y|X = 4)$ means *expected value* (average) of Y given $X = 4$.

This ideal $f(x) = E(Y|X = x)$ is called the *regression function*.

The regression function $f(x)$

- Is also defined for vector X ; e.g.

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

The regression function $f(x)$

- Is also defined for vector X ; e.g.
$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$
- 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 g at all points $X = x$.

The regression function $f(x)$

- Is also defined for vector X ; e.g.
$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$
- 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 g at all points $X = x$.
- $\epsilon = Y - f(x)$ is the *irreducible* error — i.e. even if we knew $f(x)$, we would still make errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values.

The regression function $f(x)$

- Is also defined for vector X ; e.g.
 $f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$
- 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 g at all points $X = x$.
- $\epsilon = Y - f(x)$ is the *irreducible* error — i.e. even if we knew $f(x)$, we would still make errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values.
- For any estimate $\hat{f}(x)$ of $f(x)$, we have

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

Why Estimate f

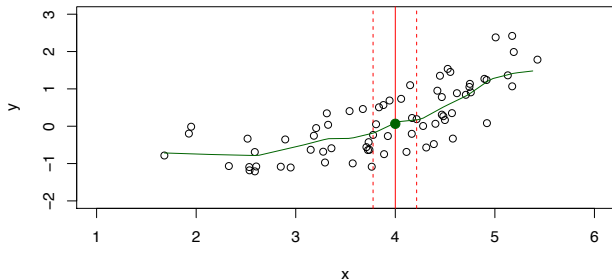
- *Inference*: We instead want to understand the relationship between X and Y , or more specifically, to understand how Y changes as a function of X_1, \dots, X_p . Now \hat{f} cannot be treated as a black box, because we need to know its exact form.
- Answering the following questions:
 1. Which predictors are associated with the response?
 2. What is the relationship between the response and each predictor?
 3. Can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?

How to estimate f

- Typically we have few if any data points with $X = 4$ exactly.
- So we cannot compute $E(Y|X = x)$!
- Relax the definition and let

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

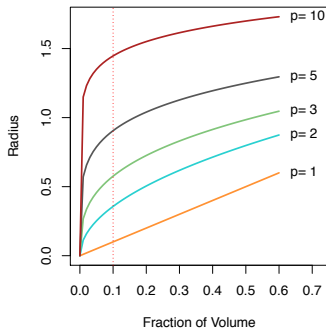
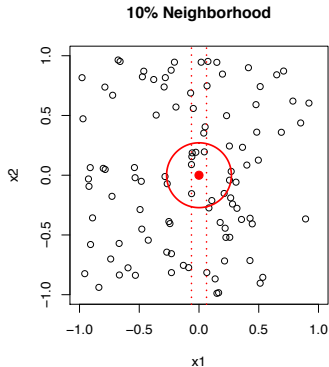
where $\mathcal{N}(x)$ is some *neighborhood* of x .



- Nearest neighbor averaging can be pretty good for small p — i.e. $p \leq 4$ and large-ish N .
- We will discuss smoother versions, such as kernel and spline smoothing later in the course.

- Nearest neighbor averaging can be pretty good for small p — i.e. $p \leq 4$ and large-ish N .
- We will discuss smoother versions, such as kernel and spline smoothing later in the course.
- Nearest neighbor methods can be *lousy* when p is large. Reason: the *curse of dimensionality*. Nearest neighbors tend to be far away in high dimensions.
 - We need to get a reasonable fraction of the N values of y_i to average to bring the variance down—e.g. 10%.
 - A 10% neighborhood in high dimensions need no longer be local, so we lose the spirit of estimating $E(Y|X = x)$ by local averaging.

The curse of dimensionality



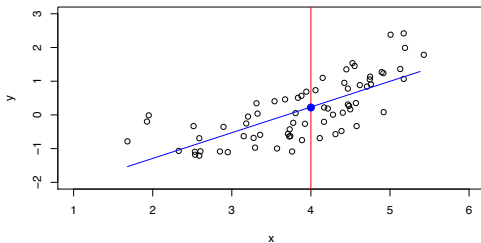
Parametric and structured models

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

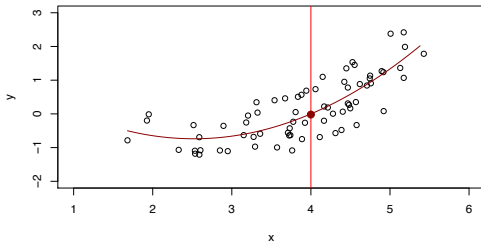
$$f_L(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots \beta_p X_p.$$

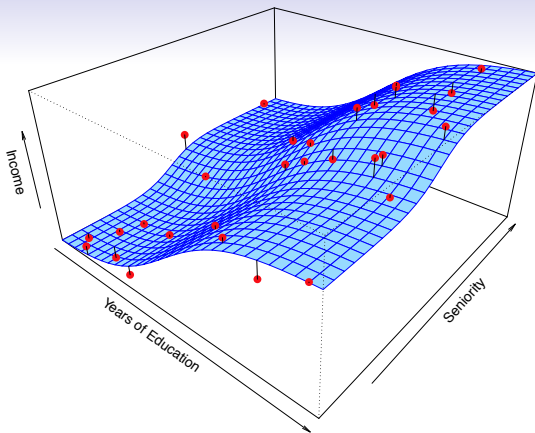
- A linear model is specified in terms of $p + 1$ parameters $\beta_0, \beta_1, \dots, \beta_p$.
- We estimate the parameters by fitting the model to training data.
- Although it is *almost never correct*, a linear model often serves as a good and interpretable approximation to the unknown true function $f(X)$.

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



A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ fits slightly better.

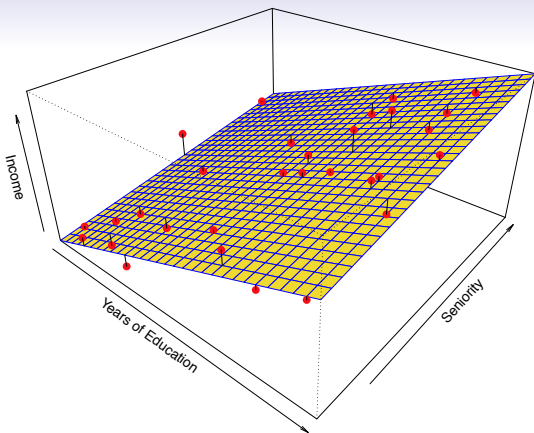




Simulated example. Red points are simulated values for **income** from the model

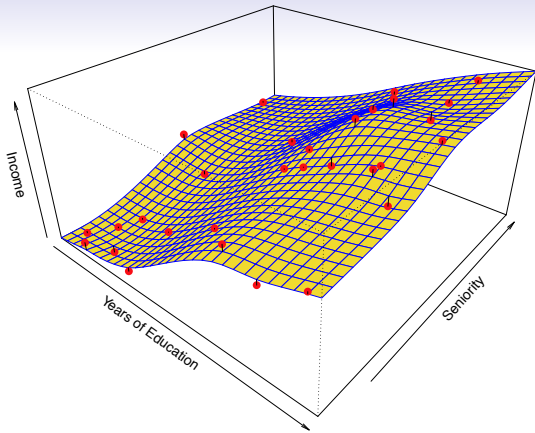
$$\text{income} = f(\text{education}, \text{seniority}) + \epsilon$$

f is the blue surface.

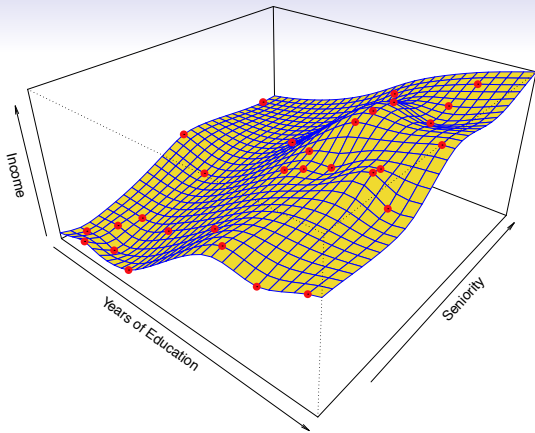


Linear regression model fit to the simulated data.

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



More flexible regression model $\hat{f}_S(\text{education}, \text{seniority})$ fit to the simulated data. Here we use a technique called a *thin-plate spline* to fit a flexible surface. We control the roughness of the fit (chapter 7).



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

Some trade-offs

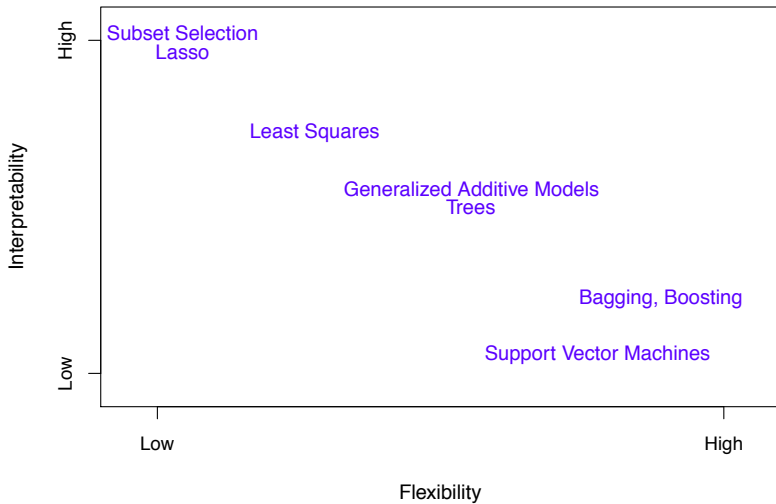
- Prediction accuracy versus interpretability.
 - Linear models are easy to interpret; thin-plate splines are not.

Some 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 just right?

Some 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 just right?
- Parsimony versus black-box.
 - We often prefer a simpler model involving fewer variables over a black-box predictor involving them all.



Supervised Versus Unsupervised Learning

- *Supervised Learning*: We wish to fit a model that relates the response to the predictors, with the aim of accurately predicting the response for future observations (prediction) or better understanding the relationship between the response and the predictors (inference). For example, linear regression and logistic regression, GAM, boosting, and support vector machines.
- *Unsupervised Learning*: we observe a vector of measurements x_i but no associated response y_i . The situation is referred to as unsupervised because we lack a response variable that can supervise our analysis. For example, clustering Analysis, PCA
- *Semi-supervised learning* : suppose we have n observations, For $m(m < n)$ of the observations, we have both predictor measurements and response measurement. For the remaining $n-m$ observations, we only have predictor measurements. In this setting, we wish to use a method that can incorporate the m observations for which response measurements are available as well as the $n-m$ observations.

Assessing Model Accuracy

- Why is it necessary to introduce so many different statistical learning approaches, rather than just a single best method?
- There is no free lunch in statistics: no one method dominates all others over all possible data sets. On a particular data set, one specific method may work best, but some other method may work better on a similar but different data set.
- Hence it is an important task to decide for any given set of data which method produces the best results.
- In order to evaluate the performance of a statistical learning method on a given data set, we need some way to measure how well its predictions actually match the observed data. That is, we need to quantify the extent to which the predicted response value for a given observation is close to the true response value for that observation.

Assessing Model Accuracy

Suppose we fit a model $\hat{f}(x)$ to some training data $\text{Tr} = \{x_i, y_i\}_1^N$, and we wish to see how well it performs.

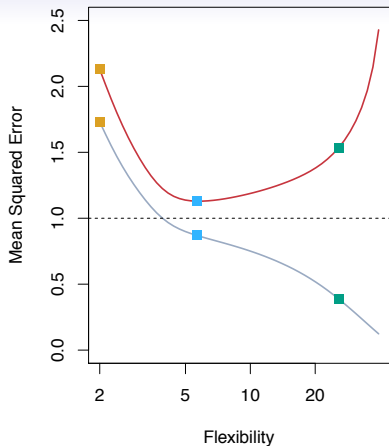
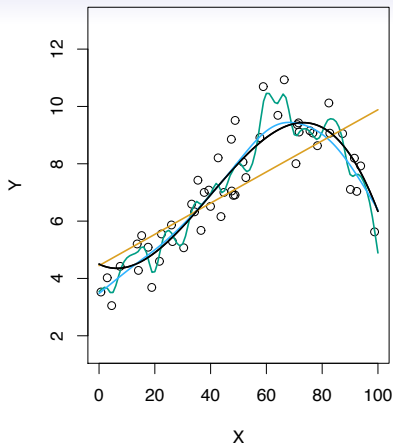
- We could compute the average squared prediction error over Tr :

$$\text{MSE}_{\text{Tr}} = \text{Ave}_{i \in \text{Tr}} [y_i - \hat{f}(x_i)]^2$$

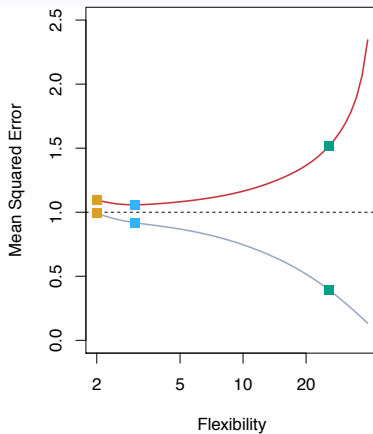
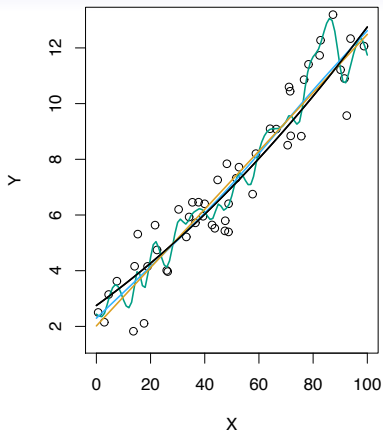
This may be biased toward more overfit models.

- Instead we should, if possible, compute it using fresh *test* data $\text{Te} = \{x_i, y_i\}_1^M$:

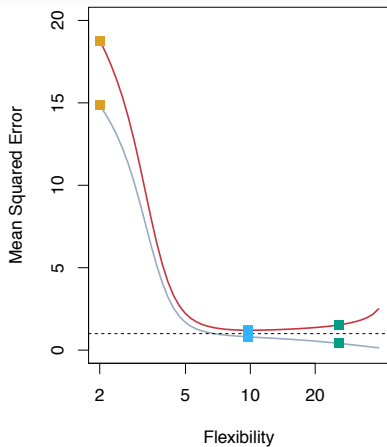
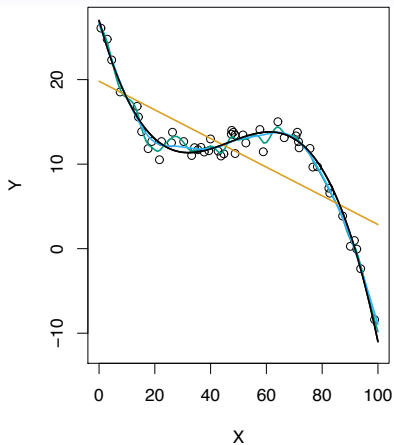
$$\text{MSE}_{\text{Te}} = \text{Ave}_{i \in \text{Te}} [y_i - \hat{f}(x_i)]^2$$



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



Here the truth is smoother, so the smoother fit and linear model do really well.



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

Bias-Variance Trade-off

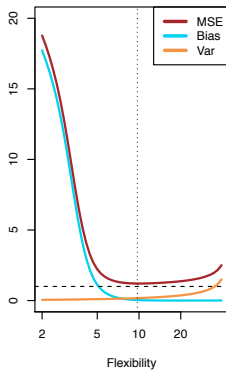
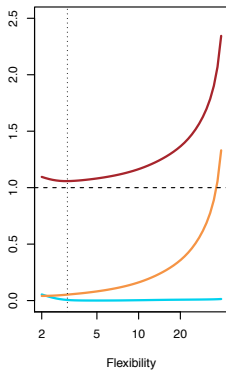
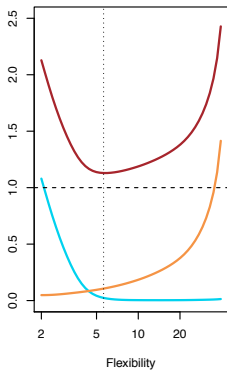
Suppose we have fit a model $\hat{f}(x)$ to some training data Tr , and let (x_0, y_0) be a test observation drawn from the population. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E(Y|X = x)$), then

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

The expectation averages over the variability of y_0 as well as the variability in Tr . Note that $\text{Bias}(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_0)$.

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*.

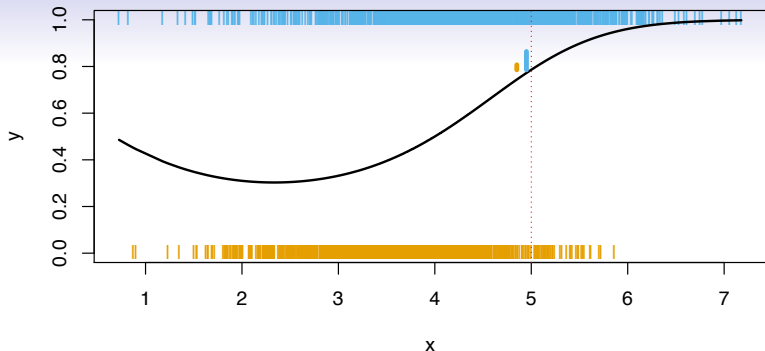
Bias-variance trade-off for the three examples



Classification Problems

Here the response variable Y is *qualitative* — e.g. email is one of $\mathcal{C} = (\text{spam}, \text{ham})$ (ham =good email), digit class is one of $\mathcal{C} = \{0, 1, \dots, 9\}$. Our goals are to:

- Build a classifier $C(X)$ that assigns a class label from \mathcal{C} to a future unlabeled observation X .
- Assess the uncertainty in each classification
- Understand the roles of the different predictors among $X = (X_1, X_2, \dots, X_p)$.

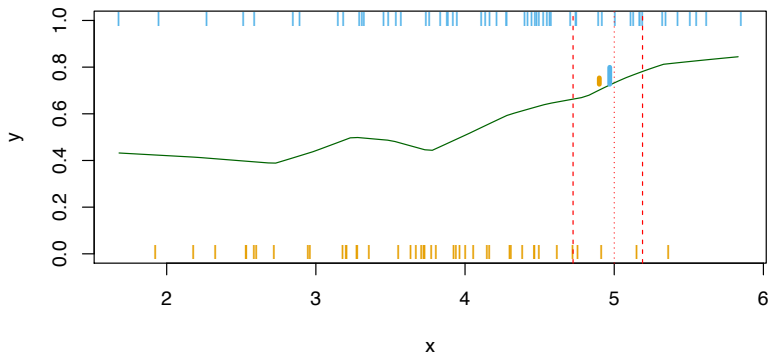


Is there an ideal $C(X)$? Suppose the K elements in \mathcal{C} are numbered $1, 2, \dots, K$. Let

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

These are the *conditional class probabilities* at x ; e.g. see little barplot at $x = 5$. Then the *Bayes optimal* classifier at x is

$$C(x) = j \text{ if } p_j(x) = \max\{p_1(x), p_2(x), \dots, p_K(x)\}$$



Nearest-neighbor averaging can be used as before.
Also breaks down as dimension grows. However, the impact on $\hat{C}(x)$ is less than on $\hat{p}_k(x)$, $k = 1, \dots, K$.

Classification: some details

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

$$\text{Err}_{\text{Te}} = \text{Ave}_{i \in \text{Te}} I[y_i \neq \hat{C}(x_i)]$$

- The Bayes classifier (using the true $p_k(x)$) has smallest error (in the population).

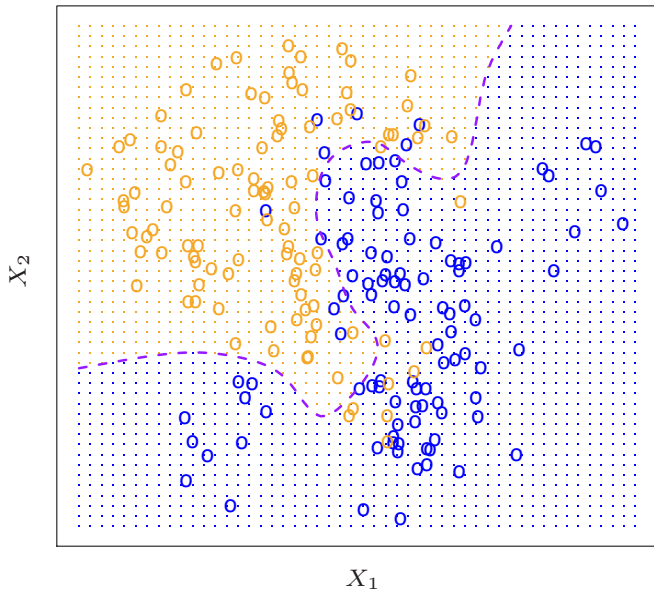
Classification: some details

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

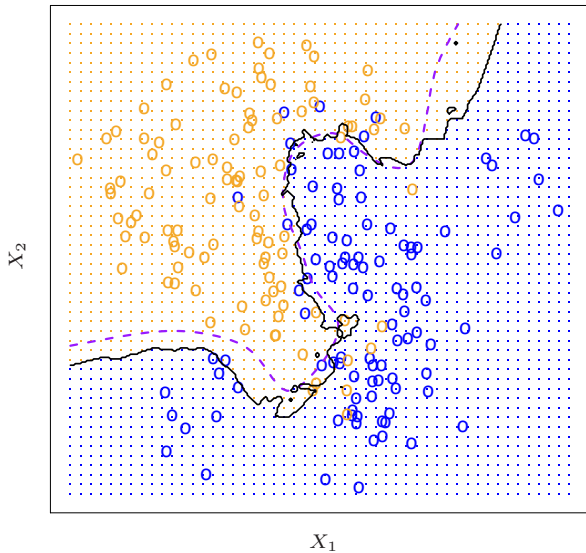
$$\text{Err}_{\text{Te}} = \text{Ave}_{i \in \text{Te}} I[y_i \neq \hat{C}(x_i)]$$

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

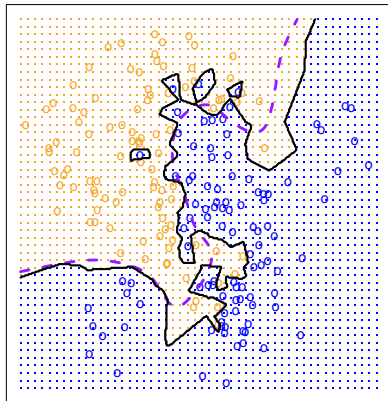
Example: K-nearest neighbors in two dimensions



KNN: K=10



KNN: $K=1$



KNN: $K=100$

