Introduction to Statistical Learning

Bin Li

IIT Lecture Series

What is statistical learning

- Statistical learning is the science of learning from the data using statistical methods.
  - Predict the price of a stock in 6 months from now, on the basis of company performance measures and economic data.
  - Predict whether a patient, hospitalized due to a heart attack, will have a second attack based on patient's demographic, diet and clinical measurements.
  - Identify the risk factors for prostate cancer.
  - Given a collection of text documents, we want to organize them according to their content similarities.
- Statistical learning plays a key role in data mining, artificial intelligence and machine learning.
- We can divide all statistical learning problems into **supervised** and **unsupervised** situations.
  - Supervised learning is where both the predictors, X_i's, and the response, Y_i, are observed (e.g. regression/classification).
  - In unsupervised learning, only X_i's are observed (e.g. clustering/market bastet analysis).

Handwritten Digit Recognition

- Data come from the handwritten ZIP codes on envelopes from U.S. postal mail.
- Each image is a segment from a five digit ZIP code, isolating a single digit.
- The images are 16×16 eight-bit graysclae maps, with each pixel ranging in intensity from 0 to 255.
- Images are nomralized to have approximately the same size and orientation.
- Task: predict from 16×16 matrix of pixel intensities, the identity of each image $(0, 1, \ldots, 9)$.
- Results:
  - Single layer neural network: 80.0%
  - Two layer network: 87%
  - Constrained neural network: 98.4%
  - Tangent distance with 1-NN: 98.9%
  - Support vector machine: 99.2%

Handwritten Digit Recognition (cont.)

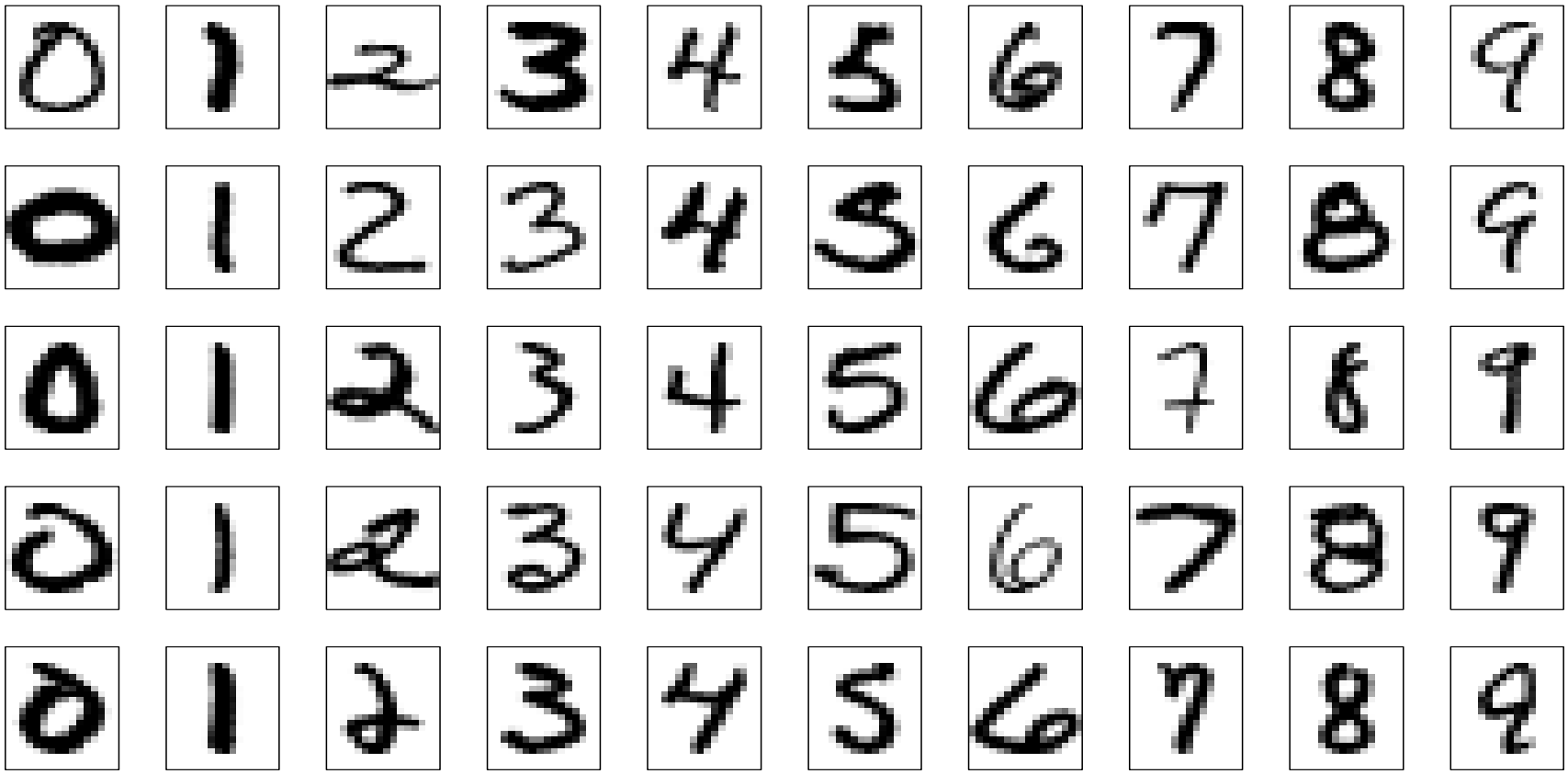
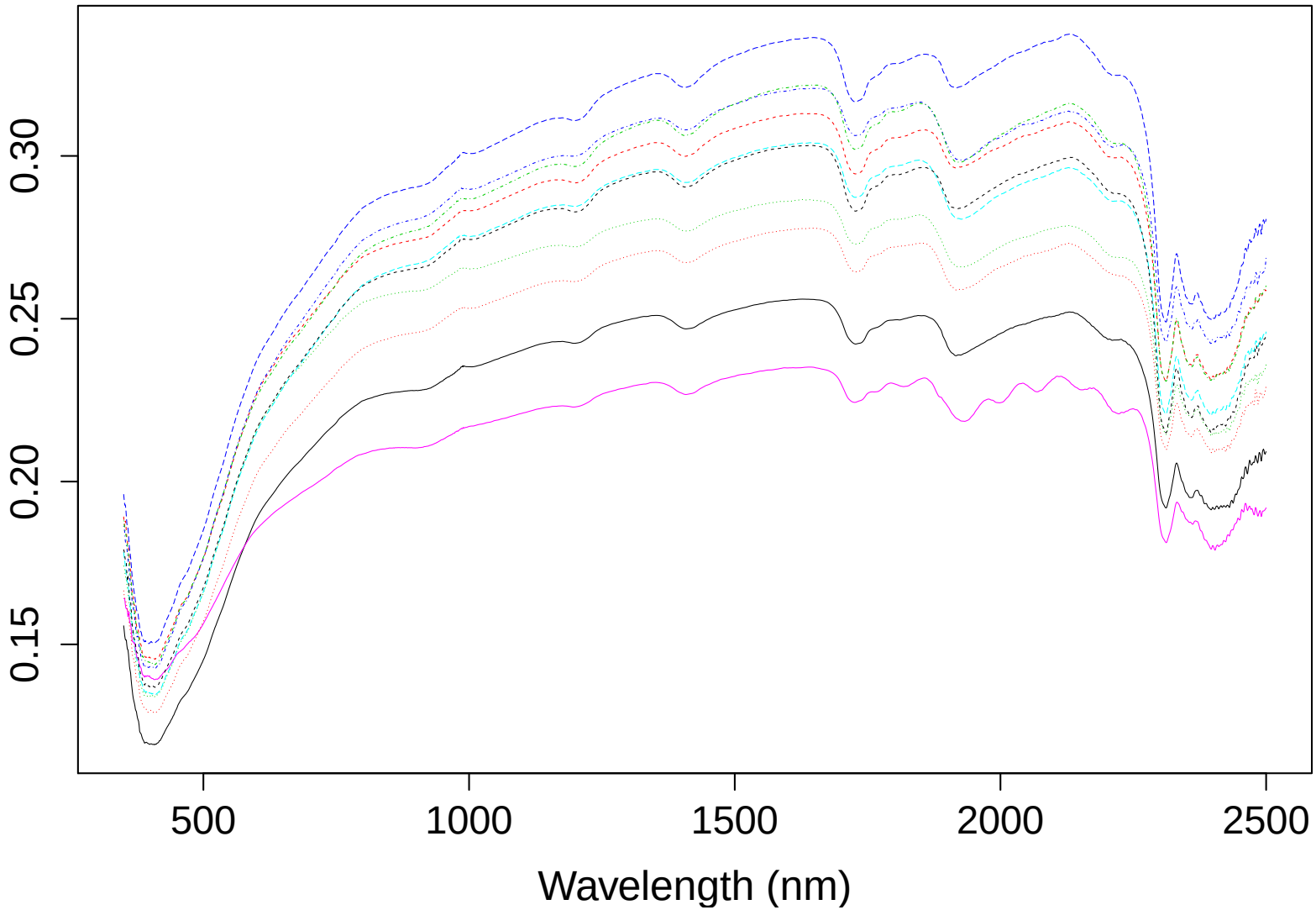


Figure from EOSL 2009

A Recent Project with Dr. Chakraborty



Statistical Science
2001, Vol. 16, No. 3, 199–231

Statistical Modeling: The Two Cultures

Leo Breiman

Abstract. There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools.

1. INTRODUCTION

Statistics starts with data. Think of the data as being generated by a black box in which a vector of input variables **x** (independent variables) go in one

The values of the parameters are estimated from the data and the model then used for information and/or prediction. Thus the black box is filled in like this:

Data and the Black Box

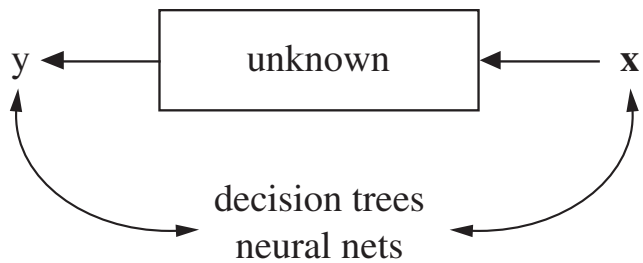
- **Prediction**. To be able to predict what the responses are going to be to future input variables.
- **Information**. To extract some information about how nature is associating the response variables to the input variables.
- Two different approaches towards the above goals.

Two Cultures

- **Data Modeling Culture** from statisticians.

  - Start with assuming a stochastic model for the black box;
  - Estimate parameters of the model from the data;
  - Use fitted model to do prediction;
  - Use hypothesis test and CI to do inference.

- **The Algorithmic Modeling Culture** from CS people.



  - Approximate the black box by some complicated function;
  - Estimate the function from some algorithm;
  - Both prediction and information are based on fitted functions;

Ozone Project

- Predictors: daily and hourly readings of over 450 meteorological variables for a period of seven years.
- Response: hourly values of ozone concentration in the Basin.
- Objetive: predict ozone concentration 12 hours in advance.
- Training set: the first five years data. Test set: the last two years data.
- Model: multiple linear regressions (including quadratic terms and interactions) with variable selection.
- Results: A failure. The false alarm rate of the final predictor was too high.
- *Q: What are the possible reasons make MLR unsuccessful in Ozone project?*

Chlorine Project

- Predictors: mass spectrum predictor with molecular weight ranges from 30 to over 10,000.
- Response: contains chlorine or not.
- Training set: 25,000 compounds with known chemical structure and mass spectra. Test set: 5,000 known compounds.
- Model: Linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and decision trees.
- Results: LDA and QDA were difficult to adapt to the variable dimensionality. Decision tree with 1,500 yes-no questions: success with 95% prediction accuracy.
- *Q: What are the possible reasons make tree successful in Chlorine project?*

Perceptions on Statistical Analysis

- Focus on finding a good solution, that's what consultants get paid for.
- Live with the data before you plunge into modeling.
- Search for a model that gives a good solution, either algorithmic or data.
- Predictive accuracy on test sets is the criterion for how good the model is.
- Computers are an indispensable partner. Programming is a necessary skill for statisticians.

What research in the university was like?

- A friend of Leo Breiman, a prominent statistician from the Berkeley Statistics Department, visited me in Los Angeles in the late 1970s. After I described the decision tree method to him, his first question was, "What's the model for the data?"
- In Annals of Statistics and JASA, almost every article contains a statement of the form:
  Assume that the data are generated by the following model ...
- Consider data modeling as the template for statistical analysis.
- The conclusions are about the model's but not the nature's mechanism.
- If the model is a poor emulation of nature, the conclusions maybe wrong.

A Study for Gender Discrimination

A study was done several decades ago by a well-known member of a university statistics department to assess whether there was gender discrimination in the salaries of the faculty.

All personnel files were examined and a data base set up which consisted of salary as the response variable and 25 other variables which characterized academic performance. Such as papers published, quality of journals published in, teaching record, evaluations, etc.

Gender appears as a binary predictor variable.

A linear regression was carried out on the data and the gender coefficient was significant at the 5% level. This was believed as strong evidence of sex discrimination.

A Study for Gender Discrimination (cont.)

- Can the data gathered answer the question posed?
- Is inference justified when your sample is the entire population?
- Should a data model be used?
- The deficiencies in analysis occurred because the focus was on the model and not on the problem.

Problems in Current Data Modeling

- The linear regression model led to many erroneous conclusions that appeared in journal articles waving the 5% significance level without knowing whether the model fit the data.
- The author set up a simulated regression problem in seven dimensions with a controlled amount of nonlinearity. Standard tests of goodness-of-fit (i.e. lack-of-fit test) did not reject linearity until the nonlinearity was extreme.
- An acceptable residual plot does not imply that the model is a good fit to the data.
- Published applications to data often show little care in checking model fit ... The question of how well the model fits the data is of secondary importance compared to the construction of an ingenious stochastic model.

Limitations of Data Modeling

- Enforcing the form of the model in data modeling.
- Relatively low prediction accuracy on data generated from complex systems.
- Old saying: "If all a man has is a hammer, then every problem looks like a nail."
- Approaching problems by looking for a data model imposes an a priori straight jacket that restricts the ability of statisticians to deal with a wide range of statistical problems.
- Takeaway message: to solve a wider range of data problems, we need a larger set of tools!

Estimating unknown function f

- Suppose we observe Y_i and $\mathbf{X}_i = (X_{i1}, X_{i2}, \ldots, X_{ip})$ for $i = 1, \ldots, n$.
- We believe that there is a relationship between Y and at least one of the X's. So we model the relationship as

$$Y_i = f(\mathbf{X_i}) + \epsilon_i \quad \text{with} \quad E\{\epsilon_i\} = 0,$$

where f is an unknown function and ϵ is a random error.

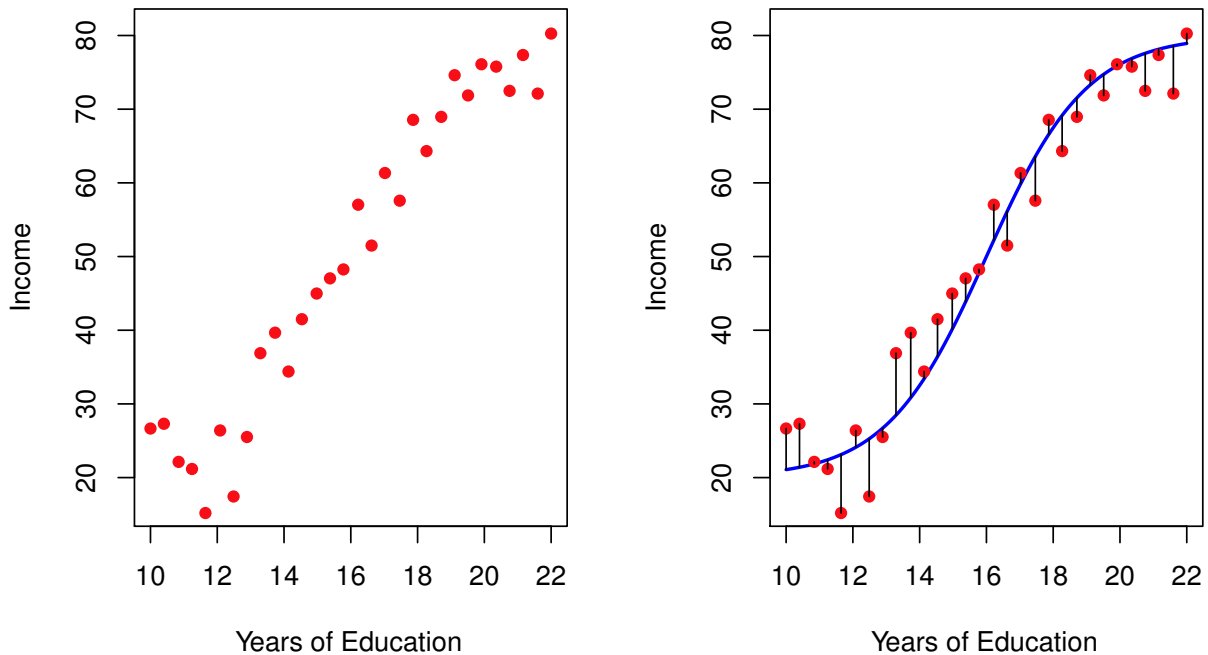


Figure from ISLR 2013

Income vs. education and seniority

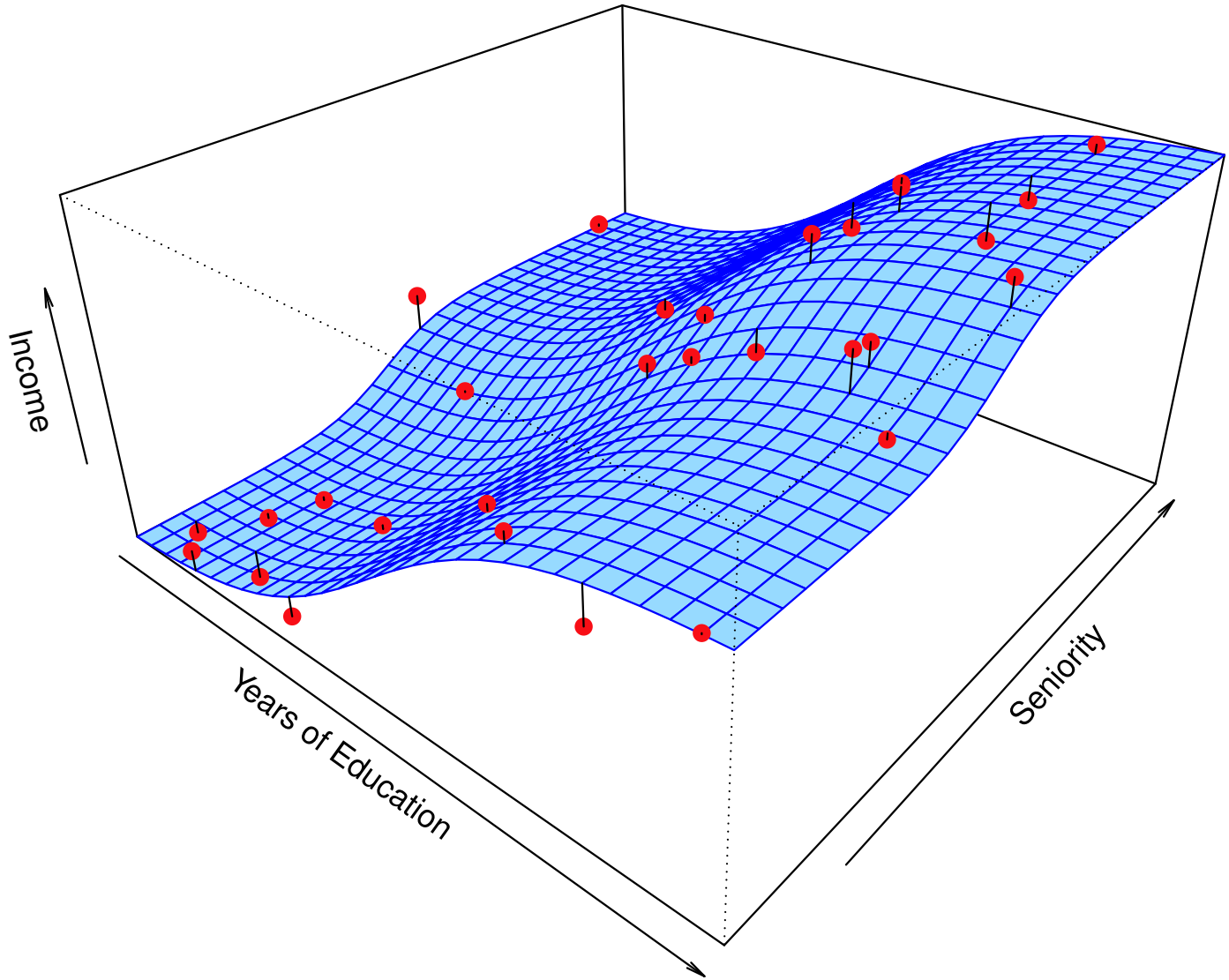
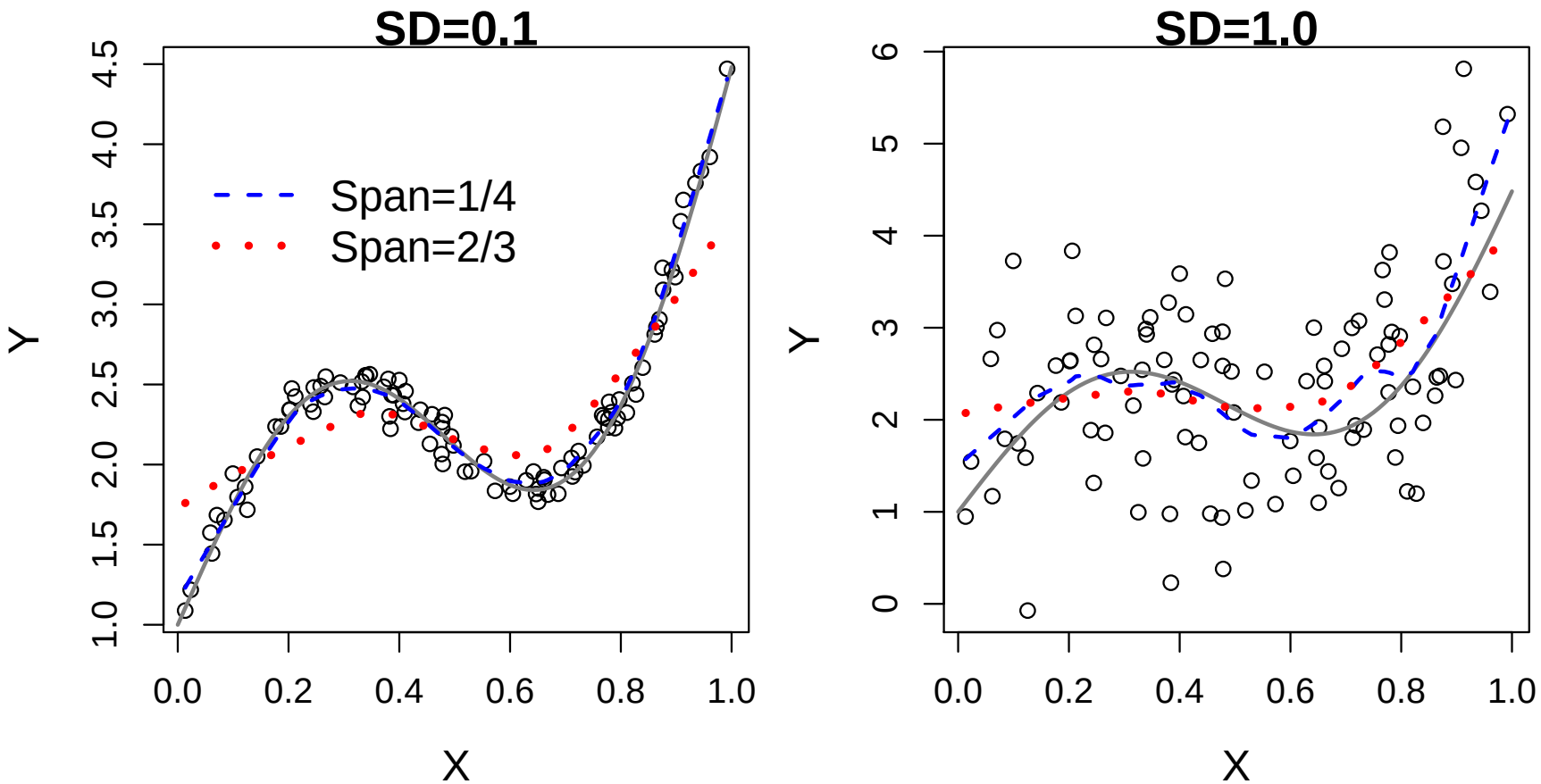


Figure from ISLR 2013

Estimating unknown function f (cont.)

- The accuracy of estimating f depends on
  - the size of variation for the ϵ_i's.
  - the complexity of fitted function $\hat{f}$



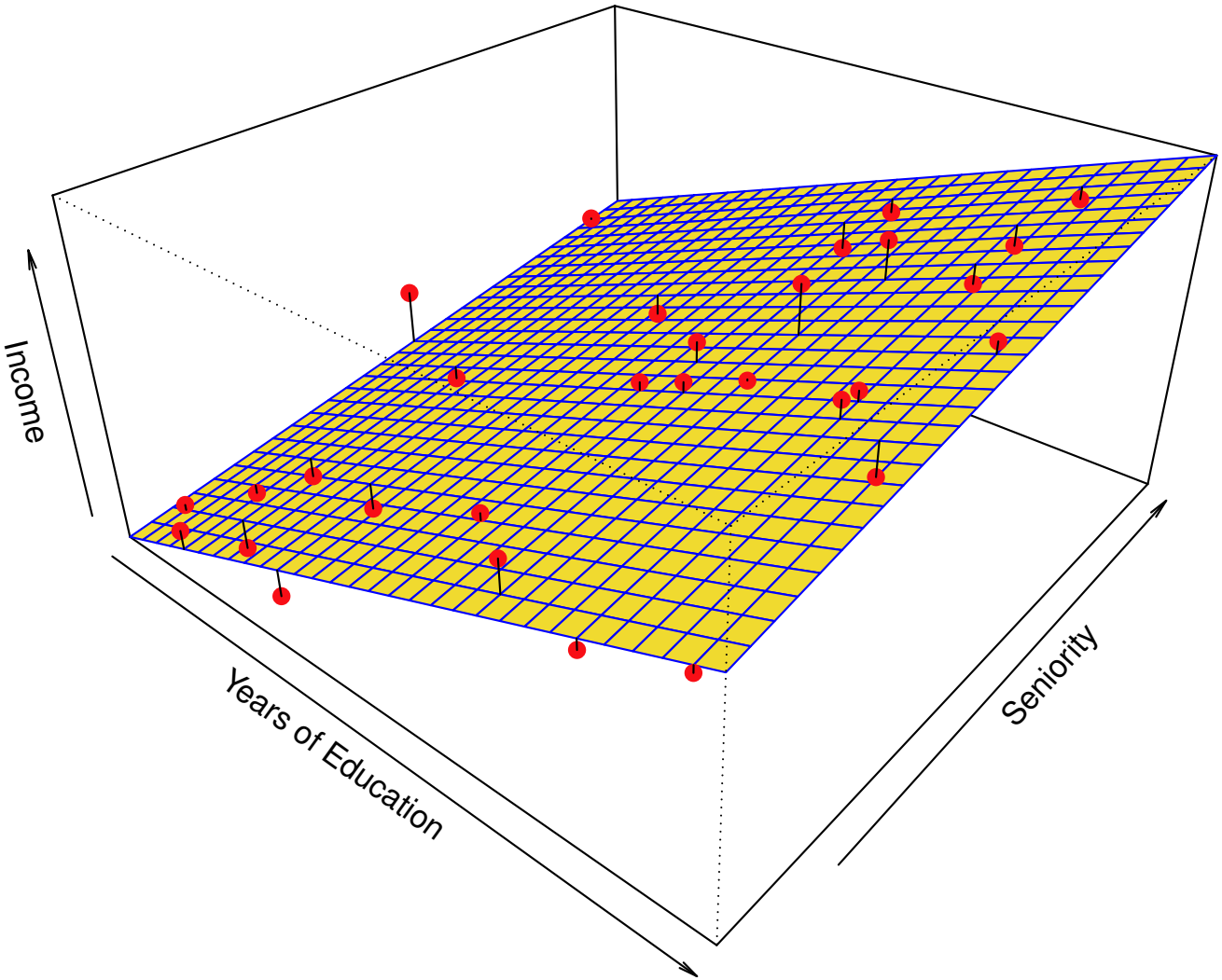
Why do we estimate f?

- Two main reasons: prediction and inference.
  - Make accurate predictions of Y based on a new value of X.
  - Which particular predictors actually affect the response?
  - Is the relationship positive or negative?
  - Is the relationship a simple linear one or is it more complicated etc.?
- Two examples:
  - Interested in predicting how much money an individual will donate based on observations from 90,000 people on which we have recorded over 400 different characteristics. For a given individual should I send out a mailing?
  - Wish to predict median house price based on 14 variables. Understand which factors have the biggest effect on the response and how big the effect is. For example how much impact does a river view have on the house value etc.

How Do We Estimate f?

- Use the training data $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ and a statistical method to estimate f.
- Two groups of statistical learning methods:
  - Parametric methods:
    - Make some assumption about the functional form of f (e.g. MLR).
    - Pros: estimating $f \Longrightarrow$ estimating a set of parameters (relatively easy task). Easy to interpret the model.
    - Cons: The form of model is too rigid. Low prediction accuracy when f is complicated.
  - Non-parametric methods:
    - Do not make explicit assumption about the functional form of f (e.g. neural network, tree).
    - Pros: accurately fit a wider range of possible shapes of f.
    - Cons: Large number of observations is required to obtain an accurate estimate of f.

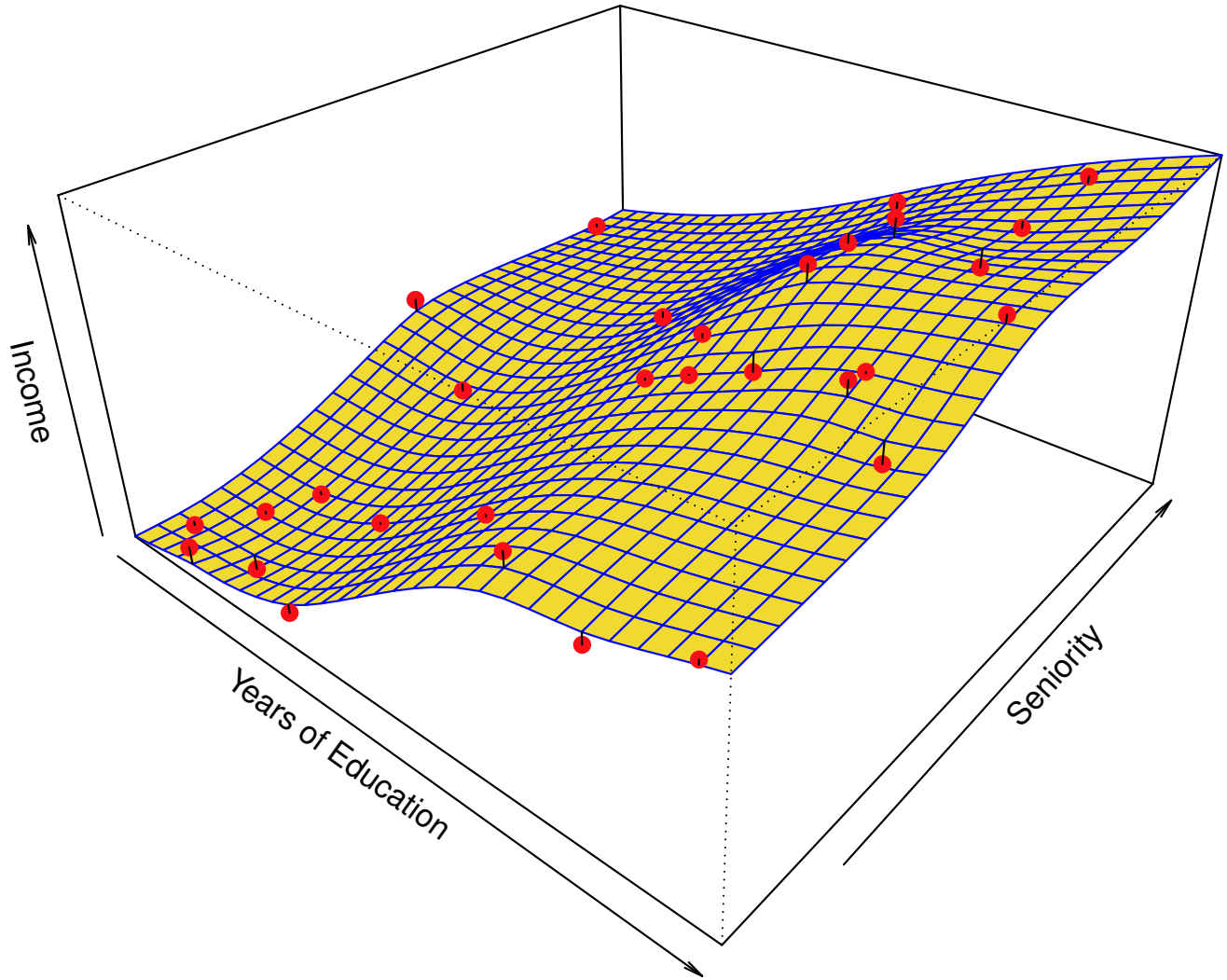
A linear regression estimate



Even if the standard deviation is low, we will still get a bad answer if we use the wrong model.

Figure from ISLR 2013

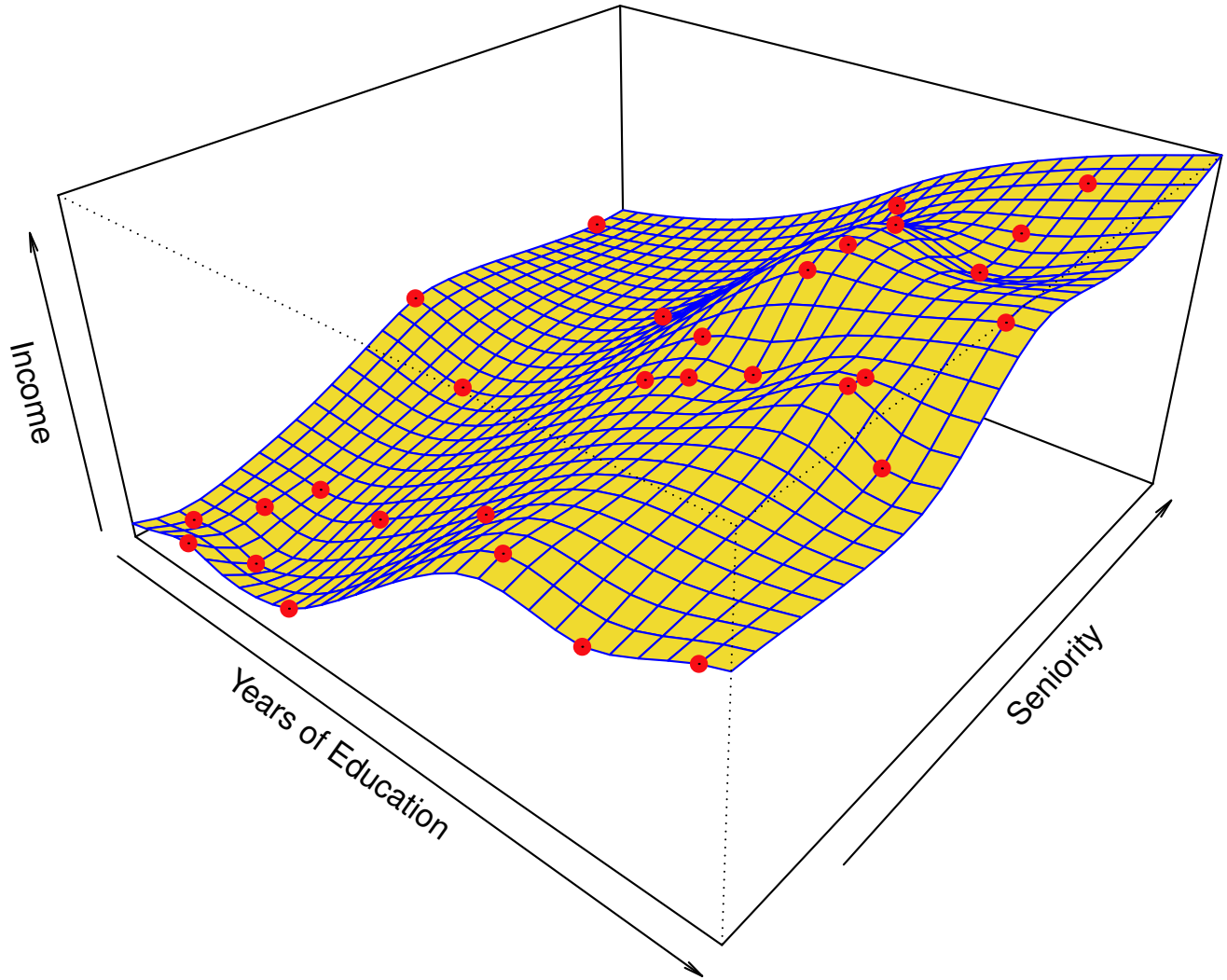
A thin-plate spline estimate



Non-linear regression methods are more flexible and can potentially provide more accurate estimates.

Figure from ISLR 2013

A poor estimate



Non-linear regression methods can also be too flexible and produce poor estimates for f.

Figure from ISLR 2013

Trade-off between model flexibility and interpretability

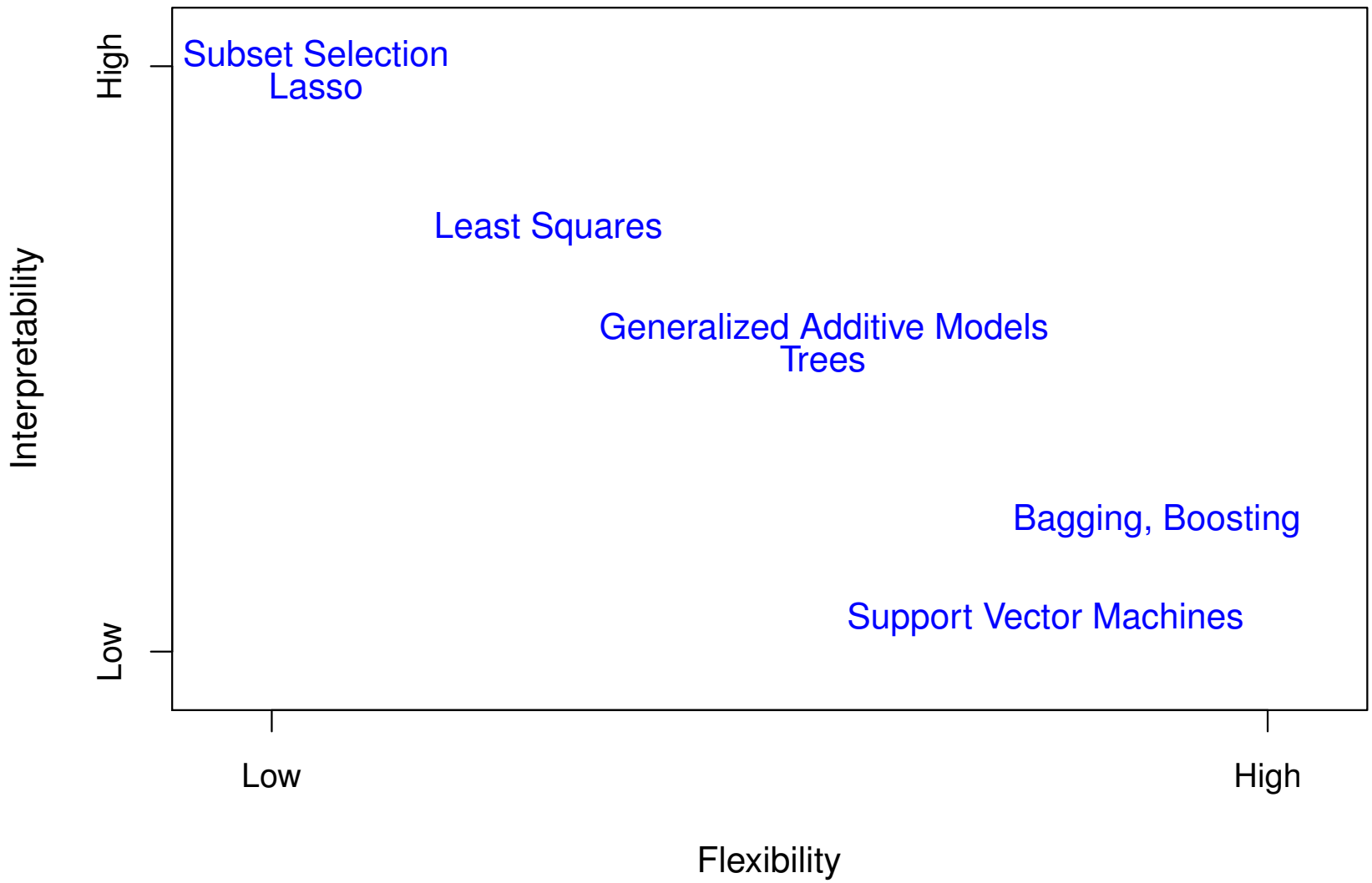
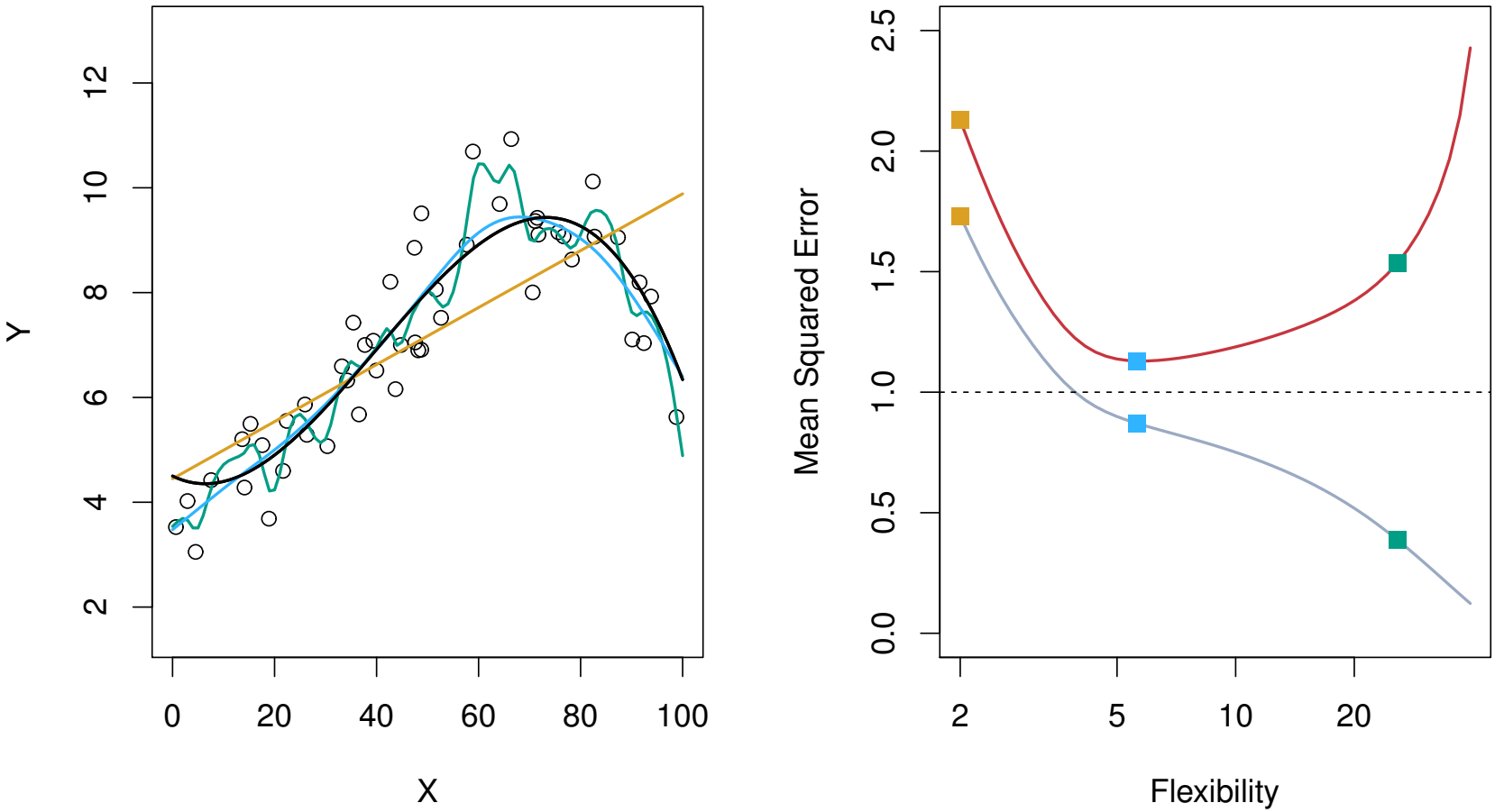


Figure from ISLR 2013

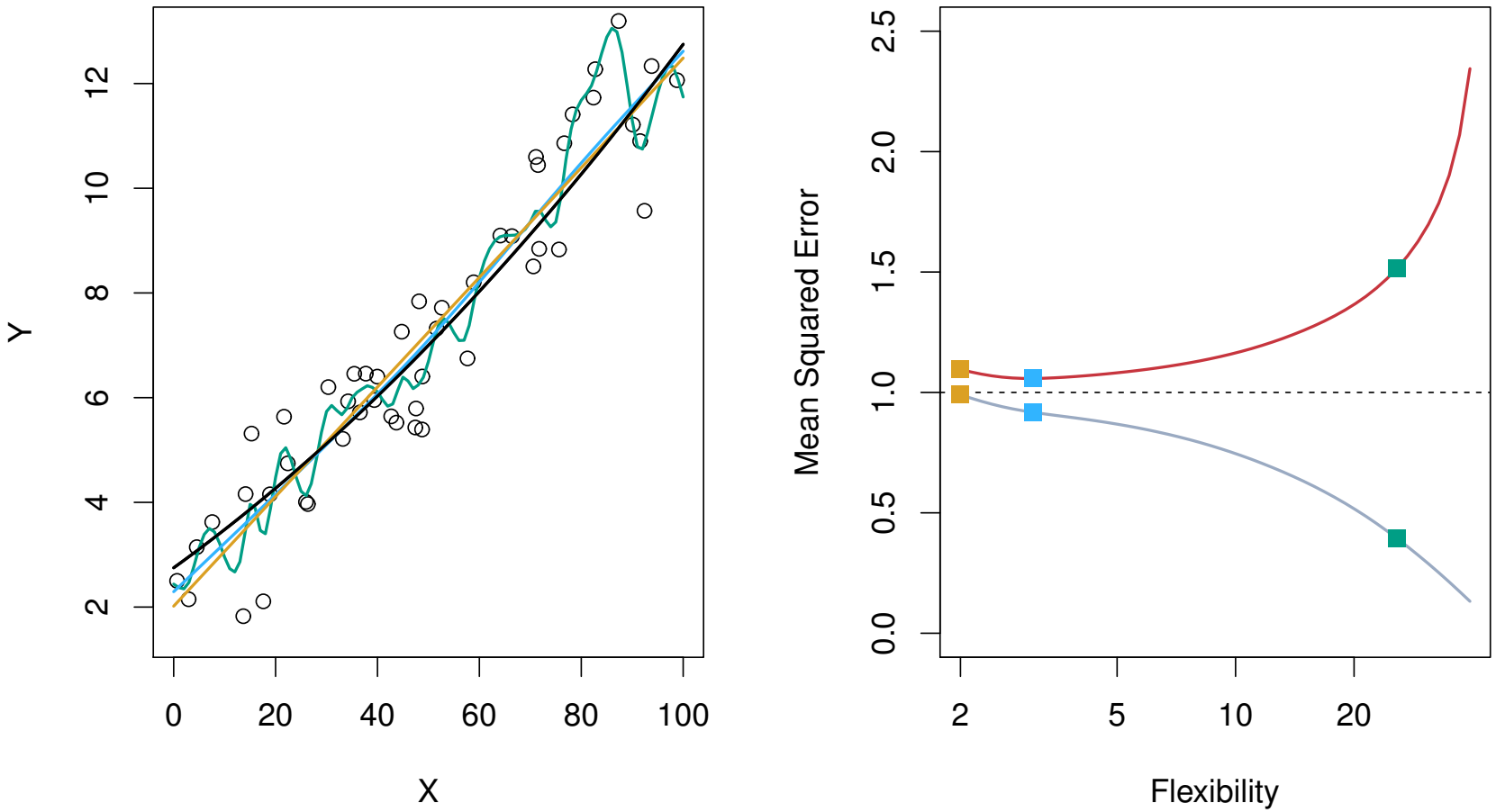
Training vs. test error: Example 1



Left: LR (orange), two smoothing spline fits (blue and green). Right: training MSE (grey), testing MSE (red), minimum possible test MSE (dash).

Figure from ISLR 2013

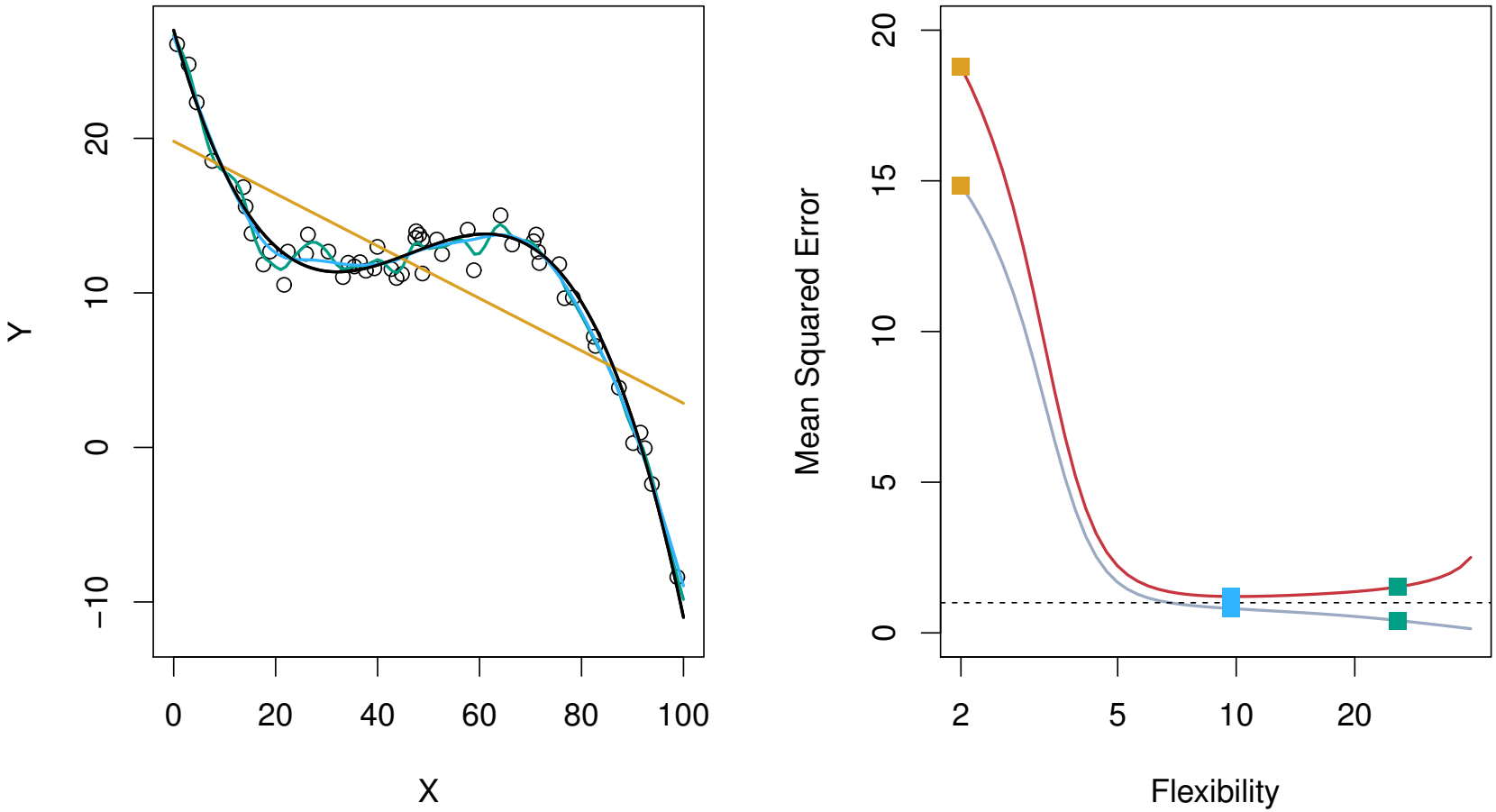
Example 2 (f is close to linear)



Left: LR (orange), two smoothing spline fits (blue and green). Right: training MSE (grey), testing MSE (red), minimum possible test MSE (dash).

Figure from ISLR 2013

Example 3 (f is far from linear)



Left: LR (orange), two smoothing spline fits (blue and green). Right: training MSE (grey), testing MSE (red), minimum possible test MSE (dash).

Figure from ISLR 2013

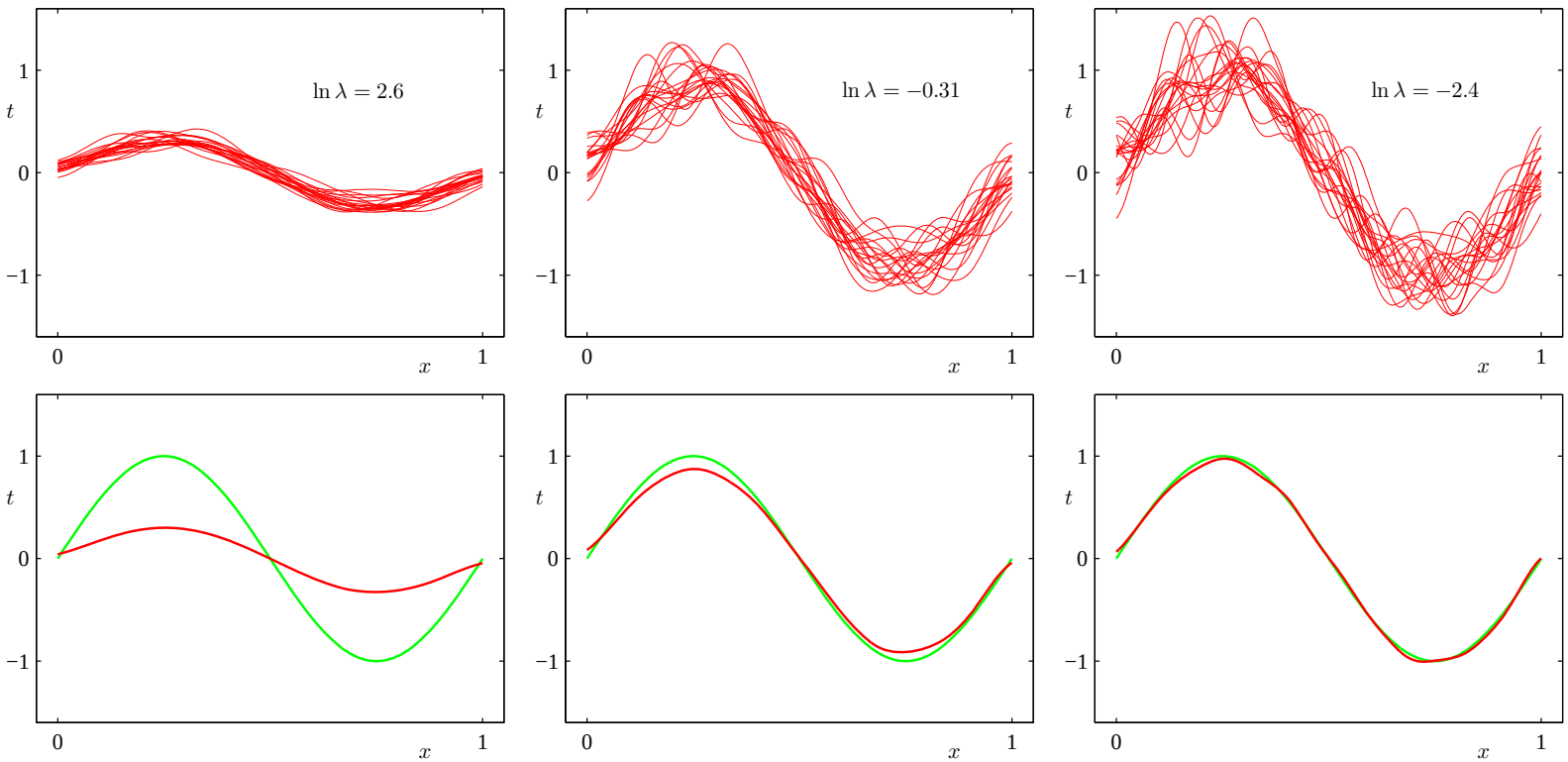
Bias variance tradeoff

- Two competing forces govern the choice of learning method, i.e. **bias** and **variance**.
- *Bias* refers to the error that is introduced by modeling a real life problem (that is usually extremely complicated) by a much simpler problem.
  - For example, linear regression assumes that there is a linear relationship between Y and X, which is unlikely in real life.
  - In general, the more flexible/complex a method is the less bias it will have.
- *Variance* refers to how much your estimate for f would change by if you had a different training data set.
  - Generally, the more flexible a method is the more variance.
  - In general, the more flexible/complex a method is the less bias it has.
- It can be shown the expected MSE for a new Y at x^{new} is:

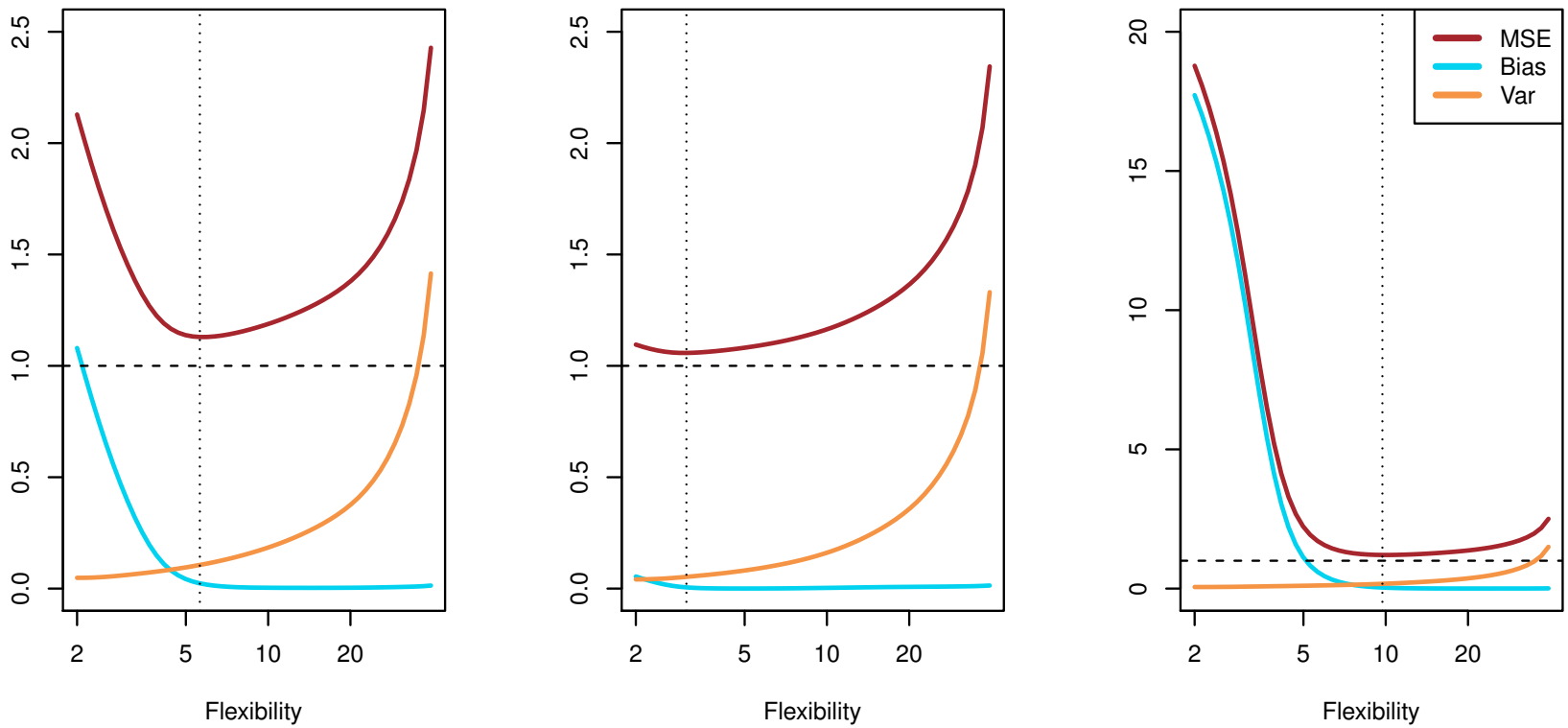
$$E[\text{MSE}(x^{new})] = \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}$$

Bias-variance tradeoff in splines

- 100 datasets with $N = 25$ points each
- Fit a model with 24 Gaussian basis functions
- Use regularized least squares with varying lambda



Bias, variance and MSE curves in example 1-3



Squared bias (blue), variance (orange) and test MSE (red) for example 1-3.
Vertical dotted line is the optimal flexibility level with the minimum test MSE.

Figure from ISLR 2013

The classification setting

- For a classification problem we can use the error rate i.e.

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

  The error rate represents the misclassifications rate.
- The Bayes error rate refers to the lowest possible error rate that could be achieved if somehow we knew exactly what the "true" probability distribution of the data looked like.
- By the Bayes rule:

$$\hat{f}(x) = \arg\max_k Pr(y = k|X = x).$$

- Decision boundary between class k and l is determined by the equation:

$$Pr(y = k|X = x) = Pr(y = l|X = x).$$

- In real life problems the Bayes error rate can't be calculated exactly.

K-Nearest Neighbors (KNN)

- k Nearest Neighbors is a flexible approach to estimate the Bayes classifier.
- For any given X we find the k closest neighbors to X in the training data, and examine their corresponding Y.
- If the majority of the Y's are orange we predict orange otherwise guess blue.
- The smaller that k is the more flexible the method will be.

KNN example with $k = 3$

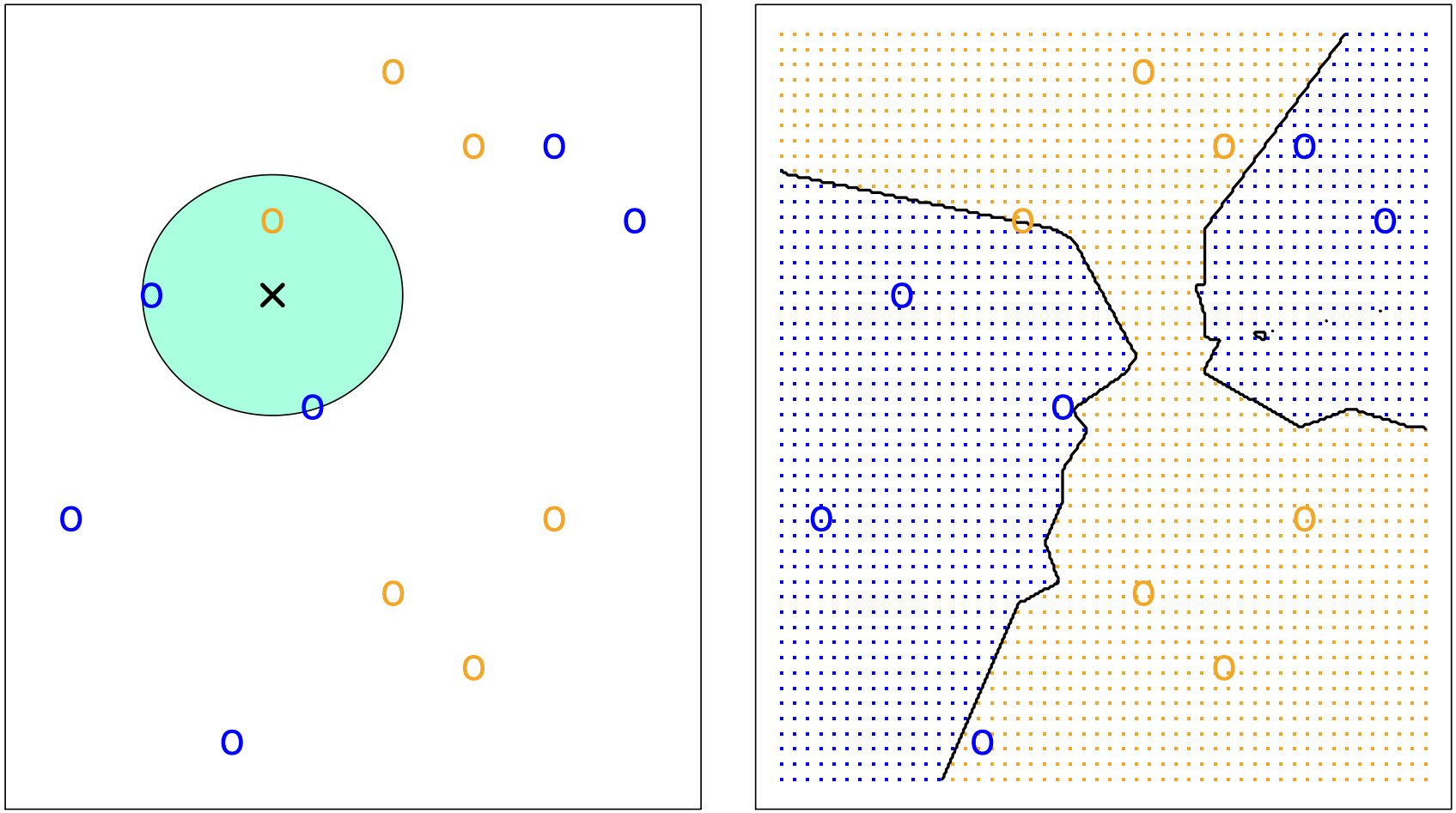
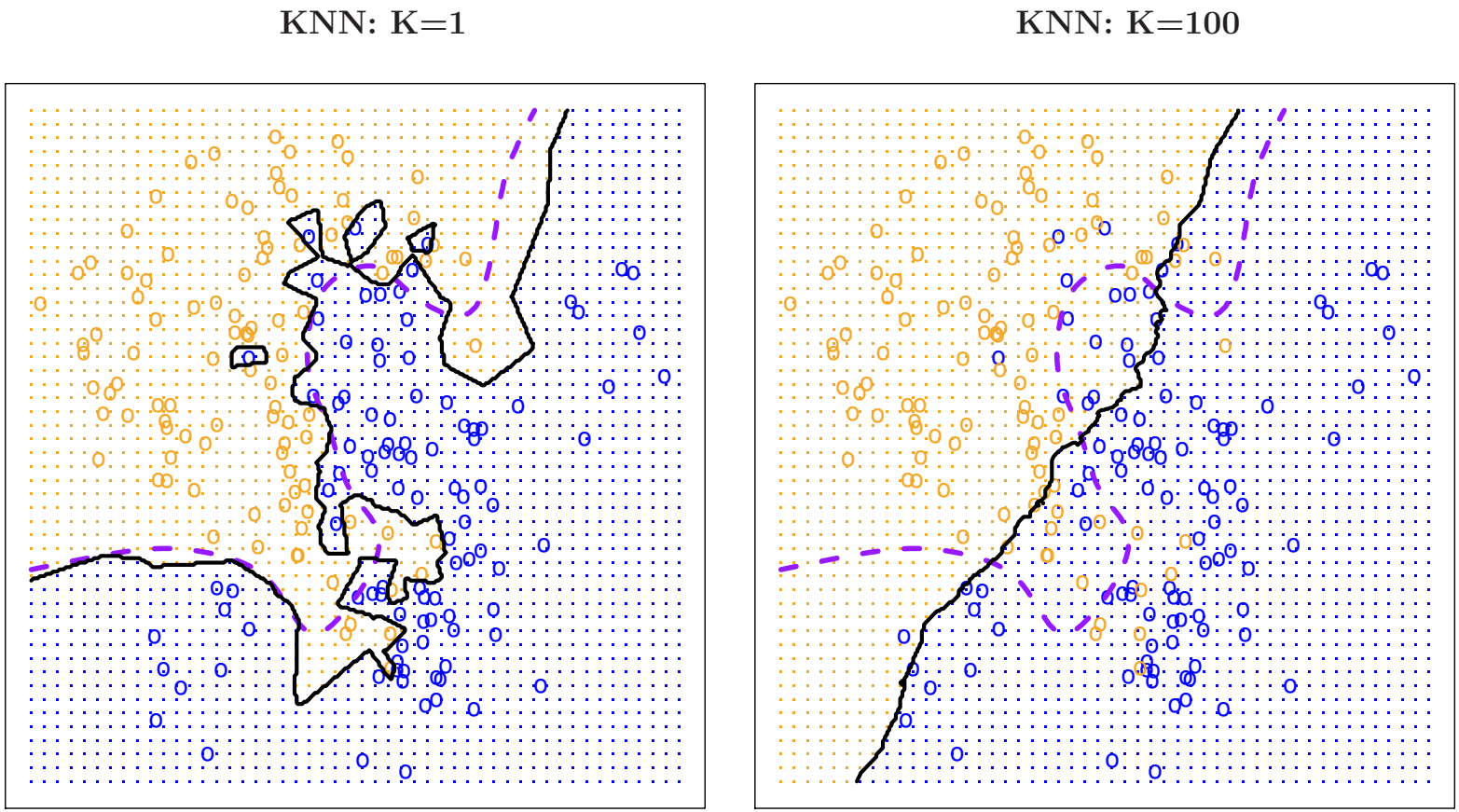


Figure from ISLR 2013

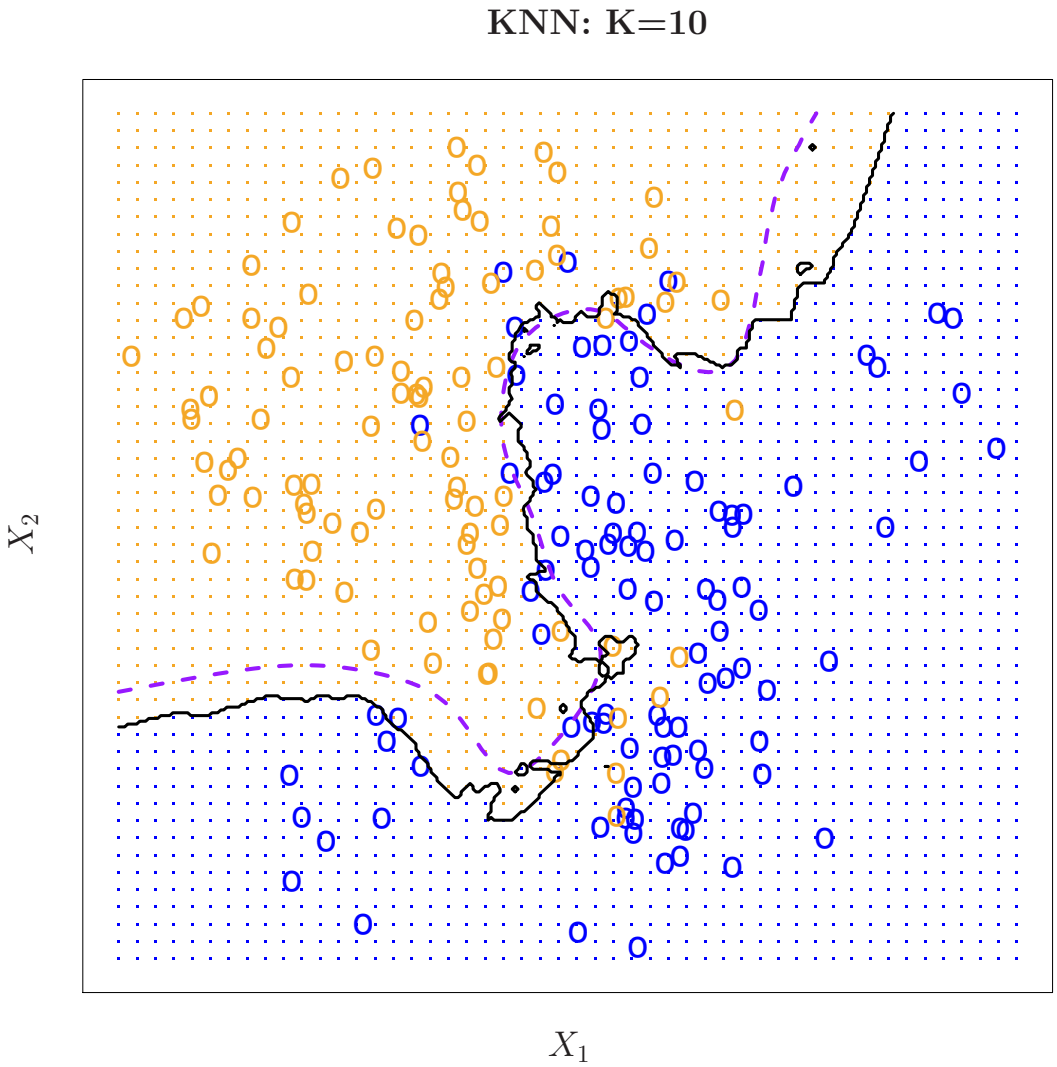
KNN with $k = 1$ and $k = 100$



Dash line is the class boudary from the Bayes classifier.
$k = 1$ overfits (too complex) and $k = 100$ underfits (too simple).

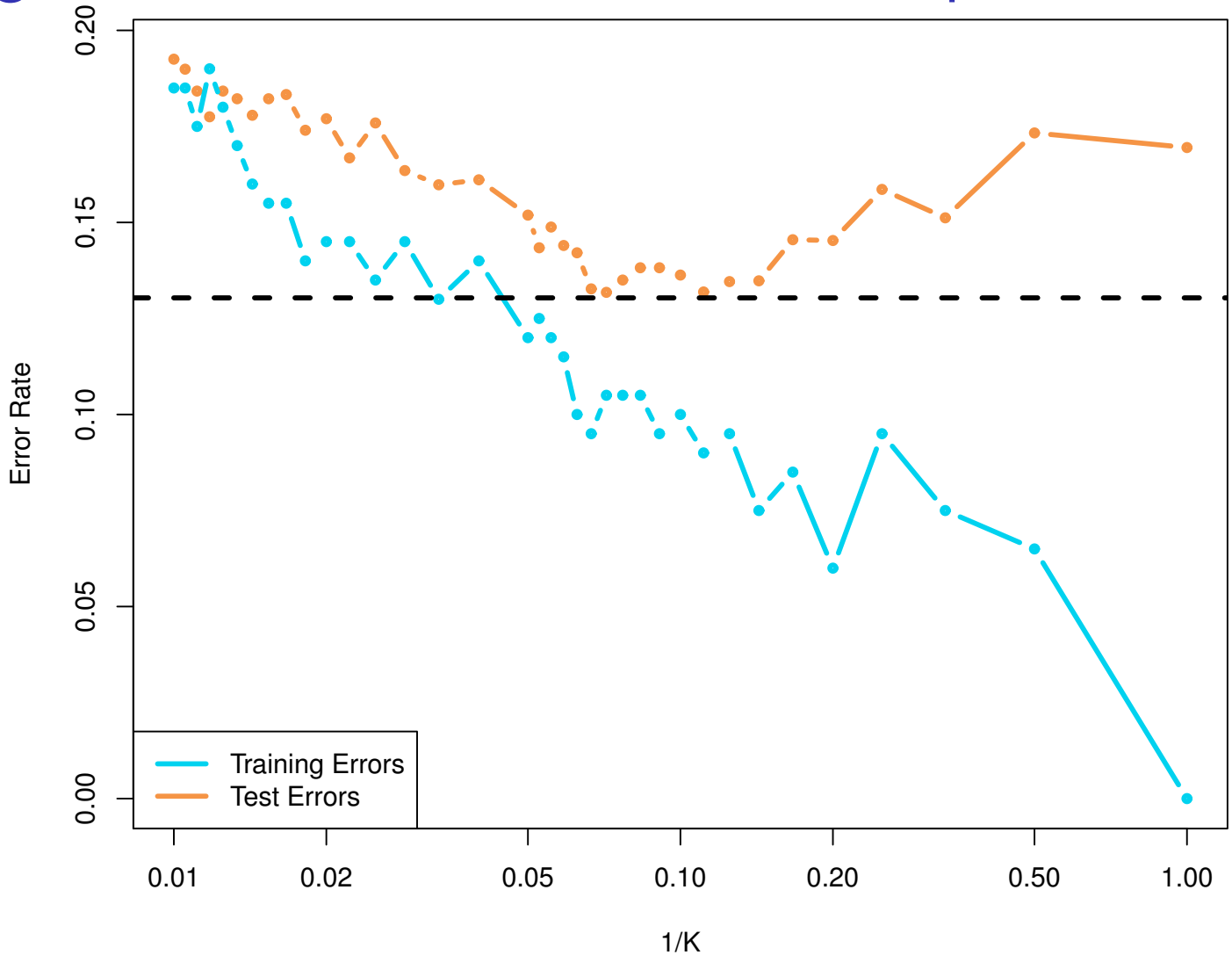
Figure from ISLR 2013

A good choice of k (Figure from ISLR 2013)



The class boundary for the knn with $k = 10$ is very similar to the one from Bayes classifier.

Training vs. test error rates in knn example



Training error rates keep going down as k decreases.

Test error rate at first decreases but then starts to increase.

Figure from ISLR 2013

A fundamental picture

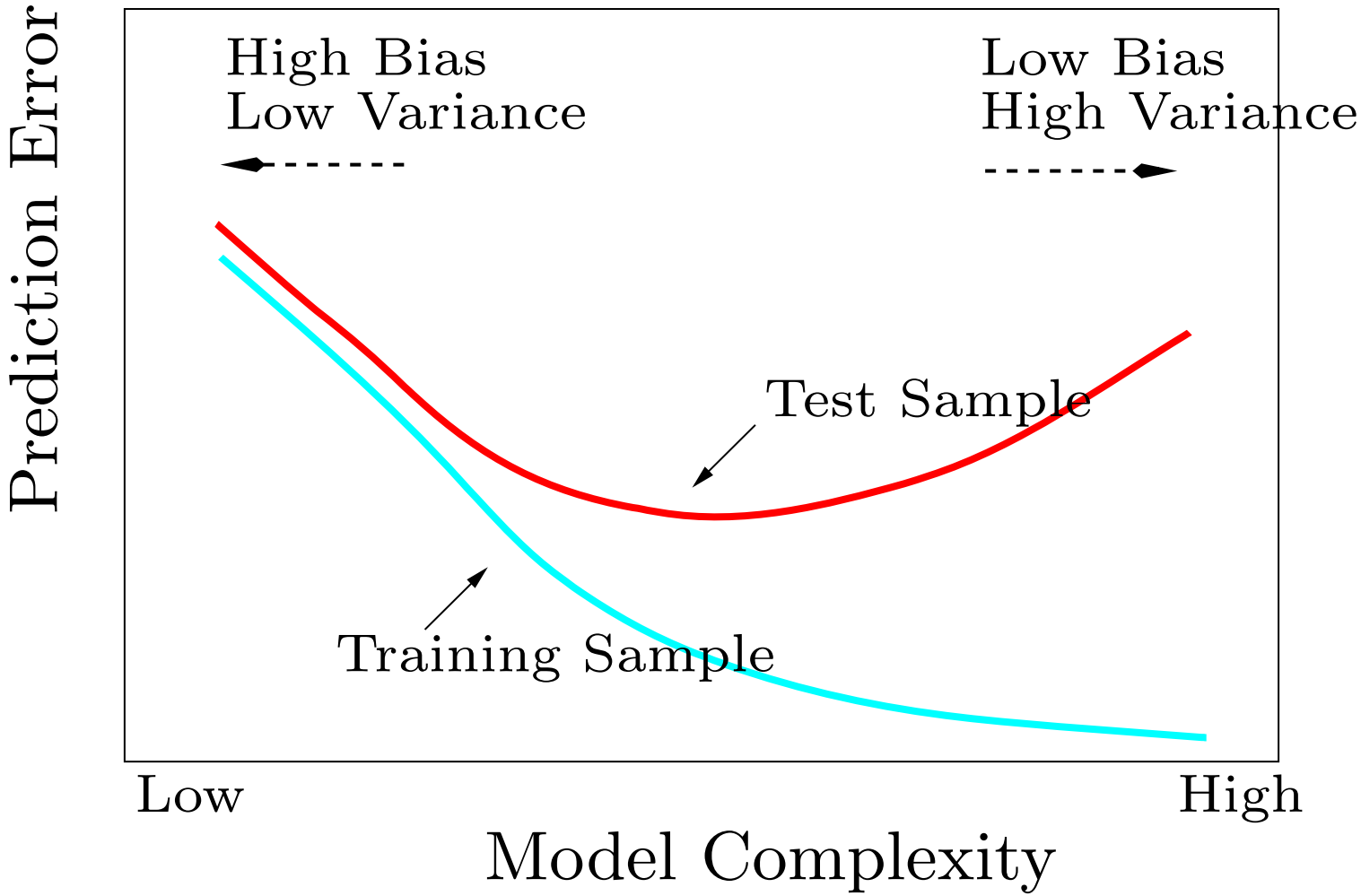


Figure from EOSL 2001.

A cautionary note

- George Box, a famous statistician and son-in-law of R.A. Fisher, once said:
  "All models are wrong, but some are useful."
- In practice, there is really NO *true* model but a *good* model.
- A good model should achieve at least one of the following:
  - an interpretable model that can be explained by some known facts or knowledge;
  - reveals some unknown truth or relationship among the variables or observations;
  - a model with accurate prediction on new samples.
- The `optimal` model depends on:
  - the purpose of the study;
  - the complexity of the underlying mechanism;
  - the quality of the data and signal-noise-ratio;
  - the sample size.

Simulation study I

- Data: 500 samples with 25 input variables and 1 numeric response Y.
  - Data generating mechanism: $y_i = \sum_{j=1}^{15} x_i + \epsilon_i$ where $\epsilon_i \sim N(0, 3^2)$.
  - Input variables: $X = (X_1, \ldots, X_{25}) \sim \text{MVN}(\mathbf{0}, \mathbf{\Sigma})$ where $\rho(X_i, X_j) = 0.5 \forall i \neq j$ and 1 o/w.

```
> library(MASS) #mvrnorm is in MASS library
> mu <- rep(0,25)
> Sigma <- matrix(0.5,25,25)+diag(.5,25)
> n <- 500
> set.seed(1)
> x <- mvrnorm(n,mu,Sigma)
> y <- as.vector(x%*%c(rep(1,15),rep(0,10)))+rnorm(n,sd=3)
> data1 <- data.frame(x,y)[1:50,]; data2 <- data.frame(x,y)
```

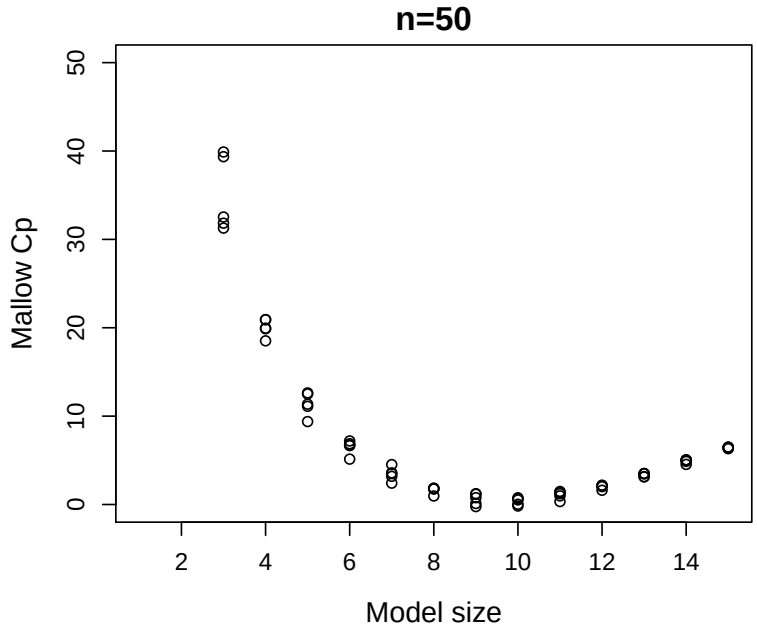
- The best subset selection is applied here using `regsubsets` function in `library(leaps)` in `R`.
- Two groups of models are generated using the first 50 obs (`data1`). and full data ($n = 500$, `data2`)

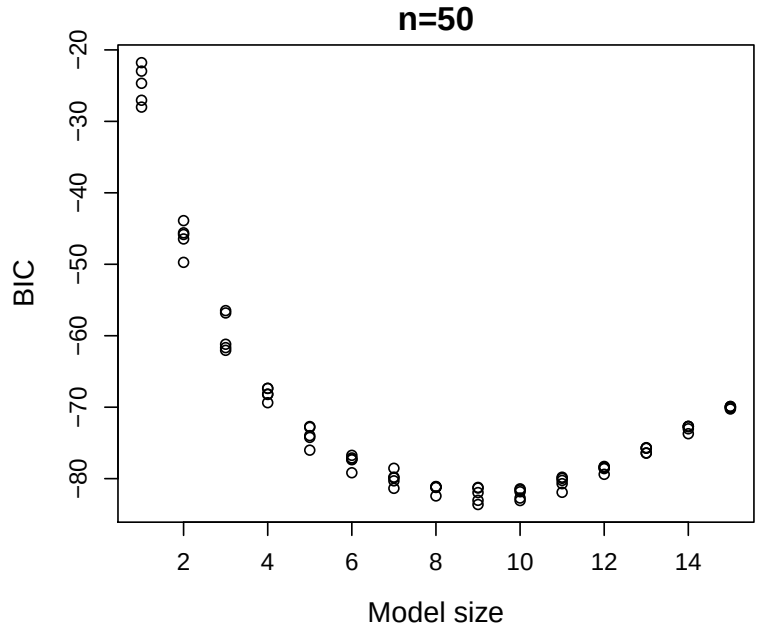
Simulation study I (cont.)

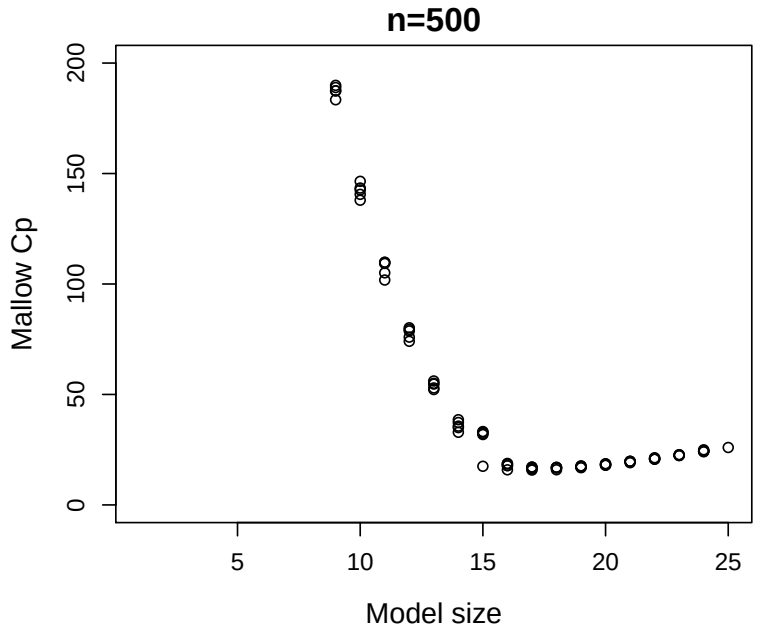
- `nvmax`: the maximum size of subsets to examine.
- `nbest`: the number of subsets of each size to record.
- There are some other useful option. For details, type `?regsubsets` in R.

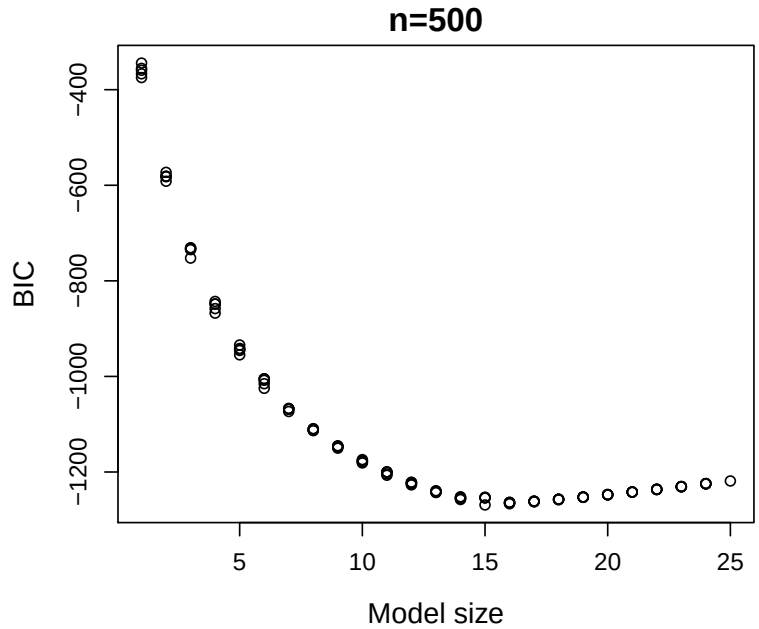
```
> library(leaps)
> sout1 <- summary(regsubsets(y ~ ., data = data1, nvmax = 15,nbest=5))
> res1 <- cbind(apply(sout1$which[,-1],1,sum),Cp=sout1$cp,bic=sout1$bic)
> sout2 <- summary(regsubsets(y ~ ., data = data2, nvmax = 25,nbest=5))
> res2 <- cbind(apply(sout2$which[,-1],1,sum),Cp=sout2$cp,bic=sout2$bic)
> par(mfrow=c(2,2))
> plot(res1[,1],res1[,2],xlim=c(1,15),ylim=c(0,50),
+      xlab="Model size",ylab="Mallow Cp")
> plot(res1[,1],res1[,3],xlim=c(1,15),ylim=range(res1[,3]),
+      xlab="Model size",ylab="BIC")
> plot(res2[,1],res2[,2],xlim=c(1,25),ylim=c(0,200),
+      xlab="Model size",ylab="Mallow Cp")
> plot(res2[,1],res2[,3],xlim=c(1,25),ylim=range(res2[,3]),
+      xlab="Model size",ylab="BIC")
```

Sample size effect



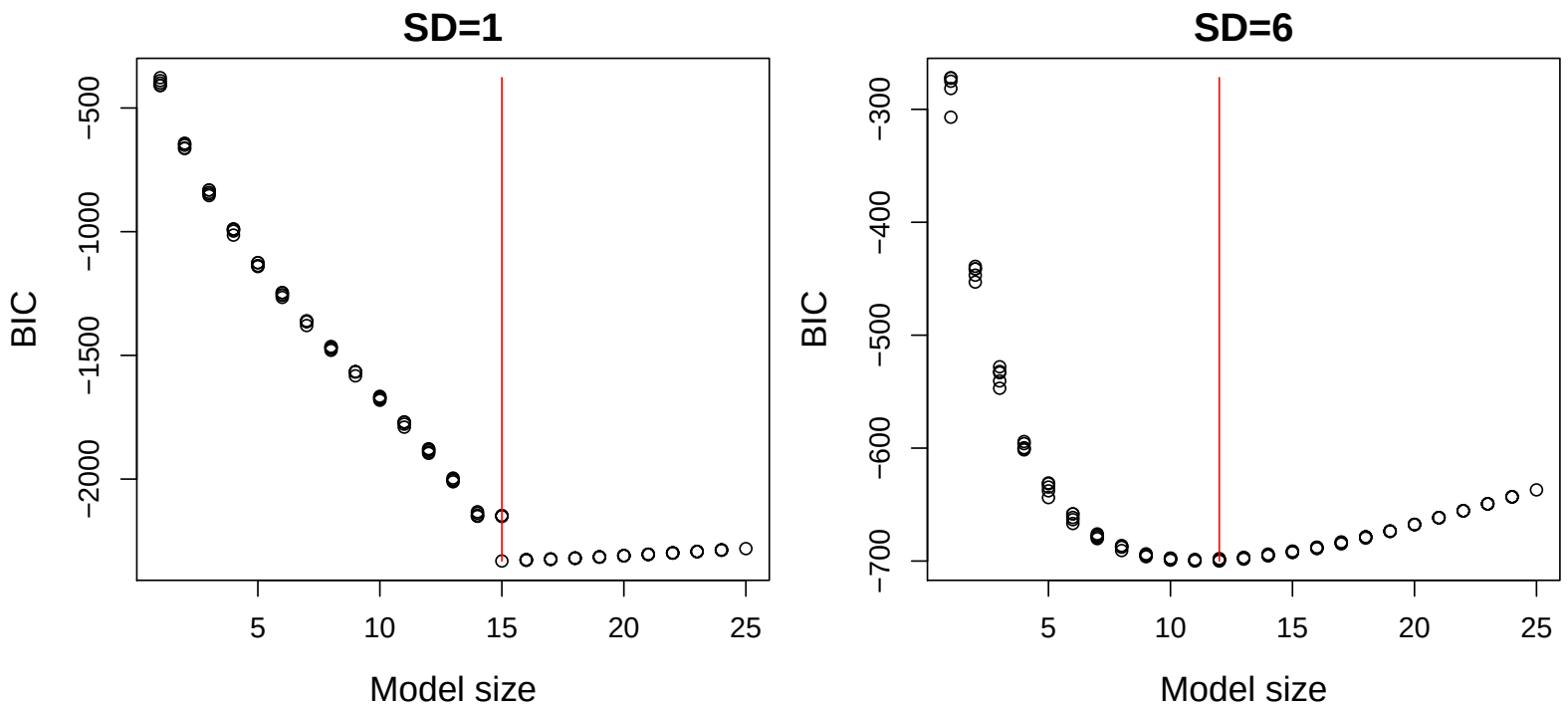






Noise effect

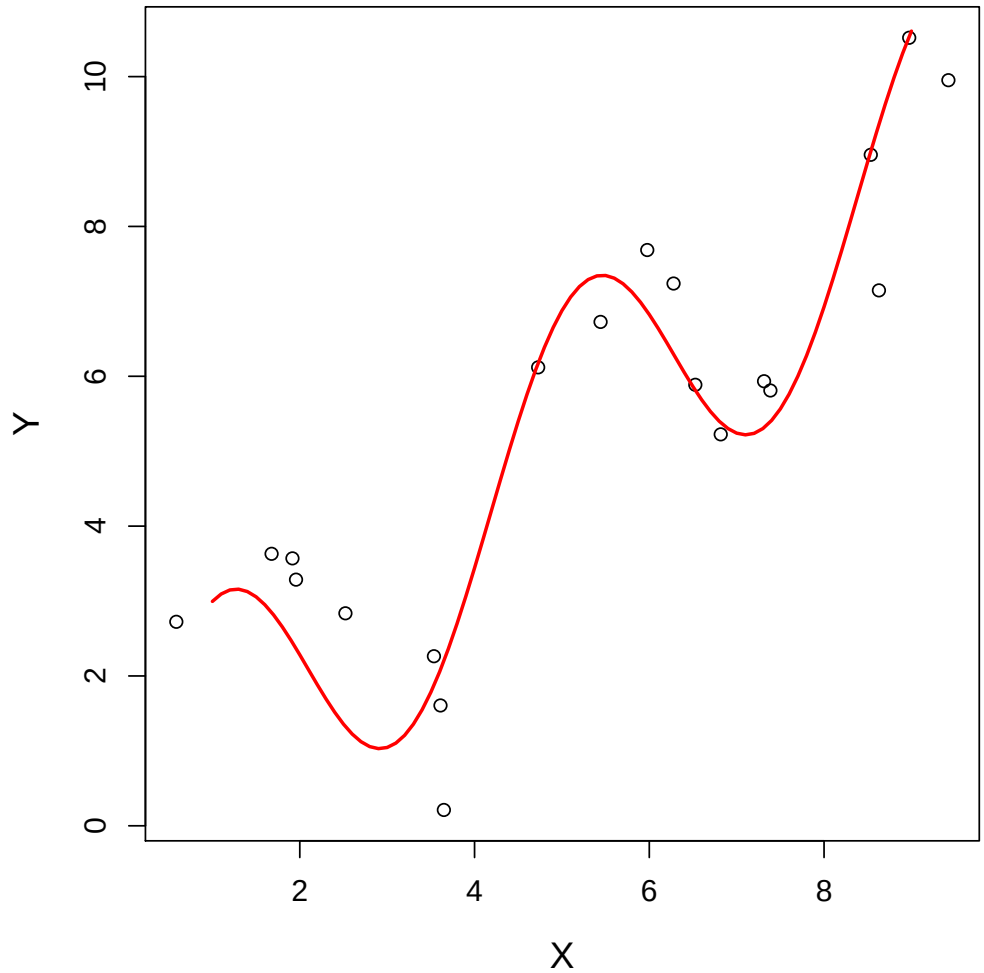
- We set two levels of standard deviation on ϵ_i: 1 and 6 with SNR=122 and 3.4, respectively.
- We use the BIC (common criterion to select models) to select the optimal model size (highlighted by red vertical line).
- Others are kept the same as previous ($n = 500$).



Simulation study II: bias-variance tradeoff

- Data:
  $y_i = 2sin(1.5x_i) + x_i + \epsilon_i$,
  where $\epsilon_i \sim N(0, 1)$
- Training set: `dat` has 20 observation.
- Fit the data using polynomial regressions.
- `dat2` has X values on a fine grid and the true function values without noise.

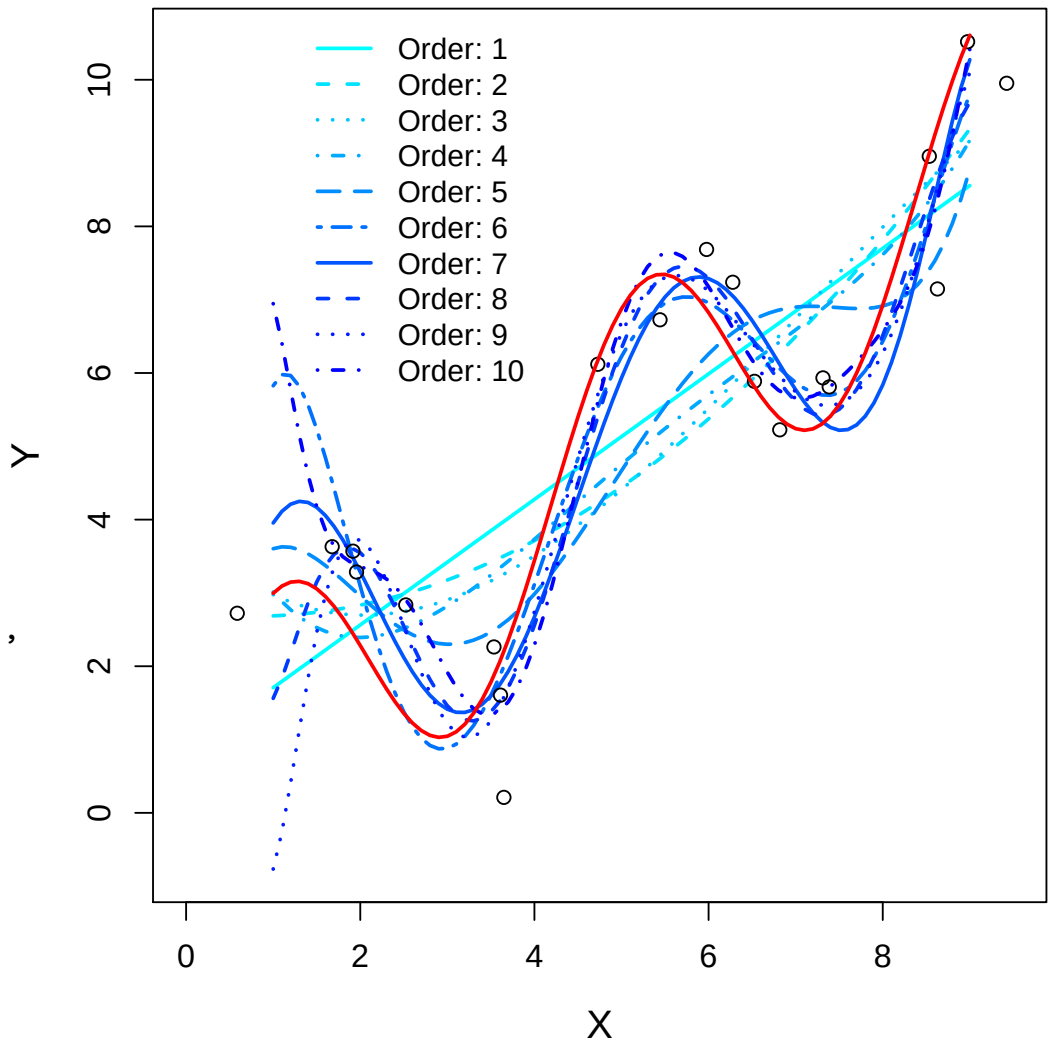
```
> n <- 20
> set.seed(1)
> dat<-data.frame(x = runif(n, 0, 9.5))
> dat$y<-with(dat,2*sin(1.5*x)+x+rnorm(n,sd=1))
> dat2 <-data.frame(x=seq(from=1,to=9,le=81))
> dat2$y<-with(dat2,2*sin(1.5*x)+x)
> plot(dat$x,dat$y, xlab="X", ylab="Y")
> lines(dat2$x,dat2$y,col="red",lwd=2)
```



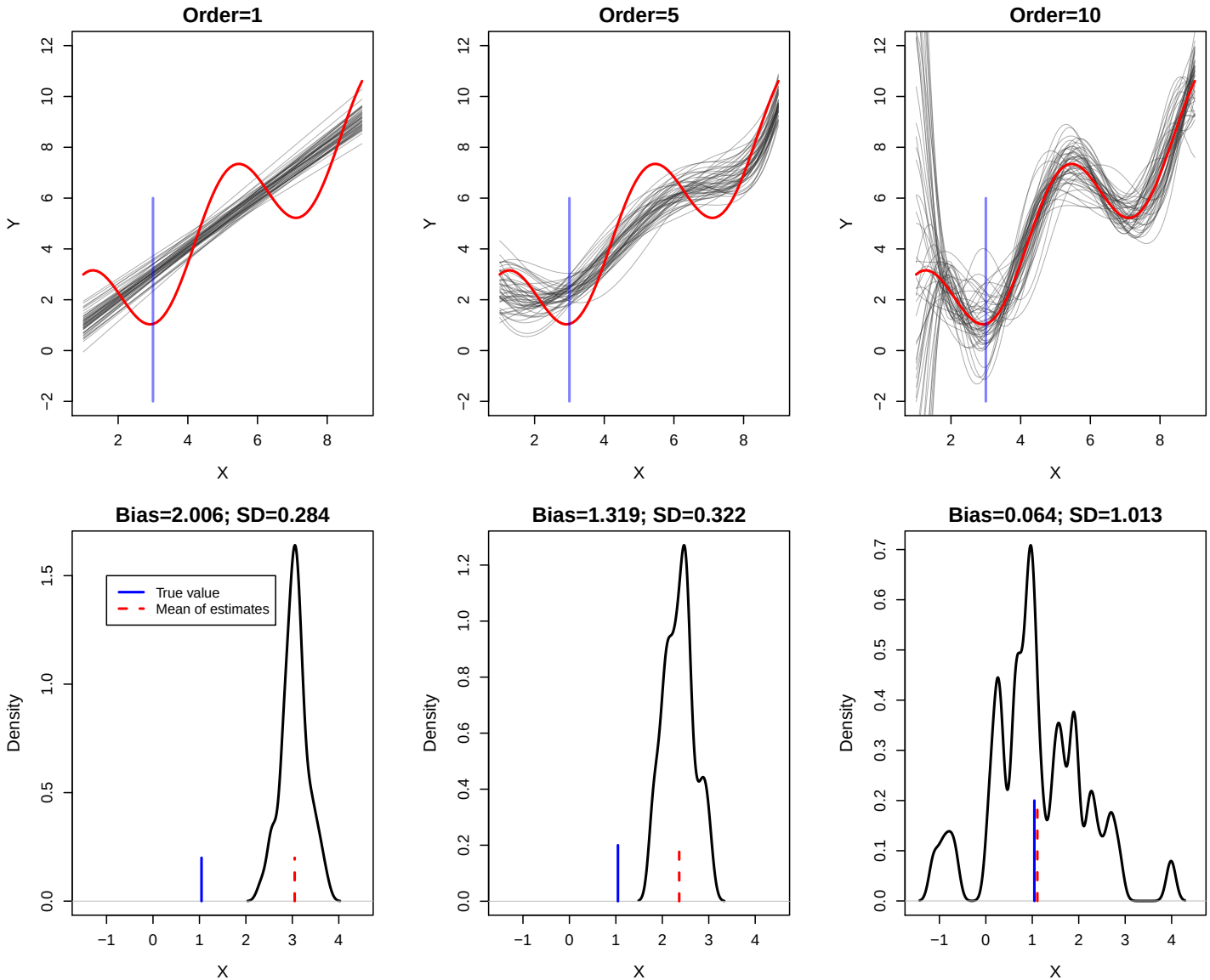
Fitting on various orders of polynomial regressions

- Fit the data using polynomial regressions from order 1 to 10.
- Predict on a fine grid of X in dat2.

```
> pred <- matrix(0,length(dat2$x),10)
> for (i in 1:10){
+  poly.fit <- lm(y~poly(x,i,raw=T),dat)
+  pred[,i]<-predict(poly.fit,dat2)
+ }
> matplot(dat2$x, pred, xlab="X", ylab="Y",
+  xlim=c(0,9.5), ylim=range(c(dat$y,pred)),
+  lty=1:10,lwd=2,type="l",
+  col=rainbow(10, start=3/6, end=4/6))
> points(dat$x, dat$y)
> lines(dat2$x, dat2$y, col="red", lwd=2)
```



Repeat 50 times on randomly generately Y



Remarks on previous figure

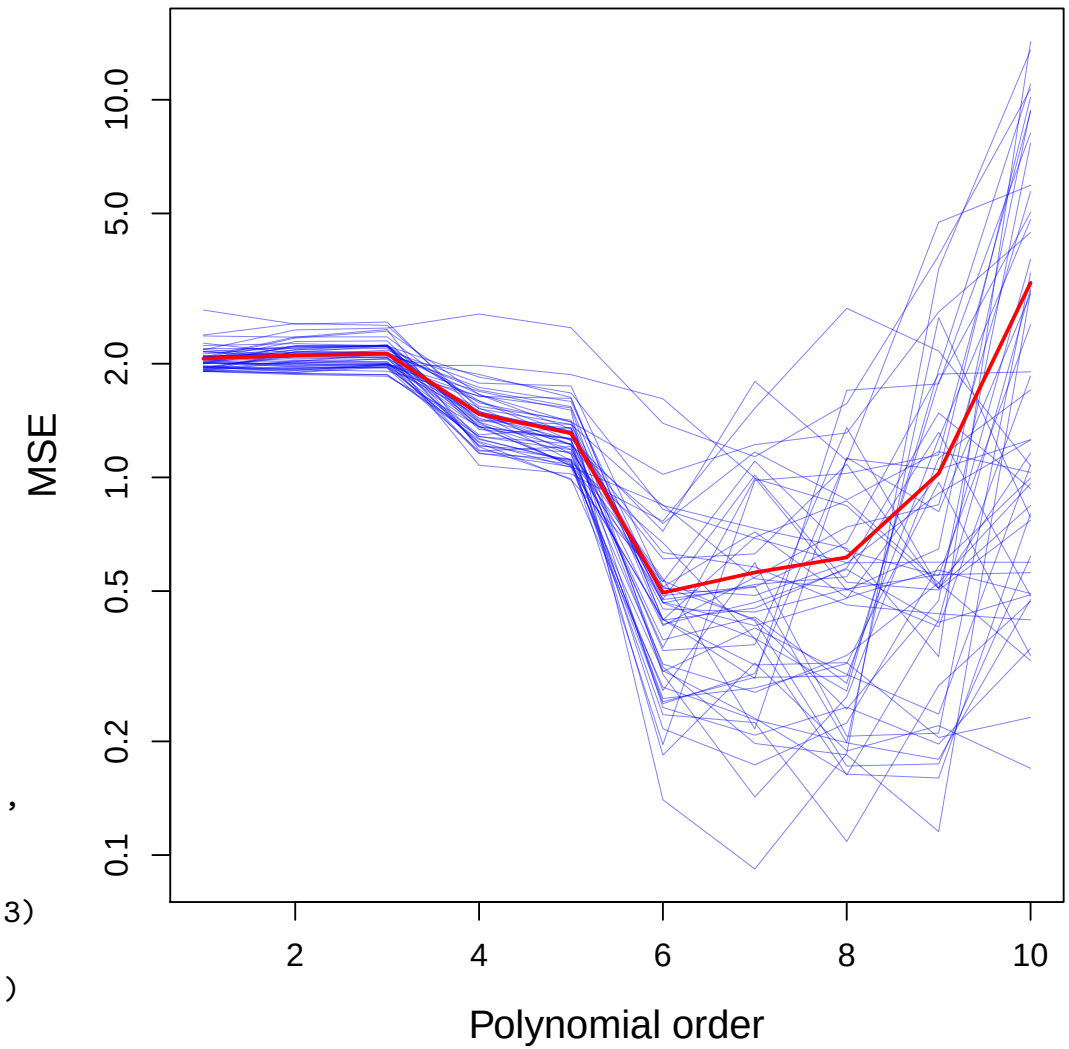
- Variance: how much $\hat{y}$ varies from one training set $\mathcal{D}$ to another.
- Bias: the difference between the true value at $X = x^*$ and expected value of $\hat{y}|X = x^*$ (average of datasets).
- Model too "simple" $\Rightarrow$ does not fit data well (a biased solution).
- Model too "complex" $\Rightarrow$ small change of data makes a big change on $\hat{y}$ (a high variance solution).

```
> iter <- 50
> pred <- list()
> for (it in 1:iter){
+   set.seed(it)
+   dat$y <- 2*sin(1.5*dat$x)+dat$x+rnorm(n,sd=1)
+   pred[[it]] <- matrix(0,length(dat2$x),10)
+   for (i in 1:10){
+     pred[[it]][,i] <- predict(lm(y~poly(x,i,raw=T),dat),dat2)
+   }
+ }
> par(mfcol=c(2,3))
> plot(dat2$x,pred[[1]][,1],xlab="X",ylab="Y",type="n")
> for (i in 1:iter){
+   lines(dat2$x,pred[[i]][,1])
+ }
> lines(dat2$x,dat2$y,col="red",lwd=2)
> segments(3,-2,3,6,lwd=2,col=rgb(0,0,1,alpha=0.5))
> title("Order=1")
> plot(density(pred.2[,1],bw=0.1),main="",xlab="X")
> lines(rep(dat2$y[ind],2),c(0,0.2),col="blue")
> lines(rep(mean(pred.2[,1]),2),c(0,0.2),col="red",lty=2)
```

MSE curves among 50 repetitions

- Curves on the background are the MSE for each sample againt polynomial order.
- Solid red line is the average MSE among 50 samples.
- Left: low variance but high bias $\Rightarrow$ Riight: high variance low bias.
- Optimal order is around 6 (true function has 4 reflection pts).

```
> mse <- matrix(0,iter,10)
> FUN1 <- function(x) mean((x-dat2$y)^2)
> for (it in 1:iter){
+   mse[it,] <- apply(pred[[it]],2,FUN1)
+ }
> plot(1:10,mse[1,],log="y",ylab="MSE",
+   xlab="Polynomial order",xlim=c(1,10),
+   ylim=range(mse),type="n")
> for (it in 1:iter){
+   lines(1:10,mse[it,]col="blue",lwd=0.3)
+ }
> lines(1:10,apply(mse,2,mean),col="red")
```



Bias-variance tradeoff in MSE

- Since we know the true function, here MSE $=$ bias^2 $+$ Variance.
- Bias is estimated by using the average over 50 replications as $E(\hat{f})$.
- Variance is estimated by using the variance of $\hat{f}$ over 50 replications.

```
> bias2 <- vari <- rep(0,10)
> for (i in 1:10){
+   tmp1 <- matrix(0,length(dat2$x),iter)
+   for (it in 1:iter){
+     tmp1[,it] <- pred[[it]][,i]
+   }
+   tmp2 <- apply(tmp1,1,mean)
> #bias2[i]: mean bias^2 for ith order
+   bias2[i] <- mean((dat2$y-tmp2)^2)
> #tmp3: variance of est. on grid for ith order
+   tmp3 <- apply(tmp1,1,var)
+   vari[i] <- mean(tmp3)
+ }
> plot(1:10,apply(mse,2,mean),xlab="Polynomial order",
+      ylab="",col="blue",ylim=range(c(bias2,vari)),
+      type="l",lwd=2)
> lines(1:10,bias2,col="red",lwd=2,lty=2)
> lines(1:10,vari,col="orange",lwd=2,lty=4)
```