Curse of dimensionality (Chapter 4, problem 4)

- When the number of features *p* is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that non-parametric approaches often perform poorly when *p* is large. We will now investigate this curse.

4(a)

- Suppose that we have a set of observations, each with measurements on p = 1 feature, X. We assume that X is uniformly (evenly) distributed on [0, 1]. Associated with each observation is a response value. Suppose that we wish to predict a test observation's response using only observations that are within 10% of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with X = 0.6, we will use observations in the range [0.55, 0.65]. On average, what fraction of the available observations will we use to make the prediction?

- On the interval (a,b) the uniform density function is $\frac{1}{b-a}$.

  Since b=1, and a=0, b-a=1 and

$$\int_{0.55}^{0.65} dx = x|_{0.55}^{0.65} = 0.65 - 0.55 = 0.1$$

4(b)

- Now suppose that we have a set of observations, each with measurements on $p = 2$ features, X_1 and X_2. We assume that (X_1,X_2) are uniformly distributed on $[0, 1] \times [0, 1]$. We wish to predict a test observation's response using only observations that are within 10% of the range of X_1 and within 10% of the range of X_2 closest to that test observation. For instance, in order to predict the response for a test observation with $X_1 = 0.6$ and $X_2 = 0.35$, we will use observations in the range $[0.55, 0.65]$ for X_1 and in the range $[0.3, 0.4]$ for X_2. On average, what fraction of the available observations will we use to make the prediction?

4(b) technical aside

- A single continuous random variable, x_1, has a density function, $f_1(x_1)$, and a distribution function,$F_1(x_1) = Prob(X_1 \leq x_1) = \int_{-\infty}^{x_1} f_1(u_1)du_1$
- There exist similar functions for multivariate random variables, X_1,.., X_p , e.g. $f(x_1, \ldots, x_p)$.
- In the case where X_1,.., X_p are independent we can conclude $f(x_1, \ldots, x_p) = f(x_1)f(x_2) \cdots f(x_p)$
- $\int_{0.55}^{0.65} dX_1 \int_{0.3}^{0.4} dX_2 = (0.1) \times (0.1) = 0.01$

4(c)

- Now suppose that we have a set of observations on *p* = 100 features. Again, the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation’s response using observations within the 10% of each feature’s range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?
- $\int_{0.45}^{0.55} dX_1 \dots \int_{0.45}^{0.55} dX_{100} = (0.1)^{100} = 10^{-100}$

4(d)

- Using your answers to parts (a)–(c), argue that a drawback of KNN when *p* is large is that there are very few training observations "near" any given test observation.
- It's impossible to get any observation within the prescribed distance.

4(e)

- Now suppose that we wish to make a prediction for a test observation by creating a *p*-dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For *p* = 1, 2, and 100, what is the length of each side of the hypercube? Comment on your answer.
- Find c_1 and c_2 such that $\int_{c_1}^{c_2} dX = 0.1$

  Let the length of side be c_2-c_1=δ. Then when *p*=1, δ =0.1
  when *p*=2, δ^2=0.1 or δ=0.31
  when *p*=100, δ^{100}=0.1 or δ=0.977
- At large *p* the hypercube sides include the vast range of each variable, so you can't get very good predictions.

Chapter 5: Resampling Methods

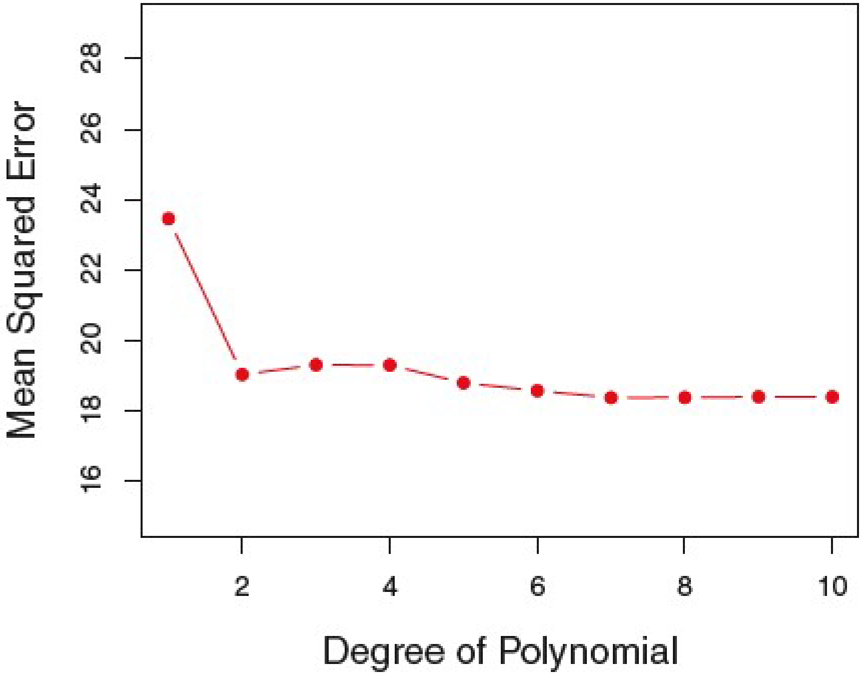
- **Cross-validation**. A method that will provide test error measurements so we may evaluate it’s performance or *assessment*.
- For techniques that may vary in flexibility, *model selection* can be made with the help of cross-validation.
- **Bootstrap**, is a computer based resampling method for making estimate of parameter bias, variance and confidence intervals.

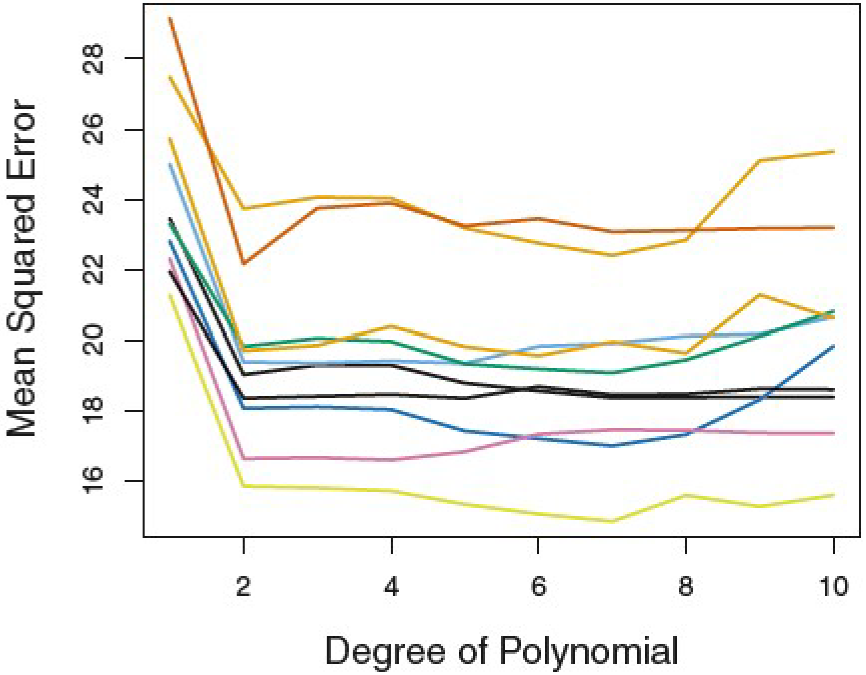
Validation Set

- We divide our observations at random into a training set, and a validation or hold-out set.
- The smaller the training set the less precise our parameter estimates will be
- The training set is used to estimate model parameters and the validation set is used as an independent means of estimating error usually via the MSE.

Validation Set

- Use the Auto mileage data and split the data evenly into a training and test set.
- The figure on the left is one split, on the right many independent splits.
- The test MSE vary a lot depending on the data included in the training and test sets.
- The model estimates should improve if we used more than 50% of the observations. Thus, these MSE estimates are probably overestimates of error we would find if trained on the entire set of observations.



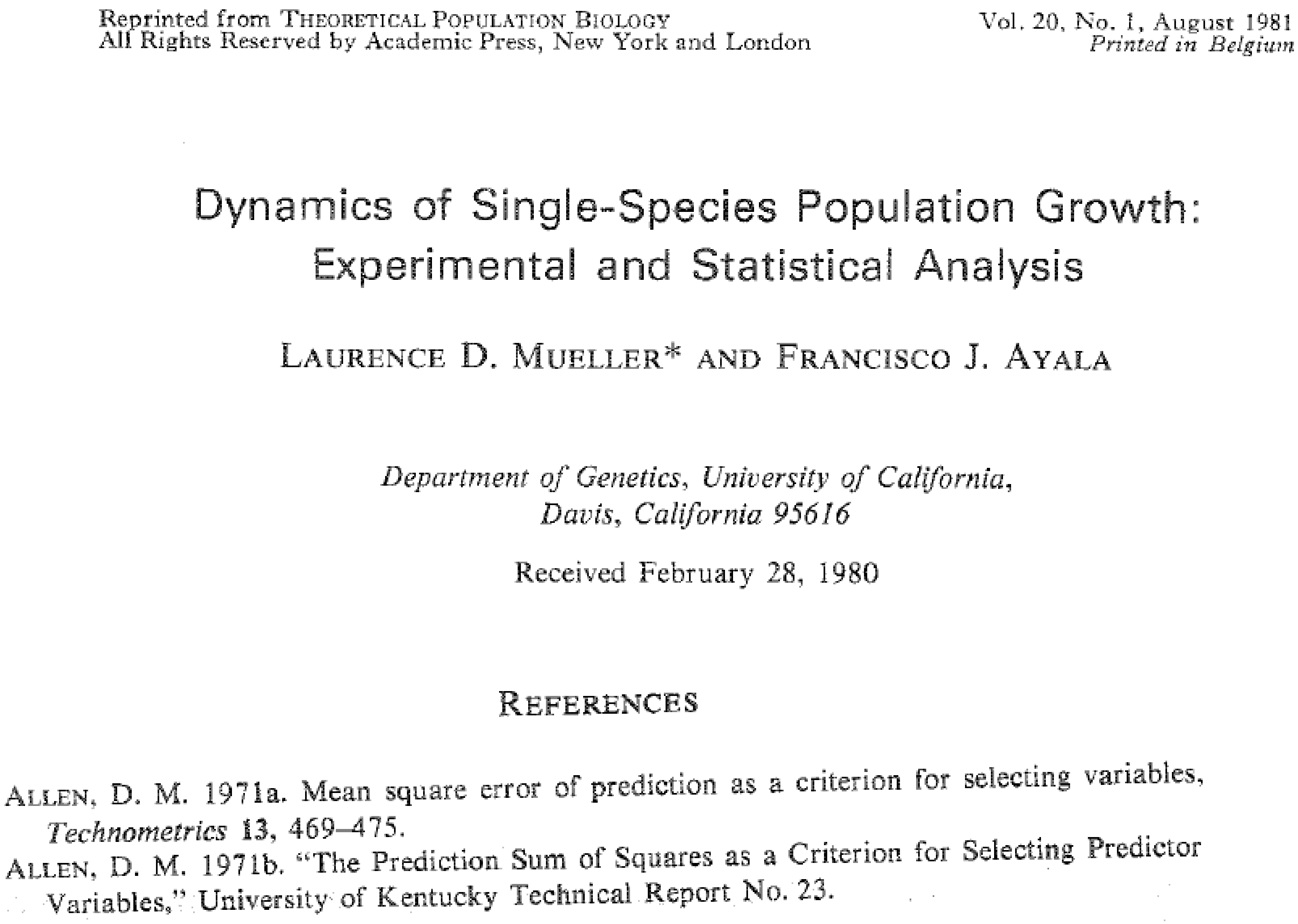


Leave-one-out Cross-validation (LOOCV)

- Suppose we had *n*-pairs of observations, $(x_1,y_1,)\ldots(x_n,y_n)$. Then we could leave out one observation, (x_i,y_i), and train the model with the remaining *n*-1 pairs.
- The single left out pair can then be used to estimate a mean squared error as, $MSE_i = (y_i - \hat{y}_i)^2$
- Repeat this *n*-times and estimate the cross-validation error as (PRESS*) $CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$
- Now the test *mse* is less likely to be severely biased, since *n*-1 is close to *n*, and it is not subject to the vagaries of which observations went into the training and test sets of data.

  * Prediction sum of squares (PRESS) was proposed in 1971 as a method for choosing the best regression equations.

PRESS



Reprinted from THEORETICAL POPULATION BIOLOGY
All Rights Reserved by Academic Press, New York and London

Vol. 20, No. 1, August 1981
Printed in Belgium

Dynamics of Single-Species Population Growth: Experimental and Statistical Analysis

LAURENCE D. MUELLER* AND FRANCISCO J. AYALA

Department of Genetics, University of California,
Davis, California 95616

Received February 28, 1980

REFERENCES

ALLEN, D. M. 1971a. Mean square error of prediction as a criterion for selecting variables, *Technometrics* **13**, 469–475.

ALLEN, D. M. 1971b. "The Prediction Sum of Squares as a Criterion for Selecting Predictor Variables," University of Kentucky Technical Report No. 23.

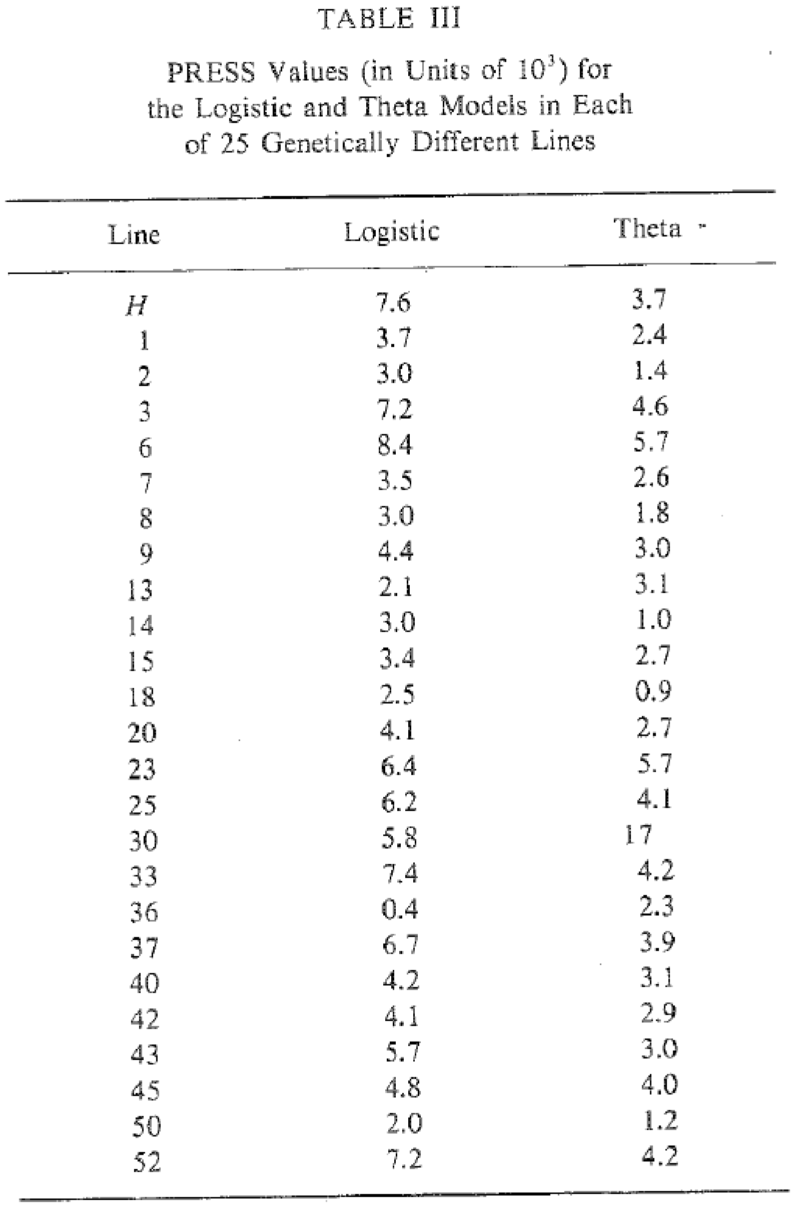


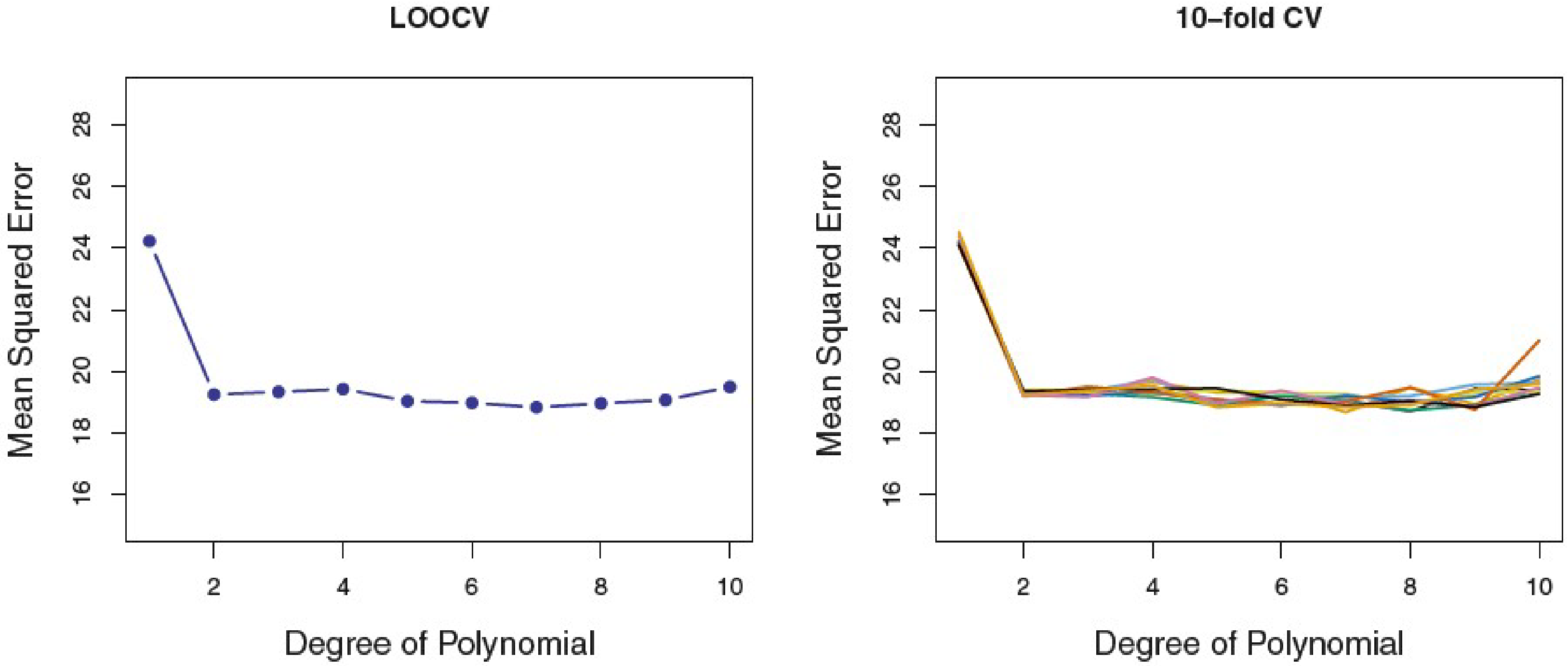
TABLE III

PRESS Values (in Units of 10^3) for the Logistic and Theta Models in Each of 25 Genetically Different Lines

Line	Logistic	Theta
H	7.6	3.7
1	3.7	2.4
2	3.0	1.4
3	7.2	4.6
6	8.4	5.7
7	3.5	2.6
8	3.0	1.8
9	4.4	3.0
13	2.1	3.1
14	3.0	1.0
15	3.4	2.7
18	2.5	0.9
20	4.1	2.7
23	6.4	5.7
25	6.2	4.1
30	5.8	17
33	7.4	4.2
36	0.4	2.3
37	6.7	3.9
40	4.2	3.1
42	4.1	2.9
43	5.7	3.0
45	4.8	4.0
50	2.0	1.2
52	7.2	4.2

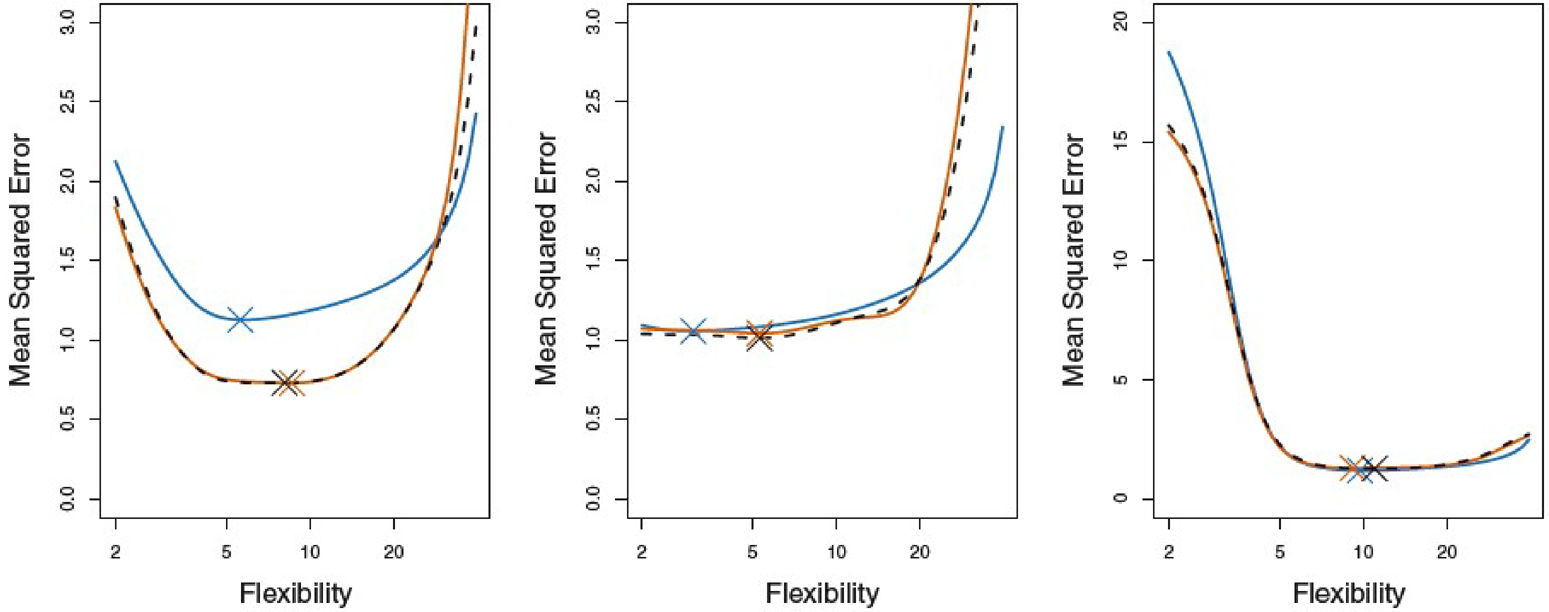
k-Fold Cross-Validation

- Now divide the observations into k groups or folds (k is often 5 or 10). Use one fold as the test set and the remaining k-1 folds as the training set. Repeat this k times. Then take the n/k observations in the test set and estimate MSE_i as $\frac{k}{n}\sum_{j=1}^{n/k}\left(y_j - \hat{y}_j\right)^2$
- Now the cross-validation MSE estimate is,
- $CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$



k-Fold Cross-Validation

- Compared to the leave one out method we only do 5 or 10 fits with our method rather than *n*.



From simulations we see both over and underestimates of the true MSE.

Usually, the value of the MSE is of little interest but the location of the minimum is.

FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9* (left)*, 2.10* (center)*, and 2.11* (right)*. The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the 10-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.*

Bias vs. Variance trade-off for *k*-fold cross validation

- The leave one out method should have very little bias since it is using almost the entire data set to estimate parameters. Validation sets (using 50% of the data) will overestimate MSE. The *k*-fold cross validation will have intermediate bias.
- However, the leave one out method should have higher variance.
- Each training set in the leave one out method has highly overlapping sets of observations. Thus, each of these training sets has a higher variance for their mean MSE. Recall that the $\text{Var}(x+y) = \text{Var}(x) + \text{Var}(y)+2\text{Cov}(xy)$. This covariance term should be positive when using almost the same data for different estimates of MSE_i.
- James et al. suggest $k=5$ or 10 have been shown to have the best balance of bias and variance.

Cross-Validation for Classification

- The leave one out cross-validation error rate is, $CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} Err_i$, where $Err_i = I(y_i \neq \hat{y}_i)$.
- In each term $\hat{y}_i$ is computed from a model trained without observation-*i*.
- Logistic regression fits using polynomials of increasing complexity.
- Thought question: how do you do LOOCV on K nearest neighbors?

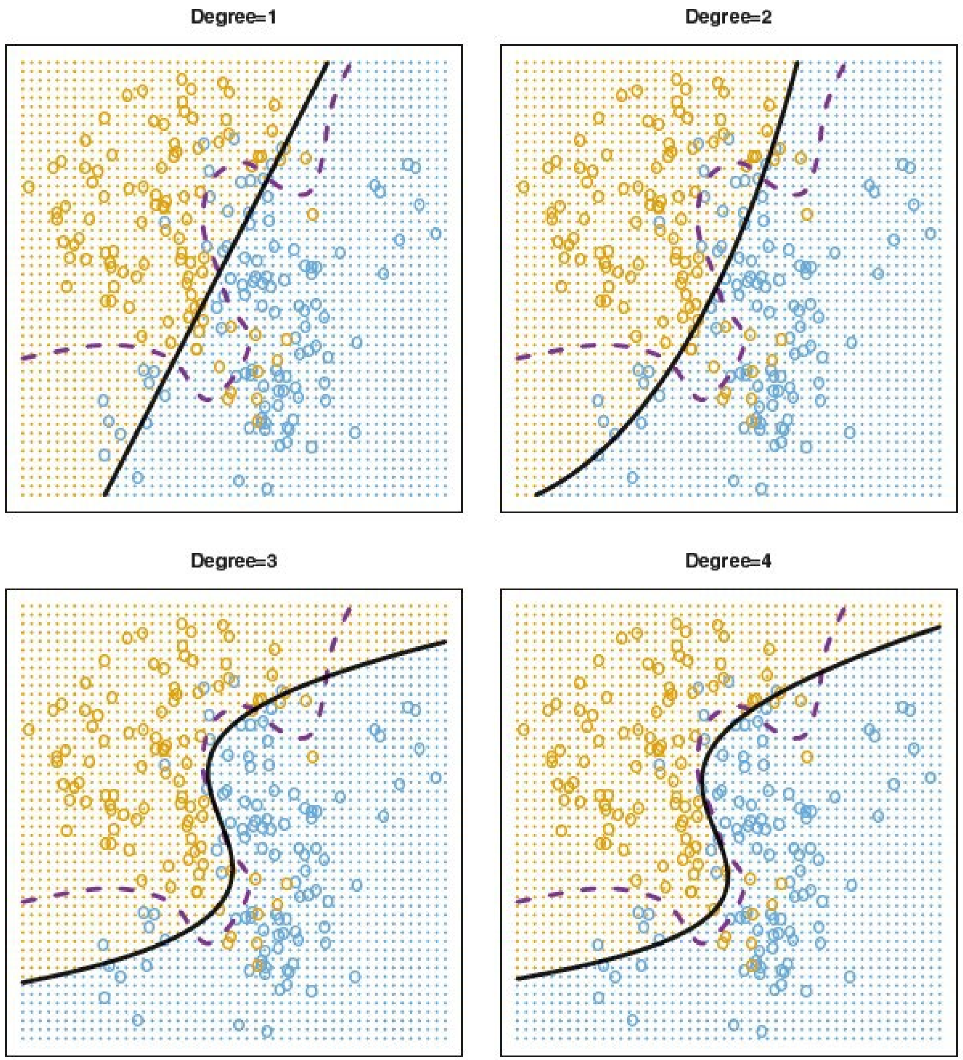
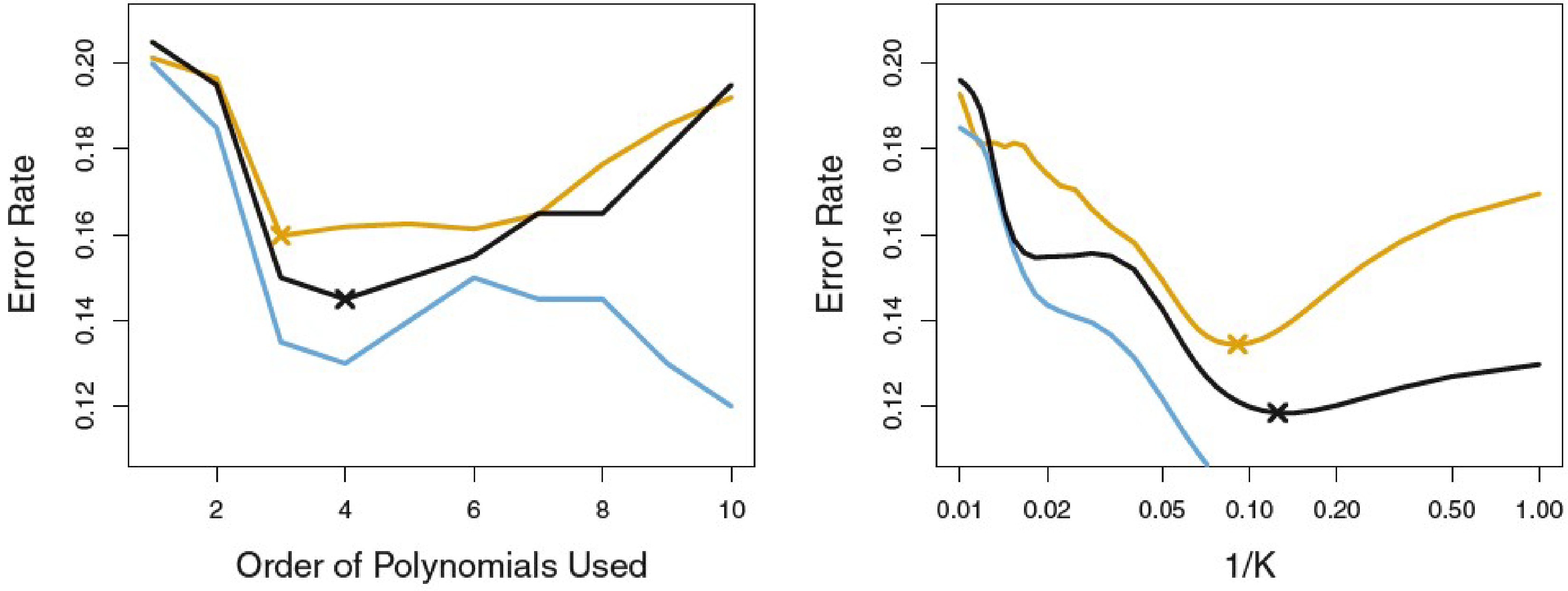


FIGURE 5.7. *Logistic regression fits on the two-dimensional classification data displayed in Figure 2.13. The Bayes decision boundary is represented using a purple dashed line. Estimated decision boundaries from linear, quadratic, cubic and quartic (degrees 1–4) logistic regressions are displayed in black. The test error rates for the four logistic regression fits are respectively* 0.201, 0.197, 0.160, *and* 0.162, *while the Bayes error rate is* 0.133.

Cross-Validation for Classification

- Logistic regression and KNN classification.
- Training error (blue) always goes down with complexity.
- 10-fold CV error (black) tends to underestimate the true error (brown). CV error predicts best model well.



The Bootstrap

- Use the data as an empirical estimate of the underlying distribution.
- Example: invest α of your assets in *X* and 1-α in *Y*. The variance and covariance of these investments are σ_X^2, σ_Y^2 and σ_{XY}.
- The variance of your returns are Var(αX+(1-α)Y)=$\alpha^2 Var(X) + (1-\alpha)^2 Var(Y) + 2\alpha(1-\alpha)Cov(XY)$. Your goal is to find α that will minimize the variation on your returns.
- Take the derivative with respect to α and set to 0, solve for α.
- This yields, $\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$

Bootstrap

To simulate these bivariate random variables

```
mean.a<- c(0,0)
var.a<- rbind(c(1,0.5),c(0.5,1.25))
library(mvtnorm)
random.a<- rmvnorm(1,mean=mean.a,var=var.a)
# For more than 1 random vector replace “1”
# with the desired number.
```

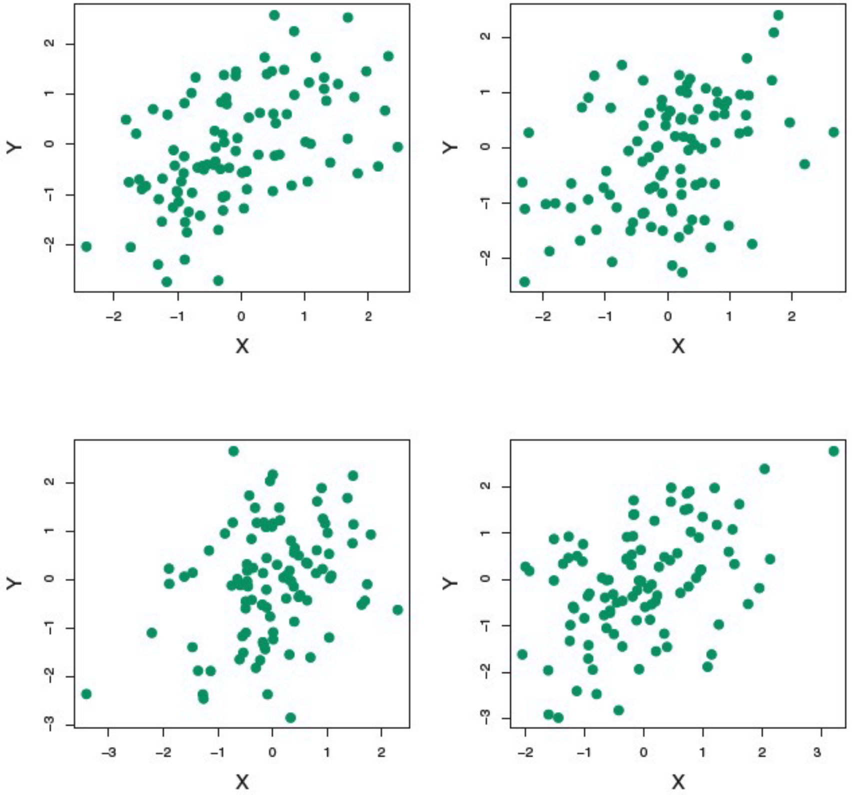


FIGURE 5.9. *Each panel displays* 100 *simulated returns for investments* X *and* Y. *From left to right and top to bottom, the resulting estimates for* α *are* 0.576, 0.532, 0.657, *and* 0.651.

Bootstrap

How to generate bootstrap samples
Let *z* be a 100 x 2 matrix with
100 observations of X and Y

```
z.boot<- z[sample(1:100,size=100,replace=T),]
```

Note some samples may be taken more than once.

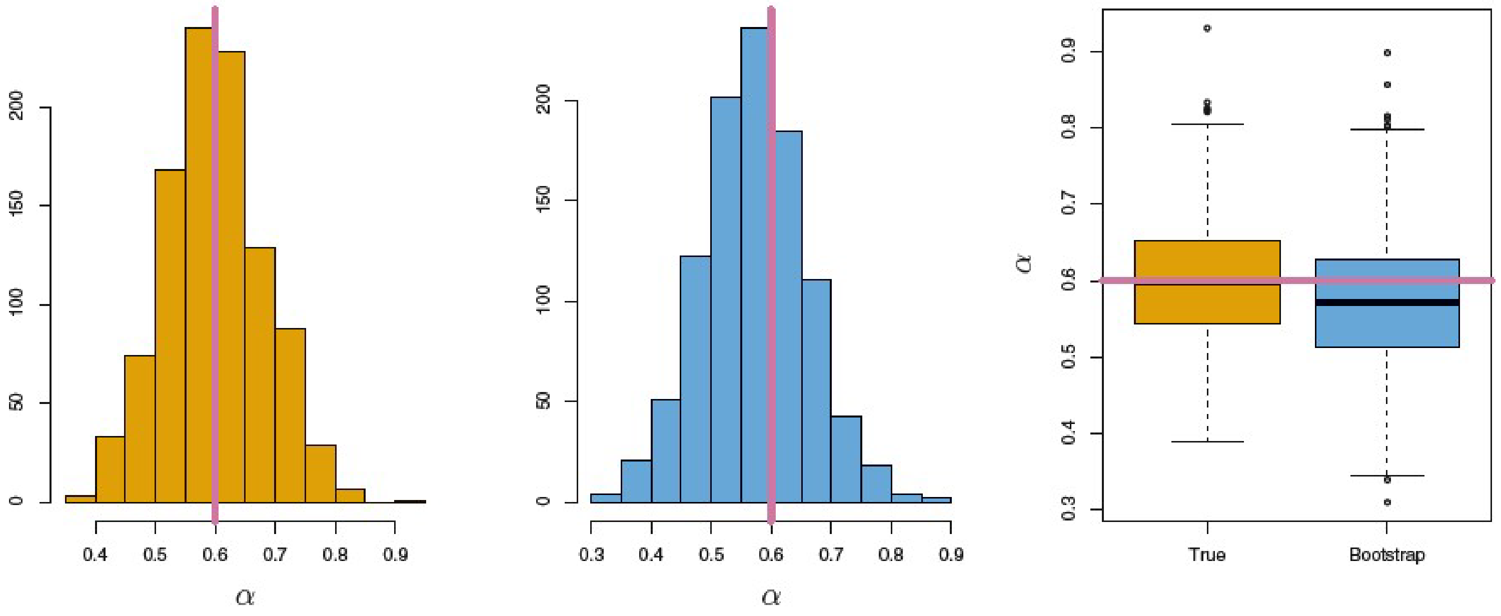


FIGURE 5.10. Left: *A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.*

Bootstrap

- If Z is the original sample of 100 observations. Then we can identify the first bootstrap sample of 100 observation with replacement as, Z^{*1}. The second will be Z^{*2}, up to Z^{*B}.
- Each bootstrap sample provides an estimate of *a*; $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \ldots, \hat{\alpha}^{*B}$
- From these *B* estimates we can now estimate the mean, standard deviation, and even an empirical confidence interval.

  $\bar{\alpha}_B = \frac{1}{B}\sum_{j=1}^{B} \hat{\alpha}^{*j}$, and $SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{j=1}^{B}(\hat{\alpha}^{*j} - \bar{\alpha}_B)^2}$
-

Bootstrap

Example: the investment problem

```
library(ISLR)
library(boot)
alpha.fn=function (data ,index){
  X=data$X [index]
  Y=data$Y [index]
  return ((var(Y)-cov (X,Y))/(var(X)+var(Y) -2* cov(X,Y)))
}
boot.alpha<- boot(Portfolio ,alpha.fn,R=1000)
boot.ci(boot.alpha)
#Output boot.alpha
ORDINARY NONPARAMETRIC BOOTSTRAP
Call:
boot(data = Portfolio, statistic = alpha.fn, R = 1000)
Bootstrap Statistics :
     original      bias    std. error
t1* 0.5758321 0.001672587  0.09039023
# Confidence intervals
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates
CALL :
boot.ci(boot.out = boot.alpha)
Intervals :
Level      Normal              Basic
95%   ( 0.3970,  0.7513 )   ( 0.4049,  0.7467 )
Level     Percentile            BCa
95%   ( 0.4050,  0.7468 )   ( 0.4024,  0.7442 )
Calculations and Intervals on Original Scale
```

Bootstrap

- Since each bootstrap sample is independent of the other, the bootstrap is a perfect method for parallelization.
- Currently the “boot” package isn’t set up to do this on Windows computers.
- Since running parallel processing is easy to set up on Windows and Linux computers, and usually doing the bootstrap sampling is easy to code it may not be worth using “boot” for large problems.
- For more bootstrap details see: Efron, B. and R. J. Tibshirani, 1993. *An Introduction to the Bootstrap*. Chapman & Hall.

Parallel Processing in R

- Two packages need to be installed, “foreach” and “doParallel”

```
library(foreach)
library(doParallel)
registerDoParallel(cores=20) #Here you allocate the
                              number of cores that will
                              be used by foreach

...
a.grid<- foreach(x=1:length(a.list),.errorhandling =
c('remove')) %dopar%{…}
```

- If you have 100 bootstrap samples would there be a difference in time to completion between using 20 cores or 22 cores, since with 22 most cores will still need to do 5 samples? -> YES, if there are random components to the analysis of a sample.

Bootstrap: example niche overlap statistics

- For each species there is vector reflecting the probability that a food item (1 to *n*) is found in the diet of the animal.
- So, for species 1 these are $p_1, \ldots, p_n$ and for species 2, $q_1, \ldots, q_n$.
- Four different indices were examined (1) coefficient of community, (2) Moristia's index, (3) Horn's index, and (4) Euclidian distance.
- For instance, the Euclidian distance statistic is, $1 - \left\{\sum_i {}^{(p_i - q_i)^2}/_{2}\right\}^{1/2}$
- Estimates and confidence intervals were compared for (i) the delta method, (ii) the jackknife, and (iii) the bootstrap.
- Suppose $\hat{S}$ is an estimator of an overlap statistic from a sample size of *n*. Let $\hat{S}_{-i}$ be the same estimate with the *ith* observation deleted. Then *n* pseudovalues may be defined as, $s_i = n\hat{S} - (n-1)\hat{S}_{-i}$
- Then the jackknife mean and variance are $\tilde{S} = \frac{1}{n}\sum_i s_i \;\; and \; Var(\tilde{S}) \frac{1}{n(n-1)} \sum_i \left(s_i - \tilde{S}\right)^2$

Bootstrap: example niche overlap statistics

When the distribution is contaminated then the confidence intervals for the delta method and jackknife perform poorly. The bootstrap on the other hand does well under all conditions.

The bootstrap MSE is slightly larger so if one knows there is no contamination then the other techniques may be better.

Lesson: MSE focusses on properties of the estimated statistic while the confidence interval focusses on the estimated variance and distribution properties of the statistic.

For more details see: Mueller, L.D. and L. Altenberg, 1985. Statistical inference on measures of niche overlap. *Ecology* **66**: 1204-1210.

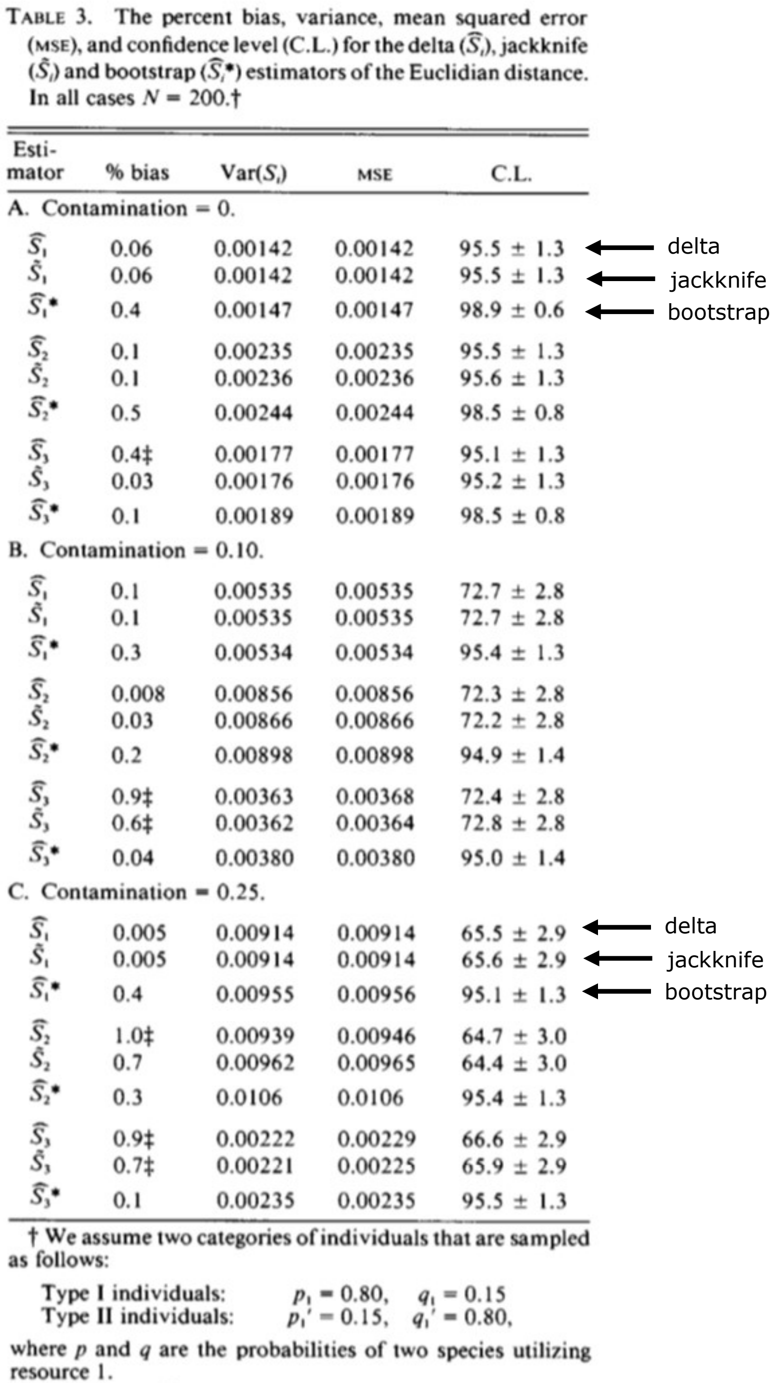


TABLE 3. The percent bias, variance, mean squared error (MSE), and confidence level (C.L.) for the delta ($\widehat{S}_i$), jackknife ($\tilde{S}_i$) and bootstrap ($\widehat{S}_i^*$) estimators of the Euclidian distance. In all cases $N = 200$.†

Estimator	% bias	Var(S_i)	MSE	C.L.
A. Contamination = 0.				
$\widehat{S}_1$	0.06	0.00142	0.00142	95.5 ± 1.3
$\tilde{S}_1$	0.06	0.00142	0.00142	95.5 ± 1.3
$\widehat{S}_1^*$	0.4	0.00147	0.00147	98.9 ± 0.6
$\widehat{S}_2$	0.1	0.00235	0.00235	95.5 ± 1.3
$\tilde{S}_2$	0.1	0.00236	0.00236	95.6 ± 1.3
$\widehat{S}_2^*$	0.5	0.00244	0.00244	98.5 ± 0.8
$\widehat{S}_3$	0.4‡	0.00177	0.00177	95.1 ± 1.3
$\tilde{S}_3$	0.03	0.00176	0.00176	95.2 ± 1.3
$\widehat{S}_3^*$	0.1	0.00189	0.00189	98.5 ± 0.8
B. Contamination = 0.10.				
$\widehat{S}_1$	0.1	0.00535	0.00535	72.7 ± 2.8
$\tilde{S}_1$	0.1	0.00535	0.00535	72.7 ± 2.8
$\widehat{S}_1^*$	0.3	0.00534	0.00534	95.4 ± 1.3
$\widehat{S}_2$	0.008	0.00856	0.00856	72.3 ± 2.8
$\tilde{S}_2$	0.03	0.00866	0.00866	72.2 ± 2.8
$\widehat{S}_2^*$	0.2	0.00898	0.00898	94.9 ± 1.4
$\widehat{S}_3$	0.9‡	0.00363	0.00368	72.4 ± 2.8
$\tilde{S}_3$	0.6‡	0.00362	0.00364	72.8 ± 2.8
$\widehat{S}_3^*$	0.04	0.00380	0.00380	95.0 ± 1.4
C. Contamination = 0.25.				
$\widehat{S}_1$	0.005	0.00914	0.00914	65.5 ± 2.9
$\tilde{S}_1$	0.005	0.00914	0.00914	65.6 ± 2.9
$\widehat{S}_1^*$	0.4	0.00955	0.00956	95.1 ± 1.3
$\widehat{S}_2$	1.0‡	0.00939	0.00946	64.7 ± 3.0
$\tilde{S}_2$	0.7	0.00962	0.00965	64.4 ± 3.0
$\widehat{S}_2^*$	0.3	0.0106	0.0106	95.4 ± 1.3
$\widehat{S}_3$	0.9‡	0.00222	0.00229	66.6 ± 2.9
$\tilde{S}_3$	0.7‡	0.00221	0.00225	65.9 ± 2.9
$\widehat{S}_3^*$	0.1	0.00235	0.00235	95.5 ± 1.3

† We assume two categories of individuals that are sampled as follows:

Type I individuals: $p_1 = 0.80$, $q_1 = 0.15$
Type II individuals: $p_1' = 0.15$, $q_1' = 0.80$,

where p and q are the probabilities of two species utilizing resource 1.