Chapter 5 Resampling Methods

– Math 313 Statistics for Data Science

Guangliang Chen

Associate Professor

cheng@hope.edu

Hope College, Fall 2023

Presentation Overview

Introduction

Consider the following questions in a supervised context (regression or classification)

- k**NN classification**: How to choose a good k for classifying test data?
- **Dimension reduction (PCA) + a classifier**: How many principal components should we keep?
- **Regularized methods**: How to choose the value of the regularization parameter (λ) in each case:
  - Lasso:

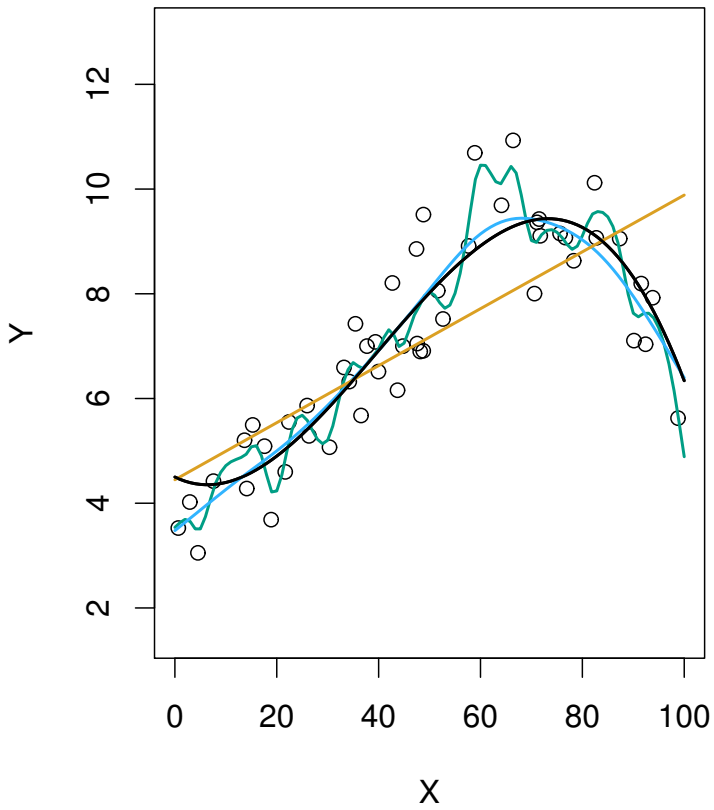
$$\min_{\hat{\boldsymbol{\beta}}} \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2 + \lambda \cdot \sum_j |\hat{\beta}_j|$$

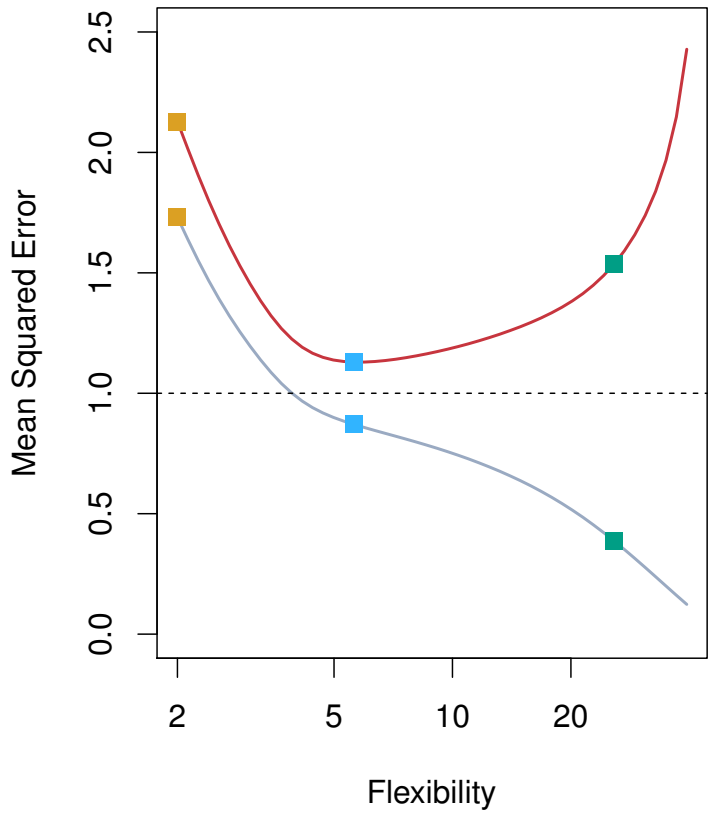
  - Regularized logistic regression:

$$\min_{\hat{\boldsymbol{\beta}}} \sum_{i=1}^{n} (-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)) + \lambda \cdot \sum_j |\hat{\beta}_j|$$

One naive idea is to compare the training errors corresponding to different choices of the respective parameter and choose the one leading to the smallest training error:

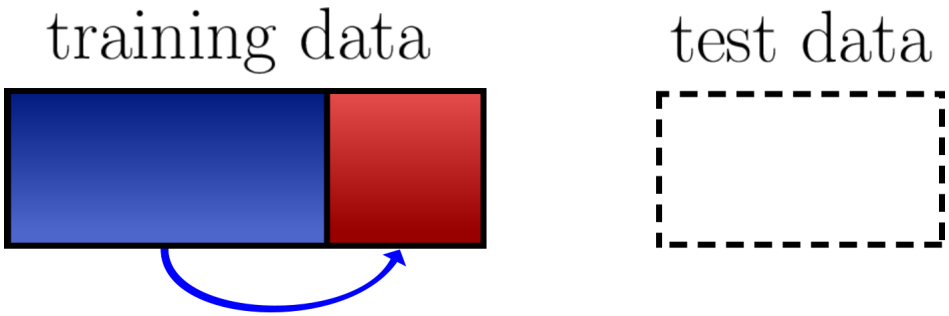
- This may lead to overfitting, and thus does not generalize well on test data
- Real target is the test error (red curve below)





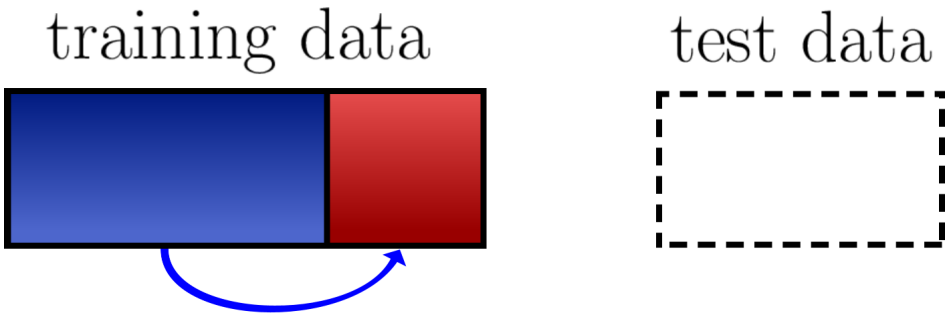
The Validation Set Approach

- Randomly divide the available set of observations into two parts, a training set and a validation set (or hold-out set).



The Validation Set Approach

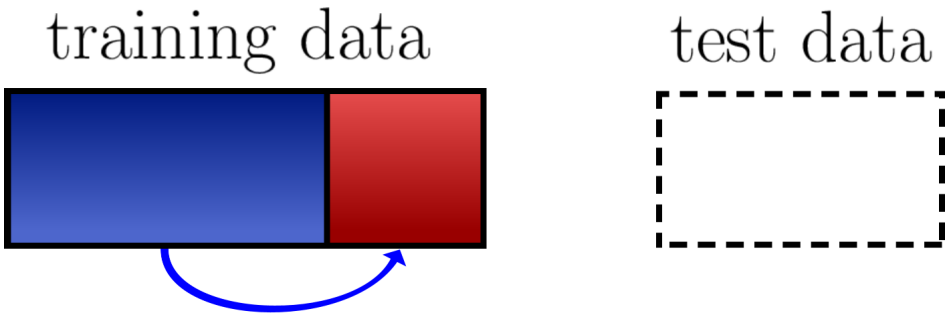
- Randomly divide the available set of observations into two parts, a training set and a validation set (or hold-out set).



- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.

The Validation Set Approach

- Randomly divide the available set of observations into two parts, a training set and a validation set (or hold-out set).



- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.
- The resulting validation set error rate – typically assessed using MSE in the case of a quantitative response – provides an estimate of the test error rate.

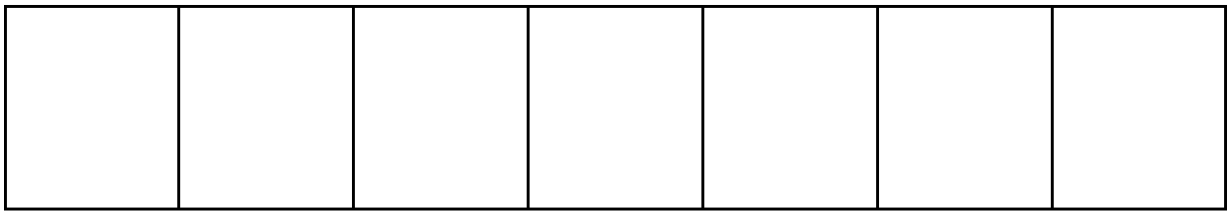
The validation set approach is conceptually simple and is easy to implement.

But it has two potential drawbacks:

- The validation estimate of the test error rate can be highly variable, due to the randomness associated with the partition.
- In the validation approach, only a subset of the observations are used to fit the model. Since statistical methods tend to perform worse when trained on fewer observations, this suggests that the validation set error rate may tend to overestimate the test error rate for the model fit on the entire data set.

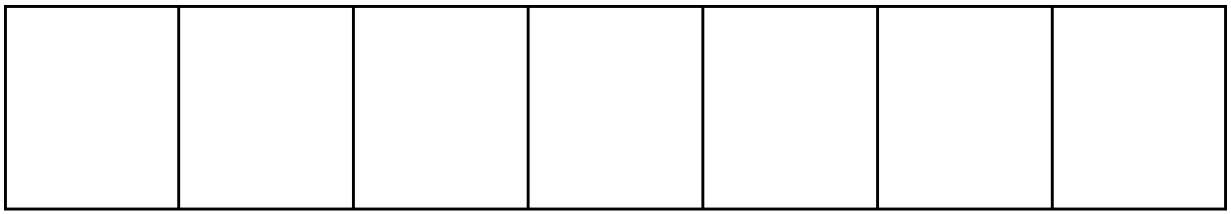
Cross-validation for regression

- This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size.

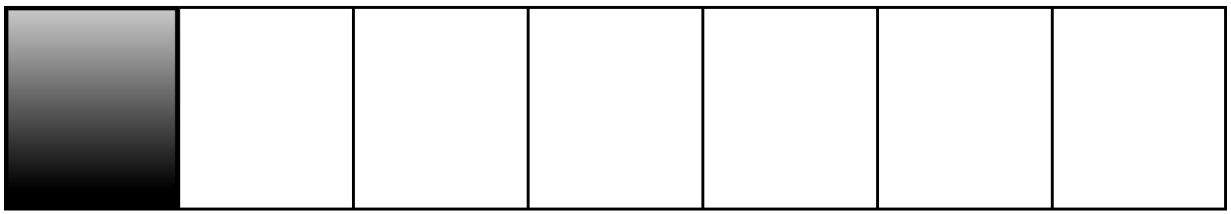


Cross-validation for regression

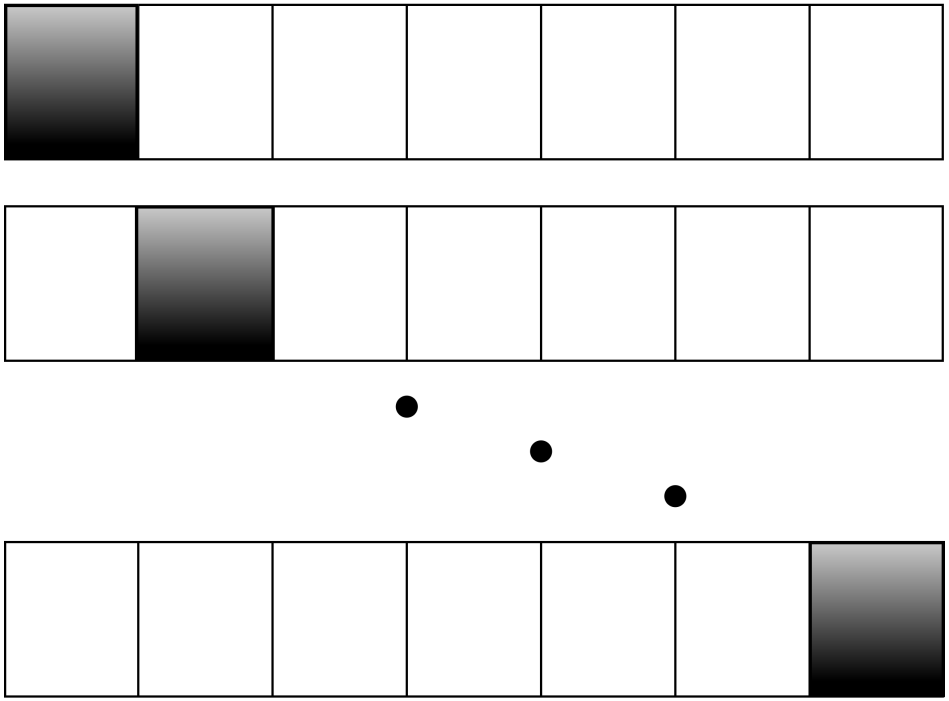
- This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size.



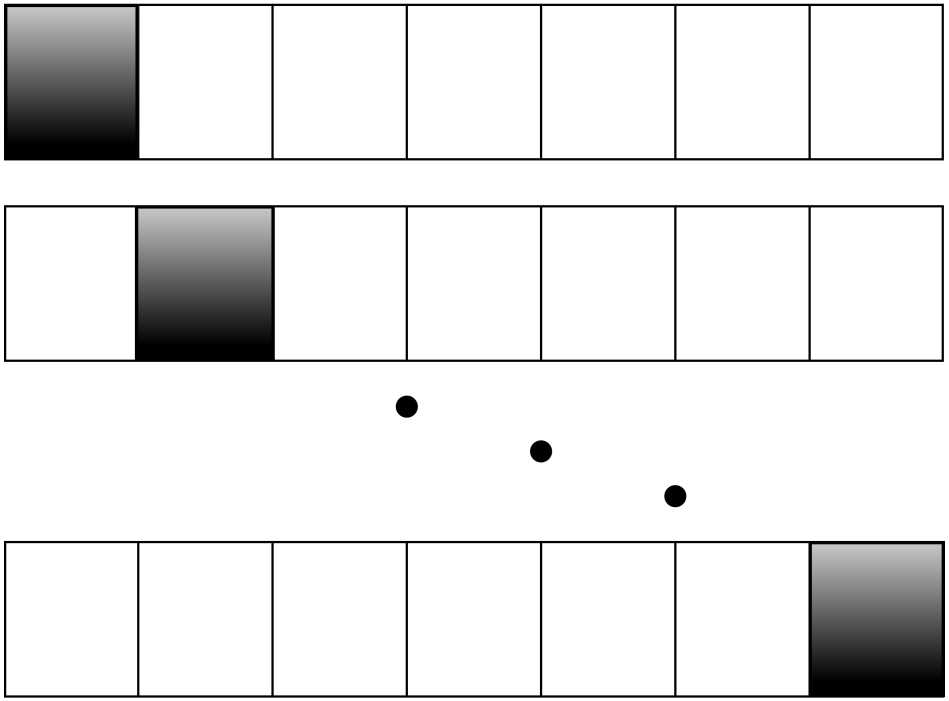
- The first fold is treated as a validation set, and the method is fit on the remaining $k-1$ folds. The mean squared error, MSE_1, is then computed on the observations in the held-out fold.



- This procedure is repeated k times; each time, a different group of observations is treated as a validation set.



- This procedure is repeated k times; each time, a different group of observations is treated as a validation set.

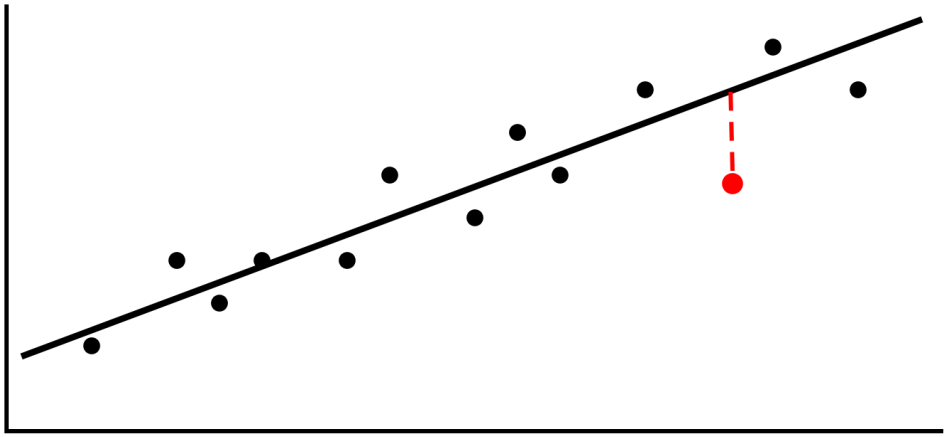


- This process results in k estimates of the test error, $\text{MSE}_1, \text{MSE}_2, \ldots, \text{MSE}_k$. The **$k$-fold CV estimate** is computed by averaging these values,

$$\text{CV}_{(k)} = \frac{1}{k} \sum_{i=1}^{k} \text{MSE}_i$$

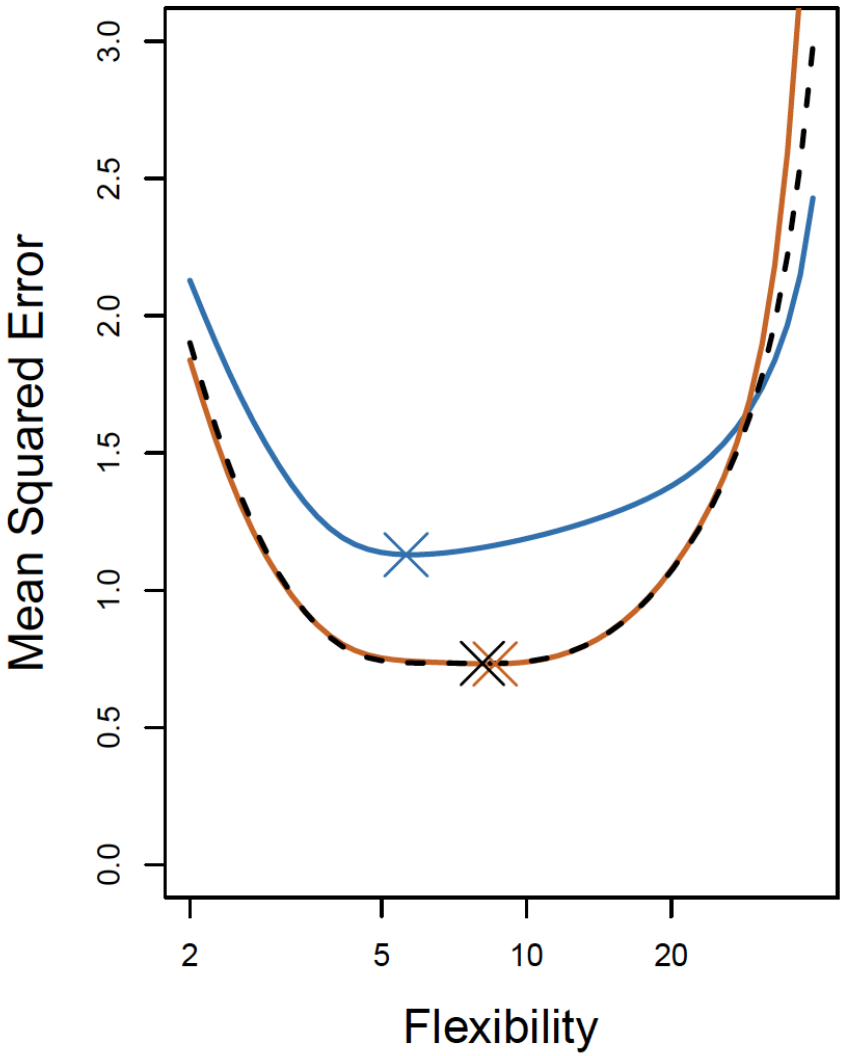
In practice, k (number of folds) is normally set to 5 or 10.

In the extreme case when $k = n$ (the total number of observations), k-fold CV is called **leave-one-out CV (LOOCV)**.



Remark. In general, k-fold CV (with small values of k such as 10) works better than LOOCV ($k = n$):

- k-fold CV with $k < n$ has a computational advantage to LOOCV.
- k-fold CV often gives more accurate estimates of the test error rate than does LOOCV.



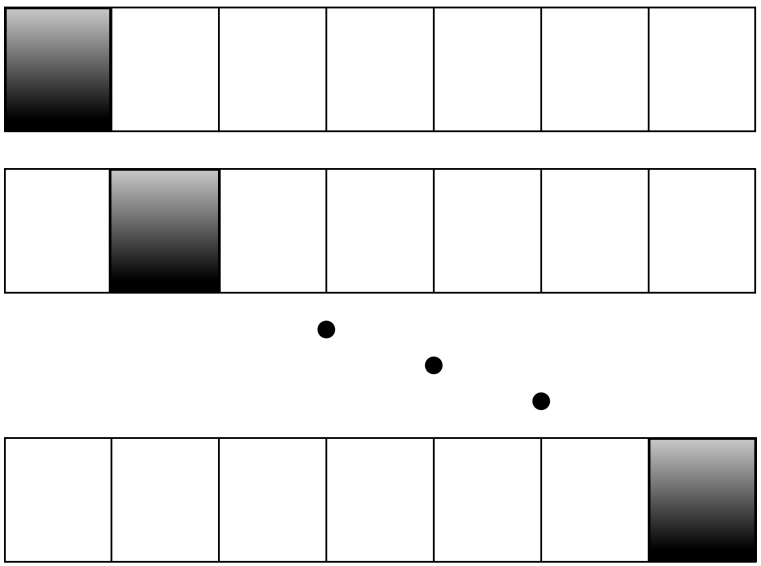
The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the 10-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.

Cross-validation on classification problems

In the classification setting, k-fold cross-validation works similarly:

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{Err}_i$$

where Err_i represents the fraction of misclassified points in fold i when it is used as validation set (and the model is trained on all other $k-1$ folds).



Demonstration using logistic regression and kNN

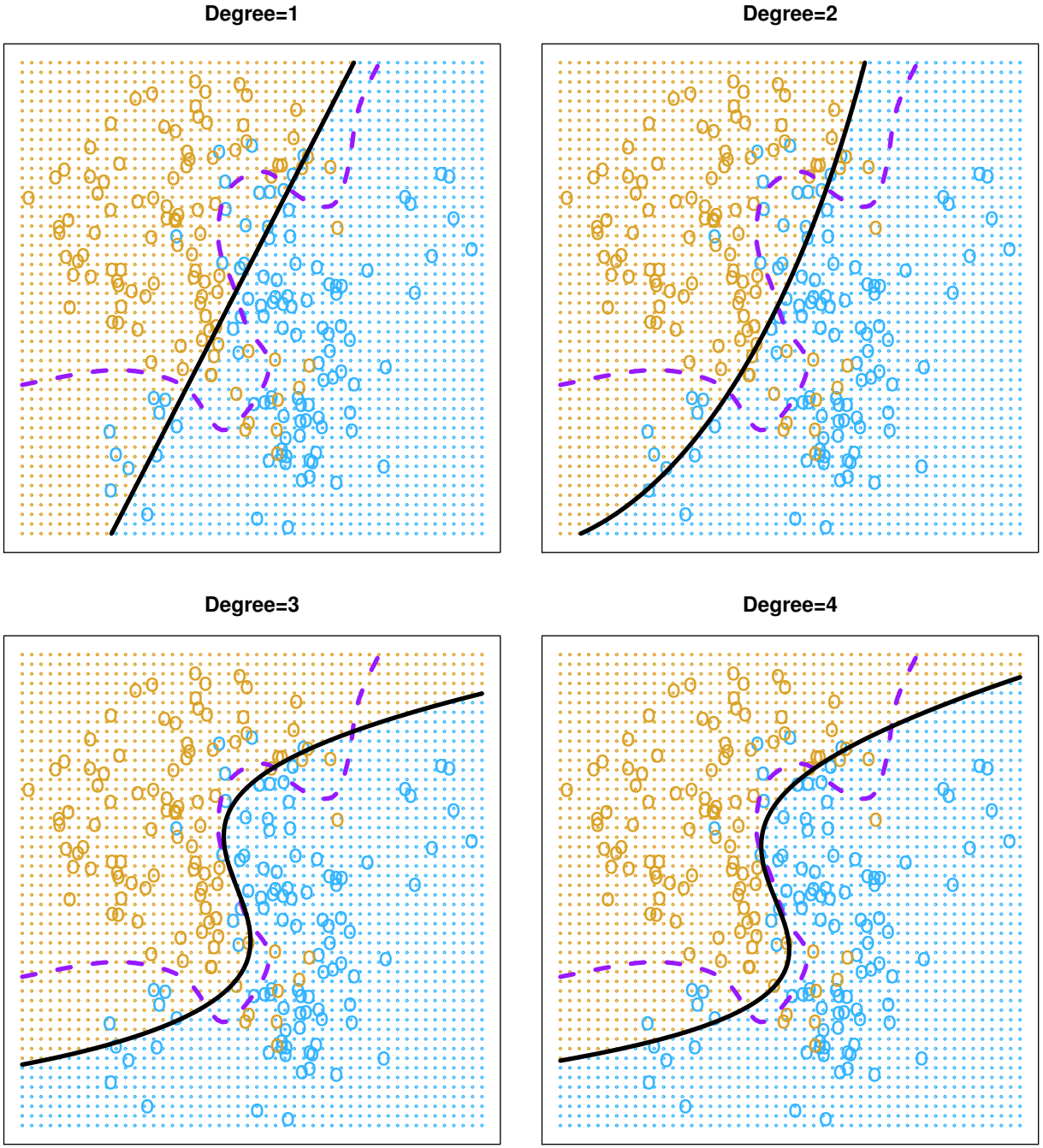
Consider logistic regression again, which is a linear classifier. One way to extend logistic regression to obtain a non-linear decision boundary is by using polynomial functions of the predictors, as we did in the regression setting.

For example, we can fit a quadratic logistic regression model (on a 2D training set) as follows

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2$$

If the quadratic terms are not enough, then we can go to third order, or fourth order.

How can we choose the order of the polynomial in this setting?


Degree=1
Degree=2
Degree=3
Degree=4

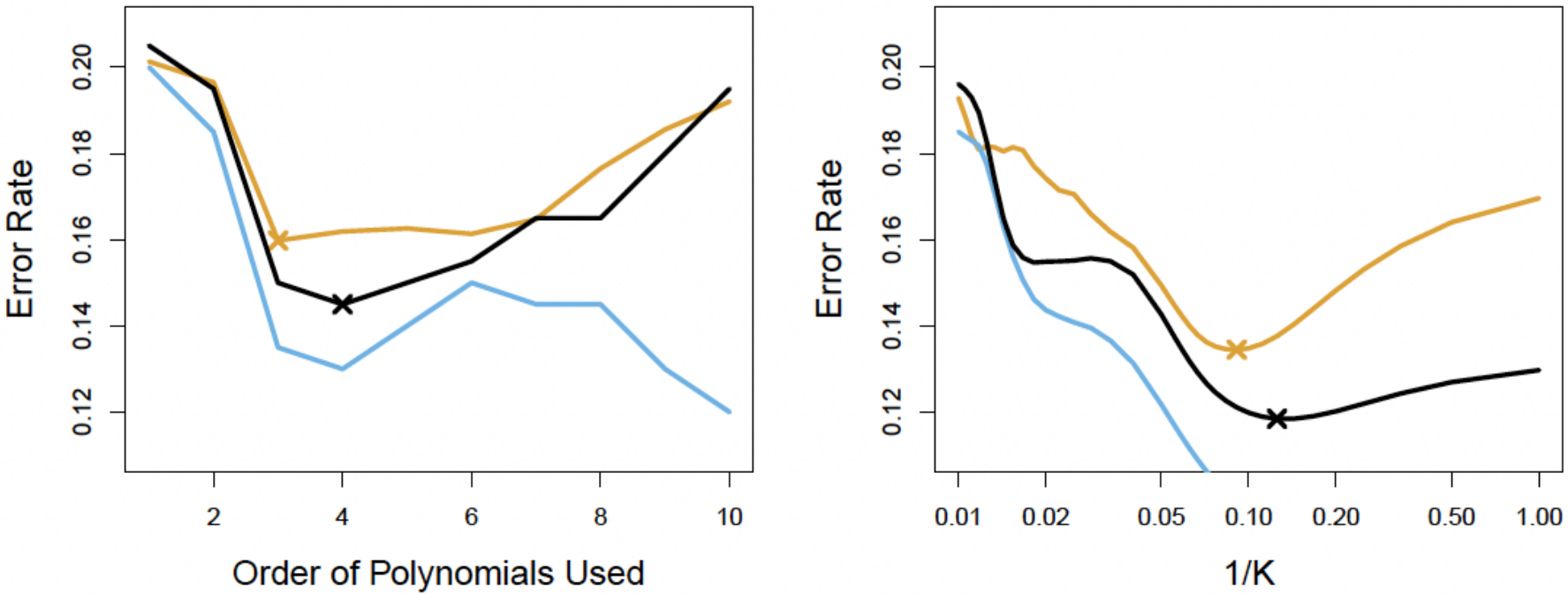


FIGURE 5.8. *Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.*

Section 5.2 The Bootstrap

Suppose you are given a sample of size 30 from some population. You then compute the value of a statistic T you choose (e.g., the sample mean) and get $t_1 = 4.9$. What can you say about the variability of the statistic?

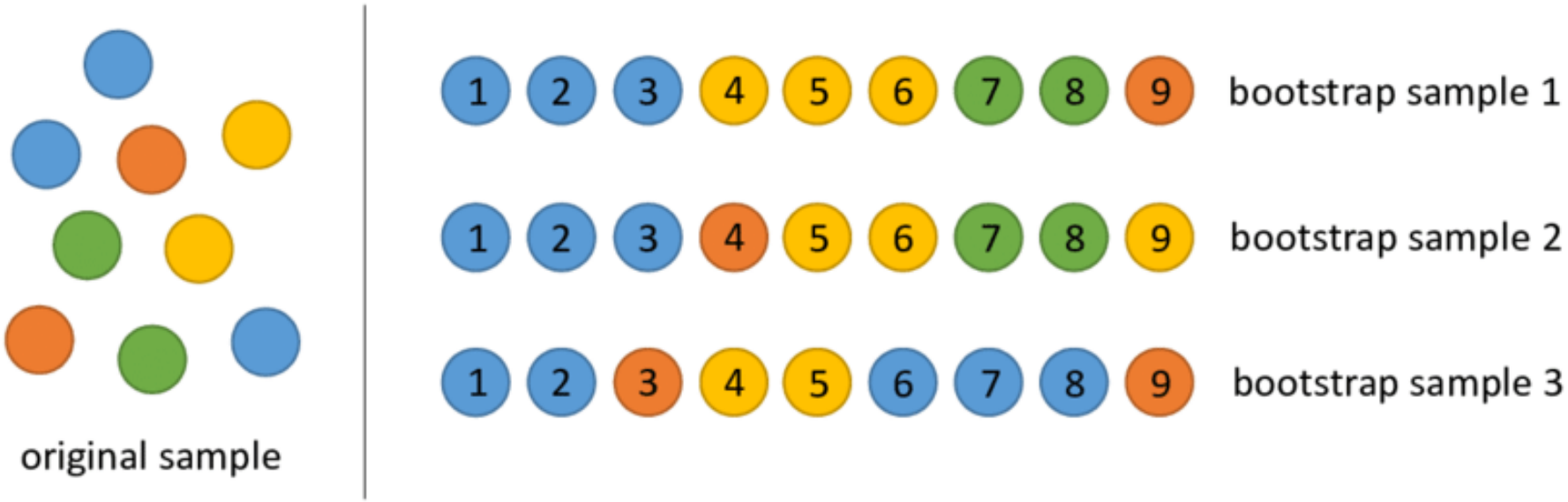
The question would be simple to answer if you can draw many more (say m) independent samples from the same population. You can then compute the corresponding values of T (say $t_2 = 4.1, t_3 = 5.3, \ldots, t_m = 4.7$) and use the following formula to compute the standard error of the statistic:

$$\text{StdErr}(T) = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(t_i - \bar{t})^2}, \qquad \text{where} \quad \bar{t} = \frac{1}{m}\sum_{i=1}^{m} t_i$$

However, in reality, it is impossible to obtain more samples from the population and the one sample given to us is what we have. What can we do?

This is where **bootstrap sampling** can help. How it works:

- Sample, *with replacement*, m data points from the data set we were originally given. Compute the value of T on this first bootstrap sample.
- Repeat the above process for more times (say 30 or 50) and obtain more values of T on those bootstrap samples.
- Compute the standard error of T using the above bootstrap values.



Why it works: Bootstrap sampling treats the one sample we have as an empirical approximation to the unknown population and then samples repeatedly from that original sample (viewed as a population).

The idea works well when the data set is an accurate representation of the population.

Bootstrap sampling will be needed when we get to ensemble learning (Chapter 8) - to be covered next.

In-class demonstration