Introduction to Data Science

DSA1101

Semester 1, 2018/2019
Week 5

n-fold cross validation in R

n-fold cross validation in `R`

- We have studied the *k*-nearest neighbor algorithm as an example of a classifier, and introduced some diagnostic metrics to evaluate the perfomance of a classifier.
- We have also been exposed to the concept of *bias-variance tradeoff*, which is a general property of predictive models.

n-fold cross validation in R

- In general, the prediction error for a model can be decomposed into

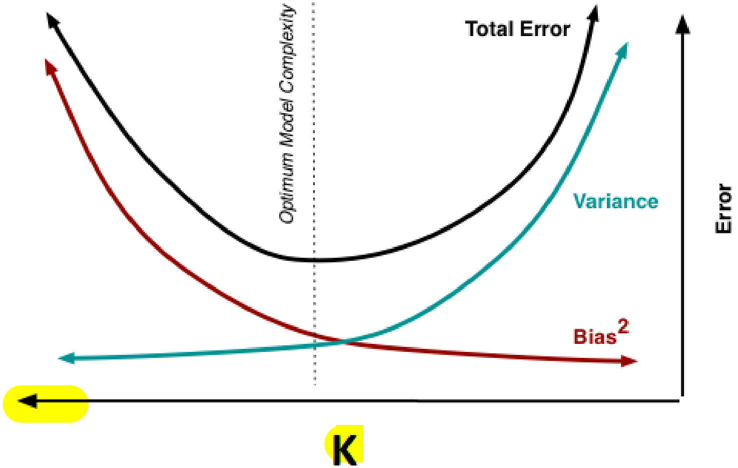
$$\text{error} = \text{bias}^2 + \text{variance} + \text{ irreducible error}$$

- Notice that in our example, for the larger value of $k = 5$, we take the average of five y values as our fitted value
- So the "variance" of our fitted value $\hat{Y}$ is smaller than when $k = 3$
- However, when $k = 5$, we are also taking data points further away from the circle to compute our fitted value. This may lead to greater "bias" in our fitted value $\hat{Y}$ compared to when $k = 3$.

n-fold cross validation in `R`

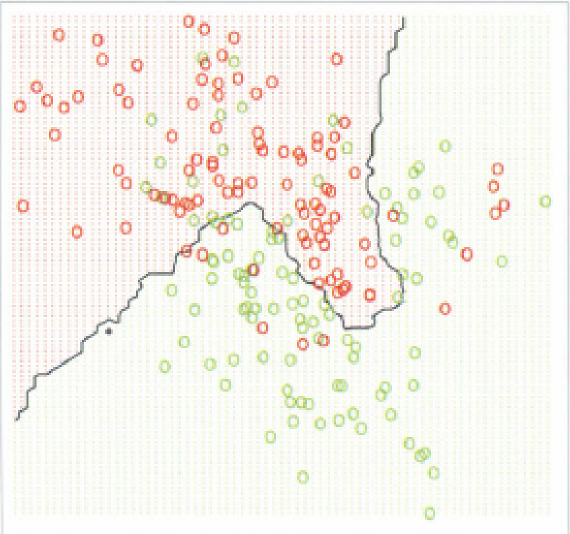
- So when *k* increases, the variance decreases, but bias increases
- This is known as the *bias-variance tradeoff*

n-fold cross validation in R



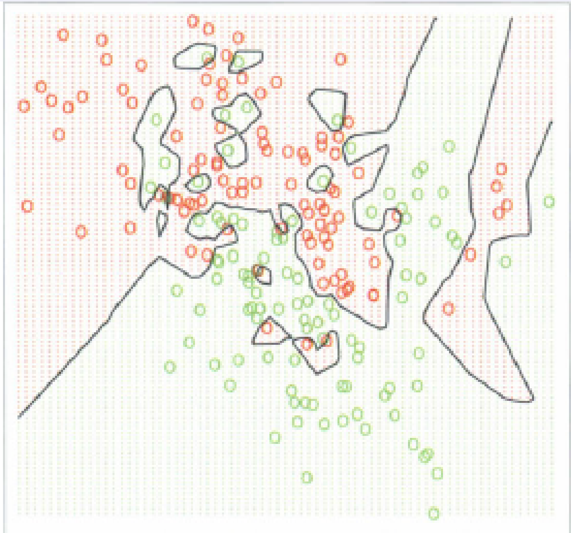
Bias-variance tradeoff. Source: *http://scott.fortmann-roe.com*

n-fold cross validation in R



Prediction by majority vote with 15 nearest neighbors. Source: *The Elements of Statistical Learning*, Hastie et al.

n-fold cross validation in R



Prediction by majority vote with one nearest neighbor. Source: *The Elements of Statistical Learning*, Hastie et al.

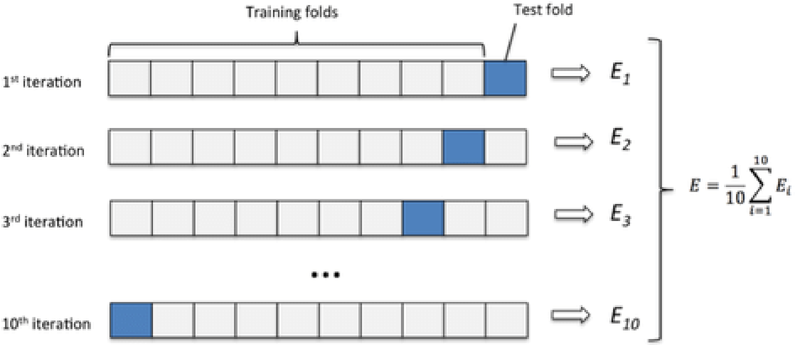
n-fold cross validation in R

- We have studied a number of measures that can be used to evaluate the performance of a classifier.
- In practice, when we are presented with a dataset, how should we go about estimating these performance measures?
- A common practice is to perform *N*-Fold Cross-Validation

n-fold cross validation in `R`

- The entire dataset is randomly split into N datasets of approximately equal size.
- N-1 of these datasets are treated as the training dataset, while the remaining one is the test dataset. A measure of the model error is obtained.
- This process is repeated across the various combinations of N datasets taken $N - 1$ at a time.
- The observed N models errors are averaged across the N folds.

Diagnostics of Classifiers



n-fold cross validation in R

- We will illustrate how to implement *n*-fold cross-validation in R to evaluate the performance of the *k*-nearest neighbor classifier
- In particular, we will attempt to estimate the optimal value of *k* (or the optimal model complexity) that will give the best classification performance

n-fold cross validation in R

- We will use a dataset with RNA expression levels for eight tissues to illustrate *n*-fold cross validation in R
- First, install and load the library ‘devtools’
- Then install the package tissuesGeneExpression from *GitHub*:

```
library(devtools)
install_github("genomicsclass/tissuesGeneExpression")
```

n-fold cross validation in R

- Load the library tissuesGeneExpression and its associated dataset
-
```r
library(tissuesGeneExpression)
data(tissuesGeneExpression)
head(e)
head(tissue)
```

```
> head(tissue)
[1] "kidney" "kidney" "kidney" "kidney" "kidney" "kidney"
```

n-fold cross validation in R

- For illustration purposes, we will remove data for *placenta* which does not have many samples:

-
  ```
  > table(tissue)
  tissue
   cerebellum       colon endometrium
           38          34          15
  hippocampus      kidney       liver
           31          39          26
     placenta
            6
  ```

```
ind <- which(tissue != "placenta")
y <- tissue[ind]
X <- t( e[,ind] )
```

n-fold cross validation in R

- Using the same dataset as both the training and testing data can give misleading results
- For example, when $k = 1$, we use each single data point to predict itself:
-
  ```
  > library(class)
  > pred <- knn(train=X, test=X, cl=y, k=1)
  > mean(y != pred)
  [1] 0
  ```
- Therefore there is a need for more principled methods to evaluate classifier performance, e.g. *n*-fold cross validation

n-fold cross validation in R

- Using the same dataset as both the training and testing data can give misleading results
- For example, when $k = 1$, we use each single data point to predict itself:

-
  ```
  > library(class)
  > pred <- knn(train=X, test=X, cl=y, k=1)
  > mean(y != pred)
  [1] 0
  ```
- Therefore there is a need for more principled methods to evaluate classifier performance, e.g. *n*-fold cross validation

n-fold cross validation in R

- We will perform 10-fold cross validation; first randomly split the 189 data points into 10 sets:
-
```r
set.seed(1)
n_folds=10
folds_i <- sample(rep(1:n_folds, length.out = 183))
```

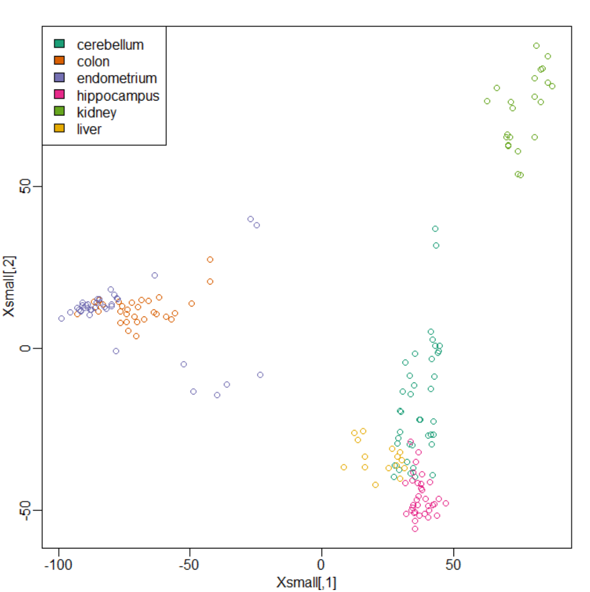
```
> table(folds_i)
folds_i
 1  2  3  4  5  6  7  8  9 10
19 19 19 18 18 18 18 18 18 18
```

n-fold cross validation in R

- For illustration purposes we will try to predict tissue type with just two dimensional features. We will reduce the dimension of our features using cmdscale from the package rafalib:
-

```r
library(rafalib)
mypar()
Xsmall <- cmdscale(dist(X))
plot(Xsmall,col=as.fumeric(y))
legend("topleft",levels(factor(y)),fill=seq_along(
    levels(factor(y))))
```

n-fold cross validation in R



n-fold cross validation in R

- We start with $k = 1$, and observe first iteration: using the first dataset as our test data and the remaining 9 datasets as training data

-
  ```r
  test_i <- which(folds_i == 1)
  pred <- knn(train=Xsmall[ -test_i, ], test=Xsmall[test_i, ], cl=y[ -test_i ], k=1)
  table(true=y[test_i ], pred)
  err=mean(y[ test_i ] != pred)
  err
  ```

n-fold cross validation in R

- We can start with the first iteration: using the first dataset as our test data and the remaining 9 datasets as training data

-
  ```
  > table(true=y[test_i ], pred)
               pred
  true           cerebellum colon endometrium hippocampus kidney
    cerebellum            4     0           0           0      0
    colon                 0     2           0           0      1
    endometrium           0     0           1           0      0
    hippocampus           1     0           0           2      0
    kidney                0     1           0           0      5
    liver                 0     0           0           0      0
               pred
  true           liver
    cerebellum       0
    colon            0
    endometrium      0
    hippocampus      0
    kidney           0
    liver            2
  > err=mean(y[ test_i ] != pred)
  > err
  [1] 0.1578947
  ```

n-fold cross validation in R

- Now we perform the whole 10-fold cross validation for $k = 1$:
-

```
err=numeric(10)
for (j in 1:10) {
  test_i <- which(folds_i == j)
  pred <- knn(train=Xsmall[ -test_i, ], test=
       Xsmall[ test_i, ], cl=y[ -test_i ], k=1)
  err[j]=mean(y[ test_i ] != pred)
}
err
error=mean(err)
error
```

n-fold cross validation in R

- Now we perform the whole 10-fold cross validation for $k = 1$:
-
```
> err
 [1] 0.15789474 0.21052632 0.26315789 0.22222222
 [5] 0.16666667 0.05555556 0.38888889 0.22222222
 [9] 0.11111111 0.16666667
> error=mean(err)
> error
[1] 0.1964912
```

- We have estimated the mis-classification error of the k-nearest neighbor classifier to be $\approx 19.6\%$ when $k = 1$
- Let us repeat the procedure for $k = 2, 3, ..., 15$

n-fold cross validation in R

- Let us repeat the procedure for $k = 2, 3, ..., 15$

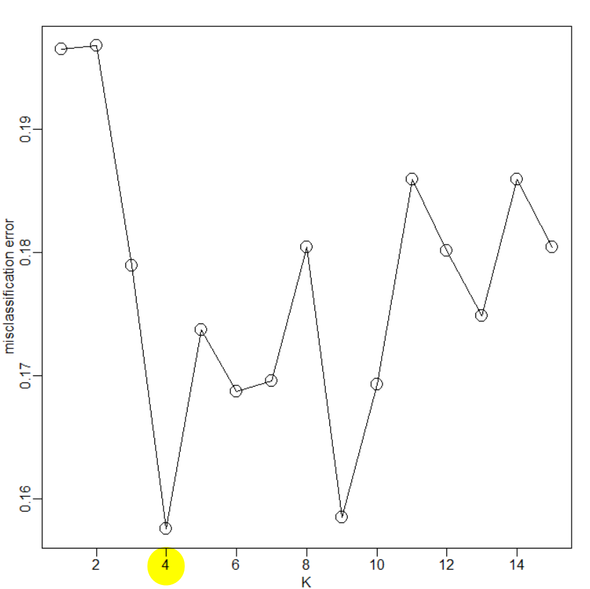
-
  ```r
  error=numeric(15)

  for (k in 1:15) {
    err=numeric(10)
    for (j in 1:10) {
      test_i <- which(folds_i == j)
      pred <- knn(train=Xsmall[ -test_i, ], test=
          Xsmall[ test_i, ], cl=y[ -test_i ], k=k)
      err[j]=mean(y[ test_i ] != pred)
    }
        error[k]=mean(err)
  }
  ```

n-fold cross validation in `R`

- We can plot the error rate against value for *k*:

-
  ```
  plot(1:15, error, type="o",ylab="misclassification
        error",xlab="K",cex.axis=1,cex=2)
  ```

n-fold cross validation in R


misclassification error
K