STAT 408
Applied Regression Analysis

Nan Miles Xi

Department of Mathematics and Statistics

Loyola University Chicago

Fall 2022

Linear Regression and Prediction

Prediction

- Prediction is a typical supervised learning task
  - Given body part measurement, predict a person's body fat
  - Given the patient's tumor information, predict if she will survive after certain period

- Given a dataset with p predictors $X_1, X_2, \dots, X_p$ and one response variable Y

- We use the data to build a model f, a function of predictors and response

$$Y = f(X_1, X_2, \dots, X_p)$$

Linear Regression for Prediction

- Given a new data point $(x_1,\ x_2, \ldots, x_p)$, we use the model output

$$f(x_1, x_2, \ldots, x_p)$$

as the prediction for response y

$$\hat{y} = f(x_1, x_2, \ldots, x_p)$$

- In the context of linear regression, function $f(X_1, X_2, \ldots, X_p)$ is a linear combination of all predictors

$$f(X_1, X_2, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Linear Regression for Prediction

- The model prediction for new observation $(x_1, x_2, \ldots, x_p)$ is calculated as

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

- The prediction task is called supervised learning because we use response Y to "supervise" or "train" the model

- In unsupervised learning (dimension reduction, clustering), there is no Y's information in the whole process
  - Either don't use or unknown

Category of Prediction

- There are two categories of prediction, depending on the type of the response variable

- If the response variable we want to predict is continuous, it is regression
  - Predict body fat or gene expression levels

- If the response variable we want to predict is categorical, it is classification
  - Predict the cancer type or if the patient can survive

Measure of Prediction Accuracy

- In regular linear model, we don't give any restriction on response variable, so

$$Y \in (-\infty, +\infty)$$

- Therefore, the regular linear model predicts continuous outcome

- How can we measure the prediction accuracy for linear model?

Measure of Prediction Accuracy

- One good accuracy measurement is the average difference between true response y and predicted response $\hat{y}$

$$\text{Prediction error} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

- This is called mean square error (MSE)
  - MSE = RSS/n

- But what is the potential issue for this measurement?

Measure of Prediction Accuracy

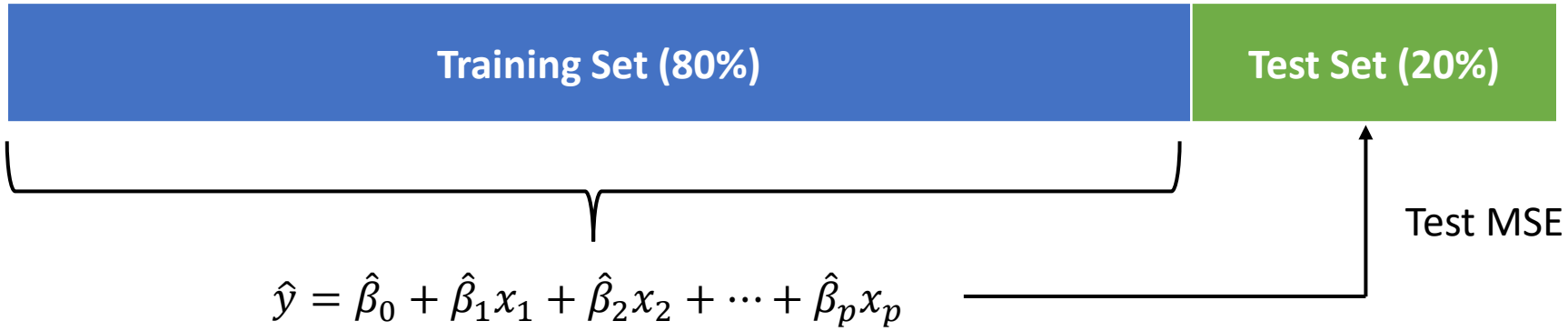
- Recall that prediction is to predict “unseen” data
  - The prediction accuracy should be measured on the “unseen” data, not the data the model already knew

- To objectively measure the prediction accuracy, we randomly split the data into a training set and a test set

- The model is fitted/trained on the training set, and predicts the response variable on the test set

- The MSE calculated on the test set would be an unbiased estimation for model prediction accuracy

Measure of Prediction Accuracy



$$\text{Test MSE} = \frac{\sum_{i \in \{test\}} (y_i - \hat{y}_i)^2}{n_{test}}$$

- The training-test split imitates the real scenario of predicting unknow data
  - The model training and prediction are performed on two different datasets

Example

- We use ‘fat’ dataset to build a multiple linear regression model for prediction

- Response variable
  - Percentage of body fat

- Predictors
  - Age, weight, height, and 10 body circumference measurements

- Sample size is 252

Prediction Accuracy in R

```
# read fat dataset
fat <- read.csv('fat.csv’)

# random split the data into 80% training set and 20% test
set.seed(2022)
index.train <- sample(1:dim(fat)[1], 0.8 * dim(fat)[1])
data.train <- fat[index.train,]
data.test <- fat[-index.train,]

# fit a linear model on the training set
lm.model <- lm(brozek ~ weight + abdom + forearm + wrist, data=data.train)
```

Prediction Accuracy in R

```
# predict body fat on the test set
yhat.test <- predict(lm.model, data.test)

# calculate test MSE
y.test <- data.test$brozek
MSE.test <- mean((y.test - yhat.test)^2)
MSE.test
```

Other Accuracy Measurement

- We can take square root to make test MSE have a same unit as response variable

$$\text{RMSE (root MSE)} = \sqrt{\frac{\sum_{i\in\{test\}}(y_i - \hat{y}_i)^2}{n_{test}}}$$

- We can divide RMSE by the average of test Y to obtain the normalized prediction error (percentage)

$$\text{NRMSE (normalized root MSE)} = \frac{\sqrt{\frac{\sum_{i\in\{test\}}(y_i - \hat{y}_i)^2}{n_{test}}}}{\bar{y}_{test}}$$

Prediction Accuracy in R

```
# calculate root MSE
RMSE.test <- sqrt(MSE.test)
RMSE.test

# calculate normalized root MSE
NRMSE.test <- RMSE.test / mean(y.test)
NRMSE.test
```

- The linear model gives a 20.9% error for body fat prediction

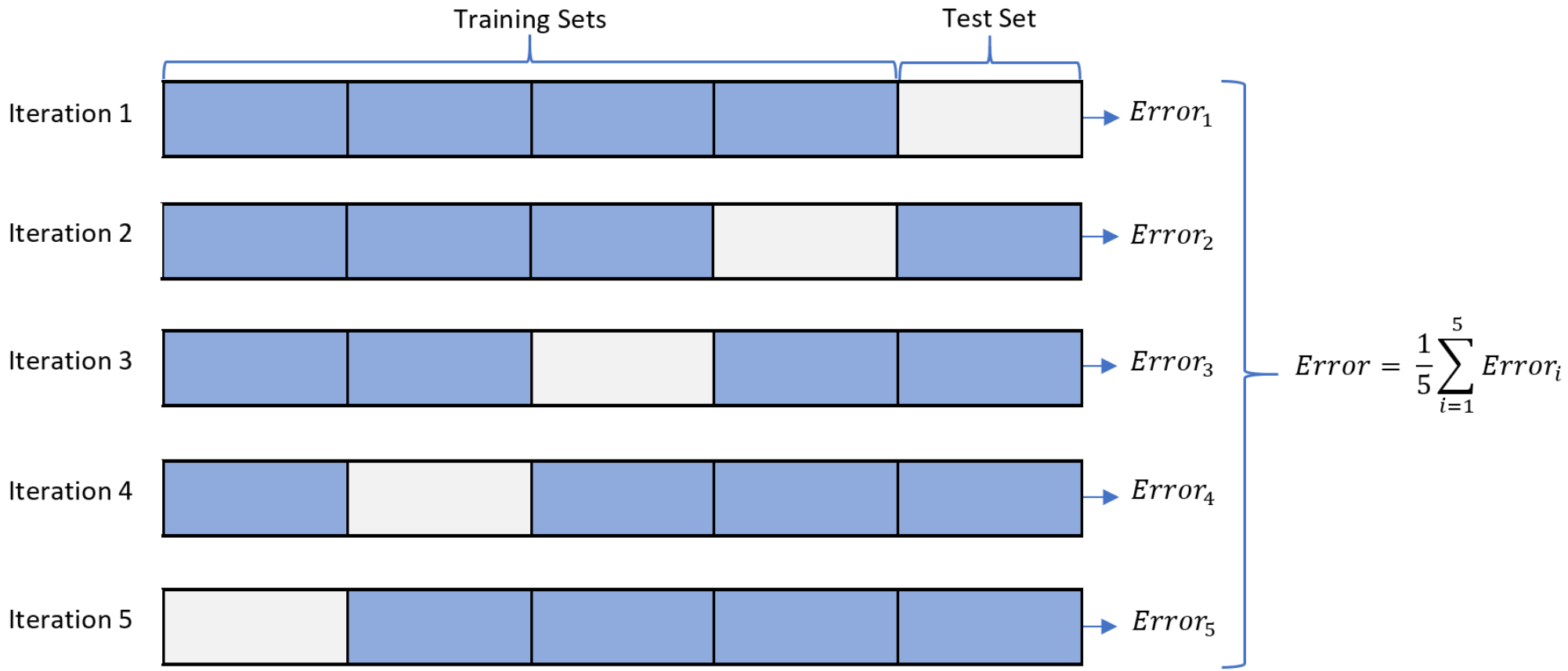
Cross-Validation

- The training-test split provides an objective estimation for prediction accuracy

- But it has two drawbacks:
  1. The split is fixed, the accuracy is measured only on 20% of data
  2. Waste the test data in the model training, reduce the effective sample size

- The solution for these two problems is cross-validation
  1. Randomly split the data into K equal-sized parts
  2. Fit the model on the K-1 parts (train), predict on the left-out part (test)
  3. Repeat by treating every part as the test set
  4. Average the K prediction accuracies or errors

Cross-Validation



- In general, it is called k-fold cross-validation (k=5 in this example)

Cross-Validation

- If we let K = n, then the model is trained on n-1 observations and predict on only one test data point
- This is called leave-one-out cross-validation (LOOCV)
- LOOCV makes use the data as much as possible, but is computationally expensive
- Check “code 11.R” to see the implementation of cross-validation in R

Prediction in Logistic Regression

- Since the output of logistic regression is binary, it is suitable for binary classification problem (yes or no)

- All the methods to measure prediction accuracy in regression apply to classification
  - Training-test split, cross validation, …

- Because the output is not continuous, the test MSE is not appropriate for the accuracy measurement

Prediction in Logistic Regression

- For binary classification, a straightforward accuracy measurement is the proportion of correct prediction

$$\text{Test accuracy} = \frac{\sum_{i=1}^{n_{test}} \mathrm{I}(y_i = \hat{y}_i)}{n_{test}}$$

- The I() function is identify function

$$I(X) = \begin{cases} 1 & if\ X = True \\ 0 & if\ X = False \end{cases}$$

Logistic Regression in R

- “Heart” dataset includes 303 patients’ clinical information and heart disease outcome
- 13 predictors on patients’ heart and lung functions
- The binary response variable is about if the patient is diagnosed with heart disease
- The detailed variable description can be found at
  - https://www.r-bloggers.com/2019/09/heart-disease-prediction-from-patient-data-in-r/
- See “code 12.R” for the implementation of model estimation

Classification Accuracy Measurement

- The simple test accuracy doesn't tell a full story about model performance
  1. False positive: $\hat{Y} = 1$ but $Y = 0$
  2. False negative: $\hat{Y} = 0$ but $Y = 1$

- Confusion matrix shows the accuracy for each class and gives a more detailed classification performance

	y.truth	
y.pred	0	1
0	32	7
1	1	20

- How many true positive, true negative, false positive, false negative?

Classification Accuracy Measurement

- True positive rate (sensitivity)

$$\frac{True\ positives}{All\ positives} = \frac{20}{20+7} = 74.07\%$$

- True negative rate (specificity)

	y.truth	
y.pred	0	1
0	32	7
1	1	20

$$\frac{True\ negatives}{All\ negatives} = \frac{32}{32+1} = 96.97\%$$

Classification Accuracy Measurement

- Precision

$$\frac{True\ positives}{Predicted\ positives} = \frac{20}{20+1} = 95.24\%$$

y.pred	y.truth 0	1
0	32	7
1	1	20

- See “code 12.R” for the implementation of different classification accuracy measurements

Classification Accuracy Measurement

- To summarize the model performance
  1. True positive rate = 74.07%
  2. True negative rate = 96.97%
  3. Precision = 95.24%

- The model is better to predict negative observations (no heart disease) than positive observations (heart disease)

- Among predicted positives, most of them are true positives (95.24%)

Classification Accuracy Measurement

- We need a single number that incorporates model's prediction capacity for both positive and negative observations

- Note that all the measurements are calculated based on

$$Y = \begin{cases} 0 & if (P = 1) < 0.5 \\ 1 & if\ (P = 1) \geq 0.5 \end{cases}$$

- The threshold decides the model behavior
  1. Larger threshold makes it harder to predict as 1: more negative predictions
  2. Smaller threshold makes it easier to predict as 0: more positive predictions

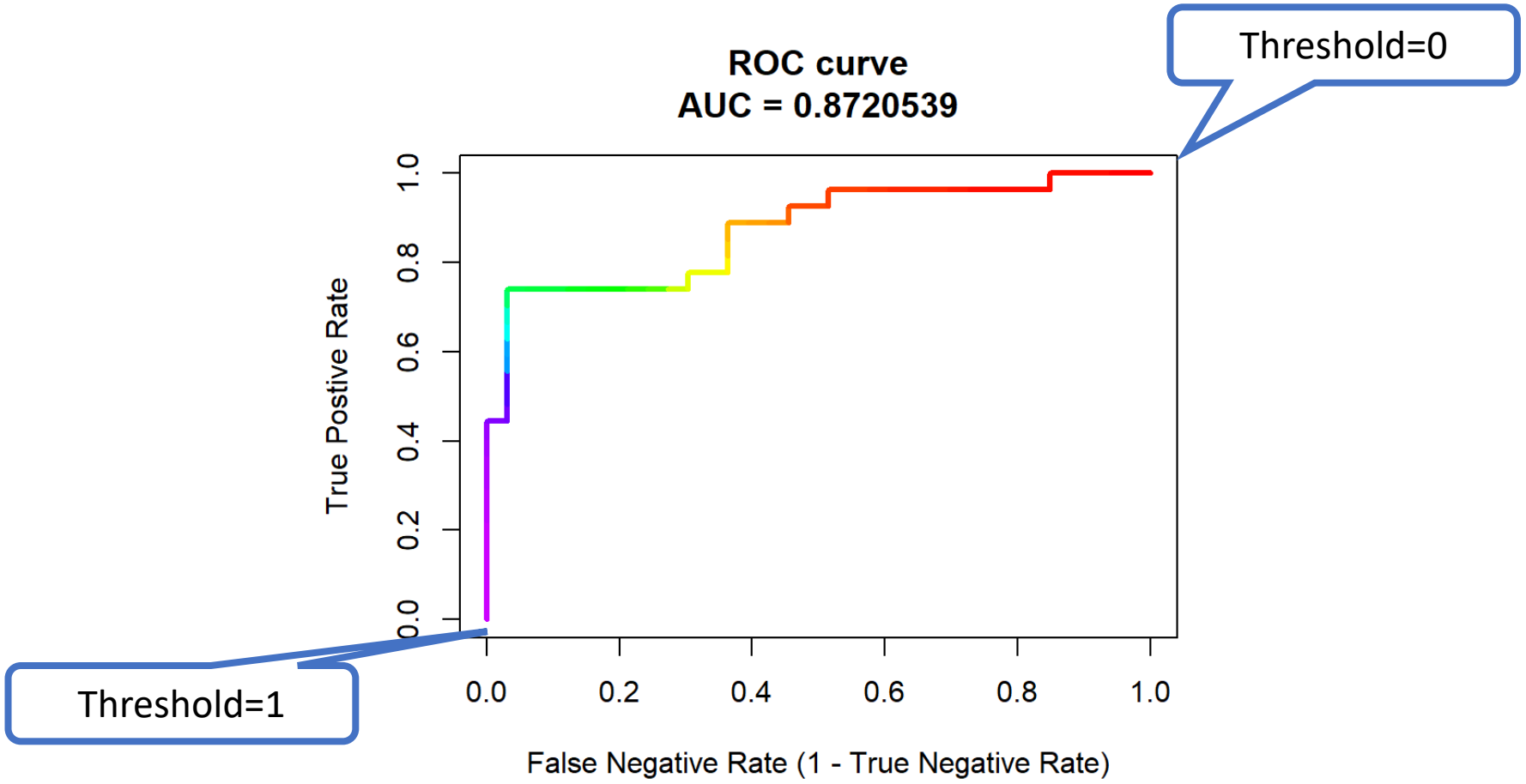
Classification Accuracy Measurement

- Adjusting the threshold will change the true positive rate and false negative rate

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
TPR	1	0.889	0.815	0.741	0.741	0.741	0.667	0.593	0.556	0.481	0
TNR	0	0.545	0.636	0.727	0.788	0.970	0.970	0.970	0.970	0.970	1

- We can plot all TPRs and TNRs in one figure to show the model behavior

Classification Accuracy Measurement



- The area under the ROC curve (AUROC) is between 0 and 1

Classification Accuracy Measurement

- AUROC is a good metric to measure the binary classification performance
  - A perfect model: AUROC = 1
  - A “random guessing model”: AUROC = 0.5
  - A good model: AUROC = 0.8 ~ 0.9

- See “code 12.R” for the implementation of ROC curve and AUROC