Data 311: Machine Learning

Lecture 2: Notation and Terminology

Dr. Irene Vrbik

University of British Columbia Okanagan

Outcomes

- Define **Statistical vs. Machine learning**
- Differentiate between **Inference vs. Prediction**
- Understand the difference between **Supervised vs Unsupervised** Learning
- Distinguish between **Classification vs. Regression** problems
- Recognize and correctly interpret the common **Notation**[1]
- **Training vs. Testing**

1. as used in **ISLR2** *Notation and Simple Matrix Algebra* (pg 9–12).

Motivating Example

- The following data sets are from the ISLR2 supported files.
- They will serve as motivating examples throughout this course.
- The easiest way to access them in through the **ISLR2 R package**

```
# install.packages("ISLR2")
library(ISLR2)
attach(Wage)
```

Wage

- The `Wage` data set comprise the wages from 3000 males from the Atlantic regions of the United States.
- There are 11 variables (type `?Wage` for details) :
  - `year`, `age`, `maritl`, `race`, `education`, `region`, `jobclass`, `health`, `health_ins`, `logwage`, and `wage`
- When analyzing this data, we might have different types of questions in mind …

Visualization adapted from **stackoverflow discussion**

Exploration

What is the age range in this data set?: `range(age)` = 18, 80
How does wage differ across education status?

Introduction

Data:

- input x (features, independent variables)
- output y (response/dependent variables)
  Source: Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical science, 16(3), 199-231. 🔗Link

Approaches

There are two goals in analyzing this data…

Data Modeling Culture

Algorithmic Modeling Culture

Two Cultures

Data Modeling Culture:
Assumes data is generated by a specific stochastic process that they wish to model.

may be preferred for **inference**

Algorithmic Modeling:
Often treats the data generation process as unknown or complex.

---

may be preferred when **prediction** is your goal

Source: Leo Breiman. "Statistical Modeling: The Two Cultures." Statist. Sci. 16 (3) 199 - 231, August 2001. **https://doi.org/10.1214/ss/1009213726**

Inference vs. Prediction

Inference goal: we might fit a linear model to understand how specific features (eg. square footage, number of bedrooms, or proximity to amenities) influence house prices.

Prediction goal: we might use random forests with the same features to predict housing prices, with less emphasis on interpreting each feature's impact.

Inference

Objective: To understand the relationship between x and y in order to draw conclusions about the population or data generating mechanism.
Focus: Emphasizes the interpretability and simplicity of the model

Prediction

Objective: To accurately forecast the output (response y) for new, unseen data points based on existing patterns in the data.
Focus: Prioritizes accuracy of predictions, often using "black box" algorithms

Examples

Inference Examples:

- Estimating the effect of a drug on blood pressure
- testing if there is a significant difference between groups
- determining the association between smoking and lung cancer

Prediction Examples:

- Predicting house prices based on features like size and location
- forecasting stock prices
- classifying whether an email is spam or not
- weather forecast

My take on the two cultures

Statistical Learning

Like the **Data Modeling Culture** “Statistical Learning” assumes a specific stochastic model (e.g., linear regression, logistic regression). They prioritize interpretability and understanding the relationship between variables.

- Primarily rooted in Statistics

Machine Learning

Like the **Algorithmic Modeling Culture**, “Machine Learning” emphasizes predictive accuracy and the ability to handle complex, high-dimensional data without predefined assumptions about the data structure.

- Primarily rooted in Computer Science

Statistical vs. Machine learning

- The authors of **ISLR** chose “Statistical Learning” to emphasize their statistics-focused approach to machine learning, as opposed to a computer science emphasis.[1]
- A distinction often made is that statistical and machine learning serve a different **purpose**.
- Statistical learning tends to focus on ***inference***, while Machine learning tends to focus on ***prediction***.

1. Source: Statistical Learning: 1.1 Opening Remarks: **https://youtu.be/LvySJGj-88U?si=q37Css-rgxkhpjUP**

Takeaway Message

- No single approach is universally superior
- Both “camps” have their strengths and weaknesses
- Our goal is to adopt a broad toolkit to handle a wide range of problems

iClicker Question

Which of the following is a strength of the algorithmic modeling culture compared to the data modeling culture?

a. Excels in predictive accuracy but lacks interpretability

b. Emphasizes interpretability and simplicity

c. Relies on strong assumptions about data structure.

d. Focuses mainly on inference

Supervised vs Unsupervised

Supervised Learning

- **Supervised learning** is characterized by the presence of "answers" in the data set which are utilized to supervise the algorithm.
- To put another way, the data is comprised of both inputs (AKA *predictor variables* or *features*) X and outputs (AKA *response variable*) Y (our "answers").
  - Y can be *categorical*, eg. cat/dog, spam/not spam, positive/negative/neutral
  - Y can be *numeric* eg. wage, stock price

Classification vs. Regression

Depending on the format of Y (i.e. categorical or numeric), supervised learning will perform one of the following tasks:

1. **Classification**: for predicting a *discrete category/label* based on the input data, or
2. **Regression**: for predicting a *numerical values* based on the input data.

Unsupervised Learning

- **Unsupervised learning** attempts to learn relationships and patterns from data that are not labeled in any way.
- In other words, we have only inputs X and no Y.
- Unsupervised learning is a more challenging task than supervised.

Source: https://vas3k.com/blog/machine_learning/

iClicker Question

What is the main difference between supervised and unsupervised learning?

a. Supervised learning only works with numbers, while unsupervised learning only works with text.

b. Supervised learning predicts categories, while unsupervised learning predicts continuous values.

c. Supervised learning and unsupervised learning are the same; they both use labeled data to make predictions.

d. Supervised learning uses labeled examples, while unsupervised learning tries to find patterns within unlabelled data.

iClicker Question

What is the main difference between classification and regression?

a. Classification only works with numbers, while regression only works with text.

b. Classification predicts categories, while regression predicts continuous values.

c. Classification is used for organizing data, while regression is used for finding patterns.

d. Classification and regression are the same; they both predict categories.

Notation

This section will cover some basic notation and refresh our memory on Matrices, Vectors, and scalars

Wage dataset

Let's consider the `Wage` dataset from the ISLR2 package.

```
# install.packages("ISLR2")
library(ISLR2)
attach(Wage)
```

- The `Wage` data set comprise the wages from 3000 males from the Atlantic regions of the United States.
- There are 11 variables: `year`, `age`, `maritl`, `race`, `education`, `region`, `jobclass`, `health`, `health_ins`, `logwage`, and `wage`

Notation

Notation is not standard across different disciplines, courses, or textbooks. We adopt the same notation used in ISLR2:

- n: the number of distinct observations in our sample
- p: the number of features available for making predictions.

> **Number of features**
>
> Most places will use p to denote the total number of columns excluding the response variable while others will use it to count all variables. We we use the former.

Viewing your data

Structure | **View spreadsheet** | **Head**

```
1 str(Wage)
```

```
'data.frame':   3000 obs. of  11 variables:
 $ year      : int  2006 2004 2003 2003 2005 2008 2009 2008 2006 2004 ...
 $ age       : int  18 24 45 43 50 54 44 30 41 52 ...
 $ maritl    : Factor w/ 5 levels "1. Never Married",..: 1 1 2 2 4 2 2 1 1 2 ...
 $ race      : Factor w/ 4 levels "1. White","2. Black",..: 1 1 1 3 1 1 4 3 2 1 ...
 $ education : Factor w/ 5 levels "1. < HS Grad",..: 1 4 3 4 2 4 3 3 3 2 ...
 $ region    : Factor w/ 9 levels "1. New England",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ jobclass  : Factor w/ 2 levels "1. Industrial",..: 1 2 1 2 2 2 1 2 2 2 ...
 $ health    : Factor w/ 2 levels "1. <=Good","2. >=Very Good": 1 2 1 2 1 2 2 1 2 2 ...
 $ health_ins: Factor w/ 2 levels "1. Yes","2. No": 2 2 1 1 1 1 1 1 1 1 ...
 $ logwage   : num  4.32 4.26 4.88 5.04 4.32 ...
 $ wage      : num  75 70.5 131 154.7 75 ...
```

The following will open a window similar to an Excel spreadsheet:

```
1 View(Wage)
```

The following shows the first 6 rows, but if your data is very wide (i.e. there are a lot of columns) it can be difficult to view

```
1 head(Wage)
```

iClicker

What is n the `Wage` data.

a. 10

b. 11

c. 2999

d. 3000

e. None of the above

iClicker

What is p for the `Wage` data?

a. 10

b. 11

c. 2999

d. 3000

e. None of the above

Matrices

Let $\mathbf{X}$ define an $n \times p$ matrix whose $(i, j)^{\text{th}}$ element is x_{ij}.

$$
\begin{array}{c}
 & \begin{array}{cccc} \text{col } 1 & \text{col } 2 & & \text{col } p \end{array} \\
\begin{array}{c} \text{row } 1 \\ \text{row } 2 \\ \\ \text{row } n \end{array} &
\begin{pmatrix}
x_{11} & x_{12} & \cdots & x_{1p} \\
x_{21} & x_{22} & \cdots & x_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & \cdots & x_{np}
\end{pmatrix}
\end{array}
$$

Some may find it helpful to think of $\mathbf{X}$ as a spreadsheet of numbers with n rows and p columns. N.B. the first index (i) of x_{ij} is the row and second index (j) is the column)

Vectors

- In reference to matrix $\mathbf{X}$ we will either be referencing a row vector, x_i, or a column vector $\mathbf{x}_j$.
- Notice the slight change in little x font here:
- rows vector x_i is curly and not bold (typically an example)
- column vector $\mathbf{x}_j$ is bold and straight (typically a feature)

Row Vector

- We refer to the i^{th}*row* of $\mathbf{X}$ using x_i
- Hence, $\mathbf{X}$ is comprised of the n row vectors $x_1, x_2, \ldots, x_n$ where x_i is a vector of length p.
- Typically, x_i stores all the variable measurements for the ith observation.
- Vectors are by default represented as columns:

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{i2} \end{pmatrix}$$

Visualization of row vector x_i

Rows in R

For example, for the `Wage` data, x_7 is a vector of length 11, consisting of `year`, `age`, `race`, and other values for the 7th individual.

```
1 Wage[7,] # extracts the 7th row
```

Column Vector

- We refer to the j^{th}*column* of $\mathbf{X}$ using $\mathbf{x}_j$
- Hence, $\mathbf{X}$ is comprised of the p column vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ where $\mathbf{x}_j$ is a vector of length n.
- Typically, $\mathbf{x}_j$ stores the measurements of a variable for all of the n observations.

$$\mathbf{x}_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

Visualization of column vector $= \mathbf{x}_j$

Columns in R

For example, for the `Wage` data, $\mathbf{x}_1$ contains the $n = 3000$ values for `year` (that year that wage information was recorded for each worker in our data set).

```
1 Wage[,1] # extracts the first column (year)
```

```
 [1] 2006 2004 2003 2003 2005 2008 2009 2008 2006 2004 2007 2007 2005 2003
[15] 2009 2009 2003 2006 2007 2003 2003 2005 2009 2007 2006 2005 2006 2004
[29] 2008 2003 2004 2005 2008 2005 2006 2003 2006 2004 2006 2003 2004 2007
[43] 2004 2005 2004 2008 2009 2007 2006 2006 2005 2008 2004 2003 2009 2004
[57] 2006 2003 2004 2003 2005 2006 2005 2004 2003 2003 2009 2003 2004 2008
[71] 2004 2006 2006 2006 2009 2003 2003 2006 2003 2008 2005 2004 2009 2003
[85] 2007 2006 2004 2009 2009 2009 2004 2003 2003 2005 2006 2009 2003 2006
[99] 2009 2005 2009 2008 2004 2005 2008 2006 2007 2003 2007 2006 2009 2008
[113] 2004 2006 2003 2007 2009 2006 2008 2005 2006 2009 2006 2008 2009 2005
[127] 2004 2004 2009 2003 2007 2008 2007 2008 2005 2004 2009 2007 2004 2003
[141] 2004 2006 2003 2007 2005 2004 2005 2004 2007 2004 2007 2008 2003 2003
[155] 2003 2008 2005 2004 2003 2005 2007 2008 2008 2007 2003 2007 2003 2004
[169] 2009 2006 2005 2008 2007 2004 2009 2006 2003 2004 2003 2006 2006 2005
[183] 2006 2009 2003 2006 2004 2008 2006 2005 2004 2005 2004 2003 2006 2009
[197] 2008 2006 2005 2004 2006 2006 2006 2003 2007 2007 2007 2006 2005 2007
[211] 2005 2006 2008 2009 2008 2003 2008 2008 2009 2009 2008 2003 2006 2004
```

Matrices revisited

Using the row and column notation just presented, the matrix X can be written:

$$\mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} = \left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p\right)$$

The ^{T} notation denotes the transpose of a matrix or vector, eg $x_i^T = (x_{i1}, x_{i2}, \ldots x_{ip})$.

Output vs. Input Variables

- We use x to denote our *input variable(s)* and y to denote our *output variable*[1]
- For instance, y_i may refer to the `wage` of the ith observation in the `Wage` data set, whose observed features are stored in x_i.
- The collection of all n *observed outcomes* form the vector $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Paired data

Our observed data consists of $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where each x_i is a vector of length p.

Visualization of inputs and outputs in tabular data

Summary

Note there is a potential for ambiguity in the usage of a (as a vector) and a as a scalar

Notation	Description	Representation
n	lower case “n”	number of samples
$\mathbf{x}, \mathbf{y}, \mathbf{a}$	lower-case bold	vectors of length n
x, x_i, a	lower-case normal	vectors of length $\neq n$
$\mathbf{X}, \mathbf{A}_{n \times p}$	capital bold letters	matrix
a	lower-case normal (beginning of alphabet)	scalars
X	capital (end of alphabet)	random variables

Ambiguous cases

In the rare cases the use of lower case normal font leads to ambiguity, clarification will be provided using the following notation:

- $a \exists \mathbf{R}$ if a scalar or $a \in \mathbb{R}$
- $a \exists \mathbf{R}^k$ if a is a vector of length k or $a \in \mathbb{R}^k$

We will indicate an $r \times s$ matrix using:

- $\mathbf{A} \exists \mathbf{R}^{r \times s}$ or $\mathbf{A} \in \mathbb{R}^{r \times s}$ or $\mathbf{A}_{r \times s}$

1. $\mathbf{y}$ only available in the Supervised setting!
To access the **details for this dataset**, type `?Wage` into R.

Statistical Learning

General Model

$$Y = f(X) + \epsilon$$

We assume that data arises from the above formula where:

- $X = (X_1, X_2, \ldots, X_p)$ are **inputs** (also referred to as *predictors*, *features*, *independent variables*, among others)
- Y is the **output** (also referred to as *response*, *dependent variable*, among others)
- ϵ is the **error** term (independent of X and with mean 0)

Use of Capital Letters

$$Y = f(X) + \epsilon$$

- Note that we are using capital letters[1] to denote *random variables* in this context.
- For ease of notation in this section, we use X to denote the input variable(s) which we distinguish using subscripts.
- eg. using the `Wage` data set we might consider the input variables: $X = (X_1, X_2)$ where X_1= `year` and X_2= `age` and response variable Y = `wage`.

Deterministic f

$$Y = f(X) + \epsilon$$

- The function f models the systematic/deterministic part of the relationship between the predictors x and the response y
- f it is in generally unknown.
- Statistical learning is concerned with estimating f.
- Estimates of f are denoted by $\hat{f}$, and $\hat{Y}$ will represent the resulting prediction for Y.

Stochastic ϵ

$$Y = f(X) + \epsilon$$

- ϵ represents the error term or noise, which captures the random variation in y that is not explained by $f(x)$.
- It accounts for the stochastic or unpredictable elements in the response variable, including measurement errors, omitted variables, or inherent randomness.
- It is commonly assumed that ϵ has a mean of zero and is normally distributed with constant variance.

Model Visualization

> All models are wrong but some are useful.

George E. P. Box

Adapted from ISLR

Simulated Bacteria Growth

This is **simulated** data representing the growth of bacterial in a population.

To mimic the real-world, let's pretend we don't know f

Goal of Statistical Learning

- The main objective is to find a function $\hat{f}(x)$ that closely approximates the true underlying function $f(x)$.
- $\hat{f}(x)$ is then used to make predictions about y given new inputs x.

> **f is only known in simulation**
>
> For real (ie. non-simulated data) data, the function f is generally unknown and must estimated based on the observed points.

Simulate using seeds

```r
set.seed(12345)
x <- seq(from = 0, to = 3, by = 0.1)
y <- exp(x) + rnorm(length(x), mean = 0, sd = 1)
errors <- y - exp(x)
```

- When simulating data you should also see a seed (in R using **`set.seed()`**) for your results to be reproducible.
- A **seed** is an initial value used by a random number generator (RNG) to start generating a sequence of random numbers.
- In programming, setting a seed ensures that the same sequence of “random” numbers is produced each time the code is run.

Recall

Reasons for finding f fall into two primary categories:

- **Prediction**: with inputs X available, our concern is predicting the output Y.
- **Inference**: we want to understand the relationship between X and Y.

Often, we will be interested in both, perhaps to varying extent. Our goal will dictate what choices for f will be "better" for us.

Motivated by Prediction

- Motivation for prediction problems often stem from the situation where X is cheap but Y is "expensive".
- Using a complicated *black box* method for f may be preferable if our many focus is prediction since we are not particularly concerned with the exact form of $\hat{f}$.
- Generally speaking, the more complex our algorithm, the harder it will be to interpret.

Motivated by Inference

- Inference aims to answer *how* Y affected by X
- Since $\hat{f}$ is used to model this relationship, we don't want a black box, we want to *understand* its nuts and bolts.

Some related question may include:

1. Which input variables (predictors) are associated with output variables (response)?
1. What is the relationship between the response and each predictor? e.g. positive, negative, linear

Takeaway Message

When choosing the “best” model for a problem, it is important to keep the underlying task in mind. Generally speaking:

- complicated models are often better at prediction but harder to understand;
- simpler models tend to be easier to interpret but will not necessary make accurate predictions.

In many cases a balancing act between the interpretation and accuracy of prediction is needed.

Interpretability vs. accuracy tradeoff

Inspired by Fig 2.7 of ISLR2 (pg 25)

General Workflow

Source: *Introduction to Machine Learning with the Tidyverse*. by Dr. Alison Hill. rstudio::conf2020

Step 1: pick a model

Source: *Introduction to Machine Learning with the Tidyverse*. by Dr. Alison Hill. rstudio::conf2020

Step 2: Train the model

Source: *Introduction to Machine Learning with the Tidyverse*. by Dr. Alison Hill. rstudio::conf2020

Step 3: Test the model

Get New Data

Source: *Introduction to Machine Learning with the Tidyverse*. by Dr. Alison Hill. rstudio::conf2020

Step 3: Test the model

Make predictions

Source: *Introduction to Machine Learning with the Tidyverse*. by Dr. Alison Hill. rstudio::conf2020

Step 4: Assess

Source: *Introduction to Machine Learning with the Tidyverse*. by Dr. Alison Hill. rstudio::conf2020

Step 5: ~~Rinse~~ Repeat

Source: *Introduction to Machine Learning with the Tidyverse*. by Dr. Alison Hill. rstudio::conf2020

Training vs. Testing

- If we do not have access to another “new” data set (as we often don’t), we can divide our data (randomly) into two non-overlapping sets:
  1. The **training set** will be the “old” data used to fit the model
  2. The **testing set** is strictly used to evaluate performance (i.e. the “new data”)

Comments

- A common split would be 80% of data for fitting and 20% for testing; however, the decision is very dependent on the problem and data available.

A word of warning:

- If training set is small, model fit may be poor
- If testing set is small, performance metrics may be unreliable

Workflow Summary

1. We denote *observed* or sampled data by lower-case.