2023WI-MATH-3561H: Lecture #01

Wesley Burr

2023/01/09

Welcome Information

Contact Details

- **Me**: Dr. Wesley Burr
- **Email**: wesleyburr@trentu.ca (only for important, personal issues!)
- **Office**: ENW/GCS 335

Digital Tech & Links

I believe in the power of technology to make teaching and learning easier. So we're going to use quite a bit of it in this class.

- **Blackboard**: official grades, class-wide communications, assignment postings, slides and video links (to Yuja)
- **WeBWorK**: some data - unique data sets for each student
- **Chat ('Teams')**: asking questions, communicating, sharing, talking to each other and me; also class-wide communications
- **RStudio**: learning to **do** statistics and data analysis (10 assignments) - preferably your own computer, but https://sage.trentu.ca/ if not
- **Video Chat**: Zoom for live-streamed workshops, office hours, etc.

WeBWorK

- only used for accessing data - no grading

‘Teams’ Chat Interface

- persistent
- multiple user
- replaces email
- invite link on Blackboard

RStudio

The **R** programming language and interface is **the** language of statistics in the 21st century.

- You will be learning (more about how) to do **data analysis** using **R** in this class
- Many ways to use RStudio - recommend you bite the bullet and install it on your own computer if at all possible. If not, our JupyterHub server is an option. I may provide some extra computational resources for folks to use later in the term, if I can find a way to make them available easily.

Course Overview

Posted Material

- lectures: 1 video per week, posted (hopefully) on Friday afternoons **before** the upcoming week
- workshops: one topic per week; two in-person slots you can attend (only attend one!); 1 recorded video (from parts of the in-person time); organized sequentially (numbered 1-12)
- extra topics: as often as needed, with announcements

Texts: First Textbook

The first textbook we are using is a Springer book, and is considered **the** standard introductory reference for statistical learning algorithms. Thankfully, while you can buy a physical copy for $94 on Amazon, there's a completely free PDF available from the authors' website. So if you want a physical copy, have at it; I own one! Otherwise, save your money.

There is a link to the PDF on Blackboard.

Texts: Second Textbook

The second textbook we are using is a CRC Press ‘red cover’ statistics book called **Statistical Rethinking**. It’s a book focused on Bayesian methods, written to be accessible to people without as much mathematics as would traditionally be required. It’s very conversational, and has tons of worked code to demonstrate the techniques. We will use it for the latter half of the course. You do need a copy, and it’s not free. Links on Blackboard.

Note: there are PDFs floating around of this book. Make sure to get the 2nd edition if you are obtaining a copy this way.

Note 2: the Kindle edition from Amazon is $10?! So that’s cheap . . .

This book should have been stocked by the bookstore, but is pricy there.

Texts: Reference Text

I may make reference to **The Book of R** periodically, which is a book you should all own from 1051H/1052H/2560H. If you're new to this course and somehow don't own it, consider picking up a copy. Details on Blackboard.

Things Worth Marks

- Assignments: 10 in total, in weeks 2-6 and 8-12, 10% each.

WeBWorK (0%)

WeBWorK is an open-source homework system with automatically graded problems. It allows for some fun things like multiple attempts, and in-response math (e.g., you can say “My Answer is [2 * 2 + 2]” and it will recognize it).

- We will only use it to generate unique data sets for each student, so that your long-form assignments can be a bit more interesting (and force you to think a bit more about your personal results!).

R Assignments (100%, 10x10%)

The R assignments are designed to assess your learning of the material covered mostly in the workshops, and demonstrated in class. Each one will examine an algorithm, or a family of algorithms, and analyze some ‘real’ data (more on this later). The reports will be written in R Markdown, and turned in as a PDF. These will be the only deliverables for the course - 10 worked assignments.

The first attempt assignments will be due on the Mondays of Weeks 3, 4, 5, 6, and the day after Reading Week ends (Monday of Week 7); plus Weeks 9, 10, 11, 12 and the first Monday of exams.

Grading and Mastery

For most of you, you've never taken one of my courses where I use a Standards-based or Mastery grading scheme. There are details in the syllabus, but here's the basic overview:

1. First grades are not final. Your assignments will be graded as 0, 4, 8 or 10 out of 10.
2. If you wish to challenge this grade, you can review the feedback, fix mistakes, and then come to a special office hour where you will defend your fixes, and discuss how you went wrong, and why. **Max of +2 to grade from this**.
3. Alternatively, you can re-do the assignment from scratch, with major changes to your work to fix fundamental flaws, and simply re-submit the assignment for complete re-grading. **Can improve by any amount**.

More on Grading

There is a limit to the system: you must challenge or resubmit within 2 weeks of receiving the grades back, or your first grade stands. No letting things slide until the end of term and then resubmitting everything.

Note: you must submit an assignment in the first round in order to then be able to re-do the assignment. This **can** be a mostly blank page saying "I was too busy to get this in, I'm taking the 0, and I'll re-do it soon."

In addition, if you choose the re-do option, **and** I've done a set of data which is not common across the class, I will make you use a fresh data set for your work. If it was a common data set, then you just fix all your mistakes and re-do the assignment.

How to Get Help

- Chat: anytime, asynchronous - I'll check in daily, Monday-Friday
- read the textbooks
- Google is surprisingly helpful for learning R stuff - there's a huge wealth of materials out there for beginners, some of which we will link to through the term
- lots of data science / stat learning blogs out there - **warning** can be hard to wade through the cruft if you don't have experience

Statistical Learning

What IS Statistical Learning?

Statistical learning refers to a set of tools for **understanding data**. These tools can be classified as:

- **supervised**: involves building a statistical model for predicting, or estimating, an output based on one or more inputs.
- **unsupervised**: there are inputs but no supervising output; nevertheless we can learn relationships and structure from such data.

In a previous course, you will have seen classes of **linear models**, often just called *regression* or *multiple regression*, when the inputs and outputs are numeric. You may also have seen **ANOVA**, which is a specific class of linear model where the inputs are categorical. Both of these fall under the classification of **supervised statistical learning**.

Some Language

Almost all models we will cover this semester share some common language.

- **input variable**: also called predictors, *independent* variables, features, or just variables
    - typically denoted by X
    - if you have more than one, X_1, X_2, through X_p
- **output variable**: also called the response, or *dependent* variable
    - typically denoted by Y

Some Notation

We will do our best to keep the mathematical notation to a minimum, but sometimes you'll want to read the textbook, and puzzle out what it is trying to say. If we assume that we observe some Y which is quantitative, and have p different predictors, $X_1, X_2, \ldots, X_p$, then what we are interested in is a **relationship** between these.

Generically, we might write:

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

where the unknown function f represents the relationship between the X predictors and the response Y. We assume there is some error in the **observations**, so that we cannot capture $f(\cdot)$ perfectly: this is ϵ.

Bringing it back to something you know

Multiple linear regression is something you're all assumed to have covered in a previous class. In that case, the unknown function is just summation:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon,$$

such that Y is related to the Xs via a linear combination, the sum of each of the Xs multiplied by a scale factor β_i, for $i = 1, 2, \cdots, p$.

Note: you will never have to do mathematical derivations in your assignments. Each of these Xs corresponds to a vector of observations in `R`, as does the Y.

Example, R

$$Y = 0.5 + 1.0X_1 + 2.5X_2 - 1.3X_3 + \epsilon$$

```
x <- data.frame(x1 = rexp(20),
                x2 = rexp(20) + 2,
                x3 = rexp(20) - 1)
y <- 0.5 + 1.0 * x$x1 + 2.5 * x$x2 - 1.3 * x$x3 + rnorm(20)
dat <- data.frame(y, x)
mod <- lm(y ~ x1 + x2 + x3, data = dat)
```

Example, R (ctd.)

$$Y = 0.5 + 1.0X_1 + 2.5X_2 - 1.3X_3 + \epsilon$$

```
mod$coefficients
```

```
## (Intercept)          x1          x2          x3
##    1.281539    1.244548    2.221301   -1.335892
```

So we generated some fake data, built a relationship between X and Y, then asked a **supervised learning** algorithm to estimate the relationship. You can see that we **estimated** the coefficients here, in the `R` code provided.

- $\beta_0 = 0.5$, we estimated 1.282
- $\beta_0 = 1.0$, we estimated 1.245
- $\beta_0 = 2.5$, we estimated 2.221
- $\beta_0 = -1.3$, we estimated -1.336

Estimating

Why Estimate f?

We would we **want** to estimate f?

There are two main reasons: for **prediction**, and for **inference**.

Prediction

In many situations (e.g., biology, ecology, physics, etc. - SCIENCE!), a set of possible inputs are easily obtained and measured. However, there are often many outputs we may be interested in which are **not** easily measured or obtained.

Thus, **prediction** of Y is often quite valuable, and something we want to do.

Notation for Prediction

We use “hat” notation to refer to predictors and prediction. So, we would say that our prediction of Y is $\hat{Y}$:

$$\hat{Y} = \hat{f}(X),$$

where $\hat{f}$ is our prediction (“estimate”) of the unknown function f that links the inputs X to the output Y.

Examples of Settings for Prediction

- blood chemistry & patient risk
- rental price for an apartment with specific characteristics ("market value")
- grade you should have gotten on your final exam, if you'd had the chance to write it (aegrotot standing)
- potential income based on education, seniority, field of work
- fitness of a species under constraints in the environment

Question of Accuracy

Remember the context: we're estimating an unknown function. And *estimating* could be rephrased as *educated guessing*. So ... technically speaking, **any** guess is an estimate! And not necessarily a **good** one!

Example: the "stopped clock" estimator. If you ask me what time it is, and I always respond 11 o'clock, I will almost always be wrong. But it **is** an estimate, and it **will** be correct ... sometimes. Namely, at 11am and 11pm!

Errors

There will be errors in your prediction, and they can be divided, as the book discusses, into **reducible error** and **irreducible error**.

- Reducible error: use the most appropriate method or methods; improve your estimate as much as is possible
- Irreducible error: no matter what you do, you can't avoid the existence of ϵ, so you can never recover the information perfectly

General Point

Because this is a survey course of statistical learning methods, we will cover a lot of different approaches. Each approach/algorithm has its strengths, and its weaknesses. Always consider more than one factor when being told something is “the best at X” - there’s almost always more to the story.

For example, certain Machine Learning algorithms are very strong at prediction, as measured by the accuracy of the method on a test set (a strength). However, once you go outside the realm of the test/train data, the methods can perform very poorly. This is a weakness. It does not mean they are not useful, simply that they are limited, as all things are.

Our Objective for Prediction in General

Our objective, any time we are predicting things, is to reduce the Reducible Error as much as is feasible. That means we will have the best prediction we can, under the constraints of the data, and our available skills.

This course will hopefully broaden your skill-base so that you have access to significantly more skills, which should make your scientific predictions significantly more capable.

A Small R Example

Using the data from before, X_1, X_2, X_3 which are linearly combined (with noise) to make Y. Let's try to predict the value of Y at a particular combination of Xs.

```
predict(mod, newdata = data.frame(x1 = 2, x2 = 1, x3 = -1))
```

```
##        1
## 7.327827
```

What is the true, average value?

$$\begin{aligned} Y = f(2, 1, -1) &= 0.5 + 1.0(2) + 2.5(1) - 1.3(-1) \\ &= 6.3 \end{aligned}$$

Since we know we used the “right model” (multiple linear regression is the perfect fit for this data), the difference between the true value of 6.3 and our predicted value of 7.33 is **irreducible error**, due to the ϵ.

Inference

The other angle for estimation is to try to understand the association between Y and X. In this case, we specifically care about the form that f takes on, not only predicting $\hat{Y}$ using our guess $\hat{f}$.

In these cases, $\hat{f}$ **cannot** be treated as a black box, because we need to know its form, and be able to understand **how it works**

In most scientific endeavours, this is what we are aiming to do.

Inference

What are some of the questions we might ask about the function $f(\cdot)$?

- which predictors are associated with the response?
- what is the relationship between the response and each predictor?
- can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?

How Do We Do It?

Estimating f, the goal of inference, is **hard**. There are many methods available that do some aspect of this, but all are subject to **constraints**.

This kind of problem is sometimes referred to as **underdetermined** - this means there just isn't enough information available to estimate $f(\cdot)$ without making assumptions and simplifications.

What Kinds of Approaches?

Primarily, we divide our approaches to estimating $f(\cdot)$ into two classes: **parametric** and **non-parametric**.

Both approaches obtain, from the available data, a **training data** set. We use this data to **train** our method on how to estimate $f(\cdot)$. In doing this, we apply the method to find an educated guess $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y).

Parametric Methods

These are what you think of when you think of statistical methods, especially the family of linear models.

1. Make an assumption about the form of $f(\cdot)$, e.g., assume f is linear in X - this is a *linear model*, of which regression is one example. For the linear option, this is a tremendous simplification.
2. Use a procedure that takes the training data, and trains (or "fits") the model. Typically, we use software for this.

The **parametric** part of the name comes from the process whereby we reduce the problem from estimating f down to only estimating a set of parameters.

Non-Parametric Methods

Non-parametric methods do not make explicit assumptions about the functional form of $f(\cdot)$. They still have some sort of framework in place (remember: it's impossible to estimate f without something!), but instead of assuming things about the functional form, they seek an estimate of f that is as close to the data as possible without being either too wiggly or too flat.

The typical restriction we require for non-parametric fits to data is that the resulting fit (estimate) be "smooth". This is somewhat vague, and deliberately so. We'll come back to this topic in a few weeks and introduce some methods with details.

How to Choose . . .

This semester, we will be looking at a variety of methods. Some are parametric, some are non-parametric. Some are highly restrictive and have strong assumptions; others are very free, and have only weak or limited assumptions. How do you decide what is appropriate for your data?

Trade-Offs: Accuracy and Interpretability

There are two, or maybe three, considerations when choosing a method.

1. Accuracy (or Flexibility): how restrictive are the assumptions to the possible outputs of the model? Can the chosen model represent a wide variety of possible relationships? (High accuracy/flexibility: deep learning, neural networks, SVMs; Low accuracy/flexibility: Lasso, least squares, GAMS)

Summarized as "how accurately can the model reproduce the given data?".

2. Interpretability: understanding the structure and implications of the model framework is important for inference and scientific understanding. How well can the model fit be interpreted to give scientific information? (High interpretability: Lasso, least squares; Low interpretability: deep learning, neural networks)

Summarized as "is there a direct interpretation of the coefficients of your model which can be translated into scientific language?"

Trade-Offs

When **prediction** is your goal, and you don't actually care about the interpretability of the model itself, complex and uninterpretable models such as neural nets can be highly powerful.

When **inference** is your goal, you do care about interpretability of your model, so less-complex and more inflexible statistical models have advantages.

We will spend most of this course talking about these trade-offs, especially once we have a few models developed and can discuss how and why you might want to use them.

Assessing Model Accuracy

Evaluate Performance & Make Decisions

If we want to compare some methods, one to another, or even evaluate the performance of a single method, we need **metrics**: ways of measuring how well predictions actually match observed data.

Formally, we want to **quantify** how close the predicted response value for a given observation is to the actual, true, response value for that observation.

In math, we want to somehow measure $y - \hat{y}$: the difference between the true y and the predicted $\hat{y}$.

Choosing a Metric

We will talk about this a lot, actually - the way of choosing how to measure that difference varies tremendously across methods. Choosing a different way of measuring it often gives an entirely different method! Some people's careers have been made on coming up with a clever choice . . .

A Metric You've Seen Before

One of, if not the, oldest methods available is the **squared error**. We want our metric to quantify how close, overall - that is, across the n data points you have. So if you have some predictions that are + and some that are -, they can balance out, and indicate the difference is 0, when it is not!

So, instead, we square the differences. This leads to **mean squared error** (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

This is the **mean** (average) across the i data points of the squared difference between y_i and $\hat{f}(x_i) = \hat{y}_i$.

You should remember this: this is what is used to give us regression!

How to Apply This?

There is a key difference between how well a model works on training data (data you've "seen before"), and how well a model works on test data ("new data"). We don't usually care **that** much about how well a model fits to training data, because ... we have all of that data! Instead, what we care about is how well a model fits to data we've never seen, because predicting those situations is why we modeled in the first place.

In practice: you may not have test data. There are ways around this. We can split the data we **do** have into two sets, train on the "training" one, and then compute the MSE or similar metrics on the "test" one. This is the traditional approach in machine learning. Or, we can get fancy ... more on this later in the term!

Course Set Up for Next Week

First Major Topic

Our first major topic will be a revisit of linear models, with only a small amount of new material. We will consider simple linear regression, multiple linear regression, and the concept of transformation of variables. It is intended to be a refresher, and give you two weeks to get your feet under you and ready to learn new ideas.

Readings

The material of this first week is Chapter 2 of ISLR, inclusive. Some parts were skipped or only skimmed (esp. 2.1.5, 2.2.2 and 2.2.3), and will be covered "as needed" (i.e., in a couple of weeks). Chapter 1 was also mentioned in part, but if you read it, just skim and look for high points. The mathematical notation at the end of Chapter 1 is important if you want to understand the equations we will use, although that isn't strictly required for success.

The material of the second week, and the first major topic, is Chapter 3, sections 3.1-3.3.3 inclusive, if you would like to (and I recommend it!) read ahead on the material.

Workshops

We will be working through a refresher on R and R Markdown, and ensuring everyone is set up to be able to do their work in a good environment during the Tuesday/Wednesday workshops this week. We will use some data from ISLR that was mentioned in Chapters 1 and 2, and recreate some of the figures!

If you want to work ahead on this, and are on your own computer, there's a technical note on Blackboard indicating the software environment you will need to be able to complete the first half of the course.