Political Science 210: Introduction to Empirical Methods

Week 8: Machine Learning

The point of machine learning

When studying a given process with independent variables X and an output Y, we might run into cases where:

- The outcome Y is hard to measure
    - It might have not happened yet!
- We have a large set of X variables and their relationship to each other or to the outcome is not clear
    - Typical when “big data” is involved – we’re saturated with possible independent variables

Machine learning gives us tools to either *predict* the outcome or to make *inferences* about variables in large datasets.

Example of a machine learning problem

Let’s say we want to know which students will get an “A” in PS 210 in future quarters of this class.

How would we know which students will get an “A” in future quarters?

- By studying students’ grades in this (or previous) quarters
  - Dependent variable: Student’s grade
  - Independent variables?
    - Amount of time student usually studies
    - Student’s class year (freshman, sophomore, junior, senior)
    - Average grade in previous math courses

Setting up the data

Let’s say we were able to collect data on the students from the current quarter, especially on the dependent variable (final grade) and independent variables of interest:

Spring 2023 class

Student	Grade	Class year	Study hrs.	Prior grades
Rashad	3.3	3	3	3.3
Jimena	3.7	1	5	4.0
Bill	2.5	4	0	3.0
Cynthia	4.0	2	7	3.7
Kevin	3.3	1	4	2.7
…	…	…	…	…

Generating a model

If we want to find the relationship between class grades and the dependent variable, we might take a *parametric* approach and assume there's a model:

$$Y_{\text{grade}} = a + b_1 X_{\text{class year}} + b_2 X_{\text{study hrs.}} + b_3 X_{\text{prior grades}}$$

We'll take OLS regression as a simple example of an *algorithm* and use it on a large-N sample of students (from Week 5) to find values of the beta coefficients:

$$Y_{\text{grade}} = (2.0) + (0.3) X_{\text{class year}} + (0.1) X_{\text{study hrs.}} + (0.1) X_{\text{prior grades}}$$

We can now use this model to *predict* the grades of the next class of students.

Predicting new outcomes

Based on what we learned from Spring 2023, we can apply it to incoming students in Fall 2023:

$$Y_{\text{grade}} = (2.0) + (0.3)X_{\text{class year}} + (0.1)X_{\text{study hrs.}} + (0.1)X_{\text{prior grades.}}$$

Fall 2023 class

Student	Grade (predicted)	Class year	Study hrs.	Prior grades
Carla	?	4	1	3.3
Malcom	?	1	7	2.7
Julio	?	2	9	3.0

Predicting new outcomes

Based on what we learned from Spring 2023, we can apply it to incoming students in Fall 2023:

$$Y_{\text{grade}} = (2.0) + (0.3)X_{\text{class year}} + (0.1)X_{\text{study hrs.}} + (0.1)X_{\text{first gen.}}$$

Fall 2023 class

Student	Grade (predicted)	Class year	Study hrs.	Prior grades
Carla	?	4	1	3.3
Malcom	?	1	7	2.7
Julio	?	2	9	3.0

Predicting new outcomes

Based on what we learned from Spring 2023, we can apply it to incoming students in Fall 2023:

$$Y_{grade} = (2.0) + (0.3)(4) + (0.1)(1) + (0.1)(3.3)_{.}$$

Fall 2023 class

Student	**Grade (predicted)**	**Class year**	**Study hrs.**	**Prior grades**
Carla	?	4	1	3.3
Malcom	?	1	7	2.7
Julio	?	2	9	3.0

Predicting new outcomes

Based on what we learned from Spring 2023, we can apply it to incoming students in Fall 2023:

$$Y_{grade} = 3.6$$

Fall 2023 class

Student	Grade (predicted)	Class year	Study hrs.	Prior grades
Carla	3.6	4	1	3.3
Malcom	?	1	7	2.7
Julio	?	2	9	3.0

Predicting new outcomes

Based on what we learned from Spring 2023, we can apply it to incoming students in Fall 2023:

$$Y_{\text{grade}} = (2.0) + (0.3)X_{\text{class year}} + (0.1)X_{\text{study hrs.}} + (0.1)X_{\text{prior grades}}$$

Fall 2023 class

Student	Grade (predicted)	Class year	Study hrs.	Prior grades
Carla	3.6	4	1	3.3
Malcom	?	1	7	2.7
Julio	?	2	9	3.0

Predicting new outcomes

Based on what we learned from Spring 2023, we can apply it to incoming students in Fall 2023:

$$Y_{grade} = 3.3.$$

Fall 2023 class

Student	Grade (predicted)	Class year	Study hrs.	Prior grades
Carla	3.6	4	1	3.3
Malcom	3.3	1	7	2.7
Julio	?	2	9	3.0

Predicting new outcomes

Based on what we learned from Spring 2023, we can apply it to incoming students in Fall 2023:

$$Y_{\text{grade}} = (2.0) + (0.3)X_{\text{class year}} + (0.1)X_{\text{study hrs.}} + (0.1)X_{\text{prior grades}}$$

<u>Fall 2023 class</u>

Student	Grade (predicted)	Class year	Study hrs.	Prior grades
Carla	3.6	4	1	3.3
Malcom	3.3	1	7	2.7
Julio	?	2	9	3.0

Predicting new outcomes

Based on what we learned from Spring 2023, we can apply it to incoming students in Fall 2023:

$$Y_{\text{grade}} = 3.8_{.}$$

Fall 2023 class

Student	Grade (predicted)	Class year	Study hrs.	Prior grades
Carla	3.6	4	1	3.3
Malcom	3.3	1	7	2.7
Julio	3.8	2	9	3.0

Validating our predictions

So far, we’ve been predicting outcomes that we can’t measure (in this case, because they haven’t happened yet).

But how do we know how well our model works? What degree of error should we expect? Are there different models that might work better?

We need a way to *validate* our model: To check how accurate it is.

Supervised learning

To test the strength of a model, we can take the data where the outcomes are known (that is, the Spring 2023 students) and *randomly* separate it into two groups:

Supervised learning

To test the strength of a model, we can take the data where the outcomes are known (that is, the Spring 2023 students) and *randomly* separate it into two groups:

Supervised learning

To test the strength of a model, we can take the data where the outcomes are known (that is, the Spring 2023 students) and *randomly* separate it into two groups:

Training group

Testing group

Supervised learning

To test the strength of a model, we can take the data where the outcomes are known (that is, the Spring 2023 students) and *randomly* separate it into two groups:

…we can fit a model to the *training group…*

$$Y_{\text{grade}} = (2.0) + (0.3)X_{\text{class year}} + (0.1)X_{\text{study hrs.}} + (0.1)X_{\text{prior grades.}}$$

Supervised learning

To test the strength of a model, we can the the data where the outcomes are known (that is, the Spring 2023 students) and *randomly* separate it into two groups:

…we can fit a model to the *training group…*

…and then use it to “predict” the grades of the *testing group*.

= 3.3

= 4.0

= 3.7

= 3.3

= 2.7

= 3.3

= 3.0

= 4.0

Supervised learning

Since we *already know* the outcome of the testing group, we can the predicted outcome with the actual outcome.

Spring 2023 class (testing set)

Student	**Predicted grade**	**Actual grade**
Rashad	3.0	3.3
Cynthia	4.0	4.0
Kevin	3.7	3.3

If we aren’t happy with its accuracy, we might go back to the training data and try different predictor variables, different models, different methods, etc.

Overfitting

We might also get an algorithm that’s TOO accurate!

If the algorithm predicts the testing data too well, it might be picking up the random error that’s specific to the quirks of the testing data but that’s less useful in predicting future cases.

We want to find an algorithm that fits “just right:” It predicts the testing data reliably, but not so reliably that it’s less useful in new cases.

Supervised vs unsupervised

In the earlier example, we tested our data using a *supervised* approach: We took a dataset where we want to know a specific outcome (grade) used that outcome to train and test a model.

In an *unsupervised* approach, we don’t have any specific outcome in mind. Instead, we typically want to classify the data into groups of similar cases.

- Classification might be useful if we don’t know much about the data’s structure
  - e.g. topic modeling: How many latent topics are there a dataset of news articles?
  - Once the algorithm is done classifying, we might make qualitative judgments about what each of the groups represent.
- Example methods include clustering, neural networks
  - Methods sound fancy and can be computationally complex, but their building blocks are often just regression techniques like OLS, which you learned about in Week 5.

Unsupervised learning (text-as-data example)

Let’s say we had a dataset of US presidential speeches that we wanted to classify. We can run an algorithm on the full dataset and group speeches based on words that are most likely to appear in a given set of speeches (and least likely to appear in other speeches). How would you label these groups/topics?

- Topic 1: Jobs, taxes, middle class, inflation
- Topic 2: Security, sanctions, negotiation, conflict
- Topic 3: Waste, spending, inefficient, government, cut
- Topic 4: Party, divide, side, common, aisle

Machine learning in social science

Advantages

- Can make better estimates when certain variables are hard to measure
  - Can "predict" certain variables when data is missing or can't be measured – better than nothing!
- Can help identify patterns in the data that we don't know about
  - Especially useful for "wide" datasets with lots of independent variables
  - Especially useful when we don't have strong assumptions about how independent variables lead to the outcome
- Can save a lot of time
  - Text-as-data methods can code a large set of documents much faster than reading them manually
- Can create better validation for classification methods
  - Rules for classifying groups will be consistent, less prone to human bias or error

Machine learning in social science

Disadvantages:

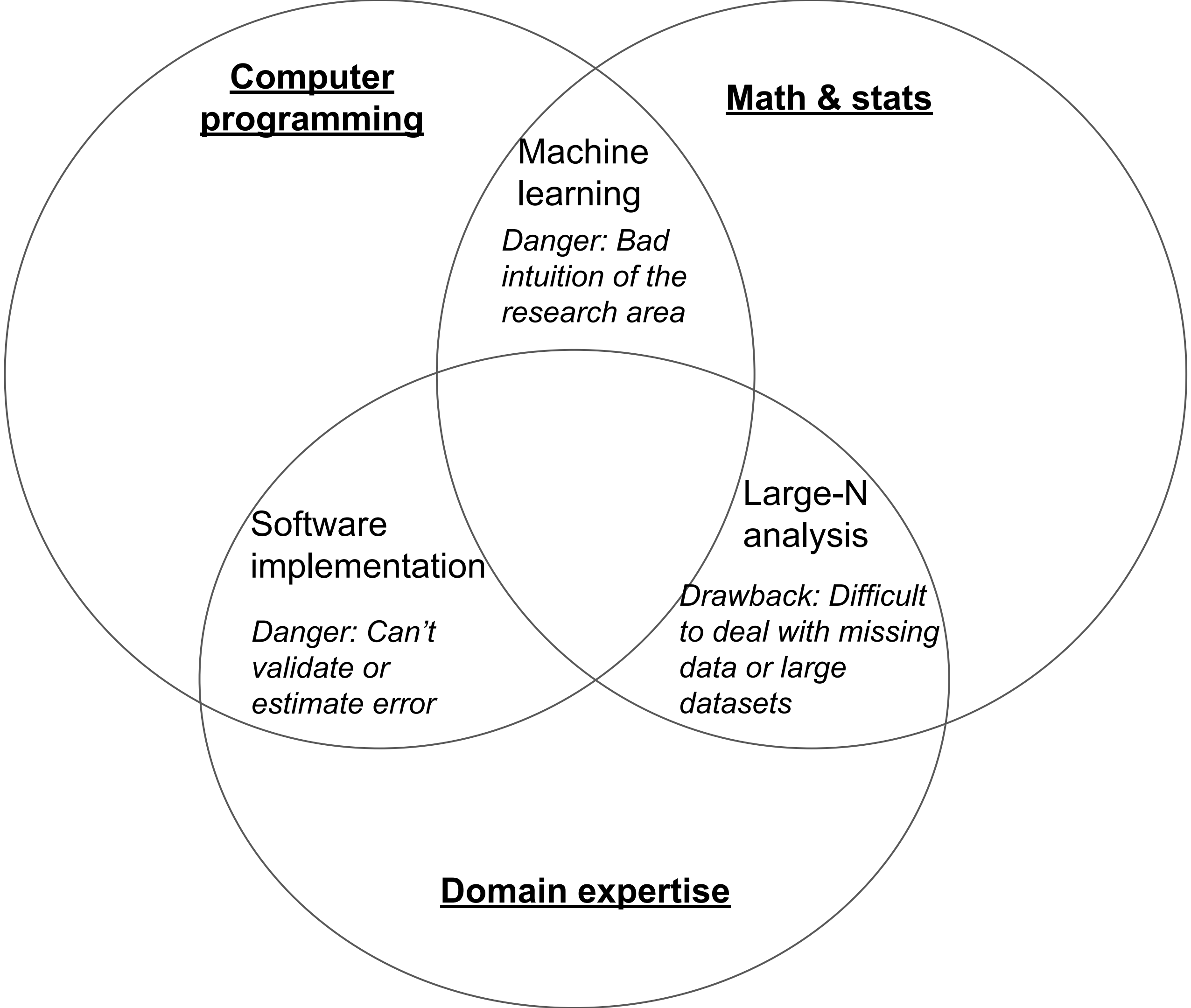
- Can’t provide causal inference
  - Causal mechanism is unclear: How or why do the variables produce the outcome?
  - Independent variables in might just be correlated with something else outside the available data that causes the outcome.
- More complex algorithms are difficult to interpret
  - The “black box” problem: Which variables are actually important in leading to the outcome? Which are just improving our predictive power?
- Training datasets might be biased
  - If conditions change over time or if new cases are from a different population than our training dataset, then predictions or classifications might be biased.
  - E.g. if we’re predicting the grades of incoming students based on previous grades, will it matter if some cohorts spent more time in COVID quarantine in high school than others?

What is machine learning, anyway?

The building blocks of machine learning algorithms are usually methods that are already common in statistics and mathematics (like in our large-N unit)…

…and ML can’t give us a *qualitative* understanding of the variables: How they work, what causes them to react to other variables (like in our small-N unit)...

…but it can at least help us address complex datasets and make inferences about missing or unclear variables based on their patterns.



Computer programming

Math & stats

Machine learning

Danger: Bad intuition of the research area

Software implementation

Danger: Can’t validate or estimate error

Large-N analysis

Drawback: Difficult to deal with missing data or large datasets

Domain expertise