An Introduction to Quantifying Text in Political Communication

Guest Lecture at Colorado State University, Spring 2022

Damon C. Roberts, PhD Candidate

March 24, 2022

University of Colorado Boulder Department of Political Science

Outline

1. Introduction
2. Basics of machine learning
3. Overview of models
4. A taxonomy
5. Takeaways

Introduction

Introduction

- Quantifying text is becoming a larger part of political communication research
- Though it uses "fancy" methods, this does not mean that you escape the common challenges of mapping concept to measurement (e.g. does asking someone if they are liberal mean we are measuring how liberal someone is?)
- These methods use a lot of jargon and sound really complex. At the end of the day, these are just extensions of familiar models like OLS and Logit. They are simply just specialized versions of these familiar models.

Basics of machine learning

Basics of machine learning

Big Picture

1000 ft. view

- Structural models versus predictive models

1000 ft. view

- Structural models versus predictive models
  - Structural models: these are interested in identifying the right estimate. We are trying to find a coefficient with the least amount of bias. That is, if we are trying to explain if X causes Y, then we are trying to find a $\hat{\beta X}$ where $\beta X - \hat{\beta X} = 0$. This is what we often do in political science.

1000 ft. view

- Structural models versus predictive models
  - Structural models: these are interested in identifying the right estimate. We are trying to find a coefficient with the least amount of bias. That is, if we are trying to explain if X causes Y, then we are trying to find a $\hat{\beta}X$ where $\beta X - \hat{\beta}X = 0$. This is what we often do in political science.
  - Predictive models: these are interested in identifying the right out-of-sample predictions. We are not as interested in correctly explaining whether and how much X explains Y, but we include X to come up with a prediction of Y. This is more common in fields like finance and computer science. Also common in election prediction.
  - Machine learning models fit in this predictive model world.

Basics of machine learning

What is it used for?

Machine Learning

- What are some phrases you often hear to colloquially refer to machine learning?

Machine Learning

- What are some phrases you often hear to colloquially refer to machine learning?
- What are the two goals of machine learning?

Machine Learning

- What are some phrases you often hear to colloquially refer to machine learning?
- What are the two goals of machine learning?
  - prediction
  - classification
  - Both prediction and classification are pretty intertwined

Basics of machine learning

Sorting Beans

Demo as we go through

- Link to Jupyter Notebook for Demo
- If you want to see the demo in the future, contact me and I'll convert it and its outputs to a PDF document. If you want to use any of the code, you can just open the file and copy and paste the code.

Let's think of sorting beans

- Causal question example: Does bean size predict whether it will be included in the slowcooker?

Let's think of sorting beans

- Causal question example: Does bean size predict whether it will be included in the slowcooker?
  - We can start with a simple regression:
- What will the regression tell us? What is the fundamental thing we are trying to address?

Let's think of sorting beans

- Causal question example: Does bean size effect whether the bean will taste good?
  - We can start with a simple regression

$$\hat{Y}_i = \alpha_i + \hat{\beta}_{1i} Size + \epsilon_i \quad \text{(Eq. 1)}$$

Let's think of sorting beans

- Causal question example: Does bean size effect whether the bean will taste good?
  - We can start with a simple regression

$$\hat{Y}_i = \alpha_i + \hat{\beta_{1i}} Size + \epsilon_i \quad \text{(Eq. 1)}$$

  - But we might think there is some other factor related to both size and whether it will taste good. So we need to control for it, right? Our estimate of $\hat{\beta_{1i}}$ in Eq. 1 might be really explained by something like whether the bean is the appropriate color. Color and size can tell us whether the bean is good or bad but size may be less important than color. But both may affect whether we want to throw it in the slowcooker.

Let's think of sorting beans

- Causal question example: Does bean size effect whether the bean will taste good??
  - We can start with a simple regression

$$\hat{Y}_i = \alpha_i + \hat{\beta}_{1i} Size + \epsilon_i \quad \text{(Eq. 1)}$$

  - But we might think there is some other factor related to both size and whether it will be included in the slowcooker. So we need to control for it, right? Our estimate of $\hat{\beta}_{1i}$ in Eq. 1 might be really explained by something like whether the bean is the appropriate color. Color and size can tell us whether the bean is good or bad but size may be less important than color. But both may affect whether we want to throw it in the slowcooker.

$$\hat{Y}_i = \alpha_i + \hat{\beta}_{1i} \text{Size} + \hat{\beta}_{2i} \text{Color} + \epsilon_i \quad \text{(Eq. 2)}$$

Let's think of sorting beans

- What will the regression tell us? What is the fundamental thing we are trying to address?

Let's think of sorting beans

- Predictive question example: What would the average bean in our slowcooker look like when we are done sorting out the ones we might expect to taste bad?

Let's think of sorting beans

- Predictive question example: What would the average bean in our slowcooker look like when we are done sorting out the ones we might expect to taste bad?
  - Are we interested in explaining the same thing as the causal question?
  - What sorts of information about the beans will we want to include in a model where we are predicting what sorts of beans will lead to the best tasting meal?
  - Ultimately, what is the output here?

Let's think of sorting beans

- Predictive question example: What would the average bean in our slowcooker look like when we are done sorting out the ones we might expect to taste bad?
  - Are we interested in explaining the same thing as the causal question?
  - What sorts of information about the beans will we want to include in a model where we are predicting what sorts of beans will lead to the best tasting meal?
  - Ultimately, what is the output here?
    - Good tasting vs. bad tasting beans, right?

The supervised cook and the unsupervised cook

- The supervised cook: You can have the chef tell you: here are the qualities of a good-tasting bean and here are the qualities of a poor tasting bean. You take a comprehensive list. You realize when you are sorting that some beans have some good qualities but also some poor qualities. But you do what you think is best. Your chef comes by and tells you, okay, well I want you to take some of these features more seriously than others. So you do that. You end up making a pretty solid pot of beans.

The supervised cook and the unsupervised cook i

- The unsupervised cook: The chef tells you:
  - I want you to make a pot of beans with only the good tasting beans. So, what do you do?
  - Start trying to group beans by different features things like color, size, plumpness, etc.
  - Now say You realize you have multiple species of beans and some species of beans taste good but might be smaller than other species. So you taste different beans. This makes the process much harder. So, the only thing you can do: You start to pick up patterns.

The supervised cook and the unsupervised cook ii

- You start to cluster beans based on these patterns. Like Pinto beans that are large and plump with a light color. So you start to put all the pinto beans with similar features in one pile. You then do that for other things. Then you end up with a bunch of piles of beans.
- Then the chef comes by and grabs the piles of beans that have features that the chef knows often is an indication of a good tasting bean

The supervised cook and the unsupervised cook

- These are two very different ways that the chef taught you to make a good pot of beans, right? Do you avoid doing work in either of the scenarios?

Overview of models

Supervised and Unsupervised Models for classification

- Supervised Models:

Supervised and Unsupervised Models for classification

- Supervised Models:
  - You do some amount of classifying of text yourself, but can only do so much. So you then "train" a model to learn about patterns in the text you already classified yourself so that you can generalize it to more texts than you have time to classify by yourself.

Supervised and Unsupervised Models for classification

- Supervised Models:
  - You do some amount of classifying of text yourself, but can only do so much. So you then "train" a model to learn about patterns in the text you already classified yourself so that you can generalize it to more texts than you have time to classify by yourself.
- Unsupervised models:

Supervised and Unsupervised Models for classification

- Supervised Models:
  - You do some amount of classifying of text yourself, but can only do so much. So you then "train" a model to learn about patterns in the text you already classified yourself so that you can generalize it to more texts than you have time to classify by yourself.
- Unsupervised models:
  - You ask the model to identify patterns for you. Machines are dumb, though. [1] You have to interpret what the patterns mean yourself.

[1] There are models where you can incorporate variables to help the model identify relevant patterns you are interested in. These are often referred to as structural or hierarchical models. Not to be mistaken with structural and hierarchical models in regression.

Overview of models

Applied Examples

Examples of supervised machine learning with text data

- Dictionaries: Make a list of words and find other words that might fall into similar categories. For example, make a list of masculine, feminine and neutral words in politics (Roberts and Utych, 2021). Then expand that list using text from interviews with members of Congress (Roberts, Utych, and Siegel, 2021).
- Similarity: Classify text based on their similarity to one another. For example: Author attribution to the Federalist papers.

Examples of unsupervised machine learning models with text data

- Clustering: Uses things like Principal Component Analysis and other fancy clustering approaches to optimize clusters which represent similar documents from dissimilar documents. For example, identifying a document as a legal document versus a novel.
- Topic models: Recognizes that texts can be similar in some ways and dissimilar in others. Common ones use what is called "mixed membership models" which basically mean that documents can have a number of "topics" covered within them. So Document A can be in the same cluster as Document B in some dimension but different in some other dimension. For example, examining transcripts from federal court confirmation hearings in the Senate to study whether different topics are discussed during the confirmation hearings for female and nominees of color (Roberts and Garrett 2022)

A taxonomy

A taxonomy of text as data

- A table of the taxonomy.

Takeaways

So, fancy ways to sort beans?

- In most applications today, people use both supervised and unsupervised models to either measure or to discover.
- Example: Partisanship over time in congress.

What you can't escape

- Many scholars see machine learning as a-theoretical and is disconnected from rigorous thinking about "modeling". Additionally, as many political scientists are interested in causal inference (explaining why x causes y), they dismiss the value of predictive models in political research methods. I disagree with both sentiments.
- Specifically to this discussion, we see that both supervised and unsupervised models require explicit decisions by the researcher. You have to make choices about the sample; in supervised machine learning, you have to label your training set; in unsupervised machine learning, you have to not only make decisions about the model and number of clusters you are interested in, but you also have to interpret and label the clusters yourself. Your computer is unable to tell you what these clusters mean. You have to find a way to describe each cluster yourself.

Questions

Contact:
damon.roberts-1@colorado.edu

Takeaways

References and recommended readings

References i

- Garrett, Tyler P. and Damon C. Roberts. 2022. "Don't Interrupt Me: The Interruption of Female and Nominees of Color in Federal Judiciary Confirmation Hearings." Presented at the Annual Meeting of the Midwestern Political Science Association. Link to paper.
- Roberts, Damon C. and Stephen M. Utych. 2021. "Linking Gender Language and Partisanship: Developing a Database of Masculine and Feminine Words." Political Research Quarterly. Link to paper.
- Roberts, Damon C., Stephen M. Utych and Alexandra Siegel. 2020. "An Expanded Gendered Language Dictionary for Political Analysis." Presented at the Annual Meeting of the American Political Science Association. Link to Paper.

References ii

- Recommended readings for those interested:
  - Less technical and good introduction: Grimmer, Justin, Margaret E. Roberts, and Brandon M. Stewart. 2022. Text as data: A new framework for machine learning and the social sciences. Princeton University Press
  - Comprehensive discussion: Eisenstein, Jacob. 2019. Introduction to natural language processing. MIT Press

Supervised machine learning with text data

- Steps for doing supervised machine learning with text data

Supervised machine learning with text data

- Steps for doing supervised machine learning with text data
    1. Collect all of your text like you would when collecting data. Either get a sample, or if you can, use your population

Supervised machine learning with text data

2. Randomly split your text into a training, validation, and testing set.

Supervised machine learning with text data

3. With your training data, manually code the documents based on what you are wanting to classify the text as

Supervised machine learning with text data

4. Then select a particular machine learning model to train using your manually coded data. What the model is doing is learning the similarities and differences between the text placed in one category versus another.

5. Once you have trained your model, try running it on your validation set. You can then see whether your model is doing okay and picking up on the patterns you want it to while also being sure it isn't trained to well so that it can't generalize. In this step, you do what is often referred to as "pruning a model's hyperparameters". Meaning, you are just simply looking at diagnostics and making a few slight tweaks to things. Once you move past this step, you cannot make adjustments to your model with the same data. If you move past this step and realize something is wrong, you will need to restart this whole process.

6. Then apply your model to your testing set. Once applied to your testing set, that is it. You then should have all of your data classified into the categories you wanted. PResent the results of your testing set in your paper.

Unsupervised machine learning with text data

- Steps for doing unsupervised machine learning with text data

Unsupervised machine learning with text data

1. Collect all of your text like you would when collecting data. Either get a sample, or if you can, use your population

2. Randomly split your text into a training, validation, and testing set.

Unsupervised machine learning with text data

3. Train a model so that it might "discover" patterns in the documents.

4. Validate the model to see if it is picking up on patterns that have some utility

5. Apply the model to the test set.

6. Look at discovered patterns, look at examples that score high on relevant metrics (e.g. semantic coherence), provide interpretations for what these patterns are and what they mean.