POL 574, Quantitative Analysis IV
Prof. Arthur Spirling
Assignment date: March 7, 2025

Homework 2

This homework must be turned in on Canvas by **5pm on Friday, March 28, 2025**. Late work will incur penalties of the equivalent of one third of a letter grade per day late.

It must be your own work, and your own work only—you must not copy anyone's work, or allow anyone to copy yours. This extends to writing code. You may consult with others, but when you write up, you must do so alone.

Your homework submission must be a PDF or HTML report, containing all written answers and code, generated from `RMarkdown`. **Raw `.R` or `.Rmd` files will not be accepted.**

Please remember the following:

- Each question part should be clearly labeled in your submission.
- Do not include written answers as code comments. We will not grade code comments.
- The code used to obtain the answer for each question part should accompany the written answer.
- **Your code must be included in full, such that your understanding of the problems can be assessed.**

There are plenty of resources to get started with `RMarkdown` online, see for instance here. You can also use the `template.Rmd` template in our GitHub repo to get started. Using this template is not required.

Part 1

1. This question uses stylometrics via the `stylest2` package. Our interest is in the Levellers, a group of English radicals from the 17th Century. In particular, we want to know who wrote two pamphlets called (among other things) "The Humble Petition" and a "Warning for all the Counties". You have three document-feature matrices available in `leveller_corpus`. These are for the two anonymous documents and the `main` set of Leveller writings from the period.

   (a) For the "main" corpus, select the optimal vocab using cross-validation. Set the seed to `123` and report the optimal cutoff (use `stylest2_select_vocab`). What is the cutoff?

   (b) Fit a `stylest2` model to the main texts, using that cutoff. Using `stylest2_predict`, generate the log odds that each author in the data wrote the texts (we know) they wrote. For the first ten documents in this subcorpus, report the true authors and the most likely authors under the model. How do they differ?

(c) What are the most influential terms in the model? (give the top ten). Are they mostly substantive or function words?

(d) Using your stylometric model, predict the author of the unknown texts. For the most likely author in each case, report the probability (via the log-odds) the model estimates they wrote the text in question

2. This question uses a Bradley-Terry approach to model contest data. The set up is as Hopkins & Noel (2022). That is, respondents are shown pairs of politicians (Senators in this case) and asked who is more conservative. Whoever that person is (of the pair) "wins" the contest. The data is in `match_plus.rds`

(a) Use the `BradleyTerry2::BTm` function to fit an unstructured contest model. Set Sanders as the reference player. Report the "abilities" (the conservatism) of all the Senators from most to least conservative. Note that the reference player is defined to have an ability (a λ) of 1.

(b) Compare the estimates from the contest model to the Senator (first dimension) NOMINATE scores (found here). What is the correlation between these measures?

(c) In the contest data, what is the empirical probability that Duckworth is more conservative than Murkowski? That is, how often does Duckworth beat Murkowski as a proportion of the times they meet?

(d) From the model ability estimates, what is the (estimated) probability that Duckworth is more conservative than Murkowski? What is the probability that Murkowski is more conservative than Duckworth?

(e) Now fit a "structured" Bradley-Terry model (see the `chameleons` example for help) using whether the Senator is from a "big state" as the only (admittedly boring) predictor. Is being from a "big state" associated with being more or less conservate?

(f) A nice feature of contest models is that we can easily add contests and re-estimate the model to include new "players". This is very different from a measure like NOMINATE. Add two contests:

- Trump beats Murkowski
- Cruz beats Trump

Re-estimate the model with these contests and Cruz as the reference player. Report the estimated conservatism of the politicians from left to right. What do you notice about the standard error (and p value) for Trump relative to the other politicians? Why?

Part 2

3. We would like you to perform some Naive Bayes classification **by hand** (that is, you may use math functions or DFM-creating functions, but not any built-in Naive Bayes functions). Make sure to show your work!

(a) Imagine a situation in which you are a political lobbyist and receive emails to the same address for both work and personal matters. The contents of those emails after all relevant preprocessing are displayed in Table 1. However, you are tired of both your work and personal emails pooled in the same inbox and want to separate them (into 'Work' and 'Personal'). Using the standard Naive Bayes classifier without smoothing, estimate for each party the posterior probability (up to a proportionality constant: i.e., the prior multiplied by the likelihood) that the following email was sent in a personal capacity: "vote support media dinner". Report these estimates. Based on these results, would you predict the mystery email is related to work or personal matters? Explain whether you trust your findings and why.

email	content
personal1	dinner weekend photo likes skiing beer
personal2	flowers partner dinner weekend together photo
personal3	social media weekend dinner photo likes
work1	morning analyze data vote lunch partner
work2	client lobbying vote tuesday support bill
work3	partner meeting lunch presentation capitol media
work4	social media vote subsidy public support

Table 1: Training set of emails

(b) Now impose Laplace smoothing on the problem. That is add one to the numerator and the size of the vocabulary to the denominator, then re-estimate the posterior probabilities. Report your findings. Based on these new results, would you predict the mystery email is related to work or personal matters? Beyond computational reasons (i.e. avoiding $log(0)$'s), can you think of any theoretical reason why smoothing might make sense? *Hint: the above data is but a sample of the types of emails you might receive in your work and personal life.*

(c) Now, suppose your boss suspects you don't check your work email and really wants to get the following email into your Personal inbox (they would otherwise be blocked by a *non-smoothed* Naive Bayes filter):

healthcare XXX XXX

From the words available in your personal emails, which ones should your boss substitute in for the two place holders (XXX) to *most* increase the probability it makes it into your Personal inbox? Make sure to explain your reasoning. *Hint: You would like to find words with the largest difference between the Work and Personal posterior probability. Note that the two words can be identical.*

(d) How would you try to alter your email rules in response to this strategy? How does this achieve your bosses goals? What do we call this general process of degrading filtering via the logic of the Naive Bayes classifier?

Part 3

For this exercise you will use a database of Amazon food reviews (source). Each review has a product rating on a 1-5 scale (with 1 being the least positive), along with a written review. You'll be asked to use some of the supervised learning techniques we've discussed in class to analyze the text from these reviews.

Before we get started, be sure to read a few of the reviews, to get a feel for the language used, and any potential imperfections in the text created during the scraping process.

4. Before we apply any classification algorithms to the food reviews, we will need a general classifier that tells us whether the review was positive or negative—the "true classification."

   (a) Divide the reviews at the empirical median rating and assign each review a label as being "positive"—if the rating was greater than or equal to the empirical median score—or "negative"—if the rating is less than the empirical median (you can use 1 and 0 as labels if you prefer, just be consistent as you do the exercises below). These are your true class labels. Report the proportion that are positive and negative, and the median rating.

5. The first method we'll use to classify reviews as being positive or negative will be dictionary based. To do so, you will use the dictionaries of positive and negative words discussed in Hu & Liu (2004)—provided to you as "negative-words.txt" and "positive-words.txt". You **must** use the dictionaries provided and may not use any substitutes from R packages.

   (a) First, preprocess the text by cleaning apostrophes, removing punctuation and setting to lower case. Create a `dfm` of the review text. Then, generate a sentiment score for each review based on the number of positive words minus the number of negative words. Create a histogram to visualize the distribution of the continuous sentiment score.

   (b) Estimate the mean sentiment score in each *true* class together with its standard error (not standard deviation!). Report your results in a table or graph. *You can, but do not have to use bootstrapping for this.*

   (c) Create a vector of dichotomous variables, of equal length to the number of reviews, in which texts that have a positive sentiment (from part (a)) are labeled "positive," while those with a negative score are labeled "negative". Report the percent of reviews in each category, and discuss the results.

   (d) Evaluate the performance of your model at identifying positive or negative reviews by creating a confusion matrix with the positive and negative values assigned by the sentiment score (created in 3(a)) on the vertical axis and the binary "true" classifications

(created in 2(a)) on the horizontal axis. Use this confusion matrix to compute the accuracy, precision, recall and F1 score of the sentiment classifier. Report these findings along with the confusion matrix. In terms of accuracy, how would you evaluate the performance of this classifier?

Hint: is there a baseline we can compare it to?

(e) Now, based on your classification of reviews, inspect the following false positives and false negatives: (1) reviews that were negative reality, but were classified as "positive" using the sentiment score and (2) reviews that were positive in reality, but were classified as "negative" using the sentiment score. Inspect the text of at least 5 of these reviews and comment on why they were misclassified by the sentiment score.

6. Next, we'll train a Naive Bayes classifier to predict if a review is positive or negative. Create a `dfm` with the following preprocessing: remove punctuation, remove numbers, set to lower case, stem terms and remove stop words.

(a) Use the `textmodel` function in `quanteda` to train a *smoothed* Naive Bayes classifier with uniform priors, using 80% of the reviews in the training set and 20% in the test set (Note: features in the test set should match the set of features in the training set. See quanteda's `dfm_match` function.). Report the accuracy, precision, recall and F1 score of your predictions. Include the confusion matrix in your answer.

(b) Were you to change the priors from "uniform" to "docfreq," would you expect this to change the performance of Naive Bayes predictions? Why? Re-estimate the smoothed Naive Bayes with the "docfreq" prior. Report the accuracy, precision, recall and F1 score of these new results. Include the confusion matrix in your answer. In terms of accuracy, how would you evaluate the performance of this classifier?

(c) Fit the model without smoothing and a uniform prior. Report the accuracy, precision, recall and F1 score of your predictions. Include the confusion matrix in your answer. How does the accuracy compare to the previous models? Why might this be?

(d) In the above exercise we only used words as features. Can you think of other features beyond words that may help classify the sentiment of a document?

7. Next, we will estimate the latent ideological space of a few notable State of the Union speeches using the `wordscores` function in `quanteda`.

(a) Using State of the Union speeches from the `sotu` package, create a vector of wordscores for the Reagan 1984 State of the Union speech (anchor repu) and, 9 years later, the Clinton 1993 speech (anchor demo) using the technique described in Laver, Benoit &

Garry (2003). That is, you should fit a `wordscores` model to the anchor texts. What are the 10 most extreme words in either direction (i.e. the 10 lowest and 10 highest wordscores)? Report your findings.

Note: Briefly state your choice regarding preprocessing and smoothing for this model. You may assign the anchor texts scores of [-1,1].

(b) Apply your wordscores model to some non-anchor documents, using all SOTU speeches from 2000-20. This should generate a wordscores estimate for each document. Report your results with a plot of the estimates over time. Did you correctly predict the party for each speech? What does the warning message suggest? What can you tell about the shrinkage of the scores? Can this be problematic? Why?

(c) Now, fit the model with the standard LBG (Laver, Benoit & Garry) rescaling (`rescaling = 'lbg'`) and plot the estimates over time. How does your results compare to the previous specification without rescaling?

8. Use `set.seed(22)` for this question. Here, we will be comparing ridge and lasso regularization. The data of interest is `leveller_predictors`. We will try to predict `outcome_y` (1 if the author of the document was a Leveller, 0 if not) using the token counts from documents they wrote. In what follows assume the problem is similar to a linear regression, such that `family` is set to "gaussian".

(a) Regress `outcome_`y on all the variables, but without an intercept, using a linear model. What do you notice about the coefficient estimates and/or their standard errors ? Why does this occur?

(b) Use `cv.glmnet` to perform a lasso regression. Use the same dependent and independent variables as previously (no intercept), and allow the function to optimize λ. What is the optimal (`1se`) value of λ it chooses?

(c) Use `cv.glmnet` to perform a ridge regression. Use the same dependent and independent variables as previously (no intercept), and allow the function to optimize λ. What is the optimal (`1se`) value of λ it chooses?

(d) Look at the coefficients estimated in both cases. Report, for each model, the variables with the ten largest (absolute) coefficients. Looking across all coefficient estimates, which regularization procedure yields more zero valued estimates?

(e) Now build a ridge "path". That is, for the ridge regression, allow λ to increase from 1 to 20, by 1 each time. Draw a plot of the coefficients on these variables (tokens) as the penalty increases: "syon", "fairer" and "inhumanity". Do these coefficients get "zero'd out"?

(f) Now build a lasso "path". That is, for the lasso regression, allow λ to increase from 0 to 3, by 0.2 each time. Draw a plot of the coefficients on these variables (tokens) as the penalty increases: "prison", "faithfull" (note the spelling) and "therefore". Do these coefficients get "zeroed out"?

9. This question gives some more intuition on ROC curves. Consider a problem in which there are 2414 emails to be classified. The true labels of these emails are given in `email.RDS`.

   (a) What proportion of the (true) labels are spam and ham?

   (b) Simulate a random classifier. Specifically, simulate the classifier that randomly labels a given email as spam with probability 0.5. Using the `ROCR` package, produce the ROC curve for this classifier.

   (c) What is the AUC value for this random classifier (rounded to the first decimal)?

   (d) Repeat this exercise, except change the predicted labels to be spam with probability 0.6. Does the random classifier curve and the AUC change, substantively? Why or why not? (Hint: check the correlation between the true labels and the predicted labels).

   (e) Simulate a classifier that has the worst possible performance on this data. What correlation will the true and predicted labels have in this case?

   (f) Draw the resulting ROC curve (using `ROCR`) and calculate the AUC for this example. Interpret the AUC value in the way we discussed in class. That is, with respect to the probability that a randomly chosen positive instance will be ranked above a randomly chosen negative instance (in terms of the probability that it is, indeed, a positive case).