Gov2018 Midterm (Spring 24)

Read the following instructions carefully:

- All answers must be typed up. Your text, math, code, and figures should be included together for each problem. RMarkdown will do this automatically and is strongly encouraged.
- You have two weeks to finish the task: 3/20 - 4/3.
- Late submission is until 4/7 midnight, your score will be discounted .
- You are encouraged to work in groups, but you should write up your answers alone and cite your peers.
- Upload your compiled PDF to Gradescope.

Problem 1

In this task, we will conduct supervised classification and unsupervised dimension reduction using a corpus of YouTube video transcripts about guns and gun control. The cleaned transcripts and associated metadata are available in `gun_transcripts.zip` and `gun_videos.csv`, respectively. Key variable definitions in the metadata are as follows:

- `rec.video.name`: The title of each video
- `rec.video.id`: The ID (unique identifier by YouTube) of each video
- `rec.channel.name`: The channel that the video is from
- `rec.channel.id`: The ID (unique identifier by YouTube) of each channel

Problem 1 is intended to be short. Begin by loading the data and constructing a document-term matrix (DTM). You may preprocess the data in any way that you see fit. For an in-depth discussion of issues in preprocessing, review

> Denny, Matthew J. and Arthur Spirling. 2018. "Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It," *Political Analysis*, Vol. 26, Iss. 2, pp. 168–189.

(a) Briefly discuss the rationale for and implications of all decisions regarding stemming, stopword removal, choice of n-gram length (potentially including $n = 1$, if you believe this is justified), removal of infrequent words, interactions, and other preprocessing steps.

(b) In addition, you may wish to rescale or reweight the data by standardizing, rescaling each row to sum to unity, or replacing the DTM with a term-frequency inverse-document-frequency (tf-idf) matrix. Again, briefly discuss your choice, particularly as it relates to long documents. Note that the the decision not to weight also constitutes a preprocessing choice that should be defended.

(c) We refer to the data matrix resulting from the preceding preprocessing regime as the "preferred" preprocessing regime. Next, select another defensible but substantially differing sequence of preprocessing decisions, then briefly report any major differences in summary statistics (e.g., dimensions of the resulting data matrix, typical values of the dropped/added variables). You will return to the "alternative" regime in Problem 3.

Problem 2

The file `gun_videos.csv` also contains a `label` variable in which a subset of videos are classified as `pro-gun` or `anti-gun`. Note that given these labels are assigned on the basis of the originating YouTube channel's characteristics and do not necessarily reflect video characteristics. Labeled channels with unclear political orientations were assigned `undetermined`, and unlabeled videos have a value of `NA`.

For the labeled subset, the `fold` variable divides videos into K roughly equally sized folds such that all videos originating from the same channel are contained within the same fold.

Before beginning, read this problem in its entirety. You may wish to write generalizable functions to accomplish common tasks to avoid duplicating effort.

Optional: `gun_recs.csv` contains the network structure of recommendations between videos. You can decide whether it is helpful for your classification task. (When you watch a YouTube video, the site recommends several related videos to watch later. This dataset was generated by iteratively crawling through these recommendations.) The following code will construct a recommendation graph from "origin" and "recommendation" video ID pairs:

```
G <- graph.edgelist(el = as.matrix(videos[, c('origin.video.name', 'rec.video.name')]))
```

Note that this recommendation network also contains a number of uncaptioned videos, as well as non-gun-related videos that were filtered out (based on title keywords) to produce the corpus under analysis.

(a) Choose two classification algorithms that require selection of a regularization parameter (or parameters). For each algorithm, select at least three widely varying values for the regularization parameter. Then, fit the corresponding models to the data from folds 1 to $K-1$. (Note that the specifics of this procedure will depend on the chosen model.)

Note that the observations labeled `undetermined` do not clearly belong to either of the categories of interest. You are free to drop these observations from the training set, treat them as a third class, fix their label at an intermediate value, or otherwise handle them as you see fit.

You will use hard true positive rate, hard true negative rate, and at least one additional metric of your choice to evaluate performance. For each of the models, compute and report (i) average in-sample performance across observations, using labels from folds 1 to $K-1$. Thes results should contain at least 18 reported values (two algorithms, three performance metrics, and three regularization parameter settings). Then report (ii) out-of-sample performance, using labels from fold K, for another 18 values. Collect these results in an informative and clearly labeled plot or table, then describe and interpret your results with a succinct caption.

(b) Rotate through the K folds, computing in-sample and out-of-sample performance each time. Average across folds, weighting by the number of observations in each. Again, collect your results in an

informative plot or table, then describe and interpret any patterns.

(c) Based on the results described above, create a final classifier that uses all labeled data. Justify your decisions. Save the resulting model.

(d) Finally, randomly sample at least 10 unique unlabeled observations and hand-code them as `pro gun`, `anti gun` on the basis of the transcript, video, or any channel metadata you wish—*without* using your model. You may view the source videos by appending the video ID to `https://www.youtube.com/watch?v=`

Report the sampled video IDs and your hand-coded labels in a table. *Subsequently*, run your classifier on these videos and report the results as well. Finally, briefly describe the basis on which your classifier made its decision.

You are free to label additional videos to better assess the generalization performance of your classifier. However, if you choose to retrain your model on the basis of these results, to ensure the integrity of the test dataset, you must clearly distinguish observations that were labeled before retraining (potentially contaminated) from those that were labeled after (true test).

If you are collaborating with other students, ensure that your hand-labeled observations are not duplicated. After submission, we will aggregate these labels into a new test set. The test labels will be stripped, and you will be sent the test IDs. You will then run your final classifier to generate predictions, which we will then compare to the test labels.

Problem 3

We now investigate the extent to which unsupervised dimension reduction procedures produce results that map onto your previous supervised classifier.

(a) First, we will conduct a brief simulation to understand the role of feature weighting. Create 1,000 observations by randomly drawing `X1 <- rnorm(1000, mean = 0, sd = 1)`, `X2 <- rnorm(1000, mean = 0, sd = .75)`, and `X3 <- rnorm(1000, mean = 0, sd = .5)`. Create the data matrix with `X <- cbind(X1, X2, X3)` and conduct a singular value decomposition.

Examine the resulting dataset, e.g. with `plot3d(X, aspect = FALSE)` from package `rgl` or by creating a bivariate scatterplot matrix (be sure to set the aspect ratio to 1). Then, superimpose the first right singular vector, e.g. with `arrow3d(p0 = c(0, 0, 0), p1 = ...)`.

Finally, standardize your `X` and repeat. Explain any differences, providing informative figures as needed. Discuss your results with reference to the term "total variance." Intuitively, what is the effect of standardization in principal component analysis? What are the implications for PCA with rare words?

(b) Center the DTM and decide whether to standardize it, using `scale(X, center = TRUE, scale = ...)`. Conduct dimension reduction on your YouTube DTM. (If absolutely necessary for computational reasons, you may consider truncated SVD or randomly subsampling of videos. Justify your decision.)

What proportion of total variation does your first dimension explain? Produce a "scree plot" depicting how the proportion decreases over the top 10 dimensions.

(c) Interpret your first dimension by examining its most positive and negative loadings. Qualitatively, what does this appear to capture? Next, sample videos that score in the top and bottom deciles on the first dimension. Describe your results. Do they confirm your initial assessment? How do videos at the extremes compare to each other (e.g. in word count, duration, views, etc.)? To the average video?

Repeat this procedure for the second dimension. If you had to label your axes in a single word or phrase (alternatively, with labels for the positive/negative directions), what would they be?

(d) Create a bivariate plot in which the x-axis represents fitted values based on your classifier from Problem 2. (For this question, only use unlabeled data.) The y-axis should represent your first dimension from this problem. How do they correspond?

Next, create a second plot in which the x- and y-axes represent your first and second latent dimensions. Indicate the fitted values using color, a contour plot, or any other clear and informative visual presentation. Does the second dimension appear to map onto your fitted values? Discuss.

(e) Finally, compute a second SVD using the data matrix from your alternative preprocessing regime. Plot the first-dimension scores from both approaches against each other and discuss the sensitivity of your results to preprocessing decisions.