

41201-01: Data Mining

Professor Taddy

**Problem Set 6**

Joanne Chen

James Hardiman

Evan Johnson

Honor code: We pledge our honor that we have not violated the Honor Code during the completion of this assignment.

Question 1: Fit K-means to speech text for K in 5,10,15,20,25. Use BIC to choose the K and interpret the selected model.

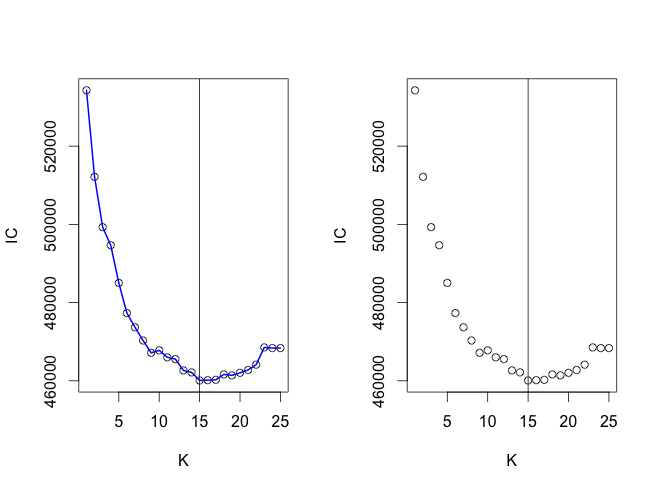
*f <- as.matrix(congress109Counts/colSums(congress109Counts))*

*fs <- scale(f)*

*kfit <- lapply(1:25, function(k) kmeans(fs,k))*

*kbic <- sapply(kfit,kIC,"B")*

*plot(kbic, xlab="K", ylab="IC", lines(kbic, col=4, lwd=2), abline(v=which.min(kbic)))*



We wound up writing code to create 25 different buckets of K to segment the x values rather than just ones of sizes 5, 10, 15, 20 , and 25 since it was a little bit easier to code, but it shows that we get the lowest deviance somewhere close to when there are 15 buckets. We ran code to confirm that:

*which.min(kbic)*

*[1] 15*

We find that 15 K buckets is the one that minimizes the bic. We then printed the top ten words that made up each of these buckets and used subjective judgment to label those buckets based on the top ten words that were coming up. The names for those buckets appears at the top of each of them next to the original number for them.

*kmfs <- kmeans(fs,which.min(kbic)) # run a clustering and look at it*

*print(apply(kmfs$centers,1,function(c) colnames(fs)[order(-c)[1:10]]))*



Question 2: Fit a topic model for the speech counts. Use Bayes factors to choose the number of topics, and interpret your chosen model.

To determine how many K buckets to create for our topics model, we first transformed the data into a simple triplet matrix.

*c <- as.simple\_triplet\_matrix(congress109Counts)*

Next, we created multiple topics models to see which size K created the highest Bayes’ Factor, but did so by stepping up in Bucket sizes of five to see generally what size we wanted.

*tpcs <- topics(c,K=5\*(1:10),tol=10)*

*log posterior increase: 5662.2, 1459.2, 441.8, 97.1, 52.8, 18.4, done.*

*log BF( 5 ) = 58785.48*

*log posterior increase: 3996.9, 420.7, 316, 133.4, 125.1, 95.8, 169.7, 82.8, 34.7, 23.8, 43.4, done.*

*log BF( 10 ) = 77275.65*

*log posterior increase: 2869.7, 358.7, 106.9, 59.6, 66.3, 43.8, 62.4, 88.3, 14, done.*

*log BF( 15 ) = 75474.02*

*log posterior increase: 2005.5, 192.1, 186.2, 107.5, 150, 52.3, 18.1, 31.3, 14, done.*

*log BF( 20 ) = 67550.3*

The bucket size around 10 appeared to be the one that created the highest Baye’s Factor. So to see more specifically which number of buckets created the highest Baye’s Factor, we started at a bucket size of 5 and then increased it to 20.

*> tpcs2 <- topics(c,K=(5:20),tol=10)*

*Estimating on a 529 document collection.*

*Fit and Bayes Factor Estimation for K = 5 ... 20*

*log posterior increase: 5662.2, 1459.2, 441.8, 97.1, 52.8, 18.4, done.*

*log BF( 5 ) = 58785.48*

*log posterior increase: 2535.7, 86.3, 30, 54.6, 62.8, 23, 44.9, 20.6, done.*

*log BF( 6 ) = 65198.5*

*log posterior increase: 2143.4, 51.2, 23, 72.6, 18.1, done.*

*log BF( 7 ) = 69892.7*

*log posterior increase: 1786.9, 94.3, 51.2, 129.7, 116.2, 48.7, 90, 14.8, 10, done.*

*log BF( 8 ) = 74850.41*

*log posterior increase: 2020.7, 72, 13.8, 10.5, done.*

*log BF( 9 ) = 77850.01*

*log posterior increase: 1580.4, 101, 37.7, 25, 12.1, 39.9, done.*

*log BF( 10 ) = 78537.91*

*log posterior increase: 1246.6, 104.7, 43, 44.9, 14.3, done.*

*log BF( 11 ) = 79549.36*

*log posterior increase: 1310.1, 52.7, done.*

*log BF( 12 ) = 80380.79*

*log posterior increase: 1373.8, 85.7, 21.1, done.*

*log BF( 13 ) = 79982.93*

*log posterior increase: 1187.8, 161.7, 31.4, done.*

*log BF( 14 ) = 79554.86*

The largest Baye’s Factor was the one where we had twelve buckets of topics. Below are the top ten most common words that appeared in each of those topics, and how often they appeared comparatively. Please note, we have also labelled all of these topic buckets using similar subjective evaluation criteria.

*summary(tpcs2, n=10)*



Question 3. Connect the unsupervised clusters to partisanship.

a) tabulate party membership by K-means cluster. Are there any non-partisan topics?

b) fit topic regressions for each of party and repshare. Compare to regression onto phrase percentages:

To better interpret our buckets by K-means cluster, we then tried to determine which ones had the most numbers of Republicans and Democrats in them, with the assumption that a bucket with more Democrats would likely have a larger share of words that we would expect them to say more commonly, and the reverse being true for Republicans.

*tapply(congress109Ideology$party,kmfs$cluster,table)*

After getting the results from which bucket had the strongest draw from each party, we then placed those results into the below table and highlighted those buckets that had the more than one Democrat or Republican. The darker the red, the more Republicans it had, and the darker the blue, the more Democrats that it had. Bucket 14, which we labelled “insurance” is in gray since it has so many people from both sides of the aisle. This is likely a non-partisan topic.



It appears then that the strongest-leaning Republican bucket, no. 11 which we have labelled “Administrative 2”, is probably because the Republicans in 2005 were in the majority, and are using words that appear to be associated with running the legislature like, “housin.urban.affair”, “buinsess.meeting”, and “circuit.court”. Certainly the last pair of words, “ronald.reagan”, would be expected to be in a strongly Republican bucket.

Bucket 10, which we have labelled, “Budget2/ Asia”, contains many more words and phrases that we would expect out of Democrats that were more opposed to the Iraq War and in favor of larger domestic spending, “climate.change”, “cost.war”, “care.cut”, “additional.funding”, and “health.care.funding”. Though we can’t explain all of the variation in the output from this cursory review, the initial results seem to make sense given our understanding of the political landscape in 2005.

We then tried to answer the second part of the question by regressing our topic model buckets to see if they could predict party membership and the percentage share of the Congress member’s district who voted for George W. Bush in 2004.

To do so, we first create our outcome variables so that we could run predictive regressions, and transformed our x variables so that we could run more naïve regressions on them using gamlr.

*party <- congress109Ideology$party*

*repshare <- congress109Ideology$repshare*

*x <- 100\*congress109Counts/colSums(congress109Counts)*

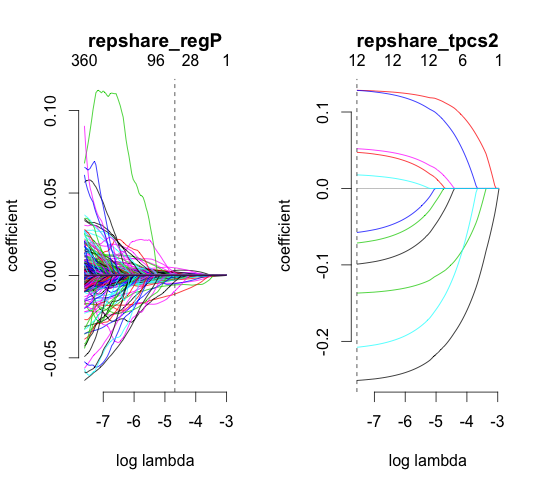
Next, we ran a simple regression to see if using our topic buckets were better at making a more accurate model for predicting representative share. Considering how much bigger the penalty was able to get before selecting the best AICc model, it looks like the topic model was in fact better.

*repshare\_regP <- gamlr(x = x, y = repshare)*

*repshare\_tpcs2 <- gamlr(x = tpcs2$omega, y = repshare)*

*plot(repshare\_regP, main="repshare\_regP")*

*plot(repshare\_tpcs2, main="repshare\_tpcs2")*



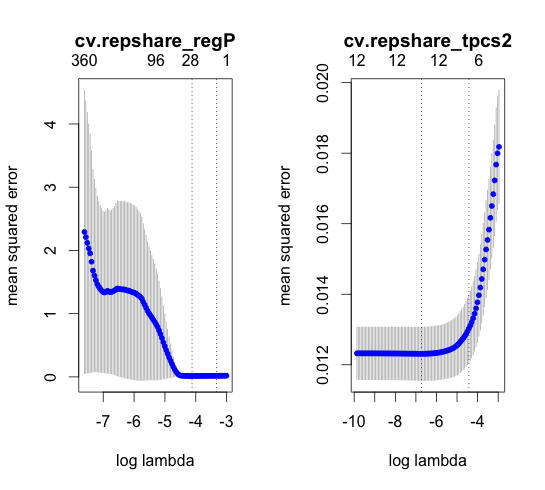
We next did essentially the same thing, using all of our variables as a giant set of covariates to predict representative share, and next using buckets, to see which was better at predicting voter share. Again, we find that using the topic buckets is better because the mean squared error it produces is less than what the much bigger data set is able to show us.

*cv.repshare\_regP <- cv.gamlr(x = x, y = repshare)*

*cv.repshare\_tpcs2 <- cv.gamlr(x = tpcs2$omega, y = repshare, lambda.min.ratio = 0.001)*

*plot(cv.repshare\_regP, main="cv.repshare\_regP")*

*plot(cv.repshare\_tpcs2, main="cv.repshare\_tpcs2")*



We next ran basically the same thing, only used a multinomial regression to predict party affiliation (we couldn’t run a simple binomial regression because there were a few independents).

We first set up our code to make the clusters necessary to create the outcome variables we are trying to predict.

*cl <- makeCluster(2,type=ifelse(.Platform$OS.type=="unix","FORK","PSOCK"))*

Then, we ran a multinomial regression to predict party affiliation using just our large dataset of X, and our also with our clusters.

*cl <- makeCluster(2,type=ifelse(.Platform$OS.type=="unix","FORK","PSOCK"))*

*party\_reg <- dmr(cl = cl, covars = tpcs2$omega, counts = party, verb = 1)*

*party\_regP <- dmr(cl = cl, covars = x, counts = party, verb = 1)*

*stopCluster(cl)*

To see what the multinomial regression looks like for predicting party affiliation, we printed the coefficients.



We then matched up those coefficients, similar to what we did in the prediction for Bush’s 2004 election share, and color coded those topics that were most predictive for Republicans as the darker red, and the ones that were strongest predictors for Democrats darker blue. Overall, these buckets seemed to make a little bit more sense than what we found for the other buckets that we came up with, as there were not as many that appeared to be as strictly about administrative procedures, and did appear a bit more partisan.

