POL 450C, Homework 6

May 22, 2017

Assigned: 5/22/2017
Due: 6/2/2017

In this problem set we're going to learn more about principal component analysis, by analyzing political speech, and regularized regression, by analyzing student drinking data.

Problem 1: When Will PCA Be Useful?

Your friend notices you looking at the slides about principal component analysis (PCA). Your friend casually remarks that the variance of the eigenvalues of the variance-covariance matrix is a useful heuristic for knowing if PCA can be fruitfully applied in some setting. Your friend goes on to explain that as the variance of the eigenvalues goes up, the more useful PCA will be.

Let's formalize your friend's suggestion. Suppose we have centered data $N \times J$ $\boldsymbol{X}$, which has variance-covariance matrix $\boldsymbol{\Sigma} = \frac{\boldsymbol{X}'\boldsymbol{X}}{N}$. And suppose that $\boldsymbol{\Sigma}$ has eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_J > 0$. Then we calculate the variance of the eigenvalues as

$$\sigma^2 = \frac{1}{J}\sum_{j=1}^{J}(\lambda_j - \bar{\lambda})^2$$

where $\bar{\lambda}$ is $\frac{1}{J}\sum_{i=1}^{J}\lambda_i$. Your friend is saying that as σ^2 gets bigger, a low-dimensional embedding via PCA will provide a better summary of our data.

In order to examine your friend's advice consider the following two scenarios.

Consider the following two data generating processes and suppose that the variance of the data is fixed across both.

Scenario 1: 5 principal components generated the data, with the remaining components comprising a very small amount of noise. Call the variance of the eigenvalues in this scenario σ_1^2

Scenario 2: 1 principal component generated the data, with the remaining components comprising a very small amount of noise. Call the variance of the eigenvalues in this scenario σ_2^2

Compare σ_1^2 and σ_2^2 to assess your friend's claim.

Problem 2: PCA Applied to A Document Term Matrix

We're going to analyze Machiavelli's *The Prince* using principal component analysis. I have included the texts in mach.zip and the *document term matrix* in `mach.csv`. Each row of `mach.csv` corresponds to a subset of the manuscript.

a) We are going to work with a standardized version of the dtm, $\boldsymbol{X}$. For each row $\boldsymbol{x}_i$ we are going to create $\boldsymbol{x}_i^*$,

$$\boldsymbol{x}_i^* \quad = \quad \frac{\boldsymbol{x}_i}{\sum_{j=1}^{500} x_{ij}}$$

Create this normalized document term matrix and call it $\boldsymbol{X}^*$

b) Find the principal components using `prcomp` in `R`. This creates an object that has the principal components `obj$rotation`, the eigenvalues `obj$sdev`, and the scores `obj$x` (the location of each observation on each principal component)

c) Using your data set, confirm the relationship between `prcomp` and singular value decomposition using `svd` in `R`. Specifically, show that the right-singular values $\boldsymbol{V}^{'}$ correspond to the eigenvectors from `prcomp`.

d) Create a plot that shows the percentage of variance each dimension explains. Using this plot, are you able to identify a clear "elbow" in the data?

e) Let's compare the estimate of the reconstruction error in (d) to the estimate of the reconstruction error that comes from cross validation. To perform the cross validation we will perform the following steps:

   i) Being sure to set a seed, assign each document (each row in the dtm) to a fold using `sample`

   ii) Write a `for` loop with the following steps

      1) Select the training data
      2) Select the test data
      3) Apply `prcomp` to the training data
      4) Using the principal components from (3), calculate the scores for the test data on each of the first 100 principal components. To do this,

suppose we have test observation $\boldsymbol{x}_i^*$ and principal component $\boldsymbol{w}_j$. The score will then be:

$$\mathrm{z}_{ij} \quad = \quad (\boldsymbol{x}_i^*)' \boldsymbol{w}_j$$

5) Using the scores, create 100 approximations for each observation, increasing the number of included principal components for each approximation. Recall that if we have M principal components in an approximation, $\tilde{x}_i^M$ our formula is $\tilde{x}_i^M = \sum_{m=1}^{M} z_{im} \boldsymbol{w}_m$.
6) Now, calculate the out of sample reconstruction error for each observation and each approxiation. (This should result a 100x 1 vector and you can store all of the reconstruction errors in an N x 100 matrix)

Compare the out of sample reconstruction error and the reconstruction error reported by the eigenvalues. What do you notice?

e) Our last step is to label the first principal component (and by extension the first dimension). The score for observation i the first dimension is $z_{i1} = (\boldsymbol{x}_i^*)' \boldsymbol{w}_1$. This implies that values of $\boldsymbol{w}_1$ with larger magnitude affect the score on the first dimension more. Using this insight, what are the 20 largest magnitude words for the first dimension? What might this imply about this score?

1 Problem 3: Student Drinking

In the next two problems we're going to use machine learning to predict student's drinking habits. The data come from a public health study of Portugese students. You can read more about the variables (and their interpretation) here:

http://archive.ics.uci.edu/ml/datasets/STUDENT+ALCOHOL+CONSUMPTION#

.

We're going to model the sum of weekday and weekend drinking activity.

The data are stored in `StudentDrinking.RData` on canvas. The dependent variable is, `alcohol`, which is a measure of alcohol consumption. The bigger `alcohol` is, the more students drink. The covariates are stored in `X`.

2 Comparing Coefficients from OLS, LASSO, Ridge, and Elastic Net

We first want to explore the behavior of OLS, LASSO, and Ridge applied to the data.

i) Fit a linear regression of alcohol on the covariates in the included data

ii) Using `cv.glmnet` fit a LASSO regression of alcohol on the covariates

iii) Using `cv.glmnet` fit a Ridge regression of alcohol on the covariates

iv) Using `cv.glmnet` fit an elastic-net regression of alcohol on the covariates, with $\alpha = 0.5$. Explain what $\alpha = 0.5$ implies about the model you're fitting.

v) Using your models from (i-iv) let's examine the behavior of the coefficient on `male` as λ increases

  a) Suppose `glmnet.obj` contains the results from applying `cv.glmnet`. To obtain the coefficient values for the sequence of λ values tested in `cv.glmnet`, we use the coefficient function `coef(glmnet.obj, s = glmnet.obj$lambda)`. Use this function to obtain a matrix of coefficients for the models used in (ii-iv).

  b) Using the matrix for each method, plot the coefficient on `male` against the value of λ from the models in ii-iv. Include the coefficient from OLS as a flat line. What do you notice as λ increases?

3 Problem 4: Cross-Validation, Super Learning and Ensembles

We're going to assess the performance of five models, an unweighted average, and a super-learning average of the methods.

i) First, set the first 20 rows to the side for use as the validation set.

ii) We'll first estimate the (unconstrained) super learner weights.

  a) On the training data (all but the first 20 rows) perform ten fold cross validation, including (1) linear regression, (2) LASSO, (3) Ridge, (4) Elastic-Net, and (5) Random Forest. Obtain 5 predictions for each observation in the training set, one from each observation

  b) Regress the dependent variable on the out of sample prediction, (without including an intercept).

  c) Store those weights as $\boldsymbol{w}$

iii) Now, fit all 5 models from (ii)-(a) to the entire training data set and predict the drinking level from the vallidation set (the data put off to the side).

iv) Obtain two ensemble predictions.

  a) Take the unweighted average of the predictions from the methods

  b) Take the weighted average, using the weights $\boldsymbol{w}$.

v) You should have 7 predictions. Store those in a matrix and report the correlation between the predictions

vi) Using the average absolute difference as a loss function assess the performance of each method. Which method performs best? Which performs the worst?

The average absolute difference for method k is defined as

$$L(\boldsymbol{Y}, \widehat{\boldsymbol{Y}}_k) \;=\; \sum_{i=1}^{N_{\text{validation}}} \frac{|Y_i - \widehat{Y}_{ik}|}{N_{\text{validation}}}$$

where $N_{\text{validation}}$ refers to the number of observations in the validation set $\widehat{\boldsymbol{Y}}_k$ refers to the predictions from the k^{th} method,