

Math for Data Science: Problem Set 3

Sofia Breganni, David Moth, Adarsh Tripathi

2025-24-11

Due Date: Wednesday, December 3 by the end of the day (23:59).

Instructions: Please submit one solution set per group and include your group members' names at the top. Please write your solutions within this Rmd file, under the relevant question. Please submit the knitted output as a pdf. Make sure to show all code you used to arrive at the answer. However, please provide a brief, clear answer to every question rather than making us infer it from your output, and please avoid printing unnecessary output.

1. More Cat Compression

I have another cat named Lev. He was upset that Laszlo got to be in my course materials and he didn't. In this exercise we will explain to Lev why my initial choice of Laszlo was nothing personal.



Figure 1: My other cat, Lev.

- a. Load the image of Lev. Extract its red, green, and blue matrices and store them as separate objects called `r`, `g`, and `b`. Center the blue matrix and compute its variance-covariance matrix. What is the dimensionality of this variance-covariance matrix and why?
- b. Find the total variance of the centered blue matrix for this image. Compare this to the total variance of the centered blue matrix for Laszlo.
- c. Use the `eigen` command to get the eigendecomposition of the blue variance-covariance matrix for Lev. Multiply the centered blue data matrix by the eigenvector corresponding to the largest eigenvalue to get the first principal component. Compute its variance.
- d. Use the answers to the previous two questions to compute the proportion of variance explained for Lev's first principal component. Check your answer against a scree plot produced by the `fviz_eig` command.¹ Compare to what we saw for Laszlo in the lab.

¹If they are similar up to the second decimal place, that's good enough.

- e. Now compute Lev's second principal component. Compute the covariance of the first principal component with the second.² Is this what you expected? Comment briefly.
- f. Now let's run PCA on Lev's `r`, `g`, and `b` matrices using the `prcomp` function with the options `center=FALSE` and `scale.=FALSE`.³ Combine these objects into a list. Now, looping over a handful of numbers of principal components, reconstitute images of Lev as we did in your lab. How many principal components does it take to start to recognize Lev as a cat?
- g. Using what you learned from parts (b), (d), and (f) above, explain why Lev requires more principal components to be minimally represented than Laszlo, despite being equally beautiful.

2. Penalized Regression

We will derive the estimator for ridge regression, which is one of several *penalized regression* methods.⁴ The process we will follow is very similar to the standard regression estimator we derived in your lab, but with a small change to the loss function. Rather than minimizing the sum of squared errors, we will minimize the sum of squared errors subject to a constraint:

$$\|\beta\|_2^2 \leq s$$

where s is just some constant chosen by the analyst. Recall that $\|\beta\|_2^2$ is the squared L_2 norm of the β vector.⁵ This question will build on the concepts and data in the lab, so please revisit those materials if anything here is unclear.

- a. Write down the Lagrangian for this constrained minimization problem. Please use matrix notation (including for the L_2 norm).

We want to minimize the sum of squared residuals:

- $\min_{\beta} \|y - \mathbf{X}\beta\|_2^2$
- Where $\|y - \mathbf{X}\beta\|_2^2 = \sum_{i=1}^n (y_i - (\mathbf{X}\beta)_i)^2 = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$

With an L_2 norm constraint:

- The L_2 norm will be: $\|\beta\|_2^2 = \sum_{i=1}^p \beta_i^2 = \boldsymbol{\beta}^T \boldsymbol{\beta}$
- And the constraint function: $g(\beta) = \boldsymbol{\beta}^T \boldsymbol{\beta} - s$

$$\mathcal{L}(\boldsymbol{\beta}, \lambda) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda(\boldsymbol{\beta}^T \boldsymbol{\beta} - s)$$

- b. Take the first derivative of the Lagrangian with respect to β and set it equal to $\mathbf{0}$ to get the first order condition. (If you're feeling retro, you can use the Matrix Cookbook to help you.) Solve for $\hat{\boldsymbol{\beta}}$.
- c. Compare your answer to the standard linear regression estimator. Under what condition are they the same?

²You can round to 10 decimal places.

³Scaling and centering is desirable when your variables are on different scales, but in this case it messes up the colors.

⁴See p. 237-244 of ISL.

⁵Also known as the Euclidian norm.

- d. Using the same `BostonHousing` dataset and the same set of variables from the lab, compute the ridge regression $\hat{\beta}$ with the equation you found in part (b) above. First standardize your \mathbf{X} matrix using the `scale()` function in R. Use the `glmnet()` function in the `glmnet` package to check your answer.⁶
- e. For the sequence of lambdas below, make a plot with `lambda_seq` on the x -axis (increasing from 0 to 100) and the estimated β coefficients for per capita crime rate in the town (in blue), proximity to the Charles River (in red), and nitric oxides concentration (in green) on the y -axis. Use `geom_line` to plot the coefficients and add a black horizontal line at $y = 0$. Label your axes and include a legend.
- f. Discussion:

- Based on your plot above, explain why ridge is one of a number of so-called “shrinkage” estimators.

Your boxed content goes here. You can put text, Markdown, equations, etc.

- Tie this back to the constrained optimization problem you solved to obtain the ridge regression $\hat{\beta}$. Can you see how a larger value of λ (or equivalently a small value of s) corresponds to greater shrinkage?
- You will learn more about this class of estimators and their virtues next semester, but do you have any intuitions as to when and why they might be desirable?

Your boxed content goes here. You can put text, Markdown, equations, etc.

⁶Note that `glmnet`'s `lambda` parameter corresponds to your $\frac{\lambda}{N}$, where N is the number of rows of your data. Also, your estimates may differ slightly from `glmnet`'s, at around the second decimal place. That's alright; this is likely due to `glmnet`'s optimization algorithm. It's computationally expensive to invert large matrices so `glmnet` is probably taking some more efficient but (slightly) less precise approach.