

# Assignment 6

## CS 750/850 Machine Learning

- **Due:** March 30th at 11:59PM
- **Submission:** Turn in both a **PDF** and the **source code** on MyCourses
- **Questions:** Piazza and Office hours:
  - Marek: Wed 1:30-2:30pm <https://unh.zoom.us/j/356262933>
  - Soheil: Mon 2pm-4pm
  - Xihong: Thu 1:30-3:30pm <https://unh.zoom.us/j/587656408>

### Problem 1 [33%]

In this problem, we will establish some basic properties of vectors and linear functions.

1. The  $L_2$  norm of a vector measures the length (size) of a vector. The norm for a vector  $x$  of size  $n$  is defined as:

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$$

Show that the norm can be expressed as the following quadratic expression:

$$\|x\|_2^2 = x^T x$$

2. Let  $a$  and  $x$  be vectors of size  $n = 3$  and consider the following linear function  $f(x) = a^T x$ . Show that the gradient of  $f$  is:  $\nabla_x f(x) = a$ .
3. Let  $A$  be a **symmetric** matrix of size  $3 \times 3$  and consider the following quadratic function  $f(x) = x^T A x$ . Show that the gradient of  $f$  is:  $\nabla_x f(x) = 2Ax$ . A matrix is symmetric if  $A_{ij} = A_{ji}$  for all  $i$  and  $j$ .

### Problem 2 [34%]

*Hint:* You can follow the slides from the March 4th class, or the LAR reference from the class website. See the class website for some recommended linear algebra references.

You will derive the formula used to compute the solution to *ridge regression*. The objective in ridge regression is:

$$f(\beta) = \|y - A\beta\|_2^2 + \lambda \|\beta\|_2^2$$

Here,  $\beta$  is the vector of coefficients that we want to optimize,  $A$  is the design matrix,  $y$  is the target, and  $\lambda$  is the regularization coefficient. The notation  $\|\cdot\|_2$  represents the Euclidean (or  $L_2$ ) norm.

Our goal is to find  $\beta$  that solves:

$$\min_{\beta} f(\beta)$$

Follow the next steps to compute it.

1. Express the ridge regression objective  $f(\beta)$  in terms of linear and quadratic terms. Recall that  $\|\beta\|_2^2 = \beta^T \beta$ . The results should be similar to the objective function of linear regression.

2. Derive the gradient:  $\nabla_{\beta} f(\beta)$  using the linear and quadratic terms above.
3. Since  $f$  is convex, its minimal value is attained when

$$\nabla_{\beta} f(\beta) = 0$$

Derive the expression for the  $\beta$  that satisfies the inequality above.

4. Implement the algorithm for computing  $\beta$  and use it on a small dataset of your choice. Do not forget about the intercept.
5. Compare your solution with `glmnet` (or another standard implementation) using a small example. Are the results the same? Why yes, or no?

### Problem 3 [33%]

Using the MNIST dataset, which we used already in Assignment 2, compare whether boosting, bagging, and random forests work the best. You may want to use only a subset of the data.

*Optional:* Use `xgboost` (also available for Python) to see whether the results are better than other boosting methods.

### Optional Problem 4 [+10%]

*Hint:* We did not cover this material in class, but it is important regardless. The slides from March 9th may be helpful when implementing this.

In this problem, you will implement a gradient descent algorithm for solving a linear regression problem. Recall that the RSS objective in linear regression is:

$$f(\beta) = \|y - A\beta\|_2^2$$

1. Consider the problem of predicting revenue as a function of spending on TV and Radio advertising. There are only 4 data points:

Revenue	TV	Radio
20	3	7
15	4	6
32	6	1
5	1	1

Write down the design matrix of predictors,  $A$ , and the response vector,  $y$ , for this regression problem. Do not forget about the intercept, which can be modeled as a predictor with a constant value over all data points. The matrix  $A$  should be  $4 \times 3$  dimensional.

2. Express the RSS objective  $f(\beta) = \|y - A\beta\|_2^2$  in terms of linear and quadratic terms.
3. Derive the gradient  $\nabla_{\beta} f(\beta)$  using the linear and quadratic terms above.
4. Implement a gradient descent method with a fixed step size in R/Python.
5. Use your implementation of linear regression to solve the simple problem above and on a small dataset of your choice. Compare the solution with linear regression from R or Python (`sklearn`). Do not forget about the intercept.