# Homework 3

Due April 24, 2020 by 11:59pm

**Instructions**: Upload your answers to the questions below to Canvas. Submit the answers to the questions in a PDF file and your code in a (single) separate file, including for the data competition exercise. Be sure to comment your code to indicate which lines of your code correspond to which question part. There are 3 study assignments and 2 exercises in this homework.

## Reading Assignments

- Review Lecture 3.
- Review Computer Lab. 3 in *canvas.uw.edu/courses/1371621/pages/course-materials* .
- Read and explore *distill.pub/2017/momentum/* .

## 1 Exercise 1

In this exercise, you will implement in **Python** a first version of *your own fast gradient algorithm* to solve the $\ell_2^2$-regularized logistic regression problem.

Recall from the lectures that the logistic regression problem writes as

$$\min_{\beta \in \mathbb{R}^d} F(\beta) := \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \, x_i^T \beta)\right) \; + \lambda \|\beta\|_2^2 \,. \tag{1}$$

We use here the machine learning convention for the labels that is $y_i \in \{-1, +1\}$.

### 1.1 Fast Gradient

The fast gradient algorithm is outlined in Algorithm 1. The algorithm requires a subroutine that computes the gradient for any $\beta$.

- Assume that $d = 1$ and $n = 1$. The sample is then of size 1 and boils down to just $(x, y)$. The function $F$ writes simply as

  $$F(\beta) = \log(1 + \exp(-yx\,\beta) \; + \lambda \beta^2 \,. \tag{2}$$

  Compute and write down the gradient $\nabla F$ of $F$.

- Assume now that $d > 1$ and $n > 1$. Using the previous result and the linearity of differentiation, compute and write down the gradient $\nabla F(\beta)$ of $F$.
- Consider the Spam dataset from *The Elements of Statistical Learning* (You can get it here: `https://web.stanford.edu/~hastie/ElemStatLearn/`). Standardize the data (i.e., center the features and divide them by their standard deviation, and also change the output labels to +/- 1).
- Write a function *computegrad* that computes and returns $\nabla F(\beta)$ for any $\beta$.
- Write a function *backtracking* that implements the backtracking rule.
- Write a function *graddescent* that implements the gradient descent algorithm with the backtracking rule to tune the step-size. The function *graddescent* calls *computegrad* and *backtracking* as subroutines. The function takes as input the initial point, the initial step-size value, and the target accuracy $\varepsilon$. The stopping criterion is $\|\nabla F\| \leq \varepsilon$.
- Write a function *fastgradalgo* that implements the fast gradient algorithm described in Algorithm 1. The function *fastgradalgo* calls *computegrad* and *backtracking* as subroutines. The function takes as input the initial step-size value for the backtracking rule and the target accuracy $\varepsilon$. The stopping criterion is $\|\nabla F\| \leq \varepsilon$.
- Use the estimate described in the course to initialize the step-size. Set the target accuracy to $\varepsilon = 5.10^{-3}$. Run *graddescent* and *fastgradalgo* on the training set of the Spam dataset for $\lambda = 0.5$. Plot the curve of the objective values $F(\beta_t)$ for both algorithms versus the iteration counter $t$ (use different colors). What do you observe?
- Denote by $\beta_T$ the final iterate of your fast gradient algorithm. Compare $\beta_T$ to the $\beta^\star$ found by *scikit-learn*. Compare the objective value for $\beta_T$ to the one for $\beta^\star$. What do you observe?
- Run cross-validation on the training set of the Spam dataset using *scikit-learn* to find the optimal value of $\lambda$. Run *graddescent* and *fastgradalgo* to optimize the objective with that value of $\lambda$. Plot the curve of the objective values $F(\beta_t)$ for both algorithms versus the iteration counter $t$. Plot the misclassification error on the training set for both algorithms versus the iteration counter $t$. Plot the misclassification error on the test set for both algorithms versus the iteration counter $t$. What do you observe?

# 2 Exercise 2

Suppose we estimate the regression coefficients in a logistic regression model by minimizing

$$F(\beta) := \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-y_i\, x_i^T \beta)\right) \; + \lambda\|\beta\|_2^2$$

for a particular value of $\lambda$. For parts (a) through (e), indicate which of (i) through (v) is correct. Justify your answer.

**Algorithm 1** Fast Gradient Algorithm

**input** step-size $\eta_0$, target accuracy $\varepsilon$
**initialization** $\beta_0 = 0$, $\theta_0 = 0$
**repeat** for $t = 0, 1, 2, \ldots$
Find $\eta_t$ with backtracking rule
$\beta_{t+1} = \theta_t - \eta_t \nabla F(\theta_t)$
$\theta_{t+1} = \beta_{t+1} + \frac{t}{t+3}(\beta_{t+1} - \beta_t)$
**until** the stopping criterion $\|\nabla F\| \leq \varepsilon$.

(a) As we increase $\lambda$ from 0, the misclassification error on the training set will:

(i) Increase initially, and then eventually start decreasing in an inverted U shape.
(ii) Decrease initially, and then eventually start increasing in a U shape.
(iii) Steadily increase.
(iv) Steadily decrease.
(v) Remain constant.
(vi) Zigzag in mysterious ways.

(b) Repeat (a) for the misclassification error on a large dataset of unseen data draw from the same probability distribution as the training set.