# Homework 7

Due May 31st, 2019 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. Be sure to comment your code to indicate which lines of your code correspond to which question part.

There are 3 exercises in this homework. Exercise 3 is a milestone for your data competition project.

## 1 Exercise 1

Study Lab #7 and do exercises therein.

## 2 Exercise 2

In this exercise, you will implement in **Python** a first version of your own kernel support vector machine with the smoothed hinge loss.

The kernel support vector machine with the smoothed hinge loss writes as

$$\min_{\alpha \in \mathbb{R}^n} F(\alpha) := \frac{1}{n} \sum_{i=1}^{n} \ell_{hh}(y_i, (K\alpha)_i) + \lambda \alpha^T K \alpha \ , \tag{1}$$

where $(K\alpha)_i$ is the $i$th entry in the vector $K\alpha$,

$$\ell_{\text{hh}}(y,t) := \begin{cases} 0 & \text{if} \quad yt > 1+h \\ \frac{(1+h-yt)^2}{4h} & \text{if} \quad |1-yt| \le h \\ 1-yt & \text{if} \quad yt < 1-h \end{cases} \tag{2}$$

and $h = 0.5$.

You know now by heart the fast gradient algorithm, so no need to recall it here.

- Compute the gradient $\nabla F(\alpha)$ of $F$.
- Write a function *computegram* that computes, for any set of datapoints $x_1, \ldots, x_n$, the kernel matrix $K$.
- Write a function *kerneleval* that computes, for any set of datapoints $x_1, \ldots, x_n$ and a new datapoint $x^\star$, the vector of kernel evaluations $[k(x_1, x^\star), \ldots, k(x_n, x^\star)]^T$.

- Consider the Digits dataset (`http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html`). Download and standardize the data, if you have not done so already.
- Write a function *mysvm* that implements the fast gradient algorithm to train the kernel support vector machine with the smoothed hinge loss. The function takes as input the initial step-size value for the backtracking rule and a stopping criterion based on the norm of the gradient.
- Train your kernel support vector machine with the smoothed hinge loss and the polynomial kernel of order 7 on the the Digits dataset, tuning the regularization parameter $\lambda$ using cross-validation. The $p$th order polynomial kernel is given by $k(x, y) = (x^T y + b)^p$. You may take $b = 1$.
- Compare the performance of kernel SVMs with different kernels (polynomial kernels with different orders, Gaussian RBF with different bandwidths, etc.).

# 3 Data Competition Project

In this exercise, you are going to train support vector machines (SVMs) using the data competition 2 project dataset (with **100** classes). You will consider here *all classes* in the dataset. You may work on this exercise on your own computer first. Note, however, that you **need** AWS to run the experiments for this entire exercise.

- In a one-vs-one fashion, for each pair of classes, train a linear SVM classifier using scikit-learn's function `LinearSVC`, with the default value for the regularization parameter. Compute the *multi-class misclassification error* obtained using these classifiers trained in a one-vs-one fashion.
- In a one-vs-rest fashion, for each class, train a linear SVM classifier using scikit-learn's function `LinearSVC`, with the default value for $\lambda_c$. Compute the multi-class misclassification error obtained using these classifiers trained in a one-vs-rest fashion.
- Redo all questions above now using your own code for the linear SVMs from Exercise 1. Make to sure to run preliminary experiments to decide how to set the stopping criterion to a value that allows the experiments to complete in a reasonable amount of time.