# Assignment 2

Bilal Baig (215734320)
bilalb@yorku.ca

March 1, 2021

**Note:** You have to work individually. You must use the same mathematical notations in textbook or lecture slides to answer these questions. You must use this latex template to write up your solutions. Remember to fill in your information (name, student number, email) at above. No handwriting is accepted. In this assignment, you need to use the *MNIST* data set. Refer to

`https://colab.research.google.com/drive/1FyahMGAE22716sUCrNXPvTrKTwS615Hd`

for how to load it in Python. Direct your queries to Hui Jiang (`hj@eecs.yorku.ca`)

## Exercise 1

**Dimension Reduction**

1. (5 marks) **PCA**: Q4.2 on page 93
2. (5 marks) **LDA**: Q4.4 on page 93
3. (10 marks) **Data visualization**: Lab Project I on page 92, parts a), b) and c)
   Note that you will have to implement PCA and LDA from scratch but you may choose to use a t-SNE implementation from any Python package.

**Your answers:**
**1.**
$e_i = ||x_i - (w^T x_i)w||^2$
**minimum distortion error** $= \frac{1}{n}\sum_{i=1}^{n} e_i$
**2.**
$w* = arg\,max_w\ w^T S_w w$
$1 - w^T S_w w = 0$
$L(w, \lambda) = w^T S_w w + \lambda(1 - w^T S_w w)$
$\frac{\partial L(w,\lambda)}{\partial w} = 0 \Longrightarrow S_b w - \lambda S_w w = 0 \Longrightarrow S_b w = \lambda S_w w \Longrightarrow S_w^{-1} S_b w = \lambda w$
**3.**
*a*)
PCA requires at least 248 dimensions to keep the total variance at 98% or above.
*b*)
LDA can only return K-1 projection directions so in this case it would be only 2.
*c*)

`t-SNE is a lot better at seperating the data than the other two methods, it takes a lot longer but it makes clear distinctions for each digit where the other two have overlaps between digits. There is a noticeable difference between the scales for PCA and LDA, this is likely due to PCA aiming to maximize variance where as LDA looks to maximize separability of the categories using their means, and so it is easier to notice the categories in LDA than it is in PCA.`

# Exercise 2

**Linear Models for Regression**

1. (10 marks) derive the formula to compute the gradients for the following linear models:
   (a) linear regression
   (b) ridge regression
   (c) LASSO

   follow the style of **Algorithm 2.3** (refer to `https://www.overleaf.com/learn/latex/algorithms`) to derive mini-batch stochastic gradient descent algorithms to optimize these models.
2. (20 marks) implement these three algorithms on a small data set, e.g. the Boston Housing Dataset (`https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html`), to predict median value of a home from the 13 attributes. You need to experimentally compare these regression models, discuss your results in terms of how the learning objective function and all learned model weights may differ among three models.

**Your answers:**

1. 1.1.
   $\frac{\partial E(w)}{\partial w} = 0$
   $\implies \frac{\partial}{\partial w}(w^T X^T X w - 2w^T X^T y + y^T y) = 0$
   $\implies 2X^T X w - 2X^T y = 0$
   $\implies w = (X^T X)^{-1} X^T y$

   1.2. $\frac{\partial Q(w)}{\partial w} = 0$
   $\implies \frac{\partial}{\partial w}(w^T X^T X w - 2w^T X^T y + y^T y + \lambda ||w||_2^2) = 0$
   $\implies X^T X w - X^T y + \lambda w = 0$
   $\implies w(X^T X + \lambda I) = X^T y$
   $\implies w = (X^T X + \lambda I)^{-1} X^T y$

   1.3. $\frac{\partial Q(w)}{\partial w} = 0$
   $\implies \frac{\partial}{\partial w}(w^T X^T X w - 2w^T X^T y + y^T y + \lambda ||w||_1) = 0$
   $\implies 2wX^T X - 2X^T y + \lambda sgn(w) = 0$
   $\implies w = (X^T X)^{-1}(X^T y - \lambda sgn(w))$
2. `The objective function for these three regressions are fairly similar, using Linear regression as the "base model" we only have to add a constant lambda*w to get ridge regression and for LASSO we add a constant lambda*sgn(w). LASSO typically took the longest to compute however it was also the most accurate of them. Assuming they all have the same starting random w, then their results are almost the exact same with some very slight differences in their weights.`

# Exercise 3

**Support Vector Machine (SVM)**

1. (10 marks) Q6.8 on page 130

2. (20 marks) use all training data of two digits '5' and '8' from the MNIST dataset to learn two binary classifiers using linear SVM and nonlinear SVM (with Gaussian RBF kernel), and compare and discuss the performance and efficiency of linear SVM and nonlinear SVM methods for these two digits. Report your best results in the test data of '5' and '8'. Don't call any off-the-shelf optimizer. Use the projected gradient descent in Algorithm 6.5 to implement the SVM optimizer yourself.

**Your answers:**

## What to submit?

You must submit:

1. one PDF document (using this latex template) for your solutions to all written questions and all results and discussions for your programming assignments
2. one zip file that includes all of your Python codes and a readme file for TA to run your codes

from eClass before the deadline. No late submission will be accepted.