This take-home portion of the midterm consists of 2 exercises, one of which requires you to use AWS. Recall that there is no collaboration allowed. Copying solutions from elsewhere (e.g., online or in books) is also forbidden. However, you are allowed to post questions in the discussion forum on Canvas.

The submission deadline is **Sunday May 19th, 11:59pm**. No extensions are allowed and no late submissions will be accepted, regardless of the excuse. Please upload to Canvas: (1) An html or pdf file with all of your output and comments; and (2) The code file(s), which may have the extension .ipynb or .py.

1 Exercise 1

In this exercise, you will implement in **Python** a first version of your own ℓ_2^2-regularized binary logistic regression with ρ-logistic loss. You will write your own codes for all functions: accelerated gradient algorithm, ℓ_2^2-regularized binary logistic regression with ρ-logistic loss, leave-one-out cross-validation, and hold-out cross-validation.The ℓ_2^2-regularized binary logistic regression with ρ-logistic loss is a supervised binary classification method, similar to ℓ_2^2-regularized binary logistic regression.

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

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

- Compute the gradient $\nabla F(\beta)$ of F.
- Consider the Spam dataset from *The Elements of Statistical Learning*. Standardize the data, if you have not done so already. Be sure to use the training and test splits from the website. You can find the link to the train/test split here: `https://web.stanford.edu/~hastie/ElemStatLearn/data.html`
- Write a function *myrhologistic* that implements the accelerated gradient algorithm to train the ℓ_2^2-regularized binary logistic regression with ρ-logistic loss. The function takes as input the initial step-size for the backtracking rule, the ε for the stopping criterion based on the norm of the gradient of the objective, and the value of ρ.
- Train your ℓ_2^2-regularized binary logistic regression with ρ-logistic loss with $\rho = 2$ and $\varepsilon = 10^{-3}$ on the the Spam dataset for the $\lambda = 1$. Report your misclassification error for this value of λ.
- Write a function *crossval* that implements leave-one-out cross-validation and hold-out cross-validation. You may either write a function that implements each variant separately depending on the case, or write a general cross-validation function that can be instantiated in each case.
- Find the optimal value of λ using leave-one-out cross-validation. Find the optimal value of λ using hold-out cross-validation with a 80%/20% split for the training set/testing set. Report your misclassification errors for the two values of λ found.

2 Data Competition Project

Read the announcement "Data Competition, Part 1" released on Canvas. We strongly recommend you perform this task on AWS. You will use *ℓ_2^2-regularized binary logistic regression with ρ-logistic*

loss for the purpose of supervised binary classification in this exercise. After completing this exercise, submit your predictions to the data competition Kaggle website.

- Pick two classes of your choice from the dataset. Train a classifier using ℓ_2^2-regularized binary logistic regression with ρ-logistic loss on the training set using your own accelerated gradient algorithm with $\rho = 2$, $\varepsilon = 10^{-3}$, and $\lambda = 1$. Be sure to use the features you previously generated with the provided script rather than the raw image features. Plot, with different colors, the *misclassification error* on the training set and on the validation set vs iterations.

- Find the value of the regularization parameter λ using using leave-one-out cross-validation. Find the value of the regularization parameter λ using using hold-out cross-validation. Train a classifier using ℓ_2^2-regularized binary logistic regression with ρ-logistic loss on the training set using your own accelerated gradient algorithm with that value of λ found by hold-out cross-validation. Plot, with different colors, the *misclassification error* on the training set and on the validation set vs. iterations.

- Consider all pairs of classes from the dataset. For each pair of classes, train a classifier using a ℓ_2^2-regularized binary logistic regression with ρ-logistic loss on the training set comprising only the data-points for that pair of classes using your own fast gradient algorithm. For each pair of classes, find the value of the regularization parameter λ using hold-out cross-validation on the training set comprising only the data-points for that pair of classes.

- Write a function that for any new data point predicts its label. To do this, you will perform the following: input the data point into each classifier (for each pair of classes) you trained above. Record the class predicted by each classifier. Then your prediction for this data point is the most frequently predicted class. If there is a tie, randomly choose between the tied classes. Report the misclassification error on the validation set and test set. Report the precision/recall on the validation set.