Introduction to Machine Learning

Home Assignment 1

This homework is due by **April 22, 2020**. It is to be returned by email to raphael.berthier@inria.fr as a **pdf** report of **maximum 3 pages** together with the ipython notebook used for the code. The results and the figures must be included into the pdf report but not the code.

The goal of this project is to automatically classify letters from different computer fonts. An example of samples of the letter "A" can be seen below.

The data comes from the `notMNIST` dataset and can be downloaded at `http://www.di.ens.fr/appstat/spring-2020/project/data.zip`. The zip archive contains two folders:

- train: contains $n = 6\,000$ labelled images of three classes "A", "B" and "C" (2 000 each)
- test: contains $n_1 = 750$ labelled images (250 for each of the three classes).

The train folder will be used to train the forecasting methods. The test folder will be used to assess their performance. If for some reasons, the datasets are too large to be used on your computer, you can use subsets of with n and n_1 sufficiently small to be computable but large enough to get prediction accuracy.
The goal is to classify if an image X_i corresponds to the letter "A": i.e., the output is $Y_i = 1$ if image i is "A" and -1 otherwise (if the image is "B" or "C").

1. Formalize the problem by defining the input space $\mathcal{X}$, the output space $\mathcal{Y}$ and the training data set. What are their dimension?

2. If $f : \mathcal{X} \to \mathcal{Y}$ is a predictor from images to $\mathcal{Y} = \{-1, 1\}$, we define for a couple image/label (X_i, Y_i):
   - the 0-1 loss: $\ell_1(f(X_i), Y_i) = \mathbb{1}_{f(X_i) \neq Y_i}$
   - the square loss: $\ell_2(f(X_i), Y_i) = \big(f(X_i) - Y_i\big)^2$
   - the logistic loss: $\ell_3(f(X_i), Y_i) = \log\big(1 + e^{-Y_i f(X_i)}\big)$.

   (a) What are the empirical risk (training error) and the true risk associated with the 0-1 loss? Why is it complicated to minimize the empirical risk in this case?

   (b) Why should we use the test data to assess the performance?

   (c) Recall the definition of the optimization problems associated with the linear least square regression and the linear logistic regression.

3. Implement the gradient descent algorithm (GD) and the stochastic gradient descent algorithm (SGD) to solve these two minimization problems.

(a) Consider the logistic regression minimization problem. Plot the training errors and the test errors as functions of the number of access to the data points[1] of GD and SGD for well-chosen (by hand) values of the step sizes.

(b) Plot the estimators $\hat{\beta}_n^{\text{logist}} \in \mathbb{R}^{28\times 28}$ and $\hat{\beta}_n^{\text{lin}} \in \mathbb{R}^{28\times 28}$ respectively associated with the logistic and linear regression as two images of size 28×28.

(c) Denote by $\hat{\beta}_n^{\text{logist}}(t) \in \mathbb{R}^{28\times 28}$ the estimator of logistic regression after t gradient iterations of SGD. Plot as images the averaged estimators $\bar{\beta}_n^{\text{logist}}(t) = \frac{1}{t}\sum_{s=1}^{t} \hat{\beta}_n^{\text{logist}}(s) \in \mathbb{R}^{28\times 28}$ for $t \in \{10, 100, 1\,000, 10\,000\}$. Repeat for the linear regression estimator.

4. (a) Recall the definition of the k-nearest neighbors classification rule with ℓ_2 metric.

(b) Implement it and plot as a function of k, its training and test errors[2].

(c) Calibrate k using K-fold cross-validation with $K = 5$.

5. Fill the following table and comment:

	Logistic regression	Linear regression	K-NN
Empirical error (0-1 loss)			
Test error (0-1 loss)			

[1] One iteration of gradient descent needs to read n data points. One iteration of SGD only needs 1 sample per iteration.

[2] Note that in practice the standard bias-variance trade-off is sometimes hard to observe and this plot hard to interpret. One explanation is that K-NN with ℓ_2 metric is not well suited for high dimensional problems and is provide poor predictive performance.