 M. Arbel, P. Gaillard, J. Mairal

From Basic Machine Learning models to Advanced Kernel Learning

Home Assignment 1

This homework should be uploaded by **November 24, 2023 at 23:59pm** as a **pdf report** together with a **code file (.py or .ipynb)** on the website

`http://pierre.gaillard.me/teaching/kernel_mosig_2023.php`

The password to upload is `kernel2023`. 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 by implementing from scratch a few simple models. Basic python librairies for linear algebra, sampling, or cross-validation may be used but not already fully implemented algorithms for SGD, linear regression or logistic regression.

An example of samples of the letter "A" can be seen below.

The data comes from the `notMNIST` dataset and can be downloaded at `https://kernel-learning.github.io/docs/data1.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 $f_\theta : \mathcal{X} \to \mathcal{Y}$ 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: $\ell_1(f_\theta(X_i), Y_i) = \mathbb{1}_{f_\theta(X_i) \neq Y_i}$
   - the square loss: $\ell_2(f_\theta(X_i), Y_i) = \big(f_\theta(X_i) - Y_i\big)^2$
   - the logistic loss: $\ell_3(f_\theta(X_i), Y_i) = \log\big(1 + e^{-Y_i f_\theta(X_i)}\big)$.

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

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

(c) What is the probability of $\mathbb{P}(Y = 1|X)$ under the logistic model?

3. Assuming that $f_\theta(x) = \langle \theta, x \rangle$. Write the update rule of SGD (for minibatches of size 1) for

   (a) the linear least-squares regression

   (b) the logistic regression

   (c) the perceptron algorithm.

4. Implement from scratch the stochastic gradient descent algorithm (SGD) with minibatch of size 1 to solve these problems.

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

   (b) Denote by $\hat{\theta}_n^{\text{logist}}(t) \in \mathbb{R}^{28\times 28}$ the estimator of logistic regression after t gradient iterations of SGD. Plot as images the estimators $\hat{\theta}_n^{\text{logist}}(t) \in \mathbb{R}^{28\times 28}$ for $t \in \{10, 100, 1\,000, 10\,000\}$. Repeat for the linear least squares regression (OLS) and perceptron.

5. k-Nearest Neighbors (KNN).

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

   (b) Implement it from scratch and plot as a function of k, its training and test errors.

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

6. Multi-layer Perceptron (MLP). Given a multi-layer perceptron with 1 input layer containing 28×28 neurons, 1 hidden layer containing 32 neurons, and 1 output layer containing 3 neurons (one for each class), how many parameters need to be trained? Implement it with ReLu activation function by using your favorite library (here you can use a function that is already implemented).

7. Fill the following table and comment:

	Logistic regression	OLS	Perceptron	KNN	MLP
Empirical error (0-1 loss)					
Test error (0-1 loss)					

8. Why is it often important to regularize? What would be the updates of the three models of question 5) with

   (a) ℓ_2 regularization $\lambda\|\theta\|_2^2$?

   (b) (optional) ℓ_1 regularization $\lambda\|\theta\|_1$?