Due: Tuesday, September 29th, 2015, at noon (Submit via NYU Classes)

Preparation:

• install the software packages and material related to the Scikit-Learn lecture available here https://github.com/amueller/nyu_ml_lectures, as described in the piazza announcement “To do before Sep 16 class: install software packages for intro. to Scikit-Learn”

• go through the material of Sec. 01.0-01.1-01.2-01.3-01.4 and 02.1 of the Scikit-Learn lecture, do and run all the exercises, and play with the codes and plots

This preparation is vital to be able to work on the assignment. The corresponding Python codes should however not be included in your submission. If you encounter any impediment, please post your questions on piazza.

Instructions: Your Python code, including the parts dedicated to the plots, should be submitted as a single “.py” file.

1 Prologue

In this homework, you will implement the Perceptron algorithm presented during the lecture, and a variant of the Perceptron algorithm related to the stochastic gradient algorithm. All the support Python code needed is presented in Sections 01.0 to 01.4 and Section 02.1 of the Scikit-Learn lecture to get you started on the right track. Take advantage of this being a relatively straightforward assignment by exploring further Python, NumPy, and Scikit-Learn and pursuing independent investigations by playing with the codes and plots. For example, experiment with several datasets, other than the Iris dataset, such as the other datasets covered in the Scikit-Learn lecture (Faces datasets, etc.) and simulated synthetic datasets you could come up with. Include any investigations you do in your submission, and we may award up to 5% extra credit.

I encourage you to develop the habit of asking “what if?” questions and then seeking the answers. I guarantee this will give you a much deeper understanding of the material (and likely better performance on the exam and job interviews, if that’s your focus). You’re also encouraged to post your interesting questions on Piazza under “questions.”

Now have fun, and go learn!
2 Perceptron

In this section, you will focus on the so-called “Batch Perceptron” algorithm, described in Sec. 9.1.2 of ?.

Write a function `batch_perceptron` that implements the Batch Perceptron algorithm. The function takes as input the following:

- the training set \((x_1, y_1), \ldots, (x_m, y_m)\), where \(x_i \in \mathbb{R}^d\) and \(y_i \in \{-1, +1\}\) for all \(i = 1, \ldots, m\)
- the maximum number of iterations `max_iter`, set to 10,000

The learning parameter \(w\) is initialised at \(w = (0, \ldots, 0)\) the null vector of \(\mathbb{R}^d\). The function returns as output the learning parameter \(w\) after `max_iter` iterations.

Set the seed of the random number generator to a particular value, corresponding to a date of your choice. For instance, run `random.seed(09162015)` if the date is Sep. 16th 2015. Of course, do not use that date, just pick a date that is special to you.

Following Sec. 01.4 of the Scikit-Learn lecture, load the Iris dataset and split the data into two sets, respectively the training set and the testing set. The split uses 50% of the data as the training set, and 50% as testing set. From now on, you will use this particular split in all your experiments.

Run the Batch Perceptron algorithm on the training set of the Iris dataset you just created. Plot the evolution of the misclassification error on the training set versus the iteration counter, as the Perceptron algorithm proceeds while working on the training set. What do you observe? How does the misclassification error curve on the training set look like? What kind of phenomenon do you observe in the last iterations? Describe what you observe as a comment in your code.

Run the Batch Perceptron algorithm on the testing set of the Iris dataset you just created. Plot the evolution of the misclassification error on the testing set versus the iteration counter, as the Perceptron algorithm proceeds while working on the training set. What do you observe? How does the misclassification error curve on the testing set look like? How does it compare to the previous curve? Describe what you observe as a comment in your code.

Plot the two curves in one single plot. Do you think it is necessary to run the algorithm for 10,000 iterations? What would happen if you could stop the algorithm earlier? Describe what you observe as a comment in your code.

3 Modified Perceptron

In this section, you will focus on a modified version of the “Batch Perceptron” algorithm, described in Exercise 9.5 of Sec. 9.6 of ?.

Write a function `modif_perceptron` that implements the Modified Batch Perceptron algorithm, described in Exercise 9.5 of Sec. 9.6 of ?. The function takes as input the following:

- the training set \((x_1, y_1), \ldots, (x_m, y_m)\), where \(x_i \in \mathbb{R}^d\) and \(y_i \in \{-1, +1\}\) for all \(i = 1, \ldots, m\)
- the maximum number of iterations `max_iter`, set to 10,000
- the parameter \(\eta\)
The learning parameter $w$ is initialised at $w = (0, \ldots, 0)$ the null vector of $\mathbb{R}^d$. The function returns as output the learning parameter $w$ after $\max\_iter$ iterations.

In the following, you will use the same training-testing split as in the previous section.

Interpret the impact of the parameter $\eta$, based on experimental results when varying $\eta$. When $\eta = 1$, to what algorithm is the Modified Batch Perceptron algorithm equivalent to? (hint: look at the previous section). Try increasing the value of $\eta$, try decreasing the value $\eta$, and plot the two curves (the evolution of the misclassification error on the training set vs the iteration counter, and the evolution of the misclassification error on the testing set). Does it seem that there are better values of $\eta$ than others? In what sense? (hint: recall the only thing you care about, in fine) Describe what you observe as a comment in your code.

4 Feedback (not graded)

1. Approximately how long did it take to complete this assignment?

2. Did you find the Python programming challenging? Did you have any impediment?

3. Any other feedback?