

# CS 434: Assignment 2

Due April 21th 11:59PM, 2017

General instructions.

1. The following languages are acceptable: Java, C/C++, Matlab, Python and R.
2. You can work in team of up to 3 people. Each team will only need to submit one copy of the source code and report.
3. You need to submit your source code (self contained, well documented and with clear instruction for how to run) and a report via TEACH. In your submission, please clearly indicate your team members' information.
4. Be sure to answer all the questions in your report. Your report should be typed, submitted in the pdf format. You will be graded based on both your code as well as the report. In particular, the clarity and quality of the report will be worth 10 % of the pts. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables.

## 1 Logistic regression with $L_2$ regularization

In this assignment you will play with the USPS handwritten digit dataset and implement the logistic regression classifier to recognize digit 4 from digit 9. The data set for this assignment can be downloaded from the class web site. This dataset contains hand-written digits 4 and 9. Each digit example is an image of by 16 by 16 pixels. Treating the gray-scale value of each pixel as a feature (between 0 and 255), each example has  $16 \times 16 = 256$  features. For each class, we have 700 training samples and 400 testing samples. You can view these images collectively at [http://www.cs.nyu.edu/~roweis/data/usps\\_4.jpg](http://www.cs.nyu.edu/~roweis/data/usps_4.jpg) and [http://www.cs.nyu.edu/~roweis/data/usps\\_9.jpg](http://www.cs.nyu.edu/~roweis/data/usps_9.jpg).

The data is in csv format and each row corresponds to a hand-written digit image (the first 256 columns) and its corresponding label (last column, 0 for digit 4 and 1 for digit 9). Note that you can use the MATLAB command `imshow` to view the image of a particular training/testing example. Say `x` is a row vector of 256 dimensions for a particular digit image, `imshow(reshape(x,16,16))` allows you to see the image.

1. (30 pts) Implement the batch gradient descent algorithm to train a binary logistic regression classifier. The behavior of Gradient descent can be

strongly influenced by the learning rate. Experiment with different learning rates, report your observation on the convergence behavior of the gradient descent algorithm. For your implementation, you will need to decide a stopping condition. You might use a fixed number of iterations, the change of the objective value (when it ceases to be significant) or the norm of the gradient (when it is smaller than a small threshold). Note, if you observe an overflow, then your learning rate is too big, so you need to try smaller learning rates.

2. (15 pts) Once you identify a suitable learning rate, rerun the training of the model from the beginning. For each gradient descent iteration, plot the training accuracy and the testing accuracy of your model as a function of the number of gradient descent iterations. What trend do you observe?
3. (15 pts) As discussed in class, Logistic regression is typically used with regularization. We will explore  $L_2$  regularization for this question. In particular, we will use the following objective with an additional regularization term that is equal to the squared Euclidean norm of the weight vector.

$$L(w) = \sum_{i=1}^n l(g(w^T x^i), y^i) + \frac{1}{2} \lambda \|w\|_2^2$$

where the loss function  $l$  is the same as introduced in class (slide 7 of the logistic regression notes). Find the gradient for this objective function and modify the batch gradient descent algorithm with this new gradient. Provide the pseudo code for your modified algorithm.

4. (30 pts) Implement the algorithm in [3], and experiment with different  $\lambda$  values (e.g.,  $10^{-3}$ ;  $10^{-2}$ ;  $\dots$ ;  $10^3$ ). Report the training and testing accuracies achieved by the weight vectors learned with different  $\lambda$  values. Discuss your results in terms of the relationship between training/testing performance and the  $\lambda$  values.

**Remark 1.** *Producing an image of weights may help you to interpret the results.*