CMPUT 466/551—Machine Learning

Assignment 3

Winter 2004
Department of Computing Science
University of Alberta

Due: in class, *Thursday, March 4*
Worth: 15% of final grade
Instructor: Dale Schuurmans, Ath409, x2-4806, dale@cs.ualberta.ca

---

Note This assignment requires you to write two Matlab programs which are to be submitted by email. When finished, please send a *single* tar file containing all of your .m files **to the TA, Qiongyun Zhang, at qiongyun@cs.ualberta.ca** with a subject heading "CMPUT 466/551 A3 solutions". The write-up should be printed out in hard copy and handed in at the start of class on March 4.

In this assignment you will design your own learning and classification algorithms for recognizing images of handwritten digits. On the course webpage download the file data3.mat. Then type "`load data3.mat`" in Matlab. This will load the training data into a matrix `X` and a vector `y`. Each row of `X` corresponds to a 256 dimensional vector representing a 16×16 grayscale image of a handwritten digit. The images are of handwritten digits from 0 to 9. The corresponding entry in the associated y-vector gives a label indicating which digit the image represents.

Note: that you can easily view the training images in Matlab by first typing "`colormap gray`", and then viewing image `i` in matrix `X` by typing "`imagesc(reshape(X(i,:),16,16))`".

The goal of this assignment is to design a learning algorithm which can examine the training data and learn a classifier that can accurately recognize which digit is depicted in an image. Your learning and classification algorithms will be tested on data that you will not see until after the assignment is submitted.

Question 1 (Learning algorithm and classifier)

The goal of this assignment is to get you to design a learning algorithm for a given application (handwritten digit recognition) in a realistic setting: you will only be given a training set and have to figure out for yourself how to learn an accurate classifier from the limited amount of data available.

You can use any learning algorithm you wish, even ones that we have not specifically studies in this course. You may even pre-process and/or filter the data as part of your learning algorithm. I expect that you will find the ideas of nonlinear basis functions, complexity control (cross validation), and error correcting output codes to be particularly useful concepts—although you need not use any of these. Some of you may also want to race ahead and try ensemble methods like "boosting" and "bagging". It is completely up to you to try whatever approach you would like. The only constraint is that you have to be able to implement your technique, and the algorithms should not take an overly excessive amount of time to run (nor require the TAs to set parameters).

(a) (5%) Write a Matlab function `[model] = learn(X,y)` which takes a $t \times n$ matrix $\mathbf{X}$ and $t \times 1$ vector of target labels `y` and returns a `model`. The `model` can be any representation you wish, as long as you can use it to classify new test data.

Also write a Matlab function `[yhat] = classify(Xtest,model)` which takes a $te \times n$ matrix `Xtest` and a `model` produced by your program `learn`, and returns a $te \times 1$ vector of classifications `yhat` on the test patterns. The classifications in `yhat` must be chosen from the set $\{0, 1, ..., 9\}$.

Your functions must be able to handle arbitrary n, t and te.

Question 2 (Performance—accuracy and efficiency)

(a) (5%) We will run your submitted program `learn` on training data, and test the resulting on classifier on test data using your `classify` function. Your programs will be evaluated according to their classification accuracy on new test data, as well as their run time and space efficiency. Mainly though, we will focus on accuracy.

Question 3 (Explanation)

(a) (5%) In short write-up, please explain how your learning algorithm works and why you chose to design it the way you did. Your write-up should be at least 2 pages, but no more than 10 pages in length.