# COMP 652 and ECSE 608: Machine Learning - Assignment 1

## Posted Thursday, January 19, 2017
## Due Thursday, February 2, 2017

You should submit an archive of your code, as well as a pdf file with your answers (either typed or scanned), uploaded to MyCourses. If you cannot access MyCourses, let us know ASAP. Assignments must be submitted by 11:59pm EST one the day the assignment is due in order to incur no penalties.

1. [55 points] **Regression**

   For this exercise, you will experiment with regression, regularization and cross-validation. You are provided with a set of data in files hw1x.dat and hw1y.dat. You are allowed to use any programming language of your choice (Python, Matlab, R, etc).

   (a) [5 points] Load the data into memory. Make an appropriate $\mathbf{X}$ matrix and $\mathbf{y}$ vector. Do not forget to add a constant input equal to 1!

   (b) [5 points] Split the data at random into one set $(\mathbf{X}_{train}, \mathbf{y}_{train})$ containing 80% of the instances, which will be used for training + validation, and a testing set $\mathbf{X}_{test}, \mathbf{y}_{test})$ (containing remaining instances).

   (c) [5 points] Give the objective of logistic regression with $L_2$ regularization.

   (d) [5 points] Run logistic regression on the data using $L_2$ regularization, varying the regularization parameter $\lambda \in \{0, 0.1, 1, 10, 100, 1000\}$. Plot on one graph the average cross-entropy for the training data and the testing data (averaged over all instances), as a function of $\lambda$ (you should use a log scale for $\lambda$). Plot on another graph the $L_2$ norm of the weight vector you obtain. Plot on the third graph the actual values of the weights obtained (one curve per weight). Finally, plot on a graph the accuracy on the training and test set. Explain briefly what you see.

   (e) [5 points] Re-format the data in the following way: take each of the input variables, and feed it through a set of Gaussian basis functions, defined as follows. For each variable (except the bias term), use 5 univariate basis functions with means evenly spaced between -10 and 10 and variance $\sigma$. You will experiment with $\sigma$ values of $0.1$, $0.5$, $1$, $5$ and $10$.

   (f) [5 points] Using no regularization and doing regression with this new set of basis functions, plot the training and testing error as a function of $\sigma$ (when using only basis functions of a given $\sigma$). Add constant lines showing the training and testing error you had obtained in part c. Explain how $\sigma$ influences overfitting and the bias-variance trade-off.

   (g) [5 points] Add in *all* the basis function and perform regularized regression with the regularization parameter $\lambda \in \{0, 0.1, 1, 10, 100, 1000, 10000\}$. Plot on one graph the average cross-entropy error for the training data and the testing data, as a function of $\lambda$ (you should use a log scale for $\lambda$). Plot on another graph the $L_2$ norm of the weight vector you obtain. Plot on a different graph the $L_2$ norm of the weights for the set of basis functions corresponding to each value of $\sigma$, as a function of $\lambda$ (this will be a graph with 5 lines on it). Explain briefly the results.

   (h) [5 points] Explain what you would need to do if you wanted to design a set of Gaussian basis functions that capture relationships between the inputs. Explain the impact of this choice on the bias-variance trade-off. No experiments are needed (although you are welcome to explore this on your own).

(i) [10 points] Suppose that instead of wanting to use a fixed set of evenly-spaced basis functions, you would like to adapt the placement of these functions. Derive a learning algorithm that computes both the placement of the basis function, $\mu_i$ and the weight vector $\mathbf{w}$ from data (assuming that the width $\sigma is fixed$. You should still allow for $L_2$ regularization of the weight vector. Note that your algorithm will need to be iterative.

(j) [5 points] Does your algorithm converge? If so, does it obtain a locally or globally optimal solution? Explain your answer.

2. [35 points] **Kernelized logistic regression**

   (a) [10 points] Following the same reasoning as discussed in class for linear regression, derive a dual view for the logistic regression algorithm (in which we have parameters associated with each point). More specifically, take the cross-entropy error function that logistic regression is optimizing. Instead of using a vector of parameters $\mathbf{w}$, you will use $\mathbf{\Phi}^T\mathbf{a}$, where $\mathbf{a}$is a new parameter vector with one entry for every instance. Then derive a learning algorithm which optimizes the error function with respect to $\mathbf{a}$. If you want to use regularization, use the $L_2$ norm. Note that in this case there is no closed-form solution, you will instead write a gradient-based update rule.

   (b) [10 points] Provide an implementation of your algorithm using a polynomial kernel, $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z} + 1)^d$, for the dataset given in hw1x.dat and hw1y.dat (where $d$ is the kernel order).

   (c) [10 points] Experiment with plain logistic regression and polynomial logistic regression with $d = 1, 2, 3$ on this data set, performing cross-validation as before. Report the training and testing cross-entropy and accuracy. Based on this data, explain the effect of the width on the approximator.

   (d) [5 points] What are the advantages and disadvantages of kernelized logistic regression over the usual version, based on your investigation? Explain your answer.

3. [10 points] **Kernels**

   In this problem, we consider constructing new kernels by combining existing kernels. Recall that for some function $K(\mathbf{x}, \mathbf{z})$ to be a kernel, we need to be able to write it as a dot product of vectors from some high-dimensional feature space:

   $$K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z})$$

   Mercer's theorem gives a necessary and sufficient condition for a function $K$ to be a kernel: its corresponding kernel matrix has to be symmetric and positive semidefinite.

   Suppose that $K_1(\mathbf{x}, \mathbf{z})$ and $K_2(\mathbf{x}, \mathbf{z})$ are kernels over $\mathbf{R}^n \times \mathbf{R}^n$. For each of the cases below, state whether $K$ is also a kernel. If it is, prove it. If it is not, give a counterexample. You can use either Mercer's theorem, or the definition of a kernel as needed.

   (a) $K(\mathbf{x}, \mathbf{z}) = aK_1(\mathbf{x}, \mathbf{z}) + bK_2(\mathbf{x}, \mathbf{z})$, where $a, b > 0$ are real numbers

   (b) $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z})K_2(\mathbf{x}, \mathbf{z})$