1. [60 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 hw2x.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 $X$ matrix and $y$ vector. Do not forget to add a constant input equal to 1!

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

(c) [5 points] Run linear 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 root mean squared 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 the third graph the actual values of the weights obtained. Explain briefly what you see.

(d) [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 10 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.

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

(f) [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 root mean squared 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.

(g) [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).
(h) [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 $w$ 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.

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

(j) [10 points] Fix $\sigma = 1$. Design and run an experiment comparing the solution obtained by your algorithm to the solution obtained using the fixed set of bases. You will need to include at least one graph showing the behaviour of your algorithm, and one comparison of the final result obtained. Explain what you see in the results.

2. [25 points] Regularization

Suppose that we have a training set $\langle x_i, y_i \rangle, i = 1 \ldots m$ of $m$ i.i.d. examples. In class we discussed that minimizing the mean squared error corresponds to an assumption that the labels of the data came from some target hypothesis $h_w$, but then were observed after being perturbed by Gaussian noise, with the noise variables drawn i.i.d. from the same distribution.

Suppose now that the standard deviation with which we observe the label of example $i$ is given by:

$$
\begin{cases} 
\sigma_0 & \text{if } ||x_i||_2 \leq R \\
\sigma_1, & \text{otherwise}
\end{cases}
$$

with $\sigma_0 < \sigma_1$ and $R$ a given real number.

- [5 points] Derive the maximum likelihood estimate of $w$ in this case.
- [5 points] Explain how this estimate would differ from the one obtained by doing regression as discussed in class (on a data set that conforms to our new assumption).
- [10 points] Suppose now that we have a different $\sigma_i$ for each example $i$, given by $\sigma_i = c||x_i||_2$ (where $c$ is a real-valued constant). Derive a maximum likelihood estimate for $w$ under this assumption, or an algorithm for computing such an estimate.
- [5 points] How do these changes affect the graphical model discussed at the end of lecture 1? Explain your answer.

3. [15 points] Using other types of error functions

Suppose that we have a regression problem in which we want to use a penalty which is differentiable, but which does not penalize large errors too much. One possibility is to use a “huberized loss”, which behaves like the squared error around 0, but like a linear error away from 0. More precisely, for an example $\langle x_i, y_i \rangle$, this loss would be:

$$
L_H(w, \delta) = \begin{cases} 
(y_i - w^T x_i)^2/2 & \text{if } |y_i - w^T x_i| \leq \delta \\
\delta |y_i - w^T x_i| - \delta^2/2 & \text{otherwise}
\end{cases}
$$

(a) [10 points] Derive a linear regression algorithm using this loss and $L_2$ regularization.

(b) [5 points] Explain the effect of $\delta$ on this loss. Give an example of a data set in which this loss might be preferable to the usual squared loss.