COMP 6321, Fall 2016 **CONCORDIA UNIVERSITY**

MACHINE LEARNING.
Assignment 1.

Due: Sept. 30, 2016.

1. [65 points] **Linear and polynomial regression**
For this exercise, you will experiment in Matlab with linear and polynomial regression on a given data set. The inputs are in the file hw1x.dat and the desired outputs in hw1y.dat.

   (a) [5 points] Load the data into memory and plot it (using the load and plot functions; use the help function if you do not know how to call them).

   (b) [5 points] Add a column vector of 1s to the inputs, then use the linear regression formula discussed in class to obtain a weight vector w. Plot both the linear regression line and the data on the same graph. (Note: matrix formulas translate almost verbatim in Matlab)

   (c) [5 points] Write a Matlab function that will evaluate the training error of the resulting fit, and report what this error is.

   (d) [5 points] Write a Matlab function called PolyRegress(x,y,d) which adds the features $x^2, x^3, ...x^d$ to the inputs and performs polynomial regression.

   (e) [5 points] Use your function to get a quadratic fit of the data. Plot the data and the fit. Report the training error. Is this a better fit?

   (f) [5 points] Repeat the previous exercise for a cubic fit.

   (g) [5 points] Suppose that the data were sorted in increasing value of the target variable y, and you simply partitioned it by putting the first m=k examples in the first fold, the next ones in the second fold, etc. Explain what would happen if you tried to perform cross-validation with these folds.

   (h) [10 points] Write a procedure that performs five-fold cross-validation on your data. Use it to determine the best degree for polynomial regression. Show the data that supports your conclusion, and explain how you have come to this conclusion. For the best fit, plot the data and the polynomial obtained.

   (i) [10 points] Change the Matlab code such that you normalize the input data in each column by the maximum absolute value in that column. What is the best degree for polynomial regression now? Justify your answer.

   (j) [10 points] As you witnessed, polynomial regression often causes the features to get extreme values, which may cause numerical problems. In such cases, it can be helpful to normalize the features, e.g. by dividing the value of each feature x_j by $\max_i |x_{i,j}|$, like you did in the example above. Prove that this change results in a scaling of the output, but has no other effect on the approximator.

2. [25 points] **Weighted linear regression**
Sometimes we might want to do linear regression but weight the different training

examples differently. This is the case, for instance, if we believe that some examples are more important than others, or that some examples are less prone to noise. In particular, suppose we want to minimize the following error function:

$$J(\mathbf{w}) = \sum_{i=1}^{m} u_i(\mathbf{w}^T\mathbf{x_i} - y_i)^2$$

where $\mathbf{x_i} \in \mathbb{R}^n, u_i \in \mathbb{R}, i = 1, \ldots, m, \mathbf{w} \in \mathbb{R}^n$ are training samples, weights of training samples and model weights, respectively.

(a) [5 points] Show that this can be re-written in matrix form as:

$$J(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^T\mathbf{U}(\mathbf{X}\mathbf{w} - \mathbf{y}).$$

Clearly state what $\mathbf{U}$ is.

(b) [5 points] Compute $\nabla_{\mathbf{w}} J(\mathbf{w})$. Set this to 0 and solve for the parameter vector $\mathbf{w}$. You should obtain a generalization of the formula we derived in class, with $\mathbf{w}$ as a function of $\mathbf{X}, \mathbf{y}$ and $\mathbf{U}$. Check that for the case in which all weights are equal to 1, you get the same formula as for linear regression.

(c) [10 points] Implement weighted linear regression for the data set used in question 1. Weight all points equally, except the point with the largest input value. Gradually increase the weight of this point. Describe what happens, and why.

(d) [5 points] Draw an example of a data set in which you would expect weighted linear regression to work a lot better than the unweighted version. Explain why you chose this data set.

3. [10 points] **Error criterion for exponential noise**
At the end of Lecture 2, we showed that one justification for minimizing the squared-error in a regression problem is probabilistic: we obtain the hypothesis under which the data has maximum likelihood if we assume that the target values are generated by a hypothesis from the same class, but perturbed by additive Gaussian noise. Now, suppose that the noise variables were not Gaussian but rather exponentially distributed. Recall that the exponential distribution has a single parameter, λ, and its density has the formula:

$$p_\lambda(t) = \begin{cases} \lambda e^{-\lambda t} & \text{if } t \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Using a similar derivation to the one in class, show that under this assumption, the hypothesis that maximizes the likelihood of the data is no longer the one that minimizes the squared error. Derive the error criterion minimized in this case. Show your final result, as well as the derivation.