HW Assignment 1 (Due date: Sep 21, Wednesday)

1 Problems

1. [**Regularized Least Squares**, 20 points]
   Exercise 1.2, page 58.

2. [**Probability Theory**, 20 points]
   Exercise 1.3, page 58.

3. [**Fisher Criterion**, 20 points]
   Exercise 4.5, page 221.

4. [**Linear separators**, 40 points]
   Consider the following training set of 2-dimensional examples with binary labels:
   $D = \{(\mathbf{x}_1, t_1)(\mathbf{x}_2, t_2)(\mathbf{x}_3, t_3), (\mathbf{x}_4, t_4)\} = \{([0,0],+), ([1,1],+), ([0,1],-), ([1,0],-)\}$.

   (a) Prove that the dataset D is not linearly separable.

   (b) What is the minimum size for a training set of 3-dimensional non-coplanar points that is not linearly separable? Provide an example and a proof.

5. [**Linear Regression**, 100 points]
   In this exercise, you are asked to run an experimental evaluation of a linear regression model, with and without regularization. The input data is available at `http://ace.cs.ohio.edu/~razvan/courses/ml6830/hw01.tar.gz`.

   (a) Select 30 values for $x \in [0,1]$ uniformly spaced, and generate corresponding t values according to $t(x) = sin(2\pi x) + x(x+1)/4 + \epsilon$, where $\epsilon = N(0, 0.005)$ is a zero mean Gaussian with variance 0.005. Save and plot all the values. Done in `dataset.dat`.

   (b) Split the 30 samples (x_n, t_n) in three sets: 10 samples for training, 10 samples for validation, and 10 samples for testing. Save and plot the 3 datasets separately. Done in `train.dat, test.dat, valid.dat`.

   (c) Consider a linear regression model with polynomial basis functions in which the sum of squared errors is normalized by the number of training examples, as shown below:

   $$E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} \left(y(x_n, \mathbf{w}) - t_n\right)^2 + \frac{\lambda}{2} ||\mathbf{w}||^2$$

   Show the closed form solution (vectorized) for the weight vector $\mathbf{w}$ that minimizes E($\mathbf{w}$).

   (d) Train and evaluate the linear regression model in the following 4 scenarios:

      1. Without regularization: use the training data to infer the parameters $\mathbf{w}$ for all values of $M \in [0,9]$. For each order M, compute E_{RMS} separately for the training and test data, and plot all the values as in Figure 1.5 in PRML.

2. Without regularization: Repeat the experiments above, but this time adding the validation data to the training dataset.
3. With regularization: Fixing $M = 9$, use the training data to infer the parameters $\mathbf{w}$, one parameter vector for each value of $\ln \lambda \in [-50, 0]$ in steps of 5. For each parameter vector, compute E_{RMS} separately for the training and validation data, and plot all the values as in Figure 1.8 in PRML. Select the regularization parameter that leads to the parameter vector that obtains best performance on the validation data, and use it to evaluate the model in two scenarios:
   (a) Train on the training data, test on the test data.
   (b) Train on the training data + validation data, test on the test data.
   Report and compare the two E_{RMS} values.

2 Tools

You are free to use MATLAB, OCTAVE, R, or packages written in C++/Java/Python to complete this assignment. If you want to use Java, I recommend using WEKA, a Java package that implements an extensive collection of machine learning algorithms. You can download the package from *http://www.cs.waikato.ac.nz/ml/weka*. There you will also find plenty of documentation on how to use it (API reference, tutorials, and manuals). There is also a local version of WEKA on the prime machines in the folder */home/razvan/ml6830/weka-3-7-1*. Other comments:

1. For this assignment, you will most likely want to use two classes: `weka.core.matrix.Matrix` and `weka.core.matrix.LinearRegression`. I recommend reading through their source code first in order to better understand how to use them. You will need to decompress the course code for WEKA, you can do this with the command 'jar xvf weka-src.jar'.
2. Gaussian samples can be generated using the method `java.util.Random.nextGaussian()`.
3. Graphs can be plot in Unix using the GNUPLOT utility.

3 Submission

Turn in a hard copy of your homework report at the beginning of class on the due date. Electronically submit a directory that has your working code, any trace files, and a concise README file describing these before class. Create a gzipped, tar ball archive of your directory, and upload it on Blackboard by the due date.

For example, if the name is John Williams, creating the archive can be done using the following commands:

> tar cvf williams_john.tar williams_john

> gzip williams_john.tar

These two steps will create the file 'williams_john.tar.gz' that you can upload on Blackboard.

Please observe the following when handing in homework:

1. Structure, indent, and format your code well.
2. Use adequate comments, both block and in-line to document your code.
3. **Do not submit third-party ML packages on Blackboard!** Just explain in the REAMDE file how you use external packages.
4. Type and nicely format the project report, including discussion points, tables, graphs etc. so that it is presentable and easy to read.
5. Working code and/or correct answers is only one part of the assignment. The project report, including discussion of the specific issues which the assignment asks about, is also a very important part of the assignment. Take the time and space to make an adequate and clear project report. On the non-programming learning-theory assignment, clear and complete explanations and proofs of your results are as important as getting the right answer.