15-830/630 – Problem Set #1

1. **Regularized least squares [10pts]**
   Consider the regularized least-squares objective mentioned in class

   $$J(\theta) = \|\Phi\theta - y\|_2^2 + \lambda\|\theta\|_2^2.$$

   By setting the gradient equal to zero and solving for θ, find a closed form expression of the optimal $\theta^\star$ that minimizes this function.

2. **Introduction to YALMIP [20pt]**
   Throughout the class we'll make extensive use of the optimization library YALMIP. In this question you'll set up and install MATLAB, YALMIP and other needed libraries, and experiment with some basic optimization formulations.

   (a) If you don't have it already, install MATLAB. CMU students can get a free (as in zero cost) copy at the Computing Serivces website: `http://www.cmu.edu/computing/software/all/matlab/download.html`.

   (b) Install YALMIP, available at `http://users.isy.liu.se/johanl/yalmip/`. Installation instruction are available on the website, and the important element is to add the various folders to your MATLAB path. You can easily add folders to your path using the command `pathtool`.

   (c) Install SeDuMi (version 1.3), a solver backend for YALMIP, available at: `http://sedumi.ie.lehigh.edu/`. Again the main installation will involve just adding folders to your MATLAB path.

   (d) **(15-830 only)** Install CPLEX, another backend solver for YALMIP. CPLEX is a commerical-grade solver that is very efficient at solving many classes of problems, and it recommended for those who expect to make substantial use of optimization in the future. You can find more information about CPLEX at `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`, and you can register for an academic copy at `http://www.ibm.com/developerworks/university/software/get_software.html`. Note that to get the copy you need to register for the IBM Academic Initiative, which will take a few days to process and you will need to verify that you are a "graduate student researcher" and point to a univerisity web page that verifies this status. If you happened to be enrolled in 15-830 but do not meet this criteria, let the instructor know.

   (e) At this point, you should be able to enter the command `yalmiptest` in MATLAB and get a list of solvers you have installed (SeDuMi and possibly CPLEX, as well as some built-in ones), then run through a suite of tests.

(f) For the following problem, you'll need this data, which is also in the file `q2.m` provided with the problem set data. Use both YALMIP and the MATLAB `linprog` command to solve the following problem, with optimization variable z:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & c^T z \\ \text{subject to} \quad & Az \le b \end{aligned}$$

Include code for both version of this solution, along with the optimal z. Verify that they do in fact get the same answer.

(g) Solve the following linear programming problem, using both `linprog` and YALMIP. Here you'll have to convert the program to standard form before using `linprog`; you could also use this standard form for the YALMIP solver, but try to write a solution in YALMIP that more directly corresponds to the problem.

$$\begin{aligned} \underset{z \in \mathbb{R}^3}{\text{minimize}} \quad & z_1 + 3z_3 \\ \text{subject to} \quad & z_1 \ge z_2 \\ & z_2 \ge 2 + z_3 \\ & 2z_1 \le z_2 + 5z_3 \\ & z_3 \le z_1 \end{aligned}$$

3. **Standard form optimization [20pts]**

(a) Formulate the problem of minimizing the sum of dead-band losses

$$\underset{\theta}{\text{minimize}} \quad \sum_{i=1}^{m} \max\{|\theta^T \phi(x_i) - y_i| - \epsilon, 0\}$$

as a linear program in standard form

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & c^T z \\ \text{subject to} \quad & Az \le b \end{aligned}$$

Hint: the resulting form looks very similar to the absolute loss case, with new variables ν_i such that $\max\{|\theta^T \phi(x_i) - y_i| - \epsilon, 0\} \le \nu_i$.

(b) The standard form of a *quadratic* program is given by

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & \frac{1}{2} z^T P z + q^T z \\ \text{subject to} \quad & Az \le b. \end{aligned}$$

Formulate the *regularized* deadband loss problem

$$\underset{\theta}{\text{minimize}} \quad \sum_{i=1}^{m} \max\{|\theta^T \phi(x_i) - y_i| - \epsilon, 0\} + \lambda \|\theta\|_2^2$$

as a quadratic program in standard form.

(c) **(15-830 only)** Write the kernelized version of deadband loss minimization as a quadratic program in standard form.

4. **Short answer [10pts]**

(a) Suppose you have a regression problem with a lot of training data, $m \approx 100,000$, and a you want to use polynomial features that result in a feature vector with dimension $k \approx 1,000$. Would you prefer to construct the features explicitly or use kernels to for this problem? Which (if any) of the difficulties discussed in class, representational and/or computational, will you likely run into?

(b) Suppose you have a problem with very little training data, $m \approx 100$, and want to use polynomial features with dimenion $k \approx= 1,000$. Would you prefer explicit features or kernels here? Which (if any) of the difficulties, representational and/or computational, will you likely run into?

(c) Finally, suppose you have only a modest amount of training examples $m \approx 1,000$, but want to use a polynomial feature vector of dimension $k \approx 100,000$. Would you prefer explicit features or kernels here? Which (if any) of the difficulties, representational and/or computational, will you likely run into?

5. **Experiments with linear and non-linear regression [25pts]**
In this program you'll recreate some of the figures and results from the class slides. All the data for this problem can be found in the data files `q5_max_temp.txt` and `q5_max_demand.txt`. Include the code and resulting figures in the answers to all these questions (these are the standard items to include for all programming assignments).

(a) Use linear and non-linear regression, with the squared loss function, to predict peak demand from high temperature. Use 1) linear regression, 2) non-linear regression with polynomial features (of max degree $d = 5$), 3) non-linear regression with RBF features (5 RBFs spaced uniformly over the input range, with bandwidth $\sigma = 20$, and 4) kernel linear regression using a Gaussian kernel (with bandwidth $\sigma = 20$ and regularization parameter $\lambda = 10^{-4}$). You should use the normal equations to learn the parameters for this problem.

Plot the resulting fits over the input data (hint: to do this, create a set of evenly spaced input points over the entire range of the input, and compute the prediction at each of these points).

(b) **(15-830 only)** Solve the same problem as in the previous question, but with the absolute loss instead of the squared loss. Here you'll probably want to use YALMIP to solve the optimization problems.

Some numerical issues pop up when solving the optimization problems in YALMIP. To address this, you should a) scale the polynomial features by 10^{-5} for part (2), and b) add a small penalty $10^{-6}\|alpha\|_2^2$ to the objective for the kernel formulation in part (4).

6. **(15-830 only) Data processing and additional features in regression [25 pts]** In the above problem, we distributed a data set that contained exactly the high temperatures and corresponding demand line-by-line. Real data is unlikely to ever appear in such a clean form, and you'll usually have to "clean" and preprocess the data yourself to make it usable for ML methods. With this in mind, in this problem you'll look at processing this data beginning with the temperature and demand data in "raw" form (the way they are downloaded from the web).

   Files `q6_demand.txt` and `q6_weather.txt` contain information about the Duquense Light electrical demand and Pittsburgh weather. Each line of the `q6_demand.txt` file contains the data:

   ```
   <UTC timestamp> <Electrical demand>
   ```

   and each line of the `q6_weather.txt` contains the data:

   ```
   <UTC timestamp> <temperature> <dewpoint> <humidity> <pressure> <wind speed> <conditions>
   ```

   where "conditions" is a discrete value corresponing to one of the conditions listed in the `q6_weather_conditions.txt` file. UTC timestamps are values that indicate the date in terms of seconds since Jan 1, 1970, in UTC time (time zone = 0).

   For this problem you'll need to do the following things:

   (a) Clean up the weather data. Datapoints are occasionally missing from the weather file, and have value -9999. You'll want to these entries from the data (you could do this by processing the file itself, but here, just do it in MATLAB after you load the data). There also are some times that have multiple entries; for these, just keep the first entry.

   (b) Synchronize the time stamps. The time stamps for the demand and weather files do not match up, and you'll need to make these correspond. Look into the MATLAB function `interp1` to do this.

   (c) Compute the maximum demand and maximum temperature for each day. For this, you can use the function `utc_to_datevec.m` included with the problem set data; the function takes a UTC timestamp and a time zone as input (Eastern time is -5, and we'll ignore daylight savings time for this problem) and outputs a vector of (year, month, day, hour, minute, second) for each input. Plot the resulting graph of maximum demand versus maximum temperature, and verify that it looks like

   (d) Compute other features from the raw weather data: minimum temperature, average dewpoint, average humidity, average pressure, average wind speed, and most common condition (breaking ties randomnly). Trying out different combinations of linear and non-linear regression (you have freedom here to construct whatever feature vectors you like), can you improve performance, as measure by squared loss, using all these features, over just using the max temperature?