CS 194-10, Fall 2011
Assignment 1

This assignment is to be done individually or in pairs. The goal is to gain experience with applying some simple learning methods to real data, where the quality of the learned model actually matters, as well as the estimate of the prediction uncertainty. When you are ready, `submit a1` as described **here**.

1. (15 pts) *Uncertainty of predictions made by linear regression*: Recall that an estimator of an unknown parameter θ is *unbiased* if its expected value (where the expectation is taken over the distribution of iid samples z_i generated from any distribution $P(z|\theta)$) equals the true value of θ. We can prove that the least-squares estimate $\hat{\mathbf{w}}$ given by the normal equations is an unbiased estimate of the true weights $\mathbf{w}$, assuming that the data are in fact generated by a linear model

$$\mathbf{Y} = \mathbf{X}\mathbf{w} + \epsilon$$

where each ϵ_i is an independent, zero-mean random error variable. (By *independent* here we mean that the errors are uncorrelated with each other: $Cov(\epsilon_i, \epsilon_j) = 0$ if $i \neq j$). When we talk about the bias of a least-squares estimate for regression, we can treat the input data $\mathbf{X}$ as fixed but arbitrary and the true weights $\mathbf{w}$ as fixed but unknown; the expectation is taken over possible realizations of the output values $\mathbf{Y}$, which in turn are determined by the errors ϵ.

The proof goes as follows:

$$\begin{aligned}
E[\hat{\mathbf{w}}] &= E[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}] \text{ from the normal equations} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T E[\mathbf{Y}] \text{ because the input values are fixed here} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T E[\mathbf{X}\mathbf{w} + \epsilon] \text{ by the model assumption above} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\mathbf{w} + E[\epsilon]) \text{ as the true parameters are fixed (albeit unknown)} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\mathbf{w}) + 0 \text{ because errors are zero-mean} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})\mathbf{w} \text{ by associativity} \\
&= \mathbf{w}\,.
\end{aligned}$$

Using a similar approach, show that the *variance* of the least-squares estimate $\hat{\mathbf{w}}$ is given by

$$Var(\hat{\mathbf{w}}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$$

where σ^2 is the variance of each ϵ_i. We will use this formula later to estimate the variance of our travel-time predictions for seismic waves. [Hint: begin from the following definition of the variance (sometimes called the covariance matrix) of a random vector $\mathbf{Z}$: $Var(\mathbf{Z}) = E[(\mathbf{Z} - E[\mathbf{Z}])(\mathbf{Z} - E[\mathbf{Z}])^T]$.]

2. (15 pts) *Weighted regression*: In lecture we discussed the linear predictor $\hat{\mathbf{w}}^T\mathbf{x}$, where $\hat{\mathbf{w}}$ was chosen as the minimizer of

$$\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2\ . \tag{1}$$

We now consider an alternative linear classifier, known as *weighted regression*. (This is an unfortunate term because "weight" is already used to refer to the parameters of the linear model.) Here, each training data point is given an *importance factor* F_i. (Call it a weight, if you like, just don't confuse it with w_j.) For example, if we are doing a *locally weighted* regression to predict the value of y at a

particular query point $\mathbf{x}_q$, we might want to define the importance factors to be higher for points closer to the query point. Some possible functions that do this:

$$\begin{aligned} F_i &= b/d(\mathbf{x}_i, \mathbf{x}_q) \\ F_i &= exp(-d(\mathbf{x}_i, \mathbf{x}_q)/b) \\ F_i &= exp(-d(\mathbf{x}_i, \mathbf{x}_q)^2/b^2) \end{aligned}$$

Weighted regression chooses $\hat{w}$ as the minimizer of

$$\sum_{i=1}^{N} F_i(y_i - \mathbf{w}^T\mathbf{x}_i)^2 \ . \tag{2}$$

(a) (5 pts) Rewrite (2) into exactly the same form as (1), except with modified values of the data y_i and $\mathbf{x}_i$. [Hint: Let $G_i = \sqrt{F_i}$.]

(b) (5pts) The standard matrix–vector form of the solution for $\hat{w}$ is

$$\hat{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \ . \tag{3}$$

We would like to find the correct solution for the weighted L_2 minimization problem. Using your previous answer, express the modified data as a data matrix $\mathbf{X}'$ and a label vector $\mathbf{Y}'$, where $\mathbf{X}'$ and $\mathbf{Y}'$ are defined in terms of $\mathbf{X}$ and $\mathbf{Y}$ and the G_i's. [Hint: put the G_i's into a matrix!]

(c) (5 pts) Finally, write out the correct solution for the weighted L_2 minimization problem, in terms of $\mathbf{X}$, $\mathbf{Y}$, and the importance factors.

Note that locally weighted regression is a reasonable solution for problems that are only locally linear, as in the case of seismic travel time prediction.

3. (70 pts) **Predicting travel times for seismic waves**

Background: As explained in class (see also **the lecture slides for Week 1**), each *phase* (P-wave, S-wave, etc.) emitted by a seismic event has a specific *travel time* from the source location to any given detector station. Accurate travel times are essential for pinpointing the event location by triangulation. A mathematical model (IASP91) has been constructed to predict these travel times; it assumes a spherically symmetric earth, so that travel time depends only on the *distance* (usually measured in degrees) between source and target locations as well as the source depth. In reality, the Earth is far from homogeneous in any given layer (nor is it a sphere!); these variations mean that IASP91 predictions are not sufficiently accurate. Assuming that the *actual* travel time from an event to a station can be measured—which is certainly true for so-called "GT0" (ground truth with zero error) events such as manmade explosions—the *residual* for that source–target pair is the difference between the actual and predicted travel times.

The data for this problem consist of *detections* (arrivals of specific phases at specific stations) described with the following attributes:

- Event ID
- Event latitude
- Event longitude
- Event depth
- Station ID
- Station latitude
- Station longitude

- Station elevation (km above sea level)
- Phase
- Travel time residual with respect to IASP91 model.

The goal is to produce, for each station and phase, a predictive model of the residual for an event in a new location; we will call this the *query location* $\mathbf{x}_q$. While the residual is certainly *not* a linear function of the event location, it may be *locally linear*, i.e., the residual varies reasonably smoothly with event location. Thus, we want to perform regression only with data *near* the query location; Section 18.8.4 of Russell and Norvig explains how to do this. The simplest method is to find the k nearest neighbors of the query location and do a regression on those data points; another method is to do a locally weighted regression using all the data.

The data are in CSV (comma-separated values) format in the file `trainingData.csv`. The code we have provided includes examples of how to process data in this form by running through the file one record at a time. We have also provided a function `dist` that computes the distance in degrees between any two points defined by latitude and longitude.

(a) (10 pts) Some practice with processing the data: Write code to find the two stations P_1 and P_2 with the largest number of P-phase detections and the two stations S_1 and S_2 with the largest number of S-phase detections.

(b) (30 pts) Write the definition for `klocalLinearRegression(station,phase,x,data,k)`, which predicts the travel time of the given `phase` from query point `x` to the `station` by finding the `k` nearest events to `x` for which a detection of that `phase` was observed at that `station`; use your function to estimate the time residual at (0,0) for each of the four station–phase pairs from part (a), using `k`=6.

(c) (30 pts) Use ten-fold cross-validation (or any other method of your choice) to plot the average per-example L_2 error of this prediction method as a function of `k` (and hence choose the best value of `k`) for each the four station–phase pairs in (a). Plug the appropriate values of `k` into the definitions of the four predictor functions `localLinearRegressionForP1` etc. Each of these will call your `klocalLinearRegression` function with the appropriate argument values. We will be testing how well these work on unseen data!

(d) (10 pts extra credit) The formula for $Var(\hat{\mathbf{w}})$ in Q.1 can be used to predict the variance of each prediction, provided the model assumptions are reasonable. Use cross-validation to estimate how well the variance formula works. Specifically, one expects the true answer to be within one standard deviation of the predicted answer roughly 68% of the time. Is this approximately true? If not, why might that be?

(e) (10 pts extra credit) Estimate the time residuals using locally weighted regression with one of the distance formulas in Q.2. Use cross-validation to choose a formula and parameter b.