Homework 1

1. *Nearest neighbor predictor.* We have a collection of n observations, $x^i \in \mathbf{R}^d$, $y^i \in \mathbf{R}$, $i = 1, \ldots, n$. Based on these observations, the nearest neighbor predictor is defined as $g^{\text{nn}}(x) = y^k$, where x^k is a nearest neighbor of x among the data points. (We can break ties arbitrarily. Recall that x^k is a nearest neighbor of x means that $\|x - x^k\| \leq \|x - x^i\|$ for $i = 1, \ldots, n$.)

   (a) Write a Julia function `y_hat = nn_predictor(x, X, y)` that implements g^{nn}, given the argument `x`. The second and third arguments give the data on which the predictor is based: `X` is an $n \times d$ matrix whose ith row is $(x^i)^T$, and `y` is an n-vector with ith entry y^i.

   (b) Report the train and test RMSE of the predictor in part (a) on the data provided in `nearest_neighbor_data.json`, training on the first 1500 data points and testing on the rest. Briefly interpret your results.

2. *Soft nearest neighbor.* In this exercise we examine an extension of the nearest neighbor predictor that can occasionally perform better. For data set $x^i \in \mathbf{R}^d$, $y^i \in \mathbf{R}$, $i = 1, \ldots, n$, the *soft nearest neighbor predictor* is defined as

$$g^{\text{snn}}(x) = \frac{\sum_{i=1}^n y^i e^{-\|x-x^i\|^2/\sigma^2}}{\sum_{i=1}^n e^{-\|x-x^i\|^2/\sigma^2}},$$

   where $\sigma \geq 0$ is a parameter. Note that σ can be thought of as a distance; the function $e^{-\|x-x^i\|^2/\sigma^2}$ is near one when the distance between x and x^i is much less than σ, and it is very small when the distance is much more than σ.

   (a) What does $g^{\text{snn}}(x)$ converge to as $\sigma \to 0$? (Explain briefly.)

   (b) What does $g^{\text{snn}}(x)$ converge to as $\sigma \to \infty$? (Explain briefly.)

   (c) Implement the soft nearest neighbor predictor in Julia as

   `y_hat = snn_predictor(x,X,y,sigma)`.

   (d) Using the data in `nearest_neighbor_data.json`, plot train and test RMSEs for g^{snn} as a function of σ over the range $[10^{-1}, 10^1]$. What value of σ would you choose? How does the test RMSE of predictor compare to the nearest neighbor predictor and the constant predictor?

3. *Sequential outlier removal.* Throughout this problem, you'll use the data `U`, `v`, found in `fitting_outliers.json`. Here, $U \in \mathbf{R}^{n \times 1}$, so there is only one (nonconstant) feature. This one feature is already (nearly) standardized, so you do not need to standardize it. The data matrix X will have two columns, the constant feature one and the feature given in U. Also, there is enough data that you do not need to use any regularization.

(a) Fit a least squares model to the dataset above and plot the data points and straight-line fit. Describe what you observe.

(b) *Sequential outlier removal.* Find the data point with the largest loss and label it as an outlier. Remove this point from your data set and fit the model again to this new dataset (which has one fewer data point). Continue doing this until your θ stops changing too much (say, the change between the components of the previous θ and the current one is no more than .01).

Show a few of the intermediate fits and the final fit plotted against the data points. Describe what you observe.

4. *All-pairs interactions.* In the following problem, we will use `U`, and `v` found in the data file `all_pairs_data.json`. The data has $U \in \mathbf{R}^{n\times 3}$. Throughout this problem, use a 50-50 train/test split.

(a) Fit a linear least-squares model directly to the data matrix, with the first feature being a constant feature $x_1 = 1$. Since we've given you enough data and the data is approximately standardized, you do not have to worry about regularization or standardization. Report the train and test RMSE of this predictor.

(b) Create an embedding which includes all of the interactions (products) between every pair of distinct variables, along with a constant feature and the variables themselves. For example, if $u \in \mathbf{R}^2$, then the embedding should be

$$\phi(u) = (1, u_1, u_2, u_1u_2).$$

Report the train and test RMSE of this predictor. Compare it with the RMSEs you got in (a).