DS 310 Machine Learning
Vasant Honavar
Fall 2018

Problemset 2: K Nearest Neighbor and Linear Regression

Available: September 5, 2018
Due: September 14, 2018

1. (25 pts.) Suppose 7 nearest neighbor search for a query sample returns $\{7, 6, 8, 4, 7, 11, 9\}$ as the values of the function at the 7 nearest data samples in the training set.
   (a) Assuming prediction using 7 nearest neighbor interpolation, what is the predicted value of the function for the query sample?
   (b) Assuming prediction using 7 nearest neighbor regression, what is the predicted value of the function for the query sample?

2. (25 pts.) Alex is building a K nearest neighbor classifier for predicting the language to which a word belongs. The words use the Latin alphabet (which is used by English, Spanish, German and several other European languages). For better or for worse, Alex chooses to represent the words using three features: length of the word, number of vowels in the word, and whether or not the word ends in the letter 'n" (0 for No, 1 for Yes). Suppose she has training data for two languages: regeln: German; pido: spanish.
   (a) What is the dimensionality of the feature space?
   (b) How are the two training data samples encoded in terms of the 3 features listed?
   (c) Using 1-nearest neighbor classifier using the Manhattan distance metric, what is the prediction for the word "neben"?

3. (25 pts.) Assume that you have a trained K nearest neighbor classifier.
   (a) What is the runtime complexity of this classifying p data samples given a K nearest neighbor classifier trained on m training samples, each represented using d features?.
   (b) Can you do better, using a carefully designed data structure with a K nearest neighbor search algorithm that is designed to take advantage of the data structure? Explain.

4. (25 pts.) Consider approximation of $f(\mathbf{X})$ where $\mathbf{X} \in \Re^d$ given a training set $\{(\mathbf{X}_1, y_1) \cdots (\mathbf{X}_n, y_n)\}$ using linear regression. That is, the goal is to find a linear approximation $y(\mathbf{X}) = \mathbf{W} \cdot \mathbf{X} + w_0$ where $\mathbf{W}$ and w_0 are learnable parameters. Suppose we further know that the function is sparse, i.e., only a few of the d features define the behavior of f. How would you modify the objective function to be minimized to ensure that the resulting linear is sparse? (Hint: Add a term to the mean squared error that when minimized attempts to drive the weights to 0).