hw4: greedy algorithm for teaching OLS; 1NN classifier

March 1, 2023

1 Q1: When enumeration stops working?

We will use the pool P in hw3pool.txt again for this homework, so we connect to hw3. Use enumeration to find the optimal OLS teaching set of size $n = 2, 3, 4, 5, \ldots$. For each n, report (1) the parameter loss $\|\hat{\theta} - \theta\|^2$ of the optimal teaching set, recall this is $(\hat{a} - a)^2 + (\hat{b} - b)^2$. Use scientific notation for the loss because we anticipate the loss to be very close to zero. (2) the number of times you need to fit OLS. (3) time in seconds.

If it is difficult for you to enumerate multisets, you can enumerate true subsets of P instead, or simple enumerate all pool items at each "position" of your teaching set.

It is possible that it takes too long for large n. You can stop and comment on the time cutoff.

2 Q2: A greedy algorithm

```
1. Uniform randomly pick the first point $(x, y)$ from the pool $P$, form the initial teaching set $D = \{(x, y)\}$.
2. For $n = 2, \ldots, |P|$
3.     For $(x, y) \in P$
4.         Compute $\hat{\theta} = OLS(D \cup \{(x, y)\})$
5.         maintain the $(x^*, y^*)$ with the smallest loss $\|\hat{\theta} - \theta\|^2$
6.     EndFor
7.     Actually grow $D = D \cup \{(x^*, y^*)\}$
8. EndFor
```

Report the same three quantities as your teaching set grows.

Our algorithm grows the teaching set until it has the same size as the pool. Looking at the loss sequence, what stopping criterion might you have used instead?

How do you compare the teaching set quality of the greedy algorithm to enumeration?

3 Q3: Implement kNN classifier

Given a training set $D = (x_1, y_1) \ldots (x_n, y_n)$ where each item $x_i \in \mathbb{R}^d$ is a d dimensional feature vector, and k, implement kNN so that you can classify any test point $x \in \mathbb{R}^d$. Your code should work for any n, d. Use Euclidean distance, and break tie arbitrarily.

For hand in, prepare a 2D training set. Color code your training points. Then, make a dense 2D grid of test points. The grid should cover the general region of your training set. Run kNN with $k = 1$ and optionally other values you choose. For each k, you should plot both the training set and the kNN classification of test points on the same plot. Make sure you can visually distinguish training and test points.