hw5: machine teaching for kNN

March 17, 2023

Similar to hw3, we now consider pool-based teaching for kNN learners.

- Given any dataset $D = (x_1, y_1) \dots (x_n, y_n)$ where $x_i \in \mathbb{R}^d$ and y_i is now a class label, The student runs kNN to learn a classifier $f : X \mapsto Y$. k is given and fixed.
- The teacher knows the above student algorithm. The teacher can only influence the student with the teaching set D. The teacher wants to make sure the student classifier f is close to a target classifier $g : X \mapsto Y$.

1 Formal Definition

Recall a pool-based teacher cannot create arbitrary data points. Instead, the teacher is given a "pool" of data points $P = \{(x_1, y_1) \dots (x_N, y_N)\}$, and the teacher must select pairs from the pool to form its teaching set D. For this homework, we allow D to be a multiset (i.e. allow repeated items from P). Exact teaching is in general infeasible in pool-based teaching. Instead, the teacher aims to approximately teach the student the target model g.

In theory, let

$$f = kNN(D)$$

where we treat kNN as a function (a learning algorithm) that takes a training set D and outputs a classifier f. The teacher has a target classifier g, and has a probability density function $p(x)$ over the input space X. Let the 0-1 loss be $\ell(y, y') = 1$ if $y \neq y'$ and 0 otherwise. Then the teacher can define the disagreement between f and g as

$$d(f, g) := \int p(x)\ell(f(x), g(x))dx. \tag{1}$$

Clearly, $d(f, g) \in [0, 1]$. $d(f, g) = 0$ if f equals g with probability one. We can now state the teaching problem as

$$D^* = \text{argmin}_{D \subseteq P} \qquad d(kNN(D), g). \tag{2}$$

2 Approximating Integral with Sampling

In practice, integrating over X can be difficult. Instead, we approximate integral with sampling. For this, we need a large sample

$$Z = \{(x'_1, y'_1 = g(x'_1)), \dots, (x'_m, y'_m = g(x'_m))\}$$

where $x'_i \sim p(x)$. In general, Z is distinct from the pool P, though the two could be the same. We can then approximate the integral by average and define

$$\hat{d}(f, g) := \frac{1}{m} \sum_{i=1}^{m} \ell(f(x'_i), y'_i). \tag{3}$$

It will be the case that $\hat{d}(f,g) \approx d(f,g)$ when m is large. Note $\hat{d}(f,g)$ is a random variable because it depends on the sample Z, while $d(f,g)$ is a constant. Given Z, the teacher can solve a slightly different problem

$$\hat{D} = \text{argmin}_{D \subseteq P} \qquad \hat{d}(kNN(D), g). \tag{4}$$

Again, there are at least two ways to solve this problem:

1. Enumeration. The teacher enumerates all subsets of P. Note a subset can be even smaller than k: kNN is well defined on any training set size. If a training set is smaller than k, kNN will simply do a majority vote on all training items.

2. Greedy. Given a partial teaching set D (initially empty), the teacher enumerates all one-item extension $D \cup \{(x,y)\}$ where $(x,y) \in P$. The teacher adds the best one-item extension (x^*, y^*) to D, where

$$(x^*, y^*) = \text{argmin}_{(x,y) \in P} \qquad \hat{d}(kNN(D \cup \{(x,y)\}), g). \tag{5}$$

   This greedy process repeats so D grows.

3 Adding a Teaching Cost

The teacher's objective so far is to guide the student close to g. Finally, we introduce the notation of a "teaching cost" function $c(D)$ which specifies how costly it is for the teacher to use a teaching set D. The new teaching problem is

$$\hat{D} = \text{argmin}_{D \subseteq P} \qquad \hat{d}(kNN(D), g) + c(D). \tag{6}$$

$c(D)$ can be thought of as a "regularizer" in the data space that encourages cheap teaching sets. For this homework, we will consider a particularly simple teaching cost: $c(D) = 0$ if $|D| \leq n^*$, and ∞ otherwise, for a given threshold n^*. This simply prevents D from being larger than n^*. For enumeration, you will consider subsets of size at most n^*; for greedy, simply stop after size n^*.

4 Hand in

For this homework let $k = 1$ (1NN), $n^* = 20$, $P = Z =$hw5data.txt. Each row x_1 x_2 y is a data item with 2D feature (x_1, x_2) and binary label y. Implement both enumeration and greedy. For each teaching set size, show:

1. the number of teaching sets of that size that you have to search through;

2. number of seconds it takes for that size;

3. $\hat{d}(kNN(D), g)$ of the best teaching set of that size

4. for enumeration, plot the best teaching set $\hat{D}$ in relation to P (i.e. plot both, but use different symbols for $\hat{D}$)

For greedy, only plot one figure for the last teaching set $\hat{D}$ with size n^* in relation to P, but see if you can mark the order of teaching items entering the greedy teaching set (e.g. use numbers instead of symbols to show items in $\hat{D}$).

Again you may not be able to enumerate teaching sets of large size, and that is OK.