CIS 399 Homework 1: Fundamentals of Machine Learning

Prof. Michael Kearns
Spring 2021

Assigned Feb. 7; Due Feb. 16

In this assignment, you are asked to creatively demonstrate your understanding of the fundamental framework and definitions we gave for supervised machine learning problems. Recall that the main elements of this framework were as follows:

- A space X of inputs x (usually multidimensional vectors)
- A space Y of outputs or outcomes y (often discrete or binary valued)
- A probability distribution P over $\langle x, y \rangle$ pairs
- A sample S of n pairs drawn from P
- A model space H of functions mapping X to Y
- For any $h \in H$, its training or empirical error $\hat{\epsilon}(h)$ on the sample S, which in the case of classification is simply the fraction of $\langle x, y \rangle$ pairs in S for which $h(x) \neq y$
- For any $h \in H$, its true or expected error $\epsilon(h)$ with respect to P, which in the case of classification is simply the probability that $h(x) \neq y$ on a pair $\langle x, y \rangle$ drawn randomly according to P

Given these definitions, the typical workflow of machine learning can be described as follows. We don't know anything directly about P, but we can sample from it to get training data S. Given the nature of the inputs x and the outcomes y we would like to predict, we choose or design our model space H. We then use an algorithm that attempts to find an $h \in H$ whose training error $\hat{\epsilon}(h)$ is as small as possible — ideally the minimum possible within H, but that's usually computationally hard to find, so our algorithm may be imperfect or heuristic. But the overall hope is that by making $\hat{\epsilon}(h)$ small, $\epsilon(h)$ will be small as well.

For example, in lecture we populated some parts of this framework by (informally) describing a consumer lending application, where the inputs x contained things like an applicant's current salary, age, employment status, whether they finished high school, etc., and the outcome y we wished to predict was whether the applicant was a good lending risk (i.e. would repay a loan if given). We then described a small and plausible decision tree for making such predictions.

In this problem, you are asked to create and describe a similar example, but in greater detail and formality. In what follows, you are asked to invent an end-to-end instance of the framework above, ideally an interesting and natural one. Because you're asked to provide a fair amount of detail, you'll want to keep your example relatively simple (e.g. low-dimensional).

1. First motivate describe informally your input space X and outcome space Y. In the example from lecture, the answer would be something like "predict whether people will repay loans based on their financial, employment and education history". For simplicity, have your Y space be binary (only two outcomes).

2. Now describe your X and Y more formally — say exactly what the components of X are (e.g. the x contain current salary, a bit indicating whether the applicant is currently employed, the age of the applicant, etc.) and what the predictions space Y is (e.g. the y are binary values indicating will or won't repay).

3. Describe a plausible or natural probability distribution P over the $\langle x, y \rangle$ pairs you have defined. This part is a bit harder to do precisely, especially depending on how complex your X is. But one way of doing it is by describing a distribution over just X, and then describing a deterministic function mapping X to Y. Here would be a simple example for the lending case: the distribution over ages is uniform from 21 to 60; the distribution over annual salaries is uniform from \$10K to \$100K; and the distribution over education is 25% did not finish high school and 75% did. For simplicity we assume these three variables are independent. Then for any x, the outcome y is 1 (will repay loan) if and only if salary is at least \$50K; *or* salary is at least \$$25K$ and applicant is a high-school graduate whose age is at least 30. (Bonus: arrange things so that the probabilities that $y = 0$ and $y = 1$ are exactly 0.5 each.)

4. We'll again let our model class H be decision trees. Carefully describe a particular decision tree h whose true error $\epsilon(h)$ on your chosen distribution is exactly 0.25 — i.e. for $\langle x, y \rangle$ pairs chosen according to your P, $h(x) \neq y$ 25% of the time. You should draw your decision tree carefully as was done in lecture, showing exactly what each of the internal tests are, and what the predictions or labels at the leaves are. You should show your calculation of $\epsilon(h)$.

5. Finally, describe carefully (i.e. give a list of) a sample S of 10 $\langle x, y \rangle$ pairs for which the decision tree h has training error $\hat{\epsilon}(h) = 0$ — i.e. h makes no mistakes on S. Obviously we want your S to be drawn randomly from your P, but since that's impossible to enforce or check in a math problem,

just try to make it seem plausible that your S was drawn according to your P.

When you've done all the above, you will have invented an instance of our ML framework that has the feel of the real world — we found a model that does well (actually perfectly) on the data S, naturally does worse on the true distribution, but is still pretty good.