Comp 451: Machine Learning

Jacob Reznikov

March 20, 2021

Abstract

My course notes for Comp 451

1 Basic concepts

1.1 Data

We describe data using:

$$D = \{(x_i, y_i), i = 1...n\}$$

Where x_i are feature sets:

- Feature sets are usually m dimensional, i.e $x_i[j]$ is a the j-th feature of the set
- Examples of a feature are
  - $\{0, 1\}$ called a binary feature
  - $\mathbb{Z}$ called a catagorical feature
  - $\mathbb{R}$ called a continuous feature, for example a time series

  Since $\{0, 1\} \subseteq \mathbb{Z} \subseteq \mathbb{R}$ we can describe a feature set as a vector in $\mathbb{R}^n$

Meanwhile y_i are labels/targets:

- One dimensional
- Same types of data as features
- If not continuous it is called a classification
- If continuous it is called a regression

1.2 Loss Function

We are trying to learn a function $f : \mathcal{X} \longrightarrow \mathcal{Y}$ that some how matches the data set.
We often do this using what's called a loss function, a loss function should be a function

$$L : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$$

That measure how good our prediction is compared to the actual data.
An example of a loss function would be

$$L_{0,1}(y, \hat{y}) = \begin{cases} 1 : y = \hat{y} \\ 0 : y \neq \hat{y} \end{cases}$$

1.3 Test sets and generalization

$$f(x) = \begin{cases} y_i : \exists(x_i, y_i) \in D : x = x_i \\ undefined \end{cases}$$

This function is useless because it doesn't generalize.
To solve this we split data into training and testing, we call these D_{trn} and D_{tst}.
We create the model based on the D_{trn} and then we test on the D_{tst} set.
These *cannot* intersect as that would lead to the undergeneralization problem.

2 Decision Boundary

Lets assume we have a binary classification problem.
We would often like to be able to draw a boundary that on its own classifies our input. If we, for example, take the input to be 2 dimensional then this is equivalent to drawing a curve that seperates the input set into 2 partitions X_0, X_1. We can then define our function as the charecteristic function of the partition.

We could write this neatly as

$$f(x) = \chi_{X_1}$$

Now in practice, calculating these boundaries is not trivial and so we employ a variety of tactics.

2.1 Instance Based Learning

As humans we have a built it visual understanding of 2D data but what if we want to program this?
We want to be able to derive a partition from the data given. The simplest example of this is to just let the set X_1 be the set of points that are closer to a datapoint with label 1 than a datapoint with label 0.

We can write this formally as

$$f(x) = \begin{cases} 1 & : \exists i : (x_i, 1) = \arg\min_{(x_i, y_i)}(d(x, x_i)) \\ 0 & : \exists \end{cases}$$