ML Section 1.4: Evaluation of Supervised Learning

Adam Hawley

March 13, 2019

Contents

1 Hypothesis Evaluation

General questions:

- How can one estimate the performance of a learned hypothesis on future data?
- How good is the estimate?

- Comparative performance evaluations.

Formally: Given a hypothesis h and a data sample containing n examples drawn at random according to the distribution D, what is the best estimate of the accuracy of h over future instances drawn from the same distribution? What is the probable error in this accuracy estimate?

2 Evaluation Problems

- Limited samples of data may be misleading (e.g. prime numbers and data set = {3,5,7} leads to hypothesis of odd numbers).
- Observed accuracy on training data is often too optimistic (e.g. due to overfitting).
- Solution: use independent test examples.
- Problem: estimate may still depend on the specific makeup of the set of training/test examples.

3 Preliminary Definitions:

f The target categorisation function to be learned (f:Examples → Categories).

h The hypothesis learned (h: Examples → Categories).

S Data sample of size n.

D Probability distribution over all data points.

Sample Error

$$error_s(h) = \frac{1}{2}\sum_{x\in S}\delta(f(x), h(x) \tag{1}$$

Where:

$$\delta(y,z) = 1 \text{ if } y \neq z, \text{ and } 0 \text{ otherwise.} \tag{2}$$

True Error

$$error)D(h) = Pr_{x\in D}[f(x) \neq h(x)] \tag{3}$$

4 Confidence Intervals (for discrete-valued hypotheses)

Given the following conditions:

1. The sample S contains n examples drawn independent of one another and independent of h, according to the probability distribution D.
2. $n \geq 30$
3. Hypothesis h commits r errors over these n examples.

Then, with $N\%$ probability, $error_D(h)$ lies in the interval:

$$error_s(h) \pm z_n \sqrt{\frac{error_s(h)(1 - error_s(h))}{n}} \tag{4}$$

5 More Evaluation Questions

- Q: Where to get independent test examples?
- A: Use subset of training data (and don't use it for training)

The problem with this is that our data set might be very small and so we would want to use every piece of it for training rather than saving it for testing.

- Q: Where can we get more samples to increase confidence?
- A: Divide the training data into several subsets. Use one of them as test data and the rest as training data. Then repeat for each subset.

6 k-fold Cross-Validation Algorithm

Partition the available data D_0 into k disjoint subsets $T_1, \ldots, T_k$ of equal size, where this size is at least 30.

-
  ```
  for i from 1 to k do {
  S_i := D_0 - T_i;
  h := learner hypothesis on training data S_i;
  $\delta_i$ := error rate of h on T_i;
  }
  ```

- Return average error rate $\delta := (1/k) \sum_{1 \leq i \leq k} \delta_{\mathrm{i}}$

The true error lies with $N\%$ probability in the interval:

$$\delta \pm t_{N,k-1} s_{\delta} \tag{5}$$

where

$$s_{\delta} = \sqrt{\frac{1}{(k(k-1))} \sum_{1 \leq i \leq k} (\delta_i - \delta)^2} \tag{6}$$

7 Comparison of Hypothesis

7.1 Standard Error from the Mean

A commonly used cheap trick for comparison which is used is to compute the standard error (STE). The standard error is equal to $\frac{STD}{\sqrt{k}}$. The confidence interval is then defined as $\delta \pm STE$. If two confidence intervals resulting from two hypotheses overlap, it can not be stated that one hypothesis is better than the other.

8 Further Problems

- Has the training set been chosen with the same distribution as future examples?
- k is limited by size of training set.
  - One alternative is to draw (possibly overlapping) subsets randomly.
  - Disadvantage: Test sets are not completely independent and have not been chosen with the same distribution as the training set.

9 Parameter Tuning

It is important that the test data is not used in any way to create the classifier. Some learning schemes operate in two stages:

1. Build the basic structure
2. Optimise parameter settings

The test data cannot be used for parameter tuning! Proper procedure uses three sets:

- Training data
- Validation data
- Test data

Validation data is used to optimise parameters.