

Intro. to machine learning (CSI 5325)

Lecture 11: evaluating hypotheses

Greg Hamerly

Spring 2008

Some content from Tom Mitchell.

- 1 Comparing hypotheses
- 2 Paired t tests
- 3 Comparing learning methods

Difference Between Hypotheses

Test h_1 on sample S_1 , test h_2 on S_2

- 1 Pick parameter to estimate: $d \equiv \text{error}_{\mathcal{D}}(h_1) - \text{error}_{\mathcal{D}}(h_2)$
- 2 Choose an estimator: $\hat{d} \equiv \text{error}_{S_1}(h_1) - \text{error}_{S_2}(h_2)$
- 3 Determine probability distribution that governs estimator

$$\sigma_{\hat{d}} \approx \sqrt{\frac{\text{error}_{S_1}(h_1)(1 - \text{error}_{S_1}(h_1))}{n_1} + \frac{\text{error}_{S_2}(h_2)(1 - \text{error}_{S_2}(h_2))}{n_2}}$$

- 4 Find interval (L, U) such that N% of probability mass falls in the interval

$$\hat{d} \pm z_N \sqrt{\frac{\text{error}_{S_1}(h_1)(1 - \text{error}_{S_1}(h_1))}{n_1} + \frac{\text{error}_{S_2}(h_2)(1 - \text{error}_{S_2}(h_2))}{n_2}}$$

Difference between hypotheses

$$\sigma_{\hat{d}} \approx \sqrt{\frac{\text{error}_{S_1}(h_1)(1 - \text{error}_{S_1}(h_1))}{n_1} + \frac{\text{error}_{S_2}(h_2)(1 - \text{error}_{S_2}(h_2))}{n_2}}$$

Comes from the facts that:

- $\text{error}_{S_1}(h_1)$ and $\text{error}_{S_2}(h_2)$ are both (approximately) normally distributed (if n is large enough)
- the difference of two normally distributed random variables is also normally distributed

Hypothesis testing

More interesting than knowing the distribution of \hat{d} is knowing whether h_1 is better than h_2 .

But for this we can just use the distribution of \hat{d} .

Hypothesis test example:

- H_0 : h_1 and h_2 are the same ('null' hypothesis)
- H_A : h_1 and h_2 are different ('alternative' hypothesis)

Estimator: $\hat{d} = error_{S_1}(h_1) - error_{S_2}(h_2)$

Test: $P(\hat{d} \neq 0)$

- which is the probability that the confidence interval on \hat{d} does NOT contain 0.

Paired t test to compare h_A, h_B

1 Partition data into k disjoint test sets T_1, T_2, \dots, T_k of equal size, where this size is at least 30.

2 For i from 1 to k , do

$$\delta_i \leftarrow \text{error}_{T_i}(h_A) - \text{error}_{T_i}(h_B)$$

3 Return the value $\bar{\delta}$, where

$$\bar{\delta} \equiv \frac{1}{k} \sum_{i=1}^k \delta_i$$

Paired t test to compare h_A, h_B

$N\%$ confidence interval estimate for d :

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

$$s_{\bar{\delta}} \equiv \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^k (\delta_i - \bar{\delta})^2}$$

Note δ_i approximately Normally distributed

Paired t test

The t distribution (or 'Student's t distribution) applies when estimating the mean of normally-distributed random variable with unknown variance.

Since the variance of the underlying distribution is unknown, the distribution of the mean estimator has a wider variance than a normal distribution.

As $n \rightarrow \infty$, the t distribution is well-approximated by the normal distribution.

It's called a paired test because each hypothesis is tested on the same sets of examples.

Student's?

Comparing learning algorithms

So far, we've just discussed comparing hypotheses. For example,

- compare tree X with tree Y
- compare neural network X with network Y
- compare neural network X with tree Y

Hypotheses are particular things that are learned, however, and we're interested in *learning algorithms*.

We want to be able to compare the performance of different learning algorithms.

Comparing learning algorithms L_A and L_B

What we'd like to estimate:

$$E_{S \subset \mathcal{D}}[\text{error}_{\mathcal{D}}(L_A(S)) - \text{error}_{\mathcal{D}}(L_B(S))]$$

where $L(S)$ is the hypothesis output by learner L on training set S

i.e., the expected difference in true error between hypotheses output by learners L_A and L_B , when trained using randomly selected training sets S drawn according to distribution \mathcal{D} .

But, given limited data D_0 , what is a good estimator?

Comparing learning algorithms L_A and L_B

$$E_{S \subset \mathcal{D}}[\text{error}_{\mathcal{D}}(L_A(S)) - \text{error}_{\mathcal{D}}(L_B(S))]$$

Given limited data D_0 , what is a good estimator?

- could partition D_0 into training set S and training set T_0 , and measure

$$\text{error}_{T_0}(L_A(S_0)) - \text{error}_{T_0}(L_B(S_0))$$

- even better, repeat this many times and average the results (next slide)

Comparing learning algorithms L_A and L_B

- 1 Partition data D_0 into k disjoint test sets T_1, T_2, \dots, T_k of equal size, where this size is at least 30.
- 2 For i from 1 to k , do
 - use T_i for the test set, and the remaining data for training set S_i
 - $S_i \leftarrow \{D_0 - T_i\}$
 - $h_A \leftarrow L_A(S_i)$
 - $h_B \leftarrow L_B(S_i)$
 - $\delta_i \leftarrow \text{error}_{T_i}(h_A) - \text{error}_{T_i}(h_B)$
- 3 Return the value $\bar{\delta}$, where

$$\bar{\delta} \equiv \frac{1}{k} \sum_{i=1}^k \delta_i$$

Comparing learning algorithms L_A and L_B

Notice we'd like to use the paired t test on $\bar{\delta}$ to obtain a confidence interval

... but it's not really correct, because training sets in this algorithm aren't independent (they overlap!)

It's more correct to view algorithm as producing an estimate of

$$E_{S \subset D_0}[\text{error}_{\mathcal{D}}(L_A(S)) - \text{error}_{\mathcal{D}}(L_B(S))]$$

instead of

$$E_{S \subset \mathcal{D}}[\text{error}_{\mathcal{D}}(L_A(S)) - \text{error}_{\mathcal{D}}(L_B(S))]$$

but even this approximation is better than no comparison