Calculating Significance for a Single Test-Train Split

Jason D. M. Rennie
jrennie@gmail.com

June 8, 2005

Consider the following common scenario. We have one classification data set, broken into test and train parts and two classification algorithms. We assume that the given data is a sample from a distribution. We would like to know what is the probability that one of the algorithms has a lower generalization error on the unseen distribution.

We train each algorithm using the training data and evaluate on the test data. We calculate the following four statistics:

- n_C - the number of test examples that both algorithms predict correctly
- n_A - the number of test examples that only algorithm A predict correctly
- n_B - the number of test examples that only algorithm B predict correctly
- n_W - the number of test examples that neither algorithm predicts correctly

The generalization error for an algorithm is the expected error on the underlying data distribution. We would like to calculate the chance that both algorithms have the same generalization error, assuming that data samples are independently drawn from the underlying distribution.

If both algorithms have the same generalization error, then we would expect to see $n_A \approx n_B$. However, if one value is much larger than the other, then chances are one algorithm has higher generalization error. In short, if both algorithms have the same generalization error, then the chance that algorithm A predicts an example correctly while algorithm B does not is the same as seeing "heads" on a fair coin flip. Clearly, the most typical scenario is $n_A = n_B$. We calculate the probability that $|n_A - n_B| \geq k$ and report this probability. If this probability is small, then it is very likely that the algorithms have different generalization errors.