

Key Questions

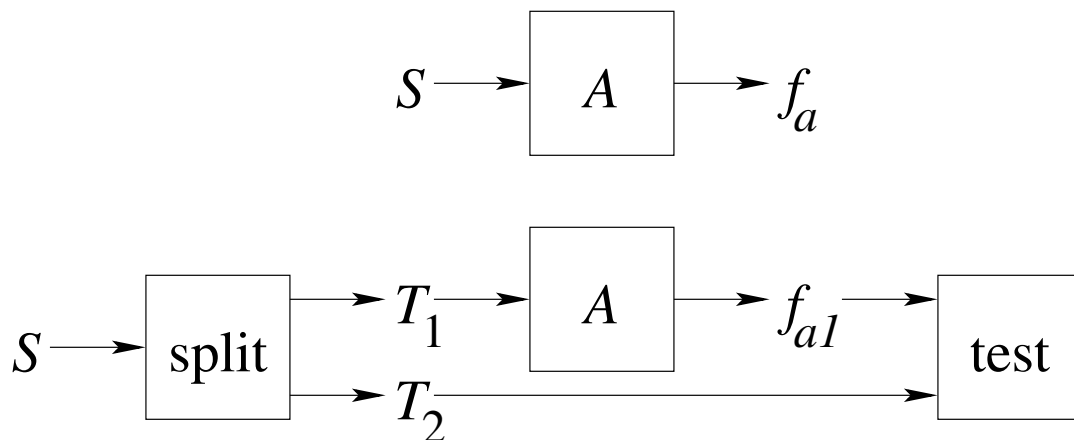
Assume that dataset S is generated from a probability distribution P .

1. When algorithm A is run on S to produce f_a , what is the error rate of f_a ?
2. When two algorithms A and A' are run on S to produce f_a and f'_a , does f_a have a lower error rate than f'_a ?

The *train-and-test* method uses S to empirically answer these questions.

Train and Test

1. Split the sample S into a training set T_1 and a test set T_2 .
2. Run the algorithm A on T_1 to produce f_{a1} .
3. Determine the error rate of f_{a1} on T_2 ?



Properties of Holdout Method

This is often called the *holdout* method.

Because $T_1 \neq S$, it is likely that $f_{a1} \neq f_a$.

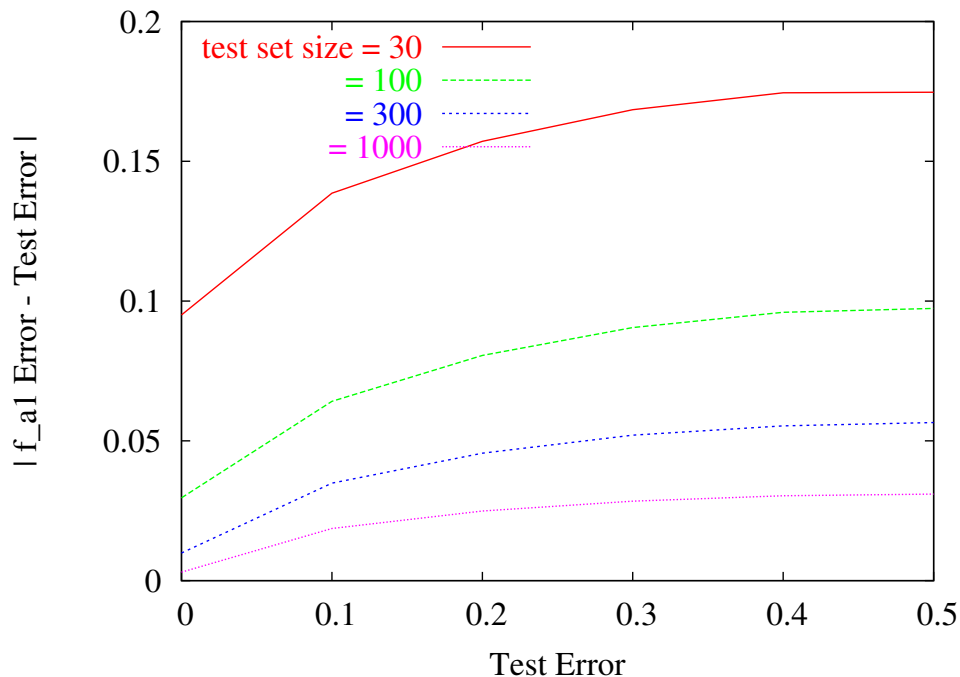
There is a tradeoff in the sizes of T_1 and T_2 .

Making T_1 larger likely decreases the difference between f_{a1} and f_a .

Making T_1 larger decreases the accuracy of the test error rate on f_{a1} .

Difference between Test Error and True Error

For different test set sizes, this shows a 95% confidence bound on $|f_{a1} \text{ error} - \text{test error}|$



More on Holdout Method

Traditionally, 2/3 of the dataset is used for training, 1/3 for testing.

Results can be analyzed using standard statistics.

Holdout tends to be pessimistic because it doesn't account for what is learned from T_2 .

Holdout doesn't account for variance due to algorithm.

Statistics for Holdout Method

$n = |T_2|$ = the number of test examples

$\delta_i = \begin{cases} 0 & \text{if } f_{a1}(\mathbf{x}_i) = y_i \\ 1 & \text{if } f_{a1}(\mathbf{x}_i) \neq y_i \end{cases}$ or 0-1 loss calc.

$u = \frac{\sum_{i=1}^n \delta_i}{n}$ = average 0-1 loss

$s^2 = \frac{\sum_{i=1}^n (\delta_i - u)^2}{n - 1}$ = sample variance

$z = \frac{s}{\sqrt{n}}$ = z statistic

Poor but quick: $u \pm 1/\sqrt{n}$ loosely estimates a 95% confidence interval.

Good: For large enough n , the following is a 95% confidence interval:

$$u \pm 1.96z$$

If $5 \geq nu(1 - u)$, then this is an acceptable approximation.

Better: Use the critical value from the t distribution instead of 1.96. Here are some critical values for a 95% confidence interval.

n	Critical Value
10	2.262
20	2.093
30	2.045
40	2.023
50	2.010
60	2.001
70	1.995
80	1.990
90	1.987
100	1.984

Multiple Train-and-Test

A single train-and-test experiment can be misleading.

1. Sampling variance especially for small samples.
2. Algorithm variance, more for NNs than for others.
3. Does not measure learning from T_2 .

Multiple train-and-test experiments can alleviate these difficulties.

Leave-One-Out Cross-Validation

For each $(\mathbf{x}_i, y_i) \in S$,

1. $T_i = S - \{(\mathbf{x}_i, y_i)\}$.
2. Use algorithm A on T_i to produce f_{ai}
3. Run f_{ai} on (\mathbf{x}_i, y_i)

Apply same statistics as holdout, using:

$$\delta_i = \begin{cases} 0 & \text{if } f_{ai}(\mathbf{x}_i) = y_i \\ 1 & \text{if } f_{ai}(\mathbf{x}_i) \neq y_i \end{cases} \text{ or 0-1 loss calculation}$$

Properties of Leave-One-Out

Generally, leave-one-out is the most accurate method, but also requires the most computation.

Each f_{ai} is likely very close to f_a because the sample only differs by one example.

Each example is used once as a test example.

Bootstrapping (see handout) might be more accurate for very small samples.

k -Fold Cross-Validation

1. Split S into k subsets F_1, F_2, \dots, F_k as evenly as possible.
2. For each fold F_i ,
 - (a) $T_i = S - F_i$
 - (b) Use algorithm A on T_i to produce f_{ai}
 - (c) Determine error rate (average 0-1 loss) of f_{ai} on F_i

Statistics for k -Fold Cross-Validation

Δ_i = average 0-1 loss of f_{ai} on F_i

$$u = \frac{\sum_{i=1}^k \Delta_i}{k} = \text{average 0-1 loss}$$

$$s^2 = \frac{\sum_{i=1}^k (\Delta_i - u)^2}{k - 1} = \text{sample variance}$$

$$t = \frac{s}{\sqrt{k}} = t \text{ statistic}$$

Use t -dist. critical value for $n = k$. E.g., for $k = 10$, confidence interval is $u \pm 2.262s/\sqrt{k}$.

Stratified k -Fold Cross-Validation

Randomly distribute the examples among the folds subject to the following conditions:

1. Each fold contains either $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$ examples.
2. If there are n_i examples of class i , then each fold contains either $\lfloor n_i/k \rfloor$ or $\lceil n_i/k \rceil$ examples of class i .

This ensures that each training set and fold approximates the prevalence of each class in the overall sample.

Properties of k -Fold Cross-Validation

10-fold CV is the most commonly used method.

It is not as accurate as leave-one-out, but it requires much less computation, though still over 10 times more than holdout.

Each example is used once as a test example.

Because each training set is 90% of the sample, some variance will result from algorithm.

Stratified CV reduces algorithm variance.