Hypothesis Evaluation

Hamid Beigy

Sharif University of Technology

Fall 1395
Table of contents

1 Introduction
2 Some performance measures of classifiers
3 Evaluating the performance of a classifier
4 Estimating true error
5 Confidence intervals
6 Paired $t$ Test
1. Introduction

2. Some performance measures of classifiers

3. Evaluating the performance of a classifier

4. Estimating true error

5. Confidence intervals

6. Paired $t$ Test
1. Machine Learning algorithms induce hypothesis that depend on the training set, and there is a need for statistical testing to
   - Assess expected performance of a hypothesis and
   - Compare expected Performances of two hypothesis to compare them.

2. Classifier evaluation criteria
   - **Accuracy (or Error)**  The ability of a hypothesis to correctly predict the label of new/previous unseen data.
   - **Speed**  The computational costs involved in generating and using a hypothesis.
   - **Robustness**  The ability of a hypothesis to make correct predictions given noisy data or data with missing values.
   - **Scalability**  The ability to construct a hypothesis efficiently given large amounts of data.
   - **Interpretability**  The level of understanding and insight that is provided by the hypothesis.
1. Given the observed accuracy of a hypothesis over a limited sample data, how well does this estimate its accuracy over additional examples.

2. Given one hypothesis outperforms another over sample data, how probable is that this hypothesis is more accurate in general.

3. When data is limited what is the best way to use this data to learn a hypothesis and estimate its accuracy.
Measuring performance of a classifier

1. Measuring performance of a hypothesis is partitioning data to
   - Training set
   - Validation set different from training set.
   - Test set different from training and validation sets.

2. Problems with this approach
   - Training and validation sets may be small and may contain exceptional instances such as noise, which may mislead us.
   - The learning algorithm may depend on other random factors affecting the accuracy such as initial weights of a neural network trained with BP. We must train/test several times and average the results.

3. Important points
   - Performance of a hypothesis estimated using training set conditioned on the used data set and cant used to compare algorithms in domain independent ways.
   - Validation set is used for model selection, comparing two algorithms, and decide to stop learning.
   - In order to report the expected performance, we should use a separate test set unused during learning.
Error of a classifier

**Definition (Sample error)**

The sample error (denoted $E_E(h)$) of hypothesis $h$ with respect to target concept $c$ and data sample $S$ of size $N$ is.

$$E_E(h) = \frac{1}{N} \sum_{x \in S} \mathbb{I}[c(x) \neq h(x)]$$

**Definition (True error)**

The true error (denoted $E(h)$) of hypothesis $h$ with respect to target concept $c$ and distribution $\mathcal{D}$ is the probability that $h$ will misclassify an instance drawn at random according to distribution $\mathcal{D}$.

$$E(h) = \mathbb{P}_{x \sim \mathcal{D}}[c(x) \neq h(x)]$$
Notions of errors

1. True error is

\[
\text{Instance space } X
\]

2. Our concern

- How we can estimate the true error \( E(h) \) of hypothesis using its sample error \( E_E(h) \) ?
- Can we bound true error of \( h \) given sample error of \( h \)?
1 Introduction

2 Some performance measures of classifiers

3 Evaluating the performance of a classifier

4 Estimating true error

5 Confidence intervals

6 Paired $t$ Test
Some performance measures of classifiers

1. **Error rate**  The error rate is the fraction of incorrect predictions for the classifier over the testing set, defined as

   \[ E_E(h) = \frac{1}{N} \sum_{x \in S} \mathbb{I}[c(x) \neq h(x)] \]

   Error rate is an estimate of the probability of misclassification.

2. **Accuracy**  The accuracy of a classifier is the fraction of correct predictions over the testing set:

   \[ \text{Accuracy}(h) = \frac{1}{N} \sum_{x \in S} \mathbb{I}[c(x) = h(x)] = 1 - E_E(h) \]

   Accuracy gives an estimate of the probability of a correct prediction.
What you can say about the accuracy of 90% or the error of 10%?

For example, if 3 – 4% of examples are from negative class, clearly accuracy of 90% is not acceptable.

Confusion matrix

Given $C$ classes, a confusion matrix is a table of $C \times C$. 
Some performance measures of classifiers

1. **Precision (Positive predictive value)** Precision is proportion of predicted positives which are actual positive and defined as:

   $$Precision(h) = \frac{TP}{TP + FP}$$

2. **Recall (Sensitivity)** Recall is proportion of actual positives which are predicted positive and defined as:

   $$Recall(h) = \frac{TP}{TP + FN}$$

3. **Specificity** Specificity is proportion of actual negative which are predicted negative and defined as:

   $$Specificity(h) = \frac{TN}{TN + FP}$$
Some performance measures of classifiers

1. **Balanced classification rate (BCR)** Balanced classification rate provides an average of recall (sensitivity) and specificity, it gives a more precise picture of classifier effectiveness. Balanced classification rate defined as

\[
BCR(h) = \frac{1}{2} \left[ \text{Specificity}(h) + \text{Recall}(h) \right]
\]

2. **F-measure** F-measure is harmonic mean between precision and recall and defined as

\[
F - \text{Measure}(h) = 2 \times \frac{\text{Precision}(h) \times \text{Recall}(h)}{\text{Precision}(h) + \text{Recall}(h)}
\]
Outline

1. Introduction

2. Some performance measures of classifiers

3. Evaluating the performance of a classifier

4. Estimating true error

5. Confidence intervals

6. Paired $t$ Test
Evaluating the performance of a classifier

1. Hold-out method
2. Random Sub-sampling
3. Cross validation method
4. Leave-one-out method
5. $5 \times 2$ Cross validation method
6. Bootstrapping method
Hold-out method

1. **Hold-out**  
   Hold-out partitions the given data into two independent sets: **training and test sets**.

   ![Diagram showing division of data into training and test sets]

   - Typically two-thirds of the data are allocated to the training set and the remaining one-third is allocated to the test set.
   - The training set is used to drive the model.
   - The test set is used to estimate the accuracy of the model.
Random sub-sampling method

- **Random sub-sampling**: Random sub-sampling is a variation of the hold-out method in which hold-out is repeated $k$ times.

- The estimated error rate is the **average of the error rates** for classifiers derived for the independently and randomly generated test partitions.

- Random subsampling can produce better error estimates than a single train-and-test partition.
**K-fold cross validation** The initial data are randomly partitioned into \( K \) mutually exclusive subsets or \textit{folds}, \( S_1, S_2, \ldots, S_K \), each of approximately equal size.

- Training and testing is performed \( K \) times.
- In iteration \( k \), partition \( S_k \) is used for test and the remaining partitions collectively used for training.
- The accuracy is the percentage of the total number of correctly classified test examples.
- The advantage of \( K \)-fold cross validation is that all the examples in the
Leave-one-out method

1. **Leave-one-out** Leave-one-out is a special case of $K$-fold cross validation where $K$ is set to the number of examples in the dataset.

- For a dataset with $N$ examples, perform $N$ experiments.
- For each experiment, use $N-1$ examples for training and the remaining example for testing.
5 × 2 Cross validation method repeats five times 2-fold cross validation method.

Training and testing is performed 10 times.

The estimated error rate is the average of the error rates for classifiers derived for the independently and randomly generated test partitions.
Bootstrapping method

1. **Bootstrapping**  The bootstrap uses sampling with replacement to form the training set.
   - Sample a dataset of $N$ instances $N$ times with replacement to form a new dataset of $N$ instances.
   - Use this data as the training set. An instance may occur more than once in the training set.
   - Use the instances from the original dataset that don't occur in the new training set for testing.
   - This method trains classifier on just 63% of the instances.
Outline

1. Introduction
2. Some performance measures of classifiers
3. Evaluating the performance of a classifier
4. Estimating true error
5. Confidence intervals
6. Paired $t$ Test
Estimating true error

1. How well does $E_E(h)$ estimate $E(h)$?
   - **Bias in the estimate** If training / test set is small, then the accuracy of the resulting hypothesis is a poor estimator of its accuracy over future examples.
     
     $bias = E[E_E(h)] - E(h)$.

     For unbiased estimate, $h$ and $S$ must be chosen independently.

   - **Variance in the estimate** Even with unbiased $S$, $E_E(h)$ may still vary from $E(h)$. The smaller test set results in a greater expected variance.

2. Hypothesis $h$ misclassifies 12 of the 40 examples in $S$. What is $E(h)$?

3. We use the following experiment
   - Choose sample $S$ of size $N$ according to distribution $D$.
   - Measure $E_E(h)$
   - $E_E(h)$ is a random variable (i.e., result of an experiment).
   - $E_E(h)$ is an unbiased estimator for $E(h)$.

4. Given observed $E_E(h)$, what can we conclude about $E(h)$?
$E_E(h)$ is a random variable with binomial distribution, for the experiment with different randomly drawn $S$ of size $N$, the probability of observing $r$ misclassified examples is

$$\mathbb{P}(r) = \frac{N!}{r!(N-r)!} E(h)^r [1 - E(h)]^{N-r}$$

For example for $N = 40$ and $E(h) = p = 0.2$, 

![Histogram of distribution of error](image)
Distribution of error

1. For binomial distribution, we have
   \[ E(r) = Np \]
   \[ Var(r) = Np(1 - p) \]

   We have shown that the random variable \( E_E(h) \) obeys a Binomial distribution.

2. Let \( p \) be the probability that the result of trial is a success.

3. The \( E_E(h) \) and \( E(h) \) are
   \[ E_E(h) = \frac{r}{N} \]
   \[ E(h) = p \]

   where
   - \( N \) is the number of instances in the sample \( S \),
   - \( r \) is the number of instances from \( S \) misclassified by \( h \), and
   - \( p \) is the probability of misclassifying a single instance drawn from \( D \).
1. It can be shown that $E_E(h)$ is unbiased estimator for $E(h)$.

2. Since $r$ is Binomially distributed, its variance is $Np(1 - p)$.

3. Unfortunately $p$ is unknown, but we can substitute our estimate $\frac{r}{N}$ for $p$.

4. In general, given $r$ errors in a sample of $N$ independently drawn test examples, the standard deviation for $E_E(h)$ is given by

\[
\text{Var} \left[ E_E(h) \right] = \sqrt{\frac{p(1 - p)}{N}} \approx \sqrt{\frac{E_E(h)(1 - E_E(h))}{N}}
\]
Outline

1. Introduction
2. Some performance measures of classifiers
3. Evaluating the performance of a classifier
4. Estimating true error
5. Confidence intervals
6. Paired $t$ Test
1. One common way to describe the uncertainty associated with an estimate is to give an interval within which the true value is expected to fall, along with the probability with which it is expected to fall into this interval.

2. How can we derive confidence intervals for $E_E(h)$?

3. For a given value of $M$, how can we find the size of the interval that contains $M\%$ of the probability mass?

4. Unfortunately, for the Binomial distribution this calculation can be quite tedious.

5. Fortunately, however, an easily calculated and very good approximation can be found in most cases, based on the fact that for sufficiently large sample sizes the Binomial distribution can be closely approximated by the Normal distribution.
In Normal Distribution $N(\mu, \sigma^2)$, $M\%$ of area (probability) lies in $\mu \pm z_M \sigma$, where

<table>
<thead>
<tr>
<th>$M%$</th>
<th>50%</th>
<th>68%</th>
<th>80%</th>
<th>90%</th>
<th>95%</th>
<th>98%</th>
<th>99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z_M$</td>
<td>0.67</td>
<td>1.0</td>
<td>1.28</td>
<td>1.64</td>
<td>1.96</td>
<td>2.33</td>
<td>2.58</td>
</tr>
</tbody>
</table>

![Diagram of normal distribution with confidence intervals and z-scores for $M\%$ of area.](image)
Comparing two hypotheses

- Test $h_1$ on sample $S_1$ and test $h_2$ on $S_2$.
  1. Pick parameter to estimate $d = E(h_1) - E(h_2)$.
  2. Choose an estimator $\hat{d} = E_E(h_1) - E_E(h_2)$.
  3. Determine probability distribution that governs estimator:
  4. $E_E(h_1)$ and $E_E(h_2)$ can be approximated by Normal Distribution.
  5. Difference of two Normal distributions is also a Normal distribution, $\hat{d}$ will also approximated by a Normal distribution with mean $d$ and variance of this distribution is equal to sum of variances of $E_E(h_1)$ and $E_E(h_2)$ . Hence, we have

$$Var[\hat{d}] = \sqrt{\frac{E_E(h_1)(1 - E_E(h_1))}{N_1} + \frac{E_E(h_2)(1 - E_E(h_2))}{N_2}}$$

6. Find interval $(L, U)$ such that $\chi\%$ of probability mass falls in interval

$$\hat{d} \pm z_{\chi} \sqrt{\frac{E_E(h_1)(1 - E_E(h_1))}{N_1} + \frac{E_E(h_2)(1 - E_E(h_2))}{N_2}}$$
Comparing two algorithms

For comparing two learning algorithms $L_A$ and $L_B$, we would like to estimate

$$\mathbb{E}_{S \sim D}[E(L_A(S)) - E(L_B(S))]$$

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

This shows 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 $D$.

But, given limited data $S_0$, what is a good estimator?

1. Could partition $S_0$ into training set $S_0^{tr}$ and test set $S_0^{ts}$, and measure

$$\hat{d} = E_{E}^{S_0^{ts}}(L_A(h_1)) - E_{E}^{S_0^{ts}}(L_B(h_2)).$$

where $h_1$ and $h_2$ are trained using training set $S_0^{tr}$ and $E_{E}^{S_0^{ts}}$ is empirical error using test set $S_0^{ts}$.

2. Even better, repeat this many times and average the results.
1 Introduction

2 Some performance measures of classifiers

3 Evaluating the performance of a classifier

4 Estimating true error

5 Confidence intervals

6 Paired $t$ Test
consider the following estimation problem

1. We are given the observed values of a set of independent, identically distributed random variables $Y_1, Y_2, \ldots, Y_K$.
2. We wish to estimate the mean $\mu$ of the probability distribution governing these $Y_i$.
3. The estimator we will use is the sample mean $\bar{Y} = \frac{1}{K} \sum_{k=1}^{K} Y_k$

The task is to estimate the sample mean of a collection of independent, identically and Normally distributed random variables.

The approximate $M\%$ confidence interval for estimating $\bar{Y}$ is given by

$$\bar{Y} \pm t_{M,K-1} S_{\bar{Y}}$$

where

$$S_{\bar{Y}} = \sqrt{\frac{1}{K(K-1)} \sum_{k=1}^{K} (Y_i - \bar{Y})^2}$$
Paired $t$ Test

Values of $t_{M,K-1}$ for two-sided confidence intervals.

<table>
<thead>
<tr>
<th>$M$</th>
<th>90%</th>
<th>95%</th>
<th>98%</th>
<th>99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K - 1 = 2$</td>
<td>2.92</td>
<td>4.30</td>
<td>6.96</td>
<td>4.03</td>
</tr>
<tr>
<td>$K - 1 = 120$</td>
<td>1.66</td>
<td>1.98</td>
<td>2.36</td>
<td>2.62</td>
</tr>
<tr>
<td>$K - 1 = \infty$</td>
<td>1.64</td>
<td>1.96</td>
<td>2.33</td>
<td>2.58</td>
</tr>
</tbody>
</table>

As $K \to \infty$, $t_{M,K-1}$ approaches $z_M$. 
1. ROC puts false positive rate \((FPR = \frac{FP}{NEG})\) on \(x\) axis.
2. ROC puts true positive rate \((TPR = \frac{TP}{POS})\) on \(y\) axis.
3. Each classifier represented by a point in ROC space corresponding to its \((FPR, TPR)\) pair.
A note on parameter tuning

- Some learning schemes operate in two stages:
  - **Stage 1**: builds the basic structure
  - **Stage 2**: optimizes parameter settings
- It is important that the test data is not used in any way to create the classifier
- The test data cant be used for parameter tuning!
- Proper procedure uses three sets: training data, validation data, and test data. Validation data is used to optimize parameters.

No Free Lunch Theorem

- For any ML algorithm there exist data sets on which it performs well and there exist data sets on which it performs badly!
- We hope that the latter sets do not occur too often in real life.
References