Machine Learning and Data Analysis
Empirical Validation of Hypotheses

Filip Železný

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics
Intelligent Data Analysis lab
http://ida.felk.cvut.cz

January 6, 2012
Remind: we want to learn \( f \) which minimizes risk \( R(f) \).

<table>
<thead>
<tr>
<th>Estimates of ( R(f) )</th>
<th>theoretical ( f ) a function of properties such as ( m, \mathcal{V}(\mathcal{F}), \delta )</th>
<th>empirical a number computed for a particular sample and learner</th>
</tr>
</thead>
<tbody>
<tr>
<td>worst-case ( \text{an upper bound on } R(f) )</td>
<td>PAC-theory</td>
<td>not interesting</td>
</tr>
<tr>
<td>average-case ( \text{the expected value of } R(f) )</td>
<td>not available</td>
<td>this lecture</td>
</tr>
</tbody>
</table>

Theoretical: reveal relationships, useful for the design of learning algorithms or experiments.

Expected-case: useful in applications of existing algorithms.
Risk Estimator

Let $S$ be a set of possible i.i.d samples. Let $L : S \rightarrow \mathcal{F}$ be a (deterministic) learning algorithm.

A risk estimator takes $L$ and $S$ and produces a number $\hat{R}(L, S)$ that should approximate $R(L(S))$, i.e. minimize

$$E_S[(R(L(S)) - \hat{R}(L, S))^2]$$

Since $S$ is drawn randomly, $L(S)$ is random, and thus $R(L(S))$ and $\hat{R}(L, S)$ are also random.

The expectation is over a probability distribution $P_S$ on samples.

For a fixed $|S| = m$, $P_S$ can be derived from $P_X$.

Our subsequent analyses of estimators will relate to any $P_S$ so we do not need to specify it.
Bias and Variance of Risk Estimates

For clarity, denote $f^S \equiv L(S)$ and $\hat{R} \equiv \hat{R}(L, S)$.

The error of an estimator can be decomposed into two components:

Bias:

$$\text{bias}(\hat{R}) = E_S[\hat{R} - R(f^S)]$$

Variance:

$$\text{var}(\hat{R}) = E_S[(E_S[\hat{R}] - \hat{R})^2]$$

This can be shown the same way we used in deriving the bias-variance trade-off in regression.

Risk estimators are usually characterized in terms of their bias and variance.
Split Sample Estimator

1. Randomly splits $S$ into $S_{\text{train}}$ and $S_{\text{test}}$
2. Learns $f = L(S_{\text{train}})$ and outputs $\hat{R}_{ss} = R(S_{\text{test}}, f)$

$f$ and $\hat{R}_{ss}$ here depend on the outcome of two random events:

1. sampling of $S$ from $X$
2. splitting of $S$, i.e. ‘subsampling’ $S_{\text{test}}$ from $S$ and letting $S_{\text{train}} = S \setminus S_{\text{test}}$
A split where $\mu = |S_{\text{test}}|/|S|$ will be called a $\mu$-split.

Given the additional random event (sample splitting), we define the *conditional* bias of $\hat{R}_{ss}$

$$\text{bias}_{\mu,S}(\hat{R}_{ss}) = E_{\mu,S}[\hat{R}_{ss}] - R(f^S)$$

where $E_{\mu,S}$ denotes the expectation over all $\mu$-splits of a fixed sample $S$.

The (unconditional) bias can be expressed as

$$\text{bias}(\hat{R}_{ss}) = E_S E_{\mu,S}[\hat{R}_{ss} - R(f^S)]$$

over all samples $S \in \mathcal{S}$ and all their $\mu$-splits.

Analogically for the conditional and unconditional variance.
Bias of $\hat{R}_{ss}$

Assuming that more examples allow learning a better classifier implies that

$$ R(f) > R(f^S) $$ (1)

since $f$ is trained on $S_{\text{train}}$, $|S_{\text{train}}| < |S|$.

Since $\hat{R}_{ss}$ is the empirical risk of $f$ is tested on a sample independent from $S_{\text{train}}$, $\hat{R}_{ss}$ is an unbiased estimator of $R(f)$:

$$ \mathbb{E}_{S} \mathbb{E}_{\mu, S}[\hat{R}_{ss} - R(f)] = 0 $$

Considering Eq. 1, $\hat{R}_{ss}$ thus has a positive bias in estimating $R(f^S)$, i.e.

$$ \text{bias}(\hat{R}_{ss}) > 0 $$
Estimating $R(f^S)$ or $R(f)$

Given that $\hat{R}_{ss}$ is an unbiased estimate of $R(f)$, we may choose to simply output $f$ with $\hat{R}_{ss}$ as the validated product of learning.

This is a compromise since $f^S$ would have likely been a better classifier than $f$.

It is a reasonable approach when $R(f)$ is not too much higher than $R(f^S)$. This occurs when $S_{\text{train}}$ is large enough so that additional data do not contribute significantly to improve $f$, i.e. the learner is saturated.

In other cases it is preferable to produce $f^S$ even if its risk estimate $\hat{R}_{ss}$ is biased.

Whether a learner is saturated follows from the learning curve.
Learner Saturation

\[ R \]

large difference

not saturated

almost saturated

small difference

min. \( R \) of \( L \)
Bayes \( R \)

\[ \left| S \right| \quad \left| S'_{\text{train}} \right| \quad \left| S' \right| \quad \left| S \right| \]

\[ \left| S_{\text{train}} \right| \]
Distribution of $\hat{R}_{ss}$

Assume a fixed $S$ and $\mu$. $\hat{R}_{ss}$ is an outcome of $\mu|S|$ Bernoulli trials (correct/incorrect classification) and for sufficiently large $S$, it is distributed normally.
Bias of $\hat{R}_{ss}$

The conditional bias

$$\text{bias}_{\mu,S}(\hat{R}_{ss})$$

grows with growing $\mu$ since also $|S_{\text{train}}|$ decays.

The trend holds as well for the unconditional bias

$$\text{bias}(\hat{R}_{ss})$$

i.e. when the conditional biases are aggregated over all samples $S \in \mathcal{S}$.

The rate of decay depends on the \textit{learning curve} of the learner $L$. 
Bias of $\hat{R}_{ss}$ (cont’d)

Consider two samples $S_1, S_2$ from the same distribution $P_{XY}$.

$|S_1| = 10, |S_2| = 60, \mu = 0.5$. 

![Graph showing large and small bias]
Conditional Variance of $\hat{R}_{ss}$

Assuming (for simplicity) that the same classifier is learned for all $\mu$-splits, the conditional variance of $\hat{R}_{ss}$ would decay with growing $\mu|S|$ as

$$\text{var}_{\mu,S}(\hat{R}_{ss}) = \frac{R(f)(1 - R(f))}{\mu|S|}$$

Rephrased: with larger test splits, estimates of $\hat{R}_{ss}$ are more reliable.

However, the assumption holds (approximately) only if $|S_{\text{train}}| = (1 - \mu)|S|$ is large enough so that $L$ is saturated.

Otherwise, different $f$ are learned from different $\mu$-splits. Since $\hat{R}_{ss}$ depends on $f$, $\text{var}_{\mu,S}(\hat{R}_{ss})$ also grows with $\text{var}_{\mu,S}(R(f))$. That in turn grows with $\mu$ with a rate depending on the learner $L$.

Thus if $|S_{\text{train}}| = (1 - \mu)|S|$ is small so that $L$ is not saturated, the trends in conditional variance cannot be predicted.
Unconditional Variance of $\hat{R}_{ss}$ (cont’d)

According to [Hastie et al., Elements of Statistical Learning, Springer, 2009], the unconditional variance

$$\text{var}(\hat{R}_{ss})$$

typically decays with growing $\mu$.

This is because for small $\mu$, the individual train splits $S_{\text{train}}$ are very similar to each other, causing high positive correlation of the measurements $e(S_{\text{test}}, f)$.

The estimate $\hat{R}_{ss}$ is thus ‘overfit’ to sample $S$. This implies large variance over different samples, i.e. high unconditional variance.

Note: Since part of the variance is due to the conditional variance, decay of $\text{var}(\hat{R}_{ss})$ with $\mu$ may be overridden by the possible growth of $\text{var}_{\mu,S}(\hat{R}_{ss})$ with $\mu$ when the learner is not saturated. (We will see an example later).
Bias-Variance Trade-off in Risk Estimation (cont’d)

Usual choice \( \mu = 0.3 \).
Part of the variance \( \text{var}(\hat{R}_{ss}) \) is the conditional variance \( \text{var}_{\mu,S}(\hat{R}_{ss}) \) which is due to the random splitting of \( S \).
**Complete Subsampling**

\[ \text{var}_{\mu,S}(\hat{R}_{ss}) \text{ can be completely eliminated by averaging estimates over all possible } \mu\text{-splits of } S \]

\[ \hat{R}_{cs} = \frac{1}{K} \sum_{S_{\text{test}} \subset 2^S} R(S_{\text{test}}, L(S \setminus S_{\text{test}})) \]

\[ |S_{\text{test}}| = \mu|S| \]

where

\[ K = \binom{|S|}{\mu|S|} \]

\( \hat{R}_{cs} \) is the *complete subsampling* estimate.

\[ \text{var}_{\mu,S}(\hat{R}_{cs}) = 0, \text{ but (conditional) bias remains,} \]

\[ \text{bias}_{\mu,S}(\hat{R}_{cs}) = \text{bias}_{\mu,S}(\hat{R}_{ss}), \text{ bias}(\hat{R}_{cs}) = \text{bias}(\hat{R}_{ss}) \]
Split Sample vs. Complete Subsampling

\[ P(\hat{R}_{ss}|\mu, S) \]

\[ \uparrow \var_{\mu,S}(\hat{R}_{ss}) \leq \var(\hat{R}_{ss}) \downarrow \]

\[ P(\hat{R}_{cs}) \leq \var(\hat{R}_{cs}) \downarrow \]

\[ \mathbb{E}_S[R(f)] \]

\[ \mathbb{E}_S[R(f)] \]
Complete subsampling and Leave-One-Out Estimate

Complete subsampling is extremely computationally difficult. Requires \( \binom{|S|}{\mu|S|} \) learning and testing sessions.

The easiest are the two extreme cases \( \mu|S| = 1 \) and \( \mu|S| = |S| - 1 \) requiring ‘only’ \( |S| \) learning and testing sessions.

\( \mu|S| = |S| - 1 \) is not useful due to the extremely high bias (learning from 1 example).

The \( \mu|S| = 1 \) case is known as leave one out estimate. We denote it \( \hat{R}_{lo} \).
Leave-One-Out: Bias and Variance

\( \hat{R}_{lo} \) has the smallest possible bias (all but one examples used to learn \( f \)).

Compared to other complete subsampling cases, it has high variance \( \text{var}(\hat{R}_{lo}) \) due to

- The positive correlations of the summands
  
  \[ R(S_{\text{test}}, L(S \setminus S_{\text{test}})) \]

  caused by the extreme similarity of the training subsamples \( S_{\text{train}} = S \setminus S_{\text{test}} \), each two differing only by 2 examples. (The estimate is ‘overfit’ to \( S \)).

- The low number \( |S| \) of summands, compared to \( \left( \frac{|S|}{\mu|S|} \right) \).
Cross-Validation

\textit{N-fold cross-validation} is a computationally feasible approximation to complete subsampling with \(1/N\)-splits.

\(S\) is randomly partitioned into sets (\textit{folds}) \(S_1, S_2, \ldots S_N\) of approximately equal size and the estimate is computed as the average

\[
\hat{R}_{cv} = \frac{1}{N} \sum_{i=1}^{N} R(S_i, L(S \setminus S_i))
\]

Cross-validation thus requires \(N\) sessions of learning and testing.

For \(N = |S|\), \(N\)-fold crossvalidation \(\hat{R}_{cv}\) is the leave-one-out estimate \(R_{lo}\).
Cross-Validation: Variance

\( \hat{R}_{cv} \) has non-zero conditional variance \( \text{var}_{N,S}(\hat{R}_{cv}) \) due to the random splitting into folds, up to the leave-one-out case where \( N = |S| \) and \( \text{var}_{N,S}(\hat{R}_{cv}) = 0 \).

The conditional variance (and consequently also the unconditional variance) can be reduced by averaging the results of \( L \) cross-validations with different splittings. This *repeated \( N \)-fold cross-validation estimate* \( \hat{R}_{rcv} \) approaches complete subsampling with \( 1/N \)-splits as \( L \to \infty \) and

\[
\lim_{L \to \infty} \text{var}_{N,S}(\hat{R}_{cv}) = 0
\]

According to experimental results [Molinaro et al., Bioinformatics, 2005] with real-life data and conventional learners, the *unconditional variance* \( \text{var}(\hat{R}_{cv}) \) of 10-fold cross-validation is comparable to \( \text{var}(\hat{R}_{lo}) \), however, much less computation is required (10 vs. \( N \) learning sessions).
Cross-Validation: Bias

$\text{bias}_{N,S}(\hat{R}_{cv})$ decays with increasing number of folds $N$ (since the training subsamples grow) to the minimum

$$\text{bias}_{N,S}(\hat{R}_{lo}) > 0$$

achieved the leave-one-out case.

For $|S| >> N$, the conditional bias can be reduced by \textit{stratification}. Stratification is an adjustment of random splitting into folds making sure that the distribution of example classes in each fold is (approximately) equal to the class distribution in $S$. 
Leave-one-out vs. 10-fold cross-validation

According to [Molinaro et al., Bioinformatics, 2005], the leave-one-out estimate has smaller error

$$E_S[(R(f^S) - \hat{R})^2]$$

than the 10-fold cross-validation estimate (and all other estimates) on real-life (genomic) data sets.
According to [Hastie et al., Springer, 2009], the leave-one-out estimate has **larger** conditional absolute error

\[ E_{N,S}[\| (R(f^S) - \hat{R}) \|] \]

than the 10-fold cross-validation estimate on simulated data sets.

However, both sources recommend the **10-fold cross-validation** (preferably stratified and repeated) as a good trade-off between estimate error and computational complexity.
Selection of Learners or Parameters

A set of learners is available $\mathcal{L} = \{L_1, \ldots, L_l\}$. $\mathcal{L}$ may refer to a single algorithm with $l$ different values of a parameter (e.g. the maximum number of literals in a conjunction).

The best learner for the available sample $S$ may be selected as

$$\arg\min_{L_i \in \mathcal{L}} \hat{R}(L_i, S)$$

Since test splits were used for the selection, $\hat{R}(L_i, S)$ would no longer be valid risk estimate of $L_i$ (it will typically have a negative bias).

Therefore, selection must be based on internal estimation.
Internal and External Estimation

When selection of a learner from a set $\mathcal{L} = \{L_1, \ldots, L_l\}$ ($L_i$ may correspond to different parameter values of different kind of algorithms) is part of learning, we formally consider a learner $L_\mathcal{L}$ that learns

$$f^S = L_\mathcal{L}(S) = L(S)$$

where

$$L = \arg \min_{L_i \in \mathcal{L}} \hat{R}(L_i, S)$$

$\hat{R}$ is some risk estimate (usually cross-validation), called the \textit{internal} estimate. Risk of $f^S$ is estimated as

$$\hat{R}(L_\mathcal{L}, S)$$

where $\hat{R}$ is some risk estimate (usually split-sample), called the \textit{external} estimate. Note that computation of $\hat{R}(L_\mathcal{L}, S)$ involves splitting of $S$, and then splitting the splits of $S$!
Example: Learner/Parameter Selection

Goal: Given sample $S$, select a learner (or parameter) and learn a classifier.

Case 1: we are not interested in the risk of $f^S$.

Using 5-fold cross-validation:

1. Perform cross-validation on $S$ for each $L \in \mathcal{L}$

   Select $L_i$ that minimizes cross-validation error

2. With $L_i$, learn classifier on the entire sample $S$, i.e. $f^S = L_i(S)$
**Example: Learner/Parameter Selection**

Case 2: we are interested in the risk of $f^S$. Now we must apply both external and internal validation.

1. **External validation using split-sample method:**
   
   ![Split-sample validation diagram](image)

2. **Perform internal cross-validation on $S_{train}$ for each $L \in \mathcal{L}$**
   
   ![Cross-validation diagram](image)

   Select $L_i$ that minimizes cross-validation error

3. **With $L_i$, learn classifier on sample $S_{train}$, i.e. $f = L_i(S_{train})$**

4. **Risk of $f^S$ estimated as $e(S_{test}, f)$**

5. **With $L_i$, learn classifier on sample $S$, i.e. $f^S = L_i(S)$**