OPPA European Social Fund
Prague & EU: We invest in your future.

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

Risk Estimates

Remind: we want to learn f which minimizes risk $R(f)$.

Estimates of $R(f)$	theoretical a *function* of properties such as m, $\mathcal{V}(\mathcal{F})$, δ	empirical a *number* computed for a particular sample and learner
worst-case an *upper bound* on $R(f)$	PAC-theory	not interesting
average-case the *expected value* of $R(f)$	not available	this lecture

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

Expected-case: useful in applications of existing algorithms.

Risk Estimator

Let $\mathcal{S}$ be a set of possible i.i.d samples. Let $L : \mathcal{S} \to \mathcal{F}$ 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

$$\mathbf{E}_{\mathcal{S}}[(R(L(S)) - \hat{R}(L, S))^2]$$

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 $P_{\mathcal{S}}$ on samples.

For a fixed $|S| = m$, $P_{\mathcal{S}}$ can be derived from P_X.

Our subsequent analyses of estimators will relate to any $P_{\mathcal{S}}$ 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:

$$\text{bias}(\hat{R}) = \mathbf{E}_{\mathcal{S}}[\hat{R} - R(f^S)]$$

Variance:

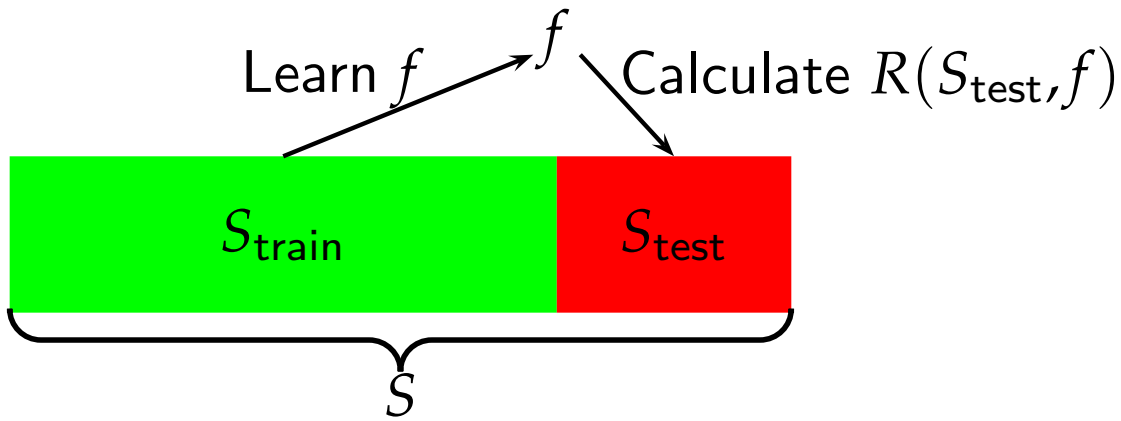
$$\text{var}(\hat{R}) = \mathbf{E}_{\mathcal{S}}[\left(\mathbf{E}_{\mathcal{S}}[\hat{R}] - \hat{R}\right)^2]$$

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



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

1. sampling of S from X
2. splitting of S, i.e. 'subsampling' S_{test} from S and letting $S_{\text{train}} = S \setminus S_{\text{test}}$

Split-Sample Estimator Bias and Variance

A split where $\mu = |S_{\text{test}}|/|S|$ will be called a μ-split.

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

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

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

The (unconditional) bias can be expressed as

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

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

Analogically for the conditional and unconditional variance.

Bias of $\hat{R}_{\text{ss}}$

Assuming that more examples allow learning a better classifier implies that

$$R(f) > R(f^S) \tag{1}$$

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

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

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

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

$$\text{bias}(\hat{R}_{\text{ss}}) > 0$$

Estimating $R(f^S)$ or $R(f)$

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

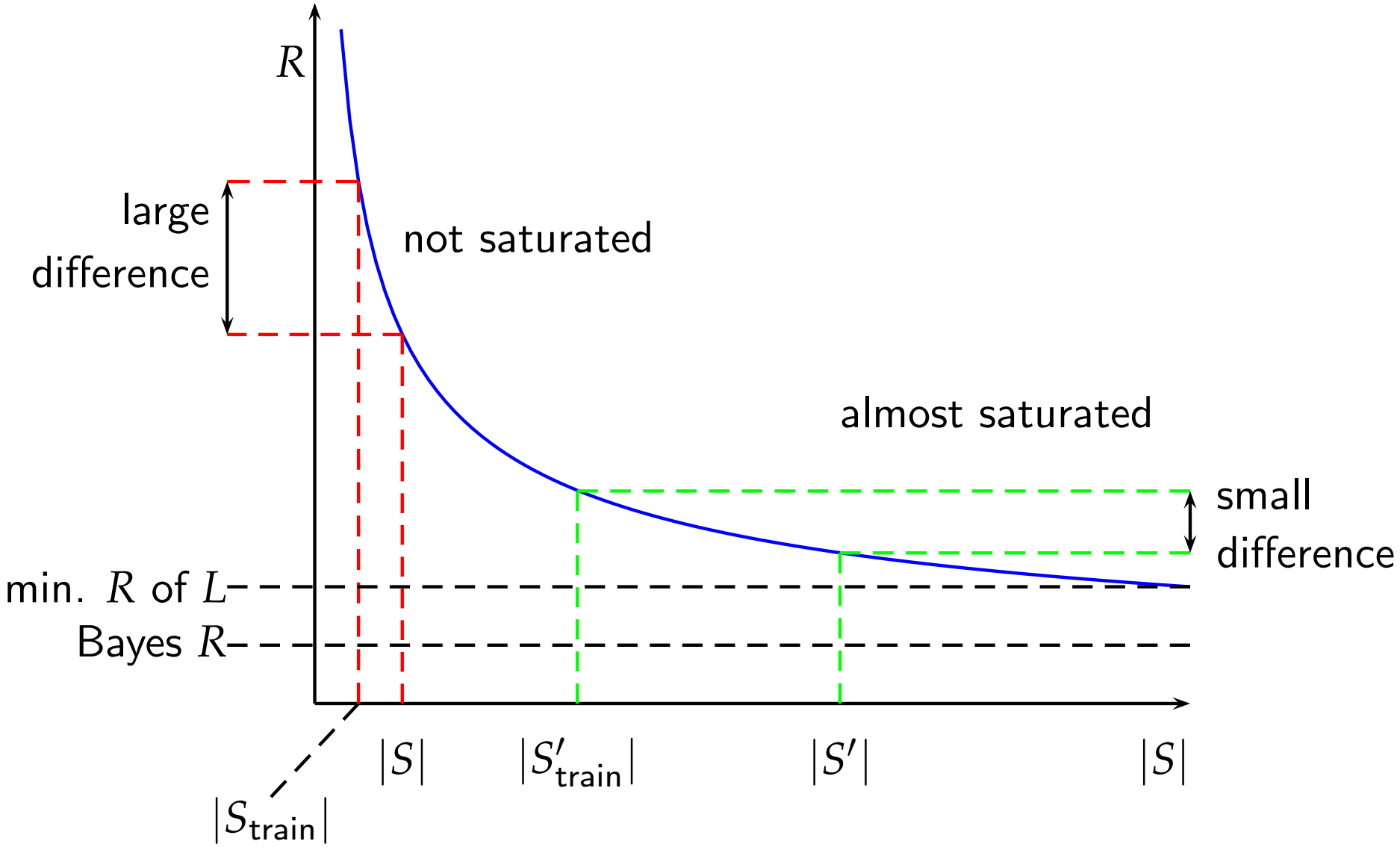
This is a compromise since f^S would have likely been a better classifier than f.

It is a reasonable approach when $R(f)$ is not too much higher than $R(f^S)$. This occurs when S_{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 $\hat{R}_{\text{ss}}$ is biased.

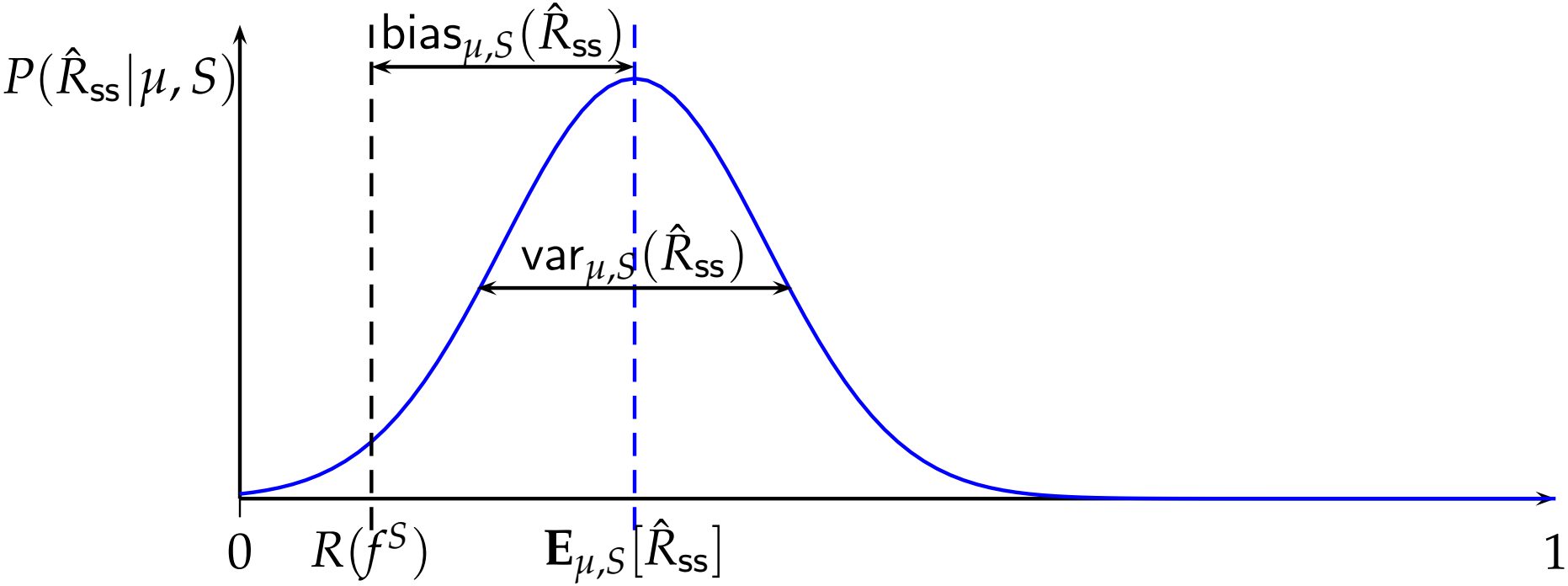
Whether a learner is saturated follows from the *learning curve*.

Learner Saturation



Distribution of $\hat{R}_{\text{ss}}$

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



Bias of $\hat{R}_{\text{ss}}$

The conditional bias

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

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

The trend holds as well for the unconditional bias

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

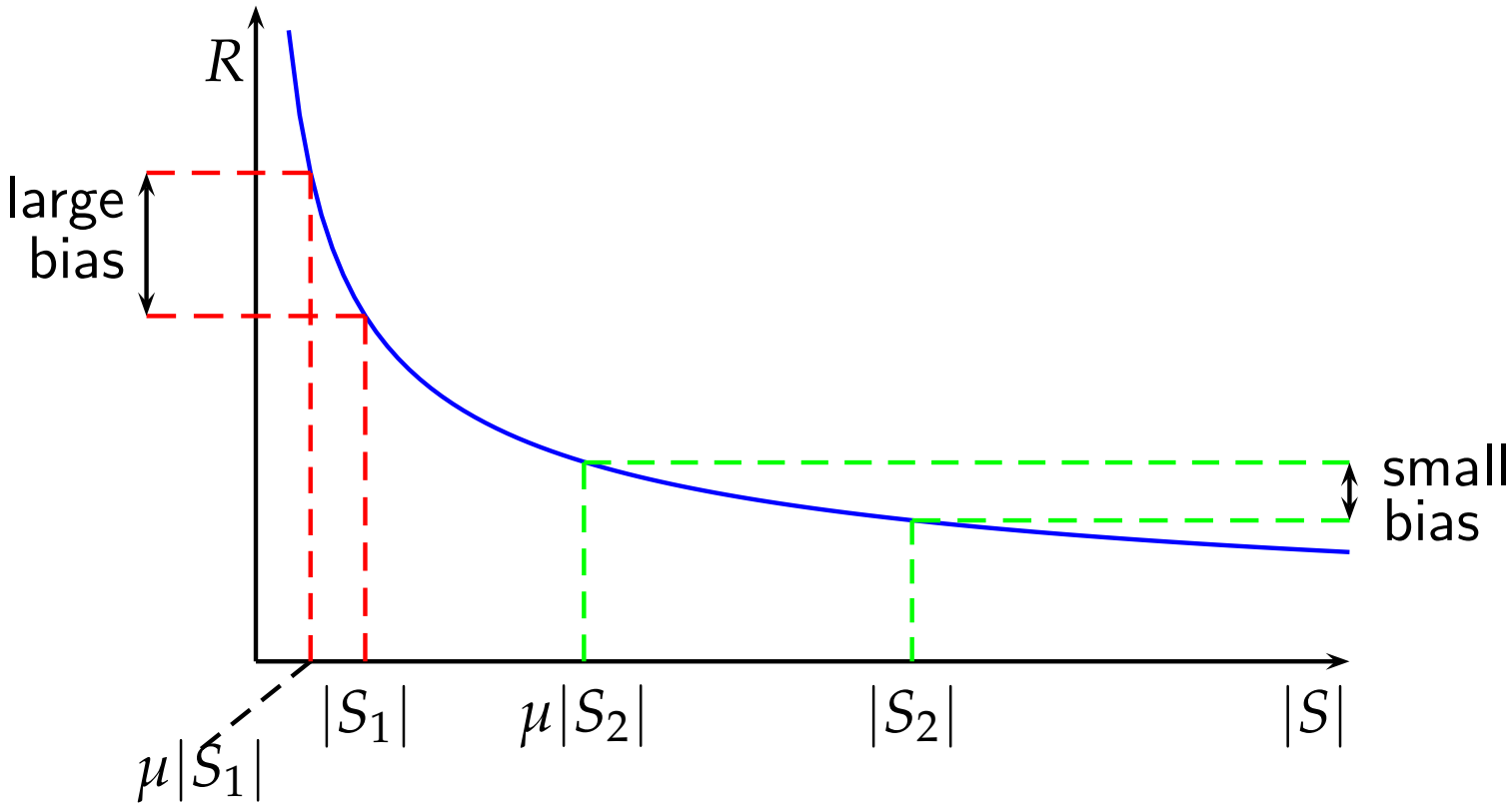
i.e. when the conditional biases are aggregated over all samples $S \in \mathcal{S}$.

The rate of decay depends on the *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}.

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



Conditional Variance of $\hat{R}_{\text{ss}}$

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

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

Rephrased: with larger test splits, estimates of $\hat{R}_{\text{ss}}$ 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 μ-splits. Since $\hat{R}_{\text{ss}}$ depends on f, $\text{var}_{\mu,S}(\hat{R}_{\text{ss}})$ also grows with $\text{var}_{\mu,S}(R(f))$. That in turn grows with μ 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 $\hat{R}_{\text{ss}}$ (cont'd)

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

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

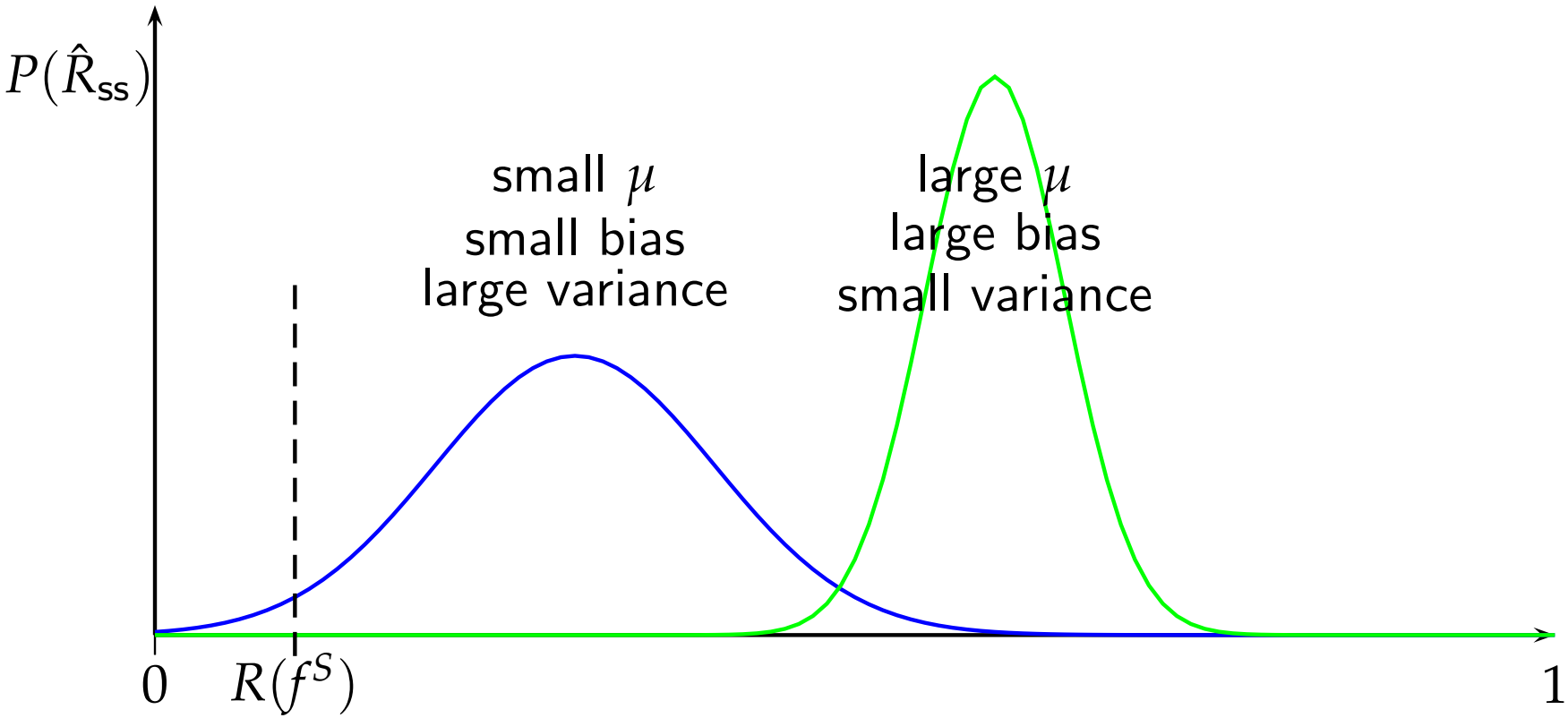
typically decays with growing μ.

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

The estimate $\hat{R}_{\text{ss}}$ 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 $\text{var}(\hat{R}_{\text{ss}})$ with μ may be overridden by the possible growth of $\text{var}_{\mu,S}(\hat{R}_{\text{ss}})$ with μ when the learner is not saturated. (We will see an example later).

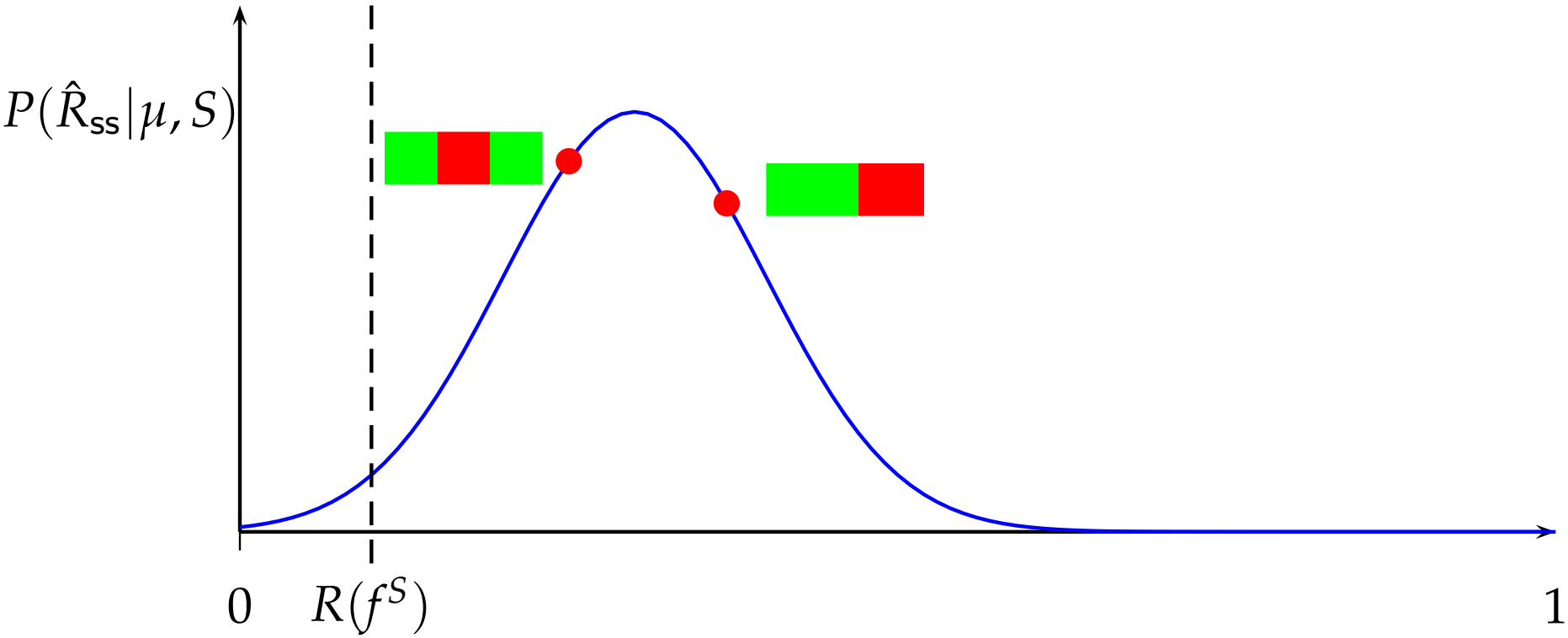
Bias-Variance Trade-off in Risk Estimation (cont'd)



Usual choice $\mu = 0.3$.

Variance of $\hat{R}_{\text{ss}}$ due to Sample Splitting

Part of the variance $\text{var}(\hat{R}_{\text{ss}})$ is the conditional variance $\text{var}_{\mu,S}(\hat{R}_{\text{ss}})$ which is due to the random splitting of S.



Complete Subsampling

$\mathrm{var}_{\mu,S}(\hat{R}_{\mathsf{ss}})$ can be completely eliminated by averaging estimates over all possible μ-splits of S

$$\hat{R}_{\mathsf{cs}} = \frac{1}{K} \sum_{\substack{S_{\mathsf{test}} \subset 2^S \\ |S_{\mathsf{test}}| = \mu|S|}} R(S_{\mathsf{test}}, L(S \setminus S_{\mathsf{test}}))$$

where

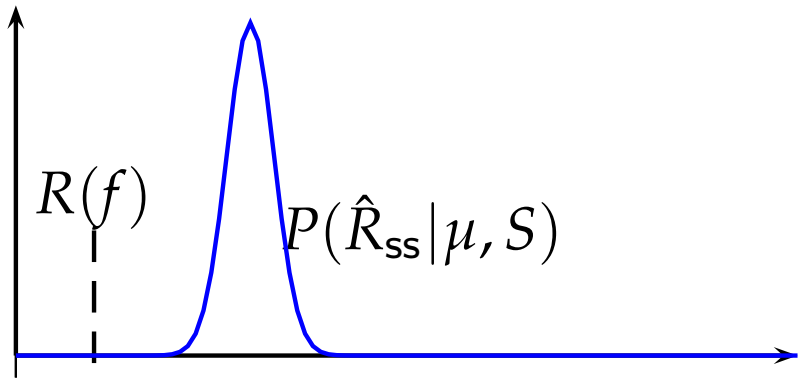
$$K = \left(\begin{array}{c} |S| \\ \mu|S| \end{array} \right)$$

$\hat{R}_{\mathsf{cs}}$ is the *complete subsampling* estimate.

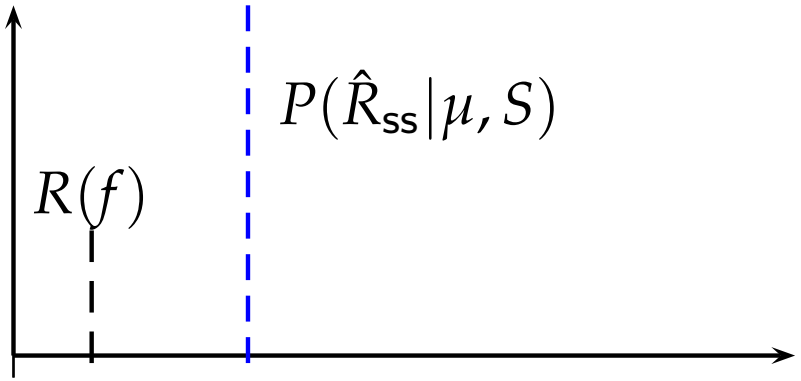
$\mathrm{var}_{\mu,S}(\hat{R}_{\mathsf{cs}}) = 0$, but (conditional) bias remains,
$\mathrm{bias}_{\mu,S}(\hat{R}_{\mathsf{cs}}) = \mathrm{bias}_{\mu,S}(\hat{R}_{\mathsf{ss}})$, $\mathrm{bias}(\hat{R}_{\mathsf{cs}}) = \mathrm{bias}(\hat{R}_{\mathsf{ss}})$

Split Sample vs. Complete Subsampling

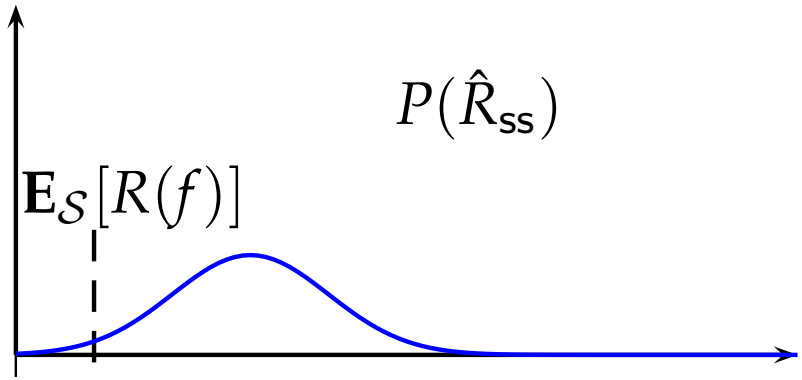
Split sample	Complete Subsampling

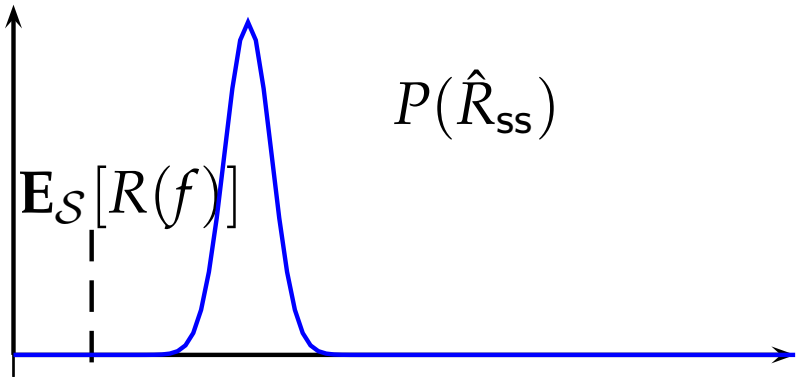


$\uparrow \text{var}_{\mu,S}(\hat{R}_{ss}) \leq \text{var}(\hat{R}_{ss}) \downarrow$



$\uparrow 0 = \text{var}_{\mu,S}(\hat{R}_{cs}) \leq \text{var}(\hat{R}_{cs}) \downarrow$





Complete Subsampling and Leave-One-Out Estimate

Complete subsampling is extremely computationally difficult. Requires $\begin{pmatrix} |S| \\ \mu|S| \end{pmatrix}$ 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 $\hat{R}_{\text{lo}}$.

Leave-One-Out: Bias and Variance

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

Compared to other complete subsampling cases, it has high variance $\text{var}(\hat{R}_{\text{lo}})$ 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 $\begin{pmatrix} |S| \\ \mu|S| \end{pmatrix}$.

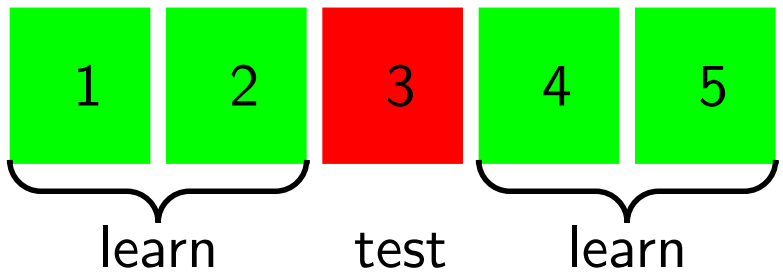
Cross-Validation

N-fold cross-validation is a computationally feasible approximation to complete subsampling with $1/N$-splits.

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

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

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



For $N = |S|$, N-fold crossvalidation $\hat{R}_{\text{cv}}$ is the leave-one-out estimate R_{lo}.

Cross-Validation: Variance

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

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* $\hat{R}_{\text{rcv}}$ approaches complete subsampling with $1/N$-splits as $L \to \infty$ and

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

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

Cross-Validation: Bias

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

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

achieved the leave-one-out case.

For $|S| >> N$, the conditional bias can be reduced by *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

$$\mathbf{E}_{\mathcal{S}}[(R(f^S) - \hat{R})^2]$$

than the 10-fold cross-validation estimate (and all other estimates) on real-life (genomic) data sets.

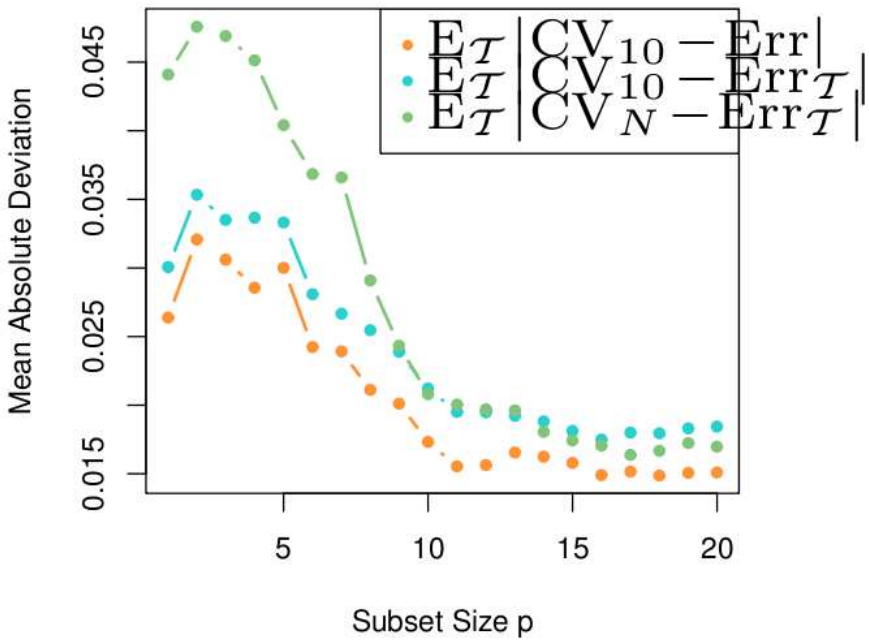
Estimator	p	Algorithm	Estimation	SD	Bias	MSE
$\tilde{\theta}_n$	0.87	LDA	0.026	0.028		
		DDA	0.073	0.058		
		NN	0.010	0.017		
		CART	0.099	0.092		
v-fold CV	0.5	LDA	0.067	0.060	0.041	0.005
		DDA	0.106	0.079	0.033	0.009
		NN	0.011	0.025	0.001	**0**
		CART	0.304	0.088	0.205	0.063
	0.2	LDA	0.034	0.045	0.008	0.002
		DDA	0.085	0.049	0.012	0.003
		NN	0.011	0.024	0.001	**0**
		CART	0.158	0.072	0.059	0.012
	0.1	LDA	0.032	0.041	0.006	**0.001**
		DDA	0.074	0.048	**0.001**	**0.002**
		NN	0.010	0.021	**0**	**0**
		CART	0.118	0.063	0.019	**0.006**
LOOCV	0.025	LDA	0.028	0.040	**0.002**	**0.001**
		DDA	0.072	0.049	**−0.001**	**0.002**
		NN	0.010	0.022	**0**	**0**
		CART	0.110	0.075	**0.011**	**0.006**
Split	0.333	LDA	0.046	0.076	0.020	0.005
		DDA	0.066	0.085	−0.007	0.008
		NN	0.007	0.029	−0.003	0.001
		CART	0.265	0.116	0.166	0.047
	0.5	LDA	0.073	0.078	0.047	0.007
		DDA	0.093	0.099	0.020	0.013
		NN	0.010	0.028	**0**	0.001
		CART	0.308	0.114	0.209	0.071
.632+	≈.368	LDA	0.037	**0.036**	0.011	**0.001**
50 repetitions		DDA	0.085	**0.036**	0.012	0.003
		NN	0.008	**0.016**	−0.002	**0**
		CART	0.160	**0.034**	0.061	0.010

Leave-one-out vs. 10-fold cross-validation (cont'd)

According to [Hastie et al., Springer, 2009], the leave-one-out estimate has larger conditional absolute error

$$\mathbf{E}_{N,S}[|(R(f^S) - \hat{R})|]$$

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 *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 *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:

2. Perform internal cross-validation on S_{train} for each $L \in \mathcal{L}$

   Select L_i that minimizes cross-validation error
3. With L_i, learn classifier on sample S_{train}, i.e. $f = L_i(S_{\text{train}})$
4. Risk of f^S estimated as $e(S_{\text{test}}, f)$
5. With L_i, learn classifier on sample S, i.e. $f^S = L_i(S)$

OPPA European Social Fund
Prague & EU: We invest in your future.