Ludwig-Maximilians-Universität München

Institut für Statistik

Efficient Computation of Unconditional Error Rate Estimators for Learning Algorithms and an Application to a Biomedical Data Set

Master Thesis

Norbert Krautenbacher

31.03.2014

Advisor:

Dr. rer. nat. Mathias Fuchs

Referee:

Prof. Dr. rer. nat. HDR Anne-Laure Boulesteix

Abstract

We derive an unbiased variance estimator for re-sampling procedures using the fact that those procedures are incomplete U-statistics. Our approach is based on careful examination of the combinatorics governing the covariances between re-sampling iterations. We establish such an unbiased variance estimator for the special case of K-Fold cross-validation. This estimator exists as soon as new observations are added to the original sample, and we specify how many additional observations are necessary. Thus we make re-sampling procedures comparable. We make no assumptions on the underlying distribution and we take the covariances between re-sampling iterations into account. Beyond that we show an approach to find a re-sampling design with minimal variance for a fixed size of learning sets. We empirically show the existence of designs with smaller variance than repeated cross-validation. We systemically compare with the complete U-statistic, the leave-p-out estimator. Our examination is completed by an application to micro-array data.

Acknowledgments

This master thesis would not have been completed without the help of my advisor, Dr. rer. nat. Mathias Fuchs. I am deeply grateful for his guidance, encouragement and supervision during the past half year. I would also like to thank Prof. Dr. rer. nat. HDR Anne-Laure Boulesteix for giving additional advice and for help in organisational affairs. At last, I am grateful to all of those who supported me in any aspect during the completion of this work.

Statutory declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources or resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Ort, Datum Norbert Krautenbacher München, 31.03.2014

Contents

Chapter 1

Introduction

This work is concerned with benchmarking in supervised learning. In particular, we consider the goal of comparing two candidate algorithms. There are several well-known procedures used in practice. The most famous one may be K-Fold cross-validation. In that case, however, practitioners often have to face the problem of high variances of the procedure. In this work we will deal with that issue by considering the theoretical aspects of such a procedure. We will especially investigate its variance and the estimator of this quantity.

In this chapter, in particular, we will at first give a general theoretical framework by Section 1.1. After that we will state the issues about the use of cross-validation and corresponding testing procedures by Section 1.2.

Section 1.3 will then explicitly describe the re-sampling procedures K-Fold cross-validation and the minimum-variance estimator leave-p-out. Both procedures will be important in the context of this work. In the last two sections of this chapter, 1.4 and 1.5, we will focus on the theoretical principles. In particular, we will treat complete and incomplete U-statistics. In fact, the introduced re-sampling procedures — we will call them "CV-like procedures" — are incomplete U-statistics. We will show this in Chapter 2 (Section 2.1).

Sections 2.2, 2.3 and 2.4 will give the main results of this work: we will apply the theory of (incomplete) U-statistics on CV-like procedures. We will train our sights on the variance of these procedures and will derive this quantity.

Let n be the total sample size and g be the learning set size. Then we will

consider the requirement that $n \geq 2g + 2$. Under this condition we will derive an unbiased estimator for this variance which is empirically estimable. Section 2.4 eventually will examine how, for a fixed number of learning sets, one could find a procedure which has minimum variance. This problem particularly has relevance for practice. Section 2.5 will show aspects about the convergence of an incomplete U-statistic to the complete U-statistic under random subsampling.
Chapter 3 contains two examples of an application on data. The first (Section 3.1) will be an artificial data example. On the basis of this we will empirically show the existence of a CV-like procedure which has smaller variance than repeated cross-validation. At last, Section 3.2 will deal with a real data problem in which we will estimate our variance estimator for 5-Fold cross-validation.

1.1 Theoretical Framework

This chapter formalizes some basic issues in machine learning or benchmarking. Thereby we will be able to investigate the comparison of the error rate estimators of two binary classification algorithms in further chapters. The following definitions and notations are similar in Fuchs et al. (2013).
In Statistical Learning we at first suppose that our data arose from an unknown distribution P (or sometimes referred to as "data generating process"):
Let $\mathcal{P}(\mathcal{Z})$ be a family of probability measures on $\mathcal{Z}$. In our case of supervised learning, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^{r-1}$ corresponds to a predictor space with $r - 1$ ($r \in \mathbb{N}$) being a fixed number of predictors, and $\mathcal{Y} \subset \mathbb{R}$ to the response space. Therefore, $P \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, and P is supposed to be defined on the standard Lebesgue σ-algebra. However, P is not required to be absolutely continuous with respect to Lebesgue measure since in binary classification we receive a discrete marginal distribution in $\mathcal{Y}$.
We suppose a given sample in $\mathcal{Z}^{\times n} = (\mathcal{X} \times \mathcal{Y})^{\times n}$ of size n. Let $\mathbf{z}_i := (\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, then

$$\{\mathbf{z}_1, \ldots, \mathbf{z}_n\} \sim P^{\otimes n},$$

where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.
Let $g \in \mathbb{N}$ be the learning sample size, and therefore $g \leq n$. We will denote a learning sample of size g shortly by $\mathfrak{L}$, i.e. $\mathfrak{L} := \{\mathbf{z}_1, \ldots, \mathbf{z}_g\} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_g, y_g)\}$.

Remark 1. Throughout this work every set can be seen as unordered as long as it is not particularly specified as ordered.

In the following, we will define a general model or learning algorithm for binary classification on which we will focus in this work.

Definition 1. A prediction model f, specifically a *binary classification algorithm* or *classifier* is defined as a measurable map:

$$\begin{aligned} f : (\mathcal{X} \times \mathcal{Y})^{\times g} \times \mathcal{X} &\to \mathcal{Y}, \\ (\mathfrak{L}, \mathbf{x}_{g+1}) &\mapsto y, \end{aligned} \tag{1.1.1}$$

where f is symmetric in the first g arguments. These correspond to the learning arguments $\mathfrak{L} = \{\mathbf{z}_1, \ldots, \mathbf{z}_g\}$ and here $y \in \mathcal{Y} = \{0, 1\}$.
Alternatively, f can be viewed as

$$f : (X \times Y)^g \to Map(X, Y)$$

but we will not take this point of view.

Our goal is to compare several candidate algorithms $f_1, \ldots, f_k$, where k is the number of candidate algorithms. Hence, a measure of performance of an algorithm is required. Therefore, it is useful to look at the discrepancy between the true response y and the value predicted by our model f_i, $i \in \{1, \ldots, k\}$, denoted by $\hat{y}$. In general, this discrepancy can be measured by the loss

$$\begin{aligned} L : \mathcal{Y} \times \mathcal{Y} &\to \mathbb{R}, \\ (y, \hat{y}) &\mapsto l \end{aligned}$$

In case of a binary classifier, l can be chosen such that $l \in \{0, 1\}$ holds and we can specify the loss function as

$$L(y, \hat{y}) = \mathbb{1}_{y \neq \hat{y}} \tag{1.1.2}$$

The (mean) measure of performance which is our main object of interest is defined by the expectation of the loss (1.1.2). Thus it is the expected value of prediction error or, in our case of binary classification, the expected misclassification rate.

Definition 2. The measure of performance is defined by the *prediction error* or (in classification) as *unconditional error rate*:

$$\begin{aligned} e_f :&= \mathbb{E}_{P^{\otimes(g+1)}} L(f(\mathfrak{L}, \mathbf{x}_{g+1}), y_{g+1}) \\ &= \mathbb{E}_{P^{\otimes(g+1)}} \mathbb{1}_{f(\mathfrak{L}, \mathbf{x}_{g+1}) \neq y_{g+1}} \\ &= P^{\otimes(g+1)}(\text{“}f \text{ is mistaken”}) \end{aligned} \tag{1.1.3}$$

We can term (1.1.3) as unconditional because the error rate is not conditional on the learning data $\mathfrak{L}$ any more, since we look at the joint expectation of the error rate. In the following, we will nonetheless shortly use the expression “error rate”, since we only will refer to the unconditional error rate from now on.
In practice, the first problem one is confronted with, is how to estimate this error rate e_f for a candidate algorithm f. On a sample of size of $n \geq g + 1$ there are many obvious unbiased estimators of e_f.

Before formulating an estimator of e_f, we will define some terms, which we will use throughout this work.

Definition 3. We define $\mathcal{S}^{(n,m)}$ as the set of all permutations $\{S(1), \ldots, S(m)\}$ of size m chosen from $\{1, \ldots, n\}$, i.e.

$$\mathcal{S}^{(n,m)} := \{s \in Map(\{1, \ldots, m\}, \{1, \ldots n\}) \mid \text{s is injective}\} \tag{1.1.4}$$

Definition 4. Let $\mathcal{S}_m$ be the set of all possible permutations of the set $\{1, \ldots, m\}$,

i.e. the special case of Definition 3 that

$$\mathcal{S}_m := \mathcal{S}_0^{(m,m)} = \{s \in Map(\{1,\ldots,m\},\{1,\ldots m\}) \mid \text{s is bijective}\} \tag{1.1.5}$$

This set is usually called the *symmetric group* $\mathcal{S}_m$.

Definition 5. Let $\mathcal{S}_0^{(n,m)}$ be the set of all unordered subsets $\{S(1),\ldots,S(m)\}$ of size m chosen from $\{1,\ldots,n\}$ without regard to the order of the sets' elements, i.e.

$$\mathcal{S}^{(n,m)} := \{S \mid S \subset \{1,\ldots,n\},\ |S| = m\} \tag{1.1.6}$$

Lemma 1. $|\mathcal{S}_m| = m!$, $|\mathcal{S}^{(n,m)}| = \binom{n}{m} \cdot m!$ and $|\mathcal{S}_0^{(n,m)}| = \binom{n}{m}$.

Proof. Elementary combinatorics, can be proved by complete induction. □

With the definitions above, we can define a term which will be useful for describing error rate estimators.

Definition 6. Let us call a pair $(S;a)$ an *evaluation tuple*, where S is an unordered subset of $\{1,\ldots,n\}$ of size g, i.e. $S \in \mathcal{S}_0^{(n,g)}$, and $a \in \{1,\ldots,n\} \setminus S$. Then, let $\mathscr{T}^{(n,g)}$ denote the collection of all evaluation tuples. For any non-empty collection of evaluation tuples, denoted by $\mathscr{T}^* \subset \mathscr{T}^{(n,g)}$, we will see $\mathcal{S}^*$ as the corresponding collection of all sets S contained in $\mathscr{T}^*$.

Our motivation for introducing this notion is that such a pair unambiguously identifies a summand for a procedure which forward we will call CV-like procedure. Any evaluation tuple defines a value $y = f(\mathfrak{L}_S;\mathbf{x}_a) \in \mathcal{Y}$, where $\mathfrak{L}_S := \{(x_i,y_i) : i \in S\}$. So $\mathcal{S}^* \subset \mathcal{S}_0^{(n,g)}$ corresponds to the collection of sets which contains the indices of the learning data sets and $\{1,\ldots,n\} \setminus S$ to the indices of the test set applied per learning set S.

Let now $\mathscr{T}^* \subset \mathscr{T}^{(n,g)}$ be a non-empty collection of evaluation tuples. Then we can formulate an estimator of e_f for a binary classifier by writing

$$\begin{aligned}\widehat{e}_f(\mathscr{T}^*) :&= \frac{1}{|\mathscr{T}^*|}\sum_{(S;a)\in\mathscr{T}^*} L\left(f(\mathfrak{L}_S, \mathbf{x}_a), y_a\right) \\ &= \frac{1}{|\mathscr{T}^*|}\sum_{(S;a)\in\mathscr{T}^*} \mathbb{1}_{f(\mathfrak{L}_S,\mathbf{x}_a)\neq y_a)}\end{aligned} \tag{1.1.7}$$

This estimator has the following property.

Lemma 2. $\widehat{e}_f(\mathscr{T}^*)$ is an unbiased estimator of e_f, i.e. $\mathbb{E}_{P^{\otimes(g+1)}}\widehat{e}_f(\mathscr{T}^*) = e_f$.

Proof.

$$\begin{aligned}\mathbb{E}_{P^{\otimes(g+1)}}\widehat{e}_f(\mathscr{T}^*) &= \mathbb{E}_{P^{\otimes(g+1)}}\left[\frac{1}{|\mathscr{T}^*|}\sum_{(S;a)\in\mathscr{T}^*} L\left(f(\mathfrak{L}_S, \mathbf{x}_a), y_a\right)\right] \\ &= \frac{1}{|\mathscr{T}^*|}\cdot|\mathscr{T}^*|\cdot\mathbb{E}_{P^{\otimes(g+1)}}L\left(f(\mathfrak{L}_S, \mathbf{x}_a), y_a\right) \\ &\overset{(1.1.3)}{=} e_f\end{aligned}$$

□

Remark 2. In a certain sense, all unbiased estimators are of this form (Halmos, 1946).

The comparison of several, say k, algorithms or unconditional error rates is in the frame of this work. Therefore, we devise our null hypothesis of interest, which is the equality of the candidate algorithms or its error rates:

$$H_0 : e_{f_1} = e_{f_2} = \cdots = e_{f_k}$$

Therefore we test against the alternative hypothesis

$$H_1 : \exists i, j \in \{1, \ldots, k\} : e_{f_i} \neq e_{f_j}$$

Since in this work the focus is merely on the comparison of two algorithms, we

can simplify our hypotheses as follows:

$$\begin{aligned} & H_0 : e_{f_1} = e_{f_2} \quad \textit{vs.} \quad H_1 : e_{f_1} \neq e_{f_2} \\ \iff \quad & H_0 : \Delta e := e_{f_1} - e_{f_2} = 0 \quad \textit{vs.} \quad H_1 : \Delta e \neq 0 \end{aligned} \tag{1.1.8}$$

Here and during this work, we investigate the *difference of two unconditional error rates* Δe rather than just the unconditional error rate e_f of Equation (1.1.3) itself. Therefore, we explicitly formulate this estimator:

$$\begin{aligned} \Delta e &= e_{f_1} - e_{f_2} \\ &= \mathbb{E}_{P^{\otimes(g+1)}}\big(L(f_1(\mathfrak{L}_S, \mathbf{x}_a), y_a) - L(f_2(\mathfrak{L}_S, \mathbf{x}_a), y_a)\big) \end{aligned} \tag{1.1.9}$$

Thus, we can extend the estimator $\widehat{e}_f$ of Equation (1.1.7) by the *estimator of the difference of two unconditional error rates* $\widehat{\Delta e}$:

$$\begin{aligned} \widehat{\Delta e}(\mathscr{T}^*) &= \frac{1}{|\mathscr{T}^*|} \sum_{(S;a)\in\mathscr{T}^*} \Big[L(f_1(\mathfrak{L}_S, \mathbf{x}_a), y_a) \\ &\quad - L(f_2(\mathfrak{L}_S, \mathbf{x}_a), y_a)\Big] \\ &= \frac{1}{|\mathscr{T}^*|} \sum_{(S;a)\in\mathscr{T}} \Gamma(S;a), \end{aligned} \tag{1.1.10}$$

where we let $\Gamma(S;a) := L(f_1(\mathfrak{L}_S, \mathbf{x}_a), y_a) - L(f_2(\mathfrak{L}_S, \mathbf{x}_a), y_a)$.
One could also use a definition of Γ where every summand has its own loss function, i.e. $L_1(f_1(\mathfrak{L}_S, \mathbf{x}_a), y_a) - L_2(f_2(\mathfrak{L}_S, \mathbf{x}_a), y_a)$. Then, the case of a single classifier is treated simultaneously since we could set $L_2 := 0$.

Equation (1.1.10) should be seen as an equality of two quantities, not as an algorithm for computation of its left-hand side. In Section 1.3.3, we will rewrite the unconditional error rate estimators in a way that will be more useful for practical computation: in Equation (1.3.3) one specific learning set may be used for fitting several times, which is unnecessarily of higher computational cost.
The next section will investigate the general problem of testing the equality of

two error rates. It will describe the functionality as well as disadvantages of a common method for error rate estimation.

1.2 Common problem

Until now, the basic expressions of Machine Learning required for this work, have been defined. On the other hand the main object of interest – the unconditional error rate – and the hypothesis we have to test for has been introduced. In this chapter, we will discuss how our hypothesis can be investigated and which problems are entailed.

Having a look at Hypothesis (1.1.8) clearly shows that we test for the equality of two means or, if its difference equals zero. If we – for the sake of simplicity – assumed that our two groups each followed a normal distribution with unknown variance, and we could additionally assume that the observations are all independent of each other, then the general assumptions for a paired t-test (or one-sample-t-test) would hold. The last aspect can in a t-test setting only be accomplished if the subsets in our set $\mathscr{T}^*$ do not overlap, i.e. are disjoint. As a matter of fact, this assumption would also be necessary for a non-parametric counterpart such as the Mann-Whitney U-test.
Thus, we would (randomly) draw non-overlapping evaluation tuples, each of size $g+1$. We would apply our algorithms to each of those sets in order to derive a prediction rule. Then we could estimate the error rate for each of the prediction rules. There are at most $\lfloor n/(n-g) \rfloor$ independent realizations of estimators of Δe. In case one took test sets of size bigger than $n-g$, one would get even fewer independent realizations. Using only independent realizations would result in an inefficient estimation of the unconditional error rate — unless we had a huge data set at hand — and doesn't allow for an application to small sample sizes. For instance, biomedical data sets often do not exceed a sample size of $n = 400$. If we assume that our algorithm f should use at least $g = 50$ learn arguments, we will get only $\left\lfloor \frac{400}{51} \right\rfloor = 7$ independent realizations of estimators of Δe.

Therefore, in practice, prediction errors or error rates usually are estimated by procedures in which evaluation tuples necessarily overlap. These are commonly called "re-sampling" procedures, which we will describe in the following section.

1.3 CV-like procedures

According to the issue described in the previous section, a more efficient use of data has to be applied in order to get an appropriate tool for error rate estimation.

Definition 7. During this work we will refer to estimators like $\widehat{\Delta e}(\mathscr{T}^*)$ given by (1.1.10) as to *CV-like procedures*, where the structure of set $\mathscr{T}^*$ may vary.

1.3.1 K-Fold Cross-validation (CV)

A common and well-known computer-intensive re-sampling method is K-Fold cross-validation (CV) (Hastie et al. (2001), for instance). In CV we consider learning subsets of size g such that our data is split into $K := \lfloor n/(n-g) \rfloor$ equal-sized parts. In this subsection we will assume that n is divisible by $n-g$ so that $K = n/(n-g) \in \mathbb{N}$. The model is fitted to the g learning arguments, i.e. to $K-1$ parts. The prediction error of the fitted model can be calculated by using the remaining part of size $n-g$ as validation part, i.e. as test sample. Applying the same procedure on every part leads to K estimates and by calculating the average of estimates we get the CV estimate of the unconditional error rate which we will formally express with the aid of Definition 6. Note that g is necessarily at least $n/2$. We will show that the CV-estimate is a CV-like procedure. This justifies the term "CV-like".

Let $\mathcal{S}_{CV} \subset \mathcal{S}_0^{(n,g)}$ be a collection of sets of learning set indices, occurring in K-fold cross-validation. So in CV-like procedures, the set $\{1, \ldots, n\}$ is split in

$K = n/(n-g)$ parts so that we get

$$\begin{aligned}\mathcal{S}_{CV} = \{&\{1,\dots,n\} \setminus \{1,\dots,n-g\},\\ &\{1,\dots,n\} \setminus \{(n-g)+1,\dots,2\cdot(n-g)\},\\ &\{1,\dots,n\} \setminus \{2\cdot(n-g)+1,\dots,3\cdot(n-g)\},\\ &\dots,\\ &\{1,\dots,n\} \setminus \{\underbrace{(\frac{n}{n-g}-1)(n-g)+1}_{=g+1},\dots,\underbrace{\frac{n}{n-g}\cdot(n-g)}_{=n}\}\},\end{aligned} \tag{1.3.1}$$

then

$$\mathscr{T}_{CV} := \{(S;a) \mid S \in \mathcal{S}_{CV},\ a \in \{1,\dots,n\} \setminus S\}, \tag{1.3.2}$$

i.e.

$$\begin{aligned}\mathscr{T}_{CV} = \{&(\{1,\dots,n\} \setminus \{1,\dots,n-g\}; 1),\\ &(\{1,\dots,n\} \setminus \{1,\dots,n-g\}; 2),\\ &\dots,\\ &(\{1,\dots,n\} \setminus \{1,\dots,n-g\}; n-g),\\ &(\{1,\dots,n\} \setminus \{(n-g)+1,\dots,2\cdot(n-g)\}; n-g+1),\\ &(\{1,\dots,n\} \setminus \{(n-g)+1,\dots,2\cdot(n-g)\}; n-g+2),\\ &\dots\\ &(\{1,\dots,n\} \setminus \{(n-g)+1,\dots,2\cdot(n-g)\}; 2\cdot(n-g)),\\ &(\{1,\dots,n\} \setminus \{2\cdot(n-g)+1,\dots,3\cdot(n-g)\}; 2\cdot(n-g)+1),\\ &\dots,\\ &\dots,\\ &(\{1,\dots,n\} \setminus \{(g+1,\dots,n)\}; n)\}\end{aligned}$$

Then the CV estimate of the unconditional error rate is

$$\widehat{\Delta e}_{CV} := \frac{1}{|\mathscr{T}_{CV}|} \sum_{(S;a)\in\mathscr{T}_{CV}} \Gamma(S;a) \overset{(1.1.10)}{=} \widehat{\Delta e}(\mathscr{T}_{CV}), \tag{1.3.3}$$

which simply is substituting the set $\mathscr{T}_{CV}$ into (1.1.10).

Therefore we have shown that ordinary cross-validation is just a special case of a CV-like procedure. There are many different possibilities of CV for fixed g and n.

Lemma 3. Provided that $K \in \mathbb{N}$, the number of ways to partition a set for K-Fold cross-validation for a fixed g and a fixed n is exactly

$$\frac{n!}{K!} \cdot \left(\left(\frac{n}{K} \right)! \right)^{-K} \tag{1.3.4}$$

Proof. The number of possibilities is equal to the number of ways to partition a set $\{1, \ldots, n\}$ into $K = n/(n-g)$ non-empty subsets, where the size of the subsets is fixed and equal to $n - g = n/K$.
Thus there are $\binom{n}{n/K}$ ways for the first partition, $\binom{n-n/K}{n/K}$ for the second, $\binom{n-2\cdot n/K}{n/K}$ for the third, and so forth. Taking into account that the order of the partitions does not matter, we get

$$\begin{aligned}
&\frac{1}{K!} \cdot \binom{n}{\frac{n}{K}} \cdot \binom{n - \frac{n}{K}}{\frac{n}{K}} \cdot \binom{n - 2 \cdot \frac{n}{K}}{\frac{n}{K}} \cdots \binom{n - K \cdot \frac{n}{K} + 1}{\frac{n}{K}} \\
&= \frac{1}{K!} \cdot \binom{n}{\frac{n}{K}} \cdot \binom{n - \frac{n}{K}}{\frac{n}{K}} \cdot \binom{n - 2 \cdot \frac{n}{K}}{\frac{n}{K}} \cdots \\
&\cdots \binom{n - (K-2) \cdot \frac{n}{K}}{\frac{n}{K}} \cdot \binom{\frac{n}{K}}{\frac{n}{K}} \\
&= \frac{1}{K!} \cdot \prod_{i=0}^{K-2} \binom{n - i \cdot \frac{n}{K}}{\frac{n}{K}} \\
&= \frac{1}{K!} \cdot \prod_{i=0}^{K-2} \frac{\left(n - i \cdot \frac{n}{K}\right)!}{\left(\frac{n}{K}\right)! \cdot \left(n - i \cdot \frac{n}{K} - \frac{n}{K}\right)!} \\
&= \frac{1}{K!} \cdot \prod_{i=0}^{K-2} \frac{\left(n - i \cdot \frac{n}{K}\right)!}{\left(\frac{n}{K}\right)! \cdot \left(n - (i+1) \cdot \frac{n}{K}\right)!} \\
&= \frac{1}{K!} \cdot \frac{\prod_{i=0}^{K-2} \left(n - i \cdot \frac{n}{K}\right)!}{\prod_{i=1}^{K-1} \left(\frac{n}{K}\right)! \cdot \left(n - i \cdot \frac{n}{K}\right)!}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{K!} \cdot \frac{\prod_{i=0}^{K-2} \left(n - i \cdot \frac{n}{K}\right)!}{\left(\prod_{i=1}^{K-1} \left(\frac{n}{K}\right)!\right) \cdot \frac{\left(n-(K-1)\cdot\frac{n}{K}\right)!}{n!} \cdot \left(\prod_{i=0}^{K-2} \left(n - i \cdot \frac{n}{K}\right)!\right)} \\
&= \frac{n!}{K! \cdot \left(\left(\frac{n}{K}\right)!\right)^{(K-1)} \cdot \left(\frac{n}{K}\right)!} \\
&= \frac{n!}{K!} \cdot \left(\left(\frac{n}{K}\right)!\right)^{-K}
\end{aligned}
$$

□

For instance, for a sample size of $n = 100$ and $K = 5$ there are $\frac{100!}{5!} \cdot \left(\left(\frac{100}{5}\right)!\right)^{-5} = 9.12 \times 10^{63}$ ways of how to partition the set for 5-Fold cross-validation.

Definition 8. For any $\mathscr{T}^* \subset \mathscr{T}^{(n,g)}$, we will in general call $|\mathscr{T}^*|$ the *size* of the CV-like procedure $\widehat{\Delta e}(\mathscr{T}^*)$.

Lemma 4. Cross-validation, i.e. $\widehat{\Delta e}_{CV}$, is a CV-like procedure of size $|\mathscr{T}_{CV}| = n$.

Proof. From (1.3.1), it is clear that $|\mathcal{S}_{CV}| = n/(n-g)$. For every $S \in \mathcal{S}_{CV}$ as part of an evaluation tuple for CV, there are $|\{1, \ldots, n\} \setminus S| = n - g$ test observations.
Thus $|\mathscr{T}_{CV}| = |\mathcal{S}_{CV}| \cdot |\{1, \ldots, n\} \setminus S| = n/(n-g) \cdot (n-g) = n$.

□

CV obviously leads to overlaps in learning and test sets, so it uses dependent instead of independent realizations. In addition, our null distribution, i.e. the distribution of our test statistic when $H_0 : \Delta e = 0$ is true, is unknown. It is pointed out in Fuchs et al. (2013) that there are neither exact nor asymptotically exact test procedures for testing our hypothesis available.
In general, a common problem of CV-like procedures like CV in practice is that such estimators have a large variance. An essential question that we have to answer in order to get an accurate estimate for the unconditional error rate is: *What can we say about the variance of CV-like error rate estimates?*

There is vast literature to answer this question. In general, if correlations among our dependent realizations are ignored, the true variance of the unconditional error rate tends to be underestimated. This is illustrated in Bengio and Grandvalet (2003).

The question of what we can actually say about the variance of a CV-like procedure, is also treated in Bengio and Grandvalet (2003). According to Theorem 1 of that work, there exists no universal unbiased estimator of the variance of K-Fold cross-validation. Since we cannot find an unbiased estimator of the variance, we have to ask our question of interest in a different way: *Can we pick a CV-like procedure with minimum variance?*

In Chapter 2, we will show that for a fixed g, there actually is a CV-like procedure with minimal variance, the leave-p-out estimator. This will turn out to be the unique *minimum-variance unbiased estimator* (MVUE). Before investigating this estimator from a more theoretical view, we will introduce it in the following section as one type of CV-like procedure.

1.3.2 Leave-p-out cross-validation (LpO)

In this section, we will introduce a further CV-like method, which will be of particular importance in further chapters. Let us investigate a CV-like procedure which takes all possible subsets of our sample of size $g+1$ into account. This corresponds to a CV-like procedure of maximum size $|\mathscr{T}^{(n,g)}|$ in fact. This procedure has been introduced in several works and is sometimes called *leave-p-out cross-validation* (Shao, 1993), where $p = n - g$.

Again, by analogy to the previous chapter, we recall the error rate difference estimator (1.1.10), derived in Section 1.1. Then we can formalize the procedure described above.

Let

$$\mathscr{T}_{LpO} := \mathscr{T}^{(n,g)} \tag{1.3.5}$$

and substitute $\mathscr{T}_{LpO}$ into (1.1.10). So, $\widehat{\Delta e}_{LpO} := \widehat{\Delta e}(\mathscr{T}_{LpO})$.

Definition 9. We will refer to the CV-like procedure $\widehat{\Delta e}_{LpO}$ as to *leave-p-out cross-validation* (LpO).

Now LpO has been introduced as a further CV-like procedure. Since the collection of sets of learning indices for CV is a sub-collection of those for LpO ($\mathcal{S}_{CV} \subset \mathcal{S}_0^{(n,g)}$), it is obvious that the LpO procedure contains more summands (meaning iterations, cf. Definition 11) than the CV procedure. In fact, LpO is not realizable in practice. However some incomplete versions of LpO are applicable, which we will see in Chapter 2.

Lemma 5. LpO, i.e. $\widehat{\Delta e}_{LpO}$, is equal to the CV-like procedure of maximum size

$$|\mathscr{T}_{LpO}| = \binom{n}{g} \cdot (n-g) = \binom{n}{g+1} \cdot (g+1).$$

Proof. Since $\mathscr{T}_{LpO} = \mathscr{T}^{(n,g)}$, the collection of sets of indices for learning is $\mathcal{S}_0^{(n,g)}$. According to Lemma 1, $|\mathcal{S}_0^{(n,g)}| = \binom{n}{g}$. For every $S \in \mathcal{S}_0^{(n,g)}$ in an evaluation tuple for LpO, there are $|\{1,\ldots,n\} \setminus S| = n - g$ test observations.
Thus

$$\begin{aligned}
|\mathscr{T}_{LpO}| &= |\mathcal{S}_0^{(n,g)}| \cdot |\{1,\ldots,n\} \setminus S| \\
&= \binom{n}{g} \cdot (n-g) \\
&= \frac{n!}{g!(n-g)!} \cdot (n-g) \\
&= \frac{n!}{(g+1)!(n-g-1)!} \cdot (g+1) \\
&= \binom{n}{g+1} \cdot (g+1)
\end{aligned}$$

The fact, that LpO is the unique CV-like procedure of maximum size, which we will see later by Conclusion 3 (Chapter 2), completes the proof. □

The next section will deal with some issues which are important for practical computation.

1.3.3 Computational aspects

In this section we will sum up some aspects which are useful in practical computation of the unconditional error rate for CV-like procedures.
Suppose an algorithm is fitted by a learning set with indices S. In practice we will consider to evaluate always by all possible remaining test observations of $\{1, \ldots, n\} \setminus S$.

Definition 10. We call a design $\mathscr{T}^* \subset \mathscr{T}_{LpO}$ a *test-complete-design*, if the following holds:

$$(S; a) \in \mathscr{T}^*,\ b \in \{1, \ldots, n\} \setminus S \ \Rightarrow (S; b) \in \mathscr{T}^*.$$

In a test-complete-design $\mathscr{T}^*$, there are exactly $n - g$ evaluation tuples for every $S \in \mathcal{S}^*$. This fact is clear, since the test set size is always 1 and in a test-complete-design $|\{1, \ldots, n\} \setminus S| = n - g$.

Conclusion 1. For any test-complete-design $\mathscr{T}^* \subset \mathscr{T}_{LpO}$ the size of $\widehat{\Delta e}(\mathscr{T}^*)$ is exactly

$$|\mathscr{T}^*| = |\mathcal{S}^*| \cdot |\{1, \ldots, n\} \setminus S| = |\mathcal{S}^*| \cdot (n - g), \tag{1.3.6}$$

where $S \in \mathcal{S}^*$.

Lemma 6. Let $\mathscr{T}^* \subset \mathscr{T}_{LpO}$. Then we can rewrite the estimator of the unconditional error rate of (1.1.10) for the case of a test-complete-design by

$$\widehat{\Delta e}(\mathscr{T}^*) = |\mathcal{S}^*|^{-1} \sum_{S \in \mathcal{S}^*} (n - g)^{-1} \sum_{a \in \{1, \ldots, n\} \setminus S} \Gamma(S; a) \tag{1.3.7}$$

Proof.

$$\begin{aligned} \widehat{\Delta e}(\mathscr{T}^*) &= |\mathscr{T}^*|^{-1} \sum_{(S;a) \in \mathscr{T}} \Gamma(S; a) \\ &= |\mathscr{T}^*|^{-1} \sum_{S \in \mathcal{S}^*} \sum_{a \in \{1, \ldots, n\} \setminus S} \Gamma(S; a) \end{aligned}$$

$$= |\mathscr{T}^*|^{-1} \cdot (n-g) \sum_{S \in \mathcal{S}^*} (n-g)^{-1} \sum_{a \in \{1,\dots,n\} \setminus S} \Gamma(S; a)$$
$$= |\mathcal{S}^*|^{-1} \sum_{S \in \mathcal{S}^*} (n-g)^{-1} \sum_{a \in \{1,\dots,n\} \setminus S} \Gamma(S; a)$$

□

In Equation (1.3.7) the algorithms f_1 and f_2 have to be fitted only once for each learning set. Thus, this is the form of the unconditional error rate we will use in practice.

Thus, in view of the computational aspect, the number of how often our algorithms has to be fitted is important, rather then the size of the CV-like procedure.

Definition 11. We will refer to $|\mathcal{S}^*|$ as to the *number of iterations* of $\widehat{\Delta e}(\mathscr{T}^*)$.

Let us therefore compare the computational cost of the introduced CV-like procedures.

Conclusion 2. Conclusion 1 implies that the number of iterations of cross-validation is $|\mathcal{S}_{CV}| = n/(n-g)$ and the number of iterations of LpO is $|\mathcal{S}_{LpO}| = \binom{n}{g}$.

So, in order to get an idea of the difference in computational costs of CV and LpO, we will evaluate the factor of how many more iterations for LpO are required compared to CV:

$$\frac{|\mathcal{S}_{LpO}|}{|\mathcal{S}_{CV}|} = \frac{\binom{n}{g}}{\frac{n}{n-g}} = \frac{(n-g) \cdot n!}{n \cdot g! \cdot (n-g)!} = \frac{(n-1)!}{g! \cdot (n-g-1)!} = \binom{n-1}{g}$$

So, our algorithms have to be trained $\binom{n-1}{g}$ times more for a LpO estimation compared to a CV estimation. This fact shows that we cannot compute a complete LpO procedure in practice.

For instance, for a sample size of $n = 100$, we have to train our algorithms 5 times for 5-Fold cross-validation, i.e. for $g = 80$. Computing the LpO takes $\binom{99}{80} = 1.07 \times 10^{20}$ times more iterations than CV.

The goal of the work is to develop a CV-like procedure whose computational cost is not much higher compared to CV, but whose variance is much smaller. Therefore we have to look at the theoretical aspects of LpO at first. The following section is an introduction to the theory of U-statistics which we will use in Chapter 2 to see CV-like procedures from a specific point of view.

1.4 U-Statistics

The theory of U-statistics has been introduced by Hoeffding (1948), although some properties of U-statistics had been established already by Halmos (1946). In this chapter we will introduce U-statistics and its properties. The following description of the theory of U-statistics and its properties are based on Hoeffding (1948) which contributes the basic results of U-statistics, and on Ferguson (2005) which refers to Hoeffding (1948).

1.4.1 Definitions

At first we will define some terms which will be required to establish some properties of a U-statistic.

Let $\mathcal{P}(\Omega)$ be a family of probability measures on $\Omega = \mathbb{R}^r$. Let Θ be a *parameter of a statistical model*, which can be defined by a map as follows

$$\Theta : \mathbb{R} \to \mathcal{P}$$

For the following definitions, however, it is more convenient to think of a parameter as a map

$$\Theta : \mathcal{P} \to \mathbb{R}$$

Now suppose that for a sample size m that there exists an unbiased estimator of $\Theta(P)$ for any $P \in \mathcal{P}(\Omega)$:

Let $\mathbf{z}_1, \ldots, \mathbf{z}_m$ be realizations of $\mathcal{Z}_1, \ldots, \mathcal{Z}_m \overset{i.i.d.}{\sim} P^{\otimes m}$. Then there exists a measurable function $\Phi(\mathbf{z}_1, \ldots, \mathbf{z}_m)$ ($\Phi : \mathcal{Z}^m \to \mathbb{R}$) such that $\forall P \in \mathcal{P}(\Omega)$:

$$\begin{aligned}\Theta(P) &= \mathbb{E}_{P^{\otimes m}} \Phi(\mathcal{Z}_1, \ldots, \mathcal{Z}_m) \\ &= \int \ldots \int \Phi(\mathbf{z}_1, \ldots, \mathbf{z}_m) \mathrm{d}P(\mathbf{z}_1) \ldots \mathrm{d}P(\mathbf{z}_m).\end{aligned} \tag{1.4.1}$$

Definition 12. We call a parameter of the form (1.4.1) a *regular parameter*. The minimal m allowing this property is called the *degree of* $\Theta(P)$.

The degree of $\Theta(P)$, however, is very difficult to determine in practice!
The function Φ in general is not necessarily supposed to be symmetric in its arguments. However, when Φ is an unbiased estimator of $\Theta(P)$, we can symmetrize Φ by applying the average of Φ to all permutations of arguments. Then

$$\Phi_0(\mathbf{z}_1, \ldots, \mathbf{z}_m) := \frac{1}{m!} \sum_{S \in \mathcal{S}_m} \Phi(\mathbf{z}_{S(1)}, \ldots, \mathbf{z}_{S(m)}) \tag{1.4.2}$$

where $\mathcal{S}_m$ is the symmetric group (s. Definition (4)).
Φ_0 is now symmetric in its arguments, still unbiased and

$$\mathbb{E}_{P^{\otimes m}}(\Phi_0(\mathbf{z}_1, \ldots, \mathbf{z}_m)) = \mathbb{E}_{P^{\otimes m}}(\Phi(\mathbf{z}_1, \ldots, \mathbf{z}_m)) \tag{1.4.3}$$

holds.
The two ways of the definition of a U-statistic are given by Hoeffding (1948).

Definition 13 (U-statistic, Hoeffding (1948))**.** Let $\mathbf{z}_1, \ldots, \mathbf{z}_n$ be a sample of n realizations of $\mathcal{Z}_1, \ldots, \mathcal{Z}_n \overset{i.i.d.}{\sim} P^{\otimes n}$ and Φ a function of $m(\leq n)$ vector arguments. Then

$$U_n = U_n(\Phi) := \frac{1}{|\mathcal{S}^{(n,m)}|} \sum_{S \in \mathcal{S}^{(n,m)}} \Phi(\mathcal{Z}_{S(1)}, \ldots, \mathcal{Z}_{S(m)}), \tag{1.4.4}$$

where $\mathcal{S}^{(n,m)}$ is the set of all permutations $\{S(1), \ldots, S(m)\}$ of size m chosen from $\{1, \ldots, n\}$ (s. Definition 3).
Any such statistic is called a *U-statistic*. Sometimes we will refer to this form as

to a U-statistic with *non-symmetrized kernel* Φ.

Note that therefore $\left|\mathcal{S}^{(n,m)}\right| = \binom{n}{m} \cdot m! = \frac{n!}{(n-m)!}$.

Definition 14 (U-statistic for a symmetric kernel, Hoeffding (1948))**.** For a *symmetric kernel* Φ_0, the U-statistic is

$$U_n = U_n(\Phi_0) := \frac{1}{|\mathcal{S}_0^{(n,m)}|} \sum_{S \in \mathcal{S}_0^{(n,m)}} \Phi_0(\mathcal{Z}_{S(1)}, \ldots, \mathcal{Z}_{S(m)}) \tag{1.4.5}$$

where $\mathcal{S}_0^{(n,m)}$ is the set of all unordered subsets $\{S(1), \ldots, S(m)\}$ of size m chosen from $\{1, \ldots, n\}$ without regard to the order of the sets' elements (s. Definition 5).

The versions above clearly lead to the same value:

$$\begin{aligned}
U_n(\Phi_0) &\overset{(1.4.5)}{=} \frac{1}{|\mathcal{S}_0^{(n,m)}|} \sum_{S \in \mathcal{S}_0^{(n,m)}} \Phi_0(\mathcal{Z}_{S(1)}, \ldots, \mathcal{Z}_{S(m)}) \\
&\overset{(1.4.2)}{=} \frac{1}{|\mathcal{S}_0^{(n,m)}|} \sum_{S \in \mathcal{S}_0^{(n,m)}} \frac{1}{m!} \sum_{S \in \mathcal{S}_m} \Phi(\mathcal{Z}_{S(1)}, \ldots, \mathcal{Z}_{S(m)}) \\
&= \frac{1}{|\mathcal{S}^{(n,m)}|} \sum_{S \in \mathcal{S}^{(n,m)}} \Phi(\mathcal{Z}_{S(1)}, \ldots, \mathcal{Z}_{S(m)}) \\
&\overset{(1.4.4)}{=} U_n(\Phi)
\end{aligned}$$

Note that the version in Definition 14 of U_n contains $m!$ times more summands than the version in Definition 13 does, since $\left|\mathcal{S}_0^{(n,m)}\right| = \binom{n}{m} = \frac{n!}{m! \cdot (n-m)!}$.

We will give a short and simple example for a U-statistic:

Example 1. Let Φ be the identity function, i.e. $\Phi(\mathcal{Z}^{\times 1}) = \mathcal{Z}^{\times 1}$. Then the corresponding U-statistic is the sample mean:

$$U_n(\Phi) = \frac{1}{|\mathcal{S}^{(n,1)}|} \sum_{S \in \mathcal{S}^{(n,1)}} \Phi(\mathcal{Z}_{S(1)}^{\times 1}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{Z}_i^{\times 1}$$

1.4.2 Properties

In this section we will introduce essential properties of U-statistics which we will need for testing our null hypothesis of equal error rates between learning algorithms.

Clearly, if $\Theta(P) = \mathbb{E}_{P^{\otimes m}}\Phi(\mathcal{Z}_1, \ldots, \mathcal{Z}_m) < \infty \; \forall P \in \mathcal{P}(\Omega)$, then U_n is an unbiased estimate of $\Theta(P)$.

The following property is particularly essential in the context of this work and has already been proved by Halmos (1946) for the univariate case and is shown in Hoeffding (1948) for the multivariate case:

"[...] [U_n] is the only unbiased estimate [...] [over $\mathcal{P}(\Omega)$] which is symmetric in [...] [$\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$], and [...] [$U_n$] has the least variance among all unbiased estimates [...] [over $\mathcal{P}(\Omega)$]." (Hoeffding (1948))

So — stated by a single expression — U_n **is unique MVUE**.

Since we want to obtain a test procedure, we will have a look at the properties of U_n with respect to its asymptotic distribution.

Theorem 1. According to Hoeffding (1948) the variance of a U-statistic is

$$\mathbb{V}(U_n) = \binom{n}{m}^{-1} \sum_{d=1}^{m} \binom{m}{d}\binom{n-m}{m-d}\zeta_d^2 \tag{1.4.6}$$

where

$$\zeta_d^2 = \mathbb{V}(\underbrace{\mathbb{E}(\mathbf{z}_1, \ldots, \mathbf{z}_d, \mathcal{Z}_{d+1}, \ldots, \mathcal{Z}_m)}_{=:\Phi_0^d}) = \mathbb{V}(\Phi_0^d(\mathcal{Z}_1, \ldots, \mathcal{Z}_d)), \tag{1.4.7}$$

so that $\zeta_m^2 = \mathbb{V}(\Phi_0(\mathcal{Z}_1, \ldots, \mathcal{Z}_m))$ and we define $\zeta_0^2 := 0$.

Hoeffding shows a further property of the quantities ζ_d^2:

Lemma 7 (Hoeffding (1948), Th. 5.1)**.** The quantities ζ_d^2, $1, \ldots, m$ satisfy

$$0 \leq \frac{\zeta_{d_1}}{d_1} \leq \frac{\zeta_{d_2}}{d_2} \tag{1.4.8}$$

if $1 \leq d_1 \leq d_2 \leq m$.

We will refer to quantities ζ_d^2 as to *Hoeffding quantities* ζ_d^2.
Moreover, in Hoeffding (1948) the following property is shown:

Theorem 2. If $\zeta_m^2 < \infty$ then

$$\sqrt{n}(U_n - \Theta) \xrightarrow{\mathcal{D}} \mathcal{N}(0, m^2 \cdot \zeta_1^2) \tag{1.4.9}$$

Assuming that $\zeta_1^2 \neq 0$, in standardized form we get:

$$\sqrt{n}\frac{U_n - \Theta}{\sqrt{m^2 \cdot \zeta_1^2}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \tag{1.4.10}$$

Thus, U_n is approximately normally distributed. So, in theory, a test procedure can be applied, since an asymptotically exact test exists. In practice, a consistent variance estimator of U_n, i.e. of the asymptotic variance $\frac{1}{n} \cdot m^2 \cdot \zeta_1^2$, has to be found.

In Chapter 2, we will show how the theory of U-statistics can be embedded in the context of machine learning.

1.5 Incomplete U-statistics

In the previous section, the theory of U-statistics has been introduced. In both versions of a U-statistic we have seen that the amount of summands is quite large so that computation might get excessive if m and n get large. This computational cost is especially high if the computation includes a machine learning algorithm in the associated kernel. However, methods for reducing the number of summands have been developed by maintaining the properties of a U-statistic approximately. Lee (1990), for instance, describes designs and properties of so-called *incomplete U-statistics*.

In the following we will introduce the basic concept of incomplete U-statistics. The descriptions are based on Lee (1990), Chapter 4.3.

Definition 15. An *incomplete U-statistic* is one of the form

$$U_n^* = \frac{1}{|\mathcal{S}^*|} \sum_{S \in \mathcal{S}^*} \Phi_0(\mathcal{Z}_{S(1)}, \ldots, \mathcal{Z}_{S(m)}) \tag{1.5.1}$$

where $\mathcal{S}^* \subset \mathcal{S}_0^{(n,m)}$.

The set $\mathcal{S}^*$ is called the *design* of the incomplete U-statistic. We aim for choosing $\mathcal{S}^*$ in such a way that the variance $\mathbb{V}(U_n^*)$ is minimized, for a fixed $|\mathcal{S}^*|$.
Clearly, subsetting $\mathcal{S}^* = \mathcal{S}_0^{(n,m)}$ into (1.5.1) corresponds to the special case that $U_n^* = U_n$ by Equation (1.4.5). Consequently, since U_n has minimum variance,

$$\forall \mathcal{S}^* \subset \mathcal{S}_0^{(n,m)} : \mathbb{V}(U_n^*) \geq \mathbb{V}(U_n)$$

However, the goal of using an incomplete U-statistic is to make the estimate U_n^* asymptotically efficient by choosing $\mathcal{S}^*$ such that $|\mathcal{S}^*| \ll |\mathcal{S}_0^{(n,m)}|$.

There are three specific theorems in Lee (1990) which will particularly be essential for accomplishing that goal for finding a CV-like procedure with small variance in Chapter 2:
Theorem 1 of Lee (1990), Chapter 4.3.1 states that the variance of an incomplete U-statistic exceeds that of a complete one by the variance of the difference of both statistics, in particular by a positive number.
Theorem 2 of that chapter shows that the variance of an incomplete U-statistic is a linear combination of very few covariances.
Theorem 3 of that chapter derives the relation between the overlap sizes of two sets and $n(S)$, the number of m-subsets in the design $\mathcal{D}$, which contain the set S. The theorem derives an alternative, much less intuitive in terms of more explicit combinatorial quantities, called B_ν.

Chapter 2

Error Rate Estimation by U-Statistics

In this chapter we will make theoretical considerations of CV-like procedures and especially of their variance. We will derive this variance and thus also have the variance of CV. After that, we will investigate an approach of minimizing this variance. At the end of this chapter, we will treat the convergence in probability of the incomplete to the complete U-statistic under random subsampling.

2.1 CV-like procedures seen as (incomplete) U-statistics

This section will show the relationship of CV-like procedures and U-statistics. We will figure out that every CV-like procedure, in fact, is an incomplete U-statistic or, in the special case of LpO, even is a complete one.

2.1.1 $\widehat{\Delta e}_{LpO}$ seen as a complete U-statistic

Lemma 8. The CV-like procedure LpO, i.e. $\widehat{\Delta e}(\mathscr{T}_{LpO})$ is a U-statistic of degree at most $g+1$.

Proof. The family of probability measures on Ω in Section 1.4 in our context

corresponds to $\mathcal{P}(\mathcal{X} \times \mathcal{Y})$ with $\mathcal{X} = \mathbb{R}^{r-1}$ and $\mathcal{Y} \subset \mathbb{R}$. In Chapter 1.1 we defined $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $\mathbf{z}_i = (\mathbf{x}_1, y_1)$ $(i = 1, \ldots, n)$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. We used the $\mathbf{z}/\mathcal{Z}$-notation already in Chapter 1.4 for the random variables or realizations of random variables distributed like P. This notation has been used, since it is directly applicable.

The application of U-statistics of Hoeffding (1948) only is justified accurately, if a further issue is taken into account, which is pointed out in Fuchs et al. (2013): according to the definitions in Hoeffding (1948) described in 1.4.1, $\Omega = \mathbb{R}^r$. However, as $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^r$ only holds, we identify P with its push-forward image $i_*(P)$ under the inclusion map $i : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^r$. Thus, P can be viewed as being supported on $\mathbb{R}^r$ and thus Hoeffding (1948) can be applied adequately. In the following, the classifier f of Definition 1 also will be viewed as a measurable map on $\mathbb{R}^{rg+r-1}$ since $f : (\mathcal{X} \times \mathcal{Y})^{\times g} \times \mathcal{X} \to \mathbb{R}^{rg+r-1}$ with respect to $i_*(P)$.

In Chapter 1.1 we defined the unconditional error rate (1.1.3). Therefore we can Δe also write as

$$\int \ldots \int \underbrace{\big(L(f_1(\mathfrak{L}_S, \mathbf{x}_a), y_{g+1}) - L(f_2(\mathfrak{L}_S, \mathbf{x}_a), y_{g+1})\big)}_{=\Gamma(S;a)} \mathrm{d}P(\mathbf{z}_1) \ldots \mathrm{d}P(\mathbf{z}_{g+1})$$

where $\Gamma : (\mathcal{X} \times \mathcal{Y})^{g+1} \to \mathbb{R}$. Thus, Δe is a regular parameter of degree $g + 1$. Again, strictly speaking, it is suggested in Fuchs et al. (2013) that Γ may be viewed as being defined on $\mathbb{R}^{r(g+1)}$ instead of $(\mathcal{X} \times \mathcal{Y})^{g+1}$ like f was extended to $\mathbb{R}^{rg+r-1}$.

□

Assumption 1. The degree of $\Theta = \Delta e$ is exactly $g + 1$.

In particular, this assumption contains the non-degeneracy of Θ.

Now we can draw the following conclusion.

Conclusion 3. $\widehat{\Delta e}(\mathscr{T}_{LpO})$ is the unique minimum-variance unbiased estimator of Δe.

By now, it is shown how the LpO can be seen as a U-statistic. In order to be able to get an asymptotically exact test, we can apply the property of asymptotic

normality of a U-statistic, shown by (1.4.10):
Let $\hat{v}(n)$ be a consistent variance estimator of $\widehat{\Delta e}$. Then

$$(\widehat{\Delta e} - \Delta e)\hat{v}(n)^{-1/2} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \tag{2.1.1}$$

for $n \to \infty$ and as long as g remains fixed and we assume that $\sigma_1^2 \neq 0$.

Attentive readers will have recognized that the size or number of summands of $\widehat{\Delta e}(\mathscr{T}_{LpO})$ neither corresponds to the number of summands of a symmetric kernel of a U-statistic, introduced by Definition 14, nor to the number of summands of a non-symmetric kernel of a U-statistic, introduced by Definition 13. The reason is that the kernel $\Gamma(S; a)$ is neither symmetric nor "completely non-symmetric". We can illustrate this fact from the direction of either the view of a symmetric or a non-symmetric kernel:
$\Gamma(S; a)$ is symmetric in $|S| = g$ arguments. Thus we would set up a U-statistic by $\binom{n}{g}$ summands. Since for every of those summands there are $|\{1, \ldots, n\} \setminus S| = n - g$ possible ways of adding the $g + 1$ argument (or the test observation a), there are $\binom{n}{g} \cdot (n - g) = |\mathscr{T}_{LpO}|$ summands.

We can also compute $\widehat{\Delta e}(\mathscr{T}_{LpO})$ by the non-symmetric kernel so that there are $\binom{n}{g+1} \cdot (g + 1)!$ summands for setting up a U-statistic. However, we do not necessarily have to take the $g!$ permutations in the g symmetric arguments into account. Thus we only need $\binom{n}{g+1} \cdot (g+1)!/g! = \binom{n}{g+1} \cdot (g+1) = |\mathscr{T}_{LpO}|$ summands.

In analogy to (1.4.2) we can symmetrize our kernel Γ:

Definition 16. Let Γ_0 be a map

$$\begin{aligned} \Gamma_0 : \mathcal{S}_0^{(n,g+1)} &\to \mathbb{R} \\ \{1, \ldots, g+1\} &\mapsto \Gamma_0(\{1, \ldots, g+1\}) \end{aligned} \tag{2.1.2}$$

and $\Gamma_0 := \frac{1}{g+1} \sum_{i=1}^{g+1} \Gamma(\{1, \ldots, g+1\} \setminus \{i\};\ i)$, where $i \in \{1, \ldots, g+1\}$.

Note that Γ_0 is the *leave-one-out* variant of the kernel Γ.

Lemma 9. Γ_0 is symmetric in its $g+1$ arguments.

Proof. Follows from (1.4.2) and the following fact: Γ is already symmetric in g arguments and thus it suffices to consider only cyclic permutations.

□

2.1.2 CV and CV-like procedures seen as an incomplete U-statistic

In Chapter 1, we have already seen that the computation of a LpO estimator in practice is not realizable, as the number of iterations needed is too large. Therefore, let us put CV-like procedures with less iterations into the context of U-statistics.

Lemma 10. Every CV-like procedure $\widehat{\Delta e}(\mathscr{T}^*)$, $\mathscr{T}^* \subset \mathscr{T}_{LpO}$ is an incomplete U-statistic.

Proof. Follows from Definition 15 and Lemma 8.

□

Knowing this fact, we can take advantage of the already developed theory of (incomplete) U-statistics in the literature.

Conclusion 4. CV is an incomplete U-statistic.

So, the most common CV-like procedure corresponds to an incomplete U-statistic. Thus, all conclusions we will make in the following sections can be applied to CV as well. Therefore, we will still be able to compare CV-like procedures among themselves.

The next section will investigate the variance of a CV-like procedure by using some theory about incomplete U-statistics. Note that the theory about the variance of an incomplete U-statistic, as well as minimum variance designs for a fixed design size have already been developed by Lee (1990), Chapter 4. However,

the problem of U-statistics with symmetric kernels are treated there only and thus it is not directly applicable on CV-like procedures.

2.2 Variance of a CV-like procedure

In this section, we derive the variance of CV-like procedures and investigate its properties including the variance of CV and LpO estimators. The found results will be required for minimizing the variance and thus finding a CV-like procedure whose variance is minimal related to a fixed size of the design.
We especially will take advantage of Theorems 1 and 2 of Chapter 4.3.1 of Lee (1990) and apply these on CV-like procedures. We will see that leaving out a large proportion of summands of $\widehat{\Delta e}_{LpO}$ still will not expand the variance excessively. The reason for that is the dependency between the terms.

2.2.1 Properties and preliminary work

The following theorem generalizes Theorem 1 of Lee (1990), Chapter 4.3.1 to U-statistics with a non-symmetric kernel and thus to CV-like procedures.

Theorem 3. Let $\widehat{\Delta e}(\mathscr{T}^*)$ be a CV-like procedure based on a fixed design $\mathscr{T}^* \subset \mathscr{T}_{LpO}$ and $\widehat{\Delta e}_{LpO} = \widehat{\Delta e}(\mathscr{T}_{LpO})$ be the leave-p-out estimator.

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) - \mathbb{V}(\widehat{\Delta e}_{LpO}) = \mathbb{V}\left(\widehat{\Delta e}(\mathscr{T}^*) - \widehat{\Delta e}_{LpO}\right) \geq 0 \tag{2.2.1}$$

Proof. Since

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*) - \widehat{\Delta e}_{LpO}) = \mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) - 2Cov(\widehat{\Delta e}(\mathscr{T}^*), \widehat{\Delta e}_{LpO}) + \mathbb{V}(\widehat{\Delta e}_{LpO}),$$

(2.2.1) holds if and only if $Cov(\widehat{\Delta e}(\mathscr{T}^*), \widehat{\Delta e}_{LpO}) = \mathbb{V}(\widehat{\Delta e}_{LpO})$.
Thus it suffices to show this equation.
We will take advantage of the fact that $Cov(\Gamma(S; a), \widehat{\Delta e}_{LpO})$ is the same for every

$(S; a) \in \mathscr{T}_{LpO}$:

$$
\begin{aligned}
\mathbb{V}\widehat{\Delta e}_{LpO} &= Cov(\widehat{\Delta e}_{LpO}, \widehat{\Delta e}_{LpO}) \\
&= Cov(|\mathscr{T}_{LpO}|^{-1} \sum_{(S;a)\in\mathscr{T}_{LpO}} \Gamma(S; a), \widehat{\Delta e}_{LpO}) \\
&= |\mathscr{T}_{LpO}|^{-1} \sum_{(S;a)\in\mathscr{T}_{LpO}} Cov(\Gamma(S; a), \widehat{\Delta e}_{LpO}) \\
&= |\mathscr{T}_{LpO}|^{-1} \cdot |\mathscr{T}_{LpO}| \cdot Cov(\Gamma(S; a), \widehat{\Delta e}_{LpO}) \\
&= |\mathscr{T}^*|^{-1} \cdot |\mathscr{T}^*| \cdot Cov(\Gamma(S; a), \widehat{\Delta e}_{LpO}) \\
&= |\mathscr{T}^*|^{-1} \sum_{(S;a)\in\mathscr{T}^*} Cov(\Gamma(S; a), \widehat{\Delta e}_{LpO}) \\
&= Cov(\widehat{\Delta e}(\mathscr{T}^*), \widehat{\Delta e}_{LpO})
\end{aligned}
$$

□

The proof differs from the one of Lee (1990) not only because we generalize his theorem, but also because he does a mistake in his proof. He assumes that the covariances of an incomplete U-statistic and a kernel are all equal, for each set of the design. This property, however, is not valid in general. We prove this fact in Section A.1 of the appendix.

From the equation we can conclude that the LpO procedure or the complete U-statistic in general always is a more efficient estimation than any other CV-like procedure or an incomplete U-statistic. We also recognize that the variance of a CV-like procedure (or an incomplete U-statistic) can be expressed by the sum of two terms. The first term is the variance of the LpO procedure $\widehat{\Delta e}_{LpO}$ (or complete U-statistic U_n), which we cannot avoid. The second one, we can consider as a penalty component: it inflates the variance of $\widehat{\Delta e}(\mathscr{T}^*)$ (or $\mathbb{V}(U_n^*)$) and — in addition — is again a variance.

In order to generalize Theorem 2 of Lee (1990), Chapter 4.3.1 to CV-like procedures we have to do some preliminary work at first, by stating some definitions and lemmas.

Let us recall formula (1.4.6) of Subsection 1.4.2, which is the variance of a U-statistic in general:

$$\mathbb{V}(U_n) = \binom{n}{m}^{-1} \sum_{d=1}^{m} \binom{m}{d} \binom{n-m}{m-d} \cdot \zeta_d^2$$

In our case of evaluation tuples let $\zeta_d^2 = \mathbb{V}(\Phi_0^d(\mathcal{Z}_1, \ldots, \mathcal{Z}_d))$ be called σ_d^2, which corresponds to the special case of ζ_d^2, where $\Phi_0 = \Gamma_0$ (and Γ_0^d is defined analogous to Φ_0^d). According to Theorem 2 of Lee (1990), Chapter 1, we can also write

$$\sigma_d^2 = Cov\Big(\Gamma_0(\{1, \ldots, g+1\}),\ \Gamma_0(\{1, \ldots, d, g+2, \ldots, 2g+2-d\}\Big). \quad (2.2.2)$$

It is shown in Fuchs et al. (2013) that σ_d^2 is a regular parameter of degree at most $2g+2$. Thus σ_d^2 is estimable by a U-statistic.

Conclusion 5. The variance of the LpO estimator is a regular parameter of degree at most $2g+2$ and is given by

$$\mathbb{V}(\widehat{\Delta e}_{LpO}) = \binom{n}{g+1}^{-1} \sum_{d=1}^{g+1} \binom{g+1}{d} \binom{n-g-1}{g+1-d} \cdot \sigma_d^2. \quad (2.2.3)$$

We will have a closer look at the variance or covariance σ_d^2, since the variance of CV-like procedures in general depends on this parameter (s. Theorem 6).

Assumption 2. Throughout this work, we will assume that $\sigma_1^2 \neq 0$.

In order to be able to express the variance of a CV-like procedure in a clear and structured way, we will make the following definition.

Definition 17. Consider the pairs of evaluation tuples in $\mathscr{T}^* \subset \mathscr{T}_{LpO}$ that have d elements in common and let us denote such a pair by $(S; a), (S'; a') \in \mathscr{T}^*$.
For the overlapping size d let
$f_d^{(1)}$ be the number of those pairs, where $a \notin S'$ and $a' \notin S$ and $a \neq a'$,
$f_d^{(2)}$ be the number of pairs, where either $a \in S'$ and $a' \notin S$ or $a \notin S'$ and $a' \in S$,
$f_d^{(3)}$ be the number of pairs, where $a \in S'$ and $a' \in S$ and

$f_d^{(4)}$ be the number of pairs, where $a = a'$ (and thus $a \notin S'$ and $a' \notin S$).
For each of the four cases, let us define $\tau_d^{(i)} := Cov(\Gamma(S;a), \Gamma(S';a'))$, $i = 1, \ldots, 4$ for the particular $(S;a)$ and $(S';a')$ of each case.
Let $(S;a)$ contain the entries $\{1, \ldots, g+1\}$ and let $(S';a')$ contain the entries $\{g+2, \ldots, 2g+2-d\}$. Then, for an overlapping size d, we get

$$\tau_d^{(1)} = Cov(\Gamma(\{1, \ldots, g\}; g+1), \Gamma(\{1, \ldots, d, g+2, \ldots, 2g+1-d\}; 2g+2-d)),$$
$$\tau_d^{(2)} = Cov(\Gamma(\{2, \ldots, g+1\}; 1), \Gamma(\{1, \ldots, d, g+2, \ldots, 2g+1-d\}; 2g+2-d)),$$
$$\tau_d^{(3)} = Cov(\Gamma(\{2, \ldots, g+1\}; 1), \Gamma(\{1, 3, \ldots, d, g+2, \ldots, 2g+2-d\}; 2)),$$
$$\tau_d^{(4)} = Cov(\Gamma(\{2, \ldots, g+1\}; 1), \Gamma(\{2, \ldots, d, g+2, \ldots, 2g+2-d\}; 1)).$$

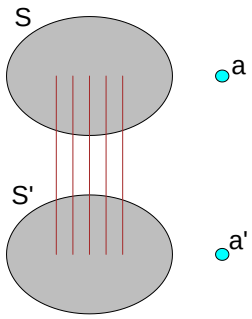
Note that such a pair can consist of two identical evaluation tuples. These definitions for $\tau_d^{(i)}$ $i = 1, \ldots, 4$ are intuitive but problematic for implementation in case of $d = 1, 2$ and $d = g+1$. We therefore reformulate those definitions in the appendix A.2 and will use those definitions also in the implementation and application on the data set.

Lemma 11. For any $(S,a), (S',a') \subset \mathscr{T}_{LpO}$ and overlapping size d there are four possible values for its covariances:

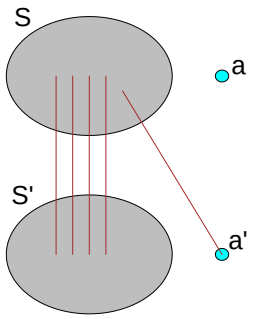
$$Cov(\Gamma(S;a), \Gamma(S';a')) = \begin{cases} \tau_d^{(1)} & \text{if } a \notin S' \text{ and } a' \notin S \text{ and } a \neq a' \\ \tau_d^{(2)} & \text{if either } a \in S' \text{ and } a' \notin S \text{ or } a \notin S' \text{ and } a' \in S \\ \tau_d^{(3)} & \text{if } a \in S' \text{ and } a' \in S \\ \tau_d^{(4)} & \text{if } a = a'. \end{cases}$$

Proof. s. Figure 2.1. □

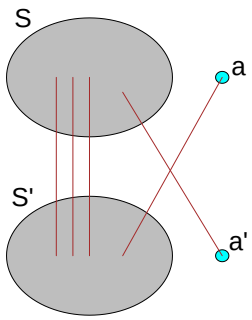
The lemma shows the difficulty of viewing the covariances of our special kernel Γ which is neither symmetric nor "completely non-symmetric". It shows the special problematic that for a fixed overlap size d, the covariance is not just always the same, as it would be for a symmetric kernel. However, we detected a fine structure, i.e. the covariances for a fixed d may not all be the same, but take



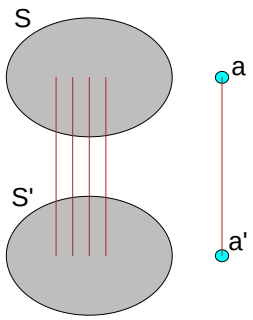
(a) Case 1



(b) Case 2



(c) Case 3



(d) Case 4

Figure 2.1: $Cov(\Gamma(S;a),\Gamma(S';a'))$ only depends on which of the four cases describes the overlap pattern. Here: example for $d = 5$

one of four values. Thus note there is an enormous number of possible pairs of covariances which we can subsume to only $4(g+1)$ covariances. Actually it will turn out (s. Lemma 12) that there are exactly $4g+1$ non-zero quantities $\tau_d^{(i)}$.

Lemma 12. In the special cases of $d = 0$ as well as $d = g + k,\ k \geq 2$ none of the four overlap cases occurs and all values for $\tau_d^{(i)},\ i = 1, \ldots, 4$ are 0. In case of $d = 1$, the third case will not occur. In case of $d = g + 1$ neither overlap case 1 nor case 2 will occur.

Therefore let us make the following definitions.

Definition 18. Let $\tau_0^{(i)} := 0$ for $i = 1, \ldots, 4$, $\tau_1^{(3)} := 0$, $\tau_{g+1}^{(1)} := 0$, $\tau_{g+1}^{(2)} := 0$ and

$\tau_{g+k}^{(i)} := 0$ for $i = 1, \ldots, 4$ for all $k \geq 2$.

The following theorem will show how the variance σ_d^2 exactly decomposes into the four covariances $\tau_d^{(i)},\ i = 1, \ldots, 4$.

Theorem 4. For any $d \in \{0, \ldots, g+1\}$

$$\begin{aligned}\sigma_d^2 = (g+1)^{-2}\Big((g+1-d)^2 \cdot \tau_d^{(1)} \ + \ (g+1-d) \cdot d \cdot 2 \cdot \tau_d^{(2)} \\ + \ (d^2-d) \cdot \tau_d^{(3)} \ + \ d \cdot \tau_d^{(4)}\Big)\end{aligned} \tag{2.2.4}$$

Proof. Let us define $\Gamma_1^d(k) := \Gamma(\{1, \ldots, g+1\} \backslash \{k\}; k)$ and $\Gamma_2^d(l) := \Gamma(\{1, \ldots, d, g+2, \ldots, 2g+2-d\} \setminus \{l\}; l)$.
Then

$$\begin{aligned}
\sigma_d^2 &= Cov\Big(\Gamma_0(\{1, \ldots, g+1\}),\ \Gamma_0(\{1, \ldots, d, g+2, \ldots, 2g+2-d\}\Big) \\
&= Cov\Big(\frac{1}{g+1} \sum_{k \in \{1,\ldots,g+1\}} \Gamma_1^d(k), \frac{1}{g+1} \sum_{l \in \{1,\ldots,d,g+2,\ldots,2g+2-d\}} \Gamma_2^d(l)\Big) \\
&= \frac{1}{(g+1)^2} \sum_{k \in \{1,\ldots,g+1\}} \sum_{l \in \{1,\ldots,d,g+2,\ldots,2g+2-d\}} Cov(\Gamma_1^d(k), \Gamma_2^d(l)) \\
&= \frac{1}{(g+1)^2} \cdot \Bigg[\underbrace{\sum_{k \in \{d+1,\ldots,g+1\}} \sum_{l \in \{g+2,\ldots,2g+2-d\}} Cov(\Gamma_1^d(k), \Gamma_2^d(l))}_{(\#\text{ occurrences of case 1}) \cdot \tau_d^{(1)}} \\
&+ \underbrace{\sum_{k \in \{d+1,\ldots,g+1\}} \sum_{l \in \{1,\ldots,d\}} Cov(\Gamma_1^d(k), \Gamma_2^d(l)) + \sum_{k \in \{1,\ldots,d\}} \sum_{l \in \{g+2,\ldots,2g+2-d\}} Cov(\Gamma_1^d(k), \Gamma_2^d(l))}_{(\#\text{ occurrences of case 2}) \cdot \tau_d^{(2)}} \\
&+ \underbrace{\sum_{k \in \{1,\ldots,d\}} \sum_{l \in \{1,\ldots,d\} \backslash \{k\}} Cov(\Gamma_1^d(k), \Gamma_2^d(l))}_{(\#\text{ occurrences of case 3}) \cdot \tau_d^{(3)}} \\
&+ \underbrace{\sum_{k,l \in \{1,\ldots,d\}, i=l} Cov(\Gamma_1^d(k), \Gamma_2^d(l))}_{(\#\text{ occurrences of case 4}) \cdot \tau_d^{(4)}} \Bigg]
\end{aligned}$$

$$\overset{figure\ 2.2}{=} (g+1)^{-2}\Big((g+1-d)^2\cdot\tau_d^{(1)} \ + \ (g+1-d)\cdot d\cdot 2\cdot\tau_d^{(2)}$$
$$+ \ (d^2-d)\cdot\tau_d^{(3)} \ + \ d\cdot\tau_d^{(4)}\Big)$$

□

The following example illustrates that the fine structure of $\tau_d^{(i)}, i = 1,\dots,4$ wouldn't exist, if we plugged in a symmetric kernel.

Example 2. If, for the sake of illustration, we subset a symmetric kernel, i.e. we considered Γ_0 instead of Γ, all $\tau_d^{(i)}$ would be equal for $i = 1,\dots,4$. This would hold, since the 4 overlap cases wouldn't differ. Then $\tau_d^{(i)} = \sigma_d^2$ and

$$\begin{aligned}(2.2.4) &= (g+1)^{-2}\Big((g+1-d)^2\cdot\sigma_d^2 \ + \ (g+1-d)\cdot d\cdot 2\cdot\sigma_d^2\\ &\quad + \ (d^2-d)\cdot\sigma_d^2 \ + \ d\cdot\sigma_d^2\Big)\\ &= (g+1)^{-2}\cdot\big((g+1)^2 - 2gd - 2d + d^2 + 2gd + 2d - 2d^2 + d^2 - d + d\big)\cdot\sigma_d^2\\ &= \sigma_d^2,\end{aligned}$$

as it should.

Yet we have investigated the structure of possible covariances between two kernels. Therefore we will be able to express the variance of a CV-like procedure by these covariances. We have recognized that the high number of covariance terms of this variance is tangible — concretely, that many of those variances are the same.

Before we will develop the specific formula of the covariance in the next section, we need some important properties about the estimation of covariances $\tau_d^{(i)},\ i = 1,\dots,4$.

Assumption 3. Θ^2 is a kernel of degree $2g+2$

Theorem 5. a) $\tau_d^{(1)}$ is a regular parameter, associated to the kernel

$$\Gamma(\{1,\dots,g\};g+1)\cdot\Gamma(\{1,\dots,d,g+2,\dots,2g+1-d\};2g+2-d))$$
$$-\Gamma(\{1,\dots,g\};g+1)\cdot\Gamma(\{g+2,\dots,2g+1\};2g+2)$$

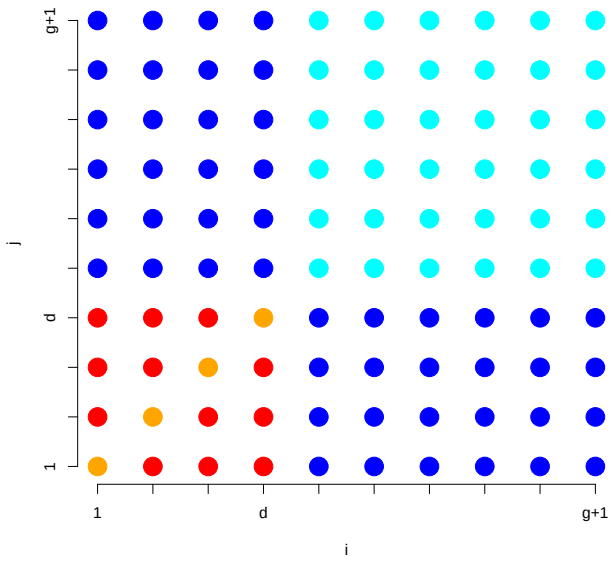


Figure 2.2: Illustration of the decomposition of σ_d^2 into four summands, each being a product of $\tau_d^{(i)}, i = 1, \ldots, 4$ and a specific pre-factor. Example for $d = 4, g = 9$: consider a square, where one side corresponds to the indices $\{1, \ldots, g+1\}$ of the first kernel and the other side to the indices $\{1, \ldots, d, g+2, \ldots, 2g+2-d\}$ of the second kernel.
The $(g+1)^2 = 100$ summands which σ_d^2 decomposes into distribute as follows: $\tau_d^{(1)}$ occurs $(g+1-d)^2 = 36$ times (cyan), $\tau_d^{(2)}$ occurs $(g+1-d) \cdot d \cdot 2 = 48$ times (blue), $\tau_d^{(3)}$ occurs $d^2 - d = 12$ times (red) and $\tau_d^{(4)}$ occurs $d = 4$ times. (orange)

and thus is estimable by a U-statistic.

b) Likewise $\tau_d^{(i)}$ is a regular parameter for $i = 2, 3, 4$

Proof. For any $i \in \{1, \ldots, 4\}$ we can write

$$\begin{aligned}\tau_d^{(i)} &= Cov(\Gamma(S; a), \Gamma(S'; a')) \\ &= \underbrace{\mathbb{E}(\Gamma(S; a) \cdot \Gamma(S'; a'))}_{=: \lambda_d^{(i)}} - \underbrace{\mathbb{E}(\Gamma(S; a)) \cdot \mathbb{E}(\Gamma(S'; a'))}_{= \Theta^2}\end{aligned}$$

where $S,\ S',\ a,\ a'$ are chosen according to one of the 4 cases (as in Lemma 11). Then

$$\begin{aligned}\Theta^2 &= \mathbb{E}(\Gamma(\{1, \ldots, g\}; g+1)) \cdot \mathbb{E}(\Gamma(\{g+2, \ldots, 2g+1\}; 2g+2)) \\ &= \int \ldots \int \Gamma(\{1, \ldots, g\}; g+1) \\ &\quad \cdot \Gamma(\{g+2, \ldots, 2g+1\}; 2g+2) \mathrm{d}P(\mathbf{z}_1) \ldots \mathrm{d}P(\mathbf{z}_{2g+2}).\end{aligned}$$

Thus Θ^2 is a regular parameter of degree at most $2g+2$. That is why we already assumed that the degree of Θ^2 is exactly $2g + 2$.
Further for $i = 1$

$$\begin{aligned}\lambda_d^{(1)} &= \mathbb{E}(\Gamma(\{1, \ldots, g\}; g+1) \cdot \Gamma(\{1, \ldots, d, g+2, \ldots, 2g+1-d\}; 2g+2-d))) \\ &= \int \ldots \int \Gamma(\{1, \ldots, g\}; g+1) \\ &\quad \cdot \Gamma(\{1, \ldots, d, g+2, \ldots, 2g+1-d\}; 2g+2-d)) \mathrm{d}P(\mathbf{z}_1) \ldots \mathrm{d}P(\mathbf{z}_{2g+2-d}).\end{aligned}$$

Thus $\lambda_d^{(1)}$ is a regular parameter of degree at most $2g + 2 - d$.
It can be shown analogically for $i = 2, 3, 4$ in b) that also $\lambda_d^{(2)},\ \lambda_d^{(3)},\ \lambda_d^{(4)}$ are regular parameters of degree at most $2g + 2 - d$.
Since linear combinations of regular parameters again are regular parameters (Hoeffding (1948), p.295), $\tau_d^{(i)} = \lambda_d^{(i)} - \Theta^2$, $i = 1, \ldots, 4$ are regular parameters and thus estimable by U-statistics.

□

Remark 3. A further property of the quantities $\tau_d^{(1)}$, $d = 1, \ldots, g$ can be shown: they can be identified as Hoeffding quantities ζ_d^2 and thus are positive and the inequality analogous to (1.4.8) holds, i.e.

$$0 \leq \frac{\tau_{d_1}^{(1)}}{d_1} \leq \frac{\tau_{d_2}^{(1)}}{d_2} \tag{2.2.5}$$

if $1 \leq d_1 \leq d_2 \leq m$. We will go without a proof, since we will not make use of this property in this work.

This preliminary work finally allows us to state the variance formula of a general CV-like procedure.

2.2.2 Variance formula for a CV-like procedure

Theorem 6. Let $\mathscr{T}^* \subset \mathscr{T}_{LpO}$ and $\widehat{\Delta e}(\mathscr{T}^*)$ be the corresponding CV-like procedure. Then its variance can — by the use of Definition 17 — be formulated by

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) = |\mathscr{T}^*|^{-2} \sum_{d=1}^{g+1} \sum_{i=1}^{4} f_d^{(i)} \cdot \tau_d^{(i)} \tag{2.2.6}$$

Proof. We have shown by Lemma 11 that the covariance of two kernels $\Gamma(S; a)$ and $\Gamma(S'; a')$ decomposes to four summands and is independent of the design $\mathscr{T}^*$. Consequently and by analogy to the proof of Theorem 2 of Chapter 4.3.1 of Lee (1990) and by including Definition 17, we get:

$$\begin{aligned}
\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) &= Cov(\widehat{\Delta e}(\mathscr{T}^*), \widehat{\Delta e}(\mathscr{T}^*)) \\
&= |\mathscr{T}^*|^{-2} \sum_{(S;a) \in \mathscr{T}^*} \sum_{(S';a') \in \mathscr{T}^*} Cov(\Gamma(S; a), \Gamma(S'; a')) \\
&= |\mathscr{T}^*|^{-2} \sum_{d=1}^{g+1} \sum_{i=1}^{4} f_d^{(i)} \cdot \tau_d^{(i)}
\end{aligned}$$

□

Since $\tau_d^{(i)}$, $i = 1, \ldots, 4$ are regular parameters of degree at most $2g + 2$, the linear

combination $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))$ is regular parameter of degree at most $2g+2$. Thus we can draw the following conclusion about its estimator.

Conclusion 6. There is an unbiased estimator for $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))$, if $n \geq 2g+2$.

Let us investigate formula (2.2.6) of Theorem 6. According to Theorem 2 of Lee (1990), Chapter 4.3.1, the four summands $\sum_{i=1}^{4} f_d^{(i)} \cdot \tau_d^{(i)}$ could be subsumed to only one f_d and to σ_d^2, if the kernel was symmetric. In fact, in that case

$$\begin{aligned}
\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) &= Cov(\widehat{\Delta e}(\mathscr{T}^*), \widehat{\Delta e}(\mathscr{T}^*)) \\
&= |\mathscr{T}^*|^{-2} \sum_{(S;a)\in\mathscr{T}^*} \sum_{(S';a')\in\mathscr{T}^*} Cov(\Gamma(S;a), \Gamma(S';a')) \\
&= |\mathscr{T}^*|^{-2} \sum_{d=1}^{g+1} Cov\Big(\Gamma_0(\{1,\ldots,g+1\}), \\
&\qquad \Gamma_0(\{1,\ldots,d,g+2,\ldots,2g+2-d\}\Big) \\
&= |\mathscr{T}^*|^{-2} \sum_{d=1}^{g+1} f_d \cdot \sigma_d^2
\end{aligned}$$

where f_d is the number of pairs of sets in $\mathscr{T}^*$ that have d elements in common.

Therefore, we might be interested in the following question: Is there a way to rewrite $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))$ such that we only have to establish a number of pairs of sets with d elements in common, independently of overlap case $i = 1, \ldots, 4$?
We will investigate such a way for a particular class of collection of designs — the test-complete designs. We introduced these designs by Definition 10 in Section 1.3.3.

Definition 19. Let N^L be the *learn-incidence-matrix* of a design $\mathscr{T}^* \subset \mathscr{T}_{LpO}$, which is defined by $N^L := (n_{ij}^L)_{n\times|\mathcal{S}^*|}$, where

$$n_{ij}^L = \begin{cases} 1 & \text{if index } i \text{ is in set } j \\ 0 & \text{otherwise.} \end{cases} \tag{2.2.7}$$

Let us denote the number of elements a pair of $S \subset \mathcal{S}^*$ and $S' \subset \mathcal{S}^*$ has in common by c. Thus $c = \sum_i n_{ij}^L \cdot n_{ij}^{L\prime}$, which is the j,j' element of $N^{L^T} N^L$.

Definition 20. Let f_c^L be the number of pairs of learning sets in the design that have c elements in common. Thus f_c^L is the number of elements of $N^{L^T} N^L$ equal to c.

For instance, a design of learning sets with index sets $\mathcal{S}^* = \{\{1,2,3,4\},\{3,4,5,6\},\{1,2,5,6\}\}$ with $n = 6$ and $g = 4$ can be represented as

$$N^L = \begin{array}{c} 1 \\ \\ \\ \\ \\ \\ n \end{array} \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Definition 21. In analogy to Definition 19, let N^{Eval} be the *evaluation-incidence-matrix*, which is defined by $N^{Eval} := (n_{ij}^{Eval})_{n \times |\mathscr{T}^*|}$, where

$$n_{ij}^{Eval} = \begin{cases} 1 & \text{if } i \in S_j \\ \beta & \text{if } i = a_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.2.8}$$

The number of elements a pair of Evaluation tuples $(S,a) \in \mathscr{T}^*$ and $(S',a') \in \mathscr{T}^*$ has in common will still be denoted by d. Thus $d = \sum_i \mathbb{1}_{n_{ij}^{Eval} \in \{\beta,1\}} \cdot \mathbb{1}_{n_{ij}^{Eval\prime} \in \{\beta,1\}}$.

The overlap case can be established from the values of $N^{Eval^T} N^{Eval}$. For comparing a pair (S,a) and (S',a'), $\sum_i n_{ij}^{Eval} \cdot n_{ij}^{Eval\prime}$ is the scalar product of two columns of N^{Eval}. The scalar product can have four different values depending on the overlap case:

- In overlap case 1, the scalar product consists of d summands of $1 \cdot 1 = 1$, all other summands equal 0.
- In overlap case 2, the scalar product consists of $d-1$ summands of $1 \cdot 1 = 1$ and one of $1 \cdot \beta = \beta$, all other summands equal 0.
- In overlap case 3, the scalar product consists of $d-2$ summands of $1 \cdot 1 = 1$ and two of $1 \cdot \beta = \beta$, all other summands equal 0.
- In overlap case 4, the scalar product consists of $d-1$ summands of $1 \cdot 1 = 1$ and one of $\beta \cdot \beta = \beta^2$, all other summands equal 0.

Conclusion 7.

$$\sum_i n_{ij}^{Eval} \cdot n_{ij}^{Eval'} = \begin{cases} d & \text{for overlap case 1} \\ \beta + d - 1 & \text{for overlap case 2} \\ 2\beta + d - 2 & \text{for overlap case 3} \\ \beta^2 + d - 1 & \text{for overlap case 4,} \end{cases} \tag{2.2.9}$$

so that the number $f_d^{(i)}$ of pairs of sets in the design $\mathscr{T}^*$ that have d elements in common for overlap case $i \in \{1, \ldots, 4\}$ is established by counting the corresponding values in $N^{Eval^T} N^{Eval}$.

Remark 4. We can express c by d by seeing c as a function

$$c = c(d) = \begin{cases} d & \text{in case 1} \\ d-1 & \text{in case 2} \\ d-2 & \text{in case 3} \\ d-1 & \text{in case 4} \end{cases} \tag{2.2.10}$$

By the use of the incidence matrices and restricting ourselves to the class test-complete-designs, we can rewrite formula (2.2.6) in the desired form, in which we discovered a fine structure.

Corollary 1. Let $\mathscr{T}^* \subset \mathscr{T}_{LpO}$ be a test-complete design.

Then

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) = |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} f_c^L \cdot \xi_c \tag{2.2.11}$$

where

$$\begin{aligned}
\xi_c := & (n-2g+c)\cdot(n-2g+c-1)\cdot\tau^{(1)}_{d=c} \\
+ & \ 2\cdot(g-c)\cdot(n-2g+c)\cdot\tau^{(2)}_{d=c+1} \\
+ & \ (g-c)^2\cdot\tau^{(3)}_{d=c+2} \\
+ & \ (n-2g+c)\cdot\tau^{(4)}_{d=c+1}
\end{aligned}$$

which can be reformulated as follows:

$$\begin{aligned}
\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) = |\mathscr{T}^*|^{-2} \Bigg[& \sum_{d=1}^{g} f_d^L \cdot (n-2g+d)\cdot(n-2g+d-1)\cdot\tau_d^{(1)} \\
& + \sum_{d=1}^{g} f_{d-1}^L \cdot (g-d+1)\cdot(n-2g+d-1)\cdot\tau_d^{(2)} \\
& + \sum_{d=2}^{g+1} f_{d-2}^L \cdot (g-d+2)^2\cdot\tau_d^{(3)} \\
& + \sum_{d=1}^{g+1} f_{d-1}^L \cdot (n-2g+d-1)\cdot\tau_d^{(4)} \Bigg].
\end{aligned} \tag{2.2.12}$$

Proof.

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) = |\mathscr{T}^*|^{-2} \sum_{(S;a)\in\mathscr{T}^*} \sum_{(S';a')\in\mathscr{T}^*} Cov(\Gamma(S;a),\Gamma(S';a'))$$

$$\begin{aligned}
& \overset{(1.3.7)}{=} |\mathscr{T}^*|^{-2} \cdot \Big(|\mathcal{S}^*|^{-2} \sum_{S\in\mathcal{S}^*} \sum_{S'\in\mathcal{S}^*} \\
& \qquad (|\mathscr{T}^*| - |\mathcal{S}^*|)^{-2} \sum_{a\in\{1,\dots,n\}\setminus S} \sum_{a'\in\{1,\dots,n\}\setminus S'} Cov(\Gamma(S;a),\Gamma(S';a'))\Big)
\end{aligned}$$

$$= |\mathscr{T}^*|^{-2} \sum_{S\in\mathcal{S}^*} \sum_{S'\in\mathcal{S}^*} \Big(\sum_{a\in\{1,\dots,n\}\setminus S} \sum_{a'\in\{1,\dots,n\}\setminus S'} Cov(\Gamma(S;a),\Gamma(S';a'))\Big)$$

$$
\begin{aligned}
&\overset{Lemma\ 11}{=} |\mathscr{T}^*|^{-2} \sum_{S\in\mathcal{S}^*} \sum_{S'\in\mathcal{S}^*} \Bigg(\underbrace{\sum_{a\notin S', a'\notin S, a\neq a'} Cov(\Gamma(S;a),\Gamma(S';a'))}_{(\#\text{ occurrences of case 1})\cdot \tau^{(1)}_{d=c}} \\
&+ \underbrace{\sum_{a\in S', a'\notin S} Cov(\Gamma(S;a),\Gamma(S';a')) + \sum_{a\notin S', a'\in S} Cov(\Gamma(S;a),\Gamma(S';a'))}_{(\#\text{ occurrences of case 2})\cdot \tau^{(2)}_{d=c+1}} \\
&+ \underbrace{\sum_{a\in S', a'\in S} Cov(\Gamma(S;a),\Gamma(S';a'))}_{(\#\text{ occurrences of case 3})\cdot \tau^{(3)}_{d=c+2}} \\
&+ \underbrace{\sum_{a=a'} Cov(\Gamma(S;a),\Gamma(S';a'))}_{(\#\text{ occurrences of case 4})\cdot \tau^{(4)}_{d=c+1}} \Bigg)
\end{aligned}
$$

$$
\begin{aligned}
&= |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} f_c^L \cdot \Big(((n-2g+c)^2 - (n-2g+c)) \cdot \tau^{(1)}_{d=c} \\
&+\ 2\cdot(g-c)\cdot(n-2g+c)\cdot \tau^{(2)}_{d=c+1} \\
&+\ (g-c)^2 \cdot \tau^{(3)}_{d=c+2} \\
&+\ (n-2g+c)\cdot \tau^{(4)}_{d=c+1} \Big) \\
&= |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} f_c^L \cdot \xi_c
\end{aligned}
$$

So, the quantities $\tau_d^{(i)}$, $i = 1,\ldots,4$ always occur as $\tau^{(1)}_{d=c}$, $\tau^{(2)}_{d=c+1}$, $\tau^{(3)}_{d=c+2}$ and $\tau^{(4)}_{d=c+1}$. This fact and the number of the occurrences for each quantity can particularly be established by looking at the values of ${N^{Eval}}^T N^{Eval}$. In particular one hast to look at a pair of sets of learn indices S and S', which we will denote by ${N_S^{Eval}}^T N_{S'}^{Eval}$:

At first we will have a look at the pair S, S' in the learn-incidence-matrix:

$$N^L = \begin{array}{c} \\ 1 \\ \\ \\ \\ \\ g \\ \\ \\ \\ \\ n \end{array} \begin{array}{c} \begin{array}{cccc} S & S' & & \end{array} \\ \begin{pmatrix} \dots & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & 1 & 0 & \dots \\ \dots & 1 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & 1 & 1 & \dots \\ \dots & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & 0 & 1 & \dots \\ \dots & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \dots & 0 & 0 & \dots \end{pmatrix} \end{array}$$

Then the evaluation-incidence-matrices for S and S' are:

$$N_S^{Eval} = \begin{array}{c} \\ 1 \\ \\ g \\ \\ \\ \\ \\ n \end{array} \begin{array}{c} \begin{array}{ccccc} & & & & \end{array} \\ \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 & 1 \\ \beta & 0 & \dots & 0 & 0 \\ 0 & \beta & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \beta & 0 \\ 0 & 0 & \dots & 0 & \beta \end{pmatrix} \end{array}, \quad N_{S'}^{Eval} = \begin{pmatrix} \beta & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \beta & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \beta & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 & \beta & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & \beta \end{pmatrix}$$

Thus

$$N_S^{Eval^T} N_{S'}^{Eval} = \begin{array}{c} \begin{array}{ccccccccc} 1 & & g-c & & & & & & n-g \end{array} \\ \begin{pmatrix} 2\beta+c & \dots & 2\beta+c & \beta+c & \beta+c & \dots & \dots & \beta+c \\ \vdots & & \vdots & \vdots & \vdots & & & \vdots \\ 2\beta+c & \dots & 2\beta+c & \beta+c & \beta+c & \dots & \dots & \beta+c \\ \beta+c & \dots & \beta+c & \beta^2+c & c & \dots & \dots & c \\ \vdots & & \vdots & c & \beta^2+c & c & & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \beta+c & \dots & \beta+c & c & \dots & c & \beta^2+c & c \\ \beta+c & \dots & \beta+c & c & \dots & \dots & c & \beta^2+c \end{pmatrix} \end{array}$$

Applying function $c = c(d)$ from Remark 4 to conclusion 7 lets us establish the number of occurrences of each overlap case and thus justifies the second last equal sign and completes the proof.

□

Example 3. Let us for a moment just formally suppose that all $\tau_d^{(i)}$ are equal, say σ^2, which are all the same for any d. We will do this in order to see that the number of summands is correct.
Then Equation (2.2.11) can be simplified to

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) = \sigma^2$$

as it should, since

$$\begin{aligned} \mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) &= |\mathscr{T}^*|^{-2} \sum_{(S;a)\in\mathscr{T}^*} \sum_{(S';a')\in\mathscr{T}^*} Cov(\Gamma(S;a), \Gamma(S';a')) \\ &= Cov(\Gamma(S;a), \Gamma(S';a')) \\ &= \sigma^2, \end{aligned}$$

if $Cov(\Gamma(S;a), \Gamma(S';a'))$ are all the same.

Proof. We will use the fact that $\xi_c = (n-g)^2 \cdot \sigma^2$ if all $\tau_d^{(i)} = \sigma^2$, because ${N_S^{Eval}}^T N_{S'}^{Eval}$ is of dimension $(n-g) \times (n-g)$.
By Definition 20, f_c^L is the number of elements of ${N^L}^T N^L$ equal to c. Since each element of ${N^L}^T N^L$ equals a value of $\{1, \ldots, g\}$ and ${N^L}^T N^L$ has dimension $|\mathcal{S}^*| \times |\mathcal{S}^*|$,

$$\sum_{c=0}^{g} f_c^L = |\mathcal{S}^*|^2 \tag{2.2.13}$$

Then we have

$$\begin{aligned}
\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) &= |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} f_c^L \cdot \xi_c \\
&= |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} f_c^L \cdot (n-g)^2 \cdot \sigma^2 \\
&= |\mathscr{T}^*|^{-2} \cdot (n-g)^2 \cdot \sigma^2 \underbrace{\sum_{c=0}^{g} f_c^L}_{\overset{(2.2.13)}{=}|\mathcal{S}^*|^2} \cdot \\
&= (|\mathcal{S}^*| \cdot (n-g))^{-2} \cdot (n-g)^2 \cdot \sigma^2 \cdot |\mathcal{S}^*|^2 \\
&= \sigma^2
\end{aligned}$$

□

Lemma 13. ξ_c is a regular parameter of degree at most and, by assumption, exactly $2g+2$ and thus estimable by a U-statistic.

Proof. Since $\tau_d^{(i)},\ i = 1, \ldots, 4$ are regular parameters, ξ_c is a linear combination of regular parameters and thus a regular parameter.

□

Remark 5. For a test-complete design

$$f_d^{(1)} = f_d^L \cdot (n-2g+d) \cdot (n-2g+d-1) \tag{2.2.14}$$

$$f_d^{(2)} = f_{d-1}^L \cdot 2 \cdot (g-d+1) \cdot (n-2g+d-1) \tag{2.2.15}$$

$$f_d^{(3)} = f_{d-2}^L \cdot (g-d+2)^2 \tag{2.2.16}$$

$$f_d^{(4)} = f_{d-1}^L \cdot (n - 2g + d - 1) \tag{2.2.17}$$

Proof. Clear from Theorem 6 and Corollary 1. □

2.2.3 Estimation of the variance of a CV-like procedure

By Corollary 1 we have found a form of a CV-like procedure's variance, which is estimable, since all components (f_c^L and pre-factors) are computable and $\tau_d^{(1)}$, $\tau_d^{(2)}$, $\tau_d^{(3)}$ and $\tau_d^{(4)}$ are regular parameters and estimable by U-statistics. We can concretely estimate $\tau_d^{(i)}$ by applying Theorem 5 and the corresponding proof:

$$\widehat{\tau_d^{(i)}} = \widehat{\lambda_d^{(i)}} - \widehat{\Theta^2} \tag{2.2.18}$$

where

$$\widehat{\Theta^2} = |\mathscr{T}_{LpO,\Theta^2}|^{-2} \sum_{|\{S\cup\{a\}\}\cap\{S'\cup\{a'\}\}=\varnothing} \Gamma(S;a)\cdot\Gamma(S';a') \tag{2.2.19}$$

where $\mathscr{T}_{LpO,\Theta^2} := \{(S;a),(S',a') : \{S\cup\{a\}\}\cap\{S'\cup\{a'\}\} = \varnothing\}$
and, using Definition 17,

$$\begin{aligned}
\widehat{\lambda_d^{(1)}} &= |\mathscr{T}_{LpO,\lambda_d^{(1)}}|^{-2} \sum_{a\notin S'\wedge a'\notin S\wedge a\neq a'} \Gamma(S;a)\cdot\Gamma(S';a'),\\
\widehat{\lambda_d^{(2)}} &= |\mathscr{T}_{LpO,\lambda_d^{(2)}}|^{-2} \sum_{(a\notin S'\wedge a'\in S)\vee(a\in S'\wedge a'\notin S)} \Gamma(S;a)\cdot\Gamma(S';a'),\\
\widehat{\lambda_d^{(3)}} &= |\mathscr{T}_{LpO,\lambda_d^{(3)}}|^{-2} \sum_{a\in S'\wedge a'\in S} \Gamma(S;a)\cdot\Gamma(S';a'),\\
\widehat{\lambda_d^{(4)}} &= |\mathscr{T}_{LpO,\lambda_d^{(4)}}|^{-2} \sum_{a=a'} \Gamma(S;a)\cdot\Gamma(S';a')
\end{aligned} \tag{2.2.20}$$

where

$$\mathscr{T}_{LpO,\lambda_d^{(1)}} := \{(S;a),(S',a') : a\notin S'\wedge a'\notin S\wedge a\neq a'\}$$

$$\mathscr{T}_{LpO,\lambda_d^{(2)}} := \{(S;a),(S',a') : (a \notin S' \wedge a' \in S) \vee (a \in S' \wedge a' \notin S)\}$$
$$\mathscr{T}_{LpO,\lambda_d^{(3)}} := \{(S;a),(S',a') : a \in S' \wedge a' \in S\}$$
$$\mathscr{T}_{LpO,\lambda_d^{(4)}} := \{(S;a),(S',a') : a = a'\}$$

One can check that Equation (2.2.18) holds.

Lemma 14. Let $\mathscr{T}^* \subset \mathscr{T}_{LpO}$ be a test-complete-design. Then the unique minimum-variance unbiased estimator of $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))$ can by formulated by

$$\widehat{\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))} = |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} f_c^L \cdot \widehat{\xi}_c \tag{2.2.21}$$

where

$$\begin{aligned}\widehat{\xi}_c := &((n-2g+c)^2 - (n-2g+c)) \cdot \widehat{\tau_{d=c}^{(1)}} \\ &+\ 2\cdot(g-c)\cdot(n-2g+c)\cdot \widehat{\tau_{d=c+1}^{(2)}} \\ &+\ (g-c)^2 \cdot \widehat{\tau_{d=c+2}^{(3)}} \\ &+\ (n-2g+c)\cdot \widehat{\tau_{d=c+1}^{(4)}}\end{aligned}$$

and where $\widehat{\tau_{d=c}^{(1)}}$, $\widehat{\tau_{d=c+1}^{(2)}}$, $\widehat{\tau_{d=c+2}^{(3)}}$ and $\widehat{\tau_{d=c+1}^{(4)}}$ are given by (2.2.18)

Proof. Follows from Corollary 1 and Theorem 5. □

2.2.4 Variance of L*p*O

For the established results above let us investigate the special case of L*p*O as an example.

Lemma 15. For L*p*O the number of pairs of learning sets which have c elements in common is

$$f_{c,LpO}^L = \binom{n}{g}\binom{g}{c}\binom{n-g}{g-c}. \tag{2.2.22}$$

Proof. We consider the number of ways of choosing a pair of learning sets of size g that have c elements in common. Then the first member of the pair of learning

sets can be chosen in $\binom{n}{g}$ ways. The c elements of the second learning set which are common with the first one can be chosen in $\binom{g}{c}$ ways. The $g-c$ elements distinct from these can be chosen in $\binom{n-g}{g-c}$ ways. Thus the number of pairs of learning sets having c elements in common is $\binom{n}{g}\binom{g}{c}\binom{n-g}{g-c}$.

□

In the following we will show how to formulate $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{LpO}))$ by re-scaling the coefficients in Hoeffding's variance formula (2.2.3).

Remark 6. The variance of LpO can be written as

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{LpO})) = |\mathscr{T}^*|^{-2} \cdot \sum_{d=1}^{g+1} f_{d,LpO} \cdot \sigma_d^2 \tag{2.2.23}$$

where

$$f_{d,LpO} = (g+1)^2 \binom{n}{g+1} \binom{g+1}{d} \binom{n-g-1}{g+1-d} \tag{2.2.24}$$

Proof.

$$\begin{aligned}
\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{LpO})) &\overset{(2.2.3)}{=} \binom{n}{g+1}^{-1} \sum_{d=1}^{g+1} \binom{g+1}{d} \binom{n-g-1}{g+1-d} \cdot \sigma_d^2 \\
&= \underbrace{\binom{n}{g+1}^{-2} (g+1)^{-2}}_{=|\mathscr{T}^*|^{-2}} \sum_{d=1}^{g+1} \underbrace{\binom{n}{g+1} (g+1)^2 \binom{g+1}{d} \binom{n-g-1}{g+1-d}}_{=:f_{d,LpO}} \cdot \sigma_d^2 \\
&= |\mathscr{T}^*|^{-2} \cdot \sum_{d=1}^{g+1} f_{d,LpO} \cdot \sigma_d^2
\end{aligned}$$

□

Example 4. The variance of the LpO procedure derived by Hoeffding's variance of a U-statistic (2.2.3) is equal to the variance of Corollary 1, (2.2.12) for the special case $\mathscr{T}^* = \mathscr{T}_{LpO}$.

Proof.

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{LpO})) \overset{(2.2.3)}{=} \binom{n}{g+1}^{-1} \sum_{d=1}^{g+1} \binom{g+1}{d}\binom{n-g-1}{g+1-d} \cdot \sigma_d^2$$

$$= \underbrace{\binom{n}{g+1}^{-2}(g+1)^{-2}}_{|\mathscr{T}^*|^{-2}} \binom{n}{g+1}(g+1)^2 \sum_{d=1}^{g+1} \binom{g+1}{d}\binom{n-g-1}{g+1-d} \cdot \sigma_d^2$$

$$= |\mathscr{T}^*|^{-2}\binom{n}{g}(g+1)(n-g) \cdot \sum_{d=1}^{g+1} \frac{(g+1)!}{d!(g+1-d)!} \cdot \frac{(n-g-1)!}{(g+1-d)!(n-2g-2+d)!} \cdot \sigma_d^2$$

$$= |\mathscr{T}^*|^{-2}\binom{n}{g} g!(n-g)!(g+1)^2 \cdot \sum_{d=1}^{g+1} \frac{1}{d!(g+1-d)!^2(n-2g-2+d)!} \cdot \sigma_d^2$$

$$\overset{2.2.4}{=} |\mathscr{T}^*|^{-2}\binom{n}{g} g!(n-g)!(g+1)^2 \cdot \sum_{d=1}^{g+1} \frac{1}{d!(g+1-d)!^2(n-2g-2+d)!} \cdot \frac{1}{(g+1)^2}\Big((g+1-d)^2 \cdot \tau_d^{(1)} \ + \ (g+1-d)\cdot d \cdot 2 \cdot \tau_d^{(2)} \ + \ (d^2-d)\cdot \tau_d^{(3)} \ + \ d \cdot \tau_d^{(4)}\Big)$$

$$\overset{\tau_{g+1}^{(1)}=\tau_{g+1}^{(2)}=\tau_1^{(3)}=0}{=} |\mathscr{T}^*|^{-2}\Bigg[\underbrace{\sum_{d=1}^{g}\binom{n}{g} g!(n-g)! \cdot \frac{1}{d!(g-d)!^2(n-2g-2+d)!} \cdot \tau_d^{(1)}}_{(1)} + \underbrace{\sum_{d=1}^{g}\binom{n}{g} g!(n-g)! \cdot \frac{2}{(d-1)!(g+1-d)!(g-d)(n-2g-2+d)!} \cdot \tau_d^{(2)}}_{(2)} + \underbrace{\sum_{d=2}^{g+1}\binom{n}{g} g!(n-g)! \cdot \frac{1}{(d-2)!(g+1-d)!^2(n-2g-2+d)!} \cdot \tau_d^{(3)}}_{(3)}$$

$$+\underbrace{\sum_{d=1}^{g+1}\binom{n}{g}g!(n-g)!\cdot\frac{1}{(d-1)!(g+1-d)!^2(n-2g-2+d)!}\cdot\tau_d^{(4)}}_{(4)}$$

$$=(2.2.12),$$

since

$$(1)=\sum_{d=1}^{g}\binom{n}{g}\underbrace{\frac{g!}{d!(g-d)!}}_{=\binom{g}{d}}\cdot\underbrace{\frac{(n-g)!}{(g-d)!(n-2g+d)!}}_{=\binom{n-g}{g-d}}\cdot(n-2g+d-1)(n-2g+d)\cdot\tau_d^{(1)}$$

$$=\sum_{d=1}^{g}f_d^L\cdot(n-2g+d-1)(n-2g+d)\cdot\tau_d^{(1)}$$

$$(2)=\sum_{d=1}^{g}\binom{n}{g}\underbrace{\frac{g!}{(d-1)!(g+1-d)!}}_{=\binom{g}{d-1}}\cdot\underbrace{\frac{(n-g)!}{(g-d+1)!(n-2g+d-1)!}}_{=\binom{n-g}{g-d+1}}$$

$$\cdot 2(n-2g+d-1)(g-d+1)\cdot\tau_d^{(2)}$$

$$=\sum_{d=1}^{g+1}f_{d-1}^L\cdot(g-d+1)(n-2g+d-1)\cdot\tau_d^{(2)}$$

$$(3)=\sum_{d=2}^{g+1}\binom{n}{g}\underbrace{\frac{g!}{(d-2)!(g-d+2)!}}_{=\binom{g}{d-2}}\cdot\underbrace{\frac{(n-g)!}{(g-d+2)!(n-2g-2+d)!}}_{=\binom{n-g}{g-d+2}}\cdot(g-d+2)^2\cdot\tau_d^{(3)}$$

$$=\sum_{d=2}^{g+1}f_{d-2}^L\cdot(g-d+2)^2\cdot\tau_d^{(3)}$$

$$(4)=\sum_{d=1}^{g+1}\binom{n}{g}\underbrace{\frac{g!}{(d-1)!(g+1-d)!}}_{=\binom{g}{d-1}}\cdot\underbrace{\frac{(n-g)!}{(g+1-d)!(n-2g+d-1)!}}_{\binom{n-g}{g-d+1}}\cdot(n-2g+d-1)\cdot\tau_d^{(4)}$$

$$=\sum_{d=1}^{g+1}f_{d-1}^L\cdot(n-2g+d-1)\cdot\tau_d^{(4)}$$

□

Let us view a further aspect about these two formulations of $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{LpO}))$ —

the distributions of $f_{d,LpO}$ and $f^{(i)}_{d,LpO},\ i = 1, \ldots, 4$ for the example of $n = 40$ for different choices of g. According to Remark 5

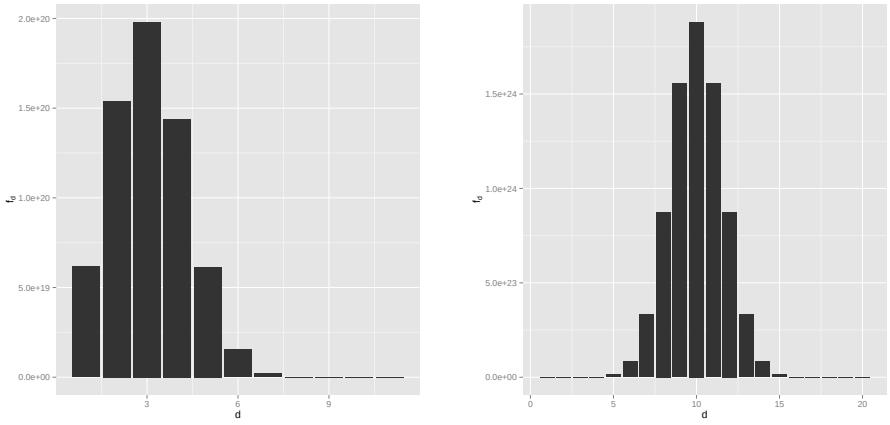
$$
\begin{aligned}
f^{(1)}_{d,LpO} &= f^{L}_{d,LpO} \cdot (n - 2g + d) \cdot (n - 2g + d - 1) \\
f^{(2)}_{d,LpO} &= f^{L}_{d-1,LpO} \cdot 2 \cdot (g - d + 1) \cdot (n - 2g + d - 1) \\
f^{(3)}_{d,LpO} &= f^{L}_{d-2,LpO} \cdot (g - d + 2)^2 \\
f^{(4)}_{d,LpO} &= f^{L}_{d-1,LpO} \cdot (n - 2g + d - 1)
\end{aligned}
$$

Figure 2.3 shows $f_{d,LpO}$ for three different choices of g. $f_{d,LpO}$ approximately consists of the hypergeometric expression $f^{L}_{d,LpO} = \binom{n}{g}\binom{g}{d}\binom{n-g}{g-d}$, since the other factors are comparatively small. Since the hypergeometric distribution approximately follows a normal distribution for big n, all three examples graphically assume the shape of a normal distribution. We can establish the means by the expected value of the hypergeometric distribution, which in our case is $(g+1)^2/n$. Thus in the case of $g + 1 = n/2 = 20$, the mean takes the value $d = 10$. For $n/2 > g = 10$ the mean decreases to $d = 3.025$ for $n/2 < g = 30$ the mean increases to $d = 24.025$.

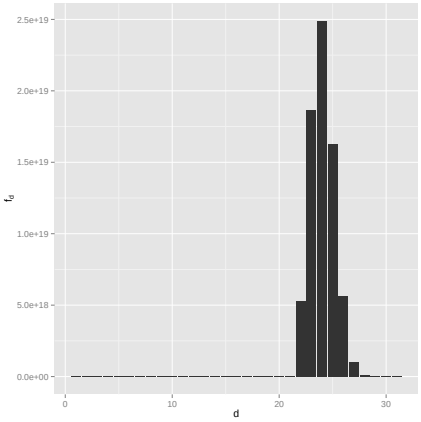
Figure 2.4 shows $f^{(i)}_{d,LpO},\ i = 1, \ldots, 4$ for the three choices of g. Since again the hypergeometric expression $f^{L}_{d,LpO} = \binom{n}{g}\binom{g}{d}\binom{n-g}{g-d}$ exceeds the other factors by far, all $f^{(i)}_{d,LpO}$ are approximately normally distributed. Varying i for a fixed g barely changes the shape but shifts the normal distribution.

2.3 Variance of cross-validation

Since there is a variance formula for a CV-like procedure, there is consequently a variance formula for CV itself. We at first will view several ways of deriving the variance for the special case of 2-Fold cross-validation, followed by deriving a general formula for the variance of K-fold cross-validation. We also will establish a special set-up, in order to make this variance estimable.



(a) n=40, g=10 **(b)** n=40, g=19



(c) n=40, g=30

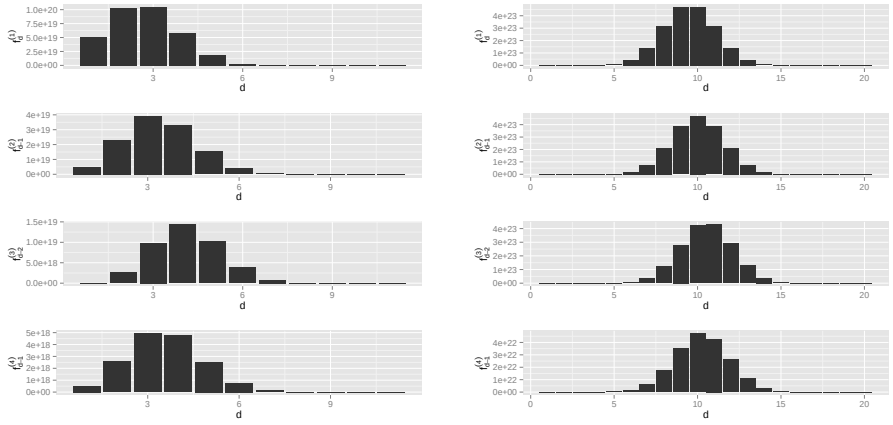
Figure 2.3: Number of occurrences ($f_{d,LpO}$) of σ_d^2 against the overlap size d for various g and $n = 40$

2.3.1 Variance of 2-Fold cross-validation

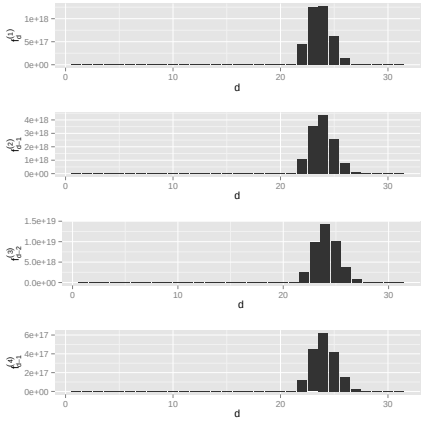
Let us rehearse several ways of deriving the variance for the simple case of 2-Fold cross-validation, so that $|\mathscr{T}_{CV}| = n$ (Lemma 4) and in special case of $K = 2$, $g = n/2$.

Variance of 2-Fold CV by the general variance formula

At first, let us derive the variance in the common way, i.e. by applying the general formula for the variance of sums: In the following let $\mathscr{T}_{2CV}$ be the set of evaluation tuples occurring in 2-Fold cross-validation.



(a) n=40, g=10 **(b)** n=40, g=19



(c) n=40, g=30

Figure 2.4: Number of occurrences ($f^{(1)}_{d,LpO}$, $f^{(2)}_{d-1,LpO}$, $f^{(3)}_{d-2,LpO}$, $f^{(4)}_{d-1,LpO}$) of $\tau^{(1)}_d$, $\tau^{(2)}_d$, $\tau^{(3)}_d$, $\tau^{(4)}_d$ against the overlap size d for various g and $n = 40$

Then

$$
\begin{aligned}
\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{2CV})) &= \mathbb{V}\big(|\mathscr{T}_{2CV}|^{-1} \sum_{(S;a)\in\mathscr{T}_{2CV}} \Gamma(S;a)\big) \\
&\overset{(1.3.7)}{=} \mathbb{V}\big(|\mathcal{S}_{2CV}|^{-1} \sum_{S\in\mathcal{S}_{2CV}} \underbrace{(n-g)}_{=n-n/2}^{-1} \sum_{a\in\{1,\dots,n\}\setminus S} \Gamma(S;a)\big) \\
&= \mathbb{V}\Big(\frac{1}{2}\big(\frac{1}{n/2} \sum_{a\in\{1,\dots,n\}\setminus S_1} \Gamma(S_1;a) + \frac{1}{n/2} \sum_{b\in\{1,\dots,n\}\setminus S_2} \Gamma(S_2;b)\big)\Big) \\
&= \frac{1}{n^2}\Big(Cov\big(\sum_{a\in\{1,\dots,n\}\setminus S_1} \Gamma(S_1;a), \sum_{b\in\{1,\dots,n\}\setminus S_1} \Gamma(S_1;b)\big) \\
&\quad + Cov\big(\sum_{a\in\{1,\dots,n\}\setminus S_2} \Gamma(S_2;a), \sum_{b\in\{1,\dots,n\}\setminus S_2} \Gamma(S_2;b)\big)
\end{aligned}
$$

$$
\begin{aligned}
&\quad + 2 \cdot Cov\Big(\sum_{a\in\{1,\dots,n\}\setminus S_1} \Gamma(S_1;a), \sum_{b\in\{1,\dots,n\}\setminus S_2} \Gamma(S_2;b)) \Big) \Big) \\
&= \frac{1}{n^2}\Big(2 \sum_{a\in\{1,\dots,n\}\setminus S_1} \sum_{b\in\{1,\dots,n\}\setminus S_1} \underbrace{Cov\big(\Gamma(S_1;a),\Gamma(S_1;b)\big)}_{\overset{Lemma\ 11}{=} \begin{cases} \tau^{(1)}_{n/2} & \text{if } a \neq b \\ \tau^{(4)}_{n/2+1} & \text{if } a = b \end{cases}} \\
&\quad + 2 \sum_{a\in\{1,\dots,n\}\setminus S_1} \sum_{b\in\{1,\dots,n\}\setminus S_2} \underbrace{Cov\big(\Gamma(S_1;a),\Gamma(S_2;b)\big)}_{\overset{Lemma\ 11}{=} \tau^{(3)}_{n/2+2}} \Big) \\
&= \frac{2}{n^2}\left(\left(\left(\frac{n}{2}\right)^2 - \frac{n}{2} \right) \cdot \tau^{(1)}_{n/2} + \frac{n}{2} \cdot \tau^{(4)}_{n/2+1} + \left(\frac{n}{2}\right)^2 \cdot \tau^{(3)}_2 \right) \\
&= \left(\frac{1}{2} - \frac{1}{n} \right) \cdot \tau^{(1)}_{n/2} + \frac{1}{2} \cdot \tau^{(3)}_2 + \frac{1}{n} \cdot \tau^{(4)}_{n/2+1}
\end{aligned}
$$

Variance of 2-Fold CV by interpretation of $N^{Eval^T} N^{Eval}$

Let us establish this variance by a second way — by directly viewing the matrix product $N^{Eval^T} N^{Eval}$ for 2-Fold cross-validation and Theorem 6.
The evaluation-incidence-matrix for 2-Fold CV is

$$
N_S^{Eval} = \begin{pmatrix}
1 & \dots & 1 & \beta & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \dots & 1 & 0 & \dots & \beta \\
\beta & \dots & 0 & 1 & \dots & 1 \\
\vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & \beta & 1 & \dots & 1
\end{pmatrix} = \begin{pmatrix} J_{n/2} & \beta I_{n/2} \\ \beta I_{n/2} & J_{n/2} \end{pmatrix}
$$

where I_K is the $k \times k$-identity matrix and J_k the $k \times k$-matrix of ones.
Then

$$
N^{Eval^T} N^{Eval} = \begin{pmatrix} \frac{n}{2} J_{n/2} + \beta^2 I_{n/2} & 2\beta J_{n/2} \\ 2\beta J_{n/2} & \frac{n}{2} J_{n/2} + \beta^2 I_{n/2} \end{pmatrix}
$$

$$= \begin{pmatrix} \beta^2+\frac{n}{2} & \frac{n}{2} & \dots & \frac{n}{2} & 2\beta & 2\beta & \dots & 2\beta \\ \frac{n}{2} & \beta^2+\frac{n}{2} & \dots & \frac{n}{2} & 2\beta & 2\beta & \dots & 2\beta \\ \vdots & \vdots & \ddots & \frac{n}{2} & \vdots & \vdots & \vdots & \vdots \\ \frac{n}{2} & \frac{n}{2} & \dots & \beta^2+\frac{n}{2} & 2\beta & 2\beta & \dots & 2\beta \\ 2\beta & 2\beta & \dots & 2\beta & \beta^2+\frac{n}{2} & \frac{n}{2} & \dots & \frac{n}{2} \\ 2\beta & 2\beta & \dots & 2\beta & \frac{n}{2} & \beta^2+\frac{n}{2} & \dots & \frac{n}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \frac{n}{2} \\ 2\beta & 2\beta & \dots & 2\beta & \frac{n}{2} & \frac{n}{2} & \dots & \beta^2+\frac{n}{2} \end{pmatrix}$$

By applying conclusion 7, we just have to count the entries in order to get $f_d^{(i)}$ of every $i \in \{1, \dots, 4\}$. Thus

$$\begin{aligned} \mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{2CV})) &\overset{(2.2.6)}{=} |\mathscr{T}_{2CV}|^{-2} \sum_{d=1}^{g+1} \sum_{i=1}^{4} f_d^{(i)} \cdot \tau_d^{(i)} \\ &= \frac{1}{n^2} \left(2 \left(\left(\frac{n}{2}\right)^2 - \frac{n}{2} \right) \cdot \tau_{n/2}^{(1)} + 2 \cdot \left(\frac{n}{2}\right)^2 \cdot \tau_{n/2+2}^{(3)} + n \cdot \tau_{n/2+1}^{(4)} \right) \\ &= \left(\frac{1}{2} - \frac{1}{n} \right) \cdot \tau_{n/2}^{(1)} + \frac{1}{2} \cdot \tau_2^{(3)} + \frac{1}{n} \cdot \tau_{n/2+1}^{(4)} \end{aligned}$$

Variance of 2-Fold CV by applying Corollary 1

At last let us subset the corresponding quantities into formula (2.2.11) of Corollary 1. In this case, all f_c^L are zero except $f_0^L = f_{n/2}^L = 2$.
Thus

$$\begin{aligned} \mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{2CV})) &\overset{(2.2.11)}{=} |\mathscr{T}_{2CV}|^{-2} \sum_{c=0}^{g} f_c^L \cdot \xi_c \\ &= |\mathscr{T}_{2CV}|^{-2} \sum_{c=0}^{g} f_c^L \cdot \Big(((n-2g+c)^2 - (n-2g+c)) \cdot \tau_{d=c}^{(1)} \\ &\quad + \ 2 \cdot (g-c) \cdot (n-2g+c) \cdot \tau_{d=c+1}^{(2)} \\ &\quad + \ (g-c)^2 \cdot \tau_{d=c+2}^{(3)} \\ &\quad + \ (n-2g+c) \cdot \tau_{d=c+1}^{(4)} \Big) \\ &\overset{2CV}{=} \frac{1}{n^2} \left(f_0^L \cdot \left(\left(\left(n - 2 \cdot \frac{n}{2} \right)^2 - \left(n - 2 \cdot \frac{n}{2} \right) \right) \cdot \tau_{d=0}^{(1)} \right. \right. \end{aligned}$$

$$
\begin{aligned}
&+\ 2\cdot\frac{n}{2}\cdot\left(n-2\cdot\frac{n}{2}\right)\cdot\tau^{(2)}_{d=1}\\
&+\ \left(\frac{n}{2}\right)^2\cdot\tau^{(3)}_{d=2}\\
&+\ \left(n-2\cdot\frac{n}{2}\right)\cdot\tau^{(4)}_{d=1}\Bigg)\\
&+f^L_{n/2}\cdot\Bigg(\left(\left(n-2\cdot\frac{n}{2}+\frac{n}{2}\right)^2-\left(n-2\cdot\frac{n}{2}+\frac{n}{2}\right)\right)\cdot\tau^{(1)}_{d=n/2}\\
&+\ 2\cdot\left(\frac{n}{2}-\frac{n}{2}\right)\cdot\left(n-2\cdot\frac{n}{2}+\frac{n}{2}\right)\cdot\tau^{(2)}_{d=n/2+1}\\
&+\ \left(\frac{n}{2}-\frac{n}{2}\right)^2\cdot\tau^{(3)}_{d=n/2+2}\\
&+\ \left(n-2\cdot\frac{n}{2}+\frac{n}{2}\right)\cdot\tau^{(4)}_{d=n/2+1}\Bigg)\Bigg)\\
&=\frac{1}{n^2}\cdot\left(f^L_0\cdot\left(\frac{2}{n}\right)^2\cdot\tau^{(3)}_2+f^L_{n/2}\cdot\left(\left(\left(\frac{n}{2}\right)^2-\left(\frac{n}{2}\right)\right)\cdot\tau^{(1)}_{n/2}+\frac{n}{2}\cdot\tau^{(4)}_{n/2+1}\right)\right)\\
&=\frac{2}{n^2}\left(\frac{n^2}{4}\cdot\tau^{(3)}_2+\left(\frac{n^2}{4}-\frac{n}{2}\right)\cdot\tau^{(1)}_{n/2}+\frac{n}{2}\cdot\tau^{(4)}_{n/2+1}\right)\\
&=\left(\frac{1}{2}-\frac{1}{n}\right)\cdot\tau^{(1)}_{n/2}+\frac{1}{2}\cdot\tau^{(3)}_2+\frac{1}{n}\cdot\tau^{(4)}_{n/2+1}
\end{aligned}
$$

We have viewed three different ways of deriving the variance of $\widehat{\Delta e}(\mathscr{T}_{2CV})$ and thus have shown that the variance of 2-Fold CV is

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{2CV}))=\left(\frac{1}{2}-\frac{1}{n}\right)\cdot\tau^{(1)}_{n/2}+\frac{1}{2}\cdot\tau^{(3)}_2+\frac{1}{n}\cdot\tau^{(4)}_{n/2+1}\tag{2.3.1}$$

2.3.2 Variance of K-Fold cross-validation

Having illustrated how the variance for 2-Fold CV looks, we will investigate the general formula for K-Fold cross-validation.

Theorem 7. The variance of K-fold cross-validation is

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{CV}))=\left(\frac{1}{K}-\frac{1}{n}\right)\cdot\tau^{(1)}_{n-n/K}+\frac{1}{K}\cdot\tau^{(3)}_{n-2n/K+2}+\frac{1}{n}\cdot\tau^{(4)}_{n-n/K+1}\tag{2.3.2}$$

Proof. In K-Fold CV the learn-incidence matrix is

$$N_{S'}^{Eval} = \begin{array}{c} \\ 1 \\ \\ n-g \\ \\ \\ \\ \\ \\ \\ n \end{array} \begin{array}{c} \begin{array}{cccc} 1 & & & K \end{array} \\ \begin{pmatrix} 0 & 1 & & 1 \\ \vdots & \vdots & & \vdots \\ 0 & 1 & & \vdots \\ 1 & 0 & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & 0 & & \vdots \\ \vdots & 1 & \ddots & 1 \\ \vdots & \vdots & & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & & 0 \end{pmatrix} \end{array}$$

From the matrix follows that the overlap size in CV is

$$c = \begin{cases} g & \text{if } S = S' \\ n - 2(n-g) = 2g - n & \text{otherwise} \end{cases}$$

Note: $c = 0$ only exists in 2-Fold CV and is covered by case $2g - n = 0$.
Further, since the number of rows of N^L is $K = n/(n-g)$, ${N^L}^T N^L$ is a $K \times K$ matrix, where only the diagonal elements equal g and all off-diagonal elements equal $2g - n$. Thus

$$f_c^L = \begin{cases} 0 & \text{if } c \notin \{g, 2g-n\} \\ K & \text{if } c = g \\ K^2 - K & \text{if } c = 2g - n \end{cases}$$

Using the facts that $|\mathscr{T}_{CV}| = n$ (by Lemma 4), and $K = n/(n-g) \iff g =$

$n - n/K$, the variance of K-Fold CV can be derived:

$$
\begin{aligned}
\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{CV})) &\overset{(2.2.11)}{=} |\mathscr{T}_{CV}|^{-2} \sum_{c=0}^{g} f_c^L \cdot \xi_c \\
&= |\mathscr{T}_{CV}|^{-2} \sum_{c=0}^{g} f_c^L \cdot \Big(((n-2g+c)^2 - (n-2g+c)) \cdot \tau^{(1)}_{d=c} \\
&\quad + \; 2 \cdot (g-c) \cdot (n-2g+c) \cdot \tau^{(2)}_{d=c+1} \\
&\quad + \; (g-c)^2 \cdot \tau^{(3)}_{d=c+2} \\
&\quad + \; (n-2g+c) \cdot \tau^{(4)}_{d=c+1} \Big) \\
&\overset{CV}{=} \frac{1}{n^2} \Big(f_g^L \Big(((n-2g+g)^2 - (n-2g+g)) \cdot \tau^{(1)}_{g} \\
&\quad + \; 2 \cdot (g-g) \cdot (n-2g+g) \cdot \tau^{(2)}_{g+1} \\
&\quad + \; (g-g)^2 \cdot \tau^{(3)}_{g+2} \\
&\quad + \; (n-2g+g) \cdot \tau^{(4)}_{g+1} \Big) \\
&\quad + f_{2g-n}^L \Big(((n-2g+2g-n)^2 - (n-2g+2g-n)) \cdot \tau^{(1)}_{2g-n} \\
&\quad + \; 2 \cdot (g-2g+n) \cdot (n-2g+2g-n) \cdot \tau^{(2)}_{2g-n+1} \\
&\quad + \; (g-2g+n)^2 \cdot \tau^{(3)}_{2g-n+2} \\
&\quad + \; (n-2g+2g-n) \cdot \tau^{(4)}_{2g-n+1} \Big) \Big) \\
&= \frac{1}{n^2} \left(K \left(\left(\left(n - n + \frac{n}{K} \right)^2 - \left(n - n + \frac{n}{K} \right) \right) \cdot \tau^{(1)}_{n-n/K} \right. \right. \\
&\quad \left. + \left(n - n + \frac{n}{K} \right) \cdot \tau^{(4)}_{n-n/K+1} \right) \\
&\quad + \left(K^2 - K \right) \cdot \left(n - n + \frac{n}{K} \right) \cdot \tau^{(3)}_{2(n-n/K)-n+2} \Big) \\
&= \left(\frac{1}{K} - \frac{1}{n} \right) \cdot \tau^{(1)}_{n-n/K} + \frac{1}{K} \cdot \tau^{(3)}_{n-2n/K+2} + \frac{1}{n} \cdot \tau^{(4)}_{n-n/K+1}
\end{aligned}
$$

□

Although this variance has to be much higher than the variance of LpO, this fact doesn't seem to be very intuitive by looking at both variances.

In Chapter 2.2.11, we claimed that there is no universal unbiased estimator for CV, which is shown in Bengio and Grandvalet (2003). However, we just showed

that there is an unbiased estimator for CV, as long as we add new observations to the sample used for CV, such that $n \geq 2g + 2$.

Example 5. Let us view the special case of leave-one-out CV which corresponds to n-Fold CV. Thus subsetting $K = n$ into (2.3.2) results in

$$\begin{aligned} \mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{nCV})) &= \left(\frac{1}{K} - \frac{1}{n}\right) \cdot \tau_{n-1}^{(1)} + \frac{1}{n} \cdot \tau_n^{(3)} + \frac{1}{n} \cdot \tau_n^{(4)} \\ &= \frac{1}{n}\left(\tau_n^{(3)} + \tau_n^{(4)}\right) \end{aligned} \tag{2.3.3}$$

2.3.3 Aspects of Estimating the variance of cross-validation

In CV usually $g \geq n/2$, so $n \leq 2g$. So, the computation of the CV-estimator for the given sample size n would not allow us to estimate the variance for K-Fold CV by the formula derived in the previous section, since $n \geq 2g + 2$ does not hold here. However, we can choose the set-up differently in practice if we are particularly interested in the variance of CV.

For the sample size n, g has to be chosen such that $n \geq 2g + 2$. Then let n_{CV} be the sample size used in K-Fold CV. Thus, since

$$g \geq \frac{K-1}{K} \cdot n_{CV} \tag{2.3.4}$$

and

$$n \geq 2g + 2, \tag{2.3.5}$$

by subsetting (2.3.4) into (2.3.5) we get:

$$\begin{aligned} n &\geq 2\left(\frac{K-1}{K} \cdot n_{CV}\right) + 2 \\ n_{CV} &\leq \frac{K(n-2)}{2(K-1)} \end{aligned} \tag{2.3.6}$$

Now, since g is fixed, let us investigate the possible choices of n_{CV}.

Consider the fact that $(K_1 - 1)/K_1 > (K_2 - 1)/K_2$ for $K_1 > K_2,\ K_1, K_2 \in \mathbb{N}$. Thus n_{CV} is at least $g + 1$, which follows from (2.3.4) by subsetting $K = n_{CV}$:

$$g = \frac{n_{CV} - 1}{n_{CV}} \cdot n_{CV} = n_{CV} - 1$$
$$n_{CV} = g + 1$$

which corresponds to leave-one-out CV.
On the other hand, the sample size for CV is at most $n_{CV} = 2g$, which follows from subsetting $K = 2$:

$$g = \frac{2 - 1}{2} \cdot n_{CV} = \frac{1}{2} n_{CV}$$
$$n_{CV} = 2g$$

Of course, if the goal in practice for a given sample size n is to find the best estimate, this is not the optimal set-up. We would use the whole sample size n instead of any $n_{CV} < n$ for setting up a CV-like procedure. However, since we are interested in the comparison of CV and another design, we need to estimate both variances, which we can only estimate by a set-up where $n \geq 2g + 2$.

2.4 Minimization of a CV-like procedure's variance

In this section, aspects of finding a CV-like procedure with minimal variance for a fixed design size will be established. Again, our investigations will be based on Lee (1990), Chapter 4 in which methods for minimizing the variance of an incomplete U-statistic for a fixed design size are derived. However, the methods introduced there only relate to U-statistics with symmetric kernels. Nonetheless we will be able to make use of them.

2.4.1 Expression of the variance for identifying a minimum variance design

As well as Lee (1990) in Chapter 4, Theorem 3 does, we will derive one additional formulation of the variance of an incomplete U-statistic or — in our case — a CV-like procedure.

Theorem 8. Let $\mathscr{T}^*$ be a test-complete-design. Then

$$\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) = |\mathscr{T}^*|^{-2} \sum_{\gamma=0}^{g} \alpha_\gamma B_\gamma^L \tag{2.4.1}$$

where

$$\alpha_\gamma := \sum_{c=0}^{\gamma} (-1)^{\gamma-c} \binom{\gamma}{c} \xi_c,$$
$$B_\gamma^L := \sum_{c=\gamma}^{g} f_c^L \binom{c}{\gamma}$$

Proof. The following equation can be shown (e.g. analogous to Lee (1990), p.191-192):

$$f_c^L = \sum_{\gamma=c}^{g} (-1)^{\gamma-c} \binom{\gamma}{c} B_\gamma^L$$

Then

$$\begin{aligned}
\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) &\overset{(2.2.11)}{=} |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} f_c^L \cdot \xi_c \\
&= |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} \left(\sum_{\gamma=c}^{g} (-1)^{\gamma-c} \binom{\gamma}{c} B_\gamma^L \right) \cdot \xi_c \\
&= |\mathscr{T}^*|^{-2} \sum_{\gamma=0}^{g} \underbrace{\left(\sum_{c=0}^{\gamma} (-1)^{\gamma-c} \binom{\gamma}{c} \xi_c \right)}_{=\alpha_\gamma} \cdot B_\gamma^L \\
&= |\mathscr{T}^*|^{-2} \sum_{\gamma=0}^{g} \alpha_\gamma B_\gamma^L
\end{aligned}$$

□

The quantities α_γ can be estimated by a U-statistic. Thus, we define $\widehat{\alpha_\gamma}$ as the associated U-statistic.

Lemma 16. B_γ^L can be interpreted as follows:
For $1 \leq \gamma \leq g$, let $S \in \mathcal{S}_0^{(n,\gamma)}$ and let $n^L(S)$ be the number of g-subsets in the design of learning sets $\mathcal{S}^*$ which contain S. For $\gamma = 0$ let $n^L(S) := |\mathcal{S}^*|$ which is justified by viewing the empty set as a subset of every learning set. Then for $0 \leq \gamma \leq g$

$$B_\gamma^L = \sum_{(n,\gamma)} \left(n^L(S)\right)^2 \tag{2.4.2}$$

Proof. The fact, that $B_\gamma^L = \sum_{(n,\gamma)} \left(n^L(S)\right)^2 = \sum_{c=\gamma}^g f_c^L \binom{c}{\gamma}$ for $\gamma = 1, \ldots, g$ can be proved analogous to Lee (1990), p.191-192.
Thus it suffices to show that for $\gamma = 0$,

$$\sum_{c=\gamma}^g f_c^L \binom{c}{\gamma} = \sum_{(n,\gamma)} \left(n^L(S)\right)^2$$

For the left side of the equation we have

$$\sum_{c=\gamma}^g f_c^L \binom{c}{\gamma} \overset{\gamma=0}{=} \sum_{c=0}^g f_c^L \binom{c}{0} = \sum_{c=0}^g f_c^L \overset{(2.2.13)}{=} |\mathcal{S}^*|^2$$

For the right side of the equation we have

$$\sum_{(n,\gamma)} \left(n^L(S)\right)^2 \overset{\gamma=0}{=} \sum_{(n,0)} |\mathcal{S}^*|^2 = |\mathcal{S}^*|^2,$$

since there are $\binom{n}{0} = 1$ ways of choosing 0 elements from n elements.

□

2.4.2 Problem of finding Minimum variance designs for a fixed size

Our goal is to find a design $\mathscr{T}^*$ which minimizes the variance $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))$ for a fixed design size. So, $|\mathscr{T}^*|$ is fixed. Therefore, let us take a closer look at Equation (2.4.1) and consider the two remaining quantities α_γ and B_γ^L. We recognize that α_γ does not depend on the design at all, so B_γ^L is the component which minimizes the variance by an appropriate design. In addition, by Equation (2.4.2), $B_\gamma^L > 0$ for all γ.
However, minimizing all B_γ^L only leads to minimum variance, if all quantities $\alpha_\gamma > 0$.

Lemma 17. A test-complete-design $\mathscr{T}^*$ has minimum variance referring to its size if the quantities B_γ^L over all possible g-subsets of $\mathcal{S}_0^{(n,g)}$ are minimized and if $\alpha_\gamma > 0$ for all $\gamma \in \{1, \ldots, g\}$.

Proof. Follows directly from the above considerations. □

In case of $\alpha_\gamma > 0$ for all $\gamma \in \{0, \ldots, g\}$, ways of minimizing B_γ^L could be found in Lee (1990), Chapter 4.3.2, where the quantity B_ν is the analogue to our B_γ^L.

In this work, however, we will treat a data example, where the quantities α_γ take also negative values.
Our method for finding CV-like procedures with small variances will simply be as follows: we will empirically estimate the variances $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))$ of several CV-like procedures of which we suspect that they have small variance.

Remark 7. Let $\mathscr{T}_1, \mathscr{T}_2 \subset \mathscr{T}_{LpO}$ be two designs. Then for the corresponding CV-like procedures $\widehat{\Delta e}(\mathscr{T}_1)$ and $\widehat{\Delta e}(\mathscr{T}_2)$,

$$\mathbb{E}\left[\min\left\{\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_1)), \mathbb{V}(\widehat{\Delta e}(\mathscr{T}_2))\right\}\right] \neq \min\left\{\mathbb{E}\left(\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_1))\right), \mathbb{E}\left(\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_2))\right\}\right) \tag{2.4.3}$$

This means that by considering several CV-like procedures, their smallest unbiased variance estimator is not unbiased. However, the error becomes smaller as $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_i))$ approaches $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_i))$.

2.5 Convergence in probability of the incomplete to the complete U-statistic under random subsampling, given the data

Before we apply the investigated theory on some data, at long last we will discuss an aspect of the numerical computation of a LpO estimator. We will deal with the question of how many iterations for an approximate estimate of a LpO procedure is needed to get appropriately close to the true LpO estimate. This is especially important for the application on a data set in this work, since we would like to estimate the regular parameters $\tau_d^{(i)},\ i = 1, \ldots, 4$ by complete U-statistics. This, however, is not realizable. Therefore we want to find an approximation by drawing randomly chosen samples from $\{1, \ldots, n\}$ in order to set up evaluation tuples and evaluate the kernel.
So, we are interested in the following question: how do we have to choose the number of samples or iterations for a CV-like procedure with random design in order to approximate LpO satisfactorily?

Therefore, let $\epsilon > 0$ and $\mathscr{T}^*$ be a collection of N randomly chosen ordered $g+1$-subsets of $\{1, \ldots, n\}$. For such a set we will see the first g elements as an unordered set. So this set corresponds to the set S used for learning. The remaining element can be seen as a, so that $(S; a) \in \mathscr{T}^*$ can be seen as an evaluation tuple. The difference of an incomplete version of the LpO procedure $\widehat{\Delta e}(\mathscr{T}^*)$ and the LpO estimate $\widehat{\Delta e}(\mathscr{T}_{LpO})$ is bounded by a pre-specified ϵ, which corresponds to the tolerance:

$$|\widehat{\Delta e}(\mathscr{T}^*) - \widehat{\Delta e}(\mathscr{T}_{LpO})| \leq \epsilon \tag{2.5.1}$$

We will apply Chebyshev's inequality, for instance in Georgii (2008) (Chapter 5) on (2.5.1). We obtain:

$$\begin{aligned} \mathbb{P}(|\widehat{\Delta e}(\mathscr{T}^*) - \widehat{\Delta e}(\mathscr{T}_{LpO})| \geq \epsilon) &\leq \frac{\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))}{N \cdot \epsilon^2} \\ \iff \mathbb{P}(|\widehat{\Delta e}(\mathscr{T}^*) - \widehat{\Delta e}(\mathscr{T}_{LpO})| < \epsilon) &\geq 1 - \frac{\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))}{N \cdot \epsilon^2} \\ &\Big(\geq 1 - \frac{1}{N \cdot \epsilon^2} \Big) \end{aligned} \tag{2.5.2}$$

since $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) \leq 1$ holds for binary classification.
Now let $\epsilon := 10^{-d},\ d \in \mathbb{N}$, so that d corresponds to the number of digits we want to fix. For instance, for the size of the CV-like procedure of $N = 10^{2d+2}$, the probability for a deviation lower than $\epsilon = 10^{-d}$ is at least $1 - \frac{1}{N \cdot \epsilon^2} = 1 - \frac{1}{10^{2d+2} \times 10^{-2d}} = 0.99$.
Note that this estimation of this size holds irrespectively of g or n.

In Fuchs et al. (2013), it is shown by using Theorem 2 of Hoeffding (1963) that bounding the probability of the approximation error can even be more restricted, so that

$$\mathbb{P}(|\widehat{\Delta e}(\mathscr{T}^*) - \widehat{\Delta e}(\mathscr{T}_{LpO})| < \epsilon) \geq 1 - 2\exp(-\epsilon^2 N/2) \tag{2.5.3}$$

Then, if we want to fix d digits, the size of the CV-like procedure has to be at most $N = 10^{2d+1}$ with a probability of at least 0.99:

$$1 - 2\exp\left(-10^{-2d} \times 10^{2d+1} \cdot 1/2\right) = 1 - 2\exp(-5) \approx 0.99$$

Note that in practice we will achieve convergence against $\widehat{\Delta e}(\mathscr{T}_{LpO})$ even faster as long as we use the following set-up:
Let $\mathcal{S}^*$ be a collection of N randomly chosen unordered g-subsets of $\{1, \ldots, n\}$ and $\mathscr{T}^*$ be the corresponding test-complete-design. In this case one would not only use one test observation for each iteration but use all remaining $n - g$ observations for testing instead. This will save computational cost and accelerate

convergence.

Chapter 3

Application on data

In this chapter, we will apply the developed theory to two data problems in order to give a numerical illustration of estimating $\widehat{\Theta}$, $\widehat{\Theta^2}$, $\widehat{\tau_d^{(i)}}$, $\widehat{\xi_c}$ and $\widehat{\alpha_\gamma}$ and thus of getting an estimation for the variance $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*))$. For simplicity, in both applications we took the second loss function of the kernel Γ to be zero, since we were interested in the variance of an error rate and not necessarily in the difference of two error rates. The first example is an illustration of how to find a small-variance design empirically by using an artificial data set and a very simple set-up.
The second example treats a real data problem by which we are going to investigate how the CV estimator behaves in practice.

3.1 Application on an artificial example

In this section, we will apply our theory on a very simple artificially created data example and will show a way of finding a design with small variance in practice. The set-up of a data set with low dimension and a prediction model with very low computational cost allows us to estimate all required quantities for estimating the variance by an appropriate precision.

3.1.1 Set-up

As data we chose a simple parabola as response variable and one predictor. For $i \in \{1, \ldots, n\}$ we have

$$x_i = 2 \cdot i/n$$
$$y_i = x_i^2$$

We chose $n = 80$ as the number of observations. Figure 3.1 shows a scatter plot of x vs. y.

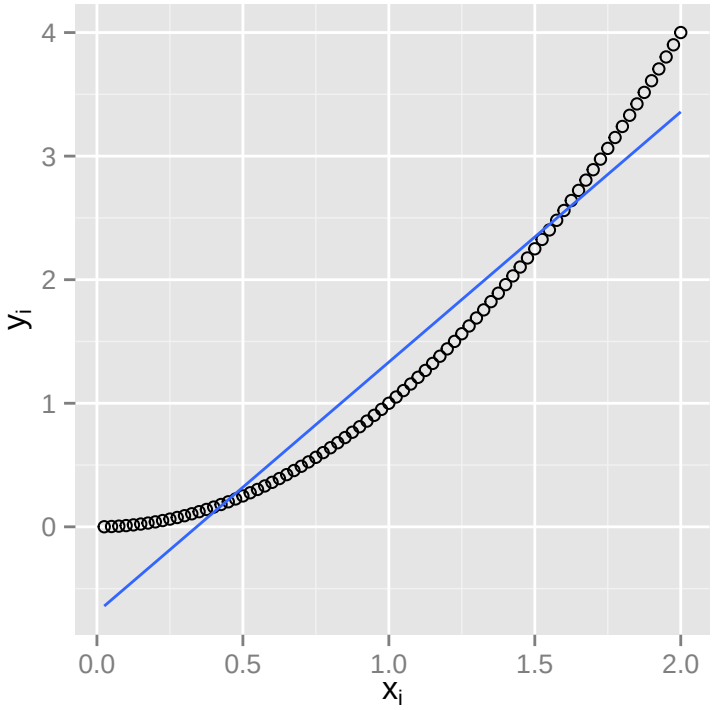


Figure 3.1: Scatterplot of x_i vs. y_i with added linear regression line

As learning set size we chose $g = 10$ in order for $n \geq 2g + 2$ to hold and such that we can investigate e.g. cross-validation for $K = 6$ and $n_{CV} = 12$.
For prediction we used a simple linear model so that our kernel-function is very quick in computation. We therefore used the fast **R**-function `fastLmPure` from the **RcppArmadillo**-package for implementation. Since we want to focus on a kernel-function which maps to values between 0 and 1, we choose the following

loss function

$$L(y_i, \hat{y}_i) = \arctan\{(y_i - \hat{y}_i)^2\} \cdot \frac{2}{\pi},$$

i.e. the squared error loss mapped to $[0; 1)$. Thus, if we let β_{ij}^S, $i = 0, 1$ be the already estimated coefficients of the linear model for the j observation, trained by $\mathfrak{L}_S$, we can formulate the kernel as

$$\Gamma(S; a) = L(\widehat{\beta}_{0,a}^S + \widehat{\beta}_{1,a}^S \cdot x_a \ ; \ y_a) \tag{3.1.1}$$

By this set-up we were able to estimate the quantities of interest.

3.1.2 Estimation of the regular parameter components of the variance

Even for this simple example the computation of the estimators $\widehat{\lambda_d^{(i)}}$, $\widehat{\Theta}$ and $\widehat{\Theta^2}$ by a U-statistic was computationally too excessive. Therefore, we estimated them by drawing between $N = 10^5$ and $N = 10^7$ random subsets (of size $g + 1$, $2g + 2$ or $2g + 2 - d$, according to the degree of the corresponding kernel) from our data set instead of using all possible evaluation tupels. This procedure is theoretically justified by the inequality (2.5.3). Hence, properly speaking, we computed $\widehat{\widehat{\lambda_d^{(i)}}}$, $\widehat{\widehat{\Theta}}$ and $\widehat{\widehat{\Theta^2}}$. However, we will do without this theoretically correct notation in the following.

Estimator $\widehat{\Theta}$ and its variance

The parameter of interest $\widehat{\Theta} = \widehat{\Delta e}(\mathscr{T}_{LpO})$ took a value of 0.0746.
For $\widehat{\Theta^2}$ we received a value of 0.0055. Then for the variance of the prediction error estimator, trivially given by

$$\mathbb{V}(\widehat{\Theta}) = \mathbb{E}(\widehat{\Theta}^2) - \left[\mathbb{E}(\widehat{\Theta})\right]^2, \tag{3.1.2}$$

we had 7.8378×10^{-4}.

Care had to be taken with computing $\widehat{\Theta}^2$: in our case Equation (3.1.2) is equivalent to

$$\mathbb{V}(\widehat{\Theta}(n_{CV})) = \mathbb{E}\left[(\widehat{\Theta}(n_{CV}))^2\right] - \left[\mathbb{E}(\widehat{\Theta}(n_{CV}))\right]^2$$

Hence, $\mathbb{E}\left[(\widehat{\Theta}(n_{CV}))^2\right]$ is estimated by $(\widehat{\Theta}(n_{CV}))^2$ which is not equal to $(\widehat{\Theta}(n))^2$. We estimated this parameter by several samples of size n_{CV} in order to improve the estimate. In particular, $(\widehat{\Theta}(n_{CV}))^2$ can be written as follows.

Let $\mathscr{T}_{LpO}^M$ be the collection of all evaluation tuples containing the set of indices M. Then

$$(\widehat{\Theta}(n_{CV}))^2 = |\mathcal{S}_0^{(n,n_{CV})}|^{-1} \sum_{M \in \mathcal{S}_0^{(n,n_{CV})}} \left(|\mathscr{T}_{LpO}^M|^{-1} \sum_{(S;a) \in \mathscr{T}_{LpO}^M} \Gamma(S;a) \right)^2 \tag{3.1.3}$$

After computing the estimators $\widehat{\lambda_d^{(i)}}$, we were able to compute the estimators $\widehat{\tau_d^{(i)}}$ by (2.2.18). Table 3.1 shows the corresponding values. Figure 3.2 depicts these estimators by scatter plots. There we can see that the estimators $\widehat{\tau_d^{(1)}}$ and $\widehat{\tau_d^{(4)}}$ are positive and grow by d, whereas $\widehat{\tau_d^{(2)}}$ and $\widehat{\tau_d^{(3)}}$ decrease and can take negative values. The fact that here $\widehat{\tau_2^{(1)}} < 0$ may be justified by the inaccuracy of the estimations.

Still taking into account that our estimations might have too less accuracy, figure 3.3 testifies that the quantities $\tau_d^{(1)}$ divided by d indeed grow, in accordance with the inequality (2.2.5). The remaining three quantities seem to decrease after dividing by d.

According to Lemma 14 the variance-MVUE of a CV-like procedure is given by (2.2.21),

$$\mathbb{V}(\widehat{\Delta e(\mathscr{T}^*)}) = |\mathscr{T}^*|^{-2} \sum_{c=0}^{g} f_c^L \cdot \widehat{\xi}_c$$

The computed required estimators $\widehat{\xi}_c$ (s. table 3.2 and figure 3.4) took both neg-

	$\widehat{\tau_d^{(1)}}$	$\widehat{\tau_d^{(2)}}$	$\widehat{\tau_d^{(3)}}$	$\widehat{\tau_d^{(4)}}$
d=0	0.000000	0.000000	0.000000	0.000000
d=1	0.000003	0.000035	0.000000	0.004127
d=2	-0.000008	0.000005	0.000536	0.004531
d=3	0.000009	-0.000090	0.000494	0.005073
d=4	0.000023	-0.000098	0.000408	0.005632
d=5	0.000104	-0.000119	0.000372	0.006244
d=6	0.000134	-0.000177	0.000276	0.006883
d=7	0.000188	-0.000207	0.000232	0.007534
d=8	0.000153	-0.000248	0.000168	0.008337
d=9	0.000328	-0.000282	0.000063	0.009234
d=10	0.000443	-0.000361	0.000036	0.010192
d=11	0.000000	0.000000	-0.000097	0.011273
d=12	0.000000	0.000000	0.000000	0.000000

Table 3.1: Estimators $\widehat{\tau_d^{(i)}}$

ative and positive values and do not seem to decrease or increase monotonically by d.

	$\widehat{\xi_c}$
c=0	0.014902
c=1	0.007851
c=2	0.003940
c=3	-0.002821
c=4	-0.008863
c=5	-0.008301
c=6	-0.008259
c=7	-0.005903
c=8	0.000143
c=9	0.009372
c=10	0.023431

Table 3.2: Estimators $\widehat{\xi_c}$

Figure 3.5 confirms Lemma 7 drawn up by Hoeffding (1948): the estimators $\widehat{\sigma_d^2}$ are all positive and grow after dividing by d for increasing d. We computed these estimators by the equation above (2.2.21).

By Theorem 8, we rewrote the CV-like procedure's variance by (2.4.1). This form enables us to find a design for minimizing this variance in case of $\alpha_\gamma > 0$ for $\gamma = 0, \ldots, g$.

The estimators $\widehat{\alpha_\gamma}$, however, take positive and also negative values (s. table 3.3).

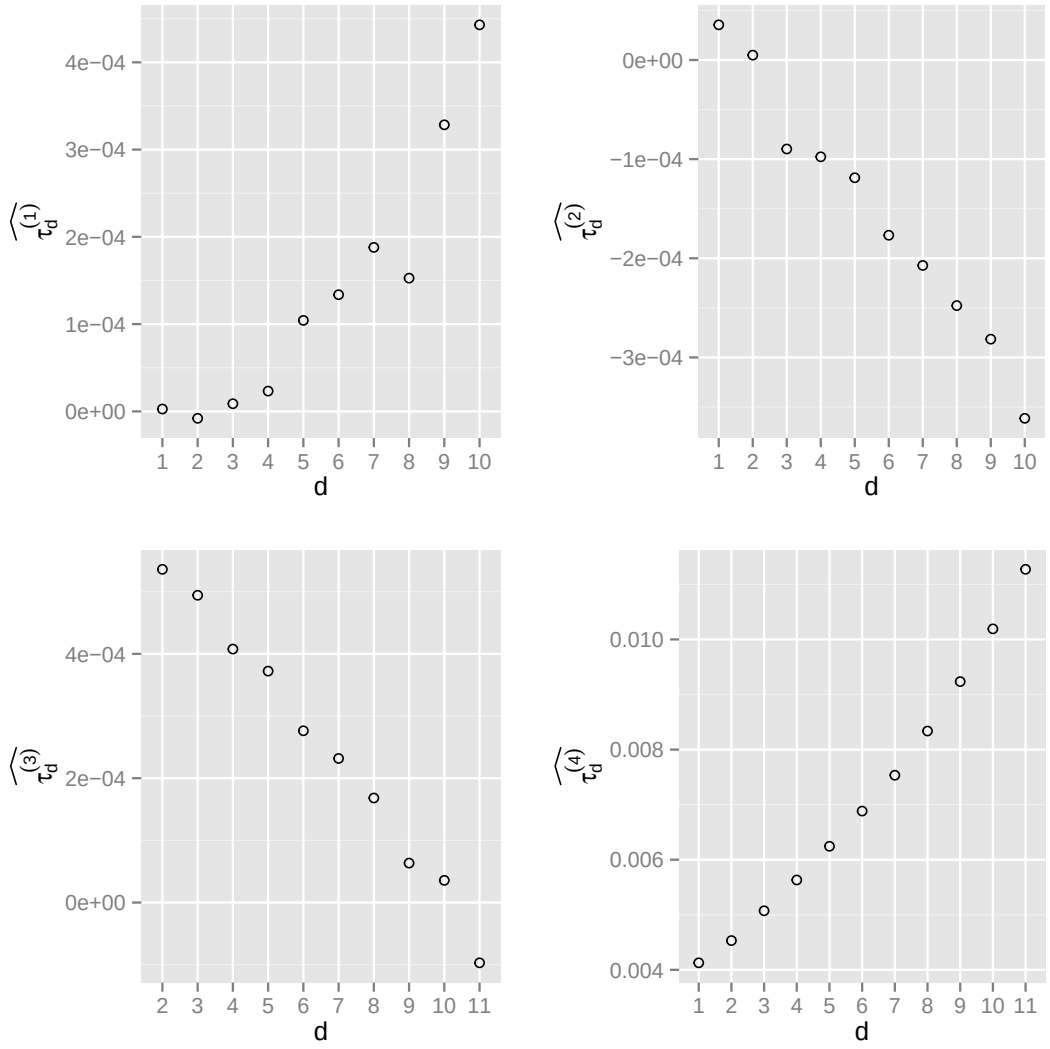


Figure 3.2: Scatter plots for the estimators $\widehat{\tau_d^{(i)}} \neq 0$. The plots show that the quantities $\tau_d^{(i)}$ $i = 2, 3$ can become negative. Thus, these quantites will not be variances.

Figure 3.6 illustrates this fact. Hence, we will not be able to find a design with minimum variance by minimizing the terms B_γ^L. However, we will consider several designs in the next sections of this chapter. We will evaluate their variances empirically to determine small-variance CV-like procedures.

3.1.3 Comparison of 6-Fold cross-validation and LpO

At first, let us investigate the variance in case of K-Fold-cross-validation compared to the variance of the LpO estimator. We chose 6-Fold-CV so that for our $g = 10$ we had $n_{CV} = 12$. For CV, we estimated a variance of $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{6CV})) =$

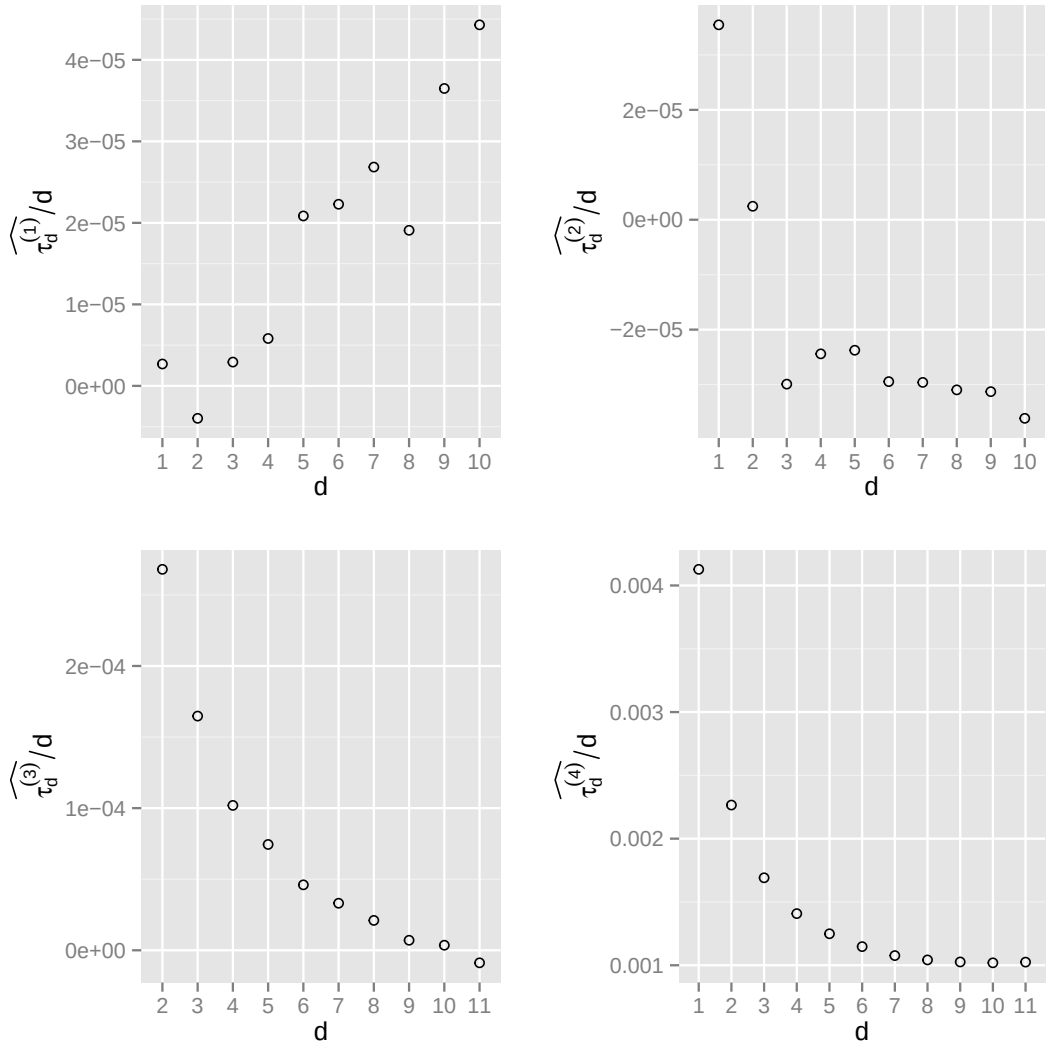


Figure 3.3: Scatter plots for estimators $\widehat{\tau_d^{(i)}} \neq 0$ divided by d. The plots show that the quantities $\tau_d^{(i)}$ $i = 2, 3, 4$ decrease by d. Thus, the examination if those were Hoeffding quantities ζ_d^2 becomes unnecessary. However, it can be confirmed that this property is valid for the quantities $\tau_d^{(i)}$, since their estimators divided by d increase.

0.001. The estimation for the variance of the LpO-estimator was computed again, this time by the estimated quantities $\widehat{\xi_c}$ by formula (2.2.11). For n_{CV}, this variance took a value of $\mathbb{V}(\widehat{\Delta e(\mathscr{T}_{LpO})}) = 8.2306 \times 10^{-4}$. This result suggests that, empirically, LpO has unambiguously smaller variance than K-Fold CV as it should. The computed variance estimator using (3.1.2) resulted in smaller variance (7.8378×10^{-4}) of the LpO compared with the variance of the CV, as well.

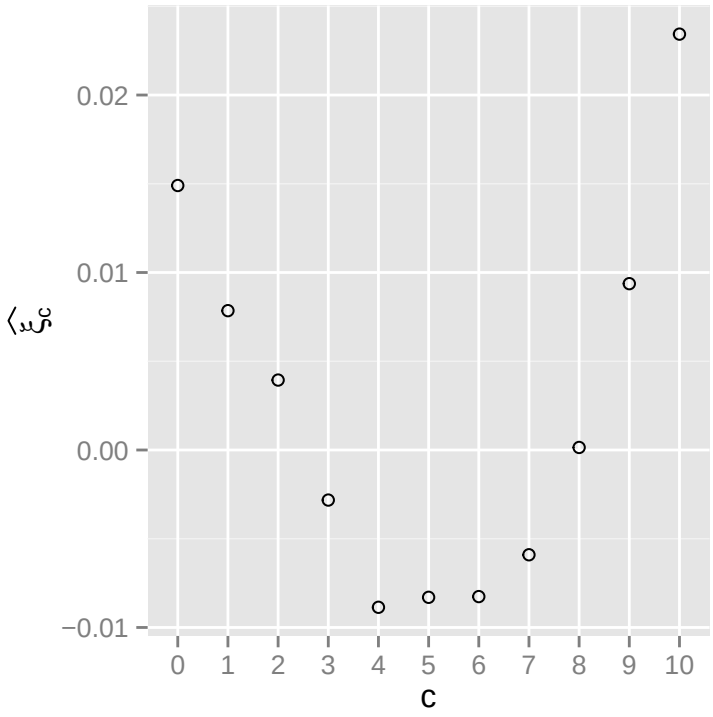


Figure 3.4: Scatter plots for the estimators $\widehat{\xi_c}$. The plot indicates how a design has to be chosen in order to minimize the variance. Since for $c = 3, \ldots, 7$, the $\widehat{\xi_c}$ are negative, these overlap cases minimize the variance. Thus a small-variance design should favor medium-sized learning overlaps over large and small sizes (since for $c = 0, 1, 2, 9, 10$, the estimated quantities $\widehat{\xi_c}$ are positive and thus maximize the variance).

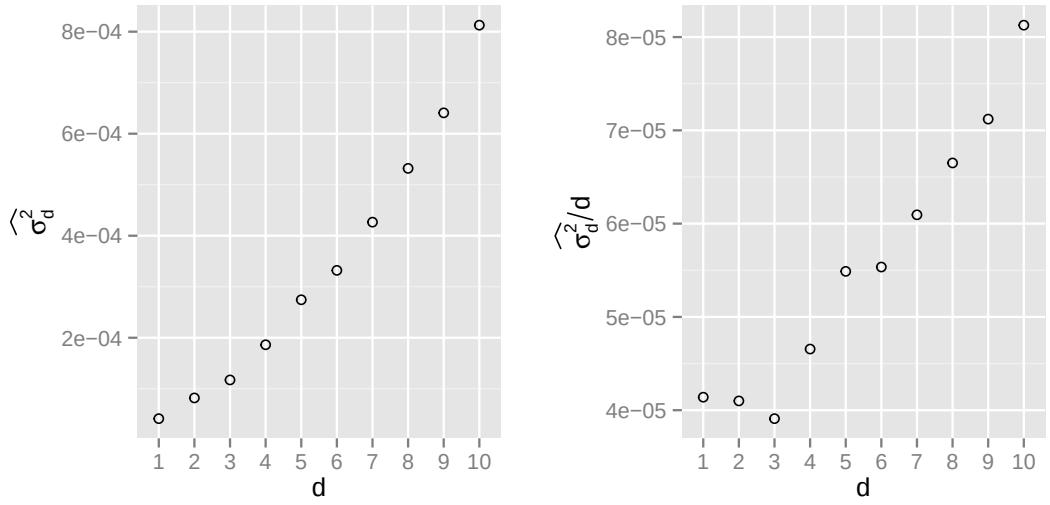


Figure 3.5: Scatter plots for the estimators $\widehat{\sigma_d^2}$ (left figure) and $\widehat{\sigma_d^2}$ divided by d (right figure). The plots show that σ_d^2 increases in both cases as they should, since they are quantities ζ_d^2. So we seem to have satisfactory estimators confirming the theoretical structure.

	$\widehat{\alpha_\gamma}$
γ =0	0.014902
γ =1	-0.007051
γ =2	0.003141
γ =3	-0.005992
γ =4	0.009563
γ =5	-0.007253
γ =6	-0.008064
γ =7	0.046343
γ =8	-0.119002
γ =9	0.237032
γ =10	-0.406962

Table 3.3: Estimators $\widehat{\alpha_\gamma}$

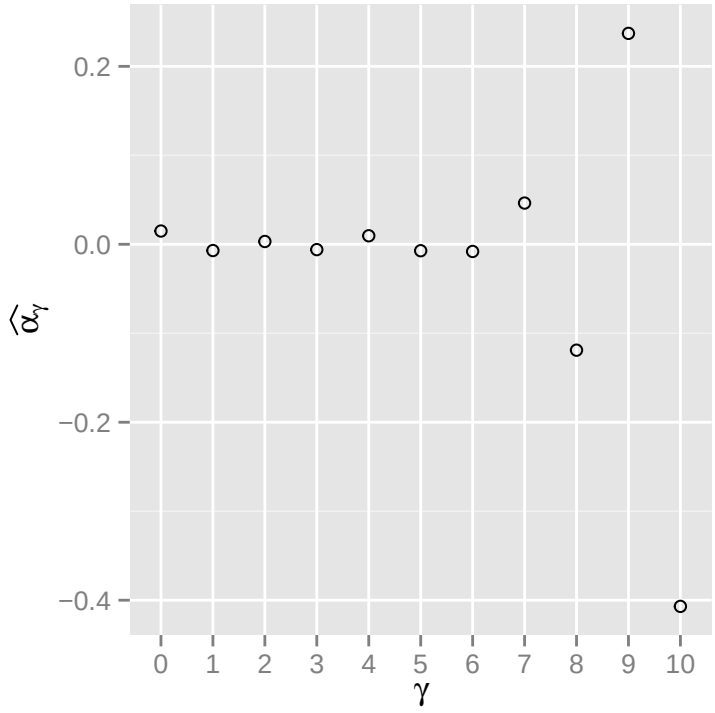


Figure 3.6: Scatter plots for the estimators $\widehat{\alpha_\gamma}$. Generally, the signs of the quantities seem to alternate between positive and negative.

3.1.4 Design with smaller variance than l-Fold-K-Fold CV

We now selected several designs which testified to have small variance. In the introduction of this work, we have already pointed out the well-known problem that K-Fold CV leads to high variance in practice. Thus, one will consider performing CV for a fixed K several times by partitioning the sets differently

(usually at random). We will refer to such a procedure as to *l-Fold-K-Fold CV*, where $K/l \in \mathbb{N}$. The number of iterations or learning sets of such a design is $|\mathcal{S}^*| = l \cdot K$.

Set-up of $|\mathcal{S}^*| = 12$, $n_{CV} = 12$

We considered 2-Fold-6-Fold CV at first so that we had $|\mathcal{S}^*| = 2 \times 6 = 12$ learning sets. We performed this kind of CV several times by partitioning each time two cross-validation-procedures at random. For every case we estimated a variance of 9.054×10^{-4}, computed by $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}^*)) = |\mathscr{T}^*|^{-2} \sum_{\gamma=0}^{g} \widehat{\alpha}_\gamma B_\gamma^L$ according to (2.4.1).

We tried one case of manually selecting the $|\mathcal{S}^*| = 12$ learning sets in a special "pseudo-balanced" design (incidence matrix cf. appendix, table B.1). However, the supposition that such a more "balanced" structure would lead to a lower variance was refuted and again we got exactly the same value as above.

Set-up of $|\mathcal{S}^*| = 12$, $n_{CV} = 13$

We changed our set-up a little by setting $n = 13$ so that we were able to apply a "formally balanced" design, which in literature is called a "Balanced Incomplete Block Design" (BIBD, e.g. in Lee (1990)). We will give the definition of a BIBD in line with our collection of learning sets.

Definition 22. A Balanced Incomplete Block Design in case of a learning set design $\mathcal{S}^*$ is a design in which each learning observation is contained in r learning sets and any pair of learning observations is contained in exactly λ learning sets of size n_{CV}.

Such a design exists for $n = 13$ and $|\mathcal{S}^*| = 26$, where $r = 20$ and $\lambda = 15$. This fact can be checked on the basis of the incidence matrix of this design (cf. appendix B.2).
In order to compare this design appropriately to K-Fold CV, we chose 5-Fold-6-Fold CV for comparison, with $n_{CV} = 13$. Thus the number of learning sets (here $l \cdot K = 30$) even exceeded the one of the BIBD-design of 26. Since for

$n_{CV}/K \notin \mathbb{N}$, we partitioned the a set of only $n_{CV} = 12$ into 6 parts at first, and replaced the observation which was left out systematically at some positions (collection of sets of learning indices, cf. B.3 in the appendix).

For comparison we generated a further design for this set-up: we chose randomly $g = 10$ indices from $\{1, \ldots, n_{CV}\}$ for every learning set. We also generated 30 learning sets. Such a CV-like procedure is sometimes called "Monte Carlo Cross-Validation" (MCCV).

We again computed the variance of the LpO estimator for $n_{CV} = 13$. Table 3.4 represents the corresponding results.

	$\mathbb{V}(\widehat{\Delta e(\mathscr{T}^*)})$
MCCV	0.000798
5-F-6-F CV	0.000762
BIBD	0.000709
LpO	0.000701

Table 3.4: Variance estimators $\mathbb{V}(\widehat{\Delta e(\mathscr{T}^*)})$ for different designs in case of $g = 10$ and $n_{CV} = 13$

The results show that the variance of the MCCV-estimator is higher then the variance of l-Fold-K CV, as one would expect.

However, we have an interesting result for the BIBD variance estimator. It is clearly smaller than the one for CV and is close to the variance of the unique MVUE, the LpO!

It should be mentioned that here we neglected the fact that, by Remark 7, our smallest variance estimator is, strictly speaking, not unbiased anymore.

3.2 Application on a micro-array data set

In this section, we will make use of the theory developed in chapter 2 by applying it on a real data problem. We will perform K-Fold CV and estimate its unbiased variance estimator by computing the required estimators for this procedure.

3.2.1 Data

We chose a data set on breast cancer by GEO (`http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE25055`). The data set contains micro-array data and some clinical variables. We focused on predicting the binary variable "response to chemotherapy" merely by the genetic variables.

According to Hatzis et al. (2011), the patients in the data set "were those with newly diagnosed ERBB2 (HER2 or HER2/neu)-negative breast cancer treated with chemotherapy containing sequential taxane and anthracycline-based regimens (then endocrine therapy if estrogen receptor [ER]-positive)".

Equivalently to Hatzis et al. (2011), we split the data into estrogen receptor(ER)-positive and ER-negative cases and used the ER-positive-reduced data set for our analysis.

Therefore, after splitting the data set and removing the observations in which the outcome was missing, there were $n = 248$ remaining observations from the original 508 observations.

The outcome variable is binary and corresponds to the two responder groups "pathologic complete response or minimal residual cancer burden" (RCB-0/RCB-I, 46 observations) and "moderate or extensive residual cancer burden" (RCB-II/III, 202 observations).

We restricted the predictor variables to those affymetrix probe set identifiers targeting genes which coded for a protein whose identifier was contained in the "Integrated Breast Cancer Pathway" of Wikipathways (`http://wikipathways.org/index.php/Pathway:WP1984`). In this way, we reduced the predictor space from originally high-dimensional to a medium-dimensional. The resulting number of variables was 190.

Hence, we had a data set of a satisfying number of observations and of medium-dimensionality at hand which is suitable for a binary classification.

3.2.2 Set-up and used learning algorithm

For the total sample size of $n = 248$, we chose $g = 80$ and $n_{CV} = 100$. Thus $n \geq 2g + 2$ holds and we had $K = n_{CV}/(n_{CV} - g) = 5$ for performing K-Fold CV.

As a learning algorithm for classification we chose dimensionality reduction by Partial Least Squares followed by linear discriminant analysis. For the implementation of this algorithm we chose the **R**-Function `pls_ldaCMA` from the **CMA**-package. We chose this algorithm because it performs quickly and is applicable to medium to high dimensional data. Here we used a number of iterations between $N = 5 \times 10^5$ and $N = 3 \times 10^6$.

3.2.3 Computation of the variance of CV

For computing the variance of K-Fold CV we had to compute merely 3 instead of all estimators for $4g + 1$ quantities $\tau_d^{(i)}$!

Namely, according to formula (2.3.2), we had to estimate the quantities $\tau^{(1)}_{n-n/K}$, $\tau^{(3)}_{n-2n/K+2}$ and $\tau^{(4)}_{n-n/K+1}$. For our set-up these correspond to $\widehat{\tau^{(1)}_{80}}$, $\widehat{\tau^{(3)}_{62}}$ and $\widehat{\tau^{(4)}_{81}}$. We were able to compute these estimators as described in Section 3.1.2. Table 3.5 shows the corresponding computed values. We observe that the estimated quantity $\widehat{\tau_d^{(4)}}$ is of enormous size compared to $\widehat{\tau_d^{(1)}}$, $\widehat{\tau_d^{(3)}}$.

$\widehat{\Theta}_{CV}$	$\widehat{\tau^{(1)}_{80}}$	$\widehat{\tau^{(3)}_{62}}$	$\widehat{\tau^{(4)}_{81}}$
0.240000	0.000410	0.001418	0.176980

Table 3.5: Estimations for the error rate and required quantities for the variance, for 5-Fold CV if $g = 80$ and $n_{CV} = 100$.

The estimator for the unconditional error rate took a value of $\widehat{\Delta e}(\mathscr{T}_{CV}) = \widehat{\Theta}_{CV} = 0.24$.

By our variance formula for CV (2.3.2) we computed the variance of CV and got $\mathbb{V}(\widehat{\Delta e}(\mathscr{T}_{CV})) = 0.0021$.

Chapter 4

Summary

Our main goal was to solve the following Problem: common CV-like procedures have high variance and, in addition, there are no unbiased variance estimators or correct test procedures available.

We considered the CV-like procedure of maximum size, the LpO, which is a complete U-statistic and thus has minimal variance among all CV-like procedures, since it is the unique MVUE. The problem is, however, that the LpO is not realizable in practice. Hence, we pursued the goal to find a CV-like procedure with still small variance but which is realizable in practice, i.e. contains way less iterations than LpO.

We therefore examined the theory of incomplete U-statistics, since every CV-like procedure is an incomplete U-statistic, in fact. We investigated the structure of its variance and tried to find a way to minimize it.

We found a general formula for the variance of a CV-like procedure for $n \geq 2g+2$ which is estimable and the estimator again is the unique MVUE. We thus found this unbiased variance estimator for K-Fold cross-validation.

We encountered several difficulties in minimizing our established variance, but tried to discover designs with small variance empirically.

For a specific set-up, we indeed found a design — a balanced incomplete block design — having smaller variance than several times repeated cross-validation.

At last, we applied the established variance estimator on a real data problem and were able to compute our unbiased variance estimator for 5-Fold CV.

Appendix A

Further proofs and equations

A.1 Mistake in proof of theorem 1 of Lee (1990), Chapter 4.3.1

Let U_n^* be an incomplete U-statistic based on a fixed design $\mathcal{D} \subset \mathcal{S}_0^{(n,m)}$ and let U_n^* be the corresponding U-statistic. Further let $S_1, \ldots, S_{|\mathcal{D}|}$ be the sets of the design.

Assertion 1. The assumption of Lee (1990) that all $Cov(U_n^*, \Phi_0(S_j))$ are the same does not hold in general, i.e.

$$\exists_{i,j \in \{1,\ldots,|\mathcal{D}|\}} \; : \; Cov(U_n^*, \Phi_0(S_j)) \neq Cov(U_n^*, \Phi_0(S_i)) \tag{A.1.1}$$

Proof. If one assumes that

$$\forall_{i,j \in \{1,\ldots,|\mathcal{D}|\}} \; : \; Cov(U_n^*, \Phi_0(S_j)) = Cov(U_n^*, \Phi_0(S_i)),$$

then this also holds for the special case that $\mathcal{D} = \{S_1, S_2, S_3\}$, where

$$|S_1 \cap S_2| = m - 1$$
$$|S_1 \cap S_3| = \varnothing$$
$$|S_2 \cap S_3| = \varnothing$$

Then

$$\begin{aligned} Cov(U_n^*, \Phi_0(S_1)) &= \frac{1}{3} \cdot \Bigg(\underbrace{Cov(\Phi_0(S_1), \Phi_0(S_1))}_{=\zeta_m^2} + \underbrace{Cov(\Phi_0(S_2), \Phi_0(S_1))}_{\zeta_{m-1}^2} \\ &\qquad + \underbrace{Cov(\Phi_0(S_3), \Phi_0(S_1))}_{=\zeta_0^2=0} \Bigg) \\ &= \frac{1}{3} \cdot \left(\zeta_m^2 + \zeta_{m-1}^2\right) \end{aligned}$$

and

$$\begin{aligned} Cov(U_n^*, \Phi_0(S_3)) &= \frac{1}{3} \cdot \Bigg(\underbrace{Cov(\Phi_0(S_1), \Phi_0(S_3))}_{=\zeta_0^2=0} + \underbrace{Cov(\Phi_0(S_2), \Phi_0(S_3))}_{=\zeta_0^2=0} \\ &\qquad + \underbrace{Cov(\Phi_0(S_3), \Phi_0(S_3))}_{=\zeta_m^2} \Bigg) \\ &= \frac{1}{3} \cdot \zeta_m^2 \\ &\neq Cov(U_n^*, \Phi_0(S_1)), \end{aligned}$$

since in general $\zeta_{m-1}^2 \neq 0$. This is a contradiction to the assumption above and thus (A.1.1) is true. □

A.2 Reformulation of definition of $\tau_d^{(i)},\ i = 1, \ldots, 4$

$$\begin{aligned} \tau_d^{(1)} &= Cov\left(\Gamma(\{1, \ldots, g\}; g+1),\right. \qquad \text{(A.2.1)} \\ &\quad \left.\Gamma(\{1, \ldots, d\} \cup (\{1, \ldots, 2g+1-d\} \setminus \{1, \ldots, g+1\}); 2g+2-d)\right) \\ \tau_d^{(2)} &= Cov\left(\Gamma(\{1, \ldots, g\}; g+1),\right. \qquad \text{(A.2.2)} \\ &\quad \left.\Gamma(\{1, \ldots, d-1, g+1\} \cup (\{1, \ldots, 2g+1-d\} \setminus \{1, \ldots, g+1\}); 2g+2-d)\right) \\ \tau_d^{(3)} &= Cov\left(\Gamma(\{1, \ldots, g\}; g+1),\right. \qquad \text{(A.2.3)} \\ &\quad \left.\Gamma(\{1, \ldots, d-2, g+1\} \cup (\{1, \ldots, 2g+2-d\} \setminus \{1, \ldots, g+1\}); d-1)\right) \end{aligned}$$

$$\tau_d^{(4)} = Cov\left(\Gamma(\{1,\ldots,g\}; g+1),\right. \tag{A.2.4}$$
$$\left.\Gamma(\{1,\ldots,d-1\} \cup (\{1,\ldots,2g+2-d\} \setminus \{1,\ldots,g+1\}); g+1\right)$$

Appendix B

Matrices for learning set designs

$$N^L = \begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0
\end{pmatrix}$$

Table B.1: Learn incidence matrix for pseudo-balanced design ($g = 10$, $n_{CV} = 12$, $|\mathcal{S}^*| = 12$)

$$N^L = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1
\end{pmatrix}$$

Table B.2: Learn incidence matrix for BIBD ($g = 10$, $n_{CV} = 13$, $|\mathcal{S}^*| = 26$)

$$\mathcal{S}^* = \begin{pmatrix}
13 & 2 & 3 & 4 & 5 & 6 & 9 & 10 & 11 & 12 \\
1 & 13 & 3 & 5 & 6 & 7 & 8 & 10 & 11 & 12 \\
2 & 3 & 13 & 5 & 6 & 7 & 8 & 9 & 10 & 12 \\
1 & 3 & 4 & 13 & 7 & 8 & 9 & 10 & 11 & 12 \\
1 & 2 & 4 & 5 & 13 & 7 & 8 & 9 & 10 & 11 \\
1 & 2 & 3 & 4 & 5 & 13 & 8 & 9 & 11 & 12 \\
2 & 3 & 4 & 6 & 7 & 8 & 13 & 10 & 11 & 12 \\
1 & 2 & 4 & 5 & 7 & 8 & 9 & 13 & 11 & 12 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 13 & 10 \\
1 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 11 & 13 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 10 & 11 & 12 \\
1 & 2 & 3 & 5 & 6 & 8 & 9 & 10 & 11 & 12 \\
13 & 2 & 3 & 4 & 5 & 7 & 8 & 10 & 11 & 12 \\
1 & 13 & 3 & 5 & 6 & 8 & 9 & 10 & 11 & 12 \\
1 & 2 & 13 & 4 & 5 & 6 & 7 & 8 & 9 & 12 \\
1 & 2 & 3 & 13 & 6 & 7 & 8 & 9 & 10 & 11 \\
2 & 4 & 5 & 6 & 13 & 8 & 9 & 10 & 11 & 12 \\
1 & 3 & 4 & 5 & 6 & 13 & 9 & 10 & 11 & 12 \\
1 & 2 & 3 & 4 & 5 & 6 & 13 & 9 & 10 & 12 \\
2 & 3 & 4 & 5 & 6 & 7 & 8 & 13 & 11 & 12 \\
1 & 2 & 4 & 5 & 6 & 7 & 8 & 9 & 13 & 11 \\
1 & 2 & 3 & 4 & 6 & 7 & 9 & 10 & 11 & 13 \\
1 & 3 & 4 & 5 & 7 & 8 & 9 & 10 & 11 & 12 \\
1 & 2 & 3 & 5 & 6 & 7 & 8 & 9 & 11 & 12 \\
13 & 3 & 4 & 5 & 6 & 7 & 8 & 10 & 11 & 12 \\
1 & 13 & 3 & 4 & 5 & 8 & 9 & 10 & 11 & 12 \\
1 & 3 & 13 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
1 & 2 & 3 & 13 & 5 & 6 & 7 & 8 & 9 & 12 \\
1 & 2 & 4 & 6 & 13 & 8 & 9 & 10 & 11 & 12 \\
1 & 2 & 3 & 4 & 5 & 13 & 7 & 9 & 10 & 11
\end{pmatrix}$$

Table B.3: Collection of sets of learning indices for 5-Fold-6-Fold CV (as matrix, one row corresponds to one learning set), for $g = 10$, $n_{CV} = 13$, $|\mathcal{S}^*| = 30$

Appendix C

Code

The attached CD-ROM contains the files and **R**-scripts used for the computations and figures in this work.

The code is available in the group folder of the computational molecular biology group on IBE.

Appendix D

R Session Info

```
## ##------ Mon Mar 31 00:55:41 2014 ------##
## R version 3.0.1 (2013-05-16)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=German_Germany.1252  LC_CTYPE=German_Germany.1252
## [3] LC_MONETARY=German_Germany.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] tikzDevice_0.7.0 filehash_2.2-2   gridExtra_0.9.1  xtable_1.7-3
## [5] ggplot2_0.9.3.1  knitr_1.5
##
## loaded via a namespace (and not attached):
##  [1] colorspace_1.2-4   dichromat_2.0-0    digest_0.6.4
##  [4] evaluate_0.5.1     formatR_0.10       gtable_0.1.2
##  [7] highr_0.3          labeling_0.2       MASS_7.3-26
## [10] munsell_0.4.2      plyr_1.8           proto_0.3-10
## [13] RColorBrewer_1.0-5 reshape2_1.2.2     scales_0.2.3
## [16] stringr_0.6.2      tools_3.0.1
```

Bibliography

Bengio, Y. and Y. Grandvalet (2003). No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research 5*, 1089–1105.

Ferguson, T. S. (2005). U-statistics: Notes for statistics.

Fuchs, M., R. Hornung, R. d. Bin, and A.-L. Boulesteix (2013). A u-statistic estimator for the variance of resampling-based error estimators. pp. 1–15.

Georgii, H. (2008). *Stochastics: Introduction to Probability and Statistics*. De Gruyter textbook. Walter De Gruyter.

Halmos, P. R. (1946). The theory of unbiased estimation. *The Annals of Mathematical Statistics 17*(1), 34–43.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. New York and NY and USA: Springer New York Inc.

Hatzis, C., L. Pusztai, V. Valero, D. J. Booser, L. Esserman, A. Lluch, T. Vidaurre, F. Holmes, E. Souchon, H. Wang, M. Martin, J. Cotrina, H. Gomez, R. Hubbard, J. I. Chacón, J. Ferrer-Lozano, R. Dyer, M. Buxton, Y. Gong, Y. Wu, N. Ibrahim, E. Andreopoulou, N. T. Ueno, K. Hunt, W. Yang, A. Nazario, A. DeMichele, J. O'Shaughnessy, G. N. Hortobagyi, and W. F. Symmans (2011). A genomic predictor of response and survival following taxane-anthracycline chemotherapy for invasive breast cancer. *JAMA 305*(18), 1873–1881.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics 19*(3), 293–325.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association 58*(301), 13–30.

Lee, A. J. (1990). *U-statistics*. Statistics. New York and NY [u.a.]: Dekker.

Shao, J. (1993). Linear model selection by cross-validation. *J. of the American Statistical Association 88*, 486–494.