MONTE CARLO CROSS-VALIDATION FOR RESPONSE SURFACE BENCHMARK

A. Beschorner, M. Voigt, K. Vogeler

Institute for Fluid Mechanics,
Faculty of Mechanical Engineering
Technische Universität Dresden
Germany
andre.beschorner@mailbox.tu-dresden.de

Abstract. When meta-models are fitted to the underlying data it is essential to validate the model by a trustworthy criterion. Many industry relevant applications are characterized by a high input parameter dimension and time consuming, cost intensive deterministic computations. In these typical cases the data size is not much higher than the amount of coefficients in the polynomial equation of the meta-model, which will lead to an overestimation of the model quality determined by the Coefficient of Determination (CoD).
An alternative benchmark criterion for response surfaces can be delivered by cross-validation (CV), where an overestimation of the meta-model quality is unusual.
This paper will use a published industry relevant example [4] to compare the simple CoD with a Monte Carlo cross-validation CoD (CoD_{MCCV}). Detailed investigations of the MCCV-method are made using a fast calculating test-model with similar characteristics as the original deterministic model. The results will show the influence of the splitting ratio, the number of cross-validation runs and the number of the deterministic samples in the database on the CoD_{MCCV} result values and their variance. To predict the variance we will give correlations, for a code internal adjustment of MCCV calculation parameters. The discussion will point out the importance of the sample to coefficients ratio (SCR) and conclude the advantages and disadvantages of the tested method.

a, b, c, d, e, f, g, h	adjustment parameter	O	order of response surface
C	number of coefficients	SCR	sample to coefficient ratio
CoD	Coefficient of Determination	SR	splitting ratio
CoD_{MCCV}	cross-validation CoD	$SRCV$	simple random CV
CV	cross-validation	x	input variable
E	number of input variables	y	result variable
i, j	count variables	$\tilde{y}$	meta-model result variable
K	number of folds	$\overline{y}$	mean of variable
$KFCV$	K-fold CV	σ	standard deviation
$MCCV$	Monte Carlo CV	$_t$	training (samples)
n	number of samples	$_v$	validation (samples)

1 Introduction

"When a validity coefficient is computed from the same data used in making an item analysis, this coefficient cannot be interpreted uncritically. And, contrary to many statements

in the literature, it cannot be interpreted 'with caution' either. There is one clear interpretation for all such validity coefficients. This interpretation is – 'Baloney!'" [1]
This conclusion by Cureton should remind the investigator of any data fitting problem, that the commonly used Coefficient of Determination (CoD) is not a stable criterion for model fitness estimation in any way. In the second part of this paper we will discuss the dependency of the CoD on the number of coefficients n in the meta-model and the number of simulations in the dataset. Afterwards we will show how this misleading model fitness interpretation by the CoD can be avoided by using a cross-validation procedure.
Cross-validation as a method for model fitness evaluation was proposed by Mosier [6]. The concept of cross-validation is to use separate data sets for model training and evaluation respectively. Since in most real applications the amount of data is limited, it is necessary to split the number of data-points n into a training data set n_t and a validation data set n_v. Over the last decades different kinds of cross-validation had been proposed, which uses different splitting procedures. A selection of them should be described here.

1.1 Cross-validation procedures

Leave-one-out cross-validation (LOOCV), which is also called delete-one CV or ordinary CV is the simplest exhaustive CV procedure. Every single data point is successively left out from the training dataset and used for validation as described by Stone [11]. The number of CV-runs is equal to the number of samples n in the dataset. As a test criteria for the LOOCV Myers and Montgomery [7] suggest the $R^2_{prediction}$ that uses the predictive error sum of squares ($PRESS$)

$$R^2_{prediction} = 1 - \frac{PRESS}{SST} = 1 - \frac{\sum_{i=1}^{n}[y_{vi} - \tilde{y}_{vi}]^2}{SST} \tag{1}$$

and the total sum of squares (SST)

$$SST = \mathbf{y}^T\mathbf{y} - \frac{\left[\sum_{i=1}^{n} y_i\right]^2}{n}. \tag{2}$$

Alternatively they show the possibility to compute the $R^2_{prediction}$ without cross validation by using the diagonal elements h_{ii} of the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$

$$R^2_{prediction} = 1 - \frac{\sum_{i=1}^{n}\left(\frac{y_i - \tilde{y}_i}{1-h_{ii}}\right)^2}{SST}. \tag{3}$$

Leave-p-out cross-validation (LPOCV) or delete-d multifold CV (Zhang [13]) is very close to the LOOCV, but here a defined number p of data points are left out from the training set and used to validate each response surface. This procedure is also exhaustive since every possible subset p is left out once.

K-fold cross-validation (KFCV) also called V-fold cross-validation is computationally less expensive compared to LOOCV and LPOCV. It was introduced by Geisser [3]. The complete dataset is split into K subsets of approximately equal size n/K. Each subset is successively used as validation set, whilst the other subsets are used to train the response surface. The number of CV-runs is equal to K – the number of folds which is defined

between $2, \ldots, n$ but should not be higher than $n/3$ in a practical sense. If K would be equal to n the procedure would be identical to LOOCV. The described method is a partially data splitting procedure since not every possible combination of validation subsets is used.

Another method with partial data splitting is **simple random cross-validation (SRCV)** or **Monte Carlo cross-validation (MCCV)** respectively. The method has been presented by Picard and Cook [8] and has been compared to the above listed procedures by Shao [10] and Xu [12], while they demonstrate outperforming results for the MCCV applied in model selection. Compared to the KFCV the subsets are selected randomly. Moreover the number of CV-runs and the validation subset size expressed by the splitting ratio $SR = n_t/n$ are independently selectable. This additional freedom of method configuration makes it necessary to identify practical ranges for the configuration parameters.

The MCCV method will be evaluated in paragraph three of this paper by using a test-model, which is emulating an industry relevant probabilistic investigation. Next to the variation of the splitting ratio $SR = n_t/n$ and the number of CV-runs we will also show the influence of n – the number of data points in the complete data set. We will also compare the CoD [2], [9]

$$CoD = \left(\frac{\sum_{i=1}^{n} [(\tilde{y}_i - \bar{\tilde{y}}) \cdot (y_i - \bar{y})]}{\sqrt{\sum_{i=1}^{n} (\tilde{y}_i - \bar{\tilde{y}})^2 \cdot \sum_{i=1}^{n} (y_i - \bar{y})^2}} \right)^2 = \left(\frac{Cov(\tilde{\mathbf{y}}, \mathbf{y})}{\sqrt{Var(\tilde{\mathbf{y}})} \cdot \sqrt{Var(\mathbf{y})}} \right)^2 \tag{4}$$

to the CoD_{CV} results that are computed in the same way but using only the selected validation samples

$$CoD_{CV} = \left(\frac{\sum_{i=1}^{n_v} [(\tilde{y}_{iv} - \bar{\tilde{y}}_v) \cdot (y_{iv} - \bar{y}_v)]}{\sqrt{\sum_{i=1}^{n_v} (\tilde{y}_{iv} - \bar{\tilde{y}}_v)^2 \cdot \sum_{i=1}^{n_v} (y_{iv} - \bar{y}_v)^2}} \right)^2 = \left(\frac{Cov(\tilde{\mathbf{y}}_\mathbf{v}, \mathbf{y}_\mathbf{v})}{\sqrt{Var(\tilde{\mathbf{y}}_\mathbf{v})} \cdot \sqrt{Var(\mathbf{y}_\mathbf{v})}} \right)^2 \tag{5}$$

to discuss the usefulness of cross-validation. Note that none of the response surfaces computed with a training subset and validated with the corresponding validation subset is used as the final meta-model. The polynomial equation of the final meta-model is computed with all data points.

In general it is very important to know about the predictability of a meta-model since it might be used to rapidly produce new data with a Monte Carlo simulation using the polynomial equation. Another application is to estimate the importance of the input parameters using the meta-model, which is expressed by the Coefficient of Importance (CoI). If the model fitness to the underlying data is lower than expected by the investigator it might lead to misinterpretation of parameter importance or erroneous interpretations with the new data calculated on the meta-model.

1.2 Deterministic test-model used for evaluation

The probabilistic investigation of the impact of manufacturing variability on the performance of a ten stage high pressure compressor, by Lange et al. [4] is in the core of interest in this paper. In his investigation Lange described the rotor blade geometry of each rotor

stage by 14 classical profile parameters. Due to the ten compressor stages there are 140 probabilistic input variables in the complete simulation.
The parameter scatter has been determined by real blade measurement with a 3D scanner. For the probabilistic post processing by meta-models, linear regression models have been used, which need 141 data points in minimum.
Since the CFD simulation of a ten stage compressor at seven operation points took about 3 days on an Intel® Core™ i7 machine, it was decided to use 200 virtual realizations created by optimized Latin Hypercube sampling. The results of this huge probabilistic simulation showed that there are mainly two important parameters in each stage, with a variation of their influence depending on the stage and the operation point. The fitness of the meta-models evaluated with the *CoD* is between 0.95 and 0.98.

For a detailed investigation of the *CoD* behavior and possible improvements on meta-model benchmark by cross-validation it was necessary to replace the above described deterministic model by a fast calculating test-model. The created test-model follows the idea of broadly emulating the main characteristics of the original model, but neglecting the operation point variation.

The here used HiPar-test-model (for **hi**gh input **par**ameter space) is using 140 input variables x_i with $i \in \{0, \ldots, 139\}$ computing six different solution variables $y_1, \ldots, y_6$. The input space of 140 dimensions is internally split into 10 groups - similar to the 10 compressor stages, controlled by $k \in \{1, \ldots, 10\}$. Hence there are 14 input parameters in each group.
The HiPar-test-model uses a combination of two public test-models mentioned in the collection of Molga et. al [5]. For $i = i_G \in \{0, 14, 28, 42, \ldots, 126\}$ a modified version of Griewangk's test function is used. Two variables x_i and x_{i+1} are computed within this function with a strong influence on the solution. Hence two variables of each group have a higher importance. If $i \neq i_G$ Rosenbrock's test function is used. It has a lower influence and is hard to approximate by a polynomial regression model. The standard formulations of the test functions can be found in [5]. The modified Griewangk function as implemented in the deterministic test-model is as follows:

$$y_{Griew.k} = a\left[\left((bx_i)^{(e_1\cdot(c_1-j)/c_1)} + d(bx_{i+1})^{(e_2\cdot(c_2-j)/c_2)}\right) + f\left(\frac{\cos(bx_i)}{\sqrt{i+1}} \cdot \frac{\cos(bx_{i+1})}{\sqrt{i+2}}\right)\right]. \quad (6)$$

The step-wise exponent reduction using variable j as a counter is deactivated by setting j to a fixed value of 1. All other model adjustment parameters are given in Tab. 1.

$$y_{Rose.k} = y_{Griew.k} + \sum_{i=p+q(k)=0}^{p+q(k)=139}\left[h_1(x_{i+1} - x_i^2)^2 + h_2(1 - x_i)^2\right] \quad (7)$$

$$y_1, \ldots, y_6 = \sum_{k=1}^{10}\left[\frac{g}{g+k} \cdot \frac{y_{Rose.k}}{14}\right] \quad (8)$$

Rosenbrock's function as implemented in the test-model is defined by equation (7), where $p \in \{1, \ldots, 13\}$ and $q(k) = 14 \cdot (k-1)$. So that, $i \in \{1, \ldots, 13\}; \{15, \ldots, 27\}; \ldots$. The results of each input variable group are added up to the result value $y_1, \ldots, y_6$, as shown in equation (8). There is a parameter influence reduction with increasing k.

variable	y_1	y_2	y_3	y_4	y_5	y_6
a	1/175	1/150	1/185	1/240	1/250	1/150
b	600	600	600	600	600	600
c_1	140	110	85	60	66	75
c_2	300	180	200	500	500	300
d	1.3	1.3	1.3	1.4	1.3	1.3
e_1	1.44	1.40	1.45	1.49	1.51	1.455
e_2	1.28	1.26	1.3	1.23	1.2	1.3
f	1000	200	400	300	2500	300
g	30	30	30	25	20	30
h_1	5	7.5	9.5	10.45	11.3	17.9
h_2	3	2	2.2	12	13	2

Table 1: Model adjustment parameter setting in the HiPar-test-model for the six different meta-models $y_1, \ldots, y_6$

The parameter settings for the six output variables has been chosen to determine first order response surfaces with Coefficients of Determination of 0.98, 0.97, 0.96, 0.95, 0.94, 0.93 using a number of samples of $n = 200$. For a higher sample size the CoD will decrease as described in the next paragraph. All 140 input parameter distributions are generated by latin hypercube sampling with an uniform distribution in the interval [0,1].
The test-model is coded in C++ and is designed very flexible. It can be easily changed in its amount of input parameters and other model characteristics, to perform further investigations e.g. on the computation time in chapter 3.3. Next to the described test-model with six solution variables, we used two other test-model types to enlarge the insight of this investigation for some aspects. So there are up to 22 meta-models included in the result analysis.

2 Characteristics of the CoD

The used test-model with 140 input variables is designed to produce data which can be approximated by first order response surfaces with CoD values between 0.93 and 0.98 for the six result variables, which is the same range as in the ten stage compressor model investigated by Lange [4]. This is of course only true for the same sample to coefficients ratio of $SCR = 1.4$ when the number of samples is $n = 200$. It is important to see the models CoD values always with respect to the existing SCR, since the CoD strongly depends on the SCR.

The left graph of Figure 1 shows the CoD for the six solution response surfaces as function of the SCR. For a SCR of 1.4 the values are equal to the original deterministic model. By increasing n, the estimated quality of the meta-models is strongly decreasing. The slope of the decrease is stronger for models with lower CoD values.
When the SCR reaches 12 the decrease stops and the CoD becomes nearly constant with values between 0.71 and 0.91. Here the meta-model quality reaches its real maximum. Below the saturation threshold of $SCR = 12$ the quality of the meta-models is clearly

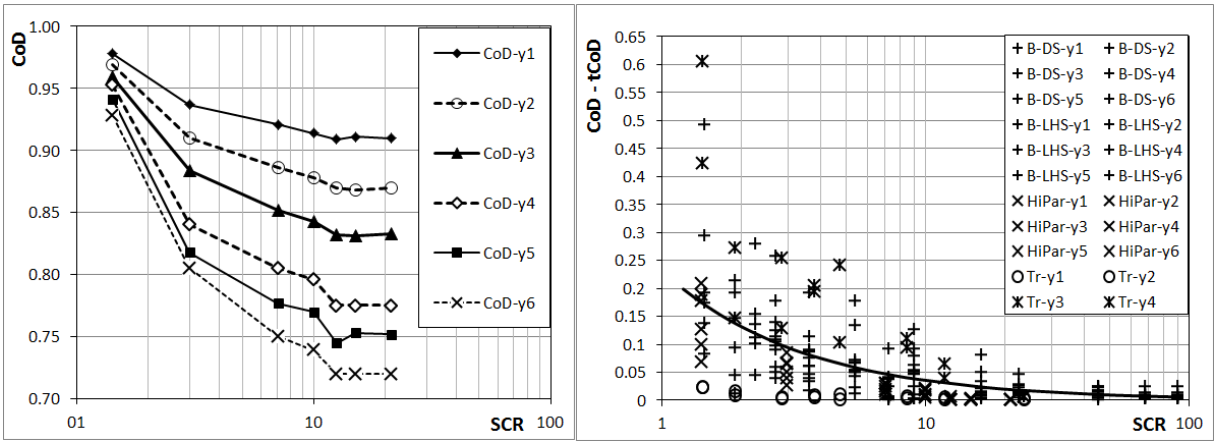

Figure 1: left: CoD of the six HiPar-response surfaces as function of the sample to coefficient ratio (SCR), right: Difference between CoD and target-CoD ($tCoD$) as function of the SCR for all tested meta-models

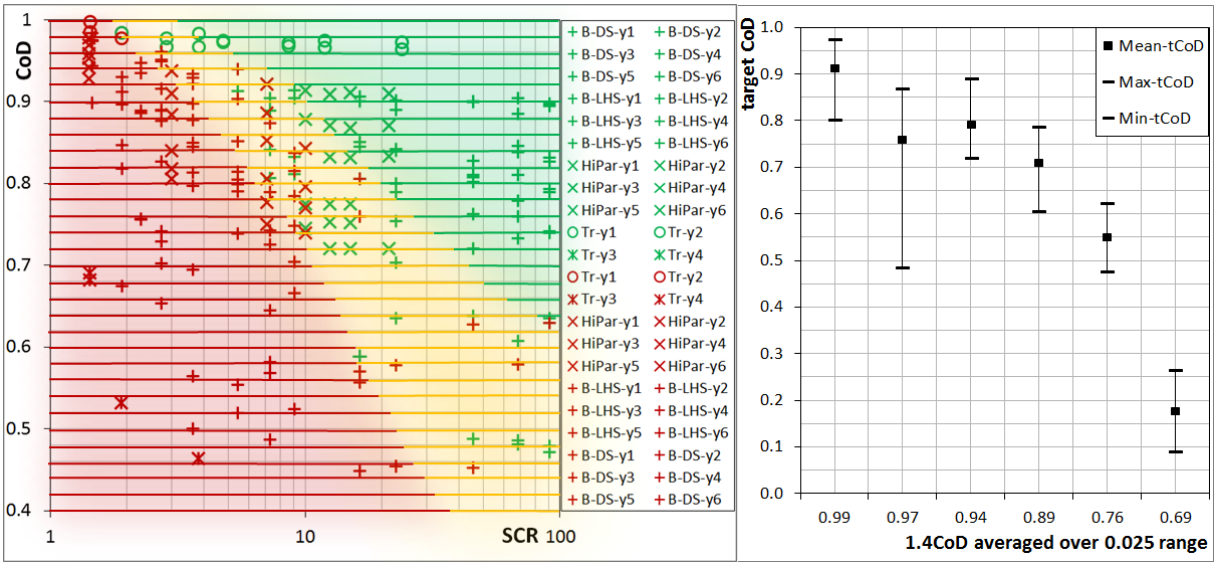

Figure 2: left: CoD as function of SCR, green symbols are above and red symbols are below the saturation threshold, right: $tCoD$ ranges as function of the $1.4CoD$ revealed by the used 22 test-models

overestimated by the CoD.

To verify this connection between the CoD saturation and the SCR, some more test-models should be used. The right graph in Figure 1 shows the absolute difference of the CoD to the target-CoD ($tCoD$), which is the average of the CoD at the two highest SCR-values. The shown data include all 22 tested meta-models.
For models with a very high $tCoD$ close to unity the deviation is small for the complete SCR-range. In this case the saturation starts at low SCR values of about 3.
The highest observed deviation values ($0.4 - 0.6$) for a SCR of 1.4 occur for meta-models with a low $tCoD < 0.5$.

All models with a $tCoD$ between 0.5 and 0.975 show a similar saturation behavior. The SCR-range of their saturation is between 7 and 22. The included trend line approximates the averages of all shown values.

Figure 2 gives the computed CoD depending on the SCR, while the color of the data points shows if the CoD is already saturated (green) or has still a difference to the $tCoD$ higher than 0.015. Most of the points (74%) achieve the saturation threshold in a SCR range between 7 and 22. Moreover there is the clear tendency, that models with a low $tCoD$ achieve the saturation at high SCR values. An explicit connection between the two variables can not be given due to the high observed scatter in the data and a quite small database. But the chart is divided into three areas, that can help the investigator to decide whether the computed CoD is reliable or not. In the red area we observed no saturated result. In the green area all computations are close or equal to the $tCoD$. In the yellow marked area we observed both, results above and below the saturation threshold.

The right graph of Figure 2 shows the connection between the CoD computed at an SCR of 1.4 ($1.4CoD$) and the $tCoD$. The data points include the $1.4CoD$ values averaged over a 0.025 wide range. For the first point this is for example the range between 0.975 and 1. The vertical axis represents the mean, min and max values of the corresponding $tCoD$ values. An interpretation of the shown data derived by 22 meta-models can be for example: If the $1.4CoD$ is 0.97, the saturated $tCoD$ can be between 0.48 and 0.87.

Generally spoken, it is important for the interpretation of any CoD value to know about the SCR as well. If the CoD should be useful as test criterion the SCR should be in or above the saturation range $(7 - 22)$.

3 Results of the Monte Carlo cross-validation

In this part we want to analyze the characteristics of the CoD_{MCCV} by varying the computation settings splitting ratio (SR), the number of samples (n) and the number of CV-runs for different meta-models. The splitting ratio has been varied between 0.5 and 0.995. The number of samples n has been varied between 200 and 1763 which corresponds to a SCR between 1.4 and 12.5. The number of CV-runs is varied between 20 and 2000, while each run includes the sample splitting, computation of a response surface using the training samples n_t and the computation of the single-run-CoD using n_v. The result CoD_{MCCV} is the average of all this CV-runs.

If the complete calculation with all its internal CV-runs is repeated, the resulting CoD_{MCCV} will not be exactly the same since each selection of n_v is randomly. But if the number of CV-runs is high enough the variance of the CoD_{MCCV} will be small. To estimate the variation of the results in chapter 3.2, each computation has been repeated 20 times with constant computation settings.
The data points shown in Figure 3 are averaged over the 20 repetitions. The splitting ratio is $SR = 0.975$. This leads to a number of validation samples $n_v = 5$ for $n = 200$.
The left graph compares the CoD with the CoD_{MCCV} for all calculated SCR values. The CoD_{MCCV} predicts a much lower response surface quality in particular if the SCR is low. For the six response surfaces of the HiPar-test-model there is no overestimation of the quality by the CoD_{MCCV}. For SCR values in the saturation range, CoD_{MCCV} and CoD

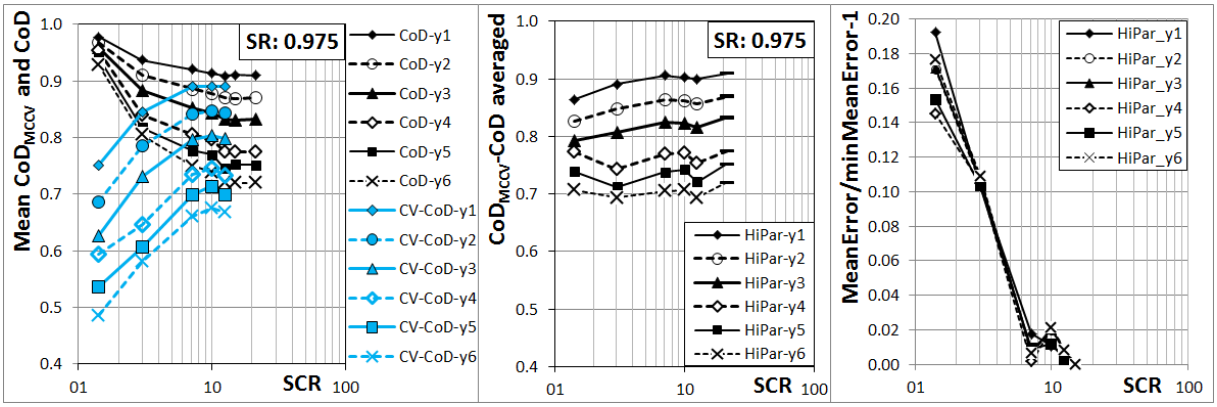

Figure 3: left: Comparison of CoD_{MCCV} that is averaged over 20 repetitions to the CoD results for the HiPar-test-model; middle: average of CoD_{MCCV} and CoD; right: mean error of response surface approximation normalized with minimum mean error tested with a complete independent sample set of $n = 3000$

converge into each other.
The improving response surface quality while increasing the SCR is shown by the right graph in Figure 3. Here we present the mean error computed with 3000 independent data points normalized with the minimum mean error that occurred. The graph shows, that the response surfaces at a $SCR = 1.4$ have about 14% to 20% higher errors than the best computed meta-models. The meta-models with minimum errors are reached at SCR values of 7 or higher.

The center graph in Figure 3 shows the average of the CoD_{MCCV} and the CoD up to an SCR of 12.5 - the last given values present the CoD. These average values are nearly constant over the complete SCR range, hence this criterion can be used as a rough estimator of the $tCoD$ that would be reached with a big database. The predictability of the $tCoD$ worked well for 91% of our test cases (20 out of 22 meta-models).

3.1 Influence of the splitting ratio

Figure 4 shows the variation of the splitting ratio for the complete test range. There is a positive correlation between the SR and the CoD_{MCCV} if the SCR is low, e.g. $SCR = 1.4$ as shown in the left graph. Here the low SR values 0.71 and 0.75 with many validation samples and the lowest possible n_t do not give useful results. The practical relevant SR-range for a low SCR is here between 0.9 and 0.975.

For a SCR of 3 the influence of the splitting ratio has decreased. Here the suggested range is between 0.9 and 0.975 as well.

Response surfaces with a large database are only marginal affected by a SR variation (Figure 4 right). The influence is only visible for an extremely high splitting ratio of 0.995, with 5 samples in the validation group. The useful SR-range is between 0.8 and 0.975.

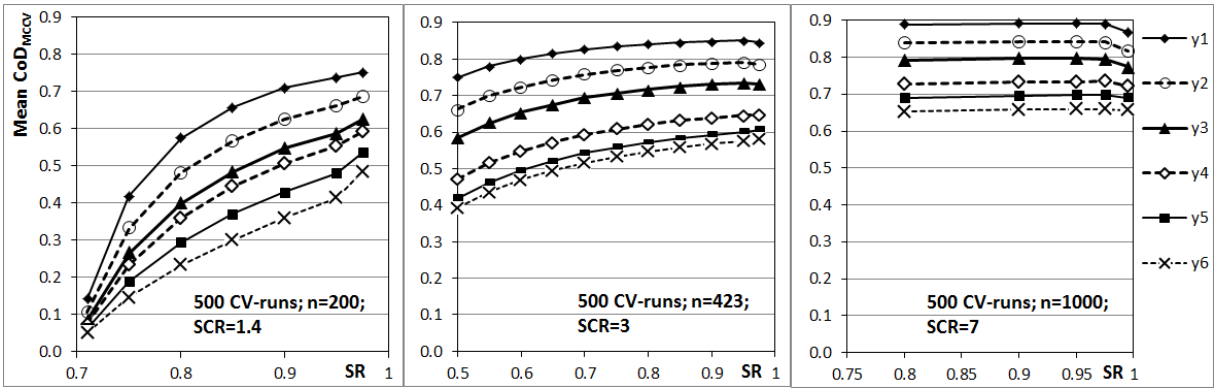

Figure 4: Influence of the splitting ratio (SR) on the CoD_{MCCV} that is averaged over 20 repetitions and shown for different data sizes

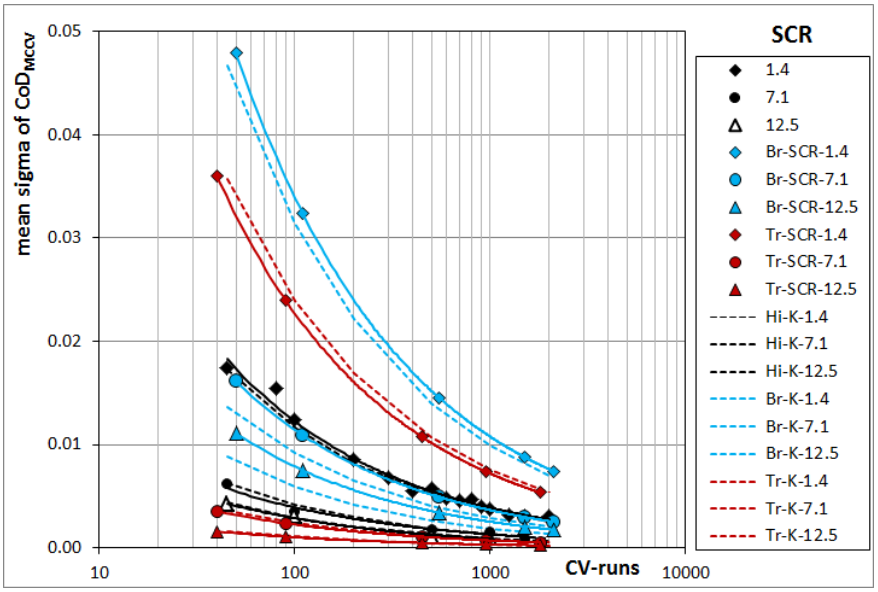

Figure 5: Variance of the MCCV results depending on model type, CV-runs and SCR

3.2 Result variation due to randomness

Due to the random selection of the test samples, the results of the CoD_{MCCV} will vary although the used data set is fixed for constant SCR. The amount of this variation depends on the number of CV-runs and on the splitting ratio.

For the used test-model the σ is strongly increasing for very high splitting ratios above 0.975. For lower SR values there is a low positive correlation between SR and σ. For the implementation of the MCCV-procedure in a probabilistic code it is important to predict the variation of the CoD_{MCCV} result value for a specific model type, to compute only as many CV-runs as necessary to reach a specific min-max-range or σ of the CoD_{MCCV}. Therefore the three different model types have been analyzed regarding their dependency of the variation expressed by sigma or the min-max-range respectively. Graph 5 shows this data with SCR as parameter. Symbols represent the results of the simulations (averaged

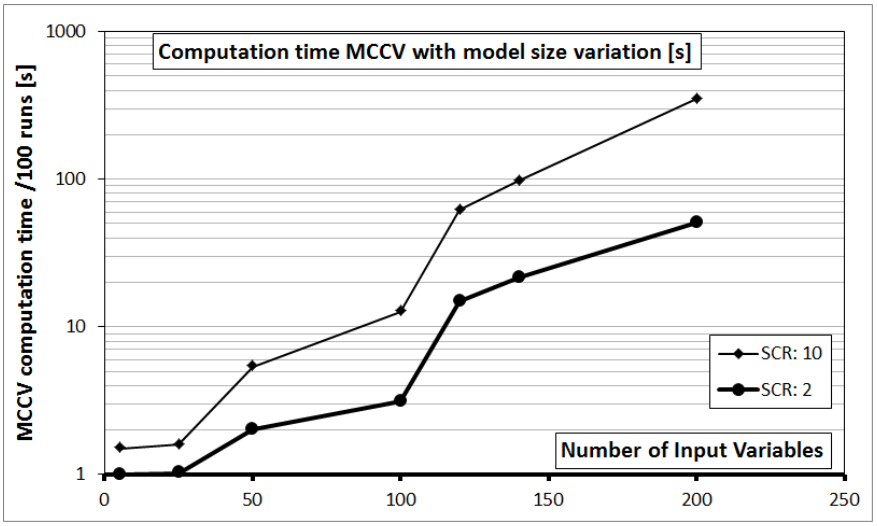

Figure 6: Computation time of the MCCV

over three relevant splitting rations and all the models of the given type) and solid lines are the trend-lines for the data-points. To estimate the necessary number of CV-runs we developed correlations

$$\text{range} = \frac{[-0.0081 \cdot C + 1.6114] \cdot SCR^{[1.5986 \cdot \frac{E \cdot O}{C} - 2.1763]}}{\sqrt{\text{CV-runs}}} \tag{9}$$

$$\sigma = \frac{[-0.002 \cdot C + 0.4265] \cdot SCR^{[1.5063 \cdot \frac{E \cdot O}{C} - 2.1283]}}{\sqrt{\text{CV-runs}}}. \tag{10}$$

The dashed lines in Figure 5 represents the approximation of σ (10).

3.3 Computation time

The computation time of the CoD_{MCCV} is influenced by the number of CV-runs and the model size. Figure 6 shows the measured time for the MCCV of a linear meta-model. The number of coefficients is equal to the number of input variables plus one. The shown data points are scaled to 100 CV-runs. The used machine is a ThinkPad® T61 with a 64bit OS, 4GB RAM and a Intel® Core™ 2 Duo T8100 at 2.1Ghz. There is plenty of room for improvements on hardware side and on the code optimization as well.
Figure 6 reveals, that there is a strong increase in computation time, when the model size exceeds 100 input parameters. Fortunately it was shown in chapter 3.2, that huge models need less CV-runs to achieve the same variance. Nevertheless the computation time will increase for models with a high input parameter dimension, since a minimum number of CV-runs will be necessary. This limit can be between 50 and 100.
Moreover the graph shows that there is a strong dependency on n, the number of samples in the database, which is here expressed by the SCR. For a model with 200 dimensions the computation needs 50s when 400 samples are used ($SCR = 2$) and 350s when 2000 samples are used ($SCR = 10$).

4 Conclusions and summary

The presented paper showed that the CoD_{MCCV} calculated by MCCV can give a more realistic quality estimation of the response surface than the CoD. This is in particular true for small sample to coefficient ratios $(3-8)$. Here we started with very small SCR-values of 1.4, where a strong overestimation of the meta-model quality by the CoD was visible. In the saturated range, where the CoD becomes constant while enlarging the database, the CoD_{MCCV} and the CoD converge into each other. Using the results 22 different meta models we developed a graph (Figure 2 left) that can be used to estimate the reliability of any computed CoD value.

The cross-validation results are influenced by the splitting ratio (SR) only for very low sample to coefficient ratios (SCR). In our test-model with 141 coefficients in the response surface equation, a high SR of 0.975 showed good results for all sample to coefficient ratios. For the small test-models with only five input dimensions, smaller splitting ratios $(0.6-0.9)$ work well.
For high SCR-values the influence of the SR is negligible.
The disadvantage of a high splitting ratio is the increasing scatter of the result values. The investigation showed that the σ, estimated by a twenty times repeated computation, is strongly increasing if the number of validation samples is below 10 in our HighParameter-test-model.

To reduce the standard deviation for a fixed SCR, the number of CV-runs can be increased. The necessary number of runs to achieve a target σ or min-max-range can be estimated with the developed correlations given in (9) and (10). But be aware that the equations are developed with the data of only three different test-model-types with all together 16 meta-models.
The reason to prefer low numbers of CV-runs, is the computation time, which is strongly increasing for high dimensional models. Fortunately these models showed lower variances.

A good estimation of the target-CoD that would be reached if the SCR is high (>7 to 22) can be found by averaging the CoD and the CoD_{MCCV}. The evaluation of 22 meta-models showed that for 91% of the tested models the difference to the target-value is small in the complete SCR-range. Hence this additional criterion can be used to predict the possible meta model improvement that can be achieved by enlarging the database.

An overestimation of the response surface quality by the CoD_{MCCV} can occur at very low SCR values (<3), combined with a low target-CoD (<0.5). This untypical CV-result occurred for three out of 22 meta-models. The observed effect must be investigated in more detail by using more test meta-models with a low quality. This was not the core of interest in the presented investigation.

Future work on this topic should reinforce the findings in this paper by using a greater variety of test-models. Moreover we will include an alternative validation sample selection algorithm. The K-fold cross-validation will deliver new data, that will be compared with the current results. Advantages and disadvantages of both procedures will be compared.

References

[1] Edward E Cureton. Validity, reliability and baloney. *Educational and Psychological Measurement*, 1950.

[2] Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. Regression, 2009.

[3] Seymour Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

[4] Alexander Lange, Matthias Voigt, Konrad Vogeler, Henner Schrapp, Erik Johann, and Volker Gümmer. Impact of manufacturing variability and nonaxisymmetry on high-pressure compressor stage performance. *Journal of Engineering for Gas Turbines and Power*, 134(3):032504, 2012.

[5] Marcin Molga and Czesław Smutnicki. Test functions for optimization needs. *Test functions for optimization needs*, 2005.

[6] Charles I Mosier. The need and means of cross-validation. i. problems and designs of cross-validation. *Educational and Psychological Measurement*, 1951.

[7] Raymond H. Myers and Douglas C. Montgomery. Response surface methodology / process and product optimization using designed experiments, 1995.

[8] Richard R Picard and R Dennis Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.

[9] Lothar Sachs. Angewandte statistik: Anwendung statistischer methoden, 2004.

[10] Jun Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88(422):pp. 486–494, 1993.

[11] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147, 1974.

[12] Qing-Song Xu and Yi-Zeng Liang. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11, 2001.

[13] Ping Zhang. Model selection via multifold cross validation. *The Annals of Statistics*, 21(1):pp. 299–313, 1993.