SFB
823

Discussion Paper

Comparison of classical and sequential design of experiments in note onset detection

Nadja Bauer, Julia Schiffner, Claus Weihs

Nr. 24/2011

Comparison of Classical and Sequential Design of Experiments in Note Onset Detection

Nadja Bauer, Julia Schiffner and Claus Weihs

Abstract Design of experiments is an established approach to parameter optimization of industrial processes. In many computer applications however it is usual to optimize the parameters via genetic algorithms. The main idea of this work is to apply design of experiment's techniques to the optimization of computer processes. The major problem here is finding a compromise between model validity and costs, which increase with the number of experiments. The second relevant problem is choosing an appropriate model, which describes the relationship between parameters and target values. One of the recent approaches here is model combination, which can be used in sequential designs in order to improve automatic prediction of the next trial point.

In this paper a musical note onset detection algorithm will be optimized using sequential parameter optimization with model combination. It will be shown that parameter optimization via design of experiments leads to better values of the target variable than usual parameter optimization via grid search or genetic optimization algorithms. Furthermore, the results of this application study reveal, whether the combination of many models brings improvements in finding the optimal parameter setting.

1 Introduction

Parameter optimization is an important issue in almost every industrial process or computer application. In general, there are one or more target variables to be optimized, which depend on a parameter vector. The relationship between target variables and parameters is usually unknown.

Nadja Bauer, Julia Schiffner and Claus Weihs,
Chair of Computational Statistics, Faculty of Statistics, TU Dortmund,
e-mail: {bauer,schiffner,weihs}@statistik.tu-dortmund.de

Due to the high costs of most real experiments, they are often replaced by appropriate simulations if possible. There are different strategies for parameter optimization: grid search, algorithms of non-linear parameter optimization (like evolutionary search techniques, simulated annealing etc.) or design of experiments. With an increasing number of parameters and an increasing function evaluation time it becomes infeasible to optimize the target variables in an acceptable period of time. Design of experiments allows to gain as much information as possible with minimal effort and therefore helps to tackle this problem in the most effective way.

There are two types of experimental designs: classical und sequential. In classical designs all trial points are fixed in advance. In sequential designs the next trial point or the decision for stopping the experiment depends on the results of previous experiments. The main challenge in design of experiments is the choice of the model, which should describe the relationship between the target variable und the parameter vector. One promising approach to cope with this problem is model combination. The aim is to find the combination which is at least as good as the best single model. In this paper we introduce and test different model combination strategies and compare them to the single models. Furthermore we will assess the influence of experimental design types on the evaluation results.

The next section provides a short overview of sequential parameter optimization and model combination strategies. Moreover, our experimental design types and model combination approaches are presented. The application problem, musical note onset detection, is discussed in section 3 and the simulation results are presented in section 4. Finally section 5 summarizes our work and provides points for future research.

2 Background and research proposal

This section provides a survey of sequential parameter optimization and model combination. For each topic we first introduce related work and common approaches and then describe our proposal.

2.1 Sequential parameter optimization

Related work

We assume that a non-linear, multimodal black-box function $f(\boldsymbol{P})$ of k numeric or integer parameters $\boldsymbol{P} = (P_1, P_2, \dots, P_k)$ is to optimize. The range of allowed values for parameter P_i is given by V_i. Let $\boldsymbol{V} = V_1 \times V_2 \times \dots \times V_k$ define the parameter space. A trial point $\tilde{\boldsymbol{P}}$ is indicated by a parameter setting $(\tilde{P}_1, \tilde{P}_2, \dots, \tilde{P}_k)$. An experimental design is a scheme that prescribes in which order the trial points are to evaluate. In case of the classical approach this scheme depends on the assumed relationship between $f(\boldsymbol{P})$ and $\boldsymbol{P}$ (i.e. the model type) and a chosen optimization criterion (like A, D-optimality, [2]) and a-priori specifies all trial points of the whole experiment. In case of a sequential approach only the initial design, whose dimension is usually much smaller than the total number of trials, has to be given in advance. The common procedure of sequential parameter optimization is as follows.

1. Let D denote the initial experimental design with $N_{initial}$ trial points and let $Y = f(D)$ be the set of function values of points in D.
2. Do the following sequential step as long as the termination criterion is not fulfilled:
   2.1 Fit the model M with response Y and design matrix D;
   2.2 Find the next trial point d_{next} to optimize the model prediction;
   2.3 Evaluate $y_{next} = f(d_{next})$ and update $D \longleftarrow (D \cup d_{next})$, $Y \longleftarrow (Y \cup y_{next})$.
3. Return the optimal value of the target variable $y_{best} \in Y$ and the associated parameter setting $d_{best} \in D$.

Usually the trial points for the initial design in step 1 are determined via Latin Hypercube Sampling (LHS, [21]), which covers the interesting parameter space $\boldsymbol{V}$ uniformly. Note, that M can be a single model but also a model ensemble (see section 2.2). The major differences between the existing algorithms for sequential parameter optimization lie in steps 2.1 and 2.2. One popular approach here is a response surface methodology proposed by Jones et al. [10]: in step 2.1 a Kriging model [12] is fitted and the next trial point in step 2.2 is chosen by maximizing the expected improvement criterion. Expected improvement can be calculated for each point $\tilde{\boldsymbol{P}} \in \boldsymbol{V}$ by consideration of two criteria: the value of model prediction and the model uncertainty at this point. For more details see [10].

Another approach is given by Bartz-Beielstein et al.[3]: in step 2.1 a user-chosen model is fitted and the optimization of the model prediction in step 2.2 is done by means of grid-search, which is realized by an LHS Design D' with N_{step} points. Note that this optimization is not time-consuming because it requires merely prediction of the model M in D' but not evaluation of the function $f(D')$. Therefore N_{step} can be set to values larger than 100.000. Another way to optimize the model prediction's value is to use an appropriate optimization algorithm like a genetic algorithm. However, for some model types (like a classical linear regression model) the optimal prediction can be found theoretically. There are several termination criteria to use in step 2.4: reaching the global optimum of f (if known), limitation of the function evaluations' number, time limitation or no improvement.

Proposal

For our research proposal we will use the above introduced procedure. Note that depending on the settings of the sequential parameter optimization algorithm different kinds of experimental designs can be used. Two of the most important issues here are the initial design and the number of sequential steps. We will propose and test three algorithm settings, where the number of the influence parameters is assumed to be three.

The first setting is a classical 3^3 factorial design with an additional inner "star" ([23], p. 250). Table 1 gives the experimental scheme for this initial design, where the values for the variables $X1$, $X2$ and $X3$ are bounded by -1 and +1 (so called extreme values). The number of trial points here is 33 and just one further evaluation will be done according to the best model prediction (verification step). This design will be called **Classic** in following. Note that in this case it is not a sequential design but a commonly used classical approach. The total number of evaluations is 34. This number should not be exceeded by all further designs in order to facilitate the comparability between them.

Table 1 3^3 full factorial design plus inner "star"

$X1$	$X2$	$X3$	$X1$	$X2$	$X3$	$X1$	$X2$	$X3$	$X1$	$X2$	$X3$
1	1	1	1	0	1	1	-1	1	0	0	0.85
0	1	1	0	0	1	0	-1	1	0	0	-0.85
-1	1	1	-1	0	1	-1	-1	1	0	0.85	0
-1	1	0	-1	0	0	-1	-1	0	0	-0.85	0
-1	1	-1	-1	0	-1	-1	-1	-1	0.85	0	0
0	1	-1	0	0	-1	0	-1	-1	-0.85	0	0
1	1	-1	1	0	-1	1	-1	-1			
1	1	0	1	0	0	1	-1	0			
0	1	0	0	0	0	0	-1	0			

The second design (**SeqICC**) is given by an inscribed central composite initial design with 15 trial points ([23], p. 151) and 19 sequential steps (see table 2). The third design (**SeqLHS**) is an LHS initial design with also 15 trial points and 19 sequential steps. The LHS initial design is commonly used in sequential parameter optimization of computer applications, while the central composite design is often applied to optimization of industrial processes. We employ both in order to assess which leads to better results.

Table 2 Inscribed central composite initial design

X1	*X2*	*X3*	*X1*	*X2*	*X3*
1	1	1	0	0	0.85
1	1	-1	0	0	-0.85
1	-1	1	0	0.85	0
1	-1	-1	0	-0.85	0
-1	1	1	0.85	0	0
-1	1	-1	-0.85	0	0
-1	-1	1	0	0	0
-1	-1	-1			

The sequential procedure terminates if the maximum number of function evaluations, here 34, is reached. For steps 2.1 and 2.2 the above presented approach of Bartz-Beielstein et al. is used with $N_{step} = 20.000$ trial points. Section 4 provides a short review of these three parameter optimization strategies. Different settings for step 2.1, the choice of a model M, will be discussed below.

2.2 Model combination

Related work

Many statistical application problems are classification or regression problems. Here, the aim is to learn the relationship between target variables and influence variables. In most cases it is not obvious which modelling approach should be used so that several model types are fitted to find the best one according to a declared accuracy criterion. The main idea behind model combination is to construct a combined model that yields better prediction accuracy than any single model.

Usually one of the following two problems is tackled: building an ensemble of one particular model type with different hyperparameter settings [20] or building an ensemble of heterogeneous models with fixed hyperparameter settings [18, 13, 24]. Hyperparameters are parameters of model or learner.

Popular approaches to model combination are Bagging and Boosting, which are based on re-sampling techniques: several training data sets are obtained from the given training data in order to fit models with different hyperparameter settings. These different models are then combined by using weighted voting for getting a classifier decision. Advantage of these ensemble-methods is that they provide better classification results then the single classifiers, but it is difficult to interpret these models [24].

A more challenging problem is developing an approach which handles different model types and optimizes their hyperparameter settings automatically. Such algorithms are proposed for example by [8] and [9]. [9] uses a so called island-model. The main idea of this algorithm is as follows: in the first step each island is inhabited just by a species of one particular model type with different hyperparameter settings. In the next steps the population in each island develops according to an evolutionary algorithm and migration between the islands is allowed so that it comes to model ensembles by crossing the species of different model types.

An important issue for this work is the combination of different model types. A good overview about some related approaches is given by [19]. One of the most popular model combination methods is linear combination: a joint prediction for a particular trial point is obtained as simple or weighted average of the individual predictions. Model outputs could be weighted for example according to an accuracy criterion like the goodness of fit or prediction accuracy. Other approaches are the Dempster-Shafer belief-based method, supra Bayesian, stacked generalization etc.

[9] use merely the simple average approach for their algorithm in order to keep its complexity low.

Proposal

The aim of this work is not only to compare different designs for sequential parameter optimization but also to test different model combination possibilities. Let us assume that m models (learners) $M_1, M_2, \dots, M_m$ are given with Y as a response and a design matrix D, which includes the settings of the influential parameters. Let us further assume that we have a minimization problem: the minimum of Y is sought. For each model compute first a model prediction accuracy criterion $M_1^{acc}, M_2^{acc}, \dots, M_m^{acc}$. We will use here the leave-one-out mean squared error estimator. Then calculate model predictions for each point d_j, $j = 1, \dots, N_{step}$, of the sequential design D' and receive for each model a prediction vector of length N_{step}: $M_1^{pred}, M_2^{pred}, \dots, M_m^{pred}$. We will assume that the smaller model prediction accuracy criterion is the better is the associated model. As first model combination method we use the weighted average approach. In order to calculate the weights the model prediction accuracies are linearly rescaled into the interval from 1 to 2 where 1 corresponds to the worst model and 2 corresponds to the best model. In this way we get a vector: $scaled(M_1^{acc}), scaled(M_2^{acc}), \dots, scaled(M_m^{acc})$.

The weighted average (**WeightAver**) then is defined as follows:

$$\boldsymbol{WeightedAverage}(d_j) = \sum_{i=1}^{m} \frac{M_i^{pred}(d_j)}{scaled(M_i^{acc})}, j = 1, \dots, N_{step}.$$

In each sequential step the next evaluation is done in that point d_j, which minimizes the ***WeightedAverage*** function.

For the second combination approach (**BestModel**) we will just choose the best model according to the model prediction accuracy criterion. Then the function f is evaluated in that point d_j with the minimal model prediction value.

The third combination method (**Best2Models**) is similar to the second method but in each step we evaluate two points according to the predictions of the two best models. That does not mean however that we will do more function evaluations than allowed (see the termination criterion in section 2.1).

The last model combination approach is based on the ten best prediction points of each model (**Best10**). In the following this approach will be called best ten points approach. First, for each model M_i the best ten predicted values $best10(M_i)$ are collected into a vector

$$\boldsymbol{Best} = (best10(M_1), best10(M_2), \dots, best10(M_m))$$

of dimension $10 \cdot m$. The vector ***ScaledBest*** is obtained by rescaling ***Best*** into the interval from 1 to 2 (2 corresponds to the biggest value of ***Best*** and 1 to the smallest). The vector ***ModelWeight*** is defined as

$$\boldsymbol{ModelWeight} = (\underbrace{scaled(M_1^{acc})}_{\times 10}, \underbrace{scaled(M_2^{acc})}_{\times 10}, \ldots, \underbrace{scaled(M_m^{acc})}_{\times 10}).$$

For each entry in ***Best*** its relative frequency in this vector is assessed and collected into the vector ***FrequencyWeight***. This is done because it could happen that many trial points belong to the best ten predictions of several model types and this makes them more influential. The final score for the best $10 \cdot m$ points is given by

$$\boldsymbol{Score}_l = \frac{\boldsymbol{ScaledBest}_l}{\boldsymbol{ModelWeight}_l} - \boldsymbol{FrequencyWeight}_l, l = 1, 2, \ldots, 10 \cdot m.$$

The next trial point for the function evaluation is that with the minimal score.

3 Application to a musical note onset detection algorithm

We will use sequential parameter optimization to find the optimal parameter set for an onset-detection algorithm. A tone onset is the time point of the beginning of a musical note or another sound. Onset detection is an important step for music transcription and other applications like timbre or meter analysis. A tutorial on onset detection is given by [5]. Here, we do not want to propose a very good approach for onset detection but to optimize effectively their algorithm parameters.

The suggested method is based on the assumption that the tone onset is marked by an amplitude increase. This assumption is fulfilled especially well for stringed instruments like piano.

The signal is analyzed merely on a low level: only the amplitude variations of the audio signal will be considered. Figure 1 shows an example of a music audio signal (the onsets are marked as vertical lines). The ongoing audio signal will be split up into windows of length $\boldsymbol{L}$ samples with overlap $\boldsymbol{U}$ samples. For each window the maximum of the absolute amplitude is calculated. An onset is detected in each window where the absolute amplitude maximum of this respective window is at least $\boldsymbol{S}$ times as large as the maximum of the previous window (see [4]). Formally this model can be written down as follows:

$$O_T(\boldsymbol{L},\boldsymbol{U}) = \overbrace{z\Big(\max(\mid x_t \mid)\Big|_{t=(T-1)\cdot(\boldsymbol{L}-\boldsymbol{U})+1}^{T\cdot\boldsymbol{L}-(T-1)\cdot\boldsymbol{U}} - \boldsymbol{S}\cdot\max(\mid x_t \mid)\Big|_{t=(T-2)\cdot(\boldsymbol{L}-\boldsymbol{U})+1}^{(T-1)\cdot\boldsymbol{L}-(T-2)\cdot\boldsymbol{U}}\Big)}^{\hat{O}_T(\boldsymbol{L},\boldsymbol{U},\boldsymbol{S})}$$

$$+e_T(\boldsymbol{L},\boldsymbol{U},\boldsymbol{S}), \text{with}$$

- N - sample length of the ongoing signal,
- $t = 1, \ldots, N$ - sample index,
- x_t - amplitude of the ongoing signal in the tth sample,
- $T = 1, \ldots, \lfloor \frac{N}{\boldsymbol{L}-\boldsymbol{U}} \rfloor$ - window index,

- $O_T(\boldsymbol{L},\boldsymbol{U})$ - vector of true onsets: $+1$, if onset in Tth window, 0, else,
- $\hat{O}_T(\boldsymbol{L},\boldsymbol{U},\boldsymbol{S})$ - vector of estimated onsets: $+1$, if onset in Tth window, 0, else,
- $O_1(\boldsymbol{L},\boldsymbol{U}) = 0$ (assumption),
- $z(x)$: $+1$, if $x > 0$, 0, else,
- $e_T(\boldsymbol{L},\boldsymbol{U},\boldsymbol{S})$ - model error,
- parameters to optimize:
  - $\boldsymbol{L}$ - window length (in samples),
  - $\boldsymbol{U}$ - overlap (in samples),
  - $\boldsymbol{S}$ - threshold.

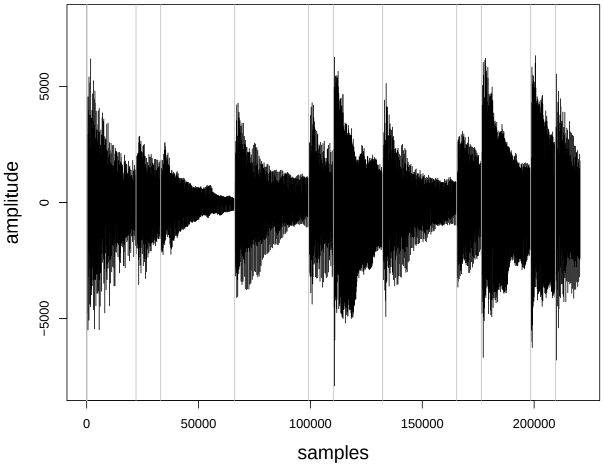

Fig. 1 Example of an audio signal. Vertical lines are the tone onsets

To illustrate this model we provide an example: Let us assume $N = 1000$, $\boldsymbol{L} = 200$ and $\boldsymbol{U} = 50$ (this means overlap is 25%). The ongoing signal is split up into 6 windows: window 1: $t \in [1,200]$, window 2: $t \in [151,350]$, window 3: $t \in [301,500]$, window 4: $t \in [451,650]$, window 5: $t \in [601,800]$ and window 6: $t \in [751,950]$.

Let us assume that the true onset occurs at $t = 480$ samples. In this case we can write the vector of true onsets as $O_T(200,50) = (0,0,1,1,0,0)'$. Due to the overlap the onset occurs in the third as well as in the fourth window. In such cases an onset is said to be detected if it is found in at least one of the two windows. O_T is regarded as a function of parameters $\boldsymbol{L}$ and $\boldsymbol{U}$ in order to clarify that its length and assignment varies with different settings of this parameter.

One of the most popular quality criteria in onset detection is the so called F-value

$$F = \frac{2c}{2c + f^+ + f^-},$$

where c is the number of correctly detected onsets, f^+ is the number of false detections and f^- represents the number of undetected onsets. Note that the F-value lies always between 0 and 1 [7]. The optimal F-value is 1.

To complete our example, let us assume that the onset vector estimated by the proposed algorithm with parameter setting $\boldsymbol{L} = 200$, $\boldsymbol{U} = 50$ and $\boldsymbol{S} = 2$ would be given as $\hat{O}(200,50,2) = (1,0,1,0,0,0)'$. Then we had $c = 1, f^+ = 1$, $f^- = 0$ and accordingly $F = \frac{2}{3} = 0.67$.

Note that the optimal parameter setting could vary depending on e.g. music tempo, number of instruments or sound volume of an audio signal. Another important factor is whether there is a synthesized audio signal or a real piano recording. To take into account some of these points we decided to differentiate between six musical epochs where each epoch is represented by two famous European composers with one music piece respectively. The epochs and composers with the corresponding abbreviations are given below:

- Medieval (Perotin[1]: PER, Adam de la Halle[2]: HAL),
- Renaissance (Orlando di Lasso[3]: DIL, Hans Leo Hassler[4]: HAS),
- Baroque (Claudio Monteverdi[5]: MON, Heinrich Schuetz[6]: STZ),
- Classic (Wolfgang A. Mozart[7]: MOZ , Franz J. Haydn[8]: HAY),
- Romance (Frédéric Chopin[9]: CHO, Robert Schumann[10]: SMN),
- New music (Arnold Schoenberg[11]: SBG, Igor Strawinski[12]: STR).

Music pieces were downloaded as MIDI-data from different internet archives given in the footnotes. MIDI-data contains all information about the recording and particularly the note onset times. As we have noticed above the proposed onset algorithm suits just for stringed instruments. For this reason the instruments of all music tracks were set to piano using the software Anvil-Studio[13]. After that the

[1] http://www.hypermusic.ca/comp/leonin.html, date: 01.07.2011.

[2] *Or est Bayard en la pature, Hure!*, http://www.midiworld.com/earlymus.html, date: 01.07.2011.

[3] *Sibylla Persica*, http://www.kunstderfuge.com/lasso.htm, date: 01.07.2011.

[4] *Ach Weh des Leiden*, http://www.kunstderfuge.com/hassler.htm, date: 01.07.2011.

[5] *Crudel! perché mi fuggi?*, http://www.kunstderfuge.com/monteverdi.htm, date: 01.07.2011.

[6] *Eile mich, Gott, zu erretten*, http://www.kunstderfuge.com/schutz.htm, date: 01.07.2011.

[7] *Sonata No. 1 in C major, KV 279 [E 189d] (1774)*, http://www.kunstderfuge.com/mozart.htm, date: 01.07.2011.

[8] *Sonata No. 1 in C major, KV 279 [E 189d] (1774)*, http://www.kunstderfuge.com/haydn.htm, date: 01.07.2011.

[9] *Sonata No. 2 in b flat minor, Op. 35*, http://www.kunstderfuge.com/chopin.htm, date: 01.07.2011.

[10] *Sonata for violin and piano in a minor, Op. 105 (1851)*, http://www.kunstderfuge.com/schumann.htm, date: 01.07.2011.

[11] *Sechs kleine Klavierstuecke, Op. 19, 3*, http://www.kunstderfuge.com/schonberg.htm, date: 01.07.2011.

[12] *Symphony of Psalms First movement: Prelude*, http://www.cco.caltech.edu/ tan/Stravinsky/download.html, date: 01.07.2011.

[13] http://www.anvilstudio.com (Version 2009.06.06), date: 01.07.2011.

true onset times were extracted using the Matlab MIDI-Toolbox[14], and then the MIDI-files were converted to WAV-files using the freely available software MIDI to WAVE Converter 6.1[15]. We converted (and then used) just the first 60 seconds of each music piece. Unfortunately in some cases either the onset times were extracted not correctly or the converter software has failed so that the onset times and the recording did not match (e.g. this was the case for several L. v. Beethoven pieces: for that reason L. v. Beethoveen was replaced by F. J. Haydn). Therefore we had to control each piece using plots like figure 1.

In design of experiments it is essential to define the region of interest for each parameter, i.e. its lower and upper boundaries. For the parameter $\boldsymbol{L}$ we will allow just the following powers of two: 256, 512, 1024, 2048, 4096 and 8192. The region of interest for $\boldsymbol{U}$ is given by an interval between 0% (no overlap) and 50%. Note that just 1% steps are allowed. The lowest possible value for $\boldsymbol{S}$ is 1.1 and the largerst 5.1 with step size 0.01.

In following we will model the relationship between the onset detection algorithm parameters ($\boldsymbol{L}$, $\boldsymbol{U}$ and $\boldsymbol{S}$) and the target variable (F-value). We actually have a maximization problem here, the sign of F will be reversed hence to get a minimization problem.

4 Results

To compare different options for the sequential parameter optimization algorithm (see section 2.1) we generate an experimental scheme with three metaparameters: *model type*, *model combination type* and *design*. The word metaparameter is used in order to differentiate between the onset algorithm parameters which have to be optimized and the parameters of the sequential parameter optimization algorithm. The metaparameter *model type* determines the model which describes the relationship between the onset detection algorithm parameters and the target variable (see section 3). We test 6 model types: a full second order model, Kriging, random forest, support vector machines, neural network and the combination of these five model types. In this work we do not aim to find the optimal hyperparameter settings for the models, so we just use the default hyperparameter setting of the corresponding R functions (the calculation was done using the R package `mlr` [6]). The second metaparameter – *model combination type* – is just meaningful for the sixth model type and has four options: weighted average, best model, best two models and best ten points. All these approaches are described in section 2.2. The last parameter - *design* - is related to the experimental design with three possibilities: classical design, sequential design with inscribed central composite initial design and sequential design with LHS initial design (see section 2.1). The nomenclature for the metaparameters is given below:

[14] https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/miditoolbox/, 01.07.2011.

[15] http://www.heise.de/software/download/midi_ to_ wav_ converter/53703, date: 01.07.2011.

- *model type*:
  - **FSOM**: Full Second Order Model (R-package `rsm` [14]),
  - **KM**: Kriging (R-package `DiceKriging` [16]),
  - **RF**: Random Forest (R-package `randomForest` [15]),
  - **SVM**: Support Vector Machines (R-package `ksvm` [11]),
  - **NN**: Neural Network (R-package `nnet` [17]),
  - **COMB**: Combination of the upper five models,
- *model combination type*:
  - **WeightAver**: weighted average approach,
  - **BestModel**: best model approach,
  - **Best2Models**: best two models approach,
  - **Best10**: best ten points approach,
- *design*:
  - **Classic**: 3^3 factorial design plus inner "star" as initial design and one verification trial,
  - **SeqICC**: inscribed central composite initial design and 19 sequential steps,
  - **SeqLHS**: LHS initial design and 19 sequential steps.

Moreover three conventional parameter optimization approaches in the field of signal analysis are conducted: grid search and two genetic algorithms. By the grid search an LHS design with 34 points is used and the two genetic algorithms are Differential Evolution Optimization (DEO) from R-package `DEoptim` [1][16] and Covariance Matrix Adapting Evolutionary Strategy (CMAES) from R-package `cmaes` [22][17]. For both evolutionary algorithms 35 function evaluations are allowed. Note that for each of the 30 proposed optimization strategies and for each music piece the evaluation is carried out ten times. This is done in order to average out the influence of chance on the outcome. The mean of the corresponding optimal values is reported in tables 3 and 4.

Table 3 presents the results for Medieval, Renaissance and Baroque and table 4 shows the results for Classic, Romance and New Music. In each column the three best values are given in bold. The counts in brackets under the composers' abbreviations provide the number of unequal onsets in a corresponding music piece (the length of all audio signals is 60 sec.). First it is to note that depending on the tempo of a music piece the F-value of the proposed onset algorithm varies considerably. Good onset detection rates are reached for Perotin (PER) and Monteverdi (MON) and worst rates for Mozart (MOZ), Chopin (CHO) and Strawinski (STR). The correlation coefficient between the number of onsets and the best reached F-value for the given 12 signals is -0.89. This illustrates that the used onset algorithm (see section 3) is suited just for rather slow pieces.

[16] Control parameters: `NP = 5, itermax = 6`

[17] Control parameters: `sigma=0.25, maxit=5`

Table 3 Simulation results for Medieval, Renaissance and Baroque

ID	model type	model combination type	design	PER (130)	HAL (233)	HAS (106)	DIL (120)	MON (82)	STZ (154)
1	**FSOM**	-	**Classic**	0.8533	0.6688	0.8161	0.7773	**0.9427**	0.7833
2	**KM**	-	**Classic**	0.8533	0.6564	0.8161	0.7732	**0.9427**	0.7809
3	**RF**	-	**Classic**	0.8533	0.6556	0.8161	0.7716	**0.9427**	0.7809
4	**SVM**	-	**Classic**	0.8968	0.6569	0.8161	0.7716	**0.9427**	0.8339
5	**NN**	-	**Classic**	0.8579	0.6614	0.8161	0.7805	**0.9427**	0.7834
6	**FSOM**	-	**SeqICC**	0.8597	0.6790	0.8189	0.8179	0.9382	0.7986
7	**KM**	-	**SeqICC**	0.9042	**0.6891**	**0.8227**	0.8187	**0.9388**	0.8571
8	**RF**	-	**SeqICC**	0.8533	0.6741	0.8097	0.7717	0.9317	0.7809
9	**SVM**	-	**SeqICC**	0.8934	0.6817	**0.8222**	**0.8235**	0.9342	0.8323
10	**NN**	-	**SeqICC**	**0.9199**	0.6811	0.8162	0.8148	0.9405	**0.8673**
11	**FSOM**	-	**seqLHS**	0.9165	0.6745	0.8102	0.8136	0.9300	0.8646
12	**KM**	-	**seqLHS**	0.8694	**0.6893**	0.8186	0.8060	0.9170	0.8495
13	**RF**	-	**seqLHS**	0.8078	0.6678	0.8076	0.7721	0.8718	0.7103
14	**SVM**	-	**seqLHS**	0.8918	0.6728	0.8066	0.8060	0.8935	0.8349
15	**NN**	-	**seqLHS**	0.9165	0.6798	0.8176	0.8044	0.9391	0.8535
16	**COMB**	**WeightAver**	**Classic**	0.8533	0.6593	0.8161	0.7783	**0.9427**	0.7842
17	**COMB**	**BestModel**	**Classic**	0.8593	0.6556	0.8161	0.7806	**0.9427**	0.7809
18	**COMB**	**Best2Models**	**Classic**	0.8533	0.6564	0.8161	0.7716	**0.9427**	0.7809
19	**COMB**	**Best10**	**Classic**	0.8533	0.6564	0.8161	0.7829	**0.9427**	0.7809
20	**COMB**	**WeightAver**	**SeqICC**	0.8892	0.6808	0.8208	**0.8219**	0.9334	0.8031
21	**COMB**	**BestModel**	**SeqICC**	0.9168	**0.6874**	0.8180	0.8113	0.9380	0.8663
22	**COMB**	**Best2Models**	**SeqICC**	**0.9180**	0.6829	0.8201	0.8190	**0.9389**	**0.8665**
23	**COMB**	**Best10**	**SeqICC**	**0.9196**	0.6840	**0.8217**	**0.8223**	0.9353	**0.8661**
24	**COMB**	**WeightAver**	**seqLHS**	0.9016	0.6773	0.8189	0.8104	0.9211	0.8597
25	**COMB**	**BestModel**	**seqLHS**	0.8879	0.6792	0.8053	0.8041	0.9054	0.8352
26	**COMB**	**Best2Models**	**seqLHS**	0.9111	0.6766	0.8177	0.8140	0.9319	0.8582
27	**COMB**	**Best10**	**seqLHS**	0.9177	0.6830	0.8198	0.8129	0.9375	0.8636
28		LHS		0.8183	0.6542	0.8005	0.7808	0.8827	0.7669
29		DEO		0.7909	0.6072	0.8051	0.7722	0.8596	0.7802
30		CMAES		0.6815	0.5581	0.7069	0.6693	0.7734	0.5553

The three conventional parameter optimization approaches (ID's 28, 29 and 30) provide obviously worse results than the proposed sequential parameter optimization strategies (especially according to table 4). This is probably due to the fact that evolutionary algorithms require many more function evaluations to perform properly than only the allowed 34 .

A further interesting fact is that for the audio signal MON the best F-value is reached in a trial point from the classical design scheme ($X1 = -1$, $X2 = 0$ and $X3 = 1$). For this reason the best value 0.9427 is attained by 9 optimization methods with classical design.

For better interpretation of the study results table 5 shows a summary of tables 3 and 4: for each optimization method the amount of the best, the second best and the third best placements over all music pieces is given. As can be seen, the classical parameter optimization approach with 33 fixed trial points and one verification

Table 4 Simulation results for Classic, Romance and New Music

ID	model type	model combination type	design	MOZ (638)	HAY (435)	CHO (483)	SMN (475)	STR (479)	SBG (159)
1	**FSOM**	-	**Classic**	0.3366	0.5298	0.4007	0.4574	0.3525	0.5637
2	**KM**	-	**Classic**	0.3364	0.6627	0.4000	0.6005	0.3648	0.5521
3	**RF**	-	**Classic**	0.3366	0.5214	0.4000	0.5429	0.3481	0.5455
4	**SVM**	-	**Classic**	0.3656	0.6144	0.4169	0.5202	0.4140	0.6347
5	**NN**	-	**Classic**	0.3364	0.6286	0.4000	0.6111	0.3505	0.5580
6	**FSOM**	-	**SeqICC**	0.3471	0.6586	0.4040	0.6142	0.3674	**0.6633**
7	**KM**	-	**SeqICC**	**0.4161**	0.6658	**0.4368**	0.5759	**0.4588**	0.6621
8	**RF**	-	**SeqICC**	0.3541	0.5086	0.4000	0.3804	0.3624	0.5642
9	**SVM**	-	**SeqICC**	0.3383	0.6231	0.4000	0.5378	0.3769	0.6572
10	**NN**	-	**SeqICC**	0.3384	**0.7070**	0.4069	**0.6385**	0.4480	**0.6641**
11	**FSOM**	-	**seqLHS**	0.3282	0.6735	0.4099	0.5692	0.4171	0.6359
12	**KM**	-	**seqLHS**	0.3541	0.6613	0.3785	0.6140	0.4190	0.6350
13	**RF**	-	**seqLHS**	0.2539	0.5397	0.3644	0.4641	0.3274	0.5984
14	**SVM**	-	**seqLHS**	0.2757	0.6000	0.4266	0.5592	0.3990	0.6332
15	**NN**	-	**seqLHS**	0.3661	0.6661	0.4120	0.5957	0.4256	0.6540
16	**COMB**	**WeightAver**	**Classic**	0.3364	0.6629	0.4021	0.5978	0.3492	0.5574
17	**COMB**	**BestModel**	**Classic**	0.3368	0.6663	0.4007	0.6132	0.3587	0.5495
18	**COMB**	**Best2Models**	**Classic**	0.3364	0.6739	0.4018	**0.6221**	0.3540	0.5481
19	**COMB**	**Best10**	**Classic**	0.3368	0.6720	0.4000	0.6214	0.3498	0.5697
20	**COMB**	**WeightAver**	**SeqICC**	0.3430	0.5811	0.4063	0.4809	0.3805	**0.6641**
21	**COMB**	**BestModel**	**SeqICC**	0.3564	**0.6770**	0.4053	0.5886	0.3846	**0.6635**
22	**COMB**	**Best2Models**	**SeqICC**	**0.4017**	0.6557	0.4076	0.6158	0.4058	0.6612
23	**COMB**	**Best10**	**SeqICC**	0.3588	0.6421	0.4288	**0.6285**	**0.4498**	0.6604
24	**COMB**	**WeightAver**	**seqLHS**	0.3669	0.6636	0.4214	0.5743	0.4322	0.6527
25	**COMB**	**BestModel**	**seqLHS**	0.3930	0.6536	0.4311	0.5718	0.4134	0.6358
26	**COMB**	**Best2Models**	**seqLHS**	0.3931	0.6709	**0.4348**	0.5808	0.4430	0.6461
27	**COMB**	**Best10**	**seqLHS**	**0.4098**	**0.7008**	**0.4328**	0.5924	**0.4508**	0.6590
28		LHS		0.2264	0.5239	0.2749	0.4268	0.3090	0.5986
29		DEO		0.2009	0.5056	0.2593	0.4434	0.3011	0.5913
30		CMAES		0.0842	0.2645	0.0989	0.2311	0.1716	0.4514

trial does not lead to appreciable results (with exception of the Monteverdi piece referred to above). The same is true for the sequential optimization via LHS initial design by using just single models. By using the model combination approach **Best10** nevertheless this metaparameter setting seems to provide some meaningful results. However those can be found just in table 4 which contains music pieces with rather worse onset error rates.

An important issue is that neither of the model combination approaches is better than the best single model. To investigate this fact we analyzed the metaparameter setting 21 of pieces HAL and CHO in detail. The model combination approach used here is **BestModel** i.e. in each step the next trial point is chosen according to the model with the best prediction accuracy (see section 2.2). To remember, the prediction accuracy criterion is the leave-one-out mean squared error estimator. Figure 2 presents the averaged error estimations for the five models in each of 19 sequential steps (the mean was calculated over the ten replications for each ID). Although for both signals the best single model is Kriging, the best prediction accuracy for CHO is reached in each step by the neural network model. For this reason the entry for CHO (see table 3) in ID 21 (0.4053) is similar to the entry for CHO in ID 10 (0.4069). For HAL the entry in ID 21 is similar to the entries in ID's 7 and 12 because Kriging as the best model achieves till sequential step 15 the best prediction accuracy. This leads to the need of further investigation regarding to appropriate accuracy criteria for model combination.

Regarding the model type the best results are achieved either with Kriging or neural network single models. Figure 2 reveals the observed characteristic of neural networks: either they perform very well or badly. The most appropriate design setting according to the simulation results is the sequential parameter optimization with classical initial design (**SeqICC**). Concerning the model combination methods the best two models and best 10 trial points approaches seem to provide acceptable results.

Table 5 Result's aggregation

ID	model type	model combination type	design	place 1	place 2	place 3
1	**FSOM**	-	**Classic**	1	0	0
2	**KM**	-	**Classic**	1	0	0
3	**RF**	-	**Classic**	1	0	0
4	**SVM**	-	**Classic**	1	0	0
5	**NN**	-	**Classic**	1	0	0
6	**FSOM**	-	**SeqICC**	0	0	1
7	**KM**	-	**SeqICC**	4	2	0
8	**RF**	-	**SeqICC**	0	0	0
9	**SVM**	-	**SeqICC**	1	1	0
10	**NN**	-	**SeqICC**	5	0	0
11	**FSOM**	-	**seqLHS**	0	0	0
12	**KM**	-	**seqLHS**	0	0	0
13	**RF**	-	**seqLHS**	0	0	0
14	**SVM**	-	**seqLHS**	0	0	0
15	**NN**	-	**seqLHS**	0	0	0
16	**COMB**	**WeightAver**	**Classic**	1	0	0
17	**COMB**	**BestModel**	**Classic**	1	0	0
18	**COMB**	**Best2Models**	**Classic**	1	0	0
19	**COMB**	**Best10**	**Classic**	1	0	0
20	**COMB**	**WeightAver**	**SeqICC**	0	0	1
21	**COMB**	**BestModel**	**SeqICC**	0	0	1
22	**COMB**	**Best2Models**	**SeqICC**	0	1	3
23	**COMB**	**Best10**	**SeqICC**	0	2	2
24	**COMB**	**WeightAver**	**seqLHS**	0	0	0
25	**COMB**	**BestModel**	**seqLHS**	0	0	0
26	**COMB**	**Best2Models**	**seqLHS**	0	1	0
27	**COMB**	**Best10**	**seqLHS**	0	3	1

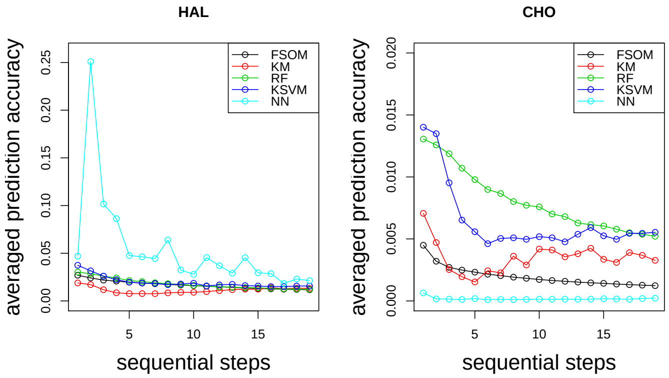

Fig. 2 Prediction accuracies for pieces HAL (table 3) and CHO (table 4) in ID 21

5 Conclusions

In this work many optimization strategies for a music signal analysis application problem – note onset detection – were introduced and compared. The usual optimization approaches in this research field are genetic algorithms. The main issue of this paper is to combine the classical design of experiments methods with ideas of sequential parameter optimization and model combination. Three different settings for sequential parameter optimization algorithm and four model combination approaches were proposed. The most important result is that the parameter optimization via design of experiments leads mostly to better results than conventional genetic algorithms. This can be stated at least for the considered low number of allowed function evaluations (34 iterations and 3 parameters). This shows the efficiency of design of experiments.

Due to the fact that none of the model combination methods is better than the best single model and that this obviously is caused by the used prediction accuracy criterion, there is a need for investigation of the influence of such criteria on the model combination outcome.

Two of the recommendable optimization strategies according to the conducted study are sequential parameter optimization approach with inscribed central composite initial design and the model combination approaches with respect to the predictions of the two best models and best ten predicted trial points. As a future research a parameter optimization of a more complex music signal analysis algorithm like an algorithm for music transcription is planned.

6 Acknowledgements

This work has been supported by the Collaborative Research Centre “Statistical Modelling of Nonlinear Dynamic Processes” (SFB 823) of the German Research Foundation (DFG), within the framework of Project C2, “Experimental Designs for Dynamic Processes”.

References

1. Ardia, D., Mullen, K., Peterson, B., Ulrich, J.: Global optimization by differential evolution. R-package (2011) http://cran.r-project.org/web/packages/DEoptim/
2. Atkinson, A. C., Donev, A. N.: Optimum Experimental Designs. University Press, Oxford (1992)
3. Bartz-Beielstein, T., Lasarczyk, C., Preu, M.: Sequential Parameter Optimization. In: McKay, B. et al. (eds.) Proceedings 2005 Congress on Evolutionary Computation (CEC’05) **1**, pp. 773–780. Piscataway NJ: IEEE Press, Edinburgh (2005)
4. Bauer, N., Schiffner, J., Weihs, C.: Einsatzzeiterkennung bei polyphonen Musikzeitreihen. SFB 823 discussion paper 22/2010 (2010)
   http://www.statistik.tu-dortmund.de/sfb823-dp2010.html
5. Bello, J. P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., Sandler, M. B.: QA Tutorial on Onset Detection in Music Signals. IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING **13** (5), 1035–1047 (2005)
6. Bischl, B.: Machine Learning in R. R-Package, TU Dortmund (2010), http://r-forge.r-project.org/projects/mlr/
7. Dixon, S.: Onset detection revisited. In Proc. DAFx-06, 133–137 (2006)
8. Escalante, H. J., Gomez, M. M., Sucar, L. E.: PSMS for neural networks. In: The IJCNN 2007 Agnostic vs Prior Knowledge Challenge, pp. 678–683 (2007)
9. Gorissen, D., Dhaene, T., De Turck, F.: Evolutionary Model Type Selection for Global Surrogate Modeling. J. of Machine Learning Research **10**, 2039–2078 (2009)
10. Jones D., Schonlau M., Welch W.: Efficient global optimization of expensive black-box functions. J. Global Optimization **13**, 455 – 492 (1998)
11. Karatzoglou, A., Meyer, D., Hornik, K.: Support Vector Machines in R. Journal of Statistical Software **15** (9) (2006)
12. Krige, D. G.: A statistical approach to some basic mine valuation problems on the witwatersrand. J. of the Chem., Metal. and Mining Soc. of South Africa, **52** (6): 119–139 (1951)
13. van der Laan, M. J., Polley, E. C., Hubbard, A. E.: Super Learner. Statistical Applications in Genetics and Molecular Biology **6** (1-25) (2007)
14. Lenth, R. V.: Response-Surface Methods in R, Using rsm. Journal of Statistical Software, **32** (7), 1-17 (2009)
15. Liaw, A., Wiener, M.: Breiman and Cutler’s random forests for classification and regression. R-package (2011) http://cran.r-project.org/web/packages/randomForest/
16. Roustant, O., Ginsbourger, D., Deville, Y.: Kriging methods for computer experiments. R-package (2011) http://cran.r-project.org/web/packages/DiceKriging/
17. Ripley, B.: Feed-forward Neural Networks and Multinomial Log-Linear Models. R-package (2009) http://cran.r-project.org/web/packages/nnet/
18. Sanchez, E., Pintos, S., Queipo, N. V.: Toward an optimal ensemble of kernel-based approximations with engineering applications. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN’06), pp. 2152–2158 (2006)
19. Sharkey, A.: On combining artificial neural nets. Connectionist Science, **8** (3), 299–314 (1996)

20. Stanley, K., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation, **10** (2), 99–127 (2002)
21. Stein, M.: Large Sample Properties of Simulations Using Latin Hypercube Sampling. Technometrics **29**, 143 – 151 (1987)
22. Trautmann, H., Mersmann, O.: Covariance Matrix Adapting Evolutionary Strategy. R-package (2011) http://cran.r-project.org/web/packages/cmaes/
23. Weihs, C., Jessenberger, J.: Statistische Methoden zur Qualittssicherung und -optimierung in der Industrie. Wiley-VCH, Weinheim (1999)
24. Zhu, D.: Hybrid approach for efficient ensembles. Decision Support Systems **48**, 480–487 (2010)