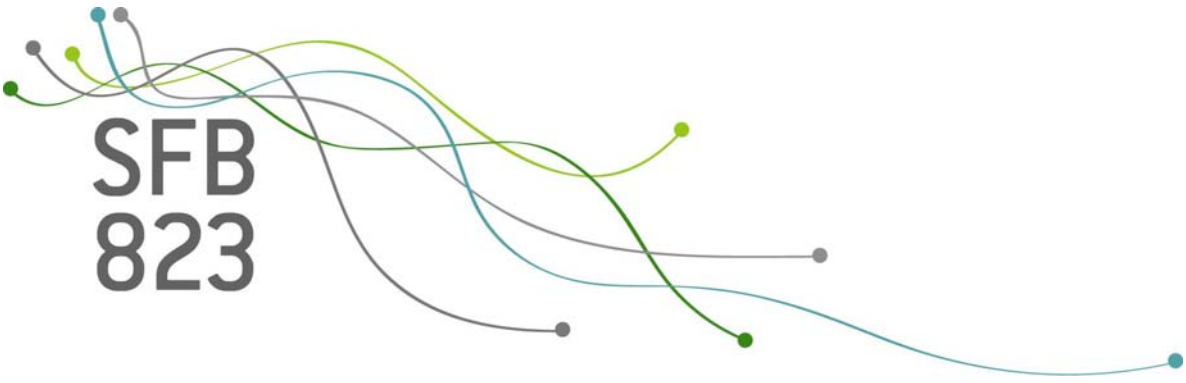
SFB 823

A multivariate approach for onset detection using supervised classification

Nadja Bauer, Klaus Friedrichs, Claus Weihs

Nr. 86/2016

Discussion Paper



A Multivariate Approach for Onset Detection Using Supervised Classification

Nadja Bauer[a,*], Klaus Friedrichs[a], Claus Weihs[a]

[a] *Department of Statistics, TU Dortmund University, D-44221 Dortmund, Germany*

Abstract

In this paper we introduce a new onset detection approach which incorporates a supervised classification model for estimating the tone onset probability in signal frames. In contrast to the most classical strategies where only one detection function can be applied for signal feature extraction, the classification model can be fitted on a large feature set. This is meaningful since, depending on the music characteristics, some detection functions can be more advantageous that the others.

Although the idea of the considering of many detection functions is not new in the literature, these functions are, so far, treated in a univariate way by, e.g., building of weighted sums. This probably lies on the difficulties of the direct transfer of the classification ideas to the onset detection task. The goodness measure of onset detection is namely based on the comparison of two time vectors while by the classification such a measure is derived from the frame-wise matches of predicted and true labels.

In this work we first construct – based on several resent publications – a comprehensive univariate onset detection algorithm which depends on many free settable parameters. Then, the new multivariate approach also depending on many free parameters is introduced. The parameters of the both onset detection strategies are optimized for online and offline cases by utilizing an appropriate validation technique. The main funding is that the multivariate strategy outperforms the univariate one significantly regarding the F-measure. Furthermore, the multivariate approach seems to be especially beneficial in online case since it requires only the halve of the future signal information comparing to the best setting of the univariate onset detection.

Keywords: Online Onset Detection, Model Based Optimization, Supervised Classification.

*Correspondence to: Department of Statistics, Vogelpothsweg 87, D-44227 Dortmund, TU Dortmund University, Germany. Tel.: +49231 755 7222; Fax: +49231 755 4387.

Email address: bauer@statistik.tu-dortmund.de (Nadja Bauer)

1. Introduction

A tone onset is the time point of the beginning of a musical note or another sound. Tone Onset Detection (OD) in music signals is an important step for many subsequent applications like music transcription and rhythm recognition. Several approaches have been proposed, but most of them can be reduced to the same basis algorithm just differing in the parameter settings ([1], [2], [3], [4]). They all follow the same scheme: windowing the signal, calculating an Onset Detection Function (ODF) for each window and localizing the tone onsets by picking the relevant peaks of the ODF. Many numerical and categorical parameters are involved in this procedure like the window size, the window overlap and the applied ODF.

In the classical procedure only one signal feature is considered for identifying tone onsets. While there exist several promising features, feature combination is an intuitive way. Since onset detection is a binary decision problem, supervised classification approach is suitable.

Another way, which is proposed in several recent publications, is to aggregate the features into one combining feature. For example, in [3] three OD features are considered in each signal frame based on spectral magnitude, phase and pitch. The onset decisions are first made for each feature separately and than merged to one feature by summing and smoothing the individual vectors. As a further example, in [5], the audio signal is disassembled in 40 frequency bins using an auditory model. The same OD feature is computed in each channel while for each signal frame the vector of 40 feature values is reduced to only one feature by using the quantile function. The general problem of such aggregating approaches is the loss of the information. Therefore, supervised classification which considers all features separately is a more appropriate solution.

Although in each signal frame a binary decision has to be performed (onset or no onset), onset detection is not a classical classification problem due to time dependencies.

The classification goodness measure considers matches between the predicted and the true class labels. However, in onset detection small detection delays are allowed, i.e, a tone onset might be correctly detected even if it found in a neighboring frame which would be counted as a misclassification in classical classification. Hence, it is not meaningful to minimize the misclassification rate since it does not automatically implies optimal OD performance. A similar problem is connected with overlapping frames where each onset obligatorily occurs in several neighboring frames. Furthermore the problem of unbalanced data arises since the most frames do not include an onset. This is a big challenge for most classification methods as the naive classification rule which assigns the label 'no onset' to all frames is already a good model w.r.t. misclassification rate.

There exist only few publications with application of supervised classification for tone onset detection (see, e.g.,[6] or DAVY).

[6] test different architectures of neural network model for onset detection. As input variables the output of the Short Time Fourier Transform is compared

with the Constant-Q transform. As the number of frequency bins (i.e., the number of input variables) in each signal frame is to high for efficient learning the neural network model, the authors use just some selected bins. The output of the learning model is not a binary vector (onset or no onset), but the smooth transitions provided by mixture of Gaussian. The peak extraction is based on the model output by its filtering, thersholding and calculation of center of mass positions inside each peak. They also remark, that using of high level features would be more beneficial in regard of training time.

? WAS ZU DAVY?

Our novel approach for multivariate OD is introduced in Section 3. Here, we consider the classical univariate OD procedure as the reference. For this reason, in Section 2 a comprehensive univariate OD algorithm is proposed first. It combines ideas of many state-of-the-art publications while instead of fixing the algorithm parameters (as done in the most publications) we will optimize them. Moreover, depending on the certain algorithm parameter settings which consider the required future signal segments, either online or offline OD can be achieved. Also the multivariate approach depends on a set of free parameters which have to be optimized.

For optimization, we use a sophisticated sequential model based approach shortly introduced in Section 4. The optimization is conducted on the data base of manual annotated music pieces described in Section 2.8. The result validation is conducted in a sophisticated manner by repeatedly dividing the data base into training and test data as provided in Section 4.2.

Section 5 presents research questions and analysis of the experimental results. Finally, in Section 6 the main findings are summarized and several ideas for future research are discussed.

2. Classical Onset Detection Algorithms

A tone onset marks, intuitively, the time point of the beginning of a new tone. However, there exist several formal definitions of the tone onset time ([1, 2, 7, 8]). The *perceptual* tone onset is defined as time point where a human listener can firstly recognize it while the *physical* onset represents time point of the first amplitude rise from zero (see [2]). There exist several statements regarding the delay in ms between the physical and perceptual onset times while in the most studies this delay does not exceed 50 ms. According to [9], the human listeners perceive – depending on the tempo of a music piece – two tone onsets withing 20 to 30 ms interval as simultaneous.

There exist two kinds of OD: *online* and *offline*. The offline OD is insofar easier as the whole signal information is allowed for making 'onset' vs. 'no onset' decision in the current signal frame. However, for many real applications (like for the hearing adds) the online OD is particularly important. In this case, just very small time delay (called latency time) is allowed for detecting a tone onset. In this paper, we also define a third detection type – *pseudo-online* OD – which is motivated through the works of [9] and [10]. The both papers

consider originally the case of the online OD as they respect the latency time. But they use audio recordings whose signal amplitudes are re-scaled in advance in a uniform interval (e.g., $[-1, 1]$). Note, for this reason, the knowledge of the absolute amplitude maximum of the whole recording is needed. Hence, the whole audio signal is incorporated indirectly. For the offline OD the amplitude re-scaling is allowed and will also be applied here.

In our pre-experiments we found that, after optimization, the pseudo-online OD significantly outperforms the online one. Although we will consider only online and offline approaches in the following, we would like to emphasize the pseudo-online OD in two respects. Firstly, the developers of online capable signal processing applications (not only limited to onset detection) should pay attention to using non standardized audio recordings as they can lead to too optimistic results which can not be achieved by the real online applications. Secondly, because of the high improvement potential, an adaptive signal amplitude re-scaling technique can be proposed for the online applications while, e.g., the amplitude maximum can be estimated in regular time intervals.

In this section, we will explain the individual steps of the classical onset detection scheme introduced in Algorithm 1. The main ideas of this algorithm are base on the tutorial of [1] while some extensions are motivated through [4, 2, 10] or are newly proposed here. Each OD step depends on many parameters. Instead of fixing, we will optimize them in the following, so that the set of possible levels or a region of interest is provided for each categorical or numerical parameter.

Algorithm 1: Classical onset detection.

1 split the signal into small (overlapping) windows;
2 pre-process the signal;
3 compute an ODF;
4 normalize the ODF;
5 threshold the normalized ODF;
6 localize the tone onsets;
7 measure the goodness of the detection.

2.1. Signal Windowing and the STFT

We assume a digital audio signal sampled with a rate of 44.1 kHz. This signal is split into l (possibly overlapping) windows of length N samples. Since we intend to carry out a Short-Time Fourier Transformation (STFT) in each of these windows, powers of 2 are chosen as window lengths. We will consider $512, 1024, 2048$ and 4096 samples as possible values for N. The hop size parameter h determines the distance in samples between the adjacent windows. We vary the hop sizes between $N/10$ and N samples (i.e., we allow the maximal frame overlap of 90%). Note, the smaller h, the more overlapping windows are generated and the more computational time is needed. This might affect the online capability of OD.

If required by the used OD feature (see line 3), a STFT is applied in each signal frame [2]):

$$X[n,j] = \frac{1}{N}\sum_{k=1}^{N} x[h\cdot(n-1)+k]w_N(k)e^{-2\pi ijk/N}, \tag{1}$$

where $X[n,j]$ is the Fourier coefficient (a complex number) of the jth frequency line in nth frame, $n = 1,\ldots,l$. $w_N()$ is the window function which can be optionally used to weight the signal amplitude in the frames before STFT calculating. A detailed overview about window functions and their characteristics is given in [11]. Here, we consider four popular functions as possible values of *window.fun* parameter: 'Uniform', 'Hanning', 'Blackman' and 'Gauss' (with $\sigma = 0.4$).

The spectral magnitudes $|X[n,j]|$ are defined as the absolute values of the Fourier coefficients while $|z| = \sqrt{Re(z)^2 + Im(z)^2}$. The Polar coordinate equivalence of the complex numbers is

$$X[n,j] = |X[n,j]|e^{i\phi[n,j]}. \tag{2}$$

$|X[n,j]|$ is the spectral magnitude and $\phi[n,j]$ is the phase (also called shift). $\phi[n,j]$ is an angle in interval $[-\pi,\pi]$ and can be calculated using the so called *atan2* function[1]:

$$\phi[n,j] = atan2\left(Re(X[n,j]), Im(X[n,j])\right). \tag{3}$$

2.2. Pre-Processing

A popular pre-processing method is the Adaptive Whitening (AW) proposed in [9] which leads to a signal based re-weighting of the STFT so that the activity variations of the different frequency lines are mapped to a similar range. The method can operate in online manner and depends on two parameters. The rounding parameter depends thereby on the maximal signal amplitude, where values greater than the spectral magnitude maximum switch off the AW[2]. As the author standardize the signal amplitude before applying of AW, the proposed region of interest for this parameter lies in interval $[0,1]$. In the real online applications the maximal amplitude can not be known in advance so that the upper limit of this interval can variate strongly. In our pre-experiments for the online OD, we varied the both parameters of the AW fixing all other OD parameters to the state-of-the-art settings while no AW effect could be recognized. For this reason, the AW option will not be considered in our algorithm.

Instead, two other pre-processing options introduced in [10] will be used her: spectral filtering and logarithmising the spectral magnitudes (only for OF

[1] `https://nf.nci.org.au/facilities/software/Matlab/techdoc/ref/atan2.html`, state: 01.03.2016.

[2] Absolute amplitude values effect directly the spectral magnitude values.

features utilizing the STFT). By spectral filtering (parameter *spec.filt* with levels 'yes' and 'no') a pseudo constant-Q filter bank is applied to the spectral magnitudes which bounds the frequency lines according to the semitones of the western music scale (from 27.5 Hz to 16 kHz). The resulting filter bank $F[j, b]$ contains then $b = 82$ frequency lines. If many Fourier coefficients are matched to one (new) frequency line, their magnitudes are weighted by a triangle window and summed. The filtered spectral magnitudes are gives as:

$$|X_{filt}[n, b]| = \sum_{j=1}^{N/2} |X[n, j]| \cdot F[j, b]. \tag{4}$$

Logarithmizing the spectral magnitudes is successfully applied by [10] and [12] and is presented in our algorithm by the categorical parameter *spec.log* with levels 'yes' and 'no'. The main idea here is to multiply the spectral magnitudes with the so called compression parameter ℓ and then take the logarithm. The region of interest for ℓ is here $[0.01, 20]$. Adding of a one is important for avoiding the negative values:

$$|X^{log}[n, j]| = \log_{10}(\ell|X[n, j]| + 1). \tag{5}$$

Note, the logarithmizing can be conducted for the original as well for the filtered spectral magnitudes.

2.3. Onset Detection Functions

Computation of an onset detection function in windows of the pre-processed signal is often called reduction ([1]), since after this step not the signal is analyzed anymore but only the vector of ODF values – *odf*. Many ODFs are based on the comparison of neighboring windows. An increase of an ODF generally indicates an onset, a decrease an offset. Also, offset information can improve onset detection ([13]). Subsequently, we will briefly discuss the 18 ODFs utilized in this study, represented by the categorical parameter *od.fun* in our optimization. Features which (in different ways) consider the tone offset information are marked with an *offset* index. Each feature is also highlighted with its individual number. Furthermore, for the purpose of better compactness, the frame index $n - 1$ is abbreviated with n'.

Signal Amplitude Based Features. The Zero-Crossing Rate (*ZCR*) is one of the simplest signal features. It gives the number of sign changes of the signal amplitude in a window. The direction of such changes can be ignored. Therefore, the absolute difference of the *ZCR* to the previous window (*ZCR.Abs.Diff*) is of interest (the greater the difference, the greater the likelihood of an onset):

$$\begin{aligned} ZCR(n) &= \frac{1}{N-1} \sum_{k=1}^{N-1} \mathbb{I}\{x[hn' + k] \cdot x[hn' + k + 1] < 0\}. \\ {}^{(1)}ZCR.Abs.Diff(n) &= |ZCR(n) - ZCR(n')|. \end{aligned} \tag{6}$$

$\mathbb{I}$ is an indicator function which takes the value 1 if the condition is fulfilled. For example, [14] use ZCR for classification of drums sounds.

The next amplitude based feature – Absolute Maximum (AM) – considers the difference between the absolute maxima of neighboring windows, as defined in [15]. In our optimization, we will consider two features, the difference and the absolute difference of AM in the neighboring windows:

$$\begin{aligned} AM(n) &= max\left(|x[hn'+1]|, \ldots, |x[hn'+N]|\right), \\ {}^{(2)}AM.Diff(n) &= AM(n) - AM(n'), \\ {}^{(3)}AM.Abs.Diff^{offset}(n) &= |AM(n) - AM(n')|. \end{aligned} \tag{7}$$

A further possibility of amplitude change measuring is the summing of all squired samples in each frame, also called as Amplitude Energy (AE, see [16]):

$$AE(n) = \sum_{k=1}^{N} (x[hn'] + k])^2. \tag{8}$$

Again, two features are of interest: the difference (${}^{(4)}AE.Diff$) and the absolute difference (${}^{(5)}AE.Abs.Diff^{offset}$) of the amplitude energy in neighboring windows whose formal definitions are analog to the Formula 7.

Spectral Magnitude Based Features. The first feature in this category is based on the spectral energy: summing the squared spectral magnitudes in each frame. However, in this form, the spectral energy would have exact the same values in each frame as the amplitude energy (due to the definition of the Fourier transformation). Furthermore, a tone onset can often be distinctly recognized in some special frequency lines, whereas other frequencies provide a blurred image. By the so called 'hard' onsets (mostly occurring by percussive or string instruments) the increase of the spectral energy is especially strong for the higher frequency lines. Therefore, [17] proposes a linear weighting of the absolute values of the Fourier coefficients – High Frequency Content (HFC) feature:

$$HFC(n) = \frac{2}{N} \sum_{j=1}^{N/2} (j \cdot |X[n,j]|)^2. \tag{9}$$

The difference (${}^{(6)}HFC.Diff$) and the absolute difference (${}^{(7)}HFC.Abs.Diff^{offset}$) of the linearly weighted spectral energy in neighboring windows are considered as OD features. According to [1], HFC feature is not well suited for other instrument classes (like wind instruments) or for detection of the 'soft' onsets.

As an alternative method, the Gauss window function can be exemplary used for the weighting (s. Section 2.1). In this way, the middle frequency lines are more influencing. This proposal will be called Gauss Frequency Content (GFC). The corresponding two features are then named ${}^{(8)}GFC.Diff$ and ${}^{(9)}GFC.Abs.Diff^{offset}$.

The following three features ([18]) – Spectral Centroid (SC), Spectral Spread (SSp) and Spectral Skewness (SSk) – consider the distribution properties of the spectral magnitude over the frequency lines. The spectral centroid indicates the location of the spectral distribution while smaller values mostly correspond to the lower tones. The direction of the change can be ignored here, so that only the absolute SC differences in the neighboring windows are considered:

$$SC(n) = \frac{\sum_{j=1}^{N/2} j \cdot |X[n,j]|}{\sum_{j=1}^{N/2} |X[n,j]|}, \tag{10}$$
$$^{(10)}SC.Abs.Diff(n) = |SC(n) - SC(n')|.$$

The spectral spread of a window represents the timbre of the playing instrument:

$$SSp(n) = \frac{\sqrt{\sum_{j=1}^{N/2}(j - SC(n))^2 |X[n,j]|}}{\sqrt{\sum_{j=1}^{N/2} |X[n,j]|}}. \tag{11}$$

Small values indicate instruments with only few overtones. Again, we are only interested in absolute differences of this feature in neighboring windows – $^{(11)}SSp.Abs.Diff$.

The spectral skewness is a measure for the skewness of the magnitude distribution.

$$SSk(n) = \frac{\sum_{j=1}^{N/2}(j - SC(n))^3 |X[n,j]|}{(SSp(n))^3 \sum_{j=1}^{N/2} |X[n,j]|}. \tag{12}$$

Low tones with few overtones will cause a positive skew. In contrast, the white noise or other unsystematic signal components should have the SSk values in the near of 0. Also here, analog to Formula 10, we only consider the absolute differences of SSk values: $^{(12)}SSk.Abs.Diff$.

Because of its particularly good recognition rate ([2, 4, 19]), the Spectral Flux (SF) is one of the most popular features for onset detection. The basic idea is to sum up the positive differences of the spectral magnitudes of neighboring windows for all frequencies. Negative differences are related to tone offsets and are hence not considered:

$$^{(13)}SF(n) = \sum_{j=1}^{N/2} H(|X[n,j]| - |X[n',j]|) \tag{13}$$

with the filter $H(x) = (x + |x|)/2$.

Alternatively, instead of the summing of the filtered absolute differences, the Euclidean distance of the spectral magnitudes in neighboring windows can be calculated. Hence, the new feature – Spectral Euclidean distance (SE) – considers tone offsets:

$$^{(14)}SE^{offset}(n) = \sum_{j=1}^{N/2} (|X[n,j]| - |X[n',j]|)^2. \tag{14}$$

Spectral Magnitude and Phase Based Features. This category of OD features concerns both the spectral magnitude and the phase (see Formula 3). It is expected that within one tone the growth of the phase between neighboring windows stays somewhat constant [12]). The Phase Deviation (PD) feature is then defined as the mean of the absolute values of the second differences of the phase over all frequencies:

$$\begin{aligned} {}^{(15)}PD(n) &= \frac{2}{N}\sum_{j=1}^{N/2}|\phi^{''}[n,j]|, \\ \phi^{''}[n,j] &= \phi[n,j] - 2\phi[n',j] + \phi[n'-1,j]. \end{aligned} \tag{15}$$

Further on, [2] proposes the Normalized Weighted Phase Deviation ($NWPD$) where the second differences are weighted by the corresponding percentage share of the absolute amplitude value regarding the signal itself:

$${}^{(16)}NWPD(n) = \frac{\sum_{j=1}^{N/2}|X[n,j]\phi^{''}[n,j]|}{\sum_{j=1}^{N/2}|X[n,j]|}. \tag{16}$$

The Complex Domain (CD) feature estimates the Fourier coefficients in the actual window according to the values in the two previous windows while assuming a stationary signal [2]). If the sum of the absolute differences of the estimated and the actual values over all frequencies is big, this can be interpreted as an indicator for a tone onset or offset.

$$\begin{aligned} {}^{(17)}CD^{offset}(n) &= \frac{2}{N}\sum_{j=1}^{N/2}|X[n,j] - \hat{X}[n,j]|, \\ \hat{X}[n,j] &= |X[n',j]|e^{i(2\phi[j,n'] - \phi[j,n'-1])}. \end{aligned} \tag{17}$$

Since it is important to distinguish between onsets and offsets, [2] proposes the Rectified Complex Domain (RCD), where magnitude differences are only taken into account if the absolute magnitude is increasing with respect to the previous window:

$$\begin{aligned} {}^{(18)}RCD(n) &= \sum_{j=1}^{N/2}H'(n,j), \\ H'(n,j) &= \begin{cases} |X[n,j] - \hat{X}[n,j]|, \text{if } |X[n,j]| > |X[n',j]|, \\ 0, \text{otherweise.} \end{cases} \end{aligned} \tag{18}$$

2.4. Normalization

The aim of *normalization* is to transform the *odf* feature vector into a standardized form for the subsequent thresholding. First, exponential smoothing with parameter $\alpha \in [0,1]$ can be applied, where for $\alpha = 1$ the time series stays

unchanged and for $\alpha = 0$ all values of a feature are equal. The smoothed vector will be termed *sm.odf*:

$$\begin{aligned} sm.odf_1 &= odf_1, \\ sm.odf_n &= \alpha \cdot odf_n + (1-\alpha) \cdot sm.odf_{n-1}. \end{aligned} \tag{19}$$

Most normalization methods are aiming at the scaling of *sm.odf* to a standard interval utilizing, e.g., $\max(sm.odf)$ and affecting the online ability of the method (see [4]). In what follows, we will, therefore, introduce threshold functions working with not normalized but only smoothed features (as also done in [10]).

2.5. Thresholding

Since not every local maximum of the *sm.odf* vector represents an onset, the threshold function aims at the distinction between relevant and irrelevant variations. A fixed value for the threshold is unfavorable since the method could then not react to dynamic changes of the signal. Instead, *moving threshold functions* are widespread ([10]):

$$T_n = \delta + \lambda \cdot mov.fun(|sm.odf_{n-l_T}|, \ldots, |sm.odf_{n+r_T}|), \tag{20}$$

where the parameter *mov.fun* (moving function) is either the median or the arithmetic mean. l_T and r_T are the numbers of windows to the left and to the right, respectively, of the nth window which are used in the calculation of *mov.fun*. Since *sm.odf* was not normalized and, hence, can lie in very differed intervals, it is difficult to a priori define the regions of interest for the parameters δ and λ which are meaningful for all OD features. For this reason, e.g., [10] optimizes δ separately for each ODF. However, as we are aiming in a global optimization using a sophisticated approach which can consider the interaction between the algorithm parameters, we optimize $\delta \in [0, 10]$ and $\lambda \in [1.1, 2.6]$ for all ODF's. These regions of interest are in accordance with the results of the similar optimizations ([10]) as well motivated through many pre-experiments.

Following [20], we also allow a moving p-quantile function as the third option of the *mov.fun* parameter. However, in this case, the parameter λ is fixed to 1 and p is optimized in the interval $[0.8, 0.98]$ instead. The allowed values for l_T and r_T will be discussed later.

2.6. Localization of Tone Onsets

The finally localized tone onsets should fulfill the following two conditions: *sm.odf* values should exceed the threshold being a local maximum. Following [10], we also use a third condition: A minimum distance *min.dist* (in number of windows) between the actual window and the window of the previous tone onset $n_{prev.onset}$ should be exceeded. To summarize:

$$O_n = \begin{cases} 1, & \text{if } sm.odf_i > T_n \text{ and} \\ & sm.odf_n = \max(sm.odf_{n-l_O}, \ldots, sm.odf_{n+r_O}), \\ & n > n_{prev.onset} + min.dist, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

$\boldsymbol{O} = (O_1, \ldots, O_l)^T$ is the tone onset vector and l_O and r_O are additional parameters, namely the number of windows to the left or right of the actual window, respectively, which are used for the calculation of the local maxima. The left limits of windows with $O_n = 1$ are taken as the time points of the tone onsets.

In [10], it is proposed to report the onsets one window later as actually detected. The authors argue that some features can increase earlier than a human listener would firstly recognize and note a tone onset. For the window length used in [10] it would correspond to a fix time shift of 10 ms. In our work, we will consider the parameter $onset.shift \in [-0.01, 0.02]$ s for optimization. The negative values are allowed as it can not be excluded that some features report tone onsets with a certain time delay.

In contrast to the most papers on the topic, we do not fix window length N and hop size h a priori but optimize them. This means, that parameter settings corresponding to the number of windows (like r_T) could stand for very different time periods depending on N and h. Therefore, all such parameters are re-defined according to the desired time length, N and h. Hence, we will not consider the parameters r_T, l_T, r_O, l_O and $min.dist$, but the times $t(r_T)$, $t(l_T)$, $t(r_O)$, $t(l_O)$ and $t(min.dist)$. For online applications, $t(r_O)$ and $t(r_T)$ are set to 0 s. In the offline case and universally for $t(l_O)$ and $t(l_T)$ these intervals are set to [0, 0.5] s. The region of interest for the parameter $t(min.dist)$ is $[0, 0.05]$ s.

To summarize, there are two application problems to optimize: online and offline OD. Offline OD has a set of 17 parameters while for online OD two parameters are fixed to 0 (i.e., 15 parameters remain for the optimization). In both cases four parameters are categorical and the remaining are numerical.

2.7. Goodness Measure for Onset Detection

The so called F-measure used here is proposed by [21] and defined as

$$F = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \quad F \in [0, 1], \tag{22}$$

where TP, FP, and FN stand for the number of *true positive* cases, *false positive* cases, and *false negative* cases, respectively, and $F = 1$ represents the optimal goodness.

The F-measure was originally defined for the classification problems but then adapted for onset detection applications (see, e.g., [2]). Hence, we will call the classification F-measure F_{class} in what follows. Since the true and the estimated onsets are compared in the time domain, a certain tolerance interval around the true onsets has to be defined in advance for the TP cases. We use here ± 25 ms as such tolerance interval (according to [10]) while ± 50 ms setting is also frequently applied in the literature ([1, 2]). This adapted F-measure will be called F_{onset}-measure in the following.

The first shortcoming of such adaption is the dependence on the defined tolerance interval and the lack of the precise measuring of the exact differences between true and estimated onset times. Secondly, as criticized in a comprehensive tutorial of [22], the common disadvantage of the F-measure is not considering

of the *true negative* (TN) cases. For example, if the same F_{onset}-value can be gained for two different music pieces while one of them is much longer then the other one, only the TN amount could help to identify the better detection rate. However, none of numerous publications cited in [22] proposes an alternative goodness measure which considers TN for comparing of two time vectors.

2.8. Music Data Base

The used data base consists of three frequently used manually annotated data sets: data base introduced in [1] with 23 pieces, in [3] with 92 recordings and in [10] with 206 music pieces. Altogether it counts 2 750 tone onsets. Many music instruments (like wind or string) and music styles (like European or oriental) are represented in this data set. According to [10], we aggregated true onset times which are reported within the 30 ms interval to only one onset.

3. Multivariate onset detection

Algorithm 2 provides our proposal for multivariate OD while the details of the individual steps are discussed in the subsequent subsections. Our approach allows overlapping signal frames and optimizes the desired F_{onset}-measure instead of F_{class} (although F_{class} is used in some steps for the reasons of simplicity).

Note that discussed procedure aims in optimization of algorithm parameters including the fitting of the best classification model. For real application many algorithm steps will be skipped.

The main idea for the optimization phase is the splitting the training data in two part: one for learning the classification rule for a given set of algorithm parameters and one for evaluating the goodness of this rule. The learning phase implies also the determination of the most important signal features which is realized by the variable selection step. In the evaluation phase only the few selected feature have to be calculated in signal frames. The novelty of the approach is that we are not interested in the predicted class labels but in the probability of a tone onset for each frame which is than handled as a single feature.

3.0.1. Spliting the Training Data Set

In the first line of Algorithm 2 the training data set is split into two parts: learning and evaluation data set. Learning data set is used for learning a classification rule which goodness is then verified on the evaluation data set. The proportion between both parts plays an important rule for the multivariate OD. On the one hand, a large learning data set implies a more common classification model, but, on the other hand, affects the model fitting time[3]. In this work we use 20% of the training data for learning purpose.

[3]The kind of the relationship between the size of the learning matrix and the model fitting time (e.g., linear, logarithmic or exponential) depends strongly on the utilized classification model.

3.0.2. Calculating of the ODF-matrix

In each signal frame 18 OD features (s. Section 2.3) can be calculated. In the most easy case the rows of the ODF-matrix would represent the signal frames and the 18 columns the values of the associated ODF's. Since the number of signal frames depends on the length of the music piece, the window size N and the hop size h, the dimension of the ODF-matrix can variate very strongly.

Obviously, not only the information of the actual frame, but also of the previews and the future frames can be employed for building the classification rule. In this manner we can consider the window overlapping effect as well the development patterns of tone onsets. Two important aspects should be taking into account here. Firstly, the inclusion of one additional frame leads to extension of the ODF-matrix by 18 new columns. Secondly, an a priori specification of the number of the additional frames is not meaningful as – depending on N and h – very different periods of time would be captured thereby. On that score, the number of frames to the right (r_M) and to the left (l_M) of the actual frame is defined here – similar to the procedure in Section 2.6 – as the function of the desired time interval in the past ($t(r_M)$) or the future ($t(l_M)$) signal, N and h. However, this number should not exceed thee frames (in each direction) in order to not substantially expand the dimension of the related ODF-matrix. For the online OD $t(r_M) = 0$ s and $t(r_M) = 0.15$ s while for offline case $t(r_M) = t(l_M) = 0.15$ s. This time limits are fixed in order to avoid additional parameters for the following optimization.

The matrix D_{lern} (line 2 of Algorithm 2) results from the row-by-row merge of all ODF-matrices of the learning data set. The subsequent analysis of the experimental results shows that the row number of D_{lern} varies in the five-figure region and the number of columns lies mostly between 55 (3 signal frames) and 126 (7 signal frames).

In the third line of Algorithm 2 the matrix D_{lern} is extended by the binary $(0, 1)$-column of the true onset information while the digit '1' notes a tone onset in the respective signal frame.

3.0.3. Variable Selection

The variable selection is an important but also very time intensive step. The aim of this step is the model based identification of the most important OD features which have to be calculated in the test phase. Reducing of the features number is especially relevant for the online OD.

The forward variable selection is implemented here via *selectFeatures* function of the **mlr** R package [23]). The required time effort depends on the used classification model and on the dimension of $\tilde{D}_{lern}$ matrix. Pre-experiments show that for the most classification models the variable selection can be conducted in acceptable time on a data matrix of 20 000 rows. Hence, if the row number of $\tilde{D}_{lern}$ exceeds this 'threshold', 20 000 rows are sampled randomly for the variable selection purpose.

Furthermore, the holdout approach is applied for measuring the goodness of fit in each variable selection step. For this reason, 50% of the (possibly previously reduced) $\tilde{D}_{lern}$ are used for the model fitting and the remaining 50%

Algorithm 2: Multivariate onset detection.

1 split the training data set into the learning and the evaluation data sets;
2 calculate for all pieces of the learning data set the ODF-matrix and merge them to one learning matrix D_{learn} by rows;
3 add the column of the true class labels to D_{learn}: $\tilde{D}_{learn}$;
4 select the most influencing features of $\tilde{D}_{learn}$ via the forward variable selection based on the used classificator;
5 fit the classification model M_{class} on the matrix of the selected features $\tilde{D}^{sel}_{learn}$;
6 **for** *music pieces of the evaluation data set* **do**
7 | calculate the ODF-matrix of the selected features;
8 | predict for each signal frame the probability p^{onset} of the tone onset according to M_{class};
9 | estimate onset times using p^{onset} vector;
10 | compute the F_{onset}-value;
11 **end**
12 mean the F_{onset}-values over of the evaluation data set.

for the goodness verification (the data split proceeds randomly). Note, the parameter `stratify` of the *makeResampleDesc* function (in **mlr** R package) was set to `TRUE` as a problem of unbalance data occurs here: There are more observation with 'no onset' as these with 'onset' labels. Through the mentioned setting, the training data is sampled in such a way that the number of positive and negative examples is approximately the same (if possible).

The goodness measure of the variable selection is the classification F-measure F_{class}. An additional feature is included only in the case if it contributes a minimum improvement of 0.01 to the already achieved F_{class}-value. The result of the selection step is the set of the selected features and the associated learning matrix $\tilde{D}^{sel}_{lern}$ (only with selected features). Note, even if the matrix $\tilde{D}_{lern}$ had to be reduced for variable selection purpose (to 20 000 rows), $\tilde{D}^{sel}_{lern}$ contains now all rows of $\tilde{D}_{lern}$.

3.0.4. Fitteng the Calssification Model

In line 5 of Algorithm 2 the final classification rule M_{class} is fitted on the $\tilde{D}^{sel}_{lern}$ data matrix. This model is then used in the succeeded evaluation step (lines 6-11). Also in the validation phase (see Section 4.2) no new model has to be fitted. Instead, the model related to the best found parameter setting is utilized. The classification rule is, hence, also an output of the optimization.

In this work, we aim to compare three classification models: logistic Regression (from R package **stats**, [24], abbr. with *logReg*), random forest (from R package **randomForest**, [25], abbr. with *randForest*) and support vector machines (from R package **e1071**, [26], abbr. with *SVM*). The theoretical foundations of the referred classification methods are provided in [27]. For each

model a separate optimization is conducted.

Since the goodness of *randForest* and *SVM* can strongly variate depending on their internal settings – so called hyper-parameters – these parameters were tuned in advance on a small data bank of 30 music pieces (real recordings as well MIDI-pieces) according to F_{class}. The details of the hyper-parameter tuning will be skipped here. The optimal found settings are:

- ν-*SVM* [28]): $\nu = 0.56$ with kernel $K(x, y) = 5x^T y + 57$;
- *randForest*: number of trees = 174, minimal node size = 9, number of candidate variables in each step = 27.

3.0.5. Evaluataion Phase

In lines 6 to 11 of Algorithm 2 the goodness of the onset detection is evalauted for each music piece of the test data set and subsequently averaged in line 12. For this reason, the ODF-matrix of selected features is calculated in line 7 and the classification model M_{class} (fitted in the training phase) is used for predicting the probability of a tone onset p^{onset} for each row (i.e., signal frame). This probability vector is then handled as an univariate OD feature by using the Formula 21 (see Section 2.6) while instead of *sm.odf* the vector p^{onset} is used:

$$O_n = \begin{cases} 1, & \text{if } p_n^{onset} > \mathit{class.thresh} \text{ and} \\ & p_n^{onset} = \max(p_{n-l_O}^{onset}, \ldots, p_{n+r_O}^{onset}) \text{ and} \\ & n > n_{\mathit{last.onset}} + \mathit{min.dist}, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

Furthermore, we use here a fix threshold *class.thresh*. The standard setting of *class.thresh* for the class assignment is 0.5. However, [29] advice to tune this threshold in the case of unbalanced classifcation problmes. So we optimize the parameter *class.thresh* in interval $[0.05, 0.95]$. Parameters r_O, l_O and *min.dist* will be optimized on the same regions of interest for online and offline approaches as by the univarite OD. The estimated onset times are detected according to the $\boldsymbol{O} = (O_1, \ldots, O_M)$ vector (analogue to Section 2.6). The individual F_{onset}-values are calculated in line 10 and then averaged over the evaluation data set in line 12.

The multivariate OD has a smaller set of influencing parameters than the univariate OD. Not considered are parameters of the exponential smoothing (as no more smoothing is required) and the moving threshold parameters. Also no onset shifting will be applied here since the optimal values of *onset.shift* in the univariate case have been found to lie in the near of 0. Furthermore, fixing this parameter to 0 reduces the optimization time. Overall, there are 10 parameters[4] to be optimized (3 categorical and 7 numerical) for the offline multivariate OD. In the online case there is one numerical parameter fewer (since $r_O = 0$ s).

[4] N, h, *window.fun*, *spec.filt*, *spec.log*, ℓ, r_O, l_O, *min.dist* and *class.thresh*.

4. Optimization Strategy and Validation Approach

4.1. Model Based Optimization

Sections 2 and 3 describe the algorithms of univariate and multivarate OD which depend on many free settable continuous and categorical parameters. These parameters have to be optimized in an appropriate manner. We use for this reason the sequential surrogate Model Based Optimization strategy (MBO) (s. Algorithm 3). Here, we will skip many details of this method as they can be found, e.g., in [30, 31, 32, 33] and mention only its main ideas instead. An onset detection algorithm is supposed to be an unknown nonlinear black-box function $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ of d parameters. Each parameter has a region of interest $[\ell_i, u_i]$ while the Cartesian product $\mathcal{X} = [\ell_1, u_1] \times \ldots \times [\ell_d, u_d]$ defines the interesting parameter space for the optimization. One possible parameter setting $\boldsymbol{x}_i \in \mathcal{X}$ is called a point while $y_i = f(\boldsymbol{x}_i)$ is the value of the target function in this point. For onset detection applications, y is the F_{onset}-value of the associated algorithm configuration. A design $\mathcal{D} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ is a set of n points and $\boldsymbol{y} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^T$ is the vector of the target values for this design.

Algorithm 3: Sequential model based optimization.

1 generate an initial design $\mathcal{D} \subset \mathcal{X}$;
2 evaluate f on the initial design: $\boldsymbol{y} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^T$;
3 **while** *optimization budget is not exhausted* **do**
4 | fit the surrogate model on $\mathcal{D}$ and $\boldsymbol{y}$;
5 | find $\boldsymbol{x}^*$ with the best infill criterion value;
6 | evaluate f on $\boldsymbol{x}^*$: $y^* = f(\boldsymbol{x}^*)$;
7 | update $\mathcal{D} \leftarrow (\mathcal{D}, \boldsymbol{x}^*)^T$ and $\boldsymbol{y} \leftarrow (\boldsymbol{y}, y^*)^T$;
8 **end**
9 **return** $y_{min} = \min(\boldsymbol{y})$ and the corresponding $\boldsymbol{x}_{min}$.

The MBO procedure in Algorithm 3 can be summarized as follows: In the first step, an initial design with n points is evaluated and a surrogate model is fitted. The surrogate model is then used for the prediction of a new design point. As long as the optimization budget is not exhausted, the new point $\boldsymbol{x}^*$ is chosen in the parameter space based on a so-called infill criterion derived from the surrogate model. The target function is evaluated in this point. The surrogate model is then updated on the design extended by the new point while the updated model is used for the next iteration. The point with the minimal target function value is taken as the result of the optimization.

In line 1 the latin hypercube sampling ([34]) designs with $5 \cdot d$ points are used for the initialization step. The number of sequential steps is set to $20 \cdot d$. Theoretically, an arbitrary regression model can be used as a surrogate. We use here the very popular ordinary Kriging model [31] which is, however, limited to only continuous parameters. As there are also categorical parameters to be

optimized, we use a simple strategy – naive Kriging – where each level is assigned to an integer resulting in a continuous parameter. The proposed values of the corresponding continuous parameter in the sequential steps are rounded and converted back to the nearest categorical level. Although we artificially define order and intervals between the levels which actually do not exist, this strategy showed satisfying results for onset detection applications [33]).

The Expected Improvement (EI) criterion, as proposed in [32], is used in line 5 as an infill criterion. EI supports the global convergence [35] and becomes the standard criterion in many applications. In each MBO iteration a new point is chosen by maximizing the infill criterion (line 6). To solve the corresponding non-linear optimization problem, we use the focus search algorithm implemented in the R package **mlrMBO** [36] which successively focuses the parameter space on the most promising regions.

4.2. Validation

When an onset detection algorithm is optimized on a music data set and then the best found F_{onset}-value is reported as the result, a strong over-fitting to the used data appears. The correct approach is to find the optimal settings of algorithm parameters on a train data set and then verify them on an additional test data set. Unfortunately, in many onset detection papers no validation is conducted, so that too optimistic F_{onset}-values are reported. Also [2] draws attention to over-fitting and other problems occurring in onset detection tasks.

Here, we apply the holdout validation approach by randomly spiting the data set in a training part (2/3 of the whole data) and a test part (the remaining 1/3). This approach is replicated 30 times so that in each replication the MBO optimization provides an own set of optimal algorithm parameter settings. Hence, a vector of 30 validated (mean) F_{onset}-values over the associated test data corresponds to each optimization problem (e.g., univariate OD in online case). The optimization problems can then be compared by these vectors both descriptively and using appropriate statistical tests. As the F_{onset}-values are not assumed to be normally distributed, the Wilcoxon signed rank test [37, p. 128 ff.] is considered here as a non-parametric alternative to the t-test. In accordance with [37] (p. 132) the sample size of 30 observations is sufficient for the desired asymptotic property of the test statistics. The significance level is assumed to be 5%.

4.3. Implementation

The experiments were executed in parallel using the **BatchExperiments** R package [38] on the Linux-HPC cluster system[5] of TU Dortmund University. The MBO optimization is conducted using a developing version of **mlrMBO** R package [36]. The univariate and multivariate OD algorithms are implemented in the *R* programming language [24] and can be provided on request.

[5] `http://lidong.itmc.tu-dortmund.de/ldw/index.php?title=System_overview&oldid=259`.

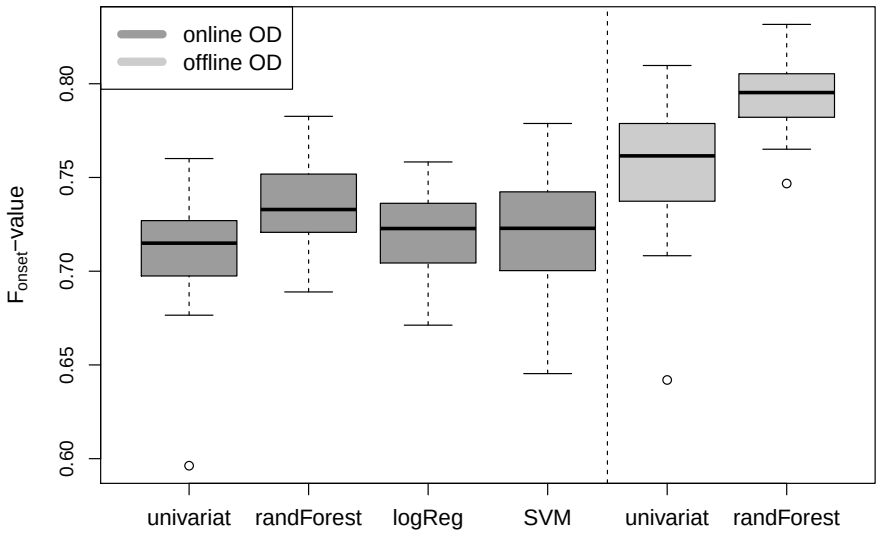


Figure 1: *Comparisson of the optimization results for the univariat and the multivariate OD (with three classificators: logReg, randForest and SVM) in online and offline cases.*

5. Experiments

The main research question is whether the multivariate OD outperforms the univariate one both in online and offline cases. The experimental results are illustrated in Figure 1 which shows the distribution of mean F_{onset}-values on the test data splits corresponding to the best parameter settings found by the used model based optimization strategy in 30 replication runs. Note, for the online OD (four left boxplots), optimization runs are conducted for each classification model (*randForest*, *logReg* and *SVM*) while for the offline OD (two right boxplots) only the best classificator (*randForest*) is used. It is obvious that the utilized classificator has an essential effect on the optimization results. While both *SVM* and *logReg* models seem to perform similarly to the univariate OD, *randForest* model outperforms it significantly. The results of multivariate OD using *randForest* clearly outperforms the results of univariate OD for online as well offline approach. For the online approach, p-value of the associated Wilcoxon signed rank test is $1.25 \cdot 10^{-4}$ and for the offline $2.91 \cdot 10^{-6}$.

Regarding the optimization effort, very different time intervals could be observed depending on the optimization problem. Figure 2 shows die distribution of the overall time for function evaluations in the sequential steps of MBO. Time for surrogate model fitting and EI optimization is not considered here as

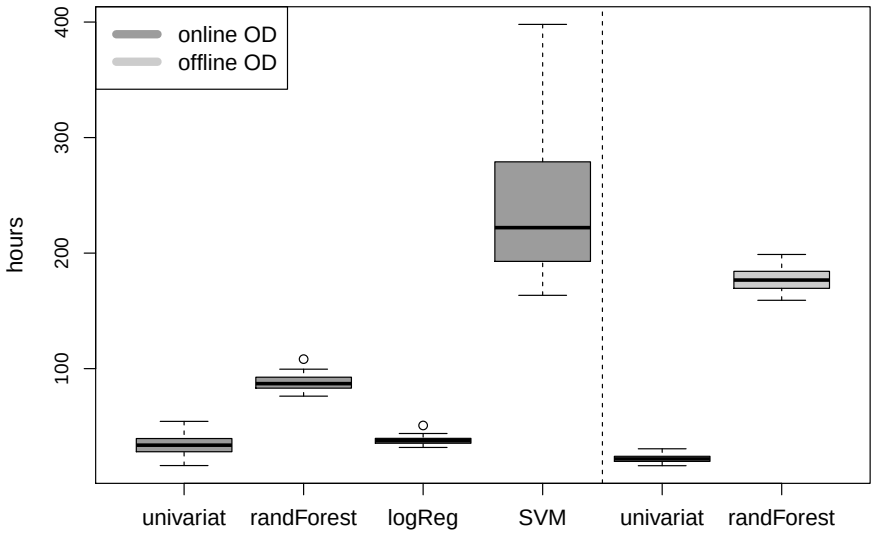


Figure 2: *Common time in hours for function evaluations in sequential steps of MBO. Optimization problems: univariate and multivariate OD (with three classificators: logReg, randForest and SVM) for online and offline cases.*

it strongly depends on the number of parameters to optimize which is different for univariat and multivaraite OD. As can be seen, the model fitting time for *SVM* model is extremely large and shows a big variation. In contrast, the fitting time variation of *randForest* and *logReg* models is small. This fact facilitates the planning of the experimental time. Optimization time of *randForest* based multivariate OD exceeds this time for univariate OD optimization in several times. However, the multivariate approach is well capable for online applications as the time intensive model fitting and variable selection steps have to be conducted only in the optimization phase. In real applications, only few OD features (s. Figure 3 discussed later) have to be calculated in the signal frames and then the already available model applied.

Table 1 compares the averaged validated F_{onset}-values and its standard deviations for univariate OD optimized with MBO, for parameter settings proposed in [10] (supposed as the state-of-the-art here) and for multivaraite OD. On the one hand, the global optimization of univariate OD does not show better performance as the state-of-the-art settings (the difference between the both approaches is, however, not significant according to the Wilcoxon rang sign test). To remain, the latter ones were optimized in [10] on the used data set so that they good performance was also expected in this study. Also the standard

Table 1: Mean validated F_{onset}-values und standard deviations (in braces) of the optimized OD algorithm paramater settings as well of the settings proposed in [10] (state-of-the-art).

approach	*mean F_{onset}-value (standard deviation)*	
	online	*offline*
univriate OD MBO optimization	0.709 (0.029)	0.756 (0.035)
univriate OD state-of-the-art	0.714 (0.018)	0.756 (0.018)
multivariate OD MBO optimization	0.735 (0.218)	0.794 (0.019)

variation for state-of-the-art setting is fewer as for the optimized ones. On the other hand, the multivaraite approach improves the mean F_{onset}-value by 0.021 in online case and by 0.035 in offline case (comparing with the state-of-the-art).

Now let as compare the best algorithm parameter settings. As in the case of the optimitzation in each otpimization run different settings were found, all 30 best combinations were applied on the whole data set. The setting with the best performance on the whole data is supposed here to be the best one for MBO and is presented in Table 2. This is done both for univariate and multivariate OD in online and offline case. Furthermore, the state-of-the-art setting as proposed in [10] is also listed[6]. Note, the full specification of the best multivariate OD setting is not possible here as the best random Forest model can not be presented in values. The associated fitted model can be provided on request.

From Table 2 can be followed that the most successful window length is 2048 samples (46 ms) both for online and for offline OD. However, the advantage of the best multivariate OD setting is the window length of 1024 samples which effects the halving of the latency time (s. Section 2). Moreover, the large hop size implies building of only 54 windows in a second which also reduces the computational time and makes the approach more interesting for real applications. Further on, the best onset detection feature in univariate case is the spectral flux. The most preferred window function is the Hunning function. Similar to [10], spectral filtering and logarithmizing of the spectral magnitudes improve the detection ability.

Other interesting funding is that much more signal windows are needed for computing the moving threshold function (see parameters $t(l_T)$ and $t(r_T)$) than for localizing the tone onsets (parameters $t(l_O)$ and $t(r_O)$). Finally, the classification threshold for multivariate OD lies in the online case by 0.310 which is noticeable smaller than the standard setting of 0.5. In offline case, in contrast, the optimal value lies in the near of the standard one.

[6] In offline case the parameter *onset.shift* is set to 0 s as it leads to much better F_{onset}-value as with the originally proposed setting *onset.shift* = 0.010 s with $F_{onset} = 0.744$.

Table 2: Comparisson of the best parameter settings of the optimized univariate and multivariate OD algorithm as well the reference state-of-the-art setting (proposed in [10]).

	F_{onset}-*value* / *parameter*	*MBO univ. OD*	*state-of-the-art*	*MBO multiv. OD*
online	F_{onset}	**0.763**	**0.725**	**0.787**
	N	2048	2048	1024
	h	389	441	816
	wind.func	Hanning	Hanning	Hanning
	spec.filter	yes	yes	yes
	spec.log	yes	yes	yes
	ℓ	0.085	1	19.250
	od.fun	*SF*	*SF*	-
	α	0.699	1	-
	mov.fun	median	mean	-
	λ	1.180	1	-
	δ	1.634	2.500	-
	$t(l_T)$	0.403	0.100	-
	$t(r_T)$	0	0	-
	$t(l_O)$	0.030	0.030	0.027
	$t(r_O)$	0	0	0
	min.dist	0.042	0.030	0.025
	onset.shift	0.008	0.010	-
	class.thresh	-	-	0.310
offline	F_{onset}	**0.800**	**0.790**	**0.838**
	N	2048	2048	2048
	h	563	441	1043
	wind.func	Hanning	Hanning	Blackman
	spec.filter	yes	yes	yes
	spec.log	yes	yes	yes
	ℓ	4.174	1	1.017
	od.fun	*SF*	*SF*	-
	α	0.711	1	-
	mov.fun	median	mean	-
	λ	1.342	1	-
	δ	1.580	2.500	-
	$t(l_T)$	0.395	0.100	-
	$t(r_T)$	0.452	0.100	-
	$t(l_O)$	0.029	0.030	0
	$t(r_O)$	0.051	0.030	0.052
	min.dist	0.041	0.030	0.037
	onset.shift	-0.009	0	-
	class.thresh	-	-	0.546

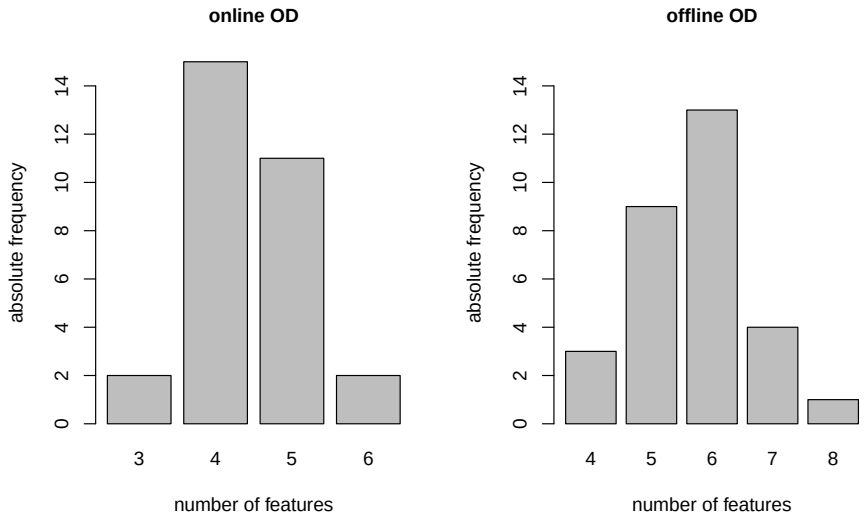


Figure 3: *Number of selected features in the variable selection step (s. line 4 of Algorithm 2) for online and offline multivariat OD with randForest model.*

The last evaluation question considers the variable selection step of multivariate OD with *randForest* classification model. To remain, this step is applied in order to reduce the number of ODF's to be computed in each signal window. Especially for online multivariate OD this number might play an important rule (for reducing the computational time).

Figure 3 shows the distribution of the number of selected features in 30 replications for online and offline multivariate OD. The larger number of selected features in offline case can be explained by the higher number of available variables in $\tilde{D}_{learn}$ matrix (s. Section 3.0.2). Overall, the variable selection step is very efficient since the number of features which have to be calculated in each window is reduced by several times. Moreover, in the most cases several variables refer to one feature: to its value in the current and next or previous windows which are already determined.

Finally, Table 3 provides the most frequently selected onset detection features (see Section 2.3) in 30 replications. The labels '_a', '_left_1' or '_right_2' indicate the feature values in the actual, first left or second right frame, respectively (see parameters l_M and r_M in Section 3.0.2). The table consist of three parts: the upper part presets the most frequently selected features (i.e., which were selected in at least 10% of replications) in online case. The right column gives the selection frequency of these features in offline case. The middle part of the table shows the features frequently selected for offline OD which consider only actual or left frames but are not (or not frequently) selected in online case. Lastly, the last part of the table provides frequently selected features for offline

OD which consider the right (i.e., future) frames and hence can not occur in online case.

Table 3: The most frequently selected features in the feature selection step of the online and offline multivariate OD with *randForest* model (s. line 4 of Algorithm 2). The frequencies of feature selection in 30 replications of the associated optimization approach are presented.

OD feature	*frame position*	*frequency in %*	
		online OD	*offline OD*
SF	actual	97	47
SE^{offset}	left_1	67	30
RCD	actual	27	3
$HFC.Diff$	actual	27	7
$GFC.Diff$	left_1	23	7
SF	left_1	23	23
CD	actual	20	3
CD^{offset}	left_3	17	10
$SC.Abs.Diff$	actual	13	0
$NWPD$	actual	10	7
RCD	left_3	10	3
$AE.Abs.Diff^{offset}$	actual	10	0
$AE.Diff$	left_1	10	0
RCD	left_1	0	17
$NWPD$	left_1	0	10
SF	left_2	0	10
SF	right_1	0	100
SE^{offset}	right_1	0	33
SF	right_3	0	33
SF	right_2	0	27
SE^{offset}	right_2	0	23
$HFC.Diff$	right_2	0	23
$AM.Diff$	right_2	0	23
$HFC.Diff$	right_3	0	20
RCD	right_2	0	10
$GFC.Diff$	right_3	0	10

According to Table 3, the most important features in online case are the spectral flux of the actual frame as well the spectral Euclidean distance (here the offset information is considered) of the first previews frame. For offline OD, the spectral flux feature of the first future frame is by far the most important one, followed by spectral flux and spectral Euclidean distance of the neighboring frames. Furthermore, in 10% - 27% of replications also different variants of high frequency content and complex domain features are selected in online and offline cases. Underrepresented or not at all selected are, in contrast, amplitude based features as well the features based on statistical measures of the spectral

magnitude distribution like spectral centroid or spectral skewness.

6. Conclusion

In this paper we first composed and optimized a comprehensive algorithm for classical (univariate) onset detection which is based on the several state-of-the-art publications. The algorithm can operate in online and offline manner, depending on certain parameter settings. For avoiding the over-fitting, the used data set is randomly split in training and test part while the best algorithm parameter settings are determined on the training data and then validated on the test data. This procedure is replicated 30 times. The optimization is conducted by means of a sophisticated sequential surrogate model based approach.

On the one hand, the state-of-the-art setting of [10] shows slightly better mean and fewer variation of the validated F_{onset}-values compared with the optimal settings found in every MBO replication ran. On the other hand, however, the best determined setting over the whole data set performs noticeable better that the state-of-the-art setting in online case.

The main contribution of this work, however, is introduction and optimizing of the new multivariate approach for onset detection where many detection features are considered in each signal frame for detecting a tone onset by utilizing a classification model. As the direct application of the classification techniques is not possible for tasks where two time vectors have to be compared for the goodness measurement, the classification model is used for computing the probability of a tone onset in each frame which is then treated as an univariate feature. Three classification models are used for this purpose: logistic regression, support vector machines and random forest. While the first both models does not show satisfying results, the multivariate approach with random Forest model outperforms the univariate one significantly for online as well offline onset detection.

Further advantage of the online multivariate algorithm is that its best parameter setting leads to halving of the latency time since the optimal frame length is found to be 23 ms in contrast to 46 ms of univariate detection algorithm. Due to the implemented feature selection step, only few features have to be determined in each signal frame in real time applications so that the online capability of the multivariate approach is well kept.

In our future research we aim to improve the multivariate onset detection in several aspects. On the one hand, also other classification models should be considered and compared for this task. On the other hand, instead of utilizing the original signal, its decomposition to many frequency bands according to an acoustic model can be meaningful. Also here the most important features and frequency bands can be determined via variable selection step for the subsequent fitting of a classification model. In this manner we would further develop our acoustic model based approach proposed in [5].

Acknowledgment

This paper is based on investigations of the projects C2 and B3 of SFB 823, which are kindly supported by the German Research Foundation (DFG).

References

[1] J. P. Bello, L. Daudet, S. A. Abdallah, C. Duxbury, M. E. Davies, M. B. Sandler, A tutorial on onset detection in music signals, IEEE Transactions on Speech and Audio Processing 13 (5-2) (2005) 1035–1047.

[2] S. Dixon, Onset detection revisited, in: Proceedings of the 9th International Conference on Digital Audio Effects (DAFx'09), 2006, pp. 133–137.

[3] A. Holzapfel, Y. Stylianou, A. Gedik, B. Bozkurt, Three dimensions of pitched instrument onset detection, IEEE Transactions on Audio, Speech, and Language Ppocessing 18 (6) (2010) 1517–1527.

[4] C. Rosão, R. Ribeiro, D. Martins De Matos, Influence of peak selection methods on onset detection, in: Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR'12), 2012, pp. 517–522.

[5] N. Bauer, K. Friedrichs, D. Kirchhoff, J. Schiffner, C. Weihs, Tone onset detection using an auditory model, in: M. Spiliopoulou, L. Schmidt-Thieme, R. Janning (Eds.), Data Analysis, Machine Learning and Knowledge Discovery, Springer International Publishing, 2014, pp. 315–324.

[6] A. Lacoste, D. Eck, A supervised classification algorithm for note onset detection, EURASIP Journal on Applied Signal Processing 2007 (1) (2007) 1–13.

[7] R. Polfreman, Comparing onset detection & perceptual attack time, in: A. de Souza Britto Jr., F. Gouyon, S. Dixon (Eds.), Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR'13), 2013, pp. 523–528.

[8] J. Vos, R. Rasch, The perceptual onset of musical tones, Perception & Psychophysics 29 (4) (1981) 323–335.

[9] D. Stowell, M. Plumbley, Adaptive whitening for improved real-time audio onset detection, in: Proceedings of the International Computer Music Conference (ICMC'07), 2007, pp. 312–319.

[10] S. Böck, F. Krebs, M. Schedl, Evaluating the online capabilities of onset detection methods, in: Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR'12), 2012, pp. 49–54.

[11] F. J. Harris, On the use of windows for harmonic analysis with the discrete fourier transform, IEEE 66 (1) (1978) 51–83.

[12] F. Eyben, S. B"ock, B. Schuller, A. Graves, Universal onset detection with bidirectional long short-term memory neural networks, in: J. S. Downie, R. C. Veltkamp (Eds.), Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR'10), International Society for Music Information Retrieval, 2010, pp. 589–594.

[13] E. Benetos, S. Dixon, Polyphonic music transcription using note onset and offset detection, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 37–40.

[14] F. Gouyon, F. Pachet, O. Delerue, Classifying percussive sounds: a matter of zero-crossing rate?, in: A. de Souza Britto Jr., F. Gouyon, S. Dixon (Eds.), Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR'13), 2013, pp. 523–528.

[15] N. Bauer, J. Schiffner, C. Weihs, Einfluss der Musikinstrumente auf die Güte der Einsatzzeiterkennung, Tech. Rep. 10/12, TU Dortmund (2012).

[16] W. A. Schloss, On the automatic transcription of percussive music - from acoustic signal to high-level analysis, Master's thesis, Stanford University (1985).

[17] P. Masri, Computer modeling of sound for transformation and synthesis of musical signal, Ph.D. thesis, University of Bristol (1996).

[18] G. Peeters, X. Rodet, A large set of audio feature for sound description (similarity and classification) in the cuidado project, Tech. rep., Ircam (2004).

[19] N. Bauer, K. Friedrichs, B. Bischl, C. Weihs, Fast model based optimization of tone onset detection by instance sampling, in: A. Wilhelm, H. A. Adalbert (Eds.), Analysis of Large and Complex Data, Springer International Publishing, 2015, pp. ?–?

[20] N. Bauer, J. Schiffner, C. Weihs, Comparison of parameter optimization techniques for a music tone onset detection algorithm, in: Proceedings of the 4th Meeting on Statistics and Data Mining (MSDM), 2013, pp. 28–34.
URL `http://www.tasa-online.com/`

[21] C. Van Rijsbergen, Information Retrieval, Butterworths, 1979.

[22] F. Salfner, M. Lenk, M. Malek, A survey of online failure prediction methods, ACM Computing Surveys 42.
URL `http://doi.acm.org/10.1145/1670679.1670680`

[23] B. Bischl, M. Lang, J. Richter, J. Bossek, L. Judt, T. Kuehn, E. Studerus, L. Kotthoff, mlr: Machine Learning in R, r package version 2.4 (2015).
URL `http://CRAN.R-project.org/package=mlr`

[24] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria (2014).
URL http://www.R-project.org/

[25] A. Liaw, M. Wiener, Classification and regression by randomforest, R News 2 (3) (2002) 18–22.

[26] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch, e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien, r package version 1.6-7 (2015).

[27] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York Inc., New York, 2001.

[28] C.-C. Chang, C.-J. Lin, Training ν-support vector regression: Theory and algorithms, Neural Computation 14 (8) (2002) 1959–1977.

[29] T. Voigt, R. Fried, M. Backes, W. Rhode, Threshold optimization for classification in imbalanced data in a problem of gamma-ray astronomy, Advances in Data Analysis and Classification 8 (2) (2014) 195.216.

[30] T. Bartz-Beielstein, C. Lasarczyk, M. Preuß, Sequential parameter optimization, in: B. McKay, et al. (Eds.), Proceedings 2005 Congress on Evolutionary Computation (CEC'05), Edinburgh, Scotland, Vol. 1, IEEE Press, Piscataway NJ, 2005, pp. 773–780.

[31] V. Picheny, T. Wagner, D. Ginsbourger, A benchmark of kriging-based infill criteria for noisy optimization, Structural and Multidisciplinary Optimization 48 (3) (2013) 607–626.

[32] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global Optimization 13 (4) (1998) 455–492.

[33] N. Bauer, K. Friedrichs, C. Weihs, Model based optimization of music onset detection, institution=TU Dortmund, number = 47/15, year=2015, Tech. rep.

[34] M. D. McKay, R. J. Beckman, W. J. Conover, Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (2) (1979) 55–61.

[35] D. R. Jones, A taxonomy of global optimization methods based on response surfaces, Journal of Global Optimization 21 (4) (2001) 345–383.

[36] B. Bischl, J. Bossek, D. Horn, M. Lang, mlrMBO: Model-Based Optimization for mlr, R package version 1.0 (2015).
URL https://github.com/berndbischl/mlrMBO

[37] S. Siegel, N. Castellan, Nonparametric statistics for the behavioral sciences, 2nd Edition, McGraw–Hill, Inc., 1988.

[38] B. Bischl, M. Lang, O. Mersmann, J. Rahnenführer, C. Weihs, BatchJobs and BatchExperiments: Abstraction mechanisms for using R in batch environments, Journal of Statistical Software 64 (11) (2015) 1–25.