Evolutionary Multi-Objective Training Set Selection of Data Instances and Augmentations for Vocal Detection

Igor Vatolkin[1][0000−0002−9454−9402] and Daniel Stoller[2][0000−0002−8615−4144]

[1] TU Dortmund
igor.vatolkin@tu-dortmund.de
[2] Queen Mary University of London
d.stoller@qmul.ac.uk

Abstract. The size of publicly available music data sets has grown significantly in recent years, which allows training better classification models. However, training on large data sets is time-intensive and cumbersome, and some training instances might be unrepresentative and thus hurt classification performance regardless of the used model. On the other hand, it is often beneficial to extend the original training data with augmentations, but only if they are carefully chosen. Therefore, identifying a "smart" selection of training instances should improve performance. In this paper, we introduce a novel, multi-objective framework for training set selection with the target to simultaneously minimise the number of training instances and the classification error. Experimentally, we apply our method to vocal activity detection on a multi-track database extended with various audio augmentations for accompaniment and vocals. Results show that our approach is very effective at reducing classification error on a separate validation set, and that the resulting training set selections either reduce classification error or require only a small fraction of training instances for comparable performance.

Keywords: Vocal detection · Evolutionary multi-objective training set selection · Data augmentation.

1 Introduction

The goal of music classification is to assign music data to categories such as music genres, emotions, harmonic properties, and instruments. To build a model for classification, a *training set* containing music examples with manually annotated labels is normally required. The size of this training set is crucial – if it is too small, overfitting occurs, and generally prediction performance tends to increase with more training data. While the available music data has grown in recent years (for example from 215 music tracks for genre recognition [10] to 106,574 pieces in the Free Music Archive [8]), labels are still often not available in great quantity, since annotation requires too much human effort. This is also the case for vocal activity detection [28].

Because of this label scarcity problem, *data augmentation* is often used to increase the number of annotated music examples [13]. In data augmentation, new training instances are created from existing ones by introducing variations, but keeping the already existing annotation. For example, annotated music tracks for genre classification can be amplified to different loudness levels, which should not affect the genre labelling, and all variants included in the training set. Due to its popularity, frameworks for music data augmentation such as the MATLAB-based *Audio Degradation Toolbox* [16] and the *muda* Python package for Musical Data Augmentation [18] were developed to allow for an easy application of various perturbations like adding noise or pitch shifting. However, data augmentation can also hurt performance when the wrong assumptions about the classification problem are made and as a result, the augmentation rules are set-up incorrectly [27].

The arrival of *big data* to music data analysis also exacerbated another problem: Complex classification models that work well with very large data sets, such as deep neural networks, are often hard to interpret for musicologists and users, who may wish to understand the defining properties of the different musical categories the model is operating with. Model prototyping also becomes more difficult, since feedback regarding model performance is obtained only infrequently due to long training times. Most importantly, corrupted or otherwise unrepresentative instances in the training set common in large datasets can unknowingly impact performance negatively, which are hard to find in these large datasets. Usually, models are trained on the full dataset, assuming instances are independent and identically distributed, so that the presence of such outlier instances effectively imposes an upper bound on the attainable performance, regardless of model choice.

Instead of training from more and more data regardless of its quality and applying a fixed set of augmentations, one may thus consider to focus on the identification of *smart data*, i.e., observations which are particularly useful to extract the most relevant properties of the target class. In this paper, we therefore propose a novel multi-objective evolutionary framework, which optimises the selection of the training instances and augmentations for a dataset to find the best solutions trading off the number of required training instances and the resulting classification performance.

As an application scenario to validate our approach, we have selected vocal detection as a prominent and well-researched task in music classification, and previous methods for the task are discussed in Section 2.1. Related work on training set selection is addressed in Section 2.2. In Section 3, we briefly introduce the backgrounds of multi-objective optimisation and outline our approach. Section 4 deals with the setup of our experimental study. The results are discussed in Section 5. We conclude with the most relevant observations and discussion of future work in Section 6.

2 Related Work

We review related work in the field of vocal activity detection and in evolutionary optimisation for training set selection in Sections 2.1 and 2.2.

2.1 Vocal Detection

Earlier approaches for singing voice detection mostly involve heavy feature engineering to enable classification [17, 25, 24], which yields moderate performance. However, attempting to further improve accuracy by refining the features suffers from diminishing returns, since it becomes harder to manually specify exactly which aspects of the audio data are relevant for classification.

More recently, approaches based on neural networks [27, 26] have been proposed that promise to reduce the required feature engineering and instead learn the relevant features directly from the data. While this avoids performance bottlenecks due to suboptimal feature design and can theoretically deliver high accuracy, it typically requires larger amounts of labelled data. Since publicly available, labelled data for singing voice detection are limited [28], ways to prevent overfitting were proposed alongside these models: Schlüter [26] applied a convolutional neural network (CNN) on weakly labelled audio excerpts in an attempt to extend the amount of usable data for training, obtaining improved performance compared to previous work, but also finding that the network also classifies simple sinusoids with pitch fluctuations as singing, which indicates the concept of a singing voice was not learned correctly. Similarly, a joint vocal separation and detection was employed by [28] aiming to exploit both singing voice separation datasets for vocal activity detection. Schlüter et. al [27] explored the benefit of different data augmentation techniques that among other aspects vary pitch and tempo of the audio excerpts. The results were mixed – some augmentations were helpful, but some were also detrimental to performance, suggesting that both data augmentation and feature extraction require prior knowledge to decide what the classifier outputs should be invariant to, and performance depends on the accuracy of this knowledge. Furthermore, all mentioned approaches can suffer from outliers in the training data. We thus aim to automate the selection of representative training instances as well as helpful data augmentations to increase vocal detection performance, which enables more robust classification models.

2.2 Multi-Objective Evolutionary Optimisation and Training Set Selection

Multi-objective optimisation evaluates solutions with regard to several optimisation criteria (see Section 3.1). In that case, many solutions become incomparable. As an example, consider a classification model which is fast but has a higher classification error, and another one which is slow, but has a lower error. The first one is better with regard to runtime, the second one with regard to classification error. However, it is still possible to create models which are both faster and have higher classification performance, but it becomes harder to identify the set of trade-off solutions. In this context, evolutionary algorithms (EA) [2] were considered [7]. EAs are often applied for such complex optimisation tasks, where a large search space makes it challenging to find a sufficiently good solution in acceptable time, and where other methods such as gradient descent can not be applied (e.g. for objective functions which are not differentiable or are multi-modal and have many local optima).

In music research, EAs were applied for instance for music composition [20] or feature selection [9]. A multi-objective EA with the target to minimise the number of selected features and the classification error was presented in [29]. EAs have proven their ability to generate new features for music classification by exploring nearly unlimited search spaces of combinations of different transforms and mathematical operations [22, 19, 15]. EAs have also been successfully applied for training set augmentation in bioinformatics, where the generation of new training data may be very expensive [11, 32], and for training set selection (TSS) [6]. In [1], TSS was explicitly formulated as a multi-objective problem of simultaneously maximising the classification accuracy and the reduction in training set size. Although training set selection was already applied for classification of acoustical events [21], we are not aware of any studies on multi-objective TSS for classification of music data.

3 Approach

Our approach consists of leveraging an evolutionary, multi-objective optimisation method, which is described in Section 3.1, and applying it to training set selection, as shown in Section 3.2.

3.1 Multi-Objective Evolutionary Optimisation

In the following, we formally introduce the problem of multi-objective optimisation with a focus on evolutionary methods. Let $\mathcal{X}$ be the decision or search space and $\mathbf{f} : \mathcal{X} \mapsto \mathbb{R}^d$ the vector-valued objective function with $d \geq 2$. $\mathcal{F} = \{f(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ is called the objective space. The goal of multi-objective optimisation is to simultaneously minimise all d dimensions $f_1, ..., f_d$ of $\mathbf{f}$ (we provide here a short definition without constraints, for the latter see [33]).

A solution $\mathbf{x}_1 \in \mathcal{X}$ dominates another solution $\mathbf{x}_2 \in \mathcal{X}$ (denoted with $\mathbf{x}_1 \prec \mathbf{x}_2$), iff[3]

$$\begin{aligned} \forall i \in \{1, ..., d\} : f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \text{ and} \\ \exists k \in \{1, ..., d\} : f_k(\mathbf{x}_1) < f_k(\mathbf{x}_2). \end{aligned} \tag{1}$$

Solutions $\mathbf{x}_1$ and $\mathbf{x}_2$ are incomparable, when neither $\mathbf{x}_1 \prec \mathbf{x}_2$ nor $\mathbf{x}_2 \prec \mathbf{x}_1$. Incomparable solutions, which are not dominated by any other solution found so far, are called the non-dominated front.

[3] For simplicity, we describe only the minimisation of objective functions, since maximisation can be achieved by minimising the function with its sign reversed.

When a multi-objective optimisation algorithm outputs the non-dominated front $\mathbf{x}_1, ..., \mathbf{x}_N$, it may be still required to evaluate solutions individually, for example, to select better solutions after evolutionary mutation. This can be done by means of the *dominated hypervolume*, or $\mathcal{S}$-metric [34], which is estimated as follows. Let $\mathbf{r} \in \mathcal{F}$ be a reference point in the objective space, which corresponds to the worst possible solution (e.g., 1 for the dimension which corresponds to the classification error). Then we define the dominated hypervolume as:

$$H(\mathbf{x}_1, ..., \mathbf{x}_N) = vol\left(\bigcup_{i=1}^{N} [\mathbf{x}_i, \mathbf{r}]\right), \tag{2}$$

where $vol(\cdot)$ describes the volume in $\mathbb{R}^d$, and $[\mathbf{x}_i, \mathbf{r}]$ the hypercube spanned between $\mathbf{x}_i$ and $\mathbf{r}$. Generally speaking, the dominated hypervolume of a given front consists of an infinite number of all theoretically possible solutions which are always dominated by solutions in this front. An individual hypervolume contribution of a solution $\mathbf{x}_i$ is then estimated as the dominated hypervolume of the front without this solution:

$$\Delta H(\mathbf{x}_i) = H(\mathbf{x}_1, .., \mathbf{x}_N) - H(\mathbf{x}_1, ..\mathbf{x}_{i-1}, \mathbf{x}_{i+1}, .., \mathbf{x}_N). \tag{3}$$

$\mathcal{S}$-metric selection evolutionary multi-objective algorithm (SMS-EMOA) [3] generates exactly one solution in each iteration. Then, all solutions are assigned to a hierarchy of non-dominated fronts, and the solution from the worst front with the smallest $\Delta H(\mathbf{x}_i)$ is removed from the population. For more details, we refer to [3].

3.2 Application to Training Set Selection

In this study, we apply SMS-EMOA to evolutionary multi-objective training set selection (EMO-TSS): We simultaneously minimise a) the ratio $f_{ir} := \frac{N}{I}$ between the number of selected instances N from the training set and its total number of instances I and b) the balanced classification error f_e achieved with a classification model trained on the given selection of instances which is computed on a separate validation set as the average of misclassification rates for positive and negative instances in binary classification. Each TSS solution is represented by a binary vector of length I called representation vector, where a one at the i-th position means that the i-th instance is used to train the classification model. Note this also supports optimising the selection of data augmentations when they are included as additional instances in the training set.

In the beginning of the evolutionary loop, the instances are selected for each individual with a given probability p. Using all instances ($p = 1$) would decrease the diversity of the initial population and thus increase the optimisation time. Additionally, selecting too few instances for each individual would increase the danger that “smart” instances not initially contained in the population are never found, so we set $p = 0.5$. To encourage the search for solutions with few training instances, we set a higher probability to remove ones than to add ones during mutation. For each position i of a new offspring solution, the i-th bit is flipped from 1 to 0 with probability w_1/N. The bit is flipped from 0 to 1 with probability $w_2 \cdot (w_1/N)$. We set these parameters so that solutions with only few training instances are explored more thoroughly (see Section 4.3).

4 Experimental Setup

In the following, we experimentally validate our proposed multi-objective training data selection algorithm for music classification tasks for the example of vocal activity detection.

4.1 Data Sets

We have selected MedleyDB [4] as a data set for vocal detection, as it contains multitrack audio recordings allowing for the application of individual data augmentation strategies for vocals and accompaniment. From the original 122 multi-track recordings, we removed 8 tracks due to source interference in the recordings[4].

To accelerate feature extraction and classification, we extracted 10 snippets[5] of 3 seconds duration from each track, equally distributed along the track length. Based on the instrument activation labels, three variants of each snippet were created that contain only vocals, only accompaniment, and a mix of both in case vocals and accompaniment were available, respectively. For each accompaniment snippet, we additionally created 4 versions whose signal amplitude was reduced to 80%, 60%, 40% and 20% and 8 further variants by applying the following degradations from the Audio Degradation Toolbox [16]: live recording, strong compression, vinyl recording, four noises with signal-to-noise ratio (SNR) of 20 dB (pink, blue, violet, white), and adding a sound ‘OldDustyRecording’ with a SNR of 10 dB. For vocal snippets, we also created 4 quieter, but no degraded versions, because we restricted the focus of this study to recognise “clean” vocals along varied accompaniment sounds. For audio snippets from time intervals which contained both accompaniment and vocals, we created mixes of the vocal snippet with the original accompaniment snippet and all augmented variants of accompaniment snippets. Together with non-mixed vocal and accompaniment snippets, this strategy leads to an overall number of 22,953 snippets with annotations. Naturally, the search space for EMO-TSS can be almost arbitrarily extended using further augmentations, but the above augmentations serve to demonstrate the potential of our approach.

[4] *Grants-PunchDrunk* contained vocals in the “sampler” stem, and for the following tracks, vocals could be heard in non-vocal stems: *ClaraBerryAndWooldog-TheBadGuys*, *Debussy-LenfantProdigue*, *Handel-TornamiAVagheggiar*, *Mozart-BesterJungling*, *Mozart-DiesBildnis*, *Schubert-Erstarrung*, *Schumann-Mignon*.

[5] As data *instances* are represented by audio *snippets* in our study, we use both terms synonymously throughout this paper.

In our experimental setup, we distinguish between three data sets. A *training* set is used by EMO-TSS to create a representation vector (see Section 3.1) and to choose the snippets used for training the vocal activity classifier. A *validation* set is used to evaluate solutions (trained classification models) created by EMO-TSS. A *test* set is used for the independent evaluation of the best solutions found after the evolutionary optimisation. From 114 MedleyDB tracks, one fourth was used as a training set, one half as a validation set, and one fourth as a test set. This partitioning was done three times similarly to a 3-fold cross-validation approach to estimate the variability of results depending on dataset selection.

As the goal of our training set selection was to identify the most relevant snippets and their augmentations for the detection of vocals in "usual" non-augmented recordings, augmented snippets were used only in the training set and not in validation and test sets. To evaluate the impact of the set of automatically selected data augmentations, we introduce two selections as a baseline for comparison: the first one ("pure snippets") uses all original snippets of the training set without augmentations, and the second one ("all snippets") uses all snippets including their augmented variants.

4.2 Features

As a first step before feature extraction, we convert the input audio to mono sampled at 22050 Hz.

As the first feature set, we use a vector of 13 Mel Frequency Cepstral Coefficients (MFCCs) [23], which were developed for speech recognition and should therefore facilitate vocal detection well. These features were computed individually for the middle of the attack interval, onset frame, and the middle of the release interval (extracted with an onset detection function from MIR Toolbox [12]), leading to a 39-dimensional feature vector for each snippet. We distinguish between these three time periods, because instrument and vocal timbre may significantly change during the time a note is played; for example, in [14] it was shown that non-harmonic properties in the attack phase may be very helpful to identify instruments, and the frames with a stable sound between two onsets performed well for genre recognition in [31].

As the second feature set, we use a Log-Mel-Spectrogram (LMS). We compute the magnitude spectrogram resulting from a 1024-point FFT with a hop size of 512, which is then converted to a Mel-frequency spectrogram using a Mel filterbank with 40 bands ranging from 60 to 8000 Hz. Finally, we normalise the features x by applying log-normalisation: $x \rightarrow \log(1 + x)$. To reduce feature dimensionality, we then downsample the spectrogram using mean-pooling across time frames with a non-overlapping window of size four, after discarding time frames at borders so that the number of time frames is divisible by four. The result is a $F \times T$ feature matrix with $F = 40$ frequencies and $T = 32$ time frames for a total of 1280 features.

We omitted the usage of larger audio feature sets, since they could lead to stronger overfitting to the validation set, especially when exploring extremely sparse solutions that use only a few instances for training. However, given a sufficiently large validation set and training time, we expect our method to work in these cases as well. Furthermore, instances important for vocal recognition performance should still be preferred over others rather independently of the feature set used.

4.3 Classification Models and Training Set Selection

We applied a random forest (RF) [5] as our classification method, as it is fast, has only few parameters, and is robust to overfitting, which keeps the meta-optimisation feasible with regards to computing time. For the same reasons, we refrained from the use of deep neural networks in this study.

However, we investigated to which extent the optimised training set selection obtained when using the simple RF classifier can also improve classification performance when it is used to train more complex classifiers. For this, we first optimised the training set selection using the RF classifier, serving as a fast surrogate model for a larger neural network we reimplemented from previous work for vocal detection [27]. This network is trained with the same settings on all, pure and the optimised EMO-TSS training set obtained using the RF, using the downsampled Log-Mel-Spectrogram features from Section 4.2.

However, we did not achieve improvements in classification performance compared to simply using pure snippets. We hypothesise that this is because of the drastically different approach neural networks employ for classification compared to RF, and that the EMO-TSS optimisation adapts closely to the behaviour of the used classification model. However, note that, in principle, our EMO-TSS approach can still be used to potentially improve the performance of this neural network by using it directly as part of the evolutionary meta-optimisation, although this significantly increases the required computing time. We leave investigations with larger models and suitable smaller surrogate models that behave sufficiently similar during evolutionary optimisation for future work.

For SMS-EMOA, we found the following parameters to work well after a preliminary study: initialisation with approximately one half of randomly selected instances, a population size of 40, $w_1 = 0.05$, and $w_2 = 0.2$. The number of generations was set to 3000, and each experiment was repeated 15 times for each fold. Please note that an exhaustive search for the optimal settings was not the target of this study and could be addressed in future work.

5 Results and Discussion

We analyse the results of training selection in Section 5.1 and investigate the properties of the chosen instances and augmentation strategies in Section 5.2.

5.1 Number of Selected Snippets and Classification Performance

Figure 1 plots the optimisation progress for the first cross-validation fold, averaged across 15 statistical experiment repetitions. In Fig. 1 (a), we observe a strong increase in hypervolume over 3000 generations for the validation set. This is due to starting with a rather "poor" solution with approximately half of all instances, so that there is enough room for improvement by removing many training instances. In Fig. 1 (b), the hypervolume for the test set also strongly increases, but remains below the hypervolume for the validation set because of higher test classification errors. In Fig. 1 (c), the progress of the lowest error $\widehat{f_e}$ among non-dominated solutions for the validation set is plotted, again averaged across 15 repetitions. Similar to the change of hypervolume with an increasing evaluation number, the optimisation progress is faster at the beginning and slows down significantly around the second third of evaluations. Fig. 1 (d) presents the progress of $\widehat{f_e}$ for the test set. As expected, the test error does not fall as rapidly as the validation error; the difference in values for the first generation is explained by a varying distribution of music tracks between the validation and test sets. Importantly, we do not observe significant overfitting on the validation set, which would be visible as an increasing test error.

Figure 2 shows the non-dominated front after all EMO-TSS experiments for fold 1. For the validation set, the solutions are marked with rectangles. Diamonds correspond to the same solutions evaluated for the test set: the ratio of selected instances remains the same, but the classification error increases slightly due to overfitting to the validation set. This increase is much more variable for smaller ratios, because the model parameters can then vary more strongly depending on the particular training set selection. Note that we assign a classification error of 1 to extremely sparse training set selections with only vocal or only non-vocal snippets and do not show them here. The solution with the smallest validation error $f_e = 0.2055$ contains 11.73% of all snippets (714 snippets), but when we further reduce the number of selected snippets to 434 snippets, f_e increases only moderately to 0.2125, indicating diminishing returns when adding more snippets and an effective optimisation on the validation set.

Table 1 lists the results after optimisation. To measure the effect of EMO-TSS, we compare the results to two baselines "pure snippets" and "all snippets", cf. Section 4.1. With "all snippets", the ratio of selected instances $\widehat{f_{ir}} = 1$, and with "pure snippets" $\widehat{f_{ir}}$ is equal to the number of non-augmented training snippets divided by the number of all snippets (including augmented ones). The complete training set with all augmentations contains 6088 snippets for fold 1, 5477 snippets for fold 2, and 5494 snippets for fold 3. We make several observations in the following.

Observation 1–comparison of baselines: Extension of pure snippets with all augmentations leads to an increase of the error in all cases: for both validation and test sets and both feature vectors. In other words, uncontrolled extension of the training set does not lead to an increase of classification performance, but even to its reduction. That not every augmentation is equally useful was also noted in work on manually selected augmentation strategies for singing voice detection [27] and demonstrates the data augmentation is not always straightforward to apply correctly.

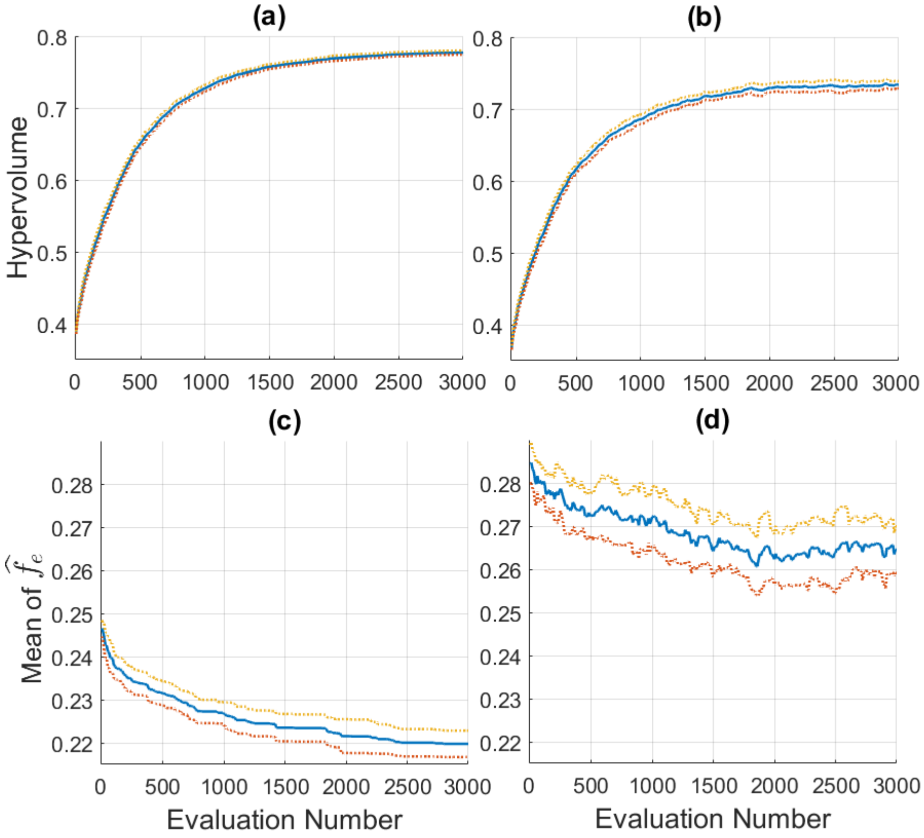

Fig. 1. Optimisation progress and 95% confidence intervals over 3000 evaluations for all statistical experiment repetitions: mean dominated hypervolume for the validation set (a) and the test set (b); mean of the smallest achieved error on the validation set (c) and the test set (d).

Observation 2–EMO-TSS reduces classification error or the number of selected instances: For all folds and features, our approach either improves the test performance $\widehat{f}_e(T)$ compared to baseline methods (e.g., $\widehat{f}_e(T) = 0.2667$ for the 1st fold using MFCCs, smaller than 0.2848 with pure snippets and 0.3489 with all snippets), or reaches similar performance, but using 39.5% less instances than the "Pure" baseline on average. This supports our initial hypothesis that many training instances can be unrepresentative or do not provide much new information to the classifier. A reduction in training instances can be beneficial for gaining an understanding of the classification problem and for example when using non-parametric models (e.g. k-nearest neighbour) whose storage space grows with the amount of training data.

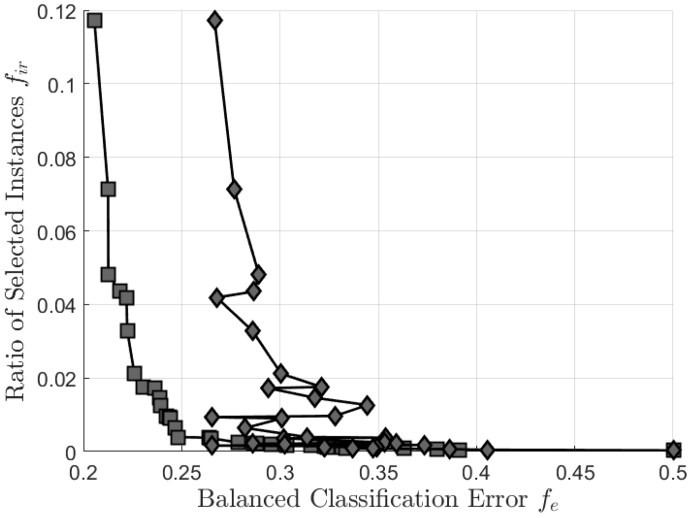

Fig. 2. Non-dominated front for the first fold, on the validation set (rectangles) and on the test set (diamonds).

Observation 3–comparison of validation errors: Using our proposed method, both the number of selected instances and the validation error is consistently reduced across all features and folds compared to the baseline methods, e.g., from {0.2411, 0.2843} to 0.2055 for fold 1 using MFCCs and from {0.1572, 0.1916} to 0.1334 using LMS. This means that EMO-TSS is very effective at optimising the training set selection to achieve good validation performances.

Observation 4–comparison of EMO-TSS validation and test errors: When applying TSS, the most critical risk is that its selection might lead to a model with good validation performance, but worse test set performance due to overfitting. Indeed, test errors sometimes increase slightly in comparison, but always remain approximately as good as the baseline test errors. To further reduce overfitting towards the validation set in the future, we suggest using a larger and more representative validation set by integrating multiple databases with vocal annotations.

Observation 5–comparison of $\widetilde{f}_e(T)$ after EMO-TSS to baselines: From all non-dominated solutions returned by EMO-TSS for an individual fold, we identified the one with the smallest test error $\widetilde{f}_e(T)$. Among all combinations of folds and features, this error is lower than the test error $\widetilde{f}_e(T)$ of the best baseline in almost all cases. This means that it is possible to obtain reduced test errors also after long-iterated TSS for the validation set. However, it is not possible to directly identify this "oracle" solution when the annotations of the test set are unseen, because the solution with the smallest test error is not the same as the solution with the smallest validation error, so this is a more theoretical result.

Table 1. Impact of our approach on classification performance. F corresponds to the fold, $\widehat{f}_{ir}$ to the ratio of selected instances for the non-dominated solution with the smallest classification error across all statistical repetitions, $\widehat{f}_e(V)$ and $\widehat{f}_e(T)$ to the classification error of the same solution on the validation and test set, and $\widetilde{f}_e(T)$ denotes the smallest test error across all final non-dominated solutions. Note that $\widehat{f}_e(T) = \widetilde{f}_e(T)$ for baseline methods "Pure" and "All" without training set selection. The best validation and test errors for each fold are marked with bold font.

F	Features	Snippets	$\widehat{f}_{ir}$	$\widehat{f}_e(V)$	$\widehat{f}_e(T)$	$\widetilde{f}_e(T)$
1	MFCCs	Pure	0.0907	0.2411	0.2848	0.2848
		All	1.0000	0.2843	0.3489	0.3489
		EMO-TSS	0.1173	0.2055	0.2667	0.2433
	LMS	Pure	0.0907	0.1572	0.1322	0.1322
		All	1.0000	0.1916	0.1903	0.1903
		EMO-TSS	0.0866	**0.1334**	0.1679	**0.1281**
2	MFCCs	Pure	0.0893	0.2752	0.3265	0.3265
		All	1.0000	0.2867	0.3297	0.3297
		EMO-TSS	0.0416	0.2127	0.3166	0.2959
	LMS	Pure	0.0893	0.1860	**0.1971**	**0.1971**
		All	1.0000	0.2609	0.3028	0.3028
		EMO-TSS	0.0310	**0.1339**	0.2199	0.2023
3	MFCCs	Pure	0.0885	0.2937	0.2592	0.2592
		All	1.0000	0.3068	0.2782	0.2782
		EMO-TSS	0.0220	0.2506	0.2700	0.2265
	LMS	Pure	0.0885	0.1642	0.1800	0.1800
		All	1.0000	0.2495	0.2049	0.2049
		EMO-TSS	0.0284	**0.1432**	0.1801	**0.1562**

Observation 6–comparison of folds: The errors are fairly different across all folds, for instance, the smallest test errors are achieved for fold 3. This is explained by a rather small size of MedleyDB: test sets contained snippets of 28 tracks (one fourth of 114 tracks).

5.2 Properties of Selected Instances

A further analysis may be done with regards to categories of selected snippets in solutions with the smallest classification errors. Can we recommend applying some particular augmentations generally, not just to specific instances? For this we use a measure describing the "popularity" of a set of snippets, obtained by dividing the frequency of their occurrence in the optimised training set selection by the frequency of their occurrence in the full training set. Table 2 shows this measure for different snippet categories computed for solutions with the smallest f_e. For vocal snippets on fold 1 with MFCCs as an example, this yields $1.028 \approx \frac{0.1149}{0.1117}$, because the proportion of vocal snippets is 0.1117 in the complete training set (680 of 6088 snippets) and 0.1149 for the training set selection with the smallest $f_e(V)$ in the first fold (82 out of 714 snippets). Numbers greater than 1

Table 2. Share of different snippet categories in relation to the share of these snippets in the complete training set for the three folds.

	MFCCs			LMS		
Snippet category	F=1	F=2	F=3	F=1	F=2	F=3
Main categories						
Vocals	1.028	1.179	1.234	1.087	1.640	1.208
Accompaniment	0.998	0.907	0.836	0.930	0.773	0.826
Mix	0.994	1.183	1.356	1.111	1.363	1.394
Applied augmentations						
Ampl. reduct. 80%	1.081	1.081	1.215	0.879	0.922	0.951
Ampl. reduct. 60%	1.128	0.884	0.747	0.858	0.857	1.207
Ampl. reduct. 40%	0.958	1.130	1.215	1.130	1.384	0.878
Ampl. reduct. 20%	1.020	1.032	0.654	1.109	0.857	1.427
Live	1.004	0.634	0.830	0.889	0.935	0.789
Compression	0.963	0.951	0.356	0.972	1.360	1.439
Vinyl recording	0.922	1.078	1.067	0.972	0.850	0.975
Noise: pink	1.127	0.761	1.186	0.916	0.680	0.418
Noise: blue	1.086	0.761	1.067	1.055	0.935	1.021
Noise: violet	0.963	1.141	1.067	0.916	0.340	0.882
Noise: white	0.840	0.824	0.474	0.972	1.615	0.696
Old dusty	0.943	1.014	1.423	1.277	0.850	1.300

show that the proportion of a given snippet category is larger in the optimised set than in the complete set, indicating its importance for high performance. Numbers lower than 1 suggest a lower importance.

Results show that vocal snippets are rather relevant, with their measures above 1 for all folds and both features. This reveals it is helpful to train vocal detection models not only with music mixtures, but also solo vocal tracks as positive examples. However, the numbers are not significantly higher than 1 for some folds. Generally, no category of snippets appears to be very important or very harmful to include; all categories contribute to models with the lowest error to some extent. Interestingly, the effect of augmentations seems to be somewhat dependent on the used features: for MFCCs, the most valuable augmentation seems to be the reduction of signal amplitude to 80% (ratios greater than 1 for all folds), while for LMS it appears harmful (ratios below 1 for all folds). Some values vary strongly across the folds. For instance, white noise degradation is less important for all folds using MFCCs and two of three LMS folds, but has a rather high value of 1.615 for the second LMS fold. Further investigations with larger data sets and also tracks from different music genres are necessary to potentially provide more conclusive findings about the relevance of individual augmentation methods.

6 Conclusions

In this work, we have proposed a multi-objective evolutionary framework for training set selection, which simultaneously minimises the number of training instances and the classification error. This approach was applied for the problem of vocal detection, together with training set augmentation by means of loudness reduction and various audio degradations. The results show that, compared to classification with a complete training set, it is possible to strongly reduce the training set size while approximately maintaining the classification performance. Using our optimised selection of training instances and augmentations, we obtain a strong performance increase on the validation set compared to using all or no augmentations for training, which mostly translates to an independent test set, albeit sometimes to a lesser degree due to over-optimisation on the validation set.

To improve the generalisation performance and make validation sets more representative, one can integrate further databases with vocal annotations in our proposed framework. It would be interesting to explore the performance of our proposed training set selection when using more feature sets, classification methods and augmentation methods and other parameters for the tested augmentation methods, and even applying it to other tasks in music information retrieval. Also, we may compare the performance of evolutionary optimisation to other training set selection techniques, like clustering or *n*-gram statistics [30].

Finally, for the application to larger classification models such as neural networks, it appears promising to investigate the use of surrogate classifier models that are fast to train and behave similarly to the large model of interest, so that the computation time for evolutionary optimisation remains feasible, and the optimal training set selection found for the surrogate model also helps the performance for the large classification model.

Acknowledgements

This work was funded by the DFG (German Research Foundation, project 336599081) and by EPSRC grant EP/L01632X/1.

References

1. Acampora, G., Herrera, F., Tortora, G., Vitiello, A.: A multi-objective evolutionary approach to training set selection for support vector machine. Knowledge-Based Systems **147**, 94–108 (2018)
2. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York (1996)
3. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research **181**(3), 1653–1669 (2007)

4. Bittner, R.M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., Bello, J.P.: Medleydb: A multitrack dataset for annotation-intensive MIR research. In: Proc. of the 15th Int'l Society for Music Information Retrieval Conf. (ISMIR). pp. 155–160 (2014)
5. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
6. Cano, J.R., Herrera, F., Lozano, M.: Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability. Data and Knowledge Engineering **60**(1), 90–108 (2007)
7. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, New York (2007)
8. Defferrard, M., Benzi, K., Vandergheynst, P., Bresson, X.: FMA: A dataset for music analysis. In: Proc. of the 18th Int'l Society for Music Information Retrieval Conf. (ISMIR). pp. 316–323 (2017)
9. Fujinaga, I.: Machine recognition of timbre using steady-state tone of acoustic musical instruments. In: Proc. of the Int'l Computer Music Conf. (ICMC). pp. 207–210 (1998)
10. Goto, M., Nishimura, T.: RWC music database: Popular, classical, and jazz music databases. In: Proc. of the 3rd Int'l Conf. on Music Information Retrieval (ISMIR). pp. 287–288 (2002)
11. Kumar, A., Cowen, L.: Augmented training of hidden markov models to recognize remote homologs via simulated evolution. Bioinformatics **25**(13), 1602–1608 (2009)
12. Lartillot, O., Toiviainen, P.: MIR in Matlab (II): A toolbox for musical feature extraction from audio. In: Proc. of the 8th Int'l Conf. on Music Information Retrieval (ISMIR). pp. 127–130 (2007)
13. Lemley, J., Bazrafkan, S., Corcoran, P.: Smart augmentation learning an optimal data augmentation strategy. IEEE Access **5**, 5858–5869 (2017)
14. Livshin, A., Rodet, X.: The significance of the non-harmonic "noise" versus the harmonic series for musical instrument recognition. In: Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR). pp. 95–100 (2006)
15. Mäkinen, T., Kiranyaz, S., Pulkkinen, J., Gabbouj, M.: Evolutionary feature generation for content-based audio classification and retrieval. In: Proc. of the 20th European Signal Processing Conf. (EUSIPCO). pp. 1474–1478 (2012)
16. Mauch, M., Ewert, S.: The audio degradation toolbox and its application to robustness evaluation. In: Proc. of the 14th Int'l Society for Music Information Retrieval Conf. (ISMIR). pp. 83–88 (2013)
17. Mauch, M., Fujihara, H., Yoshii, K., Goto, M.: Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In: Proc. of the 12th Int'l Society for Music Information Retrieval Conf. (ISMIR). pp. 233–238 (2011)
18. McFee, B., Humphrey, E.J., Bello, J.P.: A software framework for musical data augmentation. In: Proc. of the 16th Int'l Society for Music Information Retrieval Conf. (ISMIR). pp. 248–254 (2015)
19. Mierswa, I., Morik, K.: Automatic feature extraction for classifying audio data. Machine Learning Journal **58**(2-3), 127–149 (2005)
20. Miranda, E.R., Biles, J.A.: Evolutionary Computer Music. Springer, New York (2007)
21. Mun, S., Park, S., Han, D.K., Ko, H.: Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane. In: Proc. of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017) (November 2017)

22. Pachet, F., Zils, A.: Evolving automatically high-level music descriptors from acoustic signals. In: Proc. of the 1st Int'l Symp. on Computer Music Modeling and Retrieval (CMMR). Lecture Notes in Computer Science, vol. 2771, pp. 42–53. Springer (2003)
23. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice Hall, Upper Saddle River (1993)
24. Rao, V., Gupta, C., Rao, P.: Context-aware features for singing voice detection in polyphonic music. In: Adaptive Multimedia Retrieval. Large-Scale Multimedia Retrieval and Evaluation, pp. 43–57. Springer (2011)
25. Regnier, L., Peeters, G.: Singing voice detection in music tracks using direct voice vibrato detection. In: Proc. of the IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 1685–1688. IEEE (2009)
26. Schlüter, J.: Learning to pinpoint singing voice from weakly labeled examples. In: Proc. of the Int'l Society for Music Information Retrieval Conf. (ISMIR). pp. 44–50 (2016)
27. Schlüter, J., Grill, T.: Exploring data augmentation for improved singing voice detection with neural networks. In: Proc. of the 16th Int'l Society for Music Information Retrieval Conf. (ISMIR). pp. 121–126 (2015)
28. Stoller, D., Ewert, S., Dixon, S.: Jointly detecting and separating singing voice: A multi-task approach. In: Deville, Y., Gannot, S., Mason, R., Plumbley, M.D., Ward, D. (eds.) Latent Variable Analysis and Signal Separation. pp. 329–339 (2018)
29. Vatolkin, I., Preuß, M., Rudolph, G.: Multi-objective feature selection in music genre and style recognition tasks. In: Proc. of the 13th Annual Genetic and Evolutionary Computation Conf. (GECCO). pp. 411–418. ACM Press (2011)
30. Vatolkin, I., Preuß, M., Rudolph, G.: Training set reduction based on 2-gram feature statistics for music genre recognition. Tech. Rep. TR13-2-001, Faculty of Computer Science, Technische Universität Dortmund (2013)
31. Vatolkin, I., Theimer, W., Botteck, M.: Partition based feature processing for improved music classification. In: Gaul, W.A., Geyer-Schulz, A., Schmidt-Thieme, L., Kunze, J. (eds.) Proceedings of the 34th Annual Conference of the German Classification Society (GfKl), 2010. pp. 411–419. Studies in Classification, Data Analysis, and Knowledge Organization, Springer, Berlin Heidelberg (2012)
32. Velasco, J.M., Garnica, O., Contador, S., Lanchares, J., Maqueda, E., Botella, M., Hidalgo, J.I.: Data augmentation and evolutionary algorithms to improve the prediction of blood glucose levels in scarcity of training data. In: Proc. of the 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 2193–2200. IEEE (2017)
33. Zitzler, E.: Evolutionary multiobjective optimization. In: Rozenberg, G., Bäck, T., Kok, J.N. (eds.) Handbook of Natural Computing, Volume 2, pp. 871–904. Springer, Berlin Heidelberg (2012)
34. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - a comparative case study. In: Proc. of the 5th Int'l Conf. on Parallel Problem Solving from Nature (PPSN). vol. 1498, pp. 292–304. Springer (1998)