Soft Comput (2012) 16:2027–2047
DOI 10.1007/s00500-012-0874-9

FOCUS

Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures

Igor Vatolkin · Mike Preuß · Günter Rudolph · Markus Eichhoff · Claus Weihs

Published online: 3 July 2012

Abstract Instrument recognition is one of the music information retrieval research topics. This task becomes very challenging if several instruments are played simultaneously because of their varying physical characteristics: inharmonic attack noise, energy development during attack–decay–sustain–release envelope or overtone distribution. In our framework, we treat instrument detection as a machine-learning task based on a large amount of preprocessed audio features with target to build classification models. Since classification algorithms are very sensitive to feature input and the optimal feature set differs from instrument to instrument, we propose to run a multi-objective feature selection procedure before building of classification models. Two objectives are considered for evaluation: classification mean-squared error and feature rate (smaller amount of features stands for reduced costs and decreased risk of overfitting). The analysis of the extensive experimental study confirms that application of an evolutionary multi-objective algorithm is a good choice to optimize feature selection for music instrument identification.

Keywords Multi-objective feature selection · Music classification · Instrument recognition

1 Introduction

1.1 Music information retrieval and classification

Music information retrieval (MIR) is a broad research domain which deals with automatic analysis of music data and incorporates interdisciplinary studies from computer science, statistics, signal processing, data mining, music cognition and psychology (Ras 2010; Li et al. 2011). The most prominent MIR applications are music transcription, genre and artist recognition, similarity analysis and identification of high-level characteristics such as instruments, harmony or melody. Especially, the latter task can provide valuable information describing the given music and can be integrated as intermediate step into efficient organization of large music collections with limited resources such as on mobile devices (Blume et al. 2011).

Different data sources provide specific advantages and restrictions: e.g., harmonical features can be directly derived from score or MIDI data; but this representation is not always available and cannot capture the performance characteristics of a certain artist or the applied digital effects. Whereas this information is available from audio for music listeners, it is often very hard to extract it algorithmically, in particular, if there is a large number of instruments are played simultaneously.

A large part of MIR research deals with classification, either identifying high-level features, or structuring music

I. Vatolkin (✉) · M. Preuß · G. Rudolph
Fakultät für Informatik, Technische Universität Dortmund,
Otto-Hahn-Str. 14, 44227 Dortmund, Germany
e-mail: igor.vatolkin@tu-dortmund.de

M. Preuß
e-mail: mike.preuss@tu-dortmund.de

G. Rudolph
e-mail: guenter.rudolph@tu-dortmund.de

M. Eichhoff · C. Weihs
Fakultät für Statistik, Technische Universität Dortmund,
44221 Dortmund, Germany
e-mail: markus.eichhoff@tu-dortmund.de

C. Weihs
e-mail: claus.weihs@tu-dortmund.de

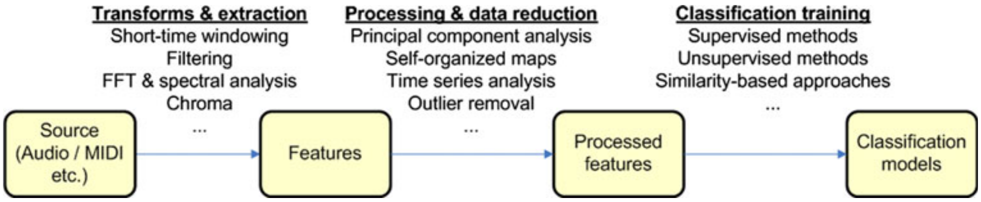

Fig. 1 Algorithm chain in music classification, from (Vatolkin et al. 2011)

with respect to genres and personal preferences (Fu et al. 2011; Blume et al. 2011; Eronen 2009; Weihs et al. 2007; Ahrendt 2006). The common steps of this approach are outlined in Fig. 1. At first, the raw data must be converted to some numerical characteristics or features. The second step is to preprocess them; typical steps are normalization, statistical feature reduction or time series analysis. Finally, the classification models can be trained to predict labels from features.

Each step of this classification chain can be optimized. However it can be argued, that the design of meaningful features should be preferred to the application of very complex classification methods (Mierswa and Morik 2005). In the recent past, many efforts have been invested in the development of such extended descriptors: examples are automatic feature construction by genetic trees (Mierswa and Morik 2005), enhanced harmonic analysis (Müller and Ewert 2010) or cepstral modulation characteristics (Nagathil et al. 2011).

1.2 Instrument recognition

The robust recognition of instruments can be treated also as an intermediate step enabling further more complex and important applications in music analysis, such as automatic correction of slightly misplaced notes and vocals or identification of wrongly tuned instruments. The better understanding of the instrument role for a certain music genre or composer also is a useful target, and such information cannot be always extracted from the score if digital effects have changed the instrument sounds or specific vocal techniques are assembled (rap vocals in hiphop, guttural singing in death metal, etc.). Moreover, the not yet completely solved automatic transcription from audio to score is hardly possible without source separation which itself may be improved by knowledge of the playing instruments.

The intensive research on automatic instrument recognition from audio is carried out since one decade or so, and many promising approaches and corresponding publications were investigated in the recent past. The complexity of categorization tasks ranges from rather simple classification of singular and very distinctive instrument tones (where different instrument characteristics can be captured by small amount of relevant features) to instrument identification in polyphonic recordings, where the spectrum is built by a mixture of overtones, non-harmonic components (such as violin bow strike or piano attack sound) and formant distributions (frequencies which are strongly resonated by instrument corpus).

One of the first publications with a deeper analysis of several features from time, spectral and cepstral domain for the recognition of woodwind tones is (Brown et al. 2001). In the work of Eronen (2003), an improved feature processing using independent component analysis (ICA) was applied for another classification of singular instrument tones. In Livshin (2006) among others, the features corresponding to non-harmonic components of instrument tones were successfully applied to instrument classification. Eronen (2009) provides a further list of publications.

Other studies deal with instrument recognition in polyphonic recordings. For example, in Essid et al. (2006), the mixtures of up to four instruments are categorized by hierarchical classification. The system benefits from the availability of domain knowledge about the processed music genre. Another interesting proposal was investigated in Kitahara et al. (2007): the robustness of audio features was measured depending on how much they had been affected by the overlapping sound of different simultaneously played instruments. An evolutionary-based learning for feature estimation was developed in Kobayashi (2009). Advanced signal separation by non-negative matrix factorization (NMF) was applied to instrument detection in Heittola et al. (2009).

1.3 Study outline

The number of existing musical instruments is very high and they have very distinctive characteristics (strings, electric and acoustic guitars, drums and percussions, piano and organ, etc.) Another quantity of instruments is synthesized completely in a digital way, and the combination number of possible effects and amplifiers, e.g., only for electric guitar is nearly infinite. Due to such variability, it is obvious that audio features which help to recognize a certain instrument may completely fail to recognize another one; since many features do not directly imply some characteristic of music theory, it is also hard for music experts to select the optimal features for identification of a concrete instrument. Therefore, automatic feature selection provides a reasonable possibility to find out the most representative features for a certain classification task.

The aim of our study was to investigate how well feature-selection may help to find the most significant features and improve the classification quality for each of the several instrument identification tasks. Further tasks were to compare several classification methods and the different settings of an optimization algorithm with regard to performance minimizing two metrics.

For this goal, we created two data sets of randomly mixed intervals (two instrument sample tones played simultaneously) and chords (with three or four tones). On the other side a large amount of different audio descriptors was available to map the feature values to binary instrument labels such as ‘chord with one or more piano tones’ or ‘chord with none of piano tones’. Because many evaluation criteria for classification performance are existing and the area of multi-objective feature selection is not well examined, especially for music data analysis, we selected two conflicting goals to be optimized simultaneously: the classification error and the cardinality of the selected feature set.

In the next chapter, we introduce the formal problem of feature selection and its multi-objective extension. The optimization algorithm is explained in detail before we describe our data sets and list the parameters of the study. In the subsequent sections different results of our study are discussed: From the necessity of feature selection in general to the comparison of classification algorithms and parameters of the optimizer itself. Finally, we discuss the major conclusions derived from the experimental results and provide directions for future research.

2 Algorithmic background

2.1 Feature selection

In general, feature or variable selection (FS) can be described as defined in Guyon et al. (2006) (we replaced arg max from the original terminology to arg min, since we *minimize* both the error and feature set size in our experiments described later):

$$\theta^* = \arg\min_{\theta}[I(Y;\Phi(X,\theta))], X \in \mathbb{R}^d. \qquad (1)$$

Here X is the original feature set, Φ the selected feature set, θ denotes the indices of the selected features and Y is the target variable, i.e., the category to identify. I is a function which is responsible for the relevance between Φ and Y. In other words, the target of feature selection is to provide the most relevant feature set (e.g., with the largest accuracy). Several benefits (sometimes conflicting!) can be considered during the reduction of the feature number by a FS process compared to the original set X:

- Reduced classification error: too many noisy or meaningless features often reduce the classification performance, as stated also in our experiments below.
- Reduced storage cost: for a certain classification task it is not required any longer to save and extract all available features.
- Reduced computation time: building a categorization model from a smaller amount of features can be done faster, especially for complex methods like Support Vector Machines (SVM), which increase feature dimensionality in search for linear separation.
- Reduced risk of overfitting: Using a very large feature number for not enough labeling instances may lead to the situation, where some of the features can be mistakenly interpreted by a classifier as relevant.

One possible categorization of the numerous feature selection algorithms is to distinguish between deterministic methods (which provide the same result for each repetition) and heuristics, which may incorporate some stochastics. To name just a few, one of the most straightforward ways to find the (sub)optimal feature number is to run forward or backward selection: start with the empty feature set, then add the most relevant features one by one due to some criterion (e.g., information gain or classification error) or remove the most irrelevant features from the complete feature set. Measuring the correlation between features can be also promising (Hall 1999). An expanded list, especially of deterministic methods, is provided in Guyon et al. (2006). For complex classification tasks, the number of possible feature subsets is typically very high, and it cannot always be directly measured that which combinations are the most promising. In that case, stochastic heuristics come into play, in particular, Evolutionary Algorithms (EA) (Rudolph 2012). The application of different enhanced EA for FS is discussed in Zhu et al. (2010). One of the first and altogether rather few applications of feature selection by EA in music classification was introduced in Fujinaga (1998). In our previous work, we have also successfully applied several hybrid EA for feature selection in music genre and style classification (Vatolkin et al. 2009; Bischl et al. 2010).

2.2 Multi-objective feature selection

The more interesting and less-investigated way to run feature selection is to do it in an explicitly multi-objective way, i.e., optimizing different conflicting targets. In this case, Eq. 1 has to be extended via

$$\theta^* = \arg\min_{\theta}[I_1(Y;\Phi(X,\theta)),\ldots,I_O(Y;\Phi(X,\theta))], \tag{2}$$

where $X \in \mathbb{R}^d$ and $O \geq 2$ denoted the number of different objectives to be optimized (minimized) simultaneously.

For multi-objective FS, it is not possible any longer to detect the one and only best feature set, since some of these sets cannot be compared directly. The concept of 'dominance' is useful in this situation: a solution $\mathbf{x}'$ *dominates* another solution $\mathbf{x}''$ if it is not worse in all objectives and better with regard to at least one objective function. Formally,

$$\begin{aligned}\mathbf{x}' \prec \mathbf{x}'' \quad &\text{if} \quad \forall j \in \{1,\ldots,O\} : I_j(\mathbf{x}') \leq I_j(\mathbf{x}'')\\ &\text{and} \quad \exists k \in \{1,\ldots,O\} : I_k(\mathbf{x}') < I_k(\mathbf{x}'')\end{aligned} \tag{3}$$

The *Pareto-front* corresponds to the best tradeoff solutions: they are not dominated by any other solution:

$$\mathbf{x} \in \mathcal{P}_f \quad \text{if} \quad \nexists\, \mathbf{x}' : \mathbf{x} \prec \mathbf{x}' \tag{4}$$

For music feature selection in genre and style classification, we proposed a set of objective categories, which are often conflicting (Vatolkin et al. 2011):

- Common quality-based metrics are based on confusion matrix data and are used very often: accuracy, precision, recall, etc. Some of the metrics in this group can be conflicting (Vatolkin 2012); in Vatolkin et al. (2011), the large Pareto-fronts maximizing recall and specificity are given.
- Specific quality-based metrics are dependent on the classification task, such as evaluation of recognized music segment boundaries (Lukashevich 2008).
- Resource metrics measure the demands on runtime and storage.
- Model complexity responds to the tradeoff between complex and possibly highly overfitted models against rather compact, robust models which may provide higher classification errors.
- User interaction metrics describe personal efforts for classification experiments, personal satisfaction and other user-related statistics (Liu 2010).

For different classification tasks, some combinations of conflicting metrics to be optimized simultaneously can be reasonable, e.g., little user impact on a definition of personal music category against classification quality, or maximization of performance both on positive and negative examples for highly unbalanced data. For our study, we decided to minimize the classification error and the selected feature rate, since the latter metric gives a good approximation of several important facets: smaller feature sets reduce the costs for feature storage and computing time; the danger of model overfitting decreases, and the model complexity may (but does not have to!) be reduced as well.

To our knowledge, no previous work applying the multi-objective feature selection by EA has been previously applied for instrument recognition. In our opinion, stochastic heuristics are a very good choice, since the number of possible instruments that are to be recognized is almost endless (and can be uncountable, if further digital effects and signal processing are applied as discussed above). Therefore, it is not possible to select a limited feature set which is sufficient for different instrument recognition tasks. Manual low-level construction of well-suited features can be, in fact, optimal for a concrete instrument, but requires very large efforts and exact knowledge of instrument characteristics. Starting with a large initial feature set boosts the possibility to have representative features for a certain task after selection procedure, and simultaneous minimization of the feature number cares for keeping the overall number of selected features small. This method can be run completely automatically without any human interaction optimizing the recognition of any possible instrument in polyphonic or monophonic recordings; after the initial feature set is integrated, only the labeling of ground truth is required, and this can be done automatically during random generation of instrument mixtures as described in Sect. 3.1.

2.3 Optimization algorithm

Many methods are available for optimization, e.g., grid search or gradient descent (Snyman 2005). However, if we deal with FS starting with a large feature set X, the number of possible solutions is very high ($2^{|X|}$) and it is not always possible to find a good set of features in a deterministic way. EA (Rudolph 2012) which incorporate randomized techniques inspired by natural evolution are well suited for solving such hard optimization tasks. The general steps of the population-based evolutionary process are sketched in Fig. 2.

At first, a set of μ problem solutions is created, often referred to as *population* of *individuals*. Then some of the solutions are selected for breeding. After the application of a crossover operator, the λ offspring individuals are created, with the aim to combine the beneficial characteristics from the parents. A mutation operator adds randomness to this process, helping to get out of the possible local optima. Finally, the next parent population is built with regard to the metric (*fitness function*) of each individual. Two basic principles are to replace always the parent population by the offspring population ((μ, λ)-EA) or to select the best λ solutions from the combined population ((μ + λ)-EA).

The application of EA for multi-objective optimization is even more promising (Coello et al. 2006). One reason to

Fig. 2 Evolutionary process

```
create $\mu$ initial parents at random and evaluate them
while stopping criterion not fulfilled do
    generate $\lambda$ offspring from $\mu$ parents by variation
    evaluate the offspring
    select $\mu$ new parents from offspring (+ parents)
end while
```

use heuristics is that the direct comparison of solutions is not always possible; many feature subsets may correspond to different tradeoffs between objectives. Another issue is that the aim of multi-objective optimization is to present a set of solutions to the decision-maker with different balance between objectives, which can be well created by population-based methods. One of the rather limited applications of multi-objective FS with EA for several classification tasks is described in Reynolds et al. (2010).

We selected $\mathcal{S}$-Metric Selection Evolutionary Multi-Objective Algorithm (SMS-EMOA) (Beume et al. 2007) as an optimization heuristic. SMS-EMOA is an (30+1) evolutionary strategy which measures both the quality and diversity of the tradeoff solutions based on the *hypervolume* or $\mathcal{S}$-metric criterion (Zitzler et al. 2007) defined by

$$\mathcal{S}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \mathrm{vol}\left(\bigcup_{i=1}^{N} [\mathbf{x}_i, \mathbf{r}]\right) \tag{5}$$

where $[\mathbf{x}_i, \mathbf{r}]$ corresponds to the hypercube between the solution $\mathbf{x}_i$ and the reference point $\mathbf{r}$ which may be set to the worst possible solution w.r.t. to all metrics. For the quality assessment of a single solution, its contribution to the $\mathcal{S}$-metric can be determined via

$$\Delta\mathcal{S}(\mathbf{x}_i) = \mathcal{S}(\mathbf{x}_1, \ldots, \mathbf{x}_N) - \mathcal{S}(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_N). \tag{6}$$

After the generation of an offspring, SMS-EMOA sorts the population into several solution fronts by fast non-dominated sorting (Deb 2001), i.e., no element of the first front of solutions is dominated by any other solution; the second front of solutions are dominated only by the first front of solutions, etc. The solution with the smallest $\Delta\mathcal{S}(\mathbf{x}_i)$ from the last front is removed from the population.

We adapted the original SMS-EMOA to FS by the integration of specific operators. As a mutation operator we used the asymmetric bit flip, so that the mutation probability of flipping 0 to 1 was:

$$p_m(i) = \frac{\gamma}{N} |m_i - p_{01}|,$$

where $N = 1{,}148$ is the total number of features, p_{01} is the general probability of switching bits on, γ the mutation step size and m_i corresponds to the ith feature selection bit (1 if the feature was selected). The probability of adding features was therewith reduced to prefer smaller feature sets.

Three different crossover possibilities were integrated: no crossover, Uniform Crossover (UC, from the two parents for each bit a parent was selected randomly with equal probability) and Commonality-Based Crossover (CBC) as introduced in Emmanouilidis et al. (2000) which should preserve building blocks in feature selection. Here, the shared bits of both parents are inherited by an offspring; the non-shared bits are inherited by an offspring from the parent i with probability

$$p_c(i) = \frac{n_i - n_c}{n_u},$$

where n_i is the number of selected bits of the ith parent, n_c is the number of shared bits equal to 1 (features to select) and n_u the number of non-shared bits equal to 1 for both parents.

Another parameter to examine was the initial feature rate if_r which corresponds to the probability of drawing 1 in starting solutions.

3 Experiment setup

3.1 Classification problems

For instrument recognition in polyphonic recordings, we created a randomly distributed set of 3,000 audio intervals and 3,000 chords. The probabilities to draw a mixture were set as follows. The first tone was randomly selected from the large collection of single instrument tones from McGill University collection[1], RWC database (Goto et al. 2003) and University of Iowa instrument samples[2]. Then for intervals, there was an equal chance of drawing the second tone as either minor third, major third, fourth, fifth, major sixth or minor seventh. For chords, after random selection of a key and tonality, there was an equal chance of drawing tonic, subdominant, dominant and submediant. Subdominant was represented by four tones, and all other

[1] http://www.music.mcgill.ca/resources/mums/html.

[2] http://www.theremin.music.uiowa.edu.

chords by three tones. The instruments for mixtures were drawn completely randomly, so it was, e.g., possible to create a chord of four piano tones, or a mix of piano, guitar, violin and trumpet. The instruments were: several different pianos, guitars (electric and acoustic), strings (violin, viola and cello) and wind (flute and trumpet). The corresponding binary instrument recognition tasks were to identify an instrument group (e.g., guitars) in an interval or chord.

3.2 Algorithm parameters

For experiments in classification and feature selection, we integrated an overall number of 1,148 different up-to-date audio features representing time, spectral (among others from mel, ERB and bark scales), cepstral and phase domain characteristics, such as energy in different sub-bands, spectral peak distribution, chroma and tonal centroid vector, linear prediction and mel frequency cepstral coefficients, etc. Most of the features are described in our technical report (Theimer et al. 2008) and the manual of MIR Toolbox (Lartillot and Toiviainen 2007). All features are available in AMUSE framework (Vatolkin et al. 2010) which provides interfaces to further feature extraction tools: MIR Toolbox, jAudio (McEnnis et al. 2006), Matlab, RapidMiner (Mierswa et al. 2006), Chroma Toolbox (Müller and Ewert 2011), etc.

As a preprocessing step, we estimated the attack–onset–release (AOR) envelope which is the approximation of the attack–decay–sustain–release stages (Park 2010): each played-note is characterized by the ATTACK phase with the increasing energy, then the DECAY phase with short energy decrease (the noisy components like bow or key stroke soften), the longer stable SUSTAIN phase and the RELEASE phase with the fading sound. The ONSET frame extracted by MIR Toolbox corresponds to the start of the sustain phase, whereas the ATTACK and the RELEASE intervals relate to any hearable components of the tone before and after the onset.

One part of the features was extracted only from the onset frame and the middle frames of attack and release intervals, so that these three extraction possibilities provided three different feature dimensions (e.g., saving 'energy from attack interval', 'onset energy' and 'energy from release interval'). Another part of features was constructed using several blocks of 4,096 samples across 1.3 s frame taking medians over the calculated characteristics in small original frames as introduced in (Eichhoff 2012).

For building of classification models, we selected four different classification methods with default parameters: Decision Tree C4.5, Random Forest (RF), Naive Bayes (NB) and Support Vector Machine (SVM) with linear kernel, all of them available in AMUSE as part of the WEKA library (Hall et al. 2009).

Some of the SMS-EMOA parameters have been set manually on the basis of a pre-study and the previous publications (Bischl et al. 2010; Vatolkin et al. 2011): $p_{01} = 0.01$, $\gamma = 32$ and $\mu = 30$. The parameters to examine were if_r (set to 0.5, 0.2 and 0.05) and crossover (CBC, UC or none). We set the evaluation number for SMS-EMOA to 2,000 evaluations (based on our previous studies, the evaluation number should be large enough to observe the convergence of optimization process, but on the other side, small enough not to enable the experiment runs which require weeks of computing time). The statistical repetition number of optimization runs was set to 10.

3.3 Evaluation

The first metric to minimize was the mean-squared error

$$E^2 = \frac{1}{L}\sum_{i=1}^{L}(\hat{s}_i - s_i)^2 \qquad (7)$$

between the labeled (true) relationship s_i and the relationship $\hat{s}_i$ predicted by a classification model. Here, L corresponds to the total number of audio mixtures. This formula provides a general definition for different possible relationships between 0 and 1; for our tasks classifying audio mixtures $s_i \in \{0;1\}$ and $\hat{s}_i \in \{0;1\}(s_i = 1$ means that the mixture belongs to a category to identify, e.g., 'chord with at least one piano tone' and $s_i = 0$ relates to 'chord without any piano'), so it is equal to a *relative error*.

The second metric was the selected feature share

$$f_r = \frac{|\Phi(X,\theta)|}{|X|}. \qquad (8)$$

It has been argued amongst others in MIR publications (Fiebrink and Fujinaga 2006) that a rational evaluation of FS should be done using independent validation sets (for general discussion of evaluation methods, see Bischl et al. 2012). Following our strategy already investigated in Vatolkin et al. (2011), we isolated 1,000 intervals and 1,000 chords for two HOLDOUT sets, which were not used for classification and optimization, but only for independent evaluation during the optimization process. The OPTIMIZATION OR EXPERIMENT SET was partitioned via tenfold cross-validation: 9/10 of mixtures were used for training of classification models, which were evaluated on the remaining 1/10. This process was repeated for changing partitions ten times, and the mean of E^2 across ten validation runs was used as optimization metric. Therefore, 2,000 optimization evaluations relate to 20,000 classification model trainings and evaluations, so especially the SVM runs required several days to be completed.

Table 1 lists the complete parameter settings of the experiments.

Table 1 Study parameters

Parameter name	Values	Number
Problem description		
Classification tasks	Binary instrument recognition from audio intervals and chords: guitar, piano, wind, strings	8
Experiment sets	2,000 intervals and 2,000 chords for classification and optimization	1
Holdout sets	1,000 intervals and 1,000 chords for independent evaluation	1
Features		
Initial feature set	1,148-dimensional audio feature vector	1
Feature processing	Frames from attack/onset/release intervals and larger building blocks	1
Classification parameters		
Algorithms	C4.5, Random Forest, Naive Bayes, Support Vector Machine	4
Hyperparameters	Default settings; linear kernel for SVM	1
Optimization parameters		
Algorithm	SMS-EMOA	1
Metrics to optimize	E^2 and f_r	1
Selection strategy	(30+1)	1
Mutation	Asymmetric bit flip with $p_{01} = 0.01$ and $\gamma = 32$	1
Crossover	No crossover, Uniform or Commonality-Based	3
Initial feature rate if_r	0.5; 0.2; 0.05	3
Evaluation number	2,000	1
Evaluation method	10-CV on optimization set (9/10 used for training); additional validation on holdout set	1
Number of statistical runs	–	10
Overall experiment number	–	2,880

4 Discussion of results

4.1 Essentiality of feature selection

Application of FS must have some reason; it makes no sense if the performance of different feature sets is approximately the same, or if the classifier already integrates some successful selection or pruning technique and does not require FS as preprocessing step. However, this is not the case in most situations as we can see—even the complex Decision Tree C4.5 algorithm with integrated tree pruning suffers from an increasing feature number.

To run a test for importance of feature selection, we performed a special study after the main experiments listed in Table 1. At first, we measured how often each feature had been selected by a certain classifier for a certain task during the optimization. Because of the overall experiment number was equal to 2,880 and the population size was 30 individuals in each experiment, the overall number of 2,880 × 30 = 86,400 solutions was taken into account. For each combination of eight categories and four classifiers $86,400/(8 \times 4) = 2,700$ solutions were analyzed. The number of selections for each feature $\xi_i \in [0; 2,700]$ can be described as EXPERIMENTAL RELEVANCE of feature $m_i, i \in [1; N]$. Then, we sorted all features due to ξ_i and built the classification models adding features one-by-one starting from the most relevant feature with the largest ξ_i and ending with the most irrelevant feature.

Figure 3 illustrates the progress of $E^2(\Phi)$ on the holdout set with the increasing number of features for model building sorted by ξ_i, starting with the most relevant feature and adding them due to their decreasing relevancy. The upper subfigure corresponds to the classification task IG and the bottom subfigure to IW (guitar and wind instrument detection in audio intervals). Some interesting observations can be stated. The smallest feature sets produce very large errors—adding more features leads to the rapid decrease in the E^2. After the (local) error optimum is achieved, the further extension of the feature set results in a slow decrease of classification performance: too many features do not provide any advantage. Naive Bayes seems to profit at most from the feature selection; the performance suffers very much from increasing the feature number. The other algorithms indicate the same (however weaker) effect. The pruning technique of C4.5 (Quinlan 1993) leads to plateaus in E^2 development, however, it cannot completely overcome the increasing feature number.

Table 2 lists the complete statistics of this study: $E^2(\Phi_{\min})$ is the optimal (minimal) error produced by feature set $\Phi_{\min}.E^2(\Phi_1)$ is calculated from the models built with the most relevant feature and $E^2(X)$ from all features. In all cases, it can be definitely stated that the optimal feature set is between the both extremes and $|\Phi_{\min}|$ is relatively closer to 1 than to 1,148. Only SVM performs well for larger feature sets. However, the optimal feature number cannot be estimated on the fly for all methods. No clear correlation can be observed for problem 'complexity' and the optimal feature number: e.g., for categorization task IP the minimal E^2 values are higher than for IG, but the cardinality of the optimal feature set increases significantly only for Naive Bayes. At any rate, we can affirm the following observation (certainly it holds only for our concrete study recognizing instruments and we cannot guarantee the same performance for any other classification task).

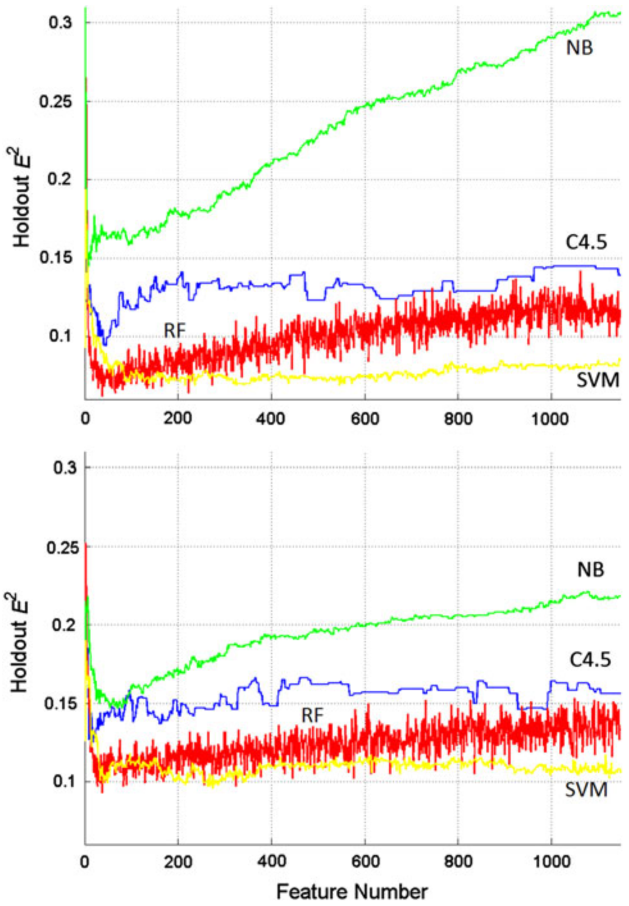

Fig. 3 Examples for E^2 progress depending on the number of selected features sorted by their experimental relevance ξ_i: *upper subfigure* identification of guitar in intervals; *bottom subfigure* wind in intervals. *Green line*: NB, *blue line*: C4.5, *red line*: RF, *yellow line*: SVM (color figure online)

Experimental observation 1 Classification models built from a very limited amount of most relevant features as well as from the complete feature set produce very large classification errors; application of external feature selection increases the classification quality and leads to smaller feature sets than the complete set, which relates to further benefits: reduced demands for storage and computing time as well as reduced danger of overfitting.

4.2 Feature analysis for different tasks

In the previous sub-section, we argued that feature selection is essential for good quality of the categorization process. However another question arises, namely if FS can be run only once for a certain group of classification tasks. Is it possible to isolate some limited and representative feature set, which may distinguish well between piano and guitar, or wind and strings? If yes, it would mean that our work optimizing FS for instrument recognition is done once; only the categorization models can be retrained for new instruments. However, the situation is more or less completely opposite of this as can be seen in Fig. 4. Here, we counted the numbers of feature selections (red: higher selection number; blue: lower; lower 10 % are faded out) and grouped the colored markers for different classifiers so that applications of the same classifier on different tasks are close to each other.

It can be observed that very often features which have been selected many times for a certain combination of a classifier and a task are not selected so often for this classifier and another task. In addition, the general number of the often selected different features varies for different classifiers and tasks (as can be also seen in the study from Table 2): for guitar recognition the number of often selected features in Fig. 4 is smaller in total than for strings. Some features seem to be adequate for several problems, e.g., the feature with id 1138 (first relative periodicity amplitude peak from the onset frame) is selected frequently for recognition of all instruments in intervals but winds. For detection of wind instruments in intervals, the feature with id 207 (unfiltered mel frequency cepstral coefficient 2) seems to be very important and is the most selected feature across all classifiers: $\xi_i = 1{,}899$ for C4.5, 1,159 for NB, 1,640 for RF and 2,143 for SVM ($\xi_{\max} = 2{,}700$).

Another interesting experiment is to switch the (sub)optimal feature sets between different categorization tasks, e.g., build a classification model for guitar recognition from the feature set which was very well suited to identify piano. For this target, we selected the feature sets $\Phi_{\min}$ from Table 2 (they may not correspond to the overall optimal feature sets but are indeed the solutions with the smallest E^2 after the experiments described in the previous sub-section) and applied them for different tasks. The results are listed in Table 3. As an example, the best $\Phi_{\min}(\text{C4.5})$ for the task identifying guitar in intervals IG has $E^2 = 0.094$. Using the best feature sets for IP, IW and IS, classifying IG leads to E^2 values of 0.122, 0.158 and 0.177, respectively. For other classifier-task combinations it can be stated from all table entries that the feature sets which are very representative for a certain combination always outperform the other feature sets, which were very representative for another task and the same classifier. So the experiment study for the analyzed categorization tasks verifies our suggestion that a feature set which is very well suited for classification of a certain task, and is often not the best for classification of other tasks. Concluding, we can state:

Experimental observation 2 Even for closely related groups of classification tasks (here, instrument recognition in intervals or chords), the relevance of features is very different. Features which are representative for one task may be completely unimportant for the other.

Table 2 Mean-squared error E^2 for three different feature sets: optimal feature set $\Phi_{\min}$ with the minimal E^2 adding the features sorted by ξ_i; only the one feature with the highest ξ_i (set Φ_1); set of all features X

Task	Alg.	$E^2(\Phi_{\min})$	$\lvert\Phi_{\min}\rvert$	$E^2(\Phi_1)$	$E^2(X)$
IG	C4.5	0.094	42	0.14	0.139
	RF	0.062	38	0.11	0.117
	NB	0.139	6	0.31	0.307
	SVM	0.069	215	0.093	0.085
IP	C4.5	0.102	15	0.129	0.137
	RF	0.079	40	0.114	0.116
	NB	0.153	52	0.45	0.454
	SVM	0.087	88	0.122	0.105
IW	C4.5	0.125	17	0.176	0.156
	RF	0.093	38	0.131	0.133
	NB	0.147	53	0.215	0.218
	SVM	0.096	274	0.126	0.106
IS	C4.5	0.135	20	0.183	0.168
	RF	0.076	121	0.135	0.127
	NB	0.147	85	0.289	0.25
	SVM	0.128	817	0.153	0.141
CG	C4.5	0.14	14	0.209	0.215
	RF	0.116	69	0.207	0.189
	NB	0.199	17	0.307	0.302
	SVM	0.118	259	0.154	0.132
CP	C4.5	0.15	21	0.226	0.214
	RF	0.122	37	0.194	0.181
	NB	0.217	53	0.415	0.405
	SVM	0.137	495	0.174	0.152
CW	C4.5	0.22	26	0.245	0.25
	RF	0.171	152	0.212	0.217
	NB	0.219	63	0.299	0.3
	SVM	0.18	240	0.197	0.188
CS	C4.5	0.156	27	0.203	0.201
	RF	0.12	28	0.169	0.168
	NB	0.185	183	0.238	0.227
	SVM	0.13	476	0.141	0.149

Detection of instruments in audio intervals: IG, guitar; IP, piano; IW, wind; IS, strings. Detection of instruments in chords: CG, guitar; CP, piano; CW, wind; CS, strings

In combination with observation 1, this also means that integration of such features for a concrete task does not mean just some noisy component, but it may significantly decrease the classification performance. From this, it follows that the FS process should be rerun each time after the definition of a new classification problem.

4.3 Evaluation of multi-objective approach

Figure 5 shows the growing dominated hypervolume (5) with a fixed reference point [1;1] for category IW, no crossover and the two $if_r = \{0.5; 0.05\}$. First notice is that the optimization performs well in general and converges: hypervolume increases rapidly at the beginning and this process slows down later. The hypervolume progress for the holdout set manifests some regions with decreasing hypervolume, however, the general performance remains very well, which is not always the case for independent evaluation (see, Loughrey 2004), where GA-based FS produced significantly lower accuracies for test sets of several classification tasks. The gaps between all four classifiers remain, also approximately, the same; RF and SVM outperform clearly C4.5 and NB.

The difference between upper and lower subfigures is in the initial feature rate if_r. It is obvious that starting with 5 % features (where feature rate is one of the two metrics to optimize) produces the larger starting hypervolume. Yet it cannot be always expected that the final hypervolumes after the optimization would be larger for higher starting hypervolumes: starting with a lower feature number may be dangerous, since the optimization can stagnate in a local optimum, especially, with the asymmetric mutation which favors switching off the bits. In the example figure, however, it appears to be slightly better to start with only 5 % features instead of 50 %.

A deeper analysis of optimization performance and also SMS-EMOA parameters is provided in Fig. 6. The upper row corresponds to hypervolumes of the last fronts after optimization. As could be already stated from Table 2, the recognition of instruments in chords is harder than in intervals. The differences between the classifiers are clearer than between different optimization settings (we will discuss the impact of different parameters later in more detail). The tasks have different complexities: for the easiest tasks IG and IP hypervolumes have higher values than 0.9.

The row in the middle lists the last hypervolumes divided by initial hypervolumes describing the relative optimization impact. Obviously, starting with a small feature number means also starting with a high hypervolume, therefore, those runs do not possess a strong overall increase in hypervolume. An impact of different crossover operators cannot be identified. Another interesting observation is that all classifiers but NB benefit approximately the same from the optimization—even if their classification performance and last hypervolumes are clearly different at the end (cf. the upper row as well as Fig. 5).

The bottom subfigure deals with the generalization ability of the optimization. Here, the last hypervolume on the independent holdout set is divided by the last hypervolume on the optimization set. A value 1 means that the classification models are well suited for application on different data sets. Indeed, values around 1 are achieved for all classifiers and tasks. For IS and CG tasks, they are even above 1 for most classifiers with optimal SMS-EMOA

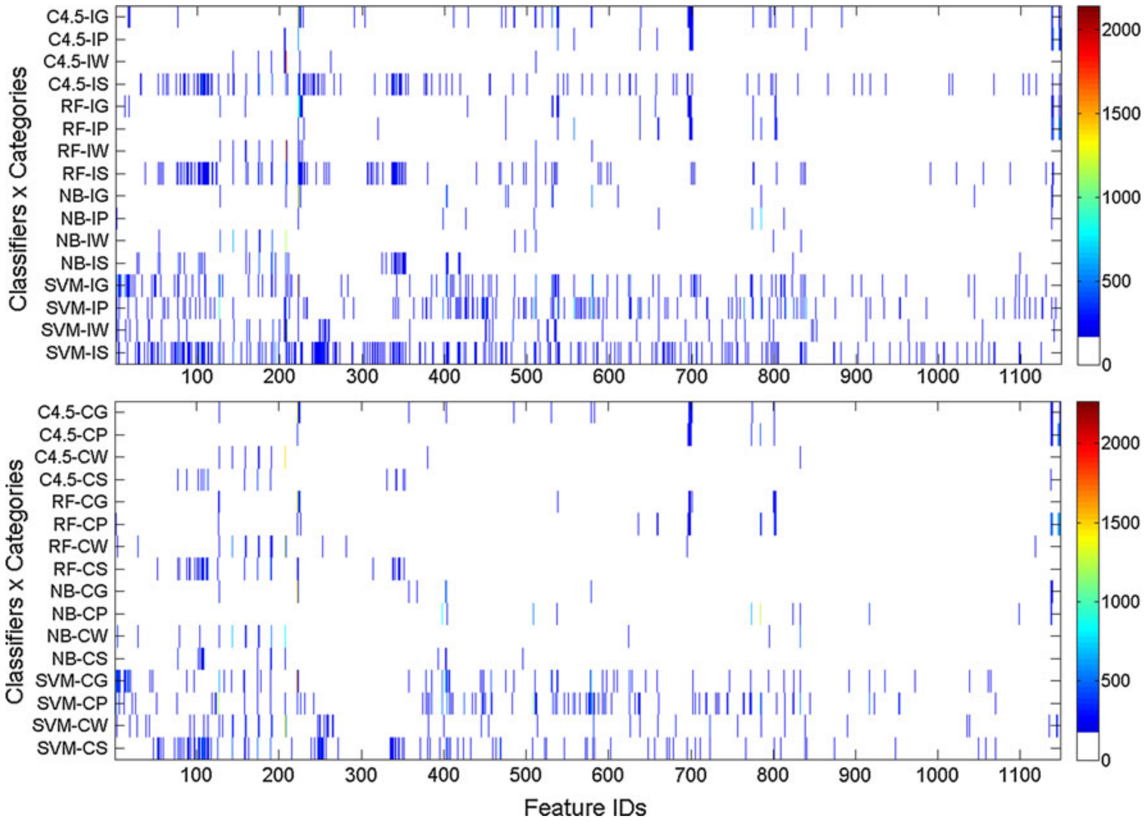

Fig. 4 Often selected features for different classifiers and tasks. Identification of instruments in intervals: IG: guitar; IP: piano; IW: wind; IS: strings. Identification of instruments in chords: CG: guitar; CP: piano; CW: wind; CS: strings

parameters. Another conclusion is that SVM has the smallest generalization ability, even though it is the best method together with RF due to the final hypervolume values. An explanation could be that SVM might choose some not really relevant features among the many selected.

To analyze more precisely, the differences between the nine different SMS-EMOA parameters (three crossovers multiplied by three initial feature rates), we estimated the relative gaps between runs with different optimization parameters for the same classifier and task. This is illustrated in Fig. 7. In each larger square (corresponding to a combination of classifier and task) consisting of nine smaller squares, the white color corresponds to the largest hypervolume and the black color to the lowest.

Maybe the most surprising observation is that the application of crossover, especially for FS-tasks-developed commonality-based crossover, does not lead to a significant increase of hypervolume relative to no crossover. For different categorization problems and classification methods, sometimes no crossover is the best possibility, sometimes UC and sometimes CBC. In addition, if we analyze different classifiers for the same task, or different tasks classified by one certain method, no recommendation can be given. In general, the omittance of crossover leads to the highest hypervolume in 13 combinations of task and classifier, UC in 11 cases, and CBC in 8 cases. Therefore, our recommendation is to use no crossover since crossover requires some (however low) computing efforts and does not provide any systematic quality increase.

For if_r, the situation is clearer. All three classifiers but SVM produce higher hypervolumes starting with smaller feature number. $if_r = 0.5$ is not a good choice; $if_r = 0.2$ is rather suitable and produces, e.g., higher hypervolumes for combinations RF-IP, RF-CS, NB-IS and NB-CS. However, the largest amount of white squares can be counted for $if_r = 0.05$. Another advantage of this feature rate is that the building of models from less features is clearly faster: the entropy calculation for tree algorithms and separation probabilities for NB must be estimated for a lower feature number. In addition, the classification is faster, since the models are less complex. For SVM, the situation is completely opposite: no single run with $if_r = 0.05$ provides a white colored square and the lowest hypervolumes (black) correspond always to the runs started with 5 % of features.

Summarizing our analysis of optimization performance, we can state the following.

Experimental observation 3 Multi-objective feature selection by SMS-EMOA performs well and leads to significant increase of hypervolume after the optimization. Furthermore, the created models are generalizable and can

Table 3 Mean-squared error E^2 for $\Phi_{\min}(\mathcal{A}_i, \mathcal{T}_j)$ selected for classification of $\mathcal{T}_k$ (i.e., application of the best feature set for a task $\mathcal{T}_j$ for classification of a current task $\mathcal{T}_k$). $\mathcal{A}_i, i \in \{1, \ldots, 4\}$: current classifier; $\mathcal{T}_j$: classification task j; corresponds to rows in the table; $\mathcal{T}_k$: classification task k; corresponds to columns in the table

	IG	IP	IW	IS		CG	CP	CW	CS
C4.5									
IG	0.094	0.11	0.195	0.245	CG	0.14	0.198	0.265	0.23
IP	0.122	0.102	0.26	0.281	CP	0.208	0.15	0.28	0.224
IW	0.158	0.202	0.125	0.212	CW	0.292	0.272	0.22	0.203
IS	0.177	0.179	0.152	0.135	CS	0.294	0.229	0.293	0.156
RF									
IG	0.062	0.103	0.117	0.165	CG	0.116	0.141	0.219	0.199
IP	0.094	0.079	0.146	0.139	CP	0.155	0.122	0.225	0.182
IW	0.112	0.149	0.093	0.108	CW	0.187	0.186	0.171	0.166
IS	0.086	0.097	0.123	0.076	CS	0.238	0.341	0.212	0.12
NB									
IG	0.139	0.228	0.36	0.302	CG	0.199	0.256	0.287	0.288
IP	0.176	0.153	0.324	0.308	CP	0.268	0.217	0.288	0.265
IW	0.267	0.388	0.147	0.301	CW	0.296	0.387	0.219	0.244
IS	0.325	0.473	0.258	0.147	CS	0.286	0.353	0.268	0.185
SVM									
IG	0.069	0.117	0.119	0.153	CG	0.118	0.173	0.191	0.167
IP	0.116	0.087	0.125	0.209	CP	0.143	0.137	0.196	0.149
IW	0.08	0.125	0.096	0.16	CW	0.138	0.186	0.18	0.17
IS	0.084	0.101	0.112	0.128	CS	0.162	0.173	0.186	0.13

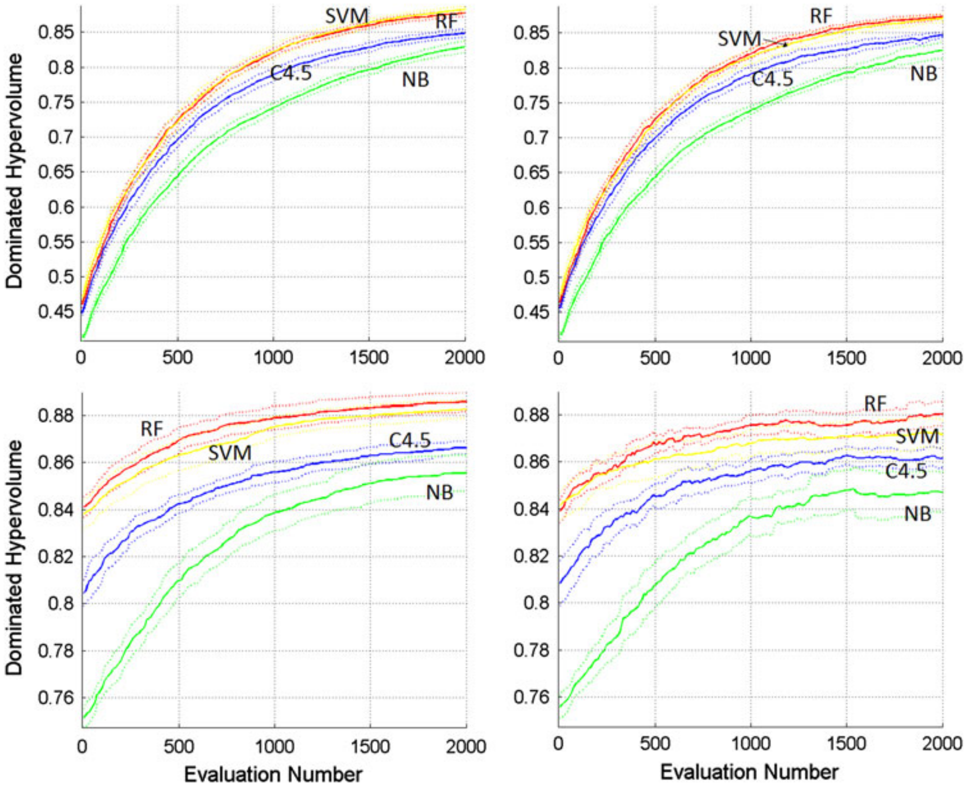

Fig. 5 Mean dominated hypervolume progress for category IW for optimization set (*left subfigures*) and holdout set (*right subfigures*). Confidence interval is marked by *dotted lines*. *Upper subfigures*: $if_r = 0.5$; *bottom subfigures*: $if_r = 0.05$. *Red lines*: RF; *yellow lines*: SVM; *blue lines*: C4.5; *green lines*: NB (color figure online)

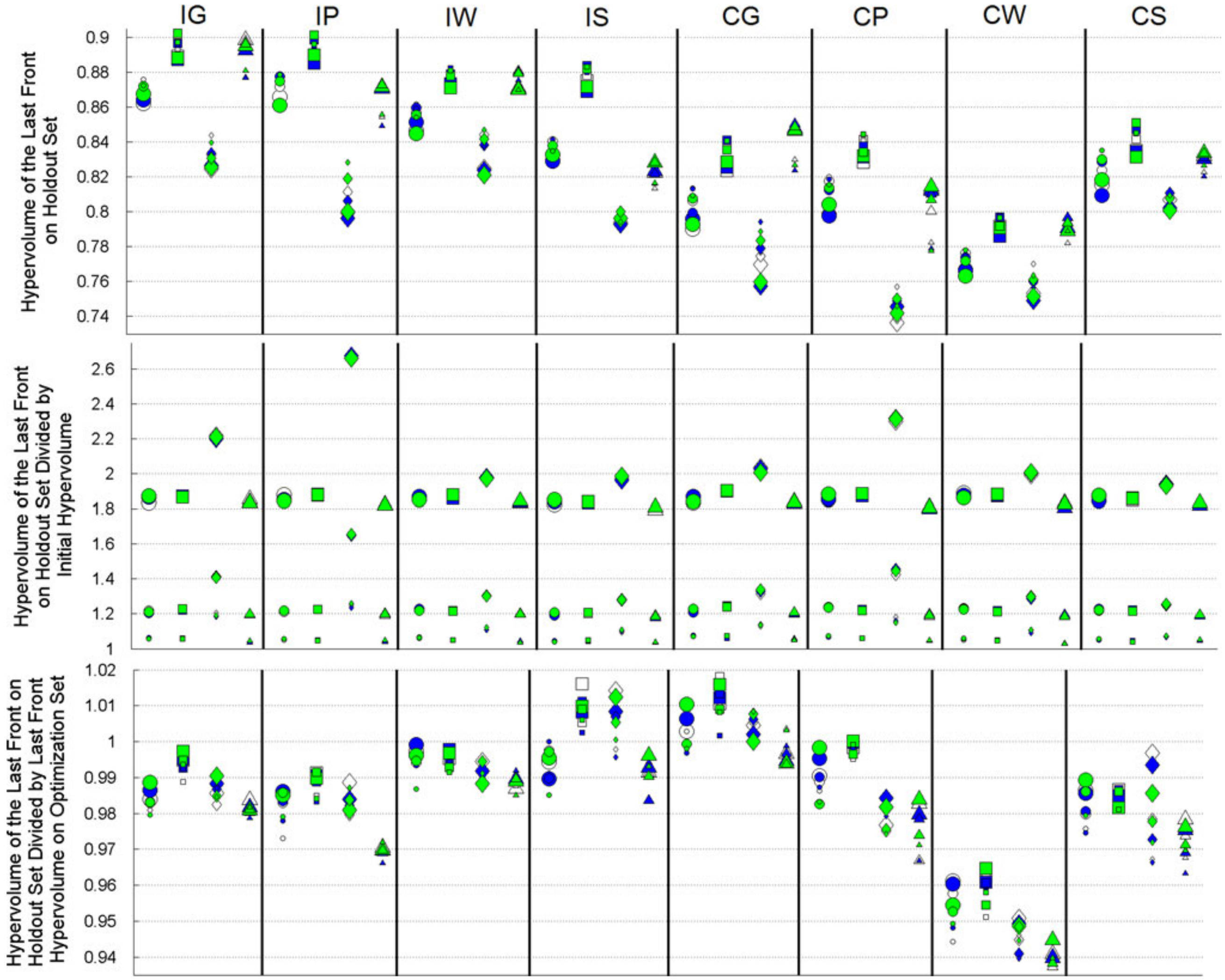

Fig. 6 Analysis of dominated hypervolume of last front after optimization. *Circles*: C4.5; *squares*: RF; *diamonds*: NB; *triangles*: SVM. *Small signs*: $if_r = 0.05$; medium signs: $if_r = 0.2$; *large signs*: $if_r = 0.5$. *Green/pale sign* background: crossover UC; *blue/dark*: CBC; *white*: no crossover (color figure online)

be applied for successful instrument identification in independent sets with comparable instrument distribution.

Experimental observation 4 Integration of two different crossover operators, one of them even developed explicitly for feature selection, does not lead to any systematic quality improvements. Actually the experiments without crossover produced the largest number of highest hypervolumes.

Experimental observation 5 Starting with less features does not necessarily imply that the optimization will get stuck in local optima. It is in most cases, even better to set if_r to smaller rates. But this parameter is sensitive to the choice of classifier: SVM fails if started with a low feature number.

4.4 Comparison of classifiers

Figure 8 illustrates some examples of the final populations of SMS-EMOA from all statistical runs for instrument identification from chords. The upper four subfigures were generated from runs with $if_r = 0.05$ and the lower four from runs with $if_r = 0.5$. As discussed above, SVM requires larger if_r for good performance whereas the other methods are better rather with smaller initial feature rates. This fact is supported by the results given in the figure: for $if_r = 0.05$ (upper subfigures) only few SVM solutions exist which belong to the overall non-dominated front. For $if_r = 0.5$ (lower subfigures), SVM occupies in three of four fronts the left region of the overall non-dominated front which corresponds to solutions with higher f_r but lower E^2.

The most important observation is that for no classification task the overall non-dominated front is occupied by solutions of only one classifier. If a larger tradeoff front should be presented to a decision maker, it means that it is reasonable to run different classification methods instead of limiting the choice to a single classifier. Nevertheless, some trends can be

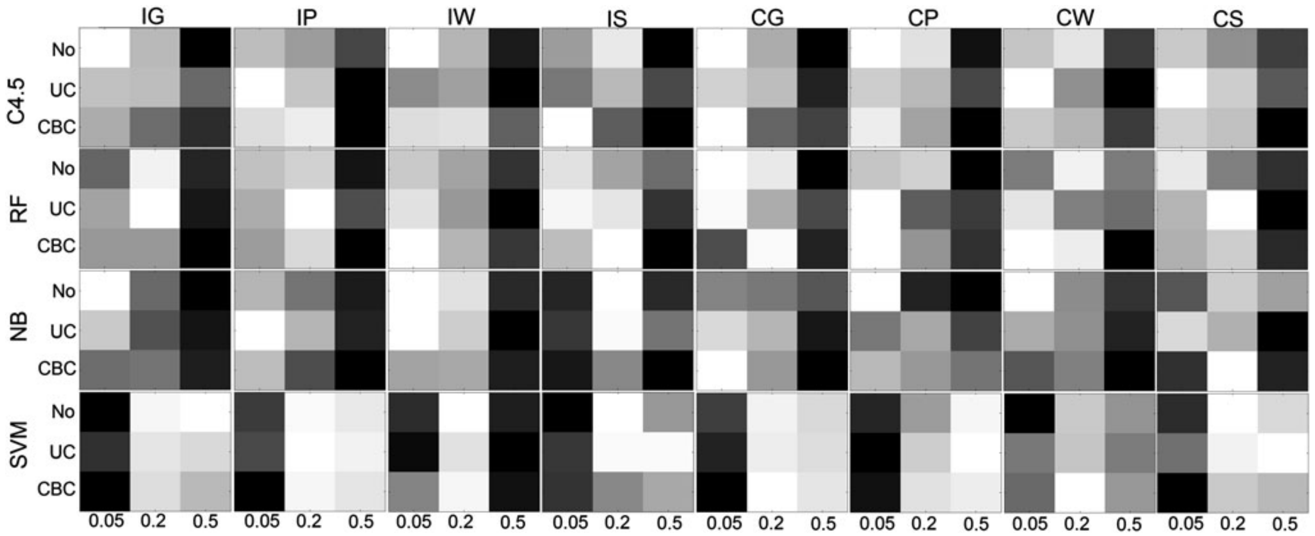

Fig. 7 Relative differences between final mean hypervolumes for nine different SMS-EMOA parameter combinations. *Columns of large squares*: categorization tasks; rows of large squares: classifiers. Scale for each 3 × 3-square: *white color* the largest mean hypervolume (better); *black color* the smallest mean hypervolume (worse). Please note that the scales are estimated for each 3 × 3-square separately

identified: SVM and RF seem to be better with regard to E^2; SVM requires rather large feature numbers to achieve small error rates. NB and C4.5 deal better with very small feature numbers at the expense of larger errors.

For comparison of classification methods, we introduced a measure (Vatolkin et al. 2011), which estimates how often an average solution of a classifier $\mathcal{A}$ after optimization is dominated by an average solution of another classifier $\mathcal{B}$ (only the last populations after the optimization are taken into account):

$$N(\mathcal{A},\mathcal{B}) := \frac{1}{p \cdot r} \cdot \sum_{i=1}^{p \cdot r} \left(\frac{1}{p \cdot r} \cdot \sum_{j \in \{1,\ldots,p \cdot r\}; a_i \prec b_i} 1 \right), \tag{9}$$

where $p = 30$ is the population size, $r = 10$ is the number of statistical repetitions of SMS-EMOA optimization runs and a_i, b_i the solutions to compare.

The mean value $\widehat{N}(\mathcal{A})$ comparing $\mathcal{A}$ to the remaining other classifiers describes the average overall danger for solutions of classifier $\mathcal{A}$ to be dominated by other methods ($c = 4$ is the overall classifier number):

$$\widehat{N}(\mathcal{A}_i) := \frac{1}{c-1} \cdot \sum_{j \in \{1,\ldots,c\} \setminus i} N(\mathcal{A}_i, \mathcal{B}_j) \tag{10}$$

Table 4 lists all $\widehat{N}(\mathcal{A}_i)$ values. Several interesting facts can be stated. At first, as discussed before, the choice of if_r seems to be very important in combination with a particular classifier not only with regard to the final hypervolume but also for generation of solutions which are less dominated by other classifiers. Best (i.e., smallest) values $\widehat{N}(\mathcal{A}_i)$ of C4.5 are achieved by runs with $if_r = 0.05$ whereas starting with a larger feature number raises a problem for this method. RF is 'satisfied', in general, with different initial feature rates, however it is more often dominated by other classifiers if $if_r = 0.05$. For NB no clear recommendation can be given, however, it prefers smaller initial feature rates. And SVM, as previously mentioned, requires more features at start and has only one outlying best $\widehat{N}(\mathcal{A}_i)$ value for $if_r = 0.05$.

Again, any systematic impact of crossover is not visible.

Table 5 summarizes how often for eight instrument identification tasks, each classifier with optimal SMS-EMOA parameters had the best (lowest) $\widehat{N}(\mathcal{A}_i)$, how often it was at second and so forth. Apparently, there is a clear order: for five tasks RF is the best method with the lowest number of dominated solutions, whereas SVM is the worst in four cases. This indicates another disadvantage of SVM: the final non-dominated hypervolume is indeed high, but many solutions are dominated by other classifiers, which means that a larger amount of statistical runs or higher population sizes may be necessary for SVM to produce solutions closer to the Pareto-front.

With the target to approximate the real Pareto-front of non-dominated solutions as good as possible, the selection of possible classifier(s) can be done in the order given by this table. However, it should not be forgotten that the overall non-dominated fronts consist of solutions of several classifiers. And this ranking list must also be treated with some caution: e.g., changing of crossover from UC to none will move NB to fourth place from the first for category IW, so only general recommendations can be given. Some robustness of the ranks can be guaranteed by the last column in Table 5 which lists the mean $\widehat{N}(\mathcal{A}_i)$ values

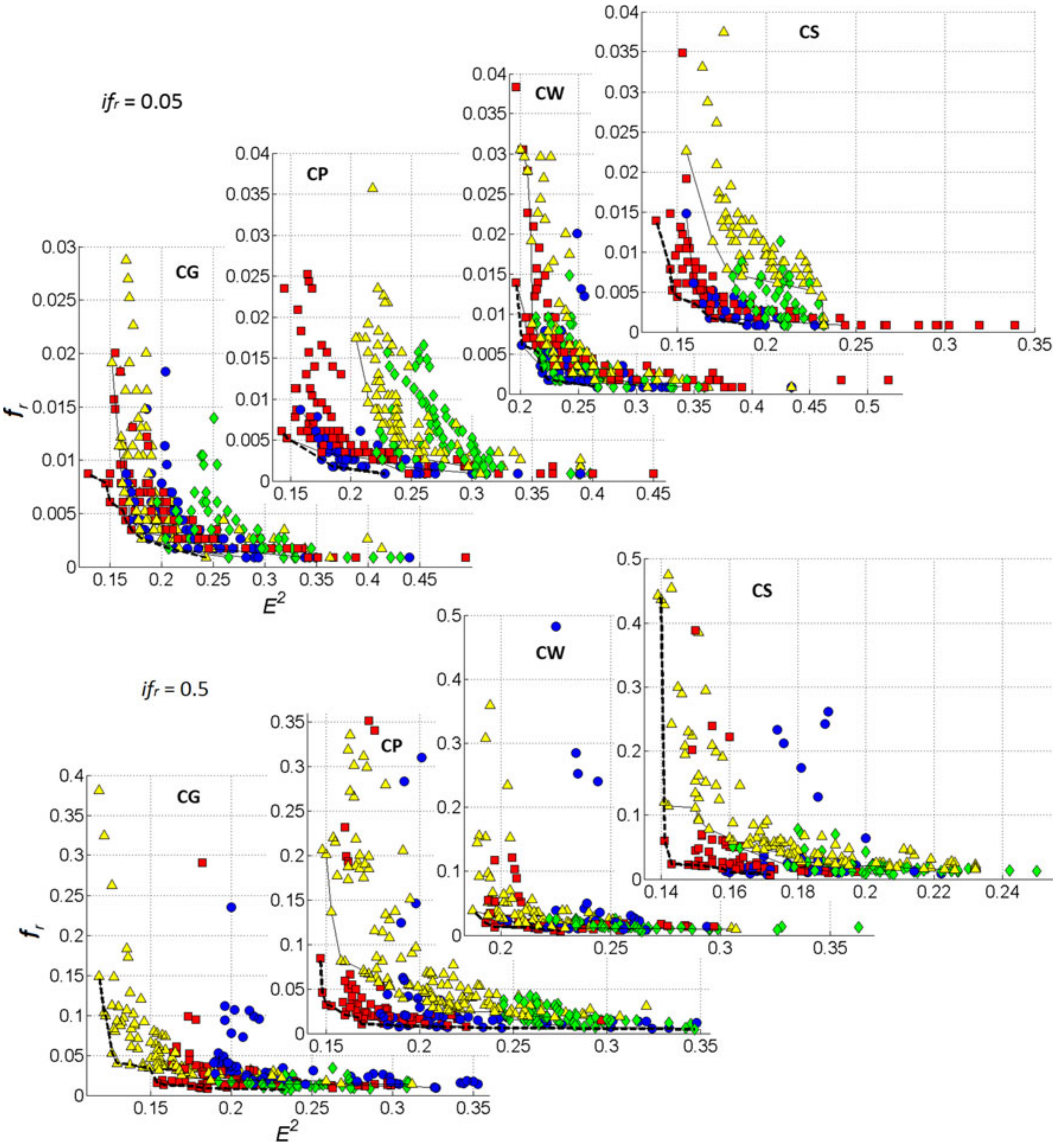

Fig. 8 Last solutions after optimization runs. Non-dominated fronts for each classifier are marked by *thin lines*. The overall non-dominated front is marked by *thick dashed line*. *Blue circles*: C4.5; *red squares*: RF; *green diamonds*: NB; *yellow triangles*: SVM (color figure online)

averaged across all eight tasks for optimal SMS-EMOA parameters and each classifier; so, if a new instrument identification task should be optimized, it can be reasonable to start with RF followed by other methods.

Another focus of classifier comparison describes the distribution of solutions. By means of different lines (boundaries) $f_r = \alpha \cdot E^2, \alpha \geq 0$ starting in the origin we can define the segments with certain balance regions between two metrics, e.g., if $\alpha = 1$, the segment between 0° and 45° corresponds to the region where f_r is optimized relatively better than E^2 and the segment between 45° and 90° where E^2 is optimized relatively better than f_r. At first, we estimate the segment share metric $\Omega_s(\mathcal{A}_i)$ defined here for minimization of two objectives:

$$\Omega_s = \sum_{k=1}^{s} \omega_k, \tag{11}$$

with $\omega_k \in \{0;1\}$ and $\omega_k = 1$ if and only if

$$\exists \Phi(X, \theta_j) : \omega_k \leq \arctan \frac{I_2(\mathcal{A}_i, P, \Phi(X, \theta_j))}{I_1(\mathcal{A}_i, P, \Phi(X, \theta_j))} \leq \omega_{k+1}. \tag{12}$$

where $\mathcal{A}_i$ is the classifier to be analyzed, P is the set of analyzed solutions after the optimization (SMS-EMOA population), $\theta_j \in 2^N$ corresponds to the indices of selected features (i.e., representation of an SMS-EMOA individual) and ω_k is the angle which defines a lower or upper segment boundary if s segments are spanned between $\omega_{\min}$ and $\omega_{\max}$.

For Table 6, we spanned $s = 20$ segments between two outer solutions (the first one with minimal f_r and the second with minimal E^2) of the overall non-dominated front from all solutions for a fixed combination of if_r and crossover and calculated here another ranking between the classifiers as for results of the Table 4. For this purpose, we estimated, at first, the average $\Omega_s(\mathcal{A}_i)$ for the different SMS-EMOA parameters and the same categorization task, and then compared these values between $\mathcal{A}_i$. Only for the task IS, the mean $\Omega_s(\mathcal{A}_i)$ values were the same for C4.5, RF and SVM, so we assigned for all three classifiers the rank 2 and for NB the rank 4. SVM seems to provide the largest distribution covering almost all segments, very closely followed by RF. C4.5 has more segments without solutions after the optimization, and NB has the smallest $\Omega_s(\mathcal{A}_i)$: almost half of the segments remains in average without NB solutions.

The last part of our comparison of classifiers measures the solution alignments with respect to the above introduced segments.

For the calculation of the centroid segment estimating the overall trend to prefer rather the first or the second metric during the optimization, we take into account again all $\mathcal{A}_i$ solutions after the optimization:

$$\Omega_a = 1 + \frac{C-1}{s-1}, \qquad (13)$$

where the centroid segment C is calculated as follows:

$$C = \frac{1}{|P|} \cdot \sum_{j=1}^{|P|} s_{\text{cur}}(\mathcal{A}_i, j) \qquad (14)$$

and the segment of the current solution P_j is

$$s_{\text{cur}}(\mathcal{A}_i, j) = \left\lceil \frac{\arctan \frac{I_2(\mathcal{A}_i, P_j)}{I_1(\mathcal{A}_i, P_j)} - \omega_{\min}}{\frac{\omega_{\max} - \omega_{\min}}{s}} \right\rceil. \qquad (15)$$

This measure gives us the information about the distribution balance. For $\Omega_a(\mathcal{A}_i) = 1.5$ the two metrics are optimized similarly. A value close to 1 would mean that most of the final solutions are in the segments with better f_r values; a value close to 2 would appear for segments with better E^2 values. It is important to mention that this holds *relative* to the segments between $\omega_{\min}$ and $\omega_{\max}$ which are spanned here between the two outer solutions of the overall non-dominated front—see also later the discussion of Fig. 9, where the segments are spanned between 0° and 90°.

Again we can calculate the ranks comparing $\Omega_a(\mathcal{A}_i)$ for different classification tasks (see Table 7). Here, we sort the ranks toward the optimization of E^2 as following: classifiers with higher $\Omega_a(\mathcal{A}_i)$ are ranked also higher and this means that their solutions have the stronger trend to optimize E^2 rather than f_r, relative to the non-dominated front of all solutions. Only once, RF and C4.5 had the same $\Omega_a(\mathcal{A}_i)$ (task CG) and we assigned to the both methods the rank 2.

The rank order is very clear. SVM generates the largest number of solutions which optimize E^2 at best among all classifiers, requiring on the other side the larger feature number. RF occupies the reliable second position, C4.5 follows and NB produces more solutions with smaller feature rate and larger classification error.

Finally, this behavior can be visually illustrated by Fig. 9. Here, we spanned 1,000 equal segments in another way, namely between the angles of 0° and 90°. The number of final solutions from ten statistical repetitions for each classifier (moving average over five segments to illustrate a smoother trend) corresponds to the ordinate. It can be seen very clearly that the first few segments which minimize f_r better at the expense of higher E^2 are almost always dominated by NB solutions, sometimes also by C4.5, whereas RF provides a larger number of compromise solutions in the middle and SVM has often more solutions than other classifiers in the higher segments. The overall segment number with a solution number above zero is significantly higher for runs with $if_r = 0.5$; here, more solutions exist which have more selected features and higher error. It is always possible due to the stochastic nature of SMS-EMOA optimization that some small number of such undesirable solutions remains, which are stuck in local optima. $if_r = 0.05$ in combination with asymmetric mutation guarantees that not so many solutions with $f_r > 0.05$ can be generated after the optimization, and the corresponding empty segments do not exist in the figure.

Concluding our comparison regarding classification, we can state the following observations:

Experimental observation 6 The two Decision Tree algorithms Random Forest and C4.5 produce the largest number of solutions which are not dominated on average by the other classifiers. However, this characteristic only holds for the optimal choice of initial feature rate.

Experimental observation 7 For RF and SVM a wider distribution of solutions which correspond to different tradeoffs of the two optimization metrics can be stated; C4.5 provides acceptable results, and NB leaves many tradeoff segments completely empty.

Experimental observation 8 The classifiers can be clearly distinguished regarding to their preference of the first or second metric: SVM and RF generate more solutions with smaller E^2 and larger if_r; C4.5 and NB solutions occupy the regions with higher E^2 and smaller if_r.

Table 4 $\widehat{N}(\mathcal{A}_i)$ for all classification tasks and SMS-EMOA parameters

if_r	Cross.	IG	IP	IW	IS	CG	CP	CW	CS
C4.5									
0.5	No	26.26	10.38	29.51	19.95	32.74	13.06	25.81	16.71
0.5	UC	20.0	21.97	27.58	19.32	32.25	11.35	18.47	17.62
0.5	CBC	25.61	17.68	18.35	18.09	23.19	12.71	18.24	28.55
0.2	No	22.72	12.01	17.52	14.36	22.60	4.86	13.58	7.27
0.2	UC	17.04	14.55	19.34	18.20	10.13	4.37	15.36	13.63
0.2	CBC	24.06	8.34	12.86	13.98	17.79	3.43	7.56	1.65
0.05	No	**7.93**	3.21	**7.89**	8.12	5.90	**2.27**	6.47	1.53
0.05	UC	7.98	4.28	10.42	**7.86**	6.65	2.60	5.69	**0.97**
0.05	CBC	10.08	**2.86**	8.63	8.01	**5.80**	4.31	**4.55**	1.91
RF									
0.5	No	3.9	1.71	4.55	**0.03**	5.30	1.85	6.37	3.66
0.5	UC	5.72	0.6	5.09	0.07	**2.77**	1.96	7.74	4.62
0.5	CBC	4.54	1.08	**4.25**	0.11	4.14	**1.17**	11.08	2.55
0.2	No	**1.13**	**0.54**	8.0	0.09	5.30	2.87	**5.54**	2.41
0.2	UC	2.21	0.63	8.75	0.04	5.32	5.06	6.40	**1.05**
0.2	CBC	2.06	1.63	9.47	0.22	5.26	3.14	5.75	5.65
0.05	No	3.48	2.7	7.11	0.77	5.27	2.73	8.04	3.37
0.05	UC	3.48	1.36	8.71	0.38	7.41	2.06	10.10	6.08
0.05	CBC	3.01	2.23	7.36	0.27	10.33	1.91	8.09	4.38
NB									
0.5	No	14.28	18.13	17.67	21.55	11.93	26.41	19.39	19.79
0.5	UC	17.74	22.27	17.53	24.82	16.30	24.88	24.90	19.73
0.5	CBC	10.29	**12.91**	22.04	18.07	20.77	31.25	20.52	13.11
0.2	No	11.75	20.95	16.48	23.81	19.90	32.96	11.83	**9.97**
0.2	UC	**2.20**	18.90	9.53	**16.82**	16.61	35.96	13.81	18.67
0.2	CBC	14.47	26.38	11.26	25.15	13.65	33.61	14.85	28.52
0.05	No	10.28	16.53	8.97	27.49	17.78	24.99	**8.67**	14.08
0.05	UC	9.15	16.07	**4.17**	25.91	10.65	29.78	9.72	14.33
0.05	CBC	10.91	18.60	12.7	26.71	**10.02**	**19.90**	9.81	17.87
SVM									
0.5	No	6.15	21.63	6.38	26.18	**2.87**	**24.30**	9.88	24.15
0.5	UC	8.15	23.57	**6.16**	28.88	6.54	25.43	**6.02**	17.66
0.5	CBC	**4.46**	**20.53**	8.54	25.86	4.52	25.60	8.33	**16.61**
0.2	No	13.72	31.95	7.31	25.84	5.17	38.60	9.87	25.42
0.2	UC	11.89	33.88	7.35	29.52	10.33	33.59	8.16	31.59
0.2	CBC	10.86	33.71	6.79	31.11	7.45	32.34	7.37	30.29
0.05	No	11.23	28.82	7.32	**20.52**	7.79	31.97	15.71	26.71
0.05	UC	13.83	29.35	7.13	25.37	12.23	32.73	12.45	28.34
0.05	CBC	9.12	35.66	6.64	23.97	12.81	31.75	9.81	26.79

The best parameter combination (if_r and crossover) for each classifier and task is indicated in bold

Experimental observation 9 The overall non-dominated fronts created from solutions of all classifiers and all statistical runs for a fixed if_r and crossover configuration consist of solutions of different classifiers; no classifier is completely absent. If many different tradeoffs should be analyzed—higher error and less features (saving storage and computing time) against more features and better classification quality—we recommend to integrate several classification methods for instrument recognition.

Table 5 Number of classifier ranks comparing $\widehat{N}(\mathcal{A}_i)$ for all classification tasks $\mathcal{T}_j$

$\mathcal{A}_i$	1st	2nd	3rd	4th	$\frac{1}{8}\sum_{j=1}^{8}\widehat{N}_{\min}(\mathcal{A}_i,\mathcal{T}_j)$
RF	5	3	–	–	2.06
C4.5	2	3	1	2	5.02
NB	1	1	4	2	10.58
SVM	–	1	3	4	12.68

Table 6 Number of classifier ranks averaging $\Omega_s(\mathcal{A}_i)$ for all SMS-EMOA parameters $\mathcal{P}_j$ and classification tasks $\mathcal{T}_k$

$\mathcal{A}_i$	1st	2nd	3rd	4th	$\frac{1}{8}\sum_{k=1}^{8}\left(\frac{1}{9}\sum_{j=1}^{9}\Omega_s(\mathcal{A}_i,\mathcal{P}_j,\mathcal{T}_k)\right)$
SVM	4	3	1	–	0.95
RF	3	3	2	–	0.94
C4.5	–	4	4	–	0.84
NB	–	–	–	8	0.56

5 Summary and further work

In our extensive study, we applied a multi-objective feature selection procedure by SMS-EMOA before the training of instrument identification models. This step definitely leads to a significant improvement of classification quality as well as reduction of the required feature number, which was experimentally verified by the increase in dominated hypervolume on optimization and holdout sets. The error E^2 is below 0.1 for optimal models applied to audio intervals (see Table 2) and varies between 0.12 and 0.18 for more complex chord mixtures. The models are also well generalizable as proved by validating their performance on independent sets.

Comparing two examined SMS-EMOA parameters, we can state the followings.

- Initial feature rate if_r has a strong impact on the feature selection quality and should be set depending on the classifier: SVM (and RF in many cases) require larger feature number for building of successful models whereas C4.5 solutions cannot overcome the local optima if too many features are provided to this classifier at the beginning of the optimization.
- Both analyzed crossover operators do not stand for any systematic improvements and may be omitted in future studies.

The classifiers have very different characteristics and it is hard to provide a general recommendation; depending on the decision-maker's preferences one or another algorithm may be preferred. At any rate, it can be reasonable to invest computing time for building of models with different classifiers, since all of them provide some parts of overall non-dominated fronts and the performance varies from classification task to classification task.

- The C4.5 is maybe the last choice if only few classifiers should be applied. It is the second slowest method due to its pruning technique which, however, cannot deal successfully with a large feature number. With regard to several evaluation possibilities discussed in Sect. 4.4, the other algorithms perform better. But some overall non-dominated front solutions (see Fig. 8) are occupied by the models of this method and it is not so often dominated by the other methods for $if_r = 0.05$ (Table 4).
- Naive Bayes has rather poor classification performance and the solution tradeoffs are not well distributed preferring the regions with higher E^2 and smaller f_r. However, it can learn well from very small feature sets (occupying corresponding regions in overall non-dominated fronts) and is the fastest method (see below Fig. 10), so it can be applied, e.g., for the estimation of classification task complexity, before other classifiers are started. It benefits at most from the integrated feature selection.
- Support Vector Machine has several advantages and drawbacks. On the one hand, it achieves often the smallest classification errors and provides a broad distribution of tradeoff solutions for the decision maker. On the other hand, it is very slow, requires larger feature sets for successful applications, many solutions are dominated by other classifiers (Table 5) and the model generalization is rather poor compared to other classifiers (Fig. 6).
- Random Forest can be the first recommendation, if only one classifier must be applied. It is fast, provides models with high generalization and acceptable distribution of different metric tradeoffs. However the larger parts of overall non-dominated front are occupied by other classifiers, so we can again recommend to use it with other methods as well.

A large number of possibilities for further research still remains. In our opinion, the most promising investigations can be done in the following directions.

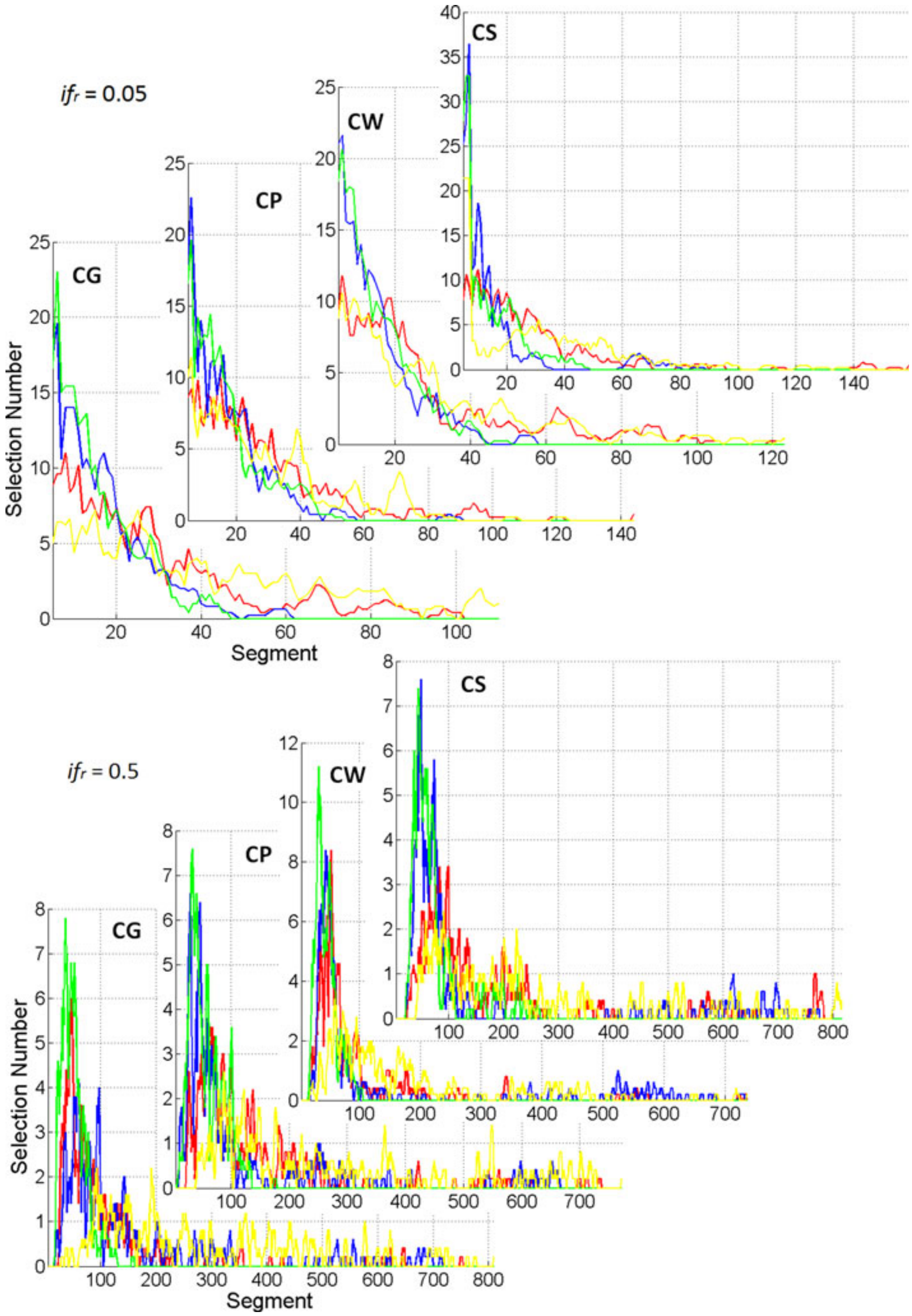

Fig. 9 Number of final solutions in $s = 1{,}000$ segments between $\omega_{\min} = 0°$ and $\omega_{\max} = 90°$ (moving average over five segments). *Blue*: C4.5, *red*: RF, *green*: NB, *yellow*: SVM (color figure online)

- One of our focus points—as also investigated in Bischl et al. (2010) and Vatolkin et al. (2011)—is to predict high-level music categories such as genres, styles or personal preferences. The previously optimized audio characteristics inspired by music theory (instruments, harmonies, structural information, etc.) can be very valuable for this task. The successful extraction of these characteristics can be applied by training and optimizing of the corresponding classification models as it was done here for instrument recognition.
- Time measurements can be reasonable for the fair comparison of classifiers. Figure 10 illustrates an example corresponding to runtime comparison averaged for ten statistical runs. It can be clearly seen, that optimization evaluations of SVM require up to ten times and more runtime than the NB runs. Especially for the right subfigure, larger overall computing times are required, since the initial solutions are started with approximately half of 1,148 features ($if_r = 0.5$). However, the evaluation with regard to runtime is very complex, since it depends on the implementation of an algorithm and other details such as varying operating system load during the statistical repetitions of experiments.

Table 7 Number of classifier ranks averaging $\Omega_a(\mathcal{A}_i)$ for all SMS-EMOA parameters $\mathcal{P}_j$ and classification tasks $\mathcal{T}_k$

$\mathcal{A}_i$	1st	2nd	3rd	4th	$\frac{1}{8}\sum_{k=1}^{8}\left(\frac{1}{9}\sum_{j=1}^{9}\Omega_a(\mathcal{A}_i,\mathcal{P}_j,\mathcal{T}_k)\right)$
SVM	7	–	1	–	1.27
RF	1	7	–	–	1.24
C4.5	–	2	6	–	1.19
NB	–	–	–	8	1.14

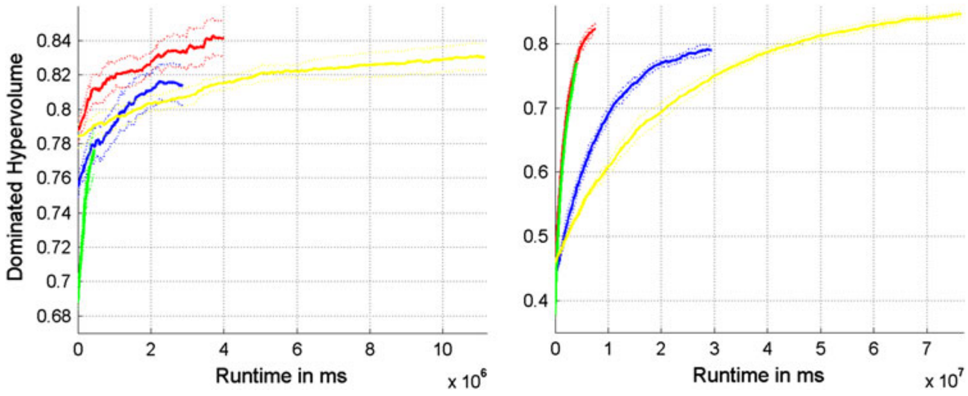

Fig. 10 Progress of dominated hypervolume over runtime for category CG and no crossover. Confidence interval is marked by *dotted lines*. The curves for NB and RF in the *right subfigure* are almost identical; the curve for NB ends approximately at ordinate value 0.76. *Left subfigure*: $if_r = 0.05$; *right subfigure*: $if_r = 0.5$. *Red*: RF, *green*: NB, *blue*: C4.5, *yellow*: SVM (color figure online)

- Deeper analysis of different algorithm steps is not only reasonable but also time consuming. Integration of further up-to-date audio features as well as in some cases, metadata features can improve the classification performance. The analysis of different classifier hyperparameters is another possibility, e.g., tree number for RF, or kernel choice for SVM. The optimizer itself can be tuned, e.g., using self-adaptation, other genetic operators (which can be also domain specific) or different population sizes (e.g., using an archive population) and so on.
- The main goal of the multi-objective optimization, namely the choice of metrics to optimize, can give very important and interesting insights into different classification problems, e.g., model-generalization ability can be optimized against classification error. If we then add the selected feature rate and some metric which measures the model complexity, we would actually switch to rather many-objective optimization with its specific challenges and possibilities.
- Certainly some of our investigations with regard to multi-objective optimization of data mining can be carried on for other real-world categorization tasks not related to music analysis.

Concluding, we would be glad if more MIR (and other research related to classification) tasks would be approached in a multi-objective way. Currently, many studies deal only with one evaluation criterion, so that the created optimized models may provide indeed very low error rates, but require large training and classification time, can be overfitted and less generalizing or are unsuitable for highly unbalanced sets.

Acknowledgements This work has been supported by the Klaus Tschira Foundation within the project 00.146.2008, “Multi-objective optimization of music classification based on high-level features with Computational Intelligence methods”.

References

Ahrendt P (2006) Music genre classification systems: Ph.D. thesis. Informatics and mathematical modelling, Technical University of Denmark

Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: multiobjective selection based on dominated hypervolume. Eur J Oper Res 181(3):1653–1669

Bischl B, Mersmann O, Trautmann H, Weihs C (2012) Resampling methods for meta-model validation with recommendations for evolutionary computation. Evol Comput 20(2):249–275

Bischl B, Vatolkin I, Preuß M (2010) Selecting small audio feature sets in music classification by means of asymmetric mutation. In: Proceedings of the 11th international conference on parallel problem solving from nature (PPSN). Springer, Berlin, pp 314–323

Blume H, Bischl B, Botteck M, Igel C, Martin R, Rötter G, Rudolph G, Theimer W, Vatolkin I, Weihs C (2011) Huge music archives on mobile devices. IEEE Signal Process Mag 28(4):24–33

Brown JC, Houix O, McAdams S (2001) Feature dependence in the automatic identification of musical woodwind instruments. J Acoust Soc Am 109(3):1064–1072

Coello CAC, Van Veldhuizen DA, Lamont GB (eds) (2006) Evolutionary algorithms for solving multi-objective problems. Kluwer Academic Publishers, New York

Deb K (2001) Multi-objective optimization using evolutionary algorithms: Wiley-Interscience Series in systems and optimization.. Wiley, Chichester

Eichhoff M, Weihs C (2012) Musical instrument recognition by high-level features. In: Gaul W, Geyer-Schulz A, Schmidt-Thieme L, Kunze J (eds) Challenges at the interface of data analysis, computer science, and optimization. Proceedings of the 34th annual conference of the Gesellschaft für Klassifikation e. V. Springer, Berlin, pp 373–381

Emmanouilidis C, Hunter A, MacIntyre J (2000) A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In: Proceedings of the 2000 Congress on Evolutionary Computation (CEC), vol 1. IEEE, pp 309–316

Eronen A (2003) Musical instrument recognition using ica-based transform of features and discriminatively trained hmms. In: Proceedings of the 7th international symposium on signal processing and its applications (ISSPA), pp 133–136

Eronen AJ (2009) Signal processing methods for audio classification and music content analysis: Ph.D. thesis. Department of Signal Processing, Tampere University of Technology, Finland

Essid S, Richard G, David B (2006) Instrument recognition in polyphonic music based on automatic taxonomies. IEEE Trans Audio Speech Lang Process 14(1):68–80

Fiebrink R, Fujinaga I (2006) Feature selection pitfalls and music classification. In: Proceedings of the 7th international conference on music information retrieval (ISMIR), pp 340–341

Fu Z, Lu G, Ting K, Zhang D (2011) A survey of audio-based music classification and annotation. IEEE Trans Multimedia 13(2):303–319

Fujinaga I (1998) Machine recognition of timbre using steady-state tone of acoustic musical instruments. In: Proceedings of the international computer music conference (ICMC), ICMA, pp 207–210

Goto M, Hashiguchi H, Nishimura T, Oka R (2003) Rwc music database: music genre database and musical instrument sound database. In: Proceedings of the 4th international conference on music information retrieval (ISMIR), pp 229–230

Guyon I, Gunn S, Nikravesh M, Zadeh L (eds) (2006) Feature extraction, foundations and applications. Springer, Berlin

Hall M (1999) Correlation-based feature selection for machine learning: Ph.D thesis. Department of Computer Science, Waikato University, New Zealand

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. SIGKDD Explor 11:10–18

Heittola T, Klapuri A, Virtanen T (2009) Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In: Proceedings of the 10th international society for music information retrieval conference (ISMIR), pp 327–332

Kitahara T, Goto M, Komatani K, Ogata T, Okuno HG (2007) Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps. EURASIP J Adv Signal Process, vol 2007

Kobayashi Y (2009) Automatic generation of musical instrument detector by using evolutionary learning method. In: Proceedings of the 10th international society for music information retrieval conference (ISMIR), pp 93–98

Lartillot O, Toiviainen P (2007) Mir in Matlab (ii): a toolbox for musical feature extraction from audio. In: Proceedings of the 8th international conference on music information retrieval (ISMIR), pp 127–130

Li T, Ogihara M, Tzanetakis G (2011) Music data mining. CRC Press, USA

Liu J, Hu X (2010) User-centered music information retrieval evaluation. In: Proceedings of the joint conference on digital libraries (JCDL) workshop: music information retrieval for the masses

Livshin A, Rodet X (2006) 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

Loughrey J, Cunningham P (2004) Overfitting in wrapper-based feature subset selection: the harder you try the worse it gets. In: M. Bramer, F. Coenen, T. Allen (eds) Proceedings of the 24th SGAI international conference on innovative techniques and applications of artificial intelligence (AI-2004), pp. 33–43

Lukashevich H (2008) Towards quantitative measures of evaluating song segmentation. In: Proceedings of the 9th international conference on music information retrival (ISMIR), pp 375–380

McEnnis D, McKay C, Fujinaga I (2006) jAudio: additions and improvements. In: Proceedigs of the 7th international conference on music information retrieval (ISMIR), pp 385–386

Mierswa I, Morik K (2005) Automatic feature extraction for classifying audio data. Mach Learn J 58(2–3):127–149

Mierswa I, Wurst M, Klinkenberg R, Scholz M, Euler T (2006) Yale rapid prototyping for complex data mining tasks. In:Ungar L, Craven M, Gunopulos D, Eliassi-Rad T (eds) Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining (KDD). ACM Press, New York, pp 935–940

Müller M, Ewert S (2010) Towards timbre-invariant audio features for harmony-based music. IEEE Trans Audio Speech Lang Process 18(3):649–662

Müller M, Ewert S (2011) Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In: Proceedings of the 12th international conference on music information retrieval (ISMIR), pp 215–220

Nagathil A, Göttel P, Martin R (2011) Hierarchical audio classification using cepstral modulation ratio regressions based on Legendre polynomials. In: Proceedings of the international conference on acoustics, speech and signal processing (ICASSP), Prague, Czech Republic, pp 2216–2219

Park TH (2010) Introduction to digital signal processing: computer musically speaking. World Scientific, USA

Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann, Menlo Park

Ras ZW, Wieczorkowska A (2010) Advances in music information retrieval: studies in computational intelligence, vol 274. Springer, Berlin

Reynolds A, Corne D, Chantler M (2010) Feature selection for multi-purpose predictive models: a many-objective task. In: Proceedings of the 11th international conference on parallel problem solving from nature (PPSN). Springer, Berlin, pp 384–393

Rudolph G (2012) Evolutionary strategies. In: Rozenberg G, Bäck T, Kok J (eds) Handbook of natural computing. Springer, Berlin

Snyman J (2005) Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms. Springer, Berlin

Theimer W, Vatolkin I, Eronen A (2008) Definitions of audio features for music content description: technical report TR08-2-001. Chair of Algorithm Engineering, University of Dortmund, Germany

Vatolkin I (2012) Multi-objective evaluation of music classification. In: Gaul W, Geyer-Schulz A, Schmidt-Thieme L, Kunze L (eds) Challenges at the interface of data analysis, computer science, and optimization. Proceedings of the 34th annual conference of the Gesellschaft für Klassifikation e. V. Springer, Berlin, pp 401–410

Vatolkin I, Preuß M, Rudolph G (2011) Multi-objective feature selection in music genre and style recognition tasks. In: Krasnogor N, Lanzi PN (eds) Proceedings of the 2011 genetic and evolutionary computation conference (GECCO). ACM Press, New York, pp 411–418

Vatolkin I, Theimer W, Botteck M (2010) Amuse (advanced music explorer): a multitool framework for music data analysis. In: Downie JS, Veltkamp RC (eds) Proceedings of the 11th international society on music information retrieval conference (ISMIR), pp 33–38

Vatolkin I, Theimer W, Rudolph G (2009) Design and comparison of different evolution strategies for feature selection and consolidation in music classification. In: Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC). IEEE Press, pp 174–181

Weihs C, Ligges U, Mörchen F, Müllensiefen D (2007) Classification in music research. Adv Data Anal Classif 1(3):255–291

Zhu Z, Jia S, Ji Z (2010) Towards a memetic feature selection algorithm. IEEE Comput Intell Mag 5(2):41–53

Zitzler E, Brockhoff D, Thiele L (2007) The hypervolume indicator revisited: on the design of Pareto-compliant indicators via weighted integration. In: Proceedings of the conference on evolutionary multi-criterion optimization (EMO), vol 4403. Springer, Berlin, pp. 862–876