Optimal filtering of dynamics in short-time features for music organization

Jerónimo Arenas-García, Jan Larsen, Lars Kai Hansen and Anders Meng
Informatics and Mathematical Modelling
Technical University of Denmark
2800 Kgs. Lyngby, Denmark
{jag,jl,lkh,am}@imm.dtu.dk

Abstract

There is an increasing interest in customizable methods for organizing music collections. Relevant music characterization can be obtained from short-time features, but it is not obvious how to combine them to get useful information. In this work, a novel method, denoted as the Positive Constrained Orthonormalized Partial Least Squares (POPLS), is proposed. Working on the periodograms of MFCCs time series, this supervised method finds optimal filters which pick up the most discriminative temporal information for any music organization task. Two examples are presented in the paper, the first being a simple proof-of-concept, where an altosax with and without vibrato is modelled. A more complex 11 music genre classification setup is also investigated to illustrate the robustness and validity of the proposed method on larger datasets. Both experiments showed the good properties of our method, as well as superior performance when compared to a fixed filter bank approach suggested previously in the MIR literature. We think that the proposed method is a natural step towards a customized MIR application that generalizes well to a wide range of different music organization tasks.

Keywords: Music organization, filter bank model, positive constrained OPLS

1. Introduction

The interest in automated methods for organizing music is increasing, which is primarily due to the large digitalization of music. Music distribution is no longer limited to physical media, but users can download music titles directly from Internet services such as e.g. *iTunes* or *Napster* [1] . Portable players easily store most users personal collections and allow the user to bring the music anywhere. The problem of navigating these seemingly endless streams of music apparently seems dubious with current technologies. However, the increased research conducted in fields of music information retrieval (MIR) will aid users in organizing and navigating their music collections. Furthermore, there has been an increasing interest in customization when organizing the music, see e.g. [1, 2], which provides a better control of the users individual collections. The problems that researchers face when working with customization, especially in MIR, are many and indeed require robust machine learning algorithms for handling the large amount of data available for an average user. User interaction could be in the sense of organizing the music collection in specific taxonomies. This could be a simple flat genre taxonomy that is frequently used in portable players, or taxonomies based on instrumentation, artist or theme, see e.g. www.allmusic.com and [1]. Customization in terms of predicting users personal music taste was investigated in [3], where a support vector machine was applied in connection with active retrieval.

The present work introduces a method for learning important dynamical structure in the short-time features [2] extracted from the music, in such a way that this information is as relevant as possible for a given music organization task. The basic idea stems from the work of [4], where the authors investigated an audio classification task using different perceptual (and non-perceptual) short-time features at larger time-scales. A periodogram was computed for each short-time feature dimension over a frame corresponding to $\sim 768\,\mathrm{ms}$, followed by a summarization of the power in 4 predefined frequency bands using a filter bank. This method was investigated in greater detail in [5], where different methods for handling dynamics of short-time features, denoted as temporal feature integration [3] , were investigated. The fixed filter bank applied in [4], was selected from the assumed importance of the dynamics in the short-time features for the given learning task. The method, however, is not general enough, since for a custom music organization task, the dynamics in the short-time features are context dependent (i.e., the relevant pattern of temporal changes in short-time features is expected to be different for, e.g., vibrato/non vibrato detection, or for genre classification tasks), which is the reason for suggesting a method where an optimal filter bank is learned for a particular music organization task.

[1] www.itunes.com and www.napster.com.

[2] Short-time features are usually extracted from music at time-levels around $5 - 100\,\mathrm{ms}$.

[3] Temporal feature integration is the process of combining all the feature vectors in a time-frame into a single new feature vector in order to capture the relevant temporal information in the frame.

The content of this paper has been structured as follows. Section 2 presents the short-time features used and shortly describes the method in [4] for capturing the dynamic structure in the short-time features. Section 3 introduces the positive constrained OPLS, which can be used to find an optimal filter bank for any given music organization task. In Section 4, two experiments are described: the first experiment is a proof-of-concept illustrating the goodness of the filters obtained using the proposed method, when discriminating between vibrato/non-vibrato playing of music instruments; in the second experiment, we compare the filter banks derived from our method with those proposed in [4], within an 11 flat taxonomy music genre classification task. Section 5 provides the conclusion and suggestions for future work.

2. Music Feature Extraction

The complete system considered in this work has been illustrated in Figure 1. The purpose of the overall system is to classify music data according to some criterion, such as genre, or presence vibrato, so we are assuming that some labelled data is available for the design. From the raw digital audio signal, an initial step towards an automated organization of music is feature extraction. This is the process of extracting relevant information from the audio signal that can be used in a sub-sequential learning algorithm. A music signal is typically stationary in periods ranging from 5-100 ms, see e.g. [6], and features extracted at this time-scale are denoted short-time features.

2.1. Short-time features

The Mel Frequency Cepstral Coefficients (MFCC) have been selected as short-time features in this work. These coefficients were originally developed for automatic speech recognition, aiming at deconvolving the effects of the vocal tract shape and the vocal cord excitation. However, they have been applied with great success in various fields of MIR, see e.g. [7, 4, 3]. The features are perceptually inspired, meaning that they resemble the auditory system of humans. The MFCCs are ranked in such a manner that the lower order MFCCs contain information about the slow variations in the spectral envelope. Hence, including the higher MFCCs a richer representation of the spectral envelope will be obtained.

For this investigation, the 6 initial MFCCs have been used, including the first coefficient, which is correlated with the perceptual dimension of loudness. In the investigations, each music snippet is power normalized prior to the MFCC extraction stage. A frame-size of 30 ms and a hop-size of 7.5 ms have been applied in all experiments to minimize aliasing in the MFCCs.

2.2. Temporal feature integration

Temporal feature integration is the process of combining all the feature vectors in a time-frame into a single new feature vector in order to capture the relevant information in the frame. Formally, this amounts to the following (see also Fig. 1):

$$\mathbf{z}_k = \mathbf{f}(\mathbf{x}_{k\cdot h_{s_x}}, \mathbf{x}_{k\cdot h_{s_x}+1}, \ldots, \mathbf{x}_{k\cdot h_{s_x}+f_{s_x}-1}), \quad (1)$$

where $\mathbf{x}$ represents the short-time features (MFCCs), f_{s_x} is the frame-size, and h_{s_x} the hop-size, both defined in a number of sample manner. Function $\mathbf{f}(\cdot)$ maps the sequence of short-time features into a single vector $\mathbf{z}_k$, for $k = 0, 1, \ldots, K-1$.

In [4] it was proposed to perform temporal feature integration by estimating the power spectrum of the MFCCs using the periodogram method [8]. In addition to this, the authors propose to summarize the energy in different frequency bands using a predefined filter bank:

$$\tilde{\mathbf{z}}_k^{(i)} = \mathbf{W}^T \mathbf{z}_k^{(i)} \quad (2)$$

where $\mathbf{z}_k^{(i)}$ is a periodogram of dimension $D_{\mathbf{z}}$ of the i-th MFCC coefficient over some frame f_{s_x}, $k = 0, \ldots, K-1$ is the index at the larger time-scale, and $\mathbf{W}$ comprises the frequency magnitude response of the filter bank. Finally, the feature vector $\tilde{\mathbf{z}}^{(i)}$, which has as many components as the number of filters in the bank, is used as an input to the subsequent classification process.

In other words, the temporal feature extraction stage consists of estimating the periodogram of each MFCC dimension independently over some time-frame f_{s_x}, after which a filter bank $\mathbf{W}$ is applied. In the coming sections we have removed the superscript i, meaning that each short-time feature dimension is processed independently, using the same filter bank for all MFCCs.

The filter bank $\mathbf{W}$ is a matrix of dimension $D_{\mathbf{z}} \times 4$, where $D_{\mathbf{z}} = \frac{f_{s_x}}{2} + 1$ (throughout this paper we will use $f_{s_x} = 256$, so that $D_{\mathbf{z}} = 129$), which simply summarizes the power components in four frequency bands:

1. 0 Hz (DC value)
2. $1 - 2$ Hz (beat rates)
3. $3 - 15$ Hz (modulation energy, e.g. vibrato)
4. $20 - \frac{s_x}{2}$ Hz (perceptual roughness)

where the sampling rate s_x is related to the hop-size (h_{s_x}). This filter bank ($\mathbf{W}$) has been suggested for general audio classification and is inherently positive, since it is applied directly on the estimated power spectrum (periodogram). The filter bank, however, can easily become sub-optimal for a specific music organization task, which is the reason for suggesting a method for finding an optimal filter bank in a supervised manner. The proposed method for the optimal design of the filter bank is the topic of the next section.

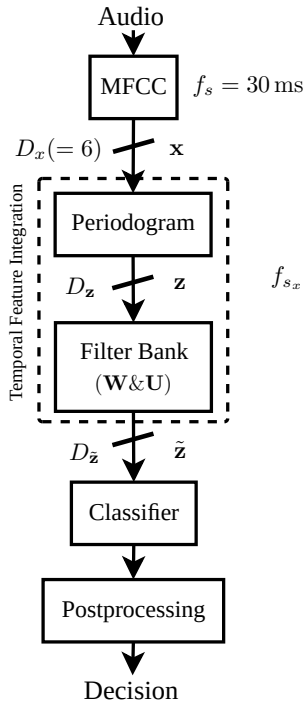

Figure 1. The figure illustrates the flow-chart of the complete process. After MFCC extraction, periodograms are computed for each MFCC. The output of the "periodogram" box is a $D_{\mathbf{z}} = 129$ dimensional vector for each MFCC, corresponding to the power in the different frequency bands. The filter bank ($\mathbf{W}$ or $\mathbf{U}$) summarizes the power in predefined frequency bands. The dimension of $\tilde{\mathbf{z}}$, denoted by $D_{\tilde{\mathbf{z}}}$ will depend on the number of MFCCs, the selected frame-size f_{s_x} and the number of filters in the filter bank $\mathbf{W}$ (fixed to 4) or $\mathbf{U}$ (n_f).

3. Supervised Design of Filter Banks

As can be understood from our previous discussion, and since the goal is to optimize the classification performance of the whole system, a better behavior can be obtained if the filter bank is designed in a supervised manner, i.e., to optimize the performance in some training dataset whose labels are used during the design process.

Then, we will assume that we are given a set of N training pairs $\{\mathbf{z}_k, \mathbf{y}_k\}_{k=1}^N$, with $\mathbf{y}_k$ being the label vector associated to $\mathbf{z}_k$. The C dimensional vector $\mathbf{y}_k$, where C is the number of classes, contains a one in the position of the true label for pattern $\mathbf{z}_k$, and zeros elsewhere. In this section, we address the issue of how one can use the training data to design a filter bank $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n_f}]$, where $\mathbf{u}_m$ is the frequency amplitude of the m-th filter and n_f is the total number of filters in the bank, in such a way that the outputs of the filters,

$$\tilde{\mathbf{z}}_k = \mathbf{U}^T \mathbf{z}_k, \tag{3}$$

are as relevant as possible for the classification task at hand [4].

From its definition, and given that this matrix operates on the power spectrum of the different MFCCs, it should be clear that all elements in $\mathbf{U}$ should be non-negative numbers (i.e., $u_{ij} \geq 0$), so that $\tilde{\mathbf{z}}_k$ can be effectively interpreted as the output energies of a filter bank. Note that a negative u_{ij} would correspond to the subtraction of the energy in a certain frequency band.

The procedure we present lies within the framework of Multivariate Analysis methods [9], and, in particular, it is a variant of Orthonormalized Partial Least Squares (OPLS). Next, we will briefly review OPLS, and explain how it can be solved under the additional constraints $u_{ij} \geq 0$, resulting in a method that we have called Positive constrained OPLS (POPLS). Readers that prefer to skip the implementation details of the method, can go directly to the experiments section.

3.1. Multi-regression model for Feature Extraction and Data Classification

For the classification process (see Fig. 1) we will consider a multi-regression model. Although other models are possible, we will see that the regression approach results in a very convenient method for computing the filter bank. The multi-regression model can be written as

$$\hat{\mathbf{y}}_k = \mathbf{B}\mathbf{U}^T \mathbf{z}_k + \mathbf{b} = \mathbf{B}\tilde{\mathbf{z}}_k + \mathbf{b}, \tag{4}$$

where $\hat{\mathbf{y}}_k$ is the predicted output, and $\{\mathbf{B}, \mathbf{b}\}$ are the free parameters of the model. In particular, $\mathbf{b}$ is a bias that compensates for the different means of the input and output variables. Note that, since $\mathbf{B}$ is $C \times n_f$ and $\mathbf{U}$ is $D_{\mathbf{z}} \times n_f$, the filter bank is effectively imposing a *bottleneck* in the system, in the sense that the $\tilde{\mathbf{z}}_k$ vectors given to the classifier are lower dimensional than the original $\mathbf{z}_k$. This dimensionality reduction is very useful to simplify the design of the classifier and to improve generalization, and is an unavoidable step when the training dataset is small. However, in order to not degrade the performance of the classifier, it is crucial that $\tilde{\mathbf{z}}_k$ retains the most discriminative information in $\mathbf{z}_k$, what can only be achieved with a good design of the filter bank.

Our aim is to adjust all parameters in the model, as well as the filter bank, to minimize the sum-of-squares of the differences between the real and estimated labels, i.e.,

$$[\mathbf{U}_o, \mathbf{B}_o, \mathbf{b}_o] = \arg\min_{\mathbf{U},\mathbf{B},\mathbf{b}} \|\mathbf{Y} - \mathbf{B}\tilde{\mathbf{Z}} - \mathbf{b}\mathbf{1}^T\|_F^2 \tag{5}$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, $\mathbf{1}$ is an all-ones vector of appropriate dimensions, and $\tilde{\mathbf{Z}} = \mathbf{U}^T\mathbf{Z}$ with $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$. Subscript '$F$' refers to the Frobenius norm of a matrix.

It is known that $\mathbf{B}_o$ can be obtained as the solution of the

[4] Note that we have opted to use $\mathbf{U}$ to denote the filter bank obtained with our method, to differentiate it from the filter bank from [4] that we will denote with $\mathbf{W}$ throughout the paper.

following modified problem:

$$\begin{aligned}\mathbf{B}_{\text{o}} &= \arg\min_{\mathbf{B}} \|\mathbf{Y}_c - \mathbf{B}\tilde{\mathbf{Z}}_c\|_F^2 \\ &= \mathbf{Y}_c\tilde{\mathbf{Z}}_c^T(\tilde{\mathbf{Z}}_c\tilde{\mathbf{Z}}_c^T)^{-1}\end{aligned} \tag{6}$$

where $\tilde{\mathbf{Z}}_c$ and $\mathbf{Y}_c$ are centered versions of $\tilde{\mathbf{Z}}$ and $\mathbf{Y}$, respectively. Then, the bias is simply given by

$$\mathbf{b}_0 = \tfrac{1}{N}(\mathbf{Y} - \mathbf{B}_0\tilde{\mathbf{Z}})\mathbf{1}. \tag{7}$$

Once we have derived a closed form expression for $\mathbf{B}_{\text{o}}$ and $\mathbf{b}_{\text{o}}$, we are ready to present our POPLS method for the selection of the optimal filter bank which minimizes (5) [10], subject to the constraint that all entries in $\mathbf{U}$ are positive.

3.2. Positive Constrained OPLS

To start with, let us introduce the optimal regression matrix, $\mathbf{B}_0$, into (5). Taking also into account that $\tilde{\mathbf{Z}}_c = \mathbf{U}^T\mathbf{Z}_c$, the minimization problem can be rewritten as

$$\begin{aligned}\mathbf{U}_{\text{o}} &= \arg\min_{\mathbf{U}} \|\mathbf{Y}_c - \mathbf{B}_{\text{o}}\tilde{\mathbf{Z}}_c\|_F^2 \\ &= \arg\min_{\mathbf{U}} \|\mathbf{Y}_c[\mathbf{I} - \mathbf{Z}_c^T\mathbf{U}(\mathbf{U}^T\mathbf{Z}_c\mathbf{Z}_c^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{Z}_c]\|_F^2\end{aligned}$$

with $\mathbf{I}$ being the N dimensional identity matrix.

Now, using the fact that $\|\mathbf{A}\|_F^2 = \text{Tr}\{\mathbf{A}\mathbf{A}^T\}$, and after some algebra, we arrive to the following optimization problem

$$\begin{aligned}&\text{maximize:} && \text{Tr}\{(\mathbf{U}^T\mathbf{C}_{zz}\mathbf{U})^{-1}\mathbf{U}^T\mathbf{C}_{zy}\mathbf{C}_{yz}\mathbf{U}\} && (8)\\ &\text{subject to:} && \mathbf{U}^T\mathbf{U} = \mathbf{I} && (9)\\ & && u_{ij} \geq 0 && (10)\end{aligned}$$

where we have defined the covariance matrices $\mathbf{C}_{zz} = \mathbf{Z}_c\mathbf{Z}_c^T$, $\mathbf{C}_{zy} = \mathbf{Z}_c\mathbf{Y}_c^T$ and $\mathbf{C}_{yz} = \mathbf{C}_{zy}^T$, and where we have made explicit the positivity constraint. The additional constraint (9) is needed to make the solution unique.

There are a number of ways to solve the above problem. We will use a procedure consisting on iteratively calculating the best filter, so that we are not only guaranteeing that $\mathbf{U}_{\text{o}}$ is the optimal bank with n_f filters, but also that any subbank consisting of some of the first columns of $\mathbf{U}_{\text{o}}$ is also optimal with respect to the number of filters used. In brief, the process consists of the following two differentiated stages:

1) Solve the "one filter" optimization problem given by:

$$\begin{aligned}&\text{maximize:} && \frac{\mathbf{u}^T\mathbf{C}_{zy}\mathbf{C}_{yz}\mathbf{u}}{\mathbf{u}^T\mathbf{C}_{zz}\mathbf{u}} && (11)\\ &\text{subject to:} && \mathbf{u}^T\mathbf{u} = 1 && (12)\\ & && u_i \geq 0 && (13)\end{aligned}$$

2) Remove from Y_c the prediction obtained from the current filter bank.

Inputs: $\mathbf{Z}$, $\mathbf{Y}$, n_f
1 - Calculate centered data matrices $\mathbf{Z}_c$ and $\mathbf{Y}_c$
2 - $\mathbf{C}_{zz} = \mathbf{Z}_c\mathbf{Z}_c^T$, $\mathbf{Y}_c^{(1)} = \mathbf{Y}_c$
3 - For $m = 1, \ldots, n_f$
  3.1 - $\mathbf{C}_{yz}^{(m)} = \mathbf{Y}_c^{(m)}{\mathbf{Z}_c}^T$; $\mathbf{C}_{zy}^{(m)} = {\mathbf{C}_{yz}^{(m)}}^T$
  3.2 - Solve (11)-(13) to obtain $\mathbf{u}_m$
  3.3 - $\mathbf{Y}^{(m+1)} = \mathbf{Y}^{(m)}\left[\mathbf{I} - \frac{\mathbf{Z}_c^T\mathbf{u}_m\mathbf{u}_m^T\mathbf{Z}_c}{\mathbf{u}_m^T\mathbf{C}_{zz}\mathbf{u}_m}\right]$
4 - Output filter bank: $\mathbf{U}_{\text{o}} = [\mathbf{u}_1, \ldots, \mathbf{u}_{n_f}]$

Table 1. POPLS pseudocode.

Table 1 summarizes our POPLS algorithm for the supervised design of filter banks. It is also worth mentioning that, in our implementation, the maximization problem (11)-(13) was solved with the *fmincon* matlab function. However, in most occasions, the convergence of this routine was not satisfactory, making it necessary to recur to an alternative representation of $\mathbf{u}$ based on hyperspherical coordinates. The advantage of this representation is that restriction (12) is directly incorporated into the representation, what simplifies the application of any optimization algorithm.

4. Experiments

This section considers two different experiments. The experiment described in Subsection 4.1 is a proof-of-concept experiment that illustrates the basic idea of POPLS for discriminating between an instrument played with and without vibrato. The second line of experiments described in Subsection 4.2 considers an 11 music genre dataset, investigated using the filter $\mathbf{W}$ from [4] and the filter obtained from the POPLS, $\mathbf{U}$.

4.1. Experiment 1: Instrument vibrato/non-vibrato detection

This experiment considers the problem of detecting vibrato or non-vibrato of a single instrument and is only intended as a proof-of-concept example.

A small dataset has been created consisting of music snippets consisting of an alto saxophone with notes ranging from $\text{D}_\text{b}3$ to $\text{A}_\text{b}5$ ($138.59 - 830.61$ Hz), with and without vibrato, resulting in a total of 64 (32 train / 32 test) small music clips each of $3 - 4$ s. The music samples were extracted from the MIS (Music Instrument Samples) database developed by the university of Iowa [11]. This database has been applied in connection with automated instrument classification in e.g. [12].

Only the first MFCC has been used in this experiment, which is known to be correlated with the perceptual dimension of loudness. A frame-size (f_{s_x}) corresponding to 960 ms and a corresponding hop-size of 240 ms were selected. The frame-size was selected to ensure a few periods of the mod-

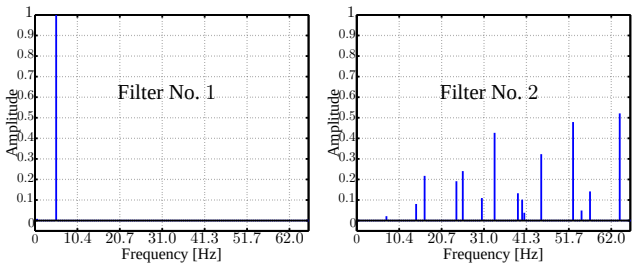

Figure 2. The left and right figure illustrates the first and second most discriminative filters extracted by the POPLS procedure, respectively.

ulation frequency of $4-5$ Hz, hence, obtaining a better spectral estimate for the instruments played with vibrato.

Leave-one-out cross-validation (LOO-CV) [13] was applied to access the test-accuracy. In each fold, the optimal filters were calculated using the POPLS method described in Subsection 3.2, and the resulting error was obtained using a linear classifier. The LOO-CV classification error obtained using $n_f = 25$ filters was 19% at the 960 ms time-scale, getting as low as 9.4% when performing weigted voting [5] across the frames in each music sample to achieve a single decision of each music sample. It is noted that, when using the fixed filter bank $\mathbf{W}$, close to random performance (48.3%, where random performance is 50%) is obtained, which is ascribed to a smearing of the relevant frequency components, since the filter is summarizing the frequencies between $3-15$ Hz.

The two filters with largest discriminative performance provided by POPLS have been illustrated in Figure 2. The left figure, which illustrates the filter with largest discriminative performance, clearly indicates that the most relevant information concerning the modulation (vibrato $\sim 4-6$ Hz) of the instrument is learned by the POPLS. Using only these two filters a classification error of 20% is obtained using weighted voting to obtain a single decision per music sample.

4.2. Experiment 2: Music genre classification

The experiment described in this subsection considers the fixed filter bank $\mathbf{W}$ and the POPLS method for determining a filter bank $\mathbf{U}$ in an 11 music genre classification setup.

4.2.1. Dataset

The dataset has previously been investigated in [5, 14], and consists of 1317 music snippets each of 30 s. distributed evenly among the 11 music genres: alternative, country, easy listening, electronica, jazz, latin, pop&dance, rap&hip-hop, r&b and soul, reggae and rock, except for latin, which only has 117 music samples. The labels have been obtained from an external reference. The music snippets consist of

[5] Weighted voting is the process of selecting class membership by summing across the output vectors of the classifier, $\hat{\mathbf{y}}_k$, corresponding to all feature vectors $\tilde{\mathbf{z}}_k$ belonging to the same clip. The class that obtains the largest sum is the "voted" class.

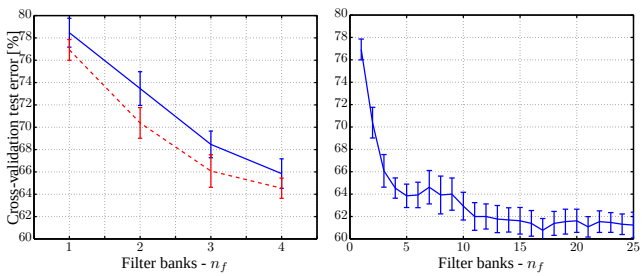

Figure 3. The left figure illustrates the mean cross-validation error for the fixed filter bank ($\mathbf{W}$, solid line) and the first 4 filters of the POPLS procedure ($\mathbf{U}$, broken line). The right figure illustrates the mean cross-validation error for the $n_f = 25$ filters obtained by the POPLS procedure. The error-bars on both plots are $\pm$ the standard deviation of the mean.

MP3 (MPEG1-layer3) encoded music with a bitrate of 128 kbps or higher, downsampled to 22050 Hz. This dataset is rather complex having on the average 1.83 songs per artist. Previous results show that this is a difficult dataset for genre classification (see, for instance, [14]).

4.2.2. Initial investigations

Previous investigations of the frame-size conducted in [5] showed that a frame-size (f_{s_x}) of approximately 2 s was optimal for the method in [4]. Since the aim is to illustrate that a supervised determination of the filter bank is superior to the fixed filter bank, the same frame-size has been used for POPLS. With a hop-size on the short-time features of 7.5 ms, this frame-size corresponds to approximately 256 samples. Due to the symmetry in the periodogram, the resulting dimensions of the filter banks become $129 \times n_f$ for $\mathbf{U}$ and 129×4 for $\mathbf{W}$. It was observed that the mean classification test error did not improve for $n_f > 25$, hence, $n_f = 25$ was the largest amount of filters investigated.

4.2.3. Results & Discussion

To access the classification accuracy of the two methods, 10-fold cross-validation has been applied. In each fold, the optimal filters were estimated from the training set as described in Section 3, and the performance of the system was subsequently evaluated on the corresponding test fold.

Figure 3 shows the 10-fold cross-validation error as a function of the number of filters in the banks. The left figure shows the cross-validation error obtained using only the first 4 filters of the POPLS, and using the fixed filter bank $\mathbf{W}$. It is observed that the filters obtained by the POPLS procedure are on the average 2% better than the fixed filter bank, shown in solid line. Furthermore, using only the first 3 filter banks obtained by POPLS the cross-validation test error is similar to the performance obtained using the fixed filter bank $\mathbf{W}$. Although most of the important dynamical structure of the MFCCs is captured by the first few filters of $\mathbf{U}$, the right plot of Figure 3 shows that a significant error reduction can be obtained when considering a larger number of filters, achieving error rates around 61% for $n_f > 15$.

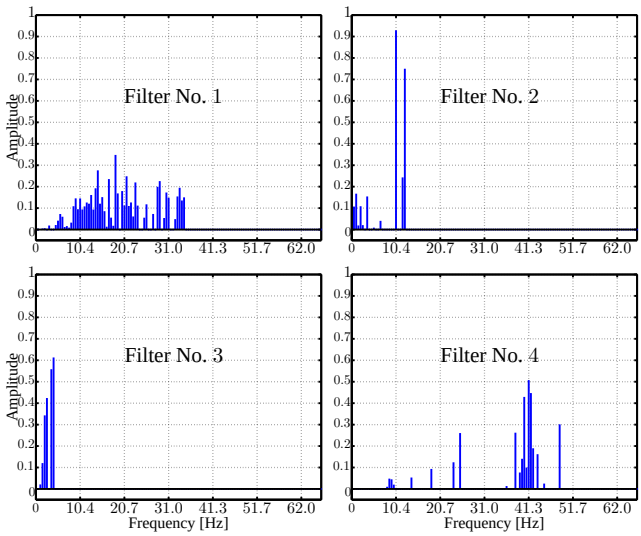

Figure 4. The four most discriminative filters, where the upper left figure illustrates the most relevant filter for the specific music genre classification setup.

Figure 4 shows the first 4 filters obtained on a single fold using the POPLS. Filter 1 includes the most important frequencies of the MFCCs periodograms, which basically cover the modulation frequencies of instruments. Filters 2 and 3 provide attention to the lower modulation frequencies, with filter 2 having frequency components at the beat-scale. Filter 4 spans the higher modulation frequencies, which are related to the perceptual roughness. The difference between the filters obtained for each training data fold is small, which partly illustrates that the proposed method is robust to noise and, further, that the specific underlying temporal structure of the MFCCs is relevant for discriminating between the different genres.

5. Conclusions and Future work

In this paper we have presented a method for designing filter banks that are able to learn the important dynamics in short-time features for a given classification task. The proposed method is very versatile, in the sense that it can be applied to any discrimination task, as we have illustrated in our experiments section, where we tackled two very different classification problems, namely, the detection of vibrato in instrument music clips, and music genre classification. The advantage of our approach over other feature extraction methods, is that it provides an elegant physical interpretation of the extracted features, in terms of the dynamical behavior of the MFCCs time series.

Although here we limited the method to provide an unique filter bank, it is straightforward to allow for different filters for each of the MFCCs. Exploiting the cross-correlation among the different filters in the bank, could also be used to improve the accuracy of the whole system. These lines, as well as the application of the method to other MIR problems, constitute logical directions for future research.

6. Acknowledgments

This work is partly supported by the Danish Technical Research Council, through the framework project 'Intelligent Sound', www.intelligentsound.org (STVF No. 26-04-0092), and by the European Commission through the sixth framework IST Network of Excellence: Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), contract no. 506778. The work of J. Arenas-García was also supported by the Spanish Ministry of Education and Science with a postdoctoral fellowship.

References

[1] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst. A benchmark dataset for audio classification and clustering. In *International Symposium on Music Information Retrieval*, pages 528–531, 2005.

[2] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kgl. Meta-features and adaboost for music classification. *Machine Learning Journal : Special Issue on Machine Learning in Music*, 2006.

[3] M. Mandel, G. Poliner, and D. Ellis. Support vector machine active learning for music retrieval. *Accepted for publication in ACM Multimedia Systems Journal*, 2006.

[4] M. F. McKinney and J. Breebart. Features for audio and music classification. In *International Symposium on Music Information Retrieval*, pages 151–158, 2003.

[5] A. Meng, P. Ahrendt, J. Larsen, and L. K. Hansen. Temporal feature integration for music genre classification. Submitted, 2006.

[6] J.-J. Aucouturier, F. Pachet, and M. Sandler. The way it sounds : Timbre models for analysis and retrieval of polyphonic music signals. *IEEE Trans. on Multimedia*, 7(6):8, December 2005. (in press).

[7] P. Ahrendt, A. Meng, and J. Larsen. Decision time horizon for music genre classification using short time features. In *EUSIPCO*, pages 1293–1296, Vienna, Austria, sept. 2004.

[8] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, N.Y.: Wiley, 1996.

[9] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, 3rd Ed., N.Y.: Wiley-Interscience, 2003.

[10] S. Roweis and C. Brody. Linear Heteroencoders. *Gatsby Unit Technical Report GCNU-TR-1999-02*, Gatsby Computational Neuroscience Unit, London, 1999.

[11] University of Iowa musical instrument sample database, `http://theremin.music.uiowa.edu/index.html`.

[12] E. Benetos, M. Kotti, C. Kotropoulos, J. J. Burred, G. Eisenberg, M. Haller, and T. Sikora. Comparison of subspace analysis-based and statistical model-based algorithms for musical instrument classification. In *Proc. of 2nd Workshop On Immersive Communication And Broadcast Systems (ICOB)*, October 2005.

[13] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[14] A. Meng and J. Shawe-Taylor. An investigation of feature models for music genre classification using the Support Vector Classifier. In *International Conference on Music Information Retrieval*, pages 604–609, 2005.