Multivariate Analysis and Kernel Methods for Music Data Analysis

Jerónimo Arenas-García, Anders Meng, Kaare Brandt Petersen and Lars Kai Hansen
Informatics and Mathematical Modelling
Technical University of Denmark
DK-2800 Kongens Lyngby, Denmark
{jag,am,kbp,lkh}@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. First, the relevant information might not be evident at the short-time level, and these features have to be combined at a larger temporal level into a new feature vector in order to capture the relevant information. Second, we need to learn a model for the new features that generalizes well to new data. In this contribution, we will study how multivariate analysis (MVA) and kernel methods can be of great help in this task. More precisely, we will present two modified versions of a MVA method known as Orthonormalized Partial Least Squares (OPLS), one of them being a kernel extension, that are well-suited for discovering relevant dynamics in large music collections. The performance of both schemes will be illustrated in a music genre classification task.

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 considers the problem of 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

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

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

for a given music organization task. After applying some kind of temporal feature integration[3], new discriminative features will be extracted using new extensions of a multivariate analysis (MVA) method known as Orthonormalized Partial Least Squares (OPLS). It will be shown that the new methods, proposed in [4] and [5], are well-suited for music data and are able to cope with large data sets, while providing competitive performance and, if desired, a clear physical interpretation for the derived features.

The rest of the paper is organized as follows: the next Section will review the standard OPLS algorithm for feature extraction; Sections 3 and 4 will be dedicated to the two OPLS extensions that we have recently proposed; in Section 5 we will illustrate the performance of both approaches in a genre music classification task, and Section 6 will conclude the paper.

2 Orthonormalized Partial Least Squares

Consider we are given a set of pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{l}$, with $\mathbf{x}_i \in \Re^N$, $\mathbf{y}_i \in \Re^M$. Let us also introduce matrices $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_l]^T$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_l]^T$, and denote by

$$\mathbf{X}' = \mathbf{X}\mathbf{U} \quad \text{and} \quad \mathbf{Y}' = \mathbf{Y}\mathbf{V}$$

two matrices, each one containing n_p projections of the original input and output data, $\mathbf{U}$ and $\mathbf{V}$ being the projection matrices of sizes $N \times n_p$ and $M \times n_p$, respectively. The objective of Multivariate Analysis (MVA) algorithms is to search for projection matrices such that the projected input and output data are maximally aligned. For instance, Canonical Correlation Analysis (CCA) finds the projections that maximize the correlation between the projected data, while Partial Least Squares (PLS) provides the directions for maximum covariance:

$$\begin{array}{lll} \text{PLS :} & \text{maximize:} & \text{Tr}\{\mathbf{U}^T\mathbf{C}_{xy}\mathbf{V}\} \\ & \text{subject to:} & \mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I} \end{array} \tag{1}$$

where $\mathbf{I}$ is the identity matrix of size n_p, the T superscript denotes matrix or vector transposition, and where we have also defined the covariance matrix $\mathbf{C}_{xy} = \tilde{\mathbf{X}}^T\tilde{\mathbf{Y}}$, $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ being the centered versions of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

In this paper, we will consider a different MVA method, namely, the Orthonormalized Partial Least Squares (OPLS) which tackles the following maximization problem:

$$\begin{array}{lll} \text{OPLS :} & \text{maximize:} & \text{Tr}\{\mathbf{U}^T\mathbf{C}_{xy}\mathbf{C}_{xy}^T\mathbf{U}\} \\ & \text{subject to:} & \mathbf{U}^T\mathbf{C}_{xx}\mathbf{U} = \mathbf{I} \end{array} \tag{2}$$

with $\mathbf{C}_{xx} = \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$.

Note that, unlike CCA or PLS, OPLS only extracts projections of the input data. It is known that OPLS is optimal for performing linear regression on the input data when a bottleneck is imposed for data dimensionality reduction [6]. In other words, the solution to (2) also minimizes the sum of squares of the residuals of the approximation of the label matrix:

$$\|\tilde{\mathbf{Y}} - \tilde{\mathbf{X}}'\hat{\mathbf{B}}\|_F^2, \qquad \hat{\mathbf{B}} = (\mathbf{C}_{xx})^{-1}\mathbf{C}_{xy} \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix and $\hat{\mathbf{B}}$ is the optimal regression matrix. Similarly to other MVA methods, OPLS is not only useful for multi-regression problems, but it can also be used as a feature extractor in supervised problems, including also the multi-label case, when $\mathbf{Y}$ is used to encode class membership information. The optimality condition suggests that the features obtained by OPLS will be more relevant than those provided by other MVA methods, in the sense that they will allow similar or better accuracy rates using fewer projections.

Before moving on to our modified versions of OPLS, it is worth pointing out two properties of MVA algorithms that make them well suited for music data analysis:

- Flexibility: The output space can be used to encode any kind of information relevant to the music organization task at hand. For instance, in this paper, we will use $\mathbf{Y}$ to encode the

[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.

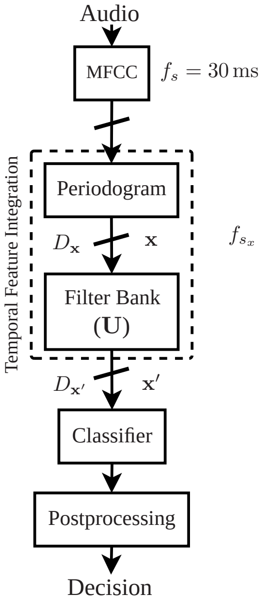


Figure 1: The figure illustrates the flow-chart of the complete process. After MFCC extraction, periodograms are computed for each MFCC. The output of the "periodogram" box is a $D_{\mathbf{x}} = 129$ dimensional vector for each MFCC, corresponding to the power in the different frequency bands. The filter bank $(\mathbf{U})$ summarizes the power in different frequency bands.

genre information for the songs in a training data set (using one-out-of-C encoding), but in previous works we also considered feature extraction for instrument classification [7] and for detecting the presence of vibrato in instrument recordings [4]. Furthermore, MVA algorithms can also be used in problems with multiple labels (e.g., when soft or multiple membership is allowed), and in regression problems (e.g., if $\mathbf{Y}$ is used to encode the ratings given by a user to different songs).

- Scalability: MVA methods extract the projection vectors using the covariance matrices only. Since these can be computed through a sum over all patterns, MVA can certainly be applied with large data sets (which is usually the case in music organization tasks), without having to keep all training data in memory. This property allows also the implementation of incremental learning schemes, which are useful when more data becomes available as time goes by.

3 Filtering of short-time dynamics using Positive Constrained OPLS

The complete system considered in this section has been illustrated in Figure 1. The purpose of the overall system is to classify music data according to some criterion, such as genre, 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. A music signal is typically stationary in periods ranging from 5-$100\,\mathrm{ms}$, see e.g., [8], and features extracted at this time-scale are denoted short-time features.

3.1 Short-time features

The Mel Frequency Cepstral Coefficients (MFCC) have been selected as short-time features in this work. 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 $30\,\mathrm{ms}$ and a hop-size of $7.5\,\mathrm{ms}$ have been applied in all experiments to minimize aliasing in the MFCCs.

3.2 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.

In [9] it was proposed to perform temporal feature integration by estimating the power spectrum of the MFCCs using the periodogram method [10]. In addition to this, the authors propose to summarize the energy in different frequency bands using a predefined filter bank:

$$\mathbf{x}'_i = \mathbf{U}^T \mathbf{x}_i \tag{4}$$

where $\mathbf{x}_i$ is a periodogram of dimension $D_{\mathbf{x}}$ of any of the MFCC coefficients over some frame f_{s_x}, and $\mathbf{U}$ comprises the frequency magnitude response of the filter bank. Finally, the feature vector $\mathbf{x}'_i$, 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 $\mathbf{U}$ is applied. In [9] it was proposed to use a filter bank with four bandpass filters whose frequency responses where selected according to the believed importance of each frequency for the genre classification task.

3.3 Supervised Design of Filter Banks

Rather than using a predetermined filter bank, we propose to make a supervised design using the periodograms extracted from some collection of labelled music data. Then, we can select the frequency response of each filter by using the OPLS algorithm; however, there is an additional constraint that we should take into consideration: note that, from its definition, and given that $\mathbf{U}$ operates on the power spectrum of the different MFCCs, all elements in $\mathbf{U}$ should be non-negative numbers so that the extracted features, $\{\mathbf{x}'_i\}$, can be effectively interpreted as the energy of the periodograms in different frequency ranges.

When consider the positivity constraint, we can formulate the Positive Constrained OPLS (POPLS) as:

$$\begin{array}{lll} \text{POPLS :} & \text{maximize:} & \text{Tr}\{\mathbf{U}^T\mathbf{C}_{xy}\mathbf{C}_{xy}^T\mathbf{U}\} \\ & \text{subject to:} & \mathbf{U}^T\mathbf{C}_{xx}\mathbf{U} = \mathbf{I} \\ & & u_{ij} \geq 0 \end{array} \tag{5}$$

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 we are computing an optimal bank with n_p filters, but also that any subbank consisting of some of the first columns of $\mathbf{U}$ is 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:

$$\text{maximize:} \quad \mathbf{u}^T\mathbf{C}_{xy}\mathbf{C}_{xy}^T\mathbf{u} \tag{6}$$

$$\text{subject to:} \quad \mathbf{u}^T\mathbf{C}_{xx}\mathbf{u} = 1 \tag{7}$$

$$u_i \geq 0 \tag{8}$$

2) Remove from the label matrix the prediction obtained from the current filter bank.

Finally, it is worth mentioning that the positivity constraint obviously leads to a suboptimal discriminative performance of the extracted features in comparison to standard OPLS. This is the price we pay in exchange for deriving features with a clear physical interpretation. We will later check that, when using enough filters, the discriminative power of the features is still quite satisfactory.

4 Sparse Kernel OPLS

When a physical interpretation of the extracted features is not required, one can increase the discrimination power of the extracted features by using more powerful, possibly non-linear, techniques. In this section we consider the kernel extension of the OPLS algorithm [5].

The overall feature extraction scheme is very similar to that in Figure 1, but important differences exist in the temporal feature integration block. First, rather than extracting the periodograms of the MFCCs, we consider adjusting an Autorregresive (AR) prediction model, as it was proposed in [11]. The parameters of this model are then stacked in a columnwise manner, forming what we call a vector of AR coefficients. These AR coefficients are projected into a very high dimensional feature space, where linear OPLS is finally applied.

To be more explicit, in this case we consider that we are given a set of pairs $\{\boldsymbol{\phi}(\mathbf{x}_i), \mathbf{y}_i\}_{i=1}^{l}$, with $\mathbf{x}_i \in \Re^N$ being vectors of AR coefficients and $\mathbf{y}_i \in \Re^M$ their associated labels, and $\boldsymbol{\phi}(\mathbf{x}) : \Re^N \rightarrow \mathcal{F}$ a function that maps the input data into some Reproducing Kernel Hilbert Space (RKHS), usually referred to as feature space, of very large or even infinite dimension. We also need to introduce matrix $\mathbf{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_l)]^T$ and $\mathbf{\Phi}' = \mathbf{\Phi}\mathbf{U}$ for the n_p dimensional extracted features. Now, $\mathbf{U}$ has dimensions $dim(\mathcal{F}) \times n_p$. Accordingly, the new Kernel OPLS problem is given by:

$$\text{KOPLS :} \qquad \begin{array}{ll} \text{maximize:} & \text{Tr}\{\mathbf{U}^T \tilde{\mathbf{\Phi}}^T \tilde{\mathbf{Y}} \tilde{\mathbf{Y}}^T \tilde{\mathbf{\Phi}} \mathbf{U}\} \\ \text{subject to:} & \mathbf{U}^T \tilde{\mathbf{\Phi}}^T \tilde{\mathbf{\Phi}} \mathbf{U} = \mathbf{I} \end{array} \tag{9}$$

where $\tilde{\mathbf{\Phi}}$ is a centered version of $\mathbf{\Phi}$.

When projecting data into an infinite dimensional space, we need to use the Representer Theorem that states that each of the projection vectors in $\mathbf{U}$ can be expressed as a linear combination of the training data. However, as explained in [5], when dealing with large data sets it is normally more convenient to impose sparsity in the projection vectors representation, i.e., we will use the approximation $\mathbf{U} = \mathbf{\Phi}_R^T \mathbf{B}$, where $\mathbf{\Phi}_R$ is a subset of the training data containing only R patterns $(R < l)$ and $\mathbf{B} = [\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_{n_p}]$ contains the parameters of the compact model. Although more sophisticated strategies can be followed in order to select the training data to be incorporated into the basis $\mathbf{\Phi}_R$, we will rely on random selection, very much in the line of the sparse greedy approximation proposed in [12] to reduce the computational burden of Support Vector Machines (SVMs).

Replacing $\mathbf{U}$ in (9) by its approximation, we get an alternative maximization problem that constitutes the basis for a KOPLS algorithm with reduced complexity (rKOPLS):

$$\text{rKOPLS :} \qquad \begin{array}{ll} \text{maximize:} & \text{Tr}\{\mathbf{B}^T \mathbf{K}_R \tilde{\mathbf{Y}} \tilde{\mathbf{Y}}^T \mathbf{K}_R^T \mathbf{B}\} \\ \text{subject to:} & \mathbf{B}^T \mathbf{K}_R \mathbf{K}_R^T \mathbf{B} = \mathbf{I} \end{array} \tag{10}$$

where we have defined $\mathbf{K}_R = \mathbf{\Phi}_R \tilde{\mathbf{\Phi}}^T$, which is a reduced kernel matrix of size $R \times l$.

There are different ways to solve the rKOPLS problem. Among them, we will rely on a two stage procedure similar to the one we used for POPLS, and basically consisting in solving the one-dimensional problem, followed by deflation (see [5] for further details).

What is interesting to our discussion here is that, apart from providing the high expressive power inherent to kernel methods, the proposed scheme still scales well with the number of training patterns, which, as we have already explained, is critical if the method is to be applied to large collections of music. Indeed, it is possible to check that matrices $\mathbf{K}_R \mathbf{K}_R^T$ and $\mathbf{K}_R \tilde{\mathbf{Y}}$ are of size $R \times R$ and $R \times M$, respectively, and that they can be computed using a sum over all training patterns. Additionally, the extraction of features for new unseen data requires only the computation of R kernels. Note, however, that the proposed scheme is very different from using simple subsampling, given that matrix $\mathbf{K}_R$ still takes into account all training data.

5 Experiments

In this section we illustrate the overall performance of the systems based on POPLS and rKOPLS in a genre classification task. To keep things as simple as possible, we use one of the simplest classifiers:

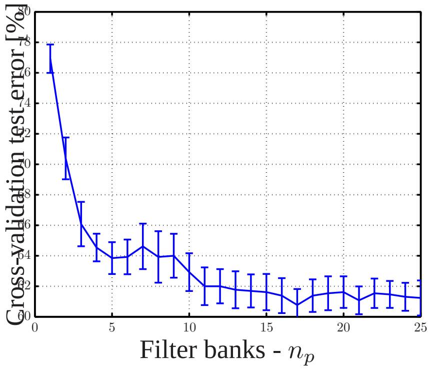


Figure 2: The figure illustrates the average number of songs misclassified when using the $n_p = 25$ filters obtained by the POPLS procedure.

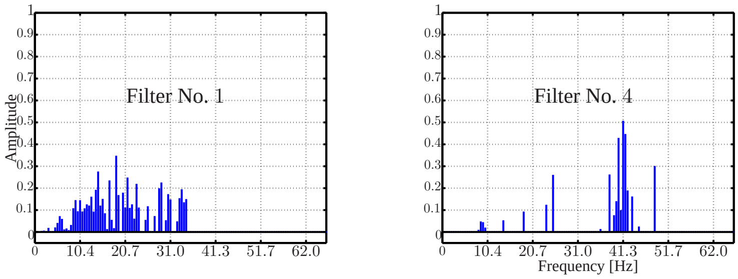


Figure 3: The first and fourth most discriminative filters for the genre classification task.

we compute the pseudoinverse of the projected training data to calculate $\hat{\mathbf{B}}$ (see Eq. (3)), and then classify data according to $\tilde{\mathbf{X}}'\hat{\mathbf{B}}$ or $\tilde{\mathbf{\Phi}}'\hat{\mathbf{B}}$ (for the POPLS and the rKOPLS approaches, respectively), using a “winner-takes-all” (w.t.a) activation function.

For the kernel approach we used a Gaussian kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / 2\sigma^2\right)$$

using 10-fold cross-validation (10-CV) on the training set to estimate σ.

5.0.1 Dataset description

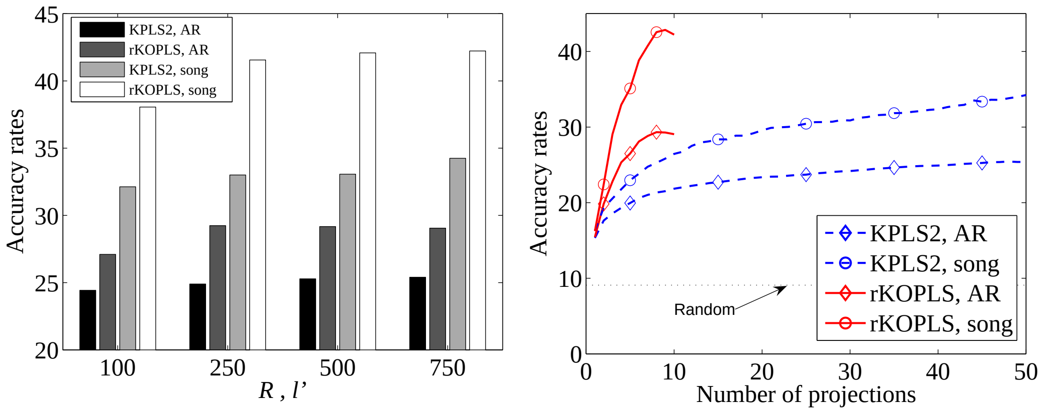
The dataset has previously been investigated in [11, 13], and consists of 1317 music snippets each of $30\,\mathrm{s}$. distributed evenly among the 11 music genres: alternative, country, easy listening, electronica, jazz, latin, pop&dance, rap&hip-hop, r&b and soul, reggae and rock, except for latin, which only has 117 music samples. The labels have been obtained from an external reference. The music snippets consist of MP3 (MPEG1-layer3) encoded music with a bitrate of $128\mathrm{kbps}$ or higher, downsampled to $22050\,\mathrm{Hz}$. 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, [13]).

Since every song consists of about seventy AR vectors, we can measure the classification accuracy in two different ways: 1) On the level of individual AR vectors or 2) by majority voting among the AR vectors of a given song.

5.1 Performance with POPLS features

Figure 2 shows the 10-fold cross-validation song classification error as a function of the number of filters in the bank. Although most of the important dynamical structure of the MFCCs is captured by the first few filters of $\mathbf{U}$, the figure shows that a significant error reduction can be obtained when considering a larger number of filters, achieving error rates around 61% for $n_p > 15$.

Figure 3 shows the first and fourth 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



(a) (b)

Figure 4: Genre classification performance of KPLS2 and rKOPLS.

frequencies of instruments. Filters 2 and 3 (not shown here) provide attention to the lower modulation frequencies. 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.

5.2 Performance with rKOPLS features

In this section we illustrate the classification accuracy of the classifiers built using rKOPLS features. For comparison purposes we include also the results achieved with a different kernel MVA technique. We have considered the Kernel Partial Least Squares algorithm described in [14], to which we refer in the following as KPLS2.

In Figure 4 the results are given both with respect to the accuracy at classifying AR vectors, and when considering the overall classification of a song. The results are very clear: compared to KPLS2, rKOPLS is not only consistently performing better as seen in Figure 4(a), but is also doing so with much fewer projections. These conclusions are very pronounced when looking at Figure 4(b) where, for $R = 750$, rKOPLS is outperforming ordinary KPLS, and is doing so with only ten projections compared to fifty projections of the KPLS2. This demonstrates that the features extracted by rKOPLS holds much more information relevant to the genre classification task than KPLS2.

6 Conclusions

In this contribution we have illustrated the relevance of MVA and kernel methods to learn the dynamics of short-time features which are relevant to a particular classification task. The features computed using modified OPLS schemes have shown to have good discrimination power, while retaining physical interpretation if needed.

We can conclude that MVA techniques are very versatile, in the sense that they can be applied to any discrimination task. Additionally, because of the nature of music data, in which both the number of dimensions and samples are very large, we believe that feature extraction methods such as POPLS and rKOPLS can become crucial to music information retrieval tasks, and hope that other researchers in the community will be able to benefit from our results.

Acknowledgments

This work was 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 Spanish Ministry of Education and Science with a Postdoctoral Felowship to the first author.

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] J. Arenas-García, J. Larsen, L. K. Hansen and A. Meng Optimal filtering of dynamics in short-time for music organization In *Proc. 7th Intl. Conf. on Music Information Retrieval*. Victoria, Canada, pp. 290–295, Oct. 2006.

[5] J. Arenas-García, K. B. Petersen and L. K. Hansen. Sparse Kernel Orthonormalized PLS for feature extraction in large data sets. To be presented at NIPS 2006.

[6] S. Roweis and C. Brody. Linear heteroencoders. Technical report, Gatsby Computational Neuroscience Unit, 1999.

[7] A. B. Nielsen, S. Sigurdsson, L. K. Hansen and J. Arenas-García, On the relevance of spectral features for instrument classification. Submitted to ICASSP'07.

[8] 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.

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

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

[11] Anders Meng, Peter Ahrendt, Jan Larsen, and Lars Kai Hansen. Temporal feature integration for music genre classification. *IEEE Trans. Audio, Speech & Language Process.*, to appear.

[12] Yuh-Jye Lee and O. L. Mangasarian. RSVM: reduced support vector machines. In *Data Mining Institute Technical Report 00-07, July 2000. CD Proceedings of the SIAM International Conference on Data Mining, Chicago, April 5-7, 2001,*, 2001.

[13] 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.

[14] John Shawe-Taylor and Nello Christiani. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.