# Musical instrument classification using non-negative matrix factorization algorithms

Emmanouil Benetos, Margarita Kotti, Constantine Kotropoulos
Artificial Intelligence and Information Analysis Laboratory
Department of Informatics, Aristotle Univ. of Thessaloniki
Box 451, Thessaloniki 541 24, Greece
E-mail: {empeneto, mkotti, costas}@aiia.csd.auth.gr

*Abstract*— **In this paper, a class of algorithms for automatic classification of individual musical instrument sounds is presented. Several perceptual features used in general sound classification applications were measured for 300 sound recordings consisting of 6 different musical instrument classes (piano, violin, cello, flute, bassoon and soprano saxophone). In addition, MPEG-7 basic spectral and spectral basis descriptors were considered, providing an effective combination for accurately describing the spectral and timbral audio characteristics. The audio files were split using 70% of the available data for training and the remaining 30% for testing. A classifier was developed based on non-negative matrix factorization (NMF) techniques, thus introducing a novel application of NMF. The standard NMF method was examined, as well as its modifications: the local, the sparse, and the discriminant NMF. Experimental results are presented to compare MPEG-7 spectral basis representations with MPEG-7 basic spectral features alongside the various NMF algorithms. The results indicate that the use of the spectrum projection coefficients for feature extraction and the standard NMF classifier yields an accuracy exceeding 95%.**

## I. Introduction

The need for analysis of musical content arises in different contexts. It has many practical applications, mainly for effectively organizing and annotating data in multimedia databases, automatic music transcription, and music retrieval. Automatic musical instrument classification is the first step in developing the aforementioned systems. However, despite the massive research which has been carried out on a similar field, namely the automatic speech recognition, limited work has been done on musical content identification systems.

The experiments carried out so far are separated into two categories: classification of isolated instrument tones and classification of sound segments. Using isolated tones, Martin and Kim [11] developed a $k$-NN classifier using 31 features on a database consisting of 15 orchestral instruments. Their study included a hierarchical procedure classifying instrument families as well as a non-hierarchical approach, achieving a 87% classification success rate at the family level and a 61% rate at the instrument level. Eronen [10] recognized individual instruments with 80% rate using samples of isolated tones covering 30 orchestral instruments, where 44 spectral and temporal features were calculated for creating Gaussian Mixture Models and building k-NN classifiers. However, since the classifier used only isolated tones, the system would have a limited use in a practical application.

Using sound segments, Brown reported correct identifications of 79-84% for four classes of instruments (oboe, sax, clarinet, and flute), using Bayes decision rules for classification [9]. Cepstral coefficients, constant-Q coefficients and autocorrelation coefficients were extracted fron the audio files of the database used in this paper, namely the MIS Database from UIOWA [1]. More recently, Synak et al [12] used MPEG-7 temporal descriptors and various spectral features for sound segments consisting of 18 instrument classes and developed 2 classifiers. The first classifier uses the $k$-NN algorithm, while the second one uses decision rules based on rough sets theory, and achieve at best a recognition rate of 68.4%.

In our work, the problem of automatically classifying musical instrument segments is addressed. Files derived from the UIOWA database [1] were used, forming 6 instrument classes. Two sets of features are proposed. The first set describes the audio timbral texture and the second one describes the spectral characteristics as defined by the MPEG-7 audio standard [2]. For the classification procedure we used non-negative matrix factorization (NMF) [4], a subspace method for basis decomposition. NMF has been mainly used in face recognition and text categorization, and in this work a novel application for the method is demonstrated. Several proposed modifications of NMF were applied, providing a comparative study of the algorithms' efficiency. Furthermore, a comparison has been performed regarding the classification accuracy of the MPEG-7 AudioSpectrumProjection (ASP) coefficients versus MPEG-7 AudioSpectrum descriptors. The results indicate that using the ASP descriptor with timbral features in the standard NMF classification algorithm yields a correct classification rate of 95.06%, which is comparable to the performance of supervised classifiers for the same experiment [13].

The remainder of this paper is organized as follows. The audio features used are discussed in detail in Section II. Section III describes the subspace method of non-negative matrix factorization and its numerous extensions. Section IV describes the classification methodology used alongside the experiments performed for its evaluation, and Section V presents conclusions and future directions.

## II. Feature Extraction

In an audio classification system a careful selection of features that are able to accurately describe the temporal and spectral sound structures is vital. In our approach, a combination of features originating from general audio data classification and the MPEG-7 Audio framework is used.

### *A. Timbral texture features*

The following features are proposed in systems concerning general audio data (GAD) classification and speech recognition, and can be considered as a short term description of the textural shape of the audio segments:

1) *Zero-Crossing Rate:* It provides a noise measure for the given signal:

$$ZCR_t = \frac{1}{N-1}\sum_{n=0}^{N-1} |sign(x_t[n]) - sign(x_t[n-1])| \tag{1}$$

where the $sign$ function is 1 for positive arguments, -1 for negative arguments, $x_t[n]$ is signal for the $t$-th frame and $N$ the number of samples in an audio frame.

2) *Delta Spectrum:* It is defined as the average variation value of the spectrum between two adjacent frames and measures the amount of local spectral flux:

$$DS_t = \frac{1}{K-1}\sum_{k=0}^{K-1} [\log(X(t,k)+\delta) - \log(X(t-1,k)+\delta)]^2 \tag{2}$$

Where $X(t,k)$ is the $k$-th frequency sample of the Discrete Fourier Transform (DFT) of the $t$-th frame, $\delta$ a very small value and $K$ the resolution of the DFT.

3) *Spectral Rolloff:* It measures the spectral shape and is defined as the frequency below which a percentage of the magnitude distribution is concentrated:

$$SRF_t = \arg\max_{h=0}^{K-1}[\sum_{k=0}^{h} X(t,k) < TH \cdot \sum_{k=0}^{K-1} X(t,k)] \tag{3}$$

where $TH$ is the percentage threshold usually set to 0.85.

### *B. MPEG-7 features*

The MPEG-7 standard, formally known as "Multimedia Content Description Interface", standardizes the description of multimedia content. The standard is divided into 8 parts, where part 4 focuses on audio description tools [2]. The Low Level Descriptors (LLD) interface, as defined in the MPEG-7 audio description framework, includes 17 descriptors, divided into 6 categories. The MPEG-7 LLDs that were used for feature extraction are:

1) *AudioSpectrumCentroid:* It describes the center of gravity of the log-frequency power spectrum, indicating whether the signal spectrum is dominated by high or low frequencies:

$$ASC_t = \frac{\sum_{k=0}^{K/2} \log_2(f(t,k)/1000)P(t,k)}{\sum_{k=0}^{K/2} P(t,k)} \tag{4}$$

where $P(t,k)$ are the modified power spectrum coefficients (coefficients below 62.5 Hz are replaced by a single coefficient with power equal to their sum) and $f(t,k)$ are their corresponding frequencies.

2) *AudioSpectrumSpread:* It describes the second moment of the log-frequency power spectrum, indicating whether it is concentrated in the vicinity of the centroid or is spread over the spectrum:

$$ASS_t = \sqrt{\frac{\sum_{k=0}^{K/2}[\log_2(f(t,k)/1000) - ASC_t]^2 P(t,k)}{\sum_{k=0}^{K/2} P(t,k)}} \tag{5}$$

3) *AudioSpectrumFlatness:* It describes the flatness properties of the short-term spectrum for a number of frequency bands, indicating the presence or absence of tonal components:

$$ASF_{t,b} = \frac{\sqrt[ih(b)-il(b)]{\prod_{k=il(b)}^{ih(b)} P(k,t)}}{\frac{1}{ih(b)-il(b)+1}\sum_{k=il(b)}^{ih(b)} P(k,t)} \tag{6}$$

where $il(b)$ and $ih(b)$ are the power spectrum coefficient indices of the lower and higher edge of band $b$, respectively.

4) *AudioSpectrumProjection:* It is the compliment to the AudioSpectrumBasis descriptor and represents low-dimensional features of a spectrum after projection onto a reduced rank basis. The coefficients of both descriptors are extracted from the normalized AudioSpectrumEnvelope coefficients, using singular value decomposition (SVD). Optionally, it is possible to produce statistically independent basis by using independent component analysis (ICA) after the SVD [3].

## III. Non-negative matrix factorization

Subspace analysis is one of the popular multivariate data analysis methods, where low dimensional structures of patterns are revealed in high dimensional spaces. Non-negative matrix factorization (NMF) has been proposed as a novel subspace method in order to obtain a parts-based representation of objects by imposing non-negative constraints [4]. The problem addressed by NMF is as follows: Given a non-negative $n \times m$ matrix $\mathbf{V}$ (data matrix, consisting of $m$ vectors of dimension $n$), it is possible to find non-negative matrix factors $\mathbf{W}$ and $\mathbf{H}$ in order to approximate the original matrix:

$$\mathbf{V} \approx \mathbf{WH} \tag{7}$$

where the $n \times r$ matrix $\mathbf{W}$ contains the basis vectors and the $r \times m$ matrix $\mathbf{H}$ contains the weights needed to properly approximate the corresponding column of matrix $\mathbf{V}$, as a linear combination of the columns of $\mathbf{W}$. Usually, $r$ is chosen so that $(n+m)r < nm$, thus resulting in a compressed version of the original data matrix.

To find an approximate factorization in (7), a suitable objective function has to be defined. The generalized Kullback-Leibler (KL) divergence between $\mathbf{V}$ and $\mathbf{WH}$ is the most frequently used. Various NMF algorithms, differing mainly in the constraints included in their objective function are presented below.

### A. Standard NMF

The standard NMF enforces the non-negativity constraints on matrices **W** and **H**, thus a data vector can be approximated by an additive combination of the basis vectors. The proposed cost function is the generalized KL divergence:

$$D(\mathbf{V}||\mathbf{WH}) = \sum_{i=1}^{n}\sum_{j=1}^{m}[v_{ij}\log\frac{v_{ij}}{y_{ij}} - v_{ij} + y_{ij}] \qquad (8)$$

where $\mathbf{WH} = \mathbf{Y} = [y_{ij}]$. $D(\mathbf{V}||\mathbf{WH})$ reduces to KL divergence when $\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} = 1$. The standard NMF optimization problem is defined as:

$$\min_{\mathbf{W},\mathbf{H}} \; D(\mathbf{V}||\mathbf{WH}) \quad subj.to \;\; \mathbf{W,H} \geq 0, \sum_{i=1}^{n} w_{ij} = 1 \; \forall j \quad (9)$$

where $\mathbf{W,H} \geq 0$ means that all elements of matrices **W** and **H** are non-negative. The optimization problem (9) can be solved by using iterative multiplicative rules [4].

### B. Local NMF (LNMF)

Aiming to impose constraints concerning spatial locality and consequently revealing local features in the data matrix **V**, LNMF incorporates 3 additional constraints into the standard NMF problem:

1) Minimize the number of basis components representing **V**.
2) Different bases should be as orthogonal as possible.
3) Only retain components giving most important information.

The above constraints are expressed in the following constrained divergence as the LNMF cost function:

$$\begin{aligned} D(\mathbf{V}||\mathbf{WH}) &= \sum_{i=1}^{n}\sum_{j=1}^{m}[v_{ij}\log\frac{v_{ij}}{y_{ij}} - v_{ij} + y_{ij}] \\ &+ \alpha\sum_{i=1}^{r}\sum_{j=1}^{r} u_{ij} - \beta\sum_{i=1}^{r}\sum_{j=1}^{r} q_{ii} \qquad (10) \end{aligned}$$

where $\alpha, \beta$ are constants, $\mathbf{W}^T\mathbf{W} = \mathbf{U} = [u_{ij}]$ and $\mathbf{HH}^T = \mathbf{Q} = [q_{ij}]$. The minimization is similar to (9) and a local solution can be found by using 3 update rules [5].

### C. Sparse NMF (SNMF)

Inspired by NMF and sparse coding, the aim of SNMF is to impose constraints that can reveal local sparse features in the data matrix **V**. The following cost function is optimized for SNMF:

$$D(\mathbf{V}||\mathbf{WH}) = \sum_{i=1}^{n}\sum_{j=1}^{m}[v_{ij}\log\frac{v_{ij}}{y_{ij}} - v_{ij} + y_{ij}] + \lambda\sum_{j=1}^{m}||h_j||_l \qquad (11)$$

where $\lambda$ is a positive constant and $||\mathbf{h}_j||_l$ the $l$-norm of the $j$-th column of **H**. The SNMF factorization is defined as in (9), including also that $\forall i ||\mathbf{w}_i||_l = 1$. In SNMF, the sparseness is measured by a linear activation penalty, the minimum $l$-norm of the column of **H**. A local solution to the minimization problem (11) can be found by the update rules in [6].

### D. Discriminant NMF (DNMF)

DNMF keeps the original constraints of the NMF algorithm, enhances the locality of basis vectors imposed in the LNMF algorithm and attempts to improve the classification accuracy by incorporating into the aforementioned constraints information about class discrimination. Two more constraints are introduced:

1) Minimize the within-class scatter matrix $\mathbf{S}_w$.
2) Maximize the between-class scatter matrix $\mathbf{S}_b$.

The modified cost function is expressed as:

$$\begin{aligned} D(\mathbf{V}||\mathbf{WH}) &= \sum_{i=1}^{n}\sum_{j=1}^{m}[v_{ij}\log\frac{v_{ij}}{y_{ij}} - v_{ij} + y_{ij}] \\ &+ \alpha\sum_{i=1}^{r}\sum_{j=1}^{r} u_{ij} - \beta\sum_{i=1}^{r}\sum_{j=1}^{r} q_{ii} \\ &+ \gamma\mathbf{S}_w - \delta\mathbf{S}_b \qquad (12) \end{aligned}$$

where $\gamma$ and $\delta$ are constants. Information on the form of the class scatter matrices and the update rules that find a local solution to the minimization of (12) can be found in [7].

## IV. Experimental procedure and results

### A. Dataset

We used audio files taken from the MIS database developed by the university of Iowa [1]. Overall 300 audio files were used, consisting of 6 different instrument classes: piano, violin, cello, flute, bassoon and soprano saxophone. In detail, 58 files contain piano recordings, 101 violin, 52 cello, 31 saxophone, 29 flute and 29 bassoon. The 300 sounds are partitioned into a training set of 210 sounds and a test set of 90 sounds, which is a typical partition for classification experiments. All recordings are discretized at 44.1 kHz sampling rate and have a duration of about 20 sec.

### B. Classification method

Musical instrument classification in the NMF subspace is performed as follows. Using data from the training set, the data matrix **V** is created (each column $\mathbf{v}_j$ contains a feature vector computed from an audio file). The training procedure is performed by applying an NMF algorithm into the data matrix, yielding the basis matrix **W** and the encoding matrix **H**.

In the test phase, for each test audio file (represented by a feature vector $\mathbf{v}_{test}$) a new test encoding vector is formed as:

$$\mathbf{h}_{test} = \mathbf{W}^{\dagger}\mathbf{v}_{test} \qquad (13)$$

where $\mathbf{W}^{\dagger}$ is defined as the Moore-Penrose generalized inverse matrix of **W**. Having formed during training 6 classes of encoding vectors $\mathbf{h}_l$ (where $l = 1, \ldots, 6$), a nearest neighbor classifier is employed to classify the new test sample by using the Cosine Similarity Measure (CSM). The class label $l'$ of the test file is defined as:

$$l' = \arg\max_{l=1,\ldots,6}\{\frac{\mathbf{h}_{test}^T\mathbf{h}_l}{||\mathbf{h}_{test}||||\mathbf{h}_l||}\} \qquad (14)$$

thus trying to maximize the cosine of the angle between $\mathbf{h}_{test}$ and $\mathbf{h}_l$.

TABLE I

CONFUSION MATRIX FOR STANDARD NMF, FEATURE MATRIX 2

| **Instr.** | Piano | Bassoon | Cello | Flute | Sax | Violin |
|---|---|---|---|---|---|---|
| Piano | **18** | 0 | 0 | 0 | 0 | 0 |
| Bassoon | 1 | **8** | 0 | 0 | 0 | 0 |
| Cello | 0 | 0 | **16** | 0 | 0 | 0 |
| Flute | 2 | 1 | 0 | **6** | 0 | 0 |
| Sax | 0 | 0 | 0 | 0 | **9** | 0 |
| Violin | 0 | 0 | 0 | 0 | 0 | **29** |

### C. *Performance Evaluation*

Two separate experiments on the various NMF algorithms have been performed by using different extracted features, in order to compare the efficiency between the MPEG-7 descriptors. The first feature vector contains the MPEG-7 statistical spectrum descriptors (ASC, ASS, and ASF) and the timbral texture features described in Section II-A. The second feature vector contains the MPEG-7 ASP descriptor and the various timbral texture features. Consequently, the efficiency of the ASP descriptor is compared with the efficiency of features more commonly used in classification experiments.

The mean classification accuracy and its standard deviation for the four NMF algorithms for both feature vectors is presented in Figure 1. The highest accuracy achieved by the standard NMF algorithm is 95.06% when ASP descriptors are used. The achieved performance is comparable to the performance of supervised GMM and HMM classifiers for the same data set, where the achieved performance was 99% and 97%, respectively [13]. However, the accuracy of NMF is deteriorated when the first feature vector is used. The LNMF is clearly outperformed by all algorithms. This may be attributed to the locality constraints the LNMF imposes when applied to holistic descriptors. The SNMF overall displays satisfactory results, but its efficiency depends on the selection of parameter $\lambda$ as can be seen in (11). Finally, DNMF outperforms both LNMF and SNMF when the first feature vector is used, but its accuracy drops to 80.5% when the ASP descriptor is used, mainly because the algorithm's accuracy depends on the values of $\gamma$ and $\delta$ as can be seen in (12).

More detailed information about the performance of the NMF algorithm using the ASP descriptor is shown in Table I in the form of a confusion matrix, where the columns correspond to the predicted musical instrument and the rows to the actual instrument. Most misclassifications occur for the flute, where flute samples are wrongly classified as piano and bassoon. In addition, there is a single miss-classification for the bassoon. It should be noted that the flute samples that were wrongly classified for piano displayed similar dynamical and spectral shape with several piano samples.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a new method of classifying musical instrument recordings by using NMF algorithms, using a variety of features, mainly the MPEG-7 Audio descriptors. The results indicate that the standard NMF algorithm can perform classification with high accuracy even compared to its modifications, which are more suitable when used in conjunction with parts-based descriptors due to their numerous constraints. In addition, it is shown that the MPEG-7 ASP descriptor yields a more discriminating representation compared to most of the traditional spectrum descriptors.

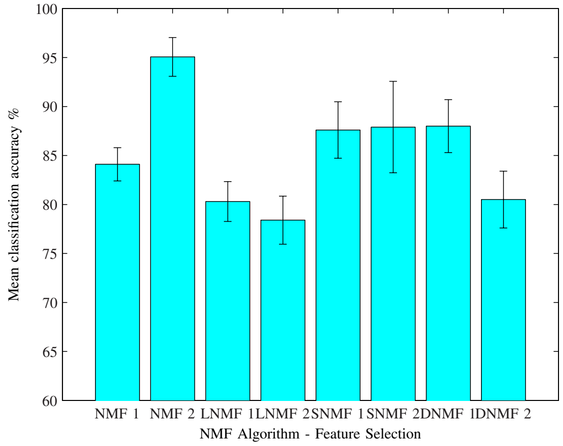

Fig. 1. Classification accuracy of NMF algorithms, where 1/2 refers to the feature vector used.

In the future, the NMF techniques can be applied to discriminate the whole spectrum of orchestral instruments. A supervised NMF classification scheme could be developed, considering information about class discrimination. Finally, for musical instrument classification, features describing the timbral shape could be employed, such as the timbral temporal and spectral descriptors proposed by the MPEG-7 standard.

## REFERENCES

[1] University of Iowa Musical Instrument Sample Database, http://theremin.music.uiowa.edu/index.html.

[2] MPEG-7 overview (version 9), *ISO/IEC JTC1/SC29/WG11 N5525*, March 2003.

[3] H. G. Kim, N. Moreau, and T. Sikora, "Audio classification based on MPEG-7 spectral basis representations," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 716-725, May 2004.

[4] D. D. Lee and H. S. Seung, "Algoritnms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556-562, 2001.

[5] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in Proc. *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-6, 2001.

[6] C. Hu, B. Zhang, S. Yan, Q. Yang, J. Yan, Z. Chen, and W. Ma, "Mining ratio rules via principal sparse non-negative matrix factorization," in Proc. *IEEE Int. Conf. Data Mining*, 2004.

[7] I. Buciu and I. Pitas, "A new sparse image representation algorithm applied to facial expression recognition," in Proc. *17th Int. Conf. Pattern Recognition*, August 2004.

[8] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 5, pp. 293-302, July 2002.

[9] J. C. Brown, O. Houix, and S. McAdams, "Feature dependence in the automatic identification of musical woodwind instruments," *J. Acoustical Society of America*, vol. 109, no. 3, pp. 1064-1072, March 2001.

[10] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in Proc. *2000 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 753-756, June 2000.

[11] K. D. Martin and Y. E. Kim, "Musical instrument identification: A pattern-recognition approach," in Proc. *136th Meeting Acoustical Society of America*, October 1998.

[12] A. Wieczorkowska, J. Wroblewski, P. Synak, and D. Slezak, "Application of temporal descriptors to musical instrument sound recognition," *J. Intelligent Information Systems*, vol. 21, no. 1, pp. 71-93, July 2003.

[13] 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," *2nd Workshop On Immersive Communication And Broadcast Systems*, October 2005.