A TENSOR-BASED APPROACH FOR AUTOMATIC MUSIC GENRE CLASSIFICATION

Emmanouil Benetos and Constantine Kotropoulos

Department of Informatics, Aristotle Univ. of Thessaloniki
Box 451, Thessaloniki 541 24, Greece
E-mail: {empeneto,costas}@aiia.csd.auth.gr

ABSTRACT

Most music genre classification techniques employ pattern recognition algorithms to classify feature vectors extracted from recordings into genres. An automatic music genre classification system using tensor representations is proposed, where each recording is represented by a feature matrix over time. Thus, a feature tensor is created by concatenating the feature matrices associated to the recordings. A novel algorithm for non-negative tensor factorization (NTF), which employs the Frobenius norm between an n-dimensional raw feature tensor and its decomposition into a sum of elementary rank-1 tensors, is developed. Moreover, a supervised NTF classifier is proposed. A variety of sound description features are extracted from recordings from the GTZAN dataset, covering 10 genre classes. NTF classifier performance is compared against multilayer perceptrons, support vector machines, and non-negative matrix factorization classifiers. On average, genre classification accuracy equal to 75% with a standard deviation of 1% is achieved. It is demonstrated that NTF classifiers outperform matrix-based ones.

1. INTRODUCTION

In the research community, automatic genre classification applications attempt to classify recordings into distinguishable genres by extracting relevant features and employing pattern recognition algorithms [1]. A large amount of literature on automatic music genre classification exists. Several benchmark audio collections have been used for experiments, making the various genre classification approaches comparable, as can be seen in Table 1. Common classifiers employed in music genre applications are Gaussian mixture models (GMMs), support vector machines (SVMs), and linear discriminant analysis (LDA). Closely related topics to music genre classification are mood and style recognition as well as artist identification [2].

In this paper, the problem of automatic genre classification is addressed by employing multilinear techniques. In the field of multilinear algebra, tensors are considered as the multidimensional equivalent of matrices or vectors [3]. In our approach, each recording is represented by a matrix containing features over time, thus creating a more detailed and natural representation of signal characteristics. Consequently, all recordings are represented by a feature tensor. Such a representation allows capturing the rich time-varying nature of short-term features and preserves their integrity without unnecessarily mixing spectral and temporal features in the same vector. A novel algorithm for analyzing and classifying multidimensional data is proposed, originating from non-negative matrix factorization (NMF). The algorithm is called non-negative tensor factorization (NTF) and is able to decompose a tensor in a sum of elementary rank-1 tensors.

Table 1: Notable results on genre classification approaches.

Reference	Dataset	Classifier	Best Accuracy
Tzanetakis et al. [8]	GTZAN	GMM	61.0%
Li et al. [9]	GTZAN	SVM - LDA	78.5%
Lidy et al. [1]	GTZAN	SVM	74.9%
Lidy et al. [1]	MIREX 2004	SVM	70.4%
Pampalk et al. [10]	MIREX 2004	NN - GMM	82.3%
Bergstra et al. [11]	MIREX 2005	Decision stumps	82.34%

The algorithm employs as a distance measure the Frobenius norm, which belongs to general category of Bregman divergences [6]. Bregman divergences have been previously used to solve the non-negative matrix approximation problem [7]. In addition, a supervised classifier for NTF is proposed, employing basis orthogonalization. Experiments are performed on the GTZAN database, which contains 1000 audio recordings covering 10 music genre classes. Several standard state-of-the-art classifiers are also tested on the same database, including multilayer perceptrons, support vector machines, and classifiers based on NMF, such as the standard NMF, the local NMF, and the sparse NMF. Genre classification accuracy results show 75% genre classification accuracy for the NTF classifier using the Frobenius norm. In general, the superiority of the NTF classifier against the just mentioned classifiers is demonstrated.

The outline of the paper is as follows. Section 2 describes the NMF method and its variants. Section 3 is devoted to the proposed NTF method and the previous approaches proposed for NTF. The proposed algorithm is presented, along with the proposed NTF classifier. Section 4 describes the data set used and the employed feature set. The standard state-of-the-art classifiers are discussed and the experimental results are presented in Section 4 as well. Conclusions are drawn and future directions are indicated in Section 5.

2. NON-NEGATIVE MATRIX FACTORIZATION

NMF is a subspace method able to obtain a parts-based representation of objects by imposing non-negative constraints [12]. The problem addressed by NMF is as follows. Given a non-negative $n \times m$ data matrix $\mathbf{V}$, find the non-negative matrix factors $\mathbf{W}$ and $\mathbf{H}$ in order to approximate the original matrix as:

$$\mathbf{V} \approx \mathbf{W}\mathbf{H}. \tag{1}$$

The $n \times r$ matrix $\mathbf{W}$ contains the basis vectors and the columns of the $r \times m$ matrix $\mathbf{H}$ contain the weights needed to properly approximate the corresponding column of matrix $\mathbf{V}$ as a linear combination of the columns of $\mathbf{W}$.

To find an approximate factorization in (1), a suitable objective function has to be defined. In [12], the generalized Kullback-Leibler (KL) divergence between $\mathbf{V}$ and $\mathbf{WH}$ was used. The minimization of the objective function can be solved by using iterative multiplicative rules [12]. The local NMF (LNMF) algorithm aims to impose spatial locality

E. Benetos was a scholar of the "Alexander S. Onassis" Public Benefit Foundation.

in the solution and consequently to reveal local features in the data matrix $\mathbf{V}$ [13]. The sparse NMF (SNMF) method is inspired by sparse coding, aiming to impose constraints that can reveal local sparse features in $\mathbf{V}$ [14]. It should be noted that the performance of the SNMF algorithm depends upon the choice of sparseness parameter. In [7], a more general view of the NMF is provided under the so-called non-negative matrix approximation (NNMA). Instead of minimizing a single objective function, Sra et al. propose the minimization of Bregman divergences (cf. Section 3.2).

3. NON-NEGATIVE TENSOR FACTORIZATION

3.1 Tensor Definition

In the field of multilinear algebra, tensors are considered as multidimensional generalizations of matrices and vectors [3]. A higher-order real-valued tensor of n dimensions is defined over the vector space $\mathbb{R}^{I_1 \times \cdots I_n}$, $I_i \in \mathbb{Z}$, $i = 1, \dots, n$ and is represented by $\mathcal{A}$. Each element of tensor $\mathcal{A}$ is addressed by n indices, $\mathcal{A}_{i_1 i_2 \dots i_n}$. Basic operations can be defined on tensors. The symbol $\times_n$ stands for the n-mode product between a tensor and a matrix [3]. For example, the n-mode product between a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and a matrix $\mathbf{B} \in \mathbb{R}^{I_2 \times J}$ is represented by a tensor $(\mathcal{A} \times_2 \mathbf{B}) \in \mathbb{R}^{I_1 \times J \times I_3}$.

3.2 Bregman Divergences

Bregman divergences were proposed by Bregman in 1967 [6]. They are defined as:

$$D_\phi(x, y) = \phi(x) - \phi(y) - \phi'(y)(x - y), \tag{2}$$

where $\phi()$ is a strictly convex function defined on a convex set $S \subseteq \mathbb{R}$ and $\phi'()$ denotes the first-order derivative of $\phi()$. Bregman divergences are non-negative [7] and their definition can be extended to tensors. For $\phi(x) = \frac{1}{2}x^2$, $D_\phi(x, y)$ corresponds to the Frobenius norm. For $\phi(x) = x \log(x)$, the corresponding Bregman divergence becomes the KL divergence, while for $\phi(x) = -\log(x)$, the resulting $D_\phi(x, y)$ is the Itakura-Saito (IS) distance. Although in this paper we deal with the Frobenius norm, the proposed algorithm in Section 3.4 can handle all the aforementioned Bregman divergences.

3.3 Previous Approaches

In 2005, Shashua and Hazan proposed a generalization of the NMF algorithm for N-dimensional tensors [15] building on the previous work of Welling and Weber [16] and provided a proof of convergence. The problem was formulated as a decomposition of a tensor into a sum of k rank-1 tensors, when the Frobenius norm was used as a distance measure. Heiler and Schnörr proposed a generalization of the SNMF algorithm for 3-dimensional tensors [17]. The Frobenius norm was used as a distance measure and the algorithm was termed as sparsity-constrained NTF. Shifted NTF algorithms, which extend the shifted NMF ones, were proposed for multichannel sound separation of harmonic instruments in [18]. In 2007, Cichocki proposed algorithms for 3-dimensional NTF using alpha and beta divergences [19]. It should be noted that this model cannot be generalized to higher-dimensional tensors or degenerate to the NMF model in the two-dimensional case.

3.4 Proposed NTF Algorithm

We aim at creating a generalized algorithm for n-dimensional tensors, which can degenerate to the NMF algorithm for $n = 2$. Our work is inspired by Shashua and Hazan's work [15], which can be applied to n-dimensional tensors and degenerates to NMF when $n = 2$. Therefore, the goal of NTF is to decompose a tensor $\mathcal{V} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ into a sum of k rank-1 tensors:

$$\mathcal{V} = \sum_{j=1}^{k} \mathbf{u}_1^j \otimes \mathbf{u}_2^j \otimes \cdots \otimes \mathbf{u}_n^j \tag{3}$$

where $\otimes$ stands for the Kronecker product and $\mathbf{u}_i^j \in \mathbb{R}_+^{I_i}$. The following minimization problem is considered using Bregman divergences:

$$\min_{\mathbf{u}_i^j \geq 0} D_\phi\left(\sum_{j=1}^{k} \mathbf{u}_1^j \otimes \mathbf{u}_2^j \otimes \cdots \otimes \mathbf{u}_n^j, \mathcal{V}\right). \tag{4}$$

The minimization of (4) can be achieved using auxiliary functions. For the Frobenius norm, the following multiplicative update rule is obtained for $\mathbf{u}_i^j$:

$$u_{i(l)}^j \leftarrow \tilde{u}_{i(l)}^j \cdot \frac{\sum_{i_1 \dots i_{i-1} i_{i+1} \dots i_n}^{I_1 \cdots I_{i-1} I_{i+1} \cdots I_n} A \cdot \mathcal{V}_{i_1 \dots l \dots i_n}}{\sum_{i_1 \dots i_{i-1} i_{i+1} \dots i_n}^{I_1 \cdots I_{i-1} I_{i+1} \cdots I_n} A \cdot B}. \tag{5}$$

where $u_{i(l)}^j$ is the l-th element of vector $\mathbf{u}_i^j$ for $j = 1, \dots, k$, $i = 1, \dots, n$, $l = 1, \dots, I_i$, $\tilde{u}_{i(l)}^j$ is the l-th element of vector $\mathbf{u}_i^j$ before updating, $A = u_{1(i_1)}^j \dots u_{i-1(i_{i-1})}^j u_{i+1(i_{i+1})}^j \dots u_{n(i_n)}^j$, and $B = \sum_{m=1}^{k} u_{1(i_1)}^m \dots \tilde{u}_{i(l)}^m \dots u_{n(i_n)}^m$. The indices on A and B are suppressed for notation simplicity. The proof of (5) is omitted.

In order to apply the aforementioned NTF algorithms for an n-dimensional tensor $\mathcal{V}$, n matrices $\mathbf{U}_i$, $i = 1, \dots, n$, should be created with dimensions $I_i \times k$. Matrices $\mathbf{U}_i$ are initialized using random numbers between 0 and 1. The update rule (5) is applied to each matrix $\mathbf{U}_i$, for $j = 1, \dots, k$ and $l = 1, \dots, I_i$.

Although MATLAB implementations of NTF are available [20, 21] we preferred to implement our NTF from first principles in order to fully control the procedure. More advanced optimization techniques, such as the projected gradient method or the fixed point alternating least squares [21], did not offer a higher classification accuracy.

3.5 Proposed 3D-NTF Classifier

The novel NTF classifier for 3-dimensional tensors extends the NMF classifier proposed in [22], where each class was trained separately and the test data are projected onto an orthogonalized basis. The proposed 3D NTF classifier considers a tensor $\mathcal{V} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1 is the data dimension, I_2, I_3 are the feature dimensions, and C is the number of classes. Let $\circledcirc$ stand for the Khatri-Rao product (column-wise tensor product). The classifier is defined as follows:

1. Perform training on each class separately:

$$\mathcal{V}_\xi = \sum_{j=1}^{k} \mathbf{u}_1^j \otimes \mathbf{u}_2^j \otimes \mathbf{u}_3^j = (\mathbf{U}_{2(\xi)} \circledcirc \mathbf{U}_{3(\xi)}) \times_3 \mathbf{U}_{1(\xi)}, \tag{6}$$

   where $\xi = 1, \dots, C$ and $\mathbf{U}_{1(\xi)}$ is a $I_{1(\xi)} \times k$ matrix, with $I_{1(\xi)}$ being the number of training data for class ξ. Matrix $\mathbf{U}_{2(\xi)}$ has dimensions $I_2 \times k$ and matrix $\mathbf{U}_{3(\xi)}$ has dimensions $I_3 \times k$. Matrices $\mathbf{U}_{1(\xi)}$, $\mathbf{U}_{2(\xi)}$, and $\mathbf{U}_{3(\xi)}$ are created by concatenating the respective vectors $\mathbf{u}_1^j$, $\mathbf{u}_2^j$, and $\mathbf{u}_3^j$. Thus, $(\mathbf{U}_{2(\xi)} \circledcirc \mathbf{U}_{3(\xi)})$ is a $I_2 \times I_3 \times k$ tensor.

2. Convert (6) into matrix unfoldings [3]:

$$\mathbf{V}_\xi = (\mathbf{U}_{2(\xi)} \odot \mathbf{U}_{3(\xi)}) \cdot \mathbf{U}_{1(\xi)}^T. \tag{7}$$

   Where $\odot$ stands for the column-wise Kronecker product. Thus, $(\mathbf{U}_{2(\xi)} \odot \mathbf{U}_{3(\xi)})$ has dimensions $I_2 I_3 \times k$, while matrix $\mathbf{V}_\xi$ has dimensions $I_2 I_3 \times I_{1(\xi)}$.

3. Perform QR decomposition on basis matrix $(\mathbf{U}_{2(\xi)} \odot \mathbf{U}_{3(\xi)})$:
$$(\mathbf{U}_{2(\xi)} \odot \mathbf{U}_{3(\xi)}) = \mathbf{Q}_\xi \cdot \mathbf{R}_\xi \qquad (8)$$
where $\mathbf{Q}_\xi$ is a $I_2 I_3 \times k$ column-orthogonal matrix (i.e. $\mathbf{Q}_\xi^T \mathbf{Q}_\xi$ is the $k \times k$ identity matrix)[1] and $\mathbf{R}_\xi$ a $k \times k$ upper triangular matrix. Store matrices $\mathbf{Q}_\xi$ and $\mathbf{H}_\xi$, where $\mathbf{H}_\xi = \mathbf{R}_\xi \cdot \mathbf{U}_{1(\xi)}^T$. It is worth noting that the Gram-Schmidt orthogonalization does not affect the non-negativity of the basis matrix. It is used to calculate **correctly** the L_2 norms in a non-orthogonal basis.
4. For testing, the feature matrix of dimensions $I_2 \times I_3$ $\mathbf{V}_{test}$ is considered. The feature matrix is projected onto basis matrices of the several classes:
$$\mathbf{h}_{test}^\xi = \mathbf{Q}_\xi^T \cdot \mathbf{V}_{test}. \qquad (9)$$
Vector $\mathbf{h}_{test}^\xi$ has length k.
5. For each class, vector $\mathbf{h}_{test}^\xi$ is compared with each column of matrix $\mathbf{H}_\xi$, using the cosine similarity measure (CSM). The vector $\mathbf{h}_j^{(\xi)}$ that maximizes the CSM for class ξ is calculated as a measure of similarity for the class:
$$CSM_\xi = \max_{j=1,2,\ldots,k} \left\{ \frac{\mathbf{h}_{test}^{\xi\ T} \mathbf{h}_j^{(\xi)}}{\|\mathbf{h}_{test}^\xi\| \|\mathbf{h}_j^{(\xi)}\|} \right\}, \qquad (10)$$
where $\mathbf{h}_j^{(\xi)}$ is the j-th column of matrix $\mathbf{H}_\xi$. Finally, the class label of the test matrix $\mathbf{V}_{test}$ is determined by the maximum among CSM_ξ, i.e.:
$$\vartheta = \arg \max_{\xi=1,2,\ldots,C} \{CSM_\xi\}. \qquad (11)$$

4. EXPERIMENTAL PROCEDURE

The GTZAN database was employed for genre classification experiments, containing 1000 audio recordings covering 10 music genres [8]. The following genres represented in the database are: Classical, Country, Disco, HipHop, Jazz, Rock, Blues, Reggae, Pop, and Metal. Each genre is represented by 100 recordings. All recordings are mono channel, are sampled at 22.050 Hz sampling rate, and have a duration of approximately 30 sec. Each recording is separated into 30 segments, therefore each segment has a duration of 1 sec.

In this paper, a combination of features originating from GAD classification and the MPEG-7 audio framework is explored. The complete list of extracted features is summarized in Table 2. It should be noted that for each audio frame of 10 msec duration, 24 Mel-frequency cepstral coefficients and 8 specific loudness sensation [10] coefficients are used. Except features 10-12, the 1st and 2nd moments of features computed on a frame basis, as well as their derivatives, are computed. This explains the factor 4 in Table 2. In total, 187 features are extracted for each segment. All extracted features apart from the MFCCs are non-negative, therefore they can be employed for NTF classification. For the MFCCs, their magnitude is employed. Computing the aforementioned features over time is facilitated within the NTF context. Finally, a data tensor $\mathcal{V}$ was created with dimensions 1000 × 187 × 30.

Table 2: Feature set.

No.	Feature	# Values per segment
1	MPEG-7 AudioPower (AP)	$1 \times 4 = 4$
2	MPEG-7 AudioFundamentalFrequency (AFF)	$1 \times 4 = 4$
3	Total Loudness (TL)	$1 \times 4 = 4$
4	Specific Loudness Sensation (SONE)	$8 \times 4 = 32$
5	MPEG-7 AudioSpectrumCentroid (ASC)	$1 \times 4 = 4$
6	Spectrum Rolloff Frequency (SRF)	$1 \times 4 = 4$
7	MPEG-7 AudioSpectrumSpread (ASS)	$1 \times 4 = 4$
8	AudioSpectrumFlatness (ASF)	$4 \times 4 = 16$
9	Mel-frequency Cepstral Coefficients (MFCCs)	$24 \times 4 = 96$
10	Auto-Correlation Values (AC)	13
11	MPEG-7 Log Attack Time (LAT)	1
12	MPEG-7 Temporal Centroid (TC)	1
13	Zero-Crossing Rate (ZCR)	$1 \times 4 = 4$
	Total number of features	**187**

Table 3: Classification accuracy for 80 selected features.

Classifier	70%-30% Split	90%-10% Split
NMF	58.77%	62.00%
LNMF	64.11%	64.33%
SNMF 1 ($\lambda = 0.1$)	57.44%	64.66%
SNMF 2 ($\lambda = 0.001$)	57.55%	66.66%
MLP	**65.33**%	72.00%
SVM	64.00%	73.00%
NTF	64.66%	**75.00**%

In order to reduce the feature set cardinality, a suitable subset for classification has to be selected. The branch-and-bound search strategy is employed for complexity reduction, using the ratio of the inter-class dispersion over the intra-class dispersion as a performance measure [5]. The feature selection algorithm is employed using the matrix unfolding of the data tensor [3]. Thus, from tensor $\mathcal{V} \in \mathbb{R}^{1000 \times 187 \times 30}$ the unfolding $\mathbf{V}_{(2)} \in \mathbb{R}^{187 \times 1000 \cdot 30}$ was created. After experimentation, 80 features were selected out of the 187. Most of the selected features belong to the class of MFCCs.

The performance of the NTF classifier is compared against that of multilayer perceptrons (MLPs), SVMs, and NMF classifiers. In particular, a 3-layered perceptron with a logistic activation function is utilized. Learning is based on the back-propagation algorithm, with learning rate equal to 0.3, for 500 training epochs, and momentum equal to 0.2. A multi-class SVM classifier with a 2nd order polynomial kernel is also used. Finally, the NMF, LNMF, and SNMF classifiers were employed. Two instances of the SNMF classifier were used with sparseness parameter equal to 0.1 and 0.001, respectively [14]. Experiments were performed using the matrix unfolding $\mathbf{V}_{(2)}$.

Experiments were performed for the subset of 80 selected features. Two different training/test set splits were tested, namely 70%-30% and 90%-10%. It is worth noting that most genre classification experiments have been tested using 90%-10% splits. Concerning the NTF classifiers, the value of parameter k is set experimentally. More specifically, it is set to 65 when 70%-30% splits are used and 62 when the 90%-10% splits are used.

The classification accuracy for the subset of 80 selected features is given in Table 3. It can be seen that the highest classification accuracy of 75.0% is achieved by the NTF classifier when the 90%-10% split is employed. It should be noted that the standard deviation of the NTF classifier accuracy is found to be 1% after 10 fold cross-validation. The classification rate outperforms that reported by Tzanetakis in [8] (61.0%) and Lidy in [1] (74.9%), but is inferior to the rate achieved by Li [9] (78.5%). In general, the NTF classifier attains higher classification rates than those achieved by the SVM classifier. As far as the NMF classifiers are concerned, it is clear that they are inferior to the SVM, MLP, and NTF classifier. The highest rate reported by the NMF classifiers is 66.6% for the SNMF 2 classifier, which still outperforms the rate reported in [8]. Therefore, it can be deduced that the NTF classifier outperforms the corresponding NMF classifiers.

Next, the statistical significance of the differences in recognition rates between the NTF classifier and the SVM and MLP classifiers is addressed, employing the method de-

[1] Obviously, $\mathbf{Q}_\xi \mathbf{Q}_\xi^T$ is **not** the identity matrix.

Table 4: Confusion matrix for the NTF classifier, using the 80 selected features set and 90%-10% splits.

Genre	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	**10**	0	0	0	0	0	0	0	0	0
Classical	0	**8**	1	0	0	0	0	0	1	0
Country	0	0	**7**	0	0	2	1	0	0	0
Disco	1	0	0	**7**	0	1	0	1	0	0
Hiphop	0	0	0	3	**7**	0	0	0	0	0
Jazz	0	0	1	0	0	**9**	0	0	0	0
Metal	0	0	0	0	1	0	**9**	0	0	0
Pop	1	0	0	0	1	0	0	**6**	1	1
Reggae	0	0	1	0	2	1	0	0	**6**	0
Rock	0	0	0	2	0	0	0	1	1	**6**

scribed in [23]. It can be shown that the performance gains for the NTF classifier are not statistically significant against the SVM and MLP classifiers at 95% confidence level. On the contrary, the performance difference between the NTF classifier and the NMF classifiers is found to be statistically significant at 95% confidence level. It should be noted that the difference of 3.5% between the one-vs-the-rest SVMs employed by Li [9] and the NTF classifier is statistically insignificant as well.

Insight to the performance of the NTF classifier is offered by the confusion matrix in Table 4. The columns of the confusion matrix correspond to the predicted music genre and the rows to the actual one. For the Frobenius NTF classifier, most misclassifications occur among the Pop, Reggae, and Rock genres.

5. CONCLUSIONS - FUTURE WORK

In this paper, a novel algorithms for NTF has been developed, as well as an NTF classifier that employs basis orthogonalization has been proposed. The classification accuracy reported in this paper indicates that multilinear techniques when employed in classification can yield promising results compared to vector-based machine learning techniques. In the future, NTF experiments can be performed using the ISMIR 2004 and 2005 datasets. Moreover, additional features could be employed, exploring the rhythmic content of the recordings, for example the rhythm and periodicity histograms. In addition, NTF algorithms could be extended to deal with multi-genre classification. Finally, various initialization techniques could be tested for NTF algorithms to speed up their convergence and various Bregman divergences could be employed.

REFERENCES

[1] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification," in Proc. *6th Int. Conf. Music Information Retrieval*, pp. 34-41, September 2005.

[2] M. I. Mandel, G. E. Poliner, and D. P. W. Ellis, "Support vector machine active learning for music retrieval," *Multimedia Systems*, vol. 12, no. 1, pp. 3-13, 2006.

[3] L. De Lathauwer, "*Signal Processing Based on Multilinear Algebra*", Ph.D. Thesis, K.U. Leuven, E.E. Dept.-ESAT, Belgium, 1997.

[4] MPEG-7, "Information Technology-Multimedia Content Description Interface-Part 4: Audio," *ISO/IEC JTC1/SC29/WG11 N5525*, March 2003.

[5] F. van der Hedjen, R. P. W. Duin, D. de Ridder, and D. M. J. Tax, *Classification, Parameter Estimation and State Estimation*, London UK: Wiley, 2004.

[6] L. M. Bregman, "The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, Vol. 7, pp. 200-217, 1967.

[7] S. Sra and I. S. Dhillon, "Nonnegative matrix approximation: algorithms and applications," *Technical Report TR-06-27*, Computer Sciences, University of Texas at Austin, 2006.

[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] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in Proc. *26th Annual ACM Conf. Research and Development in Information Retrieval*, pp. 282-289, July-August 2003.

[10] E. Pampalk, A. Flexer, and G. Widmer, "Improvements of audio based music similarity and genre classification," in Proc. *6th Int. Symp. Music Information Retrieval*, pp. 628-633, 2005.

[11] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl, "Aggregate features and AdaBoost for music classification," *Machine Learning*, vol. 65, nos 2-3, pp. 473-484, 2006.

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

[13] 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.

[14] 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. *2004 IEEE Int. Conf. Data Mining*, pp. 407-410, November 2004.

[15] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in Proc. *22nd Int. Conf. Machine Learning*, pp. 792-799, August 2005.

[16] M. Welling and M. Weber, "Positive Tensor Factorization," *Pattern Recognition Letters*, vol. 22, pp. 1255-1261, 2001.

[17] M. Heiler and C. Schnörr, "Controlling sparseness in non-negative tensor factorization," in Proc. *9th European Conf. Computer Vision*, Vol. 1, pp. 56-67, May 2006.

[18] D. FitzGerald, M. Cranitch, and F. Coyle, "Sound source separation using shifted non-negative tensor factorization," in Proc. *2006 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. V, pp. 653-658, May 2006.

[19] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S. Amari, "Non-negative tensor factorization using alpha and beta divergences," in Proc. *2007 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, April 2007.

[20] T. Kolda and B. Bader, "Matlab tensor classes", SAND2004-589, `http://csmr.ca.sandia.gov/tgkolda`

[21] A. Cichocki, R. Zdunek, and S. Amari,"Nonnegative matrix and tensor factorization," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 142-145, January 2008.

[22] E. Benetos, M. Kotti, and C. Kotropoulos, "Large scale musical instrument identification," in Proc. *4th Sound and Music Computing Conference*, July 2007.

[23] I. Guyon, J. Makhoul, R. Schwartz, and V. Vapnik, "What size test set gives good error rate estimates?," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 52-64, January 1998.