Music Genre Classification via Joint Sparse Low-Rank Representation of Audio Features

Yannis Panagakis, *Member, IEEE*, Constantine L. Kotropoulos, *Senior Member, IEEE*, Gonzalo R. Arce *Fellow, IEEE*

***Abstract*—A novel framework for music genre classification, namely the *joint sparse low-rank representation* (JSLRR) is proposed in order to: 1) smooth the noise in the test samples, and 2) identify the subspaces that the test samples lie onto. An efficient algorithm is proposed for obtaining the JSLRR and a novel classifier is developed, which is referred to as the JSLRR-based classifier. Special cases of the JSLRR-based classifier are the joint sparse representation-based classifier and the low-rank representation-based one. The performance of the three aforementioned classifiers is compared against that of the sparse representation-based classifier, the nearest subspace classifier, the support vector machines, and the nearest neighbor classifier for music genre classification on 6 manually annotated benchmark datasets. The best classification results reported here are comparable with or slightly superior than those obtained by the state-of-the-art music genre classification methods.**

***Index Terms*—Music Genre Classification, Sparse Representation, Low-Rank Representation, ℓ_1 Norm Minimization, Nuclear Norm Minimization, Auditory Representations.**

I. Introduction

Music genre is probably the most popular semantic description of music content [1]. It is worth mentioning that, 68% of the tags that appeared in *last.fm* were related to music genre [2]. Increasing the accuracy of automatic music genre classification is a cornerstone toward the deployment of robust music information retrieval systems.

Despite the considerable volume of research conducted so far, that is surveyed in [3]–[5], music genre classification remains a difficult problem due to the fuzzy boundaries between the different genres, depending on cultural, artistic, or market factors [4]. In most systems, each music recording is represented by suitable features first, which frequently undergo a dimensionality reduction [3], [4], [6] prior to their classification into music genres by machine learning algorithms. A variety of features has been tested for music genre classification in the so-called bag-of-features (BOF) approaches [7]–[14]. Such features include timbral texture ones, rhythmic ones, pitch content, or their combinations. Furthermore, spectral, cepstral, and auditory modulation-based features have been recently employed either in BOF approaches or as autonomous music representations in order to capture both the timbral and the temporal structure of music [15]–[17]. Commonly used classifiers have been the support vector machines (SVM), the nearest-neighbor (NN), and the Gaussian Mixture Model-based classifiers [3].

In pattern analysis, an underlying tenet is that the data have some type of intrinsic structure, enabling their proper representation and efficient classification. A common choice is to assume that any given set of samples, originating from a specific class, lies onto a linear subspace. Therefore, multiple data classes can be modeled by a *union* of independent linear subspaces. Such an assumption is valid in many real-world cases [18]–[20]. Accordingly, a test sample is represented as a linear combination of the training samples stemming from the class it actually belongs to and the class label assigned to it is inferred by that of the training samples weighted by non-zero coefficients. Hence, one needs to derive the representation coefficients by solving an appropriate inverse problem. In this context, one way to obtain coefficients suitable for classification is to seek for the *sparsest representation* of the test sample with respect to a dictionary formed by the training samples. If the sample dimension is much smaller than the number of training samples, such a representation is computed efficiently by solving an underdetermined system of linear equations via ℓ_1 norm minimization, which is a convex problem [21]. This idea has been employed in the *sparse representation-based classifier* (SRC) [20]. A second approach comes from recent advances on low-rank representations [19]. That is, a representation matrix suitable for classification (i.e., a matrix that contains the representation coefficients in its columns) is found by seeking for the *lowest-rank representation* of the test samples with respect to the training samples. This is achieved by solving a convex problem that involves the minimization of the nuclear norm [19]. Under the assumption made at the beginning of this paragraph, it has been proved that the latter representation possesses both dense within-class affinities and almost zero between-class affinities [19]. Thus, it reveals exactly the classification of the data, resulting into the so-called *low-rank representation-based classifier* (LRRC) [22].

Consequently, when the data samples are low-dimensional and drawn exactly from a union of independent linear subspaces the SRC, the LRRC, or even the nearest subspace-type classifiers (e.g., the *linear regression classifier* (LRC) [23]) can achieve an almost perfect classification accuracy. However, in practice, the data may not come strictly from subspace structures. Small (but densely supported) perturbations of the ideal underlying model may occur as well as arbitrarily large in magnitude (but sparse) deviations may affect a fraction of the data. Both types of errors are termed as *modeling noise* (or

Yannis Panagakis was with the Department of Informatics, Aristotle University of Thessaloniki, Greece and now is with the Department of Computing, Imperial College London, U.K. Constantine Kotropoulos is with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 541 24, Greece; Gonzalo R. Arce is with the Department of Electrical & Computer Engineering, University of Delaware, Newark, Delaware 19716-3130, U.S.A.

Corresponding author: Y. Panagakis, e-mail: yannisp@csd.auth.gr, i.panagakis@imperial.ac.uk.

noise for short) and take into account either real-world measurement/recording errors or outliers. For example, due to the semantic fuzziness between certain music genre classes, such as the boundaries between rock and metal or between rock and pop, a densely supported classification noise is observed in music genre classification. Therefore, a challenging problem is the classification of data that are *approximately* drawn from a union of independent linear subspaces in order to account for the noise. A common method to cope with the noise, is to apply dimensionality reduction to the data by using appropriate subspace learning algorithms, aiming to derive features that belong to the signal subspace and not to that of the noise. The major drawback in the just described approach is that an intermediate stage is introduced before classification, which is often highly demanding in terms of memory or computations, involving also multiple parameters to be properly chosen. In many cases, the subspaces learned are suboptimal as approximate solutions of non-convex problems, e.g., [6].

Here, the just mentioned classification problem is addressed without employing dimensionality reduction. In particular, a novel classification framework is introduced, where the unknown noise in the test data is *simultaneously* corrected and the subspaces, where the test data actually lie onto, are correctly identified. That is, given a sufficiently dense training set and a test set of noisy samples, the *joint sparse low-rank representation* (JSLRR) of the test set with respect to the training set is sought by solving an appropriate convex problem, which involves the nuclear norm, the ℓ_1 norm, and the ℓ_2/ℓ_1 norm minimization. The ℓ_2/ℓ_1 norm is adopted as a regularization term in order to fit the noise. Whenever the samples come *exactly* from a union of independent linear subspaces (i.e., there is no noise), the JSLRR is proved to be dense for within-class affinities, while exhibiting zero between-class affinities, similarly to the sparse representation [24] and the low-rank one [19]. Consequently, the subspaces are revealed, where the test samples lie onto. In the noisy case, the JSLRR is designed to be simultaneously sparse and low-rank in order to combine the benefits of both representations for classification purposes. The joint sparsity constraint implies that only a small fraction of the dictionary atoms is involved in the representation of a test sample with respect to a dictionary formed by the training samples. Such parsimonious representations are desirable, since they are robust to noise, yielding a high classification accuracy [20]. The low-rank constraint is important for two reasons. First, it captures efficiently the underlying data generation process. Indeed, the majority of subspace learning algorithms (e.g., principal component analysis, nonnegative matrix factorization) find a low-rank representation. Second, the rank of the representation increases in the presence of noise, as shown by recent investigations in matrix completion [18]. Therefore, by demanding the representation to be low-rank, noise correction is enforced. Having found the JSLRR of the test samples with respect to the training samples, each test sample is assigned to the class spanned by the subspace yielding the minimum reconstruction error, which results to a novel classier, referred to as *joint sparse low-rank representation*-based classifier. Special cases of the JSLRR are the joint sparse representation (JSR) and the robust low-rank representation (LRR). Based on these representations, another two novel classifiers are developed, namely the *joint sparse representation*-based classifier (JSR) and the *robust low-rank representation*-based one.

In this paper, each music recording is represented by 3 song-level audio features, namely the auditory cortical representations or cortical representation for short [25], the mel-frequency cepstral coefficients [26], and the chroma features [27]. The proposed 3 classifiers, namely the JSLRR-, the JSR-, and the LRR-, based classifiers are applied to the classification of audio features into music genres. Their performance is assessed by conducting experiments on 6 manually annotated benchmark datasets employing both the standard evaluation protocol for each dataset and a small sample size setting. The proposed classifiers are compared against 4 well-known classifiers, namely the SRC [20], the LRC, the SVM with a linear kernel, and the NN classifier with the cosine similarity. The experimental results indicate that the proposed classifiers exhibit a better performance with respect to the music genre classification accuracy than the classifiers they are compared to. Moreover, the best classifications results disclosed are comparable with or slightly superior than those obtained by the state-of-the-art music genre classification methods.

In summary, the contributions the paper are: 1) The development of an efficient algorithm for finding the JSLRR of the test feature vectors with respect to training feature vectors. 2) The proposal of three general purpose classifiers robust to noise that resort to the JSLRR, and its special cases the JSR, and the LRR. 3) The proposal of a novel automatic music genre classification framework by employing the JSLRR-, the JSR, and the LRR-based classifiers for classifying song-level audio features into music genres.

The paper is organized as follows. In Section II, notation conventions are introduced. The audio feature extraction process is briefly described in Section III. The JSLRR as well as its special cases (i.e., the JSR and the LRR) are detailed in Section IV. Classifiers emerging from the aforementioned representations are developed in Section IV. Datasets and experimental results are presented in Section V. Conclusions are drawn in Section VI.

II. Notations

Throughout the paper, scalars are denoted by lowercase letters (e.g., $i, \mu, \epsilon, \theta_1$), vectors appear as lowercase boldface letters (e.g., $\mathbf{x}$), and matrices are indicated by uppercase boldface letters (e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{A}$). $\mathbf{I}$ is the identity matrix of compatible dimensions. The ith column of $\mathbf{X}$ is denoted by $\mathbf{x}_i$. Let span$(\mathbf{X})$ be the linear space spanned by the columns of $\mathbf{X}$. Then, $\mathbf{Y} \in$ span$(\mathbf{X})$ implies that all column vectors of $\mathbf{Y}$ belong to span$(\mathbf{X})$. The set of real numbers is denoted by $\mathbb{R}$, while the set of nonnegative real numbers is denoted by $\mathbb{R}_+$.

A variety of norms on real-valued vectors and matrices are used. For example, $\|\mathbf{x}\|_0$ is the ℓ_0 quasi-norm counting the number of nonzero entries in $\mathbf{x}$. If $|.|$ denotes the absolute value operator, $\|\mathbf{x}\|_1 = \sum_i |x_i|$ and $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$ are the ℓ_1 and the ℓ_2 norm of $\mathbf{x}$, respectively. The mixed $\ell_{p,q}$ matrix norm is

defined as $\|\mathbf{X}\|_{p,q} = \left(\sum_i \left(\sum_j |x_{ij}|^p\right)^{\frac{q}{p}}\right)^{\frac{1}{q}}$. For $p = q = 0$, the matrix ℓ_0 quasi-norm is denoted by $\|\mathbf{X}\|_0$ and returns the number of nonzero entries in $\mathbf{X}$. For $p = q = 1$, the matrix ℓ_1 norm is defined as $\|\mathbf{X}\|_1 = \sum_i \sum_j |x_{ij}|$. The Frobenius norm is defined as $\|\mathbf{X}\|_F = \sqrt{\sum_i \sum_j x_{ij}^2}$. The ℓ_2/ℓ_1 norm of $\mathbf{X}$ is given by $\|\mathbf{X}\|_{2,1} = \sum_j \sqrt{\sum_i x_{ij}^2}$. Clearly, the $\ell_{2,1}$ norm and the ℓ_2/ℓ_1 norm are different norms. The former is the sum of ℓ_2 norms of the row vectors of $\mathbf{X}$, while the latter is the sum of the ℓ_2 norms of the column vectors of $\mathbf{X}$. This ambiguity in notation is resolved by the context. The nuclear norm of $\mathbf{X}$ (i.e., the sum of singular values of a matrix) is denoted by $\|\mathbf{X}\|_*$. The ℓ_∞ norm of $\mathbf{X}$, denoted by $\|\mathbf{X}\|_\infty$, is defined as the element of $\mathbf{X}$ with the maximum absolute value. $\mathbf{X}^T$ is the transpose of $\mathbf{X}$. If $\mathbf{X}$ is a square matrix, $\mathbf{X}^{-1}$ is its inverse, provided that the inverse matrix exists.

A vector $\mathbf{x}$ is said to be q-sparse if the size of the support of $\mathbf{x}$ (i.e., the set of indices associated to non-zero vector elements) is no larger than q: $|\text{supp}(\mathbf{x})| \leq q$. The support of a collection of vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ is defined as the union over all the individuals supports: $\text{supp}(\mathbf{X}) \triangleq \bigcup_{n=1}^{N} \text{supp}(\mathbf{x}_n)$. A matrix $\mathbf{X}$ is called q joint sparse, if $|\text{supp}(\mathbf{X})| \leq q$. That is, there are at most q rows in $\mathbf{X}$ that contain nonzero elements, because $\|\mathbf{X}\|_{0,q} = |\text{supp}(\mathbf{X})|$ for any q [28].

III. Audio Feature Extraction

Each music recording is represented by 3 song-level feature vectors.

A. Auditory cortical representations

The auditory sensations turn into perception and cognition only when they are processed by the cortical area [29]. The mechanical and neural processing in the early and central stages of the auditory system can be modeled as a two-stage process. At the first stage, which models the cochlear, the audio signal is converted into an auditory representation, i.e., the auditory spectrogram, by employing the constant-Q transform (CQT). The CQT is a time-frequency representation where the frequency bins are geometrically spaced and the Q-factors (i.e., the ratios of the center frequencies to the bandwidths) of all bins are equal [30]. The neurons in the primary auditory cortex are organized according to their selectivity on different spectral and temporal stimuli [29]. To this end, in the second stage, the spectral and temporal modulation content of the auditory spectrogram is estimated by two-dimensional (2D) multiresolution wavelet analysis, ranging from slow to fast temporal rates and from narrow to broad spectral scales. The analysis yields a four-dimensional (4D) representation of time, frequency, rate, and scale that captures the slow spectral and temporal modulations content of audio that is referred to as *auditory cortical representation*. More details on the mathematical formulation of the auditory cortical representations can be found in [25].

Parameters and implementation. The CQT is computed efficiently by employing the fast implementation scheme proposed in [30]. The audio signal is analyzed by employing 128 constant-Q filters covering 8 octaves from 44.9 Hz to 11 KHz (i.e., 16 filters per octave). The magnitude of the CQT is compressed by raising each element of the CQT matrix to the power of 0.1. At the second stage, the 2D multiresolution wavelet analysis is implemented via a bank of 2D Gaussian filters with $scales \in \{0.25, 0.5, 1, 2, 4, 8\}$ (Cycles / Octave) and (both positive and negative) $rates \in \{\pm 2, \pm 4, \pm 8, \pm 16, \pm 32\}$ (Hz) [25]. For each music recording, the extracted 4D cortical representation is time-averaged and a rate-scale-frequency 3D cortical representation is thus obtained. The overall procedure is depicted in Fig. 1. Accordingly by stacking the elements of the 3D cortical representation into a vector, each music recording can be represented by a vector $\mathbf{x} \in \mathbb{R}_+^{7680}$. The dimension of the vectorized cortical representation comes from the product of 128 frequency channels, 6 scales, and 10 rates. An ensemble of music recordings is represented by the data matrix $\mathbf{X} \in \mathbb{R}_+^{7680 \times S}$, where S is the number of the available recordings. Finally, the entries of $\mathbf{X}$ are post-processed as follows: Each row of $\mathbf{X}$ is normalized to the range $[0, 1]$ by subtracting from each entry the row minimum and then by dividing it with the difference between the row maximum and the row minimum.

B. Mel-frequency cepstral coefficients and chroma features

The MFCCs encode the timbral properties of the music signal by encoding the rough shape of the log-power spectrum on the mel-frequency scale [26]. They exhibit the desirable property that a numerical change in the MFCC coefficients corresponds to a perceptual change. The MFCC extraction employs frames of duration 92.9 ms with a hop size of 46.45 ms, and a 42 bandpass filter bank. The filters are uniformly spaced on the Mel-frequency scale. The correlation between the frequency bands is reduced by applying the discrete cosine transform along the log-energies of the bands, yielding a sequence of 20-dimensional MFCC vectors.

The chroma features [27] characterize the harmonic content of the music signal by projecting the entire spectrum onto 12 bins, representing the 12 distinct semitones (or chroma) of a musical octave. They are calculated by employing 92.9 ms frames with a hop size of 23.22 ms as follows. First, the salience of different fundamental frequencies in the range $80 - 640$ Hz is calculated. The linear frequency scale is transformed into a musical one by selecting the maximum salience value in each frequency range corresponding to one semitone. Finally, the octave equivalence classes are summed over the whole pitch range to yield a sequence of 12-dimensional chroma vectors.

Following Mandel *et al.* [31], the mean and the full covariance matrix of the MFCCs and the chroma features are computed over the duration of each music recording. Consequently, each song is represented by a 420-dimensional MFCC song-level vector and a 156-dimensional song-level chroma vector, which are obtained by stacking the mean vectors on the top of the vectorized covariance matrices.

The song-level chroma and the MFCCs, extracted from an ensemble of music recordings, are normalized over all songs to be zero-mean with unit-variance. Furthermore, they are post-processed as described in subsection III-A.

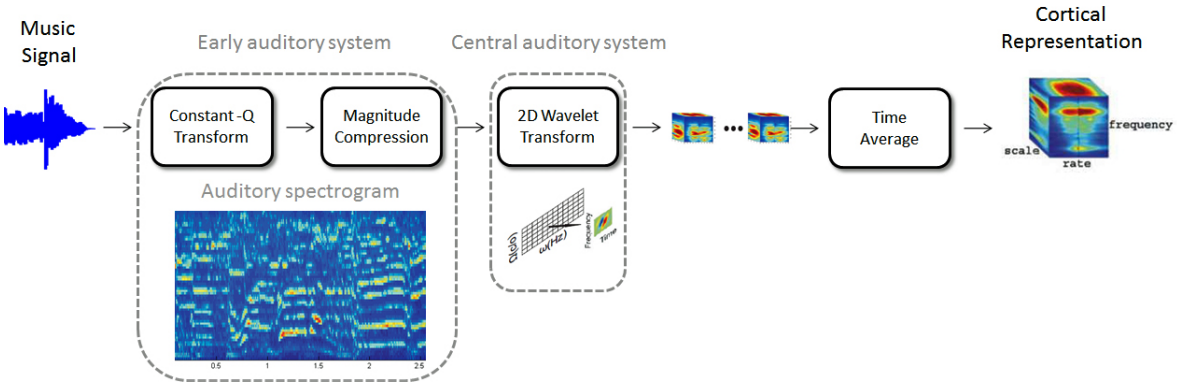

Fig. 1. Flow chart of cortical representation extraction.

IV. Classification Via Joint Sparse Low-Rank Representation

First, the *sparsest representation* (SR) [24] and the *lowest-rank representation* (LRR) [19] are briefly introduced. Next, the JSLRR is developed in order to account for the noise and the outliers. Finally, three novel classifiers are proposed, which resort to the JSLRR, and its special cases, namely the JSR, the LRR, respectively.

A. Suitable data representations for classification

Let $\mathbf{X} \in \mathbb{R}^{d\times S}$ be the data matrix that contains S vector samples of size d in its columns. That is, $\mathbf{x}_s \in \mathbb{R}^d$, $s = 1, 2, \ldots, S$. Without loss of generality, the data matrix can be partitioned as $\mathbf{X} = [\mathbf{A} \mid \mathbf{Y}]$, where $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2|\ldots|\mathbf{A}_K] \in \mathbb{R}^{d\times N}$ represents a set of N training samples that belong to K classes and $\mathbf{Y} = [\mathbf{Y}_1|\mathbf{Y}_2|\ldots|\mathbf{Y}_K] \in \mathbb{R}^{d\times M}$ contains $M = S - N$ test samples in its columns. Assume that the training samples are drawn from a union of K independent linear subspaces of unknown dimensions. The columns of $\mathbf{A}_k \in \mathbb{R}^{d\times N_k}$, $k = 1, 2, \ldots, K$ correspond to the N_k training samples originating from the kth subspace. Similarly, the columns of $\mathbf{Y}_k \in \mathbb{R}^{d\times M_k}$ refer to M_k test samples stemming from the kth class[1].

Assumption. If: 1) the data are drawn *exactly* from independent linear subspaces, i.e., span$(\mathbf{A}_k)$ linearly spans the kth class data space, $k = 1, 2, \ldots, K$, 2) $\mathbf{Y} \in$ span$(\mathbf{A})$, and 3) the data contain neither outliers nor noise, then each test vector sample that belongs to the kth class is represented as a linear combination of the training samples in $\mathbf{A}_k$. That is, $\mathbf{Y}_k = \mathbf{A}_k\mathbf{Z}_k$ with $\mathbf{Z}_k \in \mathbb{R}^{N_k\times M_k}$. Accordingly, $\mathbf{Y} = \mathbf{A}\mathbf{Z}$, where $\mathbf{Z} = \text{diag}[\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_K] \in \mathbb{R}^{N\times M}$ is the block-diagonal *representation matrix*. Therefore, the ith test sample is represented as $\mathbf{y}_i = \mathbf{A}\mathbf{z}_i \in \mathbb{R}^d$, where $\mathbf{z}_i = [\mathbf{0}^T|\ldots|\mathbf{0}^T|\mathbf{z}_k^T|\mathbf{0}^T|\ldots|\mathbf{0}^T]^T \in \mathbb{R}^N$ is the augmented coefficient vector, whose elements are non-zero if they weigh training vectors stemming from the kth class. Consequently, having found such a block-diagonal representation matrix $\mathbf{Z}$ capturing both dense within-class affinities and zero between-class affinities, the classification of the data is revealed exactly.

[1] $\sum_{k=1}^K N_k = N$ and $\sum_{k=1}^K M_k = M$.

Indeed, under the aforementioned three assumptions it has been proved that the block-diagonal representation matrix $\mathbf{Z} \in \mathbb{R}^{N\times M}$ is the *sparsest representation* (SR) [24] or the *lowest-rank representation* [19] of the test data $\mathbf{Y} \in \mathbb{R}^{d\times M}$ with respect to the training data $\mathbf{A} \in \mathbb{R}^{d\times N}$. Equivalently, the representation matrix $\mathbf{Z}$ is obtained by solving the optimization problem (1) for the SR and (2) for the lowest-rank representation:

$$\underset{\mathbf{z}_i}{\text{argmin}} \quad \|\mathbf{z}_i\|_0 \quad \text{subject to } \mathbf{y}_i = \mathbf{A}\,\mathbf{z}_i, \tag{1}$$

$$\underset{\mathbf{Z}}{\text{argmin}} \quad \text{rank}(\mathbf{Z}) \quad \text{subject to } \mathbf{Y} = \mathbf{A}\,\mathbf{Z}. \tag{2}$$

Problems (1) and (2) are non-convex and NP-hard, in general, due to the discrete nature of the ℓ_0 norm [32] and the rank function [33] and thus they are difficult to be solved. It has been proved that the convex envelope of the ℓ_0 norm is the ℓ_1 norm [21], while the convex envelope of the rank function is the nuclear norm [34]. It is worth mentioning that rank minimization generalizes the notion of vector sparsity to spectrum sparsity for matrices [34]. Consequently, convex relaxations of (1) and (2) are obtained by replacing the ℓ_0 norm and the rank function by their convex envelopes as follows:

$$\underset{\mathbf{z}_i}{\text{argmin}} \quad \|\mathbf{z}_i\|_1 \quad \text{subject to } \mathbf{y}_i = \mathbf{A}\,\mathbf{z}_i, \tag{3}$$

$$\underset{\mathbf{Z}}{\text{argmin}} \quad \|\mathbf{Z}\|_* \quad \text{subject to } \mathbf{Y} = \mathbf{A}\,\mathbf{Z}. \tag{4}$$

Under the conditions set in [21], the solution of (1) is equivalent to that of (3). Similarly, the solution of (4) is always a solution of (2) [19].

The SR matrix $\mathbf{Z} \in \mathbb{R}^{N\times M}$, which contains the representation vectors $\mathbf{z}_m$, $m = 1, 2, \ldots, M$ in its columns obtained by solving (3), is *sparse* block-diagonal [24]. That is, the within-class affinities are nonzero, but sparse and the between-class affinities are all zeros. Consequently, it has good discriminating properties, making it suitable for classification, as has been demonstrated for the SRC [20]. However, the SR seems to face some difficulties in modeling generic subspace structures. Indeed, the SR models accurately subregions on subspaces, the so-called *bouquets*, rather than generic subspaces [35]. Furthermore, the SR does not capture the global structure of the data, since it is computed for each data sample individually.

This may affect the SRC performance, when the data are heavily contaminated due to the damage of the high within-class homogeneity [19].

The solution of (4) provides the lowest-rank representation matrix $\mathbf{Z} \in \mathbb{R}^{N\times M}$, which alleviates the aforementioned drawbacks of the SR. In particular, $\mathbf{Z}$ being low-rank, it models the data generation process. Since the lowest-rank representation is computed by taking into account all the data, it preserves more accurately the data structure than the SR does [19]. In the case of clean data, the lowest-rank representation also exhibits *dense* within-class homogeneity and zero between-class affinities, making it an attractive representation for classification, as has been demonstrated for the LRR-based classifier in music mood classification [22]. For data contaminated with noise and outliers, the low-rank constraint seems to enforce noise correction [18], [19].

B. Joint sparse low-rank representation

Considering the properties of the sparse representation and the low-rank one in subspace modeling, it is interesting to identify the most *characteristic* subregions of the subspaces spanning the data. Intuitively, a representation matrix is needed that is simultaneously *row sparse* and *low-rank*. The row sparsity ensures that only a small fraction of the training samples is involved in the representation. In addition, the low-rank constraint ensures that the representation vectors (i.e., the columns of the representation matrix) are correlated in the sense that the data lying onto a particular subspace are represented as a linear combination of the same few training samples. Such a representation is referred to as JSLRR. Furthermore, in the presence of noise, both the rank and the density of the representation matrix increases, since its columns contain non-zero elements associated with more than one class. Accordingly, if one demands to reduce the rank of the representation matrix or to increase its sparsity, the noise in the test set can be smoothed and simultaneously the representation matrix admits a structure close to a block-diagonal one, which is desirable for data classification. Moreover, since the JSLRR is row sparse, the contaminated training samples are expected not to be involved in the representation of the test samples. In that sense, the JSLRR provides a more robust to noise representation than the SR and the lowest-rank representation. This property of the JSLRR is illustrated in the Example 1 at the end of this subsection.

Formally, given the test data $\mathbf{Y} \in \mathbb{R}^{d\times M}$ and the training data $\mathbf{A} \in \mathbb{R}^{d\times N}$, the JSLRR of $\mathbf{Y}$ with respect to $\mathbf{A}$ is the matrix $\mathbf{Z} \in \mathbb{R}^{N\times M}$ with rank $r \ll \min(q, M)$, where $q \ll N$ is the size of the support of $\mathbf{Z}$. Such a representation is found by minimizing the rank function regularized by the $\ell_{0,q}$ quasi-norm. The $\ell_{0,q}$ regularization term ensures that the low-rank matrix is also row sparse, since $\|\mathbf{Z}\|_{0,q} = |\text{supp}(\mathbf{Z})|$ for any q [28]. Such an optimization problem is an NP-hard non-convex problem. A convex relaxation of the just mentioned problem is considered by replacing the rank function and the $\ell_{0,q}$ quasi-norm by their convex envelopes.

In practice, the assumption stated in subsection IV-A does not hold exactly. That is, the data are *approximately* drawn from a union of subspaces. This fact introduces certain deviations from the ideal modeling assumptions. These deviations can be treated collectively as additive *noise* contaminating the ideal model (i.e., $\mathbf{Y} = \mathbf{AZ} + \mathbf{E}$). The noise term $\mathbf{E}$ models both small (but densely supported) deviations and grossly (but sparse) corrupted observations (i.e., outliers or missing data). To account for the noise, the JSLRR solves the following convex optimization problem:

$$\begin{aligned} \underset{\mathbf{Z},\mathbf{E}}{\text{argmin}} \quad & \|\mathbf{Z}\|_* + \theta_1\|\mathbf{Z}\|_1 + \theta_2\|\mathbf{E}\|_{2,1} \\ & \text{subject to } \mathbf{Y} = \mathbf{A}\,\mathbf{Z} + \mathbf{E}, \end{aligned} \tag{5}$$

where $\theta_2 > 0$ is a regularization parameter and $\|.\|_{2,1}$ denotes the ℓ_2/ℓ_1 norm. The choice of ℓ_2/ℓ_1 norm for noise characterization is attributed to the assumptions for the noise term $\mathbf{E}$ stated previously. Several JSLRR methods have been proposed in the literature. Table I indicates the optimization problem solved in each method and reveals the novelty of the proposed JSLRR formulation (5). Parameters θ_1 and θ_2 can be selected as follows: $\theta_1 = \sqrt{r/q}$, requiring a rough estimation of the rank r and sparsity-level q of the representation matrix [36] and $\theta_2 = \frac{3}{7\sqrt{\gamma\cdot M}}$, where γ denotes the estimated portion of outliers in the test set [37].

By assuming that there are no outliers in the test set (i.e., $\theta_2 = 0$ and $\mathbf{E} = \mathbf{0}$ in (5)), the JSLRR (i.e., $\mathbf{Z}$) has a block-diagonal structure, a property that makes it appealing for classification. This is guaranteed by Theorem 1, which is a consequence of Lemma 1. The proofs of both Lemma 1 and Theorem 1 can be found in the Supplementary Material.

Lemma 1: Let $\|.\|_\theta = \|.\|_* + \theta\|.\|_1$, with $\theta > 0$. For any four matrices $\mathbf{B}, \mathbf{C}, \mathbf{D}$, and $\mathbf{F}$ of compatible dimensions,

$$\left\|\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{D} & \mathbf{F} \end{bmatrix}\right\|_\theta \geq \left\|\begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix}\right\|_\theta = \|\mathbf{B}\|_\theta + \|\mathbf{F}\|_\theta. \tag{6}$$

Theorem 1: Assume that the data are exactly drawn from independent linear subspaces. That is, span$(\mathbf{A}_k)$ linearly spans the training vectors of the kth class, $k = 1, 2, \ldots, K$, $\mathbf{Y} \in$ span$(\mathbf{A})$ and $\mathbf{E} = \mathbf{0}$. Then, the minimizer of (5) is block-diagonal.

Problem (5) can be solved by employing various optimization methods, namely semi-definite programming [44], first order proximal-point methods, such as the iterative soft-thresholding algorithm (ISTA) [45], the fast iterative soft-thresholding algorithm (FISTA) [46], the SpaRSA [20], and the accelerated proximal gradient (APG) [47], as well as the algorithms that resort to the augmented Lagrange multiplier (ALM) methods [48], [49]. In this paper, the *linearized alternating direction augmented Lagrange multiplier* (LADALM) method is employed for solving (5). This choice is attributed to the fact that it suits well for large scale classification problems, yielding higher classification accuracy than other methods like FISTA [50]. The LADALM is a variant of the *alternating direction augmented Lagrange multiplier* (ADALM) method [49] for optimization problems, whose subproblems not admitting closed-formed solutions are linearized, to obtain closed-

TABLE I
POPULAR JSLRR METHODS.

Reference	Objective	Sparsity inducing representation	Error term
Here	$\operatorname{argmin}_{\mathbf{Z},\mathbf{E}} \quad \|\mathbf{Z}\|_* + \theta_1 \|\mathbf{Z}\|_1 + \theta_2 \|\mathbf{E}\|_{2,1}$ subject to $\mathbf{Y} = \mathbf{A}\,\mathbf{Z} + \mathbf{E}$	$\|\mathbf{Z}\|_1$	$\|\mathbf{E}\|_{2,1}$
[38]	$\operatorname{argmin}_{\mathbf{Z},\mathbf{E}} \quad \|\mathbf{Z}\|_* + \theta \|\mathbf{E}\|_1$ subject to $\mathbf{Y} = \mathbf{Z} + \mathbf{E}$	-	$\|\mathbf{E}\|_1$
[39]	$\operatorname{argmin}_{\mathbf{Z}} \quad \|\mathbf{Z}\|_* + \theta 1 \|\mathbf{Z}\|_1 + \theta_2 \|\mathbf{Y} - \mathbf{Z}\|_F^2$	$\|\mathbf{Z}\|_1$	$\|\mathbf{Y} - \mathbf{Z}\|_F^2$
[40]	$\operatorname{argmin}_{\mathbf{Z}} \quad \|\mathbf{Z}\|_* + \theta_1 \|\mathbf{Z}\|_1 + \theta_1 \|\mathbf{E}\|_1$ subject to $\mathcal{P}_\Omega(\mathbf{B}_1\,\mathbf{Z}\,\mathbf{B}_2^T + \mathbf{E}) = \mathcal{P}_\Omega(Y)$ for some properly chosen bases $(\mathbf{B}_1, \mathbf{B}_2)$, where $\mathcal{P}_\Omega(\cdot)$ is a linear operator that restricts the equality only on the entries which belong to the domain Ω (i.e., a subset of observations)	$\|\mathbf{Z}\|_1$	$\mathcal{P}_\Omega(\mathbf{B}_1\,\mathbf{Z}\,\mathbf{B}_2^T + \mathbf{E}) = \mathcal{P}_\Omega(Y)$
[41]	$\operatorname{argmin}_{\mathbf{Z},\mathbf{E}} \quad \|\mathbf{Z}\|_* + \theta_1 \|\mathbf{Z}\|_1 + \theta_2 \|\mathbf{E}\|_1$ subject to $\mathbf{Y} = \mathbf{A}\mathbf{Z} + \mathbf{E}$	$\|\mathbf{Z}\|_1$	$\|\mathbf{E}\|_1$
[36]	$\operatorname{argmin}_{\mathbf{Z}} \quad \|\mathbf{Z}\|_* + \theta \|\mathbf{Z}\|_{2,1}$ subject to $\|\mathbf{Y} - \mathbf{A}\mathbf{\Phi}_{2D}\mathbf{Z}\|_F^2 \leq \varepsilon$ for a threshold ε, where $\mathbf{\Phi}_{2D}$ is a 2D spatial wavelet basis	$\|\mathbf{Z}\|_{2,1}$	$\|\mathbf{Y} - \mathbf{A}\mathbf{\Phi}_{2D}\mathbf{Z}\|_F^2 \leq \varepsilon$
[42]	$\operatorname{argmin}_{\mathbf{Z}} \quad \|\mathbf{Y} - \mathbf{A}\mathbf{Z}\|_F^2 + \theta_1 \|\mathbf{Z}\|_{1,2} + \theta_2 \sum_{l=1}^K \omega_i \|\mathbf{Z} - \mathbf{Z}_l\|_F^2$ where $\mathbf{Z}$ and $\mathbf{Z}_l$, $l = 1, 2, \ldots, K$, is the sparse code of the encoded group $\mathcal{G}$ and its baseline group $\mathcal{G}_l$ and ω_l is the appearance similarity between them	$\theta_1 \|\mathbf{Z}\|_{1,2} + \theta_2 \sum_{l=1}^K \omega_i \|\mathbf{Z} - \mathbf{Z}_l\|_F^2$	$\|\mathbf{Y} - \mathbf{A}\mathbf{Z}\|_F^2$
[43]	$\operatorname{argmin}_{\mathbf{Z},\mathbf{E}} \quad \|\mathbf{Z}\|_* + \theta_1 \|\mathbf{Z}^T\|_1 + \theta_2 \|\mathbf{E}\|_{2,1}$ subject to $\mathbf{Y} = \mathbf{Y}\,\mathbf{Z} + \mathbf{E}$ and $\mathbf{Z} \geq 0$.	$\|\mathbf{Z}^T\|_1$	$\|\mathbf{E}\|_{2,1}$

formed solutions [48]. To this end, (5) is rewritten as

$$\begin{aligned} \underset{\mathbf{J},\mathbf{Z},\mathbf{W},\mathbf{E}}{\operatorname{argmin}} \quad & \|\mathbf{J}\|_* + \theta_1 \|\mathbf{W}\|_1 + \theta_2 \|\mathbf{E}\|_{2,1} \\ \text{subject to } & \mathbf{Y} = \mathbf{A}\,\mathbf{Z} + \mathbf{E},\ \mathbf{Z} = \mathbf{J},\ \mathbf{J} = \mathbf{W}, \end{aligned} \tag{7}$$

which can be solved by minimizing the augmented Lagrangian function:

$$\begin{aligned} &\mathcal{L}(\mathbf{J}, \mathbf{Z}, \mathbf{W}, \mathbf{E}, \mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3) = \|\mathbf{J}\|_* + \theta_1 \|\mathbf{W}\|_1 \\ &+\theta_2 \|\mathbf{E}\|_{2,1} + \operatorname{tr}\left(\mathbf{\Lambda}_1^T (\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E})\right) + \operatorname{tr}\left(\mathbf{\Lambda}_2^T (\mathbf{Z} - \mathbf{J})\right) \\ &+\operatorname{tr}\left(\mathbf{\Lambda}_3^T (\mathbf{J} - \mathbf{W})\right) + \frac{\mu}{2}\big(\|\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E}\|_F^2 \\ &+\|\mathbf{Z} - \mathbf{J}\|_F^2 + \|\mathbf{J} - \mathbf{W}\|_F^2\big), \end{aligned} \tag{8}$$

where $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2$, and $\mathbf{\Lambda}_3$ gather the Lagrange multipliers and $\mu > 0$ is a penalty parameter. By employing the LADALM, (8) is minimized with respect to each variable in an alternating fashion and finally the Lagrange multipliers are updated at each iteration. Let t denote the iteration index. Given $\mathbf{J}_{[t]}, \mathbf{Z}_{[t]}, \mathbf{W}_{[t]}, \mathbf{E}_{[t]}$, and μ, the iteration of LADALM for (7) reads

$$\mathbf{J}_{[t+1]} = \underset{\mathbf{J}_{[t]}}{\operatorname{argmin}} \mathcal{L}(\mathbf{J}_{[t]}, \mathbf{Z}_{[t]}, \mathbf{W}_{[t]}, \mathbf{E}_{[t]}, \mathbf{\Lambda}_{1[t]}, \mathbf{\Lambda}_{2[t]}, \mathbf{\Lambda}_{3[t]}), \tag{9}$$

$$\mathbf{Z}_{[t+1]} = \underset{\mathbf{Z}_{[t]}}{\operatorname{argmin}} \mathcal{L}(\mathbf{J}_{[t+1]}, \mathbf{Z}_{[t]}, \mathbf{W}_{[t]}, \mathbf{E}_{[t]}, \mathbf{\Lambda}_{1[t]}, \mathbf{\Lambda}_{2[t]}, \mathbf{\Lambda}_{3[t]}), \tag{10}$$

$$\mathbf{W}_{[t+1]} = \underset{\mathbf{W}_{[t]}}{\operatorname{argmin}} \mathcal{L}(\mathbf{J}_{[t+1]}, \mathbf{Z}_{[t+1]}, \mathbf{W}_{[t]}, \mathbf{E}_{[t]}, \mathbf{\Lambda}_{1[t]}, \mathbf{\Lambda}_{2[t]}, \mathbf{\Lambda}_{3[t]}) \tag{11}$$

$$\approx \underset{\mathbf{W}_{[t]}}{\operatorname{argmin}} \frac{\theta_1}{\mu} \|W_{[t]}\|_1 + \frac{1}{2} \|W_{[t]} - (\mathbf{J}_{[t+1]} + \mathbf{\Lambda}_{3[t]}/\mu)\|_F^2, \tag{12}$$

$$\mathbf{E}_{[t+1]} = \underset{\mathbf{E}_{[t]}}{\operatorname{argmin}} \mathcal{L}(\mathbf{J}_{[t+1]}, \mathbf{Z}_{[t+1]}, \mathbf{W}_{[t+1]}, \mathbf{E}_{[t]}, \mathbf{\Lambda}_{1[t]}, \mathbf{\Lambda}_{2[t]}, \mathbf{\Lambda}_{3[t]}) \tag{13}$$

$$\approx \underset{\mathbf{E}_{[t]}}{\operatorname{argmin}} \frac{\theta_2}{\mu} \|\mathbf{E}_{[t]}\|_{2,1} + \frac{1}{2} \|\mathbf{E}_{[t]} - (\mathbf{Y} - \mathbf{A}\mathbf{Z}_{[t+1]} + \mathbf{\Lambda}_{1[t]}/\mu)\|_F^2, \tag{14}$$

$$\begin{aligned} \mathbf{\Lambda}_{1[t+1]} &= \mathbf{\Lambda}_{1[t]} + \mu(\mathbf{Y} - \mathbf{A}\mathbf{Z}_{[t+1]} - \mathbf{E}_{[t+1]}), \\ \mathbf{\Lambda}_{2[t+1]} &= \mathbf{\Lambda}_{2[t]} + \mu(\mathbf{Z}_{[t+1]} - \mathbf{J}_{[t+1]}), \\ \mathbf{\Lambda}_{3[t+1]} &= \mathbf{\Lambda}_{3[t]} + \mu(\mathbf{J}_{[t+1]} - \mathbf{W}_{[t+1]}), \end{aligned}$$

In order to solve (9), we have to minimize (8) with respect to $\mathbf{J}$, which does not admit a closed form solution. Let $f(\mathbf{J})$ be the smooth term in (8) i.e., $f(\mathbf{J}) = \operatorname{tr}\left(\mathbf{\Lambda}_1^T (\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E})\right) + \operatorname{tr}\left(\mathbf{\Lambda}_2^T (\mathbf{Z} - \mathbf{J})\right) + \operatorname{tr}\left(\mathbf{\Lambda}_3^T (\mathbf{J} - \mathbf{W})\right) + \frac{\mu}{2}\big(\|\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E}\|_F^2 + \|\mathbf{Z} - \mathbf{J}\|_F^2 + \|\mathbf{J} - \mathbf{W}\|_F^2\big)$. Following [48], $f(\mathbf{J})$ is linearly approximated with respect to $\mathbf{J}$ at $\mathbf{J}_{[t]}$ as: $f(\mathbf{J}) \approx f(\mathbf{J}_{[t]}) + \operatorname{tr}\left((\mathbf{J} - \mathbf{J}_{[t]})^T \nabla f(\mathbf{J}_{[t]})\right) + \frac{\mu}{2}\|\mathbf{J} - \mathbf{J}_{[t]}\|_F^2$, where $\nabla f(\mathbf{J}) = -\mathbf{\Lambda}_{2[t]} + \mathbf{\Lambda}_{3[t]} + \mu(2\mathbf{J}_{[t]} - \mathbf{Z}_{[t]} - \mathbf{W}_{[t]})$. Therefore, an approximate solution of (9) is obtained by minimizing the linearized augmented Lagrangian function as follows:

$$\begin{aligned} \mathbf{J}_{[t+1]} &\approx \underset{\mathbf{J}}{\operatorname{argmin}} \|\mathbf{J}\|_* + f(\mathbf{J}_{[t]}) \\ &\quad + \operatorname{tr}\left((\mathbf{J} - \mathbf{J}_{[t]})^T \nabla f(\mathbf{J}_{[t]})\right) + \frac{\mu}{2}\|\mathbf{J} - \mathbf{J}_{[t]}\|_F^2 \\ &= \underset{\mathbf{J}}{\operatorname{argmin}} \|\mathbf{J}\|_* + \frac{\mu}{2}\|\mathbf{J} - (\mathbf{J}_{[t]} - \frac{1}{\mu}\nabla f(\mathbf{J}_{[t]})\|_F^2 \\ &= \mathcal{D}_{\mu^{-1}}[\mathbf{Z}_{[t]} - \mathbf{J}_{[t]} - \mathbf{\Lambda}_{3[t]}/\mu + \mathbf{W}_{[t]} + \mathbf{\Lambda}_{2[t]}/\mu], \end{aligned} \tag{15}$$

where $\mathcal{D}_\tau[\mathbf{Q}] = \mathbf{U}\mathcal{S}_\tau\mathbf{V}^\mathcal{T}$ is the singular value thresholding operator for any matrix $\mathbf{Q}$ with singular value decomposition (SVD) $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. $\mathcal{S}_\tau[q] = \operatorname{sgn}(q)\ \max(|q| - \tau, 0)$ is the shrinkage operator that is extended to matrices if applied elementwise [18]. Problem (10) is an unconstrained least-squares problem, admitting a simple closed form solution. It is easy to see that (11) and (13) reduce into (12), and (14), respectively. In particular, the subproblem (12) has a unique solution, that is obtained by the shrinkage operator: $\mathbf{W}_{[t+1]} = \mathcal{S}_{\theta_1\mu^{-1}}[\mathbf{J}_{[t+1]} + \mathbf{\Lambda}_{3[t]}/\mu]$. Let $\mathbf{M}_{[t]} = \mathbf{Y} - \mathbf{A}\mathbf{Z}_{[t+1]} + \mathbf{\Lambda}_{1[t]}/\mu$. The solution of (14) is obtained column-wise as

$\mathbf{e}_{j[t+1]} = \mathcal{S}_{\theta_2\mu^{-1}}[\|\mathbf{m}_{j[t]}\|_2]\frac{\mathbf{m}_{j[t]}}{\|\mathbf{m}_{j[t]}\|_2}$ [51]. The LADALM method for the minimization of (7) is outlined in Algorithm 1. The dominant cost of each iteration in Algorithm 1 is the computation the singular value thresholding operator (i.e., Step 3). That is, the calculation of the singular vectors of $\left(\mathbf{Z}_{[t]} - \mathbf{J}_{[t]} - \mathbf{\Lambda}_{3[t]}/\mu + \mathbf{W}_{[t]} + \mathbf{\Lambda}_{2[t]}/\mu\right)$ whose corresponding singular values are larger than the threshold μ^{-1}, yielding a $O(N^3)$ complexity at each iteration.

Algorithm 1 Solving (7) by the LADALM method.

Input: Training matrix $\mathbf{A} \in \mathbb{R}^{d\times N}$, test matrix $\mathbf{Y} \in \mathbb{R}^{d\times M}$ and the parameters θ_1, θ_2.
Output: Matrix $\mathbf{Z} \in \mathbb{R}^{N\times M}$ and matrix $\mathbf{E} \in \mathbb{R}^{d\times M}$.

1: Initialize: $\mathbf{Z}_{[0]} = \mathbf{J}_{[0]} = \mathbf{W}_{[0]} = \mathbf{0}$, $\mathbf{\Lambda}_{1[0]} = \mathbf{0}$, $\mathbf{\Lambda}_{2[0]} = \mathbf{0}$, $\mathbf{\Lambda}_{3[0]} = \mathbf{0}$, $\mu = 10^{-6}$, $\rho = 1.1$, $\epsilon = 10^{-8}$.
2: **while** not converged **do**
3: Fix $\mathbf{Z}_{[t]}$, $\mathbf{W}_{[t]}$, and $\mathbf{E}_{[t]}$, and update $\mathbf{J}_{[t+1]}$ by $\mathbf{J}_{[t+1]} \leftarrow \mathcal{D}_{\mu^{-1}}[\mathbf{Z}_{[t]} - \mathbf{J}_{[t]} - \mathbf{\Lambda}_{3[t]}/\mu + \mathbf{W}_{[t]} + \mathbf{\Lambda}_{2[t]}/\mu]$.
4: Fix $\mathbf{J}_{[t+1]}$, $\mathbf{W}_{[t]}$, and $\mathbf{E}_{[t]}$, and update $\mathbf{Z}_{[t+1]}$ by $\mathbf{Z}_{[t+1]} \leftarrow \left(\mathbf{I} + \mathbf{A}^T\mathbf{A}\right)^{-1}(\mathbf{A}^T(\mathbf{Y} - \mathbf{E}_{[t]}) + \mathbf{J}_{[t+1]} + (\mathbf{A}^T\mathbf{\Lambda}_{1[t]} - \mathbf{\Lambda}_{2[t]})/\mu)$.
5: Fix $\mathbf{J}_{[t+1]}$, $\mathbf{Z}_{[t+1]}$, and $\mathbf{E}_{[t]}$, and update $\mathbf{W}_{[t+1]}$ by $\mathbf{W}_{[t+1]} \leftarrow \mathcal{S}_{\theta_1\mu^{-1}}[\mathbf{J}_{[t+1]} + \mathbf{\Lambda}_{3[t]}/\mu]$.
6: Fix $\mathbf{Z}_{[t+1]}$, $\mathbf{J}_{[t+1]}$, and $\mathbf{W}_{[t+1]}$, form $\mathbf{M}_{[t]} = \mathbf{Y} - \mathbf{A}\mathbf{Z}_{[t+1]} + \mathbf{\Lambda}_{1[t]}/\mu$, and update $\mathbf{E}_{[t+1]}$ column-wise by $\mathbf{e}_{j[t+1]} \leftarrow \mathcal{S}_{\theta_2\mu^{-1}}[\|\mathbf{m}_{j[t]}\|_2]\frac{\mathbf{m}_{j[t]}}{\|\mathbf{m}_{j[t]}\|_2}$.
7: Update the Lagrange multipliers by $\mathbf{\Lambda}_{1[t+1]} \leftarrow \mathbf{\Lambda}_{1[t]} + \mu(\mathbf{Y} - \mathbf{A}\mathbf{Z}_{[t+1]} - \mathbf{E}_{[t+1]})$. $\mathbf{\Lambda}_{2[t+1]} \leftarrow \mathbf{\Lambda}_{2[t]} + \mu(\mathbf{Z}_{[t+1]} - \mathbf{J}_{[t+1]})$. $\mathbf{\Lambda}_{3[t+1]} \leftarrow \mathbf{\Lambda}_{3[t]} + \mu(\mathbf{J}_{[t+1]} - \mathbf{W}_{[t+1]})$.
8: Update μ by $\mu \leftarrow \min(\rho\cdot\mu, 10^6)$.
9: Check convergence conditions $\|\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E}\|_\infty < \epsilon$, $\|\mathbf{Z} - \mathbf{J}\|_\infty < \epsilon$, $\|\mathbf{J} - \mathbf{W}\|_\infty < \epsilon$.
10: $t \leftarrow t + 1$.
11: **end while**

Special cases of the JSLRR are the JSR and the robust low-rank representation (LRR). In particular, a robust JSR matrix is found by solving the convex optimization problem (16) in the presence of noise:

$$\textbf{Robust JSR}: \ \underset{\mathbf{Z},\mathbf{E}}{\operatorname{argmin}} \ \|\mathbf{Z}\|_1 + \theta_2\|\mathbf{E}\|_{2,1} \quad \text{subject to } \mathbf{Y} = \mathbf{A}\,\mathbf{Z} + \mathbf{E}. \tag{16}$$

(16) is known as the sparse multiple measurement vector (MMV) [28], which is an extension of the sparse single measurement vector (SMV) in (3). The key difference between the MMV and the SMV is that the correlations between the test samples are taken into account, as well as the test samples from a specific class are simultaneously represented by few columns of the training matrix.

A robust LRR is obtained as the solution of the following convex optimization problem:

$$\textbf{Robust LRR}: \ \underset{\mathbf{Z},\mathbf{E}}{\operatorname{argmin}} \ \|\mathbf{Z}\|_* + \theta_2\|\mathbf{E}\|_{2,1} \quad \text{subject to } \mathbf{Y} = \mathbf{A}\,\mathbf{Z} + \mathbf{E}. \tag{17}$$

Actually, (16) and (17) are ℓ_2/ℓ_1 norm regularized versions of (3) and (4) in order to account for the noise.

Similarly to the derivation of the JSLRR, the optimization problem (16) is solved by minimizing the augmented Lagrangian function:

$$\begin{aligned} &\mathcal{L}_1(\mathbf{Z},\mathbf{J},\mathbf{E},\mathbf{\Lambda}_1,\mathbf{\Lambda}_2) = \|\mathbf{J}\|_1 + \theta_2\|\mathbf{E}\|_{2,1} \\ &+ \ \operatorname{tr}\left(\mathbf{\Lambda}_1^T(\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E})\right) + \operatorname{tr}\left(\mathbf{\Lambda}_2^T(\mathbf{Z} - \mathbf{J})\right) \\ &+ \ \frac{\mu}{2}\left(\|\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E}\|_F^2 + \|\mathbf{Z} - \mathbf{J}\|_F^2\right), \end{aligned} \tag{18}$$

where $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2$ are the Lagrange multipliers and $\mu > 0$ is a penalty parameter. (18) can be minimized by employing the ADALM method [49],s as is outlined in Algorithm 2.

Algorithm 2 Solving (16) by the ADALM method.

Input: Training matrix $\mathbf{A} \in \mathbb{R}^{d\times N}$, test matrix $\mathbf{Y} \in \mathbb{R}^{d\times M}$ and the parameter θ_2.
Output: Matrix $\mathbf{Z} \in \mathbb{R}^{N\times M}$ and matrix $\mathbf{E} \in \mathbb{R}^{d\times M}$.

1: Initialize: $\mathbf{Z}_{[0]} = \mathbf{J}_{[0]} = \mathbf{0}$, $\mathbf{\Lambda}_{1[0]} = \mathbf{0}$, $\mathbf{\Lambda}_{2[0]} = \mathbf{0}$, $\mu = 10^{-6}$, $\rho = 1.1$, $\epsilon = 10^{-8}$.
2: **while** not converged **do**
3: Fix $\mathbf{Z}_{[t]}$ and $\mathbf{E}_{[t]}$, and update $\mathbf{J}_{[t+1]}$ by $\mathbf{J}_{[t+1]} \leftarrow \mathcal{S}_{\mu^{-1}}[\mathbf{Z}_{[t]} + \mathbf{\Lambda}_{2[t]}/\mu]$.
4: Fix $\mathbf{J}_{[t+1]}$, and $\mathbf{E}_{[t]}$, and update $\mathbf{Z}_{[t+1]}$ by $\mathbf{Z}_{[t+1]} \leftarrow \left(\mathbf{I} + \mathbf{A}^T\mathbf{A}\right)^{-1}(\mathbf{A}^T(\mathbf{Y} - \mathbf{E}_{[t]}) + \mathbf{J}_{[t+1]} + (\mathbf{A}^T\mathbf{\Lambda}_{1[t]} - \mathbf{\Lambda}_{2[t]})/\mu)$.
5: Fix $\mathbf{J}_{[t+1]}$, $\mathbf{Z}_{[t+1]}$, form $\mathbf{M}_{[t]} = \mathbf{Y} - \mathbf{A}\mathbf{Z}_{[t+1]} + \mathbf{\Lambda}_{1[t]}/\mu$, and update $\mathbf{E}_{[t+1]}$ column-wise by $\mathbf{e}_{j[t+1]} \leftarrow \mathcal{S}_{\theta_2\mu^{-1}}[\|\mathbf{m}_{j[t]}\|_2]\frac{\mathbf{m}_{j[t]}}{\|\mathbf{m}_{j[t]}\|_2}$.
6: Update the Lagrange multipliers by $\mathbf{\Lambda}_{1[t+1]} \leftarrow \mathbf{\Lambda}_{1[t]} + \mu(\mathbf{Y} - \mathbf{A}\mathbf{Z}_{[t+1]} - \mathbf{E}_{[t+1]})$. $\mathbf{\Lambda}_{2[t+1]} \leftarrow \mathbf{\Lambda}_{2[t]} + \mu(\mathbf{Z}_{[t+1]} - \mathbf{J}_{[t+1]})$.
7: Update μ by $\mu \leftarrow \min(\rho\cdot\mu, 10^6)$.
8: Check convergence conditions $\|\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E}\|_\infty < \epsilon$ and $\|\mathbf{Z} - \mathbf{J}\|_\infty < \epsilon$.
9: $t \leftarrow t + 1$.
10: **end while**

Problem (17), can be solved by minimizing the augmented Lagrangian function:

$$\begin{aligned} &\mathcal{L}_2(\mathbf{Z},\mathbf{J},\mathbf{E},\mathbf{\Lambda}_1,\mathbf{\Lambda}_2) = \|\mathbf{J}\|_* + \theta_2\|\mathbf{E}\|_{2,1} \\ &+ \ \operatorname{tr}\left(\mathbf{\Lambda}_1^T(\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E})\right) + \operatorname{tr}\left(\mathbf{\Lambda}_2^T(\mathbf{Z} - \mathbf{J})\right) \\ &+ \ \frac{\mu}{2}\left(\|\mathbf{Y} - \mathbf{A}\mathbf{Z} - \mathbf{E}\|_F^2 + \|\mathbf{Z} - \mathbf{J}\|_F^2\right). \end{aligned} \tag{19}$$

The minimization of (19) can be obtained, following a similar procedure to that described in Algorithm 2. The only difference is that the Step 3 in Algorithm 2 should be replaced by [19]:

$$\begin{aligned} \mathbf{J}_{[t+1]} &= \underset{\mathbf{J}_{[t]}}{\operatorname{argmin}} \ \frac{1}{\mu}\|\mathbf{J}_{[t]}\|_* + \frac{1}{2}\|\mathbf{J}_{[t]} - (\mathbf{Z}_{[t]} + \mathbf{\Lambda}_{2[t]}/\mu)\|_F^2 \\ &= \mathcal{D}_{\mu^{-1}}[\mathbf{Z}_{[t]} + \mathbf{\Lambda}_{2[t]}/\mu]. \end{aligned} \tag{20}$$

Example: For illustration, 4 linear pairwise independent subspaces are constructed, whose basis $\{\mathbf{U}_i\}_{i=1}^4$ are computed by $\mathbf{U}_{i+1} = \mathbf{R}\mathbf{U}_i$, $i = 1, 2, 3$. $\mathbf{U}_1 \in \mathbb{R}^{600\times 110}$ is a column

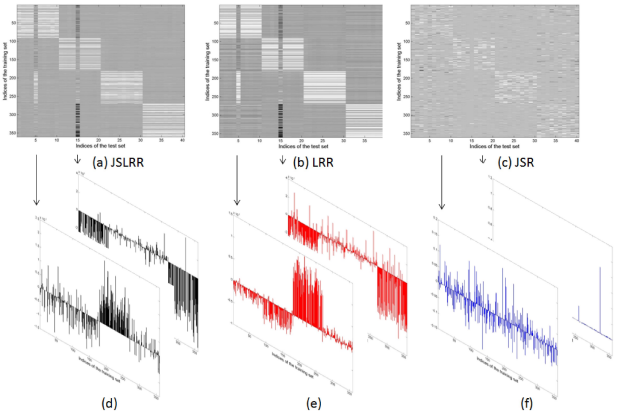

Fig. 2. Representation matrices in Example. (a) The JSLRR obtained by solving (5). (b) The LRR obtained by solving (17). (c) The JSR obtained by solving (16). Representation coefficients of the 5th and the 15th test sample obtained by (d) the JSLRR, (e) the LRR, and (f) the JSR.

orthonormal random matrix and $\mathbf{R} \in \mathbb{R}^{600\times 600}$ is a random rotation matrix. The data matrix $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4] \in \mathbb{R}^{600\times 400}$ is obtained by picking 100 samples from each subspace. That is, $\mathbf{X}_i \in \mathbb{R}^{600\times 100}$, $i = 1, 2, 3, 4$. Next, the data matrix is partitioned into the training matrix $\mathbf{A} \in \mathbb{R}^{600\times 360}$ and the test matrix $\mathbf{Y} \in \mathbb{R}^{600\times 40}$ by employing a 10-fold cross validation. The matrices $\mathbf{A}$ and $\mathbf{Y}$ have the structure described at the beginning of subsection IV-A. Next, we pick randomly 50 columns of $\mathbf{A}$ and we replace them by a linear combination of randomly chosen vectors from the two subspaces with random weights. Thus, the training set is now contaminated by outliers. The 5th column of the test matrix $\mathbf{Y}$ is replaced by a linear combination of vectors not drawn from any of the 4 subspaces and the 15th column of $\mathbf{Y}$ is replaced by a vector drawn from the 1st and the 4th subspace, as previously said. In the first row of Fig. 2, the representation matrices are depicted that are obtained by solving (5), (17), and (16), respectively. The second row in Fig. 2 depicts the representation coefficients of the 5th and the 15th test sample obtained by the JSLRR, the LRR, and the JSR, respectively. The inspection of Fig. 2 reveals that the JSLRR admits a structure closer to a block-diagonal one compared to that of the LRR and the JSR. Furthermore, the vector of representation coefficients for the 5th test sample is dense for the JSLRR (Fig. 2 (d)), the LRR (Fig. 2 (e)), and the JSR (Fig. 2 (f)). Thus, it does not provide information about the subspace the 5th test sample lies onto. For the 15th sample, the JSLRR is able to capture that this sample lies onto the intersection of two subspaces by providing two blocks of representation coefficients. In addition, there are some zero coefficients inside these coefficient blocks. This fact can be interpreted as correction, meaning that the outliers of the training set are not used to represent this test sample. This is not the case for the LRR and the JSR.

C. Joint sparse low-rank representation-based classifier

Having found the JSLRR matrix $\mathbf{Z} \in \mathbb{R}^{N\times M}$ and the noise matrix $\mathbf{E} \in \mathbb{R}^{d\times M}$, the mth test sample $\mathbf{y}_m \in \mathbb{R}^d$ is classified as follows. First, noise correction is enforced to $\mathbf{y}_m$ by subtracting $\mathbf{e}_m$ yielding $\bar{\mathbf{y}}_m = \mathbf{y}_m - \mathbf{e}_m$. Ideally, the mth column of $\mathbf{Z}$ (i.e., $\mathbf{z}_m \in \mathbb{R}^N$) contains non-zero entries in the positions associated with the columns of the training matrix $\mathbf{A}$ spanned from a single subspace, corresponding thus to a single class. Consequently, we can easily assign $\bar{\mathbf{y}}_m$ to that class. However, in practice, there are small non-zero entries in $\mathbf{z}_m$ that are associated to multiple subspaces (i.e., classes). To cope with this problem, each noise corrected test sample $\bar{\mathbf{y}}_m$ is classified to the class that minimizes the ℓ_2 squared norm residual between $\bar{\mathbf{y}}_m$ and $\hat{\mathbf{y}}_k = \mathbf{A}\,\delta_k(\mathbf{z}_m)$ divided by the squared ℓ_2 norm of $\delta_k(\mathbf{z}_m)$, where $\delta_k(\mathbf{z}_m) \in \mathbb{R}^N$ is a new vector whose nonzero entries are the entries in $\mathbf{z}_m$ that are associated to the kth class only. The procedure is outlined in Algorithm 3.

Algorithm 3 Joint Sparse Low-Rank Representation-based Classifier.

Input: Training matrix $\mathbf{A} \in \mathbb{R}^{d\times N}$ and test matrix $\mathbf{Y} \in \mathbb{R}^{d\times M}$.
Output: A class label for each column of $\mathbf{Y}$.

1: Solve (7) by employing Algorithm 1 and obtain $\mathbf{Z} \in \mathbb{R}^{N\times M}$ and $\mathbf{E} \in \mathbb{R}^{d\times M}$.
2: **for** $m = 1$ to M **do**
3: $\quad\bar{\mathbf{y}}_m = \mathbf{y}_m - \mathbf{e}_m$.
4: $\quad$**for** $k = 1$ to K **do**
5: $\quad\quad$Compute the residuals $r_k(\bar{\mathbf{y}}_m) = \|\bar{\mathbf{y}}_m - \mathbf{A}\,\delta_k(\mathbf{z}_m)\|_2^2 / \|\delta_k(\mathbf{z}_m)\|_2^2$.
6: $\quad$**end for**
7: $\quad$class$(\bar{\mathbf{y}}_m) = \text{argmin}_k\, r_k(\bar{\mathbf{y}}_m)$.
8: **end for**

The same procedure is applicable to the JSR and the LRR of the test set with respect to the training set, yielding the JSR-based and the LRR-based classifier, respectively.

The JSR-based classifier differs from the SRC [20] in that the test samples, which belong to the same class, have the same sparse support of the coefficient vectors. In other words, all the test samples drawn from a certain class are spanned by the same, few, training samples that share the same sparse pattern. Furthermore, noise correction is enforced by the JSR-based classifier. The same difference holds between the classifiers that are based on the robust low-rank representation and the standard low-rank one, e.g., [22].

V. Experimental Evaluation

A. Datasets and evaluation procedure

The performance of the three proposed classifiers based on the JSRR, the JSR, and the LRR in music genre classification is assessed by conducting experiments on 6 manually annotated benchmark datasets for which the audio files are publicly available. In particular, the GTZAN [13], the ISMIR dataset, the Ballroom, the Homburg [52], the 1517Artists [12], and the Unique [12] datasets are employed. These datasets are briefly described next.

The **GTZAN**[2] consists of 1000 excerpts, 30 sec long, equally distributed over 10 genre classes, namely blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock.

The **ISMIR**[3] contains 1458 full music recordings distributed over 6 genre classes as follows: classical (640), electronic (229), jazz-blues (52), metal-punk (90), rock-pop (203), world (244), where the number within parentheses refers to the number of recordings, which belong to each genre class. Therefore, the 43.9% of the recordings belong to the classical genre.

The **Ballroom**[4] dataset consists of 698 music excerpts, 30 sec long, from 8 different dance music styles namely cha-cha (111), jive (60), quickstep (82), rumba (98), samba (86), tango (86), viennese waltz (65), and slow waltz (110). Again, the number within parentheses refers to the number of recordings, which belong to each dance style.

The **Homburg** dataset[5] contains 1886 music excerpts, 10 sec long, by 1463 different artists. These excerpts are unequally distributed over 9 genres, namely alternative, blues, electronic, folk-country, funk/soul/rnb, jazz, pop, rap/hip-hop, rock. The largest class is the rap/hip-hop genre, containing 26.72% of the music excerpts, while the funk/soul/rnb is the smallest one, containing 2.49% of the excerpts.

The **1517Artists**[6] consists of 3180 full-length music recordings from 1517 different artists, downloaded from download.com. The 190 most popular songs, according to the number of total listenings, were selected for each of the 19 genres, i.e., alternative/punk, blues, children's, classical, comedy/spoken, country, easy listening/vocal, electronic, folk, hip-hop, jazz, latin, new age, rnb/soul, reggae, religious, rock/pop, soundtracks, world. In this dataset, the music recordings are distributed almost uniformly over the genre classes.

The **Unique**[7] consists of 3115 music excerpts of popular and well-known songs, distributed over 14 genres, namely blues, classic, country, dance, electronica, hip-hop, jazz, reggae, rock, schlager (i.e., music hits) soul/rnb/, folk, world, and spoken. Each excerpt has 30 sec duration. The class distribution is skewed. That is, the smallest class (i.e., spoken music) accounts for 0.83% and the largest class (i.e., classic) for 24.59% of the excerpts.

Two music genre classification experiments were conducted. In the first set of experiments, the performance of the proposed classifiers is compared with that of the state-of-the-art music genre classification methods by applying the standard evaluation protocol for each dataset. In particular, following [6], [12], [13], [16], [52]–[55] stratified 10-fold cross validation was applied to the GTZAN dataset, the Ballroom, the Homburg, the 1517Artists, and the Unique datasets. The experiments on the ISMIR 2004 Genre dataset were conducted according to the ISMIR2004 Audio Description Contest protocol, which defines training and evaluation sets, consisting of 729 audio files each. In content-based music classification, it is well-known that that recordings from the same artist or the same album are easily classified correctly, biasing the reported experimental results. This is attributed to the very specific artistic style and recording conditions. It is referred to as artist and album effect, respectively [56]. To prevent any artist or album effects, artist filtering has been applied to the Homburg, the 1517-Artists, and the Unique datasets, where artist information is available.

In practice, the number of annotated music recordings per genre class is often limited [3]. Therefore, a major challenge is to train the music genre classifiers for large-scale data sets from few labeled data [3]. In the second set of experiments, the performance of the proposed music genre classification framework is investigated in the just metioned challenge. Only 10% of the available recordings were exploited for training and the remaining 90% for testing, in all datasets. The experiments were repeated 10 times.

The 3 proposed classifiers are compared with another four well-known classifiers, namely the SRC [20], the LRC [23], the SVM[8] with a linear kernel, and the NN classifier with the cosine distance metric, by applying the aforementioned protocols. Since the dimensionality of the cortical representations is much larger than the cardinality of the training set, the sparse coding in the SRC[9] is obtained by the LASSO [57]. The LRC is a nearest subspace classification method, where a dense coefficient vector, (i.e., $\mathbf{z}_m$) is obtained by minimizing the ℓ_2-norm residual between the test sample $\mathbf{y}_m$ and $\mathbf{A}\mathbf{z}_m$. Next, the class label is assigned in favor of the class with the minimum reconstruction error. Due to the assumed subspace structure of the audio features, both the linear SVM, the SRC, and the NN are appropriate for separating features from various music recordings that belong to different genres. Furthermore, the aforementioned baseline classifiers are working in the same feature space with the proposed classifiers, which makes the performance comparisons fair. The performance of each classifier is assessed by reporting the music genre classification accuracy. In all the experimenters, the parameters (i.e., θ_1, θ_2) of the proposed classifiers are set as specified in subsection IV-B, namely $\theta_1 = \sqrt{r/q}$ and $\theta_2 = \frac{3}{7\sqrt{\gamma \cdot M}}$. In particular, the rank of the representation matrix is estimated as $r = M - \gamma M$, where γM denotes the number of outliers in the test set. The level of sparsity of the representation is set as $q = 3N$. The only parameter that needs tuning is the portion of outliers in the test set (i.e., γ). To this end, by employing the method in [58], (i.e., by excluding from the training set a subset, the evaluation set, and tuning the parameters in the evaluation set), it has been found that $\gamma = 0.01$ is a reasonable value for all the datasets.

B. Experimental Results

Table II, summarizes the music genre classification accuracies for the 6 datasets. These results have been obtained

[2]http://marsyas.info/download/data_sets

[3]http://ismir2004.ismir.net/ISMIR_Contest.html

[4]http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html

[5]http://www-ai.cs.uni-dortmund.de/audio.html

[6]http://www.seyerlehner.info/index.php?p=1_3_Download

[7]http://www.seyerlehner.info/index.php?p=1_3_Download

[8]The LIBSVM was used in the experiments (http://www.csie.ntu.edu.tw/~cjlin/libsvm/)

[9]The SPGL1 Matlab solver was used in the implementation of the SRC (http://www.cs.ubc.ca/~mpf/spgl1/).

by applying the standard protocol defined for each dataset. The numbers within the parentheses indicate the standard deviations obtained by a stratified 10-fold cross-validation on all datasets, but the ISMIR. Each classifier is applied to the auditory cortical representations (cortical features), the 420-dimensional MFCCs, the 156-dimensional chroma features, the fusion of cortical features and MFCCs (cm) and the fusion of all the aforementioned features (cmc). The fusion of features is obtained by constructing an augmented feature vector by stacking the cortical features on the top of the MFCCs and the chroma features. In the last rows of Table II, the figures of merit for the top performing music classification methods are included, for comparison purposes.

By inspecting Table II, the best music genre classification accuracy has been obtained by the proposed classifiers in 5 out of 6 datasets. In particular, the JSLRR-based classifier achieves the top classification accuracy in 4 out of 6 datasets, when either the cortical or the fusion of all the features has been exploited for music representation. Comparable performance has been achieved by fusing the cortical features and the MFCCs. In the ISMIR dataset, the best classification accuracy is achieved by the JSR-based classifier, when the combination of the cortical features and the MFCCs has been employed. This is not the case for the Unique dataset, where the SVMs achieve the best classification accuracy, when fusing the cortical features and the MFCCs. The JSLRR-based classifier outperforms all the classifiers being compared to in the GTZAN, the Ballroom, the Homburg, and the 1517-Artists datasets. When the cortical features as well as their combination with the MFCCs and the chroma features are exploited for music representation, the JSR-based classifier outperforms the SRC in all datasets. This can be attributed to the noise correction enforced by the JSR-based classifier. The MFCCs are classified more accurately by the JSR in the GTZAN, the ISMIR, and the 1517Artists datasets, the LRR-based classifier in the Homburg dataset, and the linear SVM in the Ballroom and the Unique datasets. The chroma features are classified more accurately by the linear SVM in the Ballroom, the Homburg, the 1517Artists, and the Unique datasets, while the SRC is proved to be more efficient in the GTZAN and the ISMIR datasets. It can be observed from Table II that the LRC performs poor in many cases. This is attributed to the fact that the training matrix (i.e., $\mathbf{A}$) is often rank-deficient and thus there are infinitely many solutions to the underlying least squares problem solved by the LRC.

The best classification accuracy obtained by the JSLRR-based classifier on all, but the Ballroom dataset, ranks high compared to that obtained by the majority of music genre classification techniques, listed in last rows of Table II. In particular, for the Homburg, the 1517-Artists, and the Unique datasets, the best accuracy is achieved by the JSLRR-based classifier. Regarding the GTZAN and ISMIR datasets, it is worth mentioning that the results in [16], have been obtained by applying feature aggregation on the combination of 4 elaborated audio features. The reported results on the Ballroom dataset are significantly inferior to those obtained by the methods in [53], [60], [61]. This is attributed to the fact that special-purpose rhythmic features have been employed for dance style discrimination there.

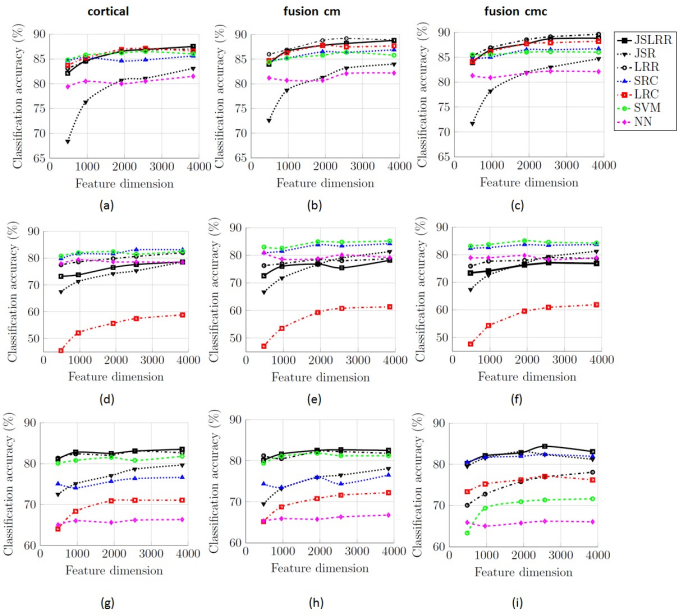

Fig. 3. Classification accuracy of the various classifiers as a function of feature dimensionality for cortical (a),(d),(g), cm (b),(e),(h), and cmc (c),(f),(i). First row, results on the GTZAN dataset. Second row, results on the ISMIR dataset. Third row, results on the Ballroom dataset.

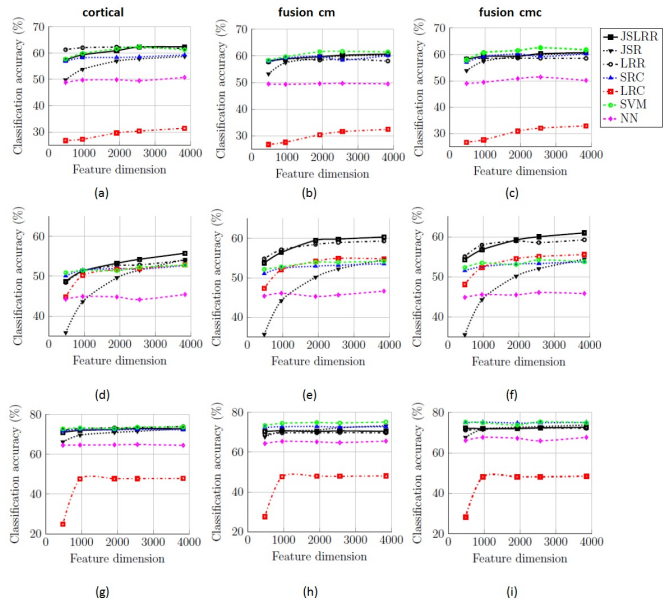

Fig. 4. Classification accuracy of the various classifiers as a function of feature dimensionality for cortical (a),(d),(g), cm (b),(e),(h), and cmc (c),(f),(i). First row, results on the Homburg dataset. Second row, results on the 1517Artists dataset. Third row, resulys on the Unique dataset.

To investigate how the classification accuracy is affected by the feature vector dimension, dimensionality reduction of the cortical and the augmented feature vectors via random projections [65] is considered. Random projections are computational inexpensive, providing with high probability a *stable embedding* [65]. Roughly speaking, a stable embedding approximately preserves the Euclidean distances between all

TABLE II
MUSIC GENRE CLASSIFICATION ACCURACY FOR VARIOUS DATASETS. THE NUMBERS WITHIN THE PARENTHESES INDICATE THE STANDARD DEVIATIONS OBTAINED BY 10-FOLD CROSS-VALIDATION. THE BEST RESULTS ARE INDICATED IN BOLDFACE.

Classifier	**Features**	GTZAN	ISMIR	Ballroom	Homburg	1517Artists	Unique
JSLRR	cmc	**89.40** (2.87)	80.52	**82.37** (3.96)	59.70 (2.07)	**61.85** (2.90)	70.36 (1.87)
	cm	88.70 (2.79)	79.28	82.22 (4.87)	59.54 (2.11)	61.63 (2.63)	70.33 (2.01)
	cortical	89.20 (2.48)	81.20	81.93 (4.86)	**63.46** (2.49)	60.18 (2.63)	73.57 (0.93)
	MFCCs	72.20 (3.35)	63.92	47.27 (5.91)	51.48 (2.34)	35.75 (2.35)	59.26 (1.98)
	chroma	29.30 (4.80)	49.93	24.07 (3.93)	35.05 (2.49)	14.77 (1.50)	42.66 (2.20)
JSR	cmc	87.10 (3.24)	85.32	79.36 (3.79)	62.08 (2.52)	58.61 (2.30)	73.99 (1.60)
	cm	87.00 (3.33)	**85.45**	79.79 (3.82)	62.08 (2.37)	58.08 (2.40)	73.86 (1.69)
	cortical	86.80 (3.01)	84.49	80.36 (3.95)	61.02 (2.90)	57.45 (2.21)	74.28 (1.41)
	MFCCs	76.90 (4.45)	79.01	43.27 (4.57)	48.83 (3.67)	42.13 (2.72)	62.47 (2.58)
	chroma	34.80 (3.88)	50.89	24.35 (4.91)	24.92 (2.54)	11.63 (0.34)	41.95 (2.67)
LRR	cmc	89.10 (3.31)	81.89	81.21 (4.17)	57.31 (2.52)	59.81 (2.85)	70.11 (1.71)
	cm	88.90 (3.28)	81.75	81.93 (3.55)	57.36 (2.70)	59.84 (2.96)	69.72 (1.80)
	cortical	87.80 (2.82)	81.20	80.93 (3.13)	60.33 (3.67)	57.92 (2.65)	74.05 (1.22)
	MFCCs	71.90 (5.21)	62.27	49.28 (4.97)	52.75 (2.26)	37.45 (2.61)	57.56 (1.04)
	chroma	30.80 (4.51)	49.93	24.64 (3.66)	32.82 (1.43)	14.77 (1.29)	45.07 (2.47)
SRC	cmc	87.00 (3.12)	84.49	77.22 (4.29)	60.17 (3.43)	53.67 (2.26)	73.57 (1.88)
	cm	87.20 (2.97)	84.77	76.93 (4.04)	60.70 (3.25)	53.49 (2.32)	73.38 (2.03)
	cortical	86.50 (2.46)	84.36	77.65 (3.44)	59.06 (2.81)	50.72 (2.61)	67.48 (1.14)
	MFCCs	75.50 (4.22)	74.75	44.26 (7.58)	49.68 (3.33)	41.19 (2.59)	63.69 (1.76)
	chroma	41.60 (3.77)	56.37	28.95 (3.75)	36.05 (2.83)	21.69 (1.31)	49.08 (2.72)
LRC	cmc	87.90 (3.14)	63.23	72.20 (4.93)	34.09 (2.23)	56.16 (2.27)	48.15 (0.60)
	cm	87.90 (3.07)	63.37	72.78 (5.29)	33.40 (1.74)	55.50 (2.43)	48.08 (0.56)
	cortical	87.30 (3.05)	60.76	71.91 (4.64)	32.18 (1.74)	53.11 (2.65)	47.89 (0.35)
	MFCCs	70.00 (5.77)	43.89	36.25 (3.85)	22.16 (3.57)	22.76 (2.98)	24.75 (2.70)
	chroma	11.90 (1.85)	29.35	16.03 (3.30)	16.43 (3.59)	8.17 (1.00)	18.58 (2.14)
SVM	cmc	86.30 (2.35)	82.99	81.66 (5.75)	62.88 (2.52)	54.24 (3.52)	74.89 (1.84)
	cm	86.60 (2.59)	83.26	81.95 (5.75)	62.77 (2.34)	54.43 (3.68)	**75.05** (2.02)
	cortical	86.10 (2.42)	82.44	80.65 (5.40)	62.40 (3.19)	53.71 (3.18)	68.89 (2.22)
	MFCCs	74.00 (2.40)	74.89	51.71 (4.67)	50.90 (3.53)	38.86 (1.99)	67.15 (2.26)
	chroma	37.20 (4.84)	52.53	29.35 (4.31)	37.43 (2.54)	18.23 (2.27)	49.34 (3.17)
NN	cmc	82.50 (3.74)	79.69	66.18 (5.89)	50.68 (4.54)	46.22 (2.85)	65.52 (2.46)
	cm	82.10 (3.28)	78.87	66.90 (5.62)	50.57 (4.45)	46.41 (2.49)	65.23 (2.31)
	cortical	81.30 (2.79)	78.60	67.90 (3.91)	49.94 (4.27)	44.84 (2.55)	64.43 (2.57)
	MFCCs	66.80 (4.56)	70.64	34.67 (5.97)	29.79 (3.13)	33.45 (2.00)	55.24 (2.43)
	chroma	37.90 (4.60)	51.30	24.07 (3.81)	27.73 (2.06)	15.09 (1.80)	38.07 (3.33)
		[16] 90.60	[16] 86.83	[53] 96.00	[59] 62.40	[59] 54.91	[59] 72.90
		[59] 87.00	[10] 83.50	[60] 90.40	[61] 61.20	[61] 41.10	[12] 72.00
		[6] 84.30	[6] 83.15	[61] 90.00	[62] 57.81	[63] 35.00	
		[61] 82.00	[59] 82.99	[54] 67.60	[63] 55.30		
		[55] 77.20	[64] 82.30	[53] 50.00	[52] 53.23		

samples of the original space in the space of reduced dimensions. Thus, the subspace structures of the original data space are also maintained into the space of reduced dimensions. It is clear from subsection IV-B that such a property is crucial for the proposed classifiers. The feature vectors of reduced dimensions are obtained by applying a random projection matrix, drawn from a zero-mean normal distribution, onto the original feature vectors. The dimensionality of the low-dimensional feature space is equal to $1/16$, $1/8$, $1/4$, $1/3$, and $1/2$ of the original feature space. In Figs. 3 and 4 classification accuracies obtained by various classifiers as a function of the dimensionality of cortical, cm, and cmc features is plotted for the 6 datasets. From the inspection of Figs. 3 and 4, it is seen that the dimensionally reduction via random projections degrades slightly the classification accuracy of all classifiers under study in all, but the Ballroom dataset.

Apart from classification accuracy, the computational cost is also of concern in practice. In Table III, the average running time of each classifier needed for training and classification per music recording is listed. These times have been computed by averaging the needed time for training and classification in all datasets and then by dividing the average time with the total number of test recordings. The JSLRR, the JSR, the LRR, and the LRC were implemented in Matlab version 2009b. For the experiments, a desktop PC with Intel Core Duo at 3.16 GHz CPU and 4 GB of RAM has been employed. The LRR-based classifier is obviously the fastest among the proposed classifiers. The JSR-based classifier is less time consuming than the JSLRR. Since in the implementation of the SRC or the SVM, C code has been employed (i.e., .mex files), their running times cannot be compared fairly with that of the proposed classifiers. It is worth mentioning that all the classifiers run significantly faster, when features of reduced dimensions are fed into them.

In Table IV, music genre classification results in the small sample size setting are summarized. These results have been obtained by employing the fusion of the cortical features, the MFCC and, the chroma features. The best classification results are obtained by the LRR-based classifier in 3 out of 6 datasets. In the GTZAN and the Homburg dataset the performance of the JSLRR-based classifier is slightly inferior to that of the LRR-based classifier. The best classification results in the ISMIR, the Ballroom, and the Unique datasets have been achieved by the linear SVM. It is worth noting the

TABLE III
AVERAGE RUNNING TIME IN CPU SECONDS PER RECORDING OF COMPETING CLASSIFIERS IN MUSIC GENRE CLASSIFICATION. CPU TIME IS SPLIT INTO 2 CELLS. THE TIME IN THE LEFT CELL IS OBTAINED BY EMPLOYING FEATURES WITHOUT DIMENSIONALITY REDUCTION, WHILE THE TIME IN THE RIGHT CELL IS OBTAINED BY EMPLOYING FEATURES WHOSE DIMENSIONALITY HAS BEEN REDUCED BY A FACTOR OF $1/2$ VIA RANDOM PROJECTIONS.

Features	JSLRR		JSR		LRR	
cmc	2.783	1.8662	2.4729	1.4775	1.6408	1.0304
cm	2.7916	1.8758	2.5154	1.4819	1.596	1.0202
cortical	2.7787	1.869	2.4843	1.4717	1.5731	1.0196
MFCCS	0.33088		0.06786		0.10764	
chroma	0.34195		0.075816		0.10748	
	SRC		LRC		SVM	
cmc	0.78172	0.51995	1.9483	1.6482	0.074568	0.037752
cm	0.76877	0.51855	1.8747	1.6154	0.077064	0.033384
cortical	0.7733	0.51808	1.8995	1.6354	0.11747	0.033696
MFCCS	0.021372		0.00546		0.00078	
chroma	0.019968		0.002028		0.000468	

LRR-based classifier and the linear SVM perform equally well in the Ballroom dataset. Given the relatively small number of training music recordings, the results in Table IV are acceptable, indicating that the LRR- and the JSLRR-based classifiers can be exploited for music genre classification in real world conditions.

TABLE IV
MUSIC GENRE CLASSIFICATION ACCURACY ON VARIOUS DATASETS BY EMPLOYING A FEW LABELED MUSIC RECORDINGS. THE NUMBERS WITHIN THE PARENTHESES INDICATE THE STANDARD DEVIATIONS.

Classifier	GTZAN	ISMIR	Ballroom
JSLRR	71.77 (1.92)	71.78 (1.34)	61.50 (2.18)
JSR	65.48 (2.21)	72.65 (1.34)	56.16 (2.66)
LRR	**73.52** (1.35)	71.91 (1.85)	62.49 (2.75)
SRC	71.00 (2.25)	74.37 (1.45)	59.04 (3.64)
LRC	72.72 (1.28)	69.47 (1.10)	59.29 (3.23)
SVM	72.03 (1.54)	**76.19** (1.25)	**62.52** (3.42)
NN	63.38 (3.60)	71.49 (1.48)	50.79 (4.74)
Classifier	Homburg	1517Artists	Unique
JSLRR	56.03 (0.72)	37.11 (0.77)	67.48 (0.77)
JSR	54.18 (1.10)	32.92 (1.65)	67.69 (0.71)
LRR	**56.10** (0.64)	**42.86** (1.08)	68.87 (0.49)
SRC	54.50 (0.74)	37.02 (0.90)	68.63 (1.05)
LRC	52.68 (1.13)	38.58 (0.78)	58.82 (0.66)
SVM	56.06 (0.73)	38.35 (1.06)	**70.04** (0.72)
NN	45.96 (1.39)	30.48 (1.16)	60.58 (1.45)

VI. CONCLUSIONS

The JSLRR has been proposed as an alternative to the sparse representation and the low-rank one in order to correct the noise and identify the subspace structures in data contaminated by outliers. Three general purpose classifiers, robust to noise, have been developed thanks to the JSLRR and tested for music genre classification. The experimental results indicate the strengths of the JSLRR in music genre classification and validate that the cortical representations are more discriminating features compared to the conventional audio features (MFCCs or chroma) for music genre classification.

REFERENCES

[1] J. J. Aucouturier and F. Pachet, "Representing musical genre: A state of the art," *Journal of New Music Research*, pp. 83–93, 2003.

[2] T. Bertin-Mahieux, D. Eck, and M. Mandel, "Automatic tagging of audio: The state-of-the-art," in *Machine Audition: Principles, Algorithms and Systems*, W. Wang, Ed. IGI Publishing, 2010.

[3] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 303–319, 2011.

[4] N. Scaringella, G. Zoia, and D. Mlynek, "Automatic genre classification of music content: A survey," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 133–141, March 2006.

[5] B. L. Sturm, "A survey of evaluation in music genre recognition," in *Proc. Int. Workshop on Adaptive Multimedia Retrieval*, 2012.

[6] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification," *IEEE Trans. Audio, Speech, and Language Technology*, vol. 18, no. 3, pp. 576–588, 2010.

[7] J. Bergstra, M. Mandel, and D. Eck, "Scalable genre and tag prediction with spectral covariance," in *Proc. 11th Int. Symp. Music Information Retrieval*, 2010, pp. 507–512.

[8] L. Chen, P. Wright, and W. Nejdl, "Improving music genre classification using collaborative tagging data," in *Proc. ACM 2nd Int. Conf. Web Search and Data Mining*, 2009, pp. 84–93.

[9] K. Chang, J. S. R. Jang, and C. S. Iliopoulos, "Music genre classification via compressive sampling," in *Proc. 11th Int. Conf. Music Information Retrieval*, pp. 387–392.

[10] A. Holzapfel and Y. Stylianou, "Musical genre classification using nonnegative matrix factorization-based features," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 424–434, February 2008.

[11] H. Lukashevich, J. Abeber, C. Dittmar, and H. Grossmann, "From multi-labeling to multi-domain-laneling: A novel two-dimensional approach to music genre classification," in *Proc. 10th Int. Conf. Music Information Retrieval*, 2009, pp. 459–464.

[12] K. Seyerlehner, M. Schedl, T. Pohle, and P. Knees, "Using block-level features for genre classification, tag classification and music similarity estimation," in *Proc. 11th Int. Symp. Music Information Retrieval*, 2010, MIREX.

[13] 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.

[14] C. Zhen and J. Xu, "Multi-modal music genre classification approach," in *Proc. 3rd IEEE Int. Conf. Computer Science and Information Technology*, 2010, pp. 398–402.

[15] D. Garcia-Garcia, J. Arenas-Garcia, E. Parrado-Hernandez, and F. Diaz-de Maria, "Music genre classification using the temporal structure of songs," in *Proc. IEEE 20th Int. Workshop Machine Learning for Signal Processing*, 2010, pp. 266–271.

[16] C. H. Lee, J. L. Shih, K. M. Yu, and H. S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 670–682, 2009.

[17] A. Nagathil, T. Gerkmann, and R. Martin, "Musical genre classification based on a highly-resolved cepstral modulation spectrum," in *Proc. 18th European Signal Processing Conf.*, Aalborg, Denmark, 2010, pp. 462–466.

[18] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[19] G. Liu, Z. Lin, S. Yan, J. Sun, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[20] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[21] D. Donoho, "For most large underdetermined systems of equations, the minimal ℓ_1-norm near-solution approximates the sparsest near-solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 7, pp. 907–934, 2006.

[22] Y. Panagakis and C. Kotropoulos, "Automatic music mood classification via low-rank representation," in *Proc. 19th European Signal Processing Conf.*, Barcelona, Spain, 2011, pp. 689–693.

[23] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.

[24] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[25] S. Shamma N. Mesgarani, M. Slaney, "Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations," *IEEE Trans. Audio, Speech, Lang. Proc.*, vol. 14, no. 3, pp. 920–930, 2006.

[26] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. 1st Int. Symposium Music Information Retrieval*, 2000, pp. 663–670.

[27] M. Ryynanen and A. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, pp. 72–86, 2008.

[28] M. Davies and Y. Eldar, "Rank awareness in joint sparse recovery," *IEEE Trans. Information Theory*, vol. 58, no. 2, pp. 1135–1146, 2012.

[29] R. Munkong and J. Biing-Hwang, "Auditory perception and cognition," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 98–117, 2008.

[30] C. Schoerkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *7th Sound and Music Computing Conf.*, Barcelona, Spain, 2010.

[31] M. Mandel and D. Ellis, "Song-level features and support vector machines for music classification," in *Proc. 6th Int. Conference Music Informatio Retrieval*, 2005, pp. 594–599.

[32] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[33] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[34] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Dept. Electrical Engineering, Stanford University, CA, USA, 2002.

[35] J. Wright and Y. Ma, "Dense error correction via ℓ_1-minimization," *IEEE Trans. Information Theory*, vol. 56, no. 7, pp. 3540–3560, 2010.

[36] M. Golbabaee and P. Vandergheynst, "Hyperspectral image compressed sensing via low-rank and joint-sparse matrix recovery," in *Proc. 2012 IEEE Int. Conf. Acoustics. Speech and Signal Processing*, Kyoto, Japan, 2012, pp. 2741–2744.

[37] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. 2010 Neural Information Processing Systems*, Vancouver, B.C., Canada, 2010, pp. 2496–2504.

[38] H. Ji, S. Huang, Z. Shen, and Y. Xu, "Robust video restoration by joint sparse and low rank matrix approximation," *SIAM J. Imaging Science*, vol. 4, no. 4, pp. 1122–1142, 2011.

[39] E. Richard, P.-A. Savalle, and N. Vayatis, "Estimation of simultaneously sparse and low rank matrices," in *Proc. Int. Conf. Machine Learning*, 2012.

[40] X. Liang, X. Ren, Z. Zhang, and Y. Ma, "Repairing sparse low-rank texture," in *Proc. European Conf. Computer Vision*, 2012, vol. Part V, pp. 482–495.

[41] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *Proc. European Conf. Computer Vision*, 2012, vol. Part VI, p. 470484.

[42] L. Zhang and C. Ma, "Low-rank decomposition and laplacian group sparse coding for image classification," *Neurocomputing*, vol. 135, pp. 339–347, 2014.

[43] M. Zhao, L. Jiao, J. Feng, and T. Liu, "A simplified low rank and sparse graph for semi-supervised learning," *Neurocomputing*, vol. 135, pp. 84–96, 2014.

[44] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and Alan S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[45] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[46] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Img. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[47] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *SIAM J. Optimization*, 2008.

[48] J. Yang and X. M. Yuan, "Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization," *Math. Comput.*, vol. 82, no. 281, 2013.

[49] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, Belmont, MA, 2nd edition, 1996.

[50] A. Yang, Z. Zhou, A. Balasubramanian, S. Sastry, and Ma Y., "Fast ℓ_1-minimization algorithms for robust face recognition.," *IEEE Trans Image Process.*, vol. 22, no. 8, pp. 3234–3246, 2013.

[51] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM J. Img. Sci.*, vol. 2, no. 2, pp. 569–592, 2009.

[52] H. Homburg, I. Mierswa, B. Moller, K. Morik, and M. Wurst, "A benchmark dataset for audio classification and clustering," in *Proc. 6th Int. Conf. Music Information Retrieval*, 2005, pp. 528–531.

[53] S. Dixon, F. Gouyon, and G. Widmer, "Towards characterisation of music via rhythmic patterns," in *Proc. 2004 Int. Conf. Music Information Retrieval*, Barcelona, Spain, 2004, pp. 509–516.

[54] F. Gouyon and S. Dixon, "Dance music classification: A tempo-based approach," in *Proc. 2004 Int. Conf. Music Information Retrieval*, Barcelona, Spain, 2004.

[55] E. Tsunoo, G. Tzanetakis, N. Ono, and S. Sagayama, "Beyond timbral statistics: Improving music classification using percussive patterns and bass lines," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 1003–1014, 2011.

[56] A. Flexer and D. Schnitzer, "Effects of album and artist filters in audio similarity computed for very large music databases," *Computer Music J.*, vol. 34, pp. 20–28, 2010.

[57] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Royal. Statist. Soc B.*, vol. 58, no. 1, pp. 267–288, 1996.

[58] S. L. Salzberg, "On comparing classifiers: Pitfalls to avoid and a recommended approach," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 317–328, 1997.

[59] Y. Panagakis and C. Kotropoulos, "Music classification by low-rank semantic mappings," *EURASIP J. Audio, Speech, and Music Processing*, vol. 3, 2013.

[60] G. Peeters, "Rhythm classification using spectral rhythm patterns," in *Proc. 2005 Int. Conf. Music Information Retrieval*, London, UK, 2005.

[61] K. Seyerlehner, G. Widmer, and Pohle T., "Fusing block-level features for music similarity estimation," in *Proc. 13th Int. Conf. Digital Audio Effects*, 2010, pp. 528–531.

[62] K. Aryafar and A. Shokoufandeh, "Music genre classification using explicit semantic analysis," in *Proc. 1st ACM Int. Workshop Music Information Retrieval with User-Centered and Multimodal Strategies*, 2011, pp. 33–38.

[63] C. Osendorfer, J. Schluter, J. Schmidhuber, and P. van der Smagt, "Unsupervised learning of low-level audio features for music similarity estimation," in *Proc. 28th Int. Conf. Machine Learning*, 2011.

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

[65] R.G. Baraniuk, V. Cevher, and M.B. Wakin, "Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 959–971, 2010.