Informed Group-Sparse Representation for Singing Voice Separation

Tak-Shing T. Chan, Member, IEEE and Yi-Hsuan Yang, Member, IEEE

Abstract—Singing voice separation attempts to separate the vocal and instrumental parts of a music recording, which is a fundamental problem in music information retrieval. Recent work on singing voice separation has shown that the low-rank representation and informed separation approaches are both able to improve separation quality. However, low-rank optimizations are computationally inefficient due to the use of singular value decompositions. Therefore, in this paper, we propose a new linear-time algorithm called informed group-sparse representation, and use it to separate the vocals from music using pitch annotations as side information. Experimental results on the iKala dataset confirm the efficacy of our approach, suggesting that the music accompaniment follows a group-sparse structure given a pre-trained instrumental dictionary. We also show how our work can be easily extended to accommodate multiple dictionaries using the DSD100 dataset.

Index Terms—Group-sparse representation, low-rank representation, singing voice separation, informed source separation.

I. INTRODUCTION

T
HE problem of recovering unknown sources from observed mixtures, known as source separation, has been successfully applied to various fields including communications, medical imaging, and audio [1]. Such an inverse problem is well-posed [2], [3] if it has a unique solution that depends on the data continuously. More specifically, in singing voice separation (SVS) [4]–[7], the aim is to separate the singing voice from the instrumentals, which has numerous applications in music information retrieval [8], [9]. Unfortunately, in the case of one microphone and more than one sources, a unique solution is mathematically impossible, so monaural source separation is generally ill-posed [10]. The prevalent approach to an ill-posed problem, say (arg) \( \min_X F(X) \), is to formulate a regularizer \( R(X) \) [2], [11], [12] which incorporates some prior assumptions:

\[
\min_X F(X) + \lambda R(X),
\]

where \( \lambda \) is a regularization parameter to be determined empirically (e.g., by cross validation). Usually, \( R(X) \) is chosen to favor a particular class of solutions. For example, one of the most popular regularizers is a sparsifier:

\[
\min_X F(X) + \lambda \|X\|_1,
\]

where the elementwise \( \ell_1 \)-norm \( \|X\|_1 \) encourages the matrix to be sparse [13]. This regularizer dominates an area of research known as sparse coding [14], [15], which is frequently used in audio separation [16], [17]. Another attractive regularizer is a low-rank regularizer:

\[
\min_X F(X) + \lambda \|X\|_*,
\]

where the trace norm \( \|X\|_* \) or the sum of the singular values of \( X \) is employed to favor low-rank solutions [18], [19]. This regularizer is often seen in singing voice separation in recent years [6], [20]–[22]. Last but not least, in informed audio source separation [22]–[24], we want to fuse external annotations into the final optimized solution, which is most helpful if the annotations are close to the correct solution (as in score-informed separation [25]). This requirement can be met by the following regularizer [20]–[28]:

\[
\min_X F(X) + \frac{\lambda}{2} \|X - X_0\|_F^2,
\]

where \( X \) can be a magnitude spectrogram and \( X_0 \) denotes the annotations on it. Such annotations may be obtained from the corresponding musical scores or from specific techniques for tracking the vocal melody contour. As evidenced above, regularization is quite versatile and it can incorporate both model assumptions and model answers into the problem itself. Still more information can be packed into the regularizer through a dictionary, as we will see in the related work below.

A. Related Work

The robust principal component analysis (RPCA) decomposes an input matrix \( X \) into a low-rank matrix \( A \) and a sparse matrix \( E \) [29], [30]:

\[
\min_{A,E} \|A\|_* + \lambda \|E\|_1 \text{ s.t. } X = A + E.
\]

Unlike traditional PCA, RPCA is robust against gross errors. For music spectrograms, if we assume that the instruments are repetitive [31] and the vocals are sparse [6], then the RPCA can be applied to the SVS problem [6]. This assumption is reasonable because musical instruments tend to have relatively stable and regular harmonic patterns while we can only sing one note at a time. The main drawback to this approach is that

Manuscript received Month xx, 2016; revised Month xx, 2016; accepted Month xx, 2017. Date of publication Month xx, 2017; date of current version Month xx, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. xxxxxxxx xxxxxxxx.

The authors are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 11564, Taiwan (e-mail: takshingchan@citil.sinica.edu.tw; yang@citil.sinica.edu.tw).

Digital Object Identifier 10.1109/LSP.2017.2647810

© 2017 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.
the resulting sparse matrix often contains instrumental solo or percussion [32, 33]. A partial solution for this problem is to incorporate reliable annotations for the sparse part using informed RPCA (hereafter RPCAi) [26]:

\[
\min_{A,E} \|A\|_* + \lambda \|E\|_1 + \frac{\gamma}{2} \|E - E_0\|_F^2 \quad \text{s.t.} \quad X = A + E,
\]

(6)

where \(E_0\) denotes the annotations (e.g., the pointwise product of \(X\) and a binary matrix). Sometimes \(A\) is not itself low-rank but is instead low-rank in a given dictionary. In this case, the low-rank representation (LRR) can be used [20, 34].

\[
\min_{Z,E} \|Z\|_1 + \lambda \|E\|_1 \quad \text{s.t.} \quad X = DZ + E,
\]

(7)

where \(D\) is a predefined (or pre-learned) dictionary such that \(A = DZ\). It can be seen that LRR is an extension of RPCA because (7) reduces to (5) when \(D = I\).

While we can perform dictionary-informed separation by simply combining LRR with the informed-separation-norm [4], LRR uses the singular value decomposition (SVD), an \(O(n^3)\) algorithm, which can be slow for larger datasets. In light of this, we will propose a dictionary-based group-sparse representation (GSR) model for SVD informed by annotated melodies [1] (GSRi). Our contributions are summarized in context in Table I.

TABLE I

| Method      | Objective                                                                 | Constraint                                                                 |
|-------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|
| RPCA [34]   | \(\|A\|_* + \gamma \|E\|_1\)                                                | \(X = A + E\)                                                             |
| RPCAi [26]  | \(\|A\|_* + \lambda \|E\|_1 + \frac{\gamma}{2} \|E - E_0\|_F\)              | \(X = A + E\)                                                             |
| LRR [34]    | \(\|Z\|_1 + \lambda \|E\|_1\)                                            | \(X = DZ + E\)                                                            |
| LRRi [26]   | \(\|Z\|_1 + \lambda \|E\|_1 + \frac{\gamma}{2} \|E - E_0\|_F\)          | \(X = DZ + E\)                                                            |
| GSR         | \(\|Z\|_2,1 + \lambda \|E\|_1\)                                         | \(X = DZ + E\)                                                            |
| GSRi        | \(\|Z\|_2,1 + \lambda \|E\|_1 + \frac{\gamma}{2} \|E - E_0\|_F\)          | \(X = DZ + E\)                                                            |

In what follows, we present our informed group-sparse representation model in Section II and the experimental results in Section III. The extension to multiple dictionaries is also described and tested before we conclude in Section IV.

II. INFORMED GROUP-SPARSE REPRESENTATION

In jazz and popular music, it is well known that a few chord symbols are enough to compactly represent the harmonic structure of a piece. To motivate our new representation, let us begin with a simple chord sequence C-G-F-G-C for the instrumental part (see [37] for chord notations). If we have a learned dictionary with the C, Dm, Em, F, G, Am, and Bbm9 chords, then the C, F, and G chords can be represented as:

\[
C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T, \quad F = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T, \quad G = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T,
\]

(8) (9) (10)

and the time-atom representation of C-G-F-G-C becomes:

\[
Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \end{bmatrix}^T, \quad 0 & 0 & 0 & 0 & 0 \quad 0 & 0 & 1 & 0 & 0 \quad 0 & 1 & 0 & 1 & 0 \quad 0 & 0 & 0 & 0 & 0 \quad 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot
\]

(11)

One observation is that there are many empty rows in this representation (because not all the chords in the dictionary are used in the given sequence). So, a promising strategy for the inverse problem is to encourage row sparsity given an instrumental dictionary. Together with the idea of informed separation incorporating vocal annotations (6), we arrive at the following formulation:

\[
\min_{Z,E} \|Z\|_{2,1} + \lambda \|E\|_1 + \frac{\gamma}{2} \|E - E_0\|_F^2 \quad \text{s.t.} \quad X = DZ + E,
\]

(12)

where \(X\) is the input spectrogram, \(D\) is the instrumental dictionary, \(E_0\) denotes the vocal annotations, \(DZ\) is the separated instrumental, \(E\) is the separated vocals, and \(\|Z\|_{2,1} = \sum_i \sqrt{\sum_j Z_{ij}^2}\) denotes the sum of the \(\ell_2\)-norms of the rows of \(Z\). As row sparsity is a kind of group sparsity, we call this the informed group-sparse representation (GSRi). In case where the vocal annotations are unavailable, we set \(\gamma\) to zero, which we simply call the group-sparse representation (GSR). Our observation is further strengthened by the fact that group sparsity has been successfully applied to other audio processing models before [39, 41].

A. Optimization

The above formulation is not trivial to solve since the \(\|\cdot\|_{2,1}\) and \(\|\cdot\|_1\) norms are nonsmooth. Moreover, there is an additional equality constraint to be satisfied. In this case, the alternating direction method of multipliers (ADMM) [42] can be applied. ADMM works by first rewriting the constraint(s) into an augmented Lagrange function, then updating each variable in an alternating fashion until convergence. Although the convergence of ADMM has not been fully proven, it often converges in practice (cf. [20]). Thus, to solve (12), we first introduce two auxiliary variables \(J\) and \(B\) for the alternating updates and rewrite the optimization problem as follows:

\[
\min_{Z,J,B} \|J\|_{2,1} + \lambda \|B\|_1 + \frac{\gamma}{2} \|E - E_0\|_F^2 \quad \text{s.t.} \quad X = DZ + E,
\]

(13)

The unconstrained augmented Lagrangian \(\mathcal{L}\) is given by:

\[
\mathcal{L} = \|J\|_{2,1} + \lambda \|B\|_1 + \frac{\gamma}{2} \|E - E_0\|_F^2 + \langle Y_1, X - DZ - E \rangle + \langle Y_2, Z - J \rangle + \langle Y_3, E - B \rangle + \frac{\mu}{2} \left( \|X - DZ - E\|_F^2 + \|Z - J\|_F^2 + \|E - B\|_F^2 \right)
\]

(14)

where \(Y_1, Y_2, \) and \(Y_3\) are the Lagrange multipliers. We then iteratively update the solutions for \(J, Z, B, \) and \(E\).
B. Relation to Low-Rank Representation

In Section I[9] we have seen that LRR is equivalent to RPCA when \( D = I \). There is a similar relation between LRR and GSR. We can factorize the matrix \( Z \) as follows (cf. [43]):

\[
Z = I_k \text{ diag}(\|Z_1\|, \ldots, \|Z_k\|) \begin{pmatrix} Z_1/\|Z_1\| & \cdots & Z_k/\|Z_k\| \end{pmatrix}.
\]

If \( Z \) has orthogonal rows, then the above is also a valid SVD, since \( I_k \) is orthonormal and the normalization above makes the rightmost term orthonormal too. As a consequence, we have:

\[
\|Z\|_F = \sum_{i=1}^k \|Z_i\| = \|Z^T\|_{2,1}.
\]

Given this condition, the equivalence between LRR and GSR can be easily established.

---

2In order to encourage reproducible research, all the code for our paper is made available at http://mac.citi.sinica.edu.tw/ikala/code.html

---

III. Experimental Results

To evaluate the performance of GSRi, we use a source separation competition dataset called the iKala dataset [22]. This dataset contains 252 30-second mono clips with human-labeled vocal pitch contours. The instrumentals and vocals are mixed at 0 dB signal-to-noise ratio. We randomly select 44 songs as the training set, leaving 208 songs for the test set. The songs are downsampled from 44 100 Hz to 22 050 Hz to reduce memory usage, then a short-time Fourier transform (STFT) with a 1 411-point Hann window with 75% overlap is used to obtain the spectrograms [22]. The magnitude spectrogram \( X \) is fed into GSRi and the separated components are reconstructed via inverse STFT using the original phase \( P \). To get the vocal annotations, we first transform the human-labeled vocal pitch contours into a time-frequency binary mask. The authors of [21] have proposed a harmonic mask similar to that of [44], where it passes only integral multiples of the vocal fundamental frequencies (cf. [5], [45]):

\[
M(f, t) = \begin{cases} 1, & \text{if } |f - nF_0(t)| < \frac{w}{2}, \ \exists n \in \mathbb{N}^+, \\ 0, & \text{otherwise.} \end{cases}
\]

Here \( F_0(t) \) is the vocal fundamental frequency at time \( t \), \( n \) is the order of the harmonic, and \( w \) is the width of the mask, which we set to \( w = 80 \) Hz as in [21]. Then we simply define the vocal annotations as \( E_0 = X \circ M \), where \( \circ \) denotes the Hadamard product. Our experimental setup is shown in Fig. [1].

A. Algorithms

The algorithms to be compared are summarized in Table I[4] All these methods (except RPCA and RPCAi) require prior training. For completeness, we further propose the informed LRR (LRRi) by simply replacing the \( \ell_{2,1} \) norm by the trace norm. The resulting subproblem \( J = \arg \min J (\|J\|_F + \frac{\lambda}{2} \|J - (Z + \mu^{-1}Y_2)\|_F^2) \) can be solved by singular value thresholding [34], [46]. The convergence criteria is \( \|X - A - E\|_F \leq \|X\|_F \leq 10^{-4} \), with \( A = DZ \) if applicable. For \( X \in \mathbb{R}^{m \times n}, \lambda \) is set to \( 1/\sqrt{\max(m, n)} \), and \( \gamma \) is set to \( 2/\sqrt{\max(m, n)} \), following a grid search on the training set. All six algorithms are implemented from scratch using IALM with the same \( \mu = 10^{-3} \) and \( \rho = 1.2 \) (see Section II[11]) to ensure fair timing comparisons.

B. Dictionary

We use non-negative sparse coding (NNSC) in the SPAMS toolbox [47], [48] to train our instrumental dictionary. Given \( n \) input frames \( x_i \in \mathbb{R}^m \), NNSC [49] learns a dictionary \( D \) by solving the following joint optimization problem:

\[
\min_{D \geq 0, \alpha} \sum_{i=1}^n \frac{1}{2} \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \quad \text{s.t. } \forall i, \ \alpha_i \geq 0,
\]

where \( \| \cdot \|_2 \) denotes the Euclidean norm and \( \lambda \) is a regularization parameter, which we set to \( 1/\sqrt{m} \) as in [47]. The input frames are extracted from the training set after STFT. Following [20], we define the dictionary size to be 100 atoms.

---

3We have tried removing the non-negative constraints on \( D \) and \( \alpha \) but this does not change the results significantly for both datasets in this section.
C. Evaluation

For both the instrumental and the vocals, separation performance is measured by BSS Eval toolbox version 3.1 in terms of source-to-distortion ratio (SDR), source-to-interferences ratio (SIR), and sources-to-artifacts ratio (SAR) \[50\], with higher values indicating better separation. We also compute the normalized SDR (NSDR) which is the improvement in SDR using the initial mixture as baseline \[35\]. We then report the average result, denoted by the G prefix, for the test set. The most important measure is GNSDR as it measures the overall performance improvement. In addition, we also report the total running time of each algorithm on an IBM System x3650 M4 (two Intel E5-2697 v2 CPUs at 2.70 GHz) with 384 GB RAM.

D. Results

We can make several observations from the results and running times shown in Table II. First, the informed algorithms (RPCAi, LRRi, GSRi) clearly outperform their uninformed counterparts, ascertaining the usefulness of informed separation. Second, the performance of GSRi and LRRi are comparable, with GSRi performing slightly better in the music accompaniment part and LRRi performing slightly better in the vocal part. Third, the advantage of the learned dictionary can be shown by the superiority of GSR and LRR families over the RPCAi family in the music accompaniment part. This means that the dictionary has successfully learned relevant information so that it performs better than plain sinusoids. Fourth, the running time of the GSR family is the fastest, showing the speed improvement by removing SVDs. Fifth, while the informed versions of RPCAi and LRRi are slower than the uninformed ones, this is not the case for GSR, for GSRi iterates much faster than LRRi. This makes GSRi more attractive than the alternatives. Finally, we remark that LRR is faster than RPCA because the SVD is applied to \( Z \), which is much smaller than \( A \).

E. The Use of Multiple Dictionaries

Suppose we concatenate \( \kappa \) instrumental dictionaries together such that \( D = (D_1 \ldots D_\kappa) \). We can then apply GSRi directly to obtain the solution of the following:

\[
\min_{Z, E} \left( \frac{1}{2} \|Z^T \ldots Z_\kappa^T\|_2 + \lambda \|E\|_1 + \frac{\gamma}{2} \|E - E_0\|^2_F \right) \tag{21}
\]

s.t. \( X = (D_1 \ldots D_\kappa) (Z_1^T \ldots Z_\kappa^T)^T + E \),

where \( Z = (Z_1^T \ldots Z_\kappa^T)^T \). As \( DZ = \sum_{i=1}^{\kappa} D_i Z_i \), both \( DZ \) and \( D_i Z_i \) can be interpreted as magnitude spectrograms: the former represents the instrumentals as a whole, while the latter represent the decomposed components associated with each dictionary. We call \[21\] the informed multiple-group-sparse representation (MGSRi). To test whether multiple dictionaries is a feasible idea, we use the DSD100 dataset which contains 50 songs for training and 50 for testing. To fit the SVS theme, we restrict ourselves to the pop/singer-songwriter subset. Each song in DSD100 contains four sources (bass, drums, other, and vocals), so we can perform SVS by either:

- **GSRi**: We mix bass, drums, and other equally into a single instrumental source. Then we learn the instrumental dictionary from this combined source and apply GSRi.
- **MGSRI**: We train \( \kappa = 3 \) dictionaries for bass, drums, and other, respectively. After solving for \( Z \) and \( E \), the instrumentals as a whole (\( DZ \)) is used for comparison.

To reduce computations, we downmix to mono and downsample from 44100 Hz to 22050 Hz. As DSD100 does not have pitch contour labels, we create them by running MELODIA \[56\] on the ground truth vocals. To eliminate the possible effect of dictionary size, we train the GSRi dictionary with 300 atoms. Here we choose \( w = 60 \) Hz and \( \lambda = \gamma = 1/\sqrt{\max(m, n)} \) for each \( X \in \mathbb{R}^{m \times n} \), after a grid search on the training set. From the results in Table III we can conclude that MGSRI is not inferior to GSRi in SVS, with the biggest advantage that it can separate all the instrumental components (as in SiSEC MUS)\[7\] while GSRi cannot.

### Table II

|                          | GNSDR | GSDR | GSIR | GSAR | Runtime  |
|--------------------------|-------|------|------|------|----------|
| RPCAi                    | E     | 2.41 | 6.21 | 8.14 | 12.53    | 02:24:54 |
|                          | A     | 4.48 | 0.76 | 3.23 | 7.00     |          |
| RPCAi                    | E     | 7.83 | 11.74| 17.82| 13.31    |          |
|                          | A     | 10.89| 7.17 | 13.31| 9.00     | 03:19:37 |
| LRR                      | E     | 3.93 | 7.73 | 11.41| 11.17    | 00:25:03 |
|                          | DZ    | 5.42 | 1.70 | 3.40 | 9.63     |          |
| LRRi                     | E     | 7.75 | 11.55| 16.92| 13.38    | 00:30:12 |
|                          | DZ    | 11.29| 7.56 | 14.92| 8.87     |          |
| GSR                      | E     | 2.50 | 6.30 | 7.36 | 14.80    | 00:13:16 |
|                          | DZ    | 5.25 | 1.53 | 5.15 | 5.89     |          |
| GSRi                     | E     | 7.71 | 11.51| 16.34| 13.63    | 00:13:15 |
|                          | DZ    | 11.31| 7.59 | 15.19| 8.82     |          |

### Table III

|                  | GNSDR | GSDR | GSIR | GSAR |
|------------------|-------|------|------|------|
| GSRi             | E     | 8.08 | 4.64 | 11.24| 6.11  |
|                  | DZ    | 5.49 | 8.94 | 13.36| 11.30 |
| MGSRI            | E     | 8.03 | 4.59 | 10.62| 6.29  |
|                  | DZ    | 5.56 | 9.01 | 13.62| 11.24 |

[1] http://bass-db.gforge.inria.fr/

---

To obtain the solution of the following:

\[
\min_{Z, E} \left( \frac{1}{2} \|Z^T \ldots Z_\kappa^T\|_2 + \lambda \|E\|_1 + \frac{\gamma}{2} \|E - E_0\|^2_F \right) \tag{21}
\]

s.t. \( X = (D_1 \ldots D_\kappa) (Z_1^T \ldots Z_\kappa^T)^T + E \),

where \( Z = (Z_1^T \ldots Z_\kappa^T)^T \). As \( DZ = \sum_{i=1}^{\kappa} D_i Z_i \), both \( DZ \) and \( D_i Z_i \) can be interpreted as magnitude spectrograms: the former represents the instrumentals as a whole, while the latter represent the decomposed components associated with each dictionary. We call \[21\] the informed multiple-group-sparse representation (MGSRi). To test whether multiple dictionaries is a feasible idea, we use the DSD100 dataset which contains 50 songs for training and 50 for testing. To fit the SVS theme, we restrict ourselves to the pop/singer-songwriter subset. Each song in DSD100 contains four sources (bass, drums, other, and vocals), so we can perform SVS by either:

- **GSRi**: We mix bass, drums, and other equally into a single instrumental source. Then we learn the instrumental dictionary from this combined source and apply GSRi.
- **MGSRI**: We train \( \kappa = 3 \) dictionaries for bass, drums, and other, respectively. After solving for \( Z \) and \( E \), the instrumentals as a whole (\( DZ \)) is used for comparison.

To reduce computations, we downmix to mono and downsample from 44100 Hz to 22050 Hz. As DSD100 does not have pitch contour labels, we create them by running MELODIA \[56\] on the ground truth vocals. To eliminate the possible effect of dictionary size, we train the GSRi dictionary with 300 atoms. Here we choose \( w = 60 \) Hz and \( \lambda = \gamma = 1/\sqrt{\max(m, n)} \) for each \( X \in \mathbb{R}^{m \times n} \), after a grid search on the training set. From the results in Table III we can conclude that MGSRI is not inferior to GSRi in SVS, with the biggest advantage that it can separate all the instrumental components (as in SiSEC MUS)\[7\] while GSRi cannot.

### IV. Conclusion

In this paper, we have proposed a novel GSRi method for SVS which incorporates both an instrumental dictionary and vocal annotations to inform the source separation process. Experimental results have shown that GSRi achieves the best performance in terms of instrumental GNSDR, GSDR, GSIR, and running time, making GSRi the best candidate for de-soloing applications. We have also successfully extended GSRi to the multiple-dictionary case. In conclusion, our experiments have shown that group sparsity achieves comparable results to low-rankness in a dictionary, but in a more efficient manner.

[1] http://bass-db.gforge.inria.fr/

---

Experimental results have shown that GSRi achieves the best performance in terms of instrumental GNSDR, GSDR, GSIR, and running time, making GSRi the best candidate for de-soloing applications. We have also successfully extended GSRi to the multiple-dictionary case. In conclusion, our experiments have shown that group sparsity achieves comparable results to low-rankness in a dictionary, but in a more efficient manner.
