REAL-TIME ONLINE SINGING VOICE SEPARATION FROM MONAURAL RECORDINGS USING ROBUST LOW-RANK MODELING

Pablo Sprechmann
University of Minnesota
sprec009@umn.edu

Alex Bronstein
Tel Aviv University
bron@eng.tau.ac.il

Guillermo Sapiro
University of Minnesota
guille@umn.edu

ABSTRACT

Separating the leading vocals from the musical accompaniment is a challenging task that appears naturally in several music processing applications. Robust principal component analysis (RPCA) has been recently employed to this problem producing very successful results. The method decomposes the signal into a low-rank component corresponding to the accompaniment with its repetitive structure, and a sparse component corresponding to the voice with its quasi-harmonic structure. In this paper we first introduce a non-negative variant of RPCA, termed as robust low-rank non-negative matrix factorization (RNMF). This new framework better suits audio applications. We then propose two efficient feed-forward architectures that approximate the RPCA and RNMF with low latency and a fraction of the complexity of the original optimization method. These approximants allow incorporating elements of unsupervised, semi- and fully-supervised learning into the RPCA and RNMF frameworks. Our basic implementation shows several orders of magnitude speedup compared to the exact solvers with no performance degradation, and allows online and faster-than-real-time processing. Evaluation on the MIR-1K dataset demonstrates state-of-the-art performance.

1. INTRODUCTION

The leading voice in musical pieces carries valuable information about the song. A system capable of separating the singing voice from the music accompaniment can be used to facilitate a number of applications such as music information retrieval, singer identification, or lyric recognition.

Separating the leading singing voice from the musical background from a monaural recording is very challenging. Existing approaches can be classified according to the level of supervision that they require. Supervised approaches tend to have a model for either the musical background, the singing voice, or both, and in general map the mixture signals onto a feature space where the separation is performed, e.g. [4, 11, 15, 19]. A common drawback of these methods is the need to identify the vocal segments beforehand, typically using features such as the Mel-Frequency Cepstrum Coefficients (MFCC). Unsupervised approaches make basic fundamental assumptions requiring no prior training or particular features. For example, in [13] the authors tackle the separation by extracting the repeating background (music) from the non-repeating foreground (voice). Most relevant for our work is the method proposed in [9]. The authors model the repetitive structure of the accompaniment with a low-rank linear model, while the singing voice is regarded as sparse and non-repetitive. The separation is performed using *robust* PCA (RPCA) [3], producing state-of-the-art results. Common drawbacks of unsupervised approaches include the requirement to observe the whole audio track to perform the separation and the fact that, unlike supervised models, the obtained sources might not follow known characteristics of the signals.

In this paper, we consider the promising results presented in [9] as a starting point. We first develop an extension of RPCA in which the low rank model is represented as a *non-negative* linear combination of *non-negative* basis vectors. This is done following recent results connecting non-convex optimization with nuclear norm optimization [17, 18] (further references are given in Section 2). As with standard non-negative matrix factorization (NMF) methods, this new model is more appropriate to represent audio signals, being applied to the magnitude of the spectrum. The use of robust NMF (RNMF) is not restricted to this application and the usage in combination with divergences in lieu of Euclidean distances is straightforward. The

proposed framework can also be seen as an extension of the robustification of NMF introduced in [22]; not only does our model consider a sparse variable accounting for outliers (singing voice), but it also adds a regularization term that minimizes the rank of the linear model.

In Section 3 we show that the RPCA and RNMF frameworks induce an architecture of multi-layer feed-forward networks designed to approximate the output of the exact optimization algorithms at a fraction of their computational cost and with no decrease in performance in our various experiments. Moreover, this new framework allows to incorporate unsupervised, semi- and fully-supervised learning into RPCA and RNMF. In this way, we aim at taking the advantages of the unsupervised methods while minimizing their drawbacks via realistic learning. When combined with learning as here proposed, the obtained networks produce over 1 dB improvement in the signal-to-distortion ratio when compared to the optimization-based RPCA (extensive experimental results are presented in Section 4), and, after the offline learning, are computable online and faster than real time without the need to observe the whole audio file.

These proposed networks are closely related to the ones introduced in [6], used to produce meaningful audio features for music style and gender classification [7]. These approaches are examples of recent successful efforts in the machine learning community to produce fast trainable (auto-)encoders of sparse features of visual and audio signals (see [5, 16] and references therein). While the work in this paper comes from these ideas, it presents a fundamental difference in the sense that the proposed networks do not compute features, but perform the full separation of the singing voice from the musical accompaniment.

2. LOW-RANK SPARSE MODELS

2.1 Robust PCA

Principal component analysis (PCA) is the most widely used statistical technique for dimensionality reduction. Its performance is, however, highly sensitive to the presence of samples not following the assumed model (subspace); even a single outlier in the data matrix can render the estimation of the low rank component arbitrarily far from the true model. In [3, 21], a very elegant remedy was developed for this shortcoming, in which the low rank matrix is determined as the minimizer of a convex program. The basic idea is to decompose the data matrix $\mathbf{X}$ as $\mathbf{X} = \mathbf{L} + \mathbf{O} \in \mathbb{R}^{m \times n}$, where $\mathbf{L}$ is a low rank matrix and $\mathbf{O}$ an error matrix with a sparse number of non-zero coefficients with arbitrarily large magnitude. RPCA can be solved by minimizing the convex program

$$\min_{\mathbf{L},\mathbf{O}} \|\mathbf{L}\|_* + \lambda \|\mathbf{O}\|_1 \text{ s.t. } \mathbf{X} = \mathbf{L} + \mathbf{O}, \quad (1)$$

where $\|\cdot\|_*$ denotes the matrix nuclear norm, defined as the sum of the singular values (the convex surrogate of the rank), and λ is a positive scalar parameter controlling the sparsity of the outliers. Several efficient optimization algorithms have been proposed for solving (1) as, for example, the augmented Lagrangian approach presented in [12].

When the observations are noisy, the equality constraint in (1) no longer holds. The RPCA model can be reformulated as

$$\min_{\mathbf{L},\mathbf{O}} \|\mathbf{L}\|_* + \lambda \|\mathbf{O}\|_1 \text{ s.t. } \|\mathbf{X} - \mathbf{L} - \mathbf{O}\|_F^2 \leq \epsilon, \quad (2)$$

with $\|\cdot\|_{\mathrm{F}}$ denoting the Frobenius norm, and ϵ a parameter controlling the approximation error [21].

2.2 Robust PCA via non-convex factorization

In this paper, we tackle the RPCA problem by solving the unconstrained optimization problem

$$\min_{\mathbf{L},\mathbf{O}} \frac{1}{2} \|\mathbf{X} - \mathbf{L} - \mathbf{O}\|_F^2 + \lambda_* \|\mathbf{L}\|_* + \lambda \|\mathbf{O}\|_1 . \quad (3)$$

This formulation is equivalent to (2) in the sense that for every $\epsilon > 0$ one can find a $\lambda_* > 0$ such that both problems admit the same solution. The unconstrained formulation can be efficiently optimized via proximal methods as in [3].

In [17] it was shown that the nuclear norm of a matrix can be reformulated as a penalty over all possible factorizations,

$$\|\mathbf{L}\|_* = \min_{\mathbf{U},\mathbf{S}} \frac{1}{2} \|\mathbf{U}\|_F^2 + \frac{1}{2} \|\mathbf{S}\|_F^2 \text{ s.t. } \mathbf{US} = \mathbf{L}, \quad (4)$$

with the minimum achieved via Singular Value Decomposition (SVD) [14]. In (3), neither the rank of $\mathbf{L}$ nor the level of sparsity in $\mathbf{O}$ are assumed known *a priori*. However, in common applications, it is reasonable to have a rough upper bound, $\text{rank}(\mathbf{L}) \leq q$. Combining this with (4), we reformulate (3) as the minimization

$$\min_{\mathbf{U},\mathbf{S},\mathbf{O}} \quad \tfrac{1}{2} \|\mathbf{X} - \mathbf{US} - \mathbf{O}\|_F^2 + \tfrac{\lambda_*}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{S}\|_F^2) + \lambda \|\mathbf{O}\|_1 \quad (5)$$

over $\mathbf{U} \in \mathbb{R}^{m \times q}$, $\mathbf{S} \in \mathbb{R}^{q \times n}$, and $\mathbf{O} \in \mathbb{R}^{m \times n}$. This decomposition reveals interesting structure hidden in the problem. The low rank component can now be thought of as an under-complete dictionary $\mathbf{U}$, with q atoms, multiplied by a matrix $\mathbf{S}$ containing the corresponding coefficients for each data vector in $\mathbf{X}$. This

interpretation brings the RPCA problem close to that of matrix factorization and sparse coding.

This new factorized formulation drastically reduces the number of optimization variables from $2nm$ to $nm + q(n + m)$. While problem (5) is no longer convex, it can be shown that any of its stationary points satisfying $||\mathbf{X} - \mathbf{US} - \mathbf{O}||_2^2 \leq \lambda_*$, is an optimal solution of (5) [14]. Thus, the problem can be solved using alternating minimization or block coordinate schemes, without the risk of remaining stuck in a local minimum. This redounds in a significant speed-up in the optimization [18].

2.3 Robust NMF

In many applications, such as spectrogram decompositions, it desirable to find non-negative factorizations. This is in the heart of the non-negative matrix factorization paradigm . We now extend (5) to consider the low rank and the outlier terms to be non-negative,

$$\min_{\mathbf{U}\geq\mathbf{0},\mathbf{S}\geq\mathbf{0},\mathbf{O}\geq\mathbf{0}} \quad \tfrac{1}{2}\left\|\mathbf{X}-\mathbf{US}-\mathbf{O}\right\|_F^2 + \tfrac{\lambda_*}{2}(\left\|\mathbf{U}\right\|_F^2 + \left\|\mathbf{S}\right\|_F^2) + \lambda\left\|\mathbf{O}\right\|_1 . \quad (6)$$

This new formulation is no longer equivalent to (3). In fact, applying (4) directly to the matrix $\mathbf{US}$, we obtain $\hat{\mathbf{U}}\hat{\mathbf{S}}$ with the factors $\hat{\mathbf{S}}$ and $\hat{\mathbf{U}}$ not being necessarily non-negative. Adding the non-negativity constraint produces the inequality

$$||\mathbf{US}||_* \leq \frac{1}{2}\min_{\hat{\mathbf{S}}\geq\mathbf{0},\hat{\mathbf{U}}\geq\mathbf{0}} ||\hat{\mathbf{U}}||_{\mathrm{F}}^2 + ||\hat{\mathbf{S}}||_{\mathrm{F}}^2. \quad (7)$$

Thus, the sum of the Frobenius norms of the non-negative matrices $\mathbf{S}$ and $\mathbf{U}$ regularizes an upper bound of the nuclear norm of their product.

Standard NMF is obtained as a particular case by setting to zero both λ_* and λ, while the robust version of NMF introduced in [22] is obtained when only λ_* is selected as zero. In this paper we use RNMF as stated in (6), however its extension to more general fitting terms such as β-divergences is straightforward. Problem (6) can be optimized using multiplicative algorithms, commonly used in the NMF context.

2.4 Robust non-negative projections

Let us now assume to be given a low dimensional model, $\mathbf{U} \in \mathbb{R}^{m\times q}$, learned from some data $\mathbf{X} \approx \mathbf{US} + \mathbf{O} \in \mathbb{R}^{m\times n}$. A new input vector $\mathbf{x}$ drawn from the same distribution as $\mathbf{X}$ can be decomposed into $\mathbf{x} = \mathbf{Us} + \mathbf{n} + \mathbf{o}$, where $\mathbf{Us}$ represents the low dimensional component, $\mathbf{n}$ is a small perturbation, and $\mathbf{o}$ is a sparse outlier vector. It can be obtained via

$$\min_{\mathbf{s}\geq\mathbf{0},\mathbf{o}\geq\mathbf{0}} \quad \tfrac{1}{2}\left\|\mathbf{x}-\mathbf{Us}-\mathbf{o}\right\|_2^2 + \tfrac{\lambda_*}{2}\left\|\mathbf{s}\right\|_2^2 + \lambda\left\|\mathbf{o}\right\|_1 , \quad (8)$$

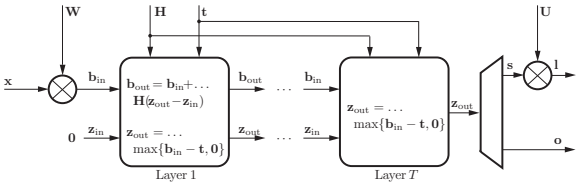

Figure 1. RNMF encoder architecture with T layers.

a convex problem similar to the one of standard sparse coding. The solution can be obtained via proximal methods [1], which split the objective function (8) into a smooth part (the first two terms), and a non-differentiable part (the ℓ_1 norm of the outliers vector). Proximal methods iterate between a gradient descent on the smooth function and an application of the proximal operator (which assumes a closed form of one-sided soft-thresholding), as detailed in Algorithm 1. This algorithm is conceptually very similar to the popular iterative shrinkage and thresholding algorithm (ISTA) [2]. We do not use this algorithm as an explicit tool, but rather as a motivation of the architecture of a feed-forward network capable of accurately performing the separation in real time, as discussed next.

input : Data $\mathbf{x}$, dictionary $\mathbf{U}$.
output: Nonnegative coefficient vector $\mathbf{s}$ and nonnegative outlier vector $\mathbf{o}$.
Define
$\mathbf{H} = \mathbf{I} - \frac{1}{\alpha}\begin{pmatrix} \mathbf{U}^{\mathrm{T}}\mathbf{U} + \lambda_*\mathbf{I} & \mathbf{U}^{\mathrm{T}} \\ \mathbf{U} & (1+\lambda_*)\mathbf{I} \end{pmatrix}$,
$\mathbf{W} = \frac{1}{\alpha}\begin{pmatrix} \mathbf{U}^{\mathrm{T}} \\ \mathbf{I} \end{pmatrix}$, and $\mathbf{t} = \frac{\lambda}{\alpha}\begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix}$.
Initialize $\mathbf{z} = \mathbf{0}$, $\mathbf{b} = \mathbf{Wx}$.
repeat
  $\mathbf{y} = \max\{\mathbf{b} - \mathbf{t}, 0\}$
  $\mathbf{b} = \mathbf{b} + \mathbf{H}(\mathbf{y} - \mathbf{z})$
  $\mathbf{z} = \mathbf{y}$
until *until convergence* ;
Output $(\mathbf{o}, \mathbf{s}) = \mathbf{z}$.
Algorithm 1: RNMF given the dictionary $\mathbf{U}$.

3. FAST ROBUST SPARSE MODELING

To avoid the computational complexity inherent to exact sparse coding algorithms, it has been recently proposed to learn non-linear regressors capable of producing good approximations in a fixed amount of time [6,10]. We follow these ideas to obtain encoders capable of efficiently approximating the solution of RPCA and RNMF. [1] We first discuss the general framework and then describe specific uses.

[1] Due to space constraints, we show details only for RNMF; RPCA can be obtained by removing the non-negativity constraints and modifying the proximal operator.

We aim at constructing a parametric regressor $\mathbf{z} = (\mathbf{o}, \mathbf{s}) = \mathbf{h}(\mathbf{x}, \mathbf{\Theta})$, with some set of parameters, collectively denoted as $\mathbf{\Theta}$, capable of accurately performing the singing voice separation for a given training sample $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Here, each $\mathbf{x}_i$ represents the magnitude spectrum of a mixture of voice and music; training samples may come from many different singers and songs.

As in [6], we design an architecture for the encoders based on an exact optimization algorithm, in this case Algorithm 1. We propose a multi-layer artificial neural networks where each layer implements a single iteration of the algorithm, as depicted in Figure 1. The parameters of the network are the matrices $\mathbf{W}$ and $\mathbf{H}$ and the thresholds $\mathbf{t}$.[2] These encoder architectures are continuous and almost everywhere $\mathcal{C}^1$ with respect to the parameters, allowing the use of (sub)gradient descent methods for training.

We train the encoders by minimizing over $\mathcal{X}$ functions of the form

$$\mathcal{L}(\mathbf{\Theta}) \quad = \quad \frac{1}{|\mathcal{X}|} \sum_{x_i \in \mathcal{X}} L(\mathbf{\Theta}, \mathbf{x}_i), \qquad (9)$$

where $L(\mathbf{\Theta}, \mathbf{x}_i)$ is a function that measures the quality of the code $\mathbf{z}_i = \mathbf{h}(\mathbf{x}_i, \mathbf{\Theta})$. Specifically, we iteratively select a random subset of $\mathcal{X}$ and then update the network parameters as $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \mu \frac{\partial \mathcal{L}(\mathbf{\Theta})}{\partial \mathbf{\Theta}}$, where μ is a decaying step, repeating the process until convergence. The decoder is just a linear operator given by a dictionary $\mathbf{U}$, see Figure 1.

Once trained, the parameters $\mathbf{\Theta}$ and the dictionary $\mathbf{U}$ are fixed, and the network is used to sequentially process new data. The latency of both the RPCA and RNMF networks (referred henceforth as *NN RPCA* and *NN RNMF*, respectively) is of the order of a single STFT frame (hundreds of milliseconds), while the exact algorithms require the entire signal to be observed.

3.1 Training regimes

Training of the proposed RPCA and RNMF encoders is possible under different regimes. We refer as *supervised* to the setting where the training set consists of the mixed signal $\mathbf{x}_i = \mathbf{o}_i^* + \mathbf{l}_i^*$, and the synchronized ground truth voice and accompaniment signals $\mathbf{o}_i^*$ and $\mathbf{l}_i^*$ (each vector corresponding to the magnitude spectrogram). In that case, we set $L(\mathbf{\Theta}, \mathbf{x}_i) = ||\mathbf{U}\mathbf{s}_i - \mathbf{l}_i^*||_2^2 + ||\mathbf{o}_i - \mathbf{o}_i^*||_2^2$, with $(\mathbf{o}_i, \mathbf{s}_i) = \mathbf{h}(\mathbf{x}_i, \mathbf{\Theta})$. For *NN RPCA*, the dictionary $\mathbf{U}$ is established using SVD applied to the clean accompaniment samples, $\mathbf{l}_i^*$, while for *NN RNMF*, the non-negative dictionary $\mathbf{U}$ is constructed running the multiplicative RNMF algorithm on the training data.

[2] In the network, extra flexibility is obtained by learning different thresholds t_i for each component.

Table 1. Performance on the recovered vocal track on MIR-1K.

Method	GNSDR	GSNR	GSAR	GSIR
Ideal freq. mask	13.48	5.46	13.65	31.22
ADMoM RPCA [9]	5.00	2.38	6.68	13.76
Proximal RPCA	5.48	3.29	7.02	13.91
NN RPCA Untrained	5.30	2.66	6.80	13.00
NN RPCA Unsupervised	5.62	2.87	6.90	14.02
NN RPCA Supervised	6.38	3.18	7.22	16.47
NN RPCA Dict. update	6.42	3.19	7.23	16.57
Multiplicative RNMF	5.60	3.39	6.94	14.67
NN RNMF Untrained	1.62	0.00	5.85	5.13
NN RNMF Unsupervised	5.00	2.66	6.63	11.89
NN NMF Supervised	6.36	3.37	7.10	16.96
NN RNMF Dict. update	**6.55**	**3.55**	**7.24**	**17.65**

We refer as *semi-supervised* to the setting in which isolated samples of voice and background are available, but are not synchronized (the $\mathbf{x}_i$ are now either the voice or the accompaniment). The training of the network is performed in the same way as the supervised case, but setting to zero the missing source.

Finally, in the *unsupervised* setting we only have access to mixtures as training data and the objective $L(\mathbf{\Theta}, \mathbf{x}_i) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{U}\mathbf{s}_i - \mathbf{o}_i\|_2^2 + \frac{\lambda_*}{2}\|\mathbf{s}_i\|_2^2 + \lambda\|\mathbf{o}_i\|_1$ is used to directly minimize the cost in (6).

Dictionary adaptation. The performance of both the RPCA and RNMF networks can be further improved if the dictionary $\mathbf{U}$ (decoder) is updated during the training. In the unsupervised setting, for *NN RPCA*, $\mathbf{U}$ is updated via gradient descent as before, while in *NN RNMF* via the standard multiplicative update,

$$\mathbf{U} \quad \leftarrow \quad \mathbf{U} \odot \frac{\mathbf{Y}\mathbf{S}^{\mathrm{T}}}{\mathbf{U}(\mathbf{S}\mathbf{S}^{\mathrm{T}} + \lambda_* \mathbf{I})}, \qquad (10)$$

where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is the input matrix, $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_n)$ is the matrix of the corresponding codes, $\mathbf{Y} = (\mathbf{x}_1 - \mathbf{o}_1, \ldots, \mathbf{x}_n - \mathbf{o}_n)$, and $\odot$ and the fraction denote, respectively, element-wise multiplication and division. This update minimizes the objective in (6) for fixed $\mathbf{O}$ and $\mathbf{S}$, and is guaranteed to preserve the non-negativity of $\mathbf{U}$. Analogously, in the semi- and fully-supervised scenarios, $\mathbf{U}$ can be updated by minimizing the corresponding $\mathcal{L}(\mathbf{\Theta})$ using the ground-truth music accompaniment. Again using gradient descent and multiplicative updates for RPCA and RNMF respectively.

4. EXPERIMENTAL RESULTS

Dataset. We evaluate the separation performance of the proposed methods on the MIR-1K dataset [8], containing 1000 16 kHz clips extracted from 110 Chinese karaoke songs performed by 19 amateur singers (11 males and 8 females). Each clip duration ranges from 4 to 13 seconds, totaling about 133 minutes. We reserved about 23 minutes of audio sang by one male

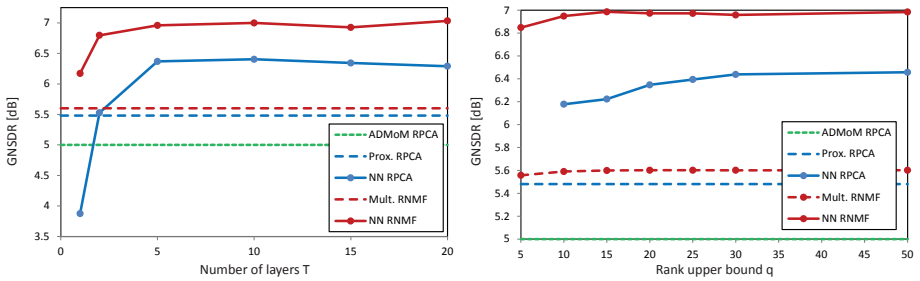

Figure 2. Performance of the supervised *NN RPCA* and *NN RNMF* on the MIR-1K dataset for different number of layers T (left, q fixed to 20), and values of the rank bound q (right, T fixed to 10). GNSDR of the recovered vocal track is used as the comparison criterion. For reference, the performance of exact RPCA and RNMF is given.

and one female singers (*abjones* and *amy*) for the purpose of training; the remaining 110 minutes of 17 singers were used for testing. The voice and the music tracks were mixed linearly with equal energy.

Evaluation. As the evaluation criteria, we used the BSS-EVAL metrics [20], which calculate the *source-to-distortion ratio* (SDR), [3] the *source-to-artifacts ratio* (SAR), and the *source-to-interference ratio* (SIR). As in [9], we computed the global normalized SDR,

$$\text{GNSDR} = \sum_{i=1}^{N} \delta_i (\text{SDR}(\hat{s}, s) - \text{SDR}(x, s)),$$

where $\hat{s}$ and s are the corresponding original and estimated voice signal, x is the mixture, δ_i is the relative duration of each of the N testing pieces. Prefix "G" indicates average sample performance,e.g. GSAR. We also computed the *signal-to-noise ratio* (SNR).

Comparison of separation methods. We evaluated the proposed *NN RPCA* and *NN RNMF* using the different training settings discussed in Section 3.1. In all our examples (except when explicitly mentioned), we used $T = 10$ layers and $q = 20$. We compare these result against three exact solvers: *ADMoM RPCA* solving (1) with $\lambda = 1/\sqrt{n}$ (as suggested in [9]) via the alternating direction method of multipliers [12], for which the code from [9] was used; *Proximal RPCA* solving (3) using the proximal method from [3], with $\lambda = \sqrt{2n}\sigma$ and $\lambda_* = \sqrt{2}\sigma$ with $\sigma = 0.3$ set following [3]; and *Multiplicative RNMF* solving (6) using the standard multiplicative algorithm.

In all experiments, the spectrogram of each mixture was computed using a window size of 1024 and a step size of 256 samples (at 16 KHz sampling rate). Training was performed using 1000 safe-guarded gradient descent iterations on a random subset of 10.000 spectral frames for training and the same amount of distinct frames for cross-validation.

[3] In this work the SDR is computed using the latest release of the BSS-EVAL code. The reported values are higher (equally for all algorithms) than the ones reported in [9], since they used the older release of that package.

Table 1 summarizes the performance of the compared methods. The best performance is achieved by the *NN RNMF* with trained dictionary. The use of the proximal RPCA algorithm allowing for inexact reconstruction of the data (thus accounting for unstructured noise) gives almost 0.5 dB improvement over [9]. The use of unsupervised training was more successful in the *NN RPCA*; however, both *NN RPCA* and *NN RNMF* outperform ADMoM RPCA.

The complexity of the proposed systems is significantly lower to the one of exact algorithms: our unoptimized Matlab code that uses GPU acceleration is capable of computing the networks about 70 faster than real time, while a preliminary implementation on iPhone 4S is online and $6 - 7$ times faster than real time (after offline training).

Parameter selection. We also evaluated the performance of the supervised RPCA and RNMF networks as a function of the two principal parameters: the number of layers T and the rank bound q, see Figure 2.

Supervised learning has a dramatic effect on the performance of the networks. With just two layers, the RPCA network already outperforms the exact RPCA algorithms; as a reference, an untrained network, with the parameters $\mathbf{W}, \mathbf{H}$, and $\mathbf{t}$ set according to Algorithm 1, requires over 15 layers to approach this performance. This phenomenon is even more pronounced in the case of RNMF. The influence of the number of layers quickly saturates; slight oscillations in the GNDSR are due to the randomization used at training.

In contrast, the effect of q is less dramatic. The networks outperform the exact algorithms already for $q = 5$ and the performance saturates for $q \geq 30$. This is radically different from the behavior of standard NMF approaches, in which setting the number of columns in the non-negative factor $\mathbf{U}$ significantly affects the performance. In fact, RNMF with $\lambda_* = 0$ as [22] yields 5.60 dB GNSDR for $q = 1$, which drops to 2.88 dB for $q = 3$ and to -2.5 dB for $q = 10$.

Supervised training settings. We evaluated the influence of the different training regimes on the perfor-

Table 2. Performance of NN RNMF on the vocal trak of *Sunrise* song. Audio files are available for download here

Method	NSDR	SNR	SAR	SIR
Ideal freq. mask	14.98	5.84	18.46	39.40
ADMoM RPCA [9]	1.61	2.99	11.13	6.60
Supervised (*MIR-1K*)	7.16	4.86	14.21	13.25
Supervised (*We are in love*)	7.85	5.47	15.35	13.59
Supervised (*Sunrise*)	10.93	5.67	16.16	19.20
Semi-supervised (*We are in love*)	7.35	4.69	11.39	20.01
Semi-supervised (*Sunrise*)	8.46	5.11	12.20	23.97

mance of the networks on Shannon Hurley's song *Sunrise*, available from `archive.org`. The song was resampled at 16 kHz and voice was artificially mixed with the guitar accompaniment with equal energies. Three distinct datasets were used for training the nets: two singers from MIR-1K used in the previous experiments; another Shannon Hurley's song *We are in love*; and the same *Sunrise*, song on which the testing was performed (given only for comparison). Supervised and semi-supervised regimes were used.

Table 2 summarizes the obtained results. RNMF networks trained using mixtures from MIR-1K outperform [9] by nearly 5.5 dB GNSDR; training on more singer-specific data (*We are in love* song) improves this result by about 0.7 dB. ; finally, training on a mixture from the same song yields over 3.5 dB improvement. We conclude that training the networks on unrelated singers and accompaniments already achieves very high performance. Semi-supervised training on the *We are in love* song yields a minor improvement over MIR-1K, and cedes 0.5 dB to the fully-supervised training. We conclude that in the absence of synchronized voice and music tracks for supervised training, semi-supervised training still produces comparable results.

5. CONCLUSION

Marrying ideas from convex optimization with multilayer neural networks, we have developed efficient architectures for real-time online single-channel separation of singing voice from musical accompaniment. Our approach achieves state-of-the-art results on the MIR-1K datasets with orders of magnitude improvement in runtime and latency. In future work, we are going to extend this framework to denoising and simultaneous separation and speaker identification.

6. REFERENCES

[1] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. In *Optimization for Machine Learning*. MIT Press, 2011.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2:183–202, March 2009.

[3] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), May 2011.

[4] J.-L. Durrieu, B. David, and G. Richard. A musically motivated mid-level representation for pitch estimation and musical audio source separation. *J. Sel. Topics Signal Processing*, 5(6):1180–1191, 2011.

[5] I. Goodfellow, Q. Le, A. Saxe, H. Lee, and A. Y. Ng. Measuring invariances in deep networks. In *NIPS*, pages 646–654. 2009.

[6] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *ICML*, pages 399–406, 2010.

[7] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun. Unsupervised learning of sparse features for scalable audio classification. In *ISMIR*, 2011.

[8] C.L. Hsu and J.S.R. Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Trans. on Audio, Speech, and Lang. Proc.*, 18(2):310–319, 2010.

[9] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *ICASSP*, 2012.

[10] K. Kavukcuoglu, M.A. Ranzato, and Y. LeCun. Fast inference in sparse coding algorithms with applications to object recognition. *arXiv:1010.3467*, 2010.

[11] Y. Li and D. L. Wang. Separation of singing voice from music accompaniment for monaural recordings. *IEEE Trans. on Audio, Speech & Lang. Proc.*, 15(4):1475–1487, 2007.

[12] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Arxiv preprint arXiv:1009.5055*, 2010.

[13] A. Liutkus, Z. Rai, R. Badeau, B. Pardo, and G. Richard. Adaptive filtering for music/voice separation exploiting the repeating musical structure. In *ICASSP*, 2012.

[14] G. Mateos and G. B. Giannakis. Robust PCA as bilinear decomposition with outlier-sparsity regularization. *arXiv.org:1111.1788*, 2011.

[15] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval. Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs. *IEEE Trans. on Audio, Speech & Lang. Proc.*, 15(5):1564–1578, 2007.

[16] M. A. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

[17] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010.

[18] B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. 2011.

[19] S. Vembu and S. Baumann. Separation of vocals from polyphonic audio recordings. In *ISMIR*, pages 337–344, 2005.

[20] E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE Trans. on Audio, Speech, and Lang. Proc.*, 14(4):1462–1469, 2006.

[21] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. In *NIPS*, pages 2496–2504. 2010.

[22] L. Zhang, Z. Chen, M. Zheng, and X. He. Robust non-negative matrix factorization. *Frontiers of Electrical and Electronic Engineering in China*, 6:192–200, 2011.