Semi-Blind Speech-Music Separation Using Sparsity and Continuity Priors

Hakan Erdogan and Emad M. Grais
Faculty of Engineering and Natural Sciences,
Sabanci University, Orhanli Tuzla, 34956,
Istanbul, Turkey.
haerdogan@sabanciuniv.edu, grais@su.sabanciuniv.edu

***Abstract*—In this paper we propose an approach for the problem of single channel source separation of speech and music signals. Our approach is based on representing each source's power spectral density using dictionaries and nonlinearly projecting the mixture signal spectrum onto the combined span of the dictionary entries. We encourage sparsity and continuity of the dictionary coefficients using penalty terms (or log-priors) in an optimization framework. We propose to use a novel coordinate descent technique for optimization, which nicely handles nonnegativity constraints and nonquadratic penalty terms. We use an adaptive Wiener filter, and spectral subtraction to reconstruct both of the sources from the mixture data after corresponding power spectral densities (PSDs) are estimated for each source. Using conventional metrics, we measure the performance of the system on simulated mixtures of single person speech and piano music sources. The results indicate that the proposed method is a promising technique for low speech-to-music ratio conditions and that sparsity and continuity priors help improve the performance of the proposed system.**

***Keywords*-single channel speech-music separation, semi-blind signal separation, sparsity.**

I. Introduction

We may encounter mixtures of speech and music audio in many places including broadcast news and other shows on TV and radio. The performance of an automatic speech recognizer quickly degrades when there is music in the background. It would be beneficial to remove music from the data for improved speech recognition. Separating speech and music from their mixture can also find application in the entertainment industry.

Blind source separation (BSS) uses minimal assumptions about the source signals, such as non-Gaussianity [1], [2]. In semi-blind source separation we assume that we have example "training" data from each source. However, this data is not transcribed or labeled in any way. Also, during testing, unlike [3] we do not impose a grammar on the speech signal since no such information is assumed available. Our goal is to reconstruct speech source signal as close as possible to the original signal and with less interference from the background music. There have been other studies on semi-blind separation of speech and music, or many musical instruments from each other [4], [5], [6], [7]. In these approaches, typically a short-time spectral dictionary of each source is developed and the mixed signal spectrum is represented as a linear combination of these dictionary entries. In [4], a non-negative sparse representation is employed and the sources are reconstructed using the Wiener filter. In [7], sparse coding with a temporal continuity objective was used. The work focuses on separating musical instruments and the continuity of the decomposition was enforced using an L_1 norm penalty. The reconstruction was done by inverse discrete Fourier transform (DFT) and overlap-add.

Our approach uses trained models of speech and music short-time spectra. It relies on non-negative sparse decomposition taking into account temporal continuity of the decomposition coefficients. The novelty in this paper is in three different aspects. First, we use both continuity and sparsity information for speech/music separation. A similar approach has been used in the context of musical instrument separation [7], but not for speech and music separation. Secondly, we introduce a new coordinate descent algorithm which is easy to implement and simple to understand for estimating the source power spectral densities. This algorithm also easily handles nonnegativity constraint. Earlier work used different algorithms based on Lagrange functionals and gradient descent [4] or a multiplicative update followed by gradient descent [7]. Thirdly, we explore both Wiener and spectral subtraction filters for reconstruction. Earlier work focused on Wiener filter and time-frequency shrinkage [5]. Using the performance criteria defined in [8], we get similar or better results as compared to the previous works [4], [5].

The remainder of this paper is organized as follows. In section II, we give a mathematical description of the problem. In section III, we show how we estimate the power spectral densities for the mixed sources. We give a brief explanation about some of the standard algorithms that are used in separation of mixed signals in section IV. In the remaining sections, we present our observations and the results of our experiments.

II. Problem formulation

Single channel signal separation problem can be defined as follows. Assume we are given an observed signal $y(t)$, which is the mixture of two sources $x(t)$ and $m(t)$. The source separation problem consists of finding estimates for $x(t)$ and $m(t)$ from $y(t)$. Algorithms presented in this paper are applied in the short time Fourier transform (STFT) domain. Denote by $Y(t,f)$ the STFT of $y(t)$, where t represents the frame index and f the frequency-index. Due to linearity of the STFT, we have:

$$Y(t,f) = X(t,f) + M(t,f). \tag{1}$$

Assuming independence of the sources, we can write the power spectral density (PSD) of the measured signal as the sum of source signal PSDs.

$$\sigma_y^2(t,f) = \sigma_x^2(t,f) + \sigma_m^2(t,f), \tag{2}$$

where $\sigma_y^2(t,f) = E(|Y(t,f)|^2)$. Here $\sigma_x^2(t,f)$ and $\sigma_m^2(t,f)$ are the unknown power spectral densities (PSDs), and need to be estimated using measured data and training speech and music spectra. The PSD for the measured signal $y(t)$ is estimated using the periodogram by taking the squared magnitude of the DFT of the windowed signal.

Once good estimates of PSDs for speech and music are determined, we can get an estimate for $x(t)$, say $\hat{x}(t)$ using one of Wiener, or spectral subtraction filters, which we explain in section IV.

III. Estimating Source Power Spectral Densities

In this section, we show how we build our codebook, and give mathematical formulation for using sparsity and continuity priors for decomposing the observed mixed signal into linear combination of the training codebook entries, and find the values of coefficients for this linear combination.

A. Learning the codebooks

We extract normalized PSDs from training data for each source type. We collect these PSDs as representative PSDs for each source. We need a way to obtain a dictionary from these PSDs. We used the k-means algorithm on the collections of PSDs for each source type. This gives us a vector quantization type of codebook for each source. We set the size of the dictionaries for each source arbitrarily. However, one can optimize the dictionary size as well. The optimal size of the dictionary may depend on the amount of training data.

B. Using codebooks for decomposition

We will write the PSDs of speech and music signals as a linear combination of the entries in a codebook. So, we will represent them as follows:

$$\sigma_x^2(t,f) \approx \sum_{k=1}^{d_x} \alpha_k(t) v_k(f), \tag{3}$$

$$\sigma_m^2(t,f) \approx \sum_{k=d_x+1}^{d} \alpha_k(t) v_k(f). \tag{4}$$

Here, $v_k(f)$'s represent codebook spectra for the speech and the music signals. In practice, we sample the frequency values so that each $\boldsymbol{v}_k$ can be represented as a vector of spectral entries. There are d_x codebook entries for speech and d_m such entries for music. The coefficients α_k are the nonnegative coefficients that multiply each codebook entry. After one learns the dictionary for each source type, one needs to estimate the coefficient vector

$$\boldsymbol{\alpha}\,(t) = [\alpha_1(t), ..., \alpha_d(t)]^T \tag{5}$$

of dimension $d = d_x + d_m$ for each time frame t. Note that the dictionary entries are neither orthogonal to each other nor form an orthogonal subspace for speech and music. In fact, we have overlapping subspaces which are difficult to separate.

C. Objective function

In order to estimate the coefficient vector $\boldsymbol{\alpha}\,(t)$, we propose using a penalized least squares approach. The penalty terms (or log-priors) have two purposes: sparsity and continuity. The data fidelity term measures the deviation from the measured PSD. For the purposes of optimization, we will represent the measured spectrum $\sigma_y^2(f)$ as the vector $\boldsymbol{y}$ of spectral entries. The optimization problem is defined as $\hat{\boldsymbol{\alpha}} = \text{argmin}\phi(\boldsymbol{\alpha})$ subject to $\alpha_k \geq 0, \quad k = 1, \ldots, d$ where

$$\phi(\boldsymbol{\alpha}) = \left\| \boldsymbol{y} - \sum_{k=1}^{d} \alpha_k \boldsymbol{v}_k \right\|_2^2 + \beta \left\| \boldsymbol{\alpha} - \boldsymbol{\alpha}^{\text{prev}} \right\|_2^2 + \gamma \sum_{k=1}^{d} |\boldsymbol{\alpha}_k|^e . \tag{6}$$

The length of each spectral vector $\boldsymbol{y}$ and $\boldsymbol{v}_k$ is $1 + N_{\text{fft}}/2$. We take only positive frequency indices since the rest can be obtained using conjugate symmetry. The term with β is the discontinuity penalty term. It penalizes the departure of $\boldsymbol{\alpha}$ from $\boldsymbol{\alpha}^{\text{prev}}$, the vector found in the previous frame. The term with γ is the term that enforces sparsity. It encourages sparseness of $\boldsymbol{\alpha}$ vector by using an L_1 type of norm with a value of e close to 1. The reason for nonnegativity of α_k is that the PSD should always be nonnegative and if the α_k are nonnegative, that will make sure that the obtained PSDs are nonnegative.

The model for the PSDs is obtained as a linear combination of dictionary entries for each type of source. One can envision the coefficient vector $\boldsymbol{\alpha}\,(t)$ as being a sparse vector which changes slowly in time. The reason for sparseness is clear since at one instant of time, one expects to find a single dictionary entry active for each source. Thus, we should expect $\boldsymbol{\alpha}\,(t)$ to be sparse. The reason for slow changing $\boldsymbol{\alpha}\,(t)$ is the fact that neighboring frames are highly correlated, so we expect $\boldsymbol{\alpha}\,(t)$ to be similar to $\boldsymbol{\alpha}\,(t-1)$.

D. Coordinate descent optimization

Due to the nonnegativity constraint and the nonquadratic nature of the sparsity prior, it is not easy to optimize the objective function directly. So, we propose using coordinate descent optimization where one parameter at a time is optimized while the others are kept constant. Enforcing nonnegativity is also easy in coordinate descent, namely if one parameter becomes negative, we just set it back to 0.

The algorithm works as follows. We have global iterations and within each global iteration we iterate sequentially through each parameter $i = 1, \ldots, d$. We calculate the partial derivative of the objective function with respect to α_i:

$$\frac{\partial\phi}{\partial\alpha_i} = -2\boldsymbol{v}_i^T\boldsymbol{y} + (\boldsymbol{v}_i^T \sum_{j=1}^{k} 2\alpha_j \boldsymbol{v}_j) + 2\beta(\alpha_i - \alpha_i^{\text{prev}}) + \gamma e \left|\alpha_i\right|^{e-1} . \tag{7}$$

For $e = 2$, we can set the partial derivative to zero and solve for the minimizer from there, and we obtain the following update:

$$\alpha_i^* = \left[\frac{1}{\boldsymbol{v}_i^T\boldsymbol{v}_i + \beta + \gamma} \left(\beta\alpha_i^{\text{prev}} + \boldsymbol{v}_i^T(\boldsymbol{y} - \sum_{l \neq i} \alpha_l \boldsymbol{v}_l) \right) \right]_+ , \tag{8}$$

where $[x]_+ = \max(0, x)$. For $e \neq 2$ we perform a few iterations (typically three) of Newtons update. For that, we need to find the second partial derivative with respect to α_i:

$$h_i = \frac{\partial^2}{\partial\alpha_i^2}(\phi(\alpha)) = 2\boldsymbol{v}_i^T\boldsymbol{v}_i + 2\beta + \gamma e(e-1)\left|\alpha_i\right|^{e-2} . \tag{9}$$

Then, the update is as follows:

$$\alpha_i^* = \left[\alpha_i - h_i^{-1} \frac{\partial}{\partial\alpha_i}\phi(\alpha) \right]_+ . \tag{10}$$

Note that, the second derivative becomes infinite for $\alpha_i = 0$. In that case, while calculating the second derivative, we set α_i to be a very small value. In practice, we stop global iterations when the change in $\boldsymbol{\alpha}$ is negligible.

IV. Reconstruction of the source signals using zero-phase filters

Once good estimates of PSDs for speech and music are determined, we can obtain an estimate for the STFT of the speech source signal $x(t)$ using filtering. We use time-varying zero-phase filters for reconstructing the source signal, namely Wiener and spectral subtraction filters. We also experimented with time-frequency shrinkage method [5] with worse results.

These filters modify the magnitude spectrum of the observed signal and keep the phase unchanged. Keeping the phase unchanged is important for the performance measures that we used since these measures assume no phase change occurs between the original and estimated speech signals.

A. Wiener filter

Wiener filter, which is optimal in the mean-squared sense, is given by:

$$H_{\text{WI}}(t,f) = \frac{\sigma_x^2(t,f)}{\sigma_x^2(t,f) + \sigma_m^2(t,f)},$$

where $\sigma_x^2(t,f)$ and $\sigma_m^2(t,f)$ are found using equations (3) and (4).

The STFT estimate can be found by $\hat{X}(t,f) = H_{\text{WI}}(t,f)Y(t,f)$. We reconstruct the speech signal $\hat{x}(t)$ by taking the inverse STFT (that is inverse DFT followed by overlap-add).

B. Spectral subtraction

Another popular method for signal estimation is spectral subtraction [9]. The expression for the frequency-domain filter is given by:

$$H_{\text{SS}}(t,f) = \left(1 - k\frac{\sigma_m^a(t,f)}{\sigma_y^a(t,f)}\right)^{1/a}, \qquad (11)$$

where k controls the degree of subtraction, and a controls whether magnitude or power spectral subtraction is used. The combination of the parameters k and a thus controls the amount of noise reduction. In this work we used $k = 1$, $a = 2$ corresponding to power spectral subtraction. Similar to the Wiener filter, we can obtain an estimate for $\hat{x}(t)$ using spectral subtraction filter on the STFT of the measured signal and performing inverse STFT.

V. Experiments and discussion

We applied the proposed algorithm on simulated mixtures of speech and music data at 16kHz sampling rate. For training speech data, we used 540 short utterances from a single speaker. We left out 20 utterances for testing. For piano music data, we downloaded piano music from piano society web site [10]. We used 38 pieces from different composers but from a single artist for training and left out one piece for the testing stage. The PSD dictionaries for speech and music data were trained using the k-means algorithm on each source data and 64 dictionary entries were obtained for each source. So $d_x = 64, d_m = 64, d = 128$. For the STFT, the frame rate was 10 ms, the window size was 30 ms, a Hamming window was used and the FFT was taken at 512 points. The test data was formed by adding random portions of the test music file to the 20 speech utterance files at different speech to music ratio (SMR) values in dB. The audio power levels of each file were found using the "audio voltmeter" program from the G.191 ITU-T STL software suite [11]. For each SMR value, we obtain 20 test utterances this way.

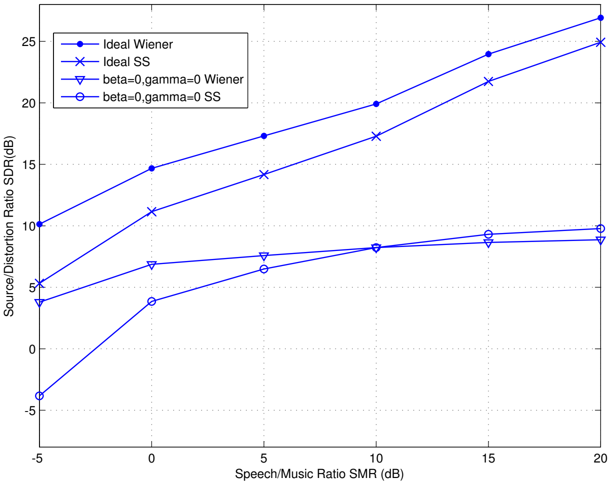

Figure 1. Source/distortion ratio SDR without using continuity and sparsity priors, using ideal Wiener filter,and ideal SS filter. Ideal means, we know the exact power spectral densities $\sigma_x^2(t,f)$ and $\sigma_m^2(t,f)$.

Performance measurement of the separation algorithms were done using metrics introduced in [8]. Projection of the predicted signal onto the original speech signal is termed as the target signal. Source distortion ratio (SDR) is defined as the ratio of the target energy to all errors in the reconstruction. Source interference ratio (SIR) is defined as the ratio of the target energy to the part of the error due to the music signal only.

In Figure 1, we show the results of applying Wiener and spectral subtraction filters for the ideal case of knowing the source PSDs exactly. These results indicate an upper bound on the performance using these filters. In practice of course, we need to estimate the source PSDs and use them in deriving the adaptive filters. Also in Figure 1, we show the baseline results for source speech signal reconstruction when no priors ($\gamma = \beta = 0$) are used. The results show that, for the ideal case, Wiener filter works best across all SMR values. Baseline results indicate that Wiener filter is better for $SMR < 10$ where spectral subtraction is slightly better for larger SMR values. Due to these results, we used Wiener filter for the rest of the experiments where we explore the effects of sparsity and continuity priors.

We experimented with applying different prior parameters, γ and β, for our reconstructions. We fixed the value of $e = 1.01$. The SDR and SIR results are presented in Table I and Table II respectively.

When no continuity prior was used ($\beta = 0$), we obtained improved SDR results using γ values of 10 and 10^2. $\gamma = 10^2$ seems better for larger SMR values. However, the SIR value is reduced as compared to no prior information as shown in Table II. So, we get improvement in SDR at the expense of a reduction in SIR. So, using sparsity priors reduces artifacts ($r(t)$) dramatically but slightly increases music interference in the estimated speech signal.

Next, to find the best value of the continuity prior, we set $\gamma = 0$

Table I
SOURCE/DISTORTION RATIO IN dB USING WIENER FILTER.

SMR	β =0					γ =0				γ =10^{-2}	γ =10	γ =100
dB	γ =0	γ =1	γ =10	γ =10^2	γ =10^3	β =10^{-3}	β =10^{-2}	β =0.1	β =1	β =10^{-2}	β =10^{-2}	β =10^{-2}
-5	3.78	3.79	3.91	2.52	-1.93	3.78	3.78	3.86	3.57	3.78	3.90	2.51
0	6.87	6.92	7.07	6.01	1.95	6.87	6.9	6.9	6.08	6.91	7.10	6.05
5	7.58	7.66	8.07	7.94	3.87	7.58	7.62	7.58	6.74	7.6	8.06	7.81
10	8.23	8.32	8.9	9.86	5.59	8.23	8.26	8.19	7.23	8.28	8.9	9.82
15	8.65	8.82	9.77	11.51	5.96	8.64	8.67	8.58	7.52	8.67	9.77	11.35
20	8.86	9.04	10.37	12.18	6.39	8.87	8.89	8.77	7.68	8.91	10.39	12.20

Table II
SOURCE/INTERFERENCE RATIO IN dB USING WIENER FILTER.

SMR	β =0					γ =0				γ =10^{-2}	γ =10	γ =100
dB	γ =0	γ =1	γ =10	γ =10^2	γ =10^3	β =10^{-3}	β =10^{-2}	β =0.1	β =1	β =10^{-2}	β =10^{-2}	β =10^{-2}
-5	19.57	19.28	18.6	14.88	4.34	19.55	19.47	19.04	15.44	19.49	18.55	14.86
0	22.34	21.76	18.42	13.19	8	22.35	22.37	22.06	19.74	22.34	18.4	13.4
5	23.44	22.19	18.33	13.94	10.64	23.42	23.49	23.25	21.41	23.49	18.31	13.85
10	24.8	22.89	18.41	16.81	15.69	24.8	24.8	24.58	23.03	24.75	18.45	16.65
15	28.03	24.27	20.97	21.62	21.29	28.02	28.07	27.85	26.18	28.04	20.91	21.66
20	30.01	24.92	24.36	25.74	25.74	29.98	30.02	29.8	28.43	29.93	24.31	25.59

and experiment with different values of β. We obtain small but consistent improvements in SDR and SIR values when using $\beta = 10^{-2}$ as shown in Tables I and II.

Finally, we combine the most promising values for γ and β to obtain combined performance results using both continuity and sparsity priors. We can improve the SDR value about 4 dB in some cases using the prior information. The SIR value can be also improved slightly around 0.05 dB using prior information.

VI. CONCLUSION

In this work, we studied semi-blind speech-music separation using sparsity and continuity priors. We analyzed how much we can improve the separation process by changing the values of the sparsity and continuity priors parameters. In addition, we experimented with applying different separation algorithms, like Wiener filter, and spectral subtraction to mixture signals with different speech to music power ratios (SMR). We introduced a new and easy to implement algorithm for source separation. We obtained some improvement in speech distortion measure by using sparsity information. We have observed that the continuity prior improved the performance only slightly. This shows that neighboring frames may not have highly correlated parameters. In the future, we may obtain better improvement by considering the probability of one set of parameters in one frame following another set of parameters in the following frame in the training data (using n-gram statistics).

REFERENCES

[1] A. J. Bell and T. J. Sejnowski, “An information maximization approach to blind separation and blind deconvolution,” *Neural Computation*, vol. 7, no. 6, pp. 1129–59, 1995.

[2] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley and Sons, 2001.

[3] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, “Super-human multi-talker speech recognition: A graphical modeling approach,” *Computer Speech and Language*, vol. 24, no. 1, pp. 45–66, Jan. 2010.

[4] L. Benaroya, L. M. Donagh, F. Bimbot, and R. Gribonval, “Non negative sparse representation for Wiener based source separation with a single sensor,” in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, 2003.

[5] L. Benaroya, F. Bimbot, G. Gravier, and G. Gribonval, “Experiments in audio source separation with one sensor for robust speech recognition,” *Speech Communication*, vol. 48, no. 7, pp. 848–54, Jul. 2006.

[6] R. Blouet, G. Rapaport, and C. Fevott, “Evaluation of several strategies for single sensor speech/music separation,” in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, 2008.

[7] T. Virtanen, “Sound source separation using sparse coding with temporal continuity objective,” in *ICMC 2003*, 2003.

[8] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Tr. Acoust. Sp. Sig. Proc.*, vol. 14, no. 4, pp. 1462–69, Jul. 2006.

[9] S. F. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Tr. Acoust. Sp. Sig. Proc.*, vol. 27, no. 2, pp. 113–20, 1979.

[10] URL, “http://pianosociety.com,” 2009.

[11] ——, “http://www.itu.int/rec/T-REC-G.191/en,” 2009.