Sound Source Separation Using Sparse Coding with Temporal Continuity Objective

Tuomas Virtanen
Tampere University of Technology, Institute of Signal Processing
email: tuomas.virtanen@tut.fi

Abstract

A data-adaptive sound source separation system is presented, which is able to extract meaningful sources from polyphonic real-world music signals. The system is based on the assumption of non-negative sparse sources which have constant spectra with time-varying gain. Temporal continuity objective is proposed as an improvement to the existing techniques. The objective increases the robustness of estimation and perceptual quality of synthesized signals. An algorithm is presented for the estimation of sources. Quantitative results are shown for a drum transcription application, which is able to transcribe 66% of the bass and snare drum hits from synthesized MIDI signals. Separation demonstrations for polyphonic real-world music signals can be found at http://www.cs.tut.fi/~tuomasv/demopage.html.

1 Introduction

Sound source separation has several applications, for example editing, analysis and automatic transcription of music. While humans are able "hear out" sounds from complex mixtures, computer modelling of this function has proven to be very difficult.

Sound source separation systems can be roughly divided into two categories: in data-adaptive techniques there is no knowledge of the sources, so that they are estimated from the data. Model-based separation systems have a parametric or statistical model of the sources, and instead of estimating the signals itself, the parameters are estimated. Both approaches have their advantages and disadvantages. The fundamental idea of estimating sources from the data is very appealing, but for polyphonic real-world signals the performance of for example independent component analysis (ICA) alone is usually poor. To increase the robustness of a data-adaptive system one can place restrictions for the sources, which moves the system towards a model-based system. It can be assumed that a good separation system has the good sides of both data-adaptive and model-based techniques, being able to adapt to the input data while preserving the robustness of model-based methods.

For one-channel audio signals the usual approach of ICA systems is to project the time-domain input signal into the frequency domain and assume that the spectra of sources is constant from frame to frame. In earlier papers the estimation of mixed signals has been done for example by assuming independence and orthogonality of source spectra (Casey and Westner 2000) and sparseness of the sources (Plumbley, *et al*. 2001). The sparseness of sources means that the sources are inactive most of the time

Temporal coherence is one of the main features that human auditory system uses in grouping spectral components (Bregman 1990). It has been one of the most difficult phenomena to model computationally. The usual approach is to estimate parameters individually in each frame and connect the frames so that the temporal continuity is maximized. In this paper, a data-adaptive separation system is proposed in which the temporal continuity between frames is achieved by using a cost function which favors temporally smooth signals, so that the continuity objective is used already in the core algorithm instead of postprocessing. The sources are assumed to be sparse and non-negative, and their spectra constant with time-varying gain. Non-negativity of spectra and gains is a necessary assumption since the estimation is done using power spectra.

For sparse non-negative sources there does not exist an unmixing matrix, multiplying by which the sources could be obtained. Instead, a specific separation algorithm was developed, which finds the optimal sources under the assumptions made. The algorithm has been designed using ideas taken from non-negative matrix factorization (Lee and Seung), which was combined with sparse coding by Hoyer (2002). The separation algorithm was implemented in Matlab and tested on different kinds of real-world music signals. The algorithm is able to extract at least some sources from most real-world music signals. The experiments show that the temporal continuity assumption increases the robustness of separation.

The experiments suggested that one application area could be separation of drums. Based on the proposed separation algorithm, an automatic drum transcription system was designed. Simulations were carried out to monitor the behaviour of the system. The results show that the system can produce applicable results in sound source separation and automatic drum transcription.

2 The Separation Algorithm

The block diagram of the separation system is illustrated in Figure 1. At first, the time-domain input signal $x(t)$ is

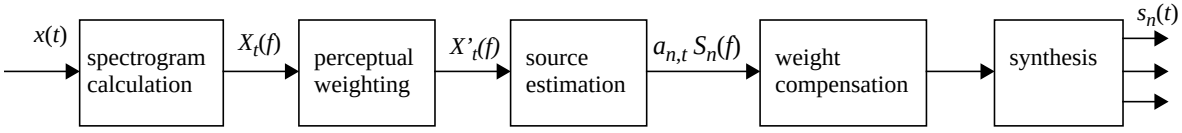

Figure 1. *Block diagram of the separation system.*

divided into frames and the power spectrum is calculated within each frame using the discrete Fourier transform (DFT). The spectrogram is weighted according to the frequency response of the human auditory system, as explained in the Section 2.3. Once the sources are estimated, the weighting is compensated by inverse weighting. Finally, the estimated sources can be synthesized separately.

As explained in the Introduction, the system is based on the assumption that the spectral shape of sources is constant over time, only the gain being time-varying. Therefore, each sound source n is characterized by its power spectrum $S_n(f)$, and the time-varying gain $a_{t,n}$. The number of sound sources is N, and the sources are assumed to sum linearly, so that the model of the input signal can be written as:

$$X_t(f) = \sum_{n=1}^{N} a_{t,n} S_n(f) + E_t(f), \quad (1)$$

where $X_t(f)$ is the power spectrum of the input signal in frame t, $a_{t,n}$ is the gain of the n^{th} source in frame t and $S_t(f)$ is the power spectrum of the source n. $E_t(f)$ is the error spectrum.

In this paper, notation $(X)_{i,j}$ is used to refer to the element (i,j) of matrix X. Let the number of frames be T and the number of discrete frequencies be F, so that the input spectra and the error spectra can be denoted by T by F matrices: $(X)_{t,f} = X_t(f)$. The spectra of the sources is denoted by a N by F matrix S: $(S)_{n,f} = S_n(f)$ and the time-varying gains by a T by N matrix A: $(A)_{t,f} = a_{t,n}$. Now Equation 1 can be written as

$$X = AS + E \quad (2)$$

The matrix X is called input data matrix, A mixing matrix, and S source matrix. Since the input data, gains and power spectra of the sources are non-negative, also the elements of corresponding matrices are restricted to non-negative values:

$$(X)_{t,f} \geq 0, (S)_{n,f} \geq 0, (A)_{t,n} \geq 0, \forall t, f, n \quad (3)$$

2.1. Cost functions

The source matrix S and mixing matrix A are unknown. The system takes the power spectrogram X as an input. The estimation of sources and mixing matrix is based on minimization of cost function, which minimizes the reconstruction error while preserving the sparseness and temporal continuity assumptions. The cost function is given by:

$$e(A,S) = w^{(g)} g(A,S) + w^{(h)} h(A) + w^{(c)} c(A), \quad (4)$$

where the functions g, h, and c and scalars $w^{(g)}$, $w^{(h)}$, and $w^{(c)}$ are the terms and weights for optimization of reconstruction, sparseness, and temporal continuity, respectively.

The reconstruction error is minimized using the cost function

$$g(A,S) = \frac{1}{2}||X - AS||^2, \quad (5)$$

where the norm operator is the sum of squared elements.

It is assumed that the sound sources are inactive most of the time. In our model this means that the elements of A have a high probability of being zero. Some sparse coding systems assume sparseness of the source matrix, but since the mixing and source matrices can swapped by $X = AS \Leftrightarrow X^T = S^T A^T$, the same optimization methods can be used for both objectives. Sparseness in achieved by minimizing a cost term (Hoyer 2002):

$$h(A) = \sum_{t=1}^{T} \sum_{n=1}^{N} |a_{t,n}| \quad (6)$$

Temporal continuity is achieved by minimizing cost term

$$c(A) = \frac{1}{2} \sum_{t=1}^{T} \sum_{n=1}^{N} |a_{t-1,n} - a_{t,n}| \quad (7)$$

Absolute value of the gain difference between frames is used instead of a squared difference, because the absolute value operator preserves rapid changes better than the squared sum. For example, a gain transition from zero to a constant level is quite common in music signals. The square operator tends to smooth these transitions, since the optimal parameters for the square operator are the ones which have equal difference in each frame. The absolute value operator is better since all the transitions which increase the gain monotonically are equal in this case.

The cost function is not well-defined yet, since for any $\hat{S}$ and $\hat{A}$ the error functions h and c can be minimized by selecting $A = \alpha\hat{A}$, $S = \hat{S}/\alpha$, and $\alpha \to 0^+$. The cost function is made well-defined by fixing the norm of each column equal to unity:

$$\sum_{t=1}^{T} a_{t,n}^2 = 1, \text{ for all } n \in [1, N]. \quad (8)$$

Weights $w^{(g)}$, $w^{(h)}$, and $w^{(c)}$ are used to balance the cost functions.

In the optimization algorithm, the gradient of the cost function with respect to the mixing matrix A is needed. The gradient is given by:

$$\nabla e = w^{(g)} \nabla g + w^{(h)} \nabla h + w^{(c)} \nabla c, \quad (9)$$

where the gradients ∇g, ∇h, and ∇c are:

$$\nabla g = (AS - X)S^T$$
$$(\nabla h)_{t,n} = 1$$
$$(\nabla c)_{t,n} = \begin{cases} -1, & a_{t,n} < a_{t-1,n} \wedge a_{t,n} < a_{t+1,n} \\ 1, & a_{t,n} > a_{t-1,n} \wedge a_{t,n} > a_{t+1,n} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

2.2. Optimization algorithm

The number of desired sources N has to be set by hand. The objective is to find A and S which minimize the cost function of Equation 5, with the restrictions given in Equations 3 and 8.

In optimization, a combination of multiplicative step and projected gradient descent seemed to be the most efficient. The algorithm has similarities to the one proposed by Hoyer (2002), except that A is assumed sparse instead of S, line search is used to estimate the optimal step size, and the scaling of A is compensated by rescaling S.

The algorithm is the following:

1. Initialize A^0 and S^0 with white noise. An absolute value is taken element-wise so that Equation 3 holds and the columns of A are scaled to unity norm so that the Equation 8 holds.

2. Update S using a multiplicative rule (Lee and Seung):

$$S^{k+1} = S^k .* (A^T X) ./ (A^T A S), \quad (11)$$

where .* and ./ are element-wise multiplication and division, respectively.

3. Update A using the steepest descend method:

$$A^{k+1} = A^k - \mu^k \nabla e^k, \quad (12)$$

where ∇e^k is the gradient of e with respect to A at point (A^k, S^{k+1}) and the optimal step size $\mu^k > 0$ is estimated using a line search, in which the effect of steps 4 and 5 is taken into account.

4. Set all negative elements of A^{k+1} to zero

5. Scale the columns of A^{k+1} to unity norm. Compensate by rescaling the rows of S^{k+1} so that the product $A^{k+1}S^{k+1}$ does not change.

Repeat steps 2 to 5 until a stopping criterion is reached. In our implementation the weights of cost functions are chosen to compensate the scale of the input signal X and the number of sources, so a fixed tolerance can be used: if the decrease of the cost function is smaller than the fixed tolerance, the iteration is terminated.

The cost function is non-increasing at each iteration. Update rule for S has been proven to be non-increasing by Lee and Seung (2001) and in the steepest descend algorithm the step size is chosen so that the cost function is non-increasing, steps 3-5 taken into account.

Naturally the convergence of the algorithm depends on the signal content and parameters. In our simulations the typical length of signals was about 20 seconds and the number of separated components ranged from two to ten. Frame size of 2048 samples and overlap factor 0.5 was used, so that the typical number of frames was about 900. The DFT length was same as the frame size. The frequency range was limited to 5kHz, so that only 233 frequency lines of the spectrum were used. With these parameters the algorithm takes about 50 to 500 iterations to reach the stopping criterion, which takes a couple of minutes on a regular PC.

2.3. Perceptual weighting

The described cost function g in Equation 5 does not take into account the characteristics of human auditory system. Music signals have most of their energy on low frequencies and the described separation algorithm without perceptual weighting tends to model only the lowest frequency components.

The frequency response from outer to inner ear can be considered to be signal-independent. Instead of calculating the norm between the input data X and estimate AS, the terms should be weighted by the frequency response of the ear. This can be implemented using a matrix multiplication:

$$X' = XW, \tag{13}$$

where W is a F-by-F diagonal matrix in which the diagonal element $(W)_{f,f}$ corresponds to the response at frequency f. As the response is the same for the input data and modelled data, the error function can now be written as:

$$g'(A, S) = ||XW - ASW||^2. \tag{14}$$

Since the operation is linear, it can be implemented by preprocessing and postprocessing steps. In the preprocessing step the input data matrix X is multiplied by W, which is compensated in the postprocessing step by multiplying separated sources S by W^{-1}. Using this procedure W does not need to be taken into account in the optimization algorithm.

2.4. Synthesis

Once the components have been separated from the signal, they can be synthesized separately. The spectrogram of a component n is given by:

$$(X)^n_{t,f} = (A)_{t,n}(S)_{n,f} \quad . \tag{15}$$

A time-domain signal can be synthesized using inverse-DFT and overlap-add. We tried to create a random initial phase, which was updated from frame to frame as an integral of frequency. However, best results were obtained by storing the phase of the original mixed spectogram and by using that for every separated component.

3 Multiple Components per Source

Unlike assumed in the signal model, the power spectra of natural sounds is not usually constant over time. A time-varying spectrogram can be modelled as a weighted sum of several components. The components are analysed using the described algorithm, and then clustered into sound sources. This kind of approach has been earlier used by Casey and Westner (2000). In clustering they used the symmetric Kullback-Leibler (KL) distance between the probability functions of the component spectra.

Our approach is to use the independence of time-varying gains in clustering the components. The classic definition of independence is that the joint density of variables is the same as the product of marginal densities. A measure for independence is the KL divergence between the joint density and product of marginal densities. The densities are estimated using the histograms of time-varying gains and the KL divergence is used a distance measure in clustering.

In real-world polyphonic signals the number of sources is usually quite large. Computationally practical number of source components is pretty low, less than ten, so that the true amount of sources is usually larger than the number of separated components. Therefore, the clustering is needed only if the number of sources in the input data is low. This kind of data is for example signals which contain only the drum track. For these signals the clustering worked well, being able to produce better sound quality than with only one component per source.

4 Application to Automatic Drum Transcription

The presented separation algorithm is able to separate different kinds of signals, the results depending on the signal. For most polyphonic signals one of the separated components is usually a drum sound. To demonstrate the performance of the separation algorithm, an automatic drum transcription system was implemented on the basis of the separation system. The transcription part was kept very simple so that the effect of the separation system could be studied as well as possible.

Since the bass and snare drums are present in most pieces of popular music, the transcription was evaluated using only these. Another reason is their suitability for the separation algorithm: bass and snare occur usually often and their energy is large enough so that the algorithm is able to adapt to their spectra. Hi-hats, which may occur even more often, have much weaker energy and their separation proved to be a more difficult task.

The transcription procedure is the following:

1. The spectrogram of the time-domain input signal is calculated. The proposed algorithm is used to separate seven most prominent components. This was found to be a good choice for the number of sources.

2. Find bass and snare sounds among the separated com-

ponents. This is done by calculating a distance between separated spectrum and the template spectra of bass and snare drums. The templates were automatically selected from the separated components of test signals during the transcription. The distance between a separated spectrum $S_n(f)$ and a template spectrum $R_m(f)$ is given by:

$$d(n,m) = \sum_{f=1}^{F} \left(R_m(f) \log \left| \frac{S_n(f)+\varepsilon}{R_m(f)+\varepsilon} \right| \right), \qquad (16)$$

where n is the index of extracted component, and $m \in \{1, 2\}$ is the index of either bass or snare drum template. ε is a small positive constant which is used to make the logarithm robust for small spectrum values.

3. Detect onsets of the found bass and snare components. The detection is based on the time-varying gains of the components. A frame contains an onset, if there is a large positive change in the gain, and no large gain in the preceding frame:

$$T_n = \{t; (a_{n,t} - a_{n,t-1}) < \beta \underline{a}_n \wedge (a_{n,t-1} - a_{n,t-2}) \geq \beta \underline{a}_n\} \qquad (17)$$

where T_n is the set of onset frames of component n, and $\underline{a}_n$ is the maximum positive change between frames:

$$\underline{a}_n = \max_t \{a_{n,t} - a_{n,t-1}\}. \qquad (18)$$

β is a threshold of detection between zero and unity. For test signals the optimum value of β was about 0.2.

4. The onset frame indices t are transformed into onset times. The result of transcription is onset times for bass and snare drum.

The transcription system was tested using signals which were synthesized from MIDI. Synthesized signals were used, because the correct drum score could be obtained from the original MIDI file, and no time-consuming annotation was needed.

The test set was 100 signals which were randomly selected 20 second excerpts from a collection of 279 General MIDI (GM) files, which were mostly Western popular music. Excerpts which contained less than ten bass or snare drum hits were rejected. The signals were synthesized using the Timidity software synthesizer. Half of the signals were used in the optimization of the parameters, and half to evaluate the performance of the transcription system. The original locations of bass and snare drum hits were stored to allow evaluation of the transcription. There is two bass drums and two snare drums in the GM drum kit, but no distinction was done between those, so that there was only one correct bass drum track and one correct snare drum track.

The signals were transcribed using the described system. The transcription was evaluated using the following procedure: for each bass/snare hit in the original file, transcribed bass/snare hits are sought which are at most 32 ms distance from the original hit. The hits for which a pair can be found are counted as correct transcriptions. Original drum hits for which pair is not found are counted as deletions, and transcribed hits for which pair is not found are counted as insertions. Once this evaluation has been done for all the signals, the overall error rate z is given by:

$$z = (N_d + N_i)/(N_c + N_d + N_i), \qquad (19)$$

where N_c is the number of correct transcriptions, N_d is the number of deletions and N_i is the number of insertions. The threshold 32 ms is about 70 percent of the frame length of the separation system. The frame length sets the output resolution for the transcription system. The maximum allowed error 32 ms is usually audible in rhythm tracks. To get more accurate timing, one should for example perform onset analysis on the original time-domain signal (Klapuri 1999).

The number of bass and snare hits in the 50 test signals was 2784. The number of correct transcriptions was 2163, number of insertions 489 and number of deletions 621, so that the error rate was 34%. The error rate for bass notes only was 27% and for snare notes only 43%. The histograms of error rates of each test signal for bass and snare are illustrated in Figure 2. For snares the error rates range more evenly, while for basses the transcription is perfect for most of the signals. When a more strict 20 ms threshold is used in the review, the overall error rate was still 37%.

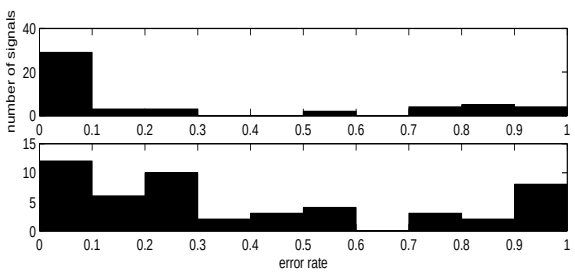

Figure 2. *Histograms of the error rates of individual signals for bass (upper plot) and snare (lower plot) transcription.*

5 Conclusions

The presented data-adaptive sound source separation system is able to extract meaningful sound sources from real-world polyphonic music signals. The proposed temporal continuity objective enhances the robustness of the system and increases the perceptual quality of separated signals. An optimization algorithm was presented to find the source signals using the assumptions made. An automatic drum transcription system was implemented on the basis of the separation algorithm. Future work includes automatic selection of the number of sources and application of the algorithm to different areas of music analysis.

References

Bregman, A.S. “Auditory Scene Analysis: The Perceptual Organization of Sound.” MIT Press, 1990.

Casey, M.A. & A. Westner. “Separation of Mixed Audio Sources By Independent Subspace Analysis.” International Computer Music Conference, 2000.

Hoyer, P.O. “Non-negative Sparse Coding.” IEEE Workshop on Neural Networks for Signal Processing, 2002.

Klapuri, A. “Sound Onset Detection by Applying Psychoacoustic Knowledge.” IEEE International Conference on Acoustics, Speech and Signal Processing, 1999.

Lee, D.D. and H.S. Seung. “Algorithms for Non-negative Matrix Factorization.” in Advances in Neural Information Processing, vol. 13. MIT Press, 2001.

Plumbley, M. D., S. A. Abdallah, J. P. Bello, M. E. Davies, J. Klingseisen, G. Monti and M. B. Sandler. “ICA and Related Models Applied to Audio Analysis and Separation.” In Proceedings of the Fourth International ICSC Symposium on Soft Computing and Intelligent Systems for Industry, Paisley, Scotland, United Kingdom, 2001.