SEPARATION OF DRUMS FROM POLYPHONIC MUSIC USING NON-NEGATIVE MATRIX FACTORIZATION AND SUPPORT VECTOR MACHINE

Marko Helén, Tuomas Virtanen

Institute of Signal Processing, Tampere University of Technology
Korkeakoulunkatu 1, FI-33720, Tampere, Finland
phone: + (358) 3 31153251, fax: + (358) 3 31153857, email: marko.helen@tut.fi, tuomas.virtanen@tut.fi
web: http://www.cs.tut.fi/~heln/

ABSTRACT

This paper presents a procedure for the separation of pitched musical instruments and drums from polyphonic music. The method is based on two-stage processing in which the input signal is first separated into elementary time-frequency components which are then organized into sound sources. Non-negative matrix factorization (NMF) is used to separate the input spectrogram into components having a fixed spectrum with time-varying gain. Each component is classified either to pitched instruments or to drums using a support vector machine (SVM). The classifier is trained using example signals from both classes. Simulation experiments were carried out using mixtures generated from real-world polyphonic music signals. The results indicate that the proposed method enables better separation quality than existing methods based on sinusoidal modeling and onset detection. Demonstration signals are available at http://www.cs.tut.fi/~heln/demopage.html.

1. INTRODUCTION

The content analysis of music signals has become a significant research topic in the past few years. One of the most interesting and challenging problems is the automatic transcription of music. Several attempts have been made for transcribing pitched instruments and drums (e.g. [1, 2, 3, 4, 5, 6]).

The transcription and analysis of pitched instruments and drums require very different approaches. When both pitched and percussive instruments are present in the signal, they disturb the estimation of each other; in the transcription of drums, pitched instruments can be considered as noise and vice versa. This problem can be addressed either by using estimation methods which are less sensitive for interference, or by using preprocessing which tries to suppress the interference.

As an example of preprocessing, sinusoidal modeling has been used to estimate the harmonic part of the signal by locating prominent spectral peaks [7, 8]. The harmonic part can be synthesized and subtracted from the original signal to obtain residual, which ideally does not contain any harmonic components. By assuming that pitched instruments are completely harmonic and that drums are completely non-harmonic, they can be analyzed separately from the harmonic part and residual, respectively.

In most cases the above mentioned assumption is not a valid. Some drum instruments, such as snare drums, contain also a significant amount of harmonic energy. This can be easily verified by listening to the sinusoids analyzed from a polyphonic music; usually the snare drum becomes partly modeled by sinusoids. This disturbs the estimation of pitched instruments. Vice versa, also pitched instruments contain non-harmonic components.

The other commonly used approach is based on the transient-likeness of drum events. They usually have a short-duration, so that the energy is highly concentrated in time. The onset times of drum events can be estimated by searching for sharp increases in the energy envelope of the signal. In drum transcription, standard pattern recognition techniques can be applied on the mixture signal at the estimated onset locations [4]. It is evident that co-occurring sounds disturb this method.

1.1 Data-driven processing

Recently, data-driven methods have produced promising results in the separation and analysis of music signals. Unsupervised learning techniques, such as independent component analysis [9, 10, 11, 12], sparse coding [13, 14], and non-negative matrix factorization [15, 16], are able to learn structures from the data without a prior knowledge. Thus, they are also suitable for the blind separation of sound sources.

However, the performance of the methods is currently very restricted in the separation of one-channel signals. For example, each sound source has to be modeled as a sum of one or more components due to the restrictions that the algorithms place on the components. Some proposals for clustering the components to sources have been made [9, 17], but basically, the clustering is still an unsolved problem. In this paper, pattern recognition techniques are used in the clustering.

It is assumed that data-driven methods can be used to solve at least partially the separation problem discussed in the previous section. For example, the separation of pitched instruments and drums could be obtained by using a method which is able to learn the spectra of instruments, so that the periodic content of drums can be correctly assigned to drum signal and the stochastic content of other instruments into the residual.

1.2 System overview

The block diagram of the proposed system is presented in Fig. 1. First, the input signal is separated into components using NMF, as explained in Section 2. Second, features are extracted from the separated components and a support vector machine is used to classify each component either to pitched class or to drum class. The feature extraction and classification procedures are explained in Section 3. Finally, the components within both classes are summed and synthesized to result in separate signals for pitched instruments and drums. The SVM is trained using training samples from both classes. This procedure avoids the need to define whether an instrument is pitched or drum, since example signals from both classes are provided.

Simulation experiments were carried out to monitor the performance of the proposed method. Simulation results and comparison to alternative approaches are presented in Section 4.

2. SEPARATION BY NON-NEGATIVE MATRIX FACTORIZATION

For one-channel signals, ICA and NMF can be applied by using a suitable representation, such as magnitude spectrogram. The representation applies some restrictions for the separated components. The spectrogram is modeled as a sum of components, each of which has a fixed spectrum with a time-varying gain. The model for short-time spectrum vector $\mathbf{x}_t$ in frame t can be written as

$$\mathbf{x}_t \approx \sum_{n=1}^{N} a_{n,t}\mathbf{s}_n, \tag{1}$$

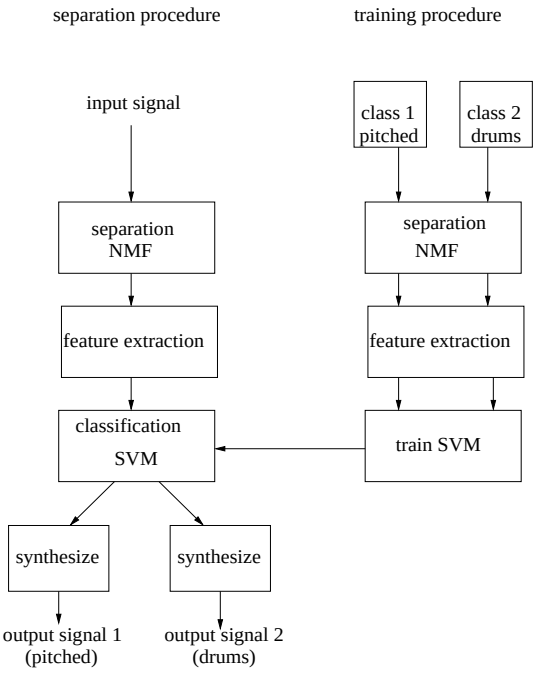

Figure 1: A block diagram of the overall system.

where vector $\mathbf{s}_n$ is the spectrum of n^{th} component, $a_{n,t}$ is the gain of n^{th} component in frame t, and N is the number of components.

The model (1) can be written in a matrix form as

$$\mathbf{X} \approx \mathbf{SA}, \tag{2}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$, $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_N]$, and $[\mathbf{A}]_{n,t} = a_{n,t}$. T is the number of frames.

The spectrogram is calculated by windowing the input signal and by applying the short-time Fourier transform. The square root of the Hanning window was used with 40 ms window length and 50 percent overlap.

There are several criteria for the estimation of the components, including the independency [9] and the sparseness [14] of the time-varying gains. In our simulations the best results were obtained using NMF and divergence [15] for the magnitude spectrogram, as used by Smaragdis and Brown in [16]. The components are restricted to be element-wise non-negative and estimated by minimizing the divergence D:

$$D(\mathbf{X}||\mathbf{AS}) = \sum_{f,t} [\mathbf{X}]_{f,t} \log\left(\frac{[\mathbf{X}]_{f,t}}{[\mathbf{AS}]_{f,t}}\right) - [\mathbf{X}]_{f,t} + [\mathbf{AS}]_{f,t}. \tag{3}$$

The divergence (3) is minimized by using update rules which are given as [15]:

$$\mathbf{S} \leftarrow \mathbf{S} .* \frac{\mathbf{A}^T(\mathbf{X}./\mathbf{AS})}{\mathbf{A}^T\mathbf{1}}, \tag{4}$$

$$\mathbf{A} \leftarrow \mathbf{A} .* \frac{(\mathbf{X}./\mathbf{AS})\mathbf{S}^T}{\mathbf{1}\mathbf{S}^T}, \tag{5}$$

where $.*$ and $./$ are element-wise multiplication and division, respectively, and $\mathbf{1}$ is a all-one matrix of the same size as $\mathbf{X}$.

For magnitude spectrogram $\mathbf{X}$ of the input signal, $\mathbf{A}$ and $\mathbf{S}$ are estimated using an iterative algorithm. $\mathbf{A}$ and $\mathbf{S}$ are initialized with absolute value of random noise, and alternatively updated by rules (4) and (5) until the divergence (3) does not significantly change.

spectral features	temporal features
MFCC (*)	kurtosis
spectral flatness	skewness
spectral centroid (*)	crest factor
spectral contrast	percussiveness
standard deviation (*)	periodicity (*)
roll off point (*)	peak time (*)
noise-likeness (*)	peak fluctuation (*)
spectral dissonance	

Table 1: Features used in the classification. The feature set which produced the best result is marked with (*).

The number of components has to be pre-defined. Some suggestions for determining it from the input signal has been made [9]. In our implementation, the number of components was chosen to be 20.

The separation is the most time-consuming part of the system. For a 10-second input signal and 20 components, the algorithm takes several hundred iterations to converge. This takes several minutes on a regular desktop computer when implemented in Matlab.

3. CLASSIFICATION

Once the polyphonic signal is separated into components, each component is classified either into pitched or into drum class. The features extracted from each component are used for classification.

In the system proposed by Uhle *et al.*, the classification was done using a set of manually tuned decision rules [11]. In our system, the classification is done using standard pattern recognition techniques. This more systematic approach enables the use of a larger number of features and automatizes the classification procedure.

3.1 Feature extraction

The spectrum $\mathbf{s}_n$ of a component n is used to obtain features which describe the frequency content of the component, and the time-varying gain $a_{n,t}$, $t \in [1,T]$ is used to extract features which describe the temporal characteristics. Table 1 presents the features used in our simulations. The following features are calculated from spectrum $\mathbf{s}_n$: Mel-frequency cepstral coefficients (MFCCs), spectral flatness, spectral centroid, spectral contrast, standard deviation, roll off point, and noise likeness. They are all commonly used in pattern recognition. 10 first MFCCs are used in our system. Noise-likeness is a correlation coefficient between the original spectrum and the spectrum convolved with a Gaussian impulse [11]. It is a rough measure of the smoothness of the spectrum. Originally the spectral dissonance was used to measure the degree of how rough or unpleasant the sound is [18]. It has turned out that it can also be used to distinguish between harmonic and noisy spectra.

The features calculated from time varying gains include kurtosis, skewness, crest factor, percussiveness, periodicity, peak time, and peak fluctuation. The percussiveness is a measure of the degree of sharp attacks in the sound. It is estimated as a correlation coefficient between the original gains and the local maxima of the gains convolved with a percussive impulses, which are modeled with instantaneous attack and linear decay [11]. The periodicity is based on the assumption that drum patterns are periodic by nature. It is calculated by locating the maximum value of the autocorrelation of the time-varying gain at delays which correspond to tempos 35-240 beats per minute [19]. Peak time means an average length of peaks in gain curve and peak fluctuation is a deviation between the length of these peaks. Peak is defined here as an area where the gain is above a threshold of 0.2*maximum.

3.2 Classification using support vector machine

The SVM is a pattern recognition method based on statistical learning theory. SVM finds the hyperplane with maximum soft-margin

for the given training set. Finding this hyperplane equals to finding the solution of the certain optimization problem which is described by Burges [20]. SVM is able to learn polynomial classifiers, radial basis function classifiers, or two layer sigmoid neural nets. The type of learning depends on the kernel function used. Our system uses, the SVM light by Joachims [21].

In this work, the SVM classifies the separated components using the features described in the previous section. The SVM is trained using components separated from training samples of both pitched and drum signals. One possible training procedure is explained in Section 4.

3.3 Synthesis

The spectrograms of the components within both classes are summed to yield separate spectrograms for pitched instruments and drums. Complex spectrograms are obtained by using the phases of the original mixture spectrogram, and time-domain signals are obtained using the inverse Fourier transform. The frames are windowed by the square root of Hanning window and combined using overlap-add. The windowing reduces the discontinuities between the frame boundaries. Because the square root of Hanning window is used both in the analysis and synthesis, adjacent windows sum to unity.

4. SIMULATION EXPERIMENTS

The performance of the proposed method was simulated and compared to alternative preprocessing methods. Quantitative evaluation of the separation performance would require reference signals. We did not have access to any material from which the pitched instruments and drums could be obtained separately. Therefore, test signals were generated by mixing harmonic signals and drums from various sources.

The harmonic test signals were 10-second excerpts from commercial CDs containing 400 pieces from different musical genres. The excerpts were chosen so that they do not contain any drums. Since there was only a few samples with singing, vocal samples were added randomly from a database which contains only singing.

Drum signals were taken from a database of acoustic drum signals, recorded using the setup described in [22]. Unlike the drum patterns used in [22], the signals used in our simulations contained also tom-toms and cymbals. In addition, synthesized drum signals were used. The MIDI samples from the commercial Drumtrax 3.0 database were synthesized with the Timidity software synthesizer. Various SoundFonts were used in the synthesis. The length of drum signals was also 10 seconds.

4.1 Training

The training was conducted on with 500 harmonic signals and 500 drum signals. Each training signal was separated into components using the NMF described in Section 2. 20 components were used for the harmonic signals and 10 components for drums. These were chosen since they worked well and testing the different number of sources would have been computationally exhaustive.

Features were extracted from each separated component. The SVM was trained with the extracted features so that the reference class (harmonic/drum) of each component was determined by the source signal.

4.2 Testing

100 samples were used for the testing. The samples were generated by mixing harmonic and drum signals from above-mentioned sources at equal energy levels. The samples were different in the training and testing. The original samples were stored as references before mixing to allow the evaluation of the separation quality.

Each test sample was separated into components using the NMF, and classified using the SVM. The components within both classes were summed and synthesized. The separation quality of harmonic and drum signals was measured by comparing the separated signals with the reference ones. The signal-to-noise ratio of a separated signal (SNR) is calculated as

$$SNR = 10\log_{10}\frac{\sum_t s(t)^2}{\sum_t (s(t)-\hat{s}(t))^2}, \quad (6)$$

where $s(t)$ is the original signal and $\hat{s}(t)$ is the separated signal. The SNR is calculated for all separated harmonic and drum signals.

method	SNR / dB
sinusoidal model	1.35
onset detection	3.05
ICA + SVM	-2.14
NMF + GMM	7.0
NMF + SVM (drums)	7.33
NMF + SVM (harmonic)	2.46
NMF + SVM (harmonic *)	7.33

Table 2: Average signal-to-noise ratios (SNR) obtained using different separation algorithms. For methods other than NMF the SNR of the harmonic and drum parts are equal. NMF + SVM (harmonic *) denotes the method in which the harmonic part is obtained by subtracting the drums from the original signal.

4.3 Comparison to other methods

To get some idea of the quality of the separation, the proposed method was compared to the approaches discussed in the Section 1. Sinusoidal modeling was done using an algorithm which detects the prominent peaks in the spectrum using the sinusoidal likeness measure [8]. The analysis and synthesis of the sinusoids was done without continuation between frames. The harmonic part of the signal was analyzed with a sinusoidal model, and the estimate of the drum part was obtained by subtracting the synthesized sinusoids from the original signal. The threshold value for the detection of the sinusoids was tuned to maximize the SNR of the separated parts.

The other tested method is based on the onset detection algorithm proposed by Klapuri [23]. The onsets were estimated by finding sharp increases of the signal energy within 21 frequency bands. A short-duration segment of the signal after each onset was judged to belong to the drum signal. The harmonic part was obtained by removing the estimated onset segments from the original signal. The threshold for the onset detection and the duration of the segments were tuned to maximize the SNR of the separated signals. The optimal segment duration was found to be 66 ms.

The separation was also tested using ICA instead of NMF, and by classifying the components using the proposed procedure. The FastICA [24] algorithm was used to estimate the independent components. In the classification, SVM was compared to Gaussian Mixture Models(GMM).

4.4 Separation results

Table 2 presents the average SNRs obtained with different methods. For methods, for which the sum of harmonic and drum signals equals exactly the original mixture signal, the SNRs are exactly the same for both separated parts, since the signals were mixed at equal levels. In the case of NMF, the situation is different since the residual $\mathbf{X} - \mathbf{SA}$ is not necessarily zero, and the residual was not classified into either part.

For the drum part, the average SNR obtained using the proposed method is clearly higher than others. However, the SNR of the harmonic part is not as high. This can be explained by the signal model, which suits better for drum signals. However, by calculating the harmonic part by subtracting the synthesized drums from the original signal, the SNR of harmonic part becomes equal to the SNR of the drum part.

The performance of ICA is poor compared to NMF. By listening to the components separated by ICA, it was observed that the

separation quality was often poor. It is obvious that the performance of the sinusoidal modeling and onset detection algorithms can be significantly improved in the separation task by using more advanced techniques. They were tested in this paper to demonstrate the differences between the methods. The average SNR obtained with all the methods is quite low, but the results are promising for the proposed method.

4.5 Classification performance

In order to measure the classification performance, each separated component in the test data was labeled as harmonic instruments or drums. Since a component may have both harmonic and drum content, it was labeled to the class, the reference signal of which it resembles the most. For each component, the residuals between the synthesized component and original signals of pitched instruments and drums were calculated. The energy ratios between the residuals and the original signals were calculated and one with the smaller residual-to-signal ratio was chosen. That is, the component was labeled as drums if

$$\frac{\sum_t (d(t) - \hat{s}(t))^2}{\sum_t d(t)^2} \leq \frac{\sum_t (h(t) - \hat{s}(t))^2}{\sum_t h(t)^2}, \qquad (7)$$

where $h(t)$ and $d(t)$ are the original pitched-instrument and drum signals respectively, and $\hat{s}(t)$ is the separated component. The obtained labeling was used as a reference in the classification.

The percent of correct classifications was used to measure the quality of the classification. The measure is an average of all the test samples. When testing different feature sets, the features calculated from the spectrum seem to work better than the ones calculated from gain. The best combination of features in our simulations included MFCCs, noise-likeness, centroid, roll off point, standard deviation, periodicity, peak time and peak fluctuation. With this feature set, the percent of correct classifications using SVM was up to 93 percent and using GMMs 92 percent. However, almost equally good results (80-90 percent) were achieved with several different feature sets. Other features with good classification capability were percussiveness and crest factor. Even MFCCs alone gave the classification percent of 82. Furthermore, the components clustered in wrong class usually had a lot of content from both classes. For these components, it is hard even for a human to decide which class they belong in.

5. CONCLUSIONS

A method for separating pitched instruments and drums from polyfonic music signals has been proposed. The method is based on factorization of the spectrogram of the input signal and classification of the separated components using a support vector machine. Simulation experiments were carried out using generated mixtures of polyphonic music. The results indicate that the proposed method enables better separation quality than the methods based on sinusoidal modeling and onset detection, for example.

REFERENCES

[1] A. Klapuri, *Signal Processing methods for the automatic transcription of music*, Ph.D. thesis, Tampere University of Technology, 2004.

[2] S. W. Hainsworth, *Techniques for the Automated Analysis of Musical Audio*, Ph.D. thesis, Cambridge Univ., 2004.

[3] J. P. Bello, *Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-Based Approach*, Ph.D. thesis, Univ. of London, 2003.

[4] O. Gillet and G. Richard, "Automatic transcription of drum loops," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Montreal, Canada, 2004.

[5] V. Sandvold, F. Gouyon, and P. Herrera, "Percussion classification in polyphonic audio recordings using localized sound models," in *Proc. 5th International Conference on Music Information Retrieval*, Barcelona, Spain, 2004.

[6] K. Yoshii, M. Goto, and H. G. Okuno, "Drum Sound Identification for Polyphonic Music Using Template Adaptation and Matching Methods", in *Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, Jeju, Korea, 2004.

[7] X. Serra, *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*, Ph.D. thesis, Stanford University, Oct. 1989.

[8] X. Rodet, "Musical Sound Signal Analysis/Synthesis: Sinusoidal+Residual and Elementary Waveform Models," in *Proc. IEEE Time-Frequency and Time-Scale Workshop*, Coventry, Grande Bretagne, 1997.

[9] M. A. Casey and A. Westner, "Separation of Mixed Audio Sources By Independent Subspace Analysis," in *Proc. International Computer Music Conference*, Berlin, Germany, 2000.

[10] S. A. Abdallah and M. D. Plumbley, "An Independent Component Analysis approach to Automatic Music Transcription," in *Proc. 114th AES Convention*, Amsterdam, March, 2003.

[11] C. Uhle, C. Dittmar, and T. Sporer, "Extraction of Drum Tracks From Polyphonic Music Using Independent Subspace Analysis," in *Proc. 4th International Symposium on Independent Component Analysis and Blind Signal Separation(ICA2003)*, Nara, Japan, April 2003.

[12] E. Vincent and X. Rodet, "Music transcription with ISA and HMM," in *Proc. 5th Int. Symp. on ICA and BSS (ICA'04)*, Granada, Spain, 2004.

[13] T. Virtanen, "Sound Source Separation Using Sparse Coding with Temporal Continuity Objective," in *Proc. International Computer Music Conference*, Singapore, 2003.

[14] S. A. Abdallah and M. D. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra," in *Proc. the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pp 318-325, Barcelona, Spain, 2004.

[15] D. D. Lee and H. S. Seung, "Algorithms for Non-negative Matrix Factorization," in *NIPS*, pp. 556-562, 2000.

[16] P. Smaragdis and J. C. Brown, "Non-Negative Matrix Factorization for polyphonic music transcription," in *Proc. 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Platz, New York, USA, 2003.

[17] S. Dubnov, "Extracting sound objects by independent subspace analysis," in *Proc. AES22 International Conference on Virtual, Synthetic and Entertainment Audio*, Finland, 2002.

[18] W. A. Sethares, "Local Consonance and the Relationship Between Timbre and Scale," *J. Acoust. Soc. Am.*, 94 (3), pt. 1, 1993.

[19] T. Heittola and A. Klapuri, "Locating Segments with Drums in Music Signals," in *Proc. 3rd International Conference on Music Information Retrieval(ISMIR 2002)*, France, October 2002.

[20] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," in *Data Mining and Knowledge Discovery*, 2(2):1-47, 1998.

[21] T. Joachims, *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*, B. Scholkopf and C. Burges and A. Smola(ed.), MIT-Press, 1999.

[22] J. Paulus and T. Virtanen, "Drum Transcription with non-negative spectrogram factorisation," submitted in *EUSIPCO 2005*, Antalya, Turkey, Sept. 4-8. 2005.

[23] A. P. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *Proc. IEEE Conference of Acoustics, Speech and Signal Processing*, Phoenix, Arizona, 1999.

[24] FastICA package for MATLAB http://www.cis.hut.fi/projects/ica/fastica/.