Blind Enhancement of the Rhythmic and Harmonic Sections by NMF: Does it help?

Björn Schuller, Alexander Lehmann, Felix Weninger, Florian Eyben, Gerhard Rigoll
Technische Universität München, Institute for Human-Machine Communication, Germany, schuller@tum.de

Introduction

Non-Negative Matrix Factorization is well known to lead to considerable successes in the blind separation of drums and melodic parts of music recordings. Such splitting may well serve as enhancement when it comes to typical Music Information Retrieval tasks as automatic key labelling or tempo detection. In this respect we introduce the combination of an NMF based blind music separation into several isolated audio tracks in combination with Support Vector classification to assign each obtained track to be either rhythmic or melodic. Thereby optimal parametrization and performances are discussed. Next, stereophonic information is further used to eliminate the key melody and bass usually panned in the centre for tempo detection or e. g. for chord labelling. We then analyse the potential for the named tasks by a number of experiments carried out on the MTV Europe Most Wanted of the 1980 ies and 90 ies in MP3 format.

Drum-beat Separation by NMF

Uhle et al. [1] designed a system for drum beat separation based on *Independent Component Analysis*. In contrast, Smaragdis and Brown [2] relied on *Non-negative Matrix Facorization* (NMF) to create a system for transcription of polyphonic music that showed remarkable results on piano music. Helen and Virtanen [3] used NMF, combined with a feature extraction and classification process, and achieved promising results in drum beat separation from popular music. Similar techniques were used by [4] and [5] for drum transcription. In this work we want to evaluate NMF for the purpose of drum beat separation as enhancement in tempo and key detection. However, we first tweak the classification process by evaluating different cost functions and parameters.

Given a matrix $V \in \mathbb{R}_{\geq 0}^{n \times m}$ and a constant $r \in \mathbb{N}$, NMF computes two matrices $W \in \mathbb{R}_{\geq 0}^{n \times r}$ and $H \in \mathbb{R}_{\geq 0}^{r \times m}$, such that

$$V \approx W \cdot H. \tag{1}$$

For $r \ll n, m$, there exists generally only an approximate solution. Factorization is usually achieved by iterative algorithms minimizing cost-functions as:

$(V - WH)^2$	Squared error
$\| (V - WH) \|_F$	Frobenius norm
$\sum_{ij} \left(V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right)$	Mod. KL Div.

Whereas the first two cost-functions are closely related to each other as both minimize some form of quadratic error, the latter interprets the matrices V and (WH) as probability distributions and minimizes their divergence. This is a modification of the Kullback-Leibler (KL) divergence because of the additional term $(WH)_{ij} - V_{ij}$, which is not only added to introduce a measurement of the absolute error but most importantly to ensure non-negativity. Of course, a lot more such cost-functions do exist. In fact, most of the available algorithms only differ by their choice of one particular cost-function. As one of the first practical algorithms for NMF, Lee and Seung [6] developed so-called "multiplicative update rules" related to the Frobenius norm as well as to the KL divergence. While these differ from a regular gradient-descent, it can be shown that under these rules, for each iteration step the error is *non-increasing*. We evaluated both of the suggested multiplicative update approaches as well as a gradient descent with respect to the squared error.

Model

The basic idea of using non-negative matrix factorization for instrument separation is the interpretation of a *monophonic* signal's short-time *magnitude spectra* as linear combinations of several distinct components' spectra. In particular, it turns out that the assumption of non-negativity of each component suffices for this model. Now, when applying non-negative matrix factorization on a signal's magnitude spectrum and considering Eq. 1, one can interpret the resulting columns of W and the rows of H as *spectral components* and their *gains over time*, respectively. Hence, the overall contribution of the i-th component to the magnitude spectrum of the original signal can be calculated as the (dyadic) product of the i-th column of W and the i-th row of H. Since only the magnitude spectrum is factorized, the separated components can be transformed back into the time domain by simply using the calculated components' magnitude spectra in conjunction with the original phase spectrum. To distinguish between the terms *instrument* separation and *component* separation, readers should note that the spectrum of one particular instrument is probably comprised of more than one component. That being said, it is indispensable to have a closer look at the available *parameters* of the described model introduced by STFT and NMF.

Parameters introduced by STFT

Parameters for STFT include the choice of a *window function*, *window overlap* and *window size*. Examples of window functions are the rectangular window, the Hann window as well as the square root of the Hann window, the latter for example being used by [3]. It seems that

window size is among the parameters that have most impact on the perceptual quality of the factorization. According to our results, we conclude that for the task of drum-beat separation window sizes between 40 to 60 ms seem to be reasonable choices. For instance, a window of 62.5 ms captures the extension of an eighth note at 120 bpm. It should be noted that STFT can produce a fairly large amount of data. For example, the magnitude- and phase-spectrum matrices for a 30 s signal have a dimension of $1\,322 \times 1\,000$, assuming a window size of 60 ms, 50 % overlap, and a sample rate of 44.1 kHz.

Number of Components

Thinking of, for example, a piece of music that only contains single notes while most notably remembering the *non-negativity constraint* as introduced, one can easily understand that every such note ought to be representable through its very own spectrum and therefore also through a single corresponding component. More obviously, Smaragdis et al [2] hence speak of *events* instead of *components*, which clearly emphasizes their singularity. Generally speaking, a higher number of components is not considered harmful as the superfluous components' contributions to the whole magnitude spectrum will be nearly zero. On the other hand, a higher number of components results in smaller absolute values and thus less maximum amplitudes of the separated components. It is thus evident that, at best, the exact number of components were known in advance. This might even be possible for a certain subset of music or due to preliminary analysis of the particular music that should be separated, yet it is not the case in general. According to our experience, an average choice of 20 to 30 components is advisable for unsupervised instrument separation of popular music.

Limitations

Despite its great potential, non-negative matrix factorization for the purpose of instrument separation is subjected to the following limitations: NMF is not guaranteed to find a global minimum of the respective cost-function. Furthermore, the randomized initialization of the two matrices W and H leads to slightly different results on each application of the algorithm. Paulus and Virtanen [4] have therefore introduced targeted initialization of the two matrices by using certain application-dependent training sets. Still, this particular topic yields promising potential for further research. Components (events) that never occur by themselves are unlikely to be separated by an algorithm with random initialization of the matrices. Intuitively speaking, the algorithm is not capable of separating events that always occur together because it can just as well achieve a good representation (in terms of the cost-function) by putting them into a single component. Also, when using NMF as a preprocessing step for feature extraction, caution must be exercised with respect to the initialization as, for instance, random initialization with small values within $[0.01, 0.02]$, in comparison to values within $[0.1, 0.2]$, often yields results with totally different scale and hence could have a non-negligible impact on the extracted features' values. As a side-effect of the *factorization's ambiguity*, which can easily be shown by regarding the product $W \cdot H$ compared to $W \cdot A^{-1} \cdot A \cdot H$, where A is some arbitrary permutation or affine transformation, the order of the separated components is non-deterministic. For our intended application, this has no effect, since normally the resulting components are classified automatically and no assumptions are made about their order. Since the model parameters have rather great influence on the (perceptual) quality of the factorization, any prior knowledge about the data to be processed should be used wherever applicable.

Convergence Characteristics

In addition to the varying perceptual quality of their factorization results, the beforementioned algorithms most notably differ in their convergence characteristics. Figure 1 shows an examplary comparison of the residual error during factorization for the gradient descent as well as for the multiplicative updates minimizing Frobenius norm (distance) and divergence. Extracts of 15 s length from the following songs were used for the comparison: *Eminem – Stan*, *Fettes Broot – Jein*, *Michael Jackson – Thriller*, *Prodigy – Firestarter*, *REM – Losing My Religion*, *Run DMC – It's Like That*, *Run DMC – Walk This Way*, *Soft Cell – Tainted Love*, *U2 – Hold Me, Thrill Me, Kiss Me, Kill Me*.

The songs were monophonic and sampled at 44.1 kHz while the 15 s intervals were chosen such that they are preferably representative for the particular songs. For every epoch, the residual error was computed as the ratio of the Frobenius norms of the absolute error and the original magnitude spectrum, i. e.

$$f(W, H, V) = \frac{||WH - V||_F}{||V||_F}, \tag{2}$$

where V represents the original spectrum and W, H are the factorization's results. The referred figure eventually shows the respective minimum and maximum error as well as the expected value during the respective epochs. One quickly notices the steeper convergence of both multiplicative update approaches versus the gradient descent. It is also straightforward to see that a certain residual error always remains. Furthermore, as the residual error is computed with respect to the Frobenius norm, it is no coincidence that the corresponding algorithm shows the fastest convergence rate. When looking at the convergence characteristics, it is also important to examine the computational speeds of the distinct algorithms. The real time factors (RTF) for 100 iterations and total time (TT) for 500 iterations were determined on a 2.6 GHz Intel Xeon with 5 GB memory as depicted in Table 1 for a total of 150 s of input data. As to memory requirements, at the time of this writing the memory consumption of our implementation for processing 1 second of monophonic samples at 44.1 kHz is at a ratio of about 1 to 3, i. e. 1 s of sound to 3 MB of memory, depending on the selected algorithm. Summa summarum it is worth noting

that in spite of being the "slowest" of the evaluated algorithms, the divergence multiplicative update shows at least equally good convergence characteristics and by far delivers the best perceptual results with respect to the separated components, because – due to its nature – it not only finds an arbitrary factorization but also respects the underlying data's distribution.

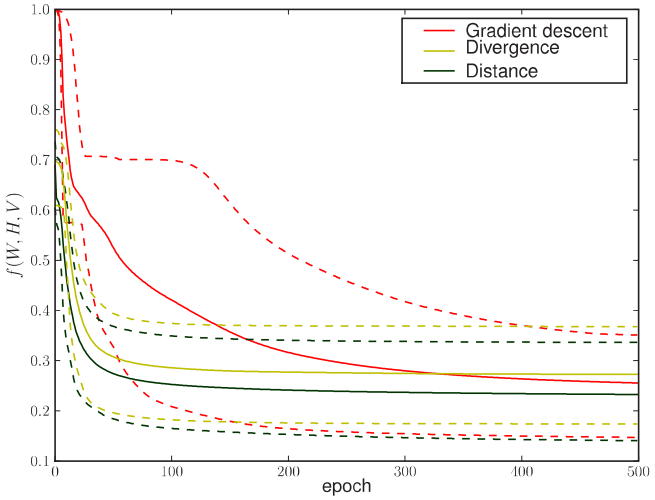

Figure 1: Exemplary comparison of NMF algorithms' convergence. Depicted is the decrease of the cost function over the course of the iteration. Dashed lines indicate extrema, continuous lines the mean.

Algorithm	**TT** (500 it.) [s]	**RTF** (100 it.)
Distance	151	0.201
Grad. Descent	181	0.241
Divergence	877	1.169

Table 1: Real time factor (RTF) and total time (TT) for different algorithms for NMF. Grad. abbreviates Gradient.

Component Assignment

As outlined, the number of components is usually greater than the number of classes that should be separated. For instance, in drum beat separation there are only two classes (drum and harmonic), while the number of components should be chosen around 20 to 30 for best separation results. Thus, components need to be classified and then superposed to generate the signals that correspond to each class. To present suited features to the classifier we implemented feature extraction for the drum beat separation task following the approaches by [1] and [3]. As shown, each component is characterized by a column of the spectral matrix W and a row of the gains matrix H that have been obtained by NMF. In the following, we will refer to these as *spectral vector* and *gains vector*, respectively. One could think of obtaining time signals for each component as described and extracting features from these time signals, but we feel that this procedure would not be optimal as features would be extracted from redundant representations of the components. Since the spectral and gains vectors contain all relevant information about the respective component, we perform feature extraction on these two vectors.

Spectral Features

From each spectral vector $\boldsymbol{s} = (s_1, \ldots, s_N)^T$, corresponding to frequencies $f_1, \ldots, f_N$, we extract the following features: 10 Mel frequency cepstral coefficients (MFCCs), sample standard deviation, spectral centroid, 95 % roll-off point, noise-likeness [1], spectral flatness and spectral dissonance [1]. For the calculation of *MFCCs*, we use a filter bank that ranges from 20 Hz to 8000 Hz. The *sample standard deviation* is computed using the common unbiased estimator.

Temporal Features

Temporal features are calculated from the gains vectors. For each gains vector $\boldsymbol{g} = (g_1, \ldots, g_M)$, we extract the following features: sample standard deviation, percussiveness, periodicity, average peak length and peak fluctuation.

Percussiveness [1] is a measurement of how accurately $\boldsymbol{g}$ can be modelled using instantaneous attacks and linear decays, a model which seems to resemble the structure of most drum patterns.

Periodicity [3] is based on the notion that drum patterns are often periodic in intervals that correspond to the tempo of the piece. We compute auto-correlation coefficients (i. e. autocorrelation values normalized by mean and variance) of $\boldsymbol{g}$, with delays that correspond to tempi of 30 to 240 bpm, at intervals of 5 bpm. We define as periodicity the maximum of the obtained autocorrelation coefficients. In order to define *average peak length* and *peak fluctuation*, we first introduce the concept of peaks. Informally speaking, a peak is any area of $\boldsymbol{g}$ that is over a threshold of 20 % of the maximum of $\boldsymbol{g}$. Formally, a peak of length l is a set of consecutive indices $\{i, i+1, \ldots, i+l-1\} \subseteq \{1, \ldots, M\}$ such that

$$g_i, g_{i+1}, \ldots, g_{i+l-1} \geq 0.2 \cdot \max\{g_i\}. \qquad (3)$$

After finding the peaks in $\boldsymbol{g}$, we can determine the average peak length, that is the sample mean of the peak lengths, and the peak fluctuation, that is their sample standard deviation [3]. Our data provides evidence that generally drum components have short peaks of similar length, whereas harmonic ones have longer peaks that vary more in length.

Synthesis

After classification, time signals for each class are obtained by the procedure proposed by [3]: for each class, we calculate a magnitude spectrogram by adding the magnitude spectrograms of the components belonging to that class. (The magnitude spectrogram of a component is the dyadic product of its spectral and gains vector.) We perform a column-wise inverse discrete-time Fourier transformation (IDFT) on the class spectrograms, using the phase values from the corresponding columns of the phase matrix of the original signal. We obtain time signals by windowing the columns with the square root of the Hann function, then using overlap-add.

Drum Beat Separation Evaluation

From the song collection "20 Years on MTV" [7], consisting of 200 songs in total, we generated one input signal per song, each about 15 to 30 s long. Using our framework, we calculated spectrograms of these signals, using the square root of the Hann function with a window size of 60 ms and 50 % window overlap. We then applied NMF, setting the number of components to 30. We selected 344 of the 6 000 resulting components by perceptual quality. Music experts assigned each to exactly one of the classes "Drum" (95 components) or "Harmonic" (249 components). We validated this data set using 10-fold stratified cross-validation (SCV). The data were scaled such that the values of every feature were in the range $[-1, 1]$. As classifier, we used a SVM with linear kernel. The "complete" feature set contains all features described above and leads to accuracy of 95.9 %. The "reduced" feature set includes 10 MFCCs, noise-likeness, standard deviation, spectral centroid and rolloff for spectral vectors, and average peak length, percussiveness, peak fluctuation and periodicity for gains vectors. This is the feature set proposed by [3], and performs best for our data set at an accuracy of 96.2 %.

Application of NMF

We now evaluate the potential gain of drum-beat separation for two highly relevant Music Information Retrieval tasks, namely tempo and key detection. Both experiments are carried out on the full "20 Years on MTV" [7] set. For tempo and meter (duple or triple) detection we use a comb-filter bank-based approach as described in [8]. Table 2 shows results for this task with the original audio and mids-removed audio by stereo channel subtraction for all, only harmonic or only drum components added accordingly. Interestingly, no improvement can be found in any of the processing steps. Apparently either spectral distortions are responsible for this effect, or the tempo simply is best reflected by all components.

Acc.	**Original**			**Channel Subtracted**		
[%]	all	drum	harm	all	drum	harm
meter	98.4	96.2	98.4	95.7	93.5	89.2
tempo	93.5	93.0	92.5	89.2	85.0	80.1
octave	75.8	74.7	74.7	72.6	69.9	66.7

Table 2: Effect on accuracy of blind drum separation by NMF for meter and tempo detection. Shown are duple or trimple meter, tempo with (tempo), and without (octave) allowance of octave errors for all components, only drum, and only harmonic (harm) such.

Next we consider recognition of 12 major keys in Table 3. Relative minor keys are mapped upon these, accordingly. We use CHROMA features (frame size 8192, frame overlap 50%, windows: Hanning (FFT), Gauss (semitone bandpasses), frequency range 126 Hz - 4 066 Hz, pitch adjustment to cope with detuning) with the same SVM in 10-fold SCV. Only the original audio with all components or exclusively harmonic components are considered: here, mid removal would eliminate important information on the scale tones carried by melody and bass. Again, no improvement is found in NMF-based separation.

[%]	**Key**	**Sub**	**Dom**	**Sum**
all	70.7	6.9	12.7	90.3
harmonic	68.6	11.7	8.5	88.8

Table 3: Effect on accuracy of blind drum separation by NMF for key detection. Shown are the percentage of correctly assigned keys (Key), confusions with the (sub-)dominants (Sub/Dom) and the respective sum.

Conclusion

In this work we have shown a highly effective separation of music into drum-beat and harmonic section. While the audible results are well usable in e. g. DeeJay applications or music remixing, no gain could be obtained by using the separation to boost performance of tempo and key detection in musical recordings. In future work we will investigate whether the separation can be better exploited on an earlier stage: in tempo detection NMF components could replace the mel-filter bank prior to the comb-filter analysis, and in key detection the CHROMA features could be based on spectral decomposition by NMF of the Gaussian semitone band filters.

Acknowledgement

The authors would like to thank the student assistant Benedikt Gollan for his highly valuable contributions.

References

[1] C. Uhle, C. Dittmar, and T. Sporer, "Extraction of drum tracks from polyphonic music using independent subspace analysis," in *Proc. ICA*, 2003.

[2] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. WASPAA*. 2003, pp. 177–180, IEEE.

[3] M. Helén and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in *Proc. EUSIPCO*, 2005.

[4] J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorisation," in *Proc. EUSIPCO*, 2005.

[5] A. Moreau and A. Flexer, "Drum transcription in polyphonic music using non-negative matrix factorisation," in *Proc. ISMIR*, 2007.

[6] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," 2001, pp. 556–562, MIT Press.

[7] "20 Years On MTV," Sony BMG Music Entertainment GmbH (ASIN: B0006H2VCC).

[8] B. Schuller, F. Eyben, and G. Rigoll, "Tango or waltz?: Putting ballroom dance style into tempo detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2008, no. Article ID 846135, pp. 12 p, 2008.