Data augmentation for deep learning source separation of HipHop songs

Hector Martel and Marius Miron

Universitat Pompeu Fabra, Music Technology Group, Barcelona
hector.martel01@estudiant.upf.edu

Abstract. Training deep learning source separation methods involves computationally intensive procedures relying on large multi-track datasets. In this paper we use data augmentation to improve hip hop source separation using small training datasets. We analyze different training strategies and data augmentation techniques with respect to their generalization capabilities. Moreover, we propose a hip hop multi-track dataset and we implemented a web demo to demonstrate our use scenario. The evaluation is done on a part of the dataset and hip-hop songs from an external dataset.

Keywords: Music Source Separation, Deep Learning, Hip Hop

1 Introduction

Audio Source Separation involves recovering individual components from an audio mixture [8]. This task is related to auditory scene analysis, however it is difficult for the current algorithms to match human ability of segregating audio streams. Matrix decomposition techniques such as Non-Negative Matrix Factorization (NMF) [3], were traditionally used for audio source separation. NMF is particularly popular in this field because of its additive reconstruction properties. However, the NMF iterative procedure is computationally expensive in contrast to newer approaches using deep learning [2]. Furthermore, frameworks as [3] rely on a pitch detection stage and assume a voiced source.

Deep Neural Networks model source separation as a regression problem, taking as an input time-frequency representations such as Short-term Fourier Transform (STFT) magnitude spectrograms, and estimating a continuous output, the magnitude spectrograms for the sources [2,5,7]. Because the estimation assumes a single feed-forward pass through the network, deep learning frameworks are less computationally intensive than NMF [2]. However, deep learning models are expensive to train and require large datasets with isolated instruments [7], which are difficult to obtain. Furthermore, data driven methods can often overfit and fail for a particular test case, which might represent a different problem in itself. To that extent, data augmentation [1] is a regularization technique that increases the robustness of an already trained model and boosts its performance on unseen data.

In this paper we study the use of data augmentation to retrain a general purpose music source separation model for hip hop music, on very small datasets. We are interested in assessing the generalization capabilities of the models trained with such data.

For the experiments we use the Convolutional Neural Network (CNN) autoencoder [1] in [2] which separates pop-rock music with low latency. The baseline architecture comprises an encoding and a decoding phase. At the encoding phase we have a vertical convolution which models timbre characteristics, a horizontal convolution which models temporal evolution, and a dense layer with a low number of units which acts as a bottleneck. The decoding phase assumes performing the inverse operations of the layers in the reverse order, namely another dense layer and two deconvolutions.

We follow the research reproducibility principles and publish the dataset, code, and a web demo.

The remainder of this paper is structured as follows. In Section 2 we present the use scenario, followed by the proposed dataset in Section 3. In Section 4 we evaluate the use scenario and discuss the results. The conclusions are presented in Section 5.

2 Use scenario

HipHop music is an interesting scenario for source separation. The most noticeable characteristic is that the voice is not sung. Thus, pitch-based methods [3] would not work properly to extract the vocals. Furthermore, the drums and the bass can be acoustic, synthesized or sampled from vinyl records, making the timbre variability of the sources very high.

Our use scenario is remixing or upmixing recordings [4] in the same production style, where the instrumentals are created by a single producer and the voices come from different musicians. Furthermore, we are interested in live remixing, where latency plays a crucial role in the overall performance, and it is advantageous to use a deep learning system. Such a system can be used by a music producer or DJ to manipulate songs within a certain genre or production style to play them live.

3 Dataset

3.1 Proposed dataset

We propose a compilation of Hip Hop songs, referred to as HHDS [1], which can be used to train a neural network. The structure of HHDS follows the convention of DSD100 [2] (Demixing Secrets Dataset). HHDS contains the separated tracks for the categories of bass, drums, vocals and others in monophonic WAV files

[1] HHDS, on Zenodo: `http://doi.org/10.5281/zenodo.823037`

[2] Demixing Secrets Dataset (DSD100), SiSEC2016: `http://liutkus.net/DSD100.zip`

with a sampling rate of 44100Hz. The mixture is calculated by normalizing the sum of the tracks. The main difference with respect to DSD100 is that in HHDS there are HipHop songs only, instead of many different genres. The total number of songs is 18, from which 13 are used for training and 5 are used for evaluation. The songs are mixed by one producer and contain vocals in Spanish from 12 different musicians to maximize timbre diversity for voice. More details about the dataset can be found at the repository page.

3.2 Data Augmentation techniques

A deep learning model can become more robust through data augmentation techniques which create more training instances [6]. To that extent, we choose transformations which are relevant for source separation and are applied to the audio signal, rather than the STFT magnitude spectrogram. Thus, similarly to [6], we discard other popular transformations such as pitch shifting and we analyze the following augmentation techniques:
a) Instrument Augmentation (IA) [7]. More renditions of the same song can be created by muting one of the instrument tracks. This transformation is useful modeling hip hop cases, e.g. an instrument does not play in certain sections.
b) Mix Augmentation (MA) [7]. We sum instrument tracks from different songs to create a new mix. The tracks are combined and picked randomly. This transformation de-correlates the harmonic relation between the instruments within a mix, however it provides more training examples of different timbre combinations.
c) Circular Shift (CS) [5, 6]. The audio signals corresponding to instrument tracks are shifted between each other with a fixed number of time frames. With this transformation we introduce small temporal deviations of $0.1, 0.2$ seconds which make the network more robust to various time patterns. While temporal alignment of the instrument tracks is slightly modified, the structure of the song does not change.

4 Evaluation

4.1 Experimental setup

a) Parameters To train the network, the spectrograms are passed through it iteratively for 40 epochs using the parameters of the baseline method [2] with mini-batch stochastic gradient descent.
b) Evaluation metrics We use the objective measures proposed in [8]: Source to Distortion Ratio (SDR) as a global quality measure, Source to Interference Ratio (SIR) related to the interferences from other sources, Source to Artifacts Ratio (SAR) related to the presence of artifacts. All measures are expressed in decibels (dB).

4.2 Experiments

The experiments evaluate 8 models, described in Table 1. A generic model trained with DSD100 [2] is used as a reference. 2 models are retrained from the reference using HHDS, and 5 models are trained with HHDS and data augmentation techniques.

Table 1. Models generated during the training phases.

DSD	Trained with DSD100 songs only, used as a reference.
DSD_HH	Retrained from DSD100 with HHDS in a new training.
DSD_HH_2	Retrained from DSD100 with HHDS in a new training, with a lower learning rate for 10 epochs.
HH	Trained with HHDS songs only, used as a specialized case.
HH_COMBI	Trained with HHDS and all the Augmentations combined.
HH_CS	Trained with HHDS and Circular Shift Augmentation.
HH_IA	Trained with HHDS and Instrument Augmentation.
HH_MA	Trained with HHDS and Mix Augmentation.

4.3 Results

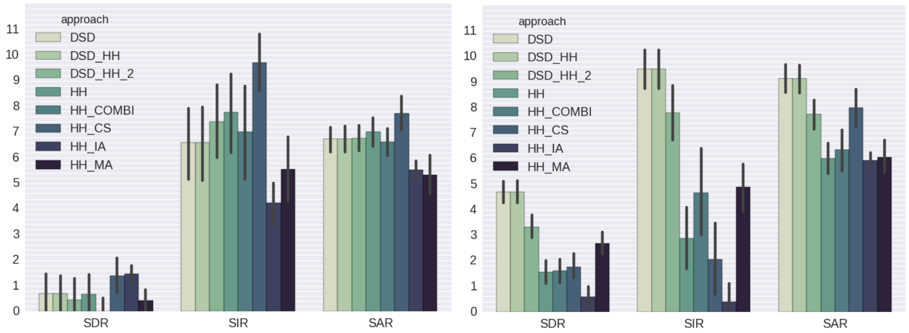

Fig. 1. Results in terms of SDR, SIR, SAR for HHDS Test (left) and the HipHop songs from DSD100 (right).

We evaluate the models for two different contexts involving different production styles, first, on the test set from HHDS (Figure 1 left), and, second, on the 3 songs labeled as HipHop from test set of DSD100 (Figure 1 right) which are not used to train any of the models. In the corresponding figures, error bars are drawn for a confidence interval of 95%. Note that the songs from HHDS comprise one production style, with mostly synthetic drums, while the ones in DSD100 have mostly acoustic drums and contain a more variety of production styles.

As seen in Figure 1 left, the generic model trained on the DSD set, DSD, has lower performance than the models trained on HHDS dataset comprising tracks with similar production style. As expected, the generic model performs better on the DSD100 hip hop set (Figure 1 right) because it models better the acoustic drums and it was trained with larger variety of timbres.

Training the model from scratch on HHDS, HH, improved 0.7dB SDR over the generic model, DSD, while decreasing 3dB over the 3 DSD100 songs which are created in a different production style. Further improvements of 3dB over the generic model and 1.5dB over the HH, are obtained with Circular Shift (CS) augmentation in HH_CS. This augmentation makes the model more robust for unseen songs the same production style (Figure 1 left), however not for the 3 songs in DSD100 (Figure 1 right). This type of augmentation is helpful in our use scenario: remixing of recordings in the same production style.

The other two augmentation techniques do not improve the results on HHDS dataset, however the Mix Augmentation model (HH_MA) obtained more robust performance on the DSD100 songs: 2dB higher. Thus, creating more combinations between different not correlated tracks, keeps the performance stable on the target context and makes it more robust on songs from different production styles. The Instrument Augmentation model (HH_IA) did not improve the baseline method because the combinations created were not realistic.

The combination of all the augmentation techniques in HH_COMBI did not result in a significant improvement in any of the two contexts. It obtained 0.5dB lower performance respect to HHDS, achieving 0.2dB over the baseline method for the same production style. Similarly to HH_IA, the combinations generated are not realistic.

Surprisingly, DSD_HH, which involved initializing the model with weights from DSD, and then re-training with HHDS, did not improve over the generic model and over the model trained from scratch. Also in DSD_HH_2, trained with 10 epochs and a lower learning rate, the improvement is not significant. Further experiments are needed to assess these problems.

5 Conclusions

We have presented a source separation scenario that is well suited for the use of small datasets under the genre-specific assumption. A producer or DJ can use this system to train a model with very few songs and separate songs of the same style. From the experiments it can be extracted that the context-specific methods outperform the general purpose one for test cases similar to the training examples. For songs that differ from the training data the performance can be improved with data augmentation, achieving a more general representation. Thus, we found that there is a trade-off between the specialization of model and its performance under unknown test data.

Future research on this topic can be focused on the development of applications such as upmixing or remixing, as well as exploring more data augmentation techniques. It would also be interesting to expand HHDS with songs from other producers.

The reader is encouraged to check a web-based demo of this paper[3].

6 Acknowledgments

The TITANX used for this research was donated by the NVIDIA Corporation. This work is partially supported by the Spanish Ministry of Economy and Competitiveness under CASAS project (TIN2015-70816-R) and by the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

References

1. Y. Bengio et al. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.
2. P. Chandna, M. Miron, J. Janer, and E. Gómez. Monoaural audio source separation using deep convolutional neural networks. *International Conference on Latent Variable Analysis and Signal Separation*, 2017.
3. J.-L. Durrieu, A. Ozerov, C. Févotte, G. Richard, and B. David. Main instrument separation from stereophonic audio signals using a source/filter model. In *Signal Processing Conference, 2009 17th European*, pages 15–19. IEEE, 2009.
4. D. Fitzgerald. Upmixing from mono-a source separation approach. In *Digital Signal Processing (DSP), 2011 17th International Conference on*, pages 1–7. IEEE, 2011.
5. P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep Learning for Monaural Speech Separation. *Acoustics, Speech and Signal Processing (ICASSP)*, pages 1562–1566, 2014.
6. M. Miron, J. Janer, and E. Gómez. Generating data to train convolutional neural networks for classical music source separation. In *Sound and Music Computing*, 2017.
7. S. Uhlich, F. Giron, and Y. Mitsufuji. Deep neural network based instrument extraction from music. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 2135–2139. IEEE, 2015.
8. E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*, 14(4):1462–1469, jul 2006.

[3] `https://hiphopss.github.io/`