AUTOMATIC SAMPLE DETECTION IN POLYPHONIC MUSIC 2017

First Author
Affiliation1
author1@ismir.edu

Second Author
Retain these fake authors in submission to preserve the formatting

Third Author
Affiliation3
author3@ismir.edu

ABSTRACT

The term 'sampling' refers to the usage of snippets or loops from existing songs or libraries in new music productions or mashups. Being able to detect samples in songs is useful in tracing and studying artist influences across generations of musicians. In this paper, an algorithm that utilizes Non-negative Matrix Factorization (NMF) and Dynamic Time Warping (DTW) is proposed to detect the instances of a given audio sample in a song. NMF is used to learn the spectral templates and their activations from the sample. Factorizing the query song with these and randomly initialized templates allows to align the sample activations with the query activations using DTW. Features derived from the DTW paths are used to train a random forest classifier to detect the presence of the sample. The algorithm is able to detect samples that are pitch-shifted and/or time-stretched and is evaluated against a dataset of real-world sample and song pairs which has been made available.

1. INTRODUCTION

Sampling, in the context of music composition and production, is the concept of reusing digital recordings, known as the 'sample', in new compositions in a way that it fits in the musical context. In digital sampling, an artist records a segment of a song or sound that he wishes to sample, may or may not modify it, and then reuses it (and possibly other recordings) by incorporating it into a new composition. Sampling of audio has become popular in mainstream pop, hip-hop and rap music.

A sample detection system enables a musicological study of the influence of older artists over newer generation artists by observing sampling patterns over the years.

Another possible use case of a sample detection system could be to detect plagiarism or copyright infringement. Sampling is legally controversial and determining fair use is largely left to the courthouse. A system that gives an objective measure of the likelihood of a sample being present in an audio file could add weight to either party's argument in a lawsuit.

The algorithm discussed in this paper focuses on solving the problem of detecting the presence of a given sample in a set of songs and also the location of where the sample is most likely present.

2. RELATED WORK

In academia, only a few publications were found that specifically tackled the problem of sample detection. However, there are several parallels that may be drawn from other areas of research that are relevant to sample detection such as cover song detection, audio fingerprinting and remix recognition. The table **??** has a brief comparison of audio fingerprinting, cover song detection and sample detection systems.

2.1 Audio Fingerprinting

Audio fingerprinting refers to the method of extracting content-based signatures from audio [5]. It is most commonly used in content-based music retrieval systems, like Shazam [1] . Van Balen proposed the use audio fingerprinting for sample detection [15]. He used a popular fingerprinting by Wang [17], in an implementation by Ellis [9].

Fingerprinting is a good choice for building systems that are robust against attacks such as pitch shifting or time stretching of audio, but in the case of sample detection, a sample is usually one component in a mixture of audio. Audio fingerprinting detects the exact audio but wouldn't perform well when the audio is mixed and masked by other audio signals.

2.2 Cover Song Detection

Cover song detection is the task of recognizing whether a given reference track has a cover song in a set of test tracks [2, 8, 13]. In cover song detection, covers may also be transposed or pitch-shifted and may vary in tempo from the original song. Dynamic Time Warping (DTW) [1] is often used to make these systems time invariant and this work uses the same. The difference lies in the fact that covers are renditions of a musical piece, while samples are snippets of audio which are usually a small part of the mix overlaid with a lot of other instruments and sounds.

Evaluating cover song detection systems and a sample detection system is highly similar. Both have a test/reference pair which is then categorized as a positive or negative match with a confidence measure.

[1] https://www.shazam.com/

Table 1. Comparison Table of Related Work

	Audio Fingerprinting	Cover Song Detection	Sample Detection
Similarity of query to reference	Exact audio is detected, with some degradation possible	Cover is not exactly the same audio as the reference	Exact same sample is present in reference, with some effects applied
Addition of extra audio tracks	Query audio isn't mixed with other audio tracks	Cover is usually a linear performance of the reference with some artistic changes	Sample could be present in a mixture of several other instruments in the reference

2.3 Remix Recognition

Work done in remix recognition by Casey et al. [6] draws inspiration from a method for web crawling called 'shingling' which utilizes a stream of text position-based features to detect if a document has already been crawled before. They snip an audio track into 4 second 'shingles' and search in a set of similarly snipped test tracks using popular low-level audio features such as MFCCs and pitch-class profiles. In sample detection, such a system wouldn't work because the features would capture information of the whole mix instead of just the sample.

2.4 Non-negative Matrix Factorization-based Approach

Dittmar et al. outlined three kinds of plagiarism in music, one of them being sample plagiarism [7]. They make use of Non-negative Matrix Factorization (NMF) to learn the spectral templates from the sample and detect the presence of these templates in the suspect audio. Correlating the activations from the sample and the song gives the likelihood of plagiarism.

While the authors provide an outline for sample detection, they do not offer a very detailed description of the specific algorithm. Sample detection, as a task in music information retrieval, hasn't yet been well defined in terms of methodology or evaluation. Publications are few and datasets related to sampling are non-existent or proprietary.

3. SAMPLE DETECTION ALGORITHM

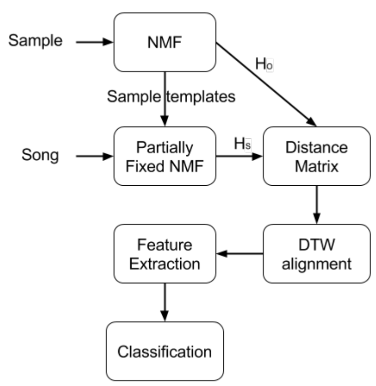

Figure 1. Block diagram showing the flow of the algorithm

The algorithm presented in this paper is based on the work by Dittmar et al. [7] The reason we chose to go with an NMF-based approach to sample detection was because of its prevalence in source separation tasks [16]. The task of sample detection is similar to a source identification problem where the sample is one of the sources present in the mix. The block diagram in Fig. 1 shows the high level processing steps of the algorithm.

3.1 Non-Negative Matrix Factorization

NMF is a widely popular algorithm in unsupervised learning with applications in recommendation systems [11] and signal processing [12]. NMF factorizes a signal $V \in \mathbb{R}^{M\times N}$ into a template matrix $W \in \mathbb{R}^{M\times K}$ and an activation matrix $H \in \mathbb{R}^{K\times N}$.

$$V = W \cdot H$$

If V is the magnitude spectrogram, W contains the K spectral or harmonic information in V while H contains temporal information about each corresponding spectral components in the template matrix [14].

Given the original sample, after RMS normalizing, downmixing and downsampling audio to 22050Hz, we compute its spectrogram(block size 4096, hop size 1024 samples). Similarly, we preprocess and compute the spectrogram of the song, which may or may not contain the sample. Using NMF, an original sample spectrogram will be factorized into its K templates, W_{o}, and activation matrix, H_{o}. A sample, used in a song, may be thought of as a source in a mixture of other sources in the song in question. Using the extracted templates W_{o} from the original sample, we can obtain the corresponding activations, H_{s}, in the song mixture by performing a partially fixed NMF [18] where the templates W_{o} are fixed and the mixture templates W_{m} are iteratively learned. In the subsequent analysis, we are only interested in the activations H_{s} corresponding to W_{o} since they indicate the presence of the sample in the song. Given that the sample wasn't pitch shifted or time-stretched, a cross-correlation between the activations H_{o} and H_{s} can be computed and peaks would show the presence of the sample. The 2-d cross-correlation between corresponding activation functions can be aggregated across the K dimensions and figure 2 shows results when the geometric mean was used for aggregation, for a true positive and a true negative detection.

3.1.1 *Pitch-shifting*

Pitch-shifting is a very common effect applied to samples before being used by artists in their own composition. It

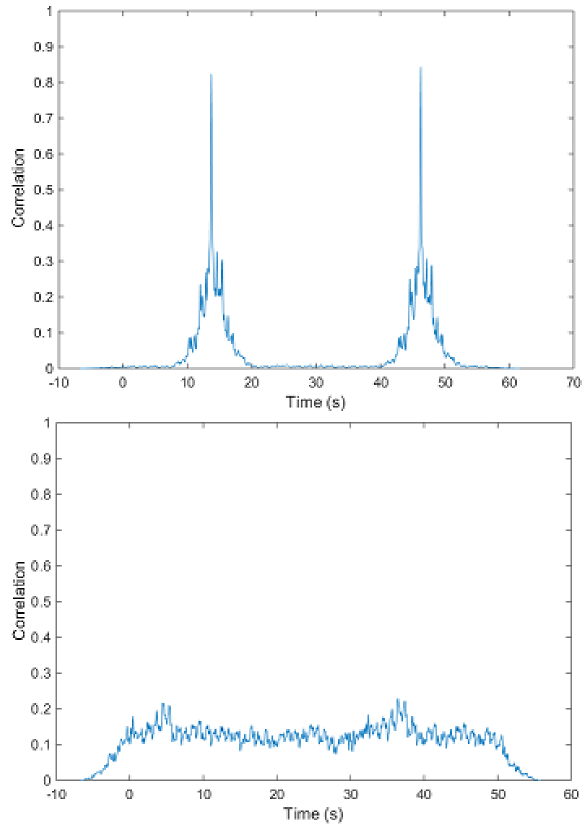

Figure 2. Geometric mean of correlation functions for when sample is present twice (above) and sample is absent (below)

refers to the process of changing the pitch of the original sample up or down. In case of pitch-shifting, the sample templates W_o will no longer be the bases in the mixture since the spectral content has shifted logarithmically in the frequency scale by the pitch-shift factor. In order to account for pitch-shifting, we scale the frequency axis of the templates by the hypothesized pitch-shift factor and create an extended W_o matrix. Now, a partially fixed NMF will be able to extract activations corresponding to each set of pitch-shifted templates and these may be compared to the activations from the original sample.

3.1.2 Time-stretching

Time-stretching is another common effect used in sampling. Artists most often change the speed of the sample in order to match their own song's tempo. In the case where a sample is time-stretched, the activations from the song will be similarly stretched and we can no longer use a cross-correlation since the activations will no longer align at a point where the sample is present.

In such a scenario, Dynamic Time Warping is used to align the activations H_o with the activations H_s at a start frame f in the song. A distance matrix is constructed using the pair-wise 2-d correlation between the K dimensional activations.

This problem is now a subsequence search for the sample activations H_o within the series of activations corresponding to the sample templates in the song, H_s. We

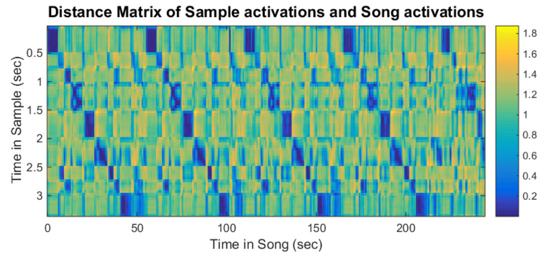

Figure 3. Distance matrix computed between activations in the case where a sample is looped

compute the cost matrix using DTW. The cost matrix is initialized by accumulating the distance matrix along the direction of sample only. The accumulated cost of alignment is an indicator of whether a sample is present at a particular frame in time or not.

3.1.3 NMF Rank Selection

We need to select parameters for the rank K of the sample spectrogram based on how many spectral templates can be used to approximate the sample. Similarly, while doing the partially fixed NMF, we need to define a rank to approximate the remaining mixture in the song so that the fixed sample templates are able to properly model the presence of the sample and reflect that in the activations.

Different songs will require different ranks for accurately approximating and factorizing the magnitude spectrograms with a low reconstruction error. In the current algorithm, fixed ranks are chosen empirically for both, the sample NMF and the mixture NMF. A possible extension could be to analyze the audio separately as a preprocessing step to obtain an approximate 'complexity' of the audio and set variable ranks based on the complexity of the spectrogram to be modeled.

3.1.4 Activation Normalization

In order for a correct sample detection, it is necessary to properly normalize the sets of activations extracted from the sample and the query song. Each set of activations is normalized by the absolute maximum across all the K activations across time. The idea is to preserve relative activation strengths for all the spectral templates of the sample.

$$H_{normalized} = \frac{H}{max(H_t^k : k \in [1, K], \forall t)}$$

3.2 Feature Extraction

To detect whether a sample is present in a song, DTW costs are computed for alignment paths backtracking from all end frames in the song and normalized by the length of the path. This mapping for every end frame in the song to the DTW cost for the path ending at that frame is called the DTW cost function. Figure 4 shows an example of this mapping. Ideally, the end frame where the sample ends will be a local or global minimum in the function.

Using an absolute threshold on the DTW cost to detect a sample is not meaningful because a low alignment

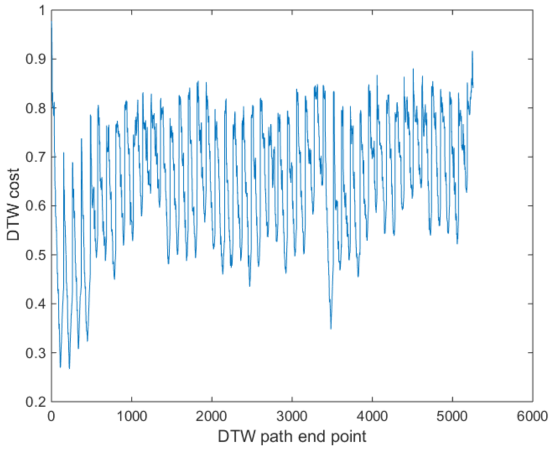

Figure 4. DTW cost function; Minima indicate the presence of the sample

cost in one song might not be a low cost in another song. The reason is that the mixing factor of samples in different songs may be different. Some might have a quiet signal at the sample sample whereas in other songs a sample might be heavily overlaid with other sounds. This leads to varying strength in activations across different song and sample pairs.

Another feature obtained from the DTW is the location in the song at which an alignment path started. Intuitively, given that a sample is present, the DTW backtracking path for end frames in the neighborhood of the exact location where the sample ends would also, after some DTW steps, merge into the optimal alignment path. Therefore, mapping the end points to the start points, we would observe a constant start point value for end points in the neighborhood of the location of the sample. This mapping is called the DTW path start function. Figure 5 shows one example of this function. A step in this function refers to ending frames that map to the same start point in the song after DTW.

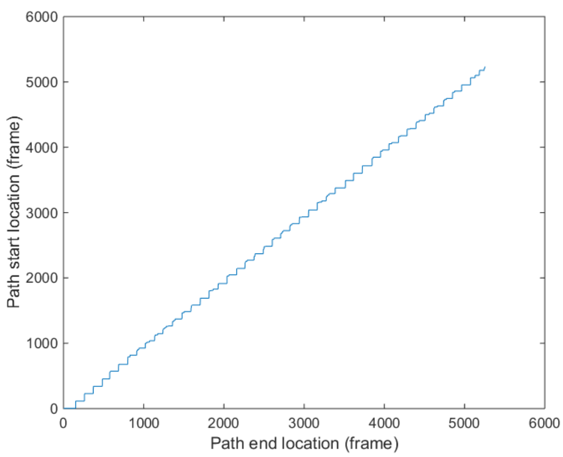

Figure 5. DTW path start function; Longer steps indicate sample

A valid assumption for each step in this function is that the local minimum of the DTW cost function should be considered as a candidate for sample detection. Hence for each song and sample pair, each unique start location in the DTW path start function is a candidate for classification and we extract the following features from the two aforementioned mappings as well as the DTW paths that were computed.

3.2.1 DTW Cost: 3 features

We extract the local minimum of the DTW cost for each end point corresponding to the unique start location. Along with the minimum, we also compute the mean and standard deviation of the cost over the set of points that map to the current step.

3.2.2 Path Features: 3 features

After extracting the local minimum DTW cost, we also extract the length and slope of the path for the minimum cost. We also compute the mean deviation of the path's slope from a straight line. In this discretized space, the deviation is computed using euclidean geometry. The distance matrix's first point is chosen as origin. The line joining the start point and end point is computed and for every other point in the path, we accumulate the perpendicular distance from the straight line and normalize by the length of the path.

In addition, we use the length of the step in the DTW path start function corresponding to the current unique start location. This gives us a 7-dimensional feature space.

3.3 Classification

Given the set of features extracted for each unique start location, the task at hand is a simple binary classification. The definition of an instance or a data point x in our case is: Every unique start location in the DTW path start function. The classifier $f(x)$ needs to decide whether a sample is present at the location denoted by x.

A random forest classifier was chosed for this task [3]. The input is a 7-dimensional feature vector described above and it classifies each datapoint as a location where the sample is present or not.

4. EVALUATION

4.1 Dataset

A dataset was compiled using whosampled.com [2] for this task. Whosampled is a website that aggregates information about songs that sample or cover other songs. The audio was downloaded using web services from streaming websites like Youtube or Dailymotion.

80 samples were selected with original songs from influential artists like James Brown, Stevie Wonder, Michael Jackson, and other artists. The songs that used these samples are from Hip-Hop, Pop and Rap genres in general with

[2] www.whosampled.com, last accessed: 1/22/2017

a few exceptions. The samples in this dataset cover one-shot samples of musical snippets or voice samples, looped drums and looped melody. The longest sample is 25 seconds, the shortest is half a second and the average length of the samples is 4.5 seconds. The total number of sampling instances is 876.

For each original song, the start and end time of the segment that was sampled was manually annotated. For each annotated sample, in the corresponding song that sampled it, all occurrences of the sample were annotated. All annotation was done using Sonic Visualizer [4]. In addition, the pitch-shift factors and the duration of the sample in the song were annotated.

These annotations along with the song names and URLs for obtaining the audio have been made available publicly. [3]

4.2 Experiments

For the current evaluation of the classifier, the first 40 samples from dataset were analyzed for the number of sample occurrences. Further, problematic samples were identified where there were no diagonals observed in the distance matrices. There were removed from the subset as well. Section **??** will talk more about these problematic samples. There are 322 sampling instances in this subset of the dataset.

For each sample and song pair, all unique start points after feature extraction were labeled as 0 or 1 based on whether a sample is present. In order to achieve this, the ground truth of the time instants of where samples were present are used. 10-fold cross-validation was used over this data. Since there is a very large number of 0 labels, another experiment is conducted with the skew between the 0 and 1 class labels is reduced to 2:1.

We also split the data into a separate training and test set to evaluate the algorithm in a similar fashion as above. A train-test split ratio of approximately 70:30 is used.

All these experiments are conducted on Weka [10].

5. FUTURE WORK

6. CONCLUSION

In this work, an algorithm for sample detection based on NMF and DTW, which is robust against pitch-shifting and time-stretching is presented. A framework for research in sample detection is described and a dataset a specifically created for this task. The paper describes a promising algorithm that uses NMF and DTW to solve the problem which is currently evaluated against a subset of the dataset. The results from the evaluation are encouraging and approaches to further improve the algorithm are proposed by way of normalization and pre-processing.

7. REFERENCES

[1] Donald J Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[2] Thierry Bertin-Mahieux and Daniel P. W. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 117–20, 2011.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] C. Cannam, C. Landone, and M. Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the ACM Multimedia 2010 International Conference*, pages 1467–1468, Firenze, Italy, October 2010.

[5] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of audio fingerprinting. *Journal of VLSI signal processing systems for signal, image and video technology*, 41(3):271–284, 2005.

[6] M. Casey and M. Slaney. Fast recognition of remixed music audio. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages 1425–1428, 2007.

[7] Christian Dittmar, Kay F Hildebrand, Daniel Gaertner, Manuel Winges, Florian Müller, and Patrick Aichroth. Audio forensics meets music information retrieval - A toolbox for inspection of music plagiarism. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1249–1253. IEEE, 2012.

[8] D. P. W. Ellis and G. E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1429–IV–1432, 2007.

[9] Dan Ellis. Robust landmark-based audio fingerprinting. `http://labrosa.ee.columbia.edu/matlab/fingerprint/`. Accessed: 2015-12-04.

[10] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[12] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[13] Joan Serrà, Emilia Gómez, and Perfecto Herrera. Audio cover song identification and similarity: Background, approaches, evaluation, and beyond. In *Advances in Music Information Retrieval*, volume 274 of

[3] www.github.com/placeholder_repo

Studies in Computational Intelligence, pages 307–332. Springer Berlin Heidelberg, 2010.

[14] Paris Smaragdis and Judith C Brown. Non-negative matrix factorization for polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 177–180. IEEE, 2003.

[15] Jan Van Balen. Automatic recognition of samples in musical audio. In *Masters thesis, Universitat Pompeu Fabra*, 2011.

[16] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE transactions on audio, speech, and language processing*, 15(3):1066–1074, 2007.

[17] Avery Wang. An industrial-strength audio search algorithm. In *Proceedings of the 4th International Conference on Music Information Retrieval*, 2003.

[18] Chih-Wei Wu and Alexander Lerch. Drum transcription using partially fixed non-negative matrix factorization with template adaptation. In *ISMIR*, pages 257–263, 2015.