AUTOMATIC SAMPLE DETECTION IN POLYPHONIC MUSIC

Siddharth Gururani, Alexander Lerch
Georgia Institute of Technology, Center for Music Technology
{siddgururani, alexander.lerch}@gatech.edu

ABSTRACT

The term 'sampling' refers to the usage of snippets or loops from existing songs or sample libraries in new songs, mashups, or other music productions. The ability to automatically detect sampling in music is, for instance, beneficial for studies tracking artist influences geographically and temporally. We present a method based on Non-negative Matrix Factorization (NMF) and Dynamic Time Warping (DTW) for the automatic detection of a sample in a pool of songs. The method comprises of two processing steps: first, the DTW alignment path between NMF activations of a song and query sample is computed. Second, features are extracted from this path and used to train a Random Forest classifier to detect the presence of the sample. The method is able to identify samples that are pitch shifted and/or time stretched with approximately 63% F-measure. We evaluate this method against a new publicly available dataset of real-world sample and song pairs.

1. INTRODUCTION

In the context of music composition and production, *sampling* stands for the concept of reusing pre-existing digital recordings in new compositions in a way that it fits the musical context. In digital sampling, an artist records a segment of a song or sound that they wish to sample, possibly modifies it, and then reuses it (and possibly other samples) by incorporating it into a new composition [11]. Sampling of audio has become popular in mainstream pop, hip-hop, and rap music.

A Sample Detection (SD) system automatically detects samples from a pool of songs and thus enables musicological studies of the influence of older artists over newer generation artists by observing sampling patterns over the years and geographically. Another possible use case of a SD system could be to detect plagiarism or copyright infringement. Sampling is legally controversial and determining fair use is largely left to the judicial system. 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 as well as its time location in the song.

2. RELATED WORK

The task of SD has been addressed in only few previous publications. There exist, however, several parallels that may be drawn from other areas of research that are relevant to SD such as cover song detection, audio fingerprinting, and remix recognition.

2.1 Audio Fingerprinting

Audio Fingerprinting (AFP) refers to the method of extracting content-based signatures from audio [1, 5, 10]. It is commonly used in content-based music identification systems such as Shazam. [1] Van Balen proposed the use of AFP for sample detection [24, 25], using a popular fingerprinting algorithm by Wang [27] in an implementation by Ellis [8].

Fingerprinting systems are robust against noise injection and, with appropriate modifications, pitch shifting and time stretching [31]. As long as the level difference between the sample and other sources is within the noise level expectations, a modified system could be a good choice for sample detection. Given that a sample might be mixed at a low level, it is questionable if this assumption really holds true for the majority of cases.

2.2 Cover Song Detection

Cover Song Detection (CSD) is the task of recognizing whether a given reference track has a cover song in a set of test tracks [2,9,19]. Covers may, for example, be transposed and deviate from the original song in terms of tempo and other properties. Dynamic Time Warping (DTW) [20] is often used to make the comparison tempo-invariant. The difference from SD is that covers are renditions of a musical piece, while samples are snippets of audio which are usually a part of the mix overlaid with multiple other instruments and sounds unrelated to the sample. The evaluation of SD systems, however, is similar to that of CSD. Both have a test/reference pair which is then categorized as a positive or negative match with or without a confidence measure.

2.3 Remix Recognition

Work done in remix recognition by Casey and Slaney [6] draws inspiration from a method for web crawling called

[1] https://www.shazam.com, last accessed: 4/26/2017

'shingling' which utilizes a stream of text position-based features to detect if a document has already been crawled before. In their work, they compute audio features such as pitch-class profiles and Log-Frequency Cepstral Coefficients using 0.1 second frames. They concatenate these feature vectors into 4 second 'shingles' and model the distribution of pair-wise distances between shingles from remixes and shingles that are not from remixes. A nearest neighbor classifier is used to identify which distribution a query 'shingle' belongs to. Such a system is not appropriate for SD since this method relies on long-term similarity between the query and the reference, however, a sample may be a short, one-shot sample (triggered samples without looping). A working remix recognition system, however, would be helpful in ways similar to SD in that it helps with tracing influences across artists. In addition, some remixing cases might also involve instances of sampling.

2.4 Sample detection with Non-negative Matrix Factorization

Dittmar et al. listed sampling as one of three kinds of plagiarism in music [7]. They utilize 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 then gives the likelihood of plagiarism. While the authors provide a general outline of a sample detection system, they neither offer a detailed algorithmic description nor a formal evaluation of their proposed system. In [28], Whitney uses NMF in a similar manner except that instead of factorizing entire spectrograms, NMF is applied to short texture windows in the sample and the song. The detection is done using pairwise 2-dimensional cross-correlation of the two activation matrices obtained. To account for pitch shifting and time stretching, the audio files are resampled using factors computed by taking the ratio of the sample and song BPM and multiple NMFs are performed. The issue with this approach is that the time stretching and pitch shift factor are not necessarily inverse when sampling. Nonetheless, NMF appears to be a good choice for a SD system as fixed templates allow to obtain activations for only the common components between the song and the sample.

The task of sample detection has been identified in music information retrieval literature, and various approaches have been proposed. However, it has not yet been well defined in terms of evaluation methodology or metrics. Reasonably sized datasets are also non-existent or proprietary. Therefore, there is no formal evaluation that can be performed to compare different sample detection methods. This paper aims to bridge this gap by providing a dataset and propose a common evaluation framework.

3. METHOD

The algorithm presented in this paper is inspired by the proposal of Dittmar et al. [7]. Since the task of sample detection is similar to a source identification problem where the sample is one of the sources present in the mix, an NMF-based approach is fitting due to its prevalence in audio source separation tasks [18, 21]. The block diagram in Figure 1 shows the processing steps of the algorithm.

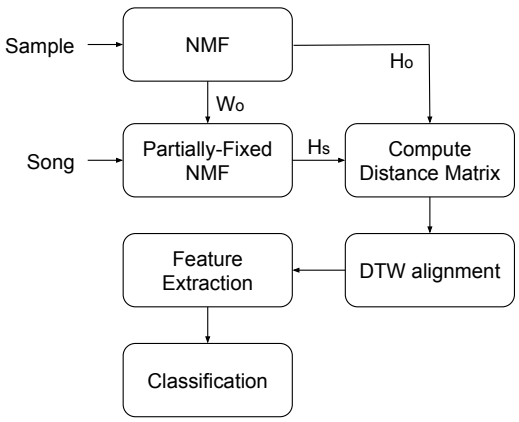

Figure 1. High Level Block Diagram

3.1 Non-Negative Matrix Factorization

NMF is a widely popular algorithm in unsupervised learning with applications in recommendation systems [12, 15, 16] and signal processing tasks, specifically source separation [13, 23, 26]. NMF factorizes a signal matrix $V \in \mathbb{R}^{M \times N}$ into a template matrix $W \in \mathbb{R}^{M \times K}$ and an activation matrix $H \in \mathbb{R}^{K \times N}$ such that:

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

If V is the magnitude spectrogram with M frequency bins and N blocks, W contains the K spectral or harmonic component templates in V while H contains temporal information about each corresponding spectral component in the template matrix [22].

In a pre-processing step, both the original sample and the paired song are RMS-normalized, downmixed, and downsampled to 22.05 kHz; then, their magnitude spectrogram is computed (block size: 4096, hop size: 1024 samples). Using NMF, the sample spectrogram will be factorized into K templates W_{o} and the activation matrix H_{o}, o indicating 'original'. A sample, used in a song, may be thought of as one source in a mixture of multiple sources in this song. The factorization of the song will then be performed using partially fixed NMF (PFNMF) [29, 30]. In this case, the template matrix W consists of a fixed, not updated, part containing the extracted templates W_{o} and a randomly initialized part W_{m} with L templates that is iteratively learned and represents what we refer to as the mixture templates. The dimension of the complete template matrix is thus $M \times (K + L)$. All activations are iteratively updated as well.

3.1.1 NMF Rank Selection

The rank K of the sample spectrogram has to be chosen based on how many spectral templates can approximate the

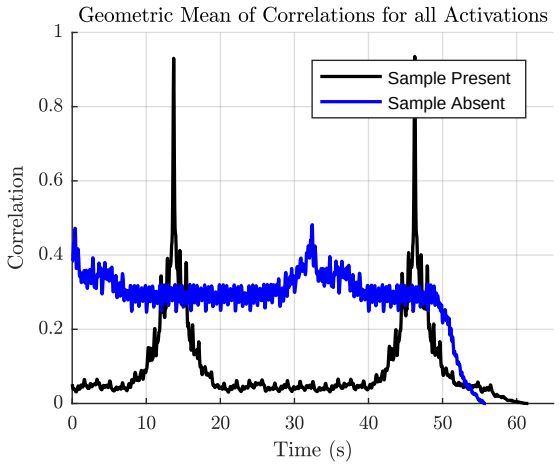

Figure 2. Geometric mean of cross-correlation functions. Black function shows sample occurrence, while blue does not have any samples.

specific sample. Similarly, while computing the PFNMF, the rank L for approximating the templates representing the remaining mixture in the song has to be selected in a way that minimizes the impact on the fixed sample template activations in order to robustly detect the sample.

Different songs will usually require different ranks for accurately approximating and factorizing their magnitude spectrograms with a low reconstruction error. In the current algorithm, however, fixed ranks are chosen: $K = 10$ and $L = 20$. The rationale behind using fixed low ranks is that for this task a perfect reconstruction is not required and that the following processing steps should be robust enough to detect the sample regardless of whether the templates are able to combine linearly to an accurate reconstruction of the original spectrogram. If the sample is used in a song, the same fixed templates should produce a roughly similar set of activations independent of the mixture rank. Larger fixed ranks were tested but no gain in performance was observed while resulting in increased computational cost.

Still, a future extension might be to analyze the audio separately as a pre-processing step to obtain a 'complexity' measure that could be used to adapt the ranks K and L based on the signals to be modeled.

3.2 Activation function processing

In the subsequent analysis, only the activations H_s are of interest. These are the activations corresponding to the original templates W_o after PFNMF and indicate the presence of the sample in the song if they match the pattern of the original activations H_o. If the sample were neither pitch shifted nor time stretched, cross-correlation functions between each corresponding activation in H_o and H_s could be computed. Peaks in the aggregated (across the K dimensions) cross-correlation function would then indicate the presence of the sample. Figure 2 shows two example functions after using the geometric mean for aggregation: a sample occurs twice in the black song (as indicated by the two local maxima), while it is not present in the blue song.

3.2.1 Activation Normalization

Proper normalization of the activation matrix is essential for accurate sample detection, however, there is no "correct" way to do this as the level (or even the presence) of the sample in the paired song is unknown. The relative activation levels between the K templates of one matrix, however, should remain identical. Thus, each activation matrix H is normalized by the absolute maximum across all the K activations across time, preserving the relative activation strengths for the spectral templates of the sample: $H_{normalized} = \frac{H}{\max(H)}$.

3.2.2 Pitch shifting & Time stretching

The assumption that the sample is neither time stretched nor pitch shifted is false for the majority of cases. In the dataset used for this study, for example, 57.5% of samples are pitch shifted. Pitch shifting is often required for the sample to match the tonality of the song, and time stretching is often required to adjust for tempo differences between the two.

In case of pitch shifting, the original templates W_o will no longer be valid templates since the frequency axis of the spectral content will be scaled by the pitch shift factor. In order to account for pitch shifting, we construct new spectral templates W_o^p by scaling the frequency axis of the original templates with a set of hypothetical pitch shift factors. Now, a partially fixed NMF will be able to extract activations corresponding to the pitch shifted templates and these may be compared to the activations from the original sample. Ideally, we would perform factorizations for a set of hypothetical pitch shifts usually ranging from one octave lower to one octave higher in semi-tone or quarter-tone steps. For this paper, however, we allow the dataset to inform our pitch shift steps. The used pitch shift set in semi-tone steps from the original sample templates W_o is:

$$P = \{p \mid p \in \{-5, -4, -3, -2, -1, 0, 0.5, 1, 2, 3, 4, 5\}\}.$$

Partially-fixed NMF is then performed individually for each pitch shift in P, i.e., 12 times. The activations H_s^p correspond to the pitch shifted templates W_o^p.

When the sample is time stretched, the activations from the song will be stretched or compressed accordingly; therefore, cross-correlation cannot be used. In such a scenario, DTW can align the activations H_o with the activations H_s^p in the song for each pitch shift p. A distance matrix D is constructed using the pair-wise correlation distance between the K-dimensional activations. The size of D is $N_\mathrm{o} \times N_\mathrm{s}$, where N_o is the number of frames in the original sample activations and N_s is the number of frames in the song activations. Correlation is used as the distance measure because it is scale independent. Preliminary tests showed that other frequently used distance measures perform either similarly or worse. The resulting distance matrix shows how similar a set of activations is to the extracted activations at each time instant. The distance matrix D and properties of the extracted path can be used to indicate the presence of a sample in a song. Figure 3 shows an example of a distance matrix with a looped sample. The low cost parallel blue paths indicate the presence of the sample.

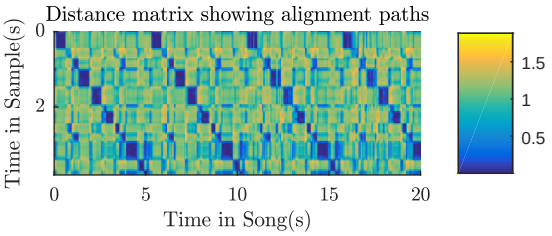

Figure 3. Distance matrix computed between activations in the case where a sample is looped

The 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}^{p}. A standard DTW implementation would initialize the cost matrix C by accumulating the first row and first column of the distance matrix D. Such a scheme can be applied when only one global optimal path that aligns the two entire time series is required. In the case of sample detection, a sample could be present in multiple locations within the song; hence, the detection of multiple alignment paths at multiple locations in D is required. Each of these paths should align the entire sample with segments of the song. Therefore, the initialization of C is modified to only accumulate the distances along the dimension of the sample, which in our case is the column. This subsequence DTW scheme [17] initializes the cost matrix C as follows:

$$\begin{aligned} C(1,j) &= D(1,j) \\ C(i,1) &= \sum_{k=1}^{i} D(k,i) \end{aligned} \tag{2}$$

Initializing in this fashion and proceeding to compute the cost matrix allows us to obtain backtracking paths from every index in the last row of D (as opposed to one path from the last element of D, which is the case for "standard" DTW). The last row of the cost matrix C corresponds to the cost of aligning the sample backtracking from every frame of the song. The backtracking paths now satisfy the requirement of aligning the entire sample with a section in the song. To summarize, every alignment path obtained is the path that aligns the sample activations H_{o} backtracking from every frame f in the song activations H_{s}^{p}. Note that the subsequence DTW is performed for each $p \in P$.

3.2.3 Pitch Candidate Selection

From the set of pitch shifts, the most likely pitch shift is inferred before feature extraction. Of the 12 cost matrices, the one corresponding to the most likely pitch candidate will be the one with the global minimum cost (minimum of the last row of the cost matrix),i.e., the one with a minimum cost lower than the minimum cost of all other matrices. All subsequent computations are based on the activation matrix of the selected candidate; the results for the 11 remaining pitch shifted templates are discarded.

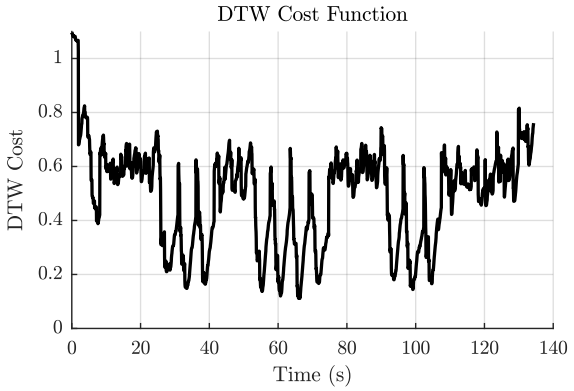

Figure 4. DTW cost function; minima indicate the end of the sample

3.3 Feature Extraction

The last row of the cost matrix C, containing the alignment path costs normalized by the length of the path, will be referred to as *DTW cost function*. Local minima in this DTW cost function indicate potential sample end points.

Using an absolute threshold on the DTW cost function to detect a sample is not meaningful because the absolute cost level depends on both sample and song characteristics. The mixing ratio of different samples in different songs will be different. Some samples might occur in a section with no other sources while others might be heavily overlaid with other sounds, leading to interference and varying strength in activations across different (song, sample) pairs. The DTW backtracking paths from all possible end points to their corresponding start location results in a set of unique start locations in the song for each (song, sample) pair. We refer to this mapping from every end point to a start point in the song as *DTW path start function*. Every unique start location is a candidate for a sample being present. Note that each unique start location can have multiple paths/end points. Figure 5 shows one example of this function. We choose to derive features from the two functions we defined above because we expect that the properties of the DTW alignment paths vary depending on whether a sample is present or not. The features extracted from this data are explained in the following sections.

3.3.1 Cost-based Features

From the multiple path end points of each start location, the following three features are extracted: (i) the minimum DTW cost across all end points, (ii) the average DTW cost across all end points, and (iii) the standard deviation of the DTW cost across all end points.

3.3.2 Path-based Features

The properties of the DTW alignment path should be meaningful for detecting the presence of the sample. For example, the tempo of the sample should stay roughly constant, meaning that the slope of the path stays constant as well. Thus, the idealized path would connect start point and end point with a straight line. Note that when we refer to end

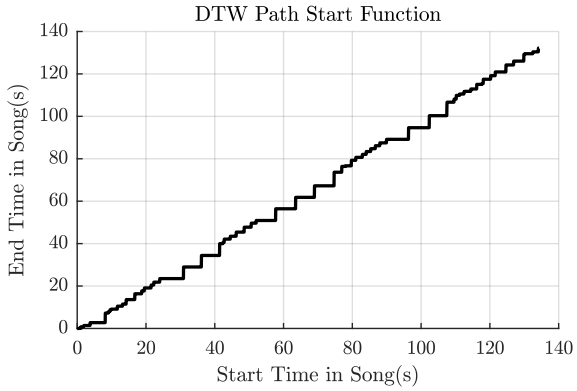

Figure 5. DTW path start function; Longer steps indicate sample; Long steps indicate that several DTW paths backtracked to the same start point.

points here, we refer to the group of end points that map to one unique start location. Overall, the following features are extracted for every unique start location: (i) the absolute length of the minimum cost path normalized by the sample length, (ii) the slope of the minimum cost path, (iii) the average perpendicular deviation of the minimum cost path from the idealized path, normalized by the length of the path, (iv) the average slope across all end point paths, (v) the standard deviation of the slope across all end point paths, (vi) the average absolute length of all end point paths normalized by the sample length, (vii) the standard deviation of the absolute length of all end point paths normalized by the sample length, (viii) the average perpendicular deviation from the idealized path across all end point paths, normalized by the length of the paths, (ix) the standard deviation of the perpendicular deviation from the idealized path across all end point paths, normalized by the length of the path, and (x) the number of end points mapping to this unique start location.

3.4 Classification

Each unique start location is represented by a 13-dimensional feature vector, which is a data point that can be used as the input to a binary classifier for detecting whether a sample is present or not. A random forest classifier with an ensemble of 200 decision trees was chosen [3]. The number of features chosen for each decision split is 4 and has been decided based on the convention of choosing $round(\sqrt{\mathrm{n}})$ where n is the number of features, 13. The output is a probability of the data point belonging to each class.

4. EVALUATION

4.1 Dataset

A dataset for Sample Detection was compiled using data from whosampled.com [2] which aggregates information about songs that sample or cover other songs. The audio was downloaded using web services from streaming websites such as Youtube or Dailymotion.

Eighty popular samples (according to the users of Whosampled) were selected from the catalog of songs by popular and frequently sampled artists such as James Brown, Stevie Wonder, Michael Jackson, and Queen. The resulting set has a balanced distribution among the genres Pop, Rock, Funk and Hip-Hop.

The samples cover several variations of sampling such as: one-shot samples of musical snippets or voice samples, looped drums, and looped melodies. The length of the longest sample is 25 s, the shortest is half a second and the average length of the samples is 4.5 s seconds. The total number of sampling instances is 876. The overall dataset contains 80 pairs of original song and sampling song.

The following annotations were added manually with the software Sonic Visualizer [4]: (i) start and end time in seconds of the sample in the original song, (ii) start time in seconds of the sample in the sampling song, and (iii) pitch shift (in semi-tones) of the sample in the sampling song. 57.5% of the samples are pitch shifted. Pitch shift was annotated by a human listener by ear. In cases where the pitch shift was difficult to ascertain, the sample and song snippet was compared in a DAW and different pitch shifts were tested until a match was found. These annotations plus additional meta-data including the song names, identifiers, and URLs for obtaining the audio have been made available publicly in an online repository. [3] The repository also contains the MATLAB source code for the algorithm.

4.2 Experiments

For each of the 80 samples in our dataset, there are 79 songs in which the sample does not occur. We randomly pick 9 songs from these 79 songs. This, in combination with the one song that includes the sample, results in a pool of 10 songs to be paired with each sample, resulting in an overall number of queries: $80 \cdot 10 = 800$ for the 80 samples. Performing experiments with all possible pairs in the entire dataset would be impractical without more compute power. This overall set is split into a training set of 50 samples/500 queries and a test set of 30 samples/300 queries.

In order to use the ground truth data for training, one additional step of interaction is necessary: all unique start points have to be labeled as '0' or '1' based on whether a sample is present. More specifically, the start points associated with minimum DTW cost within a 1 s window of the ground truth annotation are labeled '1' and the rest are labeled '0'. Multiple start points within the tolerance range are merged so that the minimum cost path remains.

For the test set, any positive detection of the sample within a 1 s tolerance window of the ground truth annotation will be regarded as True Positive. Every positive detection outside of this tolerance window and for queries not containing the sample will be regarded as False Positive.

With respect to the metrics, the False Positive Rate, Precision, Recall, and the F-measure are reported for the sample location detection. We refer to these as *Micro*-accuracy measures as they take into account the sample location and the number of occurrences.

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

[3] https://github.com/SiddGururani/sample_detection

	Precision	Recall	F-measure	Fp-rate
Micro	79.37%	34.60%	48.19%	0.04%
Macro	83.33%	50.00%	62.50%	1.11%

Table 1. Results for song-level macro accuracy and sample location-level micro accuracy measures

Macro-accuracy measures, on the other hand, report the song-level sample detection results and indicate whether a sample is present in a song or not regardless of position and number of occurrences. The same metrics are reported, but the sample detection is evaluated per song rather than per sample instance. In summary, using both the Macro and Micro-level accuracy metrics we are able to report the performance of the method in two usage scenarios: First, detecting whether a sample is present in a song (Macro), and second, detecting where in a song and how often a given sample is present (Micro).

5. RESULTS AND DISCUSSION

Table 1 reports the test accuracy of the classifier. The results show that the presented method is somewhat effective at detecting the presence of sampling in a set of songs with a low false-positive rate and a reasonably high precision. The low recall of the method can be attributed to the highly imbalanced nature of the problem as depicted in the confusion matrix in Table 5: the testing dataset has 289 instances of sampling against around 74,000 instances that are not locations of sampling. The training set is similarly skewed in its distribution of positive and negative classes. We observe an area under receiver operating characteristic curve (AUROC) of 72.54%, a result strongly impacted by the low false positive rate due to the imbalanced classes.

A possible reason for the low recall is also that the method is not always accurate when it comes to picking candidates. More specifically, it already misses approximately 7% of sampling instances during the candidate selection stage. This number is computed by classifying all unique start locations as sample instances and calculating the number of false negatives. A closer investigation of some training samples showed that sometimes the DTW cost function did not have a minimum at respective sample locations. In these cases, the distance matrix would not contain clear alignment paths like the one shown in Figure 3. The absence of clear alignment paths might stem from incorrect modeling of the fixed sample templates in the song NMF step or pitch shifts not considered in our algorithm. The choice of the distance measure might also impact the results: while the use of the correlation distance makes sense because it is scale invariant, custom distance measures might outperform it in this particular use-case.

It is worth pointing out that most problematic cases that we came across were hard to detect for humans as well. These cases included sparse drum loops, very short samples, samples that were mixed at a very low level, and samples with excessive use of audio effects applied to them. The high precision of our method, especially in the macro-accuracy use case, enables utilizing this system as a preprocessing step to a manual detection of sampling instances, e.g., for studies that trace artist influences. Given a database of songs and a set of samples to look for in the database, this method can be used to pre-label data as cases of sampling with high confidence on songs where samples are detected, allowing the human operator to focus on the remaining database. In the context of plagiarism detection, a high precision enables such a system to be used with high confidence in case of a positive detection of plagiarism. However, a low recall system "favors" the sampling artists so the involvement of human experts remains a requirement.

Micro	Not Sample	Sample
Not Sample	74126	26
Sample	189	100

Macro	No Sample	Has Sample
No Sample	267	3
Has Sample	15	15

Table 2. Confusion Matrices for Micro & Macro Accuracy

6. CONCLUSION AND FUTURE WORK

We introduced a method to detect the presence of a sample in a set of songs robust against common sampling modifications such as pitch shifting and time stretching. PFNMF is used to extract sample activations from the song, and DTW is used to align the set of activations obtained from the song and the sample. We also present a new publicly available dataset of real-world samples and songs containing fine-grained annotations for exact time locations of sample occurrences within the song. The presented method is evaluated against this dataset and we obtain 79.37% precision in detecting the exact location of the sample and 83.33% precision in song-level detection of a given sample.

Further research is required in order to improve the usability of this method. Since the algorithm works for many cases, a systematic way to improve it would be to more closely investigate the problematic cases in order to design modifications and algorithmic extensions to increase recall.

As this problem is inherently unbalanced, a possible direction is to observe best practices for machine learning on imbalanced datasets. In addition to undersampling, other techniques such as algorithmic modifications and cost-sensitive learning that may be employed to solve imbalanced classification problems [14].

Investigating custom distance measures for the DTW is another possible avenue to explore. Applying a non-linearity to the NMF activations may help in increasing the sparsity, possibly improving the differentiation between an instance containing a sample and one that does not.

Sample detection is an intriguing and challenging, yet largely untouched MIR task and it is our hope that the dataset and this paper will encourage future work on this topic in the MIR community.

7. REFERENCES

[1] Shumeet Baluja and Michele Covell. Audio fingerprinting: Combining computer vision & data stream processing. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 213–216, Honolulu, HI, USA, 2007.

[2] Thierry Bertin-Mahieux and Daniel P. W. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 117–120, New Paltz, NY, USA, 2011.

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

[4] Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proc. of the ACM International Conference on Multimedia*, 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] Michael Casey and Malcolm Slaney. Fast recognition of remixed music audio. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1425–1428, Honolulu, HI, USA, 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 *Proc. of the European Signal Processing Conference*, pages 1249–1253, Bucharest, Romania, 2012.

[8] Daniel P. W. Ellis. Robust landmark-based audio fingerprinting. `http://labrosa.ee.columbia.edu/matlab/fingerprint/`. Online; accessed: 28-April-2017.

[9] Daniel P. W. Ellis and Graham E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1429–1432, Honolulu, HI, USA, 2007.

[10] Jaap Haitsma and Ton Kalker. A highly robust audio fingerprinting system. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 107–115, Paris, France, 2002.

[11] Mark Katz. *Capturing sound: how technology has changed music*. University of California Press, Berkeley and Los Angeles, 2004.

[12] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

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

[14] Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.

[15] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284, 2014.

[16] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proc. of the ACM Conference on Information and Knowledge Management*, pages 931–940, Napa Valley, CA, USA, 2008.

[17] Meinard Müller. Dynamic time warping. In *Information Retrieval for Music and Motion*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[18] Alexey Ozerov and Cédric Févotte. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563, 2010.

[19] Suman Ravuri and Daniel P.W. Ellis. Cover song detection: from high scores to general classification. In *Proc. of the IEEE International Conference on Acoustics Speech and Signal Processing*, pages 65–68, Dallas, TX, USA, 2010.

[20] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[21] Mikkel N Schmidt and Morten Mørup. Nonnegative matrix factor 2-d deconvolution for blind single channel source separation. In *Proc. of the International Conference on Independent Component Analysis and Blind Source Separation*, pages 700–707, Chareston, SC, USA, 2006. Springer.

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

[23] Paris Smaragdis, Cedric Fevotte, Gautham J Mysore, Nasser Mohammadiha, and Matthew Hoffman. Static and dynamic source separation using nonnegative factorizations: A unified view. *IEEE Signal Processing Magazine*, 31(3):66–75, 2014.

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

[25] JMH van Balen, Martín Haro, and Joan Serrà. Automatic identification of samples in hip hop music. In *Proc. of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, pages 544–551, London, UK, 2012.

[26] 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.

[27] Avery Wang. An industrial-strength audio search algorithm. In *Proc. of the International Conference on Music Information Retrieval*, pages 7–13, Baltimore, MD, USA, 2003.

[28] Jordan L Whitney. *Automatic recognition of samples in hip-hop music through non-negative matrix factorization*. PhD thesis, University of Miami, 2013.

[29] Chih-Wei Wu and Alexander Lerch. Drum Transcription using Partially Fixed Non-Negative Matrix Factorization. In *Proc. of the European Signal Processing Conference*, pages 1281–1285, Nice, France, 2015. EURASIP.

[30] Chih-Wei Wu and Alexander Lerch. Drum transcription using partially fixed non-negative matrix factorization with template adaptation. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 257–263, Malaga, Spain, 2015.

[31] Bilei Zhu, Wei Li, Zhurong Wang, and Xiangyang Xue. A novel audio fingerprinting method robust to time scale modification and pitch shifting. In *Proc. of the ACM International Conference on Multimedia*, pages 987–990, Firenze, Italy, 2010. ACM.