klustr: a tool for dimensionality reduction and visualization of large audio datasets

Lamtharn Hantrakul, Avneesh Sarwate

MT Hauz
1065 Terrell St.
Georgia Tech Center for Music Technology
Atlanta, GA 30318

ABSTRACT

We present *klustr*: a tool for automatic dimensionality reduction and 2D visualization of large audio datasets. The tool facilitates visual navigation and discovery of similar sounding samples in a collection, allowing musicians and researchers to quickly find sounds that are similar to one another based on perceptual features such as timbre. *Klustr* forms relationships between samples based on the audio file and not high-level language-based descriptors. In this paper, we present a version of *Klustr* that is optimized for navigating drum samples typically found in pop, hip-hop and electronic music.

1. INTRODUCTION

From speech, images and sensor input, data from the natural world is often high dimensional. Instead of interacting directly with this high dimensional data however, a collection of audio sample is often instead organized using simple high-level descriptors, such as the type of sound e.g "vocal_shout". However, these labels are often not available, and when available, do not capture the nuances of relationships between sounds. In order to build relationships between samples at the timbre level, we must form representations drawn from the high dimensional audio data. These are typically in the form of STFT or MFCC features, but even these are in dimensions in the order of several thousands. This paper presents an approach of allowing users to navigate these relatively high dimensional features in 2D space.

Dimensionality reduction techniques are normally used for two purposes. Methods such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) are often used to extract key components that capture the distribution of the data using a smaller number of dimensions. This reduced data often yields better performance in classic machine learning algorithms such as Support Vector Machines (SVM) [1].

Alternatively, dimensionality reduction can be used to visualize the representation learned by non-linear models such as neural networks. A highly popular stochastic dimensionality reduction algorithm called t-distributed neighbor embedding (TSNE) [2] has become the standard tool for visualizing embeddings learnt by deep convolutional neural networks [3]. TSNE is particularly adept at preserving spatial relationships in higher dimensions but in 2 or 3 dimensions. This enables researchers to grasp an
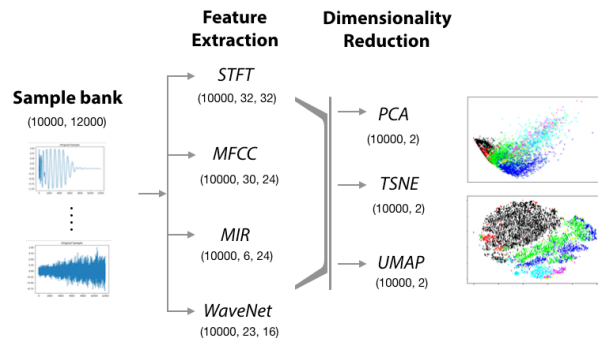

Figure 1: klustr pipeline

intuition of their model's behavior and tune hyperparameters accordingly.

In this paper, we use these dimensionality reduction techniques to directly encode features into a 2 dimensional plane. Our purpose is not to find the best set of features for a classifier, nor to understand the internal representation of a neural network. Instead, we are attempting to take advantage of dimensionality reduction to generate a navigable 2D map where neighboring samples are similar in sounds e.g a raspy snare is located next to another gritty snare, but is further away from a kick drum. We see this tool being used in navigating online sample libraries, sound archives and as a desktop tool for musicians and creators in the entertainment industry from music to Hollywood.

2. RELATED WORK

Navigable audio datasets, when labelled, are often presented to the user in an ontology of high level sound descriptors. AudioSet for example [4], divides two million hand annotated samples into categories such as "human sounds" and "sounds of things". The UrbanSound dataset uses tags such as "Mechanical" and "Music" sounds [5]. Alternatively, one can organize using features based off the audio signal itself.

In the domain of creating end-user interfaces that arrange large sound datasets in a 2D layout using audio features, Fried et al [6] used a two-step method to embed audio samples into a 2D grid. The authors first used a metric learning algorithm such as LDA to transform their feature space into a representation. Next, kernalized sorting is used to match samples to defined "tiles" on a grid. Given two sets A and B, the kernalized sorting algorithm

can use the functions f(A) and f(B) to generate a mapping A → B.

Many open-sourced projects have explored this 2D mapping. Two projects from Google Creative Lab take advantage of TSNE to create a fun-to-use 2D grid of bird sounds [7]. Another project called the *Infinite Drum Machine* mapped a collection of urban sounds into a 2D map and then colored them according to whether they were "kickdrum-like" or "snare-like" using clustering of STFT features [8]. Lastly, a recent Medium blog post outlines a comparison of various dimensionality reduction techniques used on short audio clips [9]. These projects are popular examples used in online media to demonstrate the usefulness of dimensionality reduction. However, to our knowledge, no industry tool exists that robustly implements these techniques for end-users to use on user-specified sample collections.

3. CONTRIBUTIONS

Our implementation of *Klustr* is based on work by Kyle McDonald in the Infinite Drum Machine [7] and Leon Fedden [9]. We extend their work by applying feature extraction and dimensionality over a large database of drum samples (approximately 10,000). Unlike these previous implementations, our ground truth for each sample is known. We scrape the tags included in sample bank filenames such as "kick_1.wav" and "real_tom.wav" to label the sound. This enables us to evaluate, to a certain extent, the quality of the clustering, separation and layout of samples on the 2D plot. We also experiment with novel techniques recently introduced in machine learning literature, notably UMAP [11] and WaveNet architectures [12].

4. ALGORITHM OVERVIEW AND DESCRIPTION

4.1 Extraction and Dimensionality Reduction Pipeline

Like Fried et al [6] we adopt a two step process. The first entails feature extraction from the audio file followed by a second step of dimensionality reduction. We standardize the length of all drum shots to the first 0.25 seconds at a sampling rate of 48kHz. This enables us to capture essential attack transients of different drum types, while keeping file sizes manageable. We use a collection of feature extraction methods:

- *Short Time Fourier Transform (STFT):* Window-size of 2048 and hop-size of 512
- *Mel Frequency Cepstral Coefficients (MFCC):* We selected the first 30 MFCC's
- *Hand-crafted MIR features*: RMSE, spectral centroid, spectral crest, spectral flux, spectral rolloff and zero crossing rate
- *Wavenet Autoencoder features*: We use the state-of-the-art WaveNet architecture to encode the samples into a compressed representation at the

model bottle-neck and interpret this as the extracted "feature"

We then use a variety of dimensionality reduction techniques on these features:

- *Principal Component Analysis (PCA)*
- *T-distributed Stochastic Neighbor Embedding (TSNE)*
- *Uniform Manifold Approximation and Projection (UMAP)* – *[NOTE: Unfinished work in this draft]*

The pipeline and resulting dimensions at each step are summarized in figure 1.

4.2 Dataset

The dataset is composed of short "one-shot" drum samples typically found in sample libraries. The distribution of sounds is summarized in table 1.

Drum Type	Samples
Kick	5158
Tom	422
Snare	2546
Clap	1324
High Hat	159
Ride	228
Crash	723
Total	**10,560**

Table 1: Dataset of drum "one shots"

4.3 Unsupervised Clustering Feature Selection

We use unsupervised k-means clustering to first evaluate the set of features most useful for separating samples into different classes in high dimensional space (the original dimensionality shown in "Feature Extraction" in figure 1). We reason that features useful to high-dimensional clustering should be useful to the dimensionality reduction algorithm in grouping and separating samples into classes in lower 2 dimensional space (the reduced dimensionality shown in "Dimensionality Reduction" in figure 1).

To evaluate the clusters, we take advantage of our ground truth class labels and knowledge that there are 7 distinct clusters when initializing the algorithm. We use the *silhouette coefficient* as our metric, which combines the mean intra-cluster distance (a) and mean nearest-cluster distance (b). Each sample receives a score computed by $(b - a) / max(a, b)$ and the average is computed for the class. Good clustering can be measured by values close to 1.0 while values near -1.0 indicate significant overlaps and wrongly assigned clusters [12]. To ensure the initialization does not influence the final clustering, we use three initialization methods to the k-means algorithm. The results are shown in table 2.

	STFT	MIR	MFCC	WaveNet
Kmeans++	0.214	0.140	0.175	0.272
Random	**0.246**	**0.150**	**0.192**	**0.278**
PCA-based	0.195	0.103	0.178	0.249

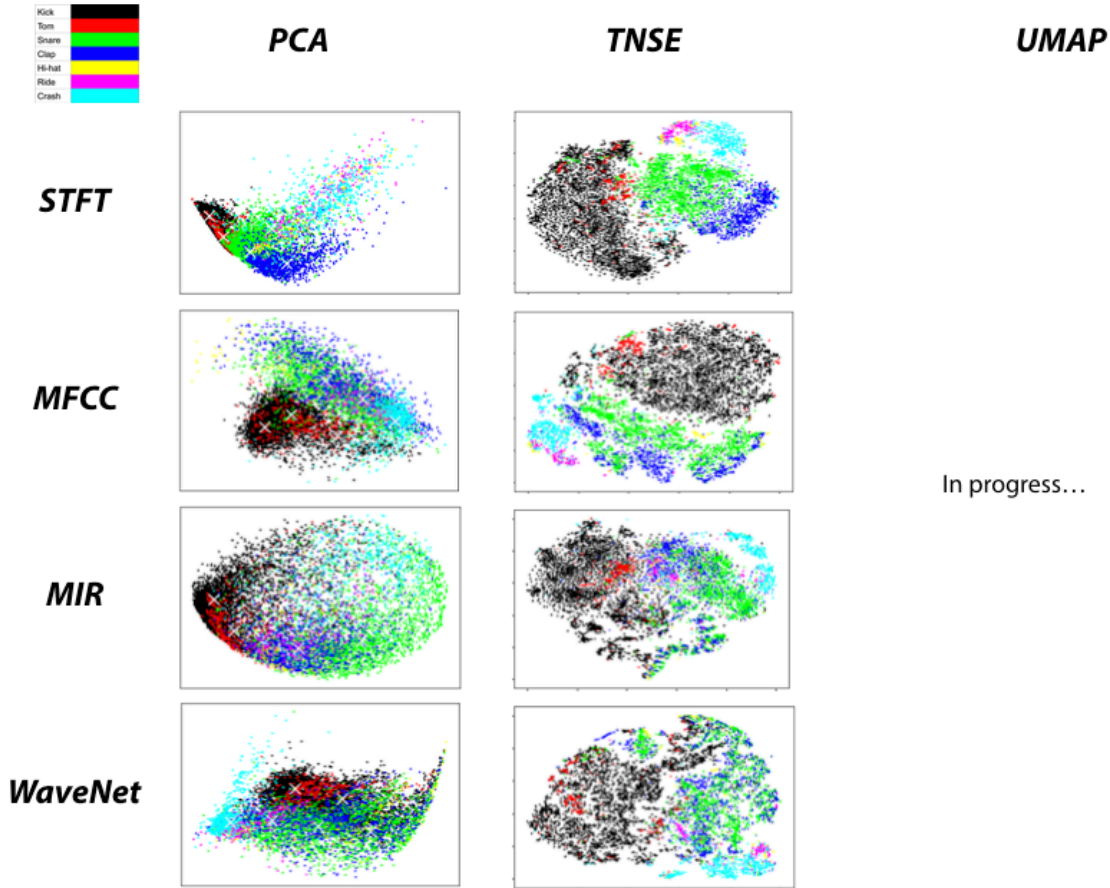**Table 2:** Silhouette coefficients for unsupervised clustering with k=7

Figure 2: 2 dimensional plots using the defined features and chosen dimensionality reduction technique. All coordinates in X and Y are scaled from 0.0 – 1.0 for visual comparison

These initial results suggest that STFT and WaveNet features are best for unsupervised clustering and should be used for dimensionality reduction in the second step, as they yielded the two highest silhouette coefficients.

5. RESULTS

Figure 2 shows the combination of features and dimensionality reduction techniques. We used the following parameters in each dimensionality reduction algorithm:

- *PCA*: plot the first two principal components on the X and Y axis

- *TSNE:* perplexity of 50 and iteration number 500 directly to 2 dimensions. Perplexity can be described as the algorithm's attention to spreading clusters out in space. Iterations refers to the number of iterations during stochastic optimization.

- *UMAP: still in progress*

On first visual inspection, we note how TSNE presents much clusters that are more separate and spread out in space.

6. EVALUATION

6.1 Silhouette Coefficient

	PCA	TSNE	UMAP
STFT	**0.115**	**0.053**	---
MFCC	-0.010	0.034	---
MIR	-0.035	-0.062	---
WaveNet	-0.102	-0.065	---

Table 3: Silhouette coefficients for the 2D embedding

The silhouette coefficients for the embedding shown in figure are presented in table 3. We note that in 2 dimensions, quick visual inspection shows that using STFT with TSNE yields the best visual separation of classes. This is confirmed by STFT features having the highest coefficients in the TSNE domain.

We note however, that the overall silhouette coefficients in 2D are lower than the scores achieved on the original dimensionality of the features. This indicates that we are losing important information in the process and two variables is not sufficient to separate clusters if this was a pure classification task. However, the plots show that the pipeline is still able to preserve important timbral relationships in 2D space.

Most importantly, comparing figure 2 plots and table 3 reveals a significant downfall of using silhouette coefficients as the evaluation metric. Note how the PCA+STFT

combination in table 3 has a *higher* coefficient (0.115) than TSNE+STFT (0.053). This is because the PCA plot contains *tight clusters* of samples which would in turn produce a high silhouette coefficient. However, from a user standpoint, the tight plot of PCA+STFT would be relatively more difficult to navigate compared to TNSE+STFT, which spaces the samples and clusters out further. The visually pleasing layout of TNSE+STFT, actually *impacts negatively* on the silhouette coefficient, since there are now samples that are further away from the mean of the class cluster. For this reason, we motivate a series of new metrics based on the geometry of the clusters and how they are positioned in 2D space. This is presented next section 6.2.

6.2 Geometry Coefficient

Given that we are projecting drum samples in a 2D space, we want to evaluate our projections using metrics that explicitly account for visual spread of clusters. We thus motivate the following "Geometric coefficients":

- *Roundness:* We use the Polsby Popper Test [13] to calculate "roundness" of clusters. For a polygon with area A and perimeter P this metric is defined as $4\pi A / P^2$. A "round" cluster is associated with a more similar in shape to a circle (roundness has values (0, 1], with a circle having a value of 1).

- *Overlap:* This is the area of the union of convex hulls of clusters divided by the sum of the areas of the clusters. Imagine two circles; if the two circles overlap, the result is a new polygon with an area less than the sum of the two original circles. Otherwise, the overlap or "union" would simply be the region covered by two separate polygons. An overlap close to 1.0 is desirable, indicating we have no overlapping areas.

	PCA		TSNE		UMAP	
	Roundness	Overlap	Roundness	Overlap	Roundness	Overlap
STFT	0.674	**0.275**	0.823	**0.240**	---	---
MFCC	0.689	0.235	0.827	0.195	---	---
MIR	**0.904**	0.182	**0.861**	0.177	---	---
WaveNet	0.696	0.209	0.838	0.240	---	---

Table 4: Geometry-based metrics for the 2D embedding

Here we see that the roundness coefficient captures the visually pleasing and spaced layout of TSNE embeddings compared to PCA embeddings. Although the MIR+PCA combination yields the highest roundness coefficient, this is counteracted by the poorest performing silhouette coefficient in table 3 (-0.035) and table 2 (0.150)

By aggregating these metrics, we can find the best feature combinations yielding the best 2D mapping. At the time of writing, this is the TSNE+STFT combination, although our work using UMAP may change these findings. We also plan to run the geometry metrics on an outlier-rejected filtering of our dimensionality reduced data.

6.3 Discussion

We were surprised to find that WaveNet features yielded the best coefficients for unsupervised k-means clustering, but performed worse than STFT features when embedding in 2D space. We believe this may be due to two reasons. Firstly, the weights in WaveNet were trained on melodic and pitched instrument sounds, not percussive sounds [11]. This would explain why STFT and MFCC features perform better than WaveNet features in 2D space. However in higher dimensions, the non-linear nature of neural networks may have captured more essential information in the compressed representation specific to musical sounds. STFT and MFCC are "static" representations of audio based on a series of mathematical transforms. WaveNet is able to define a "non-linear transform" to produce a compressed representation best suited for the task.

We also note that STFT features performed better than MFCC's, which makes sense since MFCC's are often optimized for speech, not drum samples. Lastly, we believe the consistently low performance of the concatenated MIR features could be due to its lower starting dimensionality compared to other features (order ~100 vs ~1000).

7. CONCLUSION

7.1 Optimal pipeline for *Klustr*

Our results show that TSNE and STFT features yields the best 2D representation that accounts for metrics class separability, navigability and visually pleasing results. Ideally, this would be tested through a user study interacting with various 2D maps. However, our ground truth labels and metrics have confirmed a representation that on first inspection yields a space with visually separate and defined clusters of samples. We intend on using these findings in an application where users can "drag and drop" their sample library and preview samples by navigating through the 2D map. We plan to support interfaces such as the iPad and other physical X,Y pads.

7.2 Future work

At the time of writing, a new dimensionality reduction technique called *UMAP* was recently open-sourced. UMAP is known to be faster than TSNE with better preservation of global structure [10]. It can even be used as in-place replacement for standard dimensionality reduction techniques like PCA in the pre-processing step. We will incorporate this technique in the final draft.

Lastly, due to computational limitations, we were only able to fully compute TSNE embeddings up to 500 iterations. Appendix A shows the results of computing TSNE up to 5000 iterations (overnight on a powerful CPU) for one set of features (STFT) on variety of perplexity and iteration cutoffs. We will report results based on a longer set of iterations for TSNE in the final paper for all features.

8. REFERENCES

[1] Faruqe, Md Omar, and Md Al Mehedi Hasan. "Face recognition using PCA and SVM." *Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009. 3rd International Conference on*. IEEE, 2009.

[2] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." *Journal of Machine Learning Research* 9.Nov (2008): 2579-2605.

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[4] Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., ... & Ritter, M. (2017). Audio Set: An ontology and human-labeled dataset for audio events. In *IEEE ICASSP*.

[5] Salamon, Justin, Christopher Jacoby, and Juan Pablo Bello. "A dataset and taxonomy for urban sound research." *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014.

[6] Fried, Ohad, et al. "AudioQuilt: 2D Arrangements of Audio Samples using Metric Learning and Kernelized Sorting." *NIME*. 2014.

[7] "Bird Sounds". Accessed through https://experiments.withgoogle.com/ai/bird-sounds on 10/10/2017

[8] "Infinite Drum Machine". Accessed through https://experiments.withgoogle.com/ai/drum-machine on 10/10/2017

[9] "Comparative Audio Analysis with Wavenet, MFCCs, UMAP, T-SNE and PCA" Medium Post by Leon Fedden. Published on 15/11/2017. Accessed through https://medium.com/@LeonFedden/comparative-audio-analysis-with-wavenet-mfccs-umap-t-sne-and-pca-cb8237bfce2f on 25/11/2017

[10] "Uniform Manifold Approximation and Projection UMAP". GitHub Repository accessed through https://github.com/lmcinnes/umap on 1/12/2017. Paper for this technique is unpublished as of 4/12/2017

[11] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499* (2016).

[12] "sklearn.metrics.silhouette_score" documentation in sklearn library. Accessed through http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html on 10/11/2017

[13] Polsby, Daniel D., and Robert D. Popper. "The third criterion: Compactness as a procedural safeguard against partisan gerrymandering." *Yale Law & Policy Review* 9.2 (1991): 301-353.