Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

КУРСОВАЯ РАБОТА

ПРИМЕНЕНИЕ ТОПОЛОГИЧЕСКОГО АНАЛИЗА ДАННЫХ В ПОИСКЕ МУЗЫКАЛЬНОЙ ИНФОРМАЦИИ

Application of Topological Data Analysis in Music Information Retrieval

по направлению подготовки 01.04.02 Прикладная математика и информатика

образовательная программа «Науки о данных»

Студент группы мНОД20 ИССА

Аматов Амантур

(Ф.И.О.)

Руководитель КР доцент

Чернышев Всеволод Леонидович

(должность, звание, Ф.И.О.)

Москва, 2021

Abstract

Cover song detection task is a part of Music information retrieval tasks. Cover songs may be performed in completely different style. It may be different genre, key, musical instruments, timbre and more. But still the structure of the cover song remains the same as a structure of original song. So, according to the structure, we introduce another approach to learning audio representations via Topological Data Analysis. We preprocess audios with topological methods, such as point clouds, persistence diagrams extraction with further extraction of topological features, which can help to capture the core structure of the song. This preprocessing methods were applied in training of binary classification and Siamese network models. Our results indicate that using topological preprocessing of audios can help in different approaches of cover song identification tasks.

Table of Contents

1. Introduction

One of the tasks of Music Information Retrieval (MIR) is cover song identification. A "cover song" is a different version of the same song, played with different pitch, tempo, instruments, mixing etc. Basically, these cover-reference songs have absolute different sound, but for us, people, it is still recognizable as a "cover song" of some another song. So, what is the problem you may ask, if people can differentiate cover and not cover song from each other, then a computer also can? Actually, not all audio search algorithms can provide good scores on cover song classification, e.g., Shazam algorithm [1], which can't classify a cover song played at festival due to its structure.

What does not change in cover songs in comparison to their original songs is a structure of the song. It still has almost the same verses, interludes, and choruses. So, according to this information we introduce an approach of Topological Data Analysis, which is generally persuaded by the idea that a capable approach to determining solid subjective and in some cases quantitative information about the data structure could be achieved using topology and geometry information about data. So, if the music has structure, we can use topological data analysis to inspect it. TDA was already used in Music Information Retrieval tasks, such as song's genre classification [2] or ethnic music analysis [3].

1.1 Related works

As long MIR exists as a science field, there were also existed a problem of cover songs identifications. There are a lot of articles dedicated to this area of study. Some of them used all given information about tracks, including songs metadata, lyrics and audio [4], only audio features [5], music similarity, obtained by beat-synchronized blocks [6], timbral features [7], cross recurrence plots [8]. Also a lot of works include using of neural networks, such as Convolutional Neural Networks [9] [10], Siamese neural networks [11], different type of losses [12], cross-similarity of songs network embeddings [13].

Referred papers use as main features some spectral, tonal and timbral features, which are regular for this kind of tasks. But while most of them try to capture some structure from songs, it can be also obtained by using topological techniques. There are such papers [14], that try to capture songs' geometric structures, but not to analyze them, preprocess and compare them, which we will propose in our paper.

2. Methods and materials

2.1 Dataset

In this paper was used classical dataset "Covers80" [15], which contains 80 cliques with 2-4 songs (mostly 2) in .mp3 format. Each clique represents a collection of original and cover performance of the song. The main problem was the size of the dataset. There are cover songs datasets with cliques with bigger size [16] [17] [18], but these datasets are either too big for the first experiments

in this science field, either they don't have raw audio, which is very important for the preprocessing step. So covers80 dataset fits well into this situation because we can test some approaches on raw audios without waiting too long on evaluation of our models.

2.2 Data preprocessing

All data preparation was made on Python with two main packages – Librosa for audio data preparation [19] and Gudhi [20] for topological features extraction.

At first, all the audios from dataset were converted to .wav format. It was made due to specific features of Librosa. Then we made cloud of points from each audio, what we will examine within the following section.

2.2.1 Point clouds

We need to convert every song into point cloud in order to obtain a geometric structure of the song. Every song is divided into small chunks of analysis windows and texture windows [14]. Every point is computed within every texture window. Texture windows overlapping; because of the smooth transition between every point of the cloud. Within every analysis window in a texture window audio features are computed and averaged. Length of used analysis window is 100 milliseconds, length of texture window is 3 seconds, shift between every texture window is 1 second.

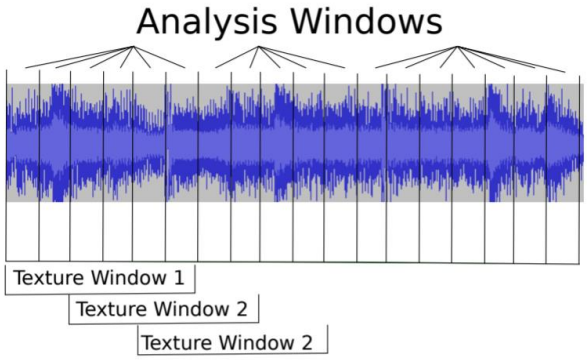

Figure 1 – Example of audio waveform cut in analysis and texture window [14].

As it was proposed, within every analysis window set of audio features were extracted. From this moment we will define 3 used variants of set of features:

1. Tonnetz, Mel-Frequency Cepstral Coefficients (MFCC), chroma features.
2. Only MFCC.
3. Only Tonnetz.

MFCC holds coarse information about the complete smoothed spectrum of each time frequency window. Chroma features summarize pitch in 12 classes of frequencies across all octaves in a time frequency window, one includes for each halfstep in the Western scale. Tonnetz

(or the Tonal Centroids) contain harmonic content of a given audio. Librosa representation of Tonnetz uses the method of projecting chroma features onto a 6-dimensional basis representing the perfect fifth, minor third, and major third each as two-dimensional coordinates.

In particular, we take 12 MFCC coefficients, 12 chroma features and 6 Tonnetz features. These features will be computed within every analysis window, sum up within every texture window, averaged and because the first dataset contains features with different scales, we need also to normalize each dimension of point so that its standard deviation is 1 and mean is 0 over the data cloud. So, for the first dataset point lives in $\mathbb{R}^{30}$, second dataset – $\mathbb{R}^{12}$, and third dataset – $\mathbb{R}^{6}$.

Also, as an experiment, we took only 60 seconds (short) of the song with the offset 10 seconds from the beginning for the sake of the faster computations. So now, we have 6 datasets to preprocess with topological features. The point clouds with full version of the songs we will refer as "full", and 60 seconds – as "short" point clouds.

As an example, we created from these point clouds graphs to show the structure using the Nearest Neighbors algorithm with number of neighbors until graph will be connected (Figure 2).

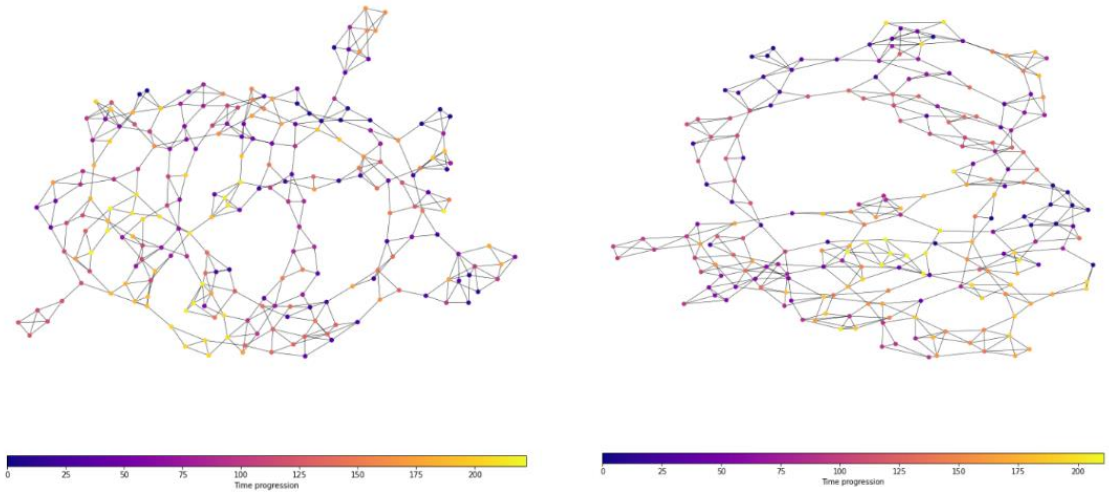

Figure 2 – Graphs of original and cover song of "Take on Me" based on point clouds of all used features. Time progression bar shows at which time moment t is a point.

There were attempts to use the graph structure for the classification task, using some graph features, such as graph radius, diameter, average shortest path, clustering coefficient, but they did not give better results due to uncontrolled number of nearest neighbors within the building the graphs.

2.2.2 Persistence diagrams

After making point clouds we can refer to topological data analysis tool using Gudhi package. From every point cloud we can build a simplicial complex, more precisely Vietoris-Rips complex,

which is one of the most common ways to measure point clouds' topological structure. It is a simplicial complex that generalizes promixity graphs to higher dimensions.

More formally, point cloud could be denoted as $\mathbb{P} = \{p_i, i = 1, \dots, N\}$, where $p_i \in \mathbb{R}^d$, $d \in \{30, 12, 6\}$, according to dimensions of clouds of all features, MFCC and Tonnetz, and N is a size of a cloud. Let us denote a matrix $N \times N$ of Euclidean distances $\mathbb{d} = \{d(p_i, p_j), i, j = 1, \dots, N\}$. For each $p_i \in \mathbb{P}$, let denote a closed ball with radius $\frac{r}{2}$, such that $B_r(p_i) = \left\{x: d(x, p_i) \leq \frac{r}{2},\ x \in \mathbb{R}^d, 0 \leq r \leq U\right\}$, where U is a pre-determined maximum of the distances in $\mathbb{d}$. The Vietoris-Rips complex corresponds to a given radius r and is defined as the set of points $\mathbb{P}_v(r) \subset \mathbb{P}$, such that any points p_i, p_j in $\mathbb{P}_v(r)$ satisfy the condition $d(p_i, p_j) \leq r$ and $i, j = 1, \dots, N$. For a given value r, a simplicial complex $\mathbb{k}(r)$ denotes the set of Vietoris-Rips simplexes such that for any two Vietoris-Rips simplexes $\mathbb{P}_v^1(r), \mathbb{P}_v^2(r) \in \mathbb{k}(r)$, we have

1) $\mathbb{P}_v^1(r) \cap \mathbb{P}_v^2(r) \in \mathbb{k}(r)$.
2) $\left(\mathbb{P}' \subset \mathbb{P}_v^1(r)\right) \Rightarrow \mathbb{P}' \in \mathbb{k}(r)$.

As a homology group, we denote as $\tilde{\alpha}_{\tilde{p},k}$, which consists of k $\tilde{p}$-dimensional simplicial complexes which are homomorphic. As a maximal dimension of this complex, we use $\tilde{p}_{max} = 1$. From Rips complex we build simplex tree, as preferred in Gudhi package. Simplex tree is a data structure, which is used to represent general (or filtered) simplicial complexes.

After we got simplex trees for every point cloud, we can extract from them persistence diagrams. As a radius r of closed ball increases, we can track births and deaths of homological groups and put them in the persistence diagram. Persistence diagram shows a topological feature born at moment x and persists at moment y, more formally consists of set of pairs $\left(b_{\tilde{\alpha}_{\tilde{p},k}}, d_{\tilde{\alpha}_{\tilde{p},k}}\right)$, where a homological group $\tilde{\alpha}_{\tilde{p},k}$ is "born" at time $b_{\tilde{\alpha}_{\tilde{p},k}}$ and persists until time $d_{\tilde{\alpha}_{\tilde{p},k}}$.

So, choosing persistence diagrams as a main representation for classification task is because of its robustness to noisy and messed data, which our dataset is about. With these diagrams we will try to capture the topological structure (or homological groups) of the data.

Here we will provide all techniques, that we used to capture persistence diagrams:

1. Persistence diagrams for full and short point clouds of three combination of audio features (Figure 4a).
2. Persistence diagrams for full sparse point clouds of three combination of audio features, where we take 120 farthest points in a point cloud (Figure 4b).
3. Persistence diagrams from DTM filtration with maximal dimension 0 [21] for full and short point clouds of three combination of audio features (Figure 4c). This type of

filtration constructs a weighted Rips complex giving larger weights to outliers, which reduces their impact on the persistence diagram.

4. Persistence diagram for full and short sparse point clouds of only Tonnetz features, where dimension reduction (2 dimensions) via PCA was used and then persistence diagrams were optimized using Tensorflow via loss as the opposite of the sum of squares of the distances to the diagonal of the points in the diagram (Figure 4d). Optimization continued for 40 epochs with ADAM optimizer (Figure 3).

In general, we took every variant of point cloud, that we acquired on the previous step, computed Rips or DTM filtration with different configurations (without changes, sparsified via founding 120 farthest points, reducing dimensionality and optimizing point clouds), and then computed simplex trees, from which we found persistence diagrams.

So, then we have already 17 variants of persistence diagrams, for which we will compute topological features to classify on the next steps.

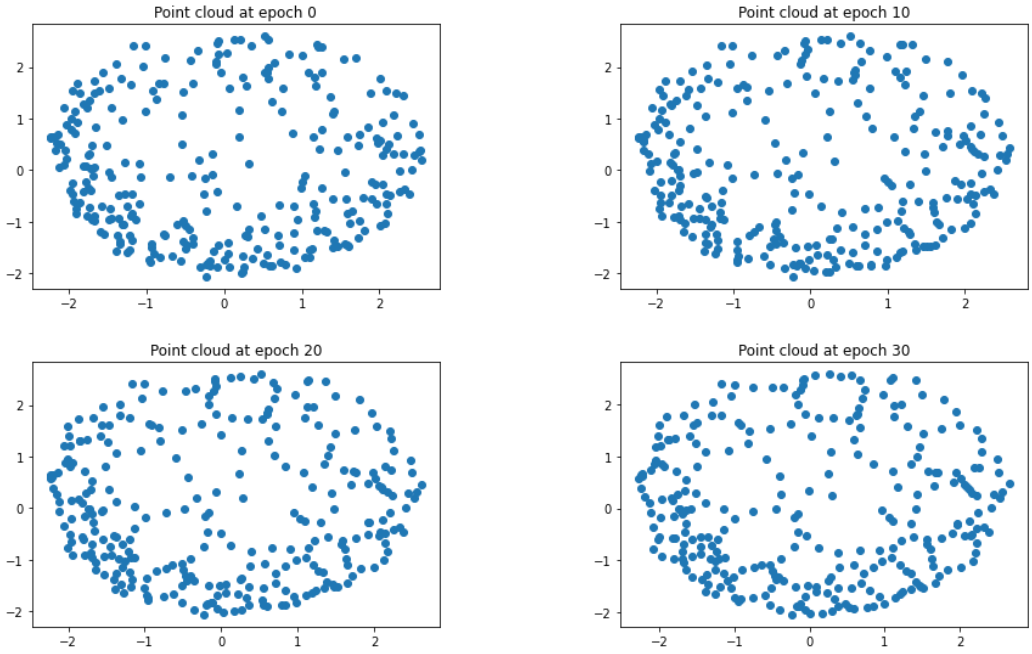

Figure 3 – Full length point clouds of Tonnetz features optimization based on Rips filtration through 30 epochs.

(a)

(b)

(c)

(d)

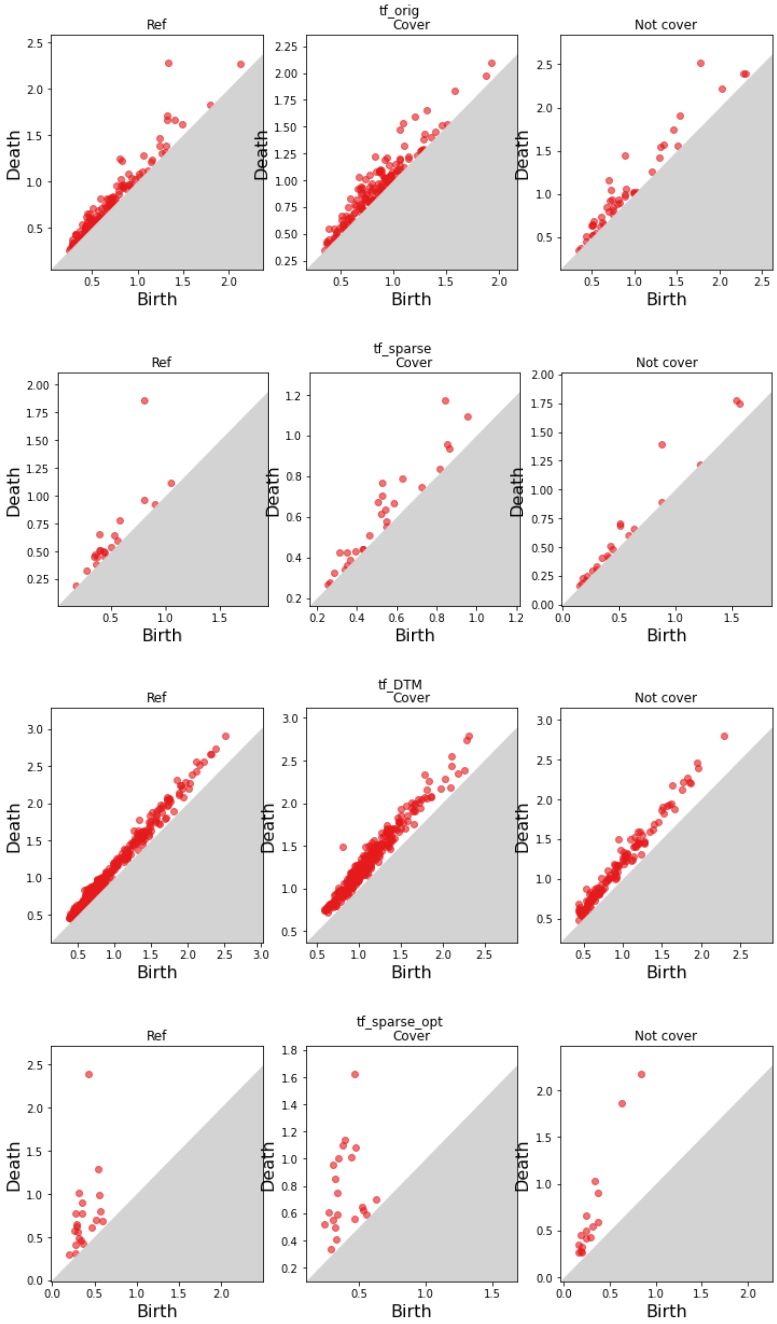

Figure 4 – Example of different variations of persistence diagram of triplet point clouds - reference song, cover song and not cover song, from Tonnetz features.

2.2.3 Topological features extraction

In this section we will show how we represented our persistence diagrams acquired on the previous step.

Before we computed topological representations, we normalized persistence diagrams. At first, were extracted only these points, which are not infinity. Then, we transformed diagrams' points by scaling them to the range [0,1]. And at last, we filtered diagrams with leaving only 25

points, which are the farthest from the diagonal. So now we got filtered diagrams (since now we will refer to these diagrams as “filtered”). After we computed topological representations.

Here are listed used topological representations of persistence homologies (Figure 5):

1) Landscapes [22]. Points from persistence landscapes are mapped into a function space. A persistence landscape is an embedding of a persistence diagram in a space of continuous functions L^2. Landscapes allow us to apply statistical and machine learning techniques on persistence data (Figure 4a).
2) Silhouettes [23]. Silhouettes are computed the same way as Landscapes, but with even averaging the sample on a given range (Figure 4b).
3) Entropy [24]. A persistence entropy is a statistic measuring the homogeneity of statistics of persistence intervals of dimension k at time t (Figure 4c).
4) Betti curves [25]. Simply, the Betti curve shows how many barcodes are at time t of persistence of homological groups (Figure 4d).

All of these features were computed within the resolution size of 500 points.

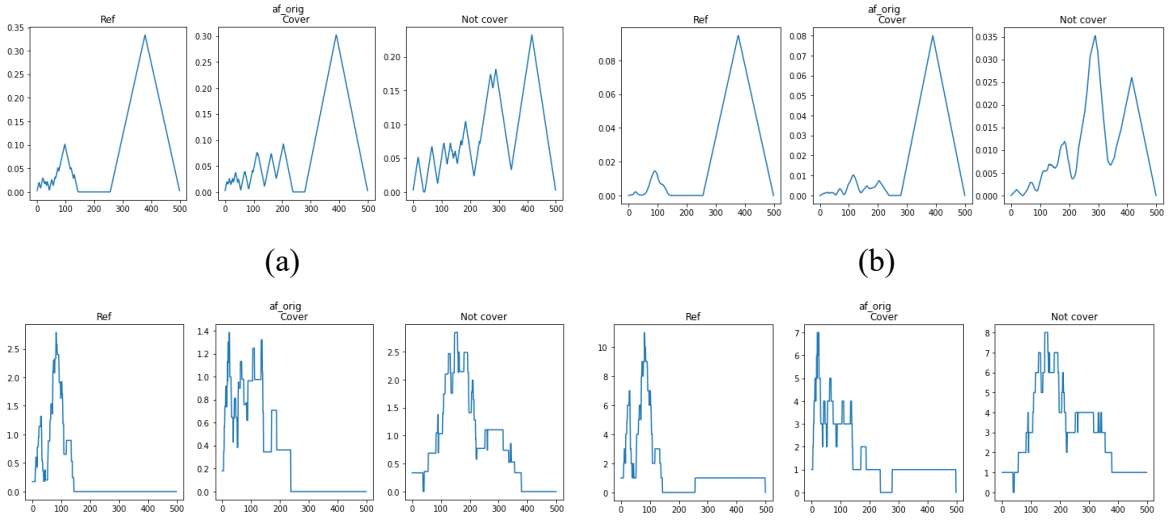

Figure 5 – Example of landscapes (a), silhouettes(b), entropies (c), Betti curves (d) of triplet persistence diagrams – reference song, cover song and not cover song, from full length point clouds of all features. Resolution size = 500.

2.3 Machine learning models

After we computed all main features for each song, we can make pair songs dataset to reduce the task of cover song detection to binary classification, where reference-cover pair is labeled as 1, and non-reference-cover is labeled as 0. As we did this artificial dataset, we acquire 84 pairs of reference-cover pairs and 3444 pairs of non-reference-cover pairs. We normalized datasets and

reduced 0-labeled examples in order to get the same-sized 1-labeled and 0-labeled samples, because other way model would overfit. So, we split our datasets in test and train, where test size is equal 0.125, and end up with train data with 158 examples and test data with 20 examples.

2.3.1 Mutual features

After we split our 17 datasets into train and test, we computed mutual features or distances. Here is the list of the mutual features, that we computed between songs features:

1) Euclidean norm between landscapes/silhouettes/entropies/Betti curves of each song in a pair.
2) Bottleneck distance between the filtered persistence diagrams D and D'. Its formula is as follows in formula 1:

$$W_\infty(D, D') = \inf_{\varphi: D \to^{\simeq} D'} \sup_j \varepsilon_j, \quad (1)$$

where $\varphi: D \to^{\simeq} D'$ is bijection between two persistence diagrams, $\varepsilon_j = ||x_j - x'_j||_\infty$, $x_j = (b_j, d_j)$ is a point of diagram, and $j \in J, |J| = |D| + |D'|$

3) 1-Wasserstein distance between the filtered persistence diagrams. It is a similarity score between two diagrams, which computes as the sum of all edges' lengths. Its formula is as follows in formula 2:

$$W_1(D, D') = \inf_{\varphi: D \to^{\simeq} D'} \sum_j \varepsilon_j \quad (2)$$

4) Kernels:
    a. Persistence Weighted Gaussian Kernel. The persistence weighted Gaussian kernel is computed by convolving the persistence diagram points with weighted Gaussian kernels.
    b. Persistence Scale Space Kernel. The persistence scale space kernel is computed by adding the symmetric to the diagonal of each point in each persistence diagram, with negative weight, and then convolving the points with a Gaussian kernel.
    c. Persistence Fisher Kernel. The persistence Fisher distance is obtained by computing the original Fisher distance between the probability distributions associated to the persistence diagrams given by convolving them with a Gaussian kernel.
    d. Sliced Wasserstein Kernel. The sliced Wasserstein kernel is computed by exponentiating the corresponding sliced Wasserstein distance with a Gaussian kernel.

2.3.2 Binary classification

Based on computed mutual features we use several machine learning approaches to do binary classification. Before doing the classification task we analyze features. We tried to find highly correlated features (Figure 6). As we can see from this figure, highly correlated features are: landscape norm, silhouette norm, bottleneck distance, Wasserstein distance and Entropy norm and Betti curves norm.

Also, we did variance threshold feature selection, univariate feature selection, sequential feature selection and recursive feature elimination.

As a classification machine learning model, we used SVM classifier with data normalization, Gradient Boosting classifier with hyperparameters: 100 estimators, learning rate equals 1.0, maximal depth equals 1 and random state equals 0, Random Forest classifier with hyperparameters: 100 estimators, maximal depth equals 5 and a random state equals 0. For these models we used all feature selection algorithms and discovered, that sequential feature selection and Variance Threshold showed the best results, what we will examine within the following sections.

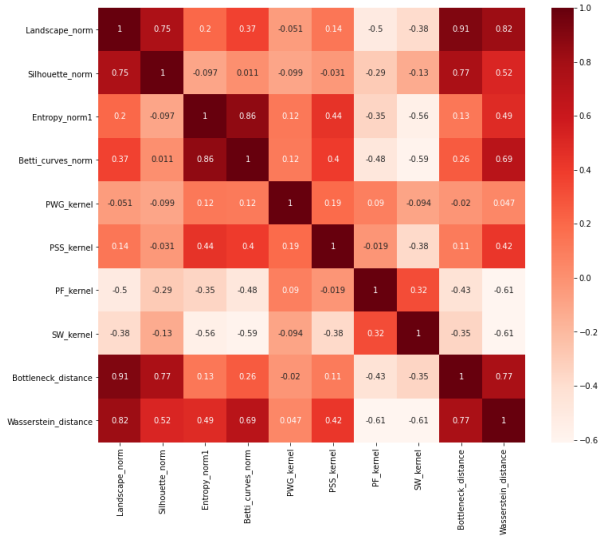

Figure 6 – Features from dataset made by Tonnetz features from short length songs.

2.3.3 Siamese neural network

In this section we will shortly discuss another approach of cover song detection using Siamese Neural Networks. The main advantage of this approach is a possibility of using a small amount of data using one shot learning. It means, that we train a neural network to extract a feature vector from input data to calculate cover-song similarity. With our datasets, which consists only of 164 songs, we need only two songs of the same clique to understand, whether a song belongs to this clique.

As an input representation, we used silhouettes of persistence diagrams of point clouds based on MFCC and full-length audio, which were divided into train, validation, and test sets, 0.6, 0.2 and 0.1, respectively.

As a network architecture, our Siamese convolutional architecture is shown in Figure 7. We trained model for 100 epochs and mini-batch size of 32 using the ADAM optimizer with learning rate 0.0001 to update weights.

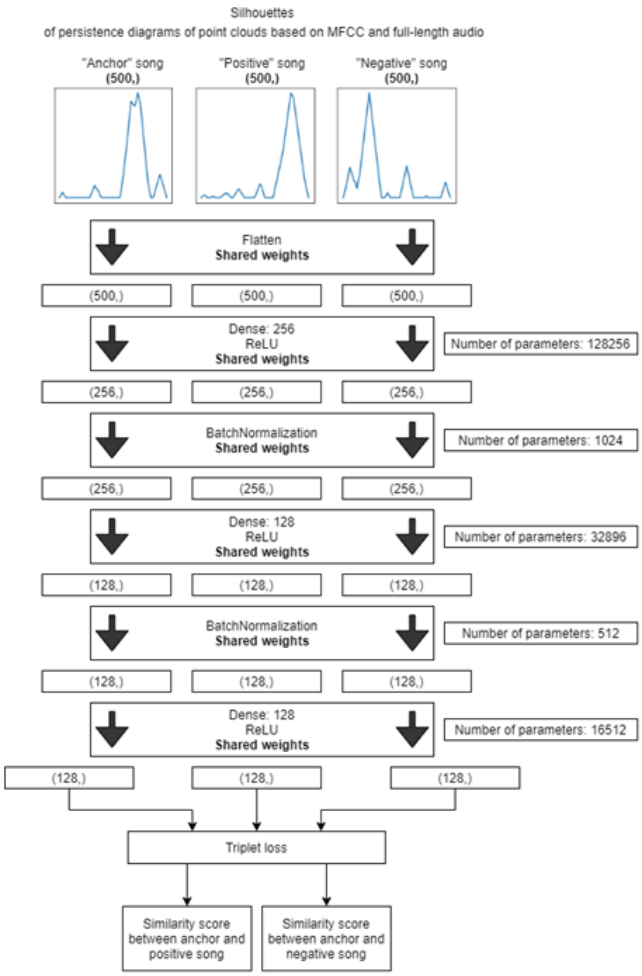

Figure 7 – Proposed Network Architecture

As an objective function, we used a triplet loss. Our network is trying to learn a feature extraction function f between three query songs x_a, x_+ and x_-, anchor sample, positive sample, and negative sample, respectfully, to minimize distance between anchor and positive sample and maximize distance between anchor and negative sample.

Our neural network computes the triplet loss using the embeddings of anchor, positive and negative samples produced by the Siamese Network.

It is defined in formula (3):

$$L(x_a, x_+, x_-) = \max(||f(x_a) - f(x_+)||^2 - ||f(x_a) - f(x_-)||^2 + m\,, 0), \quad (3)$$

where $f(\cdot)$ stands for embedding of input sample and m stands for margin.

3. Results

All results and code with the generalized structure of the whole algorithm can be obtained through the link.

3.1 Binary Classification

To evaluate models of binary classification we used three metrics – Accuracy, ROC AUC score and F1 score. Train size was 0.875 and test size 0.125.

Using a configuration of point clouds of Mel Function Cepstral Coefficients of short sized audio with SVM model and sequential selection of features: Norm of silhouettes, Norm of Betti Curves, and Persistence Weighted Gaussian Kernel, yielded best result of Accuracy 0.8, F1 score 0.82, and ROC AUC score 0.8 (Figure 8). Top 5 of the best evaluation results and configurations you can see on Table 1.

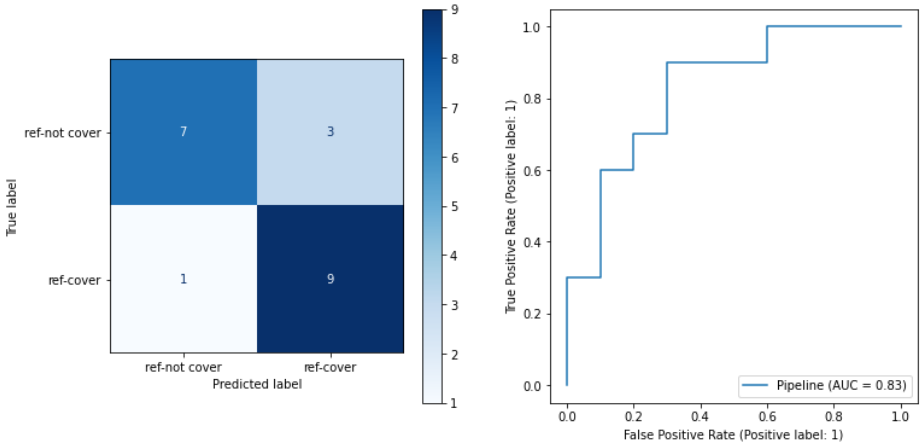

Figure 8 – Confusion matrix and ROC Curve of test set of the best results

Table 1 – Sequential Feature Selection

Configurations						Evaluation		
Audio Features	Length	Filtration	Sparse	Model	Features	Acc	F1	ROC AUC
MFCC	Short	Rips	False	SVM	Silhouette norm Betti curves norm PWG kernel	0.8	0.82	0.8
MFCC	Short	Rips	False	Grad Boost	Landscape norm PSS kernel	0.75	0.74	0.75
Tonnetz	Full	Rips	True	Random Forest	Silhouette norm	0.75	0.71	0.75
Tonnetz	Full	Rips	True	Grad Boost	Silhouette norm	0.75	0.71	0.75
All	Full	Rips	False	Grad Boost	Landscape norm PSS kernel PF kernel	0.7	0.73	0.7

3.2 Siamese Network

As a similarity score between anchor, positive and negative samples we used cosine similarity. After evaluation on test set of batches with size 32, we got mean similarity score between anchor songs and positive songs 0.726 and between anchor songs and negative songs 0.659. As we can see, similarity between anchor and positive songs is greater than similarity between anchor and negative songs, but still not as great as it should be.

4. Conclusion and Future Work

The target goal of this work was to show how topological data analysis (TDA) can be applied to Music Information Retrieval tasks. So, we show a TDA preprocessing approach to the cover song detection task with binary classification with three models and Siamese network with triplet loss and similarity score. We find the best suited parameters of point clouds, which are Mel-Function Cepstral Coefficients, with Rips filtration and SVM classification model. Also, we find the best topological features to do the classification task, which are Silhouette and Landscape norm (which are already highly correlated), PWG and PSS kernels. For Siamese network we acquire results of good embeddings similarity computations from topological feature, which is also a silhouette of point clouds of Mel-Function Cepstral Coefficients of short 60 second audios.

But still, we could not compare our results with results from other works due to using of different metrics to evaluate models. In most papers Mean Average Precision (MAP), mean rank of the first correctly identified cover (MR1) and mean number of covers identified in top 10 (MNIT10) are used, which are proposed in the MIREX for audio cover song identification task.

As a future work, we are interested in making an evaluation system, with which we can compare our results with another papers. Also, we are interested in using different datasets with more songs and bigger cliques, such as DA-TACOS [16], Kara1k [18] or The Million songs dataset [17]. Also, changing the parameters of creating point clouds to understand, which size of analysis, texture windows and window shift, fits better. According to Siamese network, we are interested to evaluate models with different layers and filtration and another type of loss (such as a contrastive loss).

5. References

[1] A. Wang, "An Industrial Strength Audio Search Algorithm," in *Proceedings of the ISMIR 2003: Proceedings of the fourth International Conference on Music Information Retrieval,* pp. 7-13, 2003.

[2] Y. Panagakis and C. Kotropoulos, "Music genre classification via Topology Preserving Non-Negative Tensor Factorization and sparse representations," in *Proceedings of 10th International Society for Music Information Retrieval Conference, ISMIR,* pp. 249-252, 2010.

[3] M. L. T. Jung and P. j.-H. Changbom, "Topological Data Analysis of Korean Music in Jeongganbo: A Cycle Structure," *arXiv preprint arXiv:2103.06620,* 2021.

[4] A. A. C. Arcos, R. Hennequin and M. Arcos, "Large-Scale Cover Song Detection in Digital Music Libraries Using Metadata, Lyrics and Audio Features," *arXiv preprint arXiv:1808.10351,* 2018.

[5] S. Ravuri and D. Ellis, "Cover song detection: From high scores to general classification," in *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* pp. 65-68, 04 2010.

[6] C. J. Tralie, "Early MFCC And HPCP Fusion for Robust Cover Song Identification," in *18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, Suzhou, China, 2017.

[7] C. J. Tralie and P. Bendich, "Cover Song Identification with Timbral Shape Sequences," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR),* pp. 38-44, 2015.

[8] J. Serrà, X. Serra and R. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics,* vol. 11, no. 9, p. 093017, 2009.

[9] Z. Yu, X. Xu, X. Chen and D. Yang, "Learning a Representation for Cover Song Identification Using Convolutional Neural Network," in *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* pp. 541-545, 2020.

[10] F. Yesiler, J. Serrà and E. Gómez, "Accurate and Scalable Version Identification Using Musically-Motivated Embeddings," in *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* pp. 21-25, 2020.

[11] M. Stamenovic, "Towards Cover Song Detection with Siamese Convolutional Neural Networks," in *35th International Conference on Machine Learning, PMLR 80*, Stockholm, Sweden, 2018.

[12] X. Du, Z. Yu, B. Zhu, X. Chen and Z. Ma, "Bytecover: Cover Song Identification Via Multi-Loss Training," in *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[13] C. Jiang, D. Yang and X. Chen, "Similarity Learning For Cover Song Identification Using Cross-Similarity Matrices of Multi-Level Deep Sequences," in *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* pp. 26-30, 2020.

[14] P. Bendich, E. Gasparovic, J. Harer and C. Tralie, "Geometric Models for Musical Audio Data," in *Leibniz International Proceedings in Informatics,* pp. 651-655, 2016.

[15] D. P. W. Ellis, "The "covers80" cover song data set," 2007. [Online]. Available: http://labrosa.ee.columbia.edu/projects/coversongs/covers80/. [Accessed 15 03 2021].

[16] F. Yesiler, C. Tralie, A. Correya, D. Silva, P. Tovstogan, E. Gomez and X. Serra, "Da-TACOS: A dataset for cover song identification and understanding," in *Proceedings of International Society for Music Information Retrieval (ISMIR),* pp. 327-334, 2019.

[17] T. Bertin-Mahieux, D. Ellis, B. Whitman and P. Lamere, "The Million Song Dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011),* pp. 591-596, 2011.

[18] Y. Bayle, L. Marsik, M. Rusek, M. Robine, P. Hanna, K. Slaninova, J. Martinovic and J. Pokorny, "Kara1k: A Karaoke Dataset for Cover Song Identification and Singing Voice Analysis," in *Proceedings on 2017 IEEE International Symposium on Multimedia (ISM),* pp. 177-184, 2017.

[19] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference,* vol. 8, pp. 18-24, 2015.

[20] T. G. Project, "GUDHI User and Reference Manual," 2021. [Online]. Available: https://gudhi.inria.fr/doc/3.4.1/. [Accessed 10 05 2021].

[21] H. Anai, F. Chazal, M. Glisse, Y. Ike, H. Inakoshi, R. Tinarrage and Y. Umeda, "DTM-based Filtrations," in *Topological Data Analysis*, Springer, 2020, pp. 33-66.

[22] P. Bubenik, "Statistical Topological Data Analysis using Persistence Landscapes," *Journal of Machine Learning Research,* pp. 77-102, 2015.

[23] F. Chazal, B. T. Fasy, F. Lecci, A. Rinaldo and L. Wasserman, "Stochastic Convergence of Persistence Landscapes and Silhouettes," *JoCG,* vol. 6, pp. 140-161, 2014.

[24] N. Atienza, R. Gonzalez-Díaz and M. Soriano-Trigueros, "On the stability of persistent entropy and new summary functions for Topological Data Analysis," in *Pattern Recognition,* vol. 107, p. 107509, November 2020.

[25] Y. Umeda, "Time Series Classification via Topological Data Analysis," in *Transactions of the Japanese Society for Artificial Intelligence,* vol. 32, no. 3, pp. 1-12, 2017.