Music Beyond Major and Minor: Visualization and Transposition Invariant Clustering of Musical Pieces

Ahmed Jellouli ahmed.jellouli@epfl.ch
Ayman Mezghani ayman.mezghani@epfl.ch
Karim Hadidane karim.hadidane@epfl.ch

CS-433 Machine Learning, EPF Lausanne, Switzerland

Supervised by: Robert Lieck, robert.lieck@epfl.ch
Digital and Cognitive Musicology Lab DCML, EPF Lausanne, Switzerland

***Abstract*—Western classical music, developed in Europe after the middle ages, is based on the Major and Minor scales. This distinction between Major and Minor is central in the definition of scale, chords and keys; This paper presents a data-driven approach to investigate the concept of Major and Minor key. Its goal is to evaluate the extent to which the structures used in music theory can be reproduced when working on pitch class distributions. We model pitch class distributions sampled from *Johann Sebastian Bach* pieces using a transposition invariant Dirichlet Mixture Model, and visualize the results for a different number of clusters using the Isomap algorithm. We find that the clustering finds patterns in the pitch class distributions beyond the binary Major and Minor dichotomy.**

I. Introduction

The project consists in building a data-driven model to cluster musical data points to find hidden patterns. We use data that comes from `MusicXML` files of different pieces by classical music composers. The pre-processing of the dataset is based on the *pitch scape* representation [1] of the musical content of a piece.
We implement, for the clustering phase, a transposition invariant Dirichlet Mixture Model. We then run it for a different number of clusters and interpret the results.
In this paper, we start by introducing the mathematical background needed such as the concept of *pitch scapes* and the Dirichlet Mixture Model. Second, the pre-processing steps will be shown in details. We will then presents the results of the data visualization and the clustering process.
Throughout the paper, we will be using data from compositions of *Johann Sebastian Bach*. The findings will then be compared to results found by running the model on pieces by a selection of other composers.

II. Pitch Scapes

A novel mathematical representation of the musical content of a piece of music was proposed by Lieck, R. and Rohrmeier, M. [1]. This representation is based on pitch scapes, which are defined as the probability distribution of pitch classes in a certain time interval of the piece.

Definition 1 A pitch scape χ of a musical piece is a function that maps each interval into a pitch class distribution:

$$\chi : \mathbb{R} \times \mathbb{R} \rightarrow [0,1]^{12} \quad , \quad \sum_{\pi=0}^{11} \chi\left(\pi | \mathrm{t_s}, \mathrm{t_e}\right) = 1 \qquad (1)$$

where π is a pitch class, and t_s, t_e define a time interval of the piece.
Pitch scape estimates for a certain pitch class are obtained in Bayesian fashion, by integrating the pitch class density over the desired interval, adding prior counts, and properly normalizing [1].
For time t and pitch class π, pitch class density corresponds to the counts of pitch class π over all tones that sound at time t, normalized by the number of tones at that instant [1].
In our work, we use prior counts of 1, which corresponds to uniform prior distribution over the the pitch classes.

III. Dirichlet Mixture Model

A. Motivation

In this work, each individual data point is a pitch class distribution, sampled from a musical piece at a certain time interval.
We model this data using a transposition invariant Dirichlet mixture model. The Dirichlet distribution is the conjugate prior of the categorical distribution, which makes it natural and simple to use in order to model pitch scapes.

B. Formal Model

The mixture model can be formalized as follows:
Variables:

- $v \in D = [0,1]^{12}$, vector representing a pitch class distribution.
- $c \in C = \{1, 2, ...|C|\}$, latent variable, cluster index for v.
- $\tau_c \in \{1, 2, ..., 12\}$, latent variable, transposition of cluster c.

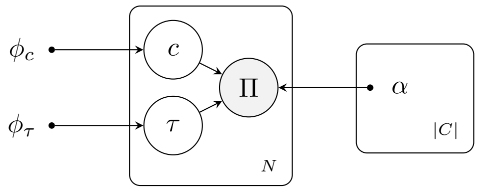


Figure 1: Plate diagram of the Mixture model, α is the cluster specific parameter, ϕ_c and ϕ_τ are categorical distributions, $\Pi = \chi(t_s, t_e)$ is the pitch scape as the interval $[t_s, t_e]$, and N is the number of data points.

Parameters:

- $\theta \in \mathbb{R}^{|C| \times 12}$ cluster specific parameters.

The model assumes that conditional on the cluster c and the transposition τ_c, the pitch class distribution v is $Dir(\alpha_c)$:

$$p(v|c, \tau_c) \sim Dir(\alpha_c) \tag{2}$$

Then:

$$p(v) = \sum_{c} \sum_{\tau_c} p(v|c, \tau_c) p(\tau_c|c) p(c) \tag{3}$$

where $p(v|c, \tau_c) = p(v_{\tau_c}|c)$ and v_{τ_c} is the vector v rotated τ_c times.

We make two additional assumptions on the model:

- τ_c is independent of the cluster c ; $p(\tau_c|c) = p(\tau_c)$
- τ_c and c are uniformly distributed over their possible values.

C. Training

The parameters θ of the above described model can be optimized using gradient descent on the negative log-likelihood of the data, assuming independence of individual data points. We can then compute the cluster c_v for each data point v:

$$c_v = \underset{c}{\operatorname{argmax}}\ p(c|v) = \underset{c}{\operatorname{argmax}}\ \frac{1}{|C| p(v)} p(v|c) \tag{4}$$

D. Transposition invariance

Modelling our data using a latent variable representing the transposition of each cluster makes the model **transposition invariant**. This means that the clusters found are not sensitive to cluster wide transpositions. The motivation behind this design is that we are looking for patterns similar to the Major and Minor dichotomy, which are inherently transposition invariant.

IV. Data Pre-Processing

Now that we are done with introducing the main theoretical material, we will dive into the data pre-processing process. As mentioned above, the data used is musical scores in `musicXML` format. The pitch scape representation is computed using the `pitchscapes` library [2].

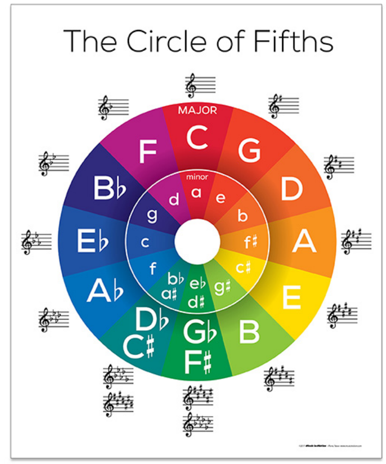


Figure 2: Circle of fifths

(Source: musicmotion.com)

We evaluate the pitch scape at different time intervals t_s, t_e, where $t_s < t_e$ and t_s, t_e are sampled from a linear space over the duration of the piece.

Data points represent distributions over the *twelve pitch classes* i.e *C*, *C#*, *D*, *D#*, *E*, *F*, *F#*, *G*, *G#*, *A*, *A#* and *B*.

We use a `Pandas DataFrame` to manipulate the data points. We augment the data frame using the metadata that contains information about the pieces such as the name, the length and the year of composition.

Other relevant columns were added. We compute the time window width, $t_e - t_s$, and relative width, $(width - min\_width)/(max\_width - min\_width)$, for each data point. We also use the Key estimator algorithm to estimate the transpositions and the Major-Minor key classifications.

V. Data Visualization

A. Music Theory Definitions

Before diving into the exploration of our data, we have to define two music theory concepts: the circle of fifths and the Tonnetz.

1) The Circle of Fifths: The circle of fifths, shown in Figure 2, is a geometrical representation of the pitch classes, different from the chromatic scale that we already know. Going in the clockwise direction, the next pitch class corresponds to the perfect fifth of the last i.e seven semitones higher. It is highly used in the analysis of classical music as its design is helpful in modulating to different keys within a composition.

2) The Tonnetz: The Tonnetz is a lattice diagram which can be used to understand harmonic relationships between chords. In addition to the *perfect fifth* relation, the Tonnetz, shown in Figure 3, links a chord to its *major third* (four semitones above) by taking a diagonally up step and its

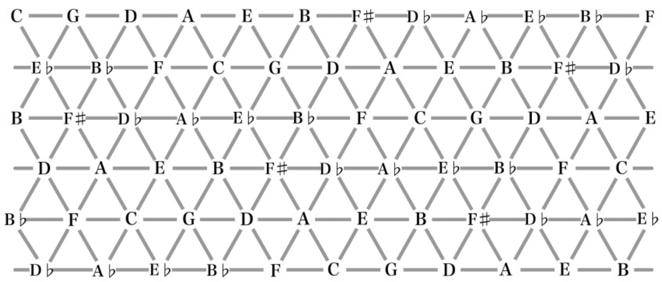


Figure 3: Tonnetz

(Source: T. Bergstrom, K. Karahalios, and J. Hart [3])

minor third (three semitones above) by taking a diagonally down step [3].

B. Dimensionality Reduction

Since a data point lives in a twelve-dimensional space, a dimensionality reduction is needed in order to visualize the data. A possible method to do so is the Isometric mapping, or so-called Isomap. The Isomap algorithm uses simple methods to estimate the intrinsic topology of the data based on the neighborhood of each data point [4]. It is highly dependent on the metric used. We experimented with multiple metrics, such as the *Euclidean distance* and the *Manhattan distance*. The most robust embedding was found when using the *Jensen-Shannon distance* [5], . The *Jensen-Shannon distance* is a metric based on the Kullback-Leibler divergence and is well suited for measuring distance between probability distributions.

We apply the Isomap algorithm on the data to embed it in a three-dimensional space. For ease of visualization, we also compute the embedding of "landmarks" in this space. These correspond to reference key profiles for each of the twenty-four possible keys, and an artificial profile, representing the uniform pitch class distribution.

C. Data Exploration

Visualizing the embedded three-dimensional data gives us great insights about its structure.

We notice that the uniform distribution landmark is placed at the center, while the key landmarks are placed in an almost circular fashion. We also see that the order of the key landmarks is almost similar to the ordering of the keys on the circle of fifths, defined in Section V-A1. The visualization reveals a **torodial-like** shape around the uniform distribution (Figure 4 in Appendix), which is very interesting in our analysis (and later in our model). This will be more clear in the following subsections.

1) Key Coloring: We first start by using the `key-estimator` provided in the `pitchscapes` library [2] to detect the transposition and tonality (*Major* or *Minor*) of each sample point. We color the data points according to their detected key (transposition + tonality) while respecting the circle of fifth coloring scheme to obtain the Figure 5a in Appendix.

Our torus, colored now, shows a hue behavior consistent with the circle of fifths. This means that data points which are estimated to belong to pitch classes that are harmonically related such as *C* and *G* are represented with close hues. Notes that are dissonant i.e distant from each other in the circle of fifths, are distant in hues.

2) The Wavy Structure: The embedding, as seen in Figure 4b, also shows a wavy structure around the uniform distribution. As we can see, there is a threefold wave around the center. This can be interpreted in different ways.

Musically speaking, this structure can be related to the concept of Tonnetz, defined in Section V-A2.

The wave structure can be seen as a result of trying to get a chord close to its major third and minor third. For example, trying to pull the *C major* and *E major* chords and *C major* and *Eb major* close would transform the two-dimension circle of fifths to an undulating circular shape in the three-dimensional space.

The intuition given about the wavy structure of the data points is not unique and we will try to interpret it in a non-musical analysis based on prior counts.

We have already introduced the prior counts in Section II. They are constants added to the pitch class counts before normalizing to pitch class distributions in order to avoid zero counts that are due to silent parts in a piece. In fact, this parameter may give the radial aspect of the structure. More precisely, with a very high prior count, the resulting pitch-class distributions are close to the uniform distribution at the center of the embedding.

3) Relative Width: Coloring the data points with respect to their relative width reveals the fact that points with small relative window widths are closer to the center. Hence their distribution tends to the uniform distribution located in the center (Figure 5b in Appendix).

VI. Clustering and Results

A. Dirichlet Mixture Model Implementation

The Dirichlet Mixture Model was implemented using `Python` as a programming language and the `PyTorch` package for training. Once the data pre-processing phase is done, we have our data in an $N \times 12$ array, where N is the number of data points. We use `PyTorch` tensors to augment the data to consider the number of possible clusters and transpositions, which yields a tensor of shape $N \times |T| \times 1 \times 12$. The generated tensor will be used in the computation of the likelihood and the optimization in the training.

B. Clustering

The clustering on the dataset was done using a Dirichlet Mixture Model, explained in Section III. We will explore in the following sections different numbers of clusters and try to interpret them. Here we will report 2D figures of the data. However, 3D interactive versions of all the plots are available on our `github.io` page [6].

1) Two Clusters: The initial and most straightforward choice was to classify the data into two clusters. The results of the clustering using our model were consistent with the key profiles estimates (provided in the `pitchscapes` library) run on the same data, as shown on Figure 6 in Appendix.

More precisely, the key estimate algorithm shows that the data points estimated to be major keys were located in the outer rim of the torus. The clusters returned by our model gives similar cluster layout. This means that the DMM model built is successfully finding the major-minor patterns.

Another possible interpretation of the clustering can be found by noticing the position of the uniform distribution at the center of the torus. This indicates that the two clusters found may be capturing the entropy of the pitch class distributions, or how similar they are to the uniform distribution. In this context, the inner ring cluster would represent points with higher entropy than those on the outer ring cluster.

2) Three Clusters: For three clusters, the algorithm seems to find more interesting patterns in the data. First, one of the clusters captures the inner ring of the torus. These are points close to the uniform distribution, and at a certain distance from the key landmarks. The second cluster forms another ring around the torus, closer to the minor key landmarks. The third cluster, however, is well localized and seems to be formed of "patches" that are close to some of the landmarks. This seems to capture a single diatonic key that combines major and minor. (Figure 7 in Appendix).

3) Four Clusters: In the case of four clusters, we notice that the clusters start to become well localized (Figure 8 in Appendix). Two of the clusters are similar in location to the three-cluster case, The inner ring is captured as a separate cluster. These are the point that are close to the uniform landmark but far from the key landmarks. The second cluster is also a continuous ring around the torus, but is close to all the minor landmarks around the circle of fifths. The new interesting clusters are the other two: they seem to consist of multiple "patches" around the torus. One of these clusters captures points between **relative key pairs**. A relative key pair is a pair of major and minor keys that share the same notes, but have different tonics. These keys are located at some point on in the circle of fifths. The last cluster's interpretation is not entirely clear, but the different patches are often well localized around minor key landmarks.

4) Twelve Clusters: When taking a higher number of clusters, it becomes increasingly unclear what the interpretation of each cluster is. All of the clusters are visually on a ring of the torus (as expected by transposition invariance), some look more discrete, i.e well localized patches around or between landmarks, and some have a much more continuous aspect.

This is an indication that the binary classification of Major and Minor is not enough to characterize all the pitch class distributions, and that there exists more ways to find similarities between pieces. The major-minor dichotomy therefore does not take into account the continuous space of pitch class distributions.

C. Clustering of data points from other composers

Until now, we have been using data points from compositions by *Johann Sebastian Bach*. It would be interesting to look at the results on data from different composers. In this section, we will discuss some of the results we found when working on *seven* specific composers: *Alexander Scriabin, Johannes Brahms, Franz Liszt, Frederic Chopin, Franz Schubert, Claude Debussy, Maurice Ravel*. These composers were chosen specifically because their work is known to be less "traditional", in terms of the keys and tonalities chosen. From the visualizations of the data, this is confirmed as the topology is different from that of *Bach*'s compositions. The torodial structure around the uniform distribution does not appear as clearly as for Bach pieces. We also observe more data points close to the uniform distribution at the center. Figure 9 in Appendix, shows an embedding of the data. It is also noteworthy to mention that the radial dimension (starting from the uniform distribution) is no longer correlated with the relative width of the data point. When performing clustering of these pieces, we find similar clusters as the ones found for *Bach* pieces (Major vs Minor, low entropy and multiple flavours of Major and Minor). However, it is harder assign meaning to these clusters. This may be due to the structure of the data, and the performance of the Isomap algorithm, especially the placement of the key landmarks (Figure 10 in Appendix).

VII. Summary

In this work, we used a transposition invariant Dirichlet Mixture model is order to cluster pitch class distributions both from *Bach* pieces, and other selected composers' pieces. We found that the clustering with two clusters yields results that can be interpreted in two ways: the clustering is either detecting the tonality of the pieces, or splitting the data according to the entropy of the pitch class distributions. We also observed that by augmenting their number, the some clusters become more localized around the key landmarks, and harder to interpret musically. Finally, we experimented with less "traditional" composers and discovered that, although there are similarities with the previous case, the topology and clustering results are less straightforward than the *Bach* pieces.

Acknowledgements

We would like to thank Dr Robert Lieck for his continuous guidance throughout the project.

REFERENCES

[1] R. Lieck and M. Rohrmeier, "Modelling hierarchical key structure with pitch scapes," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.

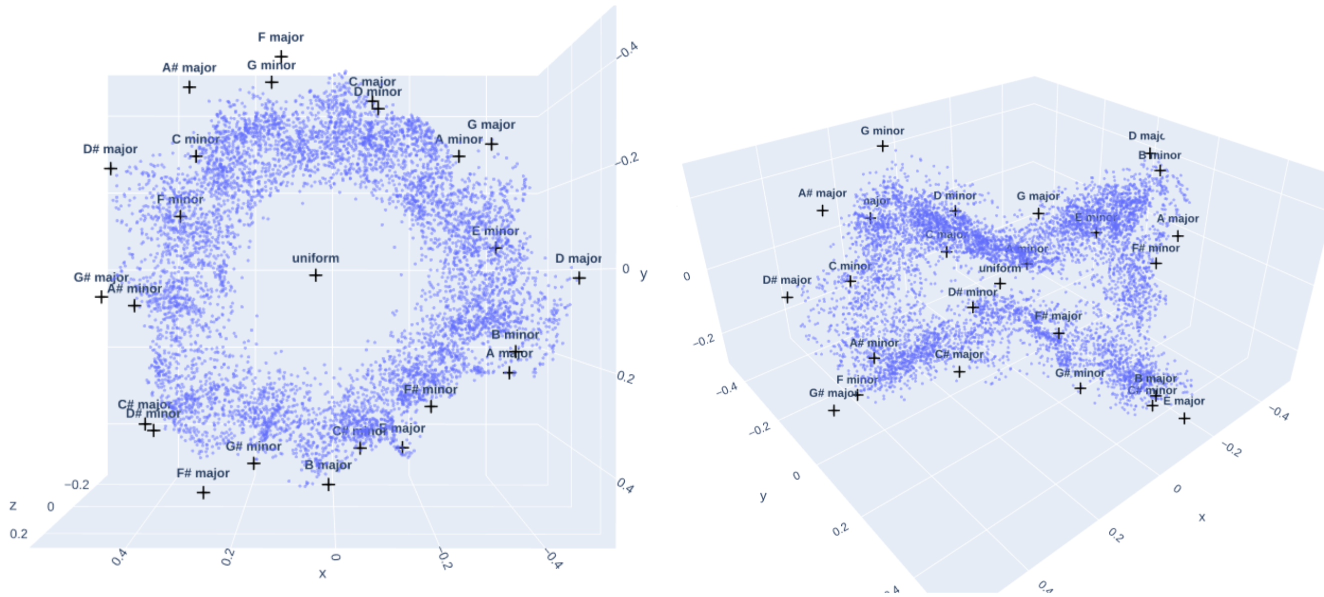
[2] Pitchscapes repository. [Online]. Available: https://github.com/robert-lieck/pitchscapes

[3] T. Bergstrom, K. Karahalios, and J. Hart, "Isochords: visualizing structure in music," 01 2007, pp. 297–304.

[4] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000. [Online]. Available: https://science.sciencemag.org/content/290/5500/2319

[5] R. Connor, F. A. Cardillo, R. Moss, and F. Rabitti, "Evaluation of jensen-shannon distance over sparse data," in *Similarity Search and Applications*, N. Brisaboa, O. Pedreira, and P. Zezula, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 163–168.

[6] Online 3d figures. [Online]. Available: https://dcmlab.github.io/music_beyond_major_and_minor_jellouli_mezghani_hadidane

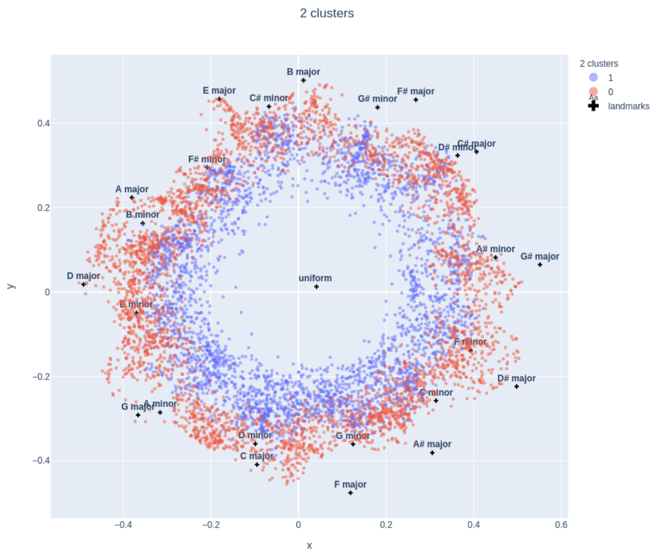


(a) Top view (b) Side view

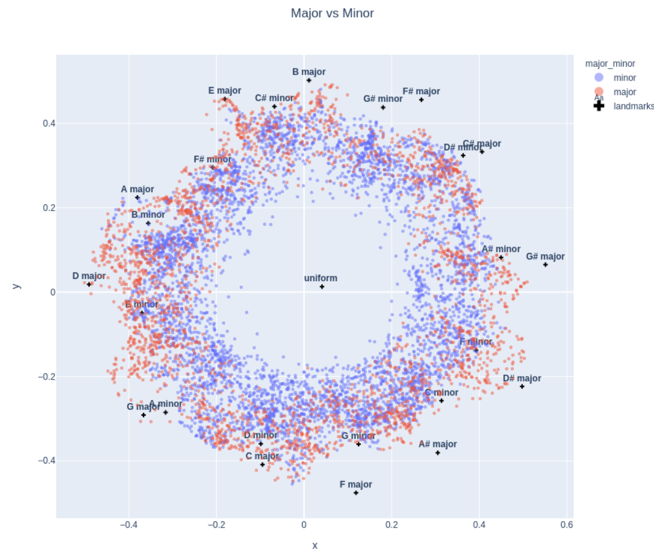
Figure 4: 3D dimensionally reduced data points

(a) Key coloring (b) Relative width coloring

Figure 5: 2D projection of data points colored by key and relative width



(a) 2 clusters coloring



(b) Tonality coloring

Figure 6: Clustering results with 2 clusters vs the tonality coloring

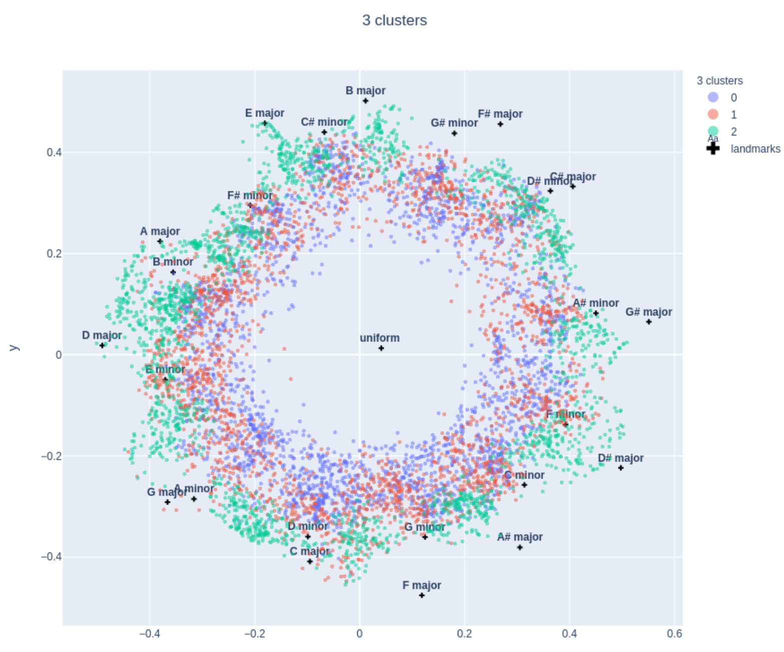
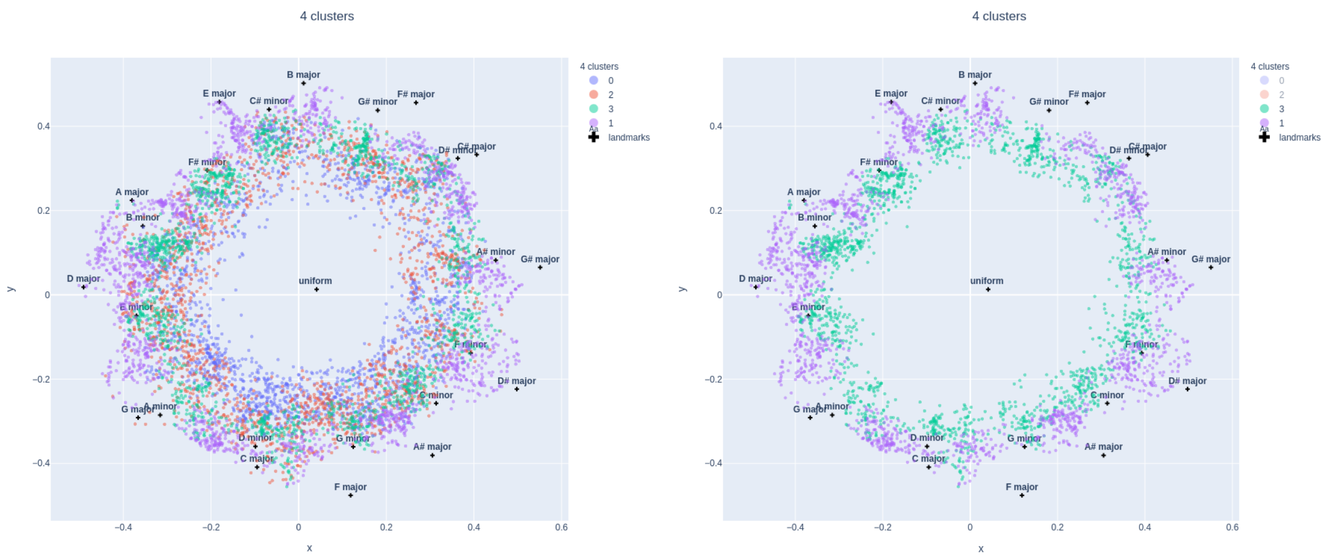


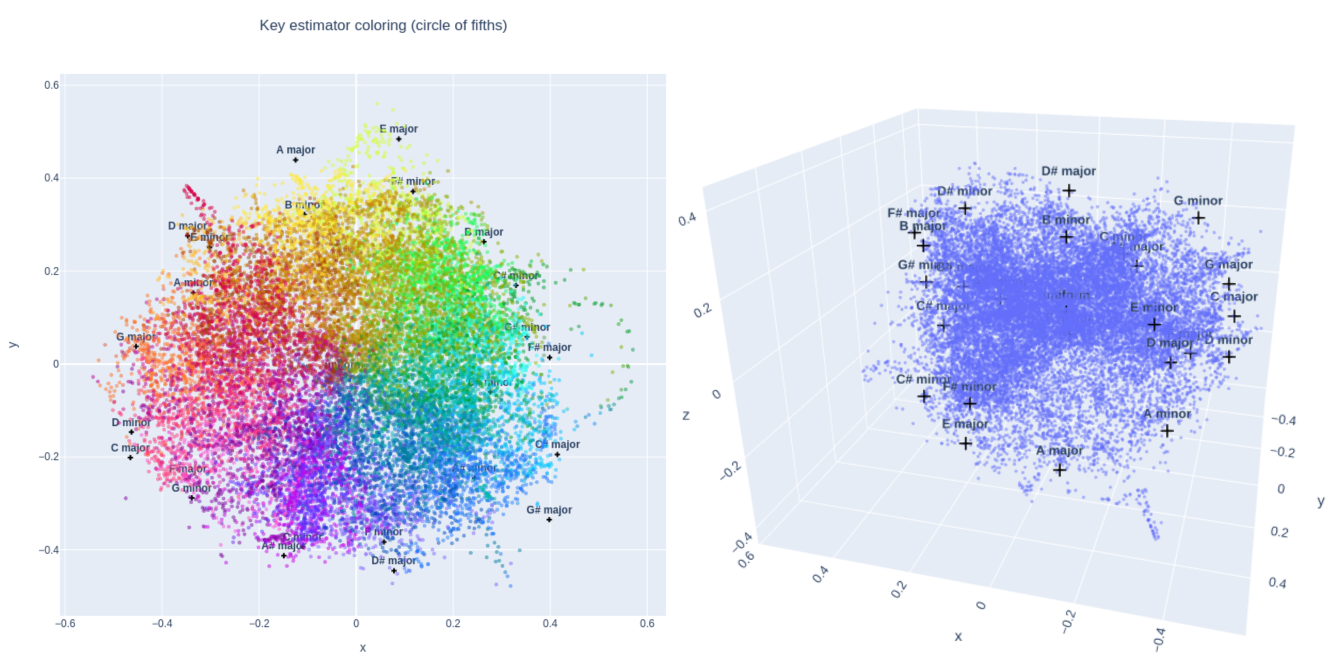
Figure 7: 3 clusters coloring



(a) 4 clusters coloring

(b) Clusters 1 and 3 isolated

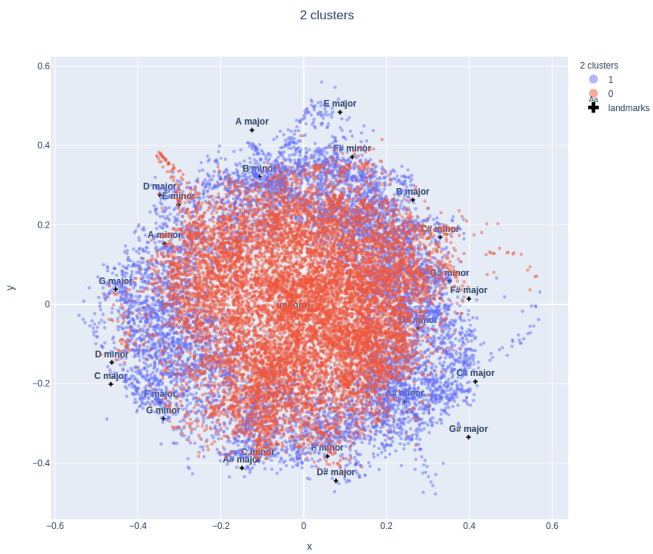
Figure 8: 2D clustering results with 4 clusters



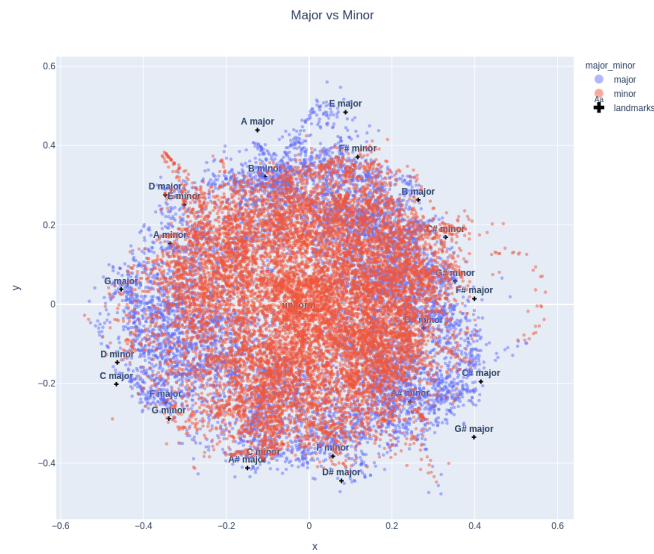
(a) Key coloring

(b) Side view

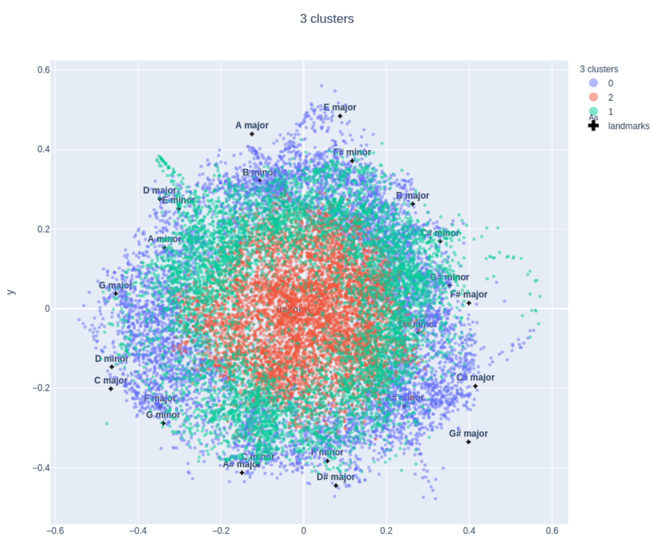
Figure 9: Visualizations of the selected composers' data points



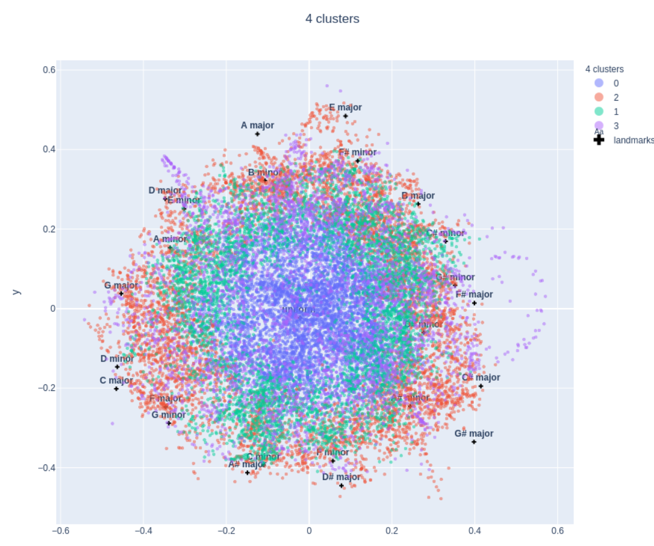
(a) 2 clusters coloring



(b) Tonality coloring



(c) 3 clusters coloring



(d) 4 clusters coloring

Figure 10: Selected composers clustering results