

# Содержание

<b>Аннотация</b>	<b>3</b>
<b>Основные термины, определения и сокращения</b>	<b>3</b>
<b>1 Введение</b>	<b>4</b>
1.1 Постановка задачи . . . . .	4
1.2 Цель и задачи . . . . .	4
1.3 Статьи и материалы . . . . .	5
<b>2 Изучение геометрических свойств имеющихся данных</b>	<b>5</b>
2.1 Построение матрицы расширенных аккордов по музыкальному произведению	5
2.2 Предварительный анализ . . . . .	5
2.3 Пространственное представление расширенных аккордов . . . . .	7
2.3.1 Использование SVD . . . . .	7
2.3.2 Анализ длительностей расширенных аккордов . . . . .	8
2.3.3 Анализ громкости звучания нот расширенных аккордов . . . . .	14
2.3.4 Анализ используемых нот в расширенных аккордов . . . . .	16
2.4 Результаты кластеризации . . . . .	20
<b>3 Алгоритмический подход</b>	<b>20</b>
3.1 Описание алгоритма . . . . .	20
3.2 Использование результатов кластеризации . . . . .	21
<b>4 Вероятностный подход</b>	<b>22</b>
4.1 Описание алгоритма . . . . .	22
4.2 Использование результатов кластеризации . . . . .	22
<b>5 Результаты</b>	<b>23</b>
<b>Список литературы</b>	<b>26</b>

## Аннотация

Еще Леонард Эйлер в XVIII-ом веке обратил внимание на диаграмму, которую он назвал «tonnetz». Диаграмма позволяет описать структуру музыкального произведения. Современные технологии позволяют использовать подобные геометрические структуры для генерации музыки. В рамках проекта я предложу два метода генерации музыкальных произведений, основывающиеся на их геометрическом представлении. Первый подход — алгоритмический (результат генерации детерминированный), генерация происходит на основе одного имеющегося музыкального произведения; второй — вероятностный (результат генерации случайный), генерация происходит на основе большого количества музыкальных произведений.

## Основные термины, определения и сокращения

- Гармония – объединение звуков в созвучия и их слаженное закономерное последование с точки зрения человеческого восприятия.
- $\Omega = \{\omega_1 \dots \omega_{128}\}$  – множество всех музыкальных нот в рассматриваемом представлении аудиофайлов в midi-формате.
- $v_t \in \mathbb{Z}^{128}$  – вектор-аккорд, для  $i \in \{1 \dots 128\}$   $i$ -ая компонента отвечает громкости ноты  $w_i$  в момент  $t$  музыкального произведения.
- $V = \{v_{t_1} \dots v_{t_n}\}$  – последовательность аккордов данного музыкального произведения, где моменты времени  $t_1 \dots t_n$  отвечают смене аккорда.
- $v^+ \in \mathbb{Z}^{129}$  – расширенный вектор-аккорд, для  $i \in \{1 \dots 128\}$   $i$ -ая компонента отвечает громкости ноты  $w_i$ ; для  $i \in \{129\}$   $i$ -ая компонента отвечает длине временного промежутка, в течение которого играл соответствующий аккорд до очередной смены в музыкальном произведении.
- $V^+ = [v_1^+ \dots v_n^+] \in \mathbb{Z}^{129 \times n}$  – матрица музыкального произведения, состоящего из  $n$  расширенных аккордов.

# 1 Введение

## 1.1 Постановка задачи

В рамках этой работы будем рассматривать музыку, созданную человеком на фортепиано. Благодаря представлению аудиофайлов в midi-формате в любой момент времени нам доступна информация об используемых в произведении нотах и их громкости. Имея доступ к большому количеству произведений, хотелось бы выяснить, есть ли зависимости между порядком нот, их громкостями и временными промежутками, в течение которого они играют. Для изучения этих зависимостей необходимо предложить объекты, которыми можно описать имеющиеся музыкальные произведения. С одной стороны, для анализа зависимостей эти объекты должны содержать значительную информацию о произведении или его части, с другой – их количество должно быть ограничено, чтобы на основе небольшого количества имеющихся произведений восстановить зависимости.

## 1.2 Цель и задачи

Цель моей работы заключается в создании детерминированного и вероятностного алгоритмов, позволяющих генерировать музыку. Для достижения цели поставлены следующие задачи:

- изучение работ, посвященных созданию музыки.
- изучение геометрических свойств имеющихся данных.
- кластеризация данных на основе выделенных свойств.
- создание детерминированного алгоритма, генерирующего музыкальные произведения на основе выделенных кластеров и одного имеющегося произведения.
- создание вероятностного алгоритма, генерирующего музыкальные произведения на основе выделенных кластеров и большого количества имеющихся произведений.
- программная реализация предложенных алгоритмов.
- сравнение результатов работы алгоритмов с музыкальными произведениями, созданными человеком.

## 1.3 Статьи и материалы

Заметные продвижения в области алгоритмического создания музыки были достигнуты авторами статьи [3]. Новизна подхода заключается в рассмотрении музыкального произведения, как последовательности токенов, отвечающих сразу нескольким изменениям звука или ритма, тогда как ранее изменения обрабатывались по одному.

Автор программы [1] визуализировал представление музыкальных произведений на графах.

После изучения статьи и программы я попробовал объединить плюсы обоих подходов в своей работе.

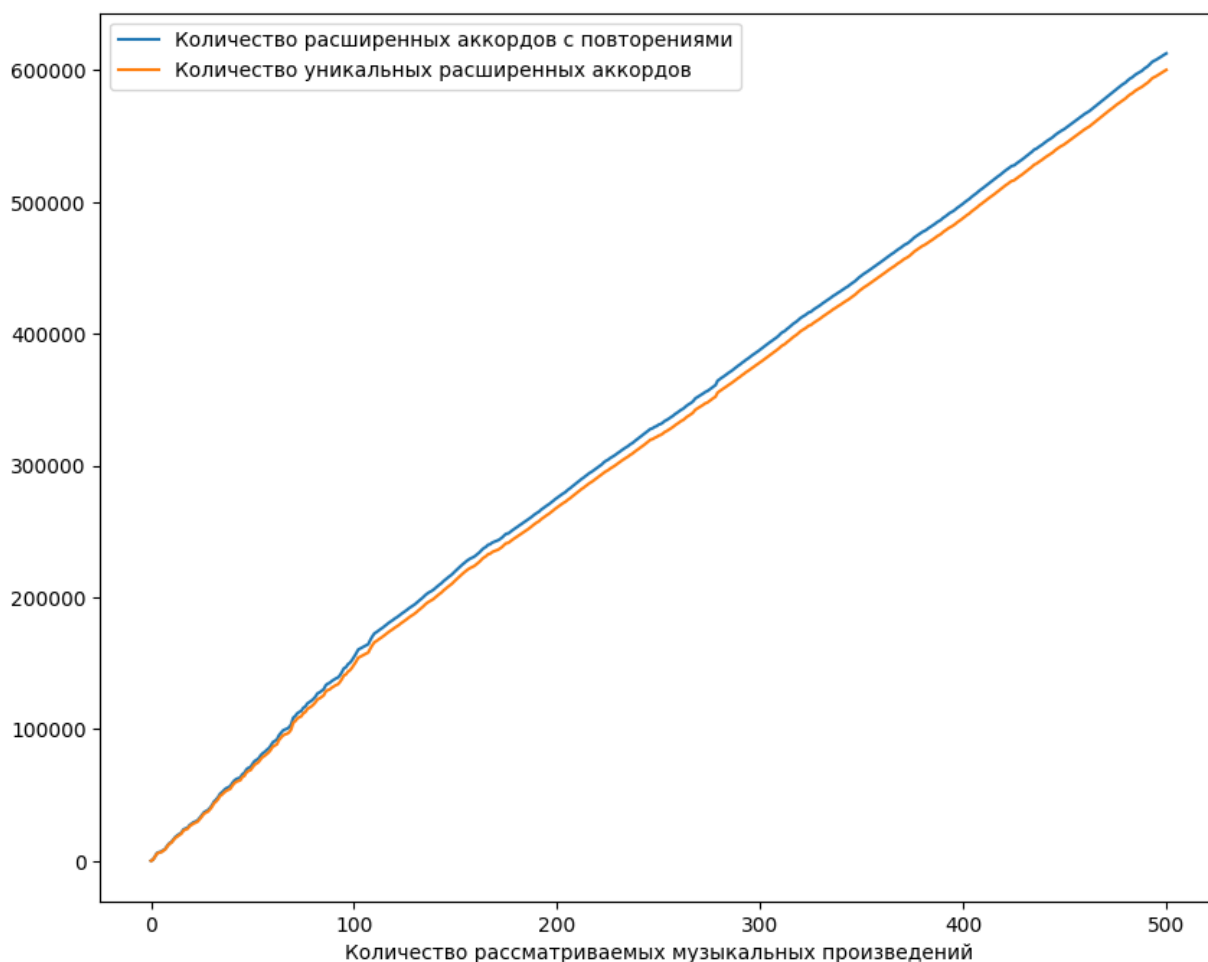
## 2 Изучение геометрических свойств имеющихся данных

### 2.1 Построение матрицы расширенных аккордов по музыкальному произведению

Рассмотрим множество всех нот  $\Omega = \{\omega_1 \dots \omega_{128}\}$  в представлении аудиофайлов в midi-формате. В каждый момент времени  $t$  выделим информацию о играющих нотах и их громкости, после чего построим вектор-аккорд  $v_t \in \mathbb{Z}^{128}$ ,  $i$ -ая компонента отвечает громкости ноты  $w_i$  в момент  $t$  музыкального произведения. Теперь любое произведение можно однозначно представить в виде последовательности  $V = \{v_{t_1} \dots v_{t_n}\}$ , где моменты времени  $t_1 \dots t_n$  отвечают смене аккорда. На основании имеющейся последовательности сформируем матрицу музыкального произведения  $V^+ \in \mathbb{Z}^{129 \times n}$ : верхний блок матрицы с размерами  $128 \times n$  будет описывать последовательные аккорды в музыкальном произведении, 129-ая строка матрицы будет описывать продолжительности соответствующих аккордов. Столбцы матрицы  $V^+$  будем называть расширенными аккордами музыкального произведения.

### 2.2 Предварительный анализ

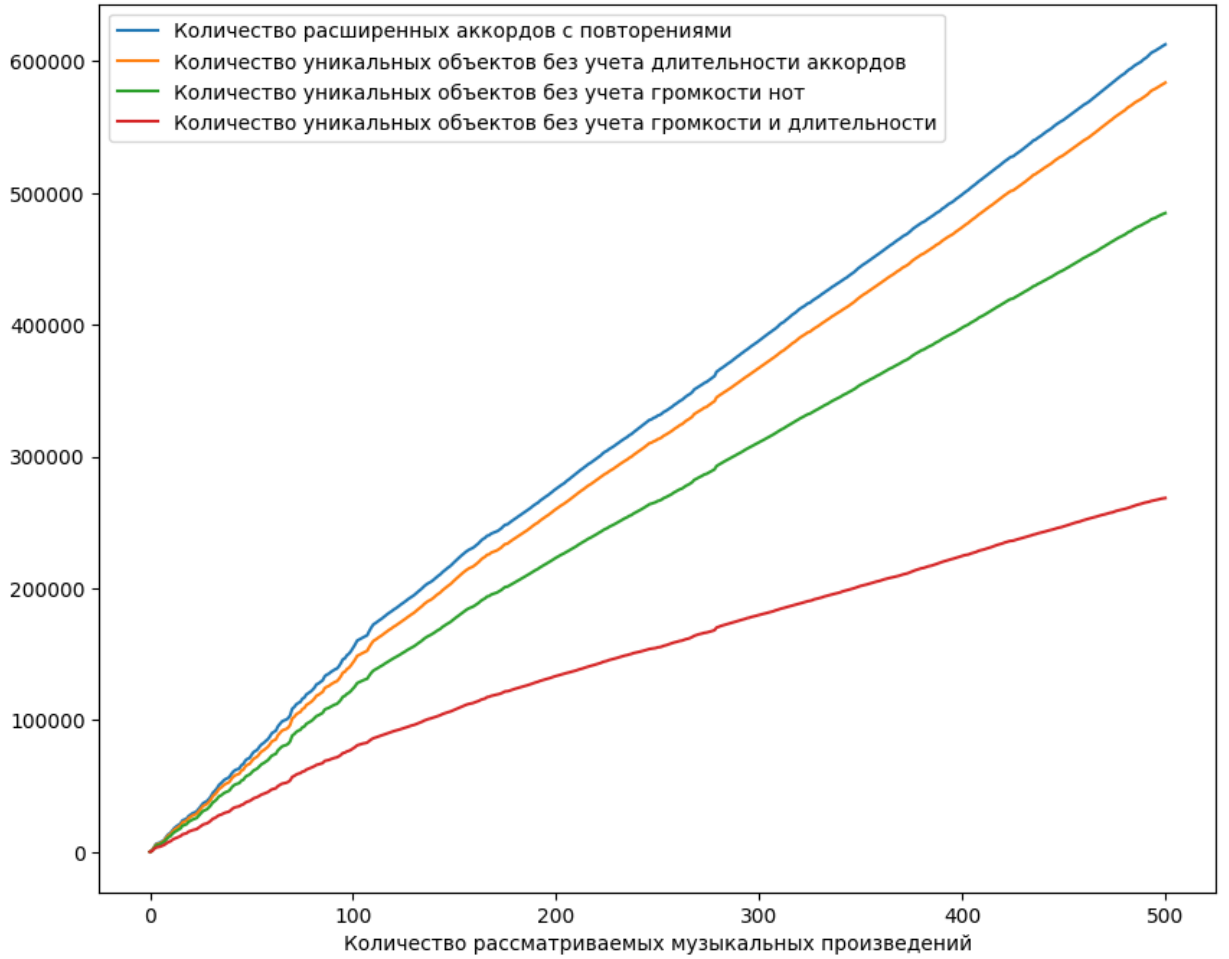
Среди рассматриваемых произведений выберем  $N = 500$  случайных. Проанализируем количество уникальных расширенных аккордов и количество расширенных аккордов с повторениями для рассматриваемых музыкальных произведений:



В рассматриваемом представлении аудиофайлов в midi-формате значения громкости нот и продолжительности аккордов описываются целыми числами, поэтому при подсчете уникальных аккордов не возникает операций с плавающей точкой и результат в точности отражает действительность. Таким образом, из полученных зависимостей становится понятно, что почти все расширенные аккорды являются уникальными, а значит они не подходят в качестве объектов, на основании которых можно выявлять зависимости в музыкальных произведениях.

Предложим несколько вариантов упрощения модели, позволяющих сократить количество уникальных рассматриваемых объектов:

- не будем учитывать громкость нот: если нота присутствовала в аккорде, положим ее громкость равной 1, иначе 0 – в этом случае верхний блок матрицы  $V^+$  музыкального произведения с размерами  $128 \times n$  будет состоять из 1 и 0.
- не будем учитывать продолжительность аккордов – в этом случае от матрицы  $V^+$  останется лишь верхний блок с размерами  $128 \times n$ .
- не будем учитывать как громкость нот, так и продолжительность аккордов.



Таким образом, даже в самой примитивной модели, теряющей большую часть данных о музыкальном произведении, отношение количества уникальных объектов по сравнению с общим количеством объектов довольно большое.

В рамках этой работы я предложу метод кластеризации расширенных аккордов на основании их геометрической структуры, благодаря которому удастся сохранить данные о громкости и продолжительности аккордов, а также об используемых нотах. После применения этого метода станет возможным исследование зависимостей между полученными объектами.

## 2.3 Пространственное представление расширенных аккордов

### 2.3.1 Использование SVD

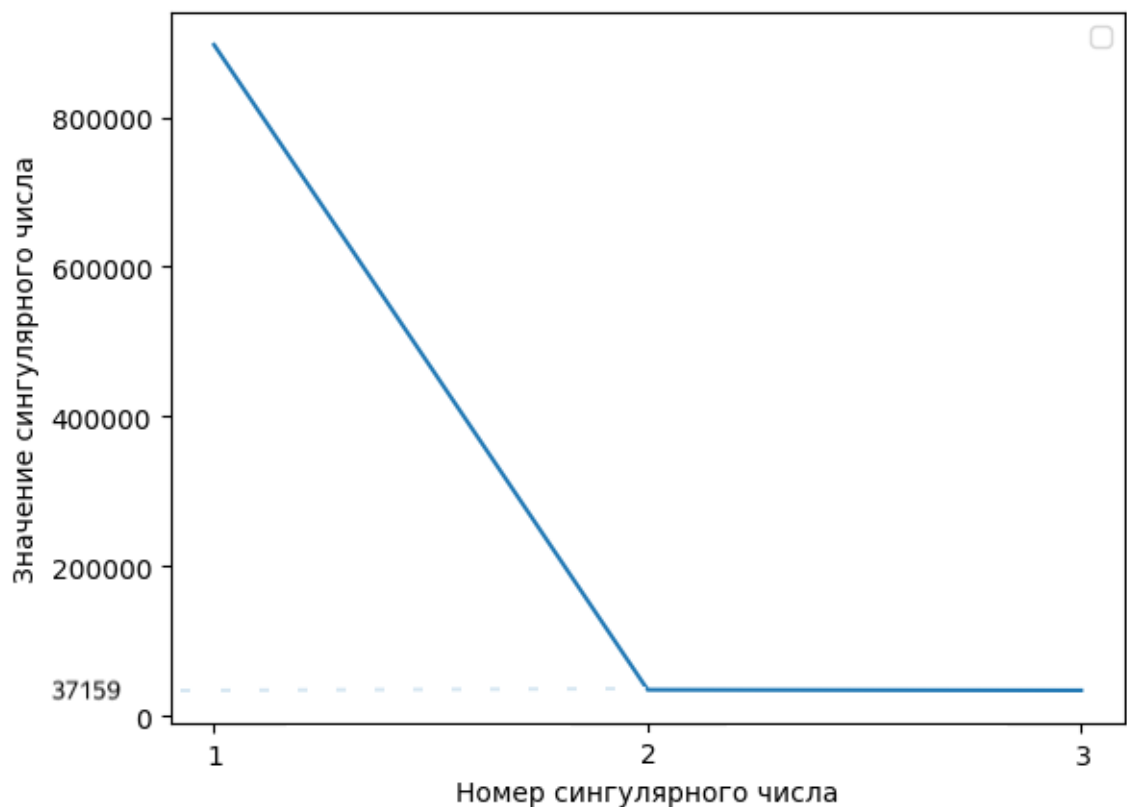
Объединим матрицы рассматриваемых музыкальных произведений в матрицу  $W = [V_1^+ \dots V_N^+]$ , после чего рассмотрим SVD разложение [4] матрицы  $W = USV^T$ . Согласно теореме Экарта-Янга [4],  $W_r = U_r S_r V_r^T$  (матрицы  $U_r, S_r, V_r$  составляют первые  $r$  столбцов матриц  $U, S, V$  соответственно) – лучшее приближение ранга  $r$  матрицы  $W$  в евклидовой норме. Для

$r = 3$  матрица  $S_r V_r^T$  будет соответствовать проекции расширенных аккордов на трехмерное пространство. В этом случае расположенные в трехмерном пространстве точки, соответствующие столбцам матрицы  $S_r V_r^T$ , будем называть пространственным представлением матрицы  $W$ .

Отметим также, что на практике ограничения по памяти и времени могут не позволить хранить или эффективно выполнять операции над рассматриваемой матрицей  $W_{129 \times [n_1 + \dots + n_N]}$ , где  $n_1 + \dots + n_N \sim 10^6$ . Будем дополнительно учитывать разреженность матрицы и использовать модуль `scipy.sparse`<sup>1</sup> из библиотеки SciPy. Хранение матрицы в формате Compressed Sparse Column matrix позволяет эффективно обращаться к столбцам матрицы  $W$  и строить SVD разложение.

### 2.3.2 Анализ длительностей расширенных аккордов

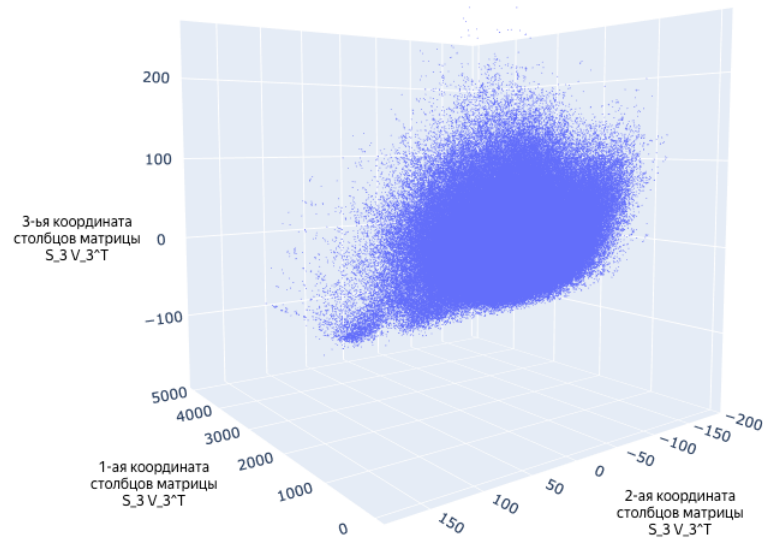
Рассмотрим значения первых трех сингулярных чисел матрицы  $W$  и компоненты матрицы  $S_3 V_3^T$ :



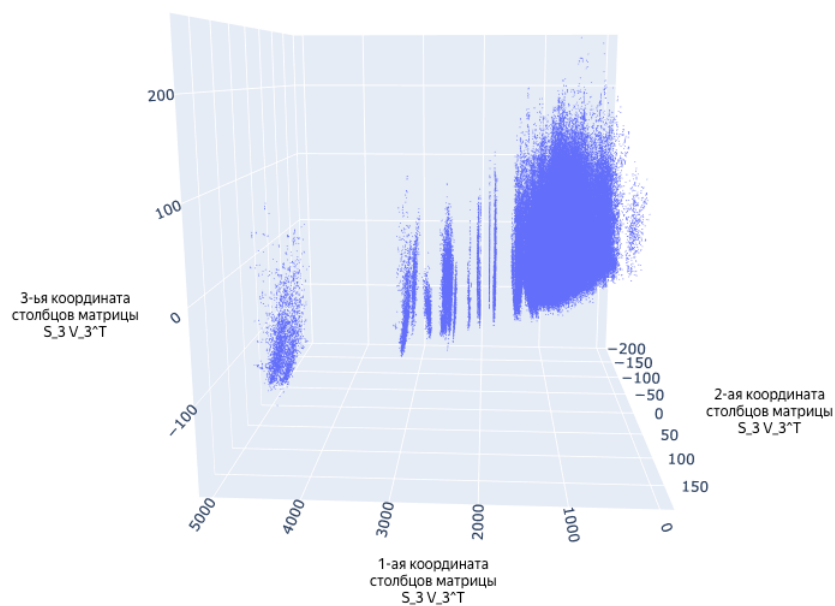
Первое сингулярное значение на порядки превосходит остальные, в силу ортонормированности столбцов  $V$  в пространственном представлении также заметен этот эффект.

<sup>1</sup><https://docs.scipy.org/doc/scipy/reference/sparse.html>, дата обр. 25.05.2023

Тут и далее пространственное представление матрицы  $W$  будем строить следующим образом: расположим точки в трехмерном пространстве так, чтобы координаты по осям отвечали компонентам столбцов матрицы  $S_3 V_3^T$ . Расположение получившейся фигуры значительно смещено по оси, соответствующей первому сингулярному вектору  $u_1$ :

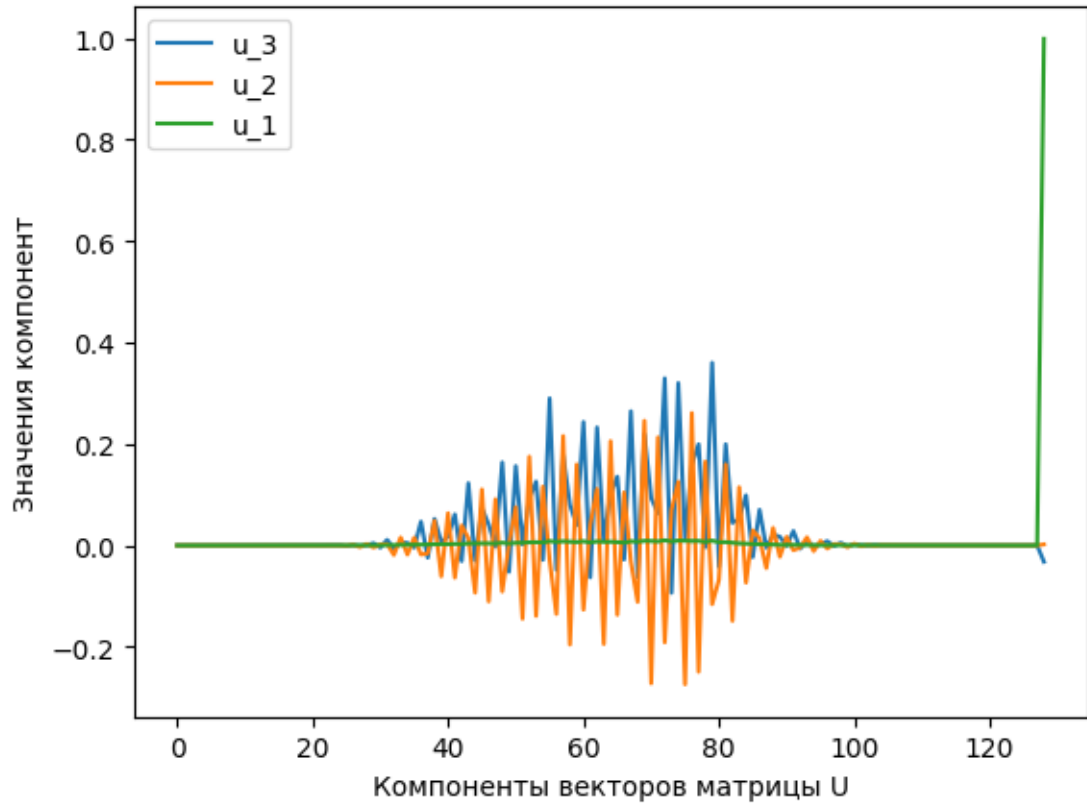


При рассмотрении проекции на ось видно, что точки распределились по нескольким кластерам, каждый из которых характеризует координата, соответствующая  $u_1$ :



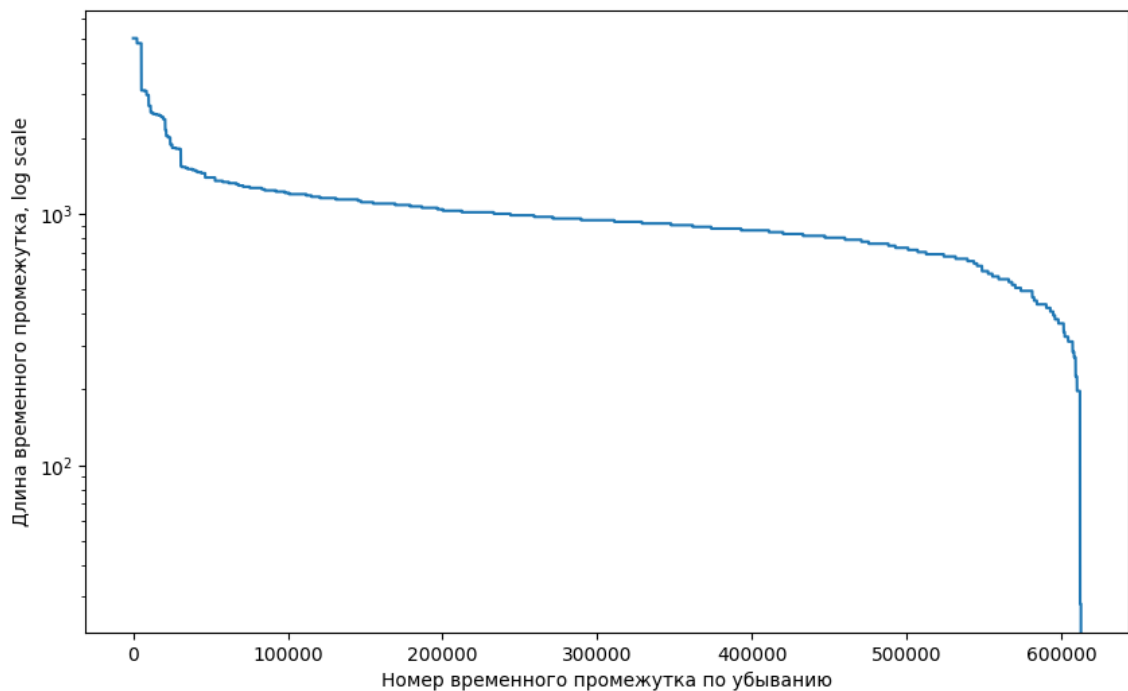


Рассмотрим компоненты сингулярных векторов матрицы  $U_3$ :

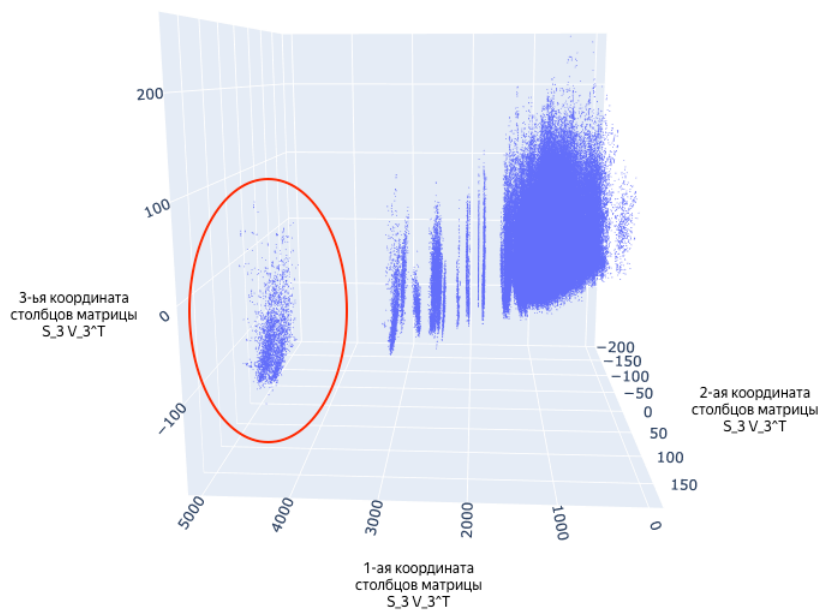


Из графика становится понятно, что вектор  $u_1$  отвечает за длительность аккордов,  $u_2$  и  $u_3$  – за звучание. Разделение признаков может быть связано с тем, что некоторые значения длительностей аккордов существенно превышают значения громкостей нот.

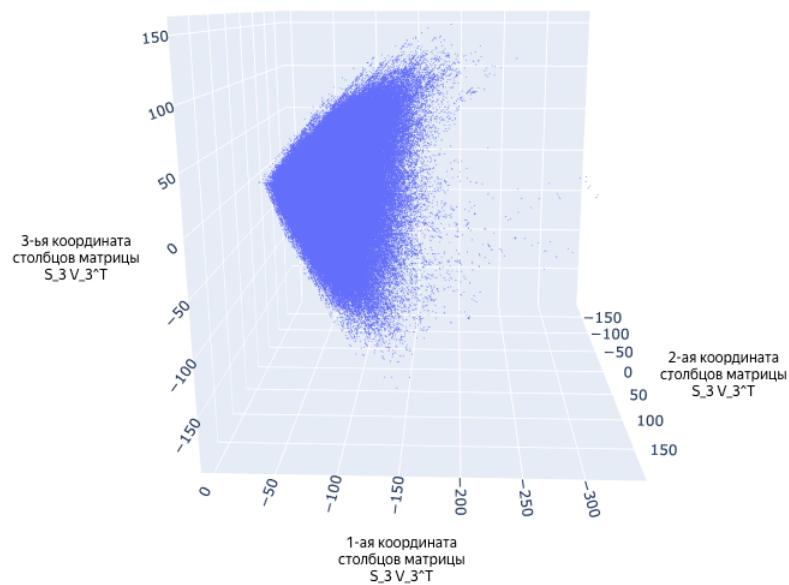
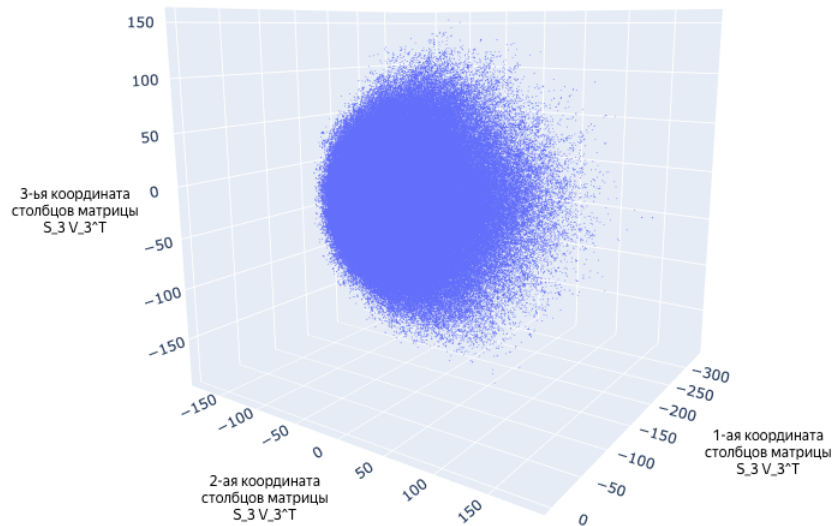
Действительно, распределение длин временных промежутков показывает, что некоторые аккорды звучат аномально долго, причем количество таких аккордов крайне мало:



Попробуем далее отказаться от рассмотрения аккордов, длительность звучания которых в разы превышает длительности звучания остальных. В рамках предложенной модели значения временных промежутков отражают не суммарную длительность определенного аккорда, а время до очередной смены, то есть описанные аккорды соответствуют произведениям, в которых долго не происходит смена звука. Описанные аккорды будут соответствовать выделенной группе точек:

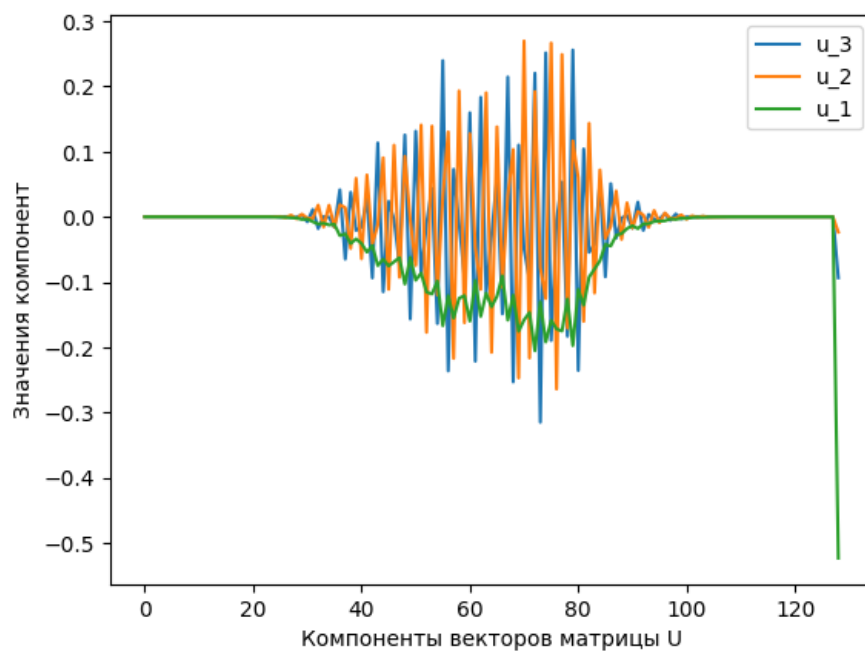


Нормируем длительности оставшихся аккордов так, чтобы значения попали в тот же отрезок, что и значения громкости нот – то есть  $[0, 127]$  и снова рассмотрим SVD разложение. В пространственном представлении аккордов теперь не выделены кластеры:



Однако даже после фильтрации и нормировки длительностей аккордов среди компонент первого вектора выделяется отвечающая за продолжительность, а остальные распределены в соответствии с частотой используемых нот. Этот эффект можно интерпретировать

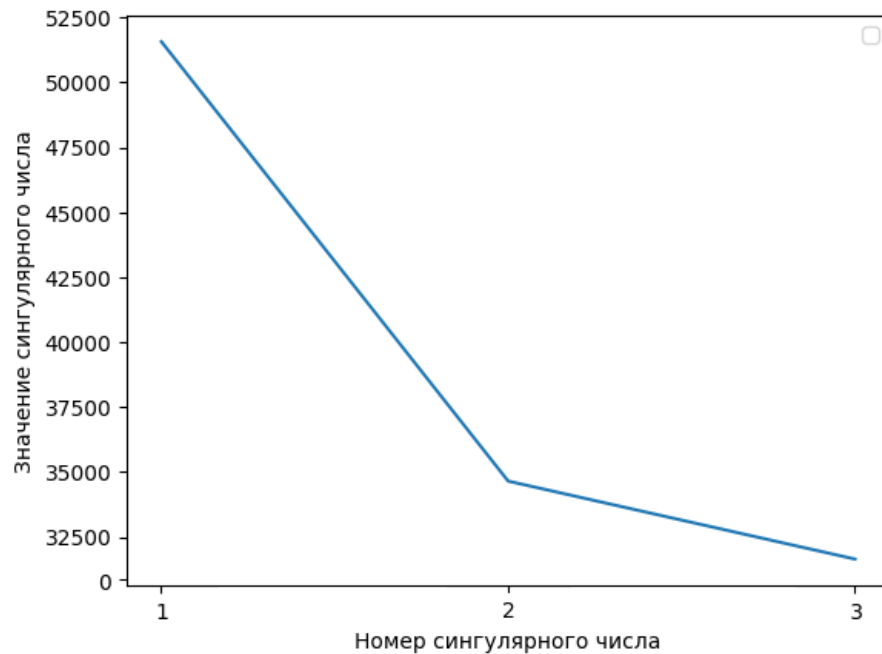
следующим образом: между длительностью аккорда и используемыми в нем нотами нет зависимости — этот результат вполне согласуется с предполагаемой действительностью.



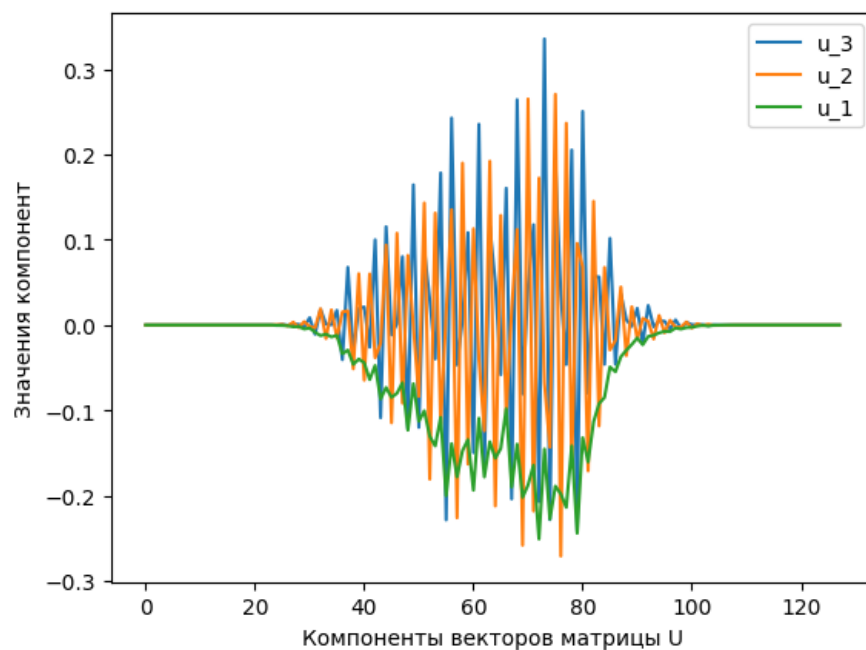
Таким образом, после пространственного анализа длительностей аккордов можно предложить вывод: продолжительность аккорда можно выделить, как самостоятельный признак, характеризующий аккорды, так как длительность не связана с используемыми в нём нотами. При этом SVD разложение матрицы  $W$  позволяет выделить кластеры длительностей аккордов.

### 2.3.3 Анализ громкости звучания нот расширенных аккордов

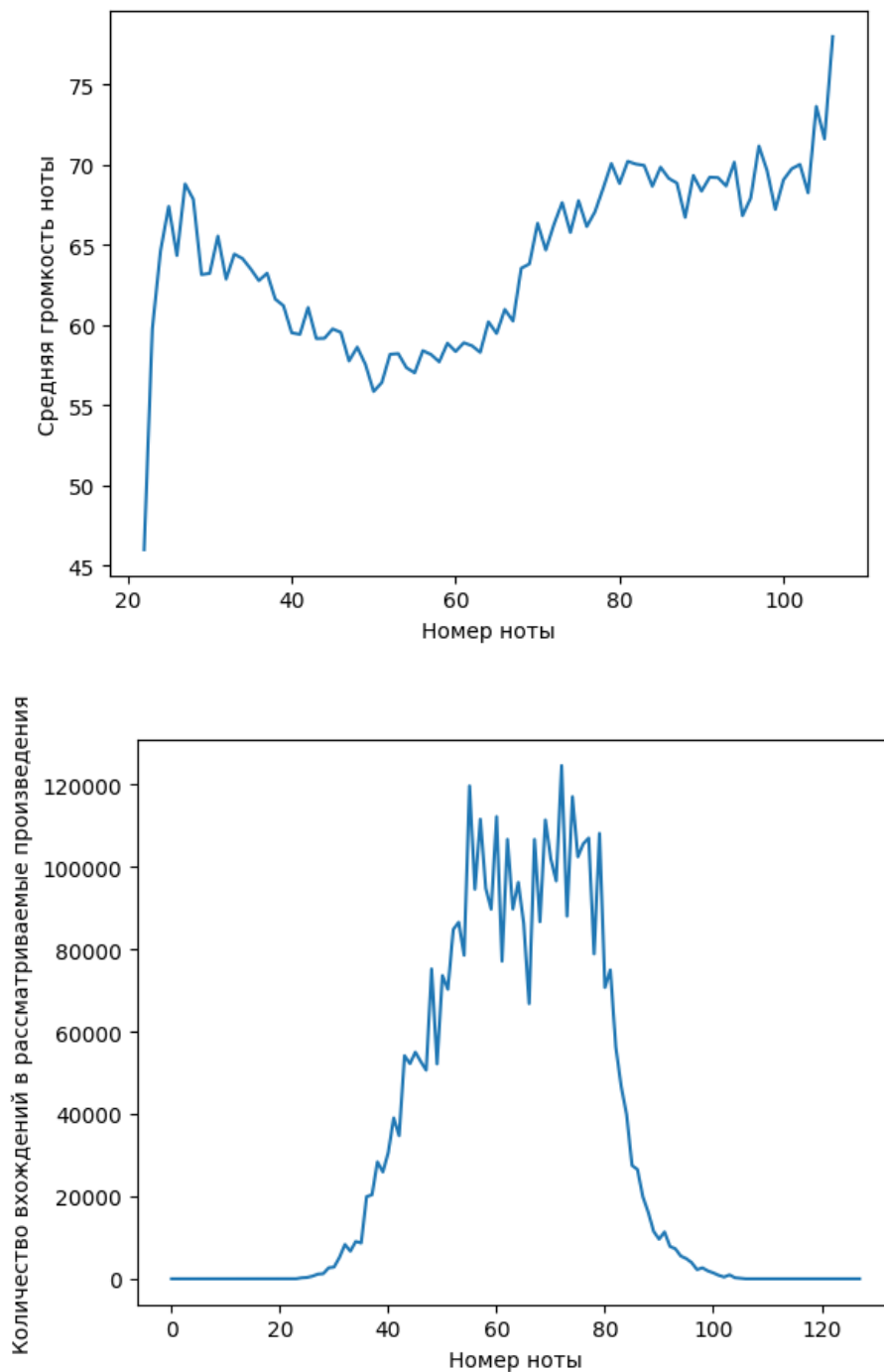
Перейдем к рассмотрению громкостей нот в аккордах, для этого снова построим SVD, но теперь для верхнего блока размеров  $128 \times [n_1 + \dots + n_N]$  матрицы  $W$ . Разница в значения первых трех сингулярных чисел существенно сократилась:



Каждый из векторов матрицы  $U_3$  отвечает громкости звучания нот, причем их компоненты распределены похожим образом:



Компоненты первого сингулярного вектора  $u_1$  можно интерпретировать как средние значения громкостей нот, взвешенных с учетом частоты их использования. Рассмотрим признаки, отвечающие громкостям и частотам нот отдельно:

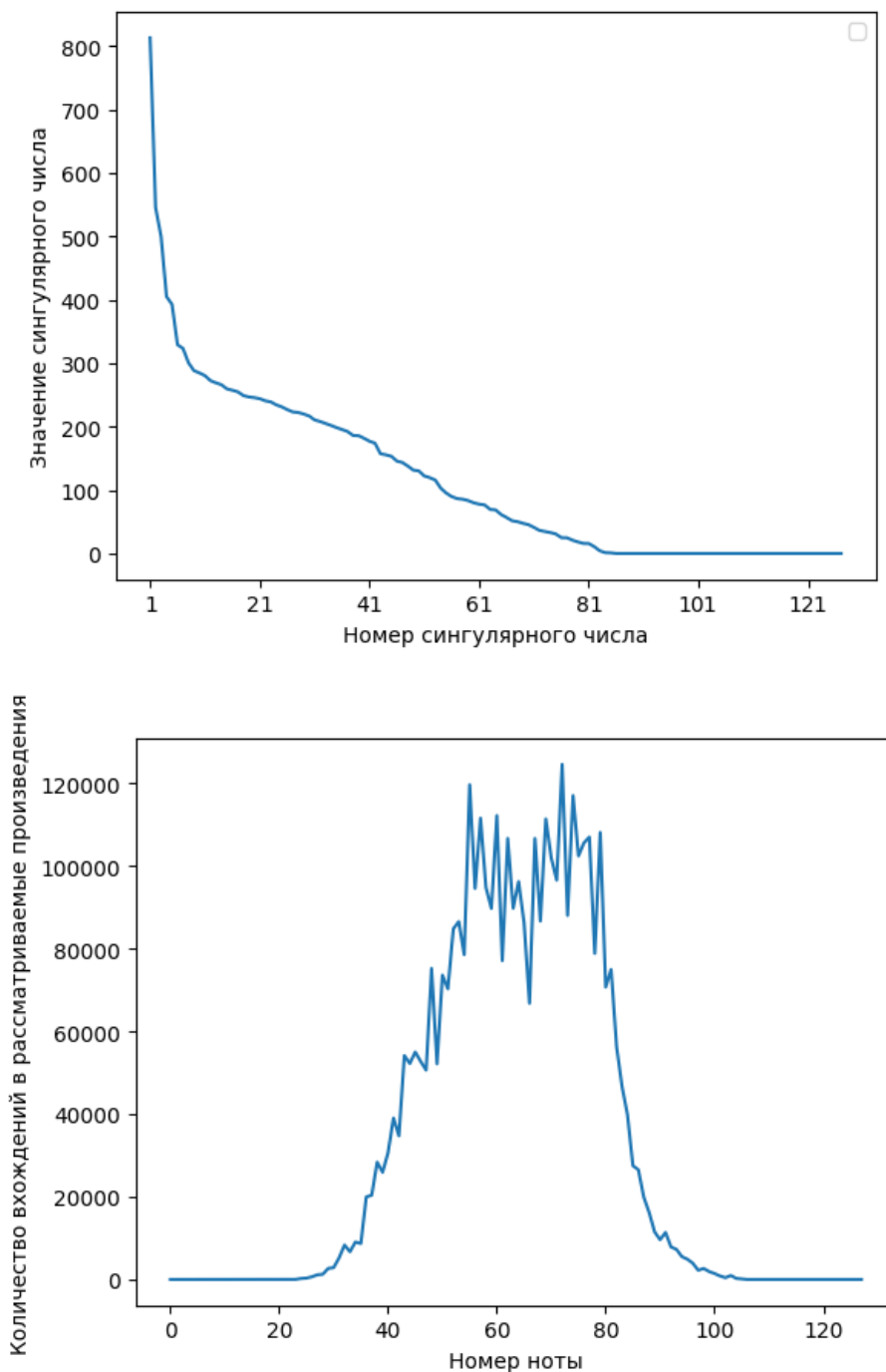


Получаем следующий результат: большинство используемых нот сосредоточено на отрезке  $[50, 80]$ , при этом для нот на первой половине этого отрезка характерно менее громкое звучание, чем для нот второй половины. Вероятно, это связано с тем, что более высокие ноты используются для задания основного мотива, а более низкие – для поддержания фоновой мелодии. В силу наблюдаемых закономерностей в рамках задачи кластеризации признаков оправдано использование средних значений громкостей для нот.

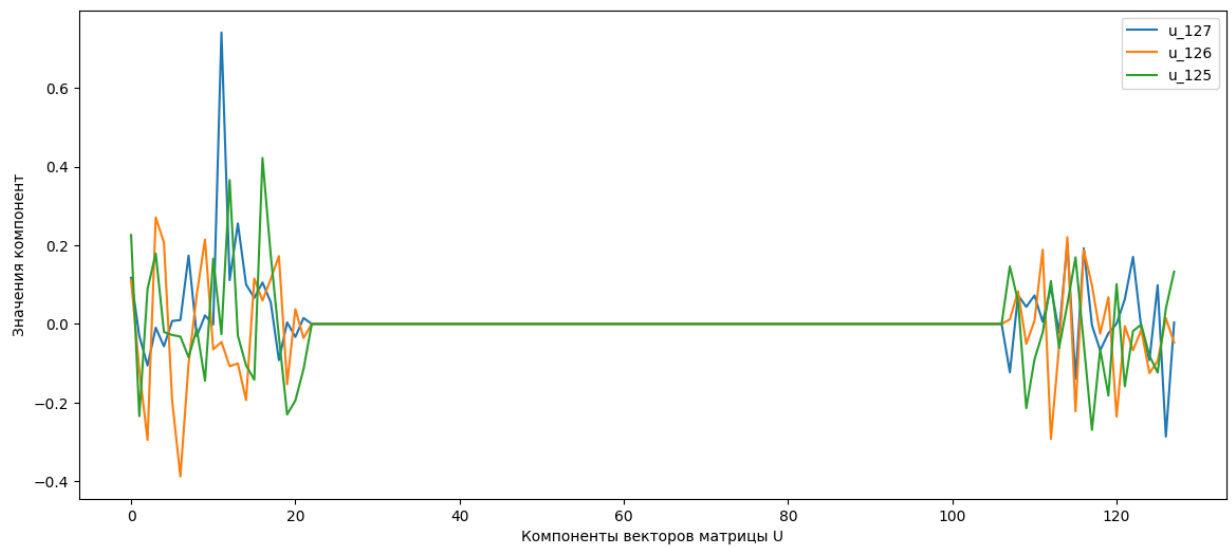
### 2.3.4 Анализ используемых нот в расширенных аккордах

Перейдем к рассмотрению зависимостей между играющими нотами в аккордах, для этого снова построим SVD, для верхнего блока размеров  $128 \times [n_1 + \dots + n_N]$  матрицы  $W$ . Также не будем учитывать громкость нот: если нота присутствовала в аккорде, положим ее громкость равной 1, иначе 0.

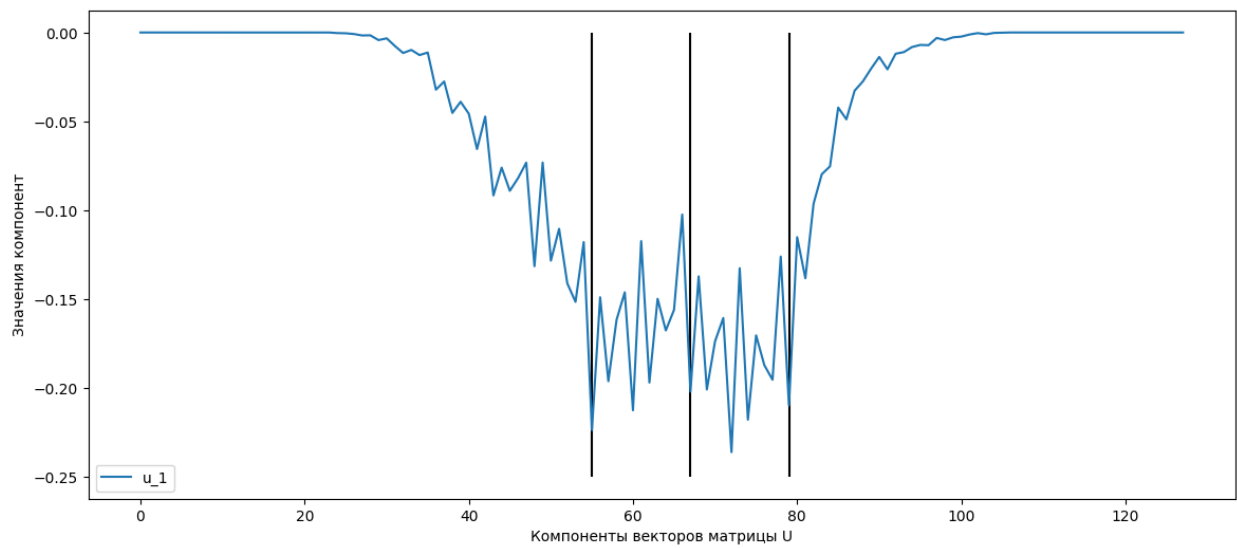
Теперь рассмотрим полное SVD разложение для описанной матрицы. Сингулярные числа с номерами  $i > 80$  имеют значения, близкие к нулю. Это объясняется тем, что в рассматриваемых произведениях используется порядка 80 различных нот – это значение определяет ранг рассматриваемой матрицы.



Компоненты левых сингулярных векторов, отвечающих младшим сингулярным числам, соответствуют наиболее редко встречающимся нотам:

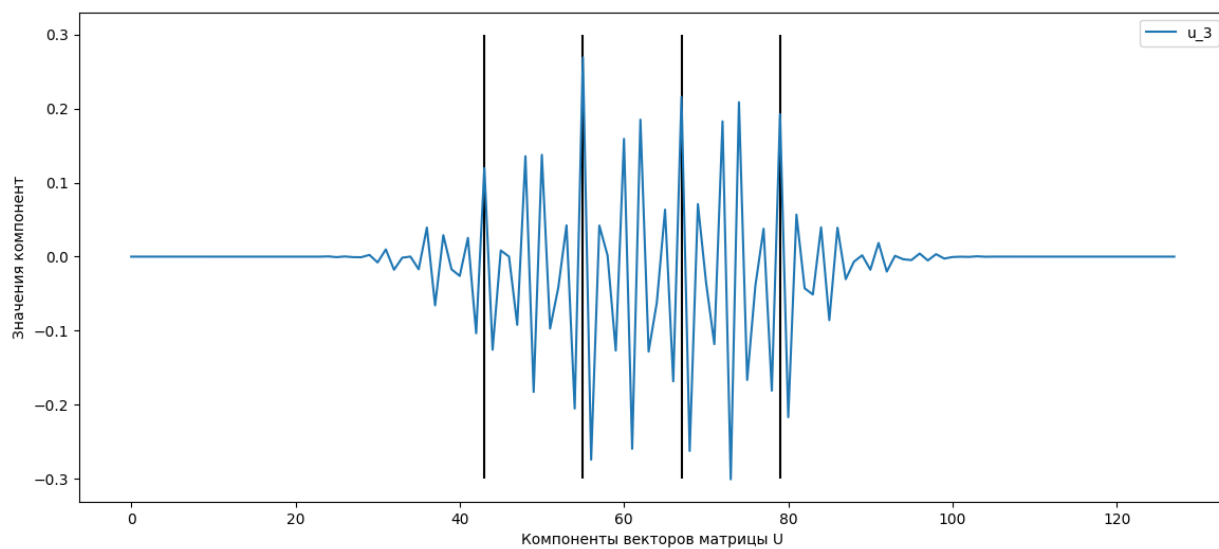
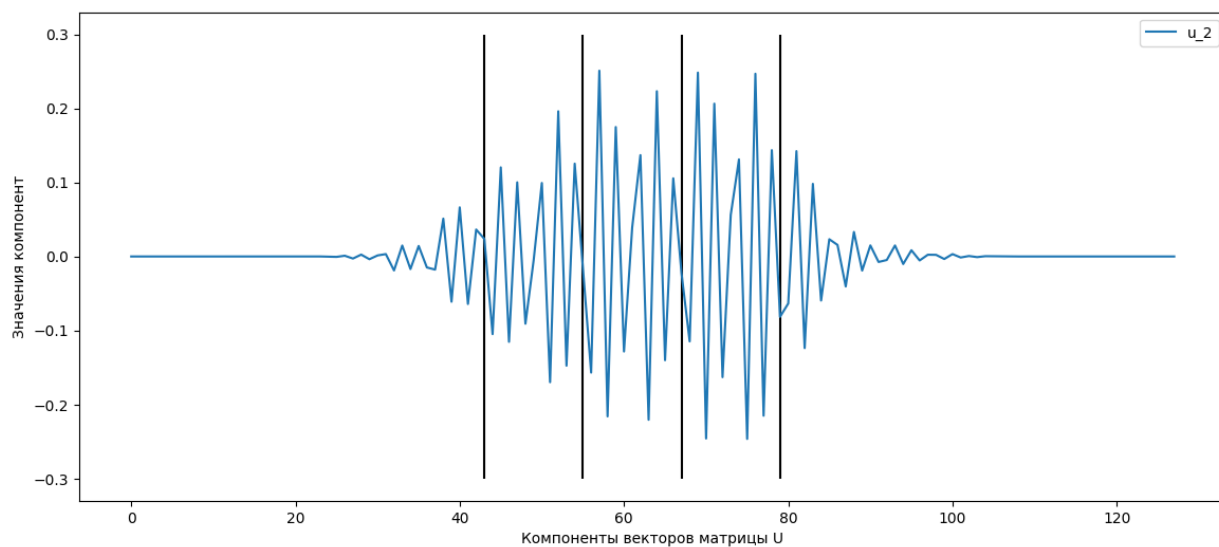


При этом распределение компонент старшего сингулярного вектора напоминает распределение частот используемых нот. Примечательно, что значения компонент на полуинтервалах  $[55, 67)$  и  $[67, 79)$  почти в точности совпадают:

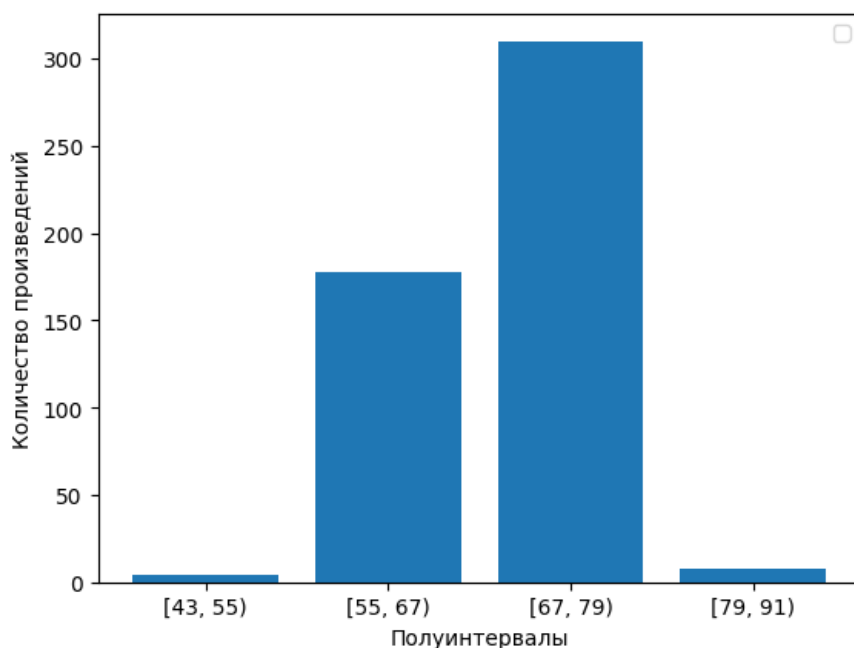




Этот же эффект наблюдается для второго и третьего левых сингулярных векторов  $u_2, u_3$ , но уже для трех полуинтервалов:  $[43, 55)$ ,  $[55, 67)$  и  $[67, 79)$ :



Для каждого из рассматриваемых произведений выясним, какой из полуинтервалов  $[43, 55)$ ,  $[55, 67)$ ,  $[67, 79)$ ,  $[79, 91)$  содержал большее количество нот:

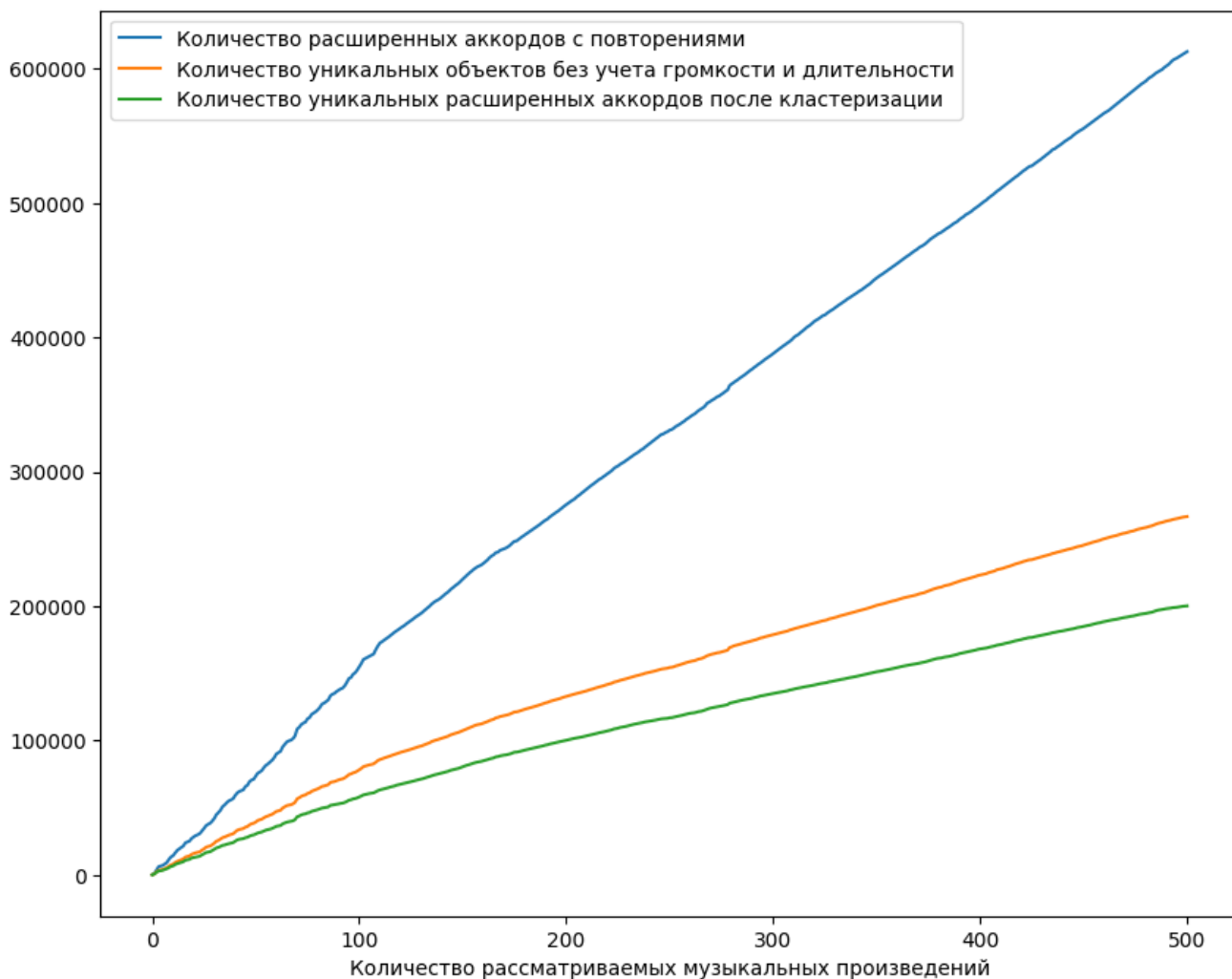


Произведения разделились на две основные группы: в первой из них большинство нот содержится в полуинтервале  $[55, 67)$ , во второй – в полуинтервале  $[67, 79)$ . Теперь становится понятно, почему среди компонент первого сингулярного вектора  $u_1$  удалось выделить два почти совпадающих полуинтервала, а не три: компоненты  $u_1$  взвешены с учетом частот соответствующих нот. Этот эффект в меньшей степени сказался на векторах  $u_2$  и  $u_3$ . Компоненты старших сингулярных векторов описывают закономерности в использовании нот в различных полуинтервалах длины 12, при этом наблюдаемые закономерности совпадают. То есть можно сделать следующее предположение: при смещении всех нот на 12 структура произведений и их звучание не должно существенно поменяться. Действительно, это соображение согласуется с музыкальной теорией: октава состоит в точности из 12 нот, при этом исполнение одно и того же музыкального произведения допускается в разных октавах – меняется только тон звучания.

Таким образом, мы получили обоснование возможности смещать ноты всего произведения на кратное 12 число нот. Воспользуемся этим в поставленной задаче кластеризации: для каждого музыкального произведения за счет смещений добьемся того, чтобы большинство нот содержалось в полуинтервале  $[55, 79)$ . Если после смещения остались ноты вне описанного полуинтервала, дополнительно сместим их в него независимо от остальных.

## 2.4 Результаты кластеризации

Сравним количество уникальных объектов для описанной ранее примитивной модели, не учитывающей как громкость нот, так и продолжительность аккордов, с количеством уникальных образов расширенных аккордов, полученных после кластеризации:



После кластеризации количество образов уникальных расширенных аккордов на основании их геометрической структуры позволяет анализировать зависимости между ними, кроме того предложенный метод кластеризации сохраняет данные о громкости и продолжительности аккордов, а также об используемых нотах.

## 3 Алгоритмический подход

### 3.1 Описание алгоритма

Прежде всего заметим, что в задаче генерации музыки на основе единственного произведения мы вынуждены в значительной степени опираться на существующую структуру и

зависимости музыкального произведения. В силу этого замечания сформулируем дополнительное предположение, обоснование разумности которого приведем позже: пусть из всевозможных последовательностей аккордов  $v_1 \dots v_n$  гармоничными кажутся музыкальные произведения, в которых смена звука происходит плавнее всего – то есть сумма  $\sum_{i=1}^{n-1} \|v_{\sigma(i)} - v_{\sigma(i+1)}\|$  наименьшая для фиксированного набора  $v_1 \dots v_n$  и всевозможных перестановок  $\sigma$  на элементах  $1 \dots n$ .

Рассмотрим неориентированный граф  $G = \{V, E\}$ ,  $E = \{(v_i, v_{i+1}) \mid i \in \{1 \dots n - 1\}\}$ . В текущей версии вес ребра  $(v_i, v_{i+1})$  определяется евклидовой нормой разности  $v_i - v_{i+1}$ . Переформулируем предложенную ранее оптимизационную задачу для графа  $G$ . Задача заключается в поиске кратчайшего замкнутого пути или цикла, который проходит через каждое ребро связного взвешенного неориентированного графа. Если граф имеет эйлеров путь, тогда этот путь служит оптимальным решением. В противном случае задачей оптимизации является поиск подмножества рёбер с минимальным возможным общим весом, так что после их добавления получающийся мультиграф имел эйлеров путь.

Решение поставленной оптимизационной задачи сведем к случайным блужданиям по графу  $G$ : при этом будем отдавать предпочтение ранее непосещенным ребрам, не являющимся мостами – то есть использовать алгоритм Флери.

Рассмотренная задача является задачей китайского почтальона [2] и имеет оптимальное решение.

## 3.2 Использование результатов кластеризации

Заметим, что если все аккорды рассматриваемого музыкального произведения являются уникальными, граф  $G$  будет представлять собой последовательно соединенные вершины  $v_1 \dots v_n$ , а решение оптимизационной задачи совпадать с оригинальным произведением.

Таким образом, для эффективного применения описанного алгоритма в задаче генерации музыки необходимо сократить количество уникальных расширенных аккордов.

Переформулируем предложенный алгоритм: вместо уникальных расширенных аккордов будем рассматривать их образы, полученные при кластеризации – это позволит усложнить структуру графа  $G$  за счет отображение некоторых вершин в одну.

Напомним предположение, в рамках которого описывается этот алгоритм: из всевозможных последовательностей аккордов  $v_1 \dots v_n$  гармоничными человеку кажутся музыкальные произведения, в которых сумма  $\sum_{i=1}^{n-1} \|v_{\sigma(i)} - v_{\sigma(i+1)}\|$  наименьшая для фиксированного набора  $v_1 \dots v_n$  и всевозможных перестановок  $\sigma$  на элементах  $1 \dots n$ . Приведем обещанное

теоретическое обоснование его разумности. Рассмотрим образы  $\{v_1 \dots v\}$  аккордов оригинального произведения, полученные после кластеризации. Построенная по ним последовательность также является решением задачи добавления ребер в граф  $G$  для существования эйлерова пути нем. Более того, такое решение оказывается близким к оптимальному на примере рассмотренных мной аудиофайлов. Если исходить из предположения о том, что изначальные музыкальные произведения звучат гармонично, то предположение соотносится с действительностью.

## 4 Вероятностный подход

### 4.1 Описание алгоритма

Идея генерации основывается на применении марковских цепей — последовательностей случайных событий с конечным или счётным числом исходов, характеризующаяся тем, что при фиксированном настоящем будущее независимо от прошлого. Процесс в каждый момент времени находится в одном из состояний. В рамках предложенной модели марковские цепи могут быть применены для построения очередного аккорда на основе одного или нескольких предыдущих. При фиксированном параметре  $D$ , отвечающем размерности марковской цепи, построим тензор  $T$  размерности  $D$  с размерами  $K \times \dots \times K$ , где  $K$  — количество уникальных расширенных аккордов среди всех произведений. Рассмотрим музыкальное произведение  $j$ , матрицей которого является  $V_j^+ \in \mathbb{Z}^{129 \times n_j}$ . Пусть  $f_j : \{1 \dots n_j\} \rightarrow \{1 \dots K\}$  — функция, сопоставляющая  $i$ -ому столбцу матрицы  $V_j^+$  музыкального произведения его порядковый номер  $f_j(i)$  среди уникальных расширенных аккордов. Далее из всех произведений выделим подпоследовательности  $v_i^+ \dots v_{i+D}^+$  аккордов длины  $D$ . Для каждой выделенной подпоследовательности увеличим значение  $T[f_j(i), \dots, f_j(i+D)]$  на 1. Теперь, если в генерируемом музыкальном произведении последние  $D-1$  расширенных аккордов имели порядковые номера  $i_1 \dots i_{D-1}$  среди уникальных расширенных аккордов, выберем очередной аккорд с вероятностями, равными весам компонент столбца  $T[i_1 \dots i_{D-1}]$  относительно суммы элементов этого столбца. Наконец, выберем случайным образом первый расширенный аккорд и длину музыкального произведения и повторим описанные шаги необходимое количество раз.

### 4.2 Использование результатов кластеризации

Для применения этого метода должно выполняться несколько условий:

- получившийся тензор не должен содержать большого числа столбцов  $T[i_1 \dots i_{D-1}]$  с

единственной ненулевой компонентой, чтобы генерируемые последовательности расширенных аккордов не повторяли в точности последовательности оригинальных произведений.

- ограничения по памяти должны позволять хранить получившийся тензор.
- ограничения по времени должны позволять производить необходимые вычислительные операции над тензором.

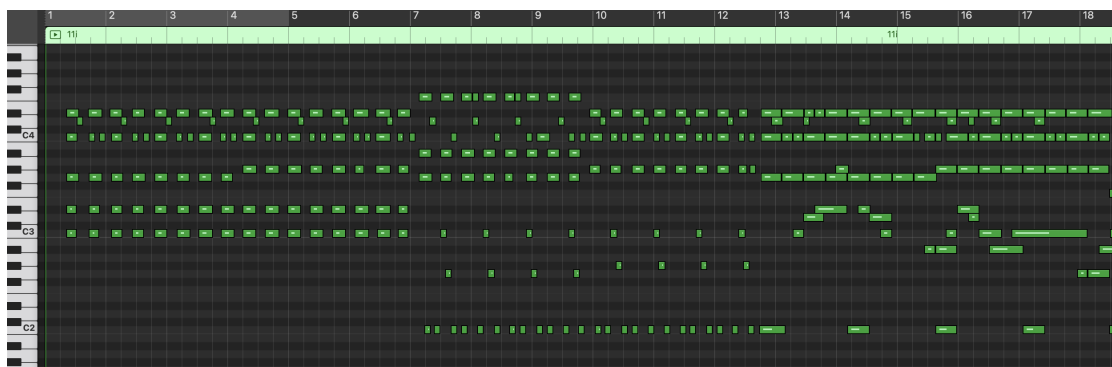
Таким образом, для эффективного применения марковских цепей в задаче генерации музыки необходимо сократить количество уникальных расширенных аккордов.

Переформулируем предложенный алгоритм: вместо уникальных расширенных аккордов будем рассматривать их образы, полученные при кластеризации – это позволит сократить параметр  $K$ , а значит сократить размер тензора  $T$ . Более того, в силу фиксированного количества рассматриваемых произведений сокращение размеров тензора приведет к увеличению плотности его значений, то есть все перечисленные условия будут выполнены.

## 5 Результаты

Я реализовал предложенные алгоритмы и разместил материалы в открытом GitHub репозитории<sup>2</sup>. Там же доступны примеры музыкальных произведений, к которым были применены описанные методы кластеризации и результаты работы алгоритмов. Рассмотрим некоторые из них. Визуализируем представления одного и того же музыкального произведения в midi-формате до и после кластеризации, для этого на временной шкале отложим используемые ноты и длительности их звучания.

- до кластеризации:



<sup>2</sup>GitHub репозиторий: [https://github.com/vvauijij/music\\_generation](https://github.com/vvauijij/music_generation), дата обр. 25.05.2023

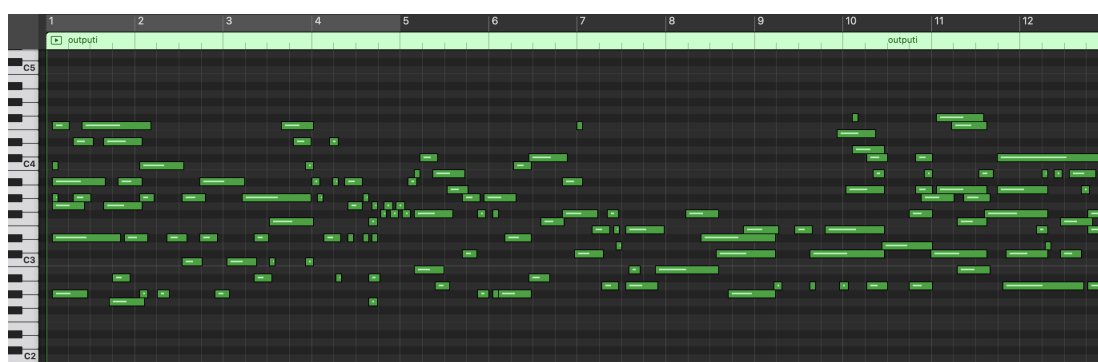
- после кластеризации:



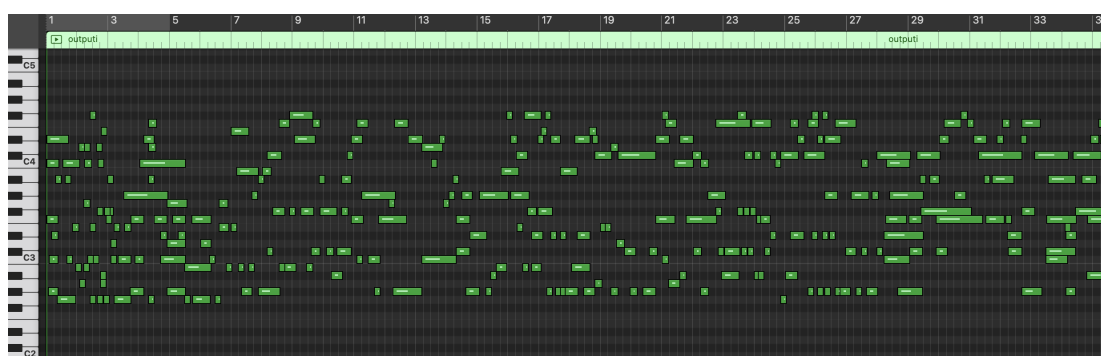
Количество различных используемых нот сократилось, их длительности изменились, однако общая структура произведения сохранилась.

Теперь рассмотрим несколько примеров генерируемых произведений, снова на временной шкале отложим используемые ноты и длительности их звучания:

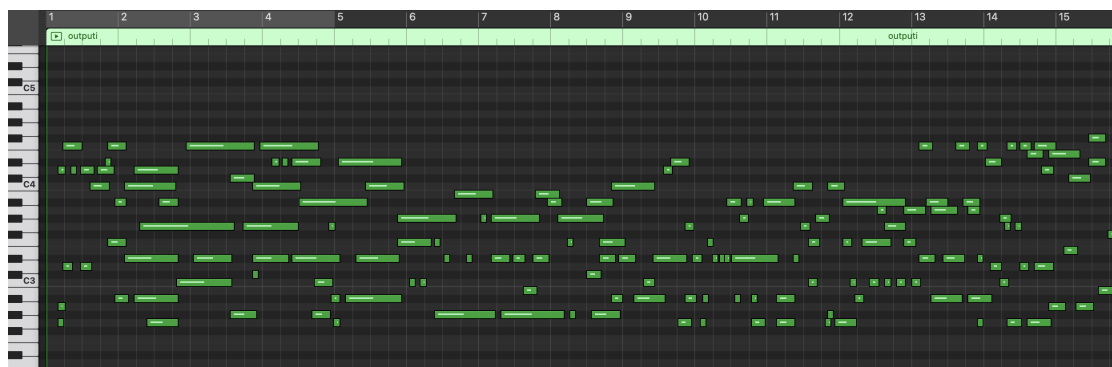
- пример №1:



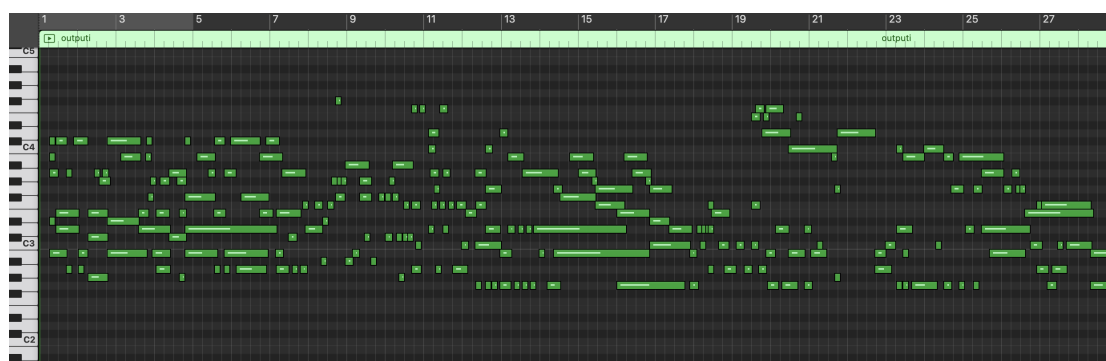
- пример №2:



- пример №3:



- пример №4:





## Список литературы

- [1] Spicher A. Bigo L. *HexaChord*. Режим доступа: свободный. URL: <https://louisbigo.com/hexachord> (дата обр. 27.11.2022).
- [2] Johnson E.L. Edmonds J. *Matching, Euler tours and the Chinese postman*. 1973. DOI: <https://doi.org/10.1007/BF01580113>.
- [3] Wen-Yi Hsiao и др. *Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs*. 2021. DOI: [10.48550/ARXIV.2101.02402](https://arxiv.org/abs/2101.02402). URL: <https://arxiv.org/abs/2101.02402>.
- [4] N. Kishore Kumar и Jan Shneider. *Literature survey on low rank approximation of matrices*. 2016. arXiv: [1606.06511](https://arxiv.org/abs/1606.06511) [math.NA].