Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет по преддипломной практике

Студент	П.С. Арабей
Руководитель	А.Н. Ткаченко
Консультант от кафедры ЭВМ	Н.Н. Иванов
Нормоконтролёр	А.С. Сидорович

МИНСК 2017

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

К ЗАЩИТЕ ДОПУСТИТЬ:
Зав. каф. ЭВМ
_____ Д.И. Самаль

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту
на тему
МОДУЛЬ ВЫДЕЛЕНИЯ ИНФОРМАЦИОННЫХ ОБРАЗОВ ИЗ МУЗЫКАЛЬНОГО ПРОИЗВЕДЕНИЯ

БГУИР ДП 1-40 02 01 01 004 ПЗ

Студент П.С. Арабей

Руководитель Н.Н. Иванов

Консультанты:

от кафедры ЭВМ Н.Н. Иванов

по экономической части И.В. Смирнов

Нормоконтролёр А.С. Сидорович

Рецензент

МИНСК 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

На сегодняшний день количество музыкальный треков в Интернете не поддаётся подсчёту. В одном только сервисе Яндекс.Музыка доступно 17 миллионов треков [1] . Сегодня выпускается тысячи песен каждый день [2]. Существует сотни жанров и поджанров музыки. Поэтому задача поиска новой музыки становится нетривиальной.

Для решение данной задачи используются рекомендательные сервисы, которые используют четыре подхода к анализу музыки для составления рекомендации:

1. Популярность трека. Использование данных о популярности трека и информации о пользователях. Этот способ доступен только большим онлайн сервисам. Качество рекомендации зависит от количества пользователей данного сервиса. Популярность треков определяется количеством и качеством отзывов и количеством покупок.

2. Метаинформация о треке. Это способ также доступен большим онлайн сервисам с большой библиотекой музыки. Качество рекомендации зависит от размеров библиотеки. Метаинформация о треке представляет собой тег с жанром, информация об исполнителе, альбом в который включён этот трек и т.д.

3. Семантика текста. Анализируя музыкальные блоги с применением технологий обработки естественных языков подход позволяет постоянно изучать веб и аналитически просматривать десятки миллионов страниц, имеющих отношения к музыке. Текст песни анализируется на основе взвешенных лексем. Произведения рекомендуется при близости его дескрипторов с дескрипторами пользователя.

4. Акустико-синтаксический анализ. На текущий момент нет методов достоверного распознавания музыкальных инструментов и музыкальных звуков. Однако, несмотря на это анализ сигнала играет очень важную роль используется в работе рекомендательных алгоритмов. Анализируются фрагменты произведения длительности от 200 мс до 4 с, в зависимости от вариабельности мелодии. Для каждого сегментов произведения определяется громкость, тембр, выявляются наиболее громко звучащие музыкальные инструменты; устанавливается к какой части композиции (припев, куплет и т. д.) относится этот сегмент.

В первом способе информационные признаки о треке получается на основе анализа поведения пользователя. Рекомендации при использовании данного метода делаются на основе коллаборативной фильтрации на основе соседей. Во втором методе является метаинформация о треке, а рекомендация делается на основе коллаборативной фильтрации на основе модели. В современных мультимедийных сервисах(youtube, lastfm, Яндекс.Музыка) используется гибридный подход, которых объединяет в себе подход осно-

ванный на соседстве и основанный на модели [3] [4].

Целью данного дипломного проекта является создание модуля, который выделял информационные образы из музыкального трека только на основании акустического анализа. Информационный образ - это признаковое описание трека как музыкального сигнала. Такой сервис должен быть лишён субъективности и обеспечить качество рекомендации.

В соответствии с поставленной целью были определены следующие задачи:

- выделение спектральных, временных и иных признаков из музыкального трека;
- проверка значимости признаков путём использовании их в задаче жанровой классификации(задача настройки классификатора решаться не будет);
- визуализация данных алгоритмом t-SNE.

1 ОБЗОР ЛИТЕРАТУРЫ

В части рассмотренных исследований решались смежные задачи, такие как жанровая классификация, идентификация музыкального трека, нахождение заимствований в музыкальном треке, распознавание звуковых источников. Для решения данных задач использовались разные подходы к выделения информационных образов из музыкальных треков. Среди рассмотренных приложений также стоит отметить разнообразие подходов к выделению информационных признаков. В данной дипломной работе стоит задача найти такие характерные информационные образы, которые позволили на основе их делать рекомендацию

1.1 Обзор методов выделения фрагментов из музыкальных произведений

В работе[5] информационные признаки, которые будут рассмотрены в следующем подразделе, рассчитываются по фрагментам музыкального трека в 1 секунду, состоящему из 40 непересекающихся окон в 25 миллисекунд. Для классификации используется 30 фрагментов. Полная сборка аудиоданных используемая в данной работе состоит из 15 жанров * 50 файлов * 30 секунд = 22500 секунд (то есть 6,25 часов аудио). Каждый фрагмент анализируется независимо. По каждому окну получают набор числовых? признаков. Их математическое ожидание и дисперсию используют в качестве признаков для фрагмента. В работе[6] используется такой же подход. Трёхсекундный фрагмент состоит из 1500 окон по 0,02 секунды. В отличии от работ[5] по автоматической жанровой классификации музыкального трека, в качестве признаков фрагмента используют среднее значение, дисперсию, коэффициент асимметрии, коэффициент эксцесса информационных признаков окон. Эти четыре значения затем используются для измерения подобия. В работе[7], где описывается принцип работы большинства аудиоидентификационных систем, используются фрагменты музыкального трека размером от 10 до 500 миллисекунд с перекрытием фрагментов от 50 до 90 процентов. В этой же работе описываются алгоритмы распознавания ремиксов.

Каждый музыкальный трек проходит передискретизацию до частоты 44100 Гц. Затем берется четырёхсекундный фрагмент.

Недостатками данных методов является потеря информации о структурах в музыкальном треке, которые не помещаются в один фрагмент.

1.2 Обзор методов представления музыкального трека в спектрально-временном виде

В работе[8] решается задача идентификации(опознавания) музыкального трека, а также поиск заимствований в жанре музыки хип-хоп. Для

получения спектрограммы используется оконное преобразования Фурье с окном Хемминга и размером окна 64 миллисекунды. После получения спектрограммы используется только амплитудная составляющая с последующим логарифмированием и применением высокочастотного фильтра к спектральной кривой. Эти шаги предпринимаются для того, чтобы сделать спектрограмму менее зависимой от абсолютного уровня и грубой спектральной формы. Также в работе рассмотрено constant Q преобразование[9]. Выделение информационных признаков в работе состоит из трёх стадий. Конечным результатом является коррелограмма. Первая стадия моделирует влияние звуковой волны на базилярную мембрану ушной улитки с помощью полосового фильтра, который реализуется четырьмя каскадными секциями фильтра второго порядка, которые реализуют фильтр восьмого порядка с импульсной характеристикой «gammatone» с центральной частотой в 9,3 кГц. Вторая стадия моделирует движение волосковых клеток. Так как клетки реагируют только на отклонение ресничек в одном направлении, и это вводит этап выпрямления в цепочке обработки сигналов. На низких частотах волосковые клетки имеют тенденцию срабатывать в определенной фазе сигнала - процесс, называемый блокировкой фазы. По мере того как частота входного сигнала увеличивается, блокировка фазы начинает заканчиваться примерно на 1,5 кГц и исчезает на 5 кГц, но при отсутствии блокировки тонкой структуры формы сигнала волосковые клетки фиксируются в амплитудной огибающей сигнала. Этот эффект моделируется посредством операции свертки с импульсным откликом фильтра типа «приподнятого-косинуса» [10] с T = 0,25 мс. Третья стадия это построение коррелограммы. Коррелограмма моделирует человеческое восприятия основного тона. Для этого используется автокорреляция. Измерением средней энергии в зависимости от задержки можно определить базовый период сигнала. После FFT-преобразования считается оконная автокорреляция для частотной области. Полученный трёхмерный массив данных(время, частота, высота тона) используется для получения информационных признаков для классификации музыкальных инструментов. В работе симулируется человеческое восприятие. Механизмы распознавания в человеческом мозге гораздо лучше приспособлены к сложным акустическим средам, чем любой искусственный механизм (рисунок 1.1).

В работе для извлечения информационных образов о ритме используется вейвлет Добеши четвёртого порядка.

1.3 Обзор методов выделения информационных признаков

В работе различается два вида признаков: спектральные и временные. Временные признаки - это значения, которые вычисляются из значений самих аудиоданных. Спектральные - на основе спектра полученного преобразованием Фурье.

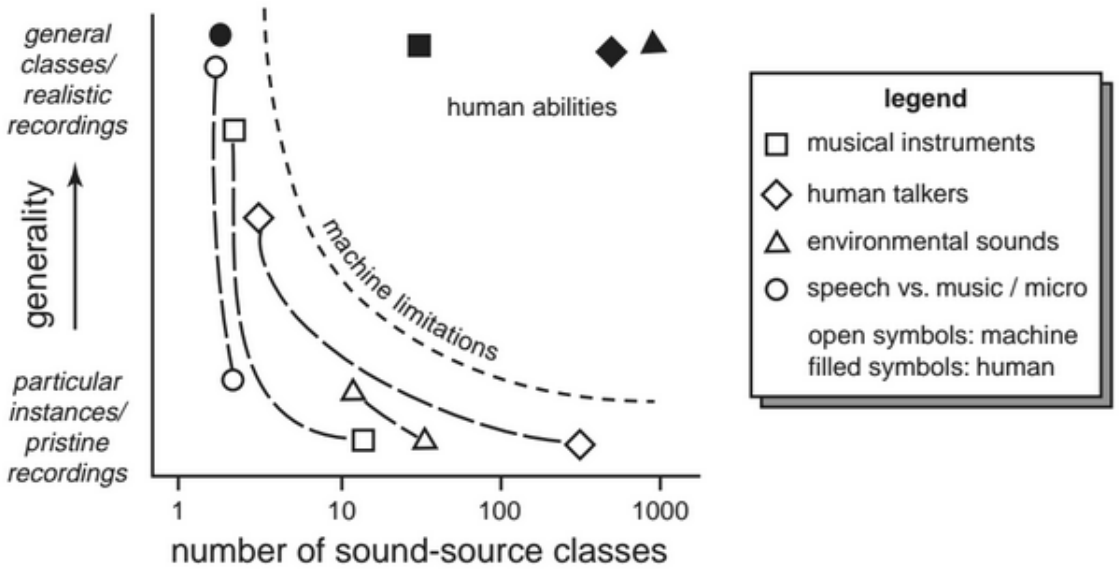

Рисунок 1.1 – График показывающий ограничения искусственных механизмов распознавания

Временные признаки:

1. Количество переходов сигнала через ноль.

$$ZeroCrossignRate = \frac{1}{N_S} \sum_{i=0}^{N_S-2} sgn(s_i) \oplus sgn(s_{i+1}) \tag{1.1}$$

где N_S - количество сэмплов в окне, $sgn()$ - функция которая возвращает 1 при положительном значении аргумента, и 0 при отрицательном. s - вектор, содержащий данные сигнала для одного окна

2. Автокорреляция первого порядка.

$$FirstOrderAutocorrelation = \frac{1}{N_S} \sum_{i=0}^{N_S-2} s_i * s_{i+1} \tag{1.2}$$

где N_S - количество сэмплов в окне, s - вектор, содержащий данные сигнала для одного окна.

3. Энергия сигнала.

$$Energy = \frac{1}{N_S} \sum_{i=0}^{N_S-1} (s_i)^2 \tag{1.3}$$

где N_S - количество сэмплов в окне, s - вектор, содержащий данные сигнала для одного окна.

Спектральные признаки:

1. Линейная регрессия спектра

$$\beta = \frac{N_B \sum_{i=1}^{N_B}(a_i F_i) - \sum_{i=1}^{N_B} a_i \sum_{i=1}^{N_B} F_i}{N_B \sum_{i=1}^{N_B}(F_i)^2 - \sum_{i=1}^{N_B}(F_i)^2} \tag{1.4}$$

где a - амплитуда, N_B - количество компонент разложения, а F - частота соответствующей амплитуды .
Это статистическое приближение использовано для нахождения наклона (β) спектра, который является мерой отношения высокочастотной составляющей к низкочастотной составляющей звука и, следовательно, тембра.

2. Среднее арифметическое взвешенное спектра.

$$SpectralCentroid = \frac{\sum_{i=1}^{N_B} a_i F_i}{\sum_{i=1}^{N_B} a_i} \tag{1.5}$$

Психоакустически это показатель воспринимаемой «яркости» звука, обеспечивающий лучшую оценку «яркого» звука, нежели тональность.

3. Гладкость спектра, как мера спектральной огибающей.

$$SSmoothness = \sum_{i=1}^{N_B} 20 \log a_i - \frac{20 \log a_{i-1} + 20 \log a_i + 20 \log a_{i+1}}{3} \tag{1.6}$$

где a - амплитуда, а N_B - количество компонент разложения. Белый шум, спектр которого имеет энергию на всех частотах будет иметь гладкость 1. Синусоидальный сигнал имеет только один пик в спектре и гладкость спектра будет равняться 0.

4. Дисперсия спектра относительно его среднего взвешенного. Оркестровые треки будут иметь большее значение, в отличии от сольных или монофонический треков. Вычисляется по следующей формуле:

$$SSpread = \sqrt{\frac{\sum_{i=1}^{N_B} a_i (F_i - SpectralCentroid)^2}{\sum_{i=1}^{N_B} a_i}} \tag{1.7}$$

где a - амплитуда, $SpectralCentroid$ - центроида спектра, N_B - количество компонент разложения, а F - частота соответствующей амплитуды .

5. Коэффициент асимметрии сигнала, является мерой того, насколько искажён спектр относительно его среднего взвешенного.

$$SDissymmetry = \sqrt{\frac{\sum_{i=1}^{N_B} a_i(F_i - SpectralCentroid)^3}{\sum_{i=1}^{N_B} a_i}} \tag{1.8}$$

В работе[5] исследованы алгоритмы автоматической жанровой классификации. Предложены набор признаков для представления «музыкальных аспектов» и ритмических структур аудиосигнала.

Термин «музыкальные аспекты» используются для обозначения особенности музыки, связанные с текстурой, тембром и музыкальными инструментами. В работе используется 4 статистических информационных образов, представляющие собой мат. ожидания и СКО следующих числовых признаков:

1. Среднее арифметическое взвешенное спектра, как мера спектральной яркости (см. формулу 1.5).

2. Энергетические спектральные окна по уровню 0,85.

$$\sum_{i=1}^{Rolloff} a_i = \sum_{i=1}^{N_B} a_i \tag{1.9}$$

где a - амплитуда, а N_B - количество компонент разложения.

3. Производная по частотам.

$$Flux = \|a_i - a_p\| \tag{1.10}$$

где a_i - амплитуда текущего окна, a_p - амплитуда прошлого окна.

4. Количества переходов через ноль сигнала, как показатель зашумленности сигнала (см. формулу 1.1).

Дополнительно выделяется процент окон, чья спектральная энергия меньше чем средняя спектральная энергия по фрагменту.

Также выделяются ритмические признаки. Вычисление признаков для представления ритмической структуры музыки основано на вейвлет преобразовании с вейвлетом Добеши. К результату вейвлет преобразования применяется автокорреляционная функция. Фиксируется первые пять пиков автокорреляционной функции и их соответствующие периодичности в ударах в минуту рассчитываются и добавляются в «ритмическую» гистограмму (см. рисунок 1.2). Этот процесс повторяется итерацией по сигналу и накоплением периодичности в гистограмме.

Пики гистограммы соответствуют различным периодичности звуко-

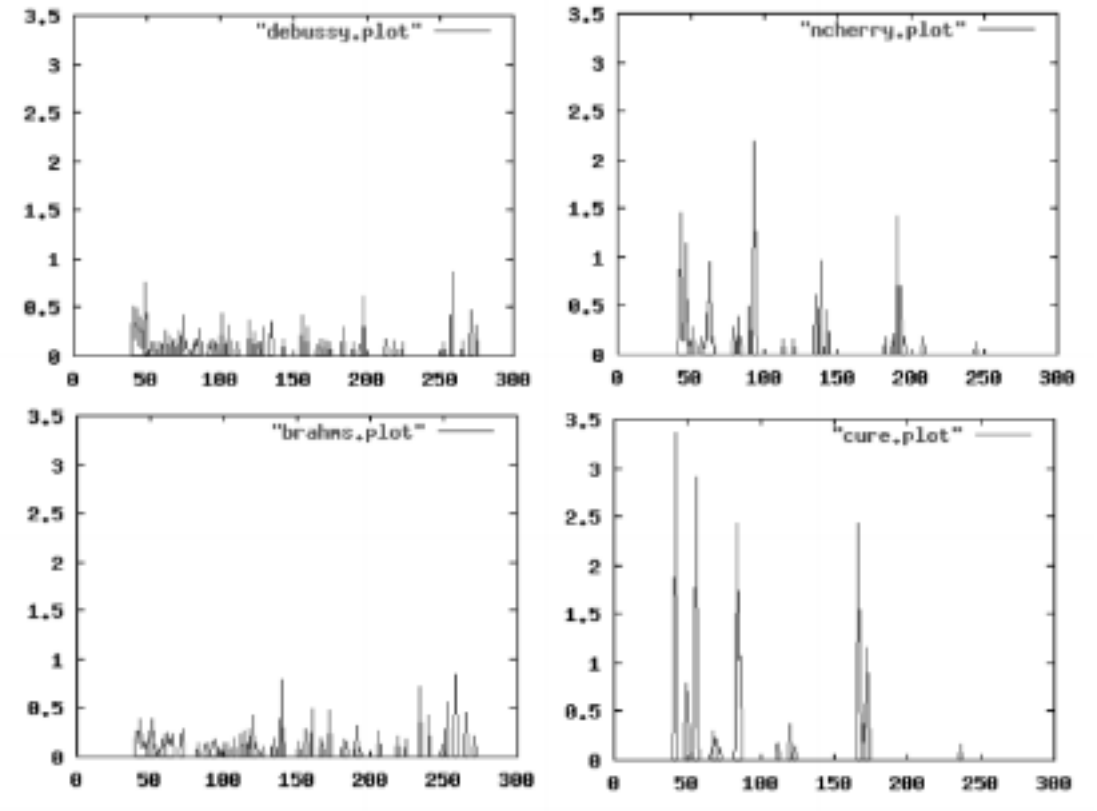

Рисунок 1.2 – Гистограмма ритма. Слева классическая музыка, справа поп-музыка

вого сигнала и используются в качестве основы для расчета признаков ритма. Используются следующие признаки, основанные на «ритмической» гистограмме :

1. Период - 0: Периодичность в ударах в минуту первого пика.
2. Амплитуда - 0: Относительная амплитуда (деленная на сумму амплитуд) первого пика.
3. Отношение периодичности - 1: отношение периодичности второго пика к периодичности первого пика.
4. Амплитуда - 1: Относительная амплитуда второго пика.
5. Отношение периодичности - 2: отношения периодичности третьего пика к периодичности второго пика.
6. Амплитуда - 2: Относительная амплитуда третьего пика.
7. Отношение периодичности - 3: отношения периодичности четвёртого пика к периодичности третьего пика.
8. Амплитуда - 3: Относительная амплитуда третьего пика.

8-мерный вектор признаков, используемый для представления ритмической структуры и силы, комбинируется с 9-мерным вектором музыкальной поверхности(4 статистических информационных образа * 2 парамет-

ра), чтобы сформировать 17-мерный вектор признаков, который используется для автоматической классификации музыкального жанра. В работе [6] вводят меру спектральной плоскостности и коэффициент амплитуды.

$$SFM_k = \frac{[\prod S_k^2(f)]^{\frac{1}{N}}}{\frac{1}{N}\sum_f S_k^2(f)} \tag{1.11}$$

$$SCF_K = \frac{max_k S_K^2(k)}{\frac{1}{N}\sum_k S_K^2(k)} \tag{1.12}$$

где S^2 - спектральная плотность мощности.

Чувственный эквивалент этих признаков можно охарактеризовать как шумоподобность и тоноподобие. Признаки с чувственным значением представляют характеристики звука, которые с большей вероятностью сохраняются в течении произведения и, следовательно, должны быть более надежными

1.4 Мел-кепстральные коэффициенты(MFCC)

Мел - психофизическая единица высоты звука, применяется главным образом в музыкальной акустике. Количественная оценка звука по высоте основана на статистической обработке большого числа данных о субъективном восприятии высоты звуковых тонов. Звуковые колебания частотой 1000 Гц при эффективном звуковом давлении $2 * 10^{-3}$ Па (то есть при уровне громкости 40 фон), воздействующие спереди на наблюдателя с нормальным слухом, вызывают у него восприятие высоты звука, оцениваемое по определению в 1000 мел. Звук частоты 20 Гц при уровне громкости 40 фон обладает по определению нулевой высотой (0 мел). Зависимость нелинейна, особенно при низких частотах (для «низких» звуков). Преобразовать значение частоты звука (Гц) в значение высоты (мел) можно по формуле:

$$m = 1127{,}01048 \ln(1 - \frac{f}{700}) \tag{1.13}$$

При обработке звука мел-частотный кепстр (MFC) представляет собой кратковременный спектр мощности звука, основанный на косинус - преобразовании Фурье на логарифмической спектральной мощности на нелинейной мел-шкале частот.

Мел-кепстральные коэффициенты (MFCC) - это коэффициенты, которые в совокупности образуют MFC. Различие между кепстром и мел-кепстром в том, что в MFC полосы частот равномерно распределены по шкале мела, что приближается к восприятию слуховой системы человека более близко, чем линейные интервалы частот, используемые в нормальном кепстре.

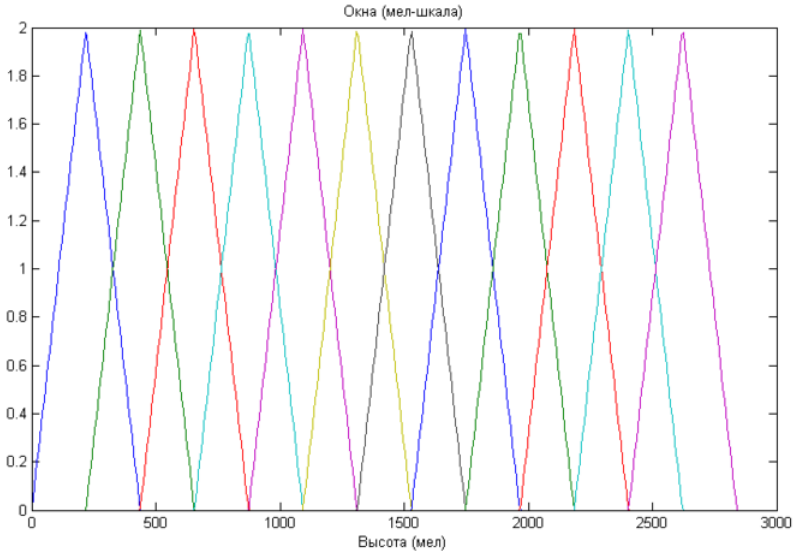

Рисунок 1.3 – Треугольные окна в мел-шкале

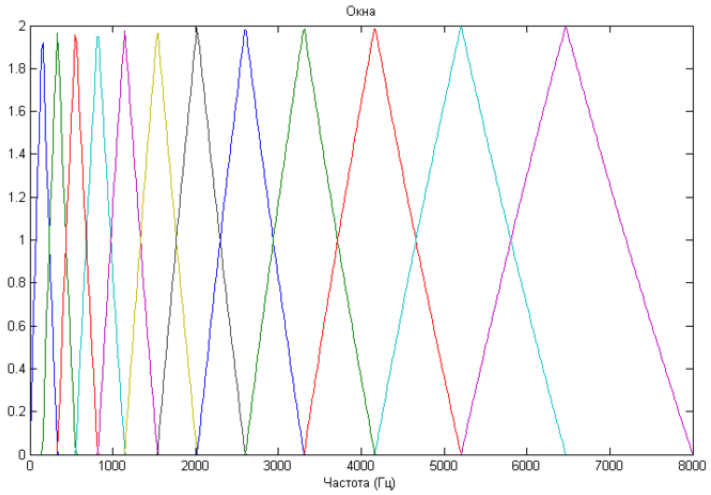

Рисунок 1.4 – треугольные окна в частотной шкале

MFCC обычно используются в качестве образов в системах распознавания речи, таких как системы, которые могут автоматически распознавать номера телефонов, произносимые в телефоне. MFCC также все чаще находят применение в приложениях поиска музыкальной информации, таких как классификация жанров[11], меры сходства звука[12] и т. д.

Для получения коэффициентов используют следующий алгоритм:

1. Используя преобразование Фурье получить спектр исходного сигнала.

2. Располагаем треугольные окна равномерно на мел-шкале (см. рисунок 1.3).

3. Переводим треугольные окна в частотную шкалу по формуле:

$$f = 700(e^{\frac{m}{1127}} - 1) \tag{1.14}$$

4. Делаем свёртку спектра с каждым окном (см. рисунок 1.4) и берём логарифм спектра можности.

5. Полученные значения переводим во временную область дискретным косинусным преобразованием.

6. MFCC представляют собой амплитуды результирующего спектра.

1.5 Обзор аналогов

1.5.1 HOLO

Данный проект - это приложение, которое по анализу музыкальной коллекции позволяет составлять плейлисты похожих на заданные образцы композиции, а также визуализировать полученные данные о музыкальной коллекции[13]. Метод получение информационных признаков основан на преобразовании Фурье с последующим получением евклидового расстояния до набора готовых спектров, с последующим построением матрицы переходов. Спектры носят служебную функцию опорных значений, и их форма выбрана по принципу отличаться друг от друга как можно сильнее, как по громкости, так и по корреляции графика частот. Формирование базы данных реализовано следующим образом:

1. Из файла извлекается фрагмент параметризуемой длины, начиная с 20% общей длины.

2. Фрагмент нарезается на параметризуемое количество окон с параметризуемым процентом перекрытия.

3. Каждое окно подвергается преобразованию Фурье, сглаживанию и очистке.

4. Имея некоторый набор заранее подготовленных центроидов, оценивается расстояние каждого окна до каждого из этих центроидов.

5. На основе расстояния окно маркируется порядковым номером ближайшей центроиды.

6. Строится матрица переходов между номерами центроид(рисунок 1.5).

7. Матрица переходов используется как вектор информационных признаков на основе которого происходит рекомендация.

Также в приложении реализовано визуализация результатов сканирования локальной музыкальной библиотеки.

Преимущества данного приложение:

- быстродействие;
- использует акустический анализ.

Минусы:

- нет взаимодействия с другими сервисами воспроизведения музыки;
- используется только 20% музыкального трека;
- анализируются только MP3-файлы 44,1кГц 16 бит;

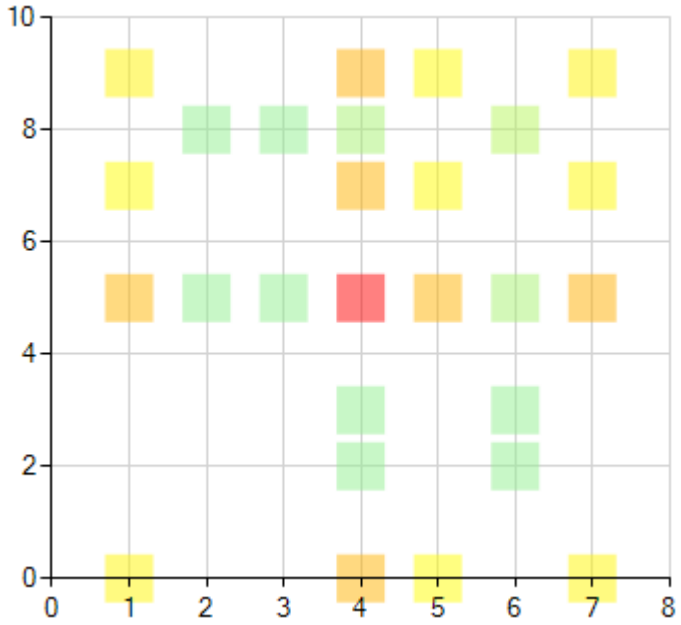

Рисунок 1.5 – Матрица переходов

1.5.2 Athena-NeuroPlay

Данный проект - это приложение, которое использует генетический алгоритм, который на основе оценки пользователя качества исходной выборки строит топологию нейронной сети прямого распространения с несколькими скрытыми слоями [14].

Первый этап - нормализация данных. Для инструментов выделяются необходимые частоты . Также принимается в расчёт и чувствительность слуха в зависимости от частоты. Таким образом задача нормализации сводится к выделению некоторой информации о частотах, которая показывает:

- как часто в композиции звучит звук из данного диапазона частот;
- как громко он звучал;
- как долго он звучал;
- для каждого определенного диапазона частот (нужно разбить весь «слышимый» спектр на определенное число диапазонов(см. рисунок 1.6)).

Для выделения частотной насыщенности в треке на каждом временном интервале используется FFT. Временной интервал имеет размер 1024 сэмплов. На основе спектра получают следующие признаки: насыщенность звука определенными спектрами, частота возникновения различных спектров звука, его громкость, его длительность. Затем данные нормализуются.

На втором этапе используется нейронная сеть прямого распространения с 1024 входными нейронами и 24*7 выходными, где каждый выходной нейрон показывает «качество» трека для того или иного времени суток. В качестве исходной выборки используется плейлист, который делится на две части: обучающая и контрольная. Размер обучающей и контрольной выборке устанавливается пользователем. При прослушивании трека из обучаю-

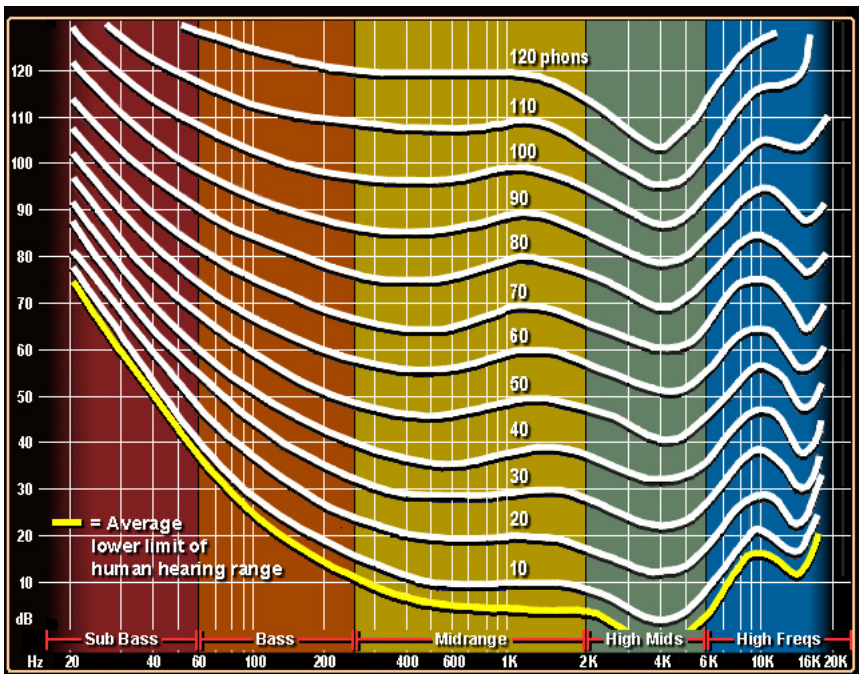

Рисунок 1.6 – Диапазоны частот

щей выборке пользователь ставит оценку от 0 до 1, где 0 - «плохой трек», а 1 - «хороший». Оценка влияет на все оценки по времени суток, но больше на ту, в которое время было поставлена оценка. После того как обучающая выборка промаркирована, идёт процесс обучения и настройки топологии нейронной сети. Топология нейронной сети определяется генетическим алгоритмом. Для этого берётся стартовая конфигурация в 10 слоев, в каждом по 100 нейронов, которая обучается за 1000 эпох, после определятся качество ее обучения. Качество обучения - это суммарная оценка контролирующей и обобщающей способности нейронной сети. На следующем этапе создаётся 10 конфигураций, каждая из которых является «мутантом» исходной, то есть у нее изменена в случайную сторону либо количество слоев, либо количество нейронов на каждом или некоторых слоях. Далее идёт обучение каждой конфигурации тем же способом, что и исходную. Выбирается лучшая из них по обобщающей способности и определяется как исходная. Данный процесс продолжаем до тех пор, пока не наступает такой момент, что мы не можем найти конфигурацию, которая обучается лучше чем исходная. Данную конфигурацию считаем лучшей, она оказалась способной лучше всех «запомнить» исходные данные и лучше всех предсказывает, то есть результат ее обучения наиболее качественный из возможных.

Из плюсов данного приложения стоит отметить:

- использует акустический анализ;
- запускается локально на компьютере пользователя.

Из недостатков:

- время работы;
- качество рекомендации.

1.5.3 Pandora Radio

Pandora (Пандора) - служба потокового воспроизведения музыки в Интернете, основанное на системе «Music Genome Project» [15]. Пользователь медиапроигрывателя Pandora выбирает музыкального исполнителя, после чего система ищет похожие композиции, используя около 400 музыкальных характеристик такие как жанр, тип инструментов, тип вокала, темп, синкопа, тональность, гармония и т. д. Используя функции «нравится» или «не нравится», слушатель может настроить «радиостанцию» по своему вкусу. В базе данных системы более миллиона композиций и более ста тысяч исполнителей[16]. Зарегистрированный пользователь может создать в своём профиле до 100 различных «радиостанций», транслирующих музыку в тех или иных жанрах. Медиапроигрыватель Pandora доступен пользователям с персональными компьютерами, смартфонами, планшетами с различными операционными системами.

Проект «Music Genome Project» - это набор более 450 атрибутов для описания песен и сложный математический алгоритм для их организации. Проект в настоящее время состоит из 5 суб-геномов : Pop / Rock, Hip-Hop / Electronica, Jazz, World Music и Classical. Песня представляется вектором, содержащим значения приблизительно для 450 «генов». Каждый ген соответствует характеристике музыки, например, пол ведущего вокалиста, уровень искажения на электрогитаре, тип фонового вокала и так далее. Рок и поп-песни имеют 150 генов, рэп-песни имеют 350 генов, а джазовые песни - приблизительно 400. Другие жанры музыки, такие как мировая и классическая музыка, имеют 300-450 генов. Система зависит от достаточного количества генов для получения полезных результатов. Учитывая вектор одной или нескольких песен, список других подобных песен построен с использованием того, что компания называет своим «алгоритмом сопоставления» [17]. Атрибуты для каждой песни выставляются музыкантом в процессе, который занимает от 20 до 30 минут на песню[18]. Десять процентов песен анализируются более чем одним музыкантом, чтобы обеспечить соответствие внутренним стандартам и статистическую надежность.

Преимущества данного сервиса:

- большая база данных музыки;
- имеются приложения для десктопов и мобильных платформ;
- быстродействие;
- взаимодействия с другими сервисами.

Из недостатков стоит отметить:

- сервис недоступен за пределами США, Австралии и Новой Зеландии;
- требуется регистрация;
- акустический анализ представлен не в чистом виде.

1.6 Выбор информационных образов для жанровой классификации музыкальных произведений

На основе анализа литературы были выбраны четыре типа информационных образов, которые будут извлекаться из музыкального трека:

- временной образ;
- спетральный образ;
- ритмический образ;
- мел-кепстральный образ.

Временные образ (здесь и далее представления музыкального трека как набора отсчётов будет называться сигналом).

1. Энергия сигнала, как мера яркости и громкости мелодии (см. формулу 1.3).

2. Количество переходов сигнала, через ноль, как мера зашумлённости (см. формулу 1.1).

3. Автокорреляция сигнала, как мера изменения резкости тембра (см. формулу 1.2.

Спектральный образ.

1. Среднее арифметическое взвешенное спектра. С точки зрения восприятия, оно имеет робастную связь с впечатлением «яркости» звука (см. формулу 1.5).

2. Линейная регрессия спектра, как мера отношения высокочастотной составляющей к низкочастотной составляющей звука и, следовательно, тембра (см. формулу 1.4).

3. Гладкость спектра, как мера гармоничности сигнала (см. формулу 1.6).

4. Дисперсия спектра относительно среднего взвешенного. С точки зрения восприятия, определяет «ширину» тембра (см. формулу 1.7).

5. Коэффициент асимметрии, как мера того, насколько искажён спектр относительно среднего взвешенного спектра, и, следовательно наклон к высоким или низким частотам (см. формулу 1.8).

6. Энтропия Винера или мера спектральной плоскостности. Определяет зашумлённость сигнала. Чем меньше значение, тем больше спектральной мощности сосредоточено в относительно небольшом числе полос (см. формулу 1.4).

7. Энергетическое спектральное окно по уровню 0,85, как мера спектральной формы (см. формулу 1.9)

Ритмический образ. Признаки считаются по коррелограмме.

1. Амплитуда первого пика.
2. Отношение частот первого пика к частоте второго пика.
3. Амплитуда второго пика.
4. Отношение частот второго пика к частоте третьего пика.
5. Амплитуда третьего пика.

6. Отношение частот третьего пика к частоте четвёртого пика.
7. Амплитуда четвёртого пика.
8. Частота первого пика.

Мел-кепстральный образ представляет из себя 16 мел-кепстральных коэффициентов.

2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

Изучив теоретические аспекты разрабатываемого модуля и выработав список требований необходимых для разработки модуля, разбиваем систему на компоненты. Компоненты в виде блоков и их взаимосвязи указаны на чертеже. В разрабатываемом модуле можно выделить следующие блоки:

- модуль чтения музыкального произведения;
- модуль препроцессинга и нарезки музыкального трека на фрагменты;
- модуль получения частотно-временного представления сигнала;
- модуль извлечения информационных образов;
- модуль обработки информационных образов;
- база данных информационных образов;
- модуль жанровой классификации музыкального произведения.
- модуль визуализация.

Структурная схема, иллюстрирующая перечисленные блоки и связи между ними приведена на чертеже ГУИР.400201.004 С1. Для решения задачи выделения спектральных, временных и иных признаков из музыкального трека будут использоваться с ледующие блоки:

- модуль препроцессинга и нарезки музыкального трека на фрагменты;
- модуль получения частотно-временного представления сигнала;
- модуль извлечения информационных образов;
- модуль обработки информационных образов.

Для решения задачи проверки значимости признаков путём использовании их в задаче жанровой классификации будет использоваться модуль жанровой классификации музыкального произведения.

Для решения задачи визуализации данных алгоритмом t-SNE используется модуль визуализации.

Модуль чтения музыкального произведения состоит из двух частей. Первая часть это консольная утилита Lame. Lame свободное приложение для кодирования аудио в формат MP3 (MPEG-1 audio layer 3) и декодирования аудио в WAV формат, который наиболее удобен для чтения и представления музыкальных треков в виде массива. Библиотека Scipy позволяет преобразовать WAV аудио в Numpy массив.

Модуль препроцессинга обеспечивает первоначальную обработку музыкальных треков. На вход модуля музыкальный трек подаётся как набор сэмплов (NumPy массив). В данном блоке происходит нормализация, фильтрация и, если необходимо, приведения стереозвука к монозвуку .

Модуль получения частотно-временного представления сигнала представляет собой набор преобразований такие как оконное преобразование Фурье и вейвлеты, которые представляются в виде многомерного NumPy - массива.Так подобные вычисления требуют высокой производительности,

то для более эффективного вычисления используются параллельные вычисления на центральном процессоре.

Модуль извлечения информационных образов. Данный модуль представляет из себя набор методов получения информационных образов из временной и частотной области музыкального произведения. Также в этом модуле считаются мел-кепстральные коэффициенты. На вход данному модулю подаётся многомерные Numpy массивы. На выходе получается одномерный Numpy массив информационных признаков для каждого музыкального трека.

Модуль обработки информационных образов. В данном модуле признаки нормализуются по МО и СКО. Удаляются выбросы и некорректные значения.

аза данных информационных образов хранит в себе результат работы всего модуля и хранит в себе набор информационных образов музыкальных треков и принадлежность к тому или иному кластеру.

При выборе базы данных были сформулированы следующие требования:

- производительность;
- объектный язык запросов;
- возможность параллельной записи и чтения.

Так как главный модуль является частью сервиса рекомендации музыки, то появляется дополнительные требования к базе данных:

- маштабируемость;
- репликация;
- балансировка нагрузки.

С учётом всех этих требований и того факта, что данные для хранения представляют собой простую структуру, выбор пал на базу данных MongoDB. MongoDB — документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных. Написана на языке C++. Имеется подробная и качественная документация, большое число примеров и драйверов под популярные языки Java, JavaScript, Node.js, C++, C#, PHP, Python, Perl, Ruby. MongoDB может работать с набором реплик. Набор реплик состоит из двух и более копий данных. Каждый экземпляр набора реплик может в любой момент выступать в роли основной или вспомогательной реплики. Все операции записи и чтения по умолчанию осуществляются с основной репликой. Вспомогательные реплики поддерживают в актуальном состоянии копии данных. В случае, когда основная реплика дает сбой, набор реплик проводит выбор, который из реплик должен стать основным. Второстепенные реплики могут дополнительно является источником для операций чтения. MongoDB масштабируется горизонтально используя шардинг. Пользователь

выбирает ключ шарда, который определяет как данные в коллекции будут распределены. Данные разделятюся на диапазоны (в зависимости от ключа шарда) и распределятся по шардам. Из преимуществ MongoDB:

1. Объектный язык запросов.
2. Поддержка индексации.
3. Поддержка Map/Reduce для распределенных операций над данным.
4. Документы, не требующие определения схемы. Одно из самых важных преимуществ. Преимущество заключается в том, что нет нужды хранить пустые ячейки данных в каждом документе.
5. Поддержка сложных массивов. Каждый элемент массива может представлять из себя объект.
6. Поддержка шардинга на уровне платформы.
7. Атомарность гарантируется только на уровне целого документа, то есть частичного обновления документа произойти не может.
8. Любые данные, которые считываются одним клиентом, могут параллельно изменяться другим клиентом.

СУБД управляет наборами JSON-подобных документов, хранимых в двоичном виде в формате BSON. Хранение и поиск файлов в MongoDB происходит благодаря вызовам протокола GridFS.

Модуль жанровой классификации необходим для проверки значимости выделенных образов. Для этого используется набор стандартных алгоритмов классификации из библиотеки Scikit-learn с параметрами по умолчанию. На вход модуля поступает двумерный Numpy-массив. На выходе матрица ошибок. В качестве исходной выборки используется репозиторий музыки GZTAN.

Модуль визуализации визуализирует данные с помощью алгоритма t-SNE. На вход подаётся двумерны Numpy-массив. На выходе изображение в формате JPEG.

Для реализации модулей был выбран язык программирования Python, так как для него существует множество библиотек выполняющих математические расчеты, и облегчающих решение задач связанных с анализом данных и машинным обучением. В работе используются библиотека NumPy - это расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами. Также существует библиотека SciKit-learn, которая содержит реализацию алгоритмов машинного обучения инструменты для работы с данными. В дипломной работе также используется библиотека Skipy, которая предназначенная для выполнения научных и инженерных расчётов.

3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

3.1 Алгоритм работы модуля выделения информационных образов из музыкального произведения

На ход модуля чтения музыкального произведения подаётся путь к музыкальному произведению. Файл с расширением ".wav" считывается в оперативную память в формате массива отсчётов. Файл с иным расширением конвертируются в WAV формат.

В модуле препроцессинга и нарезки от музыкального трека отрезается 10% длины с начала и конца трека. В случае, если трек является многоканальным, то он приводится к одноканальному посредством чередованием правого и левого канала. Данные нормализуются по МО и СКО. Затем используется экспоненциальное сглаживание с коэффициентом сглаживания 0,99. Результат нарезается на фрагменты по 5 секунд с перекрытием в 0,5 секунды.

В модуле получения частотно-временного представления используется оконное преобразование Фурье с окном Хэмминга шириной в 10 миллисекунд. Также в модуле для получения ритмического образа используется вейвлет Добеши каскадным алгоритмом с количеством каскадов 4. Результат каждого каскада сглаживается экспоненциальным сглаживанием с коэффициентом сглаживания 0,97, передискретизируется с уменьшением дискретизации в 16 раз, нормализуется по МО. Затем результаты поэлементно складываются и вычисляется автокорелляция. Результатом работы модуля является спектрограмма и коррелограмма.

В модуле выделения информационных образов из временного представления сигнала выделяется временной, из спектрограммы - спектральный образ по каждому срезу спектрограммы, из коррелограммы - ритмический. Мел-кепстральные коэффициенты считаются по окнам в 5 миллисекунд.

В модуле обработки информационных образов по всем образам кроме временного считается МО, СКО, коэффициент ассиметрии, коэффициент эксцесса.

В базу данных записывается информационный образ в формате: название произведения, номер фрагмента и результат работы модуля обработки информационных образов.

В модуле классификации информационных образов используются набор стандартных алгоритмов классификации из библиотеки scikit-learn с параметрами по умолчанию:

а) метод ближайших соседей с k = 3;
б) метод опорных векторов с полиномиальным ядром;
в) дерево принятия решений;
г) случайный лес;

д) нейронная сеть прямого распространения с 1000 нейронов на скрытом слое;

е) adaBoost;

ж) наивный баесовский классификатор;

з) квадратичный дискриминант.

Для оценки качества классификации использовался скользящий контроль количеством разбиений равным 10 и алгоритмом разбиения stratified k-fold. Также для каждого классификатора строится матрица ошибок.

В модуле визуализации используется алгоритм t-sne.Это нелинейный метод уменьшения размерности, который особенно хорошо подходит для вложения высокоразмерных данных в пространство двух или трех измерений, которое затем можно визуализировать на диаграмме рассеяния. В частности, он моделирует каждый высокоразмерный объект с помощью двух- или трехмерной точки таким образом, что аналогичные объекты моделируются соседними точками, а разнородные объекты моделируются удаленными точками. Данные визуализируются в 2д и 3д графики.

3.2 Реализация внутренних модулей

Рассмотрим в деталях функциональные части системы: для этого произведем детальный анализ компонентов, модулей, составляющих их классов и отдельных методов, реализующих логику программы.

Структурно система подразделяется на ряд модулей , объединенных под общим модулем с названием `MainModule`. Дальнейшее разделение ведется по следующим модулям:

1. Модуль чтения музыкального произведения - `WavModule`.
2. Модуль препроцессинга и нарезки - `PreprocessingModule`.
3. Модуль получения частотно-временного представления сигнала - `SpectralTransformerModule`.
4. Модуль извлечения информационных образов - `FeatureExtractorModule`.
5. Модуль обработки информационных образов - `FeatureProcessingModule`.
6. База данных информационных образов - `DatabaseModule`.
7. Модуль жанровой классификации музыкального произведения - `GenreClassificationModule`.
8. Модуль визуализации - `VisualizeDataModule`.

3.2.1 Класс `WavModule`

Класс `WavModule` предназначен для преобразования MP3 к формату WAV и считывании данных в Numpy-массив.

Класс `WavModule` имеет следующие методы:

• `create_wav(file_name)` процедура, которая принимает на вход путь к MP3-файлу и с помошью стандартной библиотеки Python вызывает либо bash-скрипт в случае запуска в операционной системе Linux, либо batch-скрипт в случае запуска в операционной системе Windows, который использует консольное приложение Lame для преобразования MP3-файла в WAV формат

• `read_wav(filename, label)` - метод, который считывает файл формата WAV и сохраняет данные в класс `Track`

3.2.2 Класс `PreprocessingModule`

Класс `PreprocessingModule` предназначен для первичной обработки трека, которая включает в себя приведения стерео звука к моно, нормализация по МО и СКО, удаления заданного процента трека с начала и с конца, экспоненциальное сглаживание и нарезка трека на фрагменты. В конструкторе задаётся коэффициент сглаживания, процент пересечения фрагментов, процент отсечения трека с начала, процент отсечения трека с конца, размер фрагмента в секундах.

Класс `PreprocessingModule` имеет следующие методы и поля:

• Поле `alpha` – коэффициент сглаживания, который может принимать значения от 0 до 1.

• Поле `overlap` – процент пересечения фрагментов.

• Поле `cut_start` – процент отсечения трека с начала трека

• Поле `cut_end` – процент отсечения трека с конца трека

• Поле `frame_size_sec` – размер фрагментов в секундах

• `stereo_to_mono(track)` – метод, который преобразует стерео звук к моно с помощью чередования правого и левого канала

• `scale(track)` – метод, который нормализует данные по МО и СКО

• `filter(track)` – метод, который преобразовывает данные искользуя экспоненциальное сглаживание:

$$s_t = \begin{cases} c_1, & t_1 = 0. \\ s_t + \alpha * (c_t - s_{t-1}), & t > 0 \end{cases} \tag{3.1}$$

где - α - коэффициент сглаживание заданные в поле `alpha`

• `cutting(track)` – метод, который отсекает процент данных трека с начала и конца, который задаётся полями `cut_start` и `cut_end`

• `framing(track)` – метод, который нарезает трека на фрагменты длинной заданной полем `frame_size_sec` и перекрытием заданным полем `overlap`.

3.2.3 Класс `SpectralTransformer`

Класс `SpectralTransformer` предназначен для получения спектрально-временного представления трека, а также для получения коррелограммы. Спектрально-временное представление получается путём оконного преобразования Фурье (см. формулу 3.2) с окном Хемминга размером 10 миллисекунд (см. формулу 3.3)

$$F(m,\omega) = \sum_{n=-\infty}^{\infty} f[n]w[n-m]e^{-j\omega n} \tag{3.2}$$

$$w_i = 0.54 - 0.46\cos\frac{2\pi i}{n-1} \tag{3.3}$$

Также в модуле получает ритмические(перкуссионные) признаки.

- Поле `alpha` – коэффициент сглаживания, который может принимать значения от 0 до 1.
- Поле `window` – массив, который содержит в себе окно Хемминга.
- Поле `level` – количество каскадов в
- Поле `rate` – частота передискретизации.
- `short_time_fourier(track)` – метод, который преобразует трек оконным преобразованием Фурье с окном Хемминга, возвращает двумерный массив амплитуд типа `ndarray`
- `wavelet_daubechies(data)` – метод, который делает вейвлет преобразования Добеши.
- `filter(track)` – метод, который преобразовывает данные искользуя экспоненциальное сглаживание.
- `resampling(data)` – метод, который передискретизирует данные в `rate` раз
- `normalize_and_sum(track)` – метод, который

4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

5 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

7 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ МОДУЛЯ ВЫДЕЛЕНИЯ ИНФОРМАЦИОННЫХ ОБРАЗОВ ИЗ МУЗЫКАЛЬНОГО ПРОИЗВЕДЕНИЯ

7.1 Описание функций, назначения и потенциальных пользователей ПО

Модуль выделения информационных признаков из музыкального произведения предназначен для получения спектральных, временных и спектрально-временных признаков из музыкального произведения. Модуль может быть использован в сервисах связанных с музыкой. Также может быть использован аналитиками данных в музыкальной сфере. Программный продукт относится к программному обеспечению по индивидуальному заказу для использования внутри организации-заказчика.

7.2 Расчёт затрат на разработку ПО

Для осуществления упрощённого расчёта затрат на разработку ПО следует произвести расчёт следующих статей:

- затраты на основную заработную плату разработчиков;
- затраты на дополнительную заработную плату разработчиков;
- отчисления на социальные нужды;
- прочие затраты (амортизация оборудования, расходы на электроэнергию, командировочные расходы, накладные расходы и т.п.).

Расчёт затрат на основную заработную плату разработчиков 7.1 осуществляется на численности и состава команды, размеров месячной заработной платы каждого из участников команды, а также общей трудоёмкости разработки программного обеспечения.

Основная заработная плата исполнителей на конкретное программное средство определяется по формуле:

$$З_о = \sum_{i=1}^{n} Т_{чi} \cdot t_i , \tag{7.1}$$

где n — количество исполнителей, занятых разработкой конкретного ПО;

$Т_{чi}$ — часовая тарифная ставка i-го исполнителя, руб;

t_i — трудоемкость работ, выполняемых i-го исполнителем, час.

В проекте занято три человека: руководитель, разработчик и специалист по анализу данных.

Часовая заработная плата определяется на основе месячной заработной планы путём деления на количество рабочих часов в месяце (примем

166 часов). Для руководителя, при заработной плате равной 600 рублей, часовая заработная плата равна 3,61 рубля. Для разработчика, при заработной плате равно 500 рублей, часовая заработная плата равна 3,01 рубля. Для специалиста по анализу данных при зарплате 14700 рублей, часовая заработная плата равна 88,55 рубля Трудоемкость определяется исходя из сложности разработки программного продукта и объема выполняемых им функций. В нашем случае она составляет 90 дней или 720 часов. Тогда основная зарплата исполнителей равна:

$$З_о = (3{,}01 + 3{,}61 + 88{,}55) \times 720 = 68\,522{,}4 \text{ руб.} \tag{7.2}$$

Таблица 7.1 – Расчет затрат на основную заработную плату команды

№	Участник команды	Выполняемые работы	Месячная заработная плата, р	Часовая заработная плата, р.	Трудоемкость работ, ч.	Основная заработная плата, р.
1	Руководитель проекта	Контроль, помощь	600	3,61	720	2599,2
2	Программист 1-й категории	Разработка	500	3,01	720	2167,2
3	Специалист по анализу данных	Разработка	14 700	88,55	720	63 756
ПРЕМИЯ (50%)						34 261,2
Итого затраты на основную заработную плату разработчиков						102 783,6

Затраты на дополнительную заработную плату команды разработчиков включает выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяется по формуле:

$$З_д = \frac{З_о \cdot Н_д}{100}, \tag{7.3}$$

где $З_о$ — затраты на основную заработную плату с учетом премии, руб;
$Н_д$ — норматив дополнительной заработной платы, 15 %.

В результате подстановки получим:

$$З_д = \frac{102\,783{,}6 \times 15}{100} = 15\,417{,}52 \text{ руб.} \tag{7.4}$$

Отчисления на социальные нужды (в фонд социальной защиты населения и на обязательное страхование) определяются в соответствии с действующими законодательными актами по формуле:

$$З_{соц} = \frac{З_о + З_д \cdot Н_{соц}}{100}, \tag{7.5}$$

где $Н_{соц}$ — норматив отчислений на социальные нужды 34% и обязательное страхование, $0{,}6$ % .

$$З_д = \frac{(102\,783{,}6 + 15\,417{,}52) \times 34}{100} = 40\,188{,}38 \text{ руб.} \tag{7.6}$$

Расчет прочих затрат осуществляется в процентах от затрат на основную заработную плату команды разработчиков с учетом премии по формуле:

$$З_{пз} = \frac{З_о \cdot Н_{пз}}{100}, \tag{7.7}$$

где $Н_{пз}$ — норматив прочих затрат, 100 % .

$$З_{пз} = \frac{102\,783{,}6 \times 100}{100} = 102\,783{,}6 \text{ руб.} \tag{7.8}$$

Полная сумма затрат на разработку программного обеспечения находится путем суммирования всех рассчитанных статей затрат (7.2).

Таблица 7.2 – Затраты на разработку программного обеспечения

Статья затрат	Сумма, руб.
Основная заработная плата команды разработчиков	102 783,6
Дополнительная заработная плата команды разработчиков	15 417,52
Отчисления на социальные нужды	40 188,38
Прочие затраты	102 783,6
Общая сумма затрат на разработку	261 173,1

7.3 Оценка результата (эффекта) от продажи ПО

Экономический эффект представляет собой прирост чистой прибыли, полученный организацией в результате использования разработанного ПО. Как правило, он может быть достигнут за счет:

• уменьшения (экономии) затрат на заработную плату за счет замены «ручных» операций и бизнес-процессов информационной системой;

• ускорения скорости обслуживания клиентов и рост возможности обслуживания большего их количества в единицу времени, т.е. рост производительности труда;

• появления нового канала сбыта продукции или получения заказов (как в случае внедрения интернет-магазина);

• и т.п.

Экономический эффект организации-разработчика программного обеспечения в данном случае заключается в получении прибыли от его продажи множеству потребителей. Прибыль от реализации в данном случае напрямую зависит от объемов продаж, цены реализации и затрат на разработку данного ПО.

Таким образом, необходимо сделать обоснование предполагаемого объема продаж – количества копий (лицензий) программного обеспечения, которое будет куплено клиентами за год (N). Принимая во внимания примерное количество скачиваний аналогичных приложений в год (2 тыс. человек) и учитывая различия в количестве функций, можно спрогнозировать 200 скачиваний за год.

Далее следует определить цену на одну копию (лицензию) ПО. Цена формируется на рынке под воздействием спроса и предложения. Тогда расчет прибыли от продажи одной копии (лицензии) ПО осуществляется по формуле:

$$П_{ед} = Ц - НДС - \frac{З_о \cdot Н_{пз}}{100}, \qquad (7.9)$$

где Ц — цена реализации одной копии (лицензии) ПО, руб.;
$З_р$ — сумма расходов на разработку и реализацию, руб.;
N — цена реализации одной копии (лицензии) ПО, руб.;
$П_{ед}$ — прибыль, получаемая организацией-разработчиком от реализации одной копии программного продукта, руб.;
НДС — цена реализации одной копии (лицензии) ПО, руб.;

Цена одной копии (лицензии) программного обеспечения выбирается на основе цены на аналогичное программное обеспечение на рынке. Для приложений данного направления, средняя стоимость составляет 1500 руб.

Сумма налога на добавленную стоимость рассчитывается по формуле:

$$НДС = \frac{Ц \cdot НДС}{100 + НДС}, \qquad (7.10)$$

где НДС — ставка налога на добавленную стоимость, 20 %.

$$НДС = \frac{200 \times 20}{20 + 100} = 33{,}33 \text{ руб.} \tag{7.11}$$

Подставив значения в формулу 6.9, получим:

$$П_{ед} = 200 - 33{,}33 - \frac{261\,173{,}1}{200} = 160{,}8 \text{ руб.} \tag{7.12}$$

Суммарная годовая прибыль по проекту в целом будет равна:

$$П = П_{ед} \cdot N\,. \tag{7.13}$$

$$П = 160{,}8 \times 200 = 32\,160 \text{ руб.} \tag{7.14}$$

Рентабельность затрат на разработку ПО рассчитывается по следующей формуле:

$$Р = 100 \times \frac{32\,160}{261\,173{,}1} \text{ руб.} \tag{7.15}$$

Проект является экономически эффективным, т.к. рентабельность затрат на разработку программного обеспечения превышает среднюю процентную ставку по банковским депозитным вкладам.

Чистая прибыль рассчитывается по формуле:

$$ЧП = П - \frac{П \cdot Н_{приб}}{100}\,, \tag{7.16}$$

где $Н_{приб}$ — ставка налога на прибыль, $18\ \%$.

$$ЧП = 32\,160 - \frac{32\,160 \cdot 18}{100} = 26\,371{,}2 \text{ руб.} \tag{7.17}$$

7.4 Расчёт показателей эффективности инвестиций в разработку ПО

Подставив значения, получим:

Перед тем, как произвести расчёт показателей эффективности инвестиций в разработку ПО, необходимо сравнить размер инвестиций в разработку и получаемый годовой экономический эффект. Т.к. сумма инвестиций меньше, чем сумма годового эффекта, инвестиции окупятся менее чем за год. В таком случае, необходимо рассчитать лишь рентабельность инвестиций:

$$Р_и = \frac{П_ч}{С_п} \cdot 100\%\,, \tag{7.18}$$

Подставив значения, получим:

$$\text{Р}_{\text{и}} = \frac{26\,371{,}2}{261\,173{,}1} \cdot 100\% = 10{,}10\%. \tag{7.19}$$

ЗАКЛЮЧЕНИЕ

В результате преддипломной практики были проведены исследования в области выделения информационных образов и сервисов рекомендации. Было проведено исследование существующих способов выделения информационных признаков при решении задач: распознования неперкуссионных инструментов, распознования заимствований и классификации жанров. Были рассмотрены существующие сервисы и приложения рекомендации музыки на основе акустического анализа, рассмотрены методы и работы, а также из положительные и отрицательные стороны. Также была разработана архитектура модуля выделения информационных образов из музыкальных произведений, и описана его структура.

В процессе реализации данного проекта в рамках дипломного проектирования будет создан конечный продукт реализующий данную архитектуру.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Yandex N. V. Десять миллионов треков на Яндекс.Музыке [Электронный ресурс]. — Электронные данные. — Режим доступа: https://yandex.ru/blog/company/69072. — Дата доступа: 05.02.2017.

2 Kevin Murnane. The US Music Industry Crossed A Threshold In 2016 [Электронный ресурс]. — Электронные данные. — Режим доступа: https://www.forbes.com/sites/kevinmurnane/2017/01/18/the-us-music-industry-passed-a-milestone-in-2016/#312060b15a90. — Дата доступа: 05.02.2017.

3 Yandex N. V. Как это работает? Рекомендации в Яндекс.Музыке [Электронный ресурс]. — Электронные данные. — Режим доступа: https://yandex.ru/blog/company/92883. — Дата доступа: 05.02.2017.

4 M. Jones. Introduction to approaches and algorithms [Электронный ресурс]. — Электронные данные. — Режим доступа: https://www.ibm.com/developerworks/opensource/library/os-recommender1/index.html?S_TACT=105AGX99&S_CMP=CP. — Дата доступа: 05.02.2017.

5 George Tzanetakis Georg Essl, Perry Cook. Automatic Musical Genre Classification Of Audio Signals. — 2001.

6 Bashi, Jamil George. Music Similarity Measures for Interpolative Playlist Generation. — 2008.

7 Balen, Jan Van. Automatic Recognition of Samples in Musical Audio. — 2011.

8 Martin, Keith Dana. Sound-Source Recognition: A Theory and Computational Model. — 1999.

9 Brown, Judith C. Calculation of a constant Q spectral transform. — 1991.

10 Ian Glover, Peter Grant. Digital Communications. — 1998.

11 Mohit Rajani, Luke Ekkizogloy. Supervised Learning in Genre Classification. — 2009.

12 Logan, Beth. Mel Frequency Cepstral Coefficient for Music Modeling. — 2000.

13 HOLO — Система анализа музыки [Электронный ресурс]. — Электронные данные. — Режим доступа: https://habrahabr.ru/post/194724/. — Дата доступа: 05.02.2017.

14 Формирование музыкальных предпочтений у нейронной сети — эксперимент по созданию умного плеера [Электронный ресурс]. — Элекстонные данные. — Режим доступа: https://habrahabr.ru/post/263811/. — Дата доступа: 05.02.2017.

15 About Pandora [Электронный ресурс]. — Электронные данные. — Режим доступа: http://www.pandora.com/about. — Дата доступа: 05.02.2017.

16 The Best Music Services Compared [Электронный ресурс]. — Электронные данные. — Режим доступа:

http://www.techlicious.com/guide/best-music-service-best-for-you/. — Дата доступа: 05.02.2017.

17 Consumer item matching method and system US 7003515 B1 [Электронный ресурс]. — Электронные данные. — Режим доступа: http://www.google.com/patents/US7003515?dq=7,003,515. — Дата доступа: 05.02.2017.

18 Tim Westergren (Music Genome Project Founder) [Электронный ресурс]. — Электронные данные. — Режим доступа: http://www.tinymixtapes.com/features/tim-westergren-music-genome-project-founder. — Дата доступа: 05.02.2017.

ПРИЛОЖЕНИЕ А
(*обязательное)*

Вводный плакат

ПРИЛОЖЕНИЕ Б
(*обязательное)*

Схема структурная