

УРАЛЬСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ ПЕРВОГО ПРЕЗИДЕНТА РОССИИ Б.Н. ЕЛЬЦИНА

На правах рукописи
УДК 004.93

ГЛАЗЫРИН НИКОЛАЙ ЮРЬЕВИЧ

**ОБ АЛГОРИТМИЧЕСКОМ РАСПОЗНАВАНИИ АККОРДОВ В
ЦИФРОВОМ ЗВУКЕ**

Специальность 05.13.18 —
«Математическое моделирование, численные методы и комплексы программ»

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
уч. степень, уч. звание
Волков М.В.

Екатеринбург – 2014

Содержание

Введение	4
1 Необходимые теоретические сведения	10
1.1 Звук	10
1.2 Свойства звука	11
1.3 Основные понятия из теории музыки	12
1.4 Цифровой звук	15
1.5 Свойства музыкальных звукозаписей	15
1.6 Формализация задачи	16
1.6.1 Частотно-временное представление	16
1.6.2 Классификация	17
2 Обзор литературы	18
2.1 Предварительная обработка	18
2.2 Спектрограмма	19
2.3 Векторы признаков	21
2.4 Классификация векторов признаков	24
2.4.1 Метод ближайшего соседа	24
2.4.2 Скрытые марковские модели и байесовские сети	25
2.4.3 Другие модели	27
2.5 Выводы	28
3 Распознавание аккордов без использования машинного обучения	29
3.1 Частотно-временное представление звукозаписи	29
3.1.1 Определение частоты настройки музыкальных инструментов	29
3.1.2 Определение ритма	30
3.1.3 Снижение влияния ударных инструментов	30
3.1.4 Получение спектра	31
3.2 Выделение мелодических компонент спектра и векторы признаков	33
3.3 Применение самоподобия	35
3.4 Классификация и исправление ошибок	36
3.4.1 Классификация хроматических векторов	36
3.4.2 Определение отсутствия звучащего аккорда	36
3.4.3 Исправление ошибок классификации	37
3.5 Выводы	37
4 Получение признаков с использованием нейронных сетей	38
4.1 Теоретические сведения	39
4.2 Построение нейронной сети и предобучение при помощи автоассоциаторов	41
4.3 Выводы	43

5 Эксперименты	44
5.1 Оценка качества распознавания аккордов	45
5.1.1 Коллекции текстовых аннотаций	45
5.1.2 Сопоставление последовательностей аккордов	46
5.1.3 Сопоставление границ сегментов	48
5.1.4 Статистическая значимость	48
5.1.5 Совокупная длительность	49
5.1.6 Типы ошибок	49
5.2 Вычисление спектрограммы	50
5.2.1 Определение ритма	50
5.2.2 Определение задержки	51
5.2.3 Определение частоты настройки	53
5.2.4 Разрешение по времени и по частоте, сглаживание	54
5.3 Преобразования спектрограммы	57
5.3.1 Применение аналога фильтра Превитт	57
5.3.2 Настройка алгоритма вычисления признаков CRP	57
5.3.3 Применение самоподобия	61
5.4 Нейронные сети	62
5.4.1 Конфигурация нейронной сети	62
5.4.2 Влияние логарифмирования спектрограммы	62
5.4.3 Влияние зашумления на этапе предварительного обучения	63
5.4.4 Влияние циклических сдвигов на этапе тестирования	63
5.5 Классификация векторов признаков	64
5.5.1 Шаблонные векторы	64
5.5.2 Эвристики	67
5.5.3 Определение отсутствия аккорда	67
5.6 Результаты MIREX Audio Chord Estimation 2013	67
5.7 Быстродействие	69
5.8 Выводы	70
Заключение	72
Список рисунков	74
Список таблиц	75
Литература	76
А Результаты распознавания аккордов в используемой коллекции	84

Введение

С давних времён в области музыки возникают задачи воспроизведения (в том числе повторного), записи, хранения, классификации, поиска. В настоящее время все они решаются в том числе с помощью компьютеров. Тематика данной работы относится к областям классификации и поиска музыки.

На текущий момент компьютер является основным средством для хранения и обработки музыки и любой информации о музыке, будь то ноты, биография композитора, год выпуска записи или график концертов группы. Сама по себе музыка, содержащаяся в цифровых звукозаписях, более ценна для человека, поскольку никакая информация о ней не может заменить собой её прослушивание. Вместе с тем, именно эта дополнительная информация даёт возможность ориентироваться в музыкальных коллекциях, находить новую музыку, организовывать существующие записи. В силу большей ценности музыки, зачастую звукозаписи не сопровождаются дополнительной информацией. Необходимость получения разнообразной информации о конкретной цифровой звукозаписи порождает множество задач, связанных с обработкой звука: идентификация композиции, нахождение разных версий одной композиции, определение заданной композиции в потоке звука с радио, поиск похожих композиций, определение мелодии композиции для последующего воспроизведения на музыкальном инструменте и другие. Эта диссертация посвящена задаче определения последовательности аккордов в звуке.

Одним из способов решения названных задач является автоматизированное при помощи компьютера извлечение мелодии и/или аккордов из цифровой записи звука с последующим использованием извлеченной информации для индексации, поиска, сравнения. Данная диссертация посвящена задаче определения последовательности аккордов в звуке, представленном в цифровом виде.

Аккорды – это основная информация, необходимая гитаристу для того, чтобы сыграть композицию. Многочисленные гитаристы-любители, не имеющие достаточно опыта или усидчивости, чтобы самостоятельно определить звучащие аккорды, смогут получить инструмент, решающий эту задачу за них. Такой инструмент может быть полезен при обучении музыке, в курсах гармонии, музыкальной формы, сольфеджио.

Информация о последовательности аккордов может быть использована также для индексации композиций и последующего поиска по запросу. Среди возможных сценариев такого поиска можно отметить следующие:

- поиск заимствований или разных версий одной и той же композиции;
- поиск композиций, которые могут гармонично сочетаться друг с другом.

Последовательность аккордов может быть представлена в текстовом виде. С одной стороны, представление звука в виде последовательности символов позволяет перенести задачи индексации и поиска музыки в хорошо проработанную область индексации и поиска текстов. С другой стороны, такое представление композиции позволяет человеку вводить поисковые запросы без использования звуковых файлов.

Первые попытки обработки музыкальной информации в символьном виде были сделаны в 1950-х годах с появлением первых компьютеров. Они были связаны с автоматическим опре-

делением закономерностей в музыке с целью использования их для создания новых мелодий (см. [91]). Тогда же было предложено использовать компьютер для распознавания и печати нотных записей, анализа схожести различных композиций и поиска по образцу. В 1960-х годах появляются первые работы (например, [29]), связанные с анализом звукозаписей, представленных в цифровом виде. Их целью было, прежде всего, понимание того, из чего состоят воспроизводимые музыкальными инструментами звуки и как они воспринимаются человеком.

В 1975 году в [83] было положено начало новому применению компьютера к анализу цифровых музыкальных звукозаписей: распознаванию в них отдельных нот. Этот процесс объединяют с компьютерным распознаванием нотных записей под общим названием *транскрибирование*. Здесь впервые теория музыки используется для анализа композиции не в виде нотной записи, а в том виде, в котором её воспринимает обычный слушатель – в виде звукозаписи. Несмотря на раннюю постановку и большое количество приложенных усилий, задача транскрибирования музыкальной звукозаписи не решена до сих пор.

В 1982 году компаниями Sony и Philips было запущено массовое производство компакт-дисков, на которых музыка была записана в цифровом формате. Со временем доступных в цифровом виде произведений стало на порядки больше, чем доступных нотных записей. Закономерно возрос интерес к автоматическому транскрибированию музыки. В [83] рассматривались только звукозаписи, содержащие не более двух одновременно звучащих музыкальных инструментов. В 1996 году в [75] был представлен один из первых методов, подходивших для любой многоголосной цифровой аудиозаписи.

Задача определения последовательности аккордов при этом не отделялась от задачи транскрибирования. Как отмечает Т. Фуджишима в [94], в 1980-1990-х годах (например, в работе [12]) проблема распознавания аккордов в музыке решалась путём распознавания отдельных нот и их последующего объединения в аккорды. Он же впервые предложил метод распознавания аккордов без предварительного транскрибирования звукозаписи. В [12] метод распознавания аккордов являлся частью системы для автоматического аккомпанемента выступлению живого человека.

В 2000-х годах определение аккордов окончательно выделяется в отдельную задачу. Начиная с 2008 года в рамках ежегодной кампании по оценке методов музыкального информационного поиска MIREX [73] проводятся соревнования среди алгоритмов распознавания аккордов в музыкальном звучании. За это время был достигнут существенный прогресс в качестве распознавания. В 2013 году на это соревнование были выставлены более 10 алгоритмов.

В 2010-х годах появляются широко доступные программные продукты, включающие в себя такие алгоритмы. Популярный пакет для профессионального создания музыки *Ableton Live 9* [9] позволяет преобразовать любую звукозапись, в том числе полифоническую, в нотное представление в редакторе. Эта возможность может быть использована в первую очередь как альтернативный способ ввода нотных данных для последующего редактирования.

Приложения для смартфонов *AnySong Chord Recognition* [11] и *Chord Detector* [24] позволяют определить аккорды в звуковом файле и показывают соответствующие гитарные табулатуры, позволяя играть на гитаре композицию одновременно с её воспроизведением. Соответственно, эти приложения нацелены в первую очередь на использование с музыкой для гитары.

Интернет-сервис Chordify [26] позволяет определить последовательность аккордов в произвольном видео с <http://youtube.com>, аудио с <http://soundcloud.com> или в загруженной пользователем звукозаписи, после чего воспроизвести звук или видео с одновременной индикацией звучащего аккорда. Наряду с недостаточным качеством распознавания, недостатком этого продукта является отсутствие возможности поиска по заданной последовательности аккордов. На сегодняшний день автору не известны какие-либо продукты, предназначенные для обработки коллекции разнообразных музыкальных звукозаписей с целью поиска похожих или гармонично сочетающихся друг с другом композиций.

Музыкальные звуки имеют длительность, которая, как правило, существенно меньше длительности всей композиции. Аккорд как совокупность звуков также имеет определенную относительно небольшую длительность. Поэтому естественно анализировать звукозапись, разделяя её на короткие фрагменты соразмерной длины.

Для каждого фрагмента вычисляется набор признаков, в сжатом виде представляющий необходимую информацию о звуке в пределах этого фрагмента. По полученному промежуточному представлению звука в виде набора признаков определяется соответствующий аккорд. Признаки позволяют представить в компактном по сравнению со звукозаписью виде основную информацию о звуке в данном фрагменте. В отличие от амплитудно-частотного представления, признаки не содержат дублирующейся информации. Было предложено большое количество разнообразных алгоритмов получения звуковых признаков, использующих особенности звучания музыкальных инструментов, особенности человеческого восприятия и возможные помехи на звукозаписях. Итоговое качество распознавания аккордов зависит как от способа получения набора признаков, так и от алгоритма, сопоставляющего этому набору конкретный аккорд.

В 2010-х годах становятся чрезвычайно популярными так называемые методы обучения представлением. Они, фактически, являются алгоритмами со множеством автоматически подбираемых параметров, позволяющими получить признаки, наилучшим образом отражающие необходимую для дальнейшего использования информацию. Но до Хамфри [47] никто не применял методы глубокого обучения к распознаванию аккордов.

Наиболее простым способом определения аккорда по набору признаков является метод ближайшего соседа: вычисление расстояний от заданного набора до «идеальных», шаблонных наборов признаков для каждого аккорда. При этом можно рассматривать разные пространства признаков. Недостатком такого подхода является невозможность учесть взаимосвязи между последовательными фрагментами: аккорды зачастую звучат дольше одного фрагмента, а каждый следующий аккорд не является независимым от предыдущего.

Вероятностные модели позволяют найти в некотором смысле наилучшую из заданного класса метрик. Большинство алгоритмов, представленных в рамках соревнований MIREX Audio Chord Estimation, используют скрытую марковскую модель или байесовскую сеть и моделируют последовательность векторов признаков как марковский процесс. При этом наблюдениями модели являются признаки в каждом фрагменте, а скрытыми состояниями – соответствующие аккорды. Параметры моделей настраиваются в процессе обучения на размеченных данных. Несмотря на достаточно высокое качество распознавания аккордов, такого рода модели имеют свои недостатки. Среди них Де Хаас в [31] выделяет следующие:

- Потребность в большом количестве данных для обучения. Подготовка таких данных весьма трудоёмка, а сами данные могут сильно различаться для разных стилей музыки, эпох, композиторов.
- Опасность переобучения. Модели с большим количеством параметров наилучшим образом подстраиваются под доступный набор обучающих данных, но непонятно, насколько хорошо они будут подходить для работы с данными не из обучающей выборки.
- Многомерность данных. Она приводит к экспоненциальному увеличению объема данных и времени их обработки, а также к росту необходимого объема обучающей выборки.
- Недостаточное использование времени. Марковское свойство предполагает зависимость только от предыдущего фрагмента. Но музыкальная композиция зачастую имеет определенную, достаточно протяжённую по времени, структуру, которая не может быть отражена в модели.
- Существуют другие условия, которые также не могут быть выражены в рамках обучаемой модели. Например, это культурный или географический контекст или сложившиеся практики и правила создания музыки.

- Сложность интерпретации модели, оперирующей в большей степени искусственными, математическими, нежели музыкальными конструкциями.

Ещё одним недостатком является то, что упомянутые вероятностные методы хорошо приспособлены для моделирования смены состояния (звучащего аккорда) и хуже – для моделирования продолжительности нахождения в одном состоянии.

Некоторые из этих недостатков могут быть преодолены при помощи методов глубокого обучения, имеющих в основе многослойную нейронную сеть (например, [86]). Каждый из её слоёв предварительно обучается без учителя на размеченных данных, после чего требуется лишь небольшое количество размеченных примеров для окончательной подстройки параметров.

Все перечисленные проблемы не могут быть разрешены в рамках модели, строящейся исключительно путём обучения на реальных данных. Де Хаас предложил другую модель, которая строится на основе правил западной тональной гармонии без использования алгоритмов машинного обучения, а следовательно, менее подверженную описанным выше недостаткам. Она допускает простую интерпретацию и может быть использована для гармонического анализа композиции. Эта модель позволяет корректировать последовательность, полученную после вычисления евклидовых расстояний между векторами признаков и шаблонами аккордов. К сожалению, при попытке применения модели ко всей последовательности расстояний требуется перебор слишком большого количества вариантов. Поэтому требуется разделять последовательность на короткие участки. Другим недостатком этой модели является необходимость привязки к фрагментам, для которых считается, что аккорд изначально был определён верно. Ошибка в таком фрагменте влечёт за собой ошибки в соседних фрагментах.

Таким образом, разработка метода для распознавания последовательности аккордов, не требующего большого объема данных для обучения, и не предполагающего использования сложной многопараметрической самообучающейся модели, но при этом сопоставимого по качеству результатов с уже существующими методами, является вполне естественной и актуальной. Именно разработка такого метода стала целью для автора данной работы.

С учётом описанных выше недостатков существующих подходов, для достижения поставленной цели перед автором данной работы были поставлены следующие задачи:

1. разработать метод для более точного выделения в спектре звука компонент, соответствующих музыкальным инструментам, с целью улучшения существующих алгоритмов вычисления признаков по фрагменту звукозаписи;
2. исследовать применимость некоторых универсальных методов обучения представлениям к получению музыкальных признаков;
3. исправить недостатки алгоритма определения аккорда по вектору признаков, использующего сопоставление с шаблонами аккордов;
4. реализовать описанные алгоритмы в виде комплекса программ, позволяющего распознавать последовательность аккордов в поданном на вход звуковом файле;
5. сравнить качество распознавания аккордов с аналогами, приняв участие в соревновании MIREX Audio Chord Estimation.

В рамках данной работы все поставленные задачи были решены.

При решении поставленных задач в работе использованы методы математического моделирования, спектральный анализ (для получения и обработки спектрограммы), алгоритмы машинного обучения (для получения признаков), методы объектно-ориентированного программирования и многопоточного программирования (для реализации описанных методов и ускорения вычислений).

В диссертации получены следующие основные результаты, которые выносятся на защиту.

1. Новый метод распознавания последовательности аккордов в звукозаписи, не использующий алгоритмов машинного обучения.
2. Новый метод представления звукозаписи в виде последовательности векторов признаков с применением многослойной нейронной сети.
3. Сравнительный анализ результатов работы предлагаемых методов на коллекции из 319 звукозаписей, подтверждающий их эффективность.
4. Реализующий предложенные методы комплекс программ на языках Java и Python, созданный в рамках данной работы.

Разработанный метод распознавания последовательности аккордов может применяться для анализа звукозаписей с целью их самостоятельного воспроизведения, с целью поиска схожих музыкальных композиций. Метод не подвержен опасности переобучения под конкретную музыкальную коллекцию.

Основные результаты диссертационной работы докладывались на всероссийской научной конференции "Анализ Изображений, Сетей и Текстов"(Екатеринбург, 2012), на всероссийской научной конференции "Анализ Изображений, Сетей и Текстов"(Екатеринбург, 2013), на 9-й международной конференции по вычислениям в области звука и музыки (Копенгаген, 2012), на 13-й конференции международного сообщества по музыкальному информационному поиску (Порто, 2012).

Результаты изложены в 4 печатных изданиях, 1 из которых изданы в журналах, рекомендованных ВАК, 3 – в тезисах докладов всероссийских и международных конференций. Алгоритм был выставлен на соревнования среди алгоритмов распознавания аккордов MIREX Audio Chord Estimation 2012 [68], [69] и MIREX Audio Chord Estimation 2013 [72], [70], [71], проводимые международной лабораторией оценки систем музыкального информационного поиска (International Music Information Retrieval Systems Evaluation Laboratory) университета Иллинойса, США.

Журналы из перечня ведущих периодических изданий:

Н. Ю. Глазырин: "О задаче распознавания аккордов в цифровых звукозаписях Известия Иркутского государственного университета, серия "Математика 2013, Т. 6, № 2. с. 2-17.

Тезисы международных конференций:

Nikolay Glazyrin, Alexander Klepinin: «Chord Recognition using Prewitt Filter and Self-Similarity», Proceedings of the 9th Sound and Music Computing Conference, Copenhagen, Denmark, 11-14 July, 2012, pp. 480-485.

Тезисы всероссийских конференций:

Николай Глазырин, Александр Клепинин: «Выделение гармонической информации из музыкальных аудиозаписей». Доклады всероссийской научной конференции "Анализ Изображений, Сетей и Текстов"(АИСТ 2012), Москва, Национальный Открытый Университет "Интуит с. 159-168.

Николай Глазырин: «Применение автоассоциаторов к распознаванию последовательностей аккордов в цифровых звукозаписях», Доклады всероссийской научной конференции "Анализ Изображений, Сетей и Текстов"(АИСТ 2013), Москва, Национальный Открытый Университет "Интуит с. 199-203.

Все исследования, результаты которых изложены в данной работе, получены лично соискателем в процессе научных исследований. Из совместных публикаций в диссертацию включен лишь тот материал, который непосредственно принадлежит соискателю.

Диссертация состоит из введения, четырех глав, заключения и двух приложений. Полный объем диссертации составляет XXX страница с XX рисунками и XX таблицами. Список литературы содержит XXX наименований.

В главе 1 представлены необходимые для дальнейшего изложения сведения из теории музыки. Делается формальная постановка и теоретические основы задачи распознавания аккордов в музыке.

Глава 2 посвящена подробному обзору литературы по рассматриваемой теме.

В главе 3 описываются улучшения для алгоритмов вычисления векторов признаков и получения аккорда по вектору признаков. Вычисление спектрограммы с повышенным разрешением по времени и частоте с последующим применением скользящего фильтра и прореживанием позволяет лучше сохранить компоненты спектра звука, соответствующие звучанию музыкальных инструментов с определённой высотой звучания. Это помогает более точно передать необходимую для определения аккорда информацию в векторах признаков. Коррекция каждого из них с использованием наиболее схожих других векторов, а также некоторые эвристические правила для коррекции последовательности распознанных аккордов дают возможность исправить некоторые ошибки определения звучащего аккорда. Описанные улучшения позволили вплотную приблизить качество распознавания аккордов к результатам алгоритмов, использующих обучаемые вероятностные модели.

В главе 4 описывается способ вычисления векторов признаков с использованием некоторых вариантов многослойных нейронных сетей с предварительным послойным обучением – многослойных автоассоциаторов. Рассматриваются также рекуррентные многослойные автоассоциаторы, позволяющие моделировать зависимость вектора признаков на текущем фрагменте от векторов признаков на предыдущих фрагментах звукозаписи.

В главе 5 описываются и анализируются результаты экспериментов. Исследуется влияние параметров описанных алгоритмов на результат, а также количественный вклад каждого из реализованных методов в повышение качества распознавания аккордов. Анализируется быстроедействие реализованного метода.

Глава 1

Необходимые теоретические сведения

Для формальной постановки задач, рассматриваемых в данной работе, требуются некоторые специальные теоретические сведения из музыкальной теории. Данная глава предназначена для знакомства читателя с соответствующей областью музыкальной теории и формализации рассматриваемых в данной работе задач. В параграфе 1.1 даются базовые понятия звука и спектра. В параграфе 1.2 описываются основные свойства звука с точки зрения теории музыки. В разделе 1.3 разъясняются основные понятия из теории музыки, необходимые для дальнейших рассуждений. В параграфе 1.4 представлены основы представления звука в цифровом виде. В параграфе 1.5 указаны характерные черты музыкальных звукозаписей, которые могут быть использованы для решения рассматриваемых задач. В параграфе 1.6 приведена формальная постановка задачи.

1.1 Звук

Большая Советская Энциклопедия [6], с. 432, определяет звук в широком смысле как колебательное движение частиц упругой среды, распространяющееся в виде волн в газообразной, жидкой или твёрдой средах. В воздухе звук передается как последовательность сгущений и разрежений. Поэтому звук можно считать непрерывной функцией $x(t)$, показывающей зависимость давления воздуха в данной точке от времени. В рамках данной работы нас будет интересовать только звук в узком смысле как явление, субъективно воспринимаемое человеком через органы слуха. Уловленные ими колебания преобразуются в нервные импульсы, которые передаются в мозг человека. Воспринимаемый человеком звук $x'(t)$ определяется как общим строением органов слуха, так и их индивидуальными особенностями для конкретного человека.

Если звук был вызван колебательным процессом с периодом T_0 и частотой $f_0 = \frac{1}{T_0}$, то полученный звуковой сигнал также будет иметь *период* T_0 и *частоту* f_0 . Будем называть такой звук *чистым тоном*. Реальные звуковые сигналы обычно вызваны множеством колебаний с различными частотами, поэтому можно говорить о *частотном спектре* звука или его *спектральной функции* $a(f)$ в заданный момент времени. Это неотрицательная функция, которая отражает зависимость интенсивности (амплитуды) колебаний на конкретной частоте от этой частоты в данном звуковом сигнале $x(t)$ в момент времени t . Также выделяют спектр мощности и фазовый спектр сигнала. В дальнейшем, если не оговорено иное, под спектром будет пониматься частотный спектр звукового сигнала.

Спектр сигнала меняется со временем, поэтому имеет смысл анализировать его в окрестности $[t_{start}, t_{end}]$ некоторой точки t . Размер окрестности должен быть не меньше периода самой низкой из частот спектра. Понятно, что для более высоких частот достаточно меньшей окрестности. Поэтому приходится принимать допущение о том, что спектр звука не меняется

(или меняется незначительно) в пределах данной окрестности. Для более высоких частот, соответственно, спектр будет усреднен по промежутку $[t_{start}, t_{end}]$.

Любой ограниченный сигнал, определённый на промежутке $[t_{start}, t_{end}]$, можно периодически продолжить на всю вещественную ось с периодом $\tau = t_{end} - t_{start}$. Продолженная таким образом функция $x(t)$ будет ограниченной. Потребуем также, чтобы она была непрерывной. Этого можно добиться, например, сделав плавное затухание сигнала рядом с точками t_{start} , t_{end} путем умножения на гладкую функцию, равную нулю в этих точках. Заданная таким образом функция будет непрерывной и ограниченной, поэтому она может быть однозначно выражена в виде ряда гармонических функций (или *гармоник*), частоты которых кратны $1/\tau$:

$$x(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos \left(2\pi \frac{k}{\tau} t - \phi_k \right),$$

где a_k – амплитуда, а ϕ_k – фаза k -й гармонической функции. Значения a_k составляют спектр звукового сигнала $x(t)$. Если $x(t)$ является чистым тоном с частотой f_0 и периодом $\tau = 1/f_0$, сумма вырождается в одно слагаемое $a_{f_0} \cos(2\pi f_0 t - \phi_{f_0})$. Данная сумма представляет собой ряд Фурье для полученной периодической функции.

Звуки, издаваемые музыкальными инструментами, не являются чистыми тонами. В каждом таком звуке можно выделить *основной тон*, имеющий наиболее низкую частоту, и *обертоны*, имеющие более высокие частоты. Обертоны, у которых частоты кратны частоте основного тона, называют гармоническими. Они характерны, например, для струнных музыкальных инструментов. Обертоны с другими частотами называют негармоническими. Обилие обертонов придаёт звуку насыщенность, но при этом затрудняет выделение чистых тонов.

1.2 Свойства звука

В классическом учебнике по элементарной теории музыки В. А. Вахромеева [4] выделяются 4 основных свойства музыкального звука: высота, длительность, громкость, тембр. Рассмотрим их более подробно.

Высота звука отражает субъективное восприятие человеком частоты звука. Высота звука нелинейно (но монотонно) зависит от его частоты. На основе экспериментов были предложены различные модели этой зависимости, в том числе шкала мёлов и шкала барков (1 барк = 100 мел). Более подробно эти модели описаны, например, в [61], с. 79-81 или в [1], с. 155-156. Высота звука может быть выражена с разной степенью ясности. Высота звуков, имеющих основной тон, определяется его частотой. Человек способен различать высоту только у периодических сигналов (по основному тону) [1]. Для остальных звуков (например, разного рода шумы, шорохи, звуки шумящих и ударных музыкальных инструментов) высота может быть неясной. Также человек не различает высоту у очень коротких звуков: короче 60 мс для низкочастотных звуков, короче 15 мс для частот 1-2 кГц. Эти значения задают разумные ограничения снизу на длину промежутка $[t_{start}, t_{end}]$, на котором имеет смысл рассматривать спектр звука.

Длительность звука соответствует длительности колебаний источника звука. Она приобретает особое значение в контексте музыкального произведения, когда последовательность звуков и их продолжительность задают ритм. То есть длительности звуков и размещение звуков на временной оси в большинстве случаев носят не случайный, а закономерный характер.

Громкость звука определяется амплитудой колебаний. Но, как в случае с высотой, эта характеристика звука является субъективной. Воспринимаемая человеком громкость зависит как от амплитуды (нелинейно и монотонно), так и от высоты звука (нелинейно и немонотонно). Эти зависимости подробно описаны в [42]. Также воспринимаемая громкость зависит от спектрального состава и длительности звуков.

Тембр или окраска звука определяется частотами, процессами вступления и затухания и интенсивностью его обертонов, которые, в свою очередь, определяются физическими свойствами музыкального инструмента и способом звукоизвлечения. Благодаря разнице в тембрах человек может отличать друг от друга разные музыкальные инструменты.

1.3 Основные понятия из теории музыки

Определения этого раздела даны в соответствии с [4] и [5].

Музыкальной системой называется отобранный практикой ряд звуков, которые находятся в определённых соотношениях по высоте. Музыкальная система является результатом длительно развивающейся музыкальной практики человеческого общества. Для нас наиболее привычна система, сформировавшаяся в европейской, в том числе русской классической музыке. Далее под музыкальной системой будет пониматься именно эта система.

Звукорядом называется совокупность звуков музыкальной системы, расположенных в порядке высоты (в восходящем или нисходящем направлении).

Степению называется звук музыкальной системы. Основные ступени соответствуют звукам, извлекаемым на фортепиано на белых клавишах. Им присвоены собственные названия: *до, ре, ми, фа, соль, ля, си*. Необходимо отдельно отметить, что слово «нота» обозначает графическое изображение звука. Тем не менее, оно часто используется как синоним для понятия «звук», например, «нота *до*» в значении «звук *до*».

Строем называется совокупность постоянных отношений по высоте между звуками музыкальной системы.

Человек воспринимает звуки с частотами f_0 и $2f_0$ (до 5000 Гц) как очень похожие и тесно связанные друг с другом. Расстояние между такими звуками называется *октавой*. Как отмечает Д. Левитин в [64], «в основе музыки каждой из известных нам культур лежит октава [. . .] даже некоторые животные – например, обезьяны и кошки, – воспринимают звуки, отличающиеся на октаву, как похожие».

Звукоряд делится на октавы на основе октавного сходства его звуков и отражающей это сходство повторности их названий. В свою очередь, каждая октава имеет своё название: субконтр-октава, контр-октава, большая октава, малая октава и октавы с первой по пятую. Началом октавы принято считать звук ступени *до*.

Темперированным называется строй, который делит каждую октаву звукоряда на равные части. С начала XVIII века в европейской музыке принята двенадцатизвуковая (двенадцатиступенная) темперация, делящая октаву на 12 равных друг другу частей, называемых *полутонами*. Полутон является наименьшим расстоянием по высоте, возможным в двенадцатизвуковом темперированном строе. Он образуется между звуками любых двух соседних ступеней звукоряда. Расстояние, образованное двумя полутонами, называется *целым тоном*. Расстояние между двумя соседними основными ступенями звукоряда (соответствующими белым клавишам фортепиано) может быть равно полутону (например, *ми–фа*) или целому тону (например, *фа–соль*).

Частоту каждой ступени звукоряда можно вычислить по формуле

$$f_j = f_0 \cdot 2^{j/12}, \quad (1.1)$$

где f_0 – частота настройки музыкальных инструментов. В 1939 году на международной конференции в Лондоне был принят стандарт для частоты настройки $f_0 = 440$ Гц. Эту частоту фиксируют для звука *ля* первой октавы. Клавиатура фортепиано охватывает 88 ступеней: от *ля* субконтроктавы до ступени *до* пятой октавы. Частота, соответствующая k -й слева клавише фортепиано (отсчитывается с нуля), может быть вычислена по формуле

$$f_k = 27.5 \cdot 2^{k/12}$$

Широко используемый в настоящее время стандарт MIDI (Musical Instrument Digital Interface) [66], задающий формат обмена данными между электронными музыкальными инструментами, определяет 128 возможных значений для частоты звука [67]. Частота, соответствующая ступени с номером k , $0 \leq k \leq 127$, может быть получена по формуле

$$f_k = 2^{\frac{k-69}{12}} \cdot 440$$

И наоборот, номер ступени может быть получен из частоты по формуле

$$k = 69 + \text{round} \left(12 \log_2 \left(\frac{f}{440} \right) \right) \quad (1.2)$$

Приведенные выше формулы справедливы для стандартного значения частоты настройки $f_0 = 440$ Гц. В рамках стандарта MIDI звук *ля* первой октавы соответствует 69-й ступени.

Производными называются ступени звукоряда, получаемые посредством повышения или понижения его основных ступеней. Повышение или понижение ступени называется *альтерацией*. Знаки альтерации указывают на повышение или понижение основной ступени. Для дальнейшего изложения важны знаки *диез* (\sharp) и *бемоль* (b), обозначающие, соответственно, повышение и понижение на один полутоном.

Интервалом называется расстояние по высоте между двумя звуками, взятыми последовательно или одновременно. Звуки интервала, взятые последовательно, образуют мелодический интервал. Звуки интервала, взятые одновременно, образуют гармонический интервал.

Каждый интервал определяется двумя величинами – количественной и качественной. Количественной называется величина, выраженная количеством ступеней, составляющих интервал. Качественной называется величина, выраженная количеством тонов и полутонов, составляющих интервал.

Интервалы, образующиеся в пределах октавы, называются простыми. Всего существует 8 простых интервалов. Их названия зависят от количества основных ступеней, которые они охватывают. Каждое название обозначает порядковый номер второго звука интервала, как если бы от первого его звука брались все ступени до него подряд: прима, секунда, терция, кварта, квинта, секста, септима, октава. Эти названия характеризуют количественную величину интервала.

Как было отмечено выше, расстояние между двумя соседними основными ступенями (образующими секунду) может быть равно полутону или целому тону. Аналогично, терция *до–ми* состоит из 2 целых тонов, а терция *ре–фа* – из 1 целого тона и 1 полутона. Качественная величина интервала определяет различие звучания однородных интервалов. Она обозначается словами: большая, чистая, увеличенная, уменьшенная. Например, терция *до–ми* называется большой, а терция *ре–фа* – малой.

Обращением интервала называется перемещение его нижнего звука на октаву вверх или верхнего звука на октаву вниз.

Аккордом называется одновременное сочетание трёх или более звуков, которые расположены по терциям или могут быть расположены по терциям. Аккорд строится от нижнего звука вверх. Аккорд, состоящий из трёх звуков, расположенных по терциям, называется *трезвучием*. Мажорное трезвучие состоит из большой и малой терций (4 и 3 полутона соответственно). Минорное трезвучие состоит из малой и большой терций. Уменьшенное трезвучие состоит из двух малых терций. Увеличенное трезвучие состоит из двух больших терций. Во всяком трезвучии, независимо от его типа, нижний звук называется *основным звуком* или *примой*, второй (по расстоянию от примы) – *терцией*, а третий – *квинтой*.

Аккорд, состоящий из четырёх звуков, расположенных по терциям, называется *септаккордом*. Его крайние звуки образуют интервал септимы. Наиболее употребительны доминант-септаккорд (большая, малая, малая терции), уменьшенный септаккорд (малая, малая, малая

терции), малый септаккорд (малая, малая, большая терции) и минорный септаккорд (малая, большая, малая терции).

Основным аккордом называется такое положение аккорда, в котором основной звук лежит ниже остальных его звуков. *Обращением* аккорда называется такое его положение, в котором нижним звуком является терция или квинта основного трезвучия. Обращения получаются посредством переноса звуков основного трезвучия вверх на октаву.

Аккорды применяются к музыке не только как сопровождение (аккомпанемент) к данной мелодии, но часто проявляются и в самой мелодии, когда её движение следует по аккордовым звукам. Последовательность, образованная несколькими аккордами, называется *гармоническим оборотом*.

Ритмом называется организованная последовательность длительностей звуков. *Ритмическим рисунком* называется последовательность звуковых длительностей, взятая отдельно от высотных соотношений звуков. Основные соотношения звуковых длительностей в музыке таковы, что каждая более крупная длительность относится к ближайшей более мелкой как 2 к 1. При этом нотные знаки обозначают только относительную длительность звуков, но не абсолютную.

Чередование звуков равными по времени долями образует в музыке равномерное движение или пульсацию. В этом движении звуки некоторых долей выделяются ударами. Такое выделение звука посредством большей громкости (часто также длительности) по сравнению с окружающими звуками называется *акцентом*. *Метром* называется непрерывно повторяющаяся последовательность акцентируемых и неакцентируемых равнодлительных ритмических единиц (отрезков времени). Эти ритмические единицы времени, образующие метр, называются, в свою очередь, *метрическими долями*. Акцентируемая доля называется *тяжёлой* или *сильной*, неакцентируемая – *лёгкой* или *слабой*. Акценты, как правило, повторяются через одинаковое количество долей: через одну, две и т.д.

Размером в нотной записи называется метр, доля которого выражена определённой ритмической длительностью (например, четвертью ноты). Размер обозначается дробью, числитель которой говорит о количестве его долей, а знаменатель – о длительности, которая принята за долю. *Тактом* называется часть музыкального произведения, которая начинается с тяжёлой доли и заканчивается перед следующей тяжёлой долей. *Темпом* называется скорость движения, частота пульсирования метрических долей. Темп иногда указывают числом, которое обозначает количество ударов метронома в минуту.

Размер и метр формируют сетку на временной шкале. Очень часто начала нот оказываются выровнены по этой сетке. Вместе с тем, эта сетка может оказаться неравномерной, поскольку понятия метра и темпа являются относительными, субъективными для исполнителя. Изменение акцентов, изменение длительности такта являются средствами музыкальной выразительности, но затрудняют автоматическую обработку звукозаписи.

Для музыкальной выразительности также необходимо объединение нескольких звуков или созвучий в систему, основанную на определённых высотных соотношениях и связях. В таких системах есть звуки, используемые как опора (в частности для окончания мелодии). Эти звуки появляются на тяжёлой доле такта, в конце музыкальной мысли (что часто бывает на чётных тактах). Кроме того, мелодия время от времени возвращается к таким звукам. Музыкальная практика выделяет среди таких звуков один, наиболее устойчивый, который называется *тоникой*. Неустойчивыми называются звуки системы, в которых выражается незавершённость музыкальной мысли. Переход неустойчивого звука в устойчивый называется *разрешением*. *Тяготением* называется притяжение неустойчивого звука системы к устойчивому, отстоящему от него на секунду.

Ладом называется система взаимоотношений между устойчивыми и неустойчивыми звуками. Многие лады состоят из 7 звуков, но существуют лады с большим и меньшим их числом. В основе отдельной мелодии и музыкального произведения в целом всегда лежит определённый

ный лад. *Тональностью* называется высотное положение лада. Название тональности состоит из обозначения тоники и обозначения лада. В двух основных ладах – мажорном и минорном – устойчивые звуки, взятые вместе, образуют соответственно мажорное и минорное трезвучия.

1.4 Цифровой звук

Звуковой сигнал $x(t)$ может быть представлен в цифровом виде при помощи операций *дискретизации* и *квантования*. Для этого с некоторой частотой ν раз в секунду измеряется амплитуда функции $x(t)$ (дискретизация), после чего каждое полученное значение $x(t_i)$ заменяется на ближайшее из заданного множества X_Q возможных значений амплитуды (квантование). Как правило, это множество содержит 2^8 , 2^{16} или 2^{24} элементов, чтобы каждое значение можно было представить целым числом байт. Частота ν часто выбирается равной 44100 Гц (по историческим причинам). При этом ν называют *частотой дискретизации*, а значения $x_Q(t_i)$ – *отсчётами* исходного сигнала $x(t)$. В соответствии с классической теоремой Котельникова, если спектр сигнала $x(t)$ ограничен сверху частотой $\nu/2$ (т.е. $a_k = 0$ для $\frac{k}{\tau} > \nu/2$), то исходный сигнал может быть восстановлен однозначно и без потерь по измеренным значениям $x(t_i)$. При квантовании эти значения заменяются на $x_Q(t_i)$, поэтому исходный сигнал может быть восстановлен из оцифрованного только с некоторой ошибкой, которая тем меньше, чем больше возможных значений амплитуды использовалось при квантовании. Для большинства звукозаписей эта ошибка незаметна на слух. Отметим ещё раз, что спектр любых оцифрованных звуковых сигналов ограничен.

1.5 Свойства музыкальных звукозаписей

Последовательность аккордов имеет смысл определять в звукозаписи, содержащей музыку в том или ином виде. Это может быть как студийная запись на компакт-диске, так и запись гитары через микрофон мобильного телефона. Музыкальные звукозаписи в целом обладают рядом свойств, которые нужно учитывать при определении последовательности аккордов. Каждое из них может быть выражено в большей или меньшей степени или вообще отсутствовать.

- Наличие музыкальной системы и строя, которым подчиняются музыкальные инструменты. Основной тон каждого из звуков музыкальных инструментов обычно принадлежит музыкальной системе, а значит его частота принимает одно из фиксированного набора значений. В целом это приводит к преобладанию энергии звука на соответствующих частотах.
- Одновременное звучание нескольких музыкальных инструментов. При этом звуковые сигналы, издаваемые разными инструментами (и даже разными звучащими элементами одного инструмента, например, струнами) складываются. Точно так же складываются спектры этих инструментов.
- Наличие гармоник у музыкальных инструментов с ясно выраженной высотой звучания. В звучании таких инструментов можно выделить отдельную ноту. При этом наряду с частотой, соответствующей этой основной ноте, звучат другие частоты. Их звучание менее выражено, но они могут соответствовать другим ступеням музыкальной системы. Математически это означает, что если k_0 таково, что a_{k_0} – наибольшая по абсолютному значению компонента спектра звучащей ноты, то существует по меньшей мере одно значение $k > k_0$ такое, что a_k существенно отлична от 0. Соотношения между парами

(k_0, k) для разных k и разных k_0 (соответствующих разным нотам) во многом задают тембр музыкального инструмента.

- Наличие инструментов с невыраженной высотой звучания. К ним относятся многие ударные инструменты, в звучании которых невозможно выделить конкретную ноту. Спектр таких инструментов характеризуется большим количеством расположенных подряд существенно отличных от нуля значений, слабо отличающихся друг от друга. Иными словами, существуют такие положительные числа A и δ , что A существенно больше δ и $A < a_k < A + \delta$ для всех k из некоторого промежутка $[k_0, k_1]$.
- Наличие ритма и метра. Метрические доли зачастую акцентируются началом звучания нот, а в некоторых музыкальных стилях – также звучанием ударных инструментов. Широко употребляются простые метры, где акцент делается один раз на две или три доли. Также широкоупотребителен метр, состоящий из четырёх долей с акцентом на первой и третьей, при этом акцент на первой доле чуть сильнее. Другие метры и размеры используются реже.
- Наличие лада и тональности. Они позволяют объединить в целостную композицию звуки различных музыкальных инструментов и голоса. Они накладывают ограничения на допустимые аккорды в композиции. Вместе с тем, эти ограничения не являются строгими и могут сознательно нарушаться композиторами. Кроме того, тональность может меняться на протяжении композиции, что влечёт за собой изменение набора «допустимых» аккордов.
- Наличие периодичностей (повторений). Как пишет Д. Левитин в [64], «музыка основана на повторениях». Одна и та же музыкальная фраза, последовательность аккордов и даже целый фрагмент композиции могут повториться в точности или с небольшими изменениями.

1.6 Формализация задачи

Теперь, с использованием введённых понятий и обозначений, можно дать формальную постановку основной задачи, решаемой в данной работе: пусть заданы звуковой сигнал $x(t)$, $t \in [t_{start}, t_{end}]$ и множество возможных названий аккордов Y . Необходимо для каждого момента времени $t \in [t_{start}, t_{end}]$ указать аккорд $y \in Y$, звучащий в этот момент.

Эту общую задачу можно разделить на отдельные шаги, которые соответствуют конкретным задачам, перечисленным во введении.

1.6.1 Частотно-временное представление

Представление звука в виде последовательности отсчётов амплитуды не является удобным для обработки: неясно, как сопоставить аккорду последовательность отсчётов и наоборот. Поэтому естественным первым шагом является часто используемый при обработке звука переход к частотно-временному представлению звукозаписи или получение её спектрограммы $C_{N \times M}$. Она представляет из себя матрицу, каждая из N строк которой соответствует определённой частоте, а каждый из M столбцов представляет из себя спектр фрагмента звукозаписи в пределах короткого промежутка. Элементами этой матрицы являются значения интенсивности данной частоты в пределах данного промежутка времени. Удобно представлять спектрограмму в виде последовательности столбцов $C_{N \times M} = \{C_m\}_{m=0}^{M-1}$.

При получении спектрограммы возникает 2 подзадачи: разбиение звукозаписи на фрагменты и вычисление спектра на каждом из них. Для того, чтобы полученная спектрограмма

наиболее полно передавала информацию о музыкальных звуках, необходимо при разбиении на фрагменты и при вычислении спектра учитывать такие особенности музыкальных звукозаписей, как наличие строя и наличие ритма. Задача 1 из списка поставленных задач предполагает разработку такого метода, который позволит как можно лучше использовать эти особенности.

1.6.2 Классификация

Обозначим за $C \subset \mathbb{R}^N$ множество всех возможных векторов-столбцов спектрограммы. Для каждого вектора из данной последовательности $\{C_m\}_{m=0}^{M-1}$ необходимо указать аккорд $y \in Y$, соответствующий этому вектору. Принимая во внимание известные из равномерно темперированного строя частоты нот, можно по спектру звука в данном фрагменте делать предположения относительно звучащих нот и аккорда. Возможно использование не только текущего, но и предыдущих векторов (а также последующих, если от алгоритма не требуется выдавать результат в реальном времени). Также возможно использование результатов классификации других векторов.

На этом этапе главными вопросами являются выбор метода классификации и отыскание такого набора преобразований множества X , который позволит упростить классификацию с использованием выбранного метода. Результатом преобразования будет последовательность векторов признаков, отличная от исходной последовательности векторов-столбцов спектрограммы. Для отыскания подходящих преобразований могут быть использованы свойства, перечисленные в разделе 1.5. Нахождению таких преобразований посвящены задачи 2 и 3 из списка поставленных задач. При этом задача 2 предполагает использование методов обучения представлением, подробнее о которых будет рассказано в главе 4.

В главе 2 описываются основные уже существующие подходы, применяемые для решения каждой из отмеченных выше подзадач.

Глава 2

Обзор литературы

Данная глава посвящена обзору литературы в области распознавания аккордов. Область исследований достаточно молодая, поэтому практически все рассматриваемые работы были опубликованы в последние 15 лет, а наиболее значимые результаты были получены в течение последних 5 лет.

Предварительная обработка звукозаписи (параграф 2.1) так или иначе присутствует практически во всех алгоритмах распознавания аккордов. Она используется для уменьшения объёма вычислений на последующих этапах, а также для подстройки некоторых параметров алгоритма под конкретную звукозапись. Первым шагом к непосредственному распознаванию аккордов является получение спектрограммы (параграф 2.2). В зависимости от наличия или отсутствия предварительной обработки, а также используемого метода для вычисления спектра, к ней применяются те или иные преобразования (параграф 2.3). Конечным итогом всех преобразований является представление спектрограммы в виде последовательности векторов признаков, различные типы которых также описаны в этом параграфе. Наконец, параграф 2.4 описывает часто применяемые методы классификации, позволяющие получить последовательность названий аккордов из последовательности векторов признаков.

2.1 Предварительная обработка

На этом этапе собирается информация, которая будет использоваться для получения частотно-временного представления звукозаписи: определяются моменты начала метрических долей и частота настройки музыкальных инструментов. Иногда дополнительно производится разделение звука на гармонические и перкуSSIONные компоненты, после чего последние удаляются из сигнала. К этому же этапу можно отнести понижение частоты дискретизации цифровой звукозаписи для ускорения вычислений на следующем этапе и преобразование стереофонических записей в монофонические.

Понижение частоты дискретизации применяется во многих работах ([93], [16], [57], [28], [59], [87], [77], [52], [79], [85], [80], [78], [53], [43], [47]) для ускорения обработки файла. Как правило, частота дискретизации понижается со стандартной для компакт-дисков 44100 Гц до 11025 Гц путем замены каждых 4 подряд идущих отсчётов $x_Q(t_i), x_Q(t_{i+1}), x_Q(t_{i+2}), x_Q(t_{i+3})$ на $x_Q(t_i)$. При этом теряется информация о частотах выше 5.5 кГц. Такая потеря не является критичной, поскольку на частотах свыше 5 кГц человеческое восприятие высоты тона существенно изменяется, в частности, нарушается октавное сходство при удвоении частоты звука. Поэтому в музыке большая часть информации находится на частотах до 5 кГц и остаётся незатронутой при таком понижении частоты дискретизации.

Такое преобразование позволяет использовать меньший размер окна при вычислении быстрого преобразования Фурье. Но при отсутствии жестких ограничений на производительность

понижение частоты дискретизации не является обязательным. В [100], [23], [27], [22], [31] частота дискретизации звукозаписей не меняется.

Общепринятым является преобразование стереофонических звукозаписей в монофонические путём взятия среднего арифметического от сигналов левого и правого каналов. При этом предполагается, что в них не будут звучать разные аккорды, что, скорее всего, звучало бы неприятно для человека.

Определение моментов начала метрических долей позволяет на следующем шаге получить спектрограмму, соотносящуюся с ритмом композиции. Смена аккорда, как и любое другое событие в музыке, очень часто подчинена ритму и происходит на границе метрических долей. Кроме того, в столбцах спектрограммы, соответствующих акцентированным долям, будет более ярко выражено звучание инструментов и соответствующие им пики спектра. Моменты начала метрических долей используются как для деления звука на фрагменты, каждый из которых соответствует одной доле или её части ([14], [13], [99], [63], [43]), так и для усреднения столбцов спектрограммы, вычисленной с фиксированным шагом по времени, в пределах одной метрической доли ([16], [79], [78], [31], [25]). В [25], помимо описанных вариантов, также применялась медианная фильтрация по всем столбцам спектрограммы в пределах одной метрической доли, но лучший результат был достигнут с использованием усреднения.

Задача определения моментов начала метрических долей обычно формулируется в рамках музыкального информационного поиска как задача отслеживания ритма (beat detection). Новые алгоритмы для её решения появляются каждый год, но в методах распознавания аккордов обычно применяется один из алгоритмов, представленных в [30], [32], [34]. Соответствующие программные модули для этих алгоритмов доступны бесплатно и удобны в подключении, что, по-видимому, является основной причиной их популярности.

Отклонение частоты настройки музыкальных инструментов от стандартного значения 440 Гц может как явно определяться на этапе предварительной обработки ([40], [87], [52], [53], [43], [10], так и неявно учитываться в процессе обработки спектрограммы ([16], [57], [80], [78], [27]). Особенно важно учитывать это отклонение при использовании преобразования постоянного качества (см. параграф 2.2), использующего заранее заданные частоты для вычисления спектральных компонент. Некоторые алгоритмы для определения частоты настройки представлены в работах [46], [101], [40], [88], [51].

Разделение звука на гармонические и перкуссионные компоненты позволяет ослабить влияние музыкальных инструментов с неясной высотой звучания на спектрограмму, получаемую на следующем этапе. Аналогичные преобразования делаются на этапе преобразования спектрограммы в последовательность векторов признаков во многих работах. Но в [80] и [43] перкуссионные компоненты удаляются из сигнала до построения спектрограммы при помощи свободно доступной для научного использования реализации алгоритма [92].

2.2 Спектрограмма

Переход к частотно-временному представлению звукозаписи в виде спектрограммы является ключевым, поскольку даёт возможность работать с отдельными частотными компонентами звука. Как было отмечено выше, для этого звукозапись делится на короткие, возможно, пересекающиеся фрагменты, на каждом из которых вычисляется спектр звука.

В алгоритмах распознавания аккордов используются следующие методы получения спектра.

1. Дискретное оконное преобразование Фурье.

$$X[n] = \sum_{j=0}^{J-1} w(j)x_Q(t_j)e^{-\frac{i2\pi nj}{J}}, \quad n = 0, 1, \dots, N-1$$

Здесь J – размер анализируемого фрагмента звукозаписи в отсчётах, $w(j)$ – функция, отличная от нуля на некотором промежутке, не выходящем за пределы этого фрагмента – оконная функция. Прямоугольная оконная функция $w(j)$, равная 1 только на анализируемом фрагменте и 0 – вне его, получается автоматически при делении на фрагменты исходной звукозаписи. Среди других оконных функций наиболее популярной является окно Хемминга:

$$w(j) = 0.53836 - 0.46164 \cos\left(\frac{2\pi j}{J-1}\right)$$

При использовании оконной функции результатом преобразования Фурье является не спектр исходного сигнала, а спектр его произведения с оконной функцией. Согласно свойству преобразования Фурье, этот спектр будет равен свёртке спектров исходного сигнала и оконной функции. Её выбор влияет на форму полученных искажений спектра. Более подробную информацию об эффектах от выбора оконной функции можно найти в [3], раздел 10.3.1.

Достоинствами дискретного оконного преобразования Фурье являются существование быстрых алгоритмов вычисления в определённых случаях и наличие большого количества реализаций на разных языках программирования. Вместе с тем, при использовании алгоритмов быстрого преобразования Фурье невозможно произвольным образом выбирать частоты его компонент. Это создаёт неудобства при дальнейшей обработке, поскольку невозможно точно определить количество звуковой энергии, приходящейся на частоты, соответствующие ступеням звукоряда. Дискретное оконное преобразование Фурье используется в [93], [40], [28], [87], [52], [99], [78], [53], [31].

2. Преобразование постоянного качества (constant-Q преобразование).

$$X[n] = \frac{1}{J(n)} \sum_{j=0}^{J(n)-1} w(n, j) x_Q(t_j) e^{-\frac{i2\pi nj}{J(n)}}, \quad n = 0, 1, \dots, N-1$$

Здесь, в отличие от преобразования Фурье, размер анализируемого фрагмента и размер оконной функции зависят от номера соответствующей частотной компоненты f_n . В свою очередь, f_n можно выбрать таким образом, что каждой ступени звукоряда будет соответствовать одинаковое число частотных компонент (одна или более). Пусть N_0 – количество компонент в одной октаве, а f_{min} – частота наименьшей из анализируемых компонент. Тогда частота n -й компоненты задается формулой $f_n = 2^{n/N_0} f_{min}$. Точно так же задаются частоты для ступеней звукоряда при использовании равномерно темперированного строя, поэтому параметр f_{min} напрямую связан с частотой настройки музыкальных инструментов. Отношение $\frac{f_n}{f_{n+1}-f_n} = \frac{1}{2^{1/N_0}-1} = Q$ называется коэффициентом качества. При таком выборе частот Q не зависит от k . Отсюда происходит название constant-Q преобразования.

Достоинством этого преобразования является легкость дальнейшей работы со спектром, поскольку его компоненты напрямую соответствуют ступеням звукоряда. Недостатками являются большая сложность вычислений и зависимость от правильного определения частоты настройки. Более быстрый алгоритм вычисления преобразования постоянного качества, использующий результат быстрого преобразования Фурье исходного сигнала, был предложен в [19]. Преобразование постоянного качества используется в [16], [57], [77], [79], [85], [80], [23], [22], [43].

3. Гребёнка фильтров (filter bank). В цифровой обработке сигналов любое преобразование сигнала называют фильтром. В данном случае под фильтром понимается полосовой фильтр – преобразование, сохраняющее в звуке только частоты, находящиеся в некоторой полосе частот. Известно (см. [2], с. 424-425), что быстрое преобразование Фурье

эквивалентно вполне определенной гребёнке достаточно грубых фильтров. Вместо них можно использовать любые другие фильтры, у каждого из которых центр полосы пропускания соответствует частоте одной из ступеней звукоряда, а ширина полосы пропускания достаточно мала, чтобы не охватывать частоты соседних ступеней. Эти фильтры можно подобрать так, чтобы они были менее грубыми, то есть более точно определяли количество звуковой энергии, приходящейся на их полосы пропускания. Недостатком данного метода является большая вычислительная сложность в сравнении с алгоритмом быстрого преобразования Фурье. Гребёнки фильтров используются в [10], [47].

Независимо от способа получения спектра, он подвергается дальнейшей обработке и в конце концов преобразуется в имеющий меньшую размерность вектор признаков.

2.3 Векторы признаков

Переход от столбцов спектрограммы к векторам признаков основан на том, что человек воспринимает звуки с частотами, отличающимися на октаву, как похожие. Эта же особенность используется композиторами, когда инструменты, звучащие в разных частотных полосах, воспроизводят одну и ту же ноту в разных октавах, или несколько голосов из разных октав составляют один аккорд. Поэтому вполне естественно просуммировать в каждом столбце спектрограммы компоненты, соответствующие одному и тому же звуку в разных октавах. Пусть спектрограмма была получена в результате преобразования постоянного качества, и N_0 – количество частотных компонент в одной октаве в столбце C_m , что соответствует шагу в $N_0/12$ полутонов. Ко всем значениям $C_m[n]$, $0 \leq n < N_0$, прибавляются значения $C_m[n + N_0]$, $C_m[n + 2N_0]$, $C_m[n + 3N_0]$, ... для каждого $0 \leq m \leq M - 1$. В результате из последовательности столбцов $\{C_i\}_{m=0}^{M-1}$ получается последовательность N_0 -мерных векторов $\{B_m\}_{m=0}^{M-1}$.

Если при получении спектрограммы использовалось быстрое преобразование Фурье с размером фрагмента J отсчётов, то необходимо сопоставить компоненты спектра частотам звукоряда. Обычно это делается по следующей формуле

$$n = 12 \log_2 \left(\frac{f_k}{f_0} \right) + 69, k = 0, 1, \dots, J - 1 \quad (2.1)$$

где f_k – частота, соответствующая k -й компоненте спектра, f_0 – частота настройки музыкальных инструментов, n – номер частоты звукоряда. Эта формула применима и для спектрограммы, полученной преобразованием постоянного качества.

Векторы признаков, полученные путём объединения спектральной информации по всем октавам, носят общее название векторов хроматических признаков или *хроматических векторов*. Впервые такой процесс был предложен в [94], а соответствующие признаки получили название *профиль тональных классов* (pitch class profile). Под *тональным классом* здесь понимается совокупность звуков, имеющих одно название, но находящихся в разных октавах, например, все звуки *до*.

В отличие от столбцов исходной спектрограммы, каждый хроматический вектор имеет всего 12 компонент, а значит, соответствующая задача классификации решается в пространстве меньшей размерности. Каждая из координатных осей в этом пространстве соответствует уровню энергии, приходящемуся на один тональный класс. Недостатками такого преобразования является потеря информации об октавах исходных звуков (влекущая потерю информации об обращении аккорда) и наложение шумовых компонент спектра на полезные. Несмотря на это, хроматические векторы используются в большинстве существующих алгоритмов распознавания аккордов.

Для определения обращения аккорда анализируют низкочастотную область спектра, что позволяет определить басовую ноту, по которой, в свою очередь, определяется обращение аккорда. В [77], [79], [53], [43], [31], [25] строятся отдельные спектрограммы для низкочастотной и высокочастотной областей спектра, граница между которыми обычно пролегает в диапазоне от 200 Гц до 250 Гц. Соответственно, получается два набора хроматических векторов, используемых в дальнейшем анализе.

Для отделения полезных компонент спектра от шумовых было предложено множество преобразований. Как правило, они не затрагивают способ получения вектора признаков из столбца спектрограммы, поэтому их итогом является хроматический вектор. Тем не менее, из-за наличия дополнительных преобразований таким векторам иногда дают собственные названия.

В [57] было предложено для случая спектрограммы, полученной быстрым преобразованием Фурье, перед вычислением (2.1) заменять каждое значение $X_m[n]$ на $\prod_{k=0}^{N_{harm}} |X[2^k \cdot n]|$, где N_{harm} – параметр, регулирующий количество гармоник. Это позволяет учесть информацию о гармониках инструментов с определённой высотой звучания в векторе признаков, который был назван *расширенный профиль тональных классов* (enhanced pitch class profile).

В [40] было предложено для случая спектрограммы, полученной быстрым преобразованием Фурье, учитывать только спектральные пики (локальные максимумы в каждом столбце). Каждый из них учитывался при вычислении не одного, а нескольких компонент вектора, с разными весами, в зависимости от разницы между частотой пика и частотой ступени звукоряда. Кроме того, чтобы учесть наличие гармоник, каждый пик с частотой f_n прибавлялся к пикам с частотами $f_n, f_n/2, f_n/3, \dots$ с соответствующими весами. Такой вектор признаков получил название *гармонический профиль тональных классов* (harmonic pitch class profile).

В [99] и [53] были предложены способы перераспределения звуковой энергии в пределах спектрограммы, полученной быстрым преобразованием Фурье, от участков с меньшим количеством энергии к участкам с большим количеством энергии (эта техника была предложена в [54]). В [99] допускается только перемещение энергии в пределах одного столбца спектрограммы, в [53] допускается также перемещение энергии между столбцами. В полученных таким образом спектрограммах более чётко выделены горизонтальные участки с большим количеством звуковой энергии, соответствующие инструментам с определённой высотой звучания и их гармоникам.

В [78] каждый столбец X_m спектрограммы, полученной быстрым преобразованием Фурье, преобразуется аналогично (2.1) в вектор C'_m из 256 компонент, расположенных с шагом в $1/3$ полутона, что соответствует охвату в чуть более, чем 7 октав. После этого для 84 ступеней звукоряда от *ля* субконтроктавы (27.5 Гц) до *фа* третьей октавы (3322 Гц) генерируются шаблонные 256-компонентные векторы-столбцы. В каждом из них элементы, соответствующие ступени звукоряда и её гармоникам, задаются как h^{k-1} , где $h = 0.6$, а k – номер гармоники; остальные элементы равны 0. Взятые вместе, они образуют матрицу E . Далее линейным методом наименьших квадратов находится вектор C_m , минимизирующий $\|C'_m - EC_m\|$, при условии, что все компоненты C_m неотрицательны. Полученные векторы C_m образуют новую спектрограмму с шагом по частоте в $1/3$ полутона, которая обрабатывается как если бы она была получена в результате преобразования постоянного качества. Полученный в результате хроматический вектор признаков получил название *NNLS chroma* (Non-Negative Least Squares chroma).

Мощность звука определяется как энергия, передаваемая звуковой волной через рассматриваемую поверхность в единицу времени. Спектр мощности звука показывает изменение его мощности с течением времени. Он может быть получен из частотного спектра путем возведения в квадрат каждой из его компонент. Как показано в [37], воспринимаемая громкость звука приблизительно пропорциональна десятичному логарифму уровня мощности звука (sound power level). Поэтому имеет смысл перед преобразованием спектрограммы в последователь-

ность хроматических векторов заменить каждое её значение $C_m[n]$ на $\log(\eta \cdot C_m[n] + 1)$, где η – положительная константа, которая обычно выбирается из диапазона $100 \leq \eta \leq 10000$. Тогда соотношение между разными компонентами спектрограммы будет приблизительно соответствовать соотношению между воспринимаемыми человеком уровнями громкости соответствующих частот. Полученные таким образом признаки называют *chroma-log-pitch* (CLP) [10].

В [82] было предложено после логарифмирования элементов спектрограммы для каждого столбца C_m вычислять дискретное косинусное преобразование, занулять первые ξ полученных коэффициентов, после чего выполнять обратное дискретное косинусное преобразование. Если представлять спектр звука как функцию интенсивности звука от частоты, данное преобразование удаляет низкочастотные компоненты в спектре этой функции, т.е. длинные последовательности отличных от нуля значений, незначительно отличающихся друг от друга. Похожие действия выполняются при вычислении мел-частотных кепстральных коэффициентов [65], широко используемых в распознавании речи. Из полученной спектрограммы обычным образом вычисляются хроматические векторы. Они получили название *chroma DCT-reduced log pitch* (CRP). Целью этого преобразования является повышение устойчивости хроматических векторов к изменению тембра музыкальных инструментов, прежде всего для сопоставления различных музыкальных записей. Но CRP-признаки были успешно применены к распознаванию аккордов в [22].

В [43] было предложено наряду с зависимостью человеческого восприятия громкости от звуковой мощности учитывать зависимость от частоты звука. Для этого на каждом фрагменте звукозаписи вместо частотного спектра вычисляется спектр мощности, от каждой его компоненты вычисляется десятичный логарифм, после чего к каждой компоненте применяется А-взвешивание [95].

В [81] были предложены преобразования последовательности хроматических векторов, направленные на повышение устойчивости к шумам. В последовательности полученных обычным способом хроматических векторов каждый вектор B_m заменяется на $B_m / \|B_m\|_1$, где $\|B_m\|_1 = \sum_{n=0}^{N_0-1} |B_m[n]|$. Затем производится квантование значений $B_m[n]$, $0 \leq B_m[n] \leq 1$ с порогом, величины которых расположены логарифмически. Далее вычисляется свёртка последовательности $\{B_m\}_{m=0}^{M-1}$ с окном Ханна длины $w \in \mathbb{N}$, а затем прореживание полученной последовательности по основанию d . Полученные в результате этих преобразований векторы признаков получили название *chroma energy normalized statistics* ($CENS_d^w$).

В [45] были предложены особые признаки, не являющиеся хроматическими. Они являются векторами в пространстве *Tonnetz* [89], [21], моделирующем взаимоотношения между ступенями равномерно темперированного строя. Согласно [45], в случае равномерно темперированного строя это 6-мерное пространство. Для удобства векторы в этом пространстве нормируют так, чтобы они попадали внутрь 6-мерного эллипса с радиусами $(r_1, r_1, r_2, r_2, r_3, r_3)$. Координаты можно разделить попарно на 3 круга. Первый из них в некотором роде соответствует квинтовому кругу. В нём точки, соответствующие ступеням звукоряда, расположены на окружности радиуса r_1 с шагом $5\pi/6$. Во втором круге эти точки расположены на окружности радиуса r_2 с шагом $\pi/4$, а в третьем – на окружности радиуса r_3 с шагом $\pi/3$. Их можно мыслить как круги малых и больших терций соответственно. Точка, соответствующая аккорду, имеет координаты, равные среднему арифметическому координат составляющих его нот. Любой хроматический вектор может быть легко преобразован в вектор в этом пространстве. Такие векторы признаков были использованы в [59], [58], [25], [47].

Сравнение результатов от применения некоторых из описанных типов признаков к задаче распознавания аккордов было проведено в [10]. Наилучшие результаты были получены с использованием признаков CRP. Авторы отмечают, что логарифмическое преобразование спектра, применяемое при вычислении признаков CLP и CRP, является важным шагом к повышению качества распознавания аккордов.

Принципиально другой подход к получению вектора признаков был предложен в [47]. Описанные выше 6-мерные признаки получаются на выходе свёрточной нейронной сети [41], входом для которой являются фрагменты спектрограммы. При этом не применяются никакие знания о свойствах спектра или музыки. Предполагается, что нейронная сеть сама определит наиболее характерные свойства в процессе обучения.

На соревнованиях MIREX Audio Chord Estimation алгоритмы, использующие современные признаки ([53], [78], [82]) показывают близкие результаты. Но на результат в значительной степени влияет используемый в алгоритме метод классификации векторов признаков.

2.4 Классификация векторов признаков

На этом этапе находится решение задачи распознавания аккордов в звукозаписи: полученная на предыдущем этапе последовательность векторов признаков преобразуется в последовательность аккордов с указанием моментов начала и конца их звучания. Перед вычислением спектрограммы звукозапись была поделена на фрагменты, моменты начала и конца которых известны. Поэтому считается, что каждый из полученных векторов признаков соответствует промежутку времени между началами текущего и следующего фрагментов.

Для определения звучащего на данном фрагменте аккорда по вектору признаков необходимо классифицировать этот вектор. В рамках задачи MIREX Audio Chord Estimation 2012 выделялись 25 возможных классов: по одному классу для каждого мажорного и минорного аккордов, а также один класс для отсутствия аккорда. Многие алгоритмы также ограничиваются этим набором ([16], [57], [52], [85], [99], [23], [27], [22], [10], [43], [25], [47]). В некоторых работах выделяют также отдельные классы для доминантсептаккордов ([93], [77], [100], [79], [78], [31]), других септаккордов ([93], [78]), уменьшенных и увеличенных ([93], [28], [58], [77], [13], [79], [78], [98]) и других видов аккордов.

2.4.1 Метод ближайшего соседа

Наиболее простой способ классификации – определение расстояния от вектора признаков до «идеальных» шаблонных векторов той же размерности, соответствующих аккордам. В качестве результата выбирается аккорд, расстояние до шаблона которого является наименьшим. Фактически, это метод k ближайших соседей для $k = 1$. Такой подход был применён в [57], [85], в одном из вариантов [10]. Мерой расстояния может выступать косинусное расстояние, евклидово расстояние, расхождение Кульбака-Лейблера и другие. Их сравнение было проведено в [85]. В качестве шаблона аккордов часто используют вектор, у которого на позициях, соответствующих входящим в аккорд нотам, стоят 1, а на остальных – 0. Например, шаблон для аккорда до-минор имеет вид (1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0) (при условии, что первая компонента вектора соответствует звуку *do*).

Важным достоинством такого способа классификации является отсутствие этапа обучения. Отсюда следует лёгкость добавления новых типов распознаваемых аккордов: для этого требуется всего лишь добавить новые шаблоны. Недостатком является невозможность учесть зависимость между подряд идущими фрагментами звукозаписи.

Иногда (например, в [85]) в шаблоны также включают информацию о гармонических обертонах входящих в аккорд звуков. Звуки, соответствующие частотам гармонических обертонов, могут быть получены из формулы (1.2). Вклад обертона в соответствующую компоненту шаблона определялся в [40] и в [85] как

$$w_{harm}(k) = h^{k-1} \quad (2.2)$$

где k – номер обертона, а $h < 1$ – параметр. Основной тон звука здесь соответствует $k = 1$. Соответствующий шаблонный вектор будет иметь компоненты со значениями, отличными от 0 и 1.

Для повышения устойчивости к шумам к последовательности векторов признаков можно предварительно применить скользящий медианный фильтр или фильтр скользящего среднего, как в [57], [85].

В [79] было предложено учитывать структуру композиции перед определением аккордов. Структура может быть задана заранее или определена автоматически. Последовательности хроматических векторов, соответствующие одинаковым структурным сегментам, усреднялись перед распознаванием аккордов. Эта идея была продолжена в [22], где было предложено использовать метод рекуррентного анализа для нахождения похожих друг на друга последовательностей хроматических векторов и их взаимного сглаживания.

2.4.2 Скрытые марковские модели и байесовские сети

Широко используемые в методах распознавания речи *скрытые марковские модели* (СММ) [7] также нашли применение в алгоритмах распознавания аккордов. В отличие от метода ближайшего соседа, они позволяют в явном виде моделировать вероятность перехода между двумя заданными аккордами. Дадим формальное определение элементов СММ.

- Набор состояний модели $Q = \{Q_1, Q_2, \dots, Q_{N_{states}}\}$. За q_t будем обозначать состояние модели в момент времени t .
- Множество наблюдаемых символов $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{M_{symbols}}\}$.
- Матрица переходных вероятностей $\Omega = \{\omega_{ij}\}$, где $\omega_{ij} = P(q_t = Q_j | q_{t-1} = Q_i), 1 \leq i, j \leq N_{states}$. Если любое состояние достижимо из любого, то все ω_{ij} неотрицательны. Для всех $i, 1 \leq i \leq N_{states}$ верно $\sum_{j=1}^{N_{states}} \omega_{ij} = 1$.
- Распределение вероятностей появления наблюдаемых символов в состоянии $Q_j, V = \{v_j(k)\}$, где $v_j(k) = P\{\lambda_k \text{ at } t | q_t = Q_j\}$ при $1 \leq j \leq N_{states}, 1 \leq k \leq M_{symbols}$.
- Начальное распределение вероятностей состояний $\pi = \{\pi_i\}$, где $\pi_i = P\{q_t = Q_i\}, 1 \leq i \leq N_{states}$.

Состояния СММ ненаблюдаемы, в каждый момент времени доступен для наблюдения только какой-либо символ из множества Λ . Важным свойством СММ является то, что вероятность перехода из состояния Q_i в состояние Q_j не зависит от предыдущих состояний модели.

Набор состояний СММ фиксируется заранее. В качестве наблюдаемых символов обычно выступают векторы признаков. Матрица переходных вероятностей, параметры распределения вероятностей появления наблюдаемых символов и параметры начального распределения вероятностей состояний могут как задаваться изначально (как в [16], в одном из вариантов [87], в нескольких вариантах [23]), так и определяться в результате обучения СММ (как в [28], в нескольких вариантах [87], [77], [52], [80], в одном из вариантов [23], [10], [53], [43]). Вероятности появления наблюдаемых символов обычно моделируются одним многомерным нормальным распределением (как в [93], [16], [87], [43], [25]) или смесью многомерных нормальных распределений (как в [28], [52], [80], [23], [53]). При обучении обычно используется итеративный метод математического ожидания – модификации (expectation-modification), также называемый методом Баума-Уэлша или методом прямого-обратного хода. В [80] минимизируется ошибка классификации, параметры модели обновляются при помощи градиентного

спуска. При распознавании наиболее вероятной последовательности скрытых состояний применяется алгоритм Витерби. Стоит отметить, что иногда алгоритм Витерби применяют, не вводя явно СММ, а задавая псевдовероятности вместо необходимых в алгоритме распределений вероятностей (например, в [22], [47]).

Несмотря на свою популярность, СММ не свободны от недостатков, ограничивающих возможность их применения. Основными из них являются очень большое количество параметров и марковское свойство, позволяющее учитывать зависимость состояния на данном шаге от состояния только на предыдущем шаге.

Обычно наблюдаемыми символами СММ являются хроматические векторы. Соответственно, вероятности появления наблюдаемых символов моделируются многомерными распределениями с числом измерений, равным размерности хроматического вектора. Оценим число параметров для типичного случая. Пусть СММ имеет $N_{states} = 25$ состояний, каждому из которых соответствует одно 12-мерное нормальное распределение, а начальное распределение вероятностей состояний равномерно. Тогда имеется $25 \cdot 24 = 600$ элементов в матрице переходных вероятностей, а также как минимум 24 параметра на каждое из 25 состояний (в предположении, что матрицы ковариации многомерных нормальных распределений диагональны). С использованием смеси нормальных распределений вместо одного распределения количество параметров для каждого состояния увеличивается пропорционально числу компонентов смеси.

Для уменьшения количества настраиваемых параметров часто предполагают, что параметры для разных состояний в некотором смысле схожи, а потому могут быть скорректированы после первоначального обучения. В хроматическом векторе каждая компонента соответствует одному классу звуков, например, всем звукам *до*. Если его первую компоненту такого вектора, соответствующую классу звуков *до*, переставить в конец, то полученный вектор останется хроматическим, но его первая компонента будет соответствовать классу звуков *до-диез*. Аналогичные циклические перестановки возможны для математических ожиданий и матрицы ковариации соответствующего многомерного распределения.

Так, если в векторе математических ожиданий для распределения, соответствующего аккорду *до-диез-мажор*, переставить одну компоненту из начала в конец, то полученный вектор математических ожиданий будет соответствовать аккорду *до-мажор*. Такими сдвигами можно привести все векторы математических ожиданий для распределений, соответствующих мажорным аккордам, к виду, в котором компонента, соответствующая основному звуку аккорда, будет первой. После этого можно усреднить все математические ожидания по всем аккордам, и обратными сдвигами вернуть усреднённые векторы математических ожиданий на свои места. Аналогично можно усреднить матрицы ковариации для всех аккордов одного типа. Также возможно усреднение компонентов матрицы переходов для случаев переходов между аккордами соответствующих типов, основные звуки которых отстоят на одинаковое число полутонов. Процедура усреднения применяется, например, в [93], [87], [23], [53]. Усреднение параметров модели полезно в случае недостатка обучающих данных или их неравномерного распределения по рассматриваемому набору аккордов.

Моделирование зависимости текущего состояния модели только от состояния на предыдущем шаге приводит к заметной проблеме. Очевидно, что смена аккорда производится не при каждой смене звукового фрагмента. Поэтому необходимо контролировать длительность нахождения модели в одном состоянии. В случае СММ первого типа это можно сделать, регулируя значения на главной диагонали матрицы переходов. А в [25] в СММ было дополнительно введено распределение, задающее вероятность нахождения модели в состоянии Q_i в течение d фрагментов, где $d \leq 20$. Процедура обучения и алгоритм Витерби были соответствующим образом модифицированы.

Другой подход к моделированию длительности нахождения СММ в одном состоянии – построение отдельной модели для каждого аккорда и связывание этих моделей в одну СММ с

общими входом и выходом для каждой из моделей. Он применялся в одном из вариантов [28], [77], [52], [53]. В этом случае можно регулировать параметры моделей каждого отдельного аккорда (как в [77]), а также добавлять штраф за переход от модели одного аккорда к модели другого аккорда (как в [52], [53]).

Были предложены различные способы для учёта информации о предыдущих состояниях модели в том числе через введение понятий жанра и тональности. В [52] использовалась языковая модель, которая позволяет учитывать более чем одно предыдущее состояние СММ. В [59] было предложено строить 24 СММ, по одной для каждой из мажорных и минорных тональностей. При распознавании аккордов для каждой модели определялась наиболее вероятная последовательность состояний. В качестве результата выбиралась та из последовательностей, вероятность которой была наибольшей. Дополнительным результатом при этом было определение тональности композиции. В [58] аналогичным образом строились отдельные СММ для 6 различных музыкальных жанров. В [98] отдельные СММ строились для 11 различных жанров, но при этом они были объединены в одну гипер-жанровую модель с более сложной процедурой обучения. Несмотря на большой потенциал такого рода комбинаций, они требуют существенно больше обучающих данных. В случае [59] и [58] использовались звукозаписи, сгенерированные из MIDI-файлов. В [98] использовался достаточно большой набор реальных музыкальных звукозаписей. Тональность может быть явным образом введена в саму СММ наряду с басовой нотой. Предложенная в [43] СММ включала в себя в том числе 12 скрытых состояний для текущей басовой ноты и 24 скрытых состояний для текущей тональности. При этом общее количество комбинаций скрытых состояний становится слишком большим, поэтому приходится накладывать дополнительные ограничения на допустимые переходы между аккордами и между тональностями и на допустимые сочетания аккордов и басовых нот.

В [78] было предложено использовать динамическую байесовскую сеть, которая, по сути, является обобщением СММ (см. [38]). В ней используются скрытые состояния для текущих метрической позиции, тональности, аккорда и басовой ноты; наблюдениями являются 2 вектора хроматических признаков: для высоких и для низких частот. Такая модель позволяет моделировать сложные музыкальные взаимоотношения. С другой стороны, она имеет множество параметров, и поэтому требует большего количества обучающих данных. Для получения наиболее вероятной последовательности в такой сети можно использовать модификацию алгоритма Витерби, но из-за размеров сети этот процесс оказывается более длительным, чем в случае СММ.

2.4.3 Другие модели

В [99] было предложено использовать более сильный алгоритм классификации, чем метод ближайшего соседа, основанный на методе опорных векторов. Помимо текущего вектора признаков этот алгоритм позволяет учитывать также признаки на предыдущем или на следующем фрагменте звукозаписи, а также попарные произведения компонент вектора признаков.

В одном из вариантов [28] было предложено заменить СММ на условное случайное поле [55]. Оно определяется следующим образом. Обозначим за \mathbf{X} и \mathbf{Y} множество наблюдений и множество случайных переменных соответственно. Пусть $G = (V, E)$ – такой граф, что $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$, то есть \mathbf{Y} можно проиндексировать вершинами этого графа. Тогда (\mathbf{X}, \mathbf{Y}) называется *условным случайным полем*, если случайные переменные \mathbf{Y}_v при условии \mathbf{X} удовлетворяют марковскому свойству с учётом графа: $p(\mathbf{Y}_v | \mathbf{X}, \mathbf{Y}_w, w \sim v) = p(\mathbf{Y}_v | \mathbf{X}, \mathbf{Y}_w, w \sim v)$, где $w \sim v$ означает, что w и v являются соседями в графе G . В случае, когда G является цепью или деревом, к соответствующему условному случайному полю можно применять алгоритмы, аналогичные методу прямого-обратного хода и алгоритму Витерби. В отличие от СММ, при определении наиболее вероятной последовательности вершин графа максимизируется не $p(\mathbf{X}, \mathbf{Y})$, а $p(\mathbf{Y} | \mathbf{X})$. Кроме того, в такой модели каждое скрытое состояние зависит не только

от текущего наблюдения, но от всей предыдущей последовательности наблюдений. В [28] отмечается, что условное случайное поле обучается существенно дольше, чем СММ.

В [31] была предложена модель тональной гармонии, построенная на основе музыкально-теоретических соотношений между аккордами. Её применение требует знания тональности, поэтому для звукозаписи предварительно определяется последовательность тональностей с ограничением на минимальную длину фрагмента в одной тональности в 16 метрических долей. На каждом фрагменте звука определяется набор наиболее вероятных аккордов (вычисляются расстояния от хроматического вектора до шаблонов аккордов), после чего модель гармонии используется для определения наиболее вероятной последовательности аккордов с учётом уже определённых аккордов на всех предыдущих фрагментах.

В [14] использовался собственный алгоритм для определения вероятности гипотез. Каждая гипотеза состоит из последовательности аккордов, определённой до данного фрагмента, и тональности. На каждом фрагменте определяется вероятность гипотез со всеми возможными вариантами текущего аккорда. В формуле для вычисления вероятности гипотезы учитываются тональность, вероятность смены аккорда, хроматический вектор, басовый звук, сочетаемость аккорда и басового звука. Очень похожий подход с другими формулами для определения вероятности гипотез был применён в [13].

Подход, в чём-то похожий на алгоритм Витерби, был предложен в [27]. Здесь на каждом фрагменте определяется набор наиболее вероятных аккордов (вычисляются расстояния от хроматического вектора до шаблонов аккордов) и тональностей (вычисляются расстояния от хроматического вектора до шаблонных векторов тональностей из [96]). Затем все наиболее вероятные кандидаты объединяются в пары. Расстояние между парами (аккорд, тональность) определяется в соответствии с [62] на основе взаимоотношений между звуками, составляющими аккорды, и звуками, входящими в тональности. Тогда методом динамического программирования можно определить последовательность пар (аккорд, тональность) по всем фрагментам, имеющую наименьшую сумму расстояний между соседними парами.

2.5 Выводы

1. В большинстве современных алгоритмов для распознавания аккордов используются методы определения ритма; методы определения частоты настройки музыкальных инструментов применяются реже.
2. Среди множества типов хроматических признаков наилучшее качество распознавания достигается при помощи тех из них, алгоритмы вычисления которых обеспечивают подавление шумовых спектральных компонент.
3. Скрытые марковские модели и динамические байесовские сети являются наиболее популярными методами классификации в алгоритмах распознавания аккордов. Эти методы позволяют добиться наилучшего качества распознавания, но требуют настройки очень большого количества параметров в процессе обучения.

Глава 3

Распознавание аккордов без использования машинного обучения

В этой главе описывается метод определения последовательности аккордов в звукозаписи, не требующий предварительного обучения. Глава разделена на параграфы в соответствии с основными реализованными в данной работе улучшениями существовавших ранее алгоритмов. В параграфе 3.1 описываются предварительная обработка и получение спектрограммы звукозаписи. Преобразование спектрограммы в векторы признаков и обработка последовательности векторов признаков описаны в параграфах 3.2 и 3.3. Наконец, используемый метод классификации и эвристики, предназначенные для уменьшения количества ошибок классификации, описаны в параграфе 3.4.

3.1 Частотно-временное представление звукозаписи

Западная музыка основывается на равномерно темперированном строе. Поэтому, как правило, все звуки, издаваемые музыкальными инструментами с определённой высотой звучания, имеют частоты, соответствующие формуле 1.1. В звучании аккорда, состоящего из нескольких нот, большая часть энергии должна приходиться на частоты этих нот. Соответственно, разница в звучании двух аккордов должна выражаться в наличии или отсутствии звуковой энергии на определённых частотах. Поэтому при построении частотно-временного представления наибольший интерес представляют частоты, соответствующие звукам западной музыкальной системы.

3.1.1 Определение частоты настройки музыкальных инструментов

В формуле 1.1 присутствует параметр f_0 , задающий частоту настройки музыкальных инструментов. Как отмечается в [61, с. 89], некоторые оркестры до сих пор используют частоты настройки 442 Гц и 443 Гц. Встречающееся гораздо чаще воспроизведение звукозаписи с изменённой скоростью приводит к аналогичному эффекту, повышая или понижая частоты звучания всех инструментов композиции. Частоту настройки необходимо определить предварительно, чтобы избежать ошибок на таких звукозаписях.

Увеличение частоты настройки в $2^{1/12} \approx 1.06$ раз (или примерно на 6%) приведёт к повышению звучания инструмента на полутон: вместо звука *си* будет звучать *до*, вместо *до* – *до#* и так далее. Аналогично, повышение скорости воспроизведения в $2^{1/12}$ раз приведёт к уменьшению периода каждого звука в $2^{-1/12}$ раз, а значит, к повышению частоты в $2^{1/12}$ раз. Очевидно, в случае такого изменения скорости воспроизведения невозможно обнаружить сам факт его наличия, не обладая дополнительной информацией об изначальной тональности ком-

позиции. Поэтому обычно фиксируют диапазон для возможных значений частоты настройки: $[440 \cdot 2^{-1/24}, 440 \cdot 2^{1/24})$, приблизительно соответствующий диапазону от 427 до 452 Гц.

Обзор некоторых алгоритмов определения частоты настройки можно найти в [60] в разделе 4.1. В рамках данной работы используется алгоритм, похожий на предложенный в [101], основанный на определении количества звуковой энергии, приходящейся на узкие частотные полосы, соответствующие разным возможным значениям отклонения частоты настройки от стандартной. Звукозапись делится на короткие фрагменты между моментами времени $t_m, m = 0, 1, 2, \dots, M'$, на каждом из которых выполняется constant- Q преобразование с $f_{min} = 440 \cdot 2^{m_0/12}$ для некоторого целого m_0 , и достаточно высоким разрешением по частоте: $N_0 = 12b_0$ компонент на октаву. В каждом фрагменте $C_m = C(t_m)$ определяется номер компоненты $C_m[n]$, $0 \leq n < N'$, которой соответствует максимальное значение спектра. Затем строится гистограмма значений функции $C_m[n]$, она состоит из N' столбцов. Значения всех столбцов, номера которых сравнимы по модулю b_0 , суммируются. В полученной гистограмме из b_0 столбцов номер столбца с наибольшим значением можно интерпретировать как отклонение f_0 от стандартной частоты настройки 440 Гц в диапазоне от $-1/2$ до $+1/2$ полутона с точностью до $1/b_0$ полутона. Если наибольшее значение приходится на 0-й столбец, то отклонения нет. Используемые здесь значения M' и N' не обязательно совпадают с соответствующими значениями M и N для основной спектрограммы.

Допустим, вместо настоящей частоты настройки f_0 была ошибочно определена $f'_0 \neq f_0$. Это приведёт к тому, что настоящие частоты звуков будут отличаться от использованных в преобразовании постоянного качества в f_0/f'_0 раз. Если этот множитель незначительно отличается от 1, разница не будет заметна. В противном случае возможно определение аккордов не в той тональности.

Влияние алгоритма определения частоты настройки на качество распознавания аккордов рассматривается в параграфе 5.2.3

3.1.2 Определение ритма

Ритм играет важную роль в западной музыке. Так же, как равномерно темперированный строй упорядочивает звуки по высоте, ритм упорядочивает и группирует их по времени начала и продолжительности звучания. Поэтому и смена звучащего аккорда должна происходить в соответствии с ритмом. Наиболее чётко воспринимаемая человеком пульсация соответствует периодической смене метрических долей. В дальнейшем будем предполагать, что смена звучащего аккорда всегда происходит в момент начала какой-то метрической доли. При этом теряется возможность определения нескольких аккордов в пределах одной метрической доли. Но расположенные в соответствии с ритмом анализируемые фрагменты позволяют анализировать звук ровно в те моменты, когда музыкальные инструменты звучат наиболее ярко, и интересующие нас частоты лучше выделены в спектре.

В рамках данной работы для определения ритма в звукозаписях использовались 2 внешние библиотеки: *Beatroot* [32] и *Beat tracker* [30] из набора *Queen Mary Vamp plugins*. Вторая библиотека потребовалась для обработки тех композиций, в которых *Beatroot* не смог определить начала метрических долей. Зависимость качества распознавания аккордов от выбора библиотеки для определения ритма рассматривается в параграфе 5.2.1.

3.1.3 Снижение влияния ударных инструментов

В звучании любого музыкального инструмента можно условно выделить 3 части: атака, стационарная часть, затухание. В процессе атаки в музыкальном инструменте устанавливаются колебания, начинают звучать основной тон и обертоны. На протяжении стационарной части

звучание меняется слабо. В процессе затухания колебания прекращаются. В [1] приводятся следующие цифры для основных категорий музыкальных инструментов:

- струнные щипковые и ударные (гитара, фортепиано): атака 10-50 мс, нет стационарной части, продолжительное затухание;
- струнные смычковые (скрипка): атака до 50 мс, продолжительная стационарная часть;
- орган: атака до 300 мс, продолжительная стационарная часть, продолжительное затухание;
- духовые (труба): атака 10-30 мс, выраженный стационарный участок, короткое затухание;
- ударные (барабаны, тарелки): атака 3-10 мс, продолжительное затухание.

Как видно, ударные инструменты начинают звучать раньше остальных. Поскольку ритм обычно задаётся именно ударными инструментами, имеет смысл анализировать спектр в моменты времени, отстоящие на несколько десятков миллисекунд от моментов начала метрических долей. В эти моменты звучание инструментов с выраженной высотой будет наиболее ярким и полным, в то время как ударные инструменты будут находиться в процессе затухания. Обозначим соответствующую задержку во времени за d .

Очевидно, введение такой задержки имеет смысл только совместно с использованием алгоритма для определения ритма. Автору неизвестно о других алгоритмах распознавания аккордов, в которых применялась бы такая задержка. Влияние величины задержки на качество распознавания аккордов анализируется в параграфе 5.2.2.

3.1.4 Получение спектра

Моменты начала метрических долей $(t_0, t_1, \dots, t_{M-1})$ (с учётом смещения) и частоты звуков равномерно темперированного строя образуют сетку на плоскости «частота-время». Особый интерес представляют значения интенсивности звука, вычисленные в узлах этой сетки. Информация о моментах начала метрических долей позволяет разделить звукозапись таким образом, чтобы на каждый из них приходилась середина одного из фрагментов. Преобразование постоянного качества позволяет в каждом фрагменте определить интенсивность звука для каждой из указанных частот.

Во многих работах (например, в [10], [22]) отмечалась важность сглаживания последовательности столбцов спектрограммы или векторов признаков. Сглаживание осуществляется путём применения фильтра скользящего среднего или скользящего медианного фильтра с шириной окна w к каждой строке спектрограммы. Оно позволяет избавиться от единичных выбросов в спектре, но при этом несколько размывает спектр, снижая разрешение по времени. Если каждый столбец спектра соответствует промежутку между двумя метрическими долями, такое размытие будет слишком сильным.

Чтобы преодолеть этот недостаток, увеличим разрешение спектрограммы по времени в T раз путём вставки между каждыми моментами (t_m, t_{m+1}) перед получением спектрограммы равномерно $T - 1$ промежуточных значений, где T – параметр. Тогда появляется возможность использовать достаточно большой размер окна при сглаживании, не приводящий к существенному размытию спектра во времени. После сглаживания разрешение спектрограммы уменьшается в T раз путем удаления добавленных промежуточных столбцов.

Равномерно темперированный строй предполагает расположение $N_0 = 12$ ступеней звукоряда в пределах октавы, поэтому удобно выбирать N_0 кратным 12. Большие значения N_0

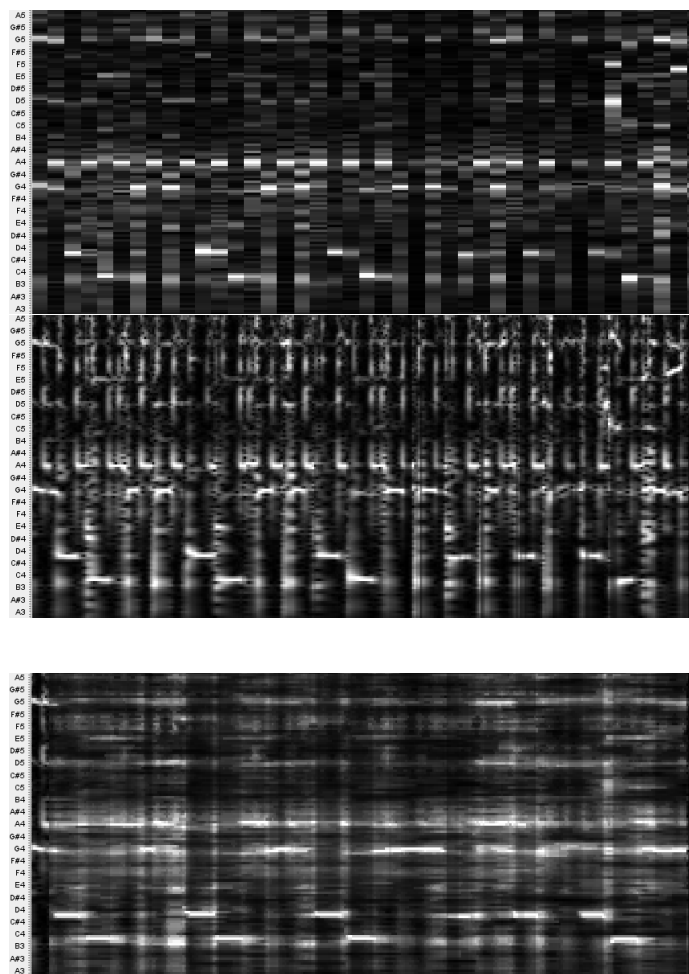


Рисунок 3.1: Фрагменты спектрограммы *The Beatles – Love Me Do* при $T = 1$ (вверху), $T = 8$ (в середине), $T = 8$ после сглаживания с $w = 19$ (внизу).

дают возможность в некоторой мере скомпенсировать ошибки при определении частоты настройки музыкальных инструментов f_0 , позволяя учесть близкие к $f_k = 2^{k/12}f_0$ частоты. В [16], [57], [77], [85], [23], [22] использовалось значение $N_0 = 36$.

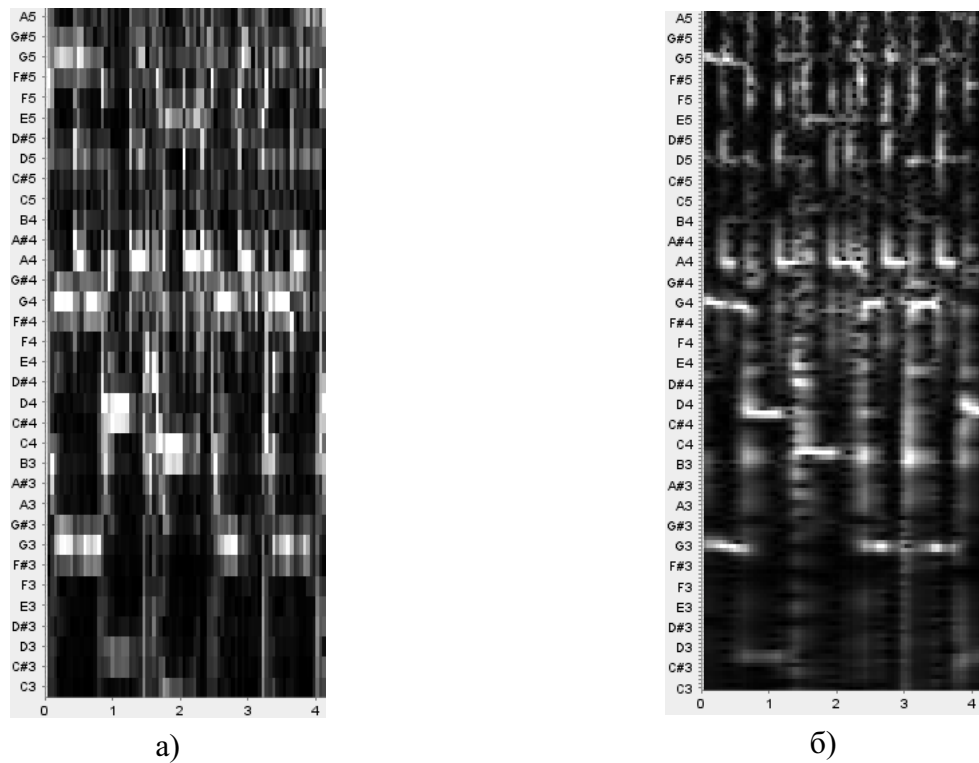


Рисунок 3.2: Фрагменты спектрограммы *The Beatles – Love Me Do* при: а) $N_0 = 12$; б) $N_0 = 60$.

Важно также правильно выбрать частоту наименьшей из компонент преобразования f_{min} и общее количество компонент N . Они определяют используемый для анализа частотный диапазон. Обычно используются частоты в пределах 50-2000 Гц. В [99] используемый диапазон ограничен сверху 1000 Гц, а в [22] – 4186 Гц. Для определения более частот ниже 50 Гц требуется слишком длинный фрагмент звука, а частоты выше 2000 Гц обычно содержат только гармоники более низких нот, затрудняющие определение аккорда.

В параграфе 5.2.4 анализируется влияние параметров T , w и N_0 на качество распознавания аккордов.

3.2 Выделение мелодических компонент спектра и векторы признаков

На этом этапе к спектрограмме применяется серия преобразований. Они нацелены на акцентирование компонент, которые несут важную для идентификации аккорда информацию, и на подавление остальных компонент. Наиболее важным является подавление шума и инструментов с неопределенной высотой звучания, поскольку их спектр не зависит от звучащего аккорда и сопоставим по уровню со спектром инструментов, задающих аккорд.

Как видно из рисунка 3.1 (в середине), барабан оставляет на спектрограмме яркие вертикальные полосы. В то же время, гитаре соответствуют горизонтальные полосы. Это свойство используют алгоритмы разделения звука на гармонические и перкуссионные компоненты, такие как [92] и [36]. Для распознавания аккордов полное разделение является излишним, необходимо только подавить перкуссионные компоненты.

Маух в [78] предложил вычитать из спектрограммы так называемый фоновый спектр. При этом каждое значение спектрограммы $C_m[n]$ заменять на $\frac{C_m[n] - \mu_m[n]}{\sigma_m^q[n]}$, где $\mu_m[n]$ представляет собой среднее значение, а $\sigma_m^q[n]$ – среднеквадратическое отклонение в пределах отрезка от $C_m[n - k]$ до $C_m[n + k]$, охватывающего одну октаву, $q \in \{0, 1\}$. Если полученное значение является отрицательным, вместо него подставляется 0.

Автором в [39] было предложено использовать аналог фильтра Превитт, используемого в обработке изображений для выделения границ. Будем для каждого фрагмента спектрограммы размера 9×3 с центром в точке $C_m[n]$ вычислять его свёртку с матрицей

$$P = \begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Если полученное значение больше 0, то заменим $C_m[n]$ на него, иначе – на 0.

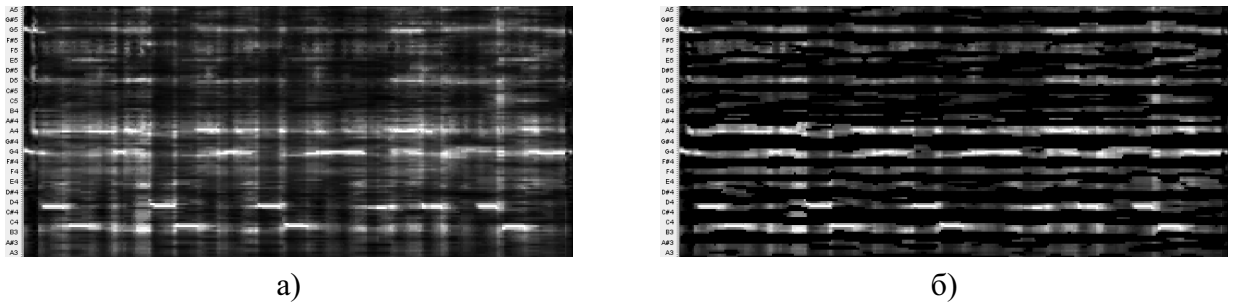


Рисунок 3.3: Фрагменты спектрограммы *The Beatles – Love Me Do*: а) до применения фильтра Превитт; б) после применения фильтра Превитт.

Еще один подход к подавлению перкуссионных компонент лёг в основу алгоритма вычисления признаков CRP [82]. Будем рассматривать $C_m[n]$ как сигнал (количество энергии, приходящейся на данную частоту, в зависимости от частоты). Применим к этой функции дискретное косинусное преобразование.

$$DC_m[k] = \sum_{j=0}^{N-1} C_m[j] \cos \left[\frac{\pi}{N} \left(j + \frac{1}{2} \right) k \right], \quad k = 0, \dots, N-1$$

В полученной последовательности значений занулим первые ξ значений, после чего произведём обратное дискретное косинусное преобразование. Зануляемые первые коэффициенты соответствуют низкочастотным компонентам сигнала $C_m[n]$, которые, в свою очередь, соответствуют достаточно длинным последовательностям существенно отличных от нуля значений. При этом имеющиеся в функции «острые» пики выделяются более чётко.

Как показывает практика, важным шагом является применение к спектрограмме логарифмического преобразования: каждая компонента $C_m[n]$ заменяется на $\log_{10}(\eta C_m[n] + 1)$. После него соотношения между компонентами спектрограммы лучше соответствуют человеческому восприятию интенсивности звука.

В параграфе 5.3.1 приводится сравнение результатов от применения различных методов очистки спектра. Параграф 5.3.2 посвящён подбору наилучших параметров для вычисления признаков CRP.

3.3 Применение самоподобия

Важным свойством музыкальных звукозаписей является наличие повторений. Музыка нравится человеку в том числе из-за повторений одного и того же мотива в разных вариациях, с некоторыми изменениями. Во многих композициях имеется достаточно продолжительный повторяющийся припев. В рамках куплета может повторяться одна и та же музыкальная фраза длительностью в несколько тактов. Можно попытаться использовать повторения для улучшения спектрограммы.

В работах [78] и [22] повторяющиеся фрагменты композиции использовались для улучшения качества распознавания аккордов. В обоих методах строились матрицы самоподобия для 12-мерных хроматических векторов признаков с использованием в качестве меры подобия коэффициента корреляции Пирсона (в [78]) и евклидова расстояния (в [22]). В полученной матрице находятся линии, параллельные главной диагонали, которые соответствуют похожим друг на друга фрагментам. Эти фрагменты затем используются для дополнительного сглаживания спектрограммы.

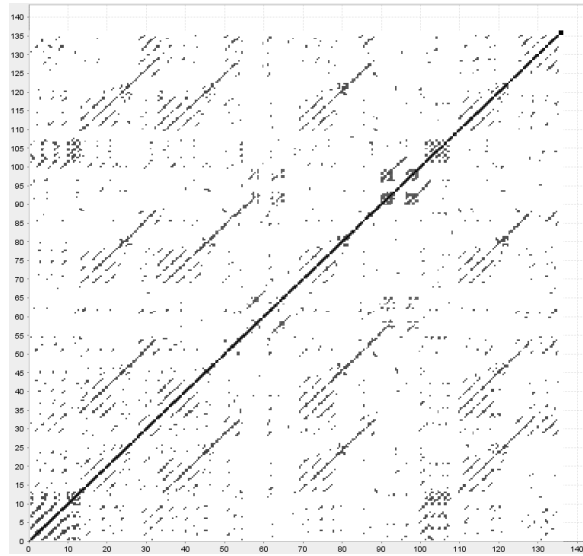


Рисунок 3.4: Матрица самоподобия для композиции *The Beatles – Love Me Do*

Однако матрицу самоподобия можно строить и для столбцов спектрограммы $\{C_i\}_{i=0}^{M-1}$, каждый из которых содержит больше информации по сравнению с соответствующим вектором признаков. Обозначим эту матрицу за $\{s_{ij}\}$, где s_{ij} – евклидово расстояние между столбцами C_i и C_j . Эта матрица имеет нули на главной диагонали. Нормализуем её таким образом, чтобы $0 \leq s_{ij} \leq 1$ для всех i, j . Затем в каждой строке сохраняются $\zeta \cdot M$ наименьших значений ($0 \leq \zeta \leq 1$), а все остальные заменяются на 1. Пример полученной матрицы показан на рисунке 3.4.

При помощи полученной матрицы можно скорректировать все столбцы C_m :

$$\hat{C}_m = \frac{\sum_{j=0}^{M-1} (1 - s_{mj}) C_j}{\sum_{j=0}^{M-1} (1 - s_{mj})}$$

На рисунке 3.5 наглядно показан эффект от такой коррекции. Спектрограмма становится существенно чётче, моменты смены аккордов становятся выделенными более явно. Количественная оценка влияния коррекции на качество распознавания аккордов приведена в параграфе 5.3.3.

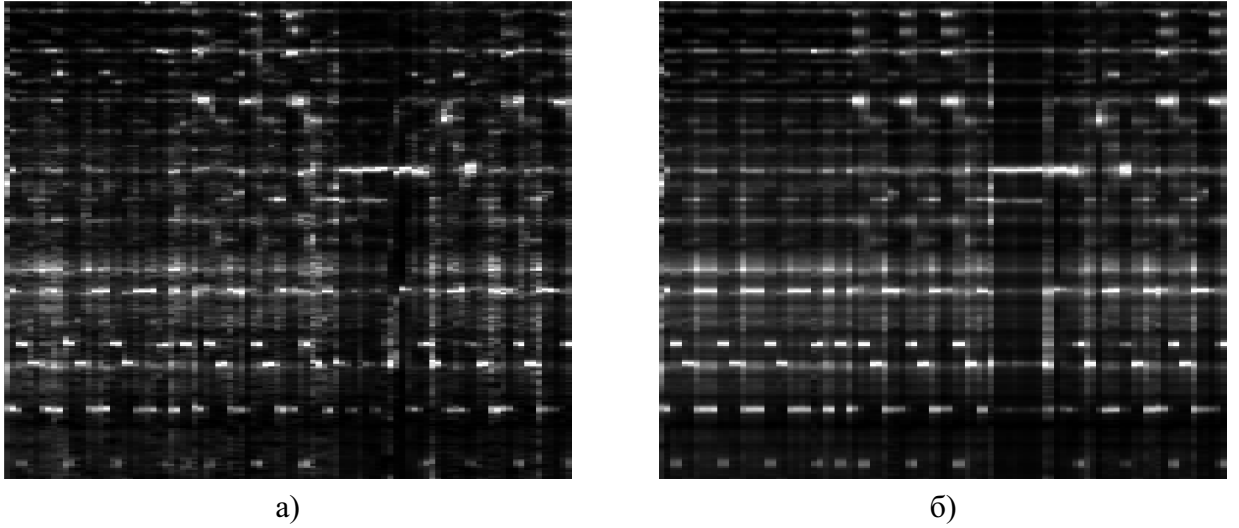


Рисунок 3.5: Фрагменты спектрограммы *The Beatles – Love Me Do*: а) без использования самоподобия; б) после коррекции с использованием самоподобия.

Важно, что применение этого преобразования требует наличия достаточно большого количества столбцов в спектрограмме. Вряд ли имеет смысл применение самоподобия к короткой звукозаписи, в которой маловероятно наличие повторяющихся элементов.

3.4 Классификация и исправление ошибок

3.4.1 Классификация хроматических векторов

Будем рассматривать в качестве множества возможных названий аккордов Y набор из названий 24 мажорных и минорных аккордов и символа «N», означающего отсутствие аккорда. За основу возьмём метод ближайшего соседа с шаблонами, учитывающими основной тон и 3 первых обертона по формуле (2.2). Эти шаблоны задаются для 12-мерных хроматических векторов. Столбцы спектрограммы охватывают несколько октав, поэтому для каждого из них потребовалось бы несколько шаблонов, чтобы учесть все возможные сочетания октав, в которых могут располагаться ноты аккорда.

Для получения 12-мерных хроматических векторов сначала ко всем значениям $\hat{C}_m[n]$, $0 \leq n < N_0$, прибавляются значения $\hat{C}_m[n + N_0]$, $\hat{C}_m[n + 2N_0]$, $\hat{C}_m[n + 3N_0]$, ... для каждого $0 \leq m \leq M - 1$, давая в результате последовательность N_0 -мерных векторов $\{B_m\}_{m=0}^{M-1}$. Далее в случае $N_0 = 12b_0$, $b_0 \geq 3$ каждый вектор B_m преобразуется в 12-мерный вектор D_m :

$$D_m[n] = B_m[b_0n - 1] + B_m[b_0n] + B_m[b_0n + 1], \quad m = 0, \dots, M - 1, \quad n = 0, \dots, 11$$

Для вычисления $D_m[0]$ в качестве $B_m[-1]$ используются $B_m[12b_0 - 1]$.

Для каждого из последовательности векторов $\{D_m\}_{m=0}^{M-1}$ определяется ближайший из шаблонов, и соответствующий ему аккорд считается аккордом, звучащим в данном фрагменте.

Параграф 5.5.1 посвящён подбору наилучших параметров для используемых шаблонов.

3.4.2 Определение отсутствия звучащего аккорда

Участки, на которых отсутствует звучащий аккорд, не обязательно являются тишиной. Это может быть также соло на ударных инструментах или какой-либо шум. Поэтому нельзя определять отсутствие аккорда как близость столбца спектрограммы или вектора признаков к

нулевому вектору. Необходимо каким-то образом учитывать информацию о наличии и выраженности звуков, соответствующих ступеням звукоряда.

Будем вычислять для каждого столбца спектрограммы следующие величины:

1. отношение максимальной по значению из компонент к сумме значений всех компонент;
2. отношение суммы значений компонент, частоты которых в точности соответствуют частотам ступеней звукоряда, к сумме значений всех компонент.

После чего будем определять отсутствие звучащего аккорда на фрагменте звукозаписи, соответствующем данному столбцу спектрограммы, если произведение указанных двух величин будет меньше некоторого порогового значения L_N . Кроме того, будем считать, что аккорды отсутствуют на участках до первой и после последней из определённых ранее метрических долей. Влияние параметра L_N на качество распознавания аккордов рассматривается в параграфе 5.5.3.

3.4.3 Исправление ошибок классификации

В результате экспериментов было обнаружено, что некоторые последовательности аккордов являются маловероятными в реальной композиции, и скорее всего является ошибочными. Для двух классов таких последовательностей предлагается метод их исправления.

A:maj - A:min

К первому классу относятся последовательности, в которых аккорды имеют общую основную ноту, но различные типы, например: A:maj-A:min-A:maj-A:min. Появление таких последовательностей возможно, поскольку соответствующие векторы признаков достаточно близки друг к другу. Для каждой такой последовательности находится вектор признаков, являющийся средним арифметическим составляющих её векторов. Аккорд, соответствующий полученному вектору признаков, приписывается всей последовательности.

A-B-C

Ко второму классу относятся последовательности из 3 разных идущих подряд аккордов: A-B-C (при этом возможно A=C). В этом случае более вероятно, что на самом деле имел место один из следующих 4 вариантов: A-A-C, A-C-C, A-B-B, B-B-C. Из них выбирается тот, для которого сумма расстояний от векторов признаков до соответствующих шаблонных векторов минимальна. Очевидно, что такая коррекция будет ошибочной в тех случаях, когда аккорд действительно звучит только в течение одной метрической доли.

Эффект от использования предложенных эвристик исследуется в параграфе 5.5.2.

3.5 Выводы

1. Разработан метод более точного вычисления спектрограммы, учитывающий свойства музыкальных звукозаписей.
2. Предложен метод для удаления шумовых компонент спектра на основе фильтра Превитт, используемого в обработке изображений.
3. Предложен набор преобразований последовательности векторов признаков, учитывающий свойства музыкальных звукозаписей.
4. Предложены эвристики для исправления распространённых ошибок классификации.

Глава 4

Получение признаков с использованием нейронных сетей

Описанный в главе 3 метод получения векторов признаков из столбцов спектрограммы состоит из нескольких преобразований, каждое из которых опирается на какие-то свойства музыкальных звукозаписей. Представление спектра звука в виде вектора признаков необходимо, чтобы облегчить последующую классификацию. Обучение представлением – это раздел машинного обучения, рассматривающий алгоритмы, направленные на получение наилучших представлений входных данных. Такие алгоритмы стремятся сохранить наиболее характерные признаки входных данных в сжатом их представлении.

В основе многих алгоритмов обучения представлением лежит многослойная нейронная сеть, при обучении которой используются так называемые методы глубокого обучения. Важным свойством таких методов является возможность предварительного обучения каждого слоя нейронной сети в отдельности без учителя, на неразмеченных данных. Во время предварительного обучения параметры каждого из слоёв достигают значений, более близких к оптимальным, чем при их случайной инициализации. Благодаря этому для окончательного обучения нейронной сети в целом требуется существенно меньше размеченных данных, чем потребовалось бы без использования предварительного обучения. В обзорной статье Й. Бенджио [17] подробно описываются техники, лежащие в основе методов глубокого обучения.

Методы обучения представлением впервые были применены к распознаванию аккордов в звукозаписях в 2012 году: Хамфри в [47] предложил использовать свёрточные нейронные сети для получения из спектрограммы признаков, позволяющих классифицировать звучащий аккорд. Этот тип нейронных сетей успешно применяется при анализе изображений. Фактически, при распознавании аккордов звуковые данные первоначально также преобразуются в изображение – спектрограмму. Также свёрточные нейронные сети применялись для классификации звуков музыкальных инструментов [49].

Многослойные рекуррентные нейронные сети были успешно использованы для распознавания речи в [74] и для распознавания аккордов в [18], где нейронная сеть возвращает на выходе распознанный аккорд, который при помощи рекуррентного соединения подаётся на вход на следующем шаге.

Хамфри и др. в [48] отмечают, что из-за своей новизны и сложности методы обучения представлением и глубокого обучения только начинают использоваться в различных задачах музыкального информационного поиска. Эти методы потенциально могут улучшить внутренние представления звуковых данных, используемые в различных задачах. При этом, в отличие от изображения, звук меняется с течением времени, что потенциально может быть использовано при разработке соответствующих алгоритмов.

В данной работе рассматриваются обычные многослойные (в том числе рекуррентные) нейронные сети, предварительно обучаемые с помощью очищающих автоассоциаторов. В

разделе 4.1 даётся определение многослойного очищающего автоассоциатора и сопутствующих понятий. В разделе 4.2 описывается построение и обучение многослойной нейронной сети с использованием автоассоциаторов, преобразующей столбец спектрограммы в вектор хроматических признаков.

4.1 Теоретические сведения

Определения в этом разделе даны в соответствии с [86].

Автоассоциатор (автоэнкодер) представляет из себя пару преобразований:

$$y = f_{\theta}(x) = s(Wx + b) \quad (4.1)$$

$$z = g_{\theta'}(y) = s(W'y + b') \quad (4.2)$$

Здесь x – входной вектор, z – реконструированный выходной вектор, y – *внутреннее представление* для x , $\theta = \{W, b\}$ и $\theta' = \{W', b'\}$ – параметры (обычно накладывают ограничение $W' = W^T$), s – нелинейная функция активации (обычно это сигмоида или функция гиперболического тангенса). Иногда в (4.2) выбирают в качестве s линейную функцию. Автоассоциатор удобно представлять в виде нейронной сети с одним скрытым слоем.

При обучении автоассоциатора минимизируется *функция стоимости* $L(X, Z(X))$, где X – множество всех возможных входных векторов. Чтобы в процессе обучения преобразования $f_{\theta}(x)$ и $g_{\theta'}(y)$ не выродились в тождественные, накладывают различные ограничения. Часто используемое ограничение: размерность вектора y должна быть меньше размерности входного вектора x . Другой возможный вариант – потребовать, чтобы размерность вектора y была больше размерности вектора x и при этом большинство компонент y были равны 0. При этом y становится разреженным представлением вектора x . Обозначим за $f_{\theta}^j(x)$ j -ю компоненту вектора y при данном входном векторе x . Тогда можем определить среднюю величину компонент вектора y :

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m f_{\theta}^j(x^{(i)}) \quad (4.3)$$

Чтобы добиться $\hat{\rho}_j = \rho$, где ρ – параметр, контролирующий разреженность, добавим слагаемое L_{ρ} в функцию стоимости L . Это слагаемое можно определять разными способами, в рамках данной работы будем использовать следующую его форму, предложенную в [8]:

$$L_{\rho} = \beta \left[\sum_{j=1}^h \left(\rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} \right) \right] \quad (4.4)$$

В дальнейшем будем использовать значение $\rho = 0.05$, также в соответствии с [8].

Очищающий автоассоциатор обучается таким образом, чтобы по повреждённому (зашумлённому) вектору \tilde{x} восстанавливать исходный вектор x . Предполагается, что такие представления более устойчивы к помехам и лучше отражают внутреннюю структуру входных данных. Показано [86], что во многих случаях внутренние представления, которые получают при помощи очищающего автоассоциатора, позволяют получить лучшие результаты в задачах классификации по сравнению с представлениями, полученными при помощи обычных автоассоциаторов. В [86] рассматриваются различные способы получения зашумлённого вектора \tilde{x} .

Из автоассоциаторов можно строить многослойные модели, отождествляя нейроны из скрытого слоя одного автоассоциатора со входными нейронами другого. Пример такой модели показан на рисунке 4.1. В полученной модели слои можно обучать друг за другом на

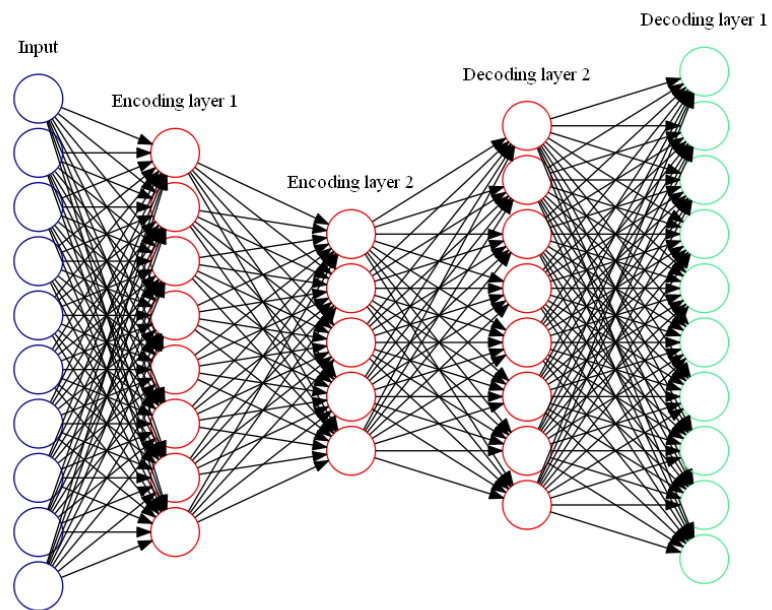


Рисунок 4.1: Многослойный автоассоциатор

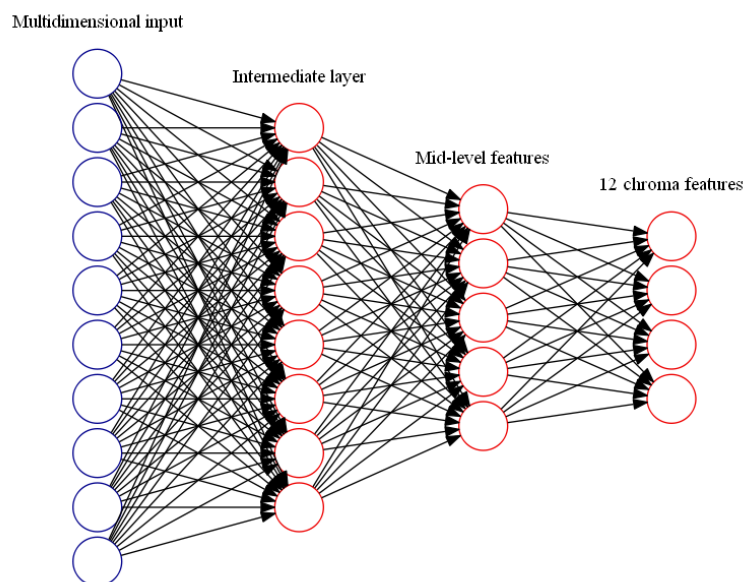


Рисунок 4.2: Многослойная нейронная сеть

неразмеченных данных. Значения, полученные в скрытом слое последнего из автоассоциаторов, могут быть использованы как векторы признаков. Пример полученной нейронной сети показан на рисунке 4.2

Рекуррентный автоассоциатор может быть получен из обычного путём добавления рекуррентных соединений, связывающих выходы скрытого слоя с дополнительными его входами, по одному дополнительному входу на каждый выход. Фактически, при этом получается сеть Эльмана, впервые описанная в [35]. Пример такой нейронной сети представлен на рисунке 4.3. Промежуточное представление $y(x_t)$ в таком случае вычисляется как

$$y(x_t) = s(Wx_t + b + Uy(x_{t-1})) \quad (4.5)$$

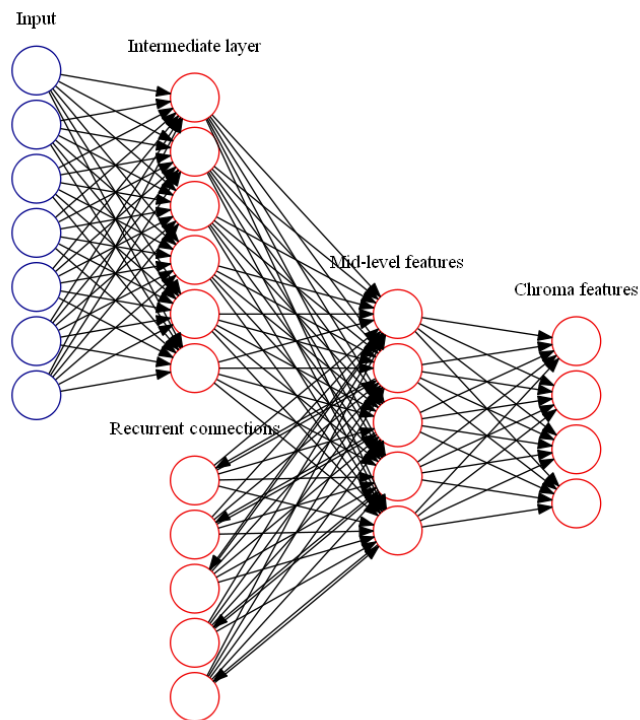


Рисунок 4.3: Многослойная нейронная сеть с рекуррентными соединениями

4.2 Построение нейронной сети и предобучение при помощи автоассоциаторов

Существенным недостатком автоассоциаторов является невозможность содержательной интерпретации значений во внутреннем слое. В частности, невозможно построить шаблонные наборы значений, соответствующие аккордам. Можно попытаться обучить алгоритм классификации на векторах значений на выходах внутреннего слоя. Но для случая 25 классов и достаточно большой размерности векторов для обучения такого классификатора может потребоваться слишком много данных.

Вместо этого соединим внутренний слой с дополнительным слоем, имеющим 12 выходов. Полученную нейронную сеть обучим на размеченных данных таким образом, чтобы на выходе получались хроматические векторы (как в разделе 3.4.1), в которых каждая компонента соответствует одному тональному классу. На вход этой сети будут подаваться столбцы спектрограммы. Таким образом, вместо задачи классификации полученная нейронная сеть решает

задачу регрессии. Классификация же полученных 12-мерных векторов делается как описано в разделе 3.4.1.

Предварительное обучение слоёв-автоассоциаторов будем производить методом мини-пакетного (mini-batch) стохастического градиентного спуска. При этом сначала обучается первый слой на всём обучающем множестве, затем полученные на его выходе значения используются для обучения второго слоя, и так далее. Окончательное обучение сети в целом также производится методом мини-пакетного стохастического градиентного спуска. При этом в качестве целевых векторов используются 12-мерные бинарные шаблоны аккордов (описанные в параграфе 2.4.1), а для случая отсутствия аккорда – нулевой вектор. При классификации, соответственно, отсутствие аккорда определяется в случае, когда ни одна из компонент полученного вектора по абсолютной величине не превосходит некоторого значения Δ , которое подбирается опытным путём.

Для очищающего автоассоциатора возможные помехи на входе можно моделировать разными способами. В [86] выделяются 3 типа помех:

1. аддитивный нормально распределенный шум $\tilde{x}|x \sim \mathcal{N}(x, \sigma_0^2 I)$, где σ_0 – параметр;
2. маскирующий шум: некоторые случайно выбранные элементы входного вектора обнуляются;
3. шум типа «соль-и-перец»: некоторым случайно выбранным элементам входного вектора присваиваются значения 0 или 1.

В случае, когда на вход подаются столбцы спектрограммы, очевидно, более естественными являются помехи первого типа. В параграфе 5.4.3 исследуется влияние различных типов помех на итоговое качество распознавания аккордов.

Естественным выбором для функции стоимости будет квадрат евклидова расстояния:

$$L(x, z) = ||x - z||^2$$

Во время предварительного обучения расстояние вычисляется между незашумлённым входным вектором и выходным вектором. Во время окончательного обучения расстояние вычисляется между шаблоном аккорда и выходным вектором. Для случая отсутствия аккорда в качестве соответствующего шаблона используется нулевой вектор.

В случае, когда в обучающей выборке большинство примеров соответствуют только одному классу, очень вероятно получить в итоге сеть, которая все векторы будет классифицировать как принадлежащие этому классу. В данном случае имеется 25 возможных классов, и желательно иметь приблизительно одинаковое количество примеров на каждый класс. Однако не все аккорды используются в музыке одинаково часто, и, например, аккорд *до-мажор* может встречаться в обучающей выборке во много раз чаще, чем *фа-диез-мажор*.

Аналогичная проблема встречается и при обучении скрытых марковских моделей и байесовских сетей для определения последовательности аккордов по последовательности векторов признаков. В этих моделях часто используют циклический сдвиг векторов признаков для усреднения параметров, соответствующих разным аккордам. Этот процесс подробно описан, например, в [93]. Идея его состоит в том, что, поскольку в хроматическом векторе каждая компонента соответствует одному тональному классу, его циклический сдвиг даёт вектор, соответствующий аккорду того же типа (мажорный или минорный) с основной нотой, сдвинутой на полутон.

В данном случае при окончательном обучении нейронной сети в целом можно также использовать сдвиг. Но входными векторами являются столбцы спектрограммы, и циклический сдвиг соответствует неестественному переносу высокочастотных компонент в область низких частот (или наоборот). Циклический сдвиг можно заменить простым сдвигом окна соответствующего размера по столбцу спектрограммы, имеющей охват на одну октаву больше.

При помощи такого сдвига из каждого столбца спектрограммы получается 12 различных столбцов, соответствующих 12 аккордам одного типа с разными основными нотами. Это позволяет уравновесить количество аккордов в пределах одного типа. Чтобы уравновесить количество аккордов между типами, потребуем, чтобы в процессе генерации обучающей выборки из спектрограмм разница между общим количеством мажорных аккордов и общим количеством минорных аккордов не превосходила заданного числа H .

Во время тестирования также можно использовать циклический сдвиг. Для этого также для каждого столбца спектрограммы генерируется 12 тестовых векторов, для каждого из которых при помощи нейронной сети получается хроматический вектор. Для полученных 12 хроматических векторов производятся соответствующие обратные сдвиги, а в качестве результата берется среднее арифметическое от полученных векторов. Эффект от такой модификации процесса тестирования рассматривается в параграфе 5.4.4.

Анализу различных конфигураций нейронной сети и подбору наилучших параметров посвящён параграф 5.4.

4.3 Выводы

1. Предложен метод для получения хроматических векторов из столбцов спектрограммы с использованием предварительно обученной многослойной нейронной сети.
2. Предложен способ для создания обучающей выборки, содержащей приблизительно одинаковое количество векторов для всех возможных аккордов.

Глава 5

Эксперименты

Описанные в главах 3 и 4 алгоритмы имеют значительное количество параметров. Для подбора их оптимальной комбинации, фактически, необходимо решить задачу многомерной оптимизации в достаточно большом пространстве. Очевидно, что в данном случае невозможно решить эту задачу аналитически. Также невозможно перебрать все возможные комбинации параметров ввиду слишком большого их числа. Однако во многих случаях можно определить разумный диапазон возможных значений параметра и исследовать изменение качества распознавания аккордов в зависимости от значений данного параметра в указанном диапазоне.

Применение методов вычисления классификации векторов признаков, основывающихся на машинном обучении, не позволяет целиком избавиться от ручного подбора параметров. Во-первых, эти алгоритмы могут иметь метапараметры, не изменяемые в процессе обучения (например, количество нейронов в j -м слое нейронной сети). Во-вторых, параметры имеются также на этапах подготовки входных данных и интерпретации результата.

Поскольку все эксперименты проводились на описанной в разделе 5.1.1 коллекции из 318 музыкальных звукозаписей, найденные значения параметров будут оптимальными только для этой коллекции. Способствовать преодолению этой проблемы могли бы достаточно большие коллекции аннотированных композиций, не существующие на данный момент.

Все эксперименты можно разделить на 4 группы. В разделе 5.2 рассматривается этап предварительной обработки звукозаписи и получения спектрограммы. В разделе 5.3 исследуется влияние различных преобразований спектрограммы на качество распознавания аккордов. Эксперименты в разделе 5.4 направлены на отыскание наилучших параметров нейронной сети, используемой для получения признаков. Преобразования над последовательностями векторов признаков и распознанных аккордов и параметры выбранного метода классификации анализируются в разделе 5.5. В разделе 5.7 сравниваются скорости работы реализованных алгоритмов.

В экспериментах были использованы следующие значения параметров (если явно не указано иное):

- $N_0 = 36$;
- частотный диапазон от 65,4 Гц (нота *до* большой октавы) до 987,77 Гц (нота *си* второй октавы);
- $T = 8$;
- $w = 15$;
- $d = 100$ мс;
- $\eta = 50000$;
- $\xi = 15$;

- $\zeta = 0.1$;
- количество учитываемых гармоник – 3;
- $h = 0.6$;
- $L_N = 0.002$.

5.1 Оценка качества распознавания аккордов

Поскольку алгоритмы распознавания аккордов предназначены для обработки музыкальных звукозаписей, необходимо оценивать качество их работы на реальных звукозаписях, а не на искусственно сгенерированных примерах. Чтобы звукозапись можно было использовать для оценки, требуется вручную решить задачу распознавания последовательности аккордов, то есть для каждого момента времени $t \in [t_{start}, t_{end}]$ указать аккорд $y \in \bar{Y}$, звучащий в этот момент. При этом набор \bar{Y} включает в себя все возможные в музыке сочетания нот и отдельные ноты. Также требуется с высокой точностью указать моменты начала и конца звучания аккордов. Всё это делает задачу подготовки тестовых коллекций очень трудоёмкой.

Для хранения этой информации используют особым образом отформатированные текстовые файлы, называемые файлами разметки или файлами текстовых аннотаций. Ниже приведён пример такого файла:

0.000	0.848	N
0.848	1.625	A: min
1.625	3.017	G: maj
3.017	3.895	F: maj
...		

Первый и второй столбцы содержат время начала и конца звучания аккорда соответственно, в третьем столбце записывается название аккорда.

5.1.1 Коллекции текстовых аннотаций

На текущий момент существует 5 коллекций текстовых аннотаций для популярной музыки разных исполнителей:

- *Isophonics* [84]. Текстовые аннотации для 180 композиций (12 альбомов) *The Beatles*, 20 композиций *Queen* (с альбома *Greatest Hits*), 18 композиций *Zweieck* (с альбома *Zweilicht*). Наиболее часто используется для исследований, несколько раз использовалась для ежегодных соревнований MIREX Audio Chord Estimation.
- *RWC Pop Music* [90]. Текстовые аннотации для 100 композиций японской и западной популярной музыки.
- *Billboard* [20]. Текстовые аннотации для 197 композиций из американского чарта *Billboard 100* за промежуток с 1958 по 1991 год. Использовалась в соревновании MIREX Audio Chord Estimation в 2012 году.
- *uspop2002* [56]. Текстовые аннотации для 195 композиций американской популярной музыки.
- *Robbie Williams annotations*. Текстовые аннотации для 65 композиций *Robbie Williams* (первые 5 альбомов).

Поскольку в аннотациях указывается точное время, важно при анализе использовать точно те же версии звукозаписей, которые были использованы при подготовке аннотаций. Это затрудняет использование некоторых коллекций. Для тестирования алгоритмов в рамках данной работы использовались коллекции *Isophonics* и *RWC Pop Music*.

5.1.2 Сопоставление последовательностей аккордов

Вопросом оценки того, насколько одна последовательность аккордов (определённая автоматически) соответствует другой (правильной, определённой человеком), занимались Харте [44] и Пауэлс и Питерс [50]. Последние предлагают следующую конструкцию для определения схожести двух последовательностей аккордов.

Результат	N	A:min						F		E:min
Эталонная разметка	N	A:min							F:maj7	
	0	1	2	3	4	5	6		7	8
Сегменты	1	2						3	4	5

Рисунок 5.1: Сопоставление последовательностей аккордов.

Пусть заданы 2 последовательности аккордов: правильная и определённая при помощи алгоритма. Объединим множества границ аккордов из обеих последовательностей в одно множество. Используя эти границы, разделим исходную композицию на сегменты (как на рисунке 5.1), на каждом из которых однозначно заданы правильный аккорд c_{ref} и определённый автоматически c_{est} . Пусть также $c_{ref} \in C_{REF}$ – множество всех аккордов, встречающихся в аннотациях, а $c_{est} \in C_{EST}$ – множество всех аккордов, которые могут быть результатом распознавания при помощи данного алгоритма.

Практически всегда $C_{EST} \subset C_{REF}$, поэтому возникает вопрос о том, как сопоставлять фрагменты, на которых $c_{ref} \notin C_{EST}$. Такие фрагменты можно либо отбрасывать, либо задать сюръективное отображение $M : C_{MI} \rightarrow C_{MO}$, которое «сложным» аккордам из множества C_{MI} будет сопоставлять «простые» аккорды из множества C_{MO} . Нужно выбрать эти множества и отображение M таким образом, чтобы $C_{EST} \subset C_{MI}$. Сравниваться при этом будут аккорды $M(c_{ref})$ и $M(c_{est})$. Сегменты, на которых $c_{ref} \notin C_{MI}$, отбрасываются. Примером отображения M может служить отображение, которое всем аккордам, состоящим из мажорного трезвучия и более высоких ступеней (например, доминантсептаккорд, нонаккорды и другие) сопоставляет мажорный аккорд, соответствующий этому трезвучию.

Если необходимо оценить качество распознавания аккордов определенного типа (например, только трезвучий или только мажорных аккордов), можно ввести дополнительные множества C_{LI} и C_{LO} , ограничивающие соответственно множества C_{MI} и C_{MO} . Тогда аккорды (c_{ref}, c_{est}) сравниваются (не отбрасываются), только если $c_{ref} \in C_{LI} \cap C_{MI}$ и $M(c_{ref}) \in M(C_{LI} \cap C_{MI}) \cap C_{LO}$.

Пусть $S : C_{SR} \times C_{SE} \rightarrow \mathbb{R}^+$ – функция оценки, причем $M(C_{LI} \cap C_{MI}) \cap C_{LO} \subset C_{SR}$ и $C_{MO} \subset C_{SE}$. Эта функция сопоставляет паре аккордов $M(c_{ref})$ и $M(c_{est})$ неотрицательное действительное число, которое выражает сходство этих аккордов между собой. Например, можно определить функцию S как равную 1 в случае, когда аккорды совпадают, и 0 иначе.

Как видно из [50], полученные цифры сильно различаются в зависимости от выбора отображения M и функции оценки S , и даже в зависимости от некоторых мелких деталей, таких как способ синтаксического разбора названий аккордов. Не всегда в статьях корректно указываются использованные метрики, что делает затруднительным непосредственное сравнение оценок качества распознавания из статей друг с другом. В этом состоит главная ценность

соревнования MIREX Audio Chord Estimation, где гарантированно используются одни и те же коллекции и метрики для оценки всех алгоритмов.

Рассмотрим 3 наиболее характерных метрики из [50].

1. “Mirex2010”. В ней не используется отображение \mathcal{M} , а функция оценки \mathcal{S} строится следующим образом. Сначала c_{ref} и c_{est} преобразуются в множества тональных классов, для которых находится пересечение. Обозначим количество элементов в пересечении за u . $\mathcal{S} = 1$ в случаях:

- c_{ref} является уменьшенным или увеличенным аккордом и $u \geq 2$;
- c_{ref} и c_{est} являются символами отсутствия аккорда;
- $u \geq 3$.

В остальных случаях $\mathcal{S} = 0$, то есть $\mathcal{S} : C_{SR} \times C_{SE} \rightarrow \{0, 1\}$.

2. “Triads”. Отображение \mathcal{M} строится следующим образом. Если существует мажорное или минорное трезвучие с основной нотой, соответствующей основной ноте аккорда, и имеющее 3 общих ноты с аккордом, то оно сопоставляется этому аккорду. Например, для доминантсептаккорда с добавленной ступенью G:7(9) результатом сопоставления будет мажорное трезвучие G:maj, а для аккорда F:aug сопоставления не существует. Аккорды, которым невозможно сопоставить мажорное или минорное трезвучие, из оценки исключаются. $\mathcal{S} = 1$ тогда и только тогда, когда c_{ref} и c_{est} совпадают.

3. “Tetrads”. Отображение \mathcal{M} строится аналогично “Triads”, но помимо мажорного и минорного трезвучий рассматриваются также мажорный и минорный септаккорды и доминантсептаккорд, а в качестве результата выбирается аккорд, имеющий наибольшее количество общих нот с заданным. Так, для доминантсептаккорда с добавленной ступенью G:7(9) результатом сопоставления будет доминантсептаккорд G:7. $\mathcal{S} = 1$ тогда и только тогда, когда c_{ref} и c_{est} совпадают.

Отметим, что при использовании метрики “Mirex2010” ни один сегмент не отбрасывается. Это может быть нежелательно, поскольку в полученном результате будут учитываться в том числе и сегменты, содержащие аккорды, которые в принципе не могут быть распознаны имеющимся методом. Ещё одним недостатком этой метрики является невозможность различить трезвучие и содержащий его септаккорд: при совпадении хотя бы трёх нот аккорды будут считаться совпадающими. Поэтому в дальнейших экспериментах будут использоваться метрики “Triads” и “Tetrads”, позволяющие более точно оценить качество распознавания аккордов. Первая метрика будет использоваться вместе с набором из 24 шаблонов для всех мажорных и минорных трезвучий, а вторая – с набором из 60 шаблонов, в который входят также шаблоны для всех доминантсептаккордов и мажорных и минорных септаккордов.

Метрика “Mirex2010” использовалась в соревновании MIREX Audio Chord Estimation в 2010, 2011 и 2012 годах. Но в 2013 году вместо неё были использованы метрики “Triads”, “Tetrads”, а также некоторые другие, позволяющие учесть также обращения аккордов и басовую ноту. Поскольку описываемый здесь алгоритм не предназначен для определения обращений аккордов, нет смысла использовать эти дополнительные метрики.

Обозначим за $\ell_1, \ell_2, \dots, \ell_{N_{segm}}$ длины всех сегментов в пределах одной композиции, а $s_1, s_2, \dots, s_{N_{segm}}$ – соответствующие значения метрики. Тогда коэффициент перекрытия (*overlap ratio*, OR) для данной композиции определяется как

$$OR = \frac{\sum_{i=1}^{N_{segm}} s_i \ell_i}{\sum_{i=1}^{N_{segm}} \ell_i} \quad (5.1)$$

При этом неважно, были ли сегменты взяты из одной и той же композиции или из нескольких разных. Но для того, чтобы иметь возможность определить статистически значимые различия между системами, в экспериментах будем определять коэффициент перекрытия отдельно для каждой композиции.

Для примера, на рисунке 5.1 $s_1 = s_2 = s_4 = 1$, $s_3 = s_5 = 0$. На сегментах 1 и 2 аккорды совпадают, на сегменте 4 аккорды F:maj и F:maj7 имеют 3 общих ступени F, A, C.

Пусть коллекция содержит N_{tracks} композиций, для каждой из которых вычислен коэффициент перекрытия OR_k . Обозначим за $L_i = \sum_{j=1}^{N_{segm}} \ell_j$ длину i -й композиции. Тогда совокупная метрика для коллекции, называемая *взвешенным средним коэффициентом перекрытия* (*weighted average overlap ratio, WAOR*), вычисляется следующим образом:

$$WAOR = \frac{\sum_{i=1}^{N_{tracks}} OR_i \cdot L_i}{\sum_{i=1}^{N_{tracks}} L_i} \quad (5.2)$$

Такой же способ усреднения применяется в соревнованиях MIREX Audio Chord Estimation. В качестве результатов экспериментов в последующих разделах приведены значения взвешенных средних коэффициентов перекрытия для метрик “Triads” и “Tetrads”.

5.1.3 Сопоставление границ сегментов

Метрика для сопоставления границ сегментов была введена Маухом в [76]. Начиная с 2013 года она также применяется в соревнованиях MIREX Audio Chord Estimation. Она позволяет оценить качество определения границ аккордов алгоритмом, игнорируя при этом сами названия аккордов.

Пусть заданы 2 разбиения звукозаписи длины L на сегменты $G^0 = (G_i^0)$ и $G = (G_i)$. Направленное расхождение Хэмминга определяется как:

$$h(G||G^0) = \sum_{i=1}^{N_G} \left(|G_i^0| - \max_j |G_i^0 \cap G_j| \right)$$

где N_G – количество сегментов в разбиении G , а $|\cdot|$ – длина сегмента. Оно определяет, насколько G фрагментировано по отношению к G^0 . Тогда *сегментация* $H(G, G^0)$ определяется как

$$H(G, G^0) = 1 - \frac{1}{L} \max\{h(G||G^0), h(G^0||G)\} \in [0, 1]$$

В экспериментах одно из разбиений задаётся правильной разметкой звукозаписи, а другое – результатом распознавания аккордов с использованием только шаблонов для мажорных и минорных трезвучий. Для усреднения значений сегментации вычислялось простое среднее арифметическое от полученных значений по всем композициям из коллекции.

5.1.4 Статистическая значимость

При сравнении нескольких вариантов алгоритма помимо средних значений метрик качества необходимо понять, действительно ли между этими вариантами имеются статистически значимые различия. Для проверки этого предположения будем использовать непараметрический критерий Фридмана. Он позволяет проверять гипотезы о различии более двух зависимых выборок. В отличие от дисперсионного анализа (ANOVA), критерий Фридмана не требует предположений о нормальности распределения значений метрик для разных композиций, а также одинаковых дисперсий этих распределений для разных вариантов алгоритма (как отмечается в [78], эти предположения не являются верными в данном случае).

Однако, если в соответствии с критерием Фридмана удастся отвергнуть нулевую гипотезу (об отсутствии различий между разными методами), необходимо выяснить, для каких пар методов имеется статистически значимая разница в качестве распознавания аккордов. Для этого вычисляется среднее Тьюки (Tukey’s honestly significant difference). В отличие от Т-теста, при его допусках множественные попарные сравнения. Этот метод используется для сравнения качества работы разных алгоритмов в рамках всех соревнований MIREX [33]. В экспериментах среднее Тьюки вычислялось для значений взвешенного среднего перекрытия, полученных с использованием метрики “Triads”.

5.1.5 Совокупная длительность

Совокупная длительность композиций в коллекции составляет 61701,61 с. Совокупная длительность участвующих в оценке фрагментов (для аккордов на которых определено отображение \mathcal{M}) составляет 58937,75 с, что составляет более 95% от совокупной длительности всех композиций.

Совокупная продолжительность звучания аккордов каждого из типов (с учётом отображения \mathcal{M} для метрик) приведена в таблице 5.1. Видно, что в звукозаписях из коллекции преобладают мажорные трезвучия и аккорды, основанные на них.

Таблица 5.1: Совокупная продолжительность звучания аккордов в секундах

Тип	Triads	%	Tetrads	%
maj	41733.38	70.81%	34381.76	58.34%
7	-	-	4862.02	8.25%
maj7	-	-	2489.6	4.22%
min	14681.45	24.91%	9921,61	16.83%
min7	-	-	4759.83	8.08%
N	2522.92	4.28%	2522.92	4.28%

5.1.6 Типы ошибок

Для удобства в дальнейшем ошибки будут сгруппированы по типам. Название типа ошибки начинается с типа ожидаемого (правильного) аккорда и может включать тип неправильно определённого аккорда, например, maj или maj-min. Ошибки maj-maj соответствуют случаю, когда вместо мажорного аккорда был определён мажорный аккорд с другой основной нотой. Поскольку используемые метрики “Triads” и “Tetrads” используют отображение \mathcal{M} для сведения в некотором смысле более сложных аккордов к более простым (например, G:7(9) к G:7 или G:maj), это же отображение будет применяться ко всем ожидаемым аккордам для уменьшения количества типов ошибок. Так, при использовании метрики “Triads” тип maj7-min будет включен в тип maj-min и не будет рассматриваться отдельно. Аккорды, которым в результате действия отображения \mathcal{M} не сопоставляется никакой другой аккорд, исключаются как из итогового результата, так и из статистики ошибок. Типы ошибок N-chord и chord-N (где chord – любой аккорд) будем объединять в один, поскольку на них оказывает влияние в первую очередь способ определения отсутствия звучащего аккорда.

Ошибки метода можно рассматривать на трёх уровнях. Каждый последующий уровень уточняет предыдущий.

1. Совокупная длительность ошибочных фрагментов в зависимости от типа звучащего аккорда. Например, maj.

2. Совокупная длительность ошибочных фрагментов в зависимости от типа звучащего аккорда и типа неверно определённого аккорда. Например, *maj-min*. Тип *maj-maj* соответствует случаям, когда вместо мажорного аккорда неверно определён мажорный аккорд с другим основным звуком.
3. Совокупная длительность ошибочных фрагментов в зависимости от типа звучащего аккорда, типа неверно определённого аккорда и расстояния в полутонах между основными звуками аккордов.

В дальнейшем на диаграммах под буквами а), б) будет показано распределение ошибок, соответствующее первому уровню. Под буквами в), г) (если присутствуют) – соответствующее второму или третьему уровню. Высота столбцов соответствует совокупной длительности фрагментов звукозаписей, содержащих ошибки соответствующего типа. Столбцы сгруппированы по типам ошибок. Цвета столбцов соответствуют разным значениям описываемого параметра.

5.2 Вычисление спектрограммы

На данном этапе необходимо выбрать наилучшие из доступных алгоритмов для определения ритма и определения частоты настройки. Наилучшее значение для параметра d , задающего задержку для моментов времени, в которых анализируется спектр звука, относительно моментов начала метрических долей, также должно быть определено на этом этапе. Кроме того, необходимо определить наилучшие значения для параметров преобразования постоянного качества: разрешение по частоте N_0 (количество компонент, приходящихся на октаву) и количество октав N/N_0 , а также для количества вставляемых промежуточных столбцов спектрограммы T и размера окна при сглаживании w .

5.2.1 Определение ритма

Были рассмотрены 3 алгоритма, позволяющие определить моменты начала метрических долей в звукозаписи: *BeatRoot* [32], *Beat tracker* от Дэвиса [30] (*DBT*) из набора плагинов *Queen Mary Vamp plugins*¹ для системы извлечения музыкальной информации из музыкальных файлов *Vamp*² и плагина *INESC Porto Beat Tracking plugin* [15] (*IBT*) для этой же системы. Выбор алгоритмов обусловлен наличием свободно доступной реализации. *BeatRoot* дополнительно потребовал небольшого вмешательства в исходный код для уменьшения потребления вычислительных ресурсов. Кроме того, на 6 композициях из анализируемого набора этот алгоритм не смог определить ритм, поэтому для этих композиций использовались значения, полученные при помощи *DBT*. Для всех алгоритмов определения ритма использовалось значение задержки $d = 100$ мс.

Таблица 5.2: Влияние алгоритма определения ритма на качество распознавания аккордов

Алгоритм	Triads	Tetrads	Сегментация
BeatRoot + DBT	0.7720	0.4310	0.8141
DBT	0.7592	0.4401	0.8007
IBT	0.7335	0.4160	0.7635
—	0.7113	0.4550	0.7584

¹<http://www.isophonics.net/QMVampPlugins>

²<http://www.vamp-plugins.org/>

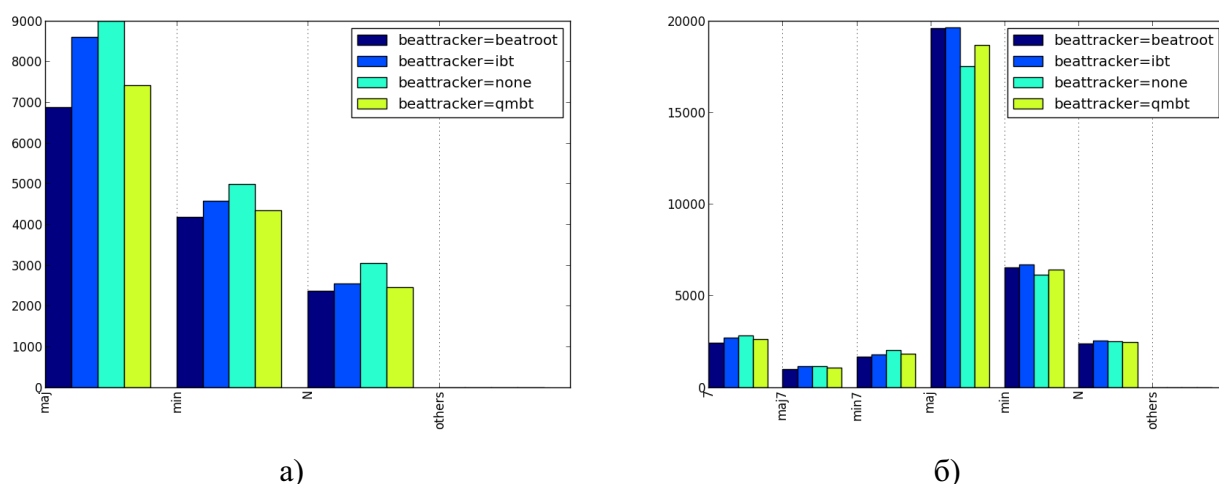


Рисунок 5.2: Диаграмма ошибок для разных методов определения ритма

Наилучшие полученные для данных алгоритмов результаты показаны в таблице 5.2. *BeatRoot* показал наилучший результат, статистически значимо превосходящий результаты, полученные с использованием алгоритмов *IBT* и *DBT*. Это достаточно удивительно, поскольку первая версия алгоритма *BeatRoot* была представлена ещё в 2001 году, а в данной работе использовалась его исправленная версия от 2007 года. При этом *DBT* был представлен в 2007 году, а *IBT*, схожий по принципу работы с *BeatRoot*, – в 2012 году.

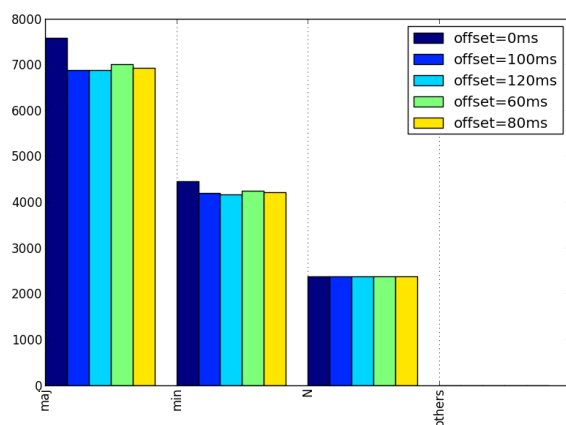
Последняя строка в таблице 5.2 показывает результат, полученный без определения ритма. При этом искусственно генерировалась последовательность значений времени с шагом в 0.5 с, которые считались моментами начала метрических долей, с задержкой $d = 0$ мс. Ожидается, это привело к наихудшему результату, статистически значимо отличающемуся от остальных.

5.2.2 Определение задержки

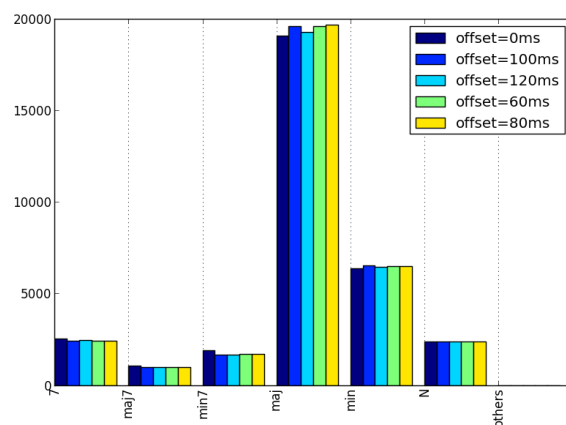
Таблица 5.3: Влияние задержки относительно моментов начала метрических долей на качество распознавания аккордов

d	Triads	Tetrads	Сегментация
0 мс	0.7558	0.4351	0.7932
60 мс	0.7689	0.4303	0.8096
80 мс	0.7711	0.4290	0.8123
100 мс	0.7720	0.4310	0.8141
120 мс	0.7724	0.4374	0.8144

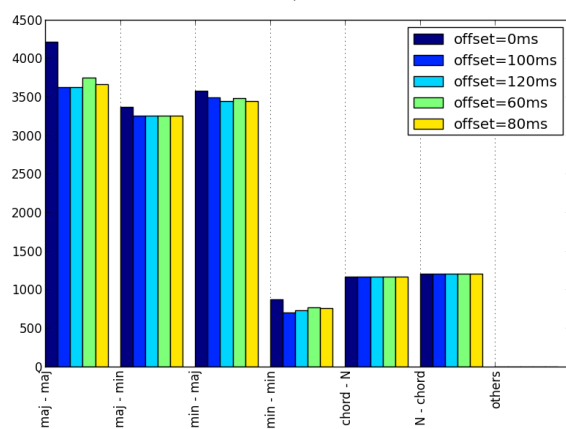
Любое из использованных значений задержки от 60 мс до 120 мс приводит к статистически значимому улучшению качества распознавания аккордов по сравнению с отсутствием задержки. При этом варианты с ненулевыми значениями задержки не имеют статистически значимых различий. Как видно из рисунка 5.3 в), в случае определения только трезвучий при нулевой задержке ухудшение качества распознавания вызвано в основном ошибочным определением основной ноты, а не типа аккорда. Рисунок 5.3 г) показывает распределение расстояний между основными звуками аккордов в полутонах для случая maj-maj. Расстояния в 5 и 7 полутонов соответствуют случаям, когда основной звук правильного аккорда является квинтой ошибочного и наоборот.



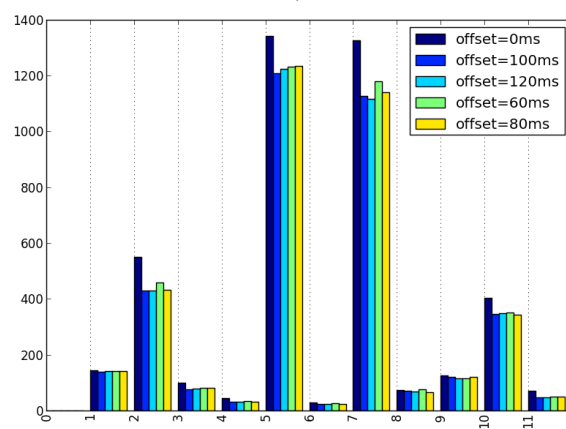
a)



б)



в)



г)

Рисунок 5.3: Диаграмма ошибок для разных значений d

5.2.3 Определение частоты настройки

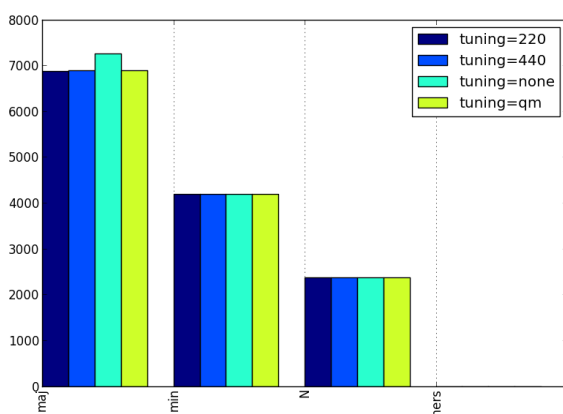
Были проведены эксперименты по распознаванию аккордов с использованием описанного в разделе 3.1.1 алгоритма для вычисления частоты настройки со следующими значениями параметров:

- $f_{min} = 220$ Гц, $N_0 = 12 \cdot 10 = 120$ компонент на октаву;
- $f_{min} = 440$ Гц, $N_0 = 12 \cdot 10 = 120$ компонент на октаву.

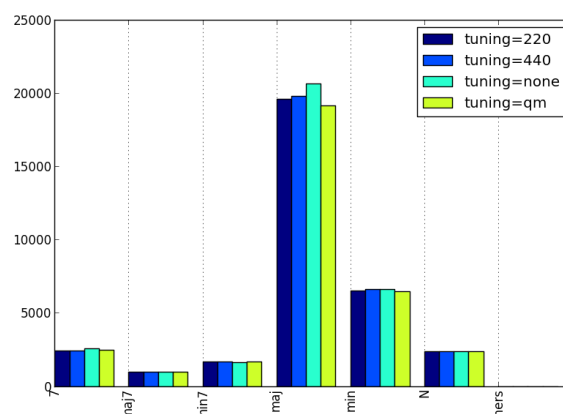
Охват во всех случаях составлял 4 октавы. Также для сравнения были проведены эксперименты без коррекции частоты настройки и с использованием алгоритма, описанного Маухом в [76], раздел 3.1.3 и реализованного в виде плагина для системы *Vamp*.

Таблица 5.4: Влияние алгоритма определения частоты настройки на качество распознавания аккордов

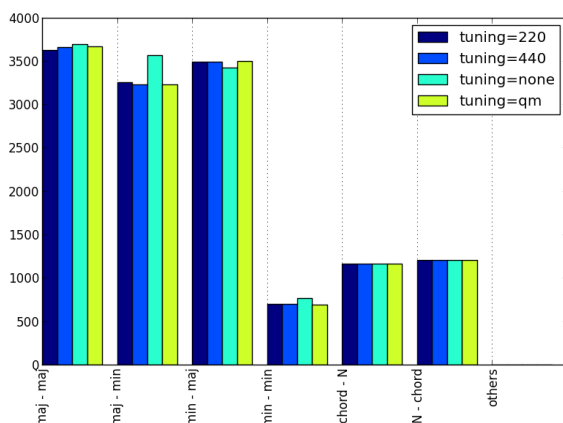
Алгоритм	Triads	Tetrads	Сегментация
$f_{min} = 220$ Гц	0.7720	0.4310	0.8141
$f_{min} = 440$ Гц	0.7719	0.4263	0.8134
Mayx [76]	0.7718	0.4382	0.8142
—	0.7657	0.4107	0.8120



a)



6)



B)

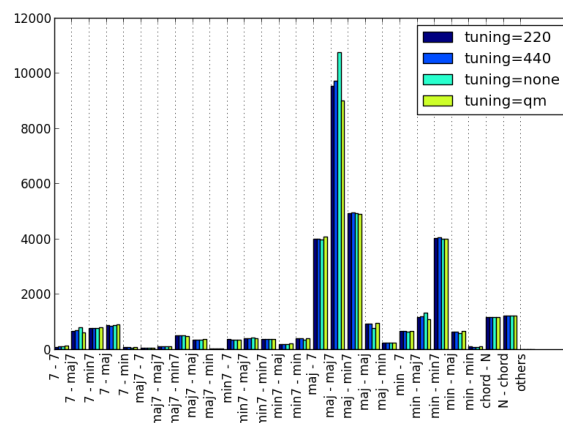
 $\Gamma)$

Рисунок 5.4: Диаграмма ошибок для разных методов определения частоты настройки

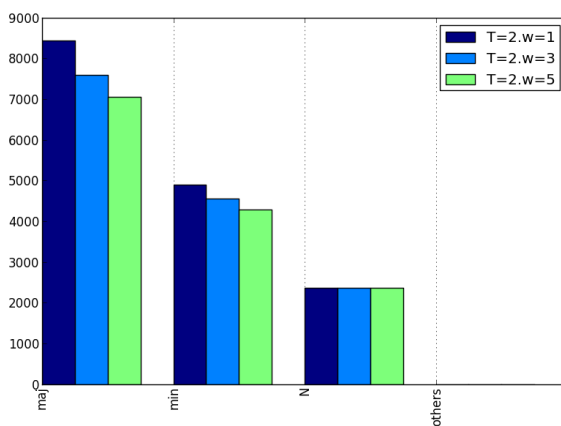
Результаты экспериментов приведены в таблице 5.4. Как видно, определение частоты настройки приводит к улучшению качества распознавания аккордов, которое, однако, не является статистически значимым ни для одного из алгоритмов. Из рисунка 5.4 в) можно заметить, что определение частоты настройки помогает уменьшить количество ошибочных определений минорного аккорда для случая трезвучий. А рисунок 5.4 г) показывает, что для случая септаккордов оно помогает уменьшить количество ошибочных определений мажорного септаккорда.

5.2.4 Разрешение по времени и по частоте, сглаживание

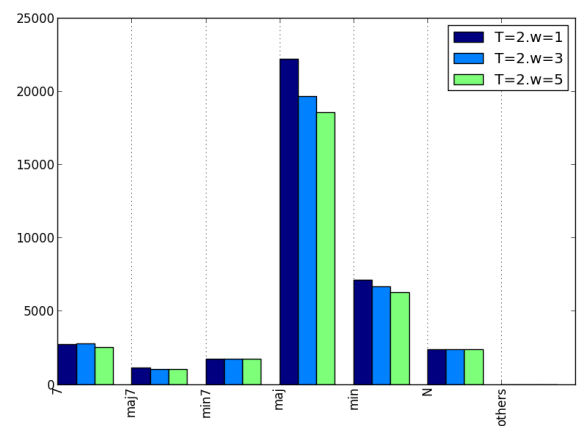
Вставка между каждыми двумя соседними моментами начала метрических долей $T - 1$ промежуточных значений позволяет повысить разрешение спектрограммы по времени. Затем, после применения скользящего медианного фильтра с размером окна w и прореживания в T раз, спектрограмма содержит ровно 1 столбец на каждую метрическую долю. Ясно, что при больших значениях T имеет смысл выбирать больше значения w и наоборот. В таблице 5.5 приведены значения для некоторых комбинаций T и w .

Таблица 5.5: Влияние параметров T и w на качество распознавания аккордов

Значения параметров	Triads	Tetrads	Сегментация
$T = 2, w = 1$	0.7334	0.3670	0.7999
$T = 2, w = 3$	0.7537	0.4200	0.8094
$T = 2, w = 5$	0.7672	0.4501	0.8149
$T = 4, w = 3$	0.7624	0.4621	0.8115
$T = 4, w = 5$	0.7619	0.4440	0.7992
$T = 4, w = 7$	0.7425	0.3909	0.7664
$T = 4, w = 9$	0.7194	0.3583	0.7307
$T = 4, w = 11$	0.6913	0.3197	0.6917
$T = 8, w = 11$	0.7715	0.4464	0.8176
$T = 8, w = 13$	0.7720	0.4364	0.8160
$T = 8, w = 15$	0.7720	0.4310	0.8141
$T = 8, w = 17$	0.7707	0.4273	0.8089
$T = 8, w = 19$	0.7696	0.3923	0.8043

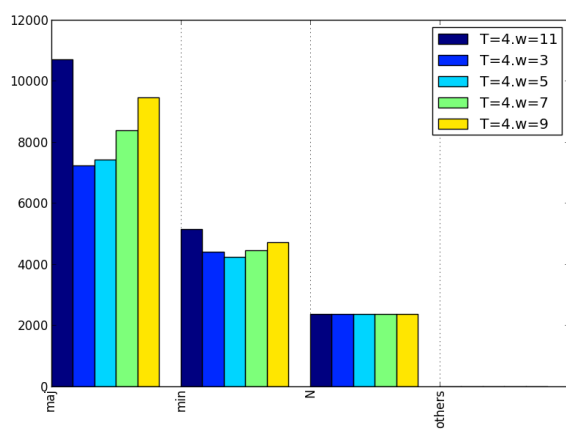


а)

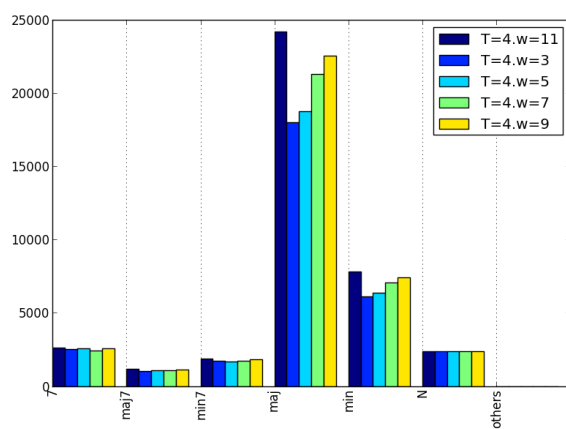


б)

Рисунок 5.5: Диаграмма ошибок для разных значений w при $T = 2$

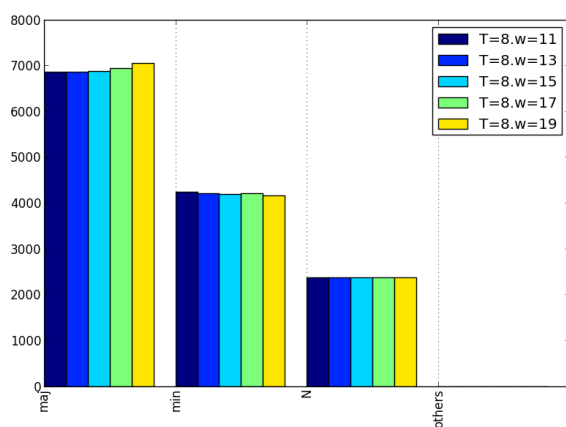


а)

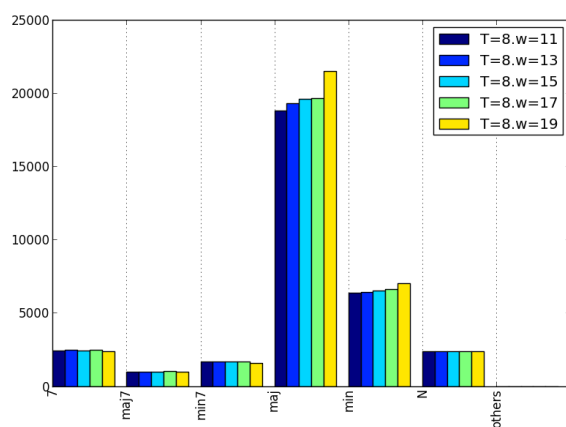


б)

Рисунок 5.6: Диаграмма ошибок для разных значений w при $T = 4$



а)



б)

Рисунок 5.7: Диаграмма ошибок для разных значений w при $T = 8$

статистически значимыми. Однако при $N_0 = 60$ требуется вычислить в 1.8 раз больше значений для компонент преобразования постоянного качества, а также в дальнейшем многократно вычислять дискретное косинусное преобразование для большего набора значений. Из рисунка 5.8 видно, что значение $N_0 = 12$ не позволяет достаточно хорошо различать мажорные и минорные аккорды. Огромное количество ошибок при таком значении N_0 вызваны ошибочным

Эксперименты показывают безусловную важность реализованных методов для улучшенного вычисления спектра. Повышение разрешения по времени с последующим сглаживанием существенно повышает качество распознавания аккордов. Введение задержки относительно моментов начала метрических долей также вносит свой вклад. Использование более высокого разрешения по частоте при вычислении преобразования постоянного качества позволяет использовать больше информации при сглаживании спектрограммы. Преварительное определение ритма при помощи внешних библиотек и предварительное определение частоты настройки при помощи реализованного алгоритма улучшают качество полученной спектрограммы, позволяя вычислять значения спектра для нужных частот в нужные моменты времени. Однако выбор алгоритма определения ритма оказывает существенно более сильное влияние на результат, чем выбор алгоритма определения частоты настройки.

5.3 Преобразования спектрограммы

На этом этапе необходимо оценить эффективность предложенных улучшений при обработке спектрограммы: применение аналога фильтра Превитт, сглаживание с использованием матрицы самоподобия. Также необходимо определить оптимальные значения для количества зануляемых первых коэффициентов ξ дискретного косинусного преобразования, для доли сохраняемых в матрице самоподобия значений ζ .

5.3.1 Применение аналога фильтра Превитт

В разделе 3.2 было предложено использовать аналог фильтра Превитт для подавления компонент спектра, соответствующих шумовым звукам ударных инструментов. Признаки CRP позволяют решить эту же задачу. Поэтому имеет смысл сравнить итоговое качество распознавания аккордов с применением каждого из этих методов в отдельности.

Таблица 5.7: Влияние разных способов подавления шумовых звуков на качество распознавания аккордов

Способ подавления	Triads	Tetrads	Сегментация
Превитт	0.7492	0.5896	0.8063
Признаки CRP	0.7720	0.4310	0.8141
Превитт + признаки CRP	0.7524	0.5520	0.8066

Результаты сравнения показаны в таблице 5.7. К сожалению, применение аналога фильтра Превитт только снижает качество распознавания аккордов, независимо от того, будут ли далее применяться признаки CRP. Все попарные различия между вариантами являются статистически значимыми.

5.3.2 Настройка алгоритма вычисления признаков CRP

Вычисление признаков CRP включает в себя взятие логарифма от значений спектрограммы, умноженных на коэффициент η . В работе [82], описывающей этот тип признаков, предлагается выбирать η из диапазона 100–10000. Далее к каждому столбцу спектрограммы при-

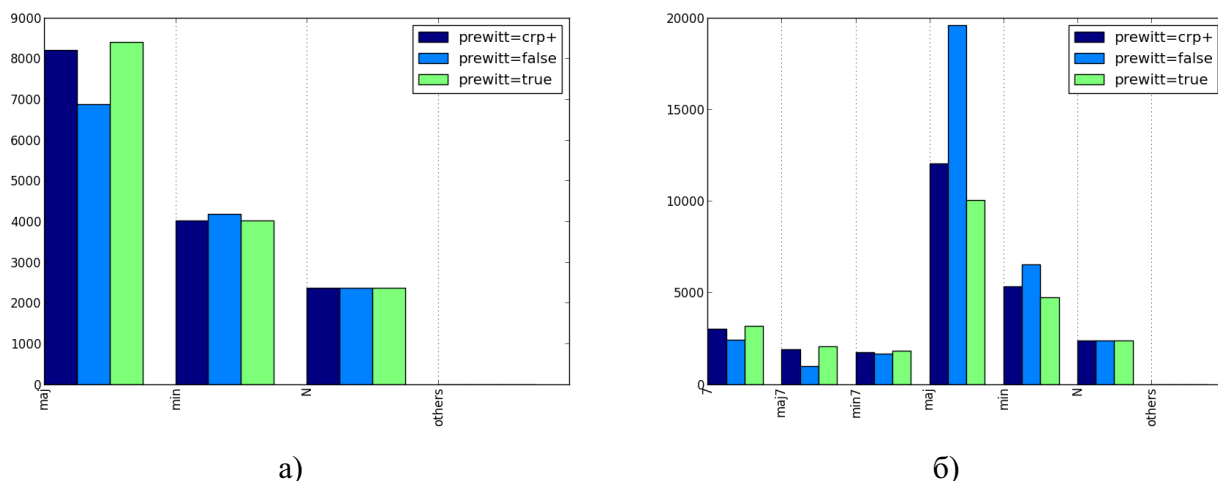


Рисунок 5.9: Диаграмма ошибок для разных способов подавления шумовых звуков

меняется дискретное косинусное преобразование. В полученном векторе зануляются первые ξ значений, после чего к нему применяется обратное дискретное косинусное преобразование. Необходимо определить влияние параметров η и ξ на качество распознавания аккордов.

Таблица 5.8: Влияние параметра η на качество распознавания аккордов

η	Triads	Tetrads	Сегментация
100	0.7348	0.4815	0.7971
1000	0.7652	0.5170	0.8102
10000	0.7721	0.4721	0.8139
50000	0.7720	0.4310	0.8141
100000	0.7719	0.4202	0.8144

В [82] не было обнаружено существенных отличий между значениями η в диапазоне от 100 до 10000 применительно к задаче подавления тембра музыкальных инструментов. Однако в задаче распознавания аккордов этот параметр оказывает достаточно заметное влияние на результат. В таблице 5.8 показаны значения среднего перекрытия и сегментации для разных значений η в промежутке от 100 до 100000. Наилучший результат для метрики “Triads” достигается при $\eta = 10000$, $\eta = 50000$, $\eta = 100000$ (статистически значимые различия отсутствуют). Все остальные различия являются статистически значимыми.

При этом для метрики “Tetrads” наилучшие результаты получаются при меньших значениях η . Из рисунка 5.10 г) можно увидеть, что при больших значениях η растёт количество ошибочно определенных септаккордов вместо соответствующих трезвучий.

Наилучшие результаты в [82] были получены для значений параметра ξ от 22 до 60 при 120-мерном векторе признаков. В этой работе в зависимости от количества охватываемых октав и значения параметра N_0 размерность вектора признаков (столбца спектрограммы) может меняться от 48 до 360. Значения в таблице 5.9 получены при охвате 4 октавы и $N_0 = 36$, что даёт 144-мерный вектор признаков. Видно, что наилучшее качество распознавания аккордов достигается при относительно небольших значениях ξ с резким падением после $\xi = 20$. Только вариант $\xi = 25$ статистически значимо отличается от всех остальных. Из рисунков 5.11 в) и г) видно, что с увеличением ξ растёт количество ошибочно определённых, соответственно, минорных трезвучий и минорных септаккордов.

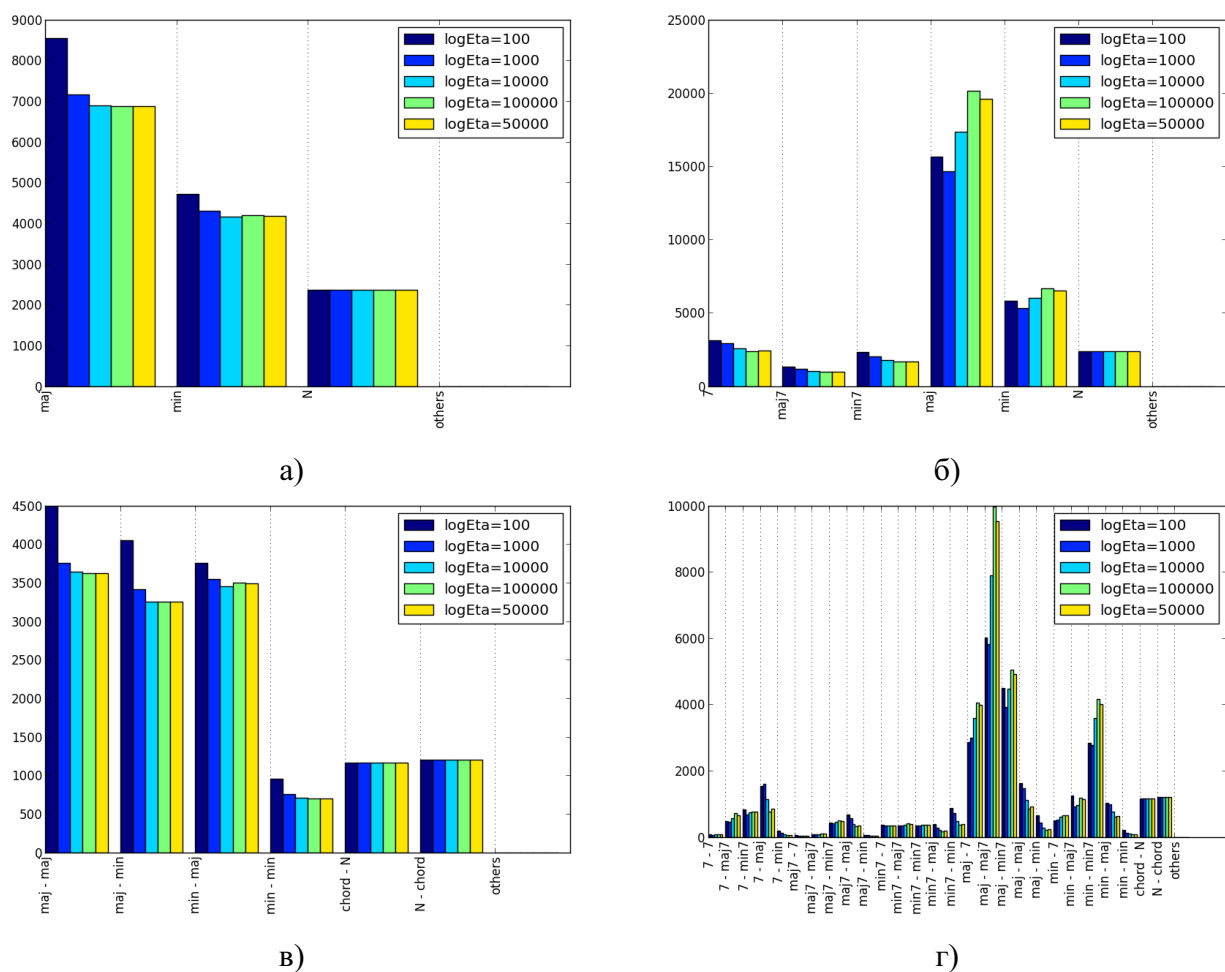
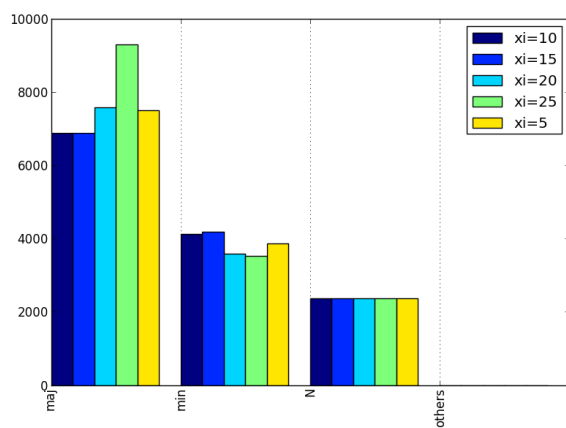


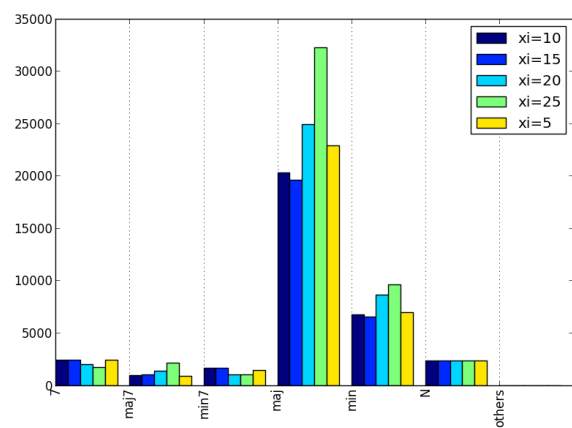
Рисунок 5.10: Диаграмма ошибок для разных значений η

Таблица 5.9: Влияние параметра ξ на качество распознавания аккордов

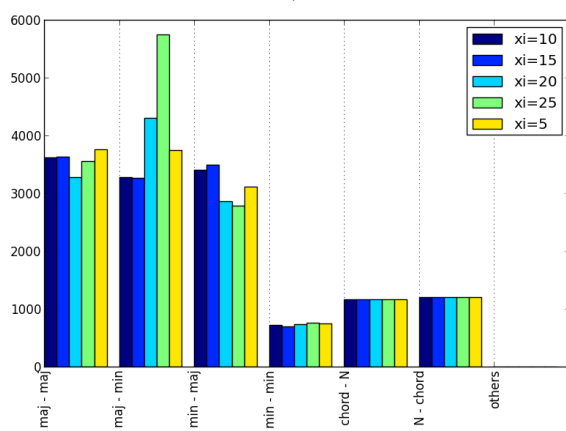
ξ	Triads	Tetrads	Сегментация
5	0.7671	0.3734	0.8067
10	0.773	0.4172	0.8138
15	0.7721	0.4310	0.8141
20	0.7705	0.3163	0.8128
25	0.7422	0.1667	0.8055



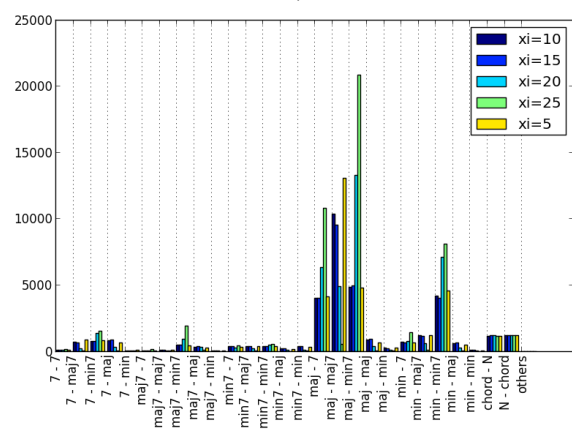
а)



б)



в)



г)

Рисунок 5.11: Диаграмма ошибок для разных значений ξ

5.3.3 Применение самоподобия

При использовании матрицы самоподобия для улучшения спектрограммы необходимо выбрать наилучшее значение для параметра ζ . Этот параметр контролирует долю столбцов спектрограммы, наиболее схожих с тем, который корректируется в данный момент.

Таблица 5.10: Влияние параметра ζ на качество распознавания аккордов

ζ	Triads	Tetrads	Сегментация
0	0.7001	0.5535	0.7827
0.05	0.7716	0.4204	0.8147
0.1	0.7720	0.4310	0.8141
0.15	0.7663	0.4313	0.8086

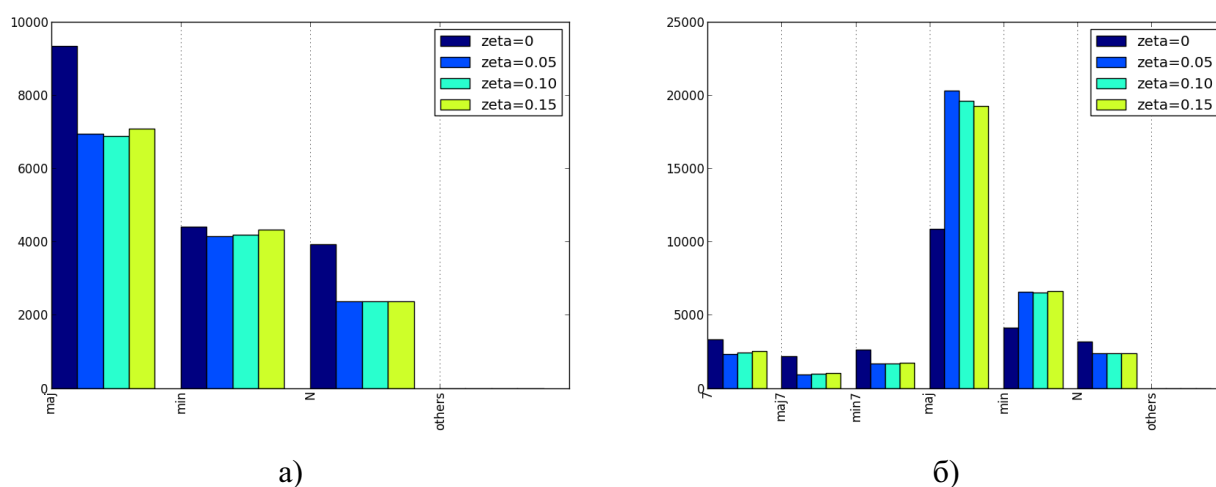


Рисунок 5.12: Диаграмма ошибок для разных значений ζ

В таблице 5.10 приведены значения среднего перекрытия и сегментации для разных значений ζ . Различия между наилучшими вариантами $\zeta = 0.05$, $\zeta = 0.1$ и $\zeta = 0.15$ не являются статистически значимыми. Значение $\zeta = 0$ соответствует отсутствию коррекции спектрограммы при помощи самоподобия. Полученные при этом значения ζ результаты оказываются существенно хуже, чем при ненулевых значениях ζ .

В работах [78] и [22] отмечалось положительное влияние от коррекции последовательности векторов признаков с использованием наиболее близких друг к другу векторов. Однако в обеих указанных работах строились матрицы самоподобия для 12-мерных хроматических векторов, в то время как в таблице 5.10 все значения получены с использованием матрицы самоподобия для столбцов спектрограммы. После применения аналогичного метода к хроматическим векторам при $\zeta = 0.08$ были получены значения для взвешенного среднего перекрытия и сегментации, соответственно, 0.7620 и 0.8119. Результат оказался статистически значимо хуже (в соответствии с критерием Вилкоксона).

Использование признаков CRP позволяет более эффективно подавлять шумовые компоненты в спектре по сравнению с применением аналога фильтра Превитт. Эти признаки были предложены в [82] для решения другой задачи, поэтому другими были и оптимальные диапазоны для параметров этих признаков. Коррекция спектрограммы с использованием очищенной матрицы самоподобия приводит к очень существенному повышению качества распознавания аккордов и, таким образом, является одним из самых важных шагов в реализованном алгоритме.

5.4 Нейронные сети

Для нейронной сети необходимо определить оптимальные значения для метапараметров, которые задаются изначально и не изменяются в процессе обучения. Это, прежде всего, конфигурация сети: количество скрытых слоёв и количество нейронов во входном и в скрытых слоях, наличие рекуррентных соединений.

Нейронная сеть была реализована на языке *Python* с использованием пакета *Theano* [97]. Обучающие данные делились на мини-пакеты по 5 векторов в каждом. Предварительное обучение каждого слоя с помощью автоассоциатора производилось в течение 15 эпох. Окончательное обучение нейронной сети также продолжалось в течение 15 эпох.

Рекуррентные соединения добавлялись к последнему из слоёв, обучаемых при помощи автоассоциаторов. При окончательном обучении рекуррентных вариантов сети сохранялась исходная последовательность обучающих векторов. При предварительном обучении слоёв и при окончательном обучении нерекуррентных вариантов обучающие векторы перемешивались в случайном порядке.

Для проведения экспериментов тестовая коллекция из 318 композиций была случайным образом поделена на 2 равные части, каждая из которых поочередно выступала в качестве обучающей и тестовой выборки.

В экспериментах предварительно вычисленные спектрограммы звукозаписей по столбцам подавались на вход нейронной сети. Описанные в параграфах 3.2 и 3.3 преобразования к спектрограмме не применялись, за исключением взятия логарифма от её компонент. Однако матрица самоподобия использовалась для сглаживания последовательности хроматических векторов, полученных на выходе нейронной сети. При определении последовательности аккордов использовались эвристики, описанные в параграфе 3.4.

5.4.1 Конфигурация нейронной сети

Были рассмотрены различные варианты нерекуррентных сетей (SDA) с 48 и с 60 входами (охватывающие, соответственно, 4 или 5 октав звукозаписи), имеющие 1, 2 или 3 скрытых слоя с разными количествами нейронов. Для всех сетей были рассмотрены также соответствующие рекуррентные варианты (RSDA). Соответствующие результаты приведены в таблице 5.11. В названии конфигурации первое число в скобках соответствует количеству входов, остальные – количеству нейронов в скрытых слоях.

Несмотря на то, что конфигурации с большим количеством нейронов показали чуть более высокие результаты, по итогам экспериментов не выявлено существенных различий в качестве распознавания аккордов между разными конфигурациями нейронной сети. Полученные результаты достаточно близки, и лишь в нескольких парах есть статистически значимые различия. При этом обучение нейронных сетей с рекуррентным слоем и последующее их тестирование отнимает значительно больше времени, чем для сетей без рекуррентного слоя (см. раздел 5.7). Количество нейронов во входном слое несущественно влияет на время работы, но большее их количество приводит к чуть лучшему результату. С учётом высказанных соображений, в дальнейших экспериментах было решено использовать конфигурацию SDA (60, 300, 300).

5.4.2 Влияние логарифмирования спектрограммы

Как видно из таблицы 5.12, применение логарифмического преобразования к столбцам спектрограммы при обучении нейронной сети и при последующем её использовании позволяет существенно улучшить качество распознавания аккордов.

Таблица 5.11: Влияние конфигурации нейронной сети на качество распознавания аккордов

Конфигурация	Triads	Tetrads	Сегментация
SDA (48, 200)	0.7639	0.6234	0.8035
SDA (48, 200, 200)	0.7616	0.6316	0.8009
SDA (60, 100)	0.7616	0.6279	0.7991
SDA (60, 200)	0.7637	0.6314	0.8010
SDA (60, 300)	0.7649	0.6322	0.8011
SDA (60, 200, 100)	0.7664	0.6389	0.8011
SDA (60, 300, 300)	0.7650	0.6363	0.8012
SDA (60, 100, 100, 100)	0.7671	0.6382	0.7996
SDA (60, 300, 300, 300)	0.7674	0.6373	0.8011
RSDA (48, 200)	0.7616	0.6226	0.7963
RSDA (48, 200, 200)	0.7660	0.6342	0.7982
RSDA (60, 300)	0.7613	0.6298	0.7955
RSDA (60, 300, 300)	0.7672	0.6373	0.7982
RSDA (60, 300, 300, 300)	0.7686	0.6360	0.7986

Таблица 5.12: Влияние логарифмирования спектрограммы до применения нейронной сети на качество распознавания аккордов

Конфигурация	Triads	Tetrads	Сегментация
SDA (60, 300, 300) nolog	0.7350	0.6050	0.7890
SDA (60, 300, 300)	0.7672	0.6373	0.7982

Таким образом, предварительная обработка данных имеет смысл также и при использовании нейронной сети для получения признаков на их основе.

5.4.3 Влияние зашумления на этапе предварительного обучения

Как описано в разделе 4.2, при предварительном обучении слоёв нейронной сети входные векторы зашумляются. В таблице 5.13 приведены результаты, полученные для конфигурации SDA (60, 300, 300) при различных типах шума и разных значениях параметра, контролирующего уровень шума: дисперсия для аддитивного шума (0 соответствует его отсутствию), вероятность обнуления компоненты входного вектора – для маскирующего шума.

Таблица 5.13: Влияние зашумления на этапе предварительного обучения нейронной сети на качество распознавания аккордов

Параметр	Тип шума	Triads	Tetrads	Сегментация
0.2	аддитивный	0.7668	0.6361	0.8013
0.2	маскирующий	0.7677	0.6336	0.8018

Ни один из вариантов зашумления (включая его отсутствие) не приводит к результату, статистически значимо отличающемуся от всех остальных.

5.4.4 Влияние циклических сдвигов на этапе тестирования

Циклический сдвиг для векторов признаков или входных данных часто применяется на этапе обучения модели, но обычно не применяется при тестировании (зачастую, например,

при использовании СММ, это излишне усложняет алгоритм). Предложенный в параграфе 4.2 метод для использования циклического сдвига на этапе тестирования модели не приводит к заметному усложнению этого этапа. Увеличение количества вычислений в 12 раз не очень заметно, учитывая относительно небольшое время обработки спектра нейронной сетью в процессе тестирования.

Таблица 5.14: Влияние циклических сдвигов на этапе тестирования нейронной сети на качество распознавания аккордов

Конфигурация	Triads	Tetrads	Сегментация
SDA (60, 300, 300) nocycle	0.7703	0.6261	0.8020
SDA (60, 300, 300)	0.7672	0.6373	0.7982

Как видно из таблицы 5.14, помимо некоторого увеличения времени обработки, это приводит даже к небольшому ухудшению результата (в метрике «Triads»). Вероятно, применение циклических сдвигов на этапе обучения нейронной сети является достаточным, и дополнительные действия на этапе тестирования не требуются.

Хроматические векторы, полученные с использованием многослойных нейронных сетей, позволяют получить сравнимые с традиционными хроматическими признаками результаты распознавания аккордов. Вместе с тем, качество полученных хроматических векторов зависит от предварительной обработки спектральных данных гораздо больше, чем от конфигурации самой нейронной сети и метода зашумления исходных данных при предварительном обучении. Добавление нейронов, слоёв и рекуррентных соединений влияет на продолжительность обучения модели в гораздо большей степени, чем на итоговое качество распознавания аккордов.

5.5 Классификация векторов признаков

Для определения последовательности аккордов по полученной последовательности хроматических векторов используется простой метод ближайшего соседа. Для этого строятся шаблонные хроматические векторы для всех аккордов из множества распознаваемых аккордов Y . При построении шаблонов используются 2 параметра: количество учитываемых обертонов и вклад h этих обертонов в шаблон (в соответствии с формулой (2.2)). Затем для каждого вектора из последовательности определяются расстояния от него до всех шаблонных векторов. Необходимо рассмотреть разные способы определения расстояния.

5.5.1 Шаблонные векторы

Поскольку первый обертон любой ступени звукоряда соответствует звуку с таким же названием, нет смысла рассматривать отдельно случай шаблонов без обертонов. Но, учитывая экспоненциальный характер убывания вклада каждого последующего обертона в шаблоны, имеет смысл рассматривать достаточно небольшое их количество. В [85] авторы ограничиваются первыми пятью обертонами. В соответствии с формулой (2.2) при $h = 0.6$ вклад пятого обертона будет составлять $0.6^5 \approx 0.078$ от вклада первого, то есть более чем в 12 раз слабее. Поэтому вряд ли имеет смысл рассматривать большее их количество.

В таблице 5.15 показаны результаты экспериментов в зависимости от количества учитываемых в шаблонах обертонов. Различия между 2 и 3 обертонами, а также между 3, 4 и 5 обертонами, не являются статистически значимыми, но максимум наблюдается при использовании в шаблонах 2 обертонов дополнительно к основному тону.

Таблица 5.15: Влияние количества обертонов в шаблонах на качество распознавания аккордов

Количество обертонов	Triads	Tetrads	Сегментация
1	0.7516	0.1586	0.8064
2	0.7720	0.4310	0.8141
3	0.7706	0.3611	0.8139
4	0.7680	0.5713	0.8134
5	0.7703	0.6173	0.8137

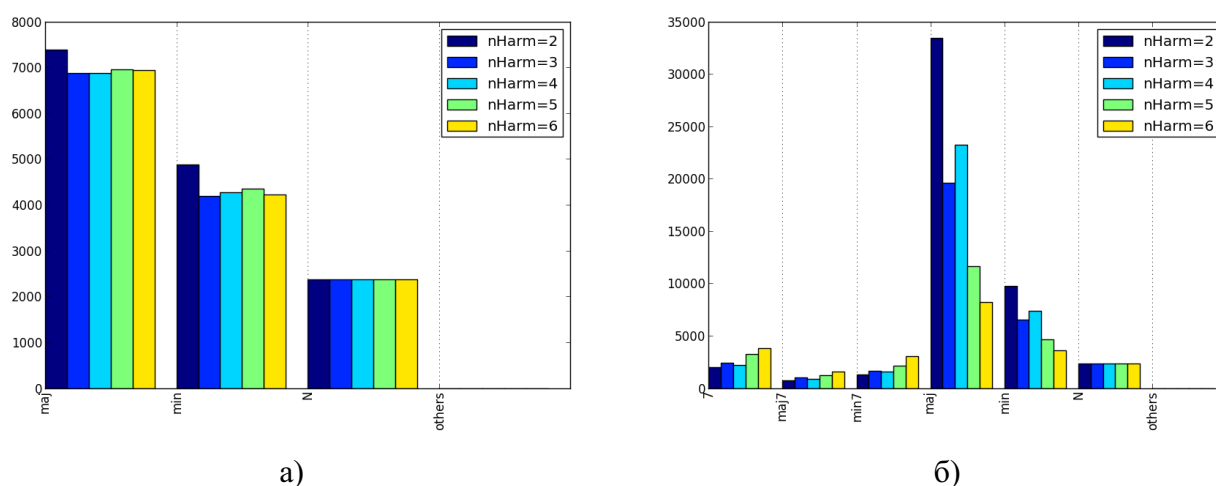


Рисунок 5.13: Диаграмма ошибок при разных количествах обертонов в шаблонах

Коэффициент h убывания вклада обертонов в [40] и [85] выбирается равным 0.6, его влияние в этих работах не исследуется. В таблице 5.16 приведены значения взвешенного среднего перекрытия и сегментации, полученные автором для разных значений h . Для случаев $h = 0.5$, $h = 0.6$, $h = 0.7$ статистически значимые различия не обнаружены, но их отличия от вариантов $h = 0.4$, $h = 0.8$ являются статистически значимыми. В целом, влияние данного параметра достаточно слабое.

Таблица 5.16: Влияние коэффициента убывания вклада обертона на качество распознавания аккордов

h	Triads	Tetrads	Сегментация
0.4	0.7660	0.2038	0.8121
0.5	0.7694	0.2881	0.8135
0.6	0.7720	0.4310	0.8141
0.7	0.7733	0.5749	0.8133
0.8	0.7708	0.6216	0.8114

Примечательно (см. рисунки 5.14, 5.13), что параметры шаблонов не оказывают почти никакого влияния в случае распознавания только мажорных и минорных трезвучий, но сильно влияют на результат при распознавании септаккордов. Это можно объяснить тем, что получаемые в реальности 12-мерные тональные векторы не являются бинарными. Определённое количество звуковой энергии приходится на все их компоненты. А поскольку шаблоны для септаккордов учитывают гармоники 4 тональных классов, почти все их компоненты оказываются ненулевыми. За счёт этого расстояние от таких шаблонов до полученных тональных векторов оказывается меньше, и септаккорд определяется там, где на самом деле звучит мажорное или

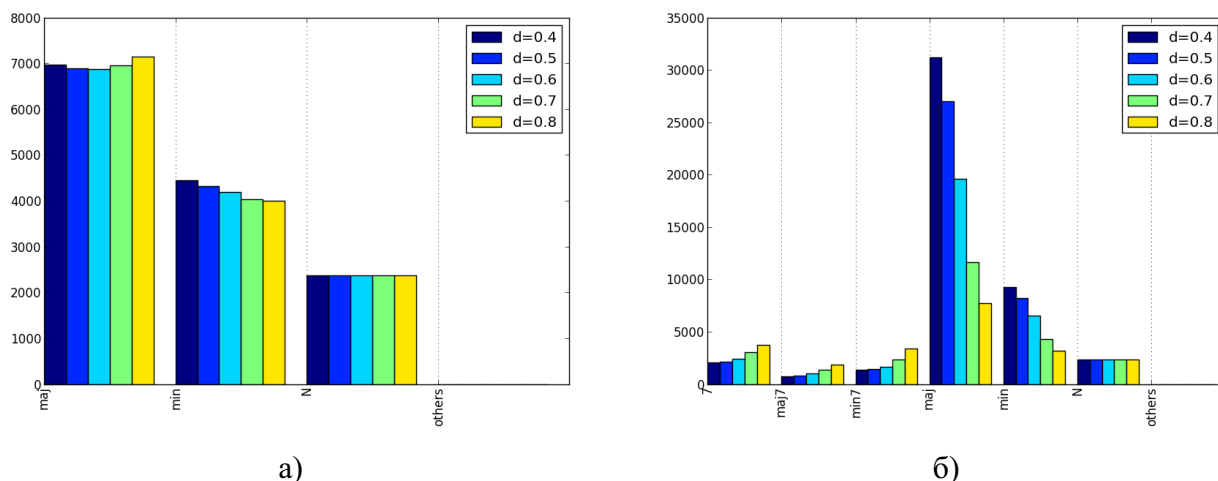


Рисунок 5.14: Диаграмма ошибок для разных значений h

минорное трезвучие. Для частичной компенсации этого эффекта количество гармоник в шаблонах для септаккордов было ограничено двумя. С увеличением же количества гармоник и вклада каждой из последующих гармоник энергия в шаблонах для трезвучий распределяется более равномерно. В результате эти шаблоны оказываются более подходящими для отделения трезвучий от септаккордов.

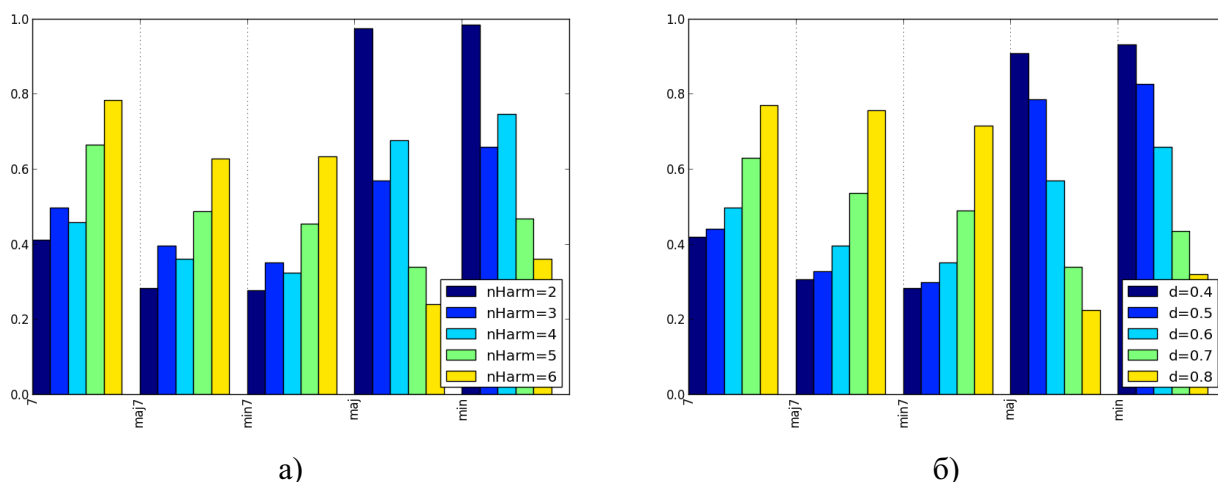


Рисунок 5.15: Нормализованная диаграмма ошибок для разных значений количества обертонов и параметра h

Вспомним, что в совокупности все фрагменты, на которых звучат септаккорды, составляют лишь чуть более 20% от длительности тестовой коллекции (см. таблицу 5.1). Поэтому падение количества ошибок на септаккордах может оказаться менее важным, чем рост количества ошибок на мажорных и минорных трезвучиях. На рисунке 5.15 высота столбцов соответствует доле фрагментов с аккордами каждого из типов, на которых была сделана ошибка распознавания. Видно, что в действительности уменьшение количества ошибок распознавания трезвучий компенсируется ростом количества ошибок распознавания септаккордов. Но из-за существенно меньшей доли фрагментов с септаккордами в коллекции значение метрики “Tetrads” растёт. Нельзя говорить, однако, что при больших значениях параметра h или количества гармоник в шаблонах алгоритм лучше распознаёт септаккорды.

Таблица 5.17: Влияние эвристик на качество распознавания аккордов

Эвристики	Triads	Tetrads	Сегментация
Без эвристик	0.7634	0.4341	0.8083
Исправление одиночных аккордов	0.7721	0.4393	0.8164
Исправление типа аккорда	0.7655	0.4270	0.8141
Обе эвристики	0.7720	0.4310	0.8141

5.5.2 Эвристики

Изменения результата при использовании предложенных эвристик показаны в таблице 5.17. Как видно, исправление типа аккорда оказывается, фактически, бесполезным. Однако исправление аккордов, которые продолжаются в течение только одной метрической доли, имеет заметный эффект и даёт статистически значимое улучшение качества распознавания аккордов.

5.5.3 Определение отсутствия аккорда

Параметр L_N , описанный в параграфе 3.4.2, позволяет определить, для каких столбцов спектрограммы будет определяться отсутствие звучащего аккорда. В таблице 5.18 приведены результаты экспериментов для разных значений L_N . При $L_N \leq 0.0025$ результат изменяться перестаёт, то есть, фактически, все столбцы спектрограммы считаются содержащими аккорд. При значениях $L_N > 0.003$ результат существенно падает, поскольку, наоборот, появляется слишком много столбцов, для которых определяется отсутствие аккорда. Это можно заметить на рисунке 5.16. Наилучший результат был получен при $L_N = 0.003$, но статистически значимых отличий между ним и результатами при меньших значениях L_N нет.

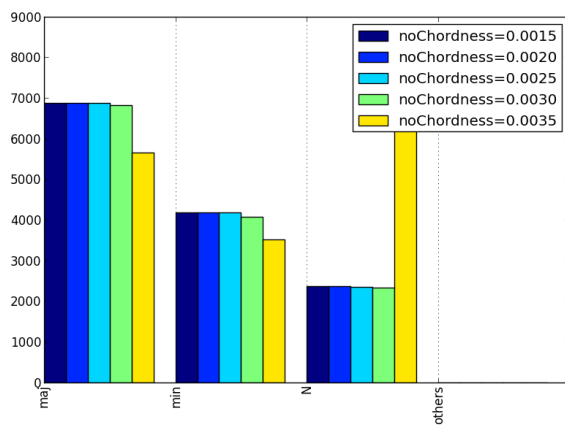
Таблица 5.18: Влияние порога для отсутствия аккорда на качество распознавания аккордов

L_N	Triads	Tetrads	Сегментация
0.0015	0.7720	0.4310	0.8141
0.0020	0.7720	0.4310	0.8141
0.0025	0.7724	0.4314	0.8141
0.0030	0.7756	0.4365	0.8140
0.0035	0.7075	0.4279	0.7720

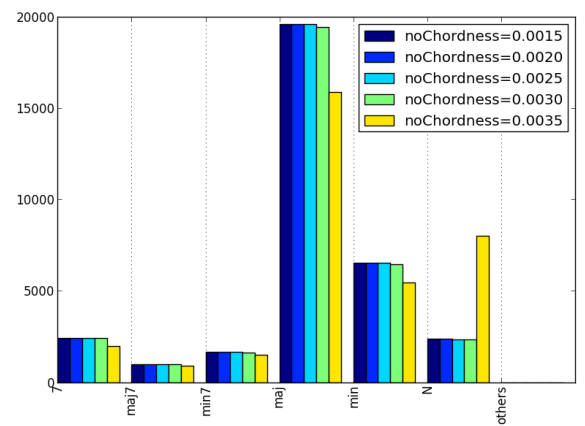
Параметры шаблонов для аккордов не оказывают практически никакого влияния на результат при распознавании только мажорных и минорных трезвучий. Для случая септаккордов они влияют только на количество ошибок, связанных с определением септаккорда вместо трезвучия, но не на качество распознавания септаккордов само по себе. Из реализованных эвристик имеет смысл применять только исправление одиночных аккордов. Пороговое значение для определения отсутствия аккорда не оказывает существенного влияния на результат.

5.6 Результаты MIREX Audio Chord Estimation 2013

Реализованный в рамках данной работы алгоритм был выставлен на ежегодное соревнование среди систем распознавания аккордов *MIREX Audio Chord Estimation 2013* [72], [70], [71]. Диаграммы результатов представлены на рисунках 5.17, 5.18 и 5.19.



а)



б)

Рисунок 5.16: Диаграмма ошибок для разных значений L_N

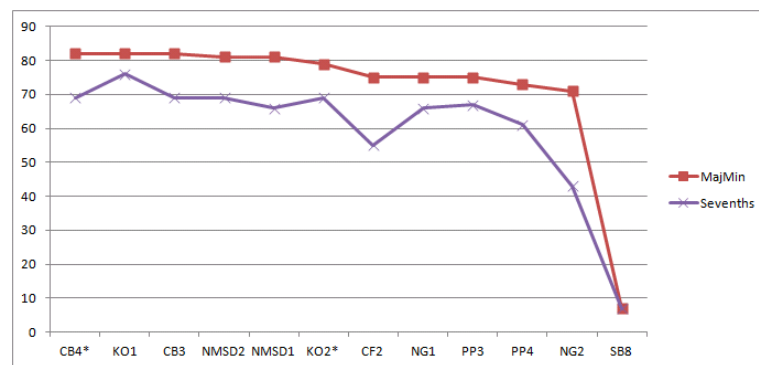


Рисунок 5.17: Результаты MIREX ACE 2013 на коллекции Mirex2009

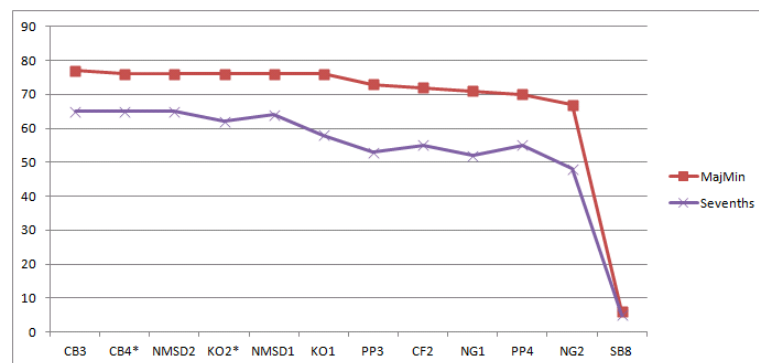


Рисунок 5.18: Результаты MIREX ACE 2013 на коллекции Billboard2012

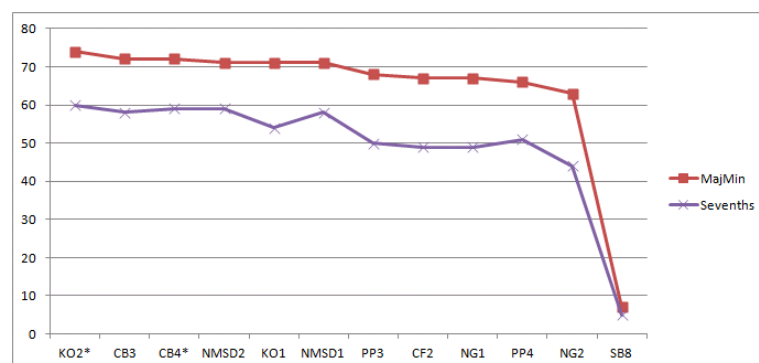


Рисунок 5.19: Результаты MIREX ACE 2013 на коллекции Billboard2013

Как видно, описанный в главе 3 метод показывает сравнимые с другими участниками результаты. При этом все остальные алгоритмы используют методы машинного обучения. Версия алгоритма под названием NG2 отличается от NG1 только тем, что содержит шаблоны для мажорного и минорного септаккордов, а также для доминантсептаккорда. За счёт ошибочного определения септаккордов вместо обычных мажорного или минорного аккорда версия NG2 показала более слабый результат.

Можно заметить, что все алгоритмы показывают более слабый результат на коллекциях *Billboard 2012* и *Billboard2013*, недоступных для участников, чем на широко используемой много лет подряд коллекции *Isophonics*. В случае, когда от алгоритмов требуется распознавать септаккорды, результаты также падают. Интересно, что даже наилучшие из алгоритмов незначительно превышают значение 0.8 для взвешенного среднего перекрытия, и до сих пор никому не удалось добиться существенного прогресса в качестве распознавания. Исключением являются случаи переобучения (пример приведен в [18], когда алгоритм обучается и тестируется на одной и той же коллекции. Очевидно, что в данном случае сравнение с результатами, полученными при обучении и тестировании на разных коллекциях, некорректны.

5.7 Быстродействие

Метод должен обладать достаточным быстродействием для того, чтобы его использование для решения реальных задач было целесообразным. В данном разделе анализируется производительность реализованного метода в зависимости от значений параметров.

Весь процесс обработки звукового файла можно разделить на 4 стадии:

1. определение частоты настройки музыкальных инструментов;
2. определение ритма при помощи внешней библиотеки;
3. вычисление спектрограммы;
4. преобразования спектрограммы и распознавание аккордов.

Процесс распознавания аккордов при наличии последовательности хроматических векторов отнимает очень небольшое время, и поэтому не был выделен в отдельную стадию. Однако вычисление преобразований спектрограммы уже отнимает заметное время. При использовании нейронной сети все дополнительные затраты времени на её обучение также могут быть отнесены к этой стадии.

На быстродействие алгоритма определения ритма автор повлиять не может. Длительность остальных стадий обработки файла зависит от параметров метода. Наибольшее влияние оказывают количество компонент преобразования постоянного качества (количество октав и количество компонент на октаву N_0) и коэффициент T , соответствующий количеству вычислений спектра звука, приходящихся на одну метрическую долю. Именно эти параметры задают количество преобразований постоянного качества, которые необходимо вычислить для получения спектрограммы звукозаписи, и количество участвующих в каждом преобразовании значений.

Полностью процесс распознавания аккордов в используемой коллекции из 318 композиций общей длительностью около 17 ч 8 мин отнимает примерно 4 ч 15 мин при следующих условиях:

- компьютер с процессором Intel Core i5 с тактовой частотой 2.8 ГГц;
- вычисления в 3 потока;
- $N_0 = 36$, охват 6 октав, начиная со звука до большой октавы (65.4 Гц);

- $T = 8$.

Взвешенное среднее перекрытие и сегментация для каждой композиции из используемой коллекции представлены в приложении А.

При тех же условиях (но с использованием 4 потоков) только наиболее продолжительные действия – определение частоты настройки и получение спектрограммы (при известных моментах начала метрических долей и без дополнительных преобразований) – занимают примерно 3 ч 40 мин. В таблице 5.19 показано время выполнения этих же действий при разных значениях N_0 . При использовании $N_0 = 60$, несмотря на большее время работы, прирост качества распознавания аккордов практически не заметен (см. таблицу 5.6). При $N_0 = 12$ выигрыш во времени невелик, при этом ухудшение результата очень заметно.

Таблица 5.19: Совокупная продолжительность определения частоты настройки и вычисления спектрограммы

N_0	Время
12	2 ч 59 мин 2 с
36	3 ч 37 мин 47 с
60	4 ч 23 мин 27 с

Зависимость времени вычисления от коэффициента T была исследована при $N_0 = 36$, охват 4 октавы, начиная со звука *до* большой октавы (65.4 Гц). Результаты показаны в таблице 5.20. В отличие от N_0 , параметр T не так существенно влияет на качество распознавания аккордов (см. таблицу 5.5). Поэтому алгоритм можно использовать при $T = 2$, когда быстроедействие алгоритма важнее, чем небольшой выигрыш в качестве распознавания.

Таблица 5.20: Совокупная продолжительность определения аккордов при разных значениях параметра T

T	Время
2	1 ч 31 мин 4 с
4	1 ч 58 мин 51 с
8	3 ч 1 мин 58 с

Время, затрачиваемое на обучение нейронной сети, оказывается на практике достаточно заметным. Однако, как видно из таблицы 5.21, добавление каждого нового слоя существенно увеличивает время обучения при незначительном улучшении результата (см. таблицу 5.11). Добавление рекуррентных соединений резко увеличивает время обучения по сравнению с аналогичной нерекуррентной сетью. Кроме того, из-за наличия зависимости от значений предыдущего шага не удаётся эффективно вычислять выходные значения во время тестов, из-за чего продолжительность тестов также резко возрастает.

5.8 Выводы

1. Произведён анализ влияния параметров реализованного метода на качество распознавания аккордов и скорость работы метода. Определена степень влияния каждого из параметров на качество распознавания аккордов.
2. Определены параметры, наиболее сильно влияющие на быстроедействие метода; показано, что можно добиться прироста скорости при незначительном снижении качества распознавания.

Таблица 5.21: Совокупная продолжительность обучения и тестов при разных конфигурациях нейронной сети

Конфигурация	Продолжительность обучения	Продолжительность тестов
SDA (60, 100)	43 мин 49 с	4 мин 38 с
SDA (60, 200)	1 ч 2 мин 19 с	4 мин 56 с
SDA (60, 300)	1 ч 20 мин 48 с	5 мин 17 с
SDA (60, 200, 100)	2 ч 19 мин 52 с	7 мин 6 с
SDA (60, 300, 300)	6 ч 21 мин 28 с	8 мин 18 с
SDA (60, 300, 300, 300)	15 ч 21 мин 19 с	13 мин 21 с
RSDA (60, 300)	6 ч 55 мин 43 с	1 ч 52 мин 01 с

3. Реализованный алгоритм показал хороший результат в международном соревновании среди алгоритмов распознавания аккордов *MIREX Audio Chord Estimation 2013*.

Заключение

Основные результаты работы заключаются в следующем.

1. На основе анализа свойств музыкальных звукозаписей был разработан метод для более точного выделения в спектре звука компонент, соответствующих музыкальным инструментам.
2. Исследование показало, что глубокие нейронные сети в применении к получению музыкальных признаков для распознавания аккордов могут показывать хорошие результаты, сравнимые с результатами традиционных подходов к получению признаков.
3. Численные исследования показали, что реализованные в рамках работы подходы позволяют добиться качества распознавания аккордов, сравнимого с лучшими из известных алгоритмов, при достаточно высокой скорости обработки.
4. Для выполнения поставленных задач был создан программный комплекс на языках Java и Python, свободно доступный через интернет.

Разработанный в рамках работы метод для более точного выделения в спектре звука компонент, соответствующих музыкальным инструментам, может быть использован как часть других алгоритмов музыкального информационного поиска, работающих с содержимым звукозаписей. Так, в задаче определения мелодии более точная спектрограмма позволит аккуратнее выделить воспроизведённые ноты. Алгоритмы для определения тональности и нахождения повторений в музыке также выиграют от повышения качества используемых спектральных данных.

Хроматические признаки, полученные из спектра с использованием предварительно обученных многослойных нейронных сетей, позволяют добиться по крайней мере такого же качества распознавания аккордов, как и в случае использования традиционных хроматических признаков. Однако, в отличие от традиционных подходов к получению признаков, для использования нейронной сети необходимо некоторое количество размеченных данных, а также существенное время на её обучение. Это делает данный подход значительно менее удобным для применения. Безусловно, глубокое обучение является весьма перспективной областью для дальнейших исследований, но его применение к звуковым данным пока не приводит к заметному скачку в качестве работы алгоритмов.

В нынешних условиях любые методы обработки информации должны быть нацелены на работу с очень большими её объёмами. Вероятно, именно чрезмерное количество времени, необходимое для анализа миллионов композиций, препятствует широкому применению методов распознавания аккордов. С другой стороны, малый объём размеченных и доступных для обучения данных затрудняет широкое применение методов, основанных на алгоритмах машинного обучения. Именно поэтому, как представляется автору, разработка достаточно быстрого метода распознавания аккордов, не использующего машинное обучение, является важным шагом на пути к обработке больших музыкальных коллекций. Реализованный в рамках работы алгоритм, использующий многопоточную обработку, является достаточно быстрым по сравнению с аналогами. Но для более комфортной работы пользователей с программой было бы желательно снизить время обработки одной композиции до нескольких секунд.

Кроме того, использованные в работе подходы могут быть применены в работе компаний, имеющих дело с многочисленными цифровыми звукозаписями: музыкальные магазины, музыкальные потоковые сервисы, площадки для размещения собственных музыкальных записей. Как правило, такие компании обладают достаточными вычислительными мощностями для обработки большого набора звуковых файлов, а от алгоритмов не требуется работы в реальном времени. Анализ содержимого звукозаписей может улучшить качество поиска и рекомендации музыки в сервисах этих компаний, предоставить новые критерии для поиска, дать большее количество информации о музыке для пользователей.

Следующие направления дальнейшей работы видятся автору наиболее перспективными.

1. Учёт контекста при распознавании аккордов, как локального, так и глобального. Дополнительная информация о структуре композиции, о её тональности, о жанре, об используемых инструментах может быть полезна в некоторых случаях.
2. Повышение быстродействия существующих алгоритмов. Этому вопросу нечасто уделяется внимание в работах, зачастую данные о времени работы алгоритмов не приводятся. Однако при использовании алгоритма в реальной жизни вопрос быстродействия выходит на первый план, иногда оказываясь более важным, чем качество решения задачи алгоритмом. Параллельная обработка звуковых данных, эксперименты на больших коллекциях звукозаписей, оптимизация используемых алгоритмов – важные темы с точки зрения применения описанных в работе методов.

Список рисунков

3.1	Фрагменты спектрограммы <i>The Beatles – Love Me Do</i> при $T = 1$ (вверху), $T = 8$ (в середине), $T = 8$ после сглаживания с $w = 19$ (внизу).	32
3.2	Фрагменты спектрограммы <i>The Beatles – Love Me Do</i> при: а) $N_0 = 12$; б) $N_0 = 60$	33
3.3	Фрагменты спектрограммы <i>The Beatles – Love Me Do</i> : а) до применения фильтра Превитт; б) после применения фильтра Превитт.	34
3.4	Матрица самоподобия для композиции <i>The Beatles – Love Me Do</i>	35
3.5	Фрагменты спектрограммы <i>The Beatles – Love Me Do</i> : а) без использования самоподобия; б) после коррекции с использованием самоподобия.	36
4.1	Многослойный автоассоциатор	40
4.2	Многослойная нейронная сеть	40
4.3	Многослойная нейронная сеть с рекуррентными соединениями	41
5.1	Сопоставление последовательностей аккордов.	46
5.2	Диаграмма ошибок для разных методов определения ритма	51
5.3	Диаграмма ошибок для разных значений d	52
5.4	Диаграмма ошибок для разных методов определения частоты настройки	53
5.5	Диаграмма ошибок для разных значений w при $T = 2$	54
5.6	Диаграмма ошибок для разных значений w при $T = 4$	55
5.7	Диаграмма ошибок для разных значений w при $T = 8$	55
5.8	Диаграмма ошибок для разных значений N_0	56
5.9	Диаграмма ошибок для разных способов подавления шумовых звуков	58
5.10	Диаграмма ошибок для разных значений η	59
5.11	Диаграмма ошибок для разных значений ξ	60
5.12	Диаграмма ошибок для разных значений ζ	61
5.13	Диаграмма ошибок при разных количествах обертонов в шаблонах	65
5.14	Диаграмма ошибок для разных значений h	66
5.15	Нормализованная диаграмма ошибок для разных значений количества обертонов и параметра h	66
5.16	Диаграмма ошибок для разных значений L_N	68
5.17	Результаты MIREX ACE 2013 на коллекции Mirex2009	68
5.18	Результаты MIREX ACE 2013 на коллекции Billboard2012	68
5.19	Результаты MIREX ACE 2013 на коллекции Billboard2013	68

Список таблиц

5.1	Совокупная продолжительность звучания аккордов в секундах	49
5.2	Влияние алгоритма определения ритма на качество распознавания аккордов . .	50
5.3	Влияние задержки относительно моментов начала метрических долей на каче- ство распознавания аккордов	51
5.4	Влияние алгоритма определения частоты настройки на качество распознавания аккордов	53
5.5	Влияние параметров T и w на качество распознавания аккордов	54
5.6	Влияние параметра N_0 на качество распознавания аккордов	56
5.7	Влияние разных способов подавления шумовых звуков на качество распозна- вания аккордов	57
5.8	Влияние параметра η на качество распознавания аккордов	58
5.9	Влияние параметра ξ на качество распознавания аккордов	59
5.10	Влияние параметра ζ на качество распознавания аккордов	61
5.11	Влияние конфигурации нейронной сети на качество распознавания аккордов .	63
5.12	Влияние логарифмирования спектрограммы до применения нейронной сети на качество распознавания аккордов	63
5.13	Влияние зашумления на этапе предварительного обучения нейронной сети на качество распознавания аккордов	63
5.14	Влияние циклических сдвигов на этапе тестирования нейронной сети на каче- ство распознавания аккордов	64
5.15	Влияние количества обертонов в шаблонах на качество распознавания аккордов	65
5.16	Влияние коэффициента убывания вклада обертона на качество распознавания аккордов	65
5.17	Влияние эвристик на качество распознавания аккордов	67
5.18	Влияние порога для отсутствия аккорда на качество распознавания аккордов .	67
5.19	Совокупная продолжительность определения частоты настройки и вычисления спектрограммы	70
5.20	Совокупная продолжительность определения аккордов при разных значениях параметра T	70
5.21	Совокупная продолжительность обучения и тестов при разных конфигурациях нейронной сети	71
A.1	Результаты распознавания аккордов	84

Литература

1. *И. А. Алдошина Р. Приттс.* Музыкальная акустика / Под ред. Т. И. Кий. — Санкт-Петербург "Композитор 2006.
2. *Рабинер Л. Гоулд Б.* Теория и применение цифровой обработки сигналов / Ed. by Л. Якименко. — Москва "МИР 1978.
3. *Оппенгейм А. Шафер Р.* Цифровая обработка сигналов / Ed. by О. Н. Кулешова. — Москва "Техносфера 2006.
4. *А. Вахромеев В.* Элементарная теория музыки / Под ред. К. Соловьева. — Москва "Государственное музыкальное издательство 1961.
5. *В. Способин И.* Элементарная теория музыки / Ed. by В. Григоренко. — Москва "КИФАРА 2012.
6. Большая советская энциклопедия. / Под ред. А. М. Прохоров. — 3-е изд. — М.: Советская энциклопедия, 1972. — Т. 9.
7. *И. Рабинер.* Скрытые марковские модели и их применение в избранных приложениях при распознавании речи: Обзор // *ТИИЭР*. — 1989. — Т. 77, № 2.
8. *A. Ng.* CS294A Lecture Notes. Sparse autoencoder. — 2014. — March. <http://www.stanford.edu/class/cs294a/sparseAutoencoder.pdf>.
9. Ableton Live 9. — 2014. — March. <https://www.ableton.com/en/live/>.
10. Analyzing Chroma Feature Types for Automated Chord Recognition / N. Jiang, P. Grosche, V. Konz, M. Müller // *Proceedings of the AES 42nd International Conference: Semantic Audio*. — Ilmenau, Germany: AES, 2011. — Pp. 285–294.
11. AnySong Chord Recognition. — 2014. — March. <https://play.google.com/store/apps/details?id=com.musprojects.chord>.
12. *Aono Y., Katayose H., Inokuchi S.* A Real-time Session Composer with Acoustic Polyphonic Instruments // *Proceedings of ICMC 1998*. — 1998. — Pp. 236–239.
13. Automatic Chord Recognition Based on Probabilistic Integration of Chord Transition and Bass Pitch Estimation / K. Sumi, K. Itoyama, K. Yoshii et al. // *Proceedings of the 9th International Conference on Music Information Retrieval*. — Philadelphia, USA: 2008. — September 14-18. — Pp. 39–44. — http://ismir2008.ismir.net/papers/ISMIR2008_236.pdf.
14. Automatic Chord Transcription with Concurrent Recognition of Chord Symbols and Boundaries / T. Yoshioka, T. Kitahara, K. Komatani et al. // *Proceedings of the 5th International Conference on Music Information Retrieval*. — Barcelona, Spain: 2004. — October 10-14. — <http://ismir2004.ismir.net/proceedings/p020-page-100-paper149.pdf>.

15. Beat Tracking for Multiple Applications: A Multi-Agent System Architecture With State Recovery / J. L. Oliveira, M. E. P. Davies, F. Gouyon, L. P. Reis // *IEEE Transactions on Audio, Speech & Language Processing*. — 2012. — Vol. 20, no. 10. — Pp. 2696–2706.
16. Bello J. P., Pickens J. A Robust Mid-Level Representation for Harmonic Content in Music Signals // *Proceedings of the 6th International Conference on Music Information Retrieval*. — London, UK: 2005. — September 11-15. — Pp. 304–311. — <http://ismir2005.ismir.net/proceedings/1038.pdf>.
17. Bengio Yoshua. Learning Deep Architectures for AI // *Found. Trends Mach. Learn.* — 2009. — January. — Vol. 2, no. 1. — Pp. 1–127. <http://dx.doi.org/10.1561/22000000006>.
18. Boulanger-Lewandowski N., Bengio Y., Vincent P. Audio chord recognition with recurrent neural networks // *Proceedings of the 14th International Society for Music Information Retrieval Conference*. — 2013. — November 4-8. — http://www.ppgia.pucpr.br/ismir2013/wp-content/uploads/2013/09/243_Paper.pdf.
19. Brown J., Puckette M. S. An efficient algorithm for the calculation of a constant Q transform // *Journal of the Acoustical Society of America*. — 1992. — November. — Vol. 92, no. 5. — Pp. 2698–2701.
20. Burgoyne J. A., Wild J., Fujinaga I. An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis // *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011* / Ed. by A. Klapuri, C. Leider. — University of Miami, 2011. — Pp. 633–638.
21. Chew E. Towards a Mathematical Model of Tonality: Ph.D. thesis / Massachusetts Institute of Technology. — 2000. — Feb. <http://www-bcf.usc.edu/~echew/papers/Dissertation2000/ec-dissertation.pdf>.
22. Cho T., Bello J. P. A Feature Smoothing Method for Chord Recognition Using Recurrence Plots // *Proceedings of the 12th International Society for Music Information Retrieval Conference*. — Miami (Florida), USA: 2011. — October 24-28. — Pp. 651–656. — <http://ismir2011.ismir.net/papers/OS8-4.pdf>.
23. Cho T., Weiss R.J., Bello J.P. Exploring common variations in state of the art chord recognition systems // *Proceedings of the Sound and Music Computing Conference (SMC)*. — Barcelona, Spain: 2010. — July. — Pp. 1–8.
24. Chord Detector. — 2014. — March. <http://www.chord-detector.com/wordpress/apps/chorddetector/>.
25. Chord Recognition Using Duration-explicit Hidden Markov Models / R. Chen, W. Shen, A. Srinivasamurthy, P. Chordia // *Proceedings of the 13th International Society for Music Information Retrieval Conference*. — Porto, Portugal: 2012. — October 8-12. — <http://ismir2012.ismir.net/event/papers/445-ismir-2012.pdf>.
26. Chordify. — 2014. — March. <http://chordify.net/>.
27. Concurrent Estimation of Chords and Keys from Audio / T. Rocher, M. Robine, P. Hanna, L. Oudre // *Proceedings of the 11th International Society for Music Information Retrieval Conference*. — Utrecht, The Netherlands: 2010. — August 9-13. — Pp. 141–146. — <http://ismir2010.ismir.net/proceedings/ismir2010-26.pdf>.

28. A Cross-Validated Study of Modelling Strategies for Automatic Chord Recognition in Audio / J. A. Burgoyne, L. Pugin, C. Kereliuk, I. Fujinaga // *Proceedings of the 8th International Conference on Music Information Retrieval*. — Vienna, Austria: 2007. — September 23-27. — Pp. 251–254. — http://ismir2007.ismir.net/proceedings/ISMIR2007_p251_burgoyne.pdf.
29. D. Freedman M. Analysis of Musical Instrument Tones // *The Journal of the Acoustical Society of America*. — 1967. — Vol. 41, no. 4A. — Pp. 793–806. <http://link.aip.org/link/?JAS/41/793/1>.
30. Davies M. E. P., Plumbley M. D. Context-Dependent Beat Tracking of Musical Audio // *Trans. Audio, Speech and Lang. Proc.* — 2007. — March. — Vol. 15, no. 3. — Pp. 1009–1020. <http://dx.doi.org/10.1109/TASL.2006.885257>.
31. De Haas W. B., Magalhães J. P., Wiering F. Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge // *Proceedings of the 13th International Society for Music Information Retrieval Conference*. — Porto, Portugal: 2012. — October 8-12. — <http://ismir2012.ismir.net/event/papers/295-ismir-2012.pdf>.
32. Dixon S. Evaluation of the Audio Beat Tracking System BeatRoot // *Journal of New Music Research*. — 2007. — March. — Vol. 36, no. 1. — Pp. 39–50.
33. Downie S. J. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research // *Acoustical Science and Technology*. — 2008. — Vol. 29, no. 4. — Pp. 247–255. http://www.jstage.jst.go.jp/article/ast/29/4/29_247/_article.
34. Ellis D. P. W. Beat tracking by dynamic programming // *Journal of New Music Research*. — 2007. — Vol. 36(1). — Pp. 51–60.
35. Elman J. L. Finding structure in time // *Cognitive Science*. — 1990. — Vol. 14, no. 2. — Pp. 179–211. <http://groups.lis.illinois.edu/amag/langev/paper/elman90findingStructure.html>.
36. Fitzgerald D. Harmonic/Percussive Separation using Median Filtering // *Audio*. — 2010. — no. 1. — Pp. 10–13. <http://arrow.dit.ie/argart/9/>.
37. Fletcher H. Munson W. A. Loudness, Its Definition, Measurement and Calculation // *The Journal of the Acoustical Society of America*. — 1933. — October. — Vol. 5, no. 2. — Pp. 82–108.
38. Ghahramani Z. An Introduction to Hidden Markov Models and Bayesian Networks // *IJPRAI*. — 2001. — Vol. 15, no. 1. — Pp. 9–42.
39. Glazyrin N., Klepinin A. Chord Recognition using Prewitt Filter and Self-Similarity // *Proceedings of the 9th Sound and Music Computing Conference*. — Copenhagen, Denmark: 2012. — July. — Pp. 480–485.
40. Gómez E. Tonal Description of Music Audio Signals: Ph.D. thesis / Universitat Pompeu Fabra. — 2006. <files/publications/emilia-PhD-2006.pdf>.
41. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // *Proceedings of the IEEE*. — 1998. — November. — Vol. 86, no. 11. — Pp. 2278–2324.

42. *H. Fastl E. Zwicker*. Psychoacoustics - Facts and Models / Ed. by Manfred R. Schroeder Thomas S. Huang, Teuvo Kohonen. — Springer-Verlag Berlin Heidelberg, 2007.
43. *Ni Y., Mcvicar M., Santos-Rodriguez R., De Bie T.* Harmony Progression Analyzer for MIREX 2011. — 2011. <http://www.music-ir.org/mirex/abstracts/2011/NMSD1.pdf>.
44. *Harte C.* Towards Automatic Extraction of Harmony Information from Music Signals: Ph.D. thesis / Queen Mary University of London, Centre for Digital Music. — 2010.
45. *Harte C., Sandler M., Gasser M.* Detecting harmonic change in musical audio // Proceedings of the 1st ACM workshop on Audio and music computing multimedia. — AMCMM '06. — New York, NY, USA: ACM, 2006. — Pp. 21–26. <http://doi.acm.org/10.1145/1178723.1178727>.
46. *Harte C. A., Sandler M.* Automatic chord identification using a quantised chromagram // *Proc. of the 118th Convention. of the AES*. — 2005.
47. *Humphrey E.J., Cho T., Bello J.P.* Learning a robust tonnetz-space transform for automatic chord recognition // Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-12). — Kyoto, Japan: 2012. — May. — Pp. 453–456.
48. *Humphrey E. J., Bello J. P., LeCun Y.* Feature learning and deep architectures: new directions for music informatics // *Journal of Intelligent Information Systems*. — 2013. — Pp. 1–21. <http://dx.doi.org/10.1007/s10844-013-0248-5>.
49. *Humphrey Eric J., Glennon Aron P., Bello Juan Pablo.* Non-Linear Semantic Embedding for Organizing Large Instrument Sample Libraries. // ICMLA (2) / Ed. by Xue wen Chen, Tharam S. Dillon, Hisao Ishbuchi et al. — IEEE Computer Society, 2011. — Pp. 142–147.
50. *J. Pauwels G. Peeters.* Evaluating automatically estimated chord sequences // Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-13). — Vancouver, Canada: 2013. — May 26-31.
51. *Khadkevich M., Omologo M.* Phase-change based tuning for automatic chord recognition. // Proceedings of the 12th International Conference on Digital Audio Effects of DAFX. — Como, Italy: 2009. — September 1-4.
52. *Khadkevich M., Omologo M.* Use of Hidden Markov Models and Factored Language Models for Automatic Chord Recognition // Proceedings of the 10th International Society for Music Information Retrieval Conference. — Kobe, Japan: 2009. — October 26-30. — Pp. 561–566. — <http://ismir2009.ismir.net/proceedings/OS7-4.pdf>.
53. *Khadkevich M., Omologo M.* Time-frequency reassigned features for automatic chord recognition // Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic. — IEEE, 2011. — Pp. 181–184.
54. *Kodera K. Gendrin R. Villedary C.* Analysis of time-varying signals with small BT values // *Acoustics, Speech and Signal Processing, IEEE Transactions on*. — 1978. — Vol. 26, no. 1. — Pp. 64–76.
55. *Lafferty J. D., McCallum A., Pereira F. C. N.* Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data // Proceedings of the Eighteenth International

- Conference on Machine Learning. — ICML '01. — San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. — Pp. 282–289. <http://dl.acm.org/citation.cfm?id=645530.655813>.
56. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures / A. Berenzweig, B. Logan, D. P. W. Ellis, B. P. W. Whitman // *Comput. Music J.* — 2004. — June. — Vol. 28, no. 2. — Pp. 63–76. <http://dx.doi.org/10.1162/014892604323112257>.
 57. Lee K. Automatic chord recognition from audio using enhanced pitch class profile // *ICMC Proceedings*. — 2006.
 58. Lee K. A System for Automatic Chord Transcription from Audio Using Genre-Specific Hidden Markov Models // *Adaptive Multimedial Retrieval: Retrieval, User, and Semantics* / Ed. by Nozha Boujemaa, Marcin Detyniecki, Andreas Nijrnberger. — Springer Berlin Heidelberg, 2008. — Vol. 4918 of *Lecture Notes in Computer Science*. — Pp. 134–146.
 59. Lee K., Slaney M. A Unified System for Chord Transcription and Key Extraction Using Hidden Markov Models // *Proceedings of the 8th International Conference on Music Information Retrieval*. — Vienna, Austria: 2007. — September 23–27. — Pp. 245–250. — http://ismir2007.ismir.net/proceedings/ISMIR2007_p245_lee.pdf.
 60. Lerch A. On the Requirement of Automatic Tuning Frequency Estimation // *International Symposium/Conference on Music Information Retrieval*. — 2006. — Pp. 212–215.
 61. Lerch A. Audio content analysis: an introduction. — John Wiley & Sons, Inc., Hoboken, New Jersey, 2012.
 62. Lerdahl F. Tonal Pitch Space. — Oxford University Press, USA, 2001.
 63. Leveraging Noisy Online Databases for Use in Chord Recognition / M. Mcvicar, Y. Ni, R. Santos-Rodriguez, T. De Bie // *Proceedings of the 12th International Society for Music Information Retrieval Conference*. — Miami (Florida), USA: 2011. — October 24–28. — Pp. 639–644. — <http://ismir2011.ismir.net/papers/OS8-2.pdf>.
 64. Levitin Daniel J. This is Your Brain on Music: Understanding a Human Obsession. — Atlantic Books Ltd., 2006.
 65. Logan B. Mel Frequency Cepstral Coefficients for Music Modeling // *Proceedings of the 1st International Conference on Music Information Retrieval*. — Plymouth (Massachusetts), USA: 2000. — October 23. — http://ismir2000.ismir.net/papers/logan_paper.pdf.
 66. MIDI Specifications. — 2014. — March. <http://www.midi.org/techspecs/>.
 67. MIDI Tuning Messages. — 2014. — March. <http://www.midi.org/techspecs/midituning.php/>.
 68. MIREX 2012: Audio Chord Description - MIREX09 Dataset. — 2013. — November. http://nema.lis.illinois.edu/nema_out/mirex2012/results/ace/mrx/.
 69. MIREX 2012: Audio Chord Description - McGill Dataset. — 2013. — November. http://nema.lis.illinois.edu/nema_out/mirex2012/results/ace/mcg/.
 70. MIREX 2013: Audio Chord Estimation Results Billboard 2012. — 2014. — March. http://www.music-ir.org/mirex/wiki/2013:Audio_Chord_Estimation_Results_Billboard_2012.

71. MIREX 2013: Audio Chord Estimation Results Billboard 2013. — 2014. — March. http://www.music-ir.org/mirex/wiki/2013:Audio_Chord_Estimation_Results_Billboard_2013.
72. MIREX 2013: Audio Chord Estimation Results MIREX 2009. — 2014. — March. http://www.music-ir.org/mirex/wiki/2013:Audio_Chord_Estimation_Results_MIREX_2009.
73. MIREX Home Page. — 2014. — March. http://www.music-ir.org/mirex/wiki/MIREX_HOME/.
74. *Maas A. Le Q. O'Neil T. Vinyals O. Nguyen P. Ng A.* Recurrent Neural Networks for Noise Reduction in Robust ASR // Proceedings of INTERSPEECH (2012). — 2012.
75. *Martin K. D.* Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing. — 1996.
76. *Mauch M.* Automatic Chord Transcription from Audio Using Computational Models of Musical Context: Ph.D. thesis / Queen Mary University of London. — 2010.
77. *Mauch M., Dixon S.* A Discrete Mixture Model for Chord Labelling // Proceedings of the 9th International Conference on Music Information Retrieval. — Philadelphia, USA: 2008. — September 14-18. — Pp. 45–50. — http://ismir2008.ismir.net/papers/ISMIR2008_214.pdf.
78. *Mauch M., Dixon S.* Approximate Note Transcription for the Improved Identification of Difficult Chords // Proceedings of the 11th International Society for Music Information Retrieval Conference. — Utrecht, The Netherlands: 2010. — August 9-13. — Pp. 135–140. — <http://ismir2010.ismir.net/proceedings/ismir2010-25.pdf>.
79. *Mauch M., Noland K., Dixon S.* Using Musical Structure to Enhance Automatic Chord Transcription // Proceedings of the 10th International Society for Music Information Retrieval Conference. — Kobe, Japan: 2009. — October 26-30. — Pp. 231–236. — <http://ismir2009.ismir.net/proceedings/PS2-7.pdf>.
80. Minimum Classification Error Training to Improve Isolated Chord Recognition / J. Reed, Y. Ueda, S. Siniscalchi et al. // Proceedings of the 10th International Society for Music Information Retrieval Conference. — Kobe, Japan: 2009. — October 26-30. — Pp. 609–614. — <http://ismir2009.ismir.net/proceedings/PS4-6.pdf>.
81. *Müller M.* Information retrieval for music and motion. — Springer-Verlag Berlin Heidelberg, 2007.
82. *Müller M., Ewert S., Kreuzer S.* Making chroma features more robust to timbre changes // Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. — ICASSP '09. — Washington, DC, USA: IEEE Computer Society, 2009. — Pp. 1877–1880. <http://dx.doi.org/10.1109/ICASSP.2009.4959974>.
83. *Moorer J. A.* — On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer. — Master's thesis, Stanford University, Stanford, CA, 1975. <https://ccrma.stanford.edu/files/papers/stanm3.pdf>.
84. OMRAS2 Metadata Project 2009 / M. Mauch, C. Cannam, M. Davies et al. // Late-breaking session at the 10th International Conference on Music Information Retrieval, Kobe, Japan. — 2009.

85. Oudre L., Grenier Y., Févotte C. Template-Based Chord Recognition : Influence of the Chord Types // Proceedings of the 10th International Society for Music Information Retrieval Conference. — Kobe, Japan: 2009. — October 26-30. — Pp. 153–158. — <http://ismir2009.ismir.net/proceedings/PS1-17.pdf>.
86. P. Vincent H. Larochelle I. Lajoie Y. Bengio, Manzagol P.-A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion // *The Journal of Machine Learning Research*. — 2010. — Vol. 11. — Pp. 3371–3408.
87. Papadopoulos H., Peeters G. Large-Scale Study of Chord Estimation Algorithms Based on Chroma Representation and HMM // CBMI / Ed. by Jenny Benois-Pineau. — IEEE, 2007. — Pp. 53–60.
88. Peeters G. Musical key estimation of audio signal based on hidden Markov modeling of chroma vectors // Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06. — 2006. — Pp. 127–131.
89. R. Cohn. Introduction to Neo-Riemannian Theory: A Survey and a Historical Perspective // *Journal of Music Theory*. — 1998. — Vol. 42, no. 2. — Pp. 167–180.
90. RWC Music Database: Popular, Classical and Jazz Music Databases / M. Goto, H. Hashiguchi, T. Nishimura, R. Oka // ISMIR. — 2002.
91. Schüler N. Reflections on the history of computer-assisted music analysis I: predecessors and the beginnings. // *Muzikoloski zbornik*. — 2005. — Vol. 41. — Pp. 31–43. <http://uweb.txstate.edu/~ns13/Schuler-CAMA-I.pdf>.
92. Separation of a Monaural Audio Signal into Harmonic/Percussive Components by Complementary Diffusion on Spectrogram / N. Ono, K.-I. Miyamoto, J Le Roux et al. // Proceedings of the EUSIPCO 2008 European Signal Processing Conference. — 2008. — August.
93. Sheh A., Ellis D. P. W. Chord segmentation and recognition using EM-trained hidden markov models. // ISMIR. — 2003.
94. T. Fujishima. Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music // *Proc. ICMC, 1999*. — 1999. — Pp. 464–467. <http://ci.nii.ac.jp/naid/10013545881/en/>.
95. Talbot-Smith M. Audio Engineer's Reference Book. — Taylor & Francis, 1999. <http://books.google.ru/books?id=LcySXZbdamEC>.
96. Temperley D. The cognition of basic musical structures. — Mit Press, 2001.
97. Theano: a CPU and GPU Math Expression Compiler / J. Bergstra, O. Breuleux, F. Bastien et al. // Proceedings of the Python for Scientific Computing Conference (SciPy). — 2010. — June. — Oral Presentation.
98. Using Hyper-genre Training to Explore Genre Information for Automatic Chord Estimation / Y. Ni, M. Mcvicar, R. Santos-Rodriguez, T. De Bie // Proceedings of the 13th International Society for Music Information Retrieval Conference. — Porto, Portugal: 2012. — October 8-12. — <http://ismir2012.ismir.net/event/papers/109-ismir-2012.pdf>.
99. Weller A., Ellis D., Jebara T. Structured Prediction Models for Chord Transcription of Music Audio // Proceedings of the 2009 International Conference on Machine Learning and Applications. — ICMLA '09. — Washington, DC, USA: IEEE Computer Society, 2009. — Pp. 590–595. <http://dx.doi.org/10.1109/ICMLA.2009.132>.

100. *Zhang X., Gerhard D.* Chord Recognition using Instrument Voicing Constraints // Proceedings of the 9th International Conference on Music Information Retrieval. — Philadelphia, USA: 2008. — September 14-18. — Pp. 33–38. — http://ismir2008.ismir.net/papers/ISMIR2008_241.pdf.
101. *Zhu Y., Kankanhalli M. S., Gao S.* Music Key Detection for Musical Audio // *Multi-Media Modeling Conference, International*. — 2005. — Vol. 0. — Pp. 30–37.

Приложение А

Результаты распознавания аккордов в используемой коллекции

Результаты отсортированы по названию композиции в алфавитном порядке.

Таблица А.1: Результаты распознавания аккордов

Композиция	Triads	Tetrads	Сегментация	Длительность (с)
01 A Kind Of Magic	0.9530	0.8122	0.9055	262.51
01 Bohemian Rhapsody	0.6700	0.3926	0.6820	347.21
01 - A Hard Day's Night	0.9563	0.7003	0.9064	148.51
01 - Come Together	0.6514	0.3628	0.7221	260.65
01 - Drive My Car	0.8242	0.2863	0.8831	134.77
01 - Help!	0.8960	0.4163	0.8908	141.09
01 - It Won't Be Long	0.7001	0.1552	0.8251	125.50
01 - I Saw Her Standing There	0.8988	0.2086	0.9310	175.80
01 - Magical Mystery Tour	0.8186	0.5323	0.8810	171.32
01 - No Reply	0.9326	0.4871	0.9237	137.85
01 - Sgt. Pepper's Lonely Hearts Club Band	0.8977	0.4675	0.8733	122.09
01 - Spiel Mir Eine Alte Melodie	0.8125	0.3936	0.8485	165.09
01 - Taxman	0.8264	0.3861	0.7555	159.01
01 - Two of Us	0.8171	0.3610	0.7561	216.69
02 Another One Bites The Dust	0.0326	0.0326	0.7176	199.50
02 - All I've Got To Do	0.9072	0.3908	0.8856	121.39
02 - Dig a Pony	0.8365	0.3557	0.8930	234.81
02 - Eleanor Rigby	0.7068	0.2434	0.8610	127.69
02 - I'm a Loser	0.8732	0.3025	0.8125	153.52
02 - I Should Have Known Better	0.9167	0.6940	0.8193	164.10
02 - Misery	0.9595	0.5765	0.9663	110.18
02 - Norwegian Wood (This Bird Has Flown)	0.8500	0.8187	0.5502	125.44
02 - Rawhide	0.6475	0.3514	0.6711	215.98
02 - Something	0.8176	0.1589	0.7629	172.33
02 - The Fool On The Hill	0.6583	0.3727	0.6822	180.19
02 - The Night Before	0.9320	0.1897	0.9091	156.71
02 - With A Little Help From My Friends	0.9241	0.5873	0.8819	164.02
03 - Across the Universe	0.8859	0.4703	0.8175	228.57
03 - All My Loving	0.9186	0.3676	0.8983	123.37
03 - Anna (Go To Him)	0.8714	0.2685	0.8985	177.58
03 - Baby's In Black	0.8371	0.2781	0.6342	127.58
03 - Flying	0.9575	0.3516	0.8811	117.86
03 - I'm Only Sleeping	0.8940	0.4747	0.9206	181.68
03 - If I Fell	0.9067	0.6513	0.8172	137.61
03 - Lucy In The Sky With Diamonds	0.7228	0.5408	0.8041	197.74
03 - Maxwell's Silver Hammer	0.9005	0.3916	0.5869	204.99
03 - She	0.8016	0.6853	0.8949	130.69
03 - You've Got To Hide Your Love Away	0.8815	0.8052	0.8500	127.61
03 - You Won't See Me	0.8451	0.5215	0.6982	190.10
04 Fat Bottomed Girls	0.8243	0.7417	0.7596	200.94
04 I Want It All	0.7206	0.5289	0.8104	212.88
04 - Blue Jay Way	0.8832	0.0987	0.5102	200.13
04 - Chains	0.9294	0.1679	0.8713	146.49
04 - Don't Bother Me	0.8543	0.2920	0.8943	149.39
04 - Erbauliche Gedanken Eines Tobackrauchers	0.5219	0.3833	0.7900	342.73
04 - Getting Better	0.5971	0.3622	0.6690	158.87

04 - I'm Happy Just To Dance With You	0.8105	0.1412	0.8414	114.45
04 - I Me Mine	0.6172	0.3298	0.8556	136.68
04 - I Need You	0.9004	0.3614	0.6931	126.83
04 - Love You To	0.7202	0.1595	0.9011	31.68
04 - Nowhere Man	0.7696	0.1156	0.8693	164.31
04 - Oh! Darling	0.9014	0.5642	0.8832	199.94
04 - Rock and Roll Music	0.9714	0.0792	0.9534	153.76
05 Bicycle Race	0.5518	0.3861	0.8560	172.02
05 I Want To Break Free	0.8987	0.7764	0.8149	257.64
05 - Andersrum	0.8100	0.5021	0.8016	110.25
05 - And I Love Her	0.8197	0.5571	0.9290	151.09
05 - Another Girl	0.8177	0.5364	0.8399	128.03
05 - Boys	0.8320	0.2787	0.8345	147.51
05 - Dig It	0.7584	0.1848	0.8006	50.02
05 - Fixing A Hole	0.7437	0.3891	0.3915	142.61
05 - Here There And Everywhere	0.7485	0.3622	0.8615	145.01
05 - I'll Follow the Sun	0.9292	0.6674	0.8659	111.05
05 - Little Child	0.9170	0.2260	0.8550	108.02
05 - Octopus's Garden	0.8765	0.5540	0.9336	171.18
05 - Think For Yourself	0.8792	0.3131	0.9591	139.34
05 - Your Mother Should Know	0.8911	0.5308	0.9019	145.61
06 You're My Best Friend	0.6416	0.3610	0.7752	167.86
06 - Ask Me Why	0.8197	0.6127	0.8269	140.58
06 - I Am The Walrus	0.8389	0.3404	0.8894	277.08
06 - I Want You	0.7653	0.4025	0.8199	389.51
06 - Let It Be	0.8233	0.6056	0.7755	243.33
06 - Mr. Moonlight	0.8788	0.6604	0.5785	125.84
06 - She's Leaving Home	0.7285	0.3655	0.7938	215.07
06 - Tell Me Why	0.7193	0.1814	0.7341	129.51
06 - The Word	0.8017	0.3886	0.8227	156.21
06 - Tigerfest	0.7430	0.3935	0.7520	194.02
06 - Till There Was You	0.8299	0.2316	0.8106	124.91
06 - Yellow Submarine	0.8320	0.5255	0.6482	160.42
06 - You're Going to Lose That Girl	0.9419	0.2551	0.9134	139.75
07 Don't Stop Me Now	0.7553	0.3628	0.8360	208.66
07 - Akne	0.7635	0.4299	0.7990	195.03
07 - Being For The Benefit Of Mr. Kite!	0.6993	0.3699	0.7431	135.32
07 - Can't Buy Me Love	0.9076	0.1648	0.8994	134.95
07 - Hello Goodbye	0.8041	0.3622	0.7732	211.49
07 - Here Comes The Sun	0.8635	0.5376	0.8211	181.09
07 - Kansas City- Hey Hey Hey Hey	0.8769	0.0666	0.9081	153.47
07 - Maggie Mae	0.8084	0.2605	0.8425	40.62
07 - Michelle	0.7954	0.5462	0.8876	138.75
07 - Please Mister Postman	0.8911	0.7488	0.9371	156.73
07 - Please Please Me	0.8709	0.0872	0.8445	123.38
07 - She Said She Said	0.8062	0.2465	0.7577	157.15
07 - Ticket To Ride	0.8217	0.7148	0.8951	192.50
08 Save Me	0.8341	0.6311	0.8900	228.60
08 - Act Naturally	0.9400	0.7925	0.8966	152.18
08 - Any Time At All	0.8698	0.6465	0.8766	133.41
08 - Because	0.8986	0.9211	0.8809	130.83
08 - Blass	0.8054	0.6921	0.8572	170.11
08 - Eight Days a Week	0.8943	0.2065	0.8376	165.43
08 - Good Day Sunshine	0.9439	0.6338	0.9281	129.78
08 - I've Got A Feeling	0.8113	0.4867	0.6874	206.01
08 - Love Me Do	0.8925	0.4118	0.8105	142.76
08 - Roll Over Beethoven	0.7917	0.3928	0.7911	167.63
08 - Strawberry Fields Forever	0.7987	0.2505	0.7885	225.87
08 - What Goes On	0.9061	0.1285	0.8085	167.12
08 - Within You Without You	0.6229	0.0214	0.3035	304.15
09 Crazy Little Thing Called Love	0.6642	0.2613	0.7929	163.83
09 Who Wants To Live Forever	0.7988	0.5258	0.7994	262.98
09 - And Your Bird Can Sing	0.9650	0.4479	0.8108	114.62
09 - Girl	0.8441	0.7843	0.7832	153.89
09 - Hold Me Tight	0.8498	0.0865	0.8235	152.58
09 - I'll Cry Instead	0.9123	0.3617	0.8655	107.76
09 - It's Only Love	0.8945	0.5877	0.8607	110.35
09 - Mr Morgan	0.7474	0.5859	0.6360	183.33
09 - One After 909	0.8730	0.0379	0.8116	171.10
09 - P. S. I Love You	0.9291	0.1234	0.8478	125.75
09 - Penny Lane	0.8172	0.5542	0.6994	158.26
09 - When I'm Sixty-Four	0.8962	0.1477	0.8199	155.17

09 - Words of Love	0.8589	0.4025	0.8742	134.74
09 - You Never Give Me Your Money	0.8240	0.4178	0.8573	236.09
10 Somebody To Love	0.6713	0.2971	0.7789	295.09
10 - Baby It's You	0.9311	0.6720	0.9318	158.07
10 - Baby You're A Rich Man	0.7890	0.1038	0.7052	183.77
10 - For No One	0.7965	0.5700	0.8413	119.15
10 - Honey Don't	0.8591	0.8093	0.7699	179.62
10 - I'm Looking Through You	0.7237	0.6017	0.8730	145.08
10 - Liebesleid	0.6712	0.4633	0.7949	252.36
10 - Lovely Rita	0.0075	0.0075	0.8526	132.47
10 - Sun King	0.6737	0.5353	0.8518	140.09
10 - The Long and Winding Road	0.8069	0.3550	0.8533	217.89
10 - Things We Said Today	0.7960	0.5388	0.4841	158.85
10 - You Like Me Too Much	0.9162	0.3274	0.8518	158.82
10 - You Really Got A Hold On Me	0.8855	0.5033	0.8496	182.14
11 - All You Need Is Love	0.7183	0.4349	0.8105	228.44
11 - Doctor Robert	0.9200	0.3583	0.8325	135.21
11 - Do You Want To Know A Secret	0.7661	0.4545	0.9047	119.35
11 - Every Little Thing	0.8423	0.1593	0.6801	124.55
11 - For You Blue	0.8852	0.1748	0.8779	151.91
11 - Good Morning Good Morning	0.7094	0.1518	0.7884	159.50
11 - Ich Kann Heute Nicht	0.7609	0.5703	0.8575	129.44
11 - In My Life	0.8815	0.6652	0.8304	147.98
11 - I Wanna Be Your Man	0.8367	0.0482	0.7944	118.99
11 - Mean Mr Mustard	0.8872	0.4497	0.9015	66.45
11 - Tell Me What You See	0.8300	0.3764	0.7532	159.71
11 - When I Get Home	0.8448	0.2850	0.8191	138.37
12 Good Old-Fashioned Lover Boy	0.6458	0.3826	0.6862	175.83
12 - A Taste Of Honey	0.7996	0.5405	0.7705	125.15
12 - Devil In Her Heart	0.8237	0.1091	0.7333	147.67
12 - Get Back	0.6515	0.4705	0.7596	182.15
12 - I've Just Seen a Face	0.8338	0.5374	0.7777	127.19
12 - I Don't Want to Spoil the Party	0.9097	0.1623	0.8272	154.96
12 - I Want To Tell You	0.8299	0.7494	0.7469	95.41
12 - Jakob Und Marie	0.6543	0.4688	0.7874	186.31
12 - Polythene Pam	0.8148	0.1666	0.8575	72.96
12 - Sgt. Pepper's Lonely Hearts Club Band (Reprise)	0.6993	0.2091	0.8567	78.28
12 - Wait	0.7305	0.6441	0.6749	128.57
12 - You Can't Do That	0.8582	0.5195	0.8516	157.65
13 Play The Game	0.7340	0.5082	0.8224	212.14
13 - A Day In The Life	0.6376	0.4894	0.8068	306.73
13 - Got To Get You Into My Life	0.7649	0.4982	0.6374	141.78
13 - I'll Be Back	0.8282	0.5633	0.8089	139.83
13 - If I Needed Someone	0.9365	0.6165	0.8553	143.86
13 - Not A Second Time	0.5931	0.2179	0.8435	126.48
13 - Paparazzi	0.8051	0.5138	0.9001	212.16
13 - She Came In Through The Bathroom Window	0.7552	0.5713	0.8516	117.73
13 - There's A Place	0.8903	0.4419	0.9273	112.88
13 - What You're Doing	0.8964	0.7863	0.9150	149.74
13 - Yesterday	0.7303	0.5884	0.7540	117.56
14 Hammer To Fall	0.5978	0.4834	0.2510	220.51
14 - Dizzy Miss Lizzy	0.9627	0.0518	0.9649	174.27
14 - Everybody's Trying to Be My Baby	0.8942	0.5514	0.7791	143.87
14 - Golden Slumbers	0.9446	0.7002	0.8643	91.59
14 - Money	0.4998	0.2709	0.6413	167.56
14 - Run For Your Life	0.9212	0.8013	0.8526	138.80
14 - Santa Donna Lucia Mobile	0.7831	0.4434	0.8251	100.83
14 - Tomorrow Never Knows	0.8552	0.0103	0.6357	177.40
14 - Twist And Shout	0.9074	0.0942	0.8810	153.27
15 Friends Will Be Friends	0.7836	0.6349	0.9432	234.36
15 Seven Seas Of Rhye	0.7925	0.5525	0.7084	170.47
15 - Carry That Weight	0.9010	0.5767	0.7549	92.71
15 - Es Wird Alles Wieder Gut Herr Professor	0.7386	0.3258	0.7840	152.32
16 We Will Rock You	0.2127	0.1395	0.3279	121.38
16 - The End	0.6804	0.4596	0.8478	136.05
16 - Zu Leise Fuer Mich	0.4697	0.2213	0.6784	229.00
17 We Are The Champions	0.6277	0.3967	0.7562	150.88
17 - Duell	0.6811	0.1264	0.6859	233.04
17 - Her Majesty	0.6131	0.3551	0.6579	20.33
18 - Zuhause	0.6690	0.3084	0.7920	202.48
CD1 - 01 - Back in the USSR	0.8519	0.2452	0.8932	163.32
CD1 - 02 - Dear Prudence	0.6112	0.3508	0.4760	234.78

CD1 - 03 - Glass Onion	0.7791	0.6871	0.7705	115.04
CD1 - 04 - Ob-La-Di Ob-La-Da	0.9442	0.4060	0.8281	184.13
CD1 - 05 - Wild Honey Pie	0.6613	0.7512	0.8424	52.92
CD1 - 06 - The Continuing Story of Bungalow Bill	0.7474	0.2944	0.6990	194.32
CD1 - 07 - While My Guitar Gently Weeps	0.7505	0.5732	0.8984	285.31
CD1 - 08 - Happiness is a Warm Gun	0.8809	0.7011	0.8756	162.38
CD1 - 09 - Martha My Dear	0.8040	0.3999	0.8465	148.74
CD1 - 10 - I'm So Tired	0.9088	0.6502	0.9096	118.27
CD1 - 11 - Black Bird	0.7674	0.7114	0.6992	111.96
CD1 - 12 - Piggies	0.9404	0.8305	0.9217	116.42
CD1 - 13 - Rocky Raccoon	0.8611	0.2435	0.8875	194.81
CD1 - 14 - Don't Pass Me By	0.9017	0.8789	0.8383	229.68
CD1 - 15 - Why Don't We Do It In The Road	0.8614	0.4970	0.8394	101.46
CD1 - 16 - I Will	0.9163	0.3991	0.9183	105.74
CD1 - 17 - Julia	0.8000	0.6858	0.8557	174.20
CD2 - 01 - Birthday	0.6555	0.1219	0.7401	161.15
CD2 - 02 - Yer Blues	0.6725	0.0927	0.6991	241.11
CD2 - 03 - Mother Nature's Son	0.7769	0.6276	0.5988	153.56
CD2 - 04 - Everybody's Got Something To Hide Except Me and My Monkey	0.6839	0.2701	0.5975	142.23
CD2 - 05 - Sexy Sadie	0.9113	0.5310	0.9322	195.34
CD2 - 06 - Helter Skelter	0.6810	0.0752	0.6941	257.50
CD2 - 07 - Long Long Long	0.9036	0.7988	0.7441	158.59
CD2 - 08 - Revolution	0.9041	0.4096	0.8564	254.55
CD2 - 09 - Honey Pie	0.8988	0.2443	0.8880	159.67
CD2 - 10 - Savoy Truffle	0.8341	0.2494	0.7553	167.00
CD2 - 11 - Cry Baby Cry	0.8787	0.4348	0.8833	156.71
CD2 - 12 - Revolution 9	0.1499	0.0641	0.4319	462.56
CD2 - 13 - Good Night	0.7106	0.3834	0.8371	189.63
N001-M01-T01	0.8031	0.2292	0.8753	197.81
N002-M01-T02	0.5913	0.3388	0.8240	209.70
N003-M01-T03	0.6189	0.3017	0.8163	192.29
N004-M01-T04	0.5967	0.5084	0.7454	192.14
N005-M01-T05	0.8588	0.1362	0.8689	205.79
N006-M01-T06	0.6174	0.5552	0.8000	206.57
N007-M01-T07	0.8603	0.5054	0.9395	284.96
N008-M01-T08	0.8048	0.5410	0.8106	179.00
N009-M01-T09	0.9092	0.6019	0.8258	265.88
N010-M01-T10	0.8341	0.4552	0.8828	211.87
N011-M01-T11	0.6336	0.3486	0.7518	263.97
N012-M01-T12	0.7960	0.3620	0.8598	193.82
N013-M01-T13	0.8633	0.6134	0.8926	215.57
N014-M01-T14	0.8572	0.7123	0.9161	234.95
N015-M01-T15	0.8253	0.2060	0.9253	162.89
N016-M01-T16	0.6670	0.4660	0.8043	262.15
N017-M02-T01	0.7892	0.3672	0.8765	241.44
N018-M02-T02	0.6888	0.4053	0.7170	249.33
N019-M02-T03	0.7457	0.5517	0.8921	287.23
N020-M02-T04	0.7481	0.6001	0.8815	226.58
N021-M02-T05	0.8268	0.4122	0.9225	257.99
N022-M02-T06	0.8215	0.6053	0.9162	209.52
N023-M02-T07	0.8636	0.4364	0.8914	201.19
N024-M02-T08	0.4451	0.2767	0.6807	207.74
N025-M02-T09	0.1988	0.1963	0.7620	231.44
N026-M02-T10	0.8178	0.2121	0.9129	199.76
N027-M02-T11	0.5467	0.4740	0.8046	291.33
N028-M02-T12	0.4329	0.5181	0.6507	217.66
N029-M02-T13	0.8702	0.5755	0.9223	206.95
N030-M02-T14	0.4567	0.3405	0.7389	151.41
N031-M02-T15	0.3591	0.0040	0.7953	250.21
N032-M02-T16	0.7712	0.2854	0.9330	252.08
N033-M03-T01	0.4682	0.3930	0.5109	287.01
N034-M03-T02	0.8863	0.4708	0.9308	207.64
N035-M03-T03	0.8651	0.2518	0.8830	174.98
N036-M03-T04	0.7680	0.4593	0.7757	316.25
N037-M03-T05	0.6512	0.1155	0.9005	234.18
N038-M03-T06	0.6607	0.4666	0.8521	273.53
N039-M03-T07	0.8080	0.8063	0.9078	283.93
N040-M03-T08	0.7164	0.4469	0.5744	224.29
N041-M03-T09	0.9778	0.2072	0.9240	57.55
N042-M03-T10	0.5990	0.2128	0.6781	246.84
N043-M03-T11	0.8111	0.3084	0.8881	202.59

N044-M03-T12	0.8340	0.2569	0.8416	240.25
N045-M03-T13	0.8286	0.5190	0.9438	217.07
N046-M03-T14	0.8652	0.6247	0.8876	194.28
N047-M03-T15	0.9280	0.7311	0.9538	208.10
N048-M03-T16	0.8033	0.3976	0.8611	265.28
N049-M04-T01	0.6705	0.2112	0.8014	251.06
N050-M04-T02	0.6930	0.1875	0.8008	193.24
N051-M04-T03	0.9025	0.7731	0.9411	353.23
N052-M04-T04	0.4092	0.3869	0.7762	215.00
N053-M04-T05	0.7841	0.4576	0.6595	219.77
N054-M04-T06	0.8463	0.5317	0.9161	222.92
N055-M04-T07	0.9040	0.7362	0.9510	234.97
N056-M04-T08	0.9438	0.7476	0.9452	322.96
N057-M04-T09	0.8313	0.4105	0.8227	252.57
N058-M04-T10	0.6856	0.4590	0.8703	179.39
N059-M04-T11	0.8500	0.5705	0.8710	205.01
N060-M04-T12	0.8691	0.6495	0.8809	230.53
N061-M04-T13	0.5998	0.3820	0.8653	283.16
N062-M04-T14	0.9287	0.5145	0.9546	187.71
N063-M04-T15	0.8754	0.4558	0.8580	246.97
N064-M04-T16	0.8517	0.5471	0.8963	270.41
N065-M05-T01	0.8336	0.5394	0.8546	227.37
N066-M05-T02	0.6762	0.3124	0.8667	272.31
N067-M05-T03	0.8513	0.5807	0.9240	241.18
N068-M05-T04	0.8375	0.5426	0.8317	288.99
N069-M05-T05	0.8814	0.5948	0.9249	315.37
N070-M05-T06	0.7584	0.6160	0.8015	242.41
N071-M05-T07	0.7506	0.4109	0.8947	259.17
N072-M05-T08	0.7168	0.4987	0.7275	181.96
N073-M05-T09	0.5759	0.3540	0.8270	129.33
N074-M05-T10	0.8820	0.5523	0.9010	187.94
N075-M05-T11	0.7460	0.5791	0.7580	188.36
N076-M05-T12	0.8601	0.4254	0.8283	175.61
N077-M05-T13	0.7182	0.5748	0.8540	205.94
N078-M05-T14	0.8242	0.6901	0.8839	243.87
N079-M05-T15	0.9091	0.7280	0.8301	228.52
N080-M05-T16	0.5673	0.5613	0.8931	198.50
N081-M06-T01	0.8216	0.4574	0.8581	226.80
N082-M06-T02	0.8440	0.6096	0.9117	328.60
N083-M06-T03	0.6843	0.3191	0.7327	193.87
N084-M06-T04	0.8006	0.4808	0.8819	203.08
N085-M06-T05	0.9332	0.1342	0.9386	200.67
N086-M06-T06	0.9024	0.6712	0.9070	248.87
N087-M06-T07	0.7807	0.5749	0.8079	244.95
N088-M06-T08	0.7026	0.5793	0.8425	237.26
N089-M06-T09	0.7899	0.0660	0.8508	218.65
N090-M06-T10	0.8262	0.5024	0.8856	272.02
N091-M07-T01	0.8964	0.4237	0.8610	221.97
N092-M07-T02	0.5730	0.3588	0.7921	178.73
N093-M07-T03	0.8435	0.7320	0.8782	248.07
N094-M07-T04	0.6503	0.4165	0.9147	223.94
N095-M07-T05	0.5842	0.0531	0.6868	232.83
N096-M07-T06	0.7474	0.2725	0.8783	265.20
N097-M07-T07	0.6147	0.1064	0.6461	243.15
N098-M07-T08	0.7177	0.7103	0.7559	211.93
N099-M07-T09	0.5211	0.4074	0.7518	287.43
N100-M07-T10	0.9243	0.7700	0.9276	280.51