НАЗВАНИЕ УЧРЕЖДЕНИЯ, В КОТОРОМ ВЫПОЛНЯЛАСЬ
ДАННАЯ ДИССЕРТАЦИОННАЯ РАБОТА

На правах рукописи
УДК 004.93

ГЛАЗЫРИН НИКОЛАЙ ЮРЬЕВИЧ

НАЗВАНИЕ ДИССЕРТАЦИОННОЙ РАБОТЫ

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 – 2013

Содержание

Введение

Музыка является неотъемлемой составляющей жизни современного человека. Она проявляет себя в разных формах: от детских колыбельных и напевания под нос до радио и сигналов вызова сотовых телефонов. Люди таких профессий как музыкант, музыковед, музыкальный критик, диджей большую часть своей жизни уделяют музыке. Те, кто не занимается музыкой профессионально, зачастую имеют несколько любимых исполнителей и слушают музыку время от времени.

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 данной 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.

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. 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.

Актуальность темы. Первые попытки обработки музыкальной информации в символьном виде были сделаны в 1950-х годах с появлением первых компьютеров. Они были связаны с автоматическим определением закономерностей в музыке и использования их для создания новых мелодий (см. [1]). Тогда же предлагается использовать компьютер для распознавания и печати нотных записей, анализа схожести различных композиций и поиска по образцу. В 1960-х годах появляются первые работы (например, [2]), 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 [3] было положено начало новому применению компьютера к анализу цифровых музыкальных звукозаписей: распознаванию в ней 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 [3] рассматривались только звукозаписи, содержащие не более двух одновременно звучащих музыкальных инструментов. В 1996 году в [4] был представлен один из первых методов, подходивших для любой полифонической цифровой аудиозаписи.

Задача определения последовательности аккордов при этом не отделялась от задачи транскрибирования. Как отмечает Т. Фуджишима в [5], в 1980-1990-х годах (например, в работе [6]) проблема распознавания аккордов в музыке решалась путём распознавания отдельных нот и их объединения в аккорды. Он же впервые предложил метод распознавания аккордов без предварительного транскрибирования звукозаписи. В [6] 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 [1] проводятся соревнования среди алгоритмов распознавания аккордов в звуке. За это время был достигнут существенный прогресс в качестве распознавания. В 2012 году на это соревнование были выставлены более 10 алгоритмов.

В 2010-х годах появляются широко доступные программные продукты, включающие в себя такие алгоритмы. Популярном пакет для профессионального создания музыки *Ableton Live 9*[2] предоставляет возможность альтернативного ввода музыкальных данных, позволяя записать аккорды или мелодию на гитаре или другом музыкальном инструменте, после чего преобразовать эту запись в нотное представление в редакторе. В данном случае задача транскрибирования облегчается тем, что на входе предполагается запись одного инструмента.

Приложения для смартфонов *AnySong Chord Recognition* [3] и *Chord Detector* [4] 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 гитарной музыкой.

Интернет-сайт `http://chordify.net` 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. На каждом фрагменте определяется набор признаков, по которому определяется соответствующий аккорд. Итоговое качество распознавания зависит как от выбора признаков, так и от алгоритма, сопоставляющего набору признаков аккорд.

[1] `http://www.music-ir.org/mirex/wiki/MIREX_HOME`
[2] См. `https://www.ableton.com/en/live/`.
[3] `https://play.google.com/store/apps/details?id=com.musprojects.chord`
[4] `http://www.chord-detector.com/wordpress/apps/chorddetector/`

Признаки позволяют представить в компактном виде основную информацию о звуке на данном фрагменте. Было предложено большое количество разнообразных алгоритмов получения звуковых признаков, использующих особенности звучания музыкальных инструментов, особенности человеческого восприятия и возможные помехи на звукозаписях.

В 2010-х годах становятся чрезвычайно популярными так называемые методы обучения представлениям. Они, фактически, являются алгоритмами со множеством автоматически подбираемых параметров, позволяющими получить хорошо работающие признаки. За исключением Хамфри [7] 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 метрики в пространстве признаков.

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 на 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 [8] выделяет следующие:

- Потребность в большом количестве данных для обучения. Подготовка таких данных весьма трудоёмка, а сами данные могут сильно разниться 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 объёма обучающей выборки.

- Недостаточное использование времени. Марковское свойство предполагает зависимость только от предыдущего фрагмента. Но музыкальная композиция зачастую имеет определённую, 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 методы хорошо приспособлены для моделирования смены состояния (звучащего аккорда) и хуже – для моделирования продолжительности нахождения в одном состоянии.

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 на таком фрагменте влечёт за собой ошибки на соседних фрагментах.

Целью работы является разработка метода для распознавания последовательности аккордов, не требующего большого количества данных для обучения. Метод должен показывать сравнимое с существующими качество распознавания аккордов.

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 звуке компонент, соответствующих музыкальным инструментам, с целью улучшения существующих алгоритмов вычисления признаков по фрагменту звукозаписи;

2. исследовать применимость некоторых методов обучения представлениям к получению признаков;

3. улучшить алгоритм определения аккорда по вектору признаков, использующий 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, поучаствовав в соревновании MIREX Audio Chord Estimation.

Все поставленные задачи были решены в рамках данной работы.

Методы исследования. При решении поставленных задач в работе использованы методы математического моделирования, спектральный анализ, алгоритмы машинного обучения, методы объектно-ориентированного программирования и многопоточного программирования.

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

1. Метод распознавания последовательности аккордов в звукозаписи, не использующий алгоритмов машинного обучения.

2. Метод представления звукозаписи в виде последовательности векторов признаков с применением многослойных очищающих автоассоциаторов.

3. Результаты работы методов на коллекции из 319 звукозаписей, подтверждающие его эффективность.

4. Реализация методов в виде комплекса программ на языках Java и Python.

Достоверность и обоснованность выносимых на защиту результатов обеспечивается условиями открытого конкурса, проводимого международной лабораторией оценки систем музыкального информационного поиска (International Music Information Retrieval Systems Evaluation Laboratory) университета Иллинойса, США. Результаты конкурса опубликованы в открытом доступе по адресам http://nema.lis.illinois.edu/nema_out/mirex2012/results/ace/mrx/, http://nema.lis.illinois.edu/nema_out/mirex2012/results/ace/mcg/.

Практическая ценность 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.

Апробация работы. 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).

Публикации. Основные результаты по теме диссертации 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 соревнование среди алгоритмов распознавания аккордов MIREX Audio Chord Estimation 2012 [5].

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.

Личный вклад соискателя. 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.

Объём и структура работы

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 описывается 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 компонеты спектра звука, соответствующие звучанию музыкальных инструментов с определённой высотой звучания. Это помогает получить векторы признаков, в большей степени сохраняющие необходимую для определения аккорда информацию.

[5] http://nema.lis.illinois.edu/nema_out/mirex2012/results/ace/mrx/, http://nema.lis.illinois.edu/nema_out/mirex2012/results/ace/mcg/

Коррекция вектора признаков с использованием наиболее схожих с ним 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 различных вариантов многослойных автоассоциаторов. Рассматриваются также рекуррентные многослойные автоассоциаторы, позволяющие моделировать зависимость вектора признаков на текущем фрагменте от вектора признаков на предыдущем фрагменте звукозаписи. Автору не удалось добиться повышения качества распознавания аккордов с использованием признаков, полученных с при помощи автоассоциаторов, в сравнении с признаками, алгоритмы вычисления которых придуманы и настроены человеком.

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.

Глава 1

Необходимые теоретические сведения

В этой главе даются теоретические сведения, необходимые для формальной постановки задачи и описания достигнутых результатов, и обзор существующих подходов к решению задачи. В разделе 1.1 даются базовые понятия звука и спектра. В разделе 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 разделе 1.4 представлены основы представления звука в цифровом виде. В разделе 1.5 указаны характерные черты музыкальных звукозаписей, которые могут быть использованы для решения поставленной задачи.

1.1 Звук

Большая Советская Энциклопедия [9] 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* $a(f)$. Это неотрицательная функция, которая показывает зависимость между частотой колебаний и интенсивностью этой частоты в данном звуковом сигнале $x(t)$. Также выделяют спектр мощности и фазовый спектр сигнала. В дальнейшем, если не оговорено иное, под спектром будет пониматься частотный спектр звукового сигнала.

Для любых существующих в природе звуковых сигналов функция $x(t)$ отлична от 0 только на некотором промежутке $[t_{start}, t_{end}]$. Поэтому любой реальный сигнал можно периодически продолжить на всю вещественную ось с периодом $\tau = t_{end} - t_{start}$. Более того, продолженная таким образом функция $x(t)$ 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 $a_{f_0} \cos(2\pi f_0 t - \phi_{f_0})$.

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.

1.2 Свойства звука

И. В. Способин в [10] выделяет 4 основных свойства звука с точки зрениия теории музыки: 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 мелов и шкала барков. Более подробно эти модели описаны, например, в [11], с. 79-81. Высота звука может быть выражена с разной степенью ясности. Высота звуков, имеющих основной тон, определяется его частотой. Для остальных звуков (например, разного рода шумы, шорохи, в том числе звуки шумящих музыкальных инструментов) высота может быть неясной.

Длительность звука соответствует длительности колебаний источника звука. Она приобретает особое значение в контексте музыкального произведения, когда последовательность звуков и их продолжительность задают ритм.

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 [12].

Тембр или окраска звука определяется частотами и интенсивностью его обертонов, которые, в свою очередь, определяются физическими свойствами музыкального инструмента. 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 [10].

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).

Ступенью называется звук музыкальной системы. Основные ступени соответствуют звукам, извлекаемым белыми клавишами фортепиано. 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 $2f_0$ как очень похожие и тесно связанные друг с другом. Расстояние между такими звуками называется *октавой*. Как отмечает Д. Левитин в [13], «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.

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 клавиш на фортепиано. Частоту каждой ступени звукоряда можно вычислить по формуле

$$f_j = f_0 \cdot 2^{j/12}, \tag{1.1}$$

где f_0 – частота настройки музыкальных инструментов. Обычно выбирают $f_0 = 440$ Гц и фиксируют эту частоту 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, задающий формат обмена данными между электронными музыкальными инструментами, определяет 128 возможных значений для частоты звука[1]. 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

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

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* (♯) и *бемоль* (♭), обозначающие соответственно повышение и понижение на один полутон.

Интервалом называется расстояние по высоте между двумя звуками, взятыми последовательно или одновременно. В октаве заключено 8 ступеней. Соответственно, имеется 8 основных названий интервалов, отражающих их величину в ступенях. 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. *Обращением* интервала называется перемещение его нижнего звука на октаву вверх или верхнего звука на октаву вниз.

Созвучием называется одновременное сочетание двух и более звуков. *Аккордом* называется созвучие, состоящее не менее, чем из трёх звуков. *Гармонией* называется объединение звуков в

[1] См. http://www.midi.org/techspecs/midituning.php/.

созвучия и последовательность созвучий. Гармония формирует контекст, сопровождает мелодию, а также может объединять несколько одновременно звучащих мелодий. В свою очередь, контекст формирует у слушателя ожидание последующих событий в музыке. Композитор может как оправдывать, так и нарушать эти ожидания для большей выразительности.

Трезвучием называется аккорд, который состоит из трёх звуков, располагающихся по терциям. 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*. *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.

Септаккордом называется четырехзвучие, располагающееся по терциям. Септаккорд может быть получен из трезвучия путём добавления к нему одной терции сверху. Наиболее употребительны доминантсептаккорд (большая, малая, малая терции), уменьшенный септаккорд (малая, малая, малая терции), малый септаккорд (малая, малая, большая терции) и минорный септаккорд (малая, большая, малая терции).

Ритмом называется организованная последовательность длительностей 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. *Ритмическим рисунком* называется последовательность звуковых длительностей, взятая отдельно от высотных соотношений звуков.

Акцентом называется выделение звука посредством большей громкости (часто также длительности) по сравнению с окружающими звуками. *Метром* называется непрерывно повторяющаяся последовательность акцентируемых и неакцентируемых равнодлительных ритмических единиц (отрезков времени). Акцентируемые и неакцентируемые равнодлительные ритмические единицы времени, образующие метр, называются *метрическими долями*. Акцентируемая доля называется *тяжёлой* или *сильной*, неакцентируемая – *легкой* 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.

Для музыкальной выразительности 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. *Тяготением* называется притяжение неустойчивого звука системы к устойчивому, отстоящему от него на секунду. *Ладом* называется система звуковысотных связей, объединённая тоникой. Многие лады состоят из 7 звуков, но существуют лады с большим и меньшим их числом. *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.

- 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 из компонент 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. Сильные метрические доли обычно акцентируются ударными инструментами и началом звучания нот. Слабые метрические доли часто выделяются ударными инструментами. 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 повторений. Как пишет Д. Левитин в [13], «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

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.

Разобьем задачу на отдельные этапы.

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 получению её спектрограммы $C_{N \times M}$. Основным инструментом для такого перехода является дискретное оконное преобразование Фурье.

Спектрограмма представляет из себя матрицу, каждая из N строк которой соответствует определённой частоте, а каждый из M столбцов – промежутку времени. Элементами её являются значения интенсивности данной частоты на данном промежутке времени. Фактически, каждый столбец представляет из себя спектр короткого фрагмента исходного сигнала. Удобно представлять спектрограмму в виде последовательности столбцов $C_{N \times M} = \{C_m\}_{m=0}^{M-1}$.

Здесь возникает 2 подзадачи: разбиение звукозаписи на фрагменты и вычисление спектра на каждом из них. Важно отметить, что для современной западной музыки характерны использование равномерно темперированного строя и наличие ритма. Поэтому большее количество звуковой энергии должно быть сосредоточено в точках, соответствующих частотам нот и моментам начала метрических долей. Исходя из этих соображений, можно скорректировать выбор моментов начала фрагментов и выбор используемого преобразования. Удобно использовать одно и то же преобразование на всех фрагментах, поскольку в этом случае все столбцы спектрограммы будут иметь одинаковый размер и смысл. После этого на каждом фрагменте необходимо решить задачу классификации.

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 на данном фрагменте делать предположения относительно звучащего 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 важными вопросами являются выбор метода классификации и выбор целевой функции (если метод предполагает нахождение наилучших параметров путём обучения). Основной подзадачей является отыскание такого набора преобразований множества X, который позволит уменьшить количество ошибок классификации 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.

В главе 2 описываются основные походы, применяемые для решения каждой из отмеченных выше подзадач.

Глава 2

Обзор литературы

Данная глава посвящена обзору литературы в области распознавания аккордов. Практически все затронутые работы вышли не позднее 15 лет с момента написания данной диссертации, наиболее значимые результаты были получены в течение последних 5 лет.

Раздел 2.1 описывает существующие подходы к разбиению звукозаписи на фрагменты, а также некоторые другие преобразования, совершаемые над звукозаписью перед её дальнейшей обработкой. Раздел 2.2 посвящён существующим подходам к получению спектрограммы. В разделе 2.3 рассмотрены существующие методы преобразования спектрограммы, позволяющие более точно классифицировать аккорд. Наконец, раздел 2.4 описывает часто применяемые методы классификации.

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 перкуссионные 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 ([14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [7]) для ускорения обработки файла. Как правило, частота дискретизации понижается с часто используемой 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 $x_Q(t_i)$. Вместе с тем, такое преобразование не является общепринятым. В [28], [29], [30], [31], [8] частота дискретизации звукозаписей не меняется. Общепринятым является преобразование стерофонических звукозаписей в монофонические путём взятия среднего арифметического от сигналов левого и правого каналов.

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 ([32], [33], [34], [35], [27]), так и для усреднения столбцов спектрограммы, вычисленной с фиксированным шагом по времени, в пределах одной метрической доли ([15], [22], [25], [8], [36]. В [36] исследовались оба этих варианта, а также дополнительно медианная фильтрация по всем столбцам спектрограммы в пределах

одной метрической доли. Наилучший результат был получен в случае усреднения столбцов спектрограммы в пределах метрической доли. В остальных случаях авторы произволно выбирали один из вариантов использования этой информации.

Отслеживание ритма (beat detection) является одной из популярных задач музыкального информационного поиска. Новые алгоритмы появляются каждый год, но далеко не все из них применются при распознавании аккордов. Практически все алгоритмы распознавания аккордов используют один из алгоритмов, представленных в [37], [38], [39]. Соответствующие программные модули для этих алгоритмов доступны бесплатно и удобны в подключении, что, по-видимому, является основной причиной их использования.

Отклонение частоты настройки музыкальных инструментов от стандартного значения 440 Гц может как явно определяться на данном этапе ([40], [19], [21], [26], [27], [41], так и неявно учитываться в процессе обработки спектрограммы ([15], [16], [24], [25], [30]). Основные алгоритмы для определения частоты настройки представлены в работах [42], [43], [40], [44], [45].

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 [24] и [27] перкуссионные компоненты удаляются из сигнала до построения спектрограммы при помощи свободно доступной для научного использования реализации алгоритма [46].

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.

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

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 разделении 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 [47], 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 создает неудобства при дальнейшей обработке, поскольку невозможно точно определить количество звуковой энергии, приходящейся на частоты, соответствующие ступеням звукоряда. Дискретное оконное преобразование Фурье используется в [14], [40], [17], [19], [21], [34], [25], [26], [8].

2. Преобразование постоянного качества (constant Q преобразование).

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

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 [48]. Преобразование постоянного качества используется в [15], [16], [20], [22], [23], [24], [29], [31], [27].

3. Гребёнка фильтров (filter bank). В цифровой обработке сигналов любое преобразование сигнала называют фильтром. Известно (см. [49], 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. Также можно выбирать полосы пропускания фильтров в соответствии с частотами ступеней звукоряда. Недостатком данного метода является большая вычислительная сложность в сравнении с алгоритмом быстрого преобразования Фурье. Гребёнки фильтров используются в [41], [7].

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 $C_m[n+N_0], C_m[n+2N_0], C_m[n+3N_0], \ldots$ 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 (ПРОВЕРИТЬ):

$$C_m[n] = \sum_{k:p[k]=n} ||X_m[k]||^2 \tag{2.1}$$

$$p[k] = \left(round \left[N_0 \ \log_2 \left(\frac{k}{J} \cdot \frac{\nu}{f_0} \right) \right] \right) \mod N_0$$

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 [5], 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 преодоления второго недостатка существует множество дополнительных преобразований.

В [16] 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 максимуы в каждом столбце). Каждый из них учитывался при вычислении не одного, а нескольких компонентов 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 $f_n, f_n/2, f_n/3, \ldots$ с соответствующими весами. Такой вектор признаков получил название *гармонический профиль тональных классов* (harmonic pitch class profile).

В [34] и [26] 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 [50]). В [34] допускается только перемещение энергии в пределах одного столбца спектрограммы, в [26] допускается таже 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 [25] 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).

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 [51], воспринимаемая громкость звука приблизительно пропорциональна десятичному логарифму уровня звуковой мощности (sound power level). Поэтому имеет смысл перед преобразованием спектрограммы в последовательность 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) [41].

В [52] 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. Похожие действия выполняются при вычислении мел-частотных кепстральных коэффициентов [53], 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 [31].

В [27] 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 A-взвешивание [54].

В [55] 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 порогами, 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 statictics* ($CENS_d^w$).

В [20], [22], [26], [27], [8], [36] строятся отдельные спектрограммы для низкочастотной и высокочастотной области спектра, граница между которыми обычно пролегает в диапазоне от 200 Гц до 250 Гц. Соответственно, получается два набора хроматических векторов, используемых в дальнейшем анализе.

В [56] были предложены особые признаки, не являющиеся хроматическими. Они являются векторами в пространстве *Tonnetz* [57], [58], моделирующем взаимоотношения между ступенями равномерно темперированного строя. Согласно [56], 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 [18], [59], [36], [7].

Сравнение качества работы некоторых из описанных типов признаков в приложении к задаче распознавания аккордов было проведено в [41]. 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 [7]. Описанные выше 6-мерные признаки получаются из спектрограммы путём применения свёрточной нейронной сети [60]. При этом не применяются никакие знания о свойствах спектра или музыки. Предполагается, что нейронная сеть сама определит наиболее характерные свойства в процессе обучения.

2.4 Классификация векторов признаков

На этом этапе находится решение задачи распознавания аккордов в звукозаписи: 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 выделялось 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 ([15], [16], [21], [23], [34], [29], [30], [31], [41], [27], [36], [7]). В некоторых работах выделяют также отдельные классы для доминантсептаккордов ([14], [20], [28], [22], [25], [8]), других септаккордов ([14], [25]), уменьшенных и увеличенных ([14], [17], [59], [20], [33], [22], [25], [61]) 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 [16], [23], в одном из вариантов [41]. Мерой расстояния может выступать косинусное расстояние, евклидово расстояние, расхождение Кульбака-Лейблера и другие. Их сравнение было проведено в [23]. 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 *до*).

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 [23]) в шаблоны также включают информацию о гармонических обертонах входящих в аккорд нот. Ноты, соответствующие частотам гармонических обертонов, могут быть получены из формулы (1.2). Вклад обертона в соответствующую компоненту шаблона определялся в [40] как

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

где k – номер обертона, а $h < 1$ – параметр. 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 [16], [23].

В [22] 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 [31], 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) [62] также нашли применение в алгоритмах распознавания аккордов. В отличие от алгоритма ближайшего соседа, они позволяют в явном виде моделировать вероятность перехода между двумя заданными аккордами. Дадим формальное определение элементов СММ.

- Набор состояний модели $Q = \{Q_1, Q_2, \ldots, Q_{N_{states}}\}$. За q_t будем обозначать состояние модели в момент времени t.
- Множество наблюдаемых символов $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_{M_{symbols}}\}$.
- 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 $v_j(k) = P\{\lambda_k \; at \; t | q_t = S_j\}$ 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 [15], в одном из вариантов [19], в нескольких вариантах [29]), так и определяться в результате обучения СММ (как в [17], в нескольких вариантах [19], [20], [21], [24], в одном из вариантов [29], [41], [26], [27]). Вероятности появления наблюдаемых символов обычно моделируются одним многомерным нормальным распределением (как в [14], [15], [19], [27], [36]) или смесью многомерных нормальных распределений (как в [17], [21], [24], [29], [26]). При обучении обычно используется итеративный метод математического ожидания – модификации (expectation-modification), также называемый методом Баума-Уэлша или методом прямого-обратного хода. В [24] 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 [31], [7]).

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, соответствущую 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 матожиданий 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 матожиданий 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 [14], [19], [29], [26]. 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 [36] 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 [17], [20], [21], [26]. В этом случае можно регулировать параметры моделей каждого отдельного аккорда (как в [20]), а также добавлять штраф за переход от модели одного аккорда к модели другого аккорда (как в [21], [26]).

Были предложены различные способы для учёта информации о предыдущих состояниях модели в том числе через введение понятий жанра и тональности. В [21] использовалась языковая модель, которая позволяет учитывать более чем одно предыдущее состояние СММ. В [18] 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 [59] аналогичным образом строились отдельные СММ для 6 различных музыкальных жанров. В [61] 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 [18] и [59] использовались звукозаписи, сгенерированные из MIDI-файлов. В [61] исползовался достаточно большой набор реальных музыкальных звукозаписей. Тональность может быть явным образом введена в саму СММ наряду с басовой нотой. Предложенная в [27] 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 [25] было предложено использовать динамическую байесовскую сеть, которая, по сути, является обобщением СММ (см. [63]). 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 [34] 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 [17] было предложено заменить СММ на условное случайное поле [64]. Оно определяется следующим образом. Обозначим за $\boldsymbol{X}$ и $\boldsymbol{Y}$ множество наблюдений и множество случайных переменных соответственно. Пусть $G = (V, E)$ – такой граф, что $\boldsymbol{Y} = (\boldsymbol{Y}_v)_{v \in V}$, то есть $\boldsymbol{Y}$ можно проиндексировать вершинами этого графа. Тогда $(\boldsymbol{X}, \boldsymbol{Y})$ называется *условным случайным полем*, если случайные переменные $\boldsymbol{Y}_v$ при условии $\boldsymbol{X}$ удовлетворяют марковскому свойству с учётом графа: $p(\boldsymbol{Y}_v | \boldsymbol{X}, \boldsymbol{Y}_w, w \sim v) = p(\boldsymbol{Y}_v | \boldsymbol{X}, \boldsymbol{Y}_w, w \sim v)$, 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 $p(\boldsymbol{X}, \boldsymbol{Y})$, а $p(\boldsymbol{Y} | \boldsymbol{X})$. Кроме того, в такой модели каждое скрытое состояние зависит не только от текущего наблюдения, но от всей предыдущей последовательности наблюдений. В [17] отмечается, что условное случайное поле обучается существенно дольше, чем СММ.

В [8] была предложена полноценная модель 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 [32] 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 [33].

Подход, в чём-то похожий на алгоритм Витерби, был предложен в [30]. 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 [65]). Затем все наиболее вероятные кандидаты объединяются в пары. Расстояние между парами (аккорд, тональность) определяется в соответствии с [66] 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. TODO

Глава 3

Распознавание аккордов без использования машинного обучения

РИСУНОК: общая схема реализованного метода

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 [11], 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 композиции. 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 [67] в разделе 4.1. В рамках данной работы используется алгоритм, похожий на предложенный в [43]. Звукозапись делится на короткие фрагменты между моментами времени $t_m, m = 0, 1, 2, \ldots, M'$, 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. На 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.

ЭКСПЕРИМЕНТ: распознавание аккордов с определением частоты настройки и без него.

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 позволят 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* [38] и *Beat tracker* [37] из набора *Queen Mary Vamp plugins*. Вторая библиотека потребовалась для обработки тех композиций, в которых *Beatroot* не смог определить начала метрических долей.

ЭКСПЕРИМЕНТ: разница в качестве распознавания с использованием одного и другого биттрекеров.

3.1.3 Получение спектра

Моменты начала метрических долей $(t_0, t_1, \ldots, t_{M-1})$ 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 на каждом фрагменте определить интенсивность звука для каждой из указанных частот.

Во многих работах (например, в [41], [31]) 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}) 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.

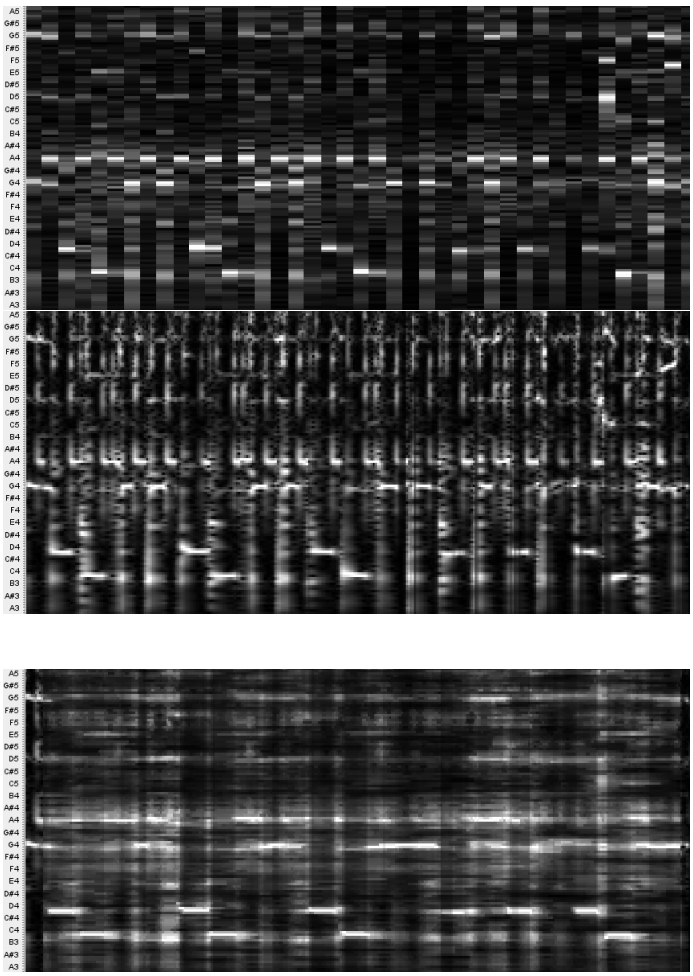
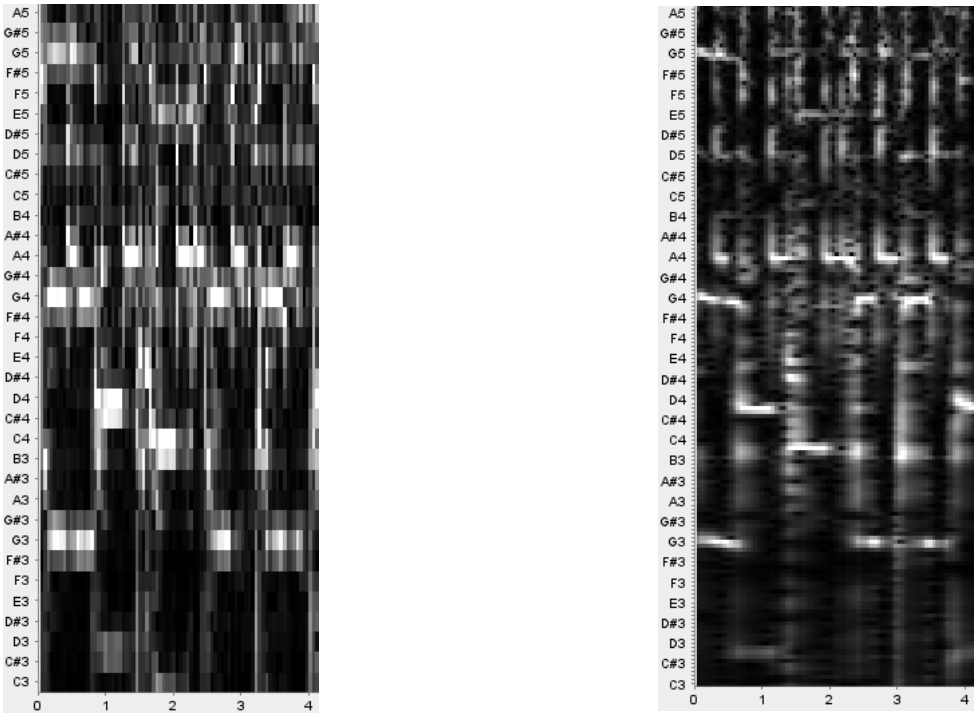


Рисунок 3.1: Фрагменты спектрограммы *The Beatles – Love Me Do* при $T = 1$ (вверху), $T = 8$ (в середине), $T = 8$ после сглаживания с $w = 19$ (внизу).

Равномерно темперированный строй предполагает расположение $N_0 = 12$ ступеней звукоряда в пределах октавы, поэтому удобно выбирать N_0 кратным 12. Большие значения N_0 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 [15], [16], [20], [23], [29], [31] использовалось значение $N_0 = 36$. Как показано ниже, $N_0 = 60$ позволяет добиться лучшего результата.

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 [34] используемый диапазон ограничен сверху 1000 Гц, а в [31] – 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.

ЭКСПЕРИМЕНТ: сравнение качества распознавания для N_0=12,36,60, и охвата в 4,5,6 октав. ЭКСПЕРИМЕНТ: сравнение с БПФ при тех же условиях (с определением ритма и частоты настройки).



а) б)

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

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.

РИСУНОК: спектр с четко выраженными ударными и гитарой и вокалом

Как видно из рисунка, 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 [46] и [68]. В данном случае полное разделение является излишним, необходимо только подавить перкуссионные компоненты.

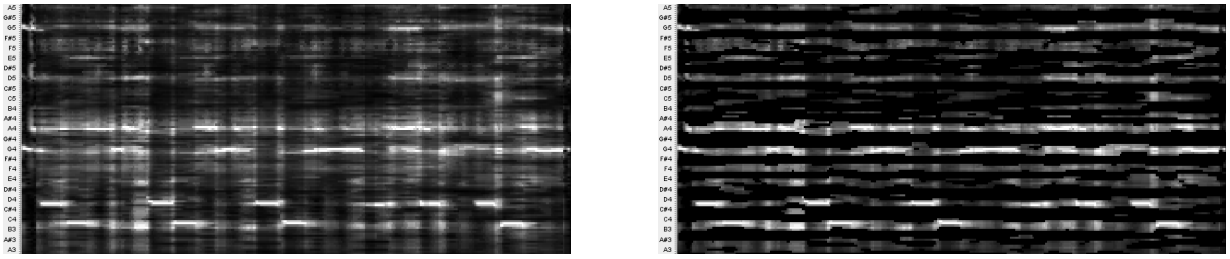
Маух в [25] предложил вычитать из спектрограммы так называемый фоновый спектр. При этом каждое значение спектрограммы $C_m[n]$ заменять на $\frac{C_m[n]-\mu_m[n]}{\sigma^q_m[n]}$, где $\mu_m[n]$ представляет собой среднее значение, а $\sigma^q_m[n]$ – среднеквадратическое отклонение в пределах отрезка 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 [69] 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.



а) б)

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

Еще один подход к подавлению перкуссионных компонент лёг в основу алгоритма вычисления признаков CRP [52]. Будем рассматривать $C_m[n]$ как сигнал (количество энергии, приходящейся на данную частоту, в зависимости от частоты). Применим к этой функции дискретное косинусное преобразование.

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

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.

Как показывает практика, важным шагом является применение к спектрограмме логарифмического преобразования: каждая компонента $C_m[n]$ заменяется на $\log_{10}(1000C_m[n] + 1)$. После него соотношения между компонентами спектрограммы лучше соответствуют человеческому восприятию интенсивности звука.

РИСУНОК: столбец спектрограммы до ДКП и после ДКП-зануление-ОДКП. РИСУНОК: спектрограмма до логарифмического преобразования и после него. ЭКСПЕРИМЕНТ: сравнение качества распознавания с применением разных методов очистки.

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 [25] и [31] повторяющиеся фрагменты композиции использовались для улучшения качества распознавания аккордов. В обеих методах строились матрицы самоподобия для 12-мерных хроматических векторов признаков с использованием в качестве меры подобия коэффициента корреляции Пирсона (в [25]) и евклидового расстояния (в [31]). В полученной матрице находятся линии, параллельные главной диагонали, которые соответствуют похожим друг на друга фрагментам. Эти фрагменты затем используются для дополнительного сглаживания спектрограммы.

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.

При помощи полученной матрицы можно скорректировать все столбцы C_m:

$$\widehat{C}_m = \frac{\sum\limits_{j=0}^{M-1}(1 - s_{mj})C_j}{\sum\limits_{j=0}^{M-1}(1 - s_{mj})}$$

РИСУНОК: матрица самоподобия ЭКСПЕРИМЕНТ: сравнение качества распознавания со сглаживанием и без него

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.

ЭКСПЕРИМЕНТ: зависимость качества распознавания от количества гармоник в шаблонах.

Для получения 12-мерных хроматических векторов сначала ко всем значениям $\widehat{C}_m[n],\ 0 \leq n < N_0$, прибавляются значения $\widehat{C}_m[n + N_0],\ \widehat{C}_m[n + 2N_0],\ \widehat{C}_m[n + 3N_0],\ \ldots$ 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:

$$D_m[n] = B_m[b_0 n - 1] + B_m[b_0 n] + B_m[b_0 n + 1], \quad m = 0, \ldots, M - 1, \quad n = 0, \ldots, 11$$

Для вычисления $D_m[0]$ в качестве $B_m[-1]$ используются $B_m[59]$.

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

3.4.2 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.

ЭКСПЕРИМЕНТ: качество распознавания до и после применения эвристик ПРИМЕРЫ композиций, где это работает и где не работает

3.5 Выводы

TODO

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 представлениям – 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 представлениям лежит многослойная нейронная сеть. Важным свойством таких алгоритмов является возможность предварительного обучения каждого слоя нейронной сети в отдельности без учителя, на неразмеченных данных. Благодаря нему требуется существенно меньше размеченных данных для окончательного обучения нейронной сети в целом.

В 2012 году Хамфри в [7] предложил использовать один из методов обучения представлениям – свёрточные нейронные сети – для получения признаков, позволяющих классифицировать звучащий аккорд. В данной работе рассматривается другой тип таких методов – многослойные очищающие автоассоциаторы, в том числе их рекуррентный вариант. Рекуррентные многослойные очищающие автоассоциаторы были успешно использованы для распознавания речи в [70].

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 и обзор литературы

Определения в этом разделе даны в соответствии с [71].

Автоассоциатор (автоэнкодер) представляет из себя пару преобразований:

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

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

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:

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

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 [72]:

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

В дальнейшем будем использовать значение $\rho = 0.05$, также в соответствии с [72].

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 [71], что во многих случаях внутренние представления, которые получаются при помощи очищающего автоассоциатора, позволяют получить лучшие результаты в задачах классификации по сравнению с представлениями, полученными при помощи обычных автоассоциаторов. В [71] 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. В полученной модели слои можно обучать друг за другом на неразмеченных данных. Значения, полученные в скрытом слое последнего из автоассоциаторов, могут быть использованы как векторы признаков.

РИСУНОК: многослойный автоассоциатор

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 [73]. Промежуточное представление $y(x_t)$ в таком случае вычисляется как

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

РИСУНОК: рекуррентный многослойный автоассоциатор

4.2 Построение нейронной сети и предобучение при помощи автоассоциаторов

Существенным недостатком многослойных 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 внутренный 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 точно так же, как и для других типов векторов признаков.

Предварительное обучение слоёв-автоассоциаторов будем производить методом мини-пакетного (mini-batch) стохастического градиентного спуска (ССЫЛКА). При этом сначала обучается первый слой на всём обучающем множестве, затем полученные значения используются для обучения второго слоя, и так далее. Окончательное обучение сети в целом также производится методом мини-пакетного стохастического градиентного спуска.

Для очищающего автоассоциатора возможные помехи на входе можно смоделировать при помощи аддитивного гауссового шума $\tilde{x}|x \sim \mathcal{N}(x, \sigma_0^2 I)$, где σ_0 – параметр. Это соответствует предположению о том, что помехи равновероятны в любой области спектра. Возможны и другие варианты моделирования помех. Для обучения последующих слоёв также будем использовать эту модель шума с параметром σ_i вместо σ_0 для i-го слоя.

Естественным выбором для функции стоимости будет квадрат евклидова расстояния между шаблоном и вектором на выходе (ССЫЛКА на другие варианты):

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

Для случая отсутствия аккорда в качестве соответствующего шаблона будем использовать 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 аккорд *до-мажор* может встречаться в обучающей выборке в разы 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 [14]. 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 эмулировать, добавив одну октаву к частотному диапазону спектрограммы, после чего просто сдвигать по столбцу спектрограммы окно на октаву короче.

ЭКСПЕРИМЕНТ: а если делать просто циклический сдвиг?

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.

4.3 Выводы

1. TODO

Рисунок 4.1: TeX.

А это две картинки под общим номером и названием:

а)

б)

Рисунок 4.2: Очень длинная подпись к изображению, на котором представлены две фотографии Дональда Кнута

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 312 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 5.6 сравниваются скорости работы реализованных алгоритмов.

5.1 Оценка качества распознавания аккордов

Поскольку алгоритмы распознавания аккордов предназначаются 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 $y \in \overline{Y}$, звучащий в этот момент. При этом набор $\overline{Y}$ 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* [74]. 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* [75]. Текстовые аннотации для 100 композиций японской и западной популярной музыки.
- *Billboard* [76]. 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* [77]. Текстовые аннотации для 195 композиций американской популярной музыки.
- *Robbie Williams annotations*. Текстовые аннтоации 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 [78] и Пауэлс и Питерс [79]. Последние предлагают следующую конструкцию для определения схожести двух последовательностей аккордов.

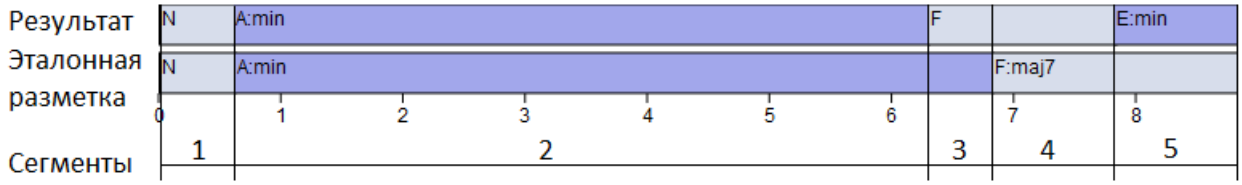


Рисунок 5.1: Сопоставление последовательностей аккордов.

Пусть заданы 2 последовательности аккордов: правильная и определённая при помощи алгоритма. Объединим множества границ аккордов из обоих 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 $\mathcal{M} : C_{MI} \to C_{MO}$, которое «сложным» аккордам из множества C_{MI} будет сопоставлять «простые» аккорды из множества C_{MO}. Нужно выбрать эти множества и отображение $\mathcal{M}$ таким образом, чтобы $C_{EST} \subset C_{MI}$. Сравниваться при этом будут аккорды $\mathcal{M}(c_{ref})$ и $\mathcal{M}(c_{est})$. Сегменты, на которых $c_{ref} \notin C_{MI}$, отбрасываются. Примером отображения $\mathcal{M}$ 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 $\mathcal{M}(c_{ref}) \in \mathcal{M}(C_{LI} \cap C_{MI}) \cap C_{LO}$.

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

Как видно из [79], полученные цифры сильно различаются в зависимости от выбора отображения $\mathcal{M}$ и функции оценки $\mathcal{S}$, 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.

В экспериментах в рамках данной работы будем использовать метрику “Mirex2010” из [79]. 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 $\mathcal{S} : C_{SR} \times C_{SE} \to \{0, 1\}$.

Отметим, что при использовании этой метрики ни один сегмент не отбрасывается. Её выбор является в достаточной степени произвольным и связан исключительно с тем фактом, что именно она использовалась в соревновании MIREX Audio Chord Estimation в 2010, 2011 и 2012 годах.

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

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

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 $L_i = \sum_{i=j}^{N_{segm}} \ell_j$ длину i-й композиции. Тогда совокупная метрика для коллекции, называемая *взвешенным средним коэффициентом перекрытия (weighted average overlap ratio, WAOR)*, вычисляется следующим образом:

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

Такой же способ усреднения применяется в соревнованиях MIREX Audio Chord Estimation.

5.1.3 Сопоставление границ сегментов

Метрика для сопоставления границ сегментов была введена Маухом в [80], но не получила широкого распространения. 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

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

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 [25], эти предположения не является верными в данном случае).

Однако, если в соответствии с критерием Фридмана удаётся 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 T-теста, при его допускаются множественные попарные сравнения. Этот метод используется для сравнения качества работы разных алгоритмов в рамках всех соревнований MIREX [81].

5.2 Вычисление спектрограммы

На данном этапе необходимо выбрать наилучшие из доступных алгоритмов для определения ритма и определения частоты настройки. 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* [38], *Beat tracker* от Дэвиса [37] (*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* [82] (*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.

Таблица 5.1: Влияние алгоритма определения ритма на качество распознавания аккордов

Алгоритм	WAOR	Сегментация
BeatRoot + DBT	0.7516	0.7907
DBT	0.7317	0.7659
IBT	0.7189	0.7425

Наилучшие полученные для данных алгоритмов результаты показаны в таблице 5.1. *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.

5.2.2 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, $N_0 = 12 \cdot 20 = 240$ компонент на октаву;
- $f_{min} = 220$ Гц, $N_0 = 12 \cdot 10 = 120$ компонент на октаву;
- $f_{min} = 440$ Гц, $N_0 = 12 \cdot 20 = 240$ 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 [80], раздел 3.1.3 и реализованного в виде плагина для системы *Vamp*.

[1] http://www.isophonics.net/QMVampPlugins
[2] http://www.vamp-plugins.org/

Таблица 5.2: Влияние алгоритма определения частоты настройки на качество распознавания аккордов

Алгоритм	WAOR	Сегментация
$f_{min} = 220$ Гц, $N_0 = 120$	0.7516	0.7892
$f_{min} = 220$ Гц, $N_0 = 240$	0.7516	0.7907
$f_{min} = 440$ Гц, $N_0 = 240$	0.7512	0.7899
Маух [80]	0.7477	0.7888
–	0.7429	0.7854

Результаты экспериментов приведены в таблице 5.2. Как видно, определение частоты настройки приводит к улучшению качества распознавания аккордов, причем при использовании любого из перечисленных алгоритмов это улучшение статистически значимо. Использование любого из описанных вариантов алгоритма из раздела 3.1.1 также приводит к статистически значимому улучшению по сравнению с алгоритмом Мауха, но за счёт большего времени работы. В то же время между собой эти 3 варианта отличаются незначительно.

5.2.3 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 5.3 приведены значения для некоторых комбинаций T и w.

Таблица 5.3: Влияние параметров T и w на качество распознавания аккордов

Значения параметров	WAOR	Сегментация
$T = 2, w = 1$	0.6780	0.7561
$T = 2, w = 3$	0.7368	0.7830
$T = 2, w = 5$	0.7278	0.7721
$T = 4, w = 5$	0.7467	0.7886
$T = 4, w = 7$	0.7496	0.7884
$T = 4, w = 9$	0.7451	0.7827
$T = 8, w = 13$	0.7504	0.7910
$T = 8, w = 15$	0.7516	0.7907
$T = 8, w = 17$	0.7494	0.7863

При $T = 2$ качество распознавания аккордов существенно хуже для случая $w = 1$, что фактически соответствует отказу от добавления промежуточных значений и последующих сглаживания и прореживания. Статистически значимых отличий между вариантами $w = 3$ и $w = 5$ нет.

При $T = 4$ влияние параметра w уже не столь существенно. Наилучший результат получен при $w = 7$, и он статистически значимо превосходит результаты, полученные при $w = 5$ и $w = 9$ (разница между которыми, в свою очередь, не является значимой).

При $T = 8$ только разница между вариантами $w = 15$ и $w = 17$ оказалась статистически значимой. При этом в абсолютных значениях различия ещё меньше, чем для $T = 4$.

Отдельно было проведено сравнение наилучших вариантов для каждого значения T. Между вариантами $T = 4, w = 7$ и $T = 8, w = 15$ нет статистически значимой разницы в качестве

распознавания аккордов; разница в абсолютных значениях метрик также незначительна. При $T = 2, w = 3$ был получен существенно худший результат.

Интересно, что наилучшие результаты достигаются при $w = 2T - 1$, что для каждого момента времени исходной последовательности соответствует фильтрации по значениям спектра, вычисленным в этот момент и в $T - 1$ добавленных промежуточных точках справа и слева. Из этого эксперимента видно, что увеличение разрешения по времени для последовательности моментов начала метрических долей по крайней мере в 4 раза приводит к существенному улучшению качества распознавания аккордов.

Таблица 5.4: Влияние параметра N_0 на качество распознавания аккордов

Значения N_0	WAOR	Сегментация
$N_0 = 12$	0.6257	0.7743
$N_0 = 36$	0.7518	0.7951
$N_0 = 60$	0.7516	0.7907

В таблице 5.4 приведены результаты, полученные при разных значениях количества компонент преобразования постоянного качества, приходящихся на одну октаву. Очевидно, что при наличии как минимум 36 компонент на октаву (3 компоненты на ноту) качество распознавания аккордов существенно повышается. Различия между $N_0 = 36$ и $N_0 = 60$ не являются 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.

5.3 Преобразования спектрограммы

Основные вопросы относительно выбора параметров, возникающие на этом этапе:

1. действительно ли применение аналога фильтра Превитт повышает качество распознавания акокрдов;

2. каково оптимальное значение для количества зануляемых первых коэффициентов дискретного косинусного преобразования ξ при вычислении признаков CRP;

3. действительно ли сглаживание с использованием матрицы самоподобия для столбцов спектрограммы лучше, чем с использованием такой матрицы для векторов признаков и без использования самоподобия вообще;

4. каково оптимальное значение для доли сохраняемых в матрице самоподобия значений ζ.

5.4 Нейронные сети

Для описанного в главе 4 метода получения векторов признаков с использованием нейронных сетей важными являются следующие вопросы:

1. какое количество элементов во входном слое сети позволяет получить наилучшие векторы признаков;

2. влияет ли предварительное применение логарифмического преобразования к спектрограмме на полученные векторы признаков;

3. каковы оптимальное количество скрытых слоёв и их размеры в нейронной сети;
4. действительно ли использование рекуррентных соединений повышает качество распознавания аккордов;
5. каковы оптимальные значения для уровней шума при обучении очищающих автоассоциаторов;
6. и др.

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.

5.5 Классификация векторов признаков

Для выбранного способа классификации векторов признаков необходимо определить наилучшие значения следующих параметров:

1. количество учитываемых при генерации шаблонов гармоник;
2. степень убывания вклада гармоник в шаблон s.

Помимо этого необходимо количественно оценить повышение качества распознавания при использовании описанных в разделе 3.4.2 эвристик.

5.6 Быстродействие

5.7 Выводы

1. TODO

Заключение

Основные результаты работы заключаются в следующем.

1. На основе анализа . . .
2. Численные исследования показали, что . . .
3. Математическое моделирование показало . . .
4. Для выполнения поставленных задач был создан . . .

И какая-нибудь заключающая фраза.

Список рисунков

Список таблиц

Литература

1. Schüler Nico. Reflections on the history of computer-assisted music analysis I: predecessors and the beginnings. // Muzikoloski zbornik. 2005. T. 41. C. 31–43. URL: http://uweb.txstate.edu/ ns13/Schuler-CAMA-I.pdf.

2. Freedman M. David. Analysis of Musical Instrument Tones // The Journal of the Acoustical Society of America. 1967. T. 41, № 4A. C. 793–806. URL: http://link.aip.org/link/?JAS/41/793/1.

3. Moorer James A. On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer: Master's thesis: Stanford University. Stanford, CA, 1975. URL: https://ccrma.stanford.edu/files/papers/stanm3.pdf.

4. Martin Keith D. Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing. 1996.

5. Fujishima Takuya. Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music // Proc. ICMC, 1999. 1999. C. 464–467. URL: http://ci.nii.ac.jp/naid/10013545881/en/.

6. Aono Y., Katayose H., Inokuchi S. A Real-time Session Composer with Acoustic Polyphonic Instruments // Proceedings of ICMC 1998. 1998. C. 236–239.

7. 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. C. 453–456.

8. De Haas W. Bas, Magalhães José Pedro, Wiering Frans. 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.

9. Большая советская энциклопедия. / под ред. А. М. Прохоров. 3-е изд. М.: Советская энциклопедия, 1972. Т. 9.

10. Способин И. В. Элементарная теория музыки / под ред. В. Григоренко. Москва "КИФА-РА 2012.

11. Lerch Alexander. Audio content analysis: an introduction. John Wiley & Sons, Inc., Hoboken, New Jersey, 2012.

12. Hugo Fastl Eberhard Zwicker. Psychoacoustics - Facts and Models / под ред. Manfred R. Schroeder Thomas S. Huang, Teuvo Kohonen. Springer-Verlag Berlin Heidelberg, 2007.

13. Levitin Daniel J. This is Your Brain on Music: Understanding a Human Obsession. Atlantic Books Ltd., 2006.

14. Sheh Alexander, Ellis Daniel P. W. Chord segmentation and recognition using EM-trained hidden markov models. // ISMIR. 2003.

15. Bello Juan Pablo, Pickens Jeremy. 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. С. 304–311. http://ismir2005.ismir.net/proceedings/1038.pdf.

16. Lee K. Automatic chord recognition from audio using enhanced pitch class profile // ICMC Proceedings. 2006.

17. A Cross-Validated Study of Modelling Strategies for Automatic Chord Recognition in Audio / John Ashley Burgoyne, Laurent Pugin, Corey Kereliuk [и др.] // Proceedings of the 8th International Conference on Music Information Retrieval. Vienna, Austria: 2007. September 23-27. С. 251–254. http://ismir2007.ismir.net/proceedings/ISMIR2007_p251_burgoyne.pdf.

18. Lee Kyogu, Slaney Malcolm. 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. С. 245–250. http://ismir2007.ismir.net/proceedings/ISMIR2007_p245_lee.pdf.

19. Papadopoulos Hélène, Peeters Geoffroy. Large-Scale Study of Chord Estimation Algorithms Based on Chroma Representation and HMM // CBMI / под ред. Jenny Benois-Pineau. IEEE, 2007. С. 53–60.

20. Mauch Matthias, Dixon Simon. A Discrete Mixture Model for Chord Labelling // Proceedings of the 9th International Conference on Music Information Retrieval. Philadelphia, USA: 2008. September 14-18. С. 45–50. http://ismir2008.ismir.net/papers/ISMIR2008_214.pdf.

21. Khadkevich Maksim, Omologo Maurizio. 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. С. 561–566. http://ismir2009.ismir.net/proceedings/OS7-4.pdf.

22. Mauch Matthias, Noland Katy, Dixon Simon. 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. С. 231–236. http://ismir2009.ismir.net/proceedings/PS2-7.pdf.

23. Oudre Laurent, Grenier Yves, Févotte Cédric. 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. С. 153–158. http://ismir2009.ismir.net/proceedings/PS1-17.pdf.

24. Minimum Classification Error Training to Improve Isolated Chord Recognition / Jeremy Reed, Yushi Ueda, Sabato Siniscalchi [и др.] // Proceedings of the 10th International Society for Music Information Retrieval Conference. Kobe, Japan: 2009. October 26-30. С. 609–614. http://ismir2009.ismir.net/proceedings/PS4-6.pdf.

25. Mauch Matthias, Dixon Simon. 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. С. 135–140. http://ismir2010.ismir.net/proceedings/ismir2010-25.pdf.

26. Khadkevich Maksim, Omologo Maurizio. 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. С. 181–184.

27. Harmony Progression Analyzer for MIREX 2011 / Yizhao Ni, Matt Mcvicar, Raul Santos-Rodriguez [и др.]. 2012.

28. Zhang Xinglin, Gerhard David. Chord Recognition using Instrument Voicing Constraints // Proceedings of the 9th International Conference on Music Information Retrieval. Philadelphia, USA: 2008. September 14-18. С. 33–38. http://ismir2008.ismir.net/papers/ISMIR2008_241.pdf.

29. 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. С. 1–8.

30. Concurrent Estimation of Chords and Keys from Audio / Thomas Rocher, Matthias Robine, Pierre Hanna [и др.] // Proceedings of the 11th International Society for Music Information Retrieval Conference. Utrecht, The Netherlands: 2010. August 9-13. С. 141–146. http://ismir2010.ismir.net/proceedings/ismir2010-26.pdf.

31. Cho Taemin, Bello Juan 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. С. 651–656. http://ismir2011.ismir.net/papers/OS8-4.pdf.

32. Automatic Chord Transcription with Concurrent Recognition of Chord Symbols and Boundaries / Takuya Yoshioka, Tetsuro Kitahara, Kazunori Komatani [и др.] // 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.

33. Automatic Chord Recognition Based on Probabilistic Integration of Chord Transition and Bass Pitch Estimation / Kouhei Sumi, Katsutoshi Itoyama, Kazuyoshi Yoshii [и др.] // Proceedings of the 9th International Conference on Music Information Retrieval. Philadelphia, USA: 2008. September 14-18. С. 39–44. http://ismir2008.ismir.net/papers/ISMIR2008_236.pdf.

34. Weller Adrian, Ellis Daniel, Jebara Tony. 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. С. 590–595. URL: http://dx.doi.org/10.1109/ICMLA.2009.132.

35. Leveraging Noisy Online Databases for Use in Chord Recognition / Matt Mcvicar, Yizhao Ni, Raul Santos-Rodriguez [и др.] // Proceedings of the 12th International Society for Music Information Retrieval Conference. Miami (Florida), USA: 2011. October 24-28. С. 639–644. http://ismir2011.ismir.net/papers/OS8-2.pdf.

36. Chord Recognition Using Duration-explicit Hidden Markov Models / Ruofeng Chen, Weibin Shen, Ajay Srinivasamurthy [и др.] // 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.

37. Davies Matthew E. P., Plumbley Mark D. Context-Dependent Beat Tracking of Musical Audio // Trans. Audio, Speech and Lang. Proc. Piscataway, NJ, USA, 2007. March. Т. 15, № 3. С. 1009–1020. URL: http://dx.doi.org/10.1109/TASL.2006.885257.

38. Dixon Simon. Evaluation of the Audio Beat Tracking System BeatRoot // Journal of New Music Research. 2007. March. Т. 36, № 1. С. 39–50. URL: http://dx.doi.org/10.1080/09298210701653310.

39. Ellis Daniel P. W. Beat tracking by dynamic programming // Journal of New Music Research. 2007. Т. 36(1). С. 51–60.

40. Gómez E. Tonal Description of Music Audio Signals. Ph.D. thesis: Universitat Pompeu Fabra. 2006. URL: files/publications/emilia-PhD-2006.pdf.

41. Analyzing Chroma Feature Types for Automated Chord Recognition / Nanzhu Jiang, Peter Grosche, Verena Konz [и др.] // Proceedings of the AES 42nd International Conference: Semantic Audio. Ilmenau, Germany: AES, 2011. С. 285–294.

42. Harte C. A., Sandler M. Automatic chord identification using a quantised chromagram // Proc. of the 118th Convention. of the AES. 2005.

43. Zhu Yongwei, Kankanhalli Mohan S., Gao Sheng. Music Key Detection for Musical Audio // Multi-Media Modeling Conference, International. Los Alamitos, CA, USA, 2005. Т. 0. С. 30–37.

44. Peeters Geoffroy. 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. С. 127–131.

45. Khadkevich Maksim, Omologo Maurizio. 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.

46. Separation of a Monaural Audio Signal into Harmonic/Percussive Components by Complementary Diffusion on Spectrogram / Nobutaka Ono, Ken-Ichi Miyamoto, Jonathan Le Roux [и др.] // Proceedings of the EUSIPCO 2008 European Signal Processing Conference. 2008. August.

47. А. Оппенгейм Р. Шафер. Цифровая обработка сигналов / под ред. О. Н. Кулешова. Москва "Техносфера 2006.

48. Brown Judith, Puckette Miller S. An efficient algorithm for the calculation of a constant Q transform // Journal of the Acoustical Society of America. 1992. November. Т. 92, № 5. С. 2698–2701.

49. Л. Рабинер Б. Гоулд. Теория и применение цифровой обработки сигналов / под ред. Л. Якименко. Москва "МИР 1978.

50. Kodera K., Gendrin R., Villedary C. Analysis of time-varying signals with small BT values // Acoustics, Speech and Signal Processing, IEEE Transactions on. 1978. Т. 26, № 1. С. 64–76.

51. Fletcher Harvey, Munson W. A. Loudness, Its Definition, Measurement and Calculation // The Journal of the Acoustical Society of America. 1933. October. Т. 5, № 2. С. 82–108.

52. Müller Meinard, Ewert Sebastian, Kreuzer Sebastian. 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. С. 1877–1880. URL: http://dx.doi.org/10.1109/ICASSP.2009.4959974.

53. Logan Beth. 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.

54. Talbot-Smith Michael. Audio Engineer's Reference Book. Taylor & Francis, 1999. URL: http://books.google.ru/books?id=LcySXZbdamEC.

55. Müller Meinard. Information retrieval for music and motion. Springer-Verlag Berlin Heidelberg, 2007.

56. Harte Christopher, Sandler Mark, Gasser Martin. 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. C. 21–26. URL: http://doi.acm.org/10.1145/1178723.1178727.

57. vv ll jj ffIntroduction to Neo-Riemannian Theory: A Survey and a Historical Perspective // Journal of Music Theory. 1998. Vol. 42, no. 2. P. pp. 167–180. URL: http://www.jstor.org/stable/843871.

58. Chew Elaine. Towards a Mathematical Model of Tonality. Ph.D. thesis: Massachusetts Institute of Technology. 2000. Feb. URL: http://www-bcf.usc.edu/˜echew/papers/Dissertation2000/ec-dissertation.pdf.

59. Lee Kyogu. A System for Automatic Chord Transcription from Audio Using Genre-Specific Hidden Markov Models // Adaptive Multimedial Retrieval: Retrieval, User, and Semantics / под ред. Nozha Boujemaa, Marcin Detyniecki, Andreas NГjrnberger. Springer Berlin Heidelberg, 2008. T. 4918 из *Lecture Notes in Computer Science*. C. 134–146.

60. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio [и др.] // Proceedings of the IEEE. Nov. T. 86, № 11. C. 2278–2324.

61. Using Hyper-genre Training to Explore Genre Information for Automatic Chord Estimation / Yizhao Ni, Matt Mcvicar, Raul Santos-Rodriguez [и др.] // 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.

62. И. Рабинер. Скрытые марковские модели и их применение в избранных приложениях при распознавании речи: Обзор // ТИИЭР. 1989. Т. 77, № 2.

63. Ghahramani Zoubin. An Introduction to Hidden Markov Models and Bayesian Networks // IJPRAI. 2001. T. 15, № 1. C. 9–42.

64. Lafferty John D., McCallum Andrew, Pereira Fernando 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. C. 282–289. URL: http://dl.acm.org/citation.cfm?id=645530.655813.

65. Temperley D. The cognition of basic musical structures. Mit Press, 2001.

66. Lerdahl F. Tonal Pitch Space. Oxford University Press, USA, 2001.

67. Lerch Alexander. On the Requirement of Automatic Tuning Frequency Estimation // International Symposium/Conference on Music Information Retrieval. 2006. C. 212–215.

68. Fitzgerald D. Harmonic/Percussive Separation using Median Filtering // Audio. 2010. № 1. C. 10–13. URL: http://arrow.dit.ie/argart/9/.

69. 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. C. 480–485.

70. 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.

71. 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. T. 11. C. 3371–3408.

72. Ng Andrew. CS294A Lecture Notes. Sparse autoencoder. URL: http://www.stanford.edu/class/cs294a/sparseAutoencoder.pdf.

73. Elman Jeffrey L. Finding structure in time // Cognitive Science. 1990. T. 14, № 2. C. 179–211. URL: http://groups.lis.illinois.edu/amag/langev/paper/elman90findingStructure.html.

74. OMRAS2 Metadata Project 2009 / M. Mauch, C. Cannam, M. Davies [и др.] // Late-breaking session at the 10th International Conference on Music Information Retrieval, Kobe, Japan. 2009.

75. RWC Music Database: Popular, Classical and Jazz Music Databases / Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura [и др.] // ISMIR. 2002.

76. Burgoyne John Ashley, Wild Jonathan, Fujinaga Ichiro. 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 / под ред. Anssi Klapuri, Colby Leider. University of Miami, 2011. C. 633–638.

77. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures / Adam Berenzweig, Beth Logan, Daniel P. W. Ellis [и др.] // Comput. Music J. Cambridge, MA, USA, 2004. June. T. 28, № 2. C. 63–76. URL: http://dx.doi.org/10.1162/014892604323112257.

78. Harte C. Towards Automatic Extraction of Harmony Information from Music Signals. Ph.D. thesis: Queen Mary University of London, Centre for Digital Music. 2010.

79. Johan Pauwels Geoffroy 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.

80. Mauch Matthias. Automatic Chord Transcription from Audio Using Computational Models of Musical Context. Ph.D. thesis: Queen Mary University of London. 2010.

81. Downie Stephen J. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research // Acoustical Science and Technology. 2008. T. 29, № 4. C. 247–255. URL: http://www.jstage.jst.go.jp/article/ast/29/4/29_247/_article.

82. Beat Tracking for Multiple Applications: A Multi-Agent System Architecture With State Recovery / João Lobato Oliveira, Matthew E. P. Davies, Fabien Gouyon [и др.] // IEEE Transactions on Audio, Speech & Language Processing. 2012. T. 20, № 10. C. 2696–2706.

Приложение А

Название первого приложения

Некоторый текст.

Приложение В

Очень длинное название второго приложения, в котором продемонстрирована работа с длинными таблицами

В.1 Подраздер приложения

Вот размещается длинная таблица:

Параметр	Умолч.	Тип	Описание
&INP			
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума

продолжение следует

(продолжение)

Параметр	Умолч.	Тип	Описание
			2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
&SURFPAR			
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$)

продолжение следует

(продолжение)			
Параметр	Умолч.	Тип	Описание
			1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

B.2 Еще один подраздер приложения

Нужно больше подразделов приложения!

B.3 Очередной подраздер приложения

Нужно больше подразделов приложения!

B.4 И еще один подраздер приложения

Нужно больше подразделов приложения!