

Санкт-Петербургский государственный университет
Математическое обеспечение и администрирование
информационных систем

Информационно-аналитические системы

Шевченко Настасья Сергеевна

Распознавание последовательности аккордов в цифровом звуке

Бакалаврская работа

Научный руководитель:
ст. преп. Ярыгина А. С.

Рецензент:
ст. преп. Луцив Д. В.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY
Information Systems Administration and Mathematical Support

Analytical Information Systems

Shevchenko Nastasia Sergeevna

Automatic chord recognition in digital audio

Graduation Thesis

Scientific supervisor:
assistant Anna Yarygina

Reviewer:
assistant Dmitry Luciv

Saint-Petersburg
2016

Оглавление

Введение	5
1. Задача распознавания аккордов	6
1.1. Мотивировка	6
1.2. Постановка задачи	6
1.3. Математическая интерпретация	7
2. Обзор методов распознавания аккордов	8
3. Теоретические основы музыки	10
4. Подходы к решению задачи	12
4.1. Машинное обучение	12
4.2. Извлечение признаков	12
4.3. Алгоритмы	13
4.3.1. Наивный Байесовский классификатор	14
4.3.2. Метод ближайших соседей	15
4.3.3. Метод опорных векторов	15
5. Эксперименты	17
5.1. Данные	17
5.1.1. Описание входных данных	17
5.1.2. Обработка коллекции	18
5.2. Методы оценки точности алгоритмов машинного обучения	19
5.2.1. Кросс-валидация по музыкальным композициям .	19
5.2.2. Assurance	20
5.2.3. Chord Symbol Recall	20
5.3. Применение алгоритмов машинного обучения для музыкаль- ной коллекции	22
5.3.1. Результаты метода кросс-валидации по музыкаль- ным композициям	22
5.3.2. Результаты оценки CSR	27

6. Заключение	29
Список литературы	31

Введение

Объем цифрового контента, проходящего и хранящегося в интернете, растет ежегодно. По данным J'son & Partners Consulting на сегодняшний день самыми быстрорастущими жанрами цифрового контента стали видео и музыка. Лавинообразный рост видео и музыкальной информации, который на 2012 год составил 62% и 99% соответственно, приводит к необходимости анализа данных и, в частности, музыкального контента [12].

Задача научного исследования музыки определяет ряд направлений, таких как классификация музыки по жанрам, определение музыкальных инструментов, звучащих в произведении, распознавание голоса или звуков отдельных музыкальных инструментов в аудиозаписи и извлечение их партии из музыкального произведения. Последнее актуализирует задачу восстановления отдельных нот музыкального произведения и автоматической идентификации аккордов в цифровом звуке.

Задачей настоящей работы является автоматическое восстановление последовательности аккордов по цифровой записи музыкальной композиции.

1. Задача распознавания аккордов

1.1. Мотивировка

Поставленная в настоящей работе задача является конкретизацией задачи научного исследования музыкальных произведений, записанных в цифровом звуке. Актуальность решения настоящей задачи достаточно очевидна.

С одной стороны, автоматическое определение аккордов композиции может позволить сэкономить время составления музыкальной нотации произведения. Более того, реализация нашего исследования, например, может позволить людям, не имеющим глубоких познаний в музыкальной теории и абсолютного слуха, раскладывать любимые музыкальные произведения, для которых нотация отсутствует, на аккорды, а также наблюдать закономерности в музыкальных композициях, тем самым познавая теорию музыки на практике.

С другой стороны, полученные последовательности аккордов можно интерпретировать как решение задачи низкого уровня и использовать ее результат в качестве признаков для решения других задач более высокого уровня, например, для определения жанра песни или композитора. Кроме того, система последовательностей аккордов может быть полезна для экспертизы музыкального произведения с целью выявления плагиата.

1.2. Постановка задачи

В общем виде задача распознавания аккордов для всей музыкальной композиции состоит в идентификации звучащего в каждый момент времени аккорда. Входные данные представляют собой музыкальные произведения, а результатом являются последовательности аккордов, звучащие в различных временных интервалах.

1.3. Математическая интерпретация

Общая постановка задачи распознавания последовательности аккордов состоит в построении функции $f: X \rightarrow Z$, которая любому музыкальному произведению $x^i \in X$ ставит в соответствие последовательность аккордов y_t^i , такую, что:

$$f(x_t^i) = y_t^i \in Z, \quad (1)$$

где $t \in [0, t_{end_i}]$, t_{end_i} - время звучания песни, $X \subset R^n$, $Y \subset Z^n$, $Z = \{z_j \mid z_j - \text{аккорд}\}$

2. Обзор методов распознавания аккордов

Первые попытки реализации детектора аккордов были предложены ещё в 90-х годах. Ключевая идея, составляющая основу этих алгоритмов, заключалась в распознавании отдельных нот и дальнейшем разделении последовательности нот на аккорды.

Существует множество подходов для решения задачи транскрибирования, то есть выделения отдельных нот музыкального произведения, но все они так или иначе допускают сильные ограничения [3]. Это позволяет предположить, что настоящая задача не имеет решения в общем виде. Сделанное заключение объясняется несколькими причинами. Во-первых, обертоны, неизбежно присутствующие в музыкальном произведении, могут маскировать тон другой ноты, затрудняя или даже не позволяя определить ее. Во-вторых, ни один музыкальный инструмент не воспроизводит идеально звучащие последовательности нот, что также даёт неточности в вычислениях.

В 1999 году Т. Фуджишима предложил способ распознавания аккордов без выделения отдельно звучащих нот, который стал основой для всех используемых в настоящее время алгоритмов [4]. Основная идея этого метода состоит в применении дискретного оконного преобразования Фурье для получения спектрограммы звукозаписи. После чего для каждого вектора спектрограммы указывается соответствующий ему аккорд по методу ближайшего соседа. Не смотря на явные преимущества по сравнению с методом распознавания с предварительным транскрибированием, алгоритм Фуджишимы имеет свои недостатки, среди которых можно отметить следующие. Во-первых, наличие шумов в аудиозаписи сильно снижает точность распознавания. Во-вторых, происходящее на границах октав даже незначительное наложение предыдущих нот на звучащие в данный момент влечёт зачастую к их неправильному распознаванию.

Дальнейшие работы основываются на применении различных методов получения спектрограммы, используя быстрое преобразование Фурье, преобразование постоянного качества (constant-Q преобразование),

вейвлеты, а также некоторых подходов к решению задачи классификации аккордов, соответствующих полученным векторам спектрограммы, такие как метод ближайших соседей, метод опорных векторов и скрытые марковские модели [1, 9, 10]. В последние годы всё чаще для решения поставленной задачи применяют нейронные сети, а в качестве моделей классификации используются, например, рекуррентные нейронные сети [2].

3. Теоретические основы музыки

Прежде чем переходить к описанию алгоритмов для решения задачи распознавания аккордов, необходимо определить значение некоторых музыкальных терминов, используемых в контексте настоящей работы.

Так, основой любой музыкальной композиции является понятие *звука*. Звук, который мы слышим, – это механические колебания воздуха с некоторой фиксированной частотой. Такое определение даёт в своей статье Е.Г.Шилов [13]. Под нотой мы будем понимать графическое представление звука.

Обозначения основных семи нот представлено на (рис. 1).

Названия нот	до	ре	ми	фа	соль	ля	си
Буквенные обозначения	C	D	E	F	G	A	B

Рис. 1: Обозначения основных нот

В настоящей работе была использована классическая 12–ступенчатая *музыкальная шкала*. Для построения музыкальной шкалы вводится понятие музыкального *интервала* (октава, квинта, терция и др.) между двумя звуками, который определяется отношением частот этих звуков. Например, октава — интервал между данным звуком и звуком двойной частоты. Минимальным интервалом является интервал между двумя соседними нотами, который называется *полутон* (рис. 2).

Важным музыкальным понятием с точки зрения математической интерпретации музыки является также понятие *класса высоты* как одной и той же ноты в разных октавах. Построение 12–ступенчатой музыкальной лестницы имеет математическое обоснование, в котором центральным аспектом выступает зависимость между частотами нот. Оказывается, что отношение частот большей (f_1) и меньшей (f_2) соседних нот подчиняется правилу:



Рис. 2: 12–ступенчатая шкала на первых двух октавах

$$\frac{f_1}{f_2} = \sqrt[12]{2} \quad (2)$$

Таким образом, существует зависимость между частотами нот, что позволяет рассчитать частоту любой ноты, зная, например, частоту ноты *ля* первой октавы равную 440 Гц.

Созвучие трёх и более нот называется *аккордом*. В европейской музыке наиболее распространены аккорды из трёх нот — *трезвучия* [5]. Название аккорда соответствует его первой ноте. Так, например, аккорд *C:maj*, называемый *До-мажор*, представлен созвучием из трёх нот: *до*, *ми* и *соль*.

Для каждого аккорда можно определить соответствующий ему *настоящий* вектор, представляющий собой 12–мерный вектор V_z , состоящий из последовательности 0 и 1, где

$$V_z[i] = \begin{cases} 1, & \text{if } i^{\text{th}} \text{ note of 12-note scale} \in \text{chord} \\ 0, & \text{otherwise} \end{cases}$$

Так, например, аккорду *C:maj*, состоящему из нот *до*, *ми* и *соль*, можно сопоставить следующий *настоящий* вектор:

$$V_{C:maj} = (1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0) \quad (3)$$

4. Подходы к решению задачи

4.1. Машинное обучение

В общем виде задачу машинного обучения можно представить следующим образом: для некоторой обучающей выборки $Train$, состоящей из множества объектов X_{train} и вычисленных значений функции признака Y_{train} (т.е. описаний объектов), необходимо построить алгоритм, выявляющий общие закономерности из набора частных эмпирических данных (в нашем случае - набора музыкальных композиций). Обученный на наборе $Train$ алгоритм призван анализировать тестовую выборку $Test$, состоящую из набора объектов (музыкальных произведений) X_{test} , с определенной степенью точности распознавания аккордов.

Первый этап, на котором происходит формирование обучающей выборки, называется задачей *извлечения признаков*, а второй этап - *обучением алгоритма*.

4.2. Извлечение признаков

Задача извлечения признаков для выбранной коллекции требует представления музыкальных произведений в виде оцифрованного сигнала и построения функции сопоставления объектов и признаков.

Оцифровка (дискретизация) сигнала подразумевает его представление в виде массива чисел. Процесс дискретизации состоит в определении значения амплитуды сигнала через определенные промежутки времени.

Следующий этап решения настоящей задачи состоит в построении функции-признака. В этой связи алгоритмы машинного обучения, применяемые в настоящей работе, используют введенный выше параметр двенадцатимерного вектора — *хромовектора* аккорда, в котором каждое из двенадцати значений соответствует одной из нот. Указанное соответствие вытекает из алгоритма построения хромовекторов, который условно можно разделить на несколько этапов:

1. Построение *спектрограммы*, т.е. зависимости амплитуды сигнала от времени для некоторого звукового интервала;
2. Вычисление энергии спектра для каждой ноты;
3. Сложение значений энергий спектра в каждом классе высоты;
4. Повтор суммирования значений энергий спектра в каждом классе высоты для каждого окна;

Для построения спектрограммы можно использовать, например, оконное преобразование Фурье для каждого окна. Результатом оконного преобразования Фурье для ограниченного звукового интервала является спектр произведения сигнала и оконной функции. Полученный таким образом спектр является оценкой спектра исходного сигнала и принципиально допускает искажения. Энергия спектра для каждого звука вычисляется в предположении, что каждая нота представляет собой наиболее высокий звук (локальные максимумы на спектрограмме) и некоторую окрестность вокруг него. Складывая значения энергий спектра в каждом классе высоты, можно получить двенадцатимерный хромовектор. Повторив суммирование значений энергий спектра в каждом классе высоты для каждого окна, можно сформировать последовательность хромовекторов, которая будет описывать исходные данные.

Вычисление функции-признака в рамках настоящей задачи распознавания аккордов интерпретируется нами как нахождение для каждого из полученных хромовекторов соответствующего аккорда.

4.3. Алгоритмы

В настоящей работе были применены следующие основные алгоритмы машинного обучения: метод ближайших соседей, метод опорных векторов и наивный байесовский классификатор. Для решения поставленной задачи использовалась библиотека Scikit-learning [8].

4.3.1. Наивный Байесовский классификатор

В основе *Наивного Байесовского Классификатора (Naive Bayesian Classifier)* лежит вероятностная теорема Байеса, а наивность обеспечивается за счет строгих предположений о независимости. В общем виде теорема Байеса формулируется следующим образом:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}, \quad (4)$$

где

$P(A)$ - априорная вероятность гипотезы А;

$P(A|B)$ - вероятность гипотеза А, при наступлении события В;

$P(B|A)$ - вероятность наступления события В при истинности гипотезы А;

$P(B)$ - полная вероятность наступления события В;

Формула (4) применительно к настоящей задаче имеет следующий вид:

$$P(A_z|B_{V_z}) = \frac{P(A_z)P(B_{V_z}|A_z)}{P(B_{V_z})}, \quad (5)$$

где событие A_z - "наступление" аккорда, событие B_{V_z} - "наступление" хромовектора.

Далее, рассматривая $V_z = (v_1, \dots, v_{12})$ и учитывая тот факт, что v_i условно независимы, можно сделать вывод:

$$P(B_{V_z}|A_z) = P(B_{v_1}|A_z) * \dots * P(B_{v_{12}}|A_z), \quad (6)$$

Заметим, что в (5) знаменатель дроби является константой, а остальные составляющие вычисляются с учётом формулы (6).

К недостатком данной модели можно отнести низкую точность классификации при большом количестве классов, что существенно скажется на результатах при попытке классификации на всех классах для используемой нами коллекции.

4.3.2. Метод ближайших соседей

Алгоритм классификации по *Методу Ближайших Соседей* (*Nearest Neighbors Algorithm*) основан на понятии расстояния между объектами выборки. Рассмотрим пространство объектов X и конечное множество классов Y . Задав на множестве X функцию расстояния $\rho : X \times X \rightarrow [0, \infty)$ можно классифицировать объекты по отношению к такому классу из Y , к которому принадлежат k -ближайших объектов обучающей выборки [11].

Таким образом, определим основные этапы работы алгоритма:

1. Рассмотрим объект, классифицируемый на данной итерации;
2. Выделим k объектов обучающей выборки, значение функции расстояния до которых от выбранного объекта минимально;
3. Отнесём классифицируемый объект к тому классу, к которому относится большая часть объектов k -выборки;

В качестве функции расстояния могут выступать различные метрики, например, евклидово расстояние или манхэттенское расстояние. В настоящей работе в качестве функции расстояния было выбрано расстояние Минковского.

К недостаткам данного алгоритма можно отнести низкую скорость работы, так как её величина пропорциональна $O(nk)$ для каждого примера, где n — количество объектов обучающей выборки, k — размерность пространства X .

4.3.3. Метод опорных векторов

Основная идея алгоритма классификации по *Методу Опорных Векторов* (*Support Vector Machine*) состоит в построении классифицирующей функции, которая в зависимости от разделяющей гиперплоскости относит объект к тому или иному классу. Если исходные данные не являются линейно разделимыми, построение разделяющей гиперплоскости невозможно. В этом случае рассматриваемый метод предлагает ис-

пользовать вспомогательное отображение, переводящее исходное пространство обучающей выборки в линейно разделимое пространство [6]. Важным аспектом данного метода является понятие ядра классификатора, так как выбор оптимальной функции-ядра может существенно влиять на результат работы алгоритма. Аппроксимирующая функция может быть линейной, полиномиальной, сигмоидной или иметь иную зависимость. В настоящей работе в качестве ядра выбрана Гауссова радиальная базисная функция (Gaussian Radial Basis Function, GRBF).

5. Эксперименты

Целью экспериментов было обучение классификаторов на различных вариациях исходной коллекции, а также сравнительный анализ результатов работы алгоритмов по различным оценкам точности.

5.1. Данные

В настоящей работе была использована размеченная коллекция музыкальных произведений, предоставленная The McGill Billboard Project [7]. Эта коллекция носит название *Hot100* и представляет собой песни самых популярных групп и певцов США в трёх периодах: 1958-1969г, 1970-1979г, 1980-1991г. Выбор временных промежутков авторами статьи обусловлен, по их мнению, соответствующими периодами зарождения различных музыкальных стилей, а популярность хитов базируется на статистике радиостанций и продаж цифрового контента.

5.1.1. Описание входных данных

Входные данные представляют собой размеченные аккорды, а также вычисленные значения признаков для каждой музыкальной композиции. Приведенные в последующем тексте примеры взяты из коллекции Billboard для песни *James Brown - I Don't mind* [7].

Разметка аккордов. Разметка аккордов каждой музыкальной композиции содержит набор последовательных временных интервалов и аккордов, звучащих на соответствующих интервалах. Каждая мелодия включает приблизительно 150 – 250 временных интервалов.

start-time	end-time	chord
0.0	1.51356009	N
1.51356009	1.80157823	N
1.80157823	3.52968707	A:min
3.52968707	5.25779591	A:min
...		

Хромовектора. Для каждой музыкальной композиции построен набор хромовекторов. В настоящей коллекции каждый хромовектор имеет две составляющие: хромовектор аккорда (12-мерный, соответствующий 12 нотам) и 12-мерный хромовектор баса. Таким образом, результирующий хромовектор имеет размерность 24. В качестве признака в работе мы использовали и хромовектор аккорда и хромовектор баса. Исходный файл представляет собой набор временных меток и соответствующий им хромовектор. Каждая мелодия содержит порядка 3500 хромовекторов.

timestamp	v1	v2		v12
0.00000000	0.198482	0	...	0.864485
0.04643990	0.310882	0	...	0.904673
	...			

5.1.2. Обработка коллекции

Первый этап настоящей работы состоял в приведении коллекции к необходимому формату, а именно, в сопоставлении каждому аккорду соответствующего хромовектора. Можно отметить, что чем больше исходная коллекция данных, тем сложнее найти опечатки или несоответствия формату, изложенному авторами [7]. Поэтому исходные данные требуют детального изучения и написания программы для приведения к нужному виду, что и было реализовано в настоящей работе.

Второй этап состоял в формировании нескольких вариаций исходной коллекции.

В первом случае был взят хромовектор, соответствующий медиане временного интервала, на котором звучит аккорд. Этот метод позволяет сократить количество хромовекторов, увеличивая таким образом скорость обучения алгоритмов. Он был предложен авторами [5] и реализован в настоящей работе.

Во втором случае формирование набора хромовекторов для каждой мелодии осуществлялось посредством взятия хромовектора с определенной частотой (каждый десятый).

Подводя итоги формирования исходной коллекции, можно выделить несколько выборок для обучения классификаторов:

1. Коллекция, построенная на медианах временных интервалов аккорда;
2. Коллекция, построенная взятием хромовектора через определенный период (каждый десятый);

Все вышеизложенные методы формирования коллекции из исходной относятся только к той части данных, которая использовалась для обучения алгоритма. Выборка коллекции, на которой осуществлялось тестирование, должна представлять собой полный набор хромовекторов. Последний аспект очень важен, потому что в реальной ситуации при тестировании отсутствуют интервалы звучания аккордов.

5.2. Методы оценки точности алгоритмов машинного обучения

5.2.1. Кросс-валидация по музыкальным композициям

Кросс-валидация или *перекрёстная проверка* - это способ оценивания эффективности алгоритмов машинного обучения. Суть данного метода состоит в разбиении исходной коллекции данных на N частей. Затем производится обучение алгоритма на $N - 1$ части, а тестирование - на одной оставшейся *контрольной* части. Подобная операция повторяется для каждой из N частей таким образом, чтобы на каждой итерации в качестве тестируемого множества выступала ещё нерассмотренная для этой цели одна из N частей коллекции. В качестве результата выполнения N итераций будет получен набор из N оценок точности, а результатом работы всего алгоритма будет являться среднее арифметическое значение полученных N оценок точности.

В настоящей работе был реализован метод кросс-валидации по мелодиям исходной коллекции для случая $N = 10$, где каждая из N частей состояла из 89 песен. Оценка качества работы алгоритма машинного

обучения на каждой итерации перекрёстной проверки для $Test$ вычислялась как среднее между оценками точности для каждой из песен X_{test} .

5.2.2. Accuracy

Качество работы алгоритма для каждой из песен контрольной выборки в настоящей работе оценивалось посредством введения относительного параметра *Accuracy* как доли хромовекторов, на которых классификатор принял правильное решение:

$$Accuracy = \frac{CorrectChromas}{AllChromas}, \quad (7)$$

где

CorrectChromas - количество хромовекторов одной песни, для которых классификатор правильно предсказал аккорд;

AllChromas - количество всех хромовекторов одной песни.

Оценка точности *Accuracy* проводилась для каждой музыкальной композиции X_{test} .

5.2.3. Chord Symbol Recall

Chord Symbol Recall (CSR) - метод оценки точности алгоритмов машинного обучения для задачи распознавания аккордов, который используется на соревнованиях по музыкальному информационному поиску MIREX (Music Information Retrieval Evaluation eXchange). Для оценки точности работы алгоритмов метод CSR предполагает вычисление параметра, определяемого как доля хромовекторов аккорда, определенных правильно, от общего числа хромовекторов, соответствующих данному аккорду.

Существуют несколько вариаций метода CSR, которые наиболее часто используются в настоящее время. К ним относятся *frame-based CSR* и *segment-based CSR*.

Оценка *frame-based CSR* основана на понятии фрейма, рамки. Основная идея алгоритма состоит в разделении коллекции хромовекторов

на части (фреймы) через равные промежутки времени, например, через 10 мс. Недостатком модели является фиксированный размер фрейма. С.Нарт в работе [5] предложил решение этой проблемы с помощью метода segment-based, суть которого заключается в следующем.

Введём понятие *функции соответствия аккордов* (chord matching function), которая определяется как

$$M(X, Y) = \begin{cases} 1, & \text{if } X = Y \\ 0, & \text{otherwise} \end{cases}$$

где X, Y - сравниваемые аккорды.

Пусть E — последовательность аккордов, предсказанная алгоритмом, а A — последовательность аккордов аннотированной коллекции, то есть последовательность правильных аккордов для хромовекторов коллекции. Разобьём последовательность E по границам интервалов звучания аккордов из A на сегменты S^j , где $S^j = S_E^j \cap S_A^j$, и сопоставим аккорд в каждом из получившихся сегментов с аккордом, звучащим на этом интервале в A с помощью функции соответствия аккордов M . Тогда, просуммировав значения по каждому из сегментов, получим количество правильно определенных хромовекторов (рис. 3).

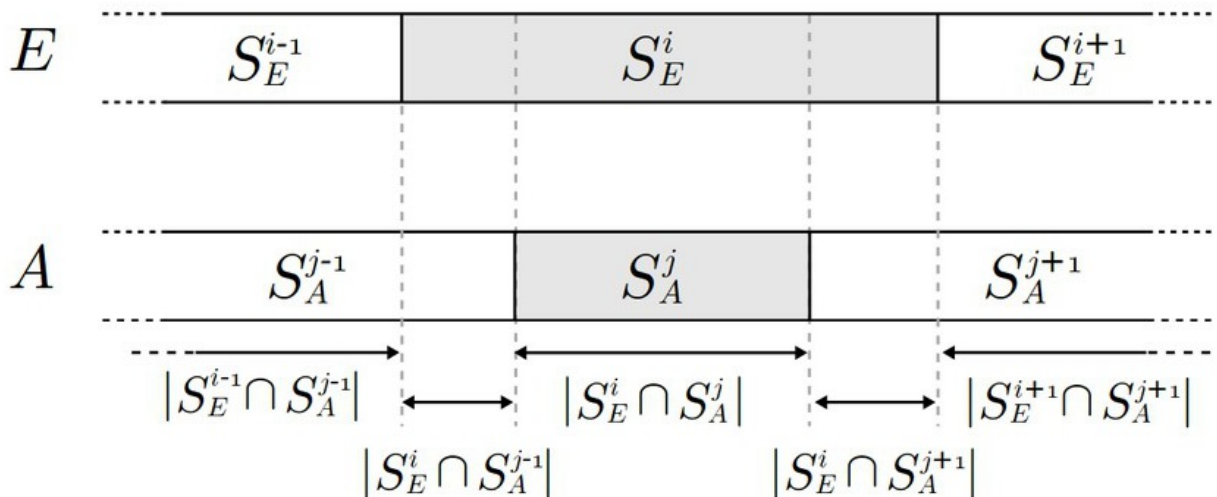


Рис. 3: E — предсказанная алгоритмом последовательность аккордов; A — последовательность аккордов аннотированной коллекции в рамках оценки CSR

Алгоритм подсчёта segment-based CSR может быть представлен в

виде формулы:

$$R(S_A, S_E) = \frac{\sum_{S_A^j} \sum_{S_E^i} |S_E^i \cap S_A^j| \cdot M(S_E^i, S_A^j)}{\sum_{S_A^j} |S_A^j|} \quad (8)$$

В рамках нашей задачи был реализован алгоритм вычисления segment-based CSR.

5.3. Применение алгоритмов машинного обучения для музыкальной коллекции

5.3.1. Результаты метода кросс-валидации по музыкальным композициям

В настоящей работе обучающая часть исходной коллекции классифицировалась как по всем аккордам, представленным в выбранной коллекции песен, так и с учетом использования определенных классов аккордов. Выбранные классы аккордов имеют наибольшую частоту встречаемости [5]. Для обучения были отобраны следующие группы аккордов GR: *maj*, *min*, *maj7*, *min7*, *7*. Полученные по методу кросс-валидации результаты представлены в таблицах 1 и 2, где

- Коллекция 1 — это обучающая выборка, построенная на медианах временных интервалов аккорда на группе аккордов GR;
- Коллекция 2 — это обучающая выборка, построенная взятием хромовектора с определенной частотой (каждый 10) на группе аккордов GR;
- Коллекция 3 — это обучающая выборка, построенная на медианах временных интервалов аккорда на всех аккордах;
- Коллекция 4 — это обучающая выборка, построенная взятием хромовектора с определенной частотой (каждый 10) на всех аккордах;

Алгоритм	Коллекция 1		Коллекция 2	
	\bar{x}	s	\bar{x}	s
Байесовский классификатор	0.341	0.180	0.355	0.186
Метод ближайших соседей	0.351	0.192	0.377	0.200
Метод опорных векторов	0.398	0.225	—	—

Таблица 1: Оценочные параметры работы алгоритмов машинного обучения на коллекциях, построенных на группе аккордов GR (метод кросс-валидации)

Алгоритм	Коллекция 3		Коллекция 4	
	\bar{x}	s	\bar{x}	s
Байесовский классификатор	0.113	0.068	0.135	0.083
Метод ближайших соседей	0.319	0.172	0.365	0.192
Метод опорных векторов	—	—	—	—

Таблица 2: Оценочные параметры работы алгоритмов машинного обучения на коллекциях, построенных на всех аккордах (метод кросс-валидации)

Отсутствие отдельных результатов в таблицах 1 и 2 (прочерки) связано с длительным временем работы программы, превышающим несколько суток.

Из таблиц 1 и 2 видно, что для каждого из примененных методов среднее значение \bar{x} имеет достаточно большой разброс s , что приводит к необходимости исследования не только средних значений параметра \bar{x} по всем мелодиям, но и анализа стандартного отклонения s по гистограммам, представленным на рис. 4, 5, 6. Анализ результатов для *коллекции 1* (см. таблицу 1) позволяет предположить, что *метод опорных векторов* в отличие от других методов показывает лучшие результаты определения среднего значения \bar{x} , а также позволяет определять некоторые песни с точностью порядка 90%. Однако этот метод показывает результат распознавания с точностью менее 10% на большем количестве песен по сравнению с остальными методами.

По данным таблиц 1 и 2 можно отметить, что обучение на *коллекциях 2* и *4*, сформированных взятием каждого 10 хромовектора, показы-

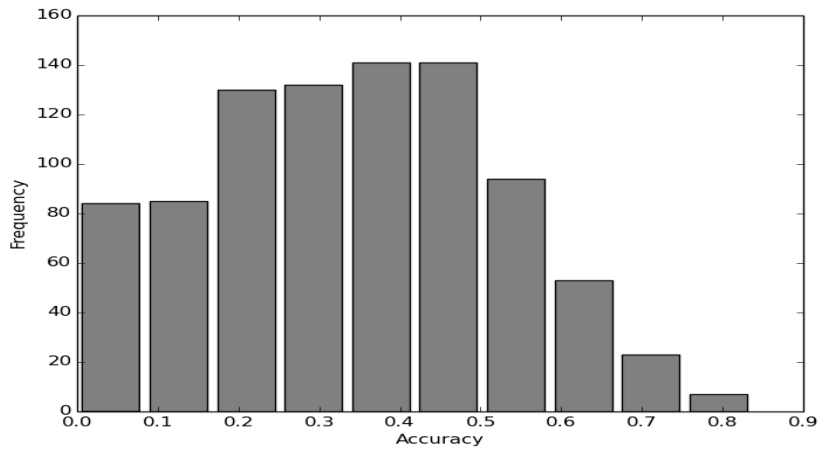


Рис. 4: Распределение количества идентифицированных с помощью метода наивного байесовского классификатора песен по параметру *Accuracy* для коллекции 1.

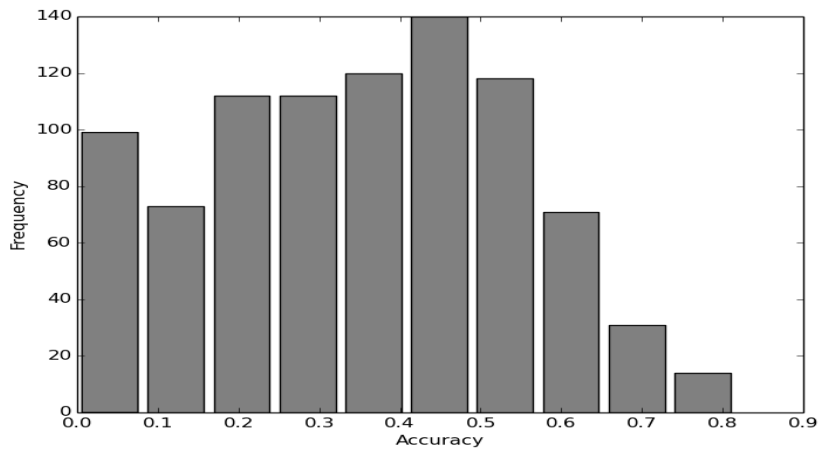


Рис. 5: Распределение количества идентифицированных с помощью метода ближайших соседей песен по параметру *Accuracy* для коллекции 1.

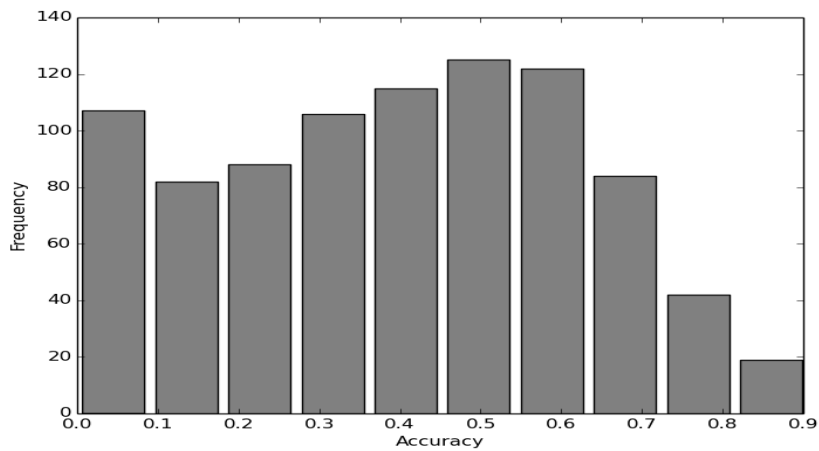


Рис. 6: Распределение количества идентифицированных с помощью метода опорных векторов песен по параметру *Accuracy* для коллекции 1.

вает лучший результат оценки параметра \bar{x} , чем на *коллекциях 1 и 3*, построенных взятием медианы. Это, по-видимому, объясняется использованием большего набора составляющих *коллекцию 2* хромовекторов (примерно 89000) по сравнению с *коллекцией 1* (порядка 8700 хромовекторов). Таким образом, с увеличением количества хромовекторов (в рассмотренных числовых пределах) точность работы алгоритмов возрастает, при этом время обучения увеличивается незначительно.

Результаты применения алгоритмов *наивного байесовского классификатора* и *метода ближайших соседей* для обучения на *коллекциях 1 и 2* представлены также на рис. 7 и 8 соответственно. Можно отметить, что при обучении обоих алгоритмов на *коллекции 2* количество песен, определенных с низкой степенью точности уменьшилось по сравнению с обучением на *коллекции 1*, увеличивая таким образом количество песен, в которых аккорды распознаются с высокой степенью точности для *коллекции 2*.

По результатам обучения алгоритмов на *коллекции 3* и *коллекции 4* (см. таблицу 2) следует отметить, что среднее значение параметра \bar{x} при работе *байесовского классификатора* значительно ниже аналогичных оценок для *коллекции 1* и *2*. Это, вероятно, является следствием "чувствительности" данного метода к количеству классов при классификации (раздел 4.3.1). На гистограммах, построенных по результатам применения *байесовского классификатора* для *коллекции 3* и *4* (см. рис. 9) видно, что, хотя средняя точность классификации \bar{x} на *коллекции 4* больше, чем на *коллекции 3*, однако увеличение количества песен, которые определились с точностью ниже 5% для *коллекции 4*, по-видимому, свидетельствует о большей целесообразности использования *коллекции 3* для метода *байесовского классификатора*.

Метод ближайших соседей для обучения на *коллекциях 3* и *4* также показал лучший результат при использовании *коллекции 4* для обучения алгоритма. Из анализа гистограмм (рис. 10), построенных по результатам тестирования данного алгоритма, следует, что в отличие от *байесовского классификатора*, *метод ближайших соседей* позволяет уменьшить количество песен, определяемых с точностью менее 30%,

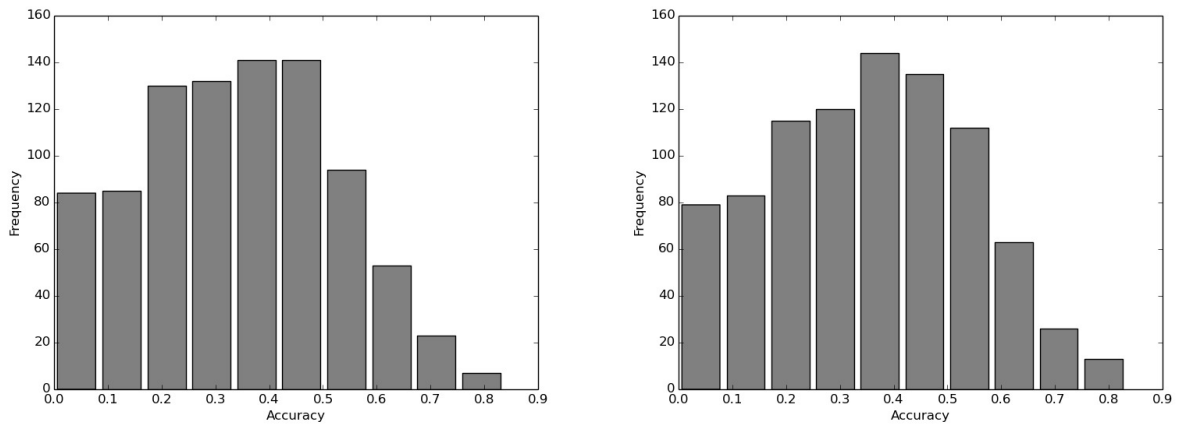


Рис. 7: Распределение количества идентифицированных с помощью метода наивного байесовского классификатора песен по параметру *Accuracy* для коллекции 1 (слева) и коллекции 2 (справа).

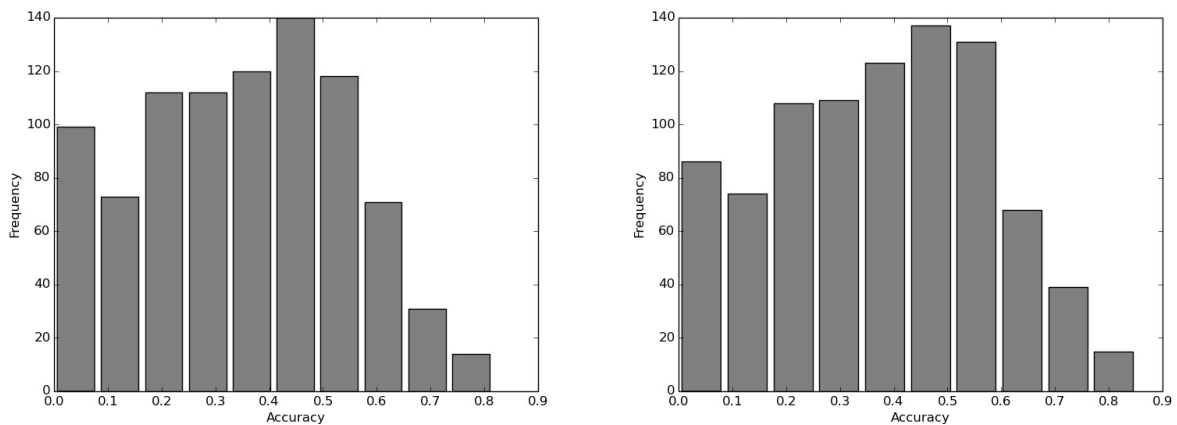


Рис. 8: Распределение количества идентифицированных с помощью метода ближайших соседей песен по параметру *Accuracy* для коллекции 1 (слева) и коллекции 2 (справа).

увеличив при этом среднее значение параметра \bar{x} , что, вероятно, позволяет сделать вывод о преимуществе этого метода по сравнению с байесовским классификатором для использованных коллекций.

5.3.2. Результаты оценки CSR

Полученные по методу оценивания CSR результаты представлены в таблицах 3, 4. Из таблиц видно, что среднее значение \bar{x} имеет аналогичную с методом кросс-валидации тенденцию увеличения точности алгоритмов при уменьшении количества классов классификации. Представленные данные показывают также высокую точность работы метода опорных векторов по сравнению с другими примененными алгоритмами для оценки CSR.

Алгоритм	Коллекция 1	Коллекция 2
Байесовский классификатор	0.313	0.343
Метод ближайших соседей	0.336	0.362
Метод опорных векторов	0.376	0.394

Таблица 3: Оценочные параметры работы алгоритмов машинного обучения на коллекциях, построенных на группе аккордов GR (метод CSR)

Алгоритм	Коллекция 3	Коллекция 4
Байесовский классификатор	0.112	0.132
Метод ближайших соседей	0.306	0.346
Метод опорных векторов	0.379	0.395

Таблица 4: Оценочные параметры работы алгоритмов машинного обучения на коллекциях, построенных на всех аккордах (метод CSR)

Можно также отметить, что оценка CSR использовалась в настоящей работе для обучения алгоритма по методу ближайших соседей при различных значениях параметра k , определяющего количество "соседей". Была посчитана оценка CSR в процессе экспериментов на коллекции 2. Из графика, приведенного на рис.11, видно, что оптимальное количество "соседей" равно 15.

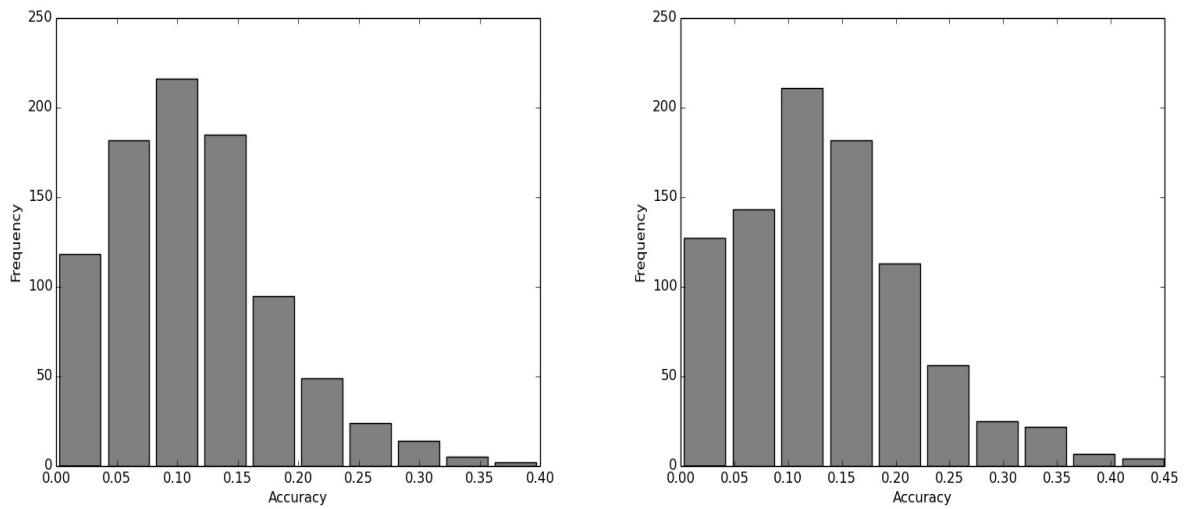


Рис. 9: Распределение количества идентифицированных с помощью метода наивного байесовского классификатора песен по параметру *Accuracy* для коллекции 3 (слева) и коллекции 4 (справа).

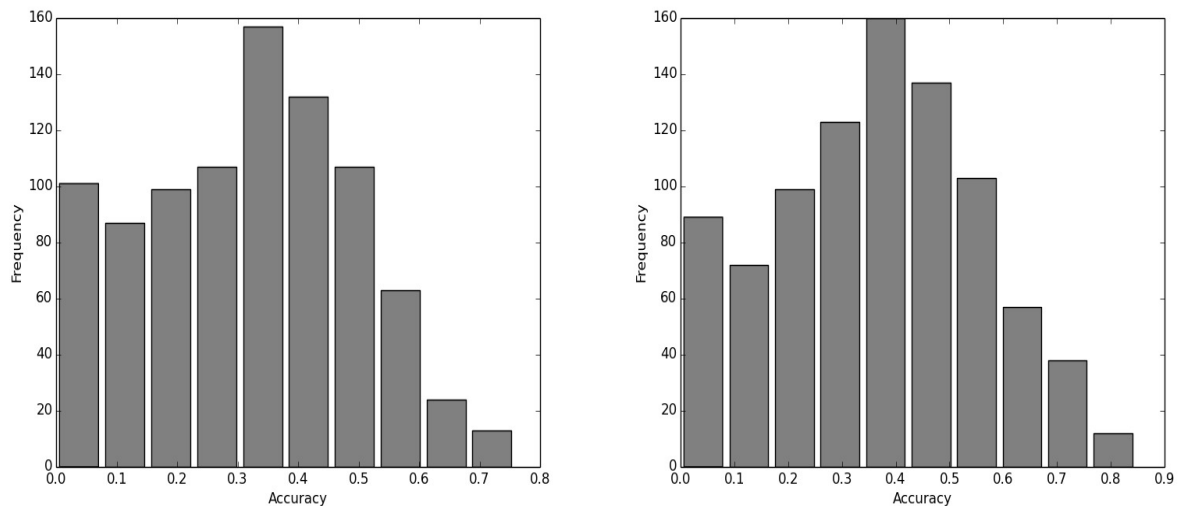


Рис. 10: Распределение количества идентифицированных с помощью метода ближайших соседей песен по параметру *Accuracy* для коллекции 3 (слева) и коллекции 4 (справа).

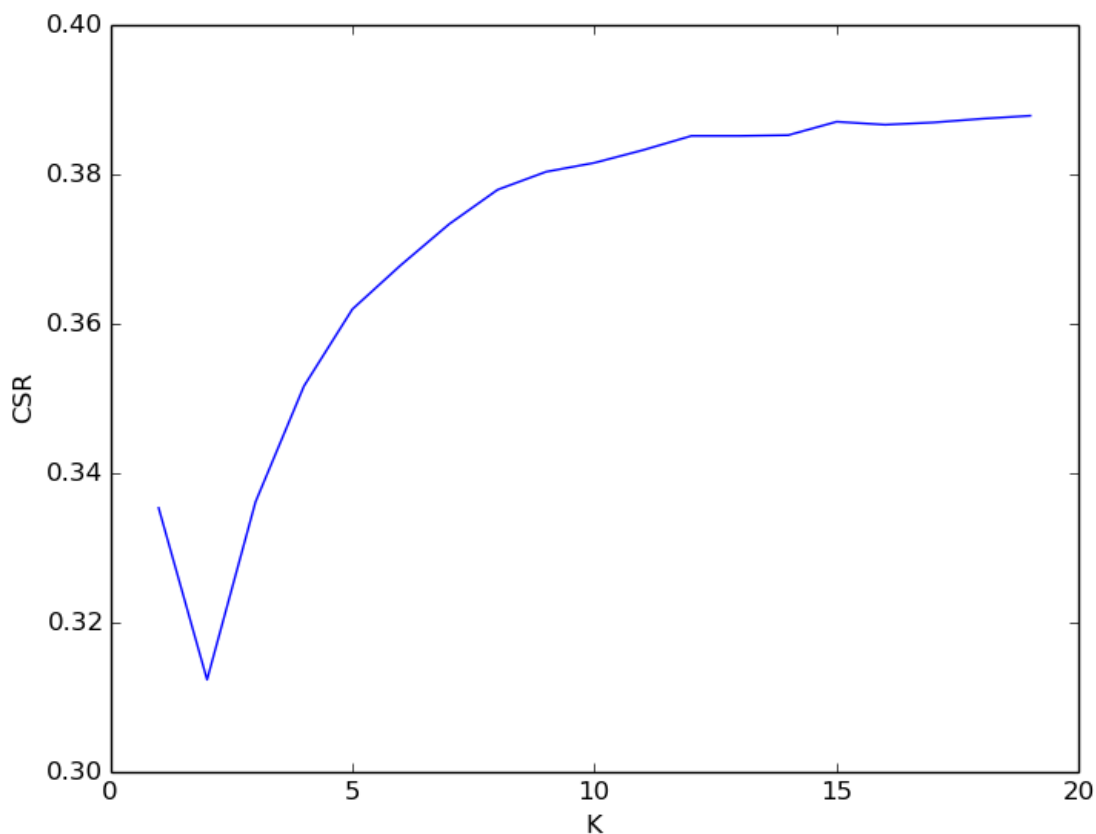


Рис. 11: Зависимость значения CSR от количества соседей (k) на коллекции 2

6. Заключение

В ходе настоящей работы были получены следующие результаты:

1. Изучена предметная область (музыкальная теория), исследованы подходы к решению настоящей задачи, и выбран метод машинного обучения;
2. Подобран и приведен к необходимому формату набор данных (размеченных мелодий);
3. Применены три основных алгоритма машинного обучения для решения настоящей задачи;
4. Реализованы две различные оценки точности работы алгоритмов;

5. Проведен сравнительный анализ результатов работы алгоритмов машинного обучения.

Список литературы

- [1] Automatic Generation of Lead Sheets from Polyphonic Music Signals. / Jan Weil, Thomas Sikora, Jean-Louis Durrieu, Gaël Richard // ISMIR. — 2009. — P. 603–608.
- [2] Boulanger-Lewandowski Nicolas, Bengio Yoshua, Vincent Pascal. Audio Chord Recognition with Recurrent Neural Networks. // ISMIR. — 2013. — P. 335–340.
- [3] Casey Michael A, Westner Alex. Separation of mixed audio sources by independent subspace analysis // Proceedings of the International Computer Music Conference. — 2000. — P. 154–161.
- [4] Fujishima Takuya. Realtime chord recognition of musical sound: A system using common lisp music // Proc. ICMC.— Vol. 1999.— 1999. — P. 464–467.
- [5] Harte Christopher. Towards automatic extraction of harmony information from music signals : Ph.D. thesis / Christopher Harte ; Department of Electronic Engineering, Queen Mary, University of London. — 2010.
- [6] Machine Learning. — URL: <http://www.machinelearning.ru> (дата обращения: 05.05.2016).
- [7] The McGill Billboard Project. — URL: <https://ddmal.music.mcgill.ca/billboard> (online; accessed: 17.12.2015).
- [8] Scikit-learn. — URL: <http://scikit-learn.org/stable> (online; accessed: 10.04.2016).
- [9] Sheh Alexander, Ellis Daniel PW. Chord segmentation and recognition using EM-trained hidden Markov models // ISMIR 2003. — 2003. — P. 185–191.

- [10] Zhou Xinquan, Lerch Alexander. CHORD DETECTION USING DEEP LEARNING // Proceedings of the 16th ISMIR Conference. — Vol. 53. — 2015.
- [11] Воронцов КВ. Лекции по метрическим алгоритмам классификации // КВ Воронцов. — 2007.
- [12] Корж РА. Криворожский национальный университет // ИНЖЕНЕРИЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕНИЯ. — 2011. — Р. 48.
- [13] Шилов ГЕ. Простая гамма (устройство музыкальной шкалы) // Шилов ГЕ—М.: Физматгиз. — 1963.