

И.А. БАЛТИЙСКИЙ, С.И. НИКОЛЕНКО
**ГРАФИЧЕСКАЯ ВЕРОЯТНОСТНАЯ МОДЕЛЬ
ДЛЯ ОЦЕНКИ СХОЖЕСТИ ГАРМОНИЙ
МУЗЫКАЛЬНЫХ ПРОИЗВЕДЕНИЙ**

Балтийский И.А., Николенко С.И. Графическая вероятностная модель для оценки схожести гармонии музыкальных произведений.

Аннотация. Методы оценки схожести музыкальных произведений позволяют реализовать полностью автоматическую рекомендательную систему для музыки, ориентированную на содержание (наподобие Pandora, но без ручного труда экспертов-музыковедов). В статье предлагается новый метод оценки схожести гармонии композиций на основе оригинальной графической вероятностной модели. Модель включает в себя информацию об аккорде и ладе для некоторого момента времени звучания композиции; мы вводим скрытую переменную, стиль, от которой зависит вероятность использования определенного аккорда в контексте определенного лада, и предлагаем сравнивать композиции как векторы-параметры распределений для стилей. По аналогии с некоторыми методами извлечения аккордов рассматриваемая модель не включает ни ритмическую информацию, ни зависимости между соседними аккордами. Описывается реализация модели в системе Infer.NET и осуществляется проверка модели на искусственных данных. Результат работы на реальных данных отрицателен, что свидетельствует о том, что простые модели не подходят для задачи оценки схожести.

Ключевые слова: информационный поиск музыки, рекомендательные системы, гармония, оценка схожести, графические модели, вероятностные модели.

Baltiyskiy I.A., Nikolenko S.I. A probabilistic graphical model for the music harmony similarity task.

Abstract. Methods for measuring music similarity allow for implementations of completely automated content-based music recommendation systems (similar to Pandora, but without the manual work of expert musicologists). This paper presents a novel method of measuring music harmony similarity based on an original probabilistic graphical model. The model includes information about the current chord and mode; we introduce a hidden parameter, style, which governs the probability of using of a certain chord within the context of a certain mode, and propose to measure the similarity as a distance between parameter vectors of the probability distribution function for style. Similar to some methods for extracting chord progressions, our model includes neither the rhythmic information nor the dependencies between neighboring chords. We describe the implementation of our model done with the Infer.NET system and show experimental results on generated data. The results of experiments with real-world data are negative, which indicates that simple bag-of-chords models are not suitable for the music similarity task.

Keywords: music information retrieval, recommendation systems, harmony, similarity task, graphical models, probabilistic models.

1. Введение. Задача автоматического анализа музыкальных произведений интересна тем, что на её основе можно построить полностью автоматическую рекомендательную систему музыкальных ком-

позиций, ориентированную на содержание [5, 30]; имеющиеся системы, такие, как Pandora [6, 14, 19], используют ручной труд экспертов-музыковедов. Задача допускает разбиение на части, соответствующие различным граням музыки [1]; нас будет интересовать подзадача, связанная с анализом схожести гармонии музыкальных произведений.

В настоящей работе мы исследуем один из возможных способов моделирования гармонической информации, с помощью обучения модели, в которой данные о гармонии представляются в виде множества аккордов без учёта порядка их следования (bag of chords). Для моделирования и обучения параметров модели мы используем формализм графических вероятностных моделей [4, 22–23, 35–39]. Способ моделирования подробно описан в разделе 2, а раздел 3 посвящён практической реализации и результатам экспериментов.

2. Графическая вероятностная модель для сравнения гармонии.

2.1. Используемые понятия и алгоритмы. Введем основные понятия, которыми будем пользоваться в дальнейшем.

2.1.1. Хрома-вектор. Здесь и далее будем рассматривать музыкальные произведения, представленные в виде цифрового звукового сигнала. Нам потребуется преобразовывать композиции в последовательность *хрома-векторов*, — представление звукового сигнала, которое мы сейчас опишем.

Воспринимаемые человеческим слухом звуки могут обладать особым свойством, *высотой*, которая главным образом определяется их спектром. Спектр звука, имеющего высоту, состоит из набора частот $f_i = if, i \geq 1$, где f — *основная частота* звука, f_i — *гармоники*. Восприятие высоты имеет спиралевидный характер: с точки зрения слушателя, звуки, с одной стороны, линейно упорядочены по высоте, с другой стороны, имеют дополнительную периодичную по высоте характеристику с периодом, равным октаве (интервалу между звуками, частоты которых отличаются в 2 раза) [21]. Окрестность некоторой частоты принято выделять в *ноту* (будем далее отождествлять ноту и ее частоту). К настоящему времени в европейской музыкальной традиции практически повсеместно применяется набор нот, задаваемый абсолютным значением частоты $f_0 = 440$ Гц и всеми нотами вида

$$f_k = f_0 2^{\frac{k}{12}}.$$

Другими словами, такой звукокоряд получен делением октавы на 12 одинаковых интервалов (под величиной интервала понимается отношение составляющих его частот). Поскольку воспринимаемый челове-

ческим ухом диапазон частот ограничен, в звукоряд включают не более 100 нот, начиная с 18-20 Гц. Рассмотрим некоторую ноту f_k ; ее *классом высот* назовем все ноты вида f_{k+12l} . По построению, количество различных классов высот равно 12.

Пользуясь спектральным разложением сигнала в коротком временном окне, можно определить значение энергии спектра вокруг нот звукоряда; полученный набор значений приближенно (с точностью до гармоник) описывает звучащие в данный момент времени ноты. Если сложить все значения энергий для нот из одного класса эквивалентности, получим 12-мерный вектор — *хрома-вектор*. Будем считать, что вектор нормализован в L^∞ , т.е. что максимальный его элемент равен 1.

Ясно, что, покрыв такими окнами весь имеющийся сигнал, можно перевести его в набор хрома-векторов. Можно «привязать» положение окон к событиям в композиции, таким как «начало ноты» или «конец ноты» (получение такой привязки представляет собой отдельную задачу анализа музыкальной информации — задачу *сегментирования*). Мы будем предполагать, что сегментирование уже произведено.

В настоящей работе не будут затронуты вопросы вычисления хрома-векторов и сегментирования; вычисление хрома-векторов подробно освещено в бакалаврской работе первого автора [32] и статье [34], о сегментировании можно прочесть в [24].

Заметим, что из периодичного характера принадлежности нот классам высот следует, что все рассуждения о порядке классов высот (а значит, и компонент хрома-вектора) должны проводиться по модулю 12. В дальнейшем мы не будем подчеркивать это; так, выражение «следующая компонента хрома-вектора» в случае последней, двенадцатой компоненты подразумевает первую компоненту, а «предыдущая» в случае первой — последнюю.

Компоненты хрома-вектора будем называть *полутонами*.

2.1.2. Лад и аккорд. Кроме отдельных нот нас также будут интересовать такие понятия музыкальной гармонии как *лад* и *аккорд*. Согласно определению, данному в [31], лад — это система устойчивых и неустойчивых звуков, объединенных тяготением к центральному звуку или созвучию — *тоники*. Под аккордом понимается сочетание из трех и более звуков, классы высот которых, как правило, разделены терциями [31]. Аккорд, как правило, воспринимается в контексте лада; в классической музыке различают устойчивые и неустойчивые аккорды в зависимости от того, какой ступенью лада является основной тон аккорда.

В первом приближении можно было бы сказать, что лад является своего рода ограничением на используемые классы высот, — например, в случае диатонического лада вместо двенадцати возможных классов используются семь. На самом деле, зачастую используются ноты, не принадлежащие ладу (альтерированные); их отличает то, что вследствие своей неустойчивости они встречаются реже ладовых нот. Забегая вперед, отметим, что если в данный момент в композиции имеется некоторый лад, можно говорить о том, что появление классов высот, принадлежащих ладу, *вероятнее* появления других классов высот. Из этого следует, что по текущему хрома-вектору можно сделать вывод о том, с какой вероятностью в данный момент времени некоторый лад используется в композиции.

Разберемся теперь с понятием аккорда. В применении к гармоническому анализу под аккордом понимается *подразумеваемый* в данный момент времени аккорд; это не значит, что должны звучать все его ноты, однако это также накладывает определенные ограничения на набор звучащих классов высот (иными словами, аккорд влияет на гармонизацию [31]). Выразить эти ограничения в явном виде сложнее; для нас важно само наличие такого ограничения.

Разумеется, и лад, и аккорд можно представить в виде хрома-векторов; поскольку мы хотим подчеркнуть именно набор составляющих их классов высот, компоненты, принадлежащие аккорду или ладу, будут равны 1, а остальные — 0. Далее опишем используемые в нашей работе аккорды и лады.

Из аккордов мы будем рассматривать только большое и малое трезвучия. Большое трезвучие (также *мажорное* трезвучие) представляет собой основной тон, терцовый тон, расположенный через 3 полутона (*большая терция*, отсюда название аккорда), и квинтовый тон, расположенный через 2 полутона от терцового. Соответствующие хрома-векторы изображены на рис. 1 и рис. 2. На рисунках компоненты, равные 1, закрашены черным, равные 0 — белым. Сразу оговоримся, что в силу энгармоничности и отсутствия тонального контекста компоненты хрома-вектора будем обозначать с использованием только диеза. В обозначениях будем придерживаться более, на наш взгляд, логичной в контексте хрома-вектора итало-французской системы, в которой нота *до* обозначается как *C*, а *си* — как *B*.

Минорный аккорд отличается тем, что терцовый тон отстоит на малую терцию от основного (см. рис. 3).

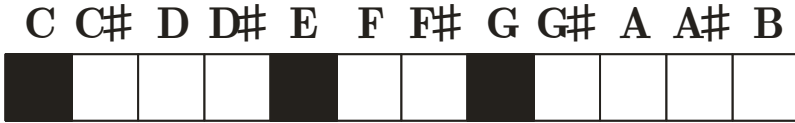


Рис. 1. До-мажорный аккорд.



Рис. 2. Си-мажорный аккорд.

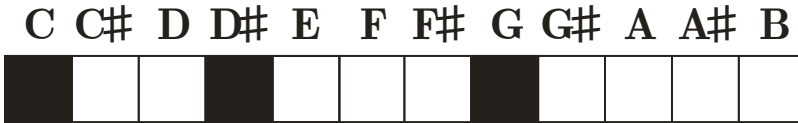


Рис. 3. До-минорный аккорд.



Рис. 4. Ионийский лад от ноты ля-диез,
или дорийский лад от ноты до.

Будем рассматривать только диатонические лады. Заметим, что, наблюдая в некоторый момент времени набор ступеней лада, невозможно сделать вывод о его основной ступени без знания о предыдущих моментах времени; поэтому вопрос первой ступени лада мы опустим и будем различать, например, только ионийский лад и говорить о *сдвиге* лада. Так, хрома-вектор для дорийского лада от ноты *до* идентичен хрома-вектору ионийского лада от ноты *ля-диез* (см. рис. 4).

2.2. Построение графической модели. В этом разделе мы построим модель, описывающую закономерности использования музыкальной гармонии в отдельной композиции. В конце главы будет приведен фактор-граф (граф функций и переменных) модели. Как отмечалось в разделе 1.1, мы будем считать, что музыкальная композиция представлена в виде последовательности хрома-векторов. Опишем генеративный процесс, результатом которого будет являться эта последовательность.

Назовем элемент последовательности хрома-векторов *сегментом*. Ключевое предположение модели заключается в том, что текущий сегмент зависит только от текущего лада и текущего (подразумеваем)

аккорда. Следующее ключевое предположение — текущий аккорд зависит от текущего лада, причем характер этой зависимости и является той особенностью использования гармонии, по которой будут сравниваться композиции. Для выражения этой зависимости введем дополнительную дискретную переменную, которую будем называть *стиль*; итак, текущий аккорд будет зависеть не только от лада, но и от стиля. Такая модель рассматривает каждый сегмент по отдельности подобно модели на основе марковской цепи нулевого порядка в работе [25].

Полученная графическая модель изображена на рис. 5. Случайную величину, характеризующую стиль, будем обозначать буквой s , аккорд — c (от англ. chord), лад — m (от англ. mode). Наблюдаемый сегмент будем обозначать g (по второй согласной букве слова segment). Лад m принимает целочисленные значения в диапазоне от 0 до 11 (будем писать $0..11$), аккорд — в диапазоне $0..23$ (12 мажорных и 12 минорных аккордов), g принимает значения в виде 12-мерных векторов вещественных чисел из $[0, 1]$.

Зафиксируем вышеуказанные зависимости в виде формул. Будем индексировать сегменты переменной k , элементы сегментов — переменной l , лады — i , аккорды — j , стили — v . Мы постулировали наличие условной зависимости между текущим аккордом от текущего лада и стиля композиции:

$$p(c = j | m = i, s = v) = c_{jiv}.$$

Соответствующие плотности распределений будут общими для всех композиций; для использования модели нам потребуется обучить параметры этих распределений.

Каждая компонента наблюдаемого сегмента генерируется текущими значениями лада и аккорда:

$$p(g_{kl} | m = i, c = j) = g_{klj}.$$

Эта зависимость устроена как *Noisy-OR* родительских вершин. Рассмотрим две случайных величины, e_0 и e_1 , которые будут означать «близко к 0» и «близко к 1» соответственно, т.е. их функции плотности вероятности должны быть определены на $[0, 1]$, иметь максимум в 0 (в 1) и убывать (возрастать) на всем отрезке. Тогда

$$g_{klj} = e_0 p_0 + e_1 (1 - p_0),$$

где p_0 имеет смысл вероятности того, что значение компоненты должно быть близко к 0:

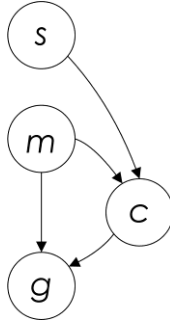


Рис. 5. Общий вид графической модели.

$$p_0 = \sum_{i=1}^{12} p(m=i) \mathfrak{I}(m_i[l]=0) \cdot \sum_{j=1}^{24} p(c=j) \mathfrak{I}(c_j[l]=0).$$

Здесь через \mathfrak{I} обозначена индикаторная функция, принимающая 1, если условие в её аргументе выполнено, и 0 в противном случае. Под m_i, c_j понимаются соответственно хрома-векторы i -го лада и j -го аккорда; l — индекс компоненты сегмента, для которой составляется это соотношение.

Как вид распределений e_0, e_1 , так и конкретные распределения остальных случайных величин обсуждаются в последующих разделах.

Рассмотрим вариацию предложенной модели. Заметим, что значения $j=0\dots 23$ содержат в себе информацию о типе аккорда и о его «сдвиге» — любой аккорд, представленный в виде хрома-вектора, можно рассматривать как циклический сдвиг аккорда с основным тоном ∂_0 на $j \bmod 12$ позиций вправо. Разделим эту переменную на две, $ct=0\dots 1$ — тип аккорда (от англ. chord type), — и $cs=0\dots 11$ — сдвиг аккорда (от англ. chord shift).

Далее, обратим внимание, что количество параметров модели можно уменьшить, что должно улучшить качество обучения. Речь идет о том, что значение лада и значение сдвига аккорда можно объединить в одно понятие, которое мы будем называть *функцией*, по аналогии с функциями аккордов в классическом гармоническом анализе, обозначаемыми римскими цифрами от I до VII. К примеру, тот факт, что аккорд соль в тональности *ре* используется в данной композиции чаще, чем аккорд *фа-диез*, в переводе на язык функций звучит так же, как тот факт, что аккорд *ля* в тональности *ми* используется чаще аккорда *си*.

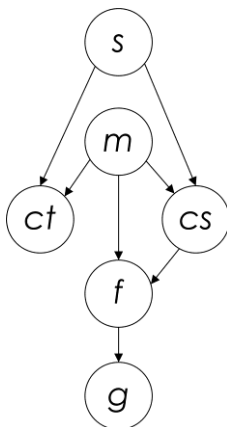


Рис. 6. Уточненная модель.

Отличие введенного понятия функции от понятия функции в гармоническом анализе заключается в том, что мы не используем знание о позиции первой ступени лада, а пользуемся лишь информацией о том, насколько аккорд сочетается с выбором нот лада.

Новая версия модели представлена на рис. 6. Случайная величина, характеризующая текущую функцию, обозначена через f .

Вместо c_{jiv} мы будем обучать

$$p(f = i | ct = t, s = v) = f_{iv}.$$

Заметим, что теперь влияние лада и аккорда на наблюдаемый сегмент можно выразить через сдвиг аккорда, тип аккорда и функцию.

В дальнейшем будем рассматривать именно такую графическую модель, включающую в себя тип аккорда ct , лад m и сдвиг аккорда cs , из которых вычисляется функция f .

Итак, в этом подразделе мы представили описание модели гармонии музыкальной композиции. Поясним теперь, как предполагается использовать описанную модель для оценки схожести музыкальных произведений.

2.3. Обучение модели. Сначала модель необходимо *обучить*. Обучение требуется для того, чтобы получить априорные распределения для f_{iv} и e_0, e_1 . Обучение производится на композициях, для которых имеется *разметка аккордов* — информация о том, какой аккорд в действительности сопоставлен каждому из сегментов. Эта информация заносится в модель в виде наблюдаемого значения c (для первого

варианта модели; про второй вариант речь пойдет ниже). Обучение производится в два этапа: *обучение с учителем* (англ. supervised learning) и *самообучение* (англ. unsupervised learning). При обучении с учителем помимо разметки аккорда задается также наблюдаемое значение стиля для s . Это необходимо для разрушения симметрии модели — до начала обучения априорное распределение для s задает одинаковые вероятности для всех исходов. При самообучении наблюдаемые значения для стиля не предоставляются; фактически, на этом этапе модель осуществляет кластеризацию.

Суть обучения заключается в следующем.

1. До начала обучения задаются априорные распределения для обучаемых параметров распределений f_{iv} и e_0, e_1 — о видах распределений речь пойдет ниже, а параметры подбираются вручную.

Для упрощения изложения будем называть параметры распределений f_{iv} и e_0, e_1 *параметрами модели*. Подчеркнем, что целью обучения является определение параметров распределений для параметров f_{iv} и e_0, e_1 , то есть для *гиперпараметров модели*.

2. Композиции из обучающего набора обрабатываются по очереди. Для очередной композиции задаются наблюдаемые значения g_k, c_k и m_k в первом варианте модели или f_k, ct_k во втором варианте; в случае обучения с учителем задается наблюдаемое значение s .

3. Производится вывод апостериорных распределений искомых параметров модели.

4. Выведенные значения гиперпараметров модели устанавливаются в качестве наблюдаемых значений случайной величины, характеризующей параметры модели. Повторяется шаг 2 для следующей композиции.

Для исключения влияния порядка обучения производится несколько проходов по всем произведениям из обучающего набора.

2.4. Использование модели. Будем считать, что сценарий использования таков: есть набор музыкальных композиций (будем называть его *фонотека*), аудиоданные которой доступны владельцу рекомендательной системы. Пользуясь моделью, мы вычисляем оценки схожести для всех пар композиций фонотеки, которые задаются рекомендательной системе.

В предложенном сценарии модель используется так.

1. Из звукового сигнала композиций фонотеки извлекаются хроматические векторы.

2. Композиции фонотеки обрабатываются по очереди. Хрома-векторы очередной композиции устанавливаются в качестве наблюдаемых значений для сегментов. Устанавливаются априорные распределения параметров модели. Производится вывод апостериорного распределения для стиля и гиперпараметров.

3. Как и на этапе обучения, апостериорное распределение параметров модели на данном шаге устанавливается в качестве априорного для следующего шага.

Как и на этапе обучения, производится несколько проходов по всем произведениям фонотеки.

После окончания вышеуказанной обработки, каждой композиции сопоставляется вектор параметров мультиномиального распределения стиля s . Можно рассматривать их как векторы в пространстве стилей; нормализовав их, расстояние между композициями можно получить как скалярное произведение соответствующих векторов: чем больше произведение, тем ближе композиции в смысле особенностей использования гармонии.

2.5. Описание случайных величин, входящих в модель.

2.5.1. Лад. До настоящего момента не указывалось, откуда берутся значения лада m в первой модели и функции f во второй. На самом деле, мы будем вычислять эти значения из текущего сегмента, получая параметры для мультиномиального распределения m (о f речь пойдет в следующих разделах); назовем вектор этих параметров pm . Значение l -й компоненты pm будем определять как величину, обратную расстоянию сегмента от l -го ионического лада (т. е. ионического лада, первой ступенью которого является l -я нота — под номером ноты понимается то же, что и в случае с компонентами хрома-вектора); после вычисления всех компонент pm , естественно, нормализуется в L^1 .

Напомним, что лад представляется в виде хрома-вектора как вектор, компоненты которого суть 0 либо 1. Сконструируем функцию расстояния между хрома-вектором и ладом. Как указывалось в главе 1, лад определяет набор наиболее возможных звучащих классов высот; появление внеладовых классов высот характерно, но менее вероятно, чем появление ладовых классов высот. Для данного сегмента мы будем уменьшать вероятность лада, для которого компоненты сегмента с большими значениями приходятся на внеладовые тона. С другой стороны, не все ладовые тона обязательно должны соответствовать большим значениям компонент сегмента; но надо увеличивать вероятность

лада, который «лучше всего» описывает наибольшие компоненты сегмента.

$$d(g, m_i) = \sum_{l=0}^{12} w(g_l, m_{i_l}),$$

$$w(b_g, b_m) = \begin{cases} \alpha_0 b_g, & b_m = 0 \\ \alpha_1 (1 - b_g), & b_m = 1 \end{cases}$$

Значения α_0, α_1 характеризуют «наказание» лада за то, что, соответственно, компонента сегмента не принадлежит ладу (чем больше значение компоненты, тем больше «наказание»), и за то, что компонента не принадлежит ладу (чем меньше значение, тем меньше наказание). Для реализации были выбраны $\alpha_0 = 1, \alpha_1 = 2$.

Приведем пример значений компоненты pm , вычисленных на реальных данных. Хрома-векторы были вычислены для первых 80 секунд записи джазового стандарта *So What* с альбома Майлза Дэвиса *Kind of Blue*. На рис. 7 и рис. 8 представлена последовательность значений компонент pm , соответствующих до-диез и ре. На иллюстрациях хорошо видно, что вычисленные значения точно схватывают тональную структуру композиции: она имеет структуру ААВА, где А сыграна в дорийском *ре*, а В — в дорийском *ре-диез*; в представлении хрома-векторами это то же, что и ионийские лады *до-диез* и *ре*. Действительно, в районе сегментов 70-122 наблюдается спад у компоненты *до-диез* и подъем у компоненты *ре*; это соответствует отрывку с 27 по 44 секунду аудиозаписи. По аудиозаписи нетрудно убедиться в том, что переход в дорийский *ре-диез* и обратно соответствует этому отрывку.

2.5.2. Аккорд. Во время обучения наблюдаемое значение аккорда берется из разметки; в первом варианте модели это целое число от 0 до 23, во втором — это тип аккорда (мажорный или минорный) и сдвиг, который влияет на получаемое значение функции (см. ниже). Во время использования модели аккорды не наблюдаются, а выводятся (производится самообучение). Самообучение для выведения текущего аккорда в нашей модели используется по аналогии с моделями, предназначенными для вычисления последовательности аккордов (см. [33]).

Не ограничивает ли нашу модель рассмотрение только мажорных и минорных аккордов? Рассмотрим более сложный вариант наподобие использованного в [25]: будем наблюдать не дискретную величину, точно описывающую текущий аккорд, а вектор вероятностей, являющийся параметром для случайной величины, описывающей текущей

аккорд. За счет этого можно рассматривать аккорд любой сложности. Покажем, как вычисляются компоненты этого вектора (обозначим его pc); отметим, что здесь мы следуем методу, изложенному в [25].

В области психоакустики проводились исследования характера восприятия слушателями различных мажорных и минорных тональностей. Испытуемые, в частности, сравнивали отношения между тональностями. В результате статистической обработки данных наблюдений были получены координаты тональностей в четырехмерном пространстве, позволяющие вычислять близость между ними как евклидово расстояние между соответствующими точками. Подробное описание методики исследований и полученных результатов приведено в книге [20].

Припишем аккордам координаты тех тональностей, тоническими трезвучиями которых они являются. Введем расстояние между произвольным аккордом, представленным в виде хрома-вектора r , компоненты которого равны 0 или 1, и «эталонным» аккордом c_j . Для начала рассмотрим вектор количества совпадений $nums$: его l -я компонента равна количеству аккордов, l -я компонента, хрома-представления которых совпадает с l -й компонентой r

$$nums[l] = |\{i : c_i[l] = r[l]\}|.$$

Тогда можно записать расстояние так:

$$d(c_j, r) = \sum_{k=1}^{24} \frac{nums[k]}{1 + d(c_i, c_j)}.$$

Мы будем учитывать возможность такого задания аккордов при практических исследованиях. Отметим, что из значения дискретной величины c можно получить значение типа и сдвига аккорда: $ct = \lfloor \frac{c}{12} \rfloor, cs = c \bmod 12$.

2.5.3. Функция. По аналогии с ладом функцию будем считать наблюдениями мультиномиальной случайной величины, параметризованной вектором вероятностей pf . Он вычисляется из наблюдаемого вектора вероятностей для лада pm и значения сдвига аккорда cs (в случае, если аккорд наблюдается как вектор вероятностей pc , будем брать значение дискретной случайной величины c с мультиномиальным распределением, параметризованным pc). А именно,

$$pf[l] = pm[(l + cs) \bmod 12],$$

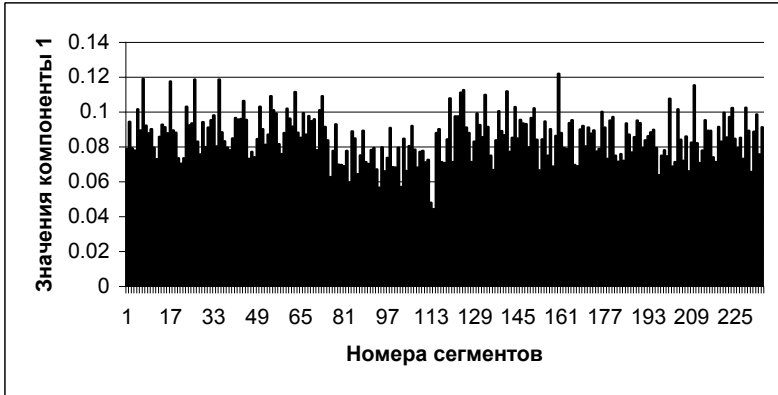


Рис. 7. Значения компоненты 1 (*до-диез*) вектора pt для первых 80 секунд So What.

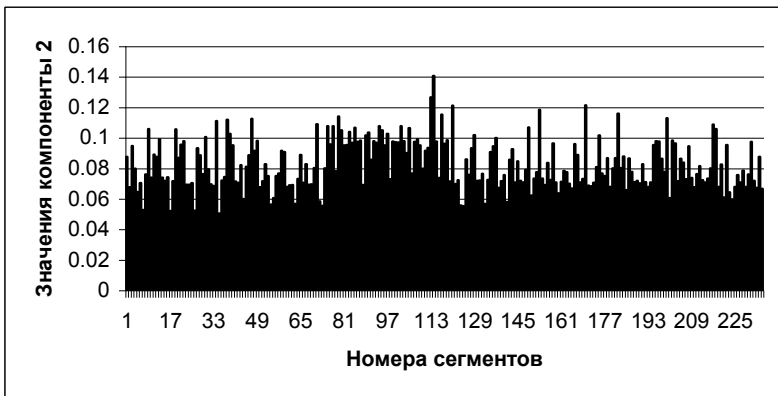


Рис. 8. Значения компоненты 2 (*ре*) вектора pt для первых 80 секунд So What.

то есть pf получается сдвигом pt влево на значение сдвига аккорда cs .

2.5.4. Виды распределений. Стиль s является дискретной случайной величиной. На этапе обучения с учителем его значения наблюдаются, а на этапе самообучения и использования нас будет интересовать параметризующий мультиномиальное распределение s вектор вероятностей ps . Случайная величина ps имеет сопряженное априорное распределение Дирихле

$$Dir(pS_\alpha): f(x_1, \dots, x_{n_s}; \alpha_1, \dots, \alpha_{n_s}) = \frac{\Gamma(a)}{\prod_i \alpha_i} \prod_i x_i^{\alpha_i - 1}, a = \sum_i \alpha_i \quad [23].$$

Тип аккорда ct при обучении — просто наблюдаемая переменная. При использовании следует, по аналогии со стилем, ввести случайную величину, pct , которая имеет сопряженное априорное распределение Дирихле и характеризует вектор вероятностей ct .

Случайная переменная, наблюдениями которой при условии стиля композиции ν и текущего типа аккорда t являются векторы вероятностей функции pf , имеет многомерное нормальное распределение $N(\underline{\mu}'_{\nu t}, \Sigma_{\nu t}^{-1})$. Индексы t, ν подчеркивают зависимость от стиля и типа аккорда.

Случайная величина $pfMeans$, характеризующая вектор средних значений $\underline{\mu}'_{\nu t}$, имеет сопряженное априорное многомерное (12-мерное) нормальное распределение $N(\underline{\mu}_{\nu t}, \Sigma_{\nu t}^{-1})$ с плотностью распределения

$$N(\bar{x}; \underline{\mu}, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\bar{x} - \underline{\mu})^T \Sigma^{-1} (\bar{x} - \underline{\mu})\right) \quad [7].$$

Случайная величина $pfPrecs$, характеризующая матрицу Σ'^{-1} , обратную к матрице ковариаций (англ. precision, будем называть ее *матрицей точности*), имеет сопряженное априорное распределение Висхарта $W_{12}(a_{\nu t}, \mathbf{B}_{\nu t})$ с плотностью распределения

$$p(\mathbf{X}) = |\mathbf{X}|^{a - \frac{d+1}{2}} e^{-\text{tr}(\mathbf{X}\mathbf{B})} \frac{|\mathbf{B}|^a}{\Gamma_d(a)} \quad [23].$$

Наконец, величины e_0, e_1 имеют распределение, получающееся из нормального путем наложения ограничения на значение (напомним, что значения должны принадлежать отрезку $[0, 1]$). Исходное нормальное распределение имеет центр в 0 для e_0 и в 1 для e_1 , нас будут интересовать распределения для его обратных дисперсий (англ. precision) $\sigma_0^{-1}, \sigma_1^{-1}$. Сопряженными априорными распределениями являются гамма-распределения $\Gamma(k, \theta)$ с плотностью распределения вероятностей

$$g(x; k, \theta) = x^{k-1} \frac{\exp\left(-\frac{x}{\theta}\right)}{\theta^k \Gamma(k)} [7].$$

Будем обучать только один из этих параметров, задавая θ вручную.

Свойства указанных здесь сопряженных априорных распределений подробно рассмотрены в [7].

Таким образом, результатом обучения являются значения $\bar{\mu}_{v_t}, \Sigma_{v_t}^{-1}, a_{v_t}, \mathbf{B}_{v_t}$ для всех t, v , а также k_0, k_1 (параметры $\sigma_0^{-1}, \sigma_1^{-1}$).

2.5.5. Фактор-граф. Результирующий фактор-граф представлен на Рис. 9.

В графе используется функция *Ntrunc*, которая, как объяснялось в первом разделе настоящей главы, является нормальным распределением с наложенными ограничениями на значение (ограничение на отрезок $[0;1]$). Также используется функция Noisy-OR, уравнение для которой в терминах лада и аккорда m и s легко переформулировать в виде уравнения с участием cs, f, g . Для соотношения, связывающего вектор вероятностей функции pf с pm и cs введено сокращение

$$pf = \text{shift}(pm, cs) \Leftrightarrow pf[l] = pm[(l + cs) \bmod 12].$$

3. Практические исследования.

3.1. Замечания о реализации.

3.1.1. Вычисление хрома-векторов. Для практических исследований методов, входными данными которых является звуковой сигнал, одна из самых острых проблем — поиск хорошего источника данных. По причинам, связанным с авторским правом, отсутствуют доступные любому исследователю фонотеки достаточно большого объема. Наличие такой фонотеки очень важно для повторяемости результатов, а также для сравнения различных методов. Выход из этой ситуации — создание баз композиций, содержащих не аудиоданные, а результаты предобработки, не подпадающие под действие авторских прав, распространяющихся на сами композиции.

Благодаря взаимодействию проекта EchoNest, предоставляющего веб-сервис, вычисляющий различного рода информацию из предоставляемого аудиосигнала, и группы исследователей из лаборатории LabRosa удалось создать набор данных для 1000000 композиций, доступный любому желающему для использования — Million Song Dataset [12, 29].

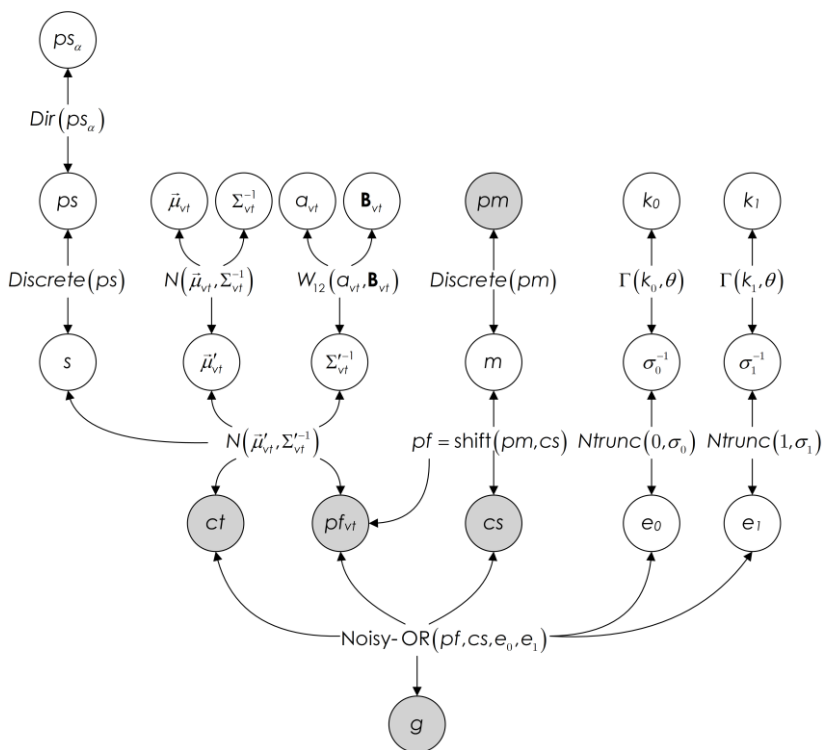


Рис. 9. Фактор-граф модели гармонии

В частности, этот набор данных содержит вычисленные хрома-векторы, сегментированные по музыкальным событиям (начало ноты, конец ноты). Хрома-векторы, другие признаки, вычисляемые из аудиосигнала, а также некоторые метаданные для каждой музыкальной композиции собраны в файл в формате HDF5 [28].

Помимо Million Song Dataset, в формате HDF5 существуют использованные ранее многими лабораториями фонотеки BEATLES (полная официальная дискография *The Beatles*) и USPOP (популярные в США песни разных исполнителей).

Реализованная в ходе данной работы модель рассчитана на использование хрома-векторов, предоставленных в формате, используемом в вышеуказанных наборах данных, что является ее преимуществом — это позволяет без изменения кода запускать программу и сравнивать результат с последующими публикациями, использующими этот формат.

3.1.2. Разметка аккордов. Для обучения модели используется разметка аккордов для набора данных BEATLES, предоставленная Queen Mary University of London [11]. Созданная в ходе данной работы реализация считывает аккорды, записанные в удобной для человека и легкой для автоматической обработке нотации, предложенной в [8].

3.1.3. Реализация байесовского вывода в графических моделях. Для осуществления вывода была использована система Infer.NET [23], разрабатываемая в Microsoft Research. В данной системе, в частности, реализован алгоритм Expectation Propagation, предложенный одним из ее разработчиков [22]. Преимущества системы — в ее индустриальной направленности: она рассчитана на работу с большими объемами данных, активно используется, в том числе в коммерческих разработках, активно поддерживается. Помимо этого, в ней реализованы удобные конструкции для описания графической модели — операции над случайными величинами, блоки If и Switch для выражения условной зависимости.

Кратко говоря, для программиста, пользующегося Infer.NET, система предоставляет собой своего рода предметно-ориентированный язык (DSL), на котором задается фактор-граф модели. Система выполнена в виде библиотеки для платформы .NET и может использоваться из любого поддерживаемого этой платформой языка, например, C#, Python (IronPython) и F#. Работа с системой выглядит так: задается модель, задаются наблюдаемые значения, вызывается функция, осуществляющая вывод и возвращающая апостериорное распределение для указанной величины. Во время работы программы, использующей Infer.NET, модель по описанию компилируется в .NET-библиотеку при первом вызове функции, производящей вывод.

На данный момент в системе поддерживаются: мультиномиальное распределение, распределение Пуассона, распределение Дирихле, нормальное распределение, гамма-распределение, бета-распределение, многомерное нормальное распределение, распределение Висхарта.

Приведем краткий мотивационный пример: используя Infer.NET, решим задачу определения параметров мультиномиального распределения для нечестной кости по наблюдаемой последовательности результатов бросков.

Так выглядит задание случайной величины, характеризующей вектор параметров 6-мерного мультиномиального распределения и имеющей априорное равномерное распределение Дирихле:

```
var diceProbs = Variable.DirichletUniform(6);
```

Для задания последовательности наблюдаемых применяется массив случайных величин:

```
var n = new Range(20000);
var values = Variable.Array<int>(n);
```

Зададим связь между случайной величиной с распределением, параметризованным *diceProbs*, и наблюдаемыми значениями:

```
values[n] = Variable.Discrete(diceProbs).ForEach(n);
```

Зададим сами наблюдаемые значения (справа от оператора присваивания генерируется последовательность из 10000 нулей, затем по 2000 всех остальных чисел):

```
values.ObservedValue = Enumerable.Range(0,
20000).Select(i => i < 10000 ? 0 : (i - 10000)/2000 +
1).ToArray();
```

Вызовем функцию для вывода, сохраним ссылку на результирующее распределение и выведем результат на экран:

```
var dist = new InferenceEngine()
    .Infer(Variable.Discrete(diceProbs));
Console.WriteLine(dist);
```

Будет выведено

```
Compiling model...done.
```

```
Discrete(0.4999 0.1 0.1 0.1 0.1 0.1)
```

3.2. Обучение модели. В рамках нашего исследования было реализовано обучение модели на размеченных данных. Рассматривалось обучение только гиперпараметров распределения pf_{vr} — $\bar{\mu}_{vr}, \Sigma_{vr}^{-1}, a_{vr}, \mathbf{V}_{vr}$. Задание наблюдаемых значений переменных расщепляет фактор-граф; фактически, обучение производится отдельно для его верхней и нижней части, поэтому практический анализ обучения одной из частей имеет смысл. Поскольку величина, характеризующая функцию, f , является жестко заданной относительно сдвига аккорда cs и лада m , а также поскольку все эти величины являются наблюдаемыми, то для обучения вышеуказанных параметров cs и m никакой роли не играют и могут быть удалены из модели.

Реализующая модель программа устроена следующим образом: для каждой композиции вычисляется набор векторов вероятностей ладов pt , из файла, содержащего разметку аккордов, считываются значения ct и cs для каждого сегмента, после чего вычисляется последовательность векторов вероятностей функции pf . В качестве наблюдаемых значений задаются последовательности ct , pf ; при обучении с учителем задается наблюдаемое значение стиля s .

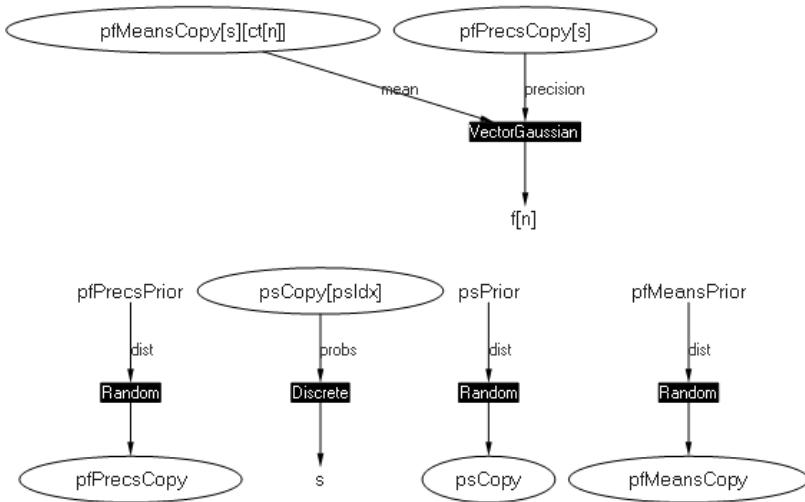


Рис. 10. Фактор-граф, построенный системой Infer.NET по описанию модели гармонии для обучения

Фактор-граф, построенный Infer.NET по описанию модели, приведен на рис. 10.

Названия вершин в целом соответствуют названиям введенных в рассмотрение величин; так, $pfMeansCopy[s][ct[n]]$ соответствует $pfMeans_v$, где $v = s$, а $ct[n] = t$; аналогично с $pfPrecs$. Под $f[n]$ подразумевается pf_k . Суффикс *Copy* у некоторых величин — техническая особенность разделяемых величин между запусками модели с различными наблюдаемыми значениями; суффикс *Prior* означает априорное распределение.

3.2.1. Обучение на искусственных данных. Для проверки того, что модель действительно определяет искомые закономерности в задаваемых данных, было осуществлено обучение на искусственных данных с известными зависимостями.

При обучении на искусственных данных логично уменьшить размерность задачи для упрощения анализа. Было решено задать 3 возможных стиля, 4 возможных компоненты хрома-вектора, 1 тип аккорда. Обучение производилось следующим образом: было сгенерировано 12 последовательностей значений pf , представляющих 12 «композиций», по 4 на каждый стиль. Схему обучения проще всего представить в виде псевдокода:

```
for i = 0..1
```

```

for style = 0..2
  LearnSupervised(GenerateSong(style), style)
for i = 0..1
  for style = 0..2
    LearnUnsupervised(GenerateSong(style))

```

Опишем, как были сгенерированы данные для заданного стиля. Все рассматриваемые последовательности имеют длины 1000.

- 1) Генерируется случайная последовательность истинных значений ладов m_{true} .
- 2) Из нее генерируется последовательность векторов вероятностей pm :

$$pm_k[l] = \begin{cases} \frac{5}{8}, & l = m_{true_k}; \\ \frac{1}{8} & \text{иначе.} \end{cases}$$

- 3) Каждый элемент последовательности m_{true} складывается по модулю 4 с результатом наблюдения, имеющего мультиномиальное распределение случайной величины, вектор параметров которого берется из заранее приготовленной таблицы для стиля.

В модель задаются в качестве наблюдаемых значений последовательность pm и вычисленная на шаге 3 последовательность сдвигов аккорда cs .

После проведения обучения исследуются выведенные параметры распределения для стиля.

Использованная для генерации таблица векторов вероятностей приведена в таблица . Фактически, эта таблица задает значения средних $\bar{\mu}_v$, которые должны получиться в результате обучения. Также следует убедиться, что полученные распределения для стиля ps_α соответствуют истинному значению стиля.

Таблица 1. Векторы вероятностей для генерации искусственных данных

| Стиль v | Вектор вероятностей ($\bar{\mu}_v$) |
|-----------|---------------------------------------|
| 0 | 0.40, 0.10, 0.40, 0.10 |
| 1 | 0.60, 0.20, 0.20, 0.20 |
| 2 | 0.25, 0.30, 0.30, 0.15 |

Приведем результаты работы модели.

Распределения для стилей:

```

[0] Dirichlet(5 1 1)
[1] Dirichlet(1 5 1)
[2] Dirichlet(1 1 5)
[3] Dirichlet(5 1 1)
[4] Dirichlet(1 5 1)

```

```
[5] Dirichlet(1 1 5)
[6] Dirichlet(5 1 1)
[7] Dirichlet(1 5 1)
[8] Dirichlet(1 1 5)
[9] Dirichlet(5 1 1)
[10] Dirichlet(1 5 1)
[11] Dirichlet(1 1 5)
```

Значения для $\bar{\mu}_i$:

```
[0] (0.3277 0.1719 0.3220 0.1784)
[1] (0.3745 0.2130 0.2079 0.2046)
[2] (0.2529 0.2015 0.2777 0.2679)
```

Значения для a_i :

```
[0] 2003.00000010452
[1] 2002.99999855706
[2] 2003.00000133842
```

3.2.2. Обучение на реальных данных. Как отмечалось выше, в качестве реальных данных был взят набор данных BEATLES, содержащий 180 песен. Для оценки качества обучения важно выбрать количество стилей и разметить песни в наборе данных, при этом композиции, объединенные по стилю в группы, должны быть похожи в смысле гармонии внутри группы и различаться между группами. В ходе подготовки исследований было поставлено два вопроса: есть ли такие группы в каталоге песен *Beatles*, и, если да, то как их выделить?

Для разрешения этого вопроса лучше всего обратиться к экспертам. К счастью, разным аспектам творчества *Beatles* посвящено множество работ. В качестве примера приведем серию статей «Заметки о...» [26], созданную профессиональным музыковедом: каждой песне посвящена отдельная статья, в которой проводится анализ, в том числе и гармонический, анализируется стиль автора песни. Поллак приходит к выводу, что песни *Beatles* весьма разнообразны с точки зрения гармонии.

Другой источник [16] на основе анализа гармонии утверждает, что стили двух основных авторов песен *Beatles*, Джона Леннона и Пола Маккартни, можно различить с точки зрения гармонии.

На основе вышеприведенных свидетельств, а также на основе частных мнений знатоков дискографии *Beatles* (в их числе исследователей в области информационного поиска в музыке) на поставленные вопросы были даны следующие ответы: да, такие группы есть, и их можно выделить на основе знания о том, кто является основным автором.

Анализ списка песен [10] показывает, что можно выделить 5 групп по следующим авторам:

- 1) отдельная группа, в которую попадают исполнения чужих композиций с ранних альбомов (как правило, все они относятся к гармонически простому ритм-энд-блюзу/рок-н-роллу; 20 песен),
- 2) Джон Леннон и Пол Маккартни (совместно; 12 песен),
- 3) Джон Леннон (64 песни),
- 4) Пол Маккартни (64 песни),
- 5) Джо Харрисон (20 песен).

Внутри каждой группы было выбрано по 4 песни для обучения с учителем, по возможности гармонически различные (всего 20 песен). Результаты обучения для распределений стилей ps_α представлены в таблица 1.

На первых 20 песнях проводилось обучение с учителем — стиль был задан. На остальных песнях проводилось самообучение. Результаты для тех песен, которые не вошли в таблицу, идентичны результатам для песен 21-40.

Были исследованы и другие варианты постановки теста. Так, из рассмотрения последовательно выключались песни, имеющие стиль 2; тем не менее, оставшиеся песни продолжали классифицироваться в один и тот же стиль. Рассмотрен был и вариант, аналогичный обучению на искусственных данных: в наборе данных оставлялось только по 4 песни для 3 стилей, заведомо гармонически различных (типичный рок-н-ролл; баллада со спуском по квинтовому кругу; песня с необычными гармоническими ходами).

Также исследовались различные варианты модели. Устранялось, например, уменьшение модели за счет введения случайной величины pf ; вместо текущего лада и аккорда рассматривались пары (текущий аккорд; следующий аккорд); вместо пары дискретных наблюдаемых величин c рассматривался вектор вероятностей значений c .

Все вышеперечисленные модификации набора данных и модели приводили к результатам, аналогичным тем, что были представлены выше.

3.3. Обсуждение результатов и выводы. Результаты обучения на искусственных данных показывают, что модель реализована адекватно выдвинутым предположениям о характере закономерностей в использовании гармонии. В то же время, результаты обучения на реальных данных свидетельствуют о том, что в них искомая закономерность отсутствует. Действительно, тот факт, что все без исключения песни на этапе самообучения классифицируются в одну и ту же группу, показывает, что наша модель не смогла различить предложенные песни.

Таблица 1. Векторы распределений ps_α для стилей для первых 60 песен

| N | ps_alpha | N | ps_alpha | N | ps_alpha |
|----|-----------|----|-----------|----|-----------|
| 1 | 3 1 1 1 1 | 21 | 1 1 2 1 1 | 41 | 1 1 3 1 1 |
| 2 | 2 1 1 1 1 | 22 | 1 1 2 1 1 | 42 | 1 1 3 1 1 |
| 3 | 3 1 1 1 1 | 23 | 1 1 3 1 1 | 43 | 1 1 3 1 1 |
| 4 | 3 1 1 1 1 | 24 | 1 1 3 1 1 | 44 | 1 1 2 1 1 |
| 5 | 1 2 1 1 1 | 25 | 1 1 3 1 1 | 45 | 1 1 2 1 1 |
| 6 | 1 2 1 1 1 | 26 | 1 1 2 1 1 | 46 | 1 1 3 1 1 |
| 7 | 1 2 1 1 1 | 27 | 1 1 3 1 1 | 47 | 1 1 3 1 1 |
| 8 | 1 1 2 1 1 | 28 | 1 1 2 1 1 | 48 | 1 1 2 1 1 |
| 9 | 1 1 2 1 1 | 29 | 1 1 2 1 1 | 49 | 1 1 3 1 1 |
| 10 | 1 1 3 1 1 | 30 | 1 1 3 1 1 | 50 | 1 1 3 1 1 |
| 11 | 1 1 3 1 1 | 31 | 1 1 3 1 1 | 51 | 1 1 2 1 1 |
| 12 | 1 1 2 1 1 | 32 | 1 1 2 1 1 | 52 | 1 1 2 1 1 |
| 13 | 1 1 3 1 1 | 33 | 1 1 3 1 1 | 53 | 1 1 3 1 1 |
| 14 | 1 1 1 2 1 | 34 | 1 1 2 1 1 | 54 | 1 1 3 1 1 |
| 15 | 1 1 1 2 1 | 35 | 1 1 2 1 1 | 55 | 1 1 2 1 1 |
| 16 | 1 1 1 2 1 | 36 | 1 1 3 1 1 | 56 | 1 1 2 1 1 |
| 17 | 1 1 1 3 1 | 37 | 1 1 3 1 1 | 57 | 1 1 2 1 1 |
| 18 | 1 1 1 2 1 | 38 | 1 1 3 1 1 | 58 | 1 1 2 1 1 |
| 19 | 1 1 1 1 2 | 39 | 1 1 2 1 1 | 59 | 1 1 2 1 1 |
| 20 | 1 1 1 1 3 | 40 | 1 1 3 1 1 | 60 | 1 1 2 1 1 |

Отсутствие искомой зависимости в реальных данных означает, что модель плохо описывает те свойства гармонии, по которым можно различать композиции. При создании модели мы опирались на предположения, которыми мы воспользовались по аналогии с работами, указанными в [33]; некоторые из них хорошо зарекомендовали себя в задаче извлечения последовательности аккордов. Полученный результат является важным свидетельством того, что такие предположения неприменимы в задаче оценки схожести музыкальных композиций. По-видимому, недостатком введенных предположений может являться либо отсутствие учета порядка аккорда в последовательности, либо отсутствие учета ритмической информации.

Укажем возможные пути развития модели. По аналогии с вероятностными моделями для извлечения аккордов можно предположить, что использование временных зависимостей — введение марковских цепей — позволит добиться эффективной оценки схожести гармонии.

4. Заключение. В работе предлагается графическая вероятностная модель, комбинирующая лад и аккорд для наблюдаемых значений хрома-векторов; музыкальное произведение представляется в виде неупорядоченного набора сегментов. Описана процедура обучения и использования модели, дается фактор-граф модели.

В ходе практических исследований предложенная модель реализуется в среде Infer.NET с целью проведения ее обучения. Полученная реализация хорошо обучается на искусственных данных, что позволяет говорить об ее адекватности модели. Обучение на реальных данных убедительно показало, что идея совместного использования информации о ладе и аккорде в отрыве от контекста музыкального произведения не позволяет давать оценку схожести произведений. Предлагаются варианты устранения недостатков модели, потенциально позволяющие построить эффективную модель для поставленной задачи.

Литература

1. *Allan H., Mullensiefen D., Wiggins G.* Methodological Considerations In Studies Of Musical Similarity // Proceedings of the International Conference on Music Information Retrieval (ISMIR 2007) (Vienna, Austria, September 23–27, 2007) URL: http://ismir2007.ismir.net/proceedings/ISMIR2007_p473_allan.pdf
2. *Anderson C.* The Long Tail: Why the Future of Business Is Selling Less of More. New York: Hyperion, 2006. 256 pp.
3. *Bas de Haas W., Veltkamp R., Wiering F.* Tonal Pitch Step Distance: a Similarity Measure for Chord Progressions // Proceedings of the Ninth International Conference on Music Information Retrieval (ISMIR 2008) (Philadelphia, Pennsylvania, September 14–18, 2008) Pp. 51–56.
4. *Bishop C.* Pattern Recognition and Machine Learning. Springer, 2006. 738 pp.
5. *Cano P., Koppenberger M., Wack N.* Content-based music audio recommendation // Proceedings of the 13th annual ACM international conference on Multimedia (MULTIMEDIA'05) (Singapore, November 6–11, 2005). Pp. 211–212.
6. *Castelluccio M.* The Music Genome Project // Strategic Finance. 2006. № 88 (6). Pp. 57–58.
7. *Fink, D.* A Compendium of Conjugate Priors. URL: <http://www.johndcook.com/CompendiumOfConjugatePriors.pdf>.
8. *Harte C., Sandler M., Abdalla, S., Gomez E.* Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations // In J. Reiss & G. Wiggins (Eds.), Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005) (London, UK, September 11–15, 2005). London, UK: University of London, 2005. Pp. 312–319.
9. Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more. URL: <http://amazon.com>
10. List of The Beatles songs // Википедия. [2011—2011]. Дата обновления: 11.11.2011. URL: http://en.wikipedia.org/w/index.php?title=List_of_The_Beatles_songs&oldid=460172268
11. Reference Annotations: The Beatles. URL: <http://isophonics.net/content/reference-annotations-beatles>
12. Million Song Dataset. URL: <http://labrosa.ee.columbia.edu/millionsong/>
13. Last.fm - Listen to internet radio and the largest music catalogue online. URL: <http://last.fm>

14. Pandora Radio — Listen to Free Internet Audio, Find New Music. URL: <http://pandora.com>
15. Apple - iTunes - Everything you need to be entertained. URL: <http://www.apple.com/itunes/>
16. Secrets of The Beatles // SeeChord. URL: <http://www.seechord.co.uk/song-writing/secrets-of-the-beatles/>
17. *Anderson C.* The Long Tail // Журнал Wired: [сайт]. URL: <http://www.wired.com/wired/archive/12.10/tail.html>
18. Netflix - Watch TV Shows Online, Watch Movies Online. URL: <https://www.netflix.com/>
19. *Joyce J.* Pandora and the Music Genome Project / Scientific Computing. 2006. № 23 (10). Pp. 14, 40–41.
20. *Krumhansl C.* Cognitive Foundations of Musical Pitch. New York: Oxford University Press, 1990. 318 pp.
21. *Krumhansl C., Shepard R.* Quantification of the hierarchy of tonal functions within a diatonic context // Journal of Experimental Psychology: Human Perception and Performance. 1979. № 5. Pp. 579–594.
22. *Minka T.* Expectation propagation for approximate Bayesian inference // In Breese, Jack S. and Koller, Daphne (eds.), Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (Seattle, WA, August 2–5, 2001). Massachusetts: Morgan Kaufmann, 2001. Pp. 362–369.
23. *Minka T., Winn J., Guiver J., Knowles D.* Infer.NET 2.4, Microsoft Research Cambridge, 2010. URL: <http://research.microsoft.com/infernet>
24. *Muller M.* Information Retrieval for Music and Motion. Berlin: Springer-Verlag, 2007. 334 pp.
25. *Pickens J., Crawford T.* Harmonic Models for Polyphonic Music Retrieval // Proceedings of International Conference on Information and Knowledge Management (CIKM 2002) (McLean, VA, November 4–9, 2002). Pp. 430–437.
26. *Pollack A.W.* Notes on ... Series, 1989–2000 [Электронный ресурс]. The 'Official' rec.music.beatles Home Page: [сайт]. URL: <http://www.recmusicbeatles.com>.
27. *Scaringella N., Zoia G., Mlynek D.* Automatic genre classification of music content: a survey // IEEE Signal Processing Magazine. 2006. Vol. 23, № 2. Pp. 133–141.
28. *The HDF Group.* Hierarchical data format version 5, 2000–2010. URL: <http://www.hdfgroup.org/HDF5>.
29. *Bertin-Mahieux T., Ellis D.P.W., Whitman B., Lamere P.* The million song dataset // In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), (Miami, Florida, October 24–28, 2011). URL: <http://ismir2011.ismir.net/papers/OS6-1.pdf>
30. *Yoshii K., Goto M., Komatani K., Ogata T., Okuno H.G.* Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences // Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006) (Victoria, Canada, October 8–12, 2006). URL: http://ismir2006.ismir.net/PAPERS/ISMIR0647_Paper.pdf.
31. *Абызова Е.А.* Гармония: Учебник. М.: Музыка, 2008, 383с.
32. *Балтийский И.А.* Сравнение методов вычисления признаков для задачи поиска музыки по голосу. Бакалаврская работа, СПбГУ ИТМО, 2009.
33. *Балтийский И.А., Николенко С.И.* Обзор графических вероятностных моделей гармонии для анализа музыкальных произведений. // Труды СПИИРАН. СПб.: Наука, 2011.
34. *Балтийский И.А., Николенко С.И.* Системы query-by-humming: обзор подходов и схема платформы для экспериментов // Труды СПИИРАН. СПб.: Наука, 2008. Вып. 7. С. 75–92.

35. Тулупьев А.Л., Николенко С.И., Сироткин А.В. Байесовские сети доверия: логико-вероятностный вывод в ациклических направленных графах. СПб.: Изд-во С.-Петербург. ун-та, 2009, 400 с.
36. Тулупьев А.Л., Николенко С.И., Сироткин А.В. Байесовские сети: логико-вероятностный подход. СПб.: Наука, 2006. 608 с.
37. Тулупьев А.Л. Алгебраические байесовские сети: реализация логико-вероятностного вывода в комплексе java-программ // Труды СПИИРАН. СПб.: Наука, 2009. Вып. 8. С. 191–232.
38. Тулупьев А.Л. Задача локального автоматического обучения в алгебраических байесовских сетях: логико-вероятностный подход // Труды СПИИРАН. 2008. Вып. 7. СПб.: Наука, 2008. С. 11–25.
39. Фильченков А.А., Тулупьев А.Л. Структурный анализ систем минимальных графов смежности // Труды СПИИРАН. 2009. Вып. 11. С. 104–127.

Балтийский Игорь Андреевич — магистр прикладной математики и информатики, студент Санкт-Петербургского государственного университета информационных технологий, механики и оптики (СПбГУ ИТМО). Область научных интересов: информационный поиск музыки, вероятностные модели, теория музыки, цифровая обработка сигналов. Число научных публикаций — 2. iosank@gmail.com; СПбГУ ИТМО, ул. Саблинская, д. 14, г. Санкт-Петербург, 197101; +7(921)792-73-22. Научный руководитель — С.И. Николенко.

Baltiyskiy Igor Andreevich — M. Sc. in Applied Mathematics and Informatics, student, Saint-Petersburg State University of Fine Mechanics and Optics (SPbSU Technical University IFMO). Research interests: music information retrieval, probabilistic models, music theory, digital signal processing. Number of publications — 2. iosank@gmail.com; Technical University IFMO, Russia, 197101, Sablinskaya street, 14; +7(921)792-73-22. Supervisor: S.I. Nikolenko.

Николенко Сергей Игоревич — к.ф.-м.н.; научный сотрудник лаборатории математической логики Учреждения РАН Санкт-Петербургского отделения Математического института им. В.А. Стеклова РАН, старший научный сотрудник проблемной лаборатории алгоритмической биологии Учреждения РАН Санкт-Петербургского Академического Университета — научно-образовательного центра нанотехнологий РАН. Число научных публикаций — 60; sergey@logic.pdmi.ras.ru, <http://logic.pdmi.ras.ru/~sergey>; наб. р. Фонтанки, 27, Санкт-Петербург, 191023, Россия; р.т. +7(812)3124058, факс +7(812)3105377.

Nikolenko Sergey Igorevich — Ph.D.; Researcher at the Laboratory of Mathematical Logic, Steklov Mathematical Institute, St. Petersburg, Senior Researcher at the Laboratory of Algorithmic Biology, St. Petersburg Academic University. Number of publications — 60; sergey@logic.pdmi.ras.ru, <http://logic.pdmi.ras.ru/~sergey>; nab. r. Fontanki, 27, St. Petersburg, 191023, Russia; tel. +7(812)3124058, fax +7(812)3105377.

Поддержка исследований. Работа второго автора была поддержана грантом Правительства РФ для государственной поддержки научных исследований, проводимых под руководством ведущих ученых (грант № 11.G34.31.0018), грантом Президента РФ для молодых кандидатов наук (грант МК-4089.2010.1), грантом Президента для ведущих научных школ НШ-5282.2010.1, ФЦП «Научные и научно-педагогические кадры инновационной России» и Российским Фондом Фундаментальных Исследований.

Статья поступила в редакцию 12.04.2011 г.

РЕФЕРАТ

Балтийский И.А., Николенко С.И. **Графическая вероятностная модель для оценки схожести гармонии музыкальных произведений.**

Оценка схожести музыкальных произведений — задача, имеющая интересное практическое применение: на основе ее решения можно построить полностью автоматическую систему рекомендации музыки, ориентированную на содержание. Известны методы, которые позволяют разбить эту задачу на подзадачи, соответствующие различным граням музыки. В нашей работе рассматривается одна из таких подзадач — оценка схожести гармонии музыкальных композиций.

Цель работы — исследовать простейший вариант модели, включающий в себя информацию о текущем аккорде и ладе в некоторый момент времени звучания произведения. Вводится понятие хрома-вектора, на котором основано представление аудиосигнала музыкальных композиций, которое мы будем использовать. Рассматриваются способы представления аккордов и ладов в виде хрома-векторов.

Для построения графической вероятностной модели предлагается генеративный процесс, результатом которого являются наблюдаемые значения хрома-векторов и вектора вероятностей события «в текущий момент времени, когда композиция находится в определенном ладе». Вводится скрытый параметр, стиль, который управляет вероятностью использования определенного аккорда в контексте определенного лада. В модели отсутствует зависимость между соседними аккордами, отсутствует и учет ритмической информации. Данный выбор мотивирован результатами сравнительного эксперимента, проведенного автором одного из методов извлечения последовательности аккордов из аудиосигнала; он позволяет упростить модель. Указывается возможность использования модели при отсутствии размеченных аккордов, описан процесс обучения модели на данных с размеченными аккордами. Предлагается способ вычисления оценки схожести композиций как расстояния между векторами параметров распределений для стилей двух композиций.

Описана реализация обучения модели с использованием библиотеки Infer.NET, позволяющей проводить приближенный байесовский вывод в произвольных графических вероятностных моделях. Описаны результаты обучения модели на искусственных данных, которые свидетельствуют о том, что реализация модели адекватна заложенным в нее предположениям.

Приводятся результаты обучения модели на реальных данных (фонотеке Beatles). Поскольку, по свидетельствам специалистов, различным авторам присуща разница в подходах к гармонии, ожидалось, что песни будут кластеризованы по стилям в несколько классов. Оказалось, что все песни были отнесены моделью к одному классу. Этот отрицательный результат свидетельствует о том, что использованных в модели простых предположений недостаточно для надежного извлечения информации о схожести гармонии. Данный результат обсуждается, предлагаются изменения, потенциально позволяющие построить эффективную модель для поставленной задачи.

SUMMARY

Baltiyskiy I.A., Nikolenko S.I. **A probabilistic graphical model for the music harmony similarity task.**

The music similarity evaluation problem has an interesting practical application: it may serve as a basis for a completely automated content-based music recommendation system. There are known methods for dividing the task into several sub-tasks each corresponding to a certain facet of music. We consider one of such sub-tasks, namely evaluating the similarity of musical harmony.

The purpose of this work is to present a simple model that includes information about the current chord and mode for some point of time in the musical piece. We introduce the notion of chroma vectors, which is the basis for the representation of audio signals used by our model. We demonstrate how to encode chord and mode information with chroma vector.

To build a probabilistic graphical model we consider a generative process which results in the observed chroma vector values and probability vectors of the event “at the current point, the piece is in mode m ”. A new hidden variable, style, σ , which governs the probability of using of a certain chord within the context of a certain mode is introduced. The model does not include nor the dependencies between the neighboring chords, neither the rhythmic structure information. This choice was motivated by the results of the comparative experiment described in the literature (conducted by the authors of one of the chord extraction models); this decision helps to keep the model simple. We point out the possibility of using of the model without the chord labels and describe the process of learning the model on the labeled chords. We propose to measure the similarity between harmonies of two pieces as the distance between parameter vectors of the distributions for the pieces’ style variables.

We describe the implementation of the model, which uses Infer.NET, a library that allows for approximate Bayesian inference in arbitrary probabilistic graphical models. We describe the results of learning the model on artificial data, which demonstrate that the implementation faithfully reproduces the basic assumptions behind the model.

We also give the results of learning the model on real data, the full Beatles official discography. As expert musicologists have suggested that different authors have different approach in terms of harmony, we expected the songs to be classified into several styles corresponding to their respective authors (Lennon, McCartney, et al.) We observed that all songs were classified into one style. This negative result shows that the basic assumptions that worked for models in the chord extraction task can not be applied to the harmony similarity task. We discuss this result and propose several modifications that may lead to an effective model for the solving of the problem at hand.