Московский авиационный институт

Реферат

на тему: «Метод идентификации музыкальных произведений по аудио фрагментам концертных исполнений»

Выполнил: студент гр. М8О-406Б
Давид Гринберг

Москва 2020

Содержание

Введение

В современном мире наблюдаются следующие тенденции:

— Люди нетерпеливы и привыкли к легкому и быстрому доступу к информации

— Количество доступной информации неукротимо растет и человек не в состоянии справиться с ее потоком без использования поисковиков

— Существенная часть информации – аудиофайлы

Конкретный пример: люди, посещающие различные музыкальные мероприятия, часто сталкиваются с ситуацией, когда на сцене выступает музыкант, а название песни или даже имя исполнителя неизвестно (например, на фестивале). Конечно, можно спросить ближайшего человека, но в таких местах обычно очень шумно. Кроме того нет гарантий, что у этого человека найдется ответ на вопрос. Существует множество методов и сервисов для нахождения музыкальных произведений по отрывку, однако у них есть ряд ограничений:

— Сервисы вроде Shazam способны распознавать только оригиналы

— Некоторые сервисы умеют искать произведения по мелодии, но у них довольно низкая точность.

— Сервисы, которые ищут каверы или ремиксы (для защиты авторских прав) не приспособлены к нахождению зашумленных отрывков, поскольку предполагают, что кавер записывался в студийных условиях

В этой работе рассмотрен метод, лишенный всех вышеобозначенных недостатков. Поскольку музыкальных произведений в мире очень много (по некоторым оценкам около 97 миллионов песен), то очень важно уметь быстро и эффективно по памяти обрабатывать эти данные. Кроме того этот метод применим не только к песням, так как аудиофайл представляет собой временной ряд.
Цели данной работы:

1. Разработать библиотеку, которая предоставляла бы гибкий и удобный интерфейс для эффективной обработки и поиска аудиофайлов

2. Реализовать клиент для идентификации концертных записей, используя разработанную библиотеку

1 Теоретическая часть

1.1 Физика звука

Звук - это вибрация, которая распространяется через воздух (или воду). Например, при прослушивании музыки с компьютера колонки производят вибрации, которые распространяются по воздуху, пока не достигнут уха человека.

Вибрации можно смоделировать с помощью синусоидальных волн.

1.1.1 Чистый тон

Чистый тон - это тон синусоидальной формы волны. Характеристики синусоиды:

— Частота: количество циклов в секунду. Единица измерения - Герц (Гц), например, 100 Гц = 100 циклов в секунду.
— Амплитуда (связана с громкостью звука): размер каждого цикла.

Эти характеристики расшифровываются человеческим ухом для формирования звука. Человек может слышать чистые тоны от 20 Гц до 20000 Гц, и этот диапазон уменьшается с возрастом. Для сравнения, свет, который видит человек, состоит из синусоид от $4 * 10^{14}$ Гц до $7.9 * 10^{14}$ Гц.

Человеческое восприятие громкости зависит от частоты чистого тона. Например, чистый тон с амплитудой равной 10 и частотой 30 Гц будет тише, чем чистый тон с амплитудой 10 и частотой 1000 Гц. Человеческие уши воспринимают звук в соответствии с психоакустической моделью.

Чистых тонов в природе не существует, однако каждый звук в мире - это сумма нескольких чистых тонов с разными амплитудами.

1.1.2 Музыкальные ноты

Ноты разделены на октавы. В большинстве западных стран октава представляет собой набор из 8 нот (A, B, C, D, E, F, G в большинстве англоязычных стран) со следующим свойством:

— Частота ноты в октаве удваивается в следующей октаве. Например, частота A4 (A в 4-й октаве) на частоте 440 Гц в 2 раза превышает частоту A3 (A в 3-й октаве) на 220 Гц и в 4 раза больше частоты A2 (A во 2-й октаве) на 110 Гц.

Частотная чувствительность ушей логарифмическая. Это означает, что:

— между 32.70 Гц и 61.74 Гц (1-я октава)
— или между 261.63 Гц и 466.16 Гц (4-я октава)
— или между 2 093 Гц и 3 951.07 Гц (7-я октава)

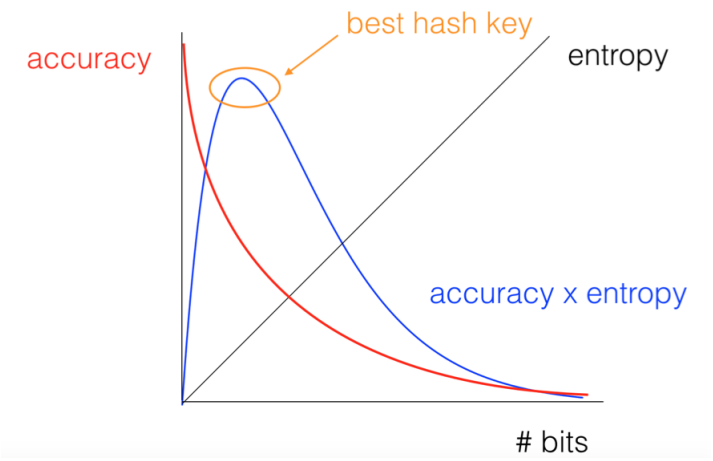
Человеческие уши распознают одинаковое количество нот.

1.2 Техника акустического отпечатка

Для того, чтобы эффективно хранить и искать аудиофайлы, нужно найти какое-нибудь компактное представление, которое при этом будет максимально правдоподобно их описывать. Это представление называется акустическим отпечатком (фингерпринтом) аудиофайла. Существует множество видов таких отпечатков, но большинство методов находят представление аудиофайлов в виде вектора хешей.

Факторы эффективности:

1. Хеши максимизируют произведение функций энтропии и точности:



2. Биты хешей сбалансированы, декоррелированы и имеют высокую дисперсию

1.2.1 Общая идея

Многие алгоритмы фингерпринтинга выглядят так:

1. Посчитать спектрограмму аудиофайла
2. Применить на ней какую-либо оконную функцию (спектрально-временные фильтры)
3. Конвертировать результат в вектор хешей

1.3 Метод хешпринтов

Этот метод предложен в [1]. Он, как и многие другие, находит представление аудиофайла в виде вектора хешей.

Метод отличается следующими характеристиками:

1. Обучение без учителя
2. Высокая адаптивность к данным
3. Независимость от силы сигнала (громкости звука)

Самой важной отличительной чертой метода является обучение без учителя. Такие методы, как, например, Chromaprint, описанный в [2], используют заранее подготовленные спектрально-временные фильтры. Метод хешпринтов находит эти фильтры непосредственно при индексации, что позволяет ему учитывать специфику данных.

Рисунок 1.1 — Фильтры, используемые Chromaprint

1.3.1 Алгоритм вычисления хешпринта

Для вычисления хешпринта, содержащего N бит, нужно проделать следующее:

1. Посчитать спектрограмму.

Результат этапа: матрица $Spectrogram \in \mathbb{R}^{B \times n}$, где B – количество частотных диапазонов, n – количество временных диапазонов.

2. Собрать контекстные фреймы полученной спектрограммы. Фреймы рассчитываются следующим образом:

$$frame_i = V_{i-w}...V_{i+w}$$

, где V_i – столбец спектрограммы, w – количество столбцов контекста.

Результат этапа: матрица $Frames \in \mathbb{R}^{Bw \times n}$

3. Применить к фреймам спектрально-временные фильтры. Фильтры представляют собой $N \times Bw$ матрицу и расситываются с помощью алгоритма обучения без учителя путем решения задачи оптимизации.

Результат этапа: матрица признаков $Features \in \mathbb{R}^{N \times n}$.

4. Посчитать дельту – изменение признаков в течение промежутка T. Дельта рассчитывается по формуле:

$$\Delta_i = feature_i - feature_{i+T}$$

5. Наложить функцию порога и упаковать признаки в хешпринты:

$$hashprint_i = intN(\Delta_i > 0)$$

1.3.2 Вычисление спектрально-временных фильтров

Фильтры подбираются таким образом, чтобы признаки, полученные при их наложении, имели максимальную дисперсию и в то же время были декоррелированы. Для этого можно применить метод главных компонент (PCA):

1. Посчитать ковариационные матрицы для всех матриц фреймов и просуммировать их.

Результат этапа: матрица $CovarianceMatrix \in \mathbb{R}^{Bw \times Bw}$

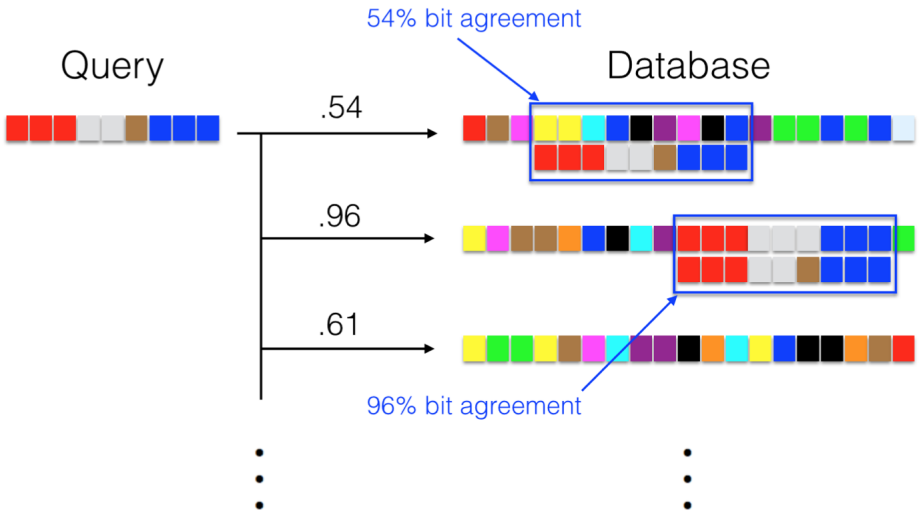
2. Найти N собственных векторов с максимальными собственными значениями.

Результат этапа: матрица $Filters \in \mathbb{R}^{N \times Bw}$

1.3.3 Специфика задачи идентификации живых отрывков

Поскольку речь идет о нечетком поиске, то мы хотим учесть как можно больше нюансов (признаков) сигнала. Поэтому будем представлять аудиофайлы в виде 64-битных хешпринтов. Также в качестве спектрограммы возьмем CQT спектрограмму - она хороша тем, что ее частотные диапазоны можно подобрать таким образом, что они будут соответствовать конкретным нотам. Из-за того что мы имеем дело с пространством довольно большой размерности, мы не можем использовать обратный индекс, поэтому поиск будет выглядеть примерно так:

1. Для каждого оригинала из базы: прикладываем к нему отрывок и ищем такой отступ, чтобы сумма расстояний Хемминга между соответствующими хешпринтами была минимальной.



2. Собираем результаты, сортируем и возвращаем top-N

2 Вторая глава

Заключение

Текст заключения

Список использованных источников

1. Tsai, T. (2016). Audio Hashprints: Theory & Application. (Doctoral dissertation, EECS Department, University of California, Berkeley).

2. Yan Ke, Derek Hoiem, Rahul Sukthankar. (2005). Computer Vision for Music Identification, Proceedings of Computer Vision and Pattern Recognition.

Приложение А Первое Приложение