Эффективность и оптимизация извлечения характеристик из символьной музыки

Чесноков Александр Михайлович[1], Ковалев Дмитрий Андреевич[2]

[1] Московский физико-технический институт, e-mail: chesnokov.am@phystech.edu
[2] SberDevices, e-mail: dmitrii.kovalev@phystech.edu

Вступление

Автоматическая классификация музыки — это процесс, при котором компьютеры автоматически распределяют музыкальную информацию по интересующим категориям. Это может быть идентификация определенных инструментов, тембров и ритмических паттернов в звуковых сигналах. На более высоком уровне можно идентифицировать аккорды, последовательности аккордов, мелодии и так далее.

Исследования в области автоматической классификации музыки имеют значительную ценность как с академической, так и с коммерческой точек зрения. Результаты различных типов классификации могут быть не только очень полезны сами по себе, например, при маркировке и систематизации больших музыкальных коллекций, но и играть важную роль в выполнении составных частей многих крупных и широкомасштабных задач. Например, в машинном обучении.

Одной из программ, которая занимается классификацией музыки, является jSymbolic2[1]. Библиотека jSymbolic2 — это бесплатный Java-инструмент с открытым исходным кодом, основная функция которого заключается в простом извлечении большого количества характеристик и свойств из музыки, записанной в символьном домене. Данная библиотека быстрее своих аналогов, разработанных на языке Python[2], однако все еще имеет недостатки в архитектуре кода и логике работы[3].

Главной целью данной работы является изучение эффективности и оптимизация извлечения характеристик из символьной музыки с помощью данного инструмента.

Результаты

На данный момент построена классовая диаграмма библиотеки для лучшего понимания ее работы рис. [1] а также сделана классовая диаграмма ожидаемой архитектуры кода после модификаций.

Была проделана работа по написанию тестов, измеряющих производительность работы библиотеки jSymbolic2. Для этого была использована библиотека Java Microbenchmark Harness (JMH) [4], позволяющая делать точные замеры по времени с учетом особенностей работы JVM. Для начала

были проведены замеры времени работы библиотеки до каких-либо изменений.

Далее была изменена архитектура кода программы и логика работы. Например, в исходной версии программа считывала файл, обрабатывала его, далее записывала результат и переходила к следующему. После модификации теперь сначала считываются сразу все файлы, затем обрабатываются и затем записываются в файл. Данная модификация будет полезна в дальнейшем для введения многопоточности в программу.

Для тестов был использована коллекция из 1000 файлов, длина которых не более чем 10000 тиков. Отдельно были проведены измерения по извлечению каждой характеристики и всех характеристик сразу. Результат замеров представлен на рисунке [2] и часть записей в таблице [1], находящейся в приложении. Как видно из таблиц и графиков, после оптимизаций производительность по обработке 1000 файлов улучшилась в среднем на 0.2 операций в минуту по каждой характеристике.

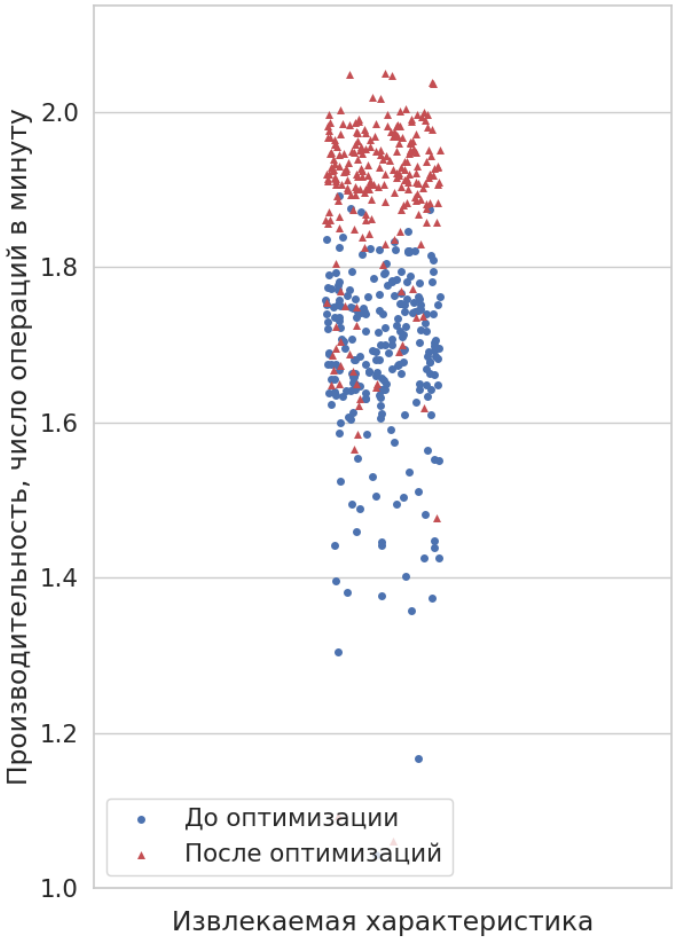

Рис. 2: Замеры производительности до и после оптимизаций.

Список литературы

[1] Cory McKay, Julie E. Cumming, Ichiro Fujinaga «JSYMBOLIC 2.2: EXTRACTING FEATURES FROM SYMBOLIC MUSIC FOR USE IN MUSICOLOGICAL AND MIR RESEARCH».

[2] EXTRACTION FOR SYMBOLIC MUSIC».

[3] Robert C. Martin, «Clean Code», 2008.

[4] JMH, https://github.com/openjdk/jmh

Приложение

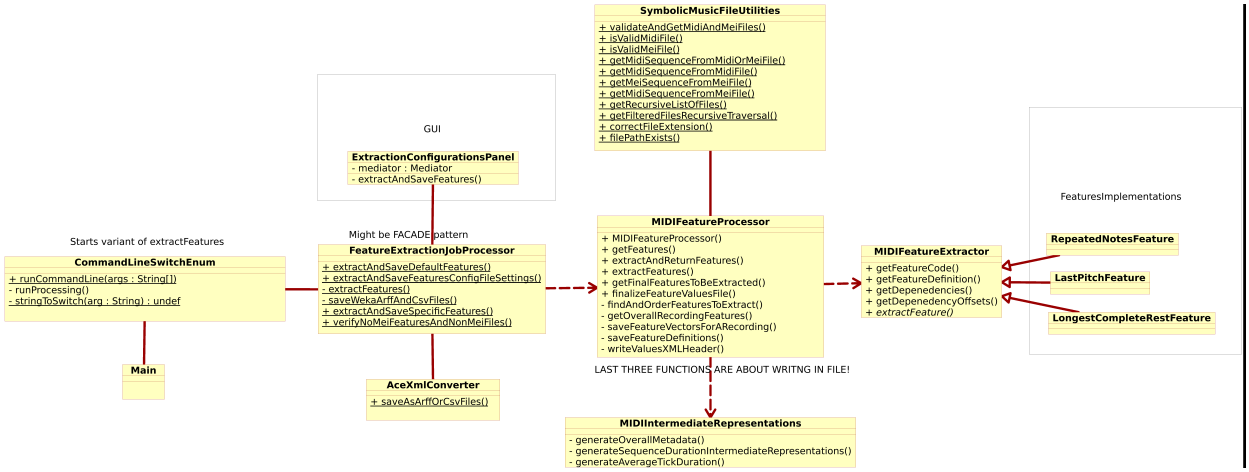

Рис. 1: Классовая диаграмма исходного кода.

Таблица 1. Производительность до и после оптимизаций

Характеристика	**До оптимизаций, операций/мин.**	**После оптимизаций, операций/мин.**
Variability-of-Voice-Separation	1.625849	1.974611
Glissando-Prevalence	1.745762	1.996505
Average-Range-of-Glissandos	1.680217	1.971806
Rhythmic-Variability—Tempo-Standardized	1.742371	1.967759
Chord-Duration	1.761353	1.970225
Voice-Equality—Range	1.611027	2.000448
Prevalence-of-Very-Short-Rhythmic-Values	1.634811	1.966729