

| Изображение | Последовательный алгоритм | Параллельный алгоритм 1 | Параллельный алгоритм 2 |
|----------------|---------------------------|-------------------------|-------------------------|
| 256x256 Lena | 6003 | 74 | 35 |
| 512x512 Nature | >36000 | 1083 | 359 |

Таблица 1 - Результат работы алгоритма, с

Стоит отметить, что не было получено точного результата измерений для изображения Nature, т.к. последовательный алгоритм работал более 10 часов, а затем был принудительно остановлен.

Результаты сжатия с помощью данного метода, а также с помощью метода JPEG:

| Изображение | Исходный размер | JPEG | Последовательный алгоритм | Параллельный алгоритм 1 | Параллельный алгоритм 2 |
|----------------|-----------------|-------|---------------------------|-------------------------|-------------------------|
| 256x256 Lena | 196662 | 11667 | 31312 | 30198 | 31332 |
| 512x512 Nature | 786486 | 60165 | N/A | 181104 | 182973 |

Таблица 1 - Результаты сжатия, байт

Качество изображений, полученных при декомпрессии сжатых изображений, различается в зависимости от изображения, однако остается на достаточно высоком уровне. Оценка проводилась как визуально, так и с помощью метрики SSIM. Результаты метрики:

| Метод | 256x256 Lena | 512x512 Nature |
|---------------------------|--------------|----------------|
| Параллельный алгоритм 1 | 0.962 | 0.978 |
| Параллельный алгоритм 2 | 0.954 | 0.979 |
| Последовательный алгоритм | 0.961 | 0.977 |

Таблица 2 - Результаты метрики SSIM

В результате работы получено программное обеспечение, позволяющее ускорить работу алгоритма в 170 раз. Однако полученные результаты всё же не позволяют говорить о том, что данный метод можно использовать в повседневной деятельности, где необходима возможность постоянно сжимать и восстанавливать изображения. Так же не удалось достичь степени сжатия на уровне JPEG. Выделяя направления для улучшения, можно отметить, что реализация на языке более низкого уровня (C/C++) позволила бы выполнять преобразования быстрее, т.к. Python является относительно медленным в плане выполнения языком. Однако негативным фактором в данном случае является многословность запуска, вызова и использования OpenCL в этих языках. Также улучшениям может подвергнуться часть, написанная с помощью OpenCL. В частности, можно попытаться использовать `__local` память или текстурную память. Помимо этого, улучшению можно подвергнуть и сам алгоритм. К примеру, большое количество исследований в области фрактального сжатия проводились в рамках изучения наилучших стратегий поиска доменных блоков. Существует несколько вариантов поиска, которые могут в различной степени ускорить алгоритм.

Список использованных источников:

1. Винер, Н. Кибернетика, или управление и связь в животном и машине; или Кибернетика и общество/ 2-е издание. - М.: Наука; Главная редакция изданий для зарубежных стран, 1983. - 344 с.
2. Hilbert, M. & Lopez, P. The world's technological capacity to store, communicate, and compute information. / M. Hilbert // Science. - 2011. №332(6025). - P. 60-65.
3. Berger, T. Rate distortion theory: A mathematical basis for data compression / Berger T. - NJ: Prentice-Hall, Englewood Cliffs, 1971
4. Алгоритмы сжатия данных [Электронный ресурс]. - Электронные данные. - Режим доступа: https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%B2#.D0.90.D0.BB.D0.B3.D0.BE.D1.80.D0.B8.D1.82.D0.BC.D1.8B.D1.81.D0.B6.D0.B0.D1.82.D0.B8.D1.8F_.D0.B4.D0.B0.D0.BD.D0.BD.D1.8B.D1.85
5. Barnsley, M. Fractals Everywhere / Michael Barnsley - San Diego: Academic Press, 1988
6. Fisher, Y. Fractal Image Compression - Theory and Application / Y. Fisher - N.Y.: Springer Verlag, 1994. - 341 p.
7. Kominek, J. Advances in Fractal Compression for Multimedia Applications / J. Kominek // Multimedia Systems. - 1997. - №4. - P. 255-270.
8. Encarta [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://www.britannica.com/topic/Encarta>
9. Python [Электронный ресурс]. - Электронные данные. - Режим доступа: <https://ru.wikipedia.org/wiki/Python>
10. OpenCL [Электронный ресурс]. - Электронные данные. - Режим доступа: <https://ru.wikipedia.org/wiki/OpenCL>
11. SSIM [Электронный ресурс]. - Электронные данные. - Режим доступа: <https://ru.wikipedia.org/wiki/SSIM>

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АНАЛИЗА СОВМЕСТИМОСТИ И ПОИСКА АНАЛОГОВ ЛЕКАРСТВЕННЫХ СРЕДСТВ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шеров Э.Б., Шелоник И.А., Холматов М.Дж.

Сиротко С.И. – канд. физ.-мат. наук, доцент

В работе рассматривается задача поиска аналогов и анализа совместимости лекарственных средств, решаемая с использованием средств обработки BIGDATA и алгоритмов машинного обучения. На основе предлагаемой модели для классификации и анализа разрабатывается программный комплекс, осуществляющий поиск лекарственных средств.

Наиболее значительным и распространённым видом воздействия на организм больного человека является фармакотерапия – воздействие лекарственными средствами. В настоящее время различные производители – фармакологические компании – выпускают широчайший ассортимент лекарственных средств, в том числе имеющих схожий состав. Лекарства могут быть как абсолютно аналогичными, с тем же действующим веществом (дженерики), так и с более сильным действием, но и с большим количеством побочных реакций и эффектов. Поскольку значительная часть лекарственных средств предлагается в свободной продаже, и их применение не требует явного назначения врачом, то возможность ориентироваться в их номенклатуре оказывается актуальной не только для специалистов-медиков, но и для широкого круга потребителей.

Цель: Автоматизировать подбор аналогов и анализ совместимости лекарственных средств с использованием технологий BIGDATA.

Для достижения цели нужно выполнить ряд задач:

1. Провести разбор и анализ различных источников.
2. Сбор первичной информации о лекарственных средствах (наполнение БД).
3. Написание парсера.
4. Применение алгоритмов машинного обучения.
5. Визуализация результатов.

В настоящее время имеются несколько главных онлайн источников, откуда потребители могут узнать различную информацию о лекарственных средствах Республики Беларусь:

- 1) Tabletka.by - сайт, где можно найти лекарства в аптеках, включая аналоги по международному непатентованному названию;
- 2) Arteka.103.by – сайт, где можно найти лекарства той же нозологии и международному непатентованному названию;
- 3) Vidal.by – справочник лекарственных средств Республики Беларусь.

Для проведения исследования загружены различные базы данных из открытых источников (база данных FDA, drugBank.com, справочник Vidal).

Разработан и написан на языке Python парсер для обработки базы данных fda, т.к. она имеет очень большой размер, данные в ней хранятся в неструктурированном виде (формат XML), и поэтому не открывается простым редактором. База данных drugBank.com загружена в json формате. Организация баз данных различна, поэтому потребовалось сделать выборку необходимых полей из всех трёх баз данных.

В качестве хранилища использовалась СУБД MongoDB, которая классифицируется как NoSQL.

Было применено несколько подходов и алгоритмов с использованием полученных данных. Применен один из контролируемых алгоритмов машинного обучения – алгоритм многоклассовой классификации [1]. В качестве базиса для обучения сделана выборка по лекарствам каждой фармакологической группы с каждым действующим веществом.

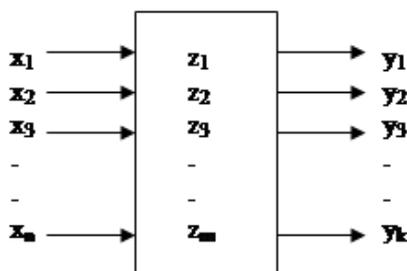


Рис. 1 – Обобщенная модель реализации классификаторов

Модель получает входное множество лекарственных средств, у каждого из которых есть определенное действующее вещество, формирует множество внутренних состояний, отражающих взаимодействия каждого с каждым, а на выход выдает множество возможных побочных эффектов.

Задача совместимости лекарств рассмотрена для рецепторного взаимодействия их друг с другом, но не на тератогенном уровне действия. (Тератогенным действием называют нарушение эмбрионального развития под воздействием тератогенных факторов – некоторых физических, химических (в том числе лекарственных препаратов) и биологических агентов (например, вирусов) с возникновением морфологических аномалий и пороков развития [2].)

При помощи библиотеки D3 языка Javascript отобраны и визуализированы полученные результаты по аналогам лекарственных средств.

Результаты исследования:

В ходе исследования получены теоретические и практические знания в областях BigData, биоинформатики и фармакологии. Спроектирована модель, обеспечивающая поиск совместимости и выявление возможных побочных эффектов по накопленным данным. Также получены практические знания разработки приложений с использованием методов индуктивного машинного обучения, реализованных программными библиотеками NumPy, pandas и Bio на языке Python.

На основе полученных результатов разрабатывается программный комплекс, способный осуществлять быстрый или более детальный и точный подбор аналогичных лекарственных средств, и проведён анализ совместимости лекарств с различным действующим веществом.

В дальнейшем планируется более масштабное тестирование и внедрение результатов, а также рассмотрение задачи совместимости на терагенном уровне с учётом ряда различных факторов.

Список использованных источников:

1. Andreas C. Müller, Sarah Guido (2016) Introduction to Machine Learning with Python
2. Википедия – Терагенное действие [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Терагенное_действие

СРЕДСТВА АДАПТИВНОЙ НАСТРОЙКИ И ПОДБОРА КОНТЕНТА ДЛЯ МУЗЫКАЛЬНОГО ПОРТАЛА.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Холматов М.Дж., Шелоник И.А., Шеров Э.Б.

Сиротко С.И. – канд. физ.-мат. наук, доцент

Рассматривается вопрос осуществления настройки и подбора музыкального контента специализированного портала. Произведен анализ существующих сервисов и приложений, позволяющих подобрать музыкальные композиции в соответствии с предпочтениями пользователя. На основании проведенного анализа предлагается разработать удобный, безопасный и привлекательный для пользователя сервис, способный быстро и эффективно подбирать музыкальный контент.

Актуальность исследования. Онлайн-музыкальные сервисы являются востребованным пользователями ресурсами как альтернатива статичным коллекциям аудиофайлов или сетевым трансляциям благодаря сочетанию индивидуального набора и потенциально неограниченному объему привлекаемых данных. Для качественного выполнения возлагаемых на них функций необходимы механизмы настройки и адаптации сервиса под личные запросы и особенности, причем с минимальными затратами усилий самого пользователя.

Цели и задачи исследования. Целью работы является разработка такого сервиса, способного подобрать актуальный контент для каждого пользователя по его индивидуальным музыкальным пристрастиям.

Для достижения цели необходимо было решить следующие 3 задачи.

Проанализировать существующие сервисы для поиска музыкального контента в сети Интернет. Одним из аналогов разрабатываемой системы является TuneGlue. Принцип работы сервиса заключается в постепенном формировании «карты музыкального путешествия» путем последовательного расширения (expand) от стартовой точки (исполнителя). Существенный минус системы – единственным критерием отбора музыкального контента является исполнитель. Для некоторых исполнителей доступна дополнительная информация о выпущенных альбомах, ссылки на официальный сайт и так далее.

Создать оптимальный список критериев подбора музыкальных композиций, способный осуществлять наиболее эффективный поиск. Подбор контента осуществляется на основе разнообразных критериев: предпочтения пользователя по жанру, настроению и мелодике композиций; географические и временные предпочтения при выборе песен (язык, страна, годы создания); любимые авторы и исполнители и т.д. Первоначально пользователю предлагается из обширной картотеки композиций создать свои плейлисты на разные случаи жизни. В дальнейшем при прослушивании определенного плейлиста пользователю предоставляется список песен, которые по всем или многим критериям являются наиболее подходящими для данного плейлиста. Таким образом, сервис помогает не просто найти новые интересные композиции, а отобрать именно те из них, которые могут в данный момент заинтересовать пользователя. Таким образом, сервис будет учиться на ваших предпочтениях, с каждым разом находя всё более подходящую музыку.

Выбрать инструменты программирования, а также разработать архитектуру приложения и алгоритм поиска, которые позволят создать удобный, безопасный и привлекательный для пользователя сервис. Приложение разработано на языке программирования PHP с использованием JavaScript для обеспечения интерактивности веб-страниц. Уровень представления был реализован при помощи фреймворка Bootstrap. Это свободный набор инструментов для создания сайтов и веб-приложений, включающий в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.