УДК 621.45.01:004.945

СРАВНИТЕЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ ПРОЕКТИРОВАНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ НА ОСНОВЕ АНАЛИЗА КРУПНОФОРМАТНЫХ ДАННЫХ О ПОТРЕБИТЕЛЬСКИХ КОРЗИНАХ

И.А. Олянич[1,a], П.Г. Серафимович[1,2,b]

[1]*Самарский национальный исследовательский университет им. академика С.П. Королева, Самара, Россия*
[a]*14124123@mail.ru,*
[2]*Институт систем обработки изображений РАН, Самара, Россия*
[b] *serafimovich.pg@ssau.ru*

Аннотация

В статье рассмотрены алгоритмы проектирования рекомендательных систем на основе анализа данных о продуктовых покупках пользователей одного из крупных онлайновых ритейлеров. Используя современные методы хранения и анализа данных, эффективные рекомендательные системы позволяют формировать покупательский интерес клиентов и повысить стоимость среднего чека в отдельных заказах. В статье описана построенная аппаратно-программная система на облачных веб-сервисах Amazon EMR и S3. С помощью данной системы изучен исходный набор данных, построены типовые примеры рекомендаций и впервые произведено сравнение алгоритмов Alternating Least Squares и Singular Value Decomposition на облачном сервисе для анализа продуктовых онлайн покупок. Рассмотрено применение фреймворков Apache Hadoop и Apache Spark для анализа крупноформатных данных о потребительских корзинах. В статье выполнен анализ пиковых дней недели и ранжирована загруженность в течение дня. Найдены популярные категории товаров. Классифицирован спрос на различные группы товаров по дням недели и частота покупок. Выявлены зависимости между первым и последующими заказами, популярные товары при первом и последующих заказах, изменения предпочтений клиентов с течением времени.

Ключевые слова: *проектирование рекомендательных систем, Hadoop, Spark, коллаборативная фильтрация, матричная факторизация.*

Цитирование: *Олянич И.А.* Сравнительное исследование алгоритмов проектирования рекомендательных систем на основе анализа крупноформатных данных о потребительских корзинах / И.А. Олянич, П.Г. Серафимович // Онтология проектирования. – 2018. – Т. 8, №4(30). - С.628-640. – DOI: 10.18287/2223-9537-2018-8-4-628-640.

Введение

Понятие Big data включает в себя набор технологий и инструментов для хранения и обработки больших объёмов данных [1]. Анализ данных в рекомендательных системах служит для поиска полезных зависимостей в поведении пользователей [2-4]. Данные зависимости могут использоваться как в режиме реального времени, так и при пакетном анализе заданного временного промежутка. Рекомендательные системы могут использоваться на различных по тематике сайтах. Например, в интернет-журнале пользователю могут рекомендоваться статьи в соответствии с его интересами. В интернет-магазине эффективная рекомендательная система позволит одновременно увеличить прибыль и повысить удобство потребителя.

Целью данной работы является *изучение* шаблонов поведения пользователей сайта, *разработка* аппаратно-программной архитектуры для обработки больших объёмов данных, *реа-*

лизация алгоритмов построения рекомендательных систем Alternating Least Squares (ALS) и Singular Value Decomposition (SVD) и их сравнение.

1 Базовые подходы

Различают два базовых подхода к построению рекомендательных систем:

- коллаборативная фильтрация (*collaborative filtering*);
- контентная фильтрация (*content-based filtering*).

Применяются также гибридные подходы, которые сочетают в себе и то, и другое. Однако сложность таких систем значительно выше [5].

Рекомендательная система строится по тренировочным данным. Оценка качества построенных рекомендательных систем осуществляется следующей метрикой:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}}$$

где P_i - предсказанная i-ая оценка; O_i - её реальное значение в тестовых данных; n - общее число тестовых оценок.

1.1 Коллаборативная фильтрация

В коллаборативной фильтрации можно выделить два базовых метода – это рекомендации, ориентированные на пользователей (*user-based collaborative filtering*) и рекомендации, ориентированные на продукты (*item-based collaborative filtering*).

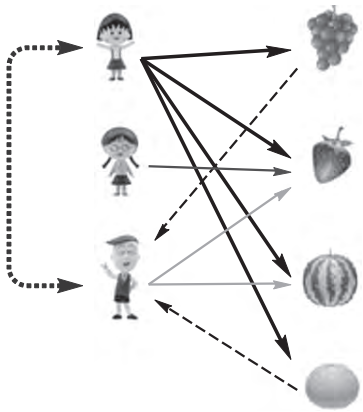

Рисунок 1 – *User-based* подход

User-based подход предполагает предложение товаров, которые покупали пользователи со схожими вкусами. При этом производится усреднение рейтинга товара, оценённого другими пользователями, с учётом степени схожести пользователей [6]. Идею данного метода иллюстрирует рисунок 1 [7]. Из рисунка можно видеть, что между пользователями 1 и 3 имеется явная схожесть в выборе продуктов, следовательно, опираясь на опыт пользователя 1, составляют рекомендации для пользователя 3.

Item-based метод предполагает предложение товаров, которые схожи с теми, что были ранее приобретены пользователем. Производится усреднение рейтинга уже оцененных товаров с учётом степени схожести с неоценённым пока товаром. Идею данного метода представляет рисунок 2 [7].

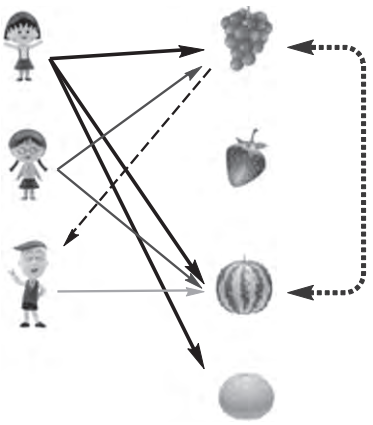

Рисунок 2 – *Item-based* подход

На рисунке 2 пользователю 3 рекомендуется виноград в дополнение к арбузу, так как, опираясь на информацию о корзинах других пользователей, можно заключить, что эти товары часто покупаются вместе [7].

Рекомендательным системам на основе коллаборативной фильтрации присущ недостаток, называемый проблемой «холодного старта». Данная проблема характерна как для *user-based*, так для *item-based* систем.

Т.к. *user-based* система не обладает информацией о новых пользователях, то, как следствие, она не может предлагать им рекомендации. В качестве разрешения данной проблемы можно предложить собирать некоторую информацию

при регистрации, т.е. выполнять так называемый явный сбор данных.

Если не предоставлять *item-based* системе информации о новых объектах, то, как следствие, они никому не будут рекомендоваться. Это можно разрешить анализом свойств объектов при их добавлении, выявляя их характеристики и особенности.

1.2 Контентная фильтрация

Основной идеей контентной фильтрации (фильтрации, основанной на содержимом) является предположение о том, что пользователю интересны объекты, похожие на объекты, которые уже были интересны пользователю ранее [8, 9]. При этом в отличие от коллаборативной фильтрации схожесть объектов определяется не набором действий пользователей, а характеристиками самого объекта. Главной трудностью при построении систем контентной фильтрации является проблема выделения признаковых описаний объектов. В последнее время для автоматического выделения признаковых описаний объектов часто используются методы глубокого обучения.

2 Алгоритм ALS

Исходные данные для рекомендательной системы обычно имеют вид крупноформатной разреженной матрицы A, которая описывает связи пользователей и продуктов. В этой матрице элемент в строке i и столбце j указывает оценку пользователя i продукту j. Для построения рекомендательной системы в соответствии с алгоритмом ALS матрица A факторизуется. Т.е. матрица A представляется как произведение двух матриц X и Y. Высота матрицы X соответствует высоте матрицы A (количеству пользователей), высота матрицы Y соответствует ширине матрицы A (количеству объектов). Две остальных размерности матриц X и Y равны одному значению k. Значение k соответствует количеству латентных факторов, которые выявляются при факторизации. Схему такой факторизации иллюстрирует на рисунок 3 [10].

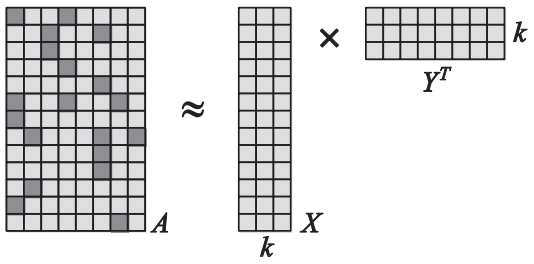

Рисунок 3 – Факторизация матрицы

Подобный алгоритм факторизации часто называется алгоритмом заполнения матрицы. Учитывая то, что исходная матрица A, как правило, является разреженной, произведение X и Y^T предоставляет значение для каждого элемента в матрице A.

Таким образом, матрица X может быть интерпретирована как отношение «пользователь-фактор», а матрица Y как отношение «фактор-продукт» [11-13].

В алгоритме ALS, при вычислении X и Y на основе чередующихся наименьших квадратов, матрица Y инициализируется случайными числами. Алгоритм является итеративным, и на каждой итерации рассчитываются матрицы X и Y. Чтобы рассчитать строку i матрицы X как функцию от Y и строки i матрицы A, используется выражение:

$$A_t \cdot Y \cdot \left(Y^T \cdot Y\right)^{-1} = X_t \quad (1)$$

Данный расчёт может быть выполнен параллельно для каждой строки матрицы X. Аналогичное (1) выражение используется, чтобы рассчитать подматрицу Y_j на основе матрицы X. В процессе итераций минимизируется квадратичная ошибка представления матрицы A произведением матриц XY^T.

3 Алгоритм SVD

В алгоритме SVD исходная матрица A представляется в виде сингулярного разложения: $A = UDV^T$. При этом матрицы U и V — ортогональные, а D — диагональная. Если размер матрицы A — $N \times M$, то размер матриц U и V - $N \times k$ и $k \times M$ соответственно, где k — ранг матрицы A. Учитывая то, что разреженные матрицы часто обладают небольшим рангом, количество параметров сокращается с $N \times M$ до $(N+M) \times k$.

$$A = UDV^T = U \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{pmatrix} V^T$$

В диагональной матрице D диагональные сингулярные элементы упорядочены по убыванию: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k$. Разложение SVD обеспечивает оптимальное приближение при обнулении заданного количества наименьших сингулярных значений.

Чтобы уточнить оценки, предоставляемые алгоритмом SVD, используются поправочные коэффициенты, называемые предикторами $b_{i,a}$:

$$b_{i,a} = \mu + b_i + b_a,$$

где μ - средний рейтинг продуктов в матрице A, b_i - средний рейтинг i пользователя, b_a - средний рейтинг продукта.

4 Технологии Big Data

Существует множество различных технологий обработки крупноформатных данных. В данной работе используются два фреймворка для сбора данных, их хранения и обработки - Hadoop и Spark [14].

4.1 Apache Hadoop

Фреймворк Hadoop состоит из четырёх главных компонент:

1) HDFS (распределенная файловая система);
2) MapReduce (алгоритмический подход к обработке данных);
3) YARN (система планирования заданий и управления ресурсами кластера);
4) Common (набор общих утилит и библиотек).

Hadoop также использует вспомогательные модули Hbase, Zookeeper, Oozie, Pig и Hive. Каждый из модулей является отдельным программным пакетом. Поэтому можно использовать только ту часть из них, которая необходима.

4.2 Apache Spark

Apache Spark – это фреймворк с открытым исходным кодом для параллельной и распределённой обработки и анализа слабоструктурированных данных в оперативной памяти [15].

Главными преимуществами Spark являются высокая производительность, особенно по сравнению с MapReduce, и поддержка четырёх языков программирования - Scala, Java, Python и R. Если при выполнении программы Spark хранит данные в оперативной памяти, то ускорение по сравнению с Hadoop может достигать 100 раз. Spark может запускаться на кластерах ApacheHadoop, на отдельном кластере или на облачных платформах и может обращаться к различным источникам данных, таким как HDFS, ApacheCassandra, ApacheHbase или же AmazonS3.

Фреймворк Spark состоит из нескольких компонент. Модуль SparkCore является основной фреймворка. Он обеспечивает распределенную диспетчеризацию, планирование и базовые функции ввода-вывода. SparkSQL использует структуру данных, называемую DataFrames и выступает в качестве распределенного механизма запросов SQL, что позволяет ускорять выполнение запросов HadoopHive в 100 раз. SparkStreaming является инструментом для обработки потоковых данных. Модуль Mlib позволяет строить модели машинного обучения. GraphX используется для работы с данными, представленными в виде графа.

5 Выбор технологий для обработки данных и построения архитектуры приложения

Для обработки крупноформатных данных использовался облачный сервис AmazonAWS, который имеет лучшее соотношение функциональности и цены по сравнению с конкурентами. В AmazonAWS были задействованы модули AmazonEMR и AmazonS3. Первоначально пользователь загружает данные в облачное хранилище AmazonS3, в котором они могут храниться при отключенном кластере. Затем данные передаются в вычислительный кластер Spark, который выполняет их обработку. Готовый результат передаётся в хранилище AmazonS3, в котором будут храниться уже обработанные данные.

6 Поиск шаблонов и разработка рекомендательных систем

Для апробации выбранной технологии исследовались данные о продуктовых покупках пользователей одного из крупных онлайновых ритейлеров. Решались следующие задачи:

- обнаружение пиковых дней недели;
- анализ загруженности в течение дня;
- нахождение популярных категорий товаров;
- анализ спроса на различные группы товаров по дням недели;
- анализ частоты покупок;
- выявление зависимостей между первым и последующими заказами;
- нахождение популярных товаров при первом и последующих заказах;
- анализ изменения предпочтений клиентов с течением времени;
- анализ применения построенных рекомендательных систем на примере нескольких людей/товаров.

Анализ спроса на продукты в разные дни недели показал (рисунок 4), что воскресенье и понедельник являются пиковыми днями. Со вторника по четверг идёт значительное снижение спроса, а в пятницу и субботу виден небольшой подъём.

Рисунок 4 – Распределение спроса на доставку продуктов по дням недели

Был изучен спрос на продукты в течение дня и выявлены пиковые часы. Частота покупок достигает наибольшей величины в период с 8:00 до 18:00 часов как показано на рисунке 5.

На рисунке 6 отображён спрос на различные группы товаров по дням недели.

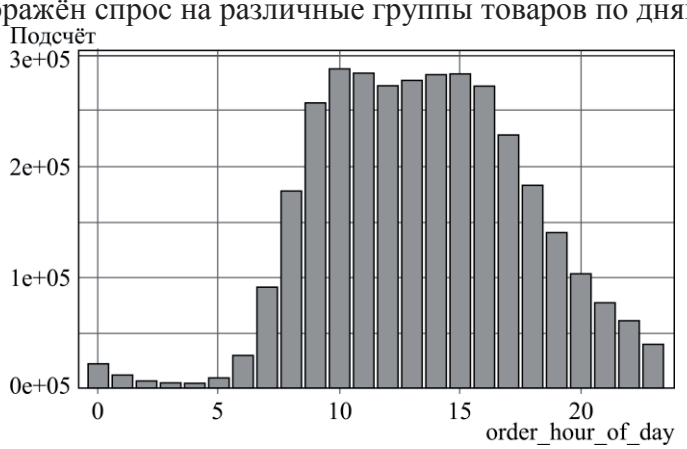

Рисунок 5 – Распределение спроса на доставку продуктов в течение суток

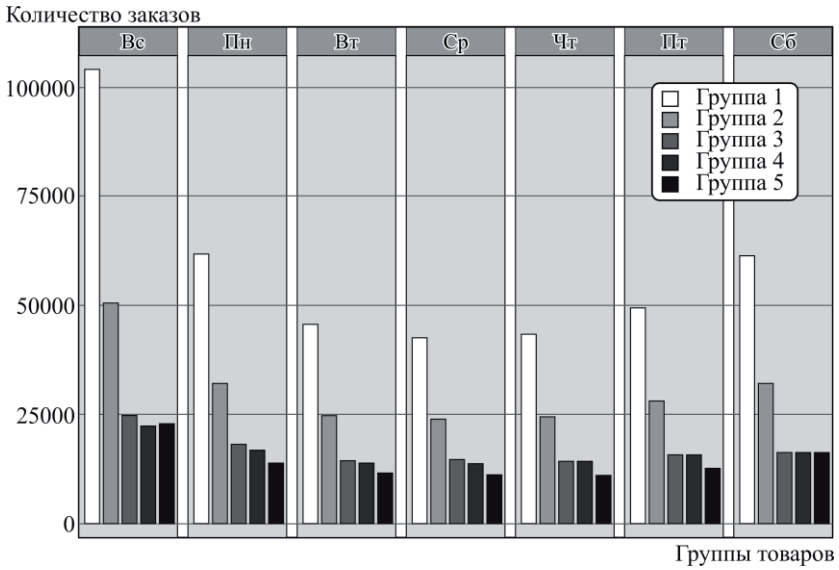

Рисунок 6 – Спрос на различные группы товаров по дням недели

Здесь 1-й столбец гистограммы (группа 1) означает овощи и фрукты, 2-й - молочные продукты, 3-й - чипсы и печенье, 4–й - напитки и чай, 5-й - замороженные продукты. Анализ гистограммы показывает, что в воскресенье присутствует повышенный спрос на свежие фрукты и овощи, а также молочную продукцию. Объём продаж других групп товаров находится в течение недели примерно на одном уровне. Данная информация может быть полезна отделу логистики компании при планировании закупок.

Выявлены дни недели, в которые ожидается наибольший рост новых клиентов. Рисунок 7 показывает, что график прихода новых пользователей имеет те же пиковые дни, что и график с общим числом заказов.

Анализ популярности категорий продуктов, которые покупаются при первом заказе, может помочь более точно настроить рекомендательную систему. Возможно, клиенты не готовы сразу покупать скоропортящиеся товары и хотят опробовать сервис, заказав что-то из бакалеи или напитков.

Однако рисунок 8 показывает, что рейтинг товаров в первом заказе практически идентичен общей популярности товаров по дням недели. Отсюда можно сделать вывод, что сделан-

ное предположение, о котором говорят некоторые маркетологи, неверно. Пользователи при первом заказе готовы выбирать из всего ассортимента товаров, не боясь, что им привезут плохо выбранные фрукты или испорченную молочную продукцию. Следовательно, в рекомендации новым пользователям стоит включать эти группы товаров.

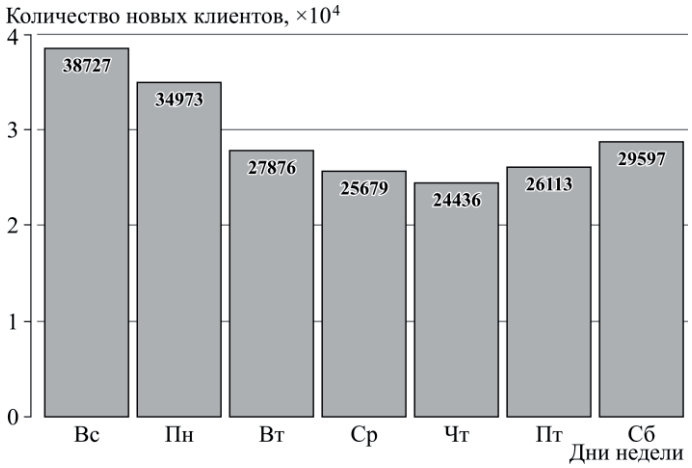

Рисунок 7 – График прихода новых клиентов по дням недели

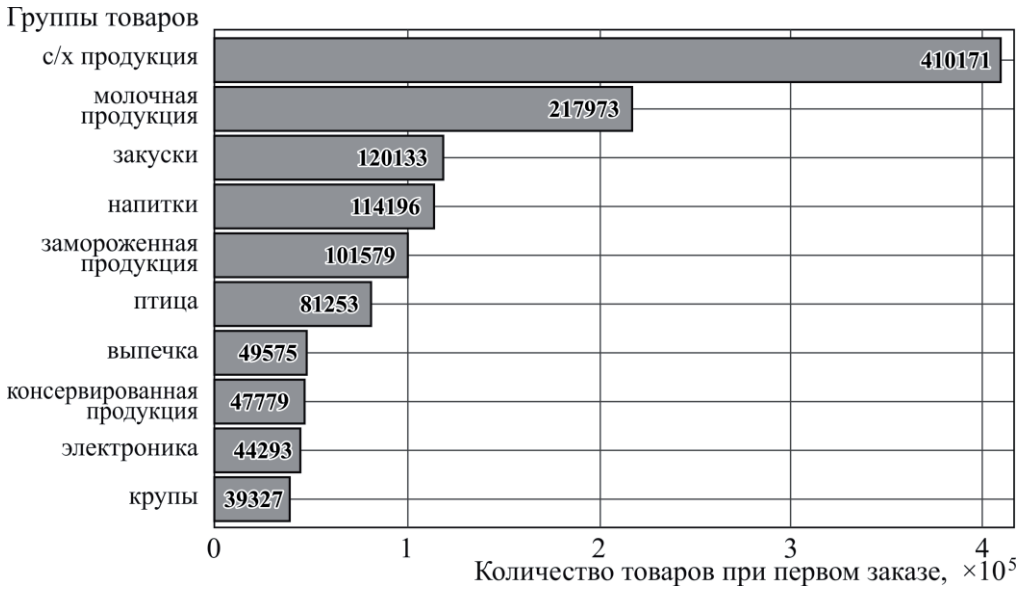

Рисунок 8 – Рейтинг групп товаров при первом заказе

Выяснялось, с какой периодичностью клиенты делают повторные покупки. Эта информация окажется полезной при проведении рекламных кампаний. Рисунок 9 показывает, что наиболее часто люди покупают с периодичностью 7 и 30 дней. Также присутствует большая группа клиентов, делающих повторный заказ спустя период от 3 до 6 дней.

Можно сделать вывод, что люди, покупающие продукты в воскресенье, понедельник и пятницу, чаще остальных делают повторную покупку через неделю. Люди, покупающие во вторник, среду и четверг, чаще других могут вернуться к покупкам раньше недельного срока. Отметим, что покупки спустя месяц показали наиболее стабильный результат. Вероятно, это можно объяснить периодичностью получения заработной платы.

Частота повторных покупок по дням недели показана на рисунке 10.

Рисунок 11 показывает распределение количества товаров в одном заказе. Как видно из графика, наиболее часто встречаются наборы от 5 до 10 товаров.

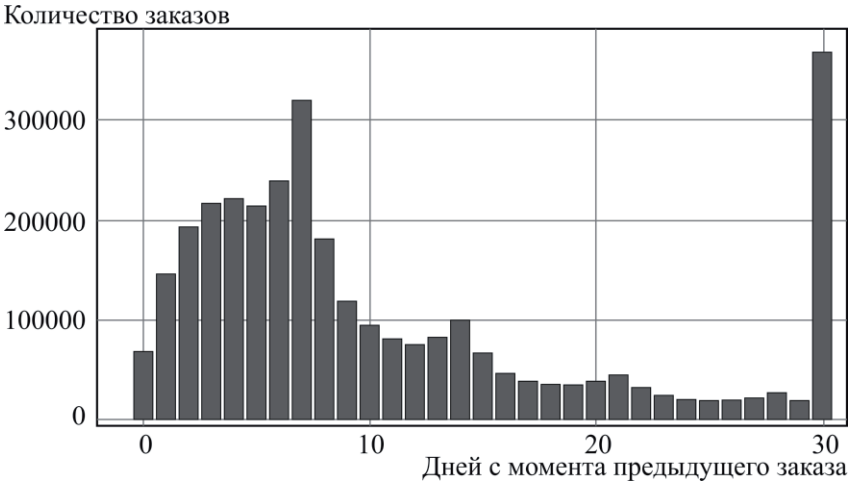

Рисунок 9 – Распределение повторных покупок по дням в течение месяца

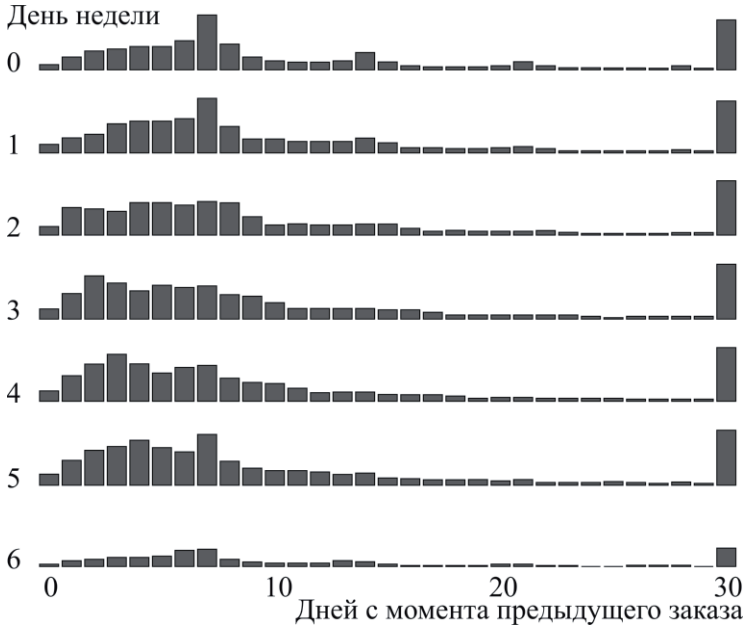

Рисунок 10 – Зависимость частоты повторных покупок от дня недели

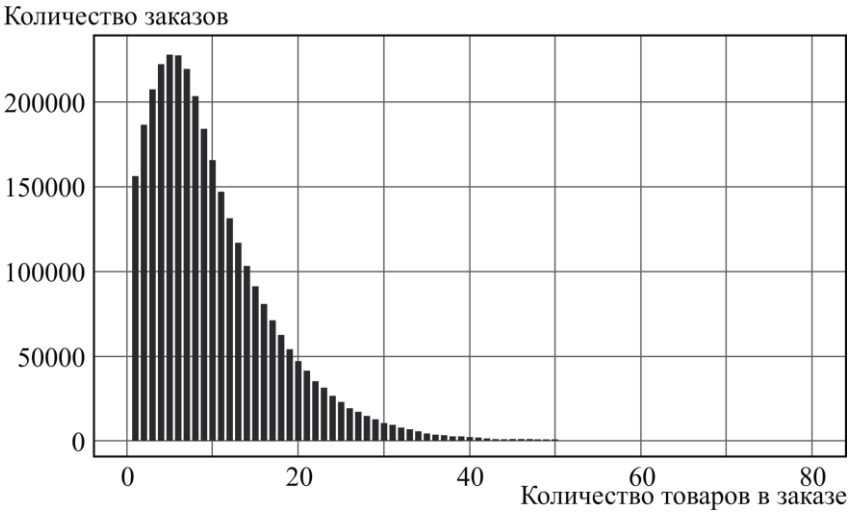

Рисунок 11 – Распределение товаров в заказе

Рисунок 12 показывает распределение спроса на различные продукты. С большим отрывом лидируют фрукты. В частности, бананы занимают два первых места. Эта информация может быть использована при составлении рекомендаций для новых пользователей, о которых первоначально ничего неизвестно.

Получена информация, как часто меняется выбор клиентов при повторном заказе. В частности, рисунок 13 показывает, что более 50% людей изменяют свою продуктовую корзину при последующих заказах. Это стимулирует строить более разнообразные рекомендации в том числе для клиентов, с устоявшимися вкусами.

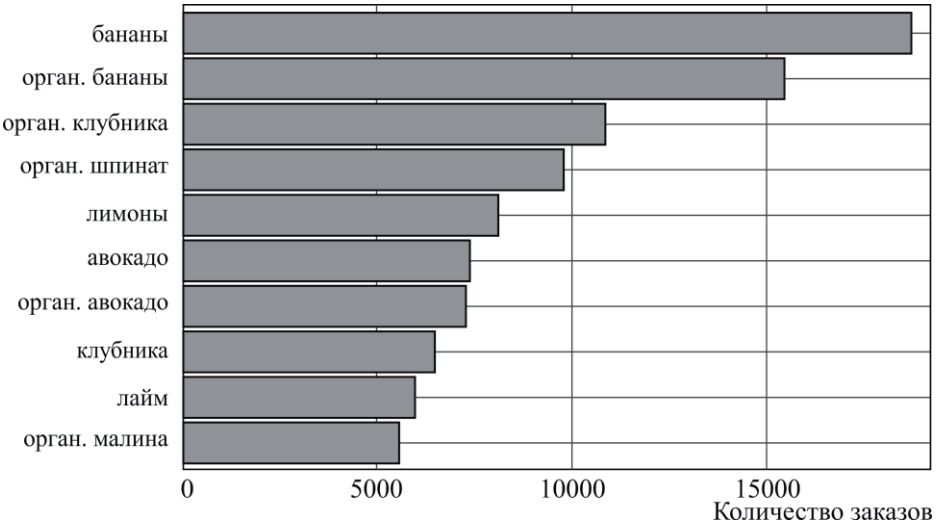

Рисунок 12 – 10 наиболее популярных продуктов

Распределение наиболее популярных продуктов среди повторных покупок (рисунок 14) показало, что лидирующие бананы опустились на шестую строчку, а остальные фрукты отсутствуют в распределении. Клиенты больше заказывают молочную продукцию и напитки. Было выявлено, что чем меньше проходит времени с момента первой покупки, тем больше вероятность изменения состава товаров в новом заказе. Установлено, что повторные покупки, осуществляемые спустя 30 дней, меньше отличаются от первой.

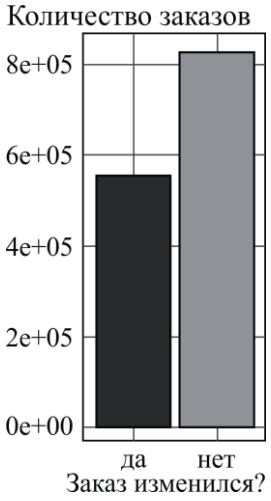

Рисунок 13 – Изменения продуктовой корзины при повторном заказе

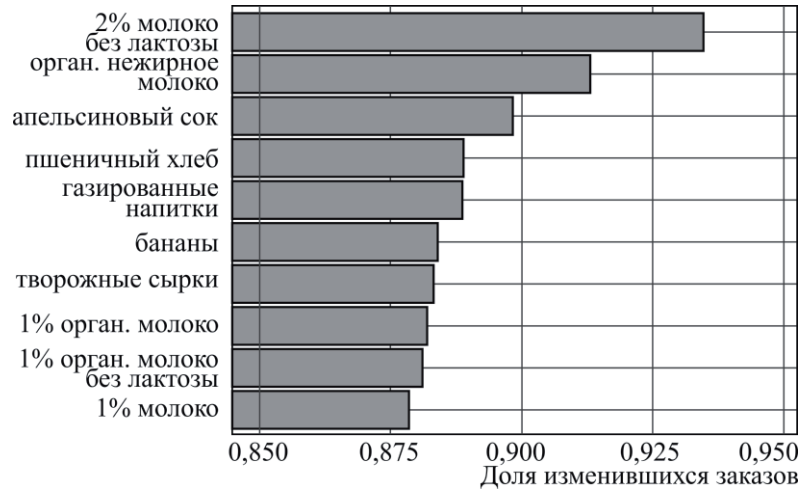

Рисунок 14 – Рейтинг популярности продуктов при повторных покупках

7 Сравнение алгоритмов построения рекомендательных систем

В вычислительных экспериментах сравнивались два алгоритма построения рекомендательных систем – ALS и SVD. В качестве исходных данных обрабатывались различные наборы пользовательских корзин - от 100 тысяч до 3 миллионов. Время построения рекомендательной системы для каждого набора данных показано на рисунке 15.

Из приведённых на рисунке 15 сведений можно сделать вывод, что при малом объёме данных оба алгоритма показывают примерно одинаковый результат, однако с увеличением объёма исходных данных алгоритм ALS (- - -) работает быстрее, чем алгоритм SVD (----). Таким образом, для интеграции с онлайн-магазином и обновления информации в режиме реального времени [14] алгоритм ALS выглядит предпочтительней.

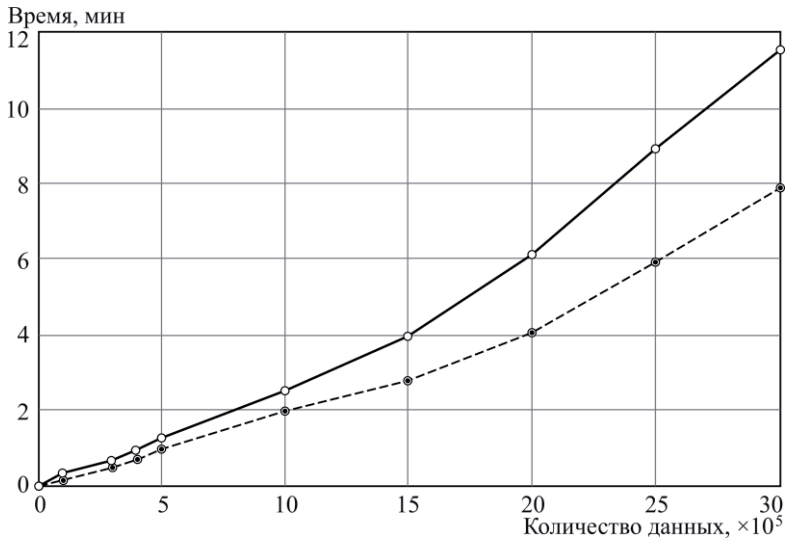

Рисунок 15 – Время построения рекомендательной системы для различных объёмов исходных данных

Для оценки качества рекомендаций принята широко используемая метрика RMSE [2, 3]. Результат исследования на тестовом наборе данных представлен в таблице 1. Видно, что ALS уступает в точности SVD. При этом изменение количества данных незначительно влияет на значение RMSE. Также алгоритм SVD показывает более точные результаты при малых объёмах исходных данных. Таким образом, при построении рекомендательной системы в пакетном режиме предпочтительным является алгоритм SVD. Для обработки данных в режиме реального времени имеет смысл выбирать алгоритм ALS.

Таблица 1 – Оценка качества полученных рекомендаций

Количество данных	RMSE (алгоритм ALS)	RMSE (алгоритм SVD)
500 000	1,59854	1,29951
1 000 000	1,59815	1,29947
1 500 000	1,59793	1,29945
2 000 000	1,59792	1,29945
2 500 000	1,59791	1,29943
3 000 000	1,59787	1,29941

Заключение

В работе построена аппаратно-программная платформа на облачных веб-сервисах Amazon, которая позволяет обрабатывать большие объёмы данных. Произведено отображение

алгоритмов ALS и SVD на архитектуру вычислительного кластера в составе облачного сервиса и выполнено их сравнение. Модульность разработанного решения позволяет при необходимости доработать его под новые требования. Произведён анализ данных одного из крупных онлайновых ритейлеров и выявлен ряд зависимостей, которые могут быть использованы при построении эффективных рекомендательных систем.

Благодарность

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 17-47-630662).

Список источников

[1] Что такое Big data: собрали все самое важное о больших данных. Rusbase. - https://rusbase/howto/chto-takoe-big-data.
[2] ***Bell, R.*** Matrix factorization techniques for recommender systems / R. Bell, Y. Koren, C. Volinsky // Computer. – 2009. – Vol. 42, Issue 8. – P. 30-37. – DOI: 10.1109/MC.2009.263.
[3] ***Wang, W.*** Using singular value decomposition approximation for collaborative filtering / W. Wang, J. Pearlman, S. Zhang, J. Ford, F. Makedon // Seventh IEEE International Conference on E-Commerce Technology (CEC'05)(CEC), Munich, Germany. – 2005. – P. 257-264. – DOI: 10.1109/ICECT.2005.102.
[4] ***Bokde, D.*** Matrix factorization model in collaborative filtering algorithms: A survey / D. Bokde, S. Girase, D. Mukhopadhyay // Procedia Computer Science. – 2015. – Vol. 49. – P. 136-146.
[5] ***Вячитов, Д.*** Data Mining – интеллектуальный анализ данных. Компания iTeam / Д. Вячитов. – http://iteam.ru/publications/it/section_92/article_1448.
[6] ***Гусев, П.*** Введение в ассоциативные правила. DataReview информационный портал / П. Гусев. – https://datareview.info/article/vvedenie-v-associativnie-pravila.
[7] Recommender systems 101. – https://d4datascience.wordpress.com/2016/07/22/recommender-systems-101/comment-page-1/.
[8] ***Чубакова, И.А.*** Data Mining / И.А. Чубакова. – М.: Бином, 2008. – 324 с.
[9] Технологии анализа данных: DataMining, VisualMining, TextMining, OLAP / А.А. Барсегян [и др.]. – М.: БХВ-Петербург, 2007. – 384 с.
[10] ***Ho, B.*** Advanced Recommendation Engines / B. Ho. – https://bigr.io/advanced-recommendation-engines/.
[11] ***Шалаев, С.*** Эволюция рекомендательных сервисов. Firma портал / С. Шалаев. – https://firma.ru/data/articles/5006.
[12] ***Nee, D.*** Collaborative filtering using alternating least squares / D. Nee. – http://danielnee.com/art/1999.
[13] Introduction to recommendation engine. – https://dataaspirant.com/2015/01/24.
[14] ***Манн, К.*** Распределённые вычисления с помощью Hadoop. IBM developer works / К. Манн. – https://www.ibm.com/developerworks/ru/library/l-hadoop.
[15] ***Пигенский, К.*** Знакомство с Apache Spark. Хабрхабр / К. Пигенский. – https://habr.ru/blog/276675.

COMPARATIVE STUDY OF THE ALGORITHMS OF DESIGN OF RECOMMENDATION SYSTEMS BASED ON THE ANALYSIS OF BIG DATA ON CONSUMER BASKETS

I.A. Olyanich[1,a], P.G. Serafimovich[1,2,b]

[1]*Samara National Research University named after academician S.P. Korolev, Samara, Russia*
[a]*14124123@mail.ru,*
[2]*Image Processing Systems Institute of the RAS, Samara, Russia*
[b]*serafimovich.pg@ssau.ru*

Abstract

The article describes the algorithms for building recommender systems based on the analysis of data on grocery purchases of users of one of the largest online retailers. Using modern methods of data storage and analysis, effective recommender systems allow, in particular, to form the customers' interest and to increase the value of the average bill in individual orders. The article describes a built-in hardware and software system on Amazon cloud web services. Using this system, the initial data set was studied, typical examples of recommendations were constructed and the ALS and SVD algorithms were compared. We considered the use of frameworks Apache Hadoop and Apache Spark for the analysis of large format data on consumer baskets. The article analyzes the peak days of the week and workload during the day. Found popular product categories. We studied the demand for various groups of goods by day of the week and the frequency of purchases. The dependencies between the first and subsequent orders, popular products for the first and subsequent orders, and also changes in customer preferences over time were identified.

Key words: *design of recommendation systems, Hadoop, Apache, collaborative filtering, matrix factorization, alternating least squares, singular-value decomposition.*

Citation: *Olyanich IA, Serafimovich PG.* Comparative study of the algorithms of design of recommendation systems based on the analysis of big data on consumer baskets [In Russian]. *Ontology of designing*. 2018; 8(4): 628-640. DOI: 10.18287/2223-9537-2018-8-4-628-640.

Acknowledgment

This work was carried out with the financial support of the Russian Foundation for Basic Research (grant No. 17-47-630662).

References

[1] What is Big data: collected all the most important about big data. Rusbase [In Russian]. - https://rusbase/howto/chto-takoe-big-data.
[2] ***Bell R, Koren Y, Volinsky C.*** Matrix factorization techniques for recommender systems. Computer 2009; 42(8): 30-37. - DOI: 10.1109/MC.2009.263.
[3] ***Wang W, Pearlman J, Zhang S, Ford J, Makedon F.*** Using singular value decomposition approximation for collaborative filtering. Seventh IEEE International Conference on E-Commerce Technology (CEC'05)(CEC), Munich, Germany 2005: 257-264. - DOI: 10.1109/ICECT.2005.102.
[4] ***Bokde D, Girase S, Mukhopadhyay D.*** Matrix factorization model in collaborative filtering algorithms: A survey, Procedia Computer Science 2015; 49: 136-146.
[5] ***Vyachitov D.*** Data Mining – data mining. ITeam Company [In Russian]. - http://iteam.ru/publications/it/section_92/article_1448.
[6] ***Gusev P.*** Introduction to associative rules. DataReview information portal [In Russian]. - https://datareview.info/article/vvedenie-v-associativnie-pravila.
[7] Recommender systems 101. – https://d4datascience.wordpress.com/2016/07/22/recommender-systems-101/comment-page-1/.
[8] ***Chubakova IA.*** Data Mining [In Russian]. Moscow: “Binom” Publ.; 2008.

[9] ***Barsegyan AA,*** [et al.]. Data analysis technologies: DataMining, VisualMining, TextMining, OLAP [In Russian]. Moscow: "BHV-Petersburg" Publ.; 2007.
[10] ***Ho, B.*** Advanced Recommendation Engines / B. Ho. – https://bigr.io/advanced-recommendation-engines/.
[11] ***Shalaev S.*** Evolution of recommendatory services. Firma portal [In Russian]. - https://firma.ru/data/articles/5006.
[12] ***Nee D.*** Collaborative filtering using alternating least squares. - http://danielnee.com/art/1999.
[13] Introduction to recommendation engine. - https://dataaspirant.com/2015/01/24.
[14] ***Mann K.*** Distributed computing using Hadoop. IBM developer works. [In Russian].- https://www.ibm.com/developerworks/ru/library/l-hadoop.
[15] ***Pigensky K.*** Acquaintance with Apache Spark. Habrahabr [In Russian]. - https://habr.ru/blog/276675.

Сведения об авторах

Олянич Игорь Анатольевич, 1994 г. рождения. Аспирант Самарского национального исследовательского университета им. академика С.П. Королева.
Igor Anatolievich Olyanich (b. 1994) postgraduate of Samara University named after academician S.P. Korolev.

Серафимович Павел Григорьевич, окончил Куйбышевский авиационный институт им. С.П. Королёва в 1989 г., д.ф.-м.н. (2016). Доцент кафедры технической кибернетики Самарского университета, старший научный сотрудник Института систем обработки изображений РАН. В списке научных трудов более 50 работ в области проектирования программных средств распределённой и параллельной обработки крупноформатных данных, методов машинного обучения, моделирования и проектирования устройств микрооптики.
Pavel Grigorievich Serafimovich graduated from the Korolyov aerospace Institute (Kuibyshev-city) in 1989, D. Sc. Phys.-Math. (2016). He is an assistant professor of technical cybernetics department of Samara University and senior researcher at the Image Processing Systems Institute of the RAS. He is an author and co-author of more than 50 publications in the field of parallel and distributed data-intensive computing, machine learning, and design of micro-optic devices.