МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ НАХОЖДЕНИЯ ПАТТЕРНОВ В УПОРЯДОЧЕННОМ МНОЖЕСТВЕ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 412 группы
направления 01.03.02 — Прикладная математика и информатика

механико-математического факультета
Богоявленского Виталия Георгиевича

Научный руководитель
Зав. кафедрой, д. ф.-м. доцент _____ С. П. Сидоров

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2024

Введение.

В данной работе рассматриваются вопросы нахождения закономерных последовательностей элементов в упорядоченных множествах путем анализа простых алгоритмов поиска паттернов и создания собственного алгоритма для нахождения закономерных последовательностей данных.

Актуальность.

Понятие "паттерн"широко используется в финансах, экономике, техническом анализе, прогностической аналитике, медицине, криминалистике и некоторых других предметных областях.

1. как сущность явления, имеющего повторяющиеся черты;
2. как свойство повторяющихся компонентов, объединенных общей структурой;
3. как процесс, фиксирующий модель взаимодействия изучаемых объектов, включающего повторения.

Во всех этих предметных областях паттерны данных могут использоваться для выделения групп схожих объектов и изучения их ключевых характеристик с проведением кластерного анализа для разбиения всех объектов выборки на непересекающиеся кластеры для формирования их классификации. Паттерн обозначает некую выявленную закономерность в данных или некую шаблонную структуру данных.

Целью данной работы является изучение основных моделей поиска паттернов, включая такие алгоритмы, как Apriori и Последовательные шаблоны, а также разработка собственного алгоритма поиска паттернов. После разработки нового алгоритма необходимо провести его проверку на подготовленных данных и сравнить результаты с рассматриваемыми моделями.

Структура бакалаврской работы

Работа состоит из введения, трех разделов, заключения и приложения.

В первом разделе даны основные понятия о паттернах и понятии получения паттернов, также указаны теоретические основы основных рассматриваемых в работе алгоритмов. Во втором разделе представлен алгоритм построения разработанного подхода по поиску паттернов и обоснование его применения в рассматриваемой задаче. В третьем разделе проведена разработка представленных алгоритмов на языке Python. Представленн анализ

поиска паттернов на реальных данных о фильмах за 2010 - 2020 года.

Основное содержание работы

Теоритические основы

Понятие паттерн широко используется в финансах, экономике, техническом анализе, прогнозной аналитике, медицине, криминалистике и некоторых других предметных областях. Что характерно, он определяется в этих областях знания по-разному:

1. как сущность явления, имеющего повторяющиеся черты;
2. как свойство повторяющихся компонентов, объединенных общей структурой;
3. как процесс, фиксирующий модель взаимодействия изучаемых объектов, включающего повторения.

Во всех этих предметных областях паттерны данных могут использоваться в смысле выделения групп схожих объектов и изучения их ключевых характеристик с проведением кластерного анализа для разбиения всех объектов выборки на непересекающиеся кластеры для формирования их классификации. В качестве паттерна обозначается некая выявленная закономерность в данных или некая шаблонная структура данных.

Рассмотрим формальную постановку задачи : Пусть $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ — набор литералов, называемых элементами. Пусть $\mathcal{D}$ — набор транзакций, где каждая транзакция T — это набор элементов, таких, что $T \subseteq \mathcal{I}$. С каждой транзакцией связан уникальный идентификатор, называемый ее TID. Мы говорим, что транзакция T содержит X, набор некоторых элементов из $\mathcal{I}$, если $X \subseteq T$. Правило ассоциации — это следствие формы $X \Longrightarrow Y$, где $X \subset \mathcal{I}, Y \subset \mathcal{I}$ и $X \cap Y = \emptyset$. Правило $X \Longrightarrow Y$ выполняется в множестве транзакций $\mathcal{D}$ с уверенностью c, если $c\%$ транзакций в $\mathcal{D}$, содержащих X, также содержат Y. Правило $X \Longrightarrow Y$ имеет поддержку s в множестве транзакций $\mathcal{D}$, если $s\%$ транзакций в $\mathcal{D}$ содержат $X \cup Y$.

Учитывая набор транзакций $\mathcal{D}$, проблема правил ассоциации интеллектуального анализа данных состоит в том, чтобы сгенерировать все правила ассоциации, поддержка и достоверность которых превышают заданную пользователем минимальную поддержку и минимальную достоверность соответственно. Полученное обобщение нейтрально по отношению к представлению

$\mathcal{D}$. Например, $\mathcal{D}$ может быть файлом данных, реляционной таблицей или результатом реляционного выражения [1].

Модель Apriori

Алгоритм Apriori - это алгоритм, используемый в машинном обучении для извлечения часто встречающихся наборов элементов из большого набора данных. Этот алгоритм основан на принципе, что все подмножества часто встречающегося набора элементов также должны быть часто встречающимися.

Генерация кандидатов Apriori

Функция apriori-gen принимает в качестве аргумента L_{k-1}, множество всех больших $(k-1)$-элементных наборов. Она возвращает супермножество множества всех больших k-элементных наборов. Функция работает следующим образом. Сначала, на этапе объединения, соединяются L_{k-1} и L_{k-1} :

1) вставить в C_k
2) из $L_{k-1}p, L_{k-1}q$
3) p. элемент $_{k_{k-1}} < q$. элемент $_{k_{k-1}}$.

Затем, на этапе обрезки, удаляются все наборы элементов $c \in C_k$, такие что некоторое $(k-1)$-подмножество c не находится в L_{k-1}.

Пример Пусть L_3 - это $\{\{1,2,3\},\{1,2,4\},\{1,3,4\},\{1,3,5\},\{2,3,4\}\}$. После этапа объединения, C_4 будет $\{\{1,2,3,4\},\{1,4,5\}\}$. Этап обрезки удалит некоторые наборы элементов. Затем останется только набор $\{1,2,3,4\}$ в C_4 [1].

Корректность

Требуется показать, что $C_k \supseteq L_k$. Очевидно, что любое подмножество большого набора элементов также должно иметь минимальную поддержку. Если расширить каждый набор элементов в L_{k-1} всеми возможными элементами, а затем удалили все те, чьи $(k-1)$-подмножества не находятся в L_{k-1}, оостанется супермножество наборов элементов в L_k.

Кандидатские наборы элементов C_k хранятся в хеш-дереве. Узел хеш-дерева может содержать либо список наборов элементов (листовой узел), ли-

бо хеш-таблицу (внутренний узел). В внутреннем узле каждый ящик хеш-таблицы указывает на другой узел. Корень хеш-дерева определен на уровне 1. Внутренний узел на глубине d указывает на узлы на глубине $d+1$. Наборы элементов хранятся в листьях. Когда добавляется набор элементов c, начинается спуск с корня по дереву, пока не достигается лист. В узле глубины d принимается решение, по какой ветке двигаться дальше, применяя хеш-функцию к d-му элементу набора. Все узлы изначально создаются как листовые узлы. Когда количество наборов элементов в листовом узле превышает определенный порог, листовой узел преобразуется во внутренний узел [2].

Ограничения

Apriori имеет ряд неэффективностей или компромиссов, которые породили другие алгоритмы. Генерация кандидатов генерирует большое количество подмножеств (алгоритм пытается загрузить набор кандидатов с как можно большим количеством подмножеств перед каждым сканированием базы данных). Исследование подмножества снизу вверх (по сути, обход решетки подмножеств в ширину) находит любое максимальное подмножество S только после того, как все $2^{|s|}-1$ его собственных подмножеств.

Алгоритм сканирует базу данных слишком много раз, что снижает общую производительность. Благодаря этому алгоритм предполагает, что база данных постоянно находится в памяти.

Кроме того, временная и пространственная сложность этого алгоритма очень высока: $O\left(2^{|D|}\right)$, где $|D|$ — ширина по горизонтали (общее количество элементов), присутствующих в базе данных.

Модель последовательные шаблоны: PrefixSpan Поскольку элементы внутри последовательности могут перечисляться в любом порядке, без потери общности, можно предположить, что они всегда перечисляются в алфавитном порядке. Например, последовательность в S перечислена как $\langle a(abc)(ac)d(cf)\rangle$ вместо $\langle a(bac)(ca)d(fc)\rangle$. С таким соглашением выражение последовательности является уникальным [3].

Затем исследуется, можно ли зафиксировать порядок проекции элементов при генерации проецированной базы данных. Интуитивно, если следовать порядку префикса последовательности и проецировать только суффикс последовательности, можно последовательно рассмотреть все возможные под-

последовательности и их соответствующие проецированные базы данных. Таким образом, сначала вводим понятие префикса и суффикса.

Определение 1 (Префикс). Предположим, что все элементы в элементе перечислены в алфавитном порядке. Дана последовательность $\alpha = \langle e_1 e_2 \cdots e_n \rangle$ (где каждый e_i соответствует часто встречающемуся элементу в S), последовательность $\beta = \langle e'_1 e'_2 \cdots e'_m \rangle$ $(m \leq n)$ называется префиксом α, если и только если 1) $e'_i = e_i$ для $(i \leq m-1)$; 2) $e'_m \subseteq e_m$; и 3) все частые элементы в $(e_m - \bar{e}'_m)$ расположены в алфавитном порядке после тех в e'_m.

Например, $\langle a \rangle, \langle aa \rangle, \langle a(ab) \rangle$ и $\langle a(abc) \rangle$ являются префиксами последовательности $s = \langle a(abc)(ac)d(cf) \rangle$, но ни $\langle ab \rangle$, ни $\langle a(bc) \rangle$ не рассматриваются как префиксы, если каждый элемент в префиксе $\langle a(abc) \rangle$ последовательности s является частым в S.

Определение 2 (Суффикс). Дана последовательность $\alpha = \langle e_1 e_2 \cdots e_n \rangle$ (где каждый e_i соответствует часто встречающемуся элементу в S). Пусть $\beta = \langle e_1 e_2 \cdots e_{m-1} e'_m \rangle$ $(m \leq n)$ будет префиксом α. Последовательность $\gamma = \langle e''_m e_{m+1} \cdots e_n \rangle$ называется суффиксом α относительно префикса β, обозначается как $\gamma = \alpha/\beta$, где $e''_m = (e_m - e'_m)$ [2] Также обозначаем $\alpha = \beta \cdot \gamma$. Обратите внимание, если β не является подпоследовательностью α, суффикс α относительно β пуст. Например, для последовательности $s = \langle a(abc)(ac)d(cf) \rangle$, $\langle (abc)(ac)d(cf) \rangle$ является суффиксом относительно префикса $\langle a \rangle$, $\langle (\_bc)\,(ac)d(cf) \rangle$ является суффиксом относительно префикса $\langle aa \rangle$, и $\langle (-c)(ac)d(cf) \rangle$ является суффиксом относительно префикса $\langle a(ab) \rangle$.

На основе понятий префикса и суффикса задача поиска последовательных шаблонов может быть разложена на набор подзадач.

Лемма 3.1 (Разделение задачи). 1. Пусть $\{\langle x_1 \rangle, \langle x_2 \rangle, \ldots, \langle x_n \rangle\}$ будет полным набором последовательных шаблонов длины 1 в базе данных последовательностей S. Полный набор последовательных шаблонов в S можно разделить на n непересекающихся подмножеств. i-е подмножество $(1 \leq i \leq n)$ является набором последовательных шаблонов с префиксом $\langle x_i \rangle$. 2. Пусть α будет последовательным шаблоном длины l и $\{\beta_1, \beta_2, \ldots, \beta_m\}$ будет набором всех последовательных шаблонов длины $(l+1)$ с префиксом α. Полный набор последовательных шаблонов с префиксом α, кроме самого α, можно разделить на m непересекающихся подмножеств. j-е подмножество $(1 \leq j \leq m)$

является набором последовательных шаблонов с префиксом β_j.

Определение 3 (Проецированная база данных). Пусть α будет последовательным шаблоном в базе данных последовательностей S. Проецированная база данных α, обозначаемая как $S|_{\alpha'}$, является совокупностью суффиксов последовательностей в S относительно префикса α.

Определение 4 (Поддержка подсчета в проецированной базе данных). Пусть α будет последовательным шаблоном в базе данных последовательностей S, а β будет последовательностью с префиксом α. Поддержка подсчета β в проецированной базе данных α $S|_{\alpha}$, обозначается как support $_{S|_0}(\beta)$, является количеством последовательностей γ в $S|_{\alpha}$, так что $\beta \sqsubseteq \alpha \cdot \gamma$.

Лемма 3.2 (Проецированная база данных). Пусть α и β будут двумя последовательными шаблонами в базе данных последовательностей S, так что α является префиксом β. 1. $S|_{\beta} = (S|_{\alpha})|_{\beta}$, 2. для любой последовательности γ с префиксом α, поддержка (γ) = support $_{S|_a}(\gamma)$, и 3. размер проецированной базы данных α не может превышать размер S. Набросок доказательства. Первая часть леммы основывается на том факте, что для последовательности γ, суффикс γ относительно β, γ/β, равен последовательности, полученной в первую очередь из проекции γ относительно α, т.е. γ/α, а затем проекции γ/α относительно β. Это означает, что $\gamma/\beta = (\gamma/\alpha)/\beta$.

Алгоритм 1 (PrefixSpan)

Последовательный анализ шаблонов с прогнозированием префиксов. Входные данные: база данных последовательностей S и минимальный порог поддержки min_support.

Результат: Полный набор последовательных шаблонов. Метод: вызов PrefixSpan $(\langle\rangle, 0, S)$. Подпрограмма PrefixSpan $(\alpha, l, S|_{\alpha})$ Параметры: 1) α — последовательный шаблон; 2) l — длина α; и 3) $S|_{\alpha}$ — это база данных α-проецирования, если $\alpha \neq \langle$, в противном случае — это база данных последовательностей S. Метод: 1. Просканируйте $S|_{\alpha}$ один раз, найдите каждый часто встречающийся элемент b, такой что (а) b можно объединить с последним элементом α, чтобы сформировать последовательный шаблон; или (б) $\langle b\rangle$ можно добавить к α для формирования последовательного шаблона. 2. Для каждого часто встречающегося элемента b добавьте его к α, чтобы сформировать последовательный шаблон α', и выведите α'. 3. Для каждого α' создать

α'-проецируемую базу данных $S|_{\alpha'}$ и вызвать PrefixSpan $(\alpha', l+1, S|_{\alpha'})$.

Основной минус PrefixSpan — создание проектируемых баз данных. В худшем случае PrefixSpan создает прогнозируемую базу данных для каждого последовательного шаблона. Если существует большое количество последовательных шаблонов, стоимость нетривиальна. Методы сокращения количества проектируемых баз данных будут обсуждаться в следующем подразделе.

Теорема 3.1 (PrefixSpan) [4]. Последовательность α является последовательным шаблоном тогда и только тогда, когда α является результатом работы PrefixSpan. **Доказательство.** Последовательность $\alpha(l \geq 1)$ длины l идентифицируется как последовательный шаблон с помощью PrefixSpan тогда и только тогда, когда α является последовательным шаблоном в проектируемой базе данных с его длиной $(l-1)$ префикс α^-. Если $l = 1$, префикс длины 0 для α равен $\alpha^- = \langle\rangle$, а проектируемая база данных — это сама S. Итак, α — это последовательный шаблон в S. Если $l > 1$, согласно лемме 3.2, $S|_{\alpha^-}$ — это в точности α^--проецируемая база данных и поддержка $_S(\alpha) =$ поддерживает $_{S|_\alpha}(\alpha)$. Следовательно, если α — последовательный образец в $S|_{\alpha^-}$, то это также последовательный образец в S. Тем самым показывается, что последовательность α является последовательным шаблоном, если так говорит PrefixSpan. Лемма 3.1 гарантирует, что PrefixSpan идентифицирует полный набор последовательных шаблонов в S.

Алгоритм нахождения паттерна в циклических последовательностях

Пусть множество X размера n, которое включает в себя циклические подмножества, обозначаемые как x_i. Эта структура формирует совокупность объектов, каждый из которых имеет свои уникальные свойства и взаимосвязи.

(Определение циклической последовательности) Циклической последовательностью с периодом $T \in \mathbb{N}$ называется бесконечная последовательность x_i, для которой $\forall i \in \mathbb{N}$ выполняется условие $x_{i_j} = x_{i_{j+kT}}$, где $k \in \mathbb{N}$. Это говорит о том, что каждый элемент последовательности повторяется через заранее заданное количество шагов, определенное периодом T.

(Определение Паттерном в циклических последовательностях определяется такая подпоследовательность элементов, которая включает в себя

часто встречающиеся объекты в каждой последовательности рассматриваемого множества призаданном расстоянии.

Матричный алгоритм

1.Введение понятия словарь и алфавит множестваю

Определение (Алфавит). Алфавитом B называеться последовательность всех уникальных элементов множества.

Определение (словарь и слово). Словарем S называется последовательность алфавита B, где каждый элемент B_i присутствует в множестве X чаще, чем α. Здесь α представляет собой пороговое значение, которое указывает, как часто элемент должен встречаться, чтобы быть включенным в словарь.

Слово, в свою очередь, это элемент x_{i_j} циклического подмножества x_i множества X. Этот базовый элемент используется для формирования всех последовательностей.

2.Построение матрицы встречи для данных.

Для словаря S с установленным расстоянием d можно построить матрицу встречи $A = (a_{ij})$, которая показывает частоту встречи слов с введеным расстоянием

В условиях предоставленного алгоритма элементы матрицы смежности определяются следующим образом:

$$a_{ij}^{[l]} = \begin{cases} 1 & \text{, если элементы } S_i \text{ и } S_j \text{ находятся рядом} \\ & \text{и расстояние между ними равно } d \text{ в последовательности } x_l, \\ 0 & \text{, в противном случае.} \end{cases}$$

Здесь d - это расстояние между словами. Для каждого подмножества x_l строятся соответствующие матрицы $A^{[l]}$

Общая матрица встречи и порог отсечения. После получения матриц $A^{[l]}$ производится получение общей матрицы встречи $D = (d_{ij})$ поэле-

ментным сложением компанент $A^{[l]}$,

$$d_{ij} = \sum_{k=1}^{l} a_{ij}^{[l]}$$

При этом диагональные элементы матрицы D будут иметь размер равный длине исследуемого множества циклических последовательностей X. (**Определение** Порогом отсечения называться величина β, которая определяеться процентом от максимального значения элемента D.

Тогда, в D элементы меньше введеного β заменяються 0 и ,соотвесвенно, большие становятся равные 1.

Построение графа

После получения D в которой отсечены малые значения, следует, что данные матрица - это матрица смежности.

(**Определение** Матрица смежности — это один из способов представления графа в виде матрицы, где каждая ячейка матрицы соответствует паре вершин графа. Если между двумя вершинами существует ребро, то в соответствующей ячейке матрицы записывается 1 (или другое значение, указывающее на наличие связи), в противном случае — 0 (или другое значение, указывающее на отсутствие связи). Матрица смежности является симметричной, что означает, что если между вершинами i и j существует ребро, то и между вершинами j и i будет ребро. Для графов без петель и кратных рёбер матрица смежности бинарна, то есть состоит из нулей и единиц. Этот метод представления графа особенно полезен для плотных графов, где число ребер близко к максимально возможному числу ребер (у полного графа) [5].

Далее требуется отыскать максимальную клику полученого графа, что и будет являтся требуемым паттерном множества X с введеным растоянием d

Решение задачи Рассматривается набор данных, размещенный на платформе Kaggle. Данный набор содержит информацию об обзорах популярных фильмов за период с 2010 по 2020 год [6]. Набор данных представляет собой CSV-файл, где информацией служат следующие признаки:

1. 'id': уникальный идентификатор фильма.
2. 'title': Название фильма.
3. 'overview': краткое содержание или синопсис фильма.
4. 'tagline': слоган фильма, часто запоминающаяся фраза или предложение, используемое в рекламных целях.
5. 'genreList': список жанров, к которым принадлежит фильм (например, ['Драма', 'Триллер']).
6. 'directingList': список режиссеров, участвовавших в фильме.
7. «writingList»: список авторов, участвовавших в создании фильма, особенно тех, кто участвовал в написании сценария и сюжета.
8. «castList»: список основных актеров фильма (ограничен пятью актерами с наибольшим количеством участников с точки зрения выставления счетов).
9. 'keywordList': список ключевых слов, связанных с фильмом.
10. 'popularity': популярность фильма по данным TMDB.
11. 'vote_average': средний голос или рейтинг фильма.
12. 'vote_count': количество голосов, полученных фильмом.
13. 'budget': бюджет фильма.
14. 'revenue': доход, полученный от фильма.
15. 'cast_popularity': расчетный показатель, показывающий коллективную популярность основного состава.
16. 'director_popularity': расчетный показатель, показывающий коллективную популярность режиссеров.
17. 'writers_popularity': расчетный показатель, показывающий коллективную популярность писателей.
18. 'recommendationList': список названий фильмов, рекомендуемых на основе рассматриваемого фильма, предоставленный TMDB.
19. 'average_recommendation_popularity': средняя популярность рекомендуемых фильмов.

Задачей работы является поиск нарративного паттерна в отношении ключевых слов фильмов, то есть поиск общего для данных множеств ключевых слов. А также задачей стоит фиксация часто встречающихся жанров фильмов за данный временной отрезок.

Реализация алгоритмов выполнена на языке Python. При реализации кода были использованы следующие библиотеки:

1. *NumPy [7]*, предназначенный для работы с многомерными массивами и математическими операциями над ними;
2. *Pandas [8]*, предоставляет структуры данных для обработки табличных данных, такие как DataFrame и Series;
3. *Matplotlib [9]* используется для визуализации данных с помощью графиков и диаграмм;
4. *networkx [10]*, используемый для работы с графами.
5. *PrefixSpan [11]*, предназначенный для поиска частых последовательных паттернов;
6. *MLxtend (apriori) [12]*, предоставляет алгоритмы для поиска частых элементов и генерации ассоциативных правил

Результаты Apriori

Входные параметры для модели, в контексте исследуемого алгоритма, есть минимальная поддержка равная 0.001, поскольку с такой поддержкой возможно получить списки паттернов длиной больше чем одно слово, и столбцы dataFrame, такие как: 'keywordList', 'genreList', в которых содержаться данные о ключевых словах присущие фильмам и присущим жанрам соответственно.

Анализируя результаты работы алгоритма, можно заметить определенные тенденции, основываясь на столбце, содержащем ключевые слова фильмов. В современном кинематографическом мире наблюдается растущий интерес к экранизированным произведениям - это подтверждают последовательности, содержащие тег "based on novel or book".

Вторым по степени заинтересованности среди пользователей выступает последовательность тегов, в которую входят такие ключевые слова как: "superhero "based on comic"и "marvel cinematic universe". Это может свидетельствовать о постепенном погружении зрителей в мир супергеройского кино, и в частности, киновселенную Marvel. Возможно, данный интерес был подогрет активной рекламной кампанией, связанной с завершением серии фильмов о "Мстителях".

Третий тег, который часто встречается в возможных паттернах, - "dystopia".

Это указывает на интерес авторов и зрителей к фильмам, в которых представлены антиутопические идеи. Скорее всего, зрители привлекаются возможностью увидеть моделирование потенциально неблагоприятного будущего.

Основываясь на полученных результатах анализа столбца жанров, можно сделать вывод о доминировании в кинематографе жанров "Adventure"и "Action". Фильмы, основанные на приключениях с элементами экшена, стали особенно популярны среди зрителей. Это говорит о любви публики к динамичным, наполненным увлекательными событиями и захватывающими моментами сюжетам. Однако, несмотря на очевидное преобладание этих жанров, крайне важно не упускать из виду и другие жанры. Научная фантастика, ужасы, триллеры и фэнтези, часто встречающиеся вместе с тегами "Adventure"и "Action также занимают важное место в кинематографе данного периода.

Результаты PrefixSpan

Входные параметры для модели, в контексте исследуемого алгоритма, есть минимальная поддержка равная 10 и столбцы dataFrame, такие как: 'keywordList', 'genreList', в которых содержаться данные о ключевых словах присущие фильмам и присущим жанрам соотвественно.

После применения алгоритма полученна информация способна дать глубокое понимание того, какие темы и сюжетные линии доминировали в фильмах исследуемого периода.

Анализ полученных паттернов позволил выявить несколько ключевых тематических направлений. В частности, бросается в глаза преобладание тем, основанных на романах или книгах. Это говорит о растущем интересе к экранизациям со стороны зрительской аудитории, а также о желании создателей фильмов перенести бумажные истории на большой экран.

Кроме того, встречаются паттерны, включающие подтексты преступлений, что указывает на популярность криминального жанра. Это свидетельствует о стремлении зрителей к реалистичным сценариям, где главенствуют жестокость и бескомпромиссность реального мира, а не выдуманные мистические существа.

Кроме того, важным элементом становится использование вселенных,

основанных на комиксах. Это говорит о растущем интересе к героям комиксов и их приключениям, что, безусловно, связано с усилиями таких гигантов индустрии, как Marvel и DC.

Результаты работы алгоритма, разработанного для выявления паттернов жанров. Важно отметить, что алгоритм PrefixSpan работает на основе анализа префиксов, то есть начала последовательности. Даже если два паттерна содержат одни и те же элементы, они считаются различными, если их префиксы отличаются. Это позволяет учитывать различные комбинации жанров и предоставляет более полную картину предпочтений зрителей. Предположим, имеются два паттерна: $a = Action, Adventure, ScienceFiction$ с уровнем поддержки 70 и $b = Adventure, Action, ScienceFiction$ с уровнем поддержки 26. Несмотря на то, что оба паттерна состоят из одних и тех же элементов, они считаются различными, поскольку их префиксы (то есть начало последовательности) различаются: у первого паттерна это "Action a у второго - "Adventure".

Результаты матричного алгоритма

Применяя матричный алгоритм описанный , для обработки столбцов, содержащих информацию о ключевых словах и жанрах следует учесть необходимость во множестве состоящим из циклических последовательностях, реализация даннного уловия заключается в том, что достягая конца последовательности вычисления продолжаються начиная с начала рассматриваемой последовательности. Характеристики наших данных накладывают определенные требования на выбор гиперпараметров.

В частности, для столбца ключевых слов из-за широкого разнообразия тегов и их относительно низкой частоты появления параметр α выбран равным 0.4 и $\beta = 0.05$, что обозначает уровень отсечения в 5%. Расстояние для окна поиска задано как $d = 3$. Эти параметры были выбраны на основе оценки распределения данных и тестирования различных вариантов подбора параметров.

С другой стороны, для данных о жанре произведений параметр α установлен равным 5.5, $\beta = 0.2$, а расстояние d также составляет 3. Эти гиперпараметры были определены путем экспериментального подбора, чтобы обеспечить наиболее оптимальное представление паттерна данных, для изу-

чения нарратива.

На рисунках 1 и 2 представлены матрицы смежности для указанных столбцов данных. Эти матрицы дают важную информацию о зависимостях и взаимосвязях между различными элементами в этих столбцах.

На основании полученных матриц смежности были построены графы, представленные на рисунках 3 и 4, где красным цветом помечены вершины вошедшие в максимальную клику и считаются паттерном в смысле циклических множеств, а синим дополнительно частовстречающиеся элементы, которые близки к элементам паттерна. Эти графы представляют собой визуальное представление данных и позволяют наглядно увидеть структуру и взаимосвязи между различными элементами.

based on novel or ...	sequel object	duringcreditssting...	murder object	based on comic ob...	based on true story (	aftercreditsstinger o.	revenge object
1	1	1	1	0	1	0	1
1	1	1	1	1	0	1	1
1	1	1	0	1	0	1	0
1	1	0	1	1	1	0	1
0	1	1	1	1	0	1	1
1	0	0	1	0	1	0	1
0	1	1	0	1	0	1	0
1	1	0	1	1	1	0	1

Рисунок 1 – Матрица встречи для ключевых слов

Drama object	Action object	Comedy object	Thriller object	Adventure object	Family object	Fantasy object	Science Fiction obj...
1	1	1	1	1	0	1	0
1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	0
1	1	0	1	1	0	0	1
1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	0
1	1	1	0	1	1	1	0
0	1	0	1	1	0	0	1

Рисунок 2 – Матрица встречи для жанров

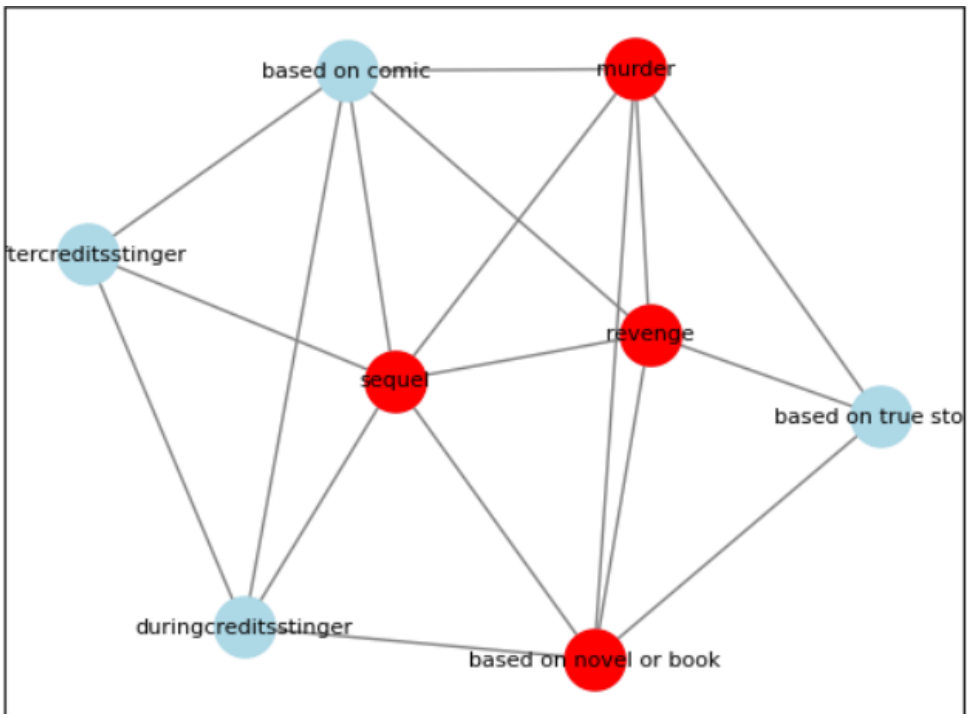

Рисунок 3 – Граф ключевых слов

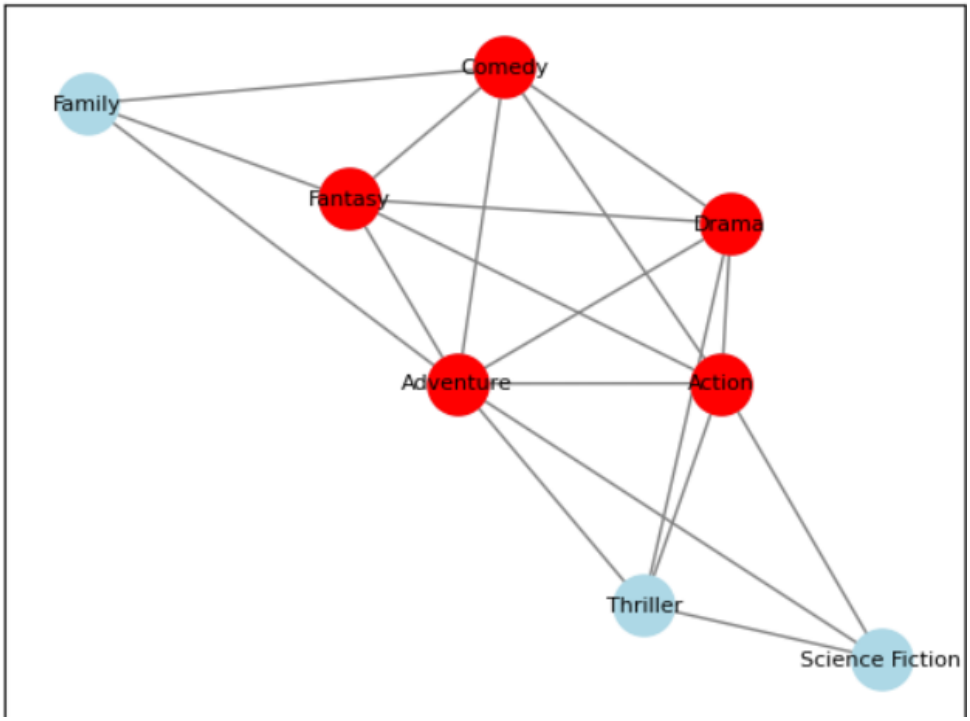

Рисунок 4 – Граф для жанров

Следующим шагом алгоритма является выделение максимальных клик для построенных графов. Максимальная клика для ключевых слов опреде-

лена следующим образом: $'sequel', 'basedonnovelorbook', 'murder', 'revenge'$, и максимальная клика для жанров определена как $'Adventure', 'Action', 'Drama', 'Fan$

Эти максимальные клики представляют собой общие паттерны для всех данных в столбце и дают ценную информацию о характеристиках общей структуры данных.

1. Результаты анализа паттернов, выявленных для ключевых слов, указывают на определенные тенденции в развитии кинематографа. Одним из ключевых направлений является акцент на темы мести и криминала, что отражает интерес зрителей к жанрам, связанным с драматическими конфликтами и моральными дилеммами. Кроме того, если рассматривать сильно связанные с паттерном элементы , наблюдается тенденция к созданию произведений, основанных на комиксах. Это может быть связано с популярностью франшиз и культурной значимостью комиксов как медиа-формата в современном обществе. Интересно отметить, что в кино все чаще используются сцены после титров. Такой подход позволяет углубить взаимодействие с аудиторией, создавая дополнительные сюжетные линии и поддерживая интерес к фильму даже после его окончания. Это свидетельствует о том, что зрители вовлечены в произведения, которые активно используют элементы, характерные для выявленного паттерна.
2. Анализ последовательности, присущей каждому набору данных, позволяет наблюдать вовлеченность и потребности зрителей и производителей фильмов в отношении определенных жанров кино. В числе наиболее привлекательных жанров выделяются Комедия, приключения, драма, фентези жанры, также сильно связанными являются жанры научная фантастика, трилерры, и симейные фильмы. Это говорит о том, что в процессе выбора картины аудитория предпочтет те произведения, в описании которых присутствует хотя бы один или несколько элементов, представленных в последовательности паттернов и дополнительно в меньшей степени предпочтут дополнительно жанры из связанных элементов графа. Это в свою очередь подчеркивает ценность введеного алгоритмического подхода для понимания предпочтений зрителей и определения стратегий, которые могут быть эффективно использованы

производителями фильмов для удовлетворения потребностей аудитории и достижения коммерческого успеха.

Заключение

В рамках проведенной работы были изучены ключевые модели и алгоритмы поиска паттернов, такие как алгоритм Apriori и метод Последовательных шаблонов. Это позволило получить глубокое понимание принципов их работы и особенностей применения в задачах анализа данных. На основе полученных знаний был разработан новый алгоритм для поиска закономерных последовательностей элементов в упорядоченных множествах. Этот алгоритм учитывает особенности упорядоченных множеств и ориентирован на максимальную точность выявления закономерных последовательностей. Проведенные тесты показали, что новый алгоритм обладает высокой точностью и может быть эффективно использован в различных прикладных областях, где требуется анализ упорядоченных множеств. Таким образом, результаты данной работы открывают новые возможности для исследований и применения методов поиска паттернов. Сконструированный алгоритм может стать основой для разработки новых методов анализа данных, способствуя прогрессу в области аналитики данных и машинного обучения

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Агравал,* Быстрые алгоритмы для поиска ассоциативных правил / Агравал, Шрикант. — Сентябрь 1994. — Pp. 487–499.

2 *Bayardo Jr, R. J.* Efficiently mining long patterns from databases // ACM SIGMOD Record / ACM. — Vol. 27. — 1998.

3 *Bechini, A.* From basic approaches to novel challenges and applications in sequential pattern mining / A. Bechini, A. Bondielli, P. Dell'Oglio, F. Marcelloni // *Applied Computing and Intelligence.* — 2023. — Vol. 3, no. 1. — Pp. 44–78.

4 Mining sequential patterns by pattern-growth: The prefixspan approach / J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal et al. // *IEEE Transactions on Knowledge and Data Engineering.* — 2004. — Vol. 16, no. 10. — Pp. 1424–1440.

5 *Kormen, T. H.* Introduction to Algorithms / T. H. Kormen, C. E. Leiserson, R. L. Rivest, C. Stein; Ed. by I. V. Krasikov. — 2 edition. — Москва: Вильямс, 2005.

6 Movie analysis: 2010-2020 popular titles. — `https://www.kaggle.com/datasets/knarasi1/movie-analysis-2010-2020-popular-titles`. — Дата обращения: 23 марта 2024 года.

7 Numpy. — `https://numpy.org/`. — Дата обращения: 23 марта 2024 года.

8 pandas. — `https://pandas.pydata.org/`. — Дата обращения: 23 марта 2024 года.

9 Matplotlib. — `https://matplotlib.org/`. — Дата обращения: 23 марта 2024 года.

10 Networkx. — `https://networkx.org/`. — Дата обращения: 23 марта 2024 года.

11 *Rathbuna, N.* Prefixspan. — `https://github.com/rathbuna/prefixspan`. — 2019. — Дата обращения: 23 марта 2024 года.

12 *Raschka, S.* Mlxtend. — `https://github.com/rasbt/mlxtend`. — 2014. — Дата обращения: 23 марта 2024 года.