УДК 004.8

# *РАЗРАБОТКА КЛАСТЕРНОГО АЛГОРИТМА РЕКОМЕНДАЦИИ ФИЛЬМОВ С ИСПОЛЬЗОВАНИЕМ СЛУЧАЙНЫХ БЛУЖДАНИЙ ПО ГРАФУ*

***Ребров М.Е.***

*магистрант,*

*МИРЭА – Российский Технологический Университет,*

*Москва, Россия*

***Иванова А.П.***

*к.ф.-м.н., доцент,*

*МИРЭА – Российский Технологический Университет,*

*Москва, Россия*

**Аннотация:**

В статье рассматривается процесс разработки гибридного метода, сочетающего в себе кластеризацию пользователей по жанровым предпочтениям и алгоритм случайных блужданий по графу с целью повышения качества рекомендаций в системах персонализации контента. Выполнено сравнение предложенного подхода с классическим методом случайных блужданий. Показано, что предварительная кластеризация позволяет повысить точность рекомендаций и сократить время вычислений за счёт уменьшения влияния шумовых связей между пользователями.

**Ключевые слова:** рекомендательные системы, кластеризация, случайные блуждания, графовые алгоритмы, машинное обучение.

# *DEVELOPMENT OF A CLUSTER-BASED MOVIE RECOMMENDATION ALGORITHM USING RANDOM WALKS ON GRAPHS*

***Rebrov M.E.***

*master's student,*

*MIREA – Russian Technological University,*
*Moscow, Russia*
***Ivanova A.P.***
*PhD, Associate Professor,*
*MIREA – Russian Technological University,*
*Moscow, Russia*

**Abstract:** This paper discusses the process of developing a hybrid method that combines clustering of users by genre preferences and a random walk graph algorithm in order to improve the quality of recommendations in content personalization systems. The proposed approach is compared with the classical random walk method. It is shown that pre-clustering makes it possible to increase the accuracy of recommendations and reduce the calculation time by reducing the influence of noise connections between users.
**Keywords:** recommender systems, clustering, random walks, graph algorithms, machine learning.

История развития рекомендательных систем тесно связана с ростом объёмов цифрового контента и необходимостью его эффективной фильтрации. С увеличением количества доступных для выбора пользователю объектов, например, фильмов, товаров, новостей, задача поиска релевантной информации становится всё более сложной [1]. В качестве ответа на данный вызов были разработаны различные подходы к построению рекомендательных алгоритмов, начиная от простых эвристических методов и заканчивая сложными моделями машинного обучения [2].

Одними из первых алгоритмов в этом направлении стали методы коллаборативной фильтрации, основанные на анализе взаимодействий пользователей с объектами. Такие методы предполагают, что пользователи с похожими предпочтениями будут выбирать схожие объекты [3]. Однако при

увеличении размерности данных и разреженности матриц взаимодействий эффективность данных подходов существенно снижается.

Альтернативой являются контент-ориентированные методы, которые используют характеристики самих объектов. В контексте рекомендательных систем фильмов такими характеристиками часто выступают жанры, позволяющие формировать интерпретируемое представление пользовательских предпочтений [4]. Однако подобные методы ограничены в способности учитывать коллективное поведение пользователей.

С развитием графовых методов особое распространение получили алгоритмы, основанные на случайных блужданиях по графу взаимодействий типа «пользователь–объект». Одним из таких методов является алгоритм $P^3$, использующий трёхшаговые переходы для выявления скрытых связей между пользователями и объектами [5]. Утверждается, что в рамках трёх произведённых переходов достигается объект, который может быть рекомендован пользователю и является для него релевантным. Данный подход позволяет учитывать структуру взаимодействий, однако подвержен влиянию шумовых связей, возникающих между пользователями с различными предпочтениями.

Для повышения качества рекомендаций активно используются методы кластеризации пользователей. Кластеризация позволяет разбить пользователей на группы, с близкими характеристиками, тем самым уменьшая влияние нерелевантных связей [6]. Наиболее распространённым алгоритмом кластеризации является метод K-Means, обладающий высокой вычислительной эффективностью и хорошей интерпретируемостью получаемых результатов [7].

В настоящей работе предлагается модификация алгоритма случайных блужданий по графу путём добавления к нему предварительной кластеризации пользователей.

На первом этапе формируется вектор жанровых предпочтений пользователя на основе истории его взаимодействий, что будет являться основой

для кластеризации. В отличие от классических методов, где пользователь описывается вектором взаимодействий высокой размерности, здесь используется компактное жанровое представление. На входе имеется история просмотров пользователя $u$, маппинг «фильм-жанры» $M$ и множество жанров $G$. На выходе получается вектор жанровых предпочтений $g_u \in R^{|G|}$, то есть вектор размерности $G$. Перед получением $g_u$ на выходе производится нормализация на количество просмотров $g_u/N_u$. Использование жанров в качестве признакового пространства обусловлено тем, что жанры являются естественным и понятным способом описания вкусов пользователя, что обеспечивает высокую интерпретируемость. Деление на количество просмотров $N_u$ принципиально важно, так как устраняет смещение в сторону пользователей, посмотревших сотни фильмов. Без нормализации такие пользователи образовывали бы отдельные кластеры, искажая структуру данных.

После построения жанровых профилей для всех пользователей формируется матрица признаков $X \in R^{|U|\times|G|}$. Для устранения влияния масштаба признаков, так как некоторые жанры встречаются чаще других, применяется стандартизация с помощью StandardScaler:

$$x'_{ij} = \frac{x_{ij-}\mu_j}{\sigma_j},$$

где $\mu_j$ – среднее по j-ому жанру, $\sigma_j$ – среднее квадратическое отклонение по $j$-ому жанру.

Для разбиения пользователей на однородные группы используется алгоритм K-Means. В экспериментах использовалось от 4 до 5 кластеров для всех трёх наборов данных. Данные значения были выбраны на основе показаний различных метрик оценки количества кластеров, таких как Elbow Method, Silhouette Score и Davies-Bouldin Score. Также это обеспечивает достаточную детализацию пользовательских сегментов и сохраняет вычислительную эффективность.

Задача K-Means формулируется как задача минимизация внутрикластерного расстояния:

$$\min(C_1, \dots, C_k) \sum_{k=1}^{K} \sum_{u \in C_k} \left\| g_u^{scaled} - \mu_k \right\|^2 ,$$

где $\mu_k$ – центроид кластера $C_k$.

Следует также отметить, что на практике могут встречаться пользователи с крайне редким сочетанием предпочтений и вкусов. В этом случае применение кластеризации может приводить к образованию малых кластеров, содержащих незначительное количество пользователей. В контексте графовых методов малые кластеры проблематичны по двум причинам. В этом случае граф внутри малого кластера слишком разрежен для построения качественной модели. Также вычислительные накладные расходы на обработку отдельного кластера обычно не окупаются из-за малого числа пользователей. Для решения этой проблемы вводится эвристическая процедура перераспределения пользователей из малых кластеров в ближайшие крупные. Она состоит в том, что получившиеся малые кластеры, то есть те, в которых количество пользователей меньше заданного порога, объединяются с крупными кластерами, ближайшими к ним по евклидову расстоянию между центроидами. В экспериментах использовался порог $\theta = 10$ пользователей. Кластеры размером менее десяти считаются статистически незначимыми для построения графовой модели. После выполнения процедуры поглощения производится перенумерация оставшихся кластеров последовательными индексами $0, 1, \dots, K' - 1$ , где $K' \leq K$.

После формирования кластеров для каждого из них независимо строится двудольный граф взаимодействий типа «пользователь–фильм». На первом шаге для кластера $C_k$ строится двудольный граф взаимодействий $G = (C_k \cup I, E_k)$, где $C_k$ – пользователи кластера, переиндексированные локально как $(0, \dots, |C_k| - 1)$, $I$ – полное множество фильмов (индексы смещены на $|C_k|$) , $E_k$ – множество рёбер, соответствующих оценкам пользователей из $C_k$ в обучающей выборке. На втором шаге происходит построение матрицы смежности графа и матрицы

переходов. Матрица смежности $A_k$ имеет размерность $(|C_k| + |I|) \times (|C_k| + |I|)$ и является симметричной:

$$A_k[u,m] = A_k[m,u] = \begin{cases} 1, & \text{если пользователь } u \text{ оценил фильм } m, \\ 0, & \text{иначе.} \end{cases}$$

Далее вычисляется стохастическая матрица переходов $P_k$ путём построчной нормализации: $P_k[i,j] = \frac{A_k[i,j]}{\sum_l A_k[i,l]}$. На третьем шаге выполняется вычисление $P^3$ для кластера. Матрица $P_k$ имеет блочную структуру:

$$P_k = \begin{bmatrix} 0 & P_{UI}^{(k)} \\ P_{IU}^{(k)} & 0 \end{bmatrix},$$

где $P_{UI}^{(k)}$ – матрица переходов от пользователей к фильмам (размер $|C_k| \times |I|$); $P_{IU}^{(k)}$ – матрица переходов от пользователей к фильмам (размер $|I| \times |C_k|$).

Итоговая матрица рекомендаций для кластера $P_3^{(k)}$ вычисляется как произведение трёх матриц $P_3^{(k)} = P_{UI}^{(k)} * P_{IU}^{(k)} * P_{UI}^{(k)}$ Размерность результирующей матрицы – $|C_k| \times |I|$. Элемент $P_3^{(k)}[u,m]$ интерпретируется как оценка релевантности фильма m для пользователя u в рамках кластера $C_k$.

Для эффективного хранения все матрицы хранятся в разреженном формате CSR с типом данных float32, что минимизирует потребление оперативной памяти.

Для кластера $C_k$ сложность умножения разреженных матриц составляет $O(|C_k| * |I| * \bar{d})$, где $\bar{d}$ – средняя степень вершины в графе. Суммарная сложность по всем кластерам $O(\sum_{k=1}^{K'} |C_k| * |I| * \bar{d}_k) \ll O(|C_k| * |I| * \bar{d}_{global})$.

После построения кластерных моделей $P_3^{(k)}$ для каждого кластера, система готова к построению рекомендаций. В случаях, когда пользователь по каким-либо причинам не был приписан к кластеру, например, это может быть новый пользователь без достаточной истории для построения жанрового профиля, система автоматически переключается на глобальную модель $P_3^{global}$. Это

обеспечивает отказоустойчивость метода и корректную обработку граничных случаев.

Для наглядного представления последовательности работы предлагаемого алгоритма приведена его схема на рис. 1.

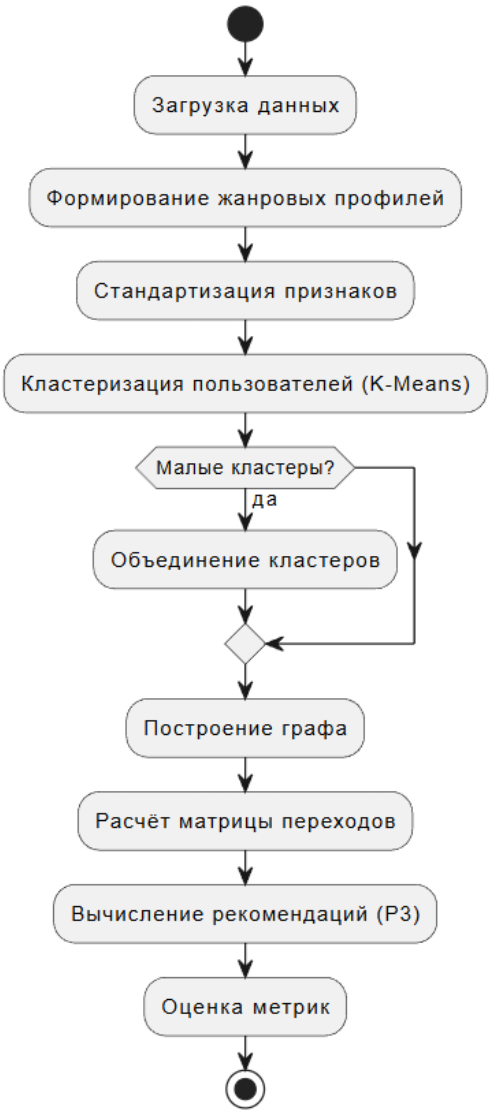


Рис. 1 – Схема работы алгоритма, авторская разработка

Для оценки эффективности предложенного метода используются две метрики, отражающие различные аспекты качества ранжирования. Метрика I оценивает процент правильно размещённых пар Pairwise Accuracy. Данная метрика была предложена в статье [5], описывающей алгоритм случайных

блужданий по графу на основе трёх переходов $P^3$, и считается наиболее строгой оценкой качества графовых рекомендательных систем. В отличие от метрик, фокусирующихся только на верхних позициях списка, таких как Precision@k или Recall@k, данная метрика оценивает глобальное качество ранжирования – насколько хорошо алгоритм упорядочивает все пары «релевантный объект-нерелевантный объект» во всём каталоге.

Пусть для пользователя $u$ заданы $I^{(u)}$ – множество всех фильмов, которые пользователь $u$ оценил, из них обучающая выборка $I^{(u)}_{train}$+ тестовая выборка $I^{(u)}_{test}$. Тогда $t_u = \left|I^{(u)}_{test}\right|$ – количество тестовых фильмов пользователя $u$, а $q = |I|$ – общее количество фильмов в каталоге. Для каждой пары фильмов $(m', m'')$, $m' \in I^{(u)}_{test}$ – тестовый фильм (релевантный объект), а $m'' \notin I^{(u)}$ – фильм, который пользователь не смотрел (нерелевантный объект), проверяется, правильно ли алгоритм расположил $m'$ выше, чем $m''$ в полном ранжировании всех фильмов. Тогда для пользователя $u$ метрика определяется следующим образом:

$$Metric_{I(u)} = \frac{1}{t_u * (q - i_u)} \sum_{m' \in I^{(u)}_{text}} \sum_{\substack{m'' \notin I^{(u)}, \\ m'' \neq m'}} 1[rank(m') < rank(m'')] \times 100\%,$$

где $i_u = \left|I^{(u)}\right|$ – общее количество фильмов, просмотренных пользователем; $rank(m)$ – позиция фильма $m$ в ранжированном списке (чем меньше значение, тем выше фильм).

Знаменатель $t_u * (q - i_u)$ представляет собой общее количество проверяемых пар релевантный-нерелевантный для пользователя $u$. Делением на это число достигается нормализация, позволяющая усреднять метрику по пользователям с разным количеством тестовых фильмов. Итоговая метрика вычисляется как среднее арифметическое по всем пользователям, имеющим хотя бы один тестовый фильм:

$$Metric_{I(u)} = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} Metric_{I(u)},$$

где $U_{test}$ – множество пользователей, для которых $t_u > 0$.

При этом значение метрики 50% – случайное угадывание, что эквивалентно подбрасыванию монетки для каждой пары. Если значение метрики больше 50%, то алгоритм значимо лучше случайного, 100% означает идеальное ранжирование, то есть, когда все тестовые фильмы расположены выше всех непросмотренных.

Ввиду квадратичной сложности относительно размера каталога $O(q^2)$, вычисление Метрики I для всех пользователей на большом датасете размером $10^6$ взаимодействий требует значительных временных затрат. По этой причине в экспериментах применялось ограничение на количество оцениваемых пользователей (sample_size = 50), выбранных случайным образом из тестовой выборки.

Метрика II является классической мерой качества рекомендаций, широко используемой в задачах оценивания качества рекомендательных систем [5]. Она оценивает практическую полезность системы с точки зрения конечного пользователя, который, как правило, просматривает лишь небольшую долю рекомендованных объектов, то есть несколько первых позиций выдачи. Для пользователя $u$ и заданного $p$% от общего размера каталога определяется абсолютное значение $k$:

$$k = \max\left(1, \left[\frac{p}{100} * q\right]\right),$$

где $q = |I|$ – общее количество фильмов.

Метрика вычисляется как доля тестовых фильмов пользователя, попавших в топ-$k$ его персональных рекомендаций:

$$Hits@p\%(u) = \frac{\left|I_{test}^{(u)} \cap Top_k(u)\right|}{\left|I_{test}^{(u)}\right|},$$

где $Top_k(u)$ – множество из $k$ фильмов с наивысшими оценками релевантности для пользователя $u$, исключая уже просмотренные фильмы из обучающей выборки.

Итоговая метрика II для заданного $p\%$ вычисляется как среднее арифметическое по всем пользователям:

$$Metric_{II_{p\%}} = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} Hits@p\%(u),$$

В экспериментах используются четыре пороговых значения: 1%, 3%, 5% и 10%. Такой набор позволяет оценить качество рекомендаций как в условиях жесткого ограничения на размер выдачи, как, например, Top-1%, так и при более мягких требованиях, как Top-10%. При этом 0 означает, что ни один тестовый фильм не попал в топ-$k$, а 1 – все тестовые фильмы попали в топ-$k$, что является идеальным результатом. Чем выше значение, тем лучше алгоритм справляется с задачей выдачи релевантных рекомендаций на верхних позициях.

Совместное использование Метрики I и Метрики II позволяет получить полную картину о поведении алгоритма. Метрика I гарантирует, что улучшение на верхних позициях не достигнуто ценой деградации общего качества ранжирования. Метрика II подтверждает, что теоретическое преимущество в глобальном упорядочивании транслируется в практическую пользу для пользователя, просматривающего лишь начало списка.

В работе использованы три набора данных MovieLens различного масштаба: $10^5$, $10^6$ и $10^7$ взаимодействий [8]. Каждый датасет содержит информацию о взаимодействиях пользователей с фильмами, включая оценки, жанры и метаданные. Для построения модели использовались идентификатор пользователя, идентификатор фильма и жанры фильмов. Жанровая информация используется для формирования компактного представления предпочтений пользователя. Результат сравнения алгоритма случайных блужданий по графу, состоящего из трёх переходов $P^3$ и его кластерной модификации на маленьком датасете размером $10^5$ взаимодействий приведён в табл. 1.

Таблица 1 – результат сравнения двух алгоритмов на датасете размером $10^5$ взаимодействий, авторская разработка

| **Evaluation** | **Global P³** | **Cluster P³** |
|---|---|---|
| Metric I | 89.27% | 89.73% |
| Metric II@1% | 0.2594 | 0.2625 |
| Metric II@3% | 0.4631 | 0.4702 |
| Metric II@5% | 0.5642 | 0.5711 |
| Metric II@10% | 0.7137 | 0.7226 |

Результат сравнения глобального и кластерного алгоритмов на среднем датасете размером $10^6$ взаимодействий представлен в табл. 2.

Таблица 2 – Результат сравнения двух алгоритмов на датасете размером $10^6$ взаимодействий, авторская разработка

| **Evaluation** | **Global P³** | **Cluster P³** |
|---|---|---|
| Metric I | 90.99% | 91.35% |
| Metric II@1% | 0.2331 | 0.2613 |
| Metric II@3% | 0.4209 | 0.4470 |
| Metric II@5% | 0.5264 | 0.5535 |
| Metric II@10% | 0.6804 | 0.7067 |

Результат сравнения двух алгоритмов на большом датасете размером $10^7$ взаимодействий представлен в табл. 3.

Таблица 3 – Результат сравнения двух алгоритмов на датасете размером $10^7$ взаимодействий, авторская разработка

| **Evaluation** | **Global P³** | **Cluster P³** |
|---|---|---|
| Metric I | 95.81% | 96.10% |

| | | |
|---|---|---|
| Metric II@1% | 0.4952 | 0.4887 |
| Metric II@3% | 0.6889 | 0.6895 |
| Metric II@5% | 0.7831 | 0.7839 |
| Metric II@10% | 0.8845 | 0.8896 |

Не менее важным аспектом сравнения является время построения моделей в этих двух алгоритмах. Результат сравнения по этому параметру в секундах представлен в табл. 4.

Таблица 4 – Результат сравнения по времени на трёх версиях датасета разного размера, авторская разработка

| **Dataset** | **Global $P^3$** | **Cluster $P^3$** |
|---|---|---|
| Small | 6.21 | 8.66 |
| Medium | 81.58 | 69.83 |
| Large | 106.35 | 96.70 |

Таким образом, предложенный подход сочетает преимущества кластеризации и графовых методов. Использование локальных моделей внутри кластеров способствует более точному учёту предпочтений пользователей и повышению эффективности алгоритма при работе с большими объёмами данных. Кластерная модификация позволила сократить время выполнения, относительно классического алгоритма случайных блужданий по графу, состоящего из трёх переходов $P^3$, примерно в 1,1 раза. Качество по метрике I предлагаемого метода оказалось выше в среднем на 0.37%, а по метрике II в среднем на 0.04. Тенденция сохраняется при увеличении объёма данных, что критически важно для такого рода алгоритмов. Однако на маленьком наборе данных размером $10^5$ взаимодействий время работы кластерного алгоритма оказалось немного выше, чем классического из-за накладных расходов на

создание моделей для каждого кластера. При общем сокращении времени работы алгоритма удалось не только не потерять в качестве рекомендаций, но и немного улучшить показатели метрик, что доказывает идею о том, что локальные модели для каждого кластера проще в вычислении, а также предлагают более точные рекомендации для пользователей.

**Библиографический список:**

1. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. – Springer, 2015. – С. 15-18.
2. Aggarwal C.C. Recommender Systems: The Textbook. – Springer, 2016. – С. 518.
3. А.Г. Гомзин, А.В. Коршунов. Системы рекомендаций: обзор современных подходов. – 2012. – №2. – С. 3-5.
4. Н.А. Клёмин, А.А. Абакумов, А.И. Егунова, А.В. Прончатов, Д.О. Грошев. Контент-ориентированный подход в системах рекомендаций: принципы, методы и метрики эффективности // Мордовский государственный университет им. Н.П. Огарёва. – 2025. – №5. – С. 2-5.
5. Cooper C., Lee S.H., Radzik T., Siantos Y. Random walks in recommender systems: Exact computation and simulations // WWW Conference. – 2014. – С. 1-6.
6. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. – Elsevier, 2011. – С. 210-218.
7. Булыга Ф. С., Курейчик В. М. Кластеризация корпуса текстовых документов при помощи алгоритма k-means. – Южный федеральный университет, г. Таганрог, Россия, 2022. – С. 4-6.
8. Harper F.M., Konstan J.A. The MovieLens Datasets: History and Context // ACM Transactions on Interactive Intelligent Systems. – 2015. – С. 1-3.