

# Предвзятость популярности в рекомендательных системах

И.С. Демьянов, Д.Е. Намиот

**Аннотация**—Рекомендательные системы играют важную роль в современных цифровых платформах, предоставляя персонализированные рекомендации, которые улучшают пользовательский опыт. На сегодняшний день рекомендации успешно внедрены в маркетплейсы, сервисы для просмотра фильмов и прослушивания музыки. Основной целью рекомендательной системы является предсказание для каждого пользователя объектов платформы, которые будут ему релевантны. Как правило, данная задача решается методами машинного обучения с учётом информации о пользователе, объектах платформы и матрицы взаимодействий пользователей и объектов. Однако такой подход нередко приводит к проблеме предвзятости популярности, при которой рекомендательные алгоритмы отдают чрезмерное предпочтение популярным объектам.

Данная статья нацелена на обзор метрик детекции предвзятости популярности в рекомендательных системах. Также в работе приводится обзор стратегий уменьшающих предвзятость популярности с перечислением их преимуществ и недостатков.

**Ключевые слова**—рекомендательные системы, предвзятость популярности, метрики популярности, разнообразие рекомендаций, персонализация, ранжирование, реранжирование.

## I. Введение

Рекомендательные системы (RecSys) представляют собой совокупность методов и алгоритмов, задача которых заключается в том, чтобы предсказать какие объекты-сущности (фильмы, книги, товары и т.д.) будут интересны пользователю на основе данных о самом пользователе и объектах [1].

Современные рекомендательные системы демонстрируют беспрецедентную способность улучшать пользовательский опыт за счет эффективного сопоставления разнообразных предпочтений отдельных пользователей с огромным количеством доступных элементов [1, 2]. Рекомендательные системы будут актуальны и далее, поскольку людям свойственно совершать покупки, общаться в социальных сетях, слушать музыку, смотреть фильмы и проводить ряд других активностей, в которых уместны релевантные рекомендации, дополняющие опыт в использовании платформы[3]. Тем более, учитывая наблюдаемый рост цифрового контента и информации в последние годы[4, 5], рекомендательные системы на основе машинного обучения прочно заняли лидирующую позицию в нахождении персонализированного релевантного контента для пользователя платформы.

Иван Сергеевич Демьянов, МГУ им. М.В. Ломоносова, (email: ivandemyanov17@gmail.com).

Дмитрий Евгеньевич Намиот, МГУ им. М.В. Ломоносова, (email: dnamiot@gmail.com).

Выделяют следующие основные типы рекомендательных систем:

- 1) Основанные на экспертных знаниях[6]
- 2) Основанные на контенте[7]
- 3) Основанные на поведении пользователей[8]
- 4) Гибридные[9]

Алгоритмы, которые учитывают взаимодействие пользователя с рекомендательной системой, считаются наиболее сильными, как следствие, наиболее часто применяются на практике и в исследованиях. Популярность данного типа рекомендательных систем привела к формированию различных подходов учёта поведения пользователей: алгоритмы основанные на близости предпочтений похожих пользователей[10], матричные факторизации[11], обучение ранжирования[12], нейросетевые архитектуры[13]. При ограничениях истории взаимодействия пользователей прибегают к ряду эвристик: составление персонализированных рекомендаций из популярных объектов, формирование рекомендаций на основе текущей сессии пользователя(Session-based Recommender Systems, SBRS)[5, 14].

В силу актуальности рекомендательных систем и обилия разнообразия подходов к формированию рекомендаций крайне важно корректно оценивать качество рекомендательных алгоритмов. Для стандартной оценки качества рекомендательной платформы принято использовать метрики классификации. Тем не менее, данные метрики никак не отражают предвзятость популярных объектов в рекомендациях.

Многие рекомендательные системы страдают от предвзятости популярности: популярные объекты рекомендуются чрезмерно часто, тогда как менее популярные объекты рекомендуются редко или не рекомендуются вообще(см. Рис. 1). Однако рекомендации таких объектов имеют решающее значение для бизнеса, поскольку вероятность того, что они будут самостоятельно обнаружены пользователями, гораздо меньше.

Для детектирования вышеописанной проблемы используются метрики популярности, оценивающие, насколько хорошо рекомендации соответствуют предпочтениям пользователей, и могут влиять на баланс между разнообразием и значимостью предложенных объектов. Во II разделе приводится обзор метрик популярности.

В последующих разделах данной работы проводится детальный обзор стратегий, направленных на снижение предвзятости популярности в рекомендательных системах на разных этапах их построения. Рассматриваются методы предварительной обработки данных, включая фильтрацию и нормализацию информации о пользователях и объектах, подходы к модификации алгоритмов обучения, учитывающие справедливость и равномерность

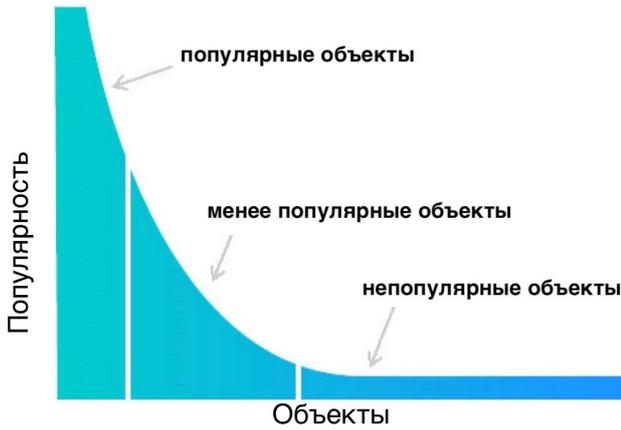


Рис. 1. Распределение объектов по их популярности [15]

рекомендаций, а также техники постобработки, такие как реранжирование выдачи. Комплексный анализ этих стратегий позволяет выявить их преимущества и ограничения, а также оценить их влияние на качество и разнообразие рекомендаций.

## II. Метрики популярности

### A. Формализация

Для постановки задачи рекомендательной системы и удобства обзора метрик введём следующие обозначения:

- $U$  – множество пользователей – users.
- $I$  – множество объектов – items.
- $R$  – множество рекомендаций, являющееся подмножеством  $U \times I$ .
- $D = \{(u, v) : d_{uv}\}$  – матрица взаимодействий.

Ввиду актуальности рекомендательных систем и их динамичного развития в различных областях жизни человека, понятие объекта(item) в зависимости от платформы может заменяться на товар, фильм, документ и другие. Матрица взаимодействий  $D$  может трактоваться как матрица предпочтений. Взаимодействия разделяют на 2 типа:

- Явный(explicit)
  - 1) лайк/дизлайк
  - 2) оценка
  - 3) покупка
  - 4) отзыв
- Неявный(implicit)
  - 1) клик/показ
  - 2) глубина просмотра
  - 3) добавление в корзину
  - 4) поисковые запросы

Основная цель рекомендательной системы – построение рекомендаций для пользователей из множества  $U$  при помощи матрицы взаимодействий:

$$f_{\theta^*} : U \times I \rightarrow R$$

где  $\theta^*$  – наилучший параметр из параметрического семейства  $\Theta$  выбранной модели, настраиваемый за счёт максимизации выбранной метрики  $\mathcal{M}_{rec}$ :

$$\theta^* = \arg \max_{\Theta} \mathcal{M}_{rec}(f_{\theta}(D))$$

Стандартно для оценки качества рекомендательной модели выбирают метрики точности и ранжирования:

- Метрики для оценки точности рекомендаций:  $MAP@k$ ,  $MAR@k$ ,  $Hit\_rate@k$
- Метрики для оценки ранжирования рекомендаций:  $NDCG@k$ ,  $MRR@k$

### B. Метрики

**Average Recommendation Popularity (ARP)** [15–17] рассчитывает среднюю популярность рекомендуемых объектов для каждого пользователя с последующим усреднением:

$$ARP = \frac{1}{|U_t|} \sum_{u \in U_t} \frac{\sum_{i \in L_u} \phi(i)}{|L_u|}$$

где  $\phi(i)$  – количество встреч объекта  $i$  в обучающем наборе данных.  $L_u$  – список рекомендаций для пользователя  $u$  и  $|U_t|$  – количество пользователей на валидации.

**Average Percentage of Long Tail Items (APLT)** [15] измеряет средний по пользователям процент рекомендаций не самых популярных объектов и определяется следующим образом:

$$APLT = \frac{1}{|U_t|} \sum_{u \in U_t} \frac{|\{i, i \in (L_u \cap \Gamma)\}|}{|L_u|}$$

здесь  $\Gamma$  – список не самых популярных объектов, а  $|L_u|$  – список рекомендаций для пользователя  $u$

Метрика **Gini** [16–18] учитывает, насколько равномерно элементы отображаются в списках рекомендаций. Равномерное распределение будет иметь индекс Джини, равный нулю, что является идеальным случаем (нижний предел).

$$Gini(L) = \frac{1}{|I| - 1} \sum_{k=1}^{|I|} (2k - |I| - 1) p(i_k | L)$$

где  $L$  – рекомендации всех пользователей и  $p(i_k | L)$  – вероятность того, что  $k$ -й наименее рекомендуемый объект войдет в  $L$  задается формулой:

$$p(i | L) = \frac{\sum_{u \in U} 1_{i \in L_u}}{\sum_{u \in U} \sum_{j \in I} 1_{j \in L_u}}$$

Метрика **Entropy** [18] учитывает распределение рекомендуемых объектов, энтропия измеряет равномерность этого распределения.

$$Entropy(L) = - \sum_{i \in I} p(i | L) \log p(i | L)$$

Равномерное распределение имеет самую высокую энтропию или прирост информации, поэтому более высокая энтропия более желательна, когда целью является увеличение разнообразия.

Метрика **Coverage** [17, 18] показывает процент элементов, которые хотя бы один раз появляются в списках рекомендаций.

$$Coverage(L) = \frac{|\cup_{u \in U} L_u|}{|I|}$$

где  $I$  — множество всех объектов рекомендательной системы.

Метрика **диверсификации** [19] измеряет усреднённое по пользователям отношение уникальных по категории рекомендаций к количеству уникальных категорий

$$\text{Diversity}(L) = \frac{1}{|U_t|} \sum_{u \in U_t} \frac{\text{unique}(\text{categories}(L_u))}{|\text{categories}(L)|}$$

Ниже приведена таблица с указанием диапазонов значений вышеупомянутых метрик.

Таблица 1  
Метрики популярности

Метрика	Описание	Диапазон
ARP	Средняя популярность рекомендуемых элементов	$[0, \infty)$
APLT	Средний процент рекомендаций не самых популярных объектов	$[0, 1]$
Gini	Оценивает неравномерность распределения популярности	$[0, 1]$ (0 – равномерно)
Entropy	Характеризует разнообразие рекомендаций по популярности	$[0, \infty)$
Coverage	Покрытие рекомендациями всего каталога	$[0, 1]$
Diversity	Отношение уникальных по категории рекомендаций к количеству уникальных категорий	$[0, 1]$

### III. Стратегии кандидатогенерации и ранжирования

Кандидатогенерация и ранжирование — ключевые этапы работы рекомендательных систем (см. Рис.2), определяющие релевантные пользователю объекты платформы и производящие дальнейшее упорядочивание. Этап кандидатогенерации позволяет отобрать из миллионов объектов платформы объекты, подходящие пользователю, которые далее, на этапе ранжирования, будут расположены в наиболее релевантном порядке для пользователя. Современные алгоритмы кандидатогенерации и ранжирования используют машинное обучение, учитывая информацию о пользователе, объектах и их взаимодействиях, чтобы максимизировать метрики качества рекомендаций.

В данном разделе будут рассмотрены стратегии уменьшения предвзятости популярности на этапах обработки данных, кандидатогенерации и ранжирования, заключающиеся в модификации рекомендательных алгоритмов.

#### A. Предобработка данных

Помимо тривиального решения в виде фильтрации взаимодействий пользователей с самыми популярными объектами рекомендательной системы, на стадии предобработки данных выделяют следующие подходы: обогащение данных перед обучением модели и разделение данных на популярные объекты и непопулярные с последующим обучением моделей отдельно на каждой из групп [20].

В качестве обогащения данных учитывают дополнительные данные о пользователях, включая их социальные

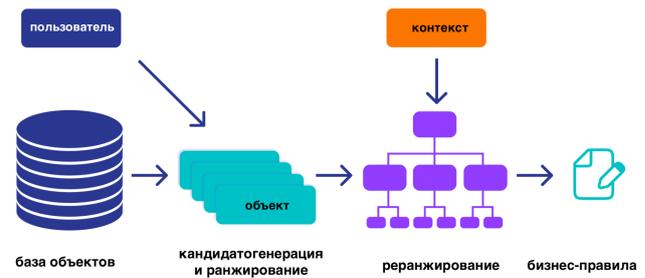


Рис. 2. Схема работы рекомендательной системы

связи [21], что расширяет представление о предпочтениях и может уменьшить влияние наиболее популярных объектов. Также в исследованиях [22] для обогащения данных применяют линейные комбинации явных и неявных откликов, что делает обратную связь более выразительной и, как следствие, разнообразные рекомендации.

#### B. Матричные факторизации и линейные модели

Обучение моделей кандидатогенерации вносит важный вклад в наличие предвзятости популярности в рекомендациях. Как правило, на этом этапе имеют место несколько моделей с целью учесть разнообразную логику формирования кандидатов в рекомендации для пользователя. Довольно часто на этом уровне используются линейные модели и матричные факторизации. В литературе выделяют несколько подходов для уменьшения предвзятости популярности во время обучения моделей кандидатогенерации.

1) *Регуляризация в кандидатогенерации:* Одним из основных направлений в этом контексте является использование методов регуляризации. Регуляризация позволяет контролировать влияние популярности путем добавления штрафного слагаемого в функцию потерь. В статье [23] предложен следующий метод, минимизирующий зависимость рекомендаций от популярности:

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \lambda \cdot R(P)$$

где  $\mathcal{L}_{\text{base}}$  — стандартная функция потерь модели,  $R(P)$  — мера влияния популярности, а  $\lambda$  — коэффициент регуляризации, позволяющий балансировать между точностью рекомендаций и снижением предвзятости.

2) *Ограничения для уменьшения предвзятости:* Методы, основанные на ограничениях, включают явные правила, направленные на контроль частоты популярных и менее популярных объектов. Предложена [24] концепция  $(\alpha, \beta)$ -справедливости, в которой вводится ограничение на формирование рекомендаций среди похожих объектов:

$$|E(i) - E(j)| \leq \beta, \quad \text{если } d(i, j) \leq \alpha$$

где  $E(i)$  — частота объекта  $i$ ,  $d(i, j)$  — мера сходства.

3) *Взвешивание примеров в обучении:* Методы взвешивания позволяют изменять важность объектов в процессе обучения модели. Предложено[25] уменьшать влияние популярных объектов, понижая вес их наблюдаемых рейтингов:

$$w_i = \frac{1}{P(i)}$$

где  $P(i)$  — популярность объекта  $i$ . Альтернативный подход заключается в увеличении веса малоизвестных объектов, что позволяет повысить их вероятность попадания в список кандидатов.

4) *Динамическое ослабление предвзятости:* В работе [26] рассматривают долгосрочное влияние кандидато-генерации на предвзятость. Предложено динамическое регулирование, при котором коэффициент ослабления предвзятости изменяется во времени:

$$\lambda_t = \lambda_0 \cdot f(t)$$

где  $\lambda_t$  — коэффициент регуляровки на момент времени  $t$ , а  $f(t)$  — функция, моделирующая динамическое усиление эффекта ослабления предвзятости.

5) *Методы обратного оценивания склонности (IPS):* Методы IPS уменьшают влияние популярности, пересчитывая веса на основе вероятности взаимодействия пользователя с объектом [27]:

$$\hat{y}_u^i = \frac{y_u^i}{p(i)}$$

где  $y_u^i$  — бинарный отклик пользователя  $u$  для объекта  $i$ ,  $p(i)$  — вероятность выбора этого объекта.

Таким образом, обучение кандидато-генерации представляет собой ключевой механизм управления популярностью в рекомендациях. Использование регуляризации, ограничений и взвешивания позволяет добиться более сбалансированных и справедливых рекомендаций.

### C. Обучение ранжированию

Ранжирование обеспечивает отношение порядка среди объектов платформы релевантных пользователю. Довольно часто ранжируются объекты-кандидаты для пользователя при помощи бустинга, поскольку зачастую данная модель показывает лучшее качество на табличных данных. На этапе ранжирования в ходе обучения модели возможно учитывать предвзятость популярности, так в качестве альтернативы AdaBoost в статье [12] предлагают использовать Fairboost, основанный на итеративном обучении базовых моделей и корректировке весов обучающих примеров с учётом предвзятости популярности. Псевдокодом 1 представлен алгоритм обучения Fairboost.

FairBoost использует жадное увеличение весов для непопулярных объектов, аналогично тому, как AdaBoost увеличивает веса ошибочно классифицированных примеров. Однако вместо фокусировки исключительно на ошибках классификации, FairBoost оптимизирует баланс между точностью предсказаний и устранением предвзятости популярности. За это отвечает множитель  $(1 + cost_{um})$  в формуле обновления весов: если объект не подвержен популярному смещению, то  $cost_{um}$  остаётся малым, и обновление веса работает почти как в

### Algorithm 1 FairBoost

**Вход:**  $A = (X_{um}, Y_{um})^N, M$

**Выход:** ансамбль базовых алгоритмов  $B$

**Инициализация:**  $w_{um} = \frac{1}{N}$  and  $cost_{um} = 0$ , для всех пар пользователь-объект

1: **for**  $j \leftarrow 1$  to  $M$  **do do**

2: Обучение базовой модели  $\hat{A}_j$  на выборке с весами  $w_{um}$

3: Вычисление ошибки:

$$err_j = \sum_{(u,m) \in S_j} w_{um} \frac{|A_{um} - \hat{A}_j|}{\max(A_{um} - \hat{A}_j)}$$

4: Вычисление веса базовой модели:

$$\alpha_j = \frac{1}{2} \log \left( \frac{1 - err_j}{err_j} \right)$$

5: Вычисление смещения популярности для текущей модели РВ  $(\hat{A}^j, \tau)$

6: Подсчет штрафа  $cost_{um}$  за смещение популярности

7: Обновление весов:

$$w_{um} \leftarrow w_{um} \cdot e^{\alpha_j \cdot (A_{um} - \hat{A}_j)} \cdot (1 + cost_{um})$$

8: **end for**

9:  $B(x) = \sum_{j=1}^M \alpha_j \hat{A}^j(x)$

AdaBoost, в случае если модель предпочитает популярные объекты, увеличивая их предсказания, то  $cost_{um}$  растёт, заставляя алгоритм увеличивать веса непопулярных объектов. Это приводит к тому, что следующие базовые модели в ансамбле будут больше учитывать ранее недооценённые объекты.

Следовательно, FairBoost представляет собой эффективный метод обучения ранжированию, обеспечивающий справедливое распределение рекомендаций между популярными и непопулярными объектами.

### D. Трансформерные архитектуры

Нейронные сети в последнем десятилетии внесли значительный вклад в улучшение качества рекомендательных систем. Особенно популярными архитектурами являются трансформеры, пришедшие из области обработки естественного языка(NLP).

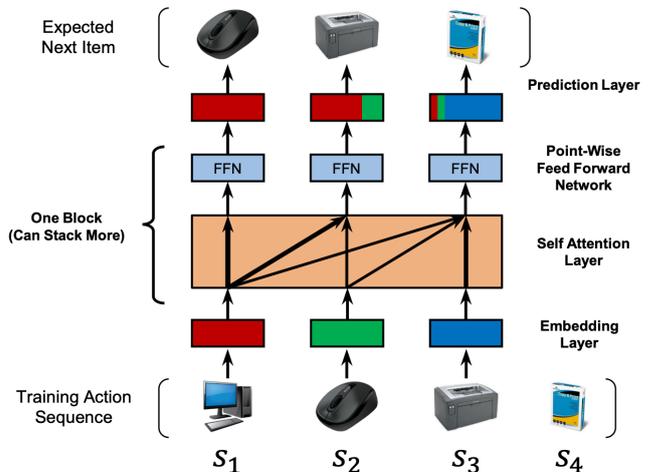


Рис. 3. Процесс обучения SASRec [28]

Данный класс рекомендательных алгоритмов особенно хорошо себя показывает в задаче рекомендации сопут-

ствующих объектов[28] и также применяется при построении рекомендательных систем на основе сессий[14]. В архитектуре трансформеров ключевую роль играет механизм внимания (attention), позволяющий модели учитывать взаимосвязи между различными элементами входной последовательности:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, M) = \text{softmax} \left( \frac{\mathbf{Q}\mathbf{K}^T + M}{\sqrt{d_k}} \right) \mathbf{V}$$

При помощи механизма внимания осуществляется преобразование векторного представления объекта последовательности в три компоненты:

- **Запрос** (*query*,  $q$ ): представляет текущий элемент, для которого вычисляется внимание.
- **Ключ** (*key*,  $k$ ): представляет элемент, к которому применяется внимание.
- **Значение** (*value*,  $v$ ): представляет информацию, связанную с элементом.

Итоговое представление для каждого элемента получается как взвешенная сумма значений всех элементов с учётом вычисленных весов внимания. Данный механизм позволяет модели учитывать контекст всей последовательности при обработке каждого отдельного элемента.

В рамках данной трансформерной архитектуры смещение популярности обусловлено неравномерным распределением градиентов – популярные объекты встречаются чаще, поэтому их эмбединги получают более мощные градиентные обновления, в то время как редкие объекты обновляются слабее. Следовательно, возникает дисбаланс внимания: популярные объекты имеют более информативные представления по причине частотного преимущества, а механизм внимания отдаёт им предпочтение.

Уменьшить предвзятость популярности трансформерной архитектуры возможно на этапе обучения: при случайном семплировании отрицательных примеров популярные объекты имеют более высокую вероятность быть выбранными, что может привести к модели, которая чрезмерно фокусируется на различии между популярными и непопулярными объектами. Это, в свою очередь, может усилить смещение рекомендаций в пользу популярных объектов, снижая разнообразие и персонализацию рекомендаций. В работе [29] предлагается семплирование отрицательных примеров  $i^-$  с учётом вероятности выбора, которая определяется следующим образом:

$$P(i^- | e_{ut}, i^+) = \phi(i^-)^\gamma,$$

где  $\phi(i^-)$  — мера популярности отрицательного объекта, а  $\gamma$  — показатель степени. При  $\gamma = 1$  вероятность пропорциональна популярности объекта, а при  $\gamma = 0$  выбор осуществляется равномерно.

В статье [30] авторы представляют новый фреймворк SPRec, использующий механизм самоигры для снижения предвзятости в рекомендациях, основанных на больших языковых моделях. В отличие от традиционных методов, SPRec не требует дополнительной разметки данных или ручного вмешательства. В каждом цикле самоигры модель проходит этапы SFT и DPO, где предсказания

предыдущей итерации используются в качестве отрицательных примеров. Это позволяет адаптивно корректировать функцию потерь DPO, нивелируя предвзятость популярности.

Другой подход заключается в модификации вычисления весов внимания, учитывая популярность объектов:

$$(QK^T)_{ij}^{\text{corr}} = \frac{\exp(q_i^T k_j / \text{pop}(j))}{\sum_{j'} \exp(q_i^T k_{j'} / \text{pop}(j'))}$$

где  $\text{pop}(j)$  — мера популярности объекта  $j$ , например, количество взаимодействий с ним. Деление скалярного произведения  $q_i^T k_j$  на показатель популярности снижает влияние популярных объектов на итоговые рекомендации, способствуя более сбалансированным и персонализированным результатам.

#### IV. Стратегии ранжирования

Важно отличать ранжирование от реранжирования: ранжирование, как правило, выполняется в рамках рекомендательной модели на основе предсказанных вероятностей взаимодействия пользователя с объектами, тогда как реранжирование применяется поверх ранжирования. Основная цель реранжирования — корректировка исходного порядка рекомендаций для улучшения визуального качества рекомендаций.

##### A. MMR

Метод максимальной маргинальной релевантности[31] направлен на балансировку между релевантностью и разнообразием при формировании итогового списка рекомендаций. Он максимизирует разнообразие, выбирая элементы, которые не только соответствуют запросу пользователя, но и отличаются от уже выбранных элементов. Формально, следующий элемент  $S$  выбирается из множества кандидатов  $D$  так, чтобы максимизировать выражение:

$$\max_{S \in D \setminus R} \left[ \lambda \cdot \text{Sim}(S, Q) - (1 - \lambda) \cdot \max_{R_i \in R} \text{Sim}(S, R_i) \right],$$

где  $R$  — текущее множество выбранных элементов,  $Q$  — запрос пользователя,  $\text{Sim}$  — функция схожести, а  $\lambda$  — параметр, регулирующий баланс между релевантностью и разнообразием. Метод MMR изначально был предложен для уменьшения избыточности в результатах поиска и автоматическом реферировании текстов, но также успешно используется для реранжирования рекомендаций с целью повышения разнообразия.

##### B. TF-IDF

TF-IDF — это статистическая мера, используемая для оценки важности термина в документе относительно всего корпуса документов. В контексте предвзятости популярности рекомендательных алгоритмов множитель IDF имеет важное значение: непопулярные, но подходящие объекты имеют высокий показатель IDF, тогда как популярные — низкий. Таким образом, комбинация значений IDF и релевантности объекта, предсказанной ранжирующей моделью, может купировать проблему предвзятости популярности, что помогает рекомендовать ключевые объекты платформы[32].

### C. *eps-greedy*

Алгоритм  $\epsilon$ -greedy используется в задачах с дилеммой «исследование-эксплуатация», таких как многорукие бандиты. В контексте ранжирования рекомендаций, алгоритм  $\epsilon$ -greedy[33] вносит предположение, что с вероятностью  $\epsilon$  система выбирает случайный элемент для исследования, а с вероятностью  $(1 - \epsilon)$  — наиболее предпочтительный элемент на основе текущей информации. Такой подход позволяет находить баланс между исследованием новых вариантов и использованием уже известных предпочтений пользователя.

### D. *Thompson Sampling*

Thompson Sampling [34] — вероятностный алгоритм, применяемый для решения дилеммы «исследование-эксплуатация» в задачах многоруких бандитов. Принцип работы данного алгоритма семплирования состоит в следующем:

- 1) Для каждого объекта моделируется апостериорное распределение ожидаемой награды на основе предыдущих наблюдений.
- 2) Из каждого апостериорного распределения случайным образом выбирается значение.
- 3) Объект с наивысшим выбранным значением рекомендуется пользователю.

Вышеописанный процесс повторяется, позволяя системе обновлять представления о предпочтениях пользователя и адаптивно улучшать рекомендации.

### V. Заключение

Рекомендательные системы в современном мире стали неотъемлемой частью цифровых платформ, взяв на себя функцию дополнения человеческого опыта среди огромного потока информации. На сегодня методы машинного обучения успешно внедрены в различные этапы формирования рекомендаций. Однако данные методы склонны к предвзятости популярности: алгоритмам свойственно продвижение уже популярных объектов платформы, что может ограничивать пользователей в доступе к менее известному, но потенциально релевантному контенту. Предвзятость популярности в рекомендательной системе снижает разнообразие выдачи и, как следствие, персонализацию платформы.

В данной статье проведён обзор метрик для выявления предвзятости популярности в рекомендательных системах, а также стратегий уменьшения предвзятости популярности на разных этапах формирования рекомендаций: предобработка данных, кандидогенерация, ранжирование, ранжирование. Работа подчёркивает важность учёта предвзятости популярности при проектировании рекомендательных систем. В перспективе дальнейшие исследования в области справедливости и персонализации рекомендаций будут способствовать разработке более адаптивных и инклюзивных систем, удовлетворяющих разнообразные потребности пользователей.

### Благодарности

Авторы благодарны сотрудникам кафедры Информационной безопасности факультета ВМК МГУ имени М.В. Ломоносова за ценные замечания и обсуждение.

Статья подготовлена в рамках работы над магистерской диссертацией. Направление работ относится к проекту факультета ВМК МГУ имени М.В. Ломоносова, посвященному устойчивым моделям машинного обучения[35]. Традиционно отмечаем, что все публикации в журнале INJOIT, связанные с цифровой повесткой, начинались с работ В.П. Куприяновского и его многочисленных соавторов[36, 37]

### Библиография

- [1] Almazro, Dhoha, et al. «A survey paper on recommender systems.» *arXiv preprint arXiv:1006.5278* (2010).
- [2] Roy, Deepjyoti, and Mala Dutta. «A systematic review and research perspective on recommender systems.» *Journal of Big Data* 9.1 (2022): 59.
- [3] Mukherjee, Subhabrata, Hemank Lamba, and Gerhard Weikum. «Item recommendation with evolving user preferences and experience.» *arXiv preprint arXiv:1705.02519* (2017).
- [4] Sankalp, K. J., et al. «Advancements in Modern Recommender Systems: Industrial Applications in Social Media, E-commerce, Entertainment, and Beyond.» (2024).
- [5] Wang, Shoujin, et al. «A survey on session-based recommender systems.» *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.
- [6] Górski, Franciszek, et al. «Integrating Expert Knowledge into Logical Programs via LLMs.» *arXiv preprint arXiv:2502.12275* (2025).
- [7] Aggarwal, Charu C., and Charu C. Aggarwal. «Content-based recommender systems.» *Recommender systems: The textbook* (2016): 139-166.
- [8] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan. «Collaborative filtering recommender systems.» *Foundations and Trends® in Human-Computer Interaction* 4.2 (2011): 81-173.
- [9] Burke, Robin. «Hybrid recommender systems: Survey and experiments.» *User modeling and user-adapted interaction* 12 (2002): 331-370.
- [10] Anelli, Vito Walter, et al. «On the discriminative power of hyper-parameters in cross-validation and how to choose them.» *Proceedings of the 13th ACM conference on recommender systems*. 2019.
- [11] Luo, Xin, et al. «An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems.» *IEEE Transactions on Industrial Informatics* 10.2 (2014): 1273-1284.
- [12] Gangwar, Ajay, and Shweta Jain. «An adaptive boosting technique to mitigate popularity bias in recommender system.» *arXiv preprint arXiv:2109.05677* (2021).
- [13] Klenitskiy, Anton, and Alexey Vasilev. «Turning dross into gold loss: is bert4rec really better than sasrec?.» *Proceedings of the 17th ACM Conference on Recommender Systems*. 2023.
- [14] Ninichuk, Marina, and Dmitry Namiot. «Survey On Methods For Building Session-Based Recommender Systems.» *International Journal of Open Information Technologies* 11.5 (2023): 22-32.

- [15] Abdollahpouri, Himan, Robin Burke, and Bamshad Mobasher. «Managing popularity bias in recommender systems with personalized re-ranking.» *arXiv preprint arXiv:1901.07555* (2019).
- [16] Abdollahpouri, Himan, et al. «User-centered evaluation of popularity bias in recommender systems.» *Proceedings of the 29th ACM conference on user modeling, adaptation and personalization*. 2021.
- [17] Klimashevskaja, Anastasiia, Mehdi Elahi, and Christoph Trattner. «Addressing popularity bias in recommender systems: An exploration of self-supervised learning models.» *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*. 2023.
- [18] Mansoury, M., et al. «FairMatch: A graph-based approach for improving aggregate diversity in recommender systems. arXiv 2020.» *arXiv preprint arXiv:2005.01148* (2005).
- [19] Hurley, Neil, and Mi Zhang. «Novelty and diversity in top-n recommendation—analysis and evaluation.» *ACM Transactions on Internet Technology (TOIT)* 10.4 (2011): 1-30.
- [20] Park, Yoon-Joo, and Alexander Tuzhilin. «The long tail of recommender systems and how to leverage it.» *Proceedings of the 2008 ACM conference on Recommender systems*. 2008.
- [21] Li, Roger Zhe, Julián Urbano, and Alan Hanjalic. «Leave no user behind: Towards improving the utility of recommender systems for non-mainstream users.» *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021.
- [22] Jadidinejad, Amir H., Craig Macdonald, and Iadh Ounis. «Unifying explicit and implicit feedback for rating prediction and ranking recommendation tasks.» *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*. 2019.
- [23] Kamishima, Toshihiro, et al. «Correcting popularity bias by enhancing recommendation neutrality.» *RecSys posters* 10 (2014).
- [24] Wang, Xiuling, and Wendy Hui Wang. «Providing item-side individual fairness for deep recommender systems.» *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022.
- [25] Steck, Harald. «Item popularity and recommendation accuracy.» *Proceedings of the fifth ACM conference on Recommender systems*. 2011.
- [26] Zhu, Ziwei, et al. «Popularity bias in dynamic recommendation.» *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery data mining*. 2021.
- [27] Schnabel, Tobias, et al. «Recommendations as treatments: Debiasing learning and evaluation.» *international conference on machine learning*. PMLR, 2016.
- [28] Kang, Wang-Cheng, and Julian McAuley. «Self-attentive sequential recommendation.» *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018.
- [29] Prakash, Arushi, Dimitrios Bermperidis, and Srivas Chennu. «Evaluating Performance and Bias of Negative Sampling in Large-Scale Sequential Recommendation Models.» *arXiv preprint arXiv:2410.17276* (2024).
- [30] Gao, Chongming, et al. «SPRec: Leveraging Self-Play to Debias Preference Alignment for Large Language Model-based Recommendations.» *arXiv preprint arXiv:2412.09243* (2024).
- [31] Carbonell, Jaime, and Jade Goldstein. «The use of MMR, diversity-based reranking for reordering documents and producing summaries.» *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 1998.
- [32] Thangaraj, Harish, et al. «A BERT Based Hybrid Recommendation System For Academic Collaboration.» *arXiv preprint arXiv:2502.15223* (2025).
- [33] Sani, S. M. F., Seyed Abbas Hosseini, and Hamid R. Rabiee. «Epsilon non-Greedy: A Bandit Approach for Unbiased Recommendation via Uniform Data.» *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2023.
- [34] Ishikawa, Shion, Young-joo Chung, and Yu Hirate. «Dynamic collaborative filtering Thompson Sampling for cross-domain advertisements recommendation.» *arXiv preprint arXiv:2208.11926* (2022).
- [35] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. «The rationale for working on robust machine learning.» *International Journal of Open Information Technologies* 9.11 (2021): 68-74.
- [36] Развитие транспортно-логистических отраслей Европейского Союза: открытый ВИМ, Интернет Вещей и кибер-физические системы / В. П. Куприяновский, В. В. Аленков, А. В. Степаненко [и др.] // *International Journal of Open Information Technologies*. – 2018. – Т. 6, № 2. – С. 54-100. – EDN YNIRFG.
- [37] Умная инфраструктура, физические и информационные активы, Smart Cities, ВИМ, GIS и IoT / В. П. Куприяновский, В. В. Аленков, И. А. Соколов [и др.] // *International Journal of Open Information Technologies*. – 2017. – Т. 5, № 10. – С. 55-86. – EDN ZISODV.

# Popularity bias in recommender systems

Ivan Demianov, Dmitry Namiot

**Abstract**—Recommender systems play a crucial role in modern digital platforms by providing personalized recommendations that enhance user experience. Today, recommendations are successfully integrated into marketplaces, movie streaming services, and music platforms. The primary goal of a recommender system is to predict the most relevant items for each user. Typically, this task is solved using machine learning methods that take into account user data, platform items, and user-item interaction matrices. However, such an approach often leads to the problem of popularity bias, where recommendation algorithms excessively favor popular items.

This paper aims to review the metrics for detecting popularity bias in recommender systems. Additionally, it provides an overview of strategies designed to mitigate popularity bias, discussing their advantages and disadvantages.

**Keywords**—recommender systems, popularity bias, popularity metrics, recommendation diversity, personalization, ranking, re-ranking.

## References

- [1] Almazro, Dhoha, et al. «A survey paper on recommender systems.» *arXiv preprint arXiv:1006.5278* (2010).
- [2] Mukherjee, Subhabrata, Hemank Lamba, and Gerhard Weikum. «Item recommendation with evolving user preferences and experience.» *arXiv preprint arXiv:1705.02519* (2017).
- [3] Roy, Deepjyoti, and Mala Dutta. «A systematic review and research perspective on recommender systems.» *Journal of Big Data* 9.1 (2022): 59.
- [4] Sankalp, K. J., et al. «Advancements in Modern Recommender Systems: Industrial Applications in Social Media, E-commerce, Entertainment, and Beyond.» (2024).
- [5] Górski, Franciszek, et al. «Integrating Expert Knowledge into Logical Programs via LLMs.» *arXiv preprint arXiv:2502.12275* (2025).
- [6] Aggarwal, Charu C., and Charu C. Aggarwal. «Content-based recommender systems.» *Recommender systems: The textbook* (2016): 139-166.
- [7] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan. «Collaborative filtering recommender systems.» *Foundations and Trends® in Human-Computer Interaction* 4.2 (2011): 81-173.
- [8] Burke, Robin. «Hybrid recommender systems: Survey and experiments.» *User modeling and user-adapted interaction* 12 (2002): 331-370.
- [9] Wang, Shoujin, et al. «A survey on session-based recommender systems.» *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.
- [10] Anelli, Vito Walter, et al. «On the discriminative power of hyper-parameters in cross-validation and how to choose them.» *Proceedings of the 13th ACM conference on recommender systems*. 2019.
- [11] Luo, Xin, et al. «An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems.» *IEEE Transactions on Industrial informatics* 10.2 (2014): 1273-1284.
- [12] Gangwar, Ajay, and Shweta Jain. «An adaptive boosting technique to mitigate popularity bias in recommender system.» *arXiv preprint arXiv:2109.05677* (2021).
- [13] Klenitskiy, Anton, and Alexey Vasilev. «Turning dross into gold loss: is bert4rec really better than sasrec?.» *Proceedings of the 17th ACM Conference on Recommender Systems*. 2023.
- [14] Abdollahpouri, Himan, Robin Burke, and Bamshad Mobasher. «Managing popularity bias in recommender systems with personalized re-ranking.» *arXiv preprint arXiv:1901.07555* (2019).
- [15] Abdollahpouri, Himan, et al. «User-centered evaluation of popularity bias in recommender systems.» *Proceedings of the 29th ACM conference on user modeling, adaptation and personalization*. 2021.
- [16] Klimashevskaja, Anastasiia, Mehdi Elahi, and Christoph Trattner. «Addressing popularity bias in recommender systems: An exploration of self-supervised learning models.» *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*. 2023.
- [17] Mansoury, M., et al. «FairMatch: A graph-based approach for improving aggregate diversity in recommender systems. arXiv 2020.» *arXiv preprint arXiv:2005.01148* (2005).
- [18] Hurley, Neil, and Mi Zhang. «Novelty and diversity in top-n recommendation—analysis and evaluation.» *ACM Transactions on Internet Technology (TOIT)* 10.4 (2011): 1-30.
- [19] Li, Roger Zhe, Julián Urbano, and Alan Hanjalic. «Leave no user behind: Towards improving the utility of recommender systems for non-mainstream users.» *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021.
- [20] Jadidinejad, Amir H., Craig Macdonald, and Iadh Ounis. «Unifying explicit and implicit feedback for rating prediction and ranking recommendation tasks.» *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*. 2019.
- [21] Park, Yoon-Joo, and Alexander Tuzhilin. «The long tail of recommender systems and how to leverage it.» *Proceedings of the 2008 ACM conference on Recommender systems*. 2008.
- [22] Kamishima, Toshihiro, et al. «Correcting popularity bias by enhancing recommendation neutrality.» *Rec-Sys posters* 10 (2014).

- [23] Wang, Xiuling, and Wendy Hui Wang. «Providing item-side individual fairness for deep recommender systems.» *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022.
- [24] Steck, Harald. «Item popularity and recommendation accuracy.» *Proceedings of the fifth ACM conference on Recommender systems*. 2011.
- [25] Zhu, Ziwei, et al. «Popularity bias in dynamic recommendation.» *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery data mining*. 2021.
- [26] Schnabel, Tobias, et al. «Recommendations as treatments: Debiasing learning and evaluation.» *international conference on machine learning*. PMLR, 2016.
- [27] Kang, Wang-Cheng, and Julian McAuley. «Self-attentive sequential recommendation.» *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018.
- [28] Ninichuk, Marina, and Dmitry Namiot. «Survey On Methods For Building Session-Based Recommender Systems.» *International Journal of Open Information Technologies* 11.5 (2023): 22-32.
- [29] Prakash, Arushi, Dimitrios Bermperidis, and Srivas Chennu. «Evaluating Performance and Bias of Negative Sampling in Large-Scale Sequential Recommendation Models.» *arXiv preprint arXiv:2410.17276* (2024).
- [30] Gao, Chongming, et al. «SPRec: Leveraging Self-Play to Debias Preference Alignment for Large Language Model-based Recommendations.» *arXiv preprint arXiv:2412.09243* (2024).
- [31] Carbonell, Jaime, and Jade Goldstein. «The use of MMR, diversity-based reranking for reordering documents and producing summaries.» *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 1998.
- [32] Thangaraj, Harish, et al. «A BERT Based Hybrid Recommendation System For Academic Collaboration.» *arXiv preprint arXiv:2502.15223* (2025).
- [33] Sani, S. M. F., Seyed Abbas Hosseini, and Hamid R. Rabiee. «Epsilon non-Greedy: A Bandit Approach for Unbiased Recommendation via Uniform Data.» *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2023.
- [34] Ishikawa, Shion, Young-joo Chung, and Yu Hirate. «Dynamic collaborative filtering Thompson Sampling for cross-domain advertisements recommendation.» *arXiv preprint arXiv:2208.11926* (2022).
- [35] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. «The rationale for working on robust machine learning.» *International Journal of Open Information Technologies* 9.11 (2021): 68-74.
- [36] Razvitie transportno-logisticheskikh otraslej Evropejskogo Sojuza: otkrytyj BIM, Internet Veshhej i kiberfizicheskie sistemy / V. P. Kuprijanovskij, V. V. Alen'kov, A. V. Stepanenko [i dr.] // *International Journal of Open Information Technologies*. – 2018. – T. 6, 2. – S. 54-100. – EDN YNIRFG.
- [37] Umnaja infrastruktura, fizicheskie i informacionnye aktivy, Smart Cities, BIM, GIS i IoT / V. P. Kuprijanovskij, V. V. Alen'kov, I. A. Sokolov [i dr.] // *International Journal of Open Information Technologies*. – 2017. – T. 5, 10. – S. 55-86. – EDN ZISODV.