ВВЕДЕНИЕ. В настоящее время в связи с ростом количества информации по любым темам встает вопрос о том, как найти интересную или необходимую пользователю информацию в сети интернет, в библиотечных системах или хранилищах данных. В связи с этим активно стали развиваться алгоритмы контекстного поиска, которые позволяют предложить пользователю то, что его скорее всего интересует.

Одной из разновидностей контекстного поиска являются рекомендательные системы. Такие системы позволяют предсказывать запросы пользователя, используя информацию о его предыдущих заказах. Например, зная, как пользователь оценил предыдущие книги, можно построить некую формальную модель для описания его интересов. Подходы к созданию таких моделей могут быть самыми разными, начиная от простого поиска k ближайших соседей, и заканчивая сложными вероятностными алгоритмами [8, 12, 13, 7].

Актуальность данной работы в необходимости повышения заинтересованности общества в чтении книг. Многие не делают этого, так как не знают с чего начать. Рекомендательная система помогает неопытным пользователям подобрать наиболее интересные книги, экономя их время. Кроме того, данная система может оказаться полезной для пользователей, которые активно подбирают книги по своим предпочтениям или начинают научное исследование в определенной области, что также требует подбора соответствующей литературы.

Цель работы: построение рекомендательной системы по подбору книг пользователям путем реализации алгоритма GroupLens, использующего коллаборативную фильтрацию, основанную на пользователях.

Работа состоит из введения, двух глав, заключения, списка использованных источников и одного приложения. Содержит 1 таблицу, 7 рисунков и 20 источников. Во введении кратко рассмотрена область применения РС, описана актуальность работы, сформулирована цель. В первой главе приведены теоретические основы рекомендательных систем,

описаны основные алгоритмы. Во второй главе приведено подробное описание разработанного приложения, рассмотрен пример использования РС. В заключении сфомулированы результаты работы. В приложении приведен полный код программы.

1 Рекомендательные системы. В данной работе был реализован алгоритм GroupLens, который активно использует коллаборативную фильтрацию, основанную на пользователях. Формула приведена ниже (1).

Простейший способ построить предсказание нового рейтинга $\hat{r}_{ui}$ – сумма рейтингов других пользователей, взвешенная их похожестью на пользователя u:

$$\hat{r}_{u,i} = \bar{r}_i + \frac{\sum_j (r_{j,i} - \bar{r}_j)\omega_{u,j}}{\sum_j |\omega_{u,j}|}, \quad (1)$$

где u всегда будет обозначать пользователей (всего пользователей N, $u = 1..N$). i – предметы (сайты, товары, книги, фильмы и т.д.), которые мы рекомендуем (всего M, $i = 1..M$). x_u – набор (вектор) признаков (features) пользователя, x_i - набор признаков предмета.

Когда пользователь u оценивает предмет i, он производит отклик (response, rating) $r_{u,i}$, этот отклик случайная величина, конечно.

Наша задача предсказывать оценки $r_{u,i}$, зная признаки x_u и x_i для всех элементов базы и, зная некоторые уже расставленные в базе $r_{u',i'}$. Предсказание будем обозначать через $\hat{r}_{u,i}$.

Следуя формуле (1), задача сводится к подсчету коэффициента $\omega[u,j]$, который описывает «похожесть», или близость пользователей u и j. В данной работе для этого была использована косинусная мера с учетом обратной частоты.

$$\omega_{u,j}^{idf} = \frac{\sum_i f_i^2 r_{u,i} r_{j,i}}{\sqrt{\sum_i (f_i r_{u,i})^2}\sqrt{\sum_i (f_i r_{j,i})^2}}, \quad (2)$$

Для подсчета похожести пользователя u и j требуется вычислить вес для каждой книги i.

$$f_i = \log \frac{N_i}{N}, \quad (3)$$

где N – общее число пользователей, N_i – число пользователей, оценивших продукт i.

Следовательно, с помощью этого коэффициента можно добиться, чтобы самые популярные книги не рекомендовались всем пользователям, независимо от их предпочтений.

Рекомендательные системы – одно из наиболее популярных приложений интеллектуального анализа данных и машинного обучения в сфере интернет-бизнеса. Рекомендательная система анализирует поведение пользователей интернет-сервиса, после чего может давать оценку предпочтения пользователем того или иного объекта рекомендаций [2]. Объектами рекомендаций могут служить товары в интернет-магазине, набор разделов веб-сайта, медиа-контент, другие пользователи веб-сервиса.

Рекомендательные системы помогают пользователям ориентироваться в большом числе контента, размещенного на сайте. В некоторых случаях это необходимая функциональность.

Можно выделить три основных подхода к построению рекомендательных систем (следуя [1]):

1 на основании признаковых описаний (content-based);

2 коллаборативная фильтрация (collaborative filtering);

3 гибридный подход.

Content-based. Подход на основании признаковых описаний предполагает, что про пользователей и про рекомендуемые объекты известно достаточно много информации. Например, все пользователи заполняют анкету, в которой указывают свою социально-демографическую информацию, интересы, и т.д. Про товары из интернет-магазина может быть известно их описание, предназначение, ценовая категория, бренд, и другие характеристики. По истории взаимодействия пользователей и объектов на

сервисе можно построить обучающую выборку и свести предсказание предпочтения к хорошо изученной задаче обучения по прецедентам [20].

На практике, использование такого похода сильно ограничено, т.к. сбор описательной информации о пользователях и объектах очень дорогостоящая процедура, которую зачастую невозможно организовать не в ущерб качеству использования сервиса, что делает рекомендательную систему неоправданно дорогой.

Коллаборативная фильтрация. Коллаборативной фильтрацией называется предсказание степени предпочтения в условиях, когда рекомендательная система не обладает какой-либо описательной информацией о пользователях и объектах (либо не использует), строит прогноз исключительно на основании взаимодействия пользователей с объектами.

Огромным толчком в исследовании математических моделей коллаборативной фильтрации послужил конкурс Netflix Prize [3]. Компания Netflix занимается интернет-прокатом кинофильмов. Целью конкурса являлось улучшение качества предсказываемой оценки пользователя некоторому фильму. Набор данных конкурса содержал четверки (дата-время, пользователь, фильм, оценка). Оценка измерялась от 1 до 5.

Гибридный подход. Несмотря на то, что алгоритмы коллаборативной фильтрации на практике показывают высокие показатели эффективности, учет дополнительной информации может сделать показатели еще выше. Одним из недостатков коллаборативной фильтрации по сравнению с методами, основанными на признаковом описании, является проблема холодного старта [16].

Гибридный подход использует композиции алгоритмов основанных на признаковых описаниях и результатов коллаборативной фильтрации.

Обратной связью (feedback) пользователя на некоторый объект в рекомендательных системах принято называть событие, по которому можно

судить о предпочтении пользователя к объекту. Вот несколько примеров обратной связи от пользователя:

- проставление оценки объекту по бальной шкале (количество звезд);
- нажатие на кнопку “нравится” (лайк) / “не нравится” (дизлайк);
- посещение страницы с описанием объекта, переход по ссылке на объект (клик);
- посещение страницы с описанием объекта более одного раза (заинтересованность);
- добавление в корзину / покупка объекта в случае, если это товар.

Именно по обратной связи пользователя на различные объекты, рекомендательная система формирует матрицу оценок предпочтений R, к которой затем применяются алгоритмы коллаборативной фильтрации. Преобразование обратной связи в числовое значение предпочтения непростая и очень важная задача в настройке рекомендательных систем. Как правило, при выборе схемы оценки предпочтения оптимизируется метрика, непосредственно связанная с ключевыми показателями эффективности (KPI) бизнеса. Техники подбора схемы оценки предпочтения выходят за рамки данной работы.

По видам обратной связи, задачи моделирования предпочтения в рекомендательных системах принято разделять на два вида:

1. с явной обратной связью (explicit feedback);
2. с неявной обратной связью (implicit feedback).

Так, например, рекомендации по оценкам из пятибальной шкалы – пример задачи с явной обратной связью. Рекомендательные системы, руководствующиеся актами покупок, посещением страниц - примеры задач с неявной обратной связью.

В случае неявной обратной связи имеется неопределенность в том, положительно или отрицательно влияют конкретный акт обратной связи на степень предпочтения. Покупка товара в интернет-магазине может означать

достижение пользователем своей потребительской цели (положительное предпочтение), но в то же время покупатель мог после получения товара в нем разочароваться и правильно было бы засчитать негативную степень предпочтения. Очевидно, что посещения страниц пользователями веб-сервиса могут происходить при совершенно разной степени заинтересованности пользователя в контенте. Стоит отметить достаточно типичную ситуацию, когда рекомендательной системе подаются на вход исключительно положительные примеры взаимодействия пользователей и объектов. Например, веб-сервис Twitter не имеет функциональности, позволяющей пользователю выразить свое низкое предпочтение контенту, а присутствует только лишь способ “поощрить” тот или иной контент, распространив его своим подписчикам посредством функции “репост”. Подобная обратная связь пользователя очень надежно (по сравнению с остальными) указывает на положительную степень предпочтения. Надежность “репостов” в сервисе Twitter подкреплена ответственностью пользователей перед своими подписчиками.

2 Алгоритм построения рекомендательной системы. Основной алгоритм, использованный в данной работе, выглядит следующим образом:

1. Для всех пользователей вычисляется средняя оценка, поставленная пользователем;
2. Для всех книг вычисляется её вес (согласно формуле (3)) и средняя оценка, поставленная пользователями, оценившими книгу;
3. На основе данных, полученных в пунктах 1 и 2, для каждой книги вычисляется ожидаемая оценка текущего пользователя (согласно формулам (1) и (2));
4. Пользователю выдаются лучшие 3 книги, которые ему будут наиболее интересны, согласно ожидаемым оценкам.

Рекомендованные книги вычисляются по формуле (1). При этом оценка считается не для каждой книги, а только для тех, которые понравились похожим пользователям.

В качестве реализации идей описанных в работе была построена следующая рекомендательная система.

Модель: пользователи, книги и оценки.

Для пользователей, книг и оценок созданы модельные классы `User`, `Book` и `Mark` соответственно.

Один из пользователей (администратор) может добавлять книги. Для опредеделения фунций администратора и админского интерфейса был создан класс `AdminFrame`. Остальные пользователи могут оценить любую книгу целым число от 1 до 10. На основе оценок пользователя ему даётся рекомендация в виде k (при реализации было выбрано значение $k = 3$) самых интересных книг, на основе оценок которые пользователь уже поставил и оценок других пользователей.

Для реализации этой модели в качестве СУБД была выбрана MySQL. База данных построена по скрипту в файле db_scripts.sql. При запуске системы открывается окно входа в систему. Для этого был создан класс `LoginFrame`. Пользователь вводит логин/пароль и, если такого пользователя ещё не было, то он создаётся. Если пользователь уже есть и введён правильный пароль, то пользователь входит в систему. В противном случае выводится соответствующее сообщение о неудачном входе в систему. Все логины и пароли хранятся в базе данных recommendersystem в таблице user. При корректном входе в систему в зависимости от того является пользователь администратором (для простоты считается, что администратор это пользователь с логином “admin”) или нет, открывается либо администраторское окно добавления/удаления книг в/из систему/системы, либо окно оценки книг. Оценки всех пользователей записываются в базу данных. Соответственно при запуске программы оценки, которые пользователь уже поставил, сразу загружаются и отображаются на окне. В любой момент пользователь может нажать кнопку “Get recommendation” для получения трёх рекомендуемых системой книг.

ЗАКЛЮЧЕНИЕ. В данной работе была построена простейшая рекомендационная система, основанная на алгоритме коллаборативной фильтрации пользователей GroupLens. Для определения дистанции между пользователями использовалась косинусная мера с коэффициентом idf, смягчающим влияние самых популярных книг.

Работа была выполнена на языке java. Графическая оболочка реализована с помощью библиотеки Java Swing. Среда разработки IntelliJ IDEA.

Данную работу можно развить, применив методы машинного обучения, например, используя классификаторы для группировки похожих книг и пользователей, учитывая не оценки, а непосредственно данные о книгах, например, название, издательство или год публикации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Adomavicius G., Tuzhilin A.* Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. // Knowledge and Data Engineering, IEEE Transactions on. – 2005. - Vol. 17(6). – Pp. :734–749.

2. *Adomavicius G., Tuzhilin A.* Context-aware recommender systems. In Recommender systems handbook. // Springer – 2011.- Pp.:217–253.

3. *BennettJ., LanningS.* The netflix prize. // In Proceedings of KDD cup and workshop. – 2007. Vol. 2007(35).

4. Developing a context-aware electronic tourist guide: some issues and experiences. // *CheverstK., DaviesN., MitchellK., FridayA., EfstratiouC.* // InProceedings of the SIGCHI conference on Human factors in computing systems. ACM – 2000. – Pp.: 17–24.

5. *DesrosiersC., KarypisG.* A comprehensive survey of neighborhood-based recommendation methods. // In Recommender systems handbook. Springer - 2011. - Pp.: 107–144.

6. *GantnerZ., RendleS., Schmidt-ThiemeL*. Factorization models for context-/time-aware movie recommendations. // In Proceedings of the Workshop on Context-Aware Movie Recommendation. ACM - 2010. – Pp.:14–19.

7. *HidasiB., TikkD*. Fast als-based tensor factorization for context-aware recommendation from implicit feedback. // In Machine Learning and Knowledge Discovery in Databases. Springer - 2012. – Pp.: 67–82.

8. *HuY., KorenY., VolinskyC*. Collaborative filtering for implicit feedback dataset. // In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. IEEE - 2008. – Pp.: 263–272.

9. *KorenY*. Factorization meets the neighborhood: a multifaceted collaborative filtering model. // In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM - 2008. – Pp.:426–434.

10. *LemireD., MaclachlanA*. Slope one predictors for online rating-based collaborative filtering. // In SDM. SIAM – 2005. – Vol 5, - Pp.: 1–5.

11. *PiatetskyG*. Interview with simon funk. // ACM SIGKDD Explorations Newsletter – 2007. – Vol 9(1) – Pp.:38–40.

12. *Pil´aszyI., ZibriczkyD., TikkD*. Fast als-based matrix factorization for explicit and implicit feedback datasets. // In Proceedings of the fourth ACM conference on Recommender systems. ACM – 2010. – Pp.: 71–78.

13. Bpr: Bayesian personalized ranking from implicit feedback. // In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. *RendleS., FreudenthalerC., GantnerZ., Schmidt-ThiemeL*. AUAI Press - 2009. – Pp.: 452–461.

14. Fast context-aware recommendations with factorization machines. // In Proceedings of the 34th ACM SIGIR Conference on Reasearch and Development in Information Retrieval. *RendleS., GantnerZ., FreudenthalerC., Schmidt-ThiemeL*. ACM - 2011. - Pp.:48-63.

15. Item-based collaborative filtering recommendation algorithms. // In Proceedings of the 10th international conference on World Wide Web. *SarwarB., KarypisG., KonstanJ., RiedlJ.* ACM - 2001. – Pp.: 285–295.

16. Methods and metrics for cold-start recommendations. // In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. *ScheinA., PopesculA., UngarL., PennockD* . ACM - 2002. – Pp.: 253–260.

17. *SteckH*. Training and testing of recommender systems on data missing not at random. // In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM – 2010. – Pp.: 713–722.

18. *WangJ., VriesA., ReindersM*. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. // In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM - 2006. - Pp.: 501–508.

19. Large-scale parallel collaborative filtering for the netflix prize. *ZhouY., WilkinsonD., SchreiberR., PanR*. // In Algorithmic Aspects in Information and Management. Springer – 2008. – Pp.: 337–348.

20. *ВоронцовК. В.* Математические методы обучения по прецедентам (теория обучения машин). [Электронный ресурс] URL: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf (дата обращения 2.04.2016). Загл. с экрана. Яз.рус.