

Министерство образования и науки Российской Федерации

Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

Ю.С. Белов, С.С. Гришунов

MANOUT. СИСТЕМА РЕКОМЕНДАЦИЙ
Методические указания по выполнению лабораторной работы
по курсу «Технологии обработки больших данных»

Калуга - 2018

УДК 004.62
ББК 32.972.5
Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий и прикладной математики».

Методические указания рассмотрены и одобрены:

- Кафедрой «Программного обеспечения ЭВМ, информационных технологий и прикладной математики» (ФН1-КФ) протокол № 6 от «12» января 2018 г.


Зав. кафедрой ФН1-КФ  д.ф.-м.н., профессор Б.М. Логинов

- Методической комиссией факультета ФНК протокол № 1 от «30» 01 2018 г.

Председатель методической комиссии факультета ФНК  к.х.н., доцент К.Л. Анфилов

- Методической комиссией КФ МГТУ им.Н.Э. Баумана протокол № 1 от «06» 02 2018 г.

Председатель методической комиссии КФ МГТУ им.Н.Э. Баумана

 д.э.н., профессор О.Л. Перерва



Рецензент:

к.т.н., зав. кафедрой ЭИУ2-КФ

 И.В. Чухраев

Авторы

к.ф.-м.н., доцент кафедры ФН1-КФ
ассистент кафедры ФН1-КФ

 Ю.С. Белов
 С.С. Гришунов

Аннотация

Методические указания по выполнению лабораторной работы по курсу «Технологии обработки больших данных» содержат краткое описание принципа работы систем рекомендации, примеры создания и тестирования эффективности систем коллаборативной фильтрации с помощью библиотеки Mahout.

Предназначены для студентов 4-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2018 г.

© Ю.С. Белов, С.С. Гришунов, 2018 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ.....	5
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ	6
РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ.....	8
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ	12
ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ.....	12
ВАРИАНТЫ ЗАДАНИЙ.....	12
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ	14
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ	14
ОСНОВНАЯ ЛИТЕРАТУРА.....	15
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА	15

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Технологии обработки больших данных» на кафедре «Программное обеспечение ЭВМ, информационные технологии и прикладная математика» факультета фундаментальных наук Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 4-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткое описание принципа работы систем рекомендации, примеры создания и тестирования эффективности систем коллаборативной фильтрации с помощью библиотеки Mahout и задание на выполнение лабораторной работы.

Методические указания составлены для ознакомления студентов с библиотекой Mahout. Для выполнения лабораторной работы студенту необходимы минимальные знания по программированию на высокоуровневом языке программирования Java.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является формирование практических навыков работы с библиотекой Mahout для создания рекомендательных систем на основе больших данных.

Основными задачами выполнения лабораторной работы являются:

1. Изучить алгоритмы системы рекомендаций на основе коллаборативной фильтрации.
2. Научиться реализовывать системы рекомендаций с помощью Apache Mahout
3. Научиться выполнять оценку правильности работы системы рекомендаций.

Результатами работы являются:

- Входные файлы с данными для обучения системы
- Программа, реализующая рекомендательную систему
- Результаты тестирования программы
- Сравнение точности примененных алгоритмов
- Подготовленный отчет

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Apache Mahout — это библиотека для работы с алгоритмами машинного обучения, которая может быть использована как надстройка к Hadoop или самостоятельно. В библиотеке реализованы методы коллаборативной фильтрации, кластеризации и классификации.

Рекомендательные системы — программы, которые пытаются предсказать, какие объекты (фильмы, музыка, книги, новости, веб-сайты) будут интересны пользователю, имея определенную информацию о его профиле.

Две основные стратегии создания рекомендательных систем — фильтрация на основе содержания и коллаборативная фильтрация. При фильтрации на основе содержания создаются профили пользователей и объектов, профили пользователей могут включать демографическую информацию или ответы на определённый набор вопросов, профили объектов могут включать названия жанров, имена актёров, имена исполнителей и другую атрибутивную информацию в зависимости от типа объекта. Определяя расстояние можно рекомендовать или не рекомендовать объект пользователю.

Коллаборативная фильтрация, совместная фильтрация — это один из методов построения прогнозов (рекомендаций) в рекомендательных системах, использующий известные предпочтения (оценки) группы пользователей для прогнозирования неизвестных предпочтений другого пользователя. Его основное допущение состоит в следующем: те, кто одинаково оценивали какие-либо предметы в прошлом, склонны давать похожие оценки другим предметам и в будущем. Например, с помощью коллаборативной фильтрации музыкальное приложение способно прогнозировать, какая музыка понравится пользователю, имея неполный список его предпочтений (симпатий и антипатий). Прогнозы составляются индивидуально для каждого пользователя, хотя используемая информация собрана от многих участников. Тем самым коллаборативная фильтрация отличается от более простого подхода, дающего усреднённую оценку

для каждого объекта интереса, к примеру, базирующуюся на количестве поданных за него голосов.

Коллаборативная фильтрация, основанная на соседстве

Этот подход является исторически первым в коллаборативной фильтрации и используется во многих рекомендательных системах. В данном подходе для активного пользователя подбирается подгруппа пользователей схожих с ним. Комбинация весов и оценок подгруппы используется для прогноза оценок активного пользователя. У данного подхода можно выделить следующие основные шаги:

1. Присвоить вес каждому пользователю с учётом схожести его оценок и активного пользователя.
2. Выбрать несколько пользователей, которые имеют максимальный вес, то есть максимально похожи на активного пользователя. Данная группа пользователей и называется соседями.
3. Вычислить предсказание оценок активного пользователя для неоценённых им предметов с учётом весов и оценок соседей.

Коллаборативная фильтрация, основанная на модели

Данный подход предоставляет рекомендации, измеряя параметры статистических моделей для оценок пользователей. Модели разрабатываются с использованием интеллектуального анализа данных, алгоритмов машинного обучения, чтобы найти закономерности на основе обучающих данных.

Этот подход является более комплексным и даёт более точные прогнозы, так как помогает раскрыть латентные факторы, объясняющие наблюдаемые оценки.

Данный подход имеет ряд преимуществ. Он обрабатывает разреженные матрицы лучше, чем подход, основанный на соседстве, что в свою очередь помогает с масштабируемостью больших наборов данных.

Java библиотека [Mahout](#) позволяет работать с различными моделями – создавать, обучать модели на локальных данных или данных hdfs, задавая различные метрики, тестировать их работу. Полный список моделей и доступных метрик схожести можно просмотреть в официальной документации.

РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ

Рассмотрим рекомендательную систему на основе коллаборативной фильтрации. Она может быть пользователе-ориентированной (user-based) или свойство-ориентированной (item-based).

Рассмотрим пример построения простой пользователе-ориентированной системы рекомендаций с помощью Mahout. Входной файл содержит оценки, поставленный пользователем разным продуктам.

```
class RecommenderIntro {
public static void main(String[] args) throws Exception {
    DataModel model = new FileDataModel (new File("intro.csv"));
    UserSimilarity similarity = new PearsonCorrelationSimilarity
(model);
    UserNeighborhood neighborhood = new
NearestNUserNeighborhood (2, similarity, model);
    Recommender recommender = new
GenericUserBasedRecommender (model, neighborhood,
similarity);
    List<RecommendedItem> recommendations =
recommender.recommend(1, 1);
    for (RecommendedItem recommendation : recommendations) {
        System.out.println(recommendation);
    }
}
}
```

DataModel хранит и предоставляет доступ ко всем предпочтениям, пользователям и предметам, нужным для вычислений.

Реализация UserSimilarity обеспечивает представление о том, как похожи вкусы пользователей; она может быть основана на одной из множества метрик или вычислений. В данном случае используется коэффициент корреляции Пирсона.

Реализация UserNeighborhood определяет понятие группы пользователей, которые наиболее близки к данному пользователю. На основе этой группы будет делаться предсказание. Для создания группы пользователей необходимо определить её размер, метрику схожести и модель данных.

Наконец, реализация Recommender связывает предыдущие три компонента вместе, чтобы сделать рекомендации пользователям. Метод recommend(int userId, int number) принимает два параметра: пользователь и количество рекомендаций которые нужно сделать этому пользователю.

Библиотека Mahout предоставляет несколько вариантов реализации класса Recommender (табл. 1), основанные на различных алгоритмах поиска рекомендаций.

Таблица 1

Реализации класса Recommender

Реализация	Ключевые параметры	Особенности
GenericUserBasedRecommender	Метрика схожести пользователей, определение группы пользователей и её размер	Стандартная реализация; Работает быстро, когда количество пользователей относительно мало
GenericItemBasedRecommender	Метрика схожести предметов	Работает быстро, когда количество предметов относительно мало. Полезно, когда есть внешняя информация о схожести предметов

Таблица 1 (продолжение)

SlopeOneRecommender	Стратегия хранения разницы оценок	Быстро работает; Требуется больших предварительных вычислений
SVDRRecommender (Singular value decomposition)	Количество признаков	Хорошие результаты; Требуется больших предварительных вычислений
KnnItemBasedRecommender	Количество средних (k), метрика схожести предметов, размер группы	Хорошие результаты, когда количество предметов относительно мало
TreeClusteringRecommender	Количество кластеров, метрика схожести кластеров, метрика схожести пользователей	Быстро работает; Требуется больших предварительных вычислений

Для оценки работы системы рекомендации в Mahout используется класс `RecommenderEvaluator`. Для рассмотренного примера можем провести оценку модели следующим образом:

```
RandomUtils.useTestSeed();
DataModel model = new FileDataModel (new File("intro.csv"));
RecommenderEvaluator evaluator =
    new AverageAbsoluteDifferenceRecommenderEvaluator ();
```

```

RecommenderBuilder builder = new RecommenderBuilder() {
    @Override
    public Recommender buildRecommender(DataModel model)
    throws TasteException {
        UserSimilarity similarity = new PearsonCorrelationSimilarity
            (model);
        UserNeighborhood neighborhood =
            new NearestNUserNeighborhood (2, similarity, model);
        return new GenericUserBasedRecommender (model,
            neighborhood, similarity);
    }
};

double score = evaluator.evaluate(builder, null, model, 0.7, 1.0);
System.out.println(score);

```

Непосредственно оценка происходит после вызова метода `evaluate()`. На вход данный метод принимает объект `RecommenderBuilder`, который создаёт экземпляр системы рекомендаций, модель данных, процент исходных данных (оценок), которые будут использоваться для обучения и тестирования (в данном случае 70% для обучения, 30% для тестирования), процент пользователей, используемых в оценке. Результат выполнения функции `evaluate` – число, представляющее среднее отклонение предсказанной оценки от реальной, т.е. чем меньше данное число, тем лучше работает система.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Для выполнения задания использовать базу данных MovieLens любого размера:

<https://grouplens.org/datasets/movielens/>

Реализовать 2 системы рекомендаций фильмов (по варианту) для пользователя на основе его оценок. В системах, в которых используются метрики, реализовать как минимум 2 версии с применением разных метрик. Сравнить оценки правильности работы всех систем. Для сравнения запускать алгоритм оценки как минимум 10 раз и использовать среднее значение оценки для каждой из систем.

ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

Приложение должно быть реализовано на языке java. Для обучения рекомендательной системы можно пользоваться как файлами, размещенными в HDFS, так и файлами в локальной файловой системе.

ВАРИАНТЫ ЗАДАНИЙ

1. GenericUserBasedRecommender. Реализовать как минимум 2 версии с различными метриками
SlopeOneRecommener
2. GenericItemBasedRecommender. Реализовать как минимум 2 версии с различными метриками
SlopeOneRecommener
3. KnnItemBasedRecommender. Реализовать как минимум 2 версии с различными метриками
SVDRecommender
4. GenericItemBasedRecommender. Реализовать как минимум 2 версии с различными метриками
SVDRecommender
5. GenericUserBasedRecommender. Реализовать как минимум 2 версии с различными метриками
SVDRecommender

6. GenericItemBasedRecommender. Реализовать как минимум 2 версии с различными метриками
KnnItemBasedRecommender. Реализовать как минимум 2 версии с различными метриками
7. TreeClusteringRecommender. Реализовать как минимум 2 версии с различными метриками
SlopeOneRecommener
8. GenericUserBasedRecommender. Реализовать как минимум 2 версии с различными метриками
GenericItemBasedRecommender. Реализовать как минимум 2 версии с различными метриками
9. KnnItemBasedRecommender. Реализовать как минимум 2 версии с различными метриками
SlopeOneRecommener
10. TreeClusteringRecommender. Реализовать как минимум 2 версии с различными метриками
SVDRecommender

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Раскройте область применения библиотеки Apache Mahout.
2. Раскройте термин «рекомендательные системы».
3. Приведите классификацию алгоритмов коллаборативной фильтрации.
4. Опишите назначение класса DataModel.
5. Опишите назначение класса UserSimilarity.
6. Опишите назначение класса UserNeighborhood.
7. Опишите назначение класса Recommender.
8. Перечислите основные реализации класса Recommender в библиотеки Mahout.
9. Перечислите основные метрики схожести.
10. Опишите назначение класса RecommenderEvaluator.
11. Приведите методику оценки эффективности работы системы рекомендации.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 3 занятия (6 академических часов: 5 часов на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Номер варианта студенту выдается преподавателем.

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания (вариант), этапы выполнения работы (со скриншотами), результаты выполнения работы. выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Федин Ф.О. Анализ данных. Часть 1. Подготовка данных к анализу [Электронный ресурс] : учебное пособие / Ф.О. Федин, Ф.Ф. Федин. — Электрон. текстовые данные. — М. : Московский городской педагогический университет, 2012. — 204 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/26444.html>
2. Федин Ф.О. Анализ данных. Часть 2. Инструменты Data Mining [Электронный ресурс] : учебное пособие / Ф.О. Федин, Ф.Ф. Федин. — Электрон. текстовые данные. — М. : Московский городской педагогический университет, 2012. — 308 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/26445.html>
3. Чубукова, И.А. Data Mining [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : , 2016. — 470 с. — Режим доступа: <https://e.lanbook.com/book/100582>. — Загл. с экрана.
4. Воронова Л.И. Big Data. Методы и средства анализа [Электронный ресурс] : учебное пособие / Л.И. Воронова, В.И. Воронов. — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 33 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/61463.html>
5. Юре, Л. Анализ больших наборов данных [Электронный ресурс] / Л. Юре, Р. Ананд, Д.У. Джефффри. — Электрон. дан. — Москва : ДМК Пресс, 2016. — 498 с. — Режим доступа: <https://e.lanbook.com/book/93571>. — Загл. с экрана.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

6. Волкова Т.В. Разработка систем распределенной обработки данных [Электронный ресурс] : учебно-методическое пособие / Т.В. Волкова, Л.Ф. Насейкина. — Электрон. текстовые данные. — Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2012. — 330 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/30127.html>
7. Кухаренко Б.Г. Интеллектуальные системы и технологии [Электронный ресурс] : учебное пособие / Б.Г. Кухаренко. —

Электрон. текстовые данные. — М. : Московская государственная академия водного транспорта, 2015. — 116 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/47933.html>

8. Воронова Л.И. Интеллектуальные базы данных [Электронный ресурс] : учебное пособие / Л.И. Воронова. — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2013. — 35 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/63324.html>
9. Николаев Е.И. Базы данных в высокопроизводительных информационных системах [Электронный ресурс] : учебное пособие / Е.И. Николаев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 163 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/69375.html>

Электронные ресурсы:

10. <http://hadoop.apache.org/> (англ.)
11. <http://mahout.apache.org/> (англ.)