

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 – 2023

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ .....	4
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ.....	5
МЕТРИКИ СХОЖЕСТИ ПОЛЬЗОВАТЕЛЕЙ И ОБЪЕКТОВ .....	8
РЕАЛИЗАЦИИ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ В MANOUT....	11
РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ .....	13
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ .....	16
ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ .....	16
ВАРИАНТЫ ЗАДАНИЙ .....	16
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ .....	18
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ.....	18
ОСНОВНАЯ ЛИТЕРАТУРА .....	19
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА .....	19

## **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» и для специальности 10.05.03 «Информационная безопасность автоматизированных систем» 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 создания масштабируемых алгоритмов машинного обучения, которая может быть использована как надстройка над Hadoop или же самостоятельно. В библиотеке реализованы популярные методы машинного обучения, в частности алгоритмы коллаборативной фильтрации, кластеризации и классификации. Основная область применения Apache Mahout – реализация рекомендательных систем.

**Рекомендательные системы** – это программы, которые пытаются предсказать, какие объекты (фильмы, музыка, книги, новости, веб-сайты) будут интересны пользователю, имея определенную информацию о его профиле или о его поведении в прошлом.

Существуют два основных подхода к созданию рекомендательных систем – фильтрация на основе содержания и коллаборативная фильтрация. Также на практике используется гибридный метод построения рекомендаций, который представляет собой объединение вышеперечисленных подходов.

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. Исходя из схожести профилей можно рекомендовать или не рекомендовать объект пользователю.

При коллаборативной фильтрации, в отличие от фильтрации на основе содержания, используется информация о поведении пользователей в прошлом — например, информация о покупках или оценках. В этом случае не имеет значения, с какими типами объектов ведётся работа, но при этом могут учитываться неявные характеристики, которые сложно было бы учесть при создании профиля. Именно этот подход использует Apache Mahout для построения рекомендательных систем.

**Коллаборативная фильтрация, совместная фильтрация** – это метод 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 спрогнозировать, 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 голосов.

Коллаборативная фильтрация в свою очередь делится на несколько основных видов, а именно: основанную на соседстве, основанную на модели, а также гибридную, объединяющую первые два вида.

**Коллаборативная фильтрация, основанная на соседстве.** Также известна как пользователь-ориентированная (user-based). 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. Высчитать предсказание оценок активного пользователя для неоценённых им предметов с учётом весов и оценок соседей.

**Коллаборативная фильтрация, основанная на модели.** Также известна как свойство-ориентированная (item-based). 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.

**Гибридный подход.** Данный подход объединяет в себе подход, основанный на соседстве и основанный на модели. Гибридный подход является самым распространённым при разработке рекомендательных систем для коммерческих сайтов, так как он помогает преодолеть ограничения изначального оригинального подхода (основанного на соседстве) и улучшить качество предсказаний. Этот подход также позволяет преодолеть проблему разреженности данных и потери информации. Однако данный подход сложен и дорог в реализации и применении.

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.

## МЕТРИКИ СХОЖЕСТИ ПОЛЬЗОВАТЕЛЕЙ И ОБЪЕКТОВ

Основа любого рекомендательного алгоритма – метрика для определения схожести двух объектов.

Как было отмечено ранее, рекомендательные системы на основе коллаборативной фильтрации, как правило, делятся на два вида: основанные на соседстве или пользователи-ориентированные, а также основанные на модели или свойство-ориентированные.

В случае с пользователи-ориентированными системами метрика определяет близость друг к другу двух пользователей (их предпочтений, интересов), а в случае со свойство-ориентированными – схожесть двух объектов по определённым критериям.

По своей сути любая реализация метрики представляют собой функцию, возвращающую значение в диапазоне от -1 до 1, причём 1 в таком случае будет означать идеальное сходство.

В качестве метрик схожести наиболее часто используются следующие:

- Косинусный коэффициент;
- Коэффициент корреляции Пирсона;
- Коэффициент корреляции Спирмана;
- Коэффициент Танимото;
- Евклидова метрика;
- Логарифмическое правдоподобие.

В Mahout метрики схожести представлены в виде различных реализаций двух классов: UserSimilarity для пользователи-ориентированных систем и ItemSimilarity для свойство-ориентированных.

Единственным обязательным параметром для всех реализаций метрик в Mahout является модель данных (DataModel), содержащая данные о предпочтениях.

Рассмотрим реализации подробнее (табл. 1).



Таблица 1

## Реализации классов UserSimilarity и ItemSimilarity

Реализация	Описание
PearsonCorrelationSimilarity (ItemSimilarity, UserSimilarity)	Реализация коэффициента корреляции Пирсона. Данную корреляцию можно интерпретировать как косинус угла между двумя векторами, определяемыми значениями предпочтений пользователей. Данная корреляция центрирует свои данные, т.е. сдвигает значения предпочтений пользователя так, чтобы каждое из их средних значений было равно 0.
UncenteredCosineSimilarity (ItemSimilarity, UserSimilarity)	Реализация косинусного коэффициента. Результатом является косинус угла, образованного между двумя векторами предпочтений пользователей. В отличие от корреляции Пирсона, данная корреляция не центрирует свои данные.
EuclideanDistanceSimilarity (ItemSimilarity, UserSimilarity)	Реализация Евклидовой метрики. Основывается на евклидовом расстоянии между двумя пользователями. Фактически представляет собой нахождение квадратного корня из суммы квадратов различий в положении (предпочтении) по каждому измерению.

Таблица 1 (продолжение)

<p>TanimotoCoefficientSimilarity (ItemSimilarity, UserSimilarity)</p>	<p>Реализация подоби́я на основе коэффициента Танимото или расширенного коэффициента Жаккара. Предназначена для «двоичных» наборов данных, где пользователь либо выражает общее предпочтение «да» для объекта, либо не имеет никаких предпочтений. Возвращаемое значение находится в границах от 0 до 1.</p>
<p>LogLikelihoodSimilarity (ItemSimilarity, UserSimilarity)</p>	<p>Реализация коэффициента логарифмического правдоподобия. Метод, лежащий в основе алгоритма, заключается в использовании оценки для анализа количества оценок.</p>
<p>SpearmanCorrelationSimilarity (UserSimilarity)</p>	<p>Реализация коэффициента корреляции Спирмана. В отличие от корреляции Пирсона, сравнивает относительное ранжирование значений предпочтений, а не самих значений предпочтений. То есть предпочтения каждого пользователя сортируются, а затем присваивается ранг в качестве значения предпочтения, причем 1 присваивается наименее предпочтительному элементу.</p>

## РЕАЛИЗАЦИИ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ В MAHOUT

Библиотека Mahout предоставляет несколько вариантов реализаций рекомендательных систем на основе различных популярных алгоритмов. За это отвечает класс Recommender. Рассмотрим его основные реализации подробнее (табл. 2).

Таблица 2

Реализации класса Recommender

Реализация	Ключевые параметры	Особенности
GenericUserBasedRecommender	Модель данных, определение близкой группы пользователей, метрика схожести пользователей.	Стандартная реализация. Работает быстро, когда количество пользователей относительно мало.
GenericItemBasedRecommender	Модель данных, метрика схожести объектов.	Работает быстро, когда количество предметов относительно мало. Полезно, когда есть внешняя информация о схожести предметов.
SlopeOneRecommender	Модель данных (стратегия хранения разницы оценок).	Быстро работает. Требуется больших предварительных вычислений.

Таблица 2 (продолжение)

SVDRecommender (Singular Value Decomposition)	Модель данных, количество признаков.	Хорошие результаты. Требует больших предварительных вычислений.
KnnItemBasedRecommender	Модель данных, количество средних (k), метрика схожести предметов, размер группы.	Хорошие результаты, когда количество предметов относительно мало.
TreeClusteringRecommender	Модель данных, количество кластеров, метрика схожести кластеров, метрика схожести пользователей.	Быстро работает. Требует больших предварительных вычислений.

## РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ

Рассмотрим пример построения простой пользователь-ориентированной (user-based) системы рекомендаций на основе коллаборативной фильтрации с помощью Java библиотеки Mahout.

В качестве Mahout реализации системы рекомендаций будем использовать `GenericUserBasedRecommender`, а в качестве реализации метрики – `PearsonCorrelationSimilarity`.

Входной файл содержит оценки, поставленные 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<recommendations> = recommender.recommend(1,
1);
        for (RecommendedItem recommendation :
recommendations) {
            System.out.println(recommendation);
        }
    }
}
```

`DataModel` представляет собой модель данных, которая хранит и предоставляет доступ ко всем предпочтениям, пользователям и предметам, нужным для вычислений. Входным параметром модели в данном случае является csv-файл с оценками пользователя различным продуктам.

Реализация `UserSimilarity` обеспечивает представление о том, как похожи вкусы пользователей; она может быть основана на одной из

множества метрик или вычислений. Как было отмечено ранее, в данном случае используется коэффициент корреляции Пирсона (PearsonCorrelationSimilarity), принимающий в качестве параметра объявленную ранее модель данных.

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.

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 для пользователя на основе его оценок. Использовать реализации рекомендательных систем и метрик, указанные в варианте задания. 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<br>UncenteredCosineSimilarity; SlopeOneRecommener    | с | метрикой |
| 2. | GenericUserBasedRecommender<br>LogLikelihoodSimilarity; SlopeOneRecommener       | с | метрикой |
| 3. | GenericUserBasedRecommender<br>EuclideanDistanceSimilarity; SlopeOneRecommener   | с | метрикой |
| 4. | GenericUserBasedRecommender<br>TanimotoCoefficientSimilarity; SlopeOneRecommener | с | метрикой |
| 5. | GenericUserBasedRecommender<br>SpearmanCorrelationSimilarity; SlopeOneRecommener | с | метрикой |
| 6. | GenericItemBasedRecommender<br>PearsonCorrelationSimilarity; SlopeOneRecommener  | с | метрикой |



7.	GenericItemBasedRecommender	с	метрикой
	LogLikelihoodSimilarity; SlopeOneRecommender		
8.	GenericItemBasedRecommender	с	метрикой
	EuclideanDistanceSimilarity; SlopeOneRecommender		
9.	GenericItemBasedRecommender	с	метрикой
	TanimotoCoefficientSimilarity; SlopeOneRecommender		
10.	GenericItemBasedRecommender	с	метрикой
	UncenteredCosineSimilarity; SlopeOneRecommender		
11.	KnnItemBasedRecommender	с	метрикой
	PearsonCorrelationSimilarity; SVDRecommender		
12.	KnnItemBasedRecommender с метрикой LogLikelihoodSimilarity; SVDRecommender		
13.	KnnItemBasedRecommender	с	метрикой
	EuclideanDistanceSimilarity; SVDRecommender		
14.	KnnItemBasedRecommender	с	метрикой
	TanimotoCoefficientSimilarity; SVDRecommender		
15.	KnnItemBasedRecommender	с	метрикой
	UncenteredCosineSimilarity; SVDRecommender		
16.	TreeClusteringRecommender	с	метрикой
	PearsonCorrelationSimilarity; SVDRecommender		
17.	TreeClusteringRecommender с метрикой LogLikelihoodSimilarity; SVDRecommender		
18.	TreeClusteringRecommender	с	метрикой
	EuclideanDistanceSimilarity; SVDRecommender		
19.	TreeClusteringRecommender	с	метрикой
	TanimotoCoefficientSimilarity; SVDRecommender		
20.	TreeClusteringRecommender	с	метрикой
	UncenteredCosineSimilarity; SVDRecommender		

## **КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ**

1. Раскройте область применения библиотеки Apache Mahout.
2. Раскройте термин «рекомендательные системы».
3. Раскройте термин «коллаборативная фильтрация».
4. Приведите и опишите основные виды коллаборативной фильтрации.
5. Перечислите основные метрики схожести.
6. Опишите назначение классов UserSimilarity и ItemSimilarity.
7. Перечислите основные реализации классов UserSimilarity и ItemSimilarity в библиотеке Mahout.
8. Опишите назначение класса Recommender.
9. Перечислите основные реализации класса Recommender в библиотеке Mahout.
10. Опишите назначение класса DataModel.
11. Опишите назначение класса UserNeighborhood.
12. Опишите назначение класса RecommenderEvaluator.
13. Приведите методику оценки эффективности работы системы рекомендаций.

## **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. Джеффри. — 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.)