

EPL451: Data Mining on the Web – Lab 6



**University of Cyprus
Department of
Computer Science**

Παύλος Αντωνίου

Γραφείο: B109, ΘΕΕ01

What is Mahout?



- Provides Scalable Machine Learning and Data Mining
- Runs on top of Apache Hadoop
- Uses the map/reduce paradigm
- <http://mahout.apache.org>
- Mahout is the person who rides the elephant



Who uses Mahout?



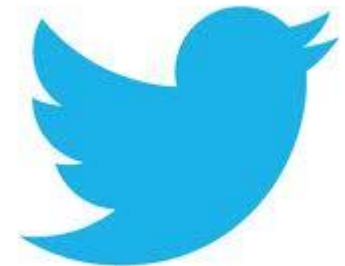
amazon.com[®]



foursquare



Adobe



... and many more

Motivation



- **Large datasets** need a **parallel approach** or processing will be infeasible.
 - We've seen how Hadoop is used for parallel processing.
 - Mahout sits on top of Hadoop and adds **machine learning techniques** and **artificial intelligence**.
 - This provides the ability to process large datasets and export useful information.
-

Mahout algorithms



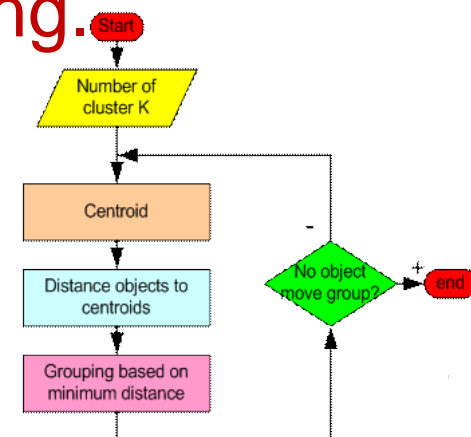
Mahout currently provides the following algorithms:

- **Clustering (ομαδοποίηση)**
 - takes data points and groups topically related data points (unsupervised learning)
- **Classification (κατηγοριοποίηση)**
 - tries to assign data points to a correct category based on prior knowledge (supervised learning)
- **Recommendation**
 - takes user behavior and tries to predict items users might like

Clustering



- Clustering is the sorting of items into groups (or categories)
- Items in the same group are more similar to one another than to items in other groups.
- Grouping is done using the notion of similarity
- Mahout uses different algorithms for clustering: k-means, Canopy (deprecated), Fuzzy k-means, Streaming k-means, Spectral clustering.
- Distance measure: Euclidean, Cosine, Tanimoto, Manhattan ...



Classification



- Classification is the automatic categorization of items into categories.
 - The categories are **predetermined**.
 - Classifier is trained from a dataset which was already manually labeled.
 - Then the classifier should be able to categorize items automatically (hopefully with high accuracy).
-

Classification



Computers & Technology



Fruits



- New item?



Recommendation



- Mahout provides a full framework for storage, online and offline computation of recommendations
 - Offline is usually done via Hadoop
 - Like clustering, there is a notion of **similarity in users or items**
 - User-based vs. Item-based recommendations
-

Mahout Recommenders



- **Item-based recommender**

- Προτείνει αντικείμενα που μπορεί να αρέσουν σε ένα χρήστη λαμβάνοντας υπόψη άλλα αντικείμενα που είναι παρόμοια με αυτά που έχει προτιμήσει στο παρελθόν

- Δεν εμπλέκονται δεδομένα άλλων χρηστών

Continue Shopping: Recommended for you



Research Methods in Applied...
Zoltan Dörnyei
★★★★☆ (2)
Paperback
£28.03
[Fix this recommendation](#)



Kenwood SB056 Smoothie 2GO Black
★★★★☆ (66)
£29.99
[Fix this recommendation](#)



How Languages are Learned
Patsy M. Lightbown
★★★★☆ (9)
Paperback
£20.46
[Fix this recommendation](#)

- **User-based recommender**

- Προτείνει αντικείμενα που μπορεί να αρέσουν σε ένα χρήστη λαμβάνοντας υπόψη αντικείμενα που προτίμησαν παρόμοιοι χρήστες στο παρελθόν

Customers Who Bought This Item Also Bought



StarTech 11 Piece PC Computer Tool Kit with Carrying Case
★★★★☆ (7)
£11.76



58 Piece Computer Repair Tool Kit
★★★★☆ (26)
£26.45



Hama PC Tool Kit (Professional)
★★★★☆ (40)
£18.28

Recommendations in Mahout



ITEM BASED

```
for every item i that u has no preference for yet
  for every item j that u has a preference for
    compute a similarity s between i and j
    add u's preference for j, weighted by s, to a running
    average
return the top items, ranked by weighted average
```

USER BASED

```
for every item i that u has no preference for yet
  for every other user v that has a preference for i
    compute a similarity s between u and v
    add v's preference for i, weighted by s, to a running
    average
return the top items, ranked by weighted average
```

Κλάσεις mahout recommender



- **Recommender** – Η γενική διεπαφή για τους recommenders. Συνήθως ο προγραμματιστής θα χρησιμοποιήσει τις κλάσεις **GenericUserBasedRecommender** για user-based recommenders και **GenericItemBasedRecommender** για item-based recommenders.
 - Η κλάση **CachingRecommender** μπορεί να χρησιμοποιηθεί παράλληλα με ένα από τους recommenders για να κρατήσει τα αποτελέσματα των εισηγήσεων στη μνήμη.
-

Κλάσεις mahout recommender



- **UserSimilarity** – Η υλοποίηση της ιδέας της ομοιότητας μεταξύ χρηστών και που χρησιμοποιείται στους user-based recommenders.
- **ItemSimilarity** – Η υλοποίηση της ιδέας της ομοιότητας μεταξύ αντικειμένων και που χρησιμοποιείται στους item-based recommenders.
- **UserNeighborhood** – Για να πραγματοποιηθούν συστάσεις σε ένα χρήστη με τη χρήση ενός user-based recommender, πρέπει να βρεθεί η «γειτονιά» με τους πιο κοντινούς χρήστες. Χρησιμοποιείται αυτή η κλάση η οποία βρίσκει π.χ. τους 10 πιο κοντινούς χρήστες σε αυτό το χρήστη.

Κλάσεις mahout recommender



- **DataModel** – Η διεπαφή αυτή καθορίζει την πηγή των δεδομένων της μορφής (user,item,value)
 - Παρέχει μεθόδους όπως `getNumUsers()` για να λάβουμε τον αριθμό των χρηστών ή `getPreferencesForItem(itemID)` για να λάβουμε όλα τις προτιμήσεις για ένα αντικείμενο
 - Διαβάστε περισσότερες πληροφορίες [εδώ](#)
- Τα δεδομένα μπορεί να προέρχονται:
 - από βάσεις δεδομένων (π.χ. MySQL) οπότε αν χρησιμοποιείται η κλάση **MySQLJDBCDataModel** η οποία υλοποιεί (implements) την διεπαφή *DataModel*
 - από αρχεία οπότε αν χρησιμοποιείται η κλάση [**FileDataModel**](#) (η οποία επίσης υλοποιεί την διεπαφή *DataModel*) που δέχεται αρχείο εισόδου της μορφής `userID,itemID[,preference[,timestamp]]`

Εργασία 1



- Το Mahout v0.11.1 είναι εγκατεστημένο στο VM
 - Κατεβάστε το GroupLens Movie dataset από την ιστοσελίδα του μαθήματος
 - <http://www.cs.ucy.ac.cy/courses/EPL451/labs/LAB06/ml-1m.zip>
- ή
- <http://www.grouplens.org/system/files/ml-1m.zip>
-

GroupLens Movie Data



- Τα δεδομένα εισόδου για την εργασία αυτή βασίζονται σε 1,000,209 ανώνυμες **βαθμολογίες** περίπου 3,900 **ταινιών** από 6,040 MovieLens **χρήστες**. Το αρχείο zip περιέχει 4 αρχεία:
 - movies.dat (movie ids με τίτλο και κατηγορία/ες)
 - ratings.dat (βαθμολογίες ταινιών)
 - README
 - users.dat (πληροφορίες χρηστών)
-

Περιγραφή αρχείου ratings.dat



- Το αρχείο βαθμολογιών είναι το πιο ενδιαφέρον διότι αποτελεί την πιο σημαντική είσοδο στο recommendation job. Κάθε γραμμή έχει τη μορφή:
 - UserID::MovieID::Rating::Timestamp
 - όπου
 - UserIDs είναι ακέραιοι
 - MovieIDs είναι ακέραιοι
 - Ratings είναι 1 έως 5 “αστέρια” (ακέραιοι)
 - Timestamp σε δευτερόλεπτα από 1/1/1970
 - Κάθε χρήστης έχει τουλάχιστον 20 βαθμολογήσεις
-

Προ-επεξεργασία δεδομένων



- Αυτό το αρχείο δεν είναι ακριβώς στη μορφή που το προτιμά το Mahout, αλλά μπορεί εύκολα να μετατραπεί.
 - Το Mahout περιμένει CSV αρχείο όπου η κάθε γραμμή έχει τη μορφή:
 - userID, itemID, value
 - Αποσυμπιέστε το zip αρχείο
 - cd στον κατάλογο ml-m1
 - Μετατρέψτε τα δεδομένα στο ζητούμενο 'csv'
 - `tr -s ':' ',' < ratings.dat | cut -f1-3 -d, > ratings.csv`
 - `tr -s ':' ',' < users.dat | cut -f1-3 -d, > users.csv`
-

Recommendation with Mahout



- Item-base recommender class:
org.apache.mahout.cf.taste.hadoop.item.RecommenderJob
- Η είσοδος στο job αυτό θα είναι το “ratings.csv” που δημιουργήσαμε προηγουμένως:
 - userID, itemID, value
- και η έξοδος θα είναι άλλο CSV αρχείο της μορφής:
 - userID [itemID, score, ...]
- που αντιπροσωπεύει τα userIDs με τα recommended itemIDs μαζί με τα preference scores.

Recommendation with Mahout



- Start hadoop (`start-all.sh`)
- Put data on Hadoop through terminal
 - `hadoop fs -put ratings.csv /user/csdeptucy/mahout/ratings.csv`
 - `hadoop fs -put users.csv /user/csdeptucy/mahout/users.csv`
- or use the eclipse plugin
- cd to Mahout directory
- `./bin/mahout recommenditembased --input /user/csdeptucy/mahout/ratings.csv --output /user/csdeptucy/output/ --tempDir /tmp --numRecommendations 10 --similarityClassname SIMILARITY_COOCCURRENCE`

Other similarity classnames



- Also, on similarityClassname, you can choose anyone you like from the above list:
 - SIMILARITY_COOCCURRENCE
 - SIMILARITY_LOGLIKELIHOOD
 - SIMILARITY_TANIMOTO_COEFFICIENT
 - SIMILARITY_CITY_BLOCK
 - SIMILARITY_COSINE
 - SIMILARITY_PEARSON_CORRELATION
 - SIMILARITY_EUCLIDEAN_DISTANCE
-

Recommendation with Mahout



- **Recommendation Output**

```
1 [3740:5.0,47:5.0,3108:5.0,2133:5.0,2420:5.0,368:5.0,832:5.0,1678:5.0,2857:5.0,2478:5.0]
2 [3260:5.0,1407:5.0,3148:5.0,3614:5.0,1805:5.0,45:5.0,914:5.0,838:5.0,2616:5.0,3504:5.0]
3 [368:5.0,1129:5.0,2133:5.0,3740:5.0,47:5.0,2478:5.0,832:5.0,3108:5.0,3697:5.0,1610:5.0]
4 [1333:5.0,368:5.0,1215:5.0,1748:5.0,3033:5.0,3608:5.0,1283:5.0,1263:5.0,2478:5.0,953:5.0]
5 [3504:5.0,1231:5.0,3108:5.0,1358:5.0,1562:5.0,2694:5.0,832:5.0,1678:5.0,2541:5.0,47:5.0]
6 [3755:5.0,2478:5.0,3108:5.0,2133:5.0,2420:5.0,368:5.0,832:5.0,1678:5.0,2779:5.0,3617:5.0]
7 [2478:5.0,1333:5.0,293:5.0,1201:5.0,1371:5.0,1339:5.0,610:5.0,1263:5.0,16:5.0,2989:5.0]
8 [3260:5.0,2599:5.0,3148:5.0,3614:5.0,1805:5.0,1566:5.0,914:5.0,838:5.0,2616:5.0,1407:5.0]
9 [2336:5.0,1320:5.0,3108:5.0,368:5.0,1610:5.0,1676:5.0,832:5.0,1678:5.0,1358:5.0,3526:5.0]
10 [1805:5.0,3507:5.0,2616:5.0,2966:5.0,2046:5.0,3793:5.0,838:5.0,3148:5.0,3614:5.0,1407:5.0]
```

...


- Each line represents the recommendation for a user. The first number is the user id and the 10 number pairs represents a movie id and a score.
- If we are looking at the first line for example, it means that for the user 1, the 10 best recommendations are for the movies 3740, 47, 3108, 2133, 2420, 368, 832, 1678, 2857, 2478.

Εργασία 2



- Θα χρησιμοποιήσουμε το αρχείο **movieRatings.dat** που περιέχει βαθμολογίες (ratings) κάποιων χρηστών για διάφορες ταινίες.
 - Το αρχείο περιέχει περίπου 1 εκατομμύριο βαθμολογίες. Το αρχείο είναι σε μορφή CSV (comma-separated values) ως εξής:
`userid,itemid,prefValue`
όπου `prefValue` είναι μια βαθμολογία από 1 μέχρι 5 την οποία έδωσε ένας χρήστης (`userid`) για μια ταινία (`itemid`).
-



- Θέλουμε ένα πρόγραμμα σε java για να μπορούμε να κάνουμε movie recommendations
 - σχετικές κλάσεις του Mahout και Hadoop πρέπει να γίνουν import
- Πρόβλημα:
 - Ένας αριθμός από jar files (και τα dependencies) πρέπει να κατεβεί και να προστεθεί στο classpath
 - Δύσκολο να ανιχνεύσουμε όλα τα dependency libraries
- Λύση: Apache **Maven**TM
 - tool for building and managing any Java-based project
 - excellent dependency management mechanism
 - easy build process

Εργασία 2 – create Maven project



- Εγκατάσταση:
 - `sudo apt-get install maven`
- Δημιουργία Maven project
 - `mvn archetype:generate -DgroupId=com.csdeptucy.app`
– `-DartifactId=recommendation`
– `-DarchetypeArtifactId=maven-archetype-quickstart`
– `-DinteractiveMode=false`
- Πλοηγηθείτε στο project folder
 - `cd recommendation`
 - see project structure [here](#)
- POM.xml file
 - core of project's configuration

POM file example



```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.csdeputy.app</groupId>
    <artifactId>recommendation</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>recommendation</name>
    <url>http://maven.apache.org</url>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.8.2</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

Maven phases



- Most common lifecycle **phases**:
 - **validate**: validate the project is correct and all necessary information is available
 - **compile**: compile the source code of the project
 - **test**: test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
 - **package**: take the compiled code and package it in its distributable format, such as a JAR
 - **integration-test**: process and deploy the package if necessary into an environment where integration tests can be run
 - **verify**: run any checks to verify the package is valid and meets quality criteria
 - **install**: install the package into the local repository, for use as a dependency in other projects locally
 - **deploy**: done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects
 - **clean**: cleans up artifacts created by prior builds
 - **site**: generates site documentation for this project
- Phases may be executed in sequence
 - `mvn clean package`

Δοκιμάστε την αρχική εφαρμογή



- Τρέξτε το JAR file με την πιο κάτω εντολή:
 - `java -cp target/recommendation-1.0-SNAPSHOT.jar com.csdeptucy.app.App`
- Θα τυπώσει: `Hello World!`

Εργασία 2



- Κατεβάστε το zip file
 - <http://www.cs.ucy.ac.cy/courses/EPL451/labs/LAB06/LAB06.zip>αποσυμπιέστε, και βάλτε το pom.xml και τα 2 .dat files στον τρέχοντα φάκελο και το .java file στον φάκελο src/main/java/com/csdeptucy/app
- Μελετήστε τον κώδικα και τρέξτε το με την πιο κάτω εντολή:
 - `java -cp target/recommendation-1.0-SNAPSHOT-jar-with-dependencies.jar com.csdeptucy.app.UserRecommender`

<https://mahout.apache.org/users/recommender/userbased-5-minutes.html>

Εργασία 3



- Το αρχείο **datingRatings.dat** περιέχει περίπου 1.7 εκατομμύρια βαθμολογίες χρηστών σε προφίλ άλλων χρηστών που έγιναν σε μια σελίδα γνωριμιών. Το αρχείο είναι στη μορφή:
 - userID, profileID, rating
 - όπου userID είναι το μοναδικό ID του χρήστη, profileID είναι το ID του προφίλ που βαθμολογήθηκε και rating είναι η βαθμολογία που έδωσε ο χρήστης από 1 μέχρι 10 (10 το πιο ψηλό).
 - Δημιουργείστε μια άλλη κλάση στο **ίδιο maven project** και ονομάστε την `DatingRecommender`.
-

Εργασία 3



- Γράψτε τον κώδικα που χρειάζεται για να εκτελέσετε **item-based recommendation** με αλγόριθμο similarity τον **EuclideanDistanceSimilarity**.
 - Θα πρέπει προτείνετε **500 προφίλ χρηστών** (=items) που μπορεί να ενδιαφέρουν τον χρήστη με **userID=19**.
 - Συμβουλευτείτε τις σημειώσεις στο πρώτο μέρος του εργαστηρίου για το ποιες κλάσεις θα χρησιμοποιήσετε για item-based recommendation.
-

Εργασία 3



- Η κλάση **GenericItemBasedRecommender** θα πρέπει να παίρνει σαν παραμέτρους μεταβλητές του τύπου (**FileDataModel**, **ItemSimilarity**).
 - Η χρήση **UserNeighborhood** δεν χρειάζεται πλέον διότι θα εκτελέσουμε item-based recommendation και όχι user-based.
 - Περιγραφή mahout API:
<https://builds.apache.org/job/Mahout-Quality/javadoc>
-

Other Mahout examples



- Twenty Newsgroups Classification Example:
<https://mahout.apache.org/users/classification/twenty-newsgroups.html>
-