Information Storage and Retrieval (CS 5604)
Collaborative Filtering
4/28/2016

Tianyi Li, Pranav Nakate, Ziqian Song

Department of Computer Science
Blacksburg, Virginia – 24061

Dr. Edward A. Fox
Virginia Tech
Agenda

- Role and goal
- User-based recommendation
  - Recommendation process
  - Algorithm
  - Implementation
- Item-based recommendation
  - Overview
  - Algorithms and Implementation
- Build Process
Project goal

• Our project is to serve the Integrated Digital Event Archiving and Library (IDEAL) project.
• IDEAL project provides services for searching, browsing, analysing, and visualization of over 1 billion tweets and over 65 million webpages.
• Building a recommendation system and recommending tweets and webpages to assist users searching and browsing the IDEAL collection.
  • User-based recommendation
  • Item-based recommendation
User-based recommendation vs. Item-based recommendation

**User recommendation**
- Clicking history
  - User-item matrix
determining relationships
  between user and item.

**Item recommendation**
- Content similarity
  - Build an item-item matrix
determining relationships
  between pairs of items.
Data requirement

User ID

tweet ID/webpage ID

Click (1/0)

Item recommendation

tweet ID/webpage ID

Content

(sim) tweet ID/webpage ID

User ID

(sim) tweet ID/webpage ID

simScore

simScore
val data = sc.textFile("fakeUser.data")
val ratings = data.map(_.split(',')).match { case Array(user, item, rate) =>
  Rating(user.toInt, item.toInt, rate.toDouble)
}

// Build the recommendation model using ALS
val rank = 10
val numIterations = 10
val model = ALS.train(ratings, rank, numIterations, 0.01)
val productuser = model.recommendProductsForUsers(3).collect()
Algorithm: Recommendation strategy

- Neighborhood methods
- Latent factor models
Algorithm: Matrix factorization

- ALS (alternating least squares)
  - Parallelization
  - Implicit data
Implementation

Randomly generated user information (user_id, document_id, rating_score)

model = ALS.train([parameters...])

new user data

recommendations (array of rdd objects)

user_id, user_id, document_id, predicted_rating_score

[65, Array(Rating(65, 468, 0.9800179177268962)), Rating(65, 797, 0.9800179177268962), Rating(65, 515, 0.6351993712498189))

HBase

result table
Item-based recommendations

• Recommendations based on the text of the documents
• Useful for retrieving related documents
• System Pipeline
Implementation details

• Document preprocessing - Word lemmatization
• Feature extraction and transformation using TF-IDF method
• Repartitioning the RDD to increase parallelization
• DIMSUM algorithm by Twitter!
• Results collection and post-processing
• Configuration parameters for spark job - Similarity threshold (DIMSUM), spark executor memory, shuffle memory size
Build process

- Code compilation
- JAR file creation
- SBT script
- Spark-submit
- Evaluation - Mean squared error (MSE)
Acknowledgement

• We want to thank Dr. Fox and our GRAs Mohamed Magdy Farag and Sunshin Lee for guiding and advising us.
• Also grateful for the effort by Front-end and Solr team to communicate and provide the data we needed to implement the system.
• We thank the whole class for discussing the problems and learning about information retrieval together.
Thank you!

Q&A