Mining Massive Datasets

Overview of Relevant Topics

Artur Andrzejak
http://pvs.ifi.uni-heidelberg.de

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Lecture Contents

Topics in **red** are relevant for the final exam

Programming and Frameworks /1

- [s01] Distributed Processing: Motivation
- [s01] Programming Paradigms
- [s01+s02] Spark Programming: Introduction
- [s02] Spark Programming: Example
- [s03] Crash Course NumPy
- [s04/s08] Spark: DataFrames and Datasets
- [s04] MapReduce (M-R): Fundamentals
- [s04] M-R: Advanced Concepts
- [s04] M-R: Expressing Algorithms in MapReduce (without “Join”)

Programming and Frameworks /2

- [s08] Spark: Execution Details
- [s11/s12] Spark Streaming
  - Overview, basic programming (concepts), [s12] Advanced Programming, Window Operations
- [s11] GraphX (not relevant)

Recommender Systems /1

- [s05] Recommender Systems: Motivation
  - Relevant only: formal model, utility matrix
- [s05] Content-based Recommendations
  - General idea, item profiles (without TF-IDF)
  - Without "Example: Star-Based Ratings"
  - Prediction and recommendation (without pros/cons)
- [s05] Collaborative Filtering
  - Predictions using similar users (user-user)
  - Item-Item Collaborative Filtering
  - Without: Common Practice, pros/cons
- [s05] Remarks & Practical Tips

Recommender Systems /2

- [s06] The Netflix Prize: Introduction
- [s06] Contrasting Recommendation Methods
  - Note: good to get an overview / comparison
- [s06] Latent Factor Models (intro)
- [s06] Finding the Latent Factors
  - Especially: concept of regularization (no formulas)
- [s06] Optimizing by Stochastic Gradient Descent
- [s06] The Netflix Prize: The Winner

Clustering

- [s02] Clustering: Introduction
- [s02] k-Means Clustering
- [s02] Implementing k-Means in Spark
- [s03] Hierarchical Clustering
  - Without “Hierarchical Clustering in Spark”
- [s03] Clustering: Metrics and Evaluation
  - Without “Dendrograms” and “Silhouetten-Plots”

Locality Sensitive Hashing

- [s07] A Word on Hash Functions and Data Structures
- [s07] Locality Sensitive Hashing: Intro
  - Relevant: Problem and Jaccard distance/similarity
- [s07] Shingling
- [s07/08] MinHashing
  - Especially relevant are:
    - min-"hash" function $h_\pi(C)$
    - Min-Hash property (without proof)
    - Min-Hash Signatures (without "Implementation")
- [s08] Locality Sensitive Hashing
  - Idea of similarity search via candidate pairs
  - Idea of using bands / Example of Bands
  - Tradeoff and analysis of LSH

Online Advertising

- [s09] Online Bipartite Matching
- [s09] Web Advertising
  - Performance-based Advertising, Adwords Problem
  - Greedy algorithm and bad scenario for it
- BALANCE Algorithm
  - Algorithm “rule” and example
  - BALANCE: Analyzing BALANCE (no slides 31-32)
  - BALANCE: General Result
  - BALANCE: Worst case and analysis (slides 34-38)
  - Generalized BALANCE

Association Rule Discovery [s10]

- The Market-Basket Model
- Outline and introduction:
  - Frequent Itemsets, Confidence and Interest, Mining Association Rules, Example
- Finding Frequent Itemsets
  - Understand the computation model and bottleneck(s)
- A-Priori Algorithm
  - Focus on this one!
- PCY (Park-Chen-Yu) Algorithm

Mining Data Streams

- [s11] Mining Data Streams – Introduction
- [s11] Filtering Data Streams
  - Problem statement and why a hash table is bad
  - First-Cut solution
  - Bloom Filter
- [s11] Sampling a fixed proportion
- [s12] Sampling a fixed-size sample
- (Repeated) [s11/s12] Spark Streaming
  - Overview, basic programming (concepts), [s12] Advanced Programming, Window Operations

Link Analysis

- [s12] Infinite Data (introduction)
- [s12] PageRank: the “Flow” Formulation
- [s12] Random Walk Interpretation
- [s13] PageRank: The Google Formulation
- [s13] RageRank: How do we actually compute the PageRank?
- (Repeated) [s13] GraphX (not relevant)

Fragen und Antworten /1

- Kommen Rechenaufgaben dran? Wird es auch aufwendige Aufgaben geben, mit Matrizen-operationen? Sind Taschenrechner erlaubt?
  - Nur kleine Beispiele, keine aufwendigen Rechenaufgaben (können im Kopf berechnet werden)
  - Taschenrechner sind OK, nicht-programmierbar

Fragen und Antworten /2

- Kommen Programmieraufgaben dran?
  - Ja, es wird Programmieraufgaben (Spark) geben
  - Der Klausur wird eine Liste der relevanten Spark-Operationen (mit Parameter) beigefügt
- Wie nah an echtem Spark-Code muss der Code sein?
  - Man sollte zeigen, dass man die Konzepte verstanden hat, Syntax ist weniger wichtig
  - Falls Name/Syntax unsicher: kurz beschreiben, was die Funktion/Transformation tun soll
- Welche Spark-Operationen muss man kennen?
  - Nur solche, die in dem Code in der VL irgendwo vorkamen