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

- [s01] Distributed Processing / Storage
- [s01] Spark Programming: Spark Architecture and Key Concepts
- [s01] Spark Fundamentals: Programming with RDDs; Example
- [s02] Structured Spark APIs: DataFrames
- [s02] Machine Learning (ML) with Spark
- …
- [s12/13] Spark Streaming (only basics)
- [s15] GraphX/GraphFrames (not relevant)

Recommender Systems (RS) /1

- [s03] Recommender Systems: Content-b & CF
  - Relevant only: formal model, utility matrix
- [s03] A. Content-based RS
  - General idea, item profiles (without TF-IDF)
  - Prediction and recommendation (without pros/cons)
- [s03] B. Collaborative Filtering
  - Predictions using similar users (user-user)
  - Item-Item Collaborative Filtering
  - Without: pros/cons
- [s03] Remarks & Practical Tips

Recommender Systems

- [s04] Contrasting Recommendation Methods
  - Note: good to repeat/compare
- [s04] C. Latent Factor Models (intro)
- [s04] Finding the Latent Factors
  - Especially: concept of regularization (no formulas, no initialisation with SVD)
- [s04] Optimizing by Stochastic Gradient Descent
- [s04] Stochastic Gradient Descent for Latent Factors
- [s04] Extra - The Netflix Prize

Link Analysis

- [s05] Link Analysis (introduction)
- [s05] PageRank: the “Flow” Formulation
- [s05] PageRank: The Google Formulation
- [s05] PageRank: Why Power Iteration works?
- [s06] PageRank: How do we actually compute the PageRank?
- [s06] Topic-Specific PageRank
- [s06] TrustRank

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

Association Rule Discovery

- [S09] The Market-Basket Model
- [S09] Outline and introduction:
  - Frequent Itemsets, Confidence and Interest, Mining Association Rules, Example
- [S09] Finding Frequent Itemsets
  - Understand the computation model and bottleneck(s)
- [S09] A-Priori Algorithm
- [S10] PCY Algorithm (simple)
- [S10] PCY Algorithm: Multistage/Multihash

Online Advertising (Only if included in the lecture)

- [s11] Online Bipartite Matching
- [s11] Web Advertising
  - Performance-based Advertising, Adwords Problem
  - Greedy algorithm and a bad scenario for it
- [s11] BALANCE Algorithm
  - Algorithm “rule” and example
  - BALANCE: Analyzing BALANCE, general result
  - Generalized BALANCE
  - BALANCE: Worst case and analysis - extra slides

Mining Data Streams 1

- [s12] Mining Data Streams – Introduction
- [s12] Filtering Data Streams
  - Problem statement and why a hash table is bad
  - First-Cut solution
  - Bloom Filter (important)
- [s12] Sampling a fixed proportion
- [s12] Sampling a fixed-size sample
- [s12/s13] Spark Streaming (only basics)

Mining Data Streams 2

- [s13] Counting Distinct Elements in a Stream
  - Flajolet-Martin Approach – the algorithm
  - Why It Works: Intuition
  - Why It Works: More formally
  - Why It Doesn't Work
- [s13] Computing Moments
  - Definition and exact computation of moments
  - Example of the surprise number
  - Alon-Matias-Szegedy (AMS) – the algorithm
  - Higher-Order Moments (k=3)

Other Topics

- There will be further topics in 2023
  - Vector Symbolic Architectures
  - SVMs / ML

Example of a Final's Problem

Problem 2. (14 points)
Clustering.

a. (5 points) Describe as a pseudocode the k-means clustering algorithm. Assume that your input data is a set of n vectors, each representing a point in R^d, and k is part of the input. Propose some simple convergence criterion of your choice. No formulas are required but can be used.

b. (2 points) We have introduced in the lecture the quotient BCV/WCV (BCV: *between-cluster variation*, WCV: *within-cluster variation*). Describe how this quotient is computed (formulas are recommended but correct text will be also accepted). Explain briefly how BCV/WCV can be used as a convergence criterion for the k-means algorithm and why it makes sense.

c. (4 points) Explain which parts of the k-means algorithm have the largest computational costs. Design and describe a distributed version of such an algorithm (as text or pseudocode). You can assume a programming model of your choice (e.g. Spark, MapReduce, or generic message passing). Especially explain how the input data is partitioned on computing nodes (in Spark: what does the input RDD look like), which operations are performed in parallel, and which data needs to be synchronized across all participating nodes at the end of each iteration.

d. (3 points) In agglomerative hierarchical clustering we merge in each step a pair of closest clusters. There are several metrics to express the cluster "distance", e.g.:

   1. centroid distance
   2. intercluster distance - single linkage (min)
   3. intercluster distance - complete linkage (max)
   4. "cohesion" of clusters, e.g. diameter of the merged clusters.

   Select two such cluster distance metrics and give for each them a brief definition / explanation. Furthermore, compute and state for each of these two metrics the order of cluster merging for the following seven points in R^1: 2; 5; 9; 15; 16; 18; 25. For presenting the merging order you might use text, or a table, or a dendrogram (not required).

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
- Kommen Programmieraufgaben dran?
  - Ja, es wird Programmieraufgaben (Spark) geben
  - Es kommt auf die Konzepte an, auch Pseudocode ist OK!

Fragen und Antworten /2

- 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
- Darf ich mein Haustier („pet animal“) mit in die Klausur mitnehmen?
  - Ja, falls es nicht beißt und kein Pferd ist
  - Evtl. Rückstände bitte anschließend entsorgen