Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Big Data Systeme & Recommendations

Timo Lange

Grundseminarvortrag SoSe 2017

HAW Hamburg
Fakultät Technik und Informatik
Studiendepartment Informatik
Studiengang: Master Informatik

Gliederung

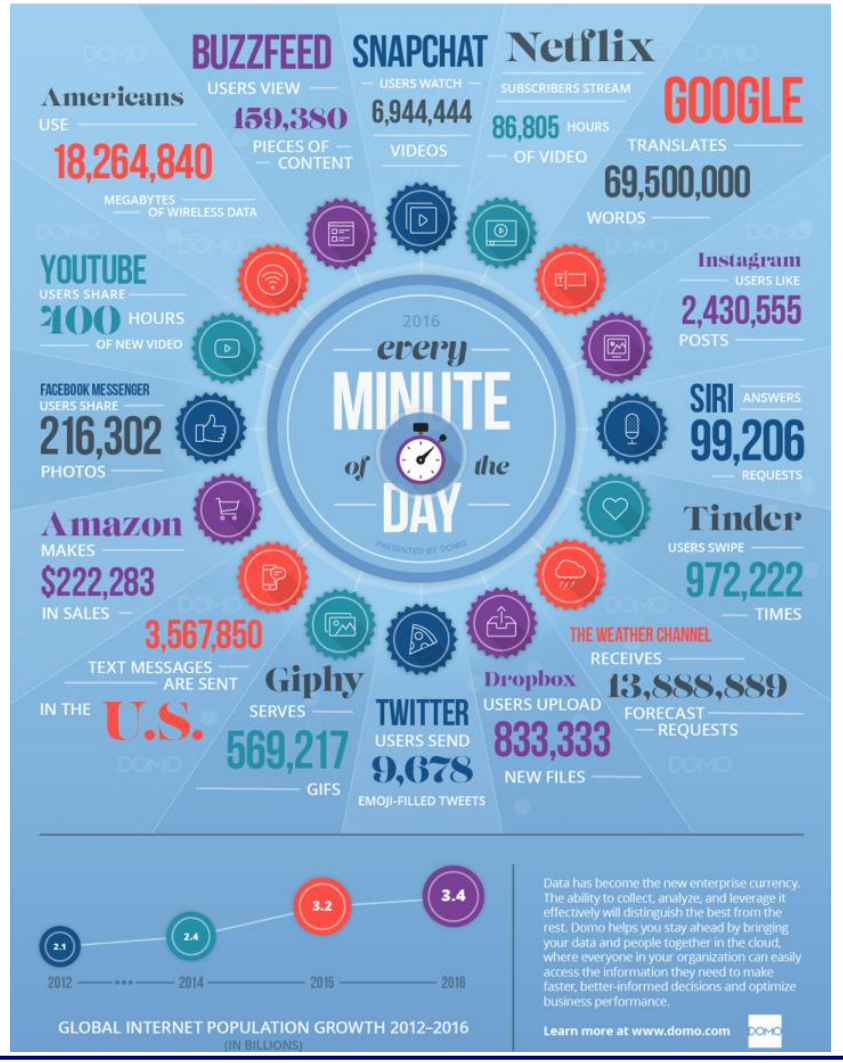
- Motivation
- Big Data
- Big Data Systeme
- Recommendation
- Recommendation - Verfahren
- Ziele
- Konferenzen

Motivation

- Bachelorarbeit
  - Big Data
  - Recommendation als Anwendungsfall

Big Data

- Keine einheitliche Definition
- 3 V‘s
  - Volume
  - Velocity
  - Variety
- Datenmenge im Big Data Bereich zu groß für einzelne Rechner
- Systeme notwendig, die im Cluster rechnen können



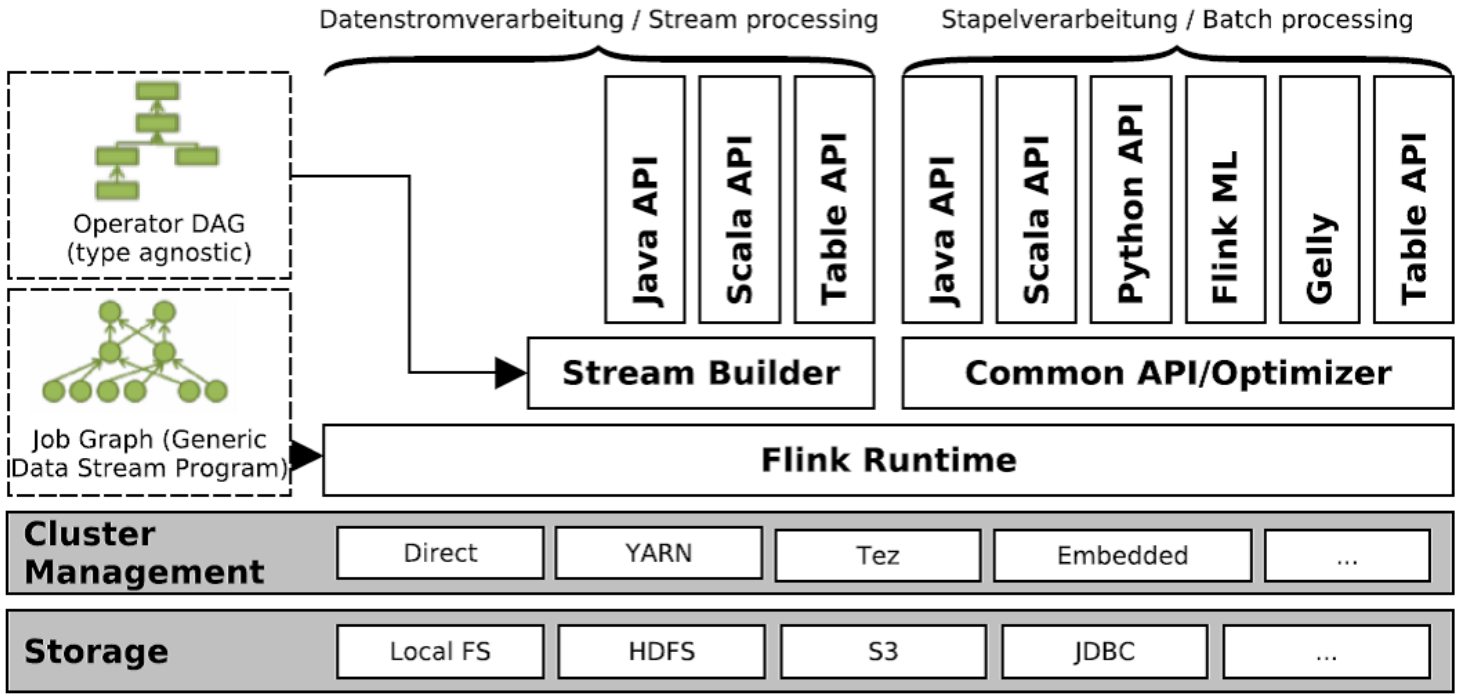
[Abb1]

Big Data Systeme

- Stream & Batch
  - Apache Flink
  - Apache Spark
  - Apache Apex
  - Apache Beam
- Batch
  - Apache Hadoop
  - Apache Tez
- Stream
  - Apache Storm
  - Apache Kafka
  - Apache Samza

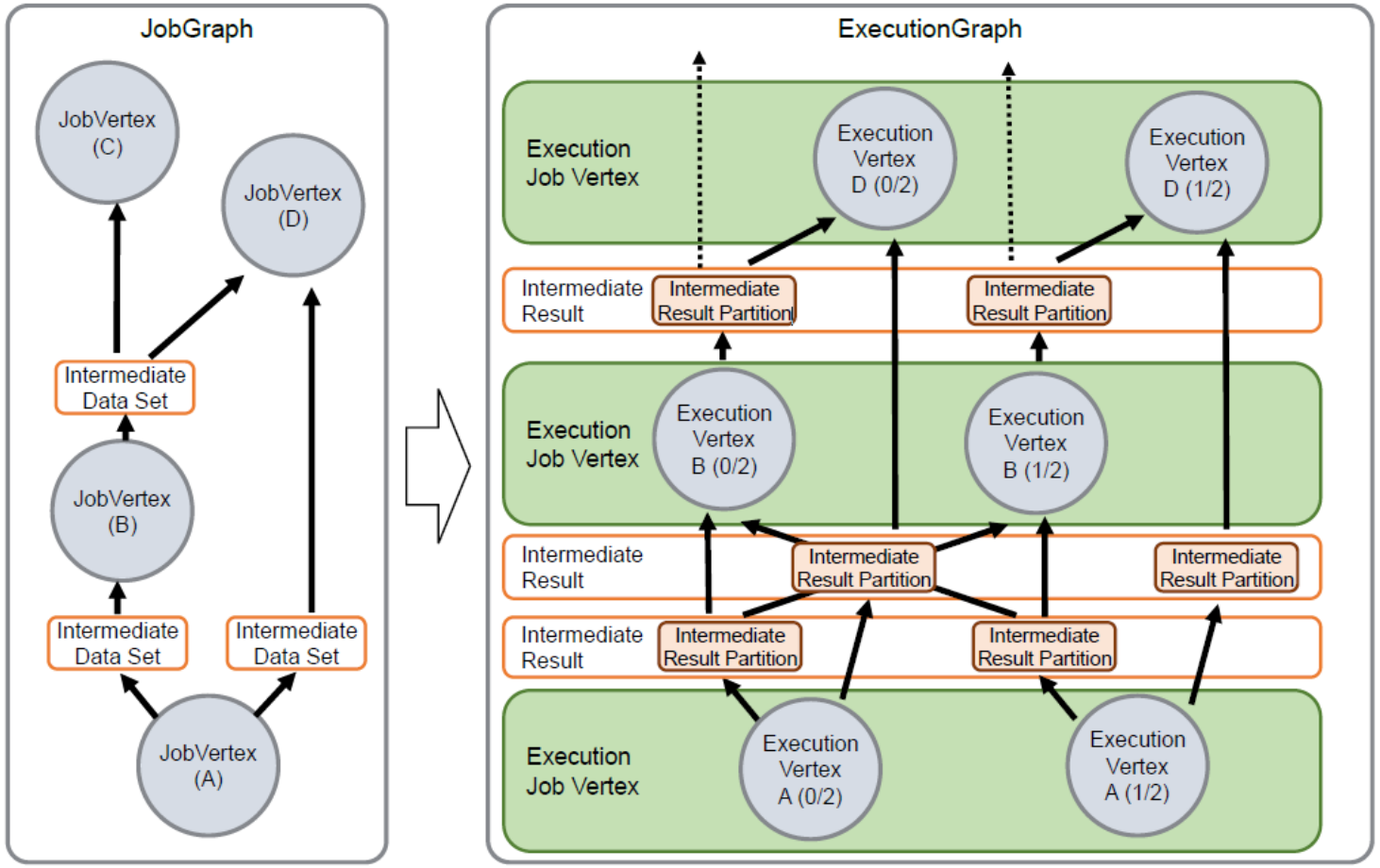
- Flink
  - Hauptabstraktion
    - DataSets
    - DataStreams

Architektur und Komponentenübersicht der Apache Flink Plattform



[Abb2]

- API erstellt Ausführungsplan (Operator-DAG)
- Aus Operator-DAG wird der JobGraph und ExecutionGraph generiert

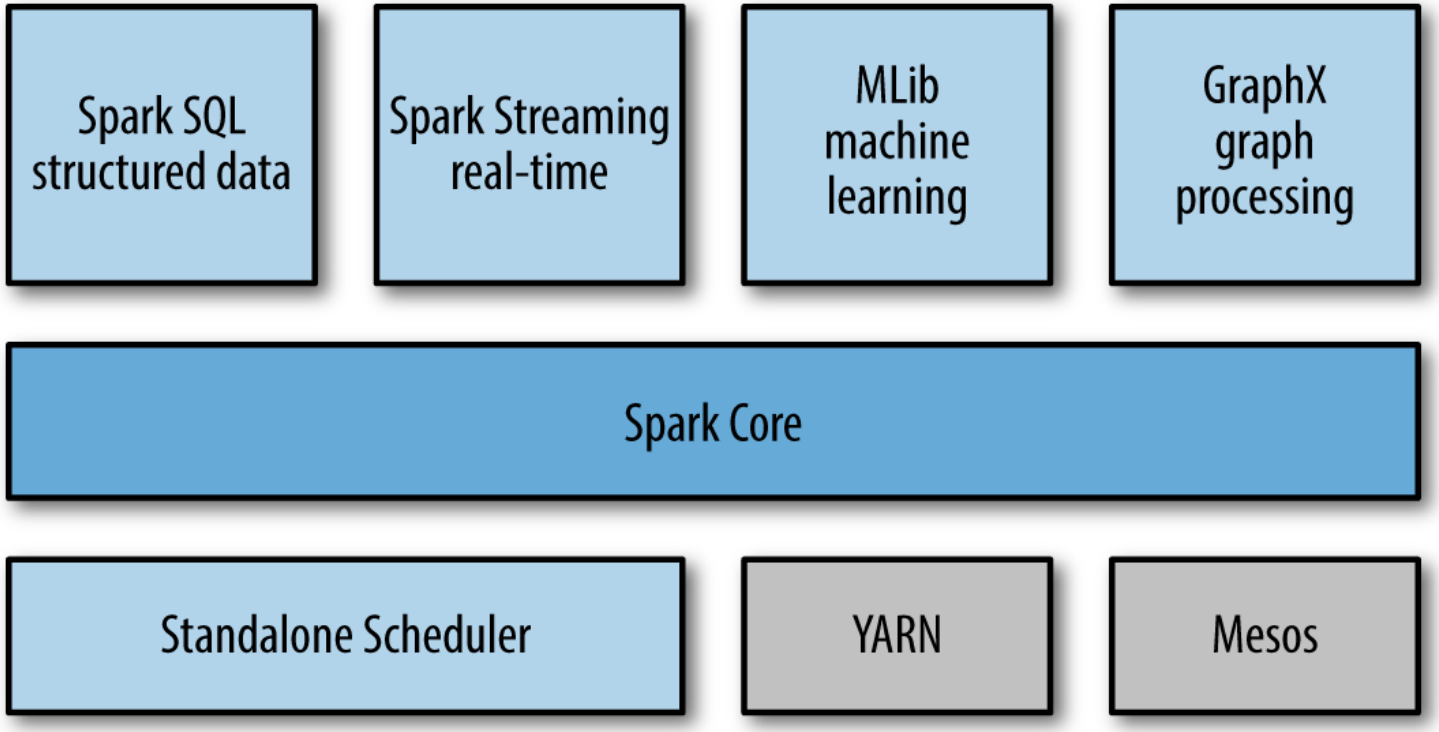


[Abb3]

Big Data Systeme

- Spark
  - Hauptabstraktion
    - RDD
    - DStream

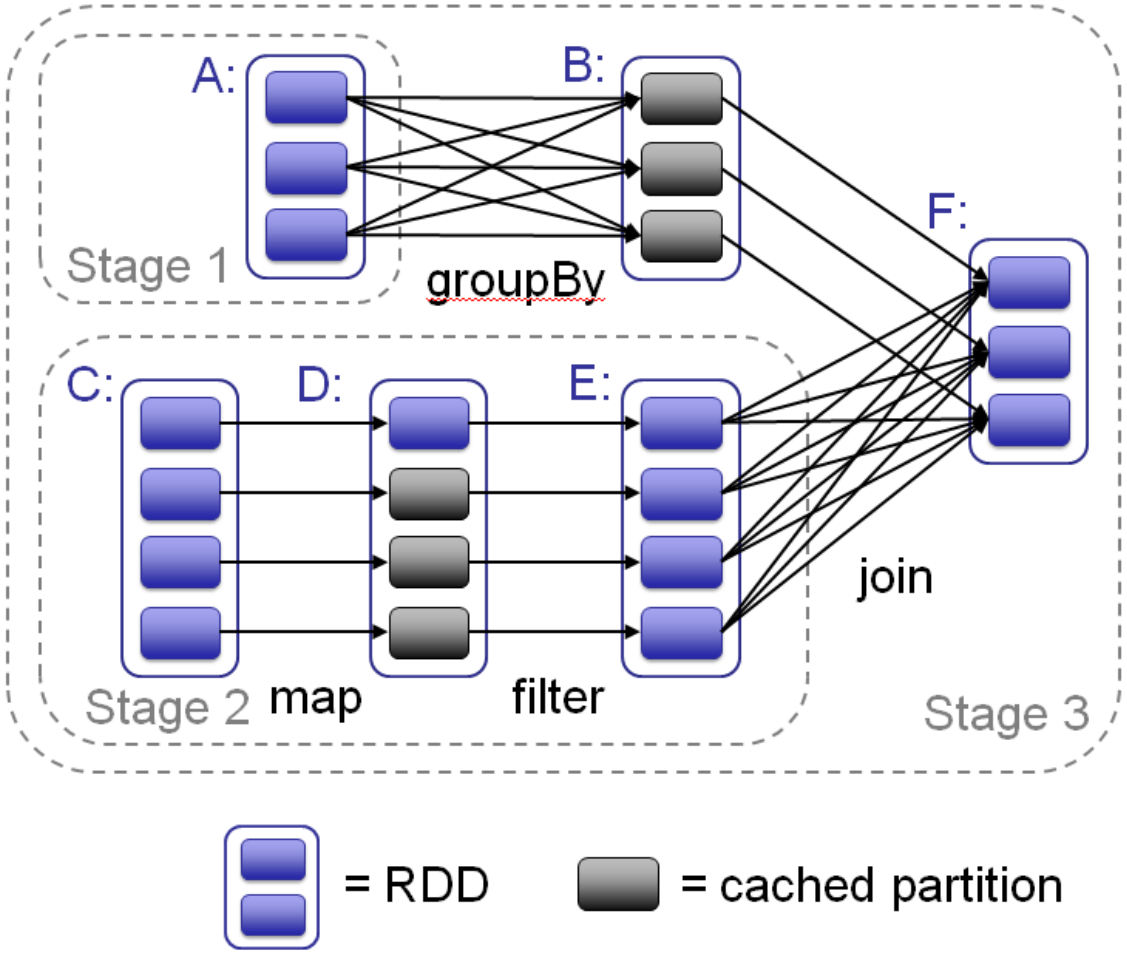
Der Komponenten-Stack von Apache Spark



[Abb4]

Big Data Systeme

- RDD Operationen aufgeteilt auf Stages



[Abb5]

- Ähnliche Konzepte in Flink und Spark
  - Typen Serialisierung
  - Speicherverwaltung
  - Off-heap Memory
  - Integrierte Bibliotheken
    - Maschinelles Lernen
    - Graph Verarbeitung
    - Relationale Datenverarbeitung

- Besonderheiten von Flink und Spark
  - Flink
    - Streaming Windows
    - Iterationen (Bulk & Delta)
    - FlinkCEP - Complex Event Processing
  - Spark
    - Persistenz
    - RDD in DStream Nutzbar

Recommendation

- Was sind Recommender Systeme
  - „Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user “ [2]
  - „The construction of systems that support users in their (online) decision making is the main goal of the field of recommender systems. In particular, the goal of recommender systems is to provide easily accessible, high-quality recommendations for a large user community “ [3]

Recommendation

- Wozu Recommender Systeme
  - Nutzer beeinflussen (z.B. etwas zu kaufen)
  - Information Overload
    - "People read around 10 MB worth of material a day, hear 400 MB a day, and see 1 MB of information every second" - The Economist, November 2006
    - U.S. media consumption [4]
      - In 2008, 33 GB per person per day
      - In 2012, 63 GB per person per day
      - In 2015, media consumption will raise to 74 GB per person a day

[Abb6]

The Age of Search has come to an end

- ... long live the Age of Recommendation! [1]
- Chris Anderson in “The Long Tail”
  - “We are leaving the age of information and entering the age of recommendation” [1]
- CNN Money, “The race to create a 'smart' Google”:
  - “The Web, they say, is leaving the era of search and entering one of discovery. What's the difference? Search is what you do when you're looking for something. Discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you.” [1]

Der Wert von Recommendations [1]

- Netflix: 2/3 der Filme wurden aufgrund von Recommendations gesehen.
- Google News: Recommendations generieren 38% mehr clickthrough.
- Amazon: 35% sales von Recommendations.
- Choicestream: 28% der Leute würden mehr Musik kaufen, wenn sie finden würden was sie mögen.

Recommendation Verfahren

- Collaborative Filtering
- Content-Based
- Knowledge-Based
- Hybrid Systems
- Und Weitere…

Collaborative Filtering

- Methoden
  - Neighborhood Methoden
  - Latent Factor Models
- Optimierungsverfahren für Latent Factor Models
  - Matrix Factorization
    - Alternating Least Squares (ALS)
    - Stochastic Gradient Descent (SGD)

Collaborative Filtering

Neighborhood Methode

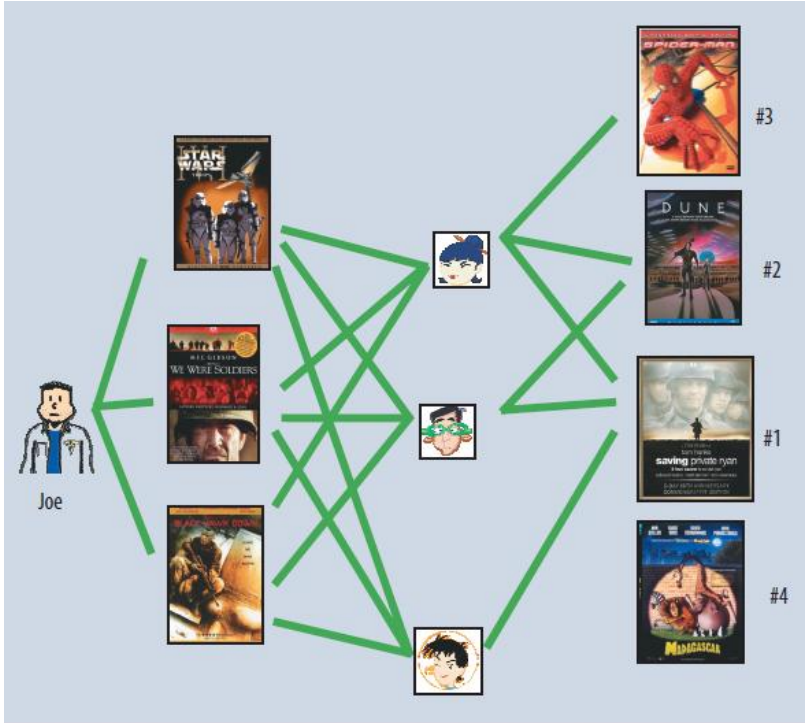


Figure 1. The user-oriented neighborhood method. Joe likes the three movies on the left. To make a prediction for him, the system finds similar users who also liked those movies, and then determines which other movies they liked. In this case, all three liked *Saving Private Ryan*, so that is the first recommendation. Two of them liked *Dune*, so that is next, and so on.

[5]

Latent Factor Methode

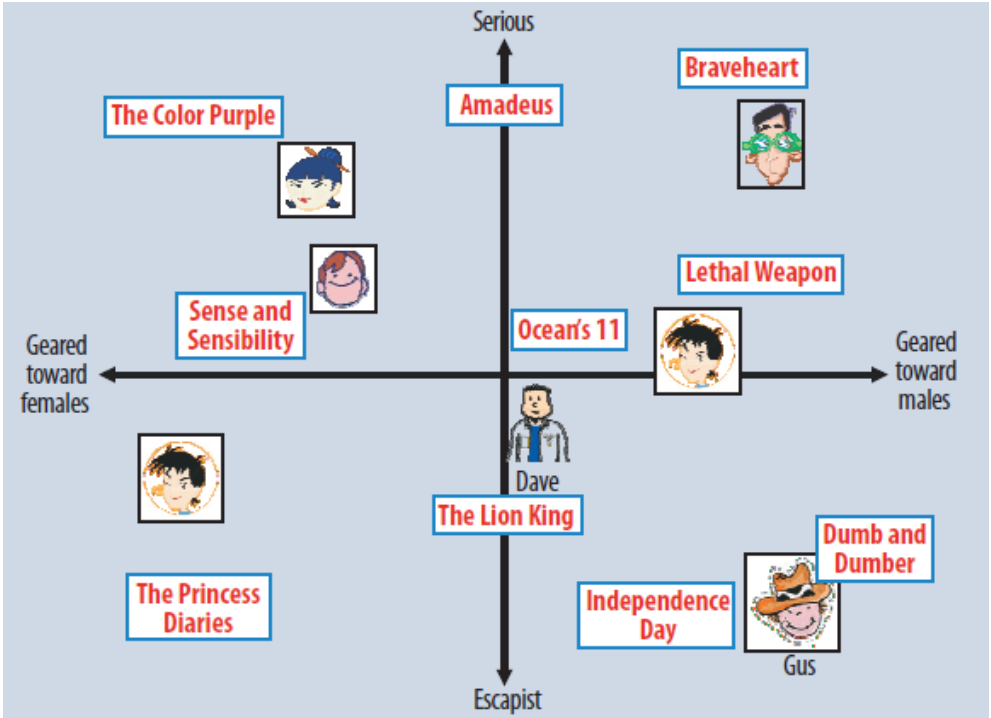


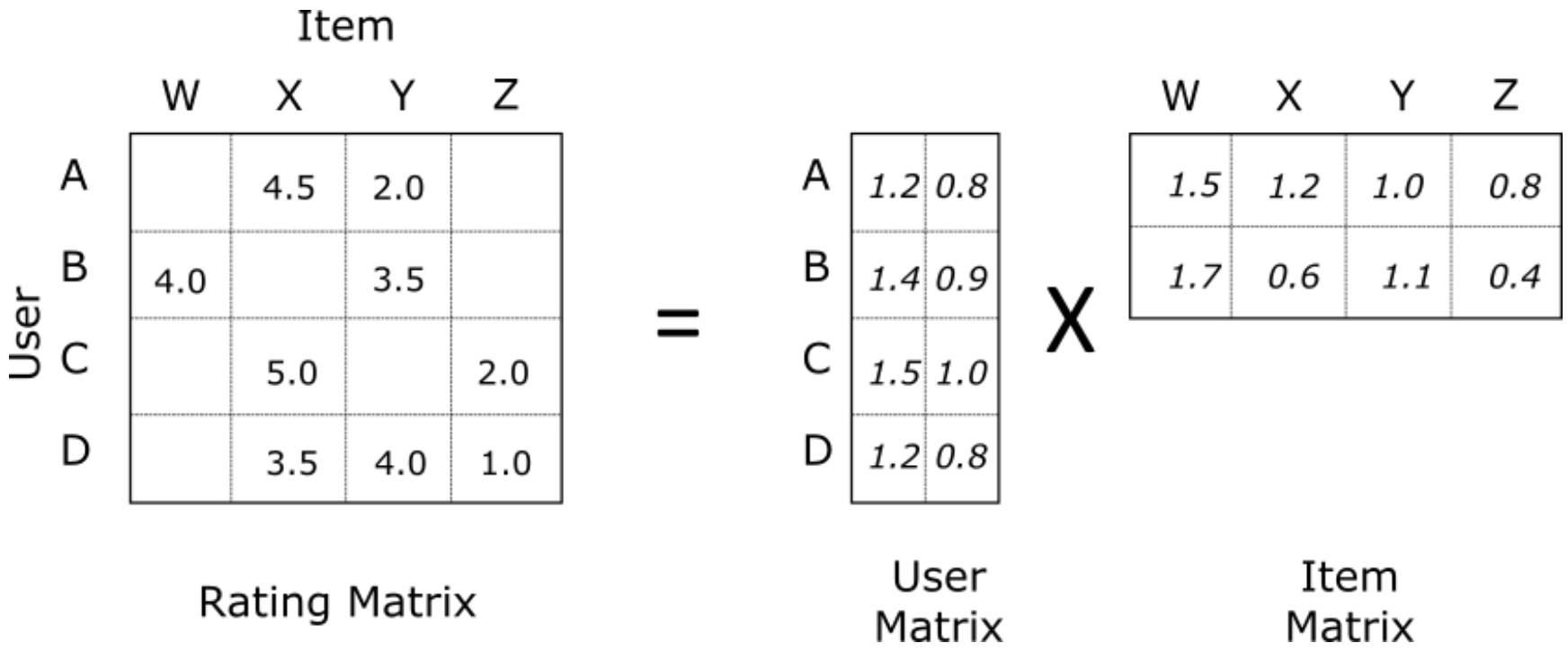
Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

[5]

Matrix Factorization

- Y. Koren, R. Bell, und C. Volinsky
  - „Matrix Factorization Techniques for Recommender Systems“, Computer, Bd. 42, Nr. 8, S. 30–37, Aug. 2009.
- Y. Zhou, D. Wilkinson, R. Schreiber, und R. Pan
  - „Large-scale parallel collaborative filtering for the netflix prize“, in International Conference on Algorithmic Applications in Management, 2008, S. 337–348.
- Netflix Preis
  - In 2006
  - 1$ Million Preisgeld für 10% Verbesserung (RMSE)
  - Forschung hat stark zugenommen

Alternating Least Squares (ALS) Matrix Factorization



[Abb7]

Alternating Least Squares (ALS) Matrix Factorization

- Mean Squared Error (MSE) berechnen
  - Lambda Regulierung um Overfitting zu vermeiden
  - K beinhaltet nur vorhandene Ratings der Ratingmatrix

$$\min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

[5]

ALS Implementation

- ALS Vorteile
  - Gute Vorhersagegenauigkeit
  - Gut parallelisierbar
- Implementierung mit Flink & Spark
  - Kommunikationsaufwand reduzieren
    - Blocked ALS
  - Zurzeit keine weiteren speziell auf Recommendation ausgelegte Verfahren implementiert

ALS Probleme

- Cold Start Problem
- Online Update
- Herausforderung
  - Implizites Feedback anstatt explizitem Feedback

Paper Matrix Factorization Online Update

- X. He, H. Zhang, M.-Y. Kan, und T.-S. Chua
  - „Fast Matrix Factorization for Online Recommendation with Implicit Feedback“, in Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, 2016, S. 549–558.

Paper Matrix Factorization Online Update

- eALS(element-wise Alternating Least Squares)
  - Latent Factor Vector Update pro Element
  - Online Update möglich
    - Änderung nur lokal für jeweils User und Item
    - Annahme: sollte das Modell aus globaler Sicht nicht zu sehr beeinflussen
  - Gewichtung des Feedbacks möglich (z.B. populäre Items relevanter)
  - Nutzt Cache Matrix für die Gewichtung der fehlenden Daten
    - Jeweils für User und Item

Weitere Probleme

- eALS Probleme bei Implementierung?
  - In Paper verwendete Caches für große Datenmengen effizient verteilt verwendbar?
  - Langsam verschlechternde Vorhersagegenauigkeit bei Online Update in realem Szenario?
  - Messungen in Paper
    - Nicht verteilt
    - Single Thread
- Recommendation allgemein
  - Filter Bubble

Konferenzen

- Big Data
  - IEEE BigData Congress 2017 (6th IEEE International Congress on Big Data, June 25 - June 30, 2017, Honolulu, Hawaii, USA, http://www.ieeebigdata.org/2017/index.html)
  - Strata Data Conference (May 23–25, 2017, London, United Kingdom, https://conferences.oreilly.com/strata)
  - ICDM 2017 (The IEEE International Conference on Data Mining, New Orleans, USA, November 18 - November 21, 2017, http://icdm2017.bigke.org/)
- Recommendation
  - ACM RecSys 2017:11th ACM Conference on Recommender Systems (Como, Italy, 27th-31st August 2017, https://recsys.acm.org/recsys17/)
    - RecSys Challenge 2017 (https://recsys.acm.org/recsys17/challenge/)
  - ACM UMAP 2017: 25th conference on User Modeling, Adaptation and Personalization (Bratislava, Slovakia, 9-12 July 2017, http://www.um.org/umap2017/)
  - ACM KDD 2017: Knowledge Discovery and Data Mining (Halifax, Nova Scotia – Canada August 13 - 17, 2017, http://www.kdd.org/kdd2017/)

Ziele/weiteres Vorgehen

- Weitere Entwicklung der Big Data Systeme verfolgen
- Recommendation Verfahren
  - Online Training Methoden verfolgen
  - Algorithmen vergleichen
- Recommendation Algorithmen mit vorgestellten Systemen implementieren
- Recommendation-Engine mit mehreren Verfahren entwickeln

Quellen Literatur

- [1] Xavier Amatriain, „Introduction to Recommender Systems: A 4-hour lecture“, 2014. Abruf 21.05.2017.
- [2]F. Ricci, L. Rokach, B. Shapira, und P. B. Kantor, Recommender Systems Handbook. Boston, MA: Springer US, 2011.
- [3]D. Jannach, Markus Zanker, Alexander Felfernig, und Gerhard Friedrich, Recommender systems: an introduction. New York: Cambridge University Press, 2011.
- [4]James E. Short, „How Much Media? 2013 Report on American Consumers“, Institute for Communications Technology at Management Marshall School of Business, University of Southern California, Aug. 2013.
- [5]Y. Koren, R. Bell, und C. Volinsky, „Matrix Factorization Techniques for Recommender Systems“, Computer, Bd. 42, Nr. 8, S. 30–37, Aug. 2009.
- [6]Y. Zhou, D. Wilkinson, R. Schreiber, und R. Pan, „Large-scale parallel collaborative filtering for the netflix prize“, in International Conference on Algorithmic Applications in Management, 2008, S. 337–348.
- [7]X. He, H. Zhang, M.-Y. Kan, und T.-S. Chua, „Fast Matrix Factorization for Online Recommendation with Implicit Feedback“, in Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, 2016, S. 549–558.

Quellen Abbildungen

- [Abb1]Josh James, „Data Never Sleeps 4.0 | Domo Blog“, 28-Juni-2016. [Online]. Verfügbar unter: https://www.domo.com/blog/data-never-sleeps-4-0/. [Zugegriffen: 22-Mai-2017].
- [Abb2]J. Traub, T. Rabl, F. Hueske, T. Rohrmann, und V. Markl, „Die Apache Flink Plattform zur parallelen Analyse von Datenströmen und Stapeldaten“, 2015.
- [Abb3]The Apache Software Foundation, „Apache Flink 1.4-SNAPSHOT Documentation: Jobs and Scheduling“, 19-Mai-2017. [Online]. Verfügbar unter: https://ci.apache.org/projects/flink/flink-docs-master/internals/job_scheduling.html. [Zugegriffen: 22-Mai-2017].
- [Abb4]H. Karau, A. Konwinski, P. Wendell, und M. Zaharia, Hrsg., Learning Spark: [lightning-fast data analysis], 1. ed. Beijing: O’Reilly, 2015.
- [Abb5]Matei Zaharia, „Parallel Programming With Spark“, gehalten auf der Strata Conference, Santa Clara, CA, Feb-2013.
- [Abb6] C. E. Pfister, „Information Overload: What is it doing to your employees?“, Renaissance Executive Forums - Peer Advisory Roundtables. [Online]. Verfügbar unter: https://www.executiveforums.com/single-post/2017/03/03/Information-Overload-What-is-it-doing-to-your-employees. [Zugegriffen: 22-Mai-2017].
- [Abb7]Till Rohrmann, „Computing Recommendations at Extreme Scale with Apache Flink™“, data Artisans, 18-März-2015. [Online]. Verfügbar unter: https://data-artisans.com/blog/computing-recommendations-at-extreme-scale-with-apache-flink. [Zugegriffen: 21-Mai-2017].