New Methods for the Automated Analysis of Massive, Parallel, and Distributed Data Streams

Volker Markl

Professor of Computer Science, TU Berlin
Chief Scientist, DFKI Berlin
Director, Berlin Institute for the Foundations of Learning and Data (BIFOLD)
AGENDA

1. Data as a Factor of Production
2. Selected Research Contributions
3. Current Research and Vision
Big Data and Machine Learning are the key drivers of innovation in AI and Data Science.
Data Management jointly with ML are disruptive in the Sciences, Humanities, and Industry.

4th Paradigm

Data as Factor of Production

Data Science (DS)

Data Management (DM)

Machine Learning (ML)

Artificial Intelligence (AI)
**Remote Sensing**

Analysis of massive satellite data archives (Sentinel-2)

- Labeled training data archive for AI models
- 3 TB / 5 days multimodal data
- Fast classification and categorization

![Satellite Image](https://scihub.copernicus.eu)

- Enables:
  - analysis of trends (e.g., deforestation)
  - predictions about regions (e.g., draught)
- Added to popular “big data” catalogs:
  - (e.g., Google Earth, Radiant MLHub, TensorFlow)

References:

3. [http://bigearth.net](http://bigearth.net)  

---

**Industrie/Industry 4.0**

Exploratory realtime analysis of sensor data streams

- Prediction of paper quality during production
- 97 heterogeneous sensors
- complex model building + information extraction and integration
- low latency

![Diagram](https://scihub.copernicus.eu)

- Enables:
  - faster reaction to paper quality issues
  - cost reduction
  - Data science on real time data streams

References:


(TU Graz, Siemens)
Data science process is complex and time-consuming

Goal of data management research: scaling the process and its operationalization with respect to human and technical latency

Draws on results from: computer architecture, statistics, machine learning, distributed systems, compilers, programming languages, etc.
Data science process is complex and time-consuming

Goal of data management research: scaling the process and its operationalization with respect to human and technical latency

Draws on results from: computer architecture, statistics, machine learning, distributed systems, compilers, programming languages, etc.
"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers
"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.
"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers
Example: "3-Means-Clustering"

Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.
"3-Means-Clustering" is a simple data analysis method that divides a dataset into \( k \) groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.
"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.
"3-Means-Clustering" is a simple data analysis method that divides a dataset into $k$ groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- **Assign each point to the closest cluster**
- Move centers
"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Choose 3 random cluster centers

Iterate until convergence:

Compute distance of each point to each center

Assign each point to the closest cluster

Move centers
"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.
Example: "3-Means-Clustering"

"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster

Move centers
"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers
"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers
"3-Means-Clustering" is a simple data analysis method that divides a dataset into \( k \) groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.
Assign each point to the closest cluster
Compute distance of each point to each center
Assign each point to the closest cluster
Move centers

"3-Means-Clustering" is a simple data analysis method that divides a dataset into k groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers
“What” not “How”
Example “k-Means-Clustering”

Hand-optimized code
(data-, workload- and system-dependent)

486 lines of code
3 days development
4 hours runtime
“What”, not “How“:
Example “k-Means-Clustering“

65 lines of code
1 hour development
1 hour runtime

(declarative specification)

Declarative data flow program with
automatic optimization, parallelization,
and hardware adaption

486 lines of code
3 days development
4 hours runtime

(imperative specification)

Hand-optimized code
(data-, workload- and system-
dependent)
AGENDA

1. Data as a Factor of Production
2. Selected Research Contributions: Reduction of Human and Technical Latency
   ① Automatic Optimization of Complex Analysis Algorithms for Big Data
   ② Analysis of Heterogeneous and Distributed Data Streams in the Internet of Things
3. Current Research and Vision
case class Path (from: Long, to: Long)
val tc = edges.iterate(10) {
  paths: DataSet[Path] =>
  val next = paths
    .join(edges)
    .where("to")
    .equalTo("from") {
    (path, edge) =>
      Path(path.from, edge.to)
    }
    .union(paths)
    .distinct()
  next
}

Auto-matic optimization,
parallelization and hardware adaption

Declarative data-analysis program

memory management
out-of-core algorithms
data stream processing
state management

monitoring of operation
fault-tolerance
resource management
coordinator

human latency
data flow graph
specification time
runtime

technical latency

Why Optimization?

Do you want to hand-tune that?

F. Hueske, M. Peters, A. Krettek, M. Ringwald, K. Tzoumas, V. Markl, J.C. Freytag: Peeking into the optimization of data flow programs with MapReduce-style UDFs. ICDE 2013: 1292-1295
Impact of Automatic Optimization

program simplification

no hand-tuning necessary

declarative data analysis program

optimization decisions e.g.:
- execution order
- parallelization strategy
- “in-core”/“out-of-core”-algorithms
- physical implementation
- materialization strategy

execution of graph A

analysis of test data on Notebook

execution of graph B

analysis of big data on a cluster

execution of graph C

later analysis, after data or environment have changed

caching, parallelization, algorithm selection

reduces human latency

reduces technical latency

27 © Volker Markl
Effect of Optimization:
Lower latency and higher throughput in particular for streaming applications

2x to 4x lower latency
50x higher throughput


VLDB Best Paper Award 2014
compression of data streams for visualization

ACM SIGMOD Research Highlight Award 2015
implicit parallelism through deep language embedding

EDBT Best Paper Award 2019
latency reduction by result sharing
Apache Flink

Flink Community

- 29,500+ Meetup Members Worldwide
- 870+ Open Source Contributors/Developers
- 49 Meetup Groups Worldwide

Flink Contributors

- 18 Countries that Regularly Hold Meetups
- 49+ Companies using Apache Flink
- Startup data Artisans (now Ververica)

Last updated: May 2021

https://www.meetup.com/topics/apache-flink/
https://flink.apache.org/poweredby.html
https://github.com/apache/flink
Some Highly Engaged Users

Largest job has > 20 operators, runs on > 5000 vCores in 1000-node cluster, processes millions of events per second

Complex jobs of > 30 operators running 24/7, processing 30 billion events daily, maintaining state of 100s of GB with exactly-once guarantees

30 Flink applications in production for more than one year; 10 billion events (2TB) processed daily

Courtesy of Kostas Tzoumas
Apache Flink Users

https://cwiki.apache.org/confluence/display/FLINK/Powered+by+Flink
AGENDA

1. Data as a Factor of Production
2. Selected Research Contributions: Reduction of Human and Technical Latency:
   ① Automatic Optimization of Complex Analysis Algorithms for Big Data
   ② Analysis of Heterogeneous and Distributed Data Streams in the Internet of Things
3. Current Research and Vision
NebulaStream: Data Processing for the IoT

Reliable and efficient data stream management in a massively distributed, heterogeneous environment

“In-network”-pre-processing and “on-demand”-data processing

Hardware-targeted code generation

Intermediate representation to optimize heterogeneous data analysis programs

Decentralized error detection and intervention

Dynamic optimization and lightweight adaptation of the distributed processing topology

ACM SIGMOD Best Paper Award 2020

Latency reduction by heterogeneous hardware

Millions of data sources Distribution

Heterogeneous hardware and data formats
Data as a Factor of Production

Selected Research Contributions: Reduction of Human and Technical Latency

Current Research and Vision:
Methods and Technologies for a single data infrastructure for AI-applications
Current Situation: Considerable backlog in AI innovations due to ...

Fragmented AI project landscape

Many little platforms
- Redundant efforts
- Competition for the best minds
- Fragmented resources and expertise

• Focus often on data exchange
• Interesting concepts exist – lack of implementation
• Multitude of efforts – lack of critical mass
Agora – **One** Data Management Infrastructure for AI-Applications

cross-sector open and protected areas collaborations billing models
AGORA: One Data Management Infrastructure for AI-Applications
-- current research activities --

Multi-Platform Data Processing

Mediator-based systems:
• expensive data transfer
• additional system
• error-prone heuristics

Goal: Mediatorless processing
• orchestration of systems

Compliant Distributed Processing

Data-flow constraints:
• transfer not permitted
• specification and enforcement of constraints

Goal: Automated derivation of compliant executions

Trustworthy Data Processing

Open environment:
• non-trusted servers
• inefficient processing

Goal: Efficient trustworthy processing through modern hardware.

Debugging of Data Analysis Programs

Online-error identification for big data streams:
• distributed processing
• continuous execution

Goal: Error correction by analyzing the processing history
Healthcare

Management and analysis of massive health data

- Efficient integrated analysis of multimodal medical datasets (ECG, MRI, -”omics”-data)
- Integration of image and time-series data
- Processing of sensitive data

- “Life-Science-API” for Flink
- Demonstrator @ Forum Digitale Technologien
- Showcase at CeBIT


Digital Humanities

Efficient navigation through historical books

- System for managing texts, illustrations and tables in historical books
- Vast number of data sources
- Text segmentation and semantic analysis
- New Paradigm for the historical sciences (”longue-durée “-studies with resolution of microstudies)
- Evolution of the role of archives and libraries

(Matteo Valeriani, MPG)
Environmental Data Analysis And “Citizen Science”

Deciphering air pollution data through data integration, explanation of outliers and visualization

- Analysis of the effects of traffic, driving restrictions, events, and weather on air quality
- Heterogeneous data formats
- Erroneous sensors
- Number of data sources

External Data Sources
- Weather Data
- Street Data
- Event Data
- Air Traffic Data

Data Preprocessing
- Clustering and Binning
- Data Cleaning
- Data Integration

Analysis
- Outlier Explanation
- Stream MAD Outlier Detection
- Time Series Analysis

Selected results:

- Sensors near runways show peaks during vacation times.
- Events significantly impact air quality even without any traffic.
- Sensors near factories, stations, or highways show peaks depending on wind directions.

- Particulate Matter Matters - The Data Science Challenge @ BTW’19. Meyer et al., Datenbank-Spektrum 2019
Conclusion

1. Data are a new **factor of production** for sciences, humanities, and industry.
   - Critical success factor is the reduction of **human latency** through intuitively usable systems.
   - Reduction of the **technical latency** enables real-time analysis.

2. Data management research is **interdisciplinary** within and outside computer science.

3. Data management research has created important **technological foundations, new systems, and novel applications**:
   - **Methods** for improving efficiency through automatic optimization, parallelization and hardware adaption.
   - **Systems** for efficient, distributed, compliant analysis of large datasets and streams.
   - **Applications** in the **Sciences and Humanities, Industry and Society** (“Citizen Science”).

4. **Current research challenges** in data management include:
   - Methods and systems for the management and analysis of **distributed, heterogeneous data streams** (“IoT”, “Industrie 4.0”).
   - Technologies for **data management infrastructures for open and protected, collaborative AI innovations** (“NFDI”, “GAIA-X”).

5. The scientific community in Berlin is tackling these challenges in a comprehensive **ecosystem**:
   - **Foundational research** (BIFOLD) and **applied research** (e.g., DFKI, Fraunhofer).
   - **Applications** in mathematics/sciences, healthcare/biotech, humanities, as well as industry and startups.
Data – a New Factor of Production

soil → prepare → fertilize → plant → harvest

data → extract → integrate → analyze → use