SCIENCE
PASSION
TECHNOLOGY

Introduction to Scientific Writing 01 Structure of Scientific Papers

Matthias Boehm

Graz University of Technology, Austria
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

Last update: Oct 23, 2021

ISDS

Announcements/Org

- **#1 Virtual Lectures**
  - https://tugraz.webex.com/meet/m.boehm
  - Optional attendance (independent of COVID)

- **#2 Course Registrations** (as of Oct 28)
  - Changes in WS20/21, now max constraints
  - **Introduction to Scientific Writing**

ISDS Group
Boehm
40/40

Agenda

- **Data Management Group**
- **Course Organization, Outline, and Projects**
- **Structure of Scientific Papers**
- **Paper Project Proposals**

Data Management Group

https://damslab.github.io/

About Me

- **09/2018 TU Graz**, Austria
  - BMK endowed chair for data management
  - **Data management for data science**
    (ML systems internals, end-to-end data science lifecycle)

https://github.com/apache/systemds

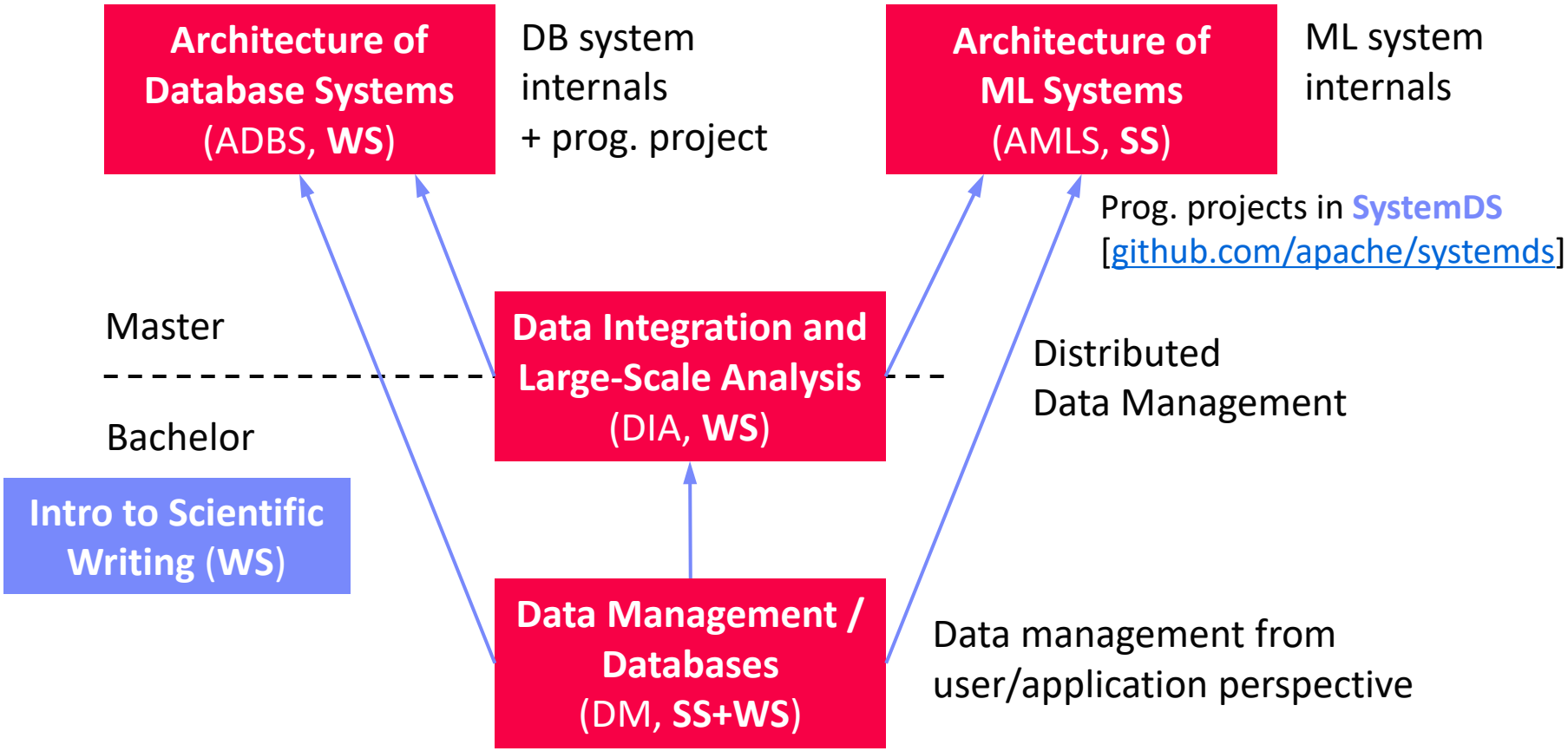
- **2012-2018 IBM Research – Almaden**, USA
  - Declarative large-scale machine learning
  - Optimizer and runtime of **Apache SystemML**

- **2011 PhD TU Dresden**, Germany
  - Cost-based optimization of integration flows
  - Systems support for time series forecasting
  - In-memory indexing and query processing

TECHNISCHE UNIVERSITÄT DRESDEN

DB group

Data Management Courses



Course Organization, Outline, Goals, and Projects

Course Logistics

- **Staff**
  - Lecturer: Univ.-Prof. Dr.-Ing. Matthias Boehm, ISDS
  - Assistant: M.Sc. Shafaq Siddiqi, ISDS

- **Language**
  - Lectures and slides: **English**
  - Communication and examination: **English**/**German**
  - Submitted paper and talk: **English**
  - **Informal language** (first name is fine), immediate feedback

- **Course Format**
  - SE 1, **2 ECTS** (0.5 ECTS lectures + 1.5 ECTS paper/talk), bachelor
  - 3 lectures, **optional attendance**
  - Mandatory paper and presentation

Course Logistics, cont.

- **Website**
  - https://mboehm7.github.io/teaching/ws2122_isw/index.htm
  - All course material (lecture slides) and dates
- **Video Recording Lectures (TUbe)? No**
  - Lectures and discussions via Webex
  - No recording in order to foster discussion, private presentations
- **Goals**
  - Understanding of / communication through scientific writing
  - Best practices for effective scientific reading, writing, and reproducibility
- **Grading**
  - **Overall: pass/fail** (no detailed 1-5 grades)
  - Includes submitted paper and final presentation
    (pass := adhere to given constraints + acceptable quality)

Outline Lectures

- **01 Structure of Scientific Papers** [Oct 28, 6.15pm, **optional**]
- **02 Scientific Reading and Writing** [Nov 04, 6pm, **optional**]
- **03 Experiments, Reproducibility, and Projects** [Nov 11, 6pm, **optional**]

...

- **04 Project Presentations** [Jan 13, 6pm, **mandatory**]

Paper Projects

Alternative: LV combined with bachelor thesis

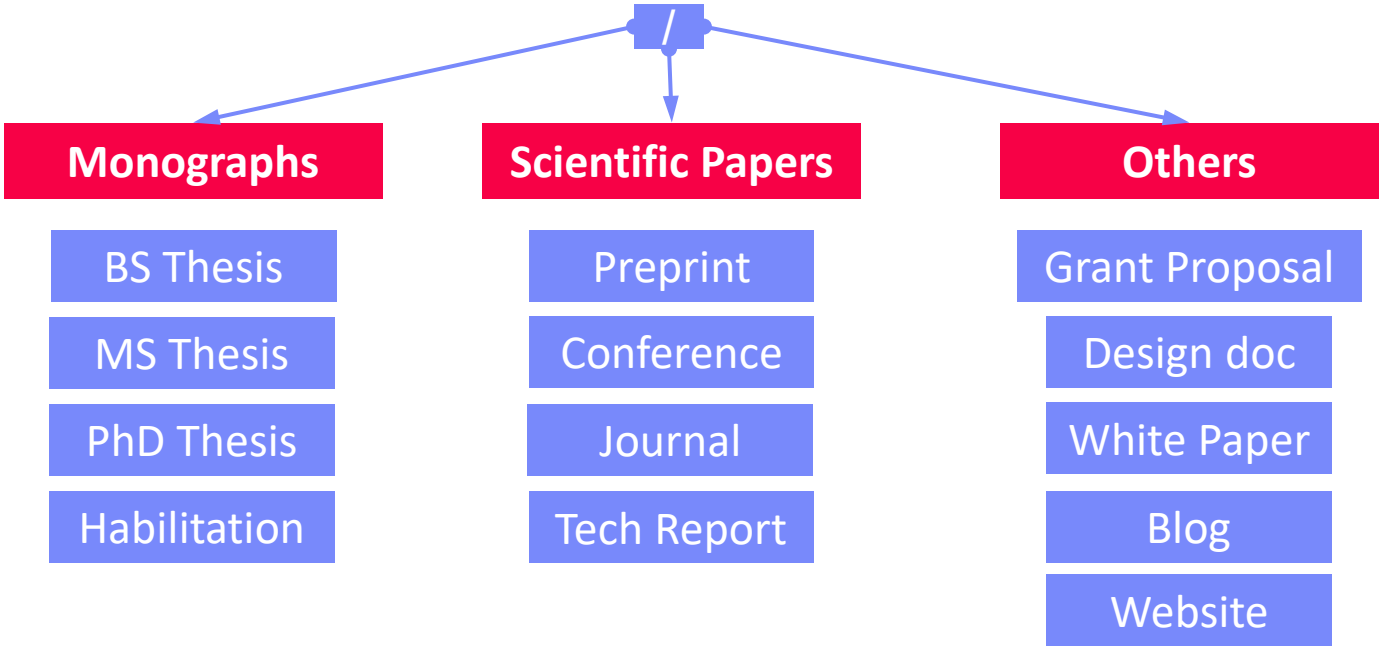
- **Team**
  - 1-4 person teams (w/ clearly separated responsibilities)
- **Project**
  - Pick from a given list of papers / groups of papers
  - **#1** Write short summary paper (#pages = 2 * team-size, written in LaTeX, ACM acmart template, document-class sigconf, PDF)
  - **#2** Prepare and present talk on paper summary (7min + 3min Q&A)
- **Timeline**
  - **Today:** List of projects proposals, feel free to bring your own
  - **Nov 11:** project selection via email to m.boehm@tugraz.at (11.59pm)
    subject: [Scientific Writing] Project Selection
  - **Dec 23:** paper submission via email to m.boehm@tugraz.at (11.59pm)
  - **Jan 13:** Final project presentation (all students)

Structure of Scientific Papers

In Computer Science (Data Management)

Overview Types of Scientific Writing

- **Classification of Scientific/Technical Documents**
  - Formal vs informal writing, cumulative?, single vs multi-author
  - Archival vs non-archival publications

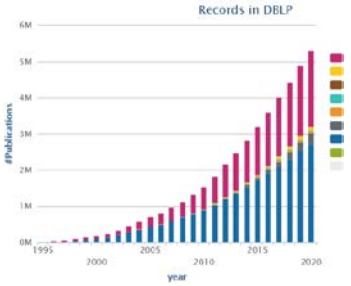


➔ **Scientific Writing Skills are crucial**

- Different types of docs share many similarities

Preparation

- **#1 Know your Audience**
- **#2 Get your Workflow in Order**
  - Writing: LaTeX (e.g., Overleaf, TeXnicCenter), versioning (e.g., git), templates
  - Plotting: R (plot, ggplot), Python (matplotlib), Gnuplot
  - Figures: MS Visio, Inkscape → pdf, eps, svg (vector graphics)
- **#3 Mindset: Quality over Quantity**
  - Aim for top-tier conferences/journals (act as filter)
  - Make the paper useful for others (ideas, evidence, code)
  - **Example** (my own theses/books)
    - Seminar (~bachelor), 5 months, **446** pages
    - Diploma (~master), 9 months, **274** pages
    - PhD thesis, 4 years, **237** pages
    - 1st book, 5+2 years, **157** pages



Your reader's time is a scarce resource

Paper Writing and Publication Process

- **Research – Writing Cycle**
  - Read lots of papers
  - ~~Idea → Research → Writing → Document~~
  - Idea → Writing/Research → Document
  - Incremental refinement of drafts

- **Paper Submission Cycle**
  - Blind vs double-blind submission
  - Revisions and Camera-ready
  - **Similar: bachelor**/master thesis
    → drafts to advisor / final version

- **Demo CMT**
  - Example DEEM 2021, PVLDB 2022

[Recommended Reading]

[Simon Peyton Jones: How to write a great research paper, MSR Cambridge]

[Eamonn Keogh: How to do good research, get it published in SIGKDD and get it cited!, **KDD 2009**]

Example SIGMOD 2020

October 15, 2019: Abstract submission
October 22, 2019: Paper submission
December 10 - 11, 2019: Author responses
January 17, 2020: Initial notification
February 19, 2020: Revised submission
March 13, 2020: Final notification
April 12, 2020: Camera ready due

Paper/Thesis Structure by Example

- **Example Paper**

[Ahmed Elgohary, Matthias Boehm, Peter J. Haas, Frederick R. Reiss, Berthold Reinwald: Compressed Linear Algebra for Large-Scale Machine Learning. **PVLDB 2016**]

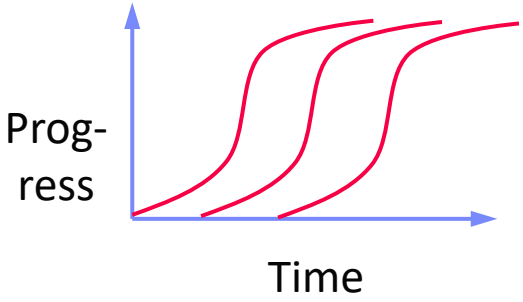
[Ahmed Elgohary, Matthias Boehm, Peter J. Haas, Frederick R. Reiss, Berthold Reinwald: Scaling Machine Learning via Compressed Linear Algebra. **SIGMOD Record 2017 46(1)**]

[Ahmed Elgohary, Matthias Boehm, Peter J. Haas, Frederick R. Reiss, Berthold Reinwald: Compressed Linear Algebra for Large-Scale Machine Learning. **VLDB Journal 2018 27(5)**]

[Ahmed Elgohary, Matthias Boehm, Peter J. Haas, Frederick R. Reiss, Berthold Reinwald: Compressed Linear Algebra for Large-Scale Machine Learning. **Commun. ACM 2019 62(5)**]

Ideas and Topic Selection

- **Problem-Oriented Research**
  - Focus on problem/observation first, not your solution
  - **Discuss early ideas** with collaborators and friends
  - Develop your taste for good research topics
  - Topic selection needs time → **pipeline model**



- **Ex. Compressed Linear Algebra**
  - **Problem:** Iterative ML algorithms + memory-bandwidth-bound operations ➔ crucial to fit data in memory ➔ automatic lossless compression
  - **Sub-problems:** #rows>>#cols, column correlation, column characteristics ➔ column-wise compression w/ heterogeneous encoding formats

Prototypes and Experiments

- **Worst Mistake: Schrödinger's Results**
  - Postpone implementation and experiments till last before the deadline
  - No feedback, no reaction time (experiments require many iterations)
  - **Karl Popper:** falsifiability of scientific results
- **Continuous Experiments**
  - Run experiments during survey / prototype building
  - Systematic experiments ➔ observations and ideas for improvements
  - Don't be afraid of throw away prototypes that don't work
- **Ex. Compressed Linear Algebra**
  - Data characteristics inspired overall design of encoding schemes
  - Initially slow compression → dedicated sampling schemes and estimators
  - Initially slow compressed operations → cache-conscious operations, selected operations with better asymptotic behavior

Title and Authors

- **List of Authors**
  - #1 by contribution (main, …, advisor)
  - #2 by last name

Compressed Linear Algebra for Large-Scale Machine Learning

Ahmed Elgohary[2]*; Matthias Boehm[1], Peter J. Haas[1], Frederick R. Reiss[1], Berthold Reinwald[1]

[1] IBM Research – Almaden; San Jose, CA, USA
[2] University of Maryland; College Park, MD, USA

- **Title**
  - Descriptive yet concise
  - Short name if possible
    → easier to cite and discuss

SPOOF: Sum-Product Optimization and Operator Fusion for Large-Scale Machine Learning

Tarek Elgamal[2]; Shangyu Luo[3]; Matthias Boehm[1], Alexandre V. Evfimievski[1], Shirish Tatikonda[4]; Berthold Reinwald[1], Prithviraj Sen[1]

[1] IBM Research – Almaden; San Jose, CA, USA
[2] University of Illinois; Urbana-Champaign, IL, USA
[3] Rice University; Houston, TX, USA
[4] Target Corporation; Sunnyvale, CA, USA

SliceLine: Fast, Linear-Algebra-based Slice Finding for ML Model Debugging

Svetlana Sagadeeva*
Graz University of Technology

Matthias Boehm
Graz University of Technology

[**Credit:** sliceline, Silicon Valley, HBO]

MNC: Structure-Exploiting Sparsity Estimation for Matrix Expressions

Johanna Sommer
IBM Germany

Matthias Boehm
Graz University of Technology

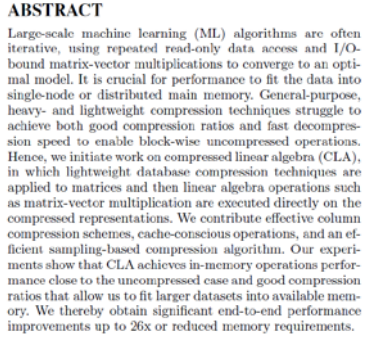
Alexandre V. Evfimievski
IBM Research – Almaden

Berthold Reinwald
IBM Research – Almaden

Peter J. Haas
UMass Amherst

Abstract

[Simon Peyton Jones: How to write a great research paper, MSR Cambridge]



ABSTRACT

Large-scale machine learning (ML) algorithms are often iterative, using repeated read-only data access and I/O-bound matrix-vector multiplications to converge to an optimal model. It is crucial for performance to fit the data into single-node or distributed main memory. General-purpose, heavy- and lightweight compression techniques struggle to achieve both good compression ratios and fast decompression speed to enable block-wise uncompressed operations. Hence, we initiate work on compressed linear algebra (CLA), in which lightweight database compression techniques are applied to matrices and then linear algebra operations such as matrix-vector multiplication are executed directly on the compressed representations. We contribute effective column compression schemes, cache-conscious operations, and an efficient sampling-based compression algorithm. Our experiments show that CLA achieves in-memory operations performance close to the uncompressed case and good compression ratios that allow us to fit larger datasets into available memory. We thereby obtain significant end-to-end performance improvements up to 26x or reduced memory requirements.

% 1. State the problem

Large-scale machine learning (ML) algorithms are often iterative, using repeated read-only data access and I/O-bound matrix-vector multiplications to converge to an optimal model. It is crucial for performance to fit the data into single-node or distributed main memory.

% 2. Say why it's an interesting problem

General-purpose, heavy- and lightweight compression techniques struggle to achieve both good compression ratios and fast decompression speed to enable block-wise uncompressed operations.

% 3. Say what your solution achieves

Hence, we initiate work on compressed linear algebra (CLA), in which lightweight database compression techniques are applied to matrices and then linear algebra operations such as matrix-vector multiplication are executed directly on the compressed representations. We contribute effective column compression schemes, cache-conscious operations, and an efficient sampling-based compression algorithm. Our experiments show that CLA achieves in-memory operations performance close to the uncompressed case and good compression ratios that allow us to fit larger datasets into available memory.

% 4. Say what follows from your solution

We thereby obtain significant end-to-end performance improvements up to 26x or reduced memory requirements.

Introduction

- **Context** (1 paragraph)
- **Problems** (1-3 paragraphs)
- [Existing Work (1 paragraph)]
- [Idea (1 paragraph)]
- **Contributions** (1 paragraph)

Contributions: Our major contribution is to make a case for *compressed linear algebra*, where linear algebra operations are directly executed over compressed matrices. We leverage ideas from database compression techniques and sparse matrix representations. The novelty of our approach is a combination of both, leading towards a generalization of sparse matrix representations and operations. The structure of the paper reflects our detailed technical contributions:

- *Workload Characterization:* We provide the background and motivation for CLA in Section 2 by giving an overview of Apache SystemML, and describing typical linear algebra operations and data characteristics.
- *Compression Schemes:* We adapt several column-based compression schemes to numeric matrices in Section 3 and describe efficient, cache-conscious core linear algebra operations over compressed matrices.
- *Compression Planning:* In Section 4, we further provide an efficient sampling-based algorithm for selecting a good compression plan, including techniques for compressed-size estimation and column grouping.
- *Experiments:* Finally, we integrated CLA into Apache SystemML. In Section 5, we study a variety of full-fledged ML algorithms and real-world datasets in both single-node and distributed settings. We also compare CLA against alternative compression schemes.


Compressed Linear Algebra for Large-Scale Machine Learning

Ahmed Elgohary[1], Matthias Boehm[2], Peter J. Haas[2], Frederick R. Reiss[2], Berthold Reinwald[2]

[1] BM Research – Almaden, San Jose, CA, USA
[2] University of Maryland, College Park, MD, USA

ABSTRACT

1. INTRODUCTION

Figure 1: Goals of Compressed Linear Algebra.

2. BACKGROUND AND MOTIVATION

2.1 SystemML Architecture

Introduction Matters

Anchoring: most reviewers reach their opinion after reading introduction and motivation and then look for evidence

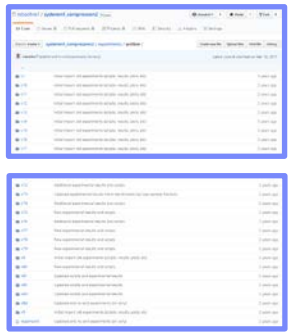
[Eamonn Keogh: How to do good research, get it published in SIGKDD and get it cited!, **KDD 2009**]

Writing the Paper (and more Experiments)

02 Scientific Reading and Writing
03 Experiments and Reproducibility

- **Easily Readable: Quality ∝ Time**
  - Make it easy to skim the paper
    → paragraph labels, self-explanatory figures (close to text), and structure
  - Avoid unnecessary formalism → as simple as possible
  - Shortening the text in favor of structure improves readability
  - Ex. Compressed Linear Algebra
    - Initial SIGMOD submission: **12+3 pages**
    - Final PVLDB submission: **12 pages**
      (+ more figures, experiments, etc)

- **Solid, Reproducible Experiments**
  - Create, use, and share dedicated benchmarks / datasets
  - Avoid weak baselines, start early w/ baseline comparisons
  - Automate your experiments as much as possible
  - Keep repository of all scripts, results, and used parameters

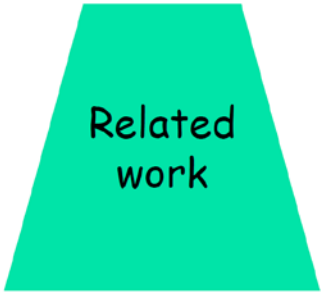


Related Work

- **Purpose of a "Related Work"-Section**
  - **Not** a mandatory task or to show you know the field
  - Put you work in context of related areas (~ 1 paragraph each)
  - Discuss closely related work
  - **Crisp separation from existing work** (what are the differences)

- **Placement**
  - Section 2 or **Section n-1**
  - Throughout the paper

[Simon Peyton Jones: How to write a great research paper, MSR Cambridge]



Your reader

Your idea

- **Give Credit**
  - Cite broadly, **give credit to inspiring ideas**, create connections
  - Honestly acknowledge **limitations of your approach**

References

- **Setup**
  - Use LaTeX **\cite**{} and BibTeX
  - Use a consistent source of bibtex entries (e.g., DBLP)

```
inproceedings{StonebrakerBPR11,
  author    = {Michael {Stonebraker et al.}},
  title     = {{The Architecture of SciDB}},
  booktitle = {{SSDBM}},
  year      = {2011}
}
```

VLDB2016.bib

\bibliographystyle{abbrv}
\bibliography{VLDB2016}

- **Different References Styles**
  - But, **not in footnotes** (unless required)
  - **8. REFERENCES**

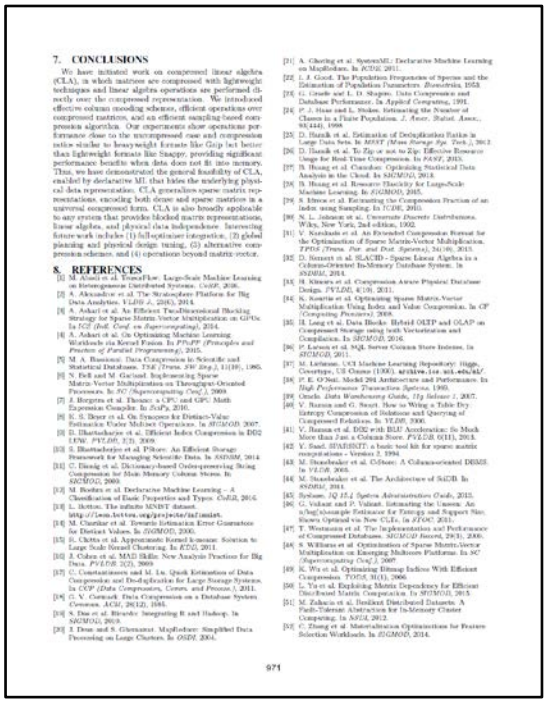
    [1] M. Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*, 2016.

    [2] A. Alexandrov et al. The Stratosphere Platform for Big Data Analytics. *VLDB J.*, 23(6), 2014.
  - **References**

    [Al14] Alexandrov, A. et al.: The Stratosphere platform for big data J. 23/6, 2014.

    [AS14] Arap, O.; Swany, M.: Offloading MPI Parallel Prefix Scan the NetFPGA. CoRR abs/1408.4939/, 2014.
  - **References**

    Jaume Amores. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence.*

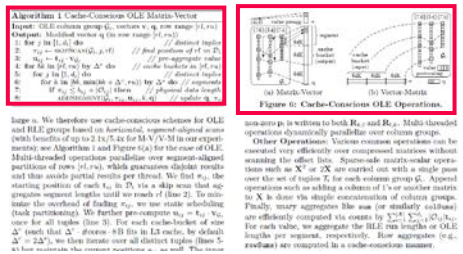


Dealing with Feedback / Criticism

- **Different Kinds of Feedback**
  - Casual discussion of early ideas
  - Comments on paper drafts
  - Reviewer comments (good and bad)

 # Always welcome feedback/criticism
 # Address all feedback w/ sincere effort

- **Example Compressed Linear Algebra**
  - SIGMOD Reviewer 2 (REJECT)
    - "The rewriting for q=Xv seems wrong: To compute q, one takes each row of the matrix X and multiplies it with the vector v."
  - PVLDB Reviewer 3 (WEAK ACCEPT)
    - "I kinda disagree with the broad definition of declarative ML from the introduction."



[Matthias Boehm et al.: Declarative Machine Learning - A Classification of Basic Properties and Types. **CoRR 2016**.]

- **Paper Rebuttal and/or Revision**
  - **Rebuttal**: seriously consider all feedback (in doubt agree), and answer with facts / ideas how to address the comments
  - **Revision** (conditional accept): address all revision requests

Paper Projects

In Computer Science (Data Management)

Paper Projects

- **Visual Analytics**
  - **#1.1** Tarique Siddiqui, Paul Luh, Zesheng Wang, Karrie Karahalios, Aditya G. Parameswaran: ShapeSearch: A Flexible and Efficient System for Shape-based Exploration of Trendlines. SIGMOD 2020
  - **#1.2** Uwe Jugel, Zbigniew Jerzak, Gregor Hackenbroich, Volker Markl: M4: A Visualization-Oriented Time Series Data Aggregation. PVLDB 7(10) 2014

- **Video Analytics**
  - **#2.1** Favyen Bastani, Songtao He, Arjun Balasingam, Karthik Gopalakrishnan, Mohammad Alizadeh, Hari Balakrishnan, Michael J. Cafarella, Tim Kraska, Sam Madden: MIRIS: Fast Object Track Queries in Video. SIGMOD 2020
  - **#2.2** Daniel Kang, Peter Bailis, Matei Zaharia: BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. PVLDB 13(4) 2019
  - **#2.3** Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, Matei Zaharia: NoScope: Optimizing Deep CNN-Based Queries over Video Streams at Scale. PVLDB 10(11) 2017

Paper Projects, cont.

- **Fairness / Causality / Diversity**
  - **#3.1** Julia Stoyanovich, Bill Howe, H. V. Jagadish: Responsible Data Management. PVLDB 13(12) 2020
  - **#3.2** Babak Salimi, Luke Rodriguez, Bill Howe, Dan Suciu: Interventional Fairness: Causal Database Repair for Algorithmic Fairness. SIGMOD 2019
  - **#3.3** Marina Drosou, Evaggelia Pitoura: DisC diversity: result diversification based on dissimilarity and coverage. PVLDB 6(1) 2012
- **Graphs**
  - **#4.1** Renchi Yang, Jieming Shi, Xiaokui Xiao, Yin Yang, Juncheng Liu, Sourav S. Bhowmick: Scaling Attributed Network Embedding to Massive Graphs. PVLDB 14(1) 2020
  - **#4.2** Siddhartha Sahu, Amine Mhedhbi, Semih Salihoglu, Jimmy Lin, M. Tamer Özsu: The Ubiquity of Large Graphs and Surprising Challenges of Graph Processing. PVLDB 11(4) 2017
  - **#4.3** Yuanyuan Tian, Andrey Balmin, Severin Andreas Corsten, Shirish Tatikonda, John McPherson: From "Think Like a Vertex" to "Think Like a Graph". PVLDB 7(3) 2013
  - **#4.4** Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, Grzegorz Czajkowski: Pregel: a system for large-scale graph processing. SIGMOD 2010

Paper Projects, cont.

- **NLP**
  - **#5.1** Orest Gkini, Theofilos Belmpas, Georgia Koutrika, Yannis E. Ioannidis: An In-Depth Benchmarking of Text-to-SQL Systems. SIGMOD 2021
  - **#5.2** Diptikalyan Saha, Avrilia Floratou, Karthik Sankaranarayanan, Umar Farooq Minhas, Ashish R. Mittal, Fatma Özcan: ATHENA: An Ontology-Driven System for Natural Language Querying over Relational Data Stores. PVLDB 9(12) 2016
  - **#5.3** Fei Li, H. V. Jagadish: Constructing an Interactive Natural Language Interface for Relational Databases. PVLDB 8(1) 2014

- **Provenance**
  - **#6.1** Pingcheng Ruan, Gang Chen, Anh Dinh, Qian Lin, Beng Chin Ooi, Meihui Zhang: Fine-Grained, Secure and Efficient Data Provenance for Blockchain. PVLDB 12(9) 2019
  - **#6.2** Daniel Deutch, Nave Frost, Amir Gilad: Provenance for Natural Language Queries. PVLDB 10(5) 2017
  - **#6.3** Adriane Chapman, H. V. Jagadish: Why not? SIGMOD 2009

Paper Projects, cont.

- **Reusing Intermediates**
  - **#7.1** Behrouz Derakhshan, Alireza Rezaei Mahdiraji, Ziawasch Abedjan, Tilmann Rabl, Volker Markl: Optimizing Machine Learning Workloads in Collaborative Environments. SIGMOD 2020
  - **#7.2** Doris Xin, Litian Ma, Jialin Liu, Stephen Macke, Shuchen Song, Aditya Parameswaran: Helix: Accelerating Human-in-the-loop Machine Learning. PVDLB 11(12) 2018
  - **#7.3** Ce Zhang, Arun Kumar, Christopher Ré: Materialization optimizations for feature selection workloads. SIGMOD 2014
  - **#7.4** Milena Ivanova, Martin L. Kersten, Niels J. Nes, Romulo Goncalves: An architecture for recycling intermediates in a column-store. SIGMOD 2009
- **Raw Query Processing**
  - **#8.1** Dominik Durner, Viktor Leis, Thomas Neumann: JSON Tiles: Fast Analytics on Semi-Structured Data. SIGMOD 2021
  - **#8.2** Manos Karpathiotakis, Ioannis Alagiannis, Anastasia Ailamaki: Fast Queries Over Heterogeneous Data Through Engine Customization. PVLDB 9(12) 2016
  - **#8.3** Ioannis Alagiannis, Renata Borovica, Miguel Branco, Stratos Idreos, Anastasia Ailamaki: NoDB: efficient query execution on raw data files. SIGMOD 2012

Paper Projects, cont.

- **Transactions / High Availability**
  - **#9.1** Jingyu Zhou et al: FoundationDB: A Distributed Unbundled Transactional Key Value Store. SIGMOD 2021
  - **#9.2** Audrey Cheng, Xiao Shi, Lu Pan, Anthony Simpson, Neil Wheaton, Shilpa Lawande, Nathan Bronson, Peter Bailis, Natacha Crooks, Ion Stoica: RAMP-TAO: Layering Atomic Transactions on Facebook's Online TAO Data Store. PVLDB 14(12) 2021
  - **#9.3** Yihe Huang, William Qian, Eddie Kohler, Barbara Liskov, Liuba Shrira: Opportunities for Optimism in Contended Main-Memory Multicore Transactions. PVLDB 13(5) 2020
  - **#9.4** Umar Farooq Minhas, Shriram Rajagopalan, Brendan Cully, Ashraf Aboulnaga, Kenneth Salem, Andrew Warfield: RemusDB: Transparent High Availability for Database Systems. PVLDB 4(11) 2011
  - **#9.5** Michael J. Cahill, Uwe Röhm, Alan D. Fekete: Serializable isolation for snapshot databases. SIGMOD 2008
- **Data Flow Systems**
  - **#10.1** Matei Zaharia et al.: Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. NSDI 2012
  - **#10.2** Christopher Olston, Shubham Chopra, Utkarsh Srivastava: Generating example data for dataflow programs. SIGMOD 2009

Paper Projects, cont.

- **Database Cracking**
  - **#11.1** Pedro Holanda, Stefan Manegold, Hannes Mühleisen, Mark Raasveldt: Progressive Indexes: Indexing for Interactive Data Analysis. PVLDB 12(13) 2019
  - **#11.2** Felix Martin Schuhknecht, Alekh Jindal, Jens Dittrich: The Uncracked Pieces in Database Cracking. PVLDB 7(2) 2013
  - **#11.3** Stratos Idreos, Martin L. Kersten, Stefan Manegold: Database Cracking. CIDR 2007

- **Index Structures**
  - **#12.1** Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, Neoklis Polyzotis: The Case for Learned Index Structures. SIGMOD 2018
  - **#12.2** Huanchen Zhang, Hyeontaek Lim, Viktor Leis, David G. Andersen, Michael Kaminsky, Kimberly Keeton, Andrew Pavlo: SuRF: Practical Range Query Filtering with Fast Succinct Tries. SIGMOD 2018
  - **#12.3** Viktor Leis, Alfons Kemper, Thomas Neumann: The adaptive radix tree: ARTful indexing for main-memory databases. ICDE 2013
  - **#12.4** Nicolas Bruno, Surajit Chaudhuri: Constrained physical design tuning. Proc. VLDB Endow. 1(1) 2008

Paper Projects, cont.

- **Data Cleaning**
  - **#13.1** Peng Li, Xi Rao, Jennifer Blase, Yue Zhang, Xu Chu, Ce Zhang: CleanML: A Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks. ICDE 2021
  - **#13.2** Mohammad Mahdavi, Ziawasch Abedjan: Baran: Effective Error Correction via a Unified Context Representation and Transfer Learning. PVLDB 13(11) 2020
  - **#13.3** Md Mahdavi et al: Raha: A Configuration-Free Error Detection System. SIGMOD 2019
  - **#13.4** Alireza Heidari, Joshua McGrath, Ihab F. Ilyas, Theodoros Rekatsinas: HoloDetect: Few-Shot Learning for Error Detection. SIGMOD 2019
  - **#13.5** Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, Christopher Ré: HoloClean: Holistic Data Repairs with Probabilistic Inference. PVLDB 10(11) 2017
  - **#13.6** Ziawasch Abedjan et al: Detecting Data Errors: Where are we and what needs to be done? PVLDB 9(12) 2016
- **Query Optimization**
  - **#14.1** Ryan Marcus, Parimarjan Negi, Hongzi Mao, Nesime Tatbul, Mohammad Alizadeh, Tim Kraska: Bao: Making Learned Query Optimization Practical. SIGMOD 2021
  - **#14.2** Guido Moerkotte, Thomas Neumann: Analysis of Two Existing and One New Dynamic Programming Algorithm for the Generation of Optimal Bushy Join Trees without Cross Products. VLDB 2006

Paper Projects, cont.

- **Query Compilation**
  - **#15.1** Timo Kersten, Viktor Leis, Alfons Kemper, Thomas Neumann, Andrew Pavlo, Peter A. Boncz: Everything You Always Wanted to Know About Compiled and Vectorized Queries But Were Afraid to Ask. PVLDB 11(13) 2018
  - **#15.2** Prashanth Menon, Andrew Pavlo, Todd C. Mowry: Relaxed Operator Fusion for In-Memory Databases: Making Compilation, Vectorization, and Prefetching Work Together At Last. PVLDB 11(1) 2017
  - **#15.3** Andrew Crotty, Alex Galakatos, Kayhan Dursun, Tim Kraska, Carsten Binnig, Ugur Çetintemel, Stan Zdonik: An Architecture for Compiling UDF-centric Workflows. PVLDB 8(12) 2015
  - **#15.4** Milos Nikolic, Mohammed Elseidy, Christoph Koch: LINVIEW: incremental view maintenance for complex analytical queries. SIGMOD 2014
  - **#15.5** Yannis Klonatos, Christoph Koch, Tiark Rompf, Hassan Chafi: Building Efficient Query Engines in a High-Level Language. PVLDB 7(10) 2014
  - **#15.6** Thomas Neumann: Efficiently Compiling Efficient Query Plans for Modern Hardware. Proc. VLDB Endow. 4(9) 2011

Paper Projects, cont.

- **Compression**
  - **#16.1** Peter A. Boncz, Thomas Neumann, Viktor Leis: FSST: Fast Random Access String Compression. Proc. VLDB Endow. 13(11) 2020
  - **#16.2** Daniel J. Abadi, Samuel Madden, Miguel Ferreira: Integrating compression and execution in column-oriented database systems. SIGMOD 2006
  - **#16.3** Marcin Zukowski, Sándor Héman, Niels Nes, Peter A. Boncz: Super-Scalar RAM-CPU Cache Compression. ICDE 2006
- **Modern Hardware**
  - **#17.1** Lasse Thostrup, Jan Skrzypczak, Matthias Jasny, Tobias Ziegler, Carsten Binnig: DFI: The Data Flow Interface for High-Speed Networks. SIGMOD 2021
  - **#17.2** Clemens Lutz, Sebastian Breß, Steffen Zeuch, Tilmann Rabl, Volker Markl: Pump Up the Volume: Processing Large Data on GPUs with Fast Interconnects. SIGMOD 2020
  - **#17.3** Ismail Oukid et al.: FPTree: A Hybrid SCM-DRAM Persistent and Concurrent B-Tree for Storage Class Memory. SIGMOD 2016
  - **#17.4** Changkyu Kim et al.: FAST: fast architecture sensitive tree search on modern CPUs and GPUs. SIGMOD 2010
  - **#17.5** René Müller, Jens Teubner, Gustavo Alonso: Data Processing on FPGAs. Proc. VLDB Endow. 2(1) 2009

Summary and Q&A

- **Data Management Group**
- **Course Organization, Outline, and Projects**
- **Structure of Scientific Papers**
- **Paper Project Proposals**

- **Remaining Questions?**

- **Next Lectures**
  - 02 Scientific Reading and Writing [Nov 04]
  - 03 Experiments, Reproducibility, and Projects [Nov 11]