Dynamic Data Citation
Andreas Rauber
30.1.2017, Brussels

Background

Business Intelligence / Data Science:

- Machine Learning, Signal Processing, Data Warehousing, IR

Complex data analytics projects

CRISP-DM: Cross-Industry Standard Process for Data Mining

Trust: Traceability, Reproducibility

ACM Statement on Algorithmic Transparency & Accountability

- How can we document the processes?
- How can we identify the data that was used?

usacm ACM US Public Policy Council

acm Europe Council

Updated May 25, 2017

Statement on Algorithmic Transparency and Accountability

by ACM U.S. Public Policy Council, approved January 12, 2017
ACM Europe Policy Committee, approved May 25, 2017

Computer algorithms are widely employed throughout our economy and society to make decisions that have far-reaching impacts, including their applications for education, access to credit, healthcare, and employment. The ubiquity of algorithms in our everyday lives is an important reason to focus on addressing challenges associated with the design and technical aspects of algorithms and preventing bias from the onset.

An algorithm is a self-contained step-by-step set of operations that computers and other 'smart' devices carry out to perform calculation, data processing, and automated reasoning tasks. Increasingly, algorithms implement institutional decision-making based on analytics, which involves the discovery, interpretation, and communication of meaningful patterns in data. Especially valuable in areas rich with recorded information, analytics relies on the simultaneous application of statistics, computer programming, and operations research to quantify performance.

There is also growing evidence that some algorithms and analytics can be opaque, making it impossible to determine when their outputs may be biased or erroneous.

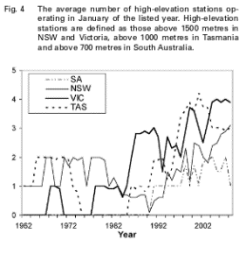
Computational models can be distorted as a result of biases contained in their input data and/or their algorithms. Decisions made by predictive algorithms can be opaque because of many factors, including technical (the algorithm may not lend itself to easy explanation), economic (the cost of providing transparency may be excessive, including the compromise of trade secrets), and social (revealing input may violate privacy expectations). Even well-engineered computer systems can result in unexplained outcomes or errors, either because they contain bugs or because the conditions of their use changes, invalidating assumptions on which the original analytics were based.

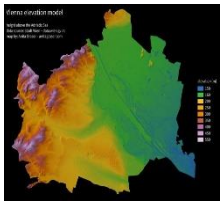
The use of algorithms for automated decision-making about individuals can result in harmful discrimination. Policymakers should hold institutions using analytics to the same standards as institutions where humans have traditionally made decisions and developers should plan and architect analytical systems to adhere to those standards when algorithms are used to make automated decisions or as input to decisions made by people.

This set of principles, consistent with the ACM Code of Ethics, is intended to support the benefits of algorithmic decision-making while addressing these concerns. These principles should be addressed during every phase of system development and deployment to the extent necessary to minimize potential harms while realizing the benefits of algorithmic decision-making.

Motivation

Identifying the data used seems trivial:

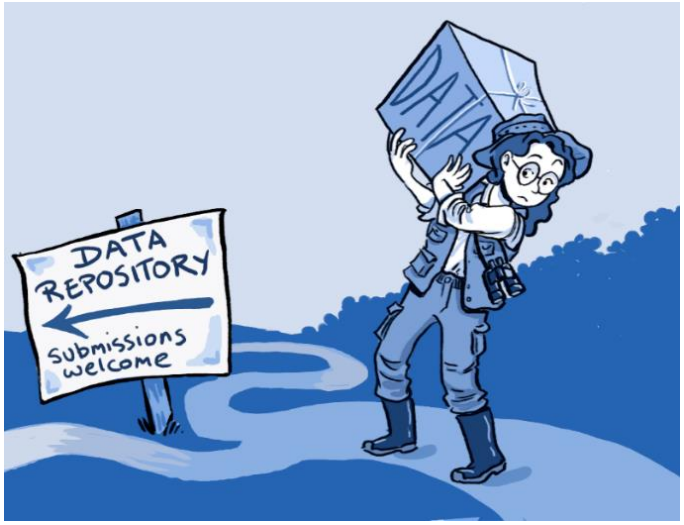




1. Put data in data repository
2. Assign identifier (DOI, Ark, URI, …)
3. Make (and keep) it accessible
4. Refer to it in analysis document / dashboard / …

So where are the challenges?

- Dynamics
- Granularity



Identification of Dynamic Data

Identifiable datasets usually have to be static

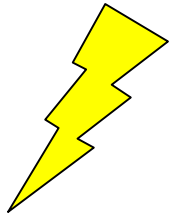
- Fixed set of data, no changes:
  no corrections to errors, no new data being added

But: data is **dynamic**

- Adding new data, correcting errors, enhancing data quality, …
- Changes sometimes highly dynamic, at irregular intervals

Current approaches

- Identifying entire data stream, without any versioning
- Using “accessed at” date
- “Artificial” versioning by identifying batches of data (e.g. annual), aggregating changes into releases (time-delayed!)



Would like to identify precisely the **data as it existed at any specific point in time**

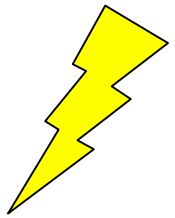
Granularity of Subsets

What about the **granularity** of data to be identified?

- Massive collections of data in any repository
- Analysts use specific subsets of data
- Need to precisely identify the subset used

Current approaches

- Storing a copy of subset as used in study -> scalability
- Citing entire dataset, providing textual description of subset (methods section) -> imprecise (ambiguity)
- Storing list of record identifiers in subset -> scalability, not for arbitrary subsets (e.g. when not entire record selected)



Would like to be able to identify precisely the **subset of (dynamic) data used** in a process

Data Citation – Requirements

- **Allow analysts to easily identify the data used**
- Dynamic data, for any type of data
  - corrections, additions, … for relational DBs, XML, files, …
- Arbitrary subsets of data (granularity)
  - rows/columns, time sequences, …
  - from single number to the entire set
- Stable across technology changes
  - e.g. migration to new database system
- Machine-actionable
  - not just machine-readable,
    definitely not just human-readable and interpretable
- Scalable to very large / highly dynamic datasets
  - But: should also work for small and/or static datasets!

Dynamic Data Citation

We have: Data + Means-of-access

Dynamic Data Citation

We have: Data + Means-of-access

Dynamic Data Citation:
Cite (dynamic) data dynamically via query!

Dynamic Data Citation

We have: Data + Means-of-access

Dynamic Data Citation:
Cite (dynamic) data dynamically via query!

Steps:

1. Data → versioned (history, with time-stamps)

Dynamic Data Citation

We have: Data + Means-of-access

Dynamic Data Citation:
Cite (dynamic) data dynamically via query!

Steps:

1. Data → versioned (history, with time-stamps)

Researcher creates working-set via some interface:

2. Access → **store & assign persistent identifier to "QUERY"**, plus
   - **Time-stamping** for re-execution against versioned DB
   - **Re-writing** for normalization, unique-sort, mapping to history
   - **Hashing** result-set: verifying identity/correctness

   (plus a few more things) leading to landing page

S. Pröll, A. Rauber. **Scalable Data Citation in Dynamic Large Databases: Model and Reference Implementation.** In IEEE Intl. Conf. on Big Data 2013 (IEEE BigData2013), 2013
http://www.ifs.tuwien.ac.at/~andi/publications/pdf/pro_ieeebigdata13.pdf

Data Citation – Output

- 14 Recommendations
  grouped into 4 phases:
  - Preparing data and query store
  - Persistently identifying specific data sets
  - Resolving PIDs
  - Upon modifications to the data infrastructure
- 2-page flyer
  https://rd-alliance.org/recommendations-working-group-data-citation-revision-oct-20-2015.html
- More detailed report: Bulletin of IEEE TCDL, 12(1), 2016
  http://www.ieee-tcdl.org/Bulletin/v12n1/papers/IEEE-TCDL-DC-2016_paper_1.pdf

Data Citation – Recommendations

Preparing Data & Query Store

- R1 – Data Versioning
- R2 – Timestamping
- R3 – Query Store

When Resolving a PID

- R11 – Landing Page
- R12 – Machine Actionability

When Data should be persisted

- R4 – Query Uniqueness
- R5 – Stable Sorting
- R6 – Result Set Verification
- R7 – Query Timestamping
- R8 – Query PID
- R9 – Store Query
- R10 – Citation Text

Upon Modifications to the Data Infrastructure

- R13 – Technology Migration
- R14 – Migration Verification

Dynamic Data Citation

- Analyst uses workbench to identify subset of data
- Upon executing selection („download“) user gets
  - Data (package, access API, …)
  - PID (e.g. DOI) (Query is time-stamped and stored)
  - Hash value computed over the data for local storage
  - Recommended citation text (e.g. BibTeX, Endnote, text)
- PID resolves to landing page
  - Provides detailed metadata, link to parent data set, subset,…
  - Option to retrieve original data OR current version OR changes
- Upon activating PID associated with a data citation
  - Query is re-executed against time-stamped and versioned DB
  - Results as above are returned
- Query store aggregates data usage

Dynamic Data Citation

Note: query string provides excellent provenance information on the data set!

- A … of data
- … gets
  - Data (package, access API, …)
  - PID (e.g. DOI) (Query time-stamped and stored)
  - Hash value computed over the data for local storage
  - Recommended citation text (e.g. BibTeX)
- PID resolves to landing page
  - Provides detailed metadata, link to parent data set, subset,…
  - Option to retrieve original data OR current version OR changes
- Upon activating PID associated with a data citation
  - Query is re-executed against time-stamped and versioned DB
  - Results as above are returned
- Query store aggregates data usage

Dynamic Data Citation

- Note: query string provides excellent provenance information on the data set!
- This is an important advantage over traditional approaches relying on, e.g. storing a list of identifiers/DB dump!!!
- [illegible] of data
- [illegible] gets
  - Data (pack
  - PID (e.g. D
  - Hash value
  - Recommended citation text (e.g. BibTeX)
- PID resolves to landing page
  - Provides detailed metadata, link to parent data set, subset,…
  - Option to retrieve original data OR current version OR changes
- Upon activating PID associated with a data citation
  - Query is re-executed against time-stamped and versioned DB
  - Results as above are returned
- Query store aggregates data usage

Dynamic Data Citation

- A[illegible] of data
- [illegible]gets
  - Data (pack[illegible]
  - PID (e.g. D[illegible]
  - Hash value[illegible]
  - Recommended citatio[illegible]ext (e.g. L[illegible]TeX)
- P[illegible]subset,…
  [illegible]R changes
- Upon activating PI[illegible]associated with a data citation
  - Query is re-execute[illegible] against time-stamped and versioned DB
  - Results as above a[illegible]returned
- Query store aggregates data usage

Note: query string provides excellent provenance information on the data set!

This is an important advantage over traditional approaches relying on, e.g. storing a list of identifiers/DB dump!!!

Identify which parts of the data are used. If data changes, identify which queries (studies) are affected

Pilots / Adopters

- Series of Webinars presenting implementations
  - Recordings, slides, supporting papers https://www.rd-alliance.org/group/data-citation-wg/ webconference/webconference-data-citation-wg.html
  - Implementing of the RDA Data Citation Recommendations by the **Climate Change Centre Austria (CCCA) for a repository of NetCDF files**
  - Implementing the RDA Data Citation Recommendations for **Long-Tail Research Data / CSV files**
  - Implementing the RDA Data Citation Recommendations in the **Distributed Infrastructure of the Virtual and Atomic Molecular Data Center (VAMDC)**
  - Implementation of Dynamic Data Citation at the **Vermont Monitoring Cooperative**
  - Adoption of the RDA Data Citation of Evolving Data Recommendation to **Electronic Health Records**

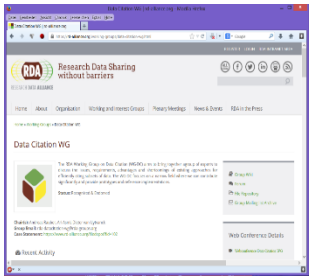
Industry Partners

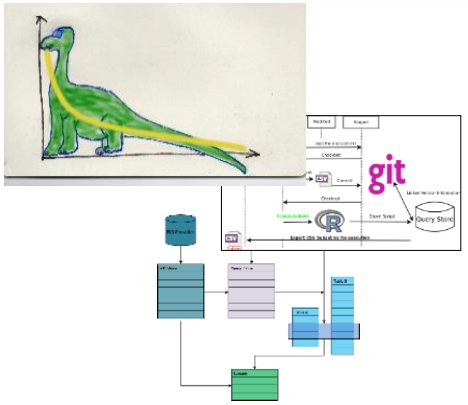
- (Less willing to share insights into their DM practices)
- IT Solutions company
  - Enterprise offering data management / transformation / data migration / ETL services
  - Relational Databases (Oracle, MySQL, Postgres): study on different versioning / historization approaches
  - Question: how to deal with schema evolution?
  - Likely will get permission to publish the study
- In many cases core building blocks already in place (versioning, query processing, …)
- Straightforward mechanism to add auditability for source data used in analysis / processing
- Add query re-writing, storing queries, interface adaptations
- **Effort** required: it depends – pilots: **5-8 PM**

Summary - Advantages

- **Precisely identify any arbitrary subset of data**
- Principles applicable to all types of data
- Straightforward to implement in most settings
- Optimizations for high-volume / very dynamic data possible
- Transparent for the analyst / data scientist
- Reduces documentation effort for analysts / data scientist
- Reduces data management complexity for data centre
- Increases traceability of results, **trust**

Thank you!





Thanks!

https://rd-alliance.org/working-groups/data-citation-wg.html