CS594

Provenance & Explanations

1 - Introduction & Overview

Course webpage

Boris Glavic

bglavic@uic.edu

Introduction

Course organization

Provenance & Explanations - Introduction

Course organization

Course organization

- What will you learn in this course?
- Logistics

Provenance & Explanations - Introduction

Course organization

Course organization

- What will you learn in this course?
- Logistics

Motivation

- **Provenance and explanations are essential tools for building trust-worthy, secure, transparent, and fair data-intensive systems and machine learning pipelines.**
- These tools are used to
  - debug analysis results
  - to comprehend the results of complex queries
  - to explore the impact of hypothetical changes to data and/or policies
  - to audit sensitive computations
  - to justify and understand predictions made by machine learning models

Covered Topics

The following topics will be covered in the course:

- **Provenance & Explanations - Introduction**
  - Motivation & use cases
  - Provenance graphs
  - Explanations for query answers
- **Provenance models**
- **Hypothetical reasoning: what-if and how-to**
  - Incremental view maintenance / what-if queries
  - View update & how-to
- **Explanations**
  - Counterfactual explanations
  - Explanations as (provenance) summarization
  - Attribution and degrees of responsibility (including game theoretic notions of attribution)

Covered Topics (continued)

- **Explaining missing answers**
- **Provenance capture & management**
  - How to compute provenance efficiently?
  - Storage and computation trade-offs
- **Building provenance-aware & explanation-ready systems**
  - Strategies for capturing and managing provenance
  - How to compute explanations efficiently?

Course organization

Course organization

- What will you learn in this course?
- Logistics

Reading Materials

- The following overview articles and textbooks will be helpful, but are optional
  - **Data Provenance - Origins, Applications, Algorithms, and Models.**, *Boris Glavic.* Foundations and Trends® in Databases, vol. 9 (3-4), 209-441, 2021. http://www.cs.uic.edu/%7ebglavic/dbgroup/assets/pdfpubls/G21.pdf
  - **Trends in Explanations: Understanding and Debugging Data-Driven Systems.**, *Boris Glavic, Alexandra Meliou, Sudeepa Roy*. Foundations and Trends® in Databases, vol. 11 (3), 226-318, 2021. http://www.cs.uic.edu/%7ebglavic/dbgroup/assets/pdfpubls/GMR21.pdf
  - **Principles of Data Integration**, 1th Edition, *Doan, Halevy, and Ives*, Morgan Kaufmann, 2012

Textbooks

- Depending on your background, a standard database textbook may be useful:
  - *Elmasri* and *Navathe*. **Fundamentals of Database Systems**, 6th Edition, Addison-Wesley, 2003
  - *Ramakrishnan* and *Gehrke*. **Database Management Systems**, 3nd Edition, McGraw-Hill, 2002
  - *Silberschatz*, *Korth*, and *Sudarshan*. **Database System Concepts**, 6th Edition, McGraw Hill, 2010
  - *Garcia-Molina*, *Ullman*, and *Widom*. **Database Systems: The Complete Book**, 2nd Edition, Prentice Hall, 2008

Prerequisites

- No formal prerequisites, but some background in databases (roughly equivalent to *CS480*) is expected.

Workload

1. Work on a semester-long research project related to implementing provenance or explanation techniques based on a research paper or working on developing new techniques
2. Review and present a state-of-the-art research paper from the field
3. Actively participate in class
4. Homework assignments / quizzes

Research project

- **research project options:**
  - implement and evaluate a technique from a recent research paper
  - do original research (possibly leading to a publication if there is interest)
- you can **choose** either
  - one of the example projects
  - propose your own project
- there will be several **meetings** to help you stay on track
  - decide project topic
  - project design
  - implementation / evaluation
  - presentation

Literature review

- **select a research paper**
  - from this list: paper list
  - you can access pdfs for these papers at: google drive
- **read the paper**
- **present & discuss the paper in class**
- **write summary & critique**

Grading

- Project: **40%**
- Paper review and presentation: **40%**
- Homework assignment & Quizzes: **10%**
- Active participation in class: **10%**

Class Outline

- **Background** - **first half of the semester**
  - lectures and discussion
  - introduce important background on provenance & explanations
- **Literature review presentations** - **second half of the semester**
  - students present research papers
- **Project presentations** - **end of the semester**
  - students present and demo their projects

Important Dates

- **Literature review**
  - Select a paper to review: **09/12**
  - Written paper summary due: **11/15**
  - Present & discuss paper in class: **starting mid / late October**
- **Research project**
  - Select a project topic: **09/17**
  - Finalize project plan and initial results: **10/15**
  - Finish project: **12/01**
  - Project presentations: **12/03**

Provenance & Explanations - Introduction

Course organization

Provenance & Explanations - Introduction
- Motivation & Use Cases
- Provenance & The W3C Prov Standard
- Recap

Provenance & Explanations - Introduction

What is Data Provenance?

- **Provenance** is **metadata** describing the **origin** and **creation process** of data
- **entity** - a piece of information or physical artifact whose origin we want to track
  - *file*, *database table*, *a database table's row or cell*, *physical contract*, *biological sample*, *python object*
- **activity** - a computation or physical process that may **use** and **generate** entities
  - *database query*, *python script*, *cell in a jupyter notebook*,
- **actor** - a person or machine that controls or executes an activity
  - *data analyst*, *DBA*, *developer*, *scientist*, *computer*, *cluster*, *cloud service*, *OS process*, *physical instrument*
- **dependencies** - track relationships between entities, activities and actors
  - **data dependencies** - an entity E_1 was derived from an entity E_2
  - **transformation dependencies**
    - an entity E was generated by an activity A
    - an entity E was accessed by an activity A

What are Explanations?

- **Explanations justify** and **explain** an outcome of a **computation**
- **"What" - the target** - the phenomenon we would like to explain
  - the prediction of an ML model
  - a query result
  - a point in a plot created by a computational notebook
  - ...
- **"Who" - the audience** - what is the target audience for the explanation
  - a data analysis expert
  - a domain scientist
  - a lay person
  - ...

What are Explanations?

- **"Why" - the purpose** - for which purpose is the explanation created
  - debugging
  - understanding
  - trust / transparency
  - auditing / justification
- **"How" - the methodology** - the form of the explanation and how it is created
  - provenance-based
  - summary-based
  - attribution / responsibility
  - differences / evolution

Usecase I

University	Department	NumStud	AvgGPA	Cred
UIC	CS	2404	3.5	14650
UIC	BIO	354	2.5	9.42
UIC	LAW	560	3.7	1650
Northwestern	CS	1450	3.1	12.2
UChicago	CS	780	-0.25	14923
ETH Zurich	CS	1200	1.6	3.2
University of Zurich	CS	560	1.3	0

Usecase II

Age	MaritalStatus	Gender	Property	Education	LoanGranted
30s	false	female	no	BS	no
40s	true	female	yes	BS	no
50s	true	female	yes	MS	no
40s	true	male	yes	highschool	yes
30s	false	male	no	highschool	no
50s	true	male	yes	BS	yes
30s	false	male	no	MS	yes

Usecase III

City	Neighborhood	AvgSal	NumCrimes
Chicago	X	45k	5000
Chicago	Y	35k	30
Chicago	Z	105k	150000
Chicago	A	50k	50
NY	G	145M	100540
NY	H	1.2M	600

Applications (end user)

- **error diagnosis and debugging**
  - tracing erroneous / interesting outputs back to problematic inputs / parts of the computation
  - which outputs are affected by problematic inputs
- **understanding / trust**
  - help users trust a result by helping them to understand how it was derived
- **data discovery / search**
  - *"find datasets that are based on 2022 census data"*
- **auditing**
  - prove how data was derived / handled / accessed

Applications (supporting technology)

- **probabilistic query processing**
- **hypothetical reasoning**
  - **what-if** analysis
  - **how-to** analysis
- **view update / incremental computation**
- **threat analysis**
- **improving query / computation performance**
- **automatic storage organization**
- **fine-grained access control**

Summary (Provenance)

- **Provenance is information about the creation process and origin of data**
  - **entities** (data items), **activities** (transformations), **actors** (humans, machines, . . .)
  - **dependencies**
    - **data-to-data**: entity `A` is **derived from** entity `B`
    - **data-and-computation**: entity `A` **was generated by** computation `C`, entity `A` **was accessed by** computation `C`
    - **data-and-actors**: entity `E` **is attributed to** actor `A`
    - **actors-and-activity**: actor `A` **controlled** activity `C`
  - **granularity**
    - **entities**: e.g., file, line, character, . . .
    - **activities**:, e.g., transaction, SQL statement, operator, . . .

Summary (Explanations)

- **Explanations help users understand outcomes of computations and data analysis**
  - typically meant for end users (understandability is critical)
  - often high-level
  - may be based on provenance, but not necessarily

Provenance & Explanations - Introduction

Provenance & Explanations - Introduction

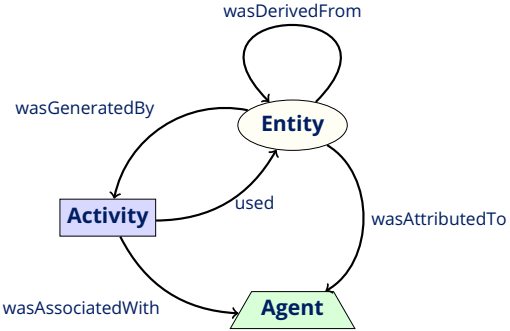
- Motivation & Use Cases
- Provenance & The W3C Prov Standard
- Recap

The W3 Prov Standard

- **Standard for representing and sharing provenance information**
  - https://www.w3.org/TR/prov-overview/
  - Extensible
  - multiple predefined serialization formats, e.g., JSON, XML, ...
- **Tooling & tutorials**
  - tutorial: https://github.com/lucmoreau/ProvToolbox
    - tutorial in ProvStore: https://openprovenance.org/store/documents/303
  - check consistency of PROV doc: https://github.com/pgroth/prov-check (online version: https://openprovenance.org/service/validator.html)
  - serializations and transformations: https://github.com/lucmoreau/ProvToolbox (online version: https://openprovenance.org/service/translator.html)
- **Public Prov Documents**
  - https://openprovenance.org/store/

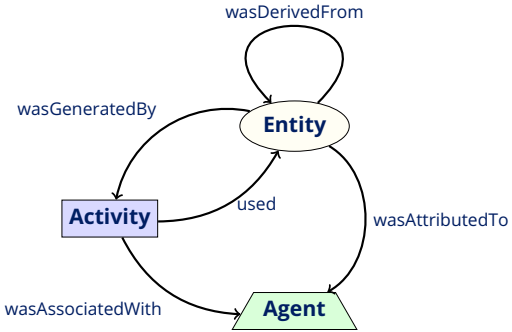
Prov - Nodes & Edge Types

- **Node types**
  - **Entities**
  - **Activities**
  - **Agents**
- **Edge types**
  - **wasDerivedFrom** (entity - entity)
    - data dependencies
  - **wasGeneratedBy** (entity - activity)
    - Activity outputs
  - **used** (activity - entity)
    - Activity inputs



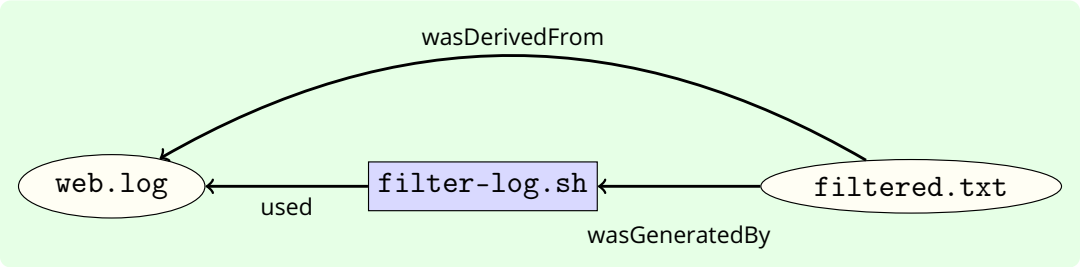
Additional Edge Types

- **Edge types**
  - **wasAttributedTo** (entity - agent)
    - the agent facilitated the creation of the entity or owns the entity
  - **wasAssociatedWith** (activity - agent)
    - the agent trigger / controlled / or executed the activity



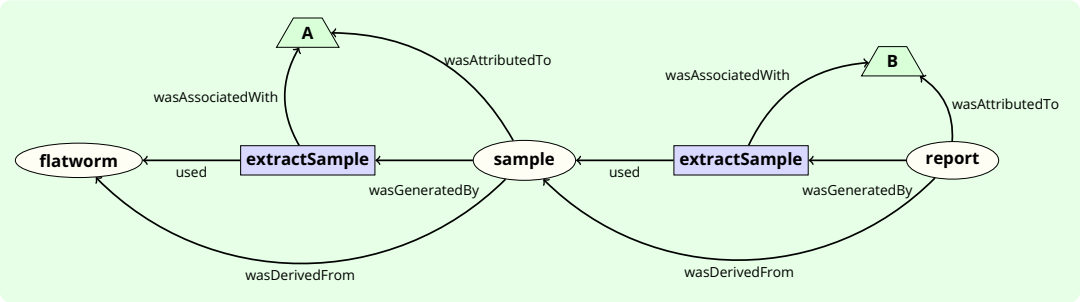
Prov Example - Log Filtering

- bash script `filter-log.sh` read log file `web.log` and writes out log entries (lines) that contain the string `apache`



Prov Example - Analyzing a Biological Sample

- Student `A` did prepare a tissue sample from a flatworm. Student `B` did test for the presence of a particular molecule in the sample.



Prov Serializations

- **The prov standard defines multiple serializations of Prov graphs including**
  - prov-n (prov notation - a human readable notation) - `https://www.w3.org/TR/2013/REC-prov-n-20130430/`
  - prov-json `https://www.w3.org/submissions/2013/01/`
  - prov-O (RDF turtle encoding) `https://www.w3.org/TR/2013/REC-prov-o-20130430/`
  - prov-xml - `https://www.w3.org/TR/2013/NOTE-prov-xml-20130430/`

prov-n Notation

- uses a **functional notation**, e.g., `entity(e1)`
- **identifiers** of entries are always the first argument
  - use namespaces that are declared in a document to map to a URL
  - use the format `namespace:id`
  - ids are requires for entities, activities, and agents
- each entry type declares **required** and **optional arguments**
  - some entry types allow a list of key value pairs as the last argument
  - optional ids are separated from the remaining arguments by `;` not `,`
- **documents** are enclosed by `document` and `endDocument`

prov-n Important Entry Types

- **entities** - `entity(identifier, optionalAttributeValuePairs)`

```
entity(ex:declarationOfIndependence, [ prov:type="document" ])
```

- **activity** - `activity( identifier (timeOrMarker timeOrMarker)? optionalAttributeValuePairs)`

```
activity(ex:log-fitering)
activity(ex:log-fitering, -, -, [prov:type="filter"])
activity(ex:a10, 2011-11-16T16:00:00, 2011-11-16T17:00:00)
```

- **agent** - `agent(identifier optionalAttributeValuePairs)`

```
agent(ex:ag, [ prov:type='prov:Person', ex:name="David" ])
agent(ex:ag, [ prov:type='prov:Person', ex:name="David" ])
```

prov-n Important Entry Types II

- **used** - `used(optionalIdentifier (eIdentifierOrMarker timeOrMarker)? optionalAttributeValuePairs)`

```
used(ex:a, ex:e, -)
used(ex:a, ex:e, 2011-11-16T16:00:00)
```

- **wasDerivedFrom** - `wasDerivedFrom(optionalIdentifier eIdentifier eIdentifier (aIdentifierOrMarker gIdentifierOrMarker uIdentifierOrMarker)? optionalAttributeValuePairs)`

```
wasDerivedFrom(ex:e2, ex:e1, -, -, -)
```

- **wasGeneratedBy** - `wasGeneratedBy( optionalIdentifier (aIdentifierOrMarker timeOrMarker)? optionalAttributeValuePairs)`

```
wasGeneratedBy(ex:e2, ex:a1, -)
wasGeneratedBy(ex:e2, ex:a1, [ex:fct="save"])
```

prov-n Example

- Reconsider the log file filtering example

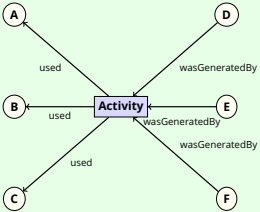
```
document
  prefix var <http://openprovenance.org/var#>

  entity(var:log)
  entity(var:filteredlog)
  activity(var:filter-script)
  used(var:filter-script,var:log,-)
  wasGeneratedBy(var:filteredlog,var:filter-script,-)
  wasDerivedFrom(var:filteredlog,var:log,-,-,-)
endDocument
```

Granularity

- Important design choice: at which granularity of entities and activities do we need to track provenance
- **Entities**
  - CSV file, row, field, individual character
  - Database, table, row, cell
  - Distributed file system, directory, file, . . .
- **Activities**
  - Transaction, SQL statement, relational operator
  - Distributed simulation, workload on one machine in the cluster, individual task
  - Publishing a research paper, individual phases (data collection, analysis, writing, peer-review process), . . .

Data Dependencies

- Not all **entities** generated by an **activity** are necessarily **derived from** all **entities** used by the **activity**
- Need more detailed information about the activity to understand which `wasDerivedFrom` relationships hold



Collections

- Entities may be part of collections
  - A databases entity contains table entities, a table entity contains row entities, a row entity contains cell entities
  - A document entity contains paragraphs which in turn contain sentence which contain words

```
document
  prefix ex <http://example.org>
  entity(ex:table, [ prov:type='prov:Collection' ])
  entity(ex:row1)
  entity(ex:row2)
  hasMember(ex:table, ex:row1)
  hasMember(ex:table, ex:row2)
endDocument
```

Provenance & Explanations - Introduction

Provenance & Explanations - Introduction

- Motivation & Use Cases
- Provenance & The W3C Prov Standard
- Recap

Recap

- **Provenance** - information about the origin and creation process of data
  - **W3C PROV** - general model for representing and storing provenance with several serializations
  - **Entities**, **Activities**, **Agents**
  - **data dependencies**
  - **granularity** of entities and activities
- **Explanations** - explaining computations for end users, often high-level
  - black-box or white-box
  - often involves **summarization**
  - often utilizes **attribution** metrics, e.g., Shapley values