Logic Programming for Data Tagging

Stephen Chong
Privacy Tools Project NSF Site Visit
Monday October 19

with Alexandra Wood
Berkman

Micah Altman
MIT

Kevin Wang
Harvard Undergrad

Aaron Bembenek
Harvard Grad & Privacy Tools Summer Intern 2015
Logic Programming for Data Tagging

Knowledge Acquisition
- Legal Scholars
- IRB

Knowledge Codification
- Logic Programming/Legal Model
- Decision-Tree Language Generator

Data Ingestion
- DataTags Interview Tool
- DUA Generator Tool

Secure Infrastructure
- Blue
- Green
- Yellow
- Orange
- Red
- Crimson

Data Retrieval
- Privacy Budget Tool
- Privacy Analysis Tool
- Access Control

End-to-End Systems

Technology Science
Dataverse
Logic Programming for Data Tagging

- **Formal legal model** for tag recommendation and license generation
- **Optimization** of DataTags Questionnaires
- Datalog implementation
Tag Recommendations and Licenses

Use and share data

Data Tags

Social Scientist

This Work

Tag characterization

Formal model of regulatory requirements

Permitted/denied actions

Analysis in legal memos

Statutory text

Privacy Legislation

Stephen Chong, Harvard University.
Tag Recommendations and Licenses

Motivation and context
- Formalize aspects of privacy legislation
  - Using a logic programming language
- Answer whether legislation/best practice permits or denies specific actions on data sets
  - Expert-system-like ability
- Explore legislation
  - e.g., find conditions where best practice contradictory

Combines
- computer science (formal modeling),
- law (legal research & analysis),
- social science (survey design),
- information science (taxonomies)
System design

This work

Question text
License text
Logic rules

Logic program

Recommended privacy attributes
Recommended Tags
Recommended license text

Data deposit
Transformations
Access Control

Data deposit
Deposit license

Data Owner

Data User
Formal model: Actions

\( dd : \text{Data depositor} \)

Deposit(\( dd, \ ds, \ r, \ cs \))

Accept(\( r, \ ds, \ dd, \ cs \))

Release(\( r, \ ds, \ du, \ dd, \ cs \))

\( ds : \text{Dataset} \)

\( du : \text{Data user} \)

\( cs : \text{Condition set} \) (provides further details about action)

Stephen Chong, Harvard University.
Permitted or Denied

Actions can be permitted or denied

\[
\text{Permitted}(\text{leg}, \ a) \\
\text{Denied}(\text{leg}, \ a)
\]

- Or neither permitted or denied
- E.g., \( \text{Denied}(\text{ferpa}, \ \text{Release}(\text{harvardDataverse}, \ \text{cs152grades-2015sp}, \ \text{jon@doe.com}, \ \text{chong@seas.harvard.edu}, \ [\text{dataverseClickthrough}])) \)
Example formalization

Let \( dd \) be the data depositor
Let \( du \) be the data user
Let \( ds \) be the data set
Let \( r \) be the repository
Let \( cs \) be a set of conditions

IF \( \text{CMR:depositorInScope}(dd, ds) \)
AND \( \text{CMR:identifiable}(ds) \)
AND NOT (\( \text{CMR:secure}(r) \)
AND \( \text{CMR:isAcceptableConditionsForRelease}(cs) \))
THEN \( \text{DENIED(Release}(r, ds, du, dd, cs)) \)

Let \( l \) be a license
Let \( cs \) be a set of conditions
IF \( \text{License}(l) \in cs \)
AND \( \text{licenseImplies}(l, \text{CMR:TransmissionEncrypted}) \)
THEN \( \text{CMR:isAcceptableConditionsForRelease}(cs) \)
Formalization process

**Define Use Cases In Scope**
- Data Deposit
- Retention
- Transformation
- Dissemination

**Parse Use Case Components**
- Actors: data controller, repository, user
- Actions: accept data, store, release,...
- Entities: Data set, Record,...

**Legal Review of Law**
- Identify Restrictions on Use Cases
- Identify Restrictions on Actors, Actions, Entities

**Expert Coding of Use Case Actions**
- Map rules permitting and restricting action to law-specific characteristics
- Map law-specific characteristics to license text affirmations
- Map law-specific characteristics to general properties
Demo
DataTags

Permitted and denied actions are the interface between DataTags and legislation

- May require more powerful language than Prolog...

Let $dd$ be the data depositor
Let $du$ be the data user
Let $ds$ be the data set
Let $r$ be the repository
Let $t$ be the data tag

IF $\text{isDataTag}(t, r, ds)$
AND FOR ALL condition sets $cs$
    PERMITTED(Release($r, ds, du, dd, cs$))
    IMPLIES conditionsRequire($cs, \text{ReidentificationProhibited}$)
THEN atLeast($t, \text{Yellow}$)
Logic Programming for Data Tagging

**Formal legal model for tag recommendation and license generation**

**Datalog implementation**

**Optimization of DataTags Questionnaires**
DataTags Questionnaire in Datalog

- When accepting dataset, ask depositor series of questions to determine DataTag
- Currently: nice domain specific language
  - But imperative with explicit control flow (i.e., `gos`)
- Goal: express more **declaratively** using Datalog
  - Separate questions from control flow
  - Facilitate composition of questionnaires
  - Re-run old answers when questionnaire changes
  - ...

Stephen Chong, Harvard University.
“Optimize” question order

- Given declarative questionnaire, what is "best" order to ask questions?
  - Fewest questions to reach decision?
  - Ask questions from general to specific?
  - Ask related questions at same time?
- Assume some cost function for the question order
- Characterize as a game
  - Player asks question, opponent gives answer
  - Player's goal: reach decision with lowest cost
  - Determine strategy with lowest (expected) cost
- Game tree too big to explore exhaustively
  - E.g., with $n$ questions, 3 answers per question, there are $n! \times 3^n$ paths/final states
  - But analysis of Datalog program can significantly reduce search
Logic Programming for Data Tagging

Formal legal model for tag recommendation and license generation

Datalog implementation

Optimization of DataTags Questionnaires

Knowledge Acquisition
- Legal Scholars
- IRB

Knowledge Codification
- Logic Programming/Legal Model
- Decision-Tree Language Generator

Data Ingestion
- DataTags Interview Tool
- DUA Generator Tool

Secure Infrastructure
- Blue/Green/Yellow/Orange/Red/Crimson

Data Retrieval
- Privacy Budget Tool
- Privacy Analysis Tool

Access Control

End-to-End Systems
- Dataverse
- Dataverse
Our very own Datalog...

- Developed our own Datalog implementation
  - Can extend with language features
  - More flexible interface/efficient interaction
  - Make use of modern concurrent hardware
  - Will be used in Harvard undergrad PL course
  - ...

- Will be released open source
Current state

- Current state:
  - Six evaluation engines
    - Top-down, bottom up, concurrent bottom up, ...
    - Exploring different concurrent techniques to improve scalability
      - Preliminary results: 1.2–5.5× speedup over XSB Prolog on OpenRuleBench transitive closure tests
  - Implemented hypotheticals
  - Graphical user interface
    - Suitable for use by undergrad class!
Moving forward

- Formal legal model
  - License generation (from required conditions)
  - Review/independent validation of rules and license text
  - Independent validation of formalization process
  - Engagement with practitioners
    - IRBs, state and local govt. agencies, educational data controllers, ...
- Questionnaire representation and optimization
- Datalog
  - Release and use
  - Develop right logical extension for, e.g., connecting to DataTags
Questions?

- **Formal legal model** for tag recommendation and license generation
- **Datalog implementation**
- **Optimization of DataTags Questionnaires**