Privacy-Preserving Distributed Queries
for a Clinical Case Research Network

Gunther Schadow, Regenstrief Institute

Overview

- Objectives, Use Cases
- Architectural Assumptions
- Privacy Protecting Distributed Joins
- Special issues with record linkage
- Discussion
Objective

- To support medical researchers locating appropriate study “material”
- by querying a large loosely coupled network of various medical data bases,
- while maintaining reasonable patient privacy in the querying process and its results.

Medical Research Studies

- Retrospective Cohort Studies
  - find cohorts of exposed and control subjects, link each with outcome.
- Case–Control Studies
  - find outcomes (study and control) and link each with data on exposure.
- Cross–Sectional Studies
  - find cases and look for common features.
- Prospective Studies
  - requires contact with individual patients.
Kinds of study “material”

- Cases (medical information) for retrospective study.
- Tissue samples related to certain kinds cases for tissue examinations.
- Potentially: human subjects for inclusion in interventional studies.

Locating study material, present

- Chart review – manually scan through paper charts.
  - still very common practice (tedious)
- Isolated databases / warehouses
  - may not contain all data needed (outpatient visits, prescriptions)
- Shared databases with compilations of case abstracts.
  - only contain select data elements
... and proposed future

- A loosely coupled (“federated”) distributed multi-database.
  - Data remains at the location of origin.
  - Dynamically joined for each query.
- But how can we do distributed joins and still avoid revealing patient identifiers?

Architecture, Assumptions

- Simple Data Schema
  - One simple relation: \( R(p, e, t, v) \)
  - patient identifier \( (p, \) abstract\)
  - event code \( (e) \)
  - time of event \( (t) \)
  - value of event \( (v) \)

<table>
<thead>
<tr>
<th>patient id</th>
<th>time</th>
<th>event</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jimmy</td>
<td>1999–01–10</td>
<td>birth</td>
<td></td>
</tr>
<tr>
<td>Jimmy</td>
<td>1999–01–17</td>
<td>prescription</td>
<td>erythromycin</td>
</tr>
<tr>
<td>Jimmy</td>
<td>1999–03–07</td>
<td>diagnosis</td>
<td>pyloric stenosis</td>
</tr>
<tr>
<td>Carly</td>
<td>1998–09–21</td>
<td>birth</td>
<td></td>
</tr>
<tr>
<td>Carly</td>
<td>1998–12–24</td>
<td>procedure</td>
<td>pylorotomy</td>
</tr>
<tr>
<td>Carly</td>
<td>1999–08–15</td>
<td>diagnosis</td>
<td>neuroblastoma</td>
</tr>
</tbody>
</table>
Data distribution

- Diagnosis and surgery from a hospital.
- Prescription information from outpatient pharmacies.
- Birth and death records from public records.
- Special case information from cancer registries, etc.

Distributed Join Queries

- Select query:
  - pass the criterion and receive all matching ids, then intersect with ids you already have.
- Semi-join:
  - pass the criterion plus a set of ids, then receive all ids from that set that match the criterion.
- Bloom-join:
  - semi-join where the set of ids passed is a set of hash values, Bloom-filter.
Distributed join and privacy

- Common surrogate keys do not exist in loosely-coupled systems.
- Join keys must be real identifiers
  - name
  - date of birth
  - social security number
- Conventional distributed join protocols would effectively broadcast these identifiers.
Hashing for privacy

- Protecting identifiers through keyed hashing (HMAC)
  - $h_k(p) = h(h(p \circ k) \circ k)$
  - one-way operation
  - pseudo-random
  - uniformly distributed
  - $(p \equiv q) \Rightarrow (h_k(p) = h_k(q))$
- Protects privacy from the Mediator
  - If the mediator is kept from knowing the key (ensured by policy, organization).

Vulnerabilities of hashing

- Dictionary attacks
  - Attacker finds known patients of interest in semi-join filters by hashing the identifiers he knows.
  - Easy for a data source, since key is shared by all data sources.
  - Hashing alone is not safe.
  - Protect privacy from data sources by making ambiguous.
Hash–collisions for privacy

- Number of individuals $N \approx 10^9$
- 128 bit HMAC
  - $10^{36}$ codes, practically 1:1
- HMAC truncated to any length
  - exploiting uniform distribution and pseudo–randomness
- False positive probability of an HMAC match:
  $$P(h^b \in F | q \notin R) = 1 - (1 - 1/2^b)^m$$

Simple Bayesian privacy model

- Posterior probability for a person $q$ to have a condition $C$ when $h(q)$ is in the semi–join filter $F$ for $C$.
  $$P(C | h \in F) = \frac{P(h \in F | C) P(C)}{P(h \in F | C) P(C) + P(h \in F | \overline{C}) P(\overline{C})}$$
Likelihood of inference

- In odds / likelihood ratio form
  \[ O(C \mid h \in F) = \frac{P(h \in F \mid C)}{P(h \in F \mid \overline{C})} O(C) = L \cdot O(C) \]
- Worst case assumption
  \[ L = \frac{1}{P(h \in F \mid \overline{C})} = \frac{1}{1-(1-1/n)^m} \]

Diagnostic likelihood ratios

- Common prior probability
  - \( P(\text{HIV}) = 0.006 \)
  - \( P(\text{cancer}) = 0.03 \)
- Likelihood ratios:
  - 1  no information
  - 1–2 minor increase
  - 2–5 small increase
  - 5–10 moderate increase
  - >10 large increase, often conclusive
Practical interpretation of Likelihood ratio

- Linear amplification of prior probability

Adjusting likelihood ratios

- hash function range $n$ for $L$
  \[ n = (1 - (1 - 1/L)^{1/m})^{-1} \]

<table>
<thead>
<tr>
<th>$L$</th>
<th>$m$</th>
<th>$n$ [1]</th>
<th>$b$ [bit]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$10^5$</td>
<td>$2.5 \times 10^5$</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>$10^5$</td>
<td>$10^6$</td>
<td>20</td>
</tr>
<tr>
<td>50</td>
<td>$10^5$</td>
<td>$5 \times 10^6$</td>
<td>23</td>
</tr>
</tbody>
</table>

- false-positive retrieval rate
  \[ P(h(q) \in F | q \not\in R) = 1/L \]
Real identifiers as join-keys

- Real identifiers can be wrong or incomplete.
- Links that should be made are not made ("false negatives")
- Vector of identifier components.
- Matching relation \( \Rightarrow \)

Record Linkage

- Heuristic linkage
  - also known as "deterministic":
  - guess a set of identifiers,
  - guess matching rules
  - statistically test overall performance
  - typically two outcomes
  - quite commonly used

- Probabilistic linkage
  - Fellegi and Sunter (1969)
  - guess a set of identifiers,
  - guess comparison operation
  - assess performance of each component
  - typically three outcomes based on likelihood score
Example Heuristic Rule

- Using the following data
  - social security number (SSN)
  - first name (FN), last name (LN)
  - birth year (YB), month (MB), day (DB)
  - phonetic code of first name (cFN)
- One of the following sets must match completely.
  1.) SSN, cFN, YB;
  2.) SSN, cFN, MB;
  3.) SSN, cFN, DB; and
  4.) LN, FN, YB, MB, DB;

Privacy for $k$ components

- False-positives for $k$ hash codes
  \[ P(F) = 1 - (1 - 1/n)^{km} \]
- Likelihood ratio $L$ for $P(C \mid f \in F)$
  \[ L = (1 - (1 - 1/n)^{km})^{-1} \]
- hash function range $n$ for $L$
  \[ n = (1 - (1 - 1/L)^{1/km})^{-1} \]
- false-positive retrieval rate is still
  \[ P(\bigvee_i h(q_i) \in F \mid q \notin R) = 1/L \]
Likelihood ratios for $k$ components

<table>
<thead>
<tr>
<th>$L$</th>
<th>$m$</th>
<th>$n [1]$</th>
<th>$b$ [bit]</th>
<th>$k = 1$</th>
<th>$k = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$10^5$</td>
<td>$2.5 \times 10^5$</td>
<td>18</td>
<td>$10^6$</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>$10^5$</td>
<td>$10^6$</td>
<td>20</td>
<td>$4 \times 10^6$</td>
<td>22</td>
</tr>
<tr>
<td>50</td>
<td>$10^5$</td>
<td>$5 \times 10^6$</td>
<td>23</td>
<td>$2 \times 10^7$</td>
<td>25</td>
</tr>
</tbody>
</table>

Privacy for $k$ components

- The “intruder” can require that more than one ($\alpha$) identifier combinations match, giving a likelihood ratio
  $$L(C | \land \alpha h(q_i) \in F) = (1 - (1 - 1/n)^{km})^\alpha$$

- The intruder therefore can get a very good likelihood ratio.
Privacy for multiple identifiers

- Semi-joins with disjunctive identifier vectors gives too much of an advantage to the intruder.
- Can we find a single identifier code?
- Loss of sensitivity is a great problem!

Discussion: Fellegi–Sunter

- Comparison vector $\gamma (p, q)$

\[
\frac{P(\gamma (p, q) | p=q)}{P(\gamma (p, q) | p\neq q)} = \frac{m(\gamma)}{u(\gamma)}
\]

- Two thresholds $T_\mu, T_\lambda$
  - $\gamma > T_\mu$: assume match
  - $\gamma < T_\lambda$: assume non-match
  - $T_\mu \geq \gamma \geq T_\lambda$: undetermined (review)
Fellegi–Sunter

- Comparison vector \( \gamma(p, q) \) is not restricted in any way.
  - “deterministic” linkage is a special case
- Commonly the components of \( \gamma \) correspond to the components of the identifier vectors.
- Independence of components of \( \gamma \) is important for the common simplification:

\[
w(\gamma) = \sum w(\gamma_i)
\]

Fellegi–Sunter

- Independent identifier vector components are nice, but
- render components vulnerable to frequency attacks;
- lose uniform distribution of hash values
Outlook

- For semi-join filter, reduce number of rules
- Merge rules 1–3
  - dropping the birth date component
  - only affects specificity
- Consider dropping rule 4
  - and lose up to 30% of true matches

Conclusion

- Without a surrogate key that has good retrieval properties, privacy protecting semi-join filters are hard to accomplish.
- Policy and network organization and a variable trust model where privacy protection can be modulated for each data source seem necessary.
Thank you!