Privacy-Preserving Big Data Publishing

Hessam Zakerzadeh\textsuperscript{1}, Charu C. Aggarwal\textsuperscript{2}, Ken Barker\textsuperscript{1}

SSDBM’15

\textsuperscript{1} University of Calgary, Canada
\textsuperscript{2} IBM TJ Watson, USA
Data Publishing

- OECD* declaration on access to research data
- Policy in Canadian Institutes of Health Research (CIHR)

* - Organization for Economic Co-operation and Development

Data Publishing

Benefits:

• Facilitating the research community to confirm published results.
• Ensuring the availability of original data for meta-analysis.
• Making data available for instruction and education.

Requirement:

• **Privacy** of individuals whose data is included must be preserved.
Tug of War

Preserving the privacy of individuals whose data is included (needs anonymization).

Usefulness (utility) of the published data.

Ref: http://www.picgifs.com
Key Question in Data Publishing

*How to preserve the privacy of individuals while publishing data of high utility?*
Privacy Models

Privacy-preserving models:

– Interactive setting (e.g. differential privacy)
– **Non-interactive** setting (e.g. k-anonymity, l-diversity)
  
  • Randomization
  • **Generalization**
K-Anonymity

Attributes

• Identifiers
• Quasi-identifier
• Sensitive
K-Anonymity

Table 1: A patient data set
(a) Original data

<table>
<thead>
<tr>
<th>t#</th>
<th>SIN</th>
<th>Gender</th>
<th>Zipcode</th>
<th>DOB</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12543222</td>
<td>M</td>
<td>12499</td>
<td>30 June, 1985</td>
<td>HIV</td>
</tr>
<tr>
<td>2</td>
<td>23988880</td>
<td>M</td>
<td>12423</td>
<td>1 June, 1986</td>
<td>flu</td>
</tr>
<tr>
<td>3</td>
<td>34340012</td>
<td>M</td>
<td>13001</td>
<td>12 Sept, 1982</td>
<td>flu</td>
</tr>
<tr>
<td>4</td>
<td>12001234</td>
<td>M</td>
<td>13078</td>
<td>17 Sept, 1982</td>
<td>gastritis</td>
</tr>
<tr>
<td>5</td>
<td>33336788</td>
<td>F</td>
<td>13223</td>
<td>22 Aug, 1971</td>
<td>miocarditis</td>
</tr>
<tr>
<td>6</td>
<td>33789901</td>
<td>F</td>
<td>13009</td>
<td>2 Nov, 1974</td>
<td>miocarditis</td>
</tr>
</tbody>
</table>

(b) 2-anonymized data

<table>
<thead>
<tr>
<th>t#</th>
<th>Gender</th>
<th>Zipcode</th>
<th>DOB</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>124**</td>
<td>June [1985-1986]</td>
<td>HIV</td>
</tr>
<tr>
<td>2</td>
<td>M</td>
<td>124**</td>
<td>June [1985-1986]</td>
<td>flu</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>130**</td>
<td>Sept 1982</td>
<td>flu</td>
</tr>
<tr>
<td>4</td>
<td>M</td>
<td>130**</td>
<td>Sept 1982</td>
<td>gastritis</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>13***</td>
<td>[1971-1974]</td>
<td>miocarditis</td>
</tr>
</tbody>
</table>
L-Diversity

Table 1: A patient data set

(a) Original data

<table>
<thead>
<tr>
<th>t#</th>
<th>SIN</th>
<th>Gender</th>
<th>Zipcode</th>
<th>DOB</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12543222</td>
<td>M</td>
<td>12499</td>
<td>30 June, 1985</td>
<td>HIV</td>
</tr>
<tr>
<td>2</td>
<td>23988880</td>
<td>M</td>
<td>12423</td>
<td>1 June, 1986</td>
<td>flu</td>
</tr>
<tr>
<td>3</td>
<td>34340012</td>
<td>M</td>
<td>13001</td>
<td>12 Sept, 1982</td>
<td>flu</td>
</tr>
<tr>
<td>4</td>
<td>12001234</td>
<td>M</td>
<td>13078</td>
<td>17 Sept, 1982</td>
<td>gastritis</td>
</tr>
<tr>
<td>5</td>
<td>33336788</td>
<td>F</td>
<td>13223</td>
<td>22 Aug, 1971</td>
<td>miocarditis</td>
</tr>
<tr>
<td>6</td>
<td>33789901</td>
<td>F</td>
<td>13009</td>
<td>2 Nov, 1974</td>
<td>miocarditis</td>
</tr>
</tbody>
</table>

(c) 2-diverse data

<table>
<thead>
<tr>
<th>t#</th>
<th>Gender</th>
<th>Zipcode</th>
<th>DOB</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>1****</td>
<td>1982-1986</td>
<td>HIV</td>
</tr>
<tr>
<td>2</td>
<td>M</td>
<td>1****</td>
<td>1982-1986</td>
<td>flu</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>1****</td>
<td>1982-1986</td>
<td>flu</td>
</tr>
<tr>
<td>4</td>
<td>*</td>
<td>13***</td>
<td>1971-1982</td>
<td>gastritis</td>
</tr>
<tr>
<td>5</td>
<td>*</td>
<td>13***</td>
<td>1971-1982</td>
<td>miocarditis</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
<td>13***</td>
<td>1971-1982</td>
<td>miocarditis</td>
</tr>
</tbody>
</table>
Assumptions in state-of-the-art of anonymization

Implicit assumptions in current anonymization:
• Small- to moderate-size data
• Batch and one-time process

Focus on
• Quality of the published data
Still valid?

New assumptions:

• Small- to moderate-size data
  vs. Big data (in Tera or Peta bytes)

• e.g. health data, or web search logs
  vs. Repeated application

• Batch and one-time process
  vs. Scalability

Focus on

• Quality of the published data + Scalability
Naïve solution?

Divide & conquer

- Inspired by streaming data anonymization techniques
- Divide the big data into small parts (fragments)
- Anonymize each part (fragment) individually and in isolation
Naïve solution?

Divide & conquer

- Inspired by streaming data anonymization techniques
- Divide the big data into small parts (fragments)
- Anonymize each part (fragment) individually and in isolation

- What we lose?
  - **Rule of thumb**: more data, less generalization/perturbation (large crowd effect)
  - Quality
Main question to answer in Big Data Privacy

Is it somehow possible to take advantage of the entire data set in the anonymization process without losing scalability?
Map-Reduce-Based Anonymization

**Idea:** distribute the high-computational process of anonymization among different processing nodes such that distribution **does not affect** the quality (utility) of the anonymized data.
Map-Reduce Paradigm

Data flow of a mapreduce job
Mondrian-like Map-Reduce Alg.

Traditional Mondrian

- Pick a dimension (e.g. dim with widest range), called **cut dimension**
- Pick a point along the cut dimension, called **cut point**
- Split data into two equivalence classes along the **cut dimension** and at the **cut point**, provided that privacy condition is not violated.
- Repeat until no further split is possible
Mondrian-like Map-Reduce Alg.

Traditional Mondrian

![Diagram of Mondrian-like Map-Reduce Alg.]

Privacy-Preserving Big Data Publishing
Mondrian-like Map-Reduce Alg.

Traditional Mondrian

![Diagram of Mondrian-like Map-Reduce Alg.](image)
Mondrian-like Map-Reduce Alg.

Traditional Mondrian

![Diagram of Mondrian-like Map-Reduce Alg.]

Privacy-Preserving Big Data Publishing
Mondrian-like Map-Reduce Alg.

Preliminaries:

• Each equivalence class is divided into at most $q$ equivalence classes.
• A global file is shared among all nodes. This file contains equivalence classes formed so far organized in a tree structure (called equivalence classes tree).
• Initially contains the most general equivalence class.
Mapper

Algorithm 1 Mapper in $k$-anonymity

1: $KMapper(k, v)$
2:   //obtain the global file from distributed file system
3:   $eq-id = findFinestEQ(v)$
4:   $output-value = empty$ // an array of size dimensionality
5:   foreach $dim$ in $v$
6:     append pair ($dim, 1$) to $output-value$
7:   emit($eq-id, output-value$)
Mapper: Example

iteration 1: only one equivalence class exists (called eq₁)

\[ eq₁ = [1:100],[1:100],[1:100] \]

Data records:

- 12, 33, 5
- 56, 33, 11
- 12, 99, 5

Mapper’s output:

- \(< \text{eq₁}, <(12,1),(33,1),(5,1)> >\>
- \(< \text{eq₁}, <(56,1),(33,1),(11,1)> >\>
- \(< \text{eq₁}, <(12,1),(99,1),(5,1)> >\>
Combiner

Algorithm 2 Combiner in $k$-anonymity

1: $KCombiner(k, V)$
2: $output-value = empty$ // an array of size dimensionality
3: foreach $v$ in $V$
4:     for $i$ in $[1, ..., dimensionality]$
5:         add($v[i], output-value[i])$
6:     emit($k, output-value$)
Combiner example

Mapper’s output (combiner’s input):

\[< \text{eq}_1, <(12,1),(33,1),(5,1)>>\]
\[< \text{eq}_1, <(56,1),(33,1),(11,1)>>\]
\[< \text{eq}_1, <(12,1),(99,1),(5,1)>>\]

Combiner’s output:

\[< \text{eq}_1, \begin{array}{cc} 12 & 2 \\ 33 & 2 \\ 5 & 2 \\ 56 & 1 \\ 99 & 1 \\ 11 & 1 \end{array}>\]
Reducer

Algorithm 3 Reducer in $k$-anonymity

1: $KReduction(k, V)$
2: // all the data records belong to equivalence class $eq$
3: // $q$ is the maximum number of split
4: for ($i$ in $[1, \ldots, \text{dimensionality}]$)
5: \hspace{1em} $c$-$dim$ = findDimToCut($V, i$)
6: for ($p$ in $[q, \ldots, 2]$)
7: \hspace{2em} $cp_1, cp_2, \ldots, cp_{p-1} = \text{findCutPoint}_p(V, c$-$dim$)
8: if (splitting $eq$ at $cp_1, cp_2, \ldots, cp_{p-1}$ does not violate
9: \hspace{2em} the $k$-anonymity)
10: \hspace{3em} $eq_1, eq_2, \ldots, eq_{p-1} = \text{Cut}_p(eq, c$-$dim, cp_1, \ldots, cp_{p-1})$
11: for ($j$ in $[1, \ldots, p - 1]$)
12: \hspace{4em} emit($eq_j$, “1”)
13: exit()
14: emit($eq$, “0”)

Reducer’s output

Combiner’s output (reducer’s input):
< eq₁, 12 2, 33 2, 5 2, 56 1, 99 1, 11 1 >

Reducer’s output:
- If eq₁ is splittable:
  - <[12:56],[33:45],[5:11], “1”> → Added to the global file
  - <[12:56],[45:99],[5:11], “1”> → Added to the global file
- If eq₁ is un-splittable:
  - <[12:56],[33:99],[5:11], “0”> → Added to the global file
Improvement 1 (Data transfer improvement)

• Mapper outputs only records belonging to **splittable** equivalence classes.

• Requirement:
  • Global file includes a flag for each equivalence class indicating whether it is **splittable** or not.
  • If a data record belongs to an **un-splittable** EQ, do not output it.
  • e.g. [1:100],[1:100],[1:100], “1”
    [12:56],[33:45],[5:11], “1”
    [12:56],[45:99],[5:11], “1”
    [12:30],[33:45],[5:11], “1”
    [30:56],[33:45],[5:11], “0”
Improvement 2 (Memory improvement)

• What if the global file doesn’t fit into memory of mapper nodes?
  • Break down the global file into multiple small files
  • How?
    • Split the big data into small files.
    • Create a global file per small input file.
    • Each global file contains only a subset of total equivalence classes.
    • Each small global file is referred to as global subset equivalence class file (gsec file).
Improved Algorithm

Mapper’s output:
\[
< eq_1, <f_1, (12,1),(33,1),(5,1)>> \\
< eq_1, <f_1, (56,1),(33,1),(11,1)>> \\
< eq_1, <f_1, (12,1),(99,1),(5,1)>>
\]

Reducer’s output:
If eq_1 is splittable:
\[
<[12:56],[33:45],[5:11], “1”, f_1 > \\
<[12:56],[45:99],[5:11], “1”, f_1>
\]
If eq_1 is un-splittable:
\[
<[12:56],[33:99],[5:11], “0”, f_1>
\]

Combiner’s output:
\[
< eq_1, <f_1, \\
12 | 2 \\
56 | 1 \\
33 | 2 \\
99 | 1 \\
56 | 2 \\
11 | 1 \\
> 
\]
$K,q = 2$
Further Analysis

• Time Complexity (per round)
  • Mapper
  • Combiner
  • Reducer

• Data Transfer (per round)
  • Mapper and Combiner (Local)
  • Combiner and Reducer (Across the network)
Experiments

Answer the following questions:

• How well the algorithm scales up?
• How much information is lost in the anonymization process?
• How much data is transferred between mappers/combiners and combiners/reducers in each iteration?
Experiments

Data sets

**Poker data set**: 1M records, each 11 dimensions

**Synthetic data set**: 10M records, each 15 dimensions, 1.4 GB

**Synthetic data set**: 100M records, each 15 dimensions, 14.5 GB

Information loss baseline

Each data set is split into 8 fragments. Each fragment is anonymized individually

State-Of-The-Art

MapReduce Top-Down Specialization (MRTDS) [Zhang et. al’14]
Experiments Settings

• Hadoop cluster on AceNet
• 32 nodes, each having 16 cores and 64 GB RAM
• Running RedHat Enterprise Linux 4.8
Information Loss vs. K
Information Loss vs. L

![Bar charts showing information loss for synthetic and poker datasets.](image)
Running time vs. K (L)
Data Transfer vs. Iteration # (between mappers and combiners)
Data Transfer vs. Iteration # (between combiners and reducers)
Future work

• Extension to other data types (graph data anonymization, set-value data anonymization, etc)
• Extension to other privacy models