Tutorial LV 185.A83

PAML

Privacy Aware Machine Learning

.. ML on perturbed knowledge bases ..

Bernd Malle

b.malle@hci-kdd.org

SBA Research / Holzinger Group, HCI-KDD
Overview / Contents

1. Introduction & Motivation
   • The right to be forgotten & it’s consequences

2. Perturbation vs. Anonymization

3. ML performance on perturbed data


5. Limits of anonymization (k, l, t)

6. (Some) Algorithmic Approaches
   • Greedy clustering
   • SaNGreeA

7. ML performance on anonymized data

8. Can iML help in anonymization?
Privacy in the 21st century...??

Data protection laws

Technological progress

Privacy
The right to be forgotten

- Basically: A user has the right to have their data deleted from a database upon request.

- In past cases, the requirement only meant deletion from a search index (due to EU tech ignorance).

- From 2018 onwards, the “right to be forgotten” will be part of the new EU data protection rules.

- Since one cannot foresee which (non-existing) laws will be enforced by the European bureaucracy in the future (see Apple..), it would be wise to be prepared...

- There is even a proposal by German data protection advocates to restrict automated processing of anonymized data which “might” be de-anonymizable...
Impact on different DB layers

- System failures?
- Incomplete statistics?
- OLTP service?
- Backups

- Less convenient
- ML performance…?
To whom it may concern...

Less convenient service?

Re-appearing items?

Incomplete statistics?

ML performance...?
Two different experimental scenarios

1. Simulate users exercising their “right to be forgotten” in the worst way possible – requesting the erasure of the most valuable data points in the knowledge base.

   In the future extendable to
   - Outlier deletion first (anomalous users have higher probability to request their data deleted)
   - Perturbation via addition of ‘targeted’ noise

2. Try to circumnavigate the re-creation of our ML databases by anonymizing them in the first place and applying our learning algorithms on that anonymized datasets.
Scenario One

implies

Selectively deleting (valuable) data points
Adult dataset original distribution

- capital-gain
- education-num
- marital-status

- relationship
- occupation
- hours-per-week

Bernd Malle <b.malle@hci-kdd.org>
1. Preprocess dataset

2. Train some logistic classifier on it

3. Retrieve the coefficients learned by the Log.Class.

4. Sort & use the best xyz as most valuable columns
Selective deletion – prepared datasets

- After extracting the 3 attribute values contributing the most information to the classifier
- We construct new datasets with 0.2, 0.4, 0.6, 0.8 and 1.0 fractions of those data rows missing
- Thereby constructing 15 new data sets
- To use 4 different classifiers on...
Selective deletion - Results

- F1 score dependent on perturbation, gradient boosting
- F1 score dependent on perturbation, linear SVC
- F1 score dependent on perturbation, logistic regression
- F1 score dependent on perturbation, random forest

[Graphs showing F1 score decrease with perturbation level for different features: Capital gain > 2k, Education num > 10, Marital status, civ spouse]
Scenario Two

implies

Wholesale anonymization of the knowledge base
Introduction & Motivation

• Public release of sensitive information is useful for
  • Statistics => education, grant proposals ;-)
  • Research => prediction of disease spreading etc.

• However, personal identities need to be concealed

• In the past, simple approaches have failed to provide sufficient security:
  • data linkage of publicly available datasets
  • Netflix database, which was linked with the IMDB movie ratings database (via date of rating) => at least one user was re-identified
Re-Identifying the NYC Taxi Ride Dataset

1. Find suspicious data
2. Figure out what ONE hash represents (‘0’)
3. Figure out input domain for hashes
   => Medallions are 4-5 digits
   => ~20M possibilities
4. Construct inverted LUT
5. DS hacked !!!

We need robust anonymization techniques
Properties & General Approach

Data properties => Reduce granularity

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Zip</th>
<th>Gender</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alex</td>
<td>25</td>
<td>41076</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

• Identifiers := immediately reveal identity
  • name, email, phone nr., SSN
  => DELETE

• Sensitive data
  • medical diagnosis, symptoms, drug intake, income
  => NECESSARY, KEEP

• Quasi-Identifiers := used in combination to retrieve identity
  • Age, zip, gender, race, profession, education
  => MAYBE USEFUL
  => MANIPULATE / GENERALIZE
**k-anonymity**: for every entry in the DS, there must be at least k-1 identical entries (w.r.t. QI's) => this is 3-anon:

<table>
<thead>
<tr>
<th>Node</th>
<th>Name</th>
<th>Age</th>
<th>Zip</th>
<th>Gender</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>Alex</td>
<td>25</td>
<td>41076</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>X2</td>
<td>Bob</td>
<td>25</td>
<td>41075</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>X3</td>
<td>Charlie</td>
<td>27</td>
<td>41076</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>X4</td>
<td>Dave</td>
<td>32</td>
<td>41099</td>
<td>Male</td>
<td>Diabetes</td>
</tr>
<tr>
<td>X5</td>
<td>Eva</td>
<td>27</td>
<td>41074</td>
<td>Female</td>
<td>Flu</td>
</tr>
<tr>
<td>X6</td>
<td>Dana</td>
<td>36</td>
<td>41099</td>
<td>Female</td>
<td>Gastritis</td>
</tr>
<tr>
<td>X7</td>
<td>George</td>
<td>30</td>
<td>41099</td>
<td>Male</td>
<td>Brain Tumor</td>
</tr>
<tr>
<td>X8</td>
<td>Lucas</td>
<td>28</td>
<td>41099</td>
<td>Male</td>
<td>Lung Cancer</td>
</tr>
<tr>
<td>X9</td>
<td>Laura</td>
<td>33</td>
<td>41075</td>
<td>Female</td>
<td>Alzheimer</td>
</tr>
</tbody>
</table>

There are 2 possible attacks on k-anonymity though...
1. Homogeneity attack:
   • all entries contain the same piece of sensitive information (Allergies)

<table>
<thead>
<tr>
<th>Node</th>
<th>Age</th>
<th>Zip</th>
<th>Gender</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>25-27</td>
<td>410*</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>X2</td>
<td>25-27</td>
<td>410*</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>X3</td>
<td>25-27</td>
<td>410*</td>
<td>Male</td>
<td>Allergies</td>
</tr>
</tbody>
</table>

2. Background knowledge attack:
   • Given two entries with identical QI sets: One has lung cancer, the other diabetes…

<table>
<thead>
<tr>
<th>Node</th>
<th>Age</th>
<th>Zip</th>
<th>Gender</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>X8</td>
<td>27-33</td>
<td>410**</td>
<td>*</td>
<td>Lung Cancer</td>
</tr>
<tr>
<td>X9</td>
<td>27-33</td>
<td>410**</td>
<td>*</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>
**I-diversity:** for every "equivalence class" of (at least k) QI-duplicates, there must be at least l different "well represented" values for the sensitive attribute

2 possible attacks:

1. **Skewness attack:**
   - cancer = positive 1% / negative 99%
   - Chances within group are 50%...

2. **Semantic closeness attack:** (similarity attack)
   - gastritis $\Leftrightarrow$ gastric ulcer ??

<table>
<thead>
<tr>
<th>Node</th>
<th>QI</th>
<th>Cancer</th>
<th>Drugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>*</td>
<td>Y</td>
<td>xyz...</td>
</tr>
<tr>
<td>X2</td>
<td>*</td>
<td>Y</td>
<td>xyz...</td>
</tr>
<tr>
<td>X3</td>
<td>*</td>
<td>Y</td>
<td>xyz...</td>
</tr>
<tr>
<td>X4</td>
<td>*</td>
<td>N</td>
<td>xyz...</td>
</tr>
<tr>
<td>X5</td>
<td>*</td>
<td>Y</td>
<td>xyz...</td>
</tr>
<tr>
<td>X6</td>
<td>*</td>
<td>Y</td>
<td>xyz...</td>
</tr>
<tr>
<td>X7</td>
<td>*</td>
<td>N</td>
<td>xyz...</td>
</tr>
<tr>
<td>X8</td>
<td>*</td>
<td>N</td>
<td>xyz...</td>
</tr>
<tr>
<td>X9</td>
<td>*</td>
<td>N</td>
<td>xyz...</td>
</tr>
</tbody>
</table>
t-closeness: an equivalence class has t-closeness if the intra-class distribution of a sensitive attribute differs no more than a threshold t from its global distribution (whole dataset). The whole DS has t-closeness if this holds for every equivalence class it contains.

basic idea:
• we do not want an attacker to gain too much insight (additional information) by looking at the data
• additional information => surprise (delta expectation)
• the closer our local and global distributions are => the less our local group deviates from expectations
Limits of anonymization

Trade-off between:

- Data utility => min. information loss
- Privacy => max. information loss

Both can be easily achieved (but not together 😊)

<table>
<thead>
<tr>
<th>Node</th>
<th>Name</th>
<th>Age</th>
<th>Zip</th>
<th>Gender</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>Alex</td>
<td>25</td>
<td>41076</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>X2</td>
<td>Bob</td>
<td>25</td>
<td>41075</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>X3</td>
<td>Charlie</td>
<td>27</td>
<td>41076</td>
<td>Male</td>
<td>Allergies</td>
</tr>
<tr>
<td>X4</td>
<td>Dave</td>
<td>32</td>
<td>41099</td>
<td>Male</td>
<td>Diabetes</td>
</tr>
<tr>
<td>X5</td>
<td>Eva</td>
<td>27</td>
<td>41074</td>
<td>Female</td>
<td>Flu</td>
</tr>
<tr>
<td>X6</td>
<td>Dana</td>
<td>36</td>
<td>41099</td>
<td>Female</td>
<td>Gastritis</td>
</tr>
<tr>
<td>X7</td>
<td>George</td>
<td>30</td>
<td>41099</td>
<td>Male</td>
<td>Brain Tumor</td>
</tr>
<tr>
<td>X8</td>
<td>Lucas</td>
<td>28</td>
<td>41099</td>
<td>Male</td>
<td>Lung Cancer</td>
</tr>
<tr>
<td>X9</td>
<td>Laura</td>
<td>33</td>
<td>41075</td>
<td>Female</td>
<td>Alzheimer</td>
</tr>
</tbody>
</table>
Different kinds of data input format

1. Microdata
   • data at the granularity of individuals (table row)

2. Graph data -> social network data, in which
   • nodes represent microdata
   • edges represent their structural context
   • graph data are harder to anonymize
     o It's harder to model the background knowledge of an attacker.
     o It is harder to quantify the information loss of modifications.
     o Modifications can propagate through the network.
Perturbative

- Adding noise only distribution counts
  - Value perturbation => numerical attributes
  - Graph perturbation
    - (randomly) adding / deleting nodes / edges

- Microaggregation / Clustering
  - Replace node data by centroid data
  - good for numerical data, but possible also for others given rules
  - Ensures k-anonymity only when computed over all attributes at the same time
  - Exact optimal only in P when computed over just 1 attribute (else heuristic)
Algorithmic approaches 3/3

Non-perturbative

- Generalization (hierarchies)
  - fixed ruleset
  - range partitioning (numerical values...)
  - Manual generation for many application domains (even ZIP...)

```
Level 2
  {A+, A, A-, B+, B, B-}

Level 1
  {A+, A, A-}
  {B+, B, B-}
```

- Suppression
  - Special case of generalization (with one level)

Figure 1: A possible generalization hierarchy for the attribute “Quality”.

“Social Network Greedy Anonymization” (SaNGreeA)

• Anonymizes a dataset w.r.t 2 information categories:
  • Feature vector values => traditional, tabular
  • Graph structure => edge configuration

• Based on the concept of ‘greedy’ clustering

• Which poses the question:
  • How do we choose the next node to add to a cluster w.r.t the above two criteria?

  ! We need some (good) cost functions !
• Generalization Information loss (GIL)
  • Based on content of nodes

• We assume
  • Continuous properties (age, body height, ...)
    • Candidate Nodes hold a particular value
    • Clusters have either particular value (at the start) or a
      generalized range
    • In order to incorporate the node into the cluster, we may
      have to generalize this range further, increasing the cost.

• Categorical properties (work class, native-country, ...)
  • Same preconditions as above
  • We use generalization hierarchies to determine the cost of
    clustering
• Generalization information loss function:

\[ GIL(cl) = |cl| \cdot \left( \sum_{j=1}^{s} \frac{\text{size}(\text{gen}(cl)[N_j])}{\text{size}(\min_{X \in \mathcal{N}}(X[N_j]), \max_{X \in \mathcal{N}}(X[N_j]))} + \sum_{j=1}^{t} \frac{\text{height}(\Lambda(\text{gen}(cl)[C_j]))}{\text{height}(\mathcal{H}_{C_j})} \right), \]

where:
- \( |cl| \) denotes the cluster \( cl \)'s cardinality;
- \( \text{size}([i_1, i_2]) \) is the size of the interval \([i_1, i_2]\), i.e., \((i_2 - i_1)\);
- \( \Lambda(w), w \in \mathcal{H}_{C_j} \) is the subhierarchy of \( \mathcal{H}_{C_j} \) rooted in \( w \);
- \( \text{height}(\mathcal{H}_{C_j}) \) denotes the height of the tree hierarchy \( \mathcal{H}_{C_j} \).

• Example GIL:

  • age_range overall = [11 – 91]
  • In order to cluster some nodes, we need to generalize 27 to [20 - 30]
  • Cost = (30-20)/(91-11) = 1/8

• Given a generalization hierarchy ‘native-country’ with 4 levels
• In order to cluster, we need to generalize ‘Austria’, ‘France’, or ‘Portugal’ to ‘Western Europe’, which is 1 level higher
• Cost = 1/4
Greedy anonymization Main Loop

```python
## MAIN LOOP
for node in adults:
    if node in added and added[node] == True:
        continue
    # Initialize new cluster with given node
    cluster = CL.NodeCluster(node, adults, adj_list, gen_hierarchies)
    # Mark node as added
    added[node] = True
    # SaNGreeA inner loop - Find nodes that minimize costs and
    # add them to the cluster since cluster_size reaches k
    while len(cluster.getNodes()) < GLOB.K_FACTOR:
        best_cost = float('inf')
        for candidate, v in ((k, v) for (k, v) in adults.items() if k > node):
            if candidate in added and added[candidate] == True:
                continue
            cost = cluster.computeNodeCost(candidate)
            if cost < best_cost:
                best_cost = cost
                best_candidate = candidate
        cluster.addNode(best_candidate)
        added[best_candidate] = True
    # We have filled our cluster with k entries, push it to clusters
    clusters.append(cluster)
```
Anonymization – prepared datasets

• We used k-factors of:

• 3, 7, 11, 15 and 19

• Each combined with three different weight vectors
  
  ➢ Equal weights for all columns
  ➢ Age preferred (0.88 vs 0.01 rest)
  ➢ Race preferred (0.88 vs. 0.01 rest)

• Resulting in 15 differently anonymized data sets
ML on Anonymization - Results

F1 score dependent on anonymization, gradient boosting

- equal weights
- age preferred
- race preferred

F1 score dependent on anonymization, linear SVC

- equal weights
- age preferred
- race preferred

F1 score dependent on anonymization, logistic regression

- equal weights
- age preferred
- race preferred

F1 score dependent on anonymization, random forest

- equal weights
- age preferred
- race preferred
Initial Conclusions

1. Succumbing to the “right-to-be-forgotten” still seems better than performing ML on anonymized DBs

2. A whole lot of future research is needed in order to corroborate and expand on those results

- Extension to other ML approaches
  => Multi-Class, Prediction, Dim. Reduction, Pattern Rec.
  (clustering not such a good candidate... why?)
- Other perturbation techniques
- Graph-based datasets
Can iML help anonymization 1/4?

Examples of iML?

• The CAT (Cornell anonymization toolkit) as well as ARX (TU Munich) allow you to run utility / risk analysis
• However, they are not interactive, but only support re-running your experiment with new settings...

Figure 2: Anonymization process
Possibilities to bring iML into anonymization?

1. Distance functions for Clustering
2. Information loss measures

• Both are subjective

• “Optimality” will also depend on the specific domain (medical vs. financial data)

• So (inter)active learning could be applied by involving a domain expert => the Human-in-the-loop approach...
Can iML help anonymization 3/4 ?
Can iML help anonymization 4/4?

Case: data similarity:

\[ \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}. \]

- Subset of Data
- Sample presented to User
- Update data + learn Heuristics
- User decides

Which two are more similar?
| [55 - 76] | * | North_Americ | Male | * | Married-civ-spouse |
| [55 - 76] | * | North_Americ | Male | * | Married-civ-spouse |
| [55 - 76] | * | North_Americ | Male | * | Married-civ-spouse |

51 | Private | United-States | Male | White | Married-civ-spouse |

| [48 - 70] | Private | America | Male | White | * |
| [48 - 70] | Private | America | Male | White | * |
| [48 - 70] | Private | America | Male | White | * |
Applying a weight vector to our desired columns will change our cost function and thereby produce different anonymization results:

<table>
<thead>
<tr>
<th></th>
<th>age</th>
<th>workclass</th>
<th>native-country</th>
<th>sex</th>
<th>race</th>
<th>marital-status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>age</th>
<th>workclass</th>
<th>native-country</th>
<th>sex</th>
<th>race</th>
<th>marital-status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.95</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Graph structure anonymization

\[
\text{intraSIL}(cl) = \left(\left(\frac{|cl|}{2} - |E_{cl}|\right) \cdot |E_{cl}| / \left(\frac{|cl|}{2}\right) + |E_{cl}| \cdot \left(1 - |E_{cl}| / \left(\frac{|cl|}{2}\right)\right)\right) = \\
2 \cdot |E_{cl}| \cdot \left(1 - \frac{|E_{cl}|}{\left(\frac{|cl|}{2}\right)}\right).
\]

\[
\text{interSIL}(cl_1, cl_2) = (|cl_1| \cdot |cl_2| - |E_{cl_1,cl_2}|) \cdot \frac{|E_{cl_1,cl_2}|}{|cl_1| \cdot |cl_2|} + |E_{cl_1,cl_2}| \cdot \left(1 - \frac{|E_{cl_1,cl_2}|}{|cl_1| \cdot |cl_2|}\right) = \\
2 \cdot |E_{cl_1,cl_2}| \cdot \left(1 - \frac{|E_{cl_1,cl_2}|}{|cl_1| \cdot |cl_2|}\right).
\]
• Conclusion: the level of privacy / security of data will always remain subjective with regard to the data set as well as potential attackers !!

• You can never answer the question: "Will this algorithm be good enough for our purposes?" without testing it thoroughly for your specific use cases on YOUR OWN DATA...

• Data that might seem safe today might become unsafe again in the future (additional
Thank you!

Bernd Malle <b.malle@hci-kdd.org>