Approximation algorithms for *k*-anonymity and privacy preservation in query logs

Aristides Gionis

Yahoo! Research, Barcelona, Spain

NATO Advanced Study Institute on
Mining Massive Data Sets for Security

September 10–21, 2007
Villa Cagnola, Gazzada, Italy

- privacy preservation
- k-anonymization
  - models, metrics, trade-offs, algorithms
- privacy preservation of query logs

Privacy preservation

- Many organizations collect and hold large volumes of data
  - credit card companies
  - real estate companies
  - hospitals
  - search engines
- Would like to publish the data for the purposes of data mining
- At the same time data contain a lot of sensitive information
- Would like to preserve the privacy of the individuals represented in the data

How to preserve privacy?

- Manipulate data in order to hide sensitive information
  - remove data records
  - perturb values (e.g., add random noise)
  - generalize entries (e.g., Barcelona $\Rightarrow$ Catalunya)
- Trade-off between data manipulation and utility

k-anonymity

- A successful paradigm for privacy preservation in data-mining and algorithms community

  [Samarati and Sweeney, 1998, Samarati, 2001, Sweeney, 2002, Meyerson and Williams, 2004, Aggarwal et al., 2005a, Bayardo and Agrawal, 2005, LeFevre et al., 2005, Machanavajjhala et al., 2006, Nergiz and Clifton, 2006]
- Main idea: ensure that in the released data each data record is indistinguishable from $k - 1$ other records

k-anonymity — motivating example

Name	Age	Edu	Job	Owns Apart	Post Code	Salary	Loan Appr
John	22	hs	sales	No	00510	28 K	No
Mary	30	hs	sales	Yes	00510	35 K	Yes
Alex	25	hs	sales	No	00510	30 K	No
Mark	28	BS	IT	Yes	00512	40 K	No
Jane	32	MS	IT	Yes	00510	45 K	Yes
Mike	40	BS	IT	Yes	00512	55 K	Yes

k-anonymity — motivating example

Name	Age	Edu	Job	Owns Apart	Post Code	Salary	Loan Appr
John	22	hs	sales	No	00510	28 K	No
Mary	30	hs	sales	Yes	00510	35 K	Yes
Alex	25	hs	sales	No	00510	30 K	No
Mark	28	BS	IT	Yes	00512	40 K	No
Jane	32	MS	IT	Yes	00510	45 K	Yes
Mike	40	BS	IT	Yes	00512	55 K	Yes

k-anonymity — motivating example

Name	Age	Edu	Job	Owns Apart	Post Code	Salary	Loan Appr
John	*	hs	sales	*	00510	28 K	No
Mary	*	hs	sales	*	00510	35 K	Yes
Alex	*	hs	sales	*	00510	30 K	No
Mark	*	*	IT	Yes	*	40 K	No
Jane	*	*	IT	Yes	*	45 K	Yes
Mike	*	*	IT	Yes	*	55 K	Yes

k-anonymity — motivating example

Name	Age	Edu	Job	Owns Apart	Post Code	Salary	Loan Appr
John	22	hs	sales	No	00510	28 K	No
Mary	30	hs	sales	Yes	00510	35 K	Yes
Alex	25	hs	sales	No	00510	30 K	No
Mark	28	BS	IT	Yes	00512	40 K	No
Jane	32	MS	IT	Yes	00510	45 K	Yes
Mike	40	BS	IT	Yes	00512	55 K	Yes

k-anonymity — motivating example

Name	Age	Edu	Job	Owns Apart	Post Code	Salary	Loan Appr
John	22–30	hs	sales	*	00510	28 K	No
Mary	22–30	hs	sales	*	00510	35 K	Yes
Alex	22–30	hs	sales	*	00510	30 K	No
Mark	28–40	BS,MS	IT	Yes	00510–2	40 K	No
Jane	28–40	BS,MS	IT	Yes	00510–2	45 K	Yes
Mike	28–40	BS,MS	IT	Yes	00510–2	55 K	Yes

k-anonymity — motivating example

Name	Age	Edu	Job	Owns Apart	Post Code	Salary	Loan Appr
John	22–30	hs	sales	*	00510	30 K	No
Mary	22–30	hs	sales	*	00510	30 K	No
Alex	22–30	hs	sales	*	00510	30 K	No
Mark	28–40	BS,MS	IT	Yes	00510–2	40 K	No
Jane	28–40	BS,MS	IT	Yes	00510–2	45 K	Yes
Mike	28–40	BS,MS	IT	Yes	00510–2	55 K	Yes

Still not perfect privacy ensured; *l*-diversity

k-anonymity

- Quasi-identifier: a set of attributes in the public database that can be linked with external information (e.g., Name, Age, Edu, Job, PostCode)
- k-anonymity property
  Let D be a database and Q_D quasi-identifier attributes. We say that D is k-anonymized iff each set of values in $D(Q_D)$ appears in at least k records of D
- k-anonymization problem

  Given a database table (public part)
  - hide minimum amount of information, and
  - ensure k-anonymity

k-anonymity issues

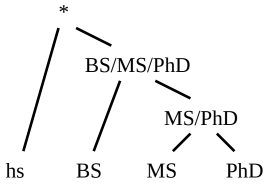
- What kind of operations are allowed to modify the data?
- How to measure information loss?
- Complexity of the problem and algorithms

k-anonymity operations

- Suppression
  a data entry is always replaced by *****
- Generalization
  a data entry is replaced by a more general value
  e.g., 08003 $\Rightarrow$ 0800*, or MS $\Rightarrow$ PostGraduate

k-anonymity generalization operation

- Domain generalization hierarchy (DGH)
  generalize along a hierarchy (usually provided by the user)



- Natural domain generalization hierarchy (NDGH) [Nergiz and Clifton, 2006]
  generalize along arbitrary lattices (not restricted to trees)
- Unrestricted generalization
  as NDGH

How to apply the generalization operation?

- Single-dimension generalization (SDG)
  a value is mapped always to the same generalized value e.g., 08003 $\Rightarrow$ 0800* (always)
- Multiple-dimension generalization (MDG)
  a set of values is mapped always to the same generalized set of values, e.g.,

  {08003, high-school} $\Rightarrow$ {0800*, elem/high } (always)
  {08003, BS } $\Rightarrow$ {080**, uni } (always)
- Cell-based generalization (CBG)
  the same value in different data records can be mapped to different generalized values

SDG $\subseteq$ MDG $\subseteq$ CBG

SDG vs. CBG

- SDG +
  - resulting table easier to use in certain data mining applications
- CBG +
  - less information loss, better utility
  - less sensitive to outliers

In the next of the talk we focus on CBG

Some notation

Dataset:	D	n records, m attributes
i-th row:	D_i	$i = 1 \dots n$
j-th value of i-th row:	$D_i(j)$	$j = 1 \dots m$
Values of the j-th attribute:	A_j	$j = 1 \dots m$

Some notation

- Generalization:
  a mapping $g : A_j \to \overline{A}_j \subseteq \mathcal{P}(A_j)$
  natural assumption $A_j \subseteq \overline{A}_j$
  suppression: $\overline{A}_j = A_j \cup \{*\}$
  hierarchical: $\overline{A}_j$ forms a hierarchy under $\subseteq$
  unrestricted: $\overline{A}_j = \mathcal{P}(A_j)$
- Generalization of a dataset D:
  $g_i : A_1 \times \cdots \times A_n \to \overline{A}_1 \times \cdots \times \overline{A}_n$ generalization operators
  $\overline{D}_i := g_i(D_i)$, $1 \leq i \leq n$
  $g(D) := \{\overline{D}_1, \ldots, \overline{D}_n\}$ generalization of D

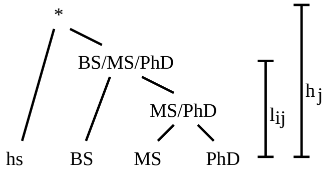
Measures of information loss

Recall: *k*-anonymization problem

Given a database table
- hide minimum amount of information, and
- ensure *k*-anonymity

Measures of information loss

- Suppressions: number of *****'s [Meyerson and Williams, 2004]
- Generalization by hierarchies [Aggarwal et al., 2005a]



Cost: $\sum \frac{l_{ij}}{h_j}$

Measures of information loss

Unrestricted generalizations

- LM measure:

$$\mathrm{LM}(g(D)) = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{|\overline{D}_i(j)| - 1}{|A_j| - 1}$$

- DM measure:

$$\mathrm{DM}(g(D)) = \sum_{i=1}^{n} |I_{g(D)}(\overline{D}_i)|,$$

  where $I_{g(D)}(t)$ the set of records indistinguishable from t
- and more: CM measure, AM measure

(Recall, $D = \{D_1, \ldots, D_n\}$ generalizes to $g(D) = \{\overline{D}_1, \ldots, \overline{D}_n\}$)

Entropy-based measures of information-loss

- [Gionis and Tassa, 2007]

  $\{\text{Male, Female}\} \rightarrow *$ loses 1 bit

  Age $\rightarrow *$ loses many bits

$$P[X_j = a] = \frac{\#\{1 \le i \le n : D_i(j) = a\}}{n}$$

$$H(X_j) = -\sum_{a \in A_j} P[X_j = a] \log_2 P[X_j = a]$$

$$H(X_j | B_j) = -\sum_{b \in B_j} P[X_j = b | X_j \in B_j] \log_2 P[X_j = b | X_j \in B_j]$$

Then, find the mapping $g_i : D_i(j) \rightarrow \overline{A}_j$ to minimize

$$I(g(D)) = \sum_{i=1}^{n} \sum_{j=1}^{m} H(A_j | g_i(D_i(j)))$$

Example – how accurate is the entropy measure

- Inter-dependencies among columns not taken into account
- Inter-dependencies among rows not taken into account

$$D = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 3 \end{bmatrix}, \quad g(D) = \begin{bmatrix} * \\ * \\ 3 \\ 3 \end{bmatrix}, \quad p_1 = \frac{1}{4}, \quad p_2 = \frac{1}{4}, \quad p_3 = \frac{1}{2}$$

- $H(A|*) = 1.5$, implying $I(g(D)) = 1.5 + 1.5 + 0 + 0 = 3$
- However, the actual information loss is 2 bits
- Entropy measure overestimates the actual information loss

Non monotonicity

- Consider $A_j = \{1, 2, 3\}$ with probabilities $(1 - 2\epsilon, \epsilon, \epsilon)$
- Replacing 2 and 3 by $\{2, 3\}$ loses 1 bit
- Replacing 2 and 3 by * loses $O(\epsilon)$ bits
- Non monotonicity: generalizing to a larger set gives smaller amount of information loss
- Does not seem natural to have non-monotone measure
- But we can design $\overline{A}_j$ so that entropy is monotone

Rectification of non monotonicity

- Recall

$$I(g(D)) = \sum_{i=1}^{n} \sum_{j=1}^{m} H(A_j | \overline{D}_i(j))$$

1. We can design $\overline{A}_j$ so that entropy is monotone
   e.g., use a bottom-up algorithm that merges pairs of sets in which monotonicity is violated
2. Monotone entropy measure:

$$I'(g(D)) = \sum_{i=1}^{n} \sum_{j=1}^{m} P(\overline{D}_i(j)) H(A_j | \overline{D}_i(j))$$

Hardness results and approximation algorithms

[Meyerson and Williams, 2004]

- Only suppressions (*)
- k-anonymization problem is NP-hard, for $k \geq 3$ and large alphabet,
  (where alphabet is the domain of each attribute)
- $O(k \log k)$-approximation algorithm
  algorithm based on set cover
  running time $O(n^{2k})$
- $O(k \log n)$-approximation algorithm
  strongly polynomial

Hardness results and approximation algorithms

[Aggarwal et al., 2005b, Aggarwal et al., 2005a]

- Extended to generalization by suppressions
- NP-hardness for ternary alphabets and $k \geq 2$
- $O(k)$-approximation algorithm for any k
  1.5 approximation algorithm for $k = 2$
  2 approximation algorithm for $k = 3$
- $\Omega(k)$ lower bound on the approximation ratio,
  if graph representation is used

The forest algorithm

[Aggarwal et al., 2005b, Aggarwal et al., 2005a]

- Based on the graph representation
- Provides $O(k)$-approximation for the k-anonymization problem
- Consider a graph $G = (V, E)$, in which V is the set of rows in the database and the cost of an edge $(u, v) \in E$ denotes the minimum generalization cost paid for u and v to become identical
- Outline:
- Find a forest of G with total cost at most OPT and such that all trees have size at least k
- Split large trees, so that all trees have size at most $L = O(k)$
- Approximation ratio is L

New results and the entropy measures

[Gionis and Tassa, 2007]

- NP-hardness for the entropy measure and the monotone entropy measure
- $O(\log k)$-approximation algorithm for both of the entropy measures, as well as the previous measures
- Running time is $O(n^{2k})$
- Adapt the Forest algorithm to obtain $O(k)$-approximation algorithm for the entropy measures in time polynomial in both n and k

k-anonymity algorithm for the entropy measure

- In a k-anonymity solution, let a cluster be a set of identical rows (after the information hiding process)
- Observation 1: All clusters in an optimal k-anonymity solution have between k and $2k - 1$ rows
- Observation 2: Given a set of rows X to be clustered together, we can compute the optimal information loss $I(X)$ required for the rows to become identical

k-anonymity algorithm for the entropy measure

- Generate all subsets of rows of sizes between k and $2k-1$ ($O(n^{2k})$ such sets)
- for each such set X compute its cost $I(X)$
- Solve set cover on those subsets

  /* find a minimum-cost collection of the sets that cover all
   * rows can be approximated within factor $O(\log k)$ */
- Convert cover to clustering

  /* check each pair of overlapping sets and eliminate overlaps
   * need to assume that collection $\overline{A}_j$ is *proper* */

Summary (so far)

- Entropy-based measures for k-anonymity
- $\mathrm{O}(\log k)$ approximation algorithm for the entropy measure and other previous measures (non-strongly polynomial)
- $O(k)$ approximation algorithm for the entropy measures (strongly polynomial)
- $O(k)$ ratio has been shown to be optimal for the graph representation case

Privacy preservation in query logs

- Search engines collect a large amount of query logs
- Contain a lot of interesting information that can be used for
  - analyzing users' behavior
  - creating user profiles
  - personalization
  - creating knowledge bases and folksonomies
  - finding similar concepts
  - building systems for query suggestions and recommendations
  - using statistics for improving systems' performance
  - etc.

Privacy preservation in query logs

- Users' online behavior is sensitive
- People would not like to have their web searches revealed
- Some searches are more sensitive than other
- On the other hand, releasing query logs can be beneficial for analysis, research, improving systems' performance and functionality

Trade-off between utility and privacy

- Release of a query log can be possible after sufficient obfuscation of the data
- Trade-off between utility of released information and privacy
- The more complex the data mining task needed to be performed,
  the more the amount of privacy that needs to be taken away from the users
  - For example, for building a model for query inter-arrival times, can release a query log with no private information
  - For building a model for predicting the salary of a user user on their queries, it is not so clear...

The AOL query-log release

- American Online (AOL) query log released in August 2006
- Objective was to contribute to IR research
- Query log rough statistics
  - 20 million queries
  - 650 K users
  - from over 3 months
- Social security numbers, credit card numbers, driver license numbers, etc.
- Possible to uniquely identify many users by combining information from queries and yellow pages
- Big media scandal, big damage to AOL and the privacy of its users

A typical query log

Entries of the format

<cookie, query, rank, clickURL, timeStamp, IP, country,...>

Anonymizing query logs

[Adar, 2007]

- Argue that anonymization is potentially possible
- Two main techniques:

1 Eliminate infrequent queries

2 Splitting personalities

- Additionally:

3 Eliminate identifying information
(SSN, credit card numbers, etc.)

Anonymizing query logs

Eliminate infrequent queries

- Keep only queries generated by a large number of users
- Computationally possible using counters
- How to do it on-the-fly?

Online elimination of infrequent queries

- Background: How to split a secret among n people so that every coalition of k persons can access the secret?
- Answer:
  Let the secret be the coefficients of a $(k-1)$-degree polynomial

$$f(x) = a_{k-1}x^{k-1} + \ldots + a_1 x + a_0$$

- For the i-th person, select a number x_i, and give to the person the pair $(x_i, f(x_i))$
- Any k persons can cooperate and recover the polynomial, while no $k-1$ persons can recover it

Online elimination of infrequent queries

- Straightforward application in eliminating infrequent queries
- A query q is decoded as a $(k-1)$-degree polynomial f_q
- For a person u_i who makes the query q, print

$$(u_i, f_q(u_i))$$

- If k or more people type the query q, it is possible to decrypt q!

Split personality

- Split the queries of the same user into sessions
- E.g., queries about food recipes, sport results, buying books, music, etc.
- Assign each of those sessions to a different virtual user
- Released query log can be still useful for many applications
- More difficult to identify users by combining queries
- Finding similar queries and finding query sessions is quite hard problem

More on anonymizing query logs: negative results

[Kumar et al., 2007]

- Anonymization via token-based hashing:
- The query is split into terms and each term is hashed to a token
- Co-occurrence analysis and frequency analysis can be used to reveal the query terms
- Assume access to an unencrypted query log
- Query term statistics remain constant across different query logs
- Provide practical graph-matching algorithms and analysis of real query logs

Summary

- Releasing data so that privacy is preserved is an important research area
- Studied *k*-anonymity, new notions, algorithms
- Privacy preservation in query logs — a lot of care is needed

Adar, E. (2007).

User 4XXXXX9: Anonymizing Query Logs.

In *WWW*.

Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., and Zhu, A. (2005a).

Approximation algorithms for *k*-anonymity.

Journal of Privacy Technology.

Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., and Zhu, A. (2005b).

Approximation algorithms for *k*-anonymity.

In *ICDT*.

Bayardo, R. and Agrawal, R. (2005).

Data privacy through optimal *k*-anonymization.

In *ICDE*.

Gionis, A. and Tassa, T. (2007).

k-anonymization with minimal loss of information.

Kumar, R., Novak, J., Pang, B., and Tomkins, A. (2007).
On anonymizing query logs via token-based hashing.
In *WWW*.

LeFevre, K., DeWitt, D., and Ramakrishnan, R. (2005).
Incognito: efficient full-domain k-anonymity.
In *SIGMOD*.

Machanavajjhala, A., Gehrke, J., Kifer, D., and Venkitasubramaniam, M. (2006).
l-diversity: Privacy beyond k-anonymity.
In *ICDE*.

Meyerson, A. and Williams, R. (2004).
On the complexity of optimal k-anonymity.
In *PODS*.

Nergiz, M. E. and Clifton, C. (2006).
Thoughts on k-anonymization.
In *ICDE Workshops*.

Samarati, P. (2001).

Protecting respondent's privacy in microdata release.

IEEE Transactions on Knowledge and Data Engineering, 13:1010–1027.

Samarati, P. and Sweeney, L. (1998).

Generalizing data to provide anonymity when disclosing information (abstract).

In *PODS*.

Sweeney, L. (2002).

k-anonymity: A model for protecting privacy.

International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5):557–570.