Spatial Indexing Techniques for Privacy-Preserving Data Publishing

Guillaume Raschia, Adeel Anjum

▶ **To cite this version:**

Guillaume Raschia, Adeel Anjum. Spatial Indexing Techniques for Privacy-Preserving Data Publishing. Atelier Protection de la Vie Privée (APVP) 2011, Jun 2011, Sorèze (81), France. hal-00603192

HAL Id: hal-00603192
https://hal.science/hal-00603192v1

Submitted on 24 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Spatial Indexing Techniques for Privacy-Preserving Data Publishing (*position paper*)

L'indexation spatiale au service de la publication de données qui préserve la vie privée.

Guillaume Raschia
LINA, University of Nantes
guillaume.raschia@univ-nantes.fr

Adeel Anjum
LINA, University of Nantes
adeel.anjum@etu.univ-nantes.fr

ABSTRACT

Il existe une littérature foisonnante au sujet de l'anonymisation de données par généralisation des valeurs d'enregistrement. Les contributions portent soit sur le modèle de généralisation, enrichissant le K-anonymat de L. Sweeney, soit sur un algorithme de calcul d'une version anonymisée d'un jeu de données, étant entendu qu'il a été prouvé que le problème est $\mathcal{NP}$-difficile, et donc que la plupart des algorithmes offrent des solutions approchées.

Les méthodes d'accès spatiales (MAS), bien étudiées dans le champ des bases de données, suscitent un intérêt grandissant dans le contexte de l'anonymisation de données par généralisation des valeurs d'enregistrements, étant données (i) leur capacité à atteindre des grandeurs d'échelle inégalées par d'autres approches, et (ii) l'adéquation du format des entrées de l'index avec les données anonymisées.

Nous proposons dans cette communication un état des lieux et une analyse critique des MASs étudiées sous l'angle de l'anonymisation de données. Nous évaluons les propositions existantes et suggérons des pistes de travail encore non explorées à ce jour.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Management—*Database applications*; E.1 [**Data**]: Data Structures

General Terms

Privacy-Preserving Data Publishing

Keywords

Anonymization, Spatial Indexing, Point Access Method

1. INTRODUCTION

1.1 Motivation

As stated by Jim Gray [11], we are entering the fourth age of science defined by a new paradigm where data play a central role in the production of *science* and *innovation*. To achieve that bright vision, *scientific* data must be unleashed from private repositories, and publicly released for the all research community. The Open Access movement, first concentrated on free access to scientific publications, turns now to Open Data initiative. In the same time, new business models have emerged to offer valuable services and take benefits from open data.

Then, organizations are strongly encouraged to release their micro-data to support data analysis, to provide new business opportunities and to allow every kind of scientific study and to support data journalism as well. For example, patients' medical records may be released by a clinic to support medical research and epidemiological studies.

However, releasing medical records about individuals violates their privacy thus, *Privacy-Preserving Data Publishing* (PPDP) has become a critical issue for companies and organizations. To obviate identity disclosure, many organizations usually remove the uniquely identifying information like name, SSN or IP address from the public release. However, as stated first by Latanya Sweeney in [23], observing a 30-year tradition of inference problem in statistical databases, this sanitization of data might not be helpful in keeping the secrecy of given individuals since several attributes, coined *quasi-identifiers*, if they are put together, could surely lead to identity disclosure.

1.2 Preliminaries

This gave rise to the need for robust sanitization methods to publish sensitive individual data keeping their privacy intact. The seminal *K-anonymization* paradigm [23] was proposed to achieve this goal by means of a generalization model. Basically, anonymization based on generalization consists in decreasing accuracy of values from quasi-identifiers. For instance, 44100 Zip code would become 44XXX and 70 pounds would be said to range between 50 and 80 pounds. More precisely, *a table satisfies K-anonymity if every record is indistinguishable on quasi-identifiers from at least $K-1$ other records.* This indistinguishability principle supports an equivalence relationship on the records of an anonymous public release and prevents from identity disclosure of individuals with a probability of

APVP '2011 Toulouse, France

Id	Age	Zipcode	Gender	Disease
(1)	[48−62] (62)	441XX (44120)	* (F)	Flu
(2)	[48−62] (51)	441XX (44190)	* (M)	Flu
(3)	[48−62] (48)	441XX (44100)	* (M)	HIV
(4)	[59−77] (59)	444XX (44470)	* (F)	Flu
(5)	[59−77] (77)	444XX (44420)	* (M)	Gastritis
(6)	[59−77] (66)	444XX (44420)	* (M)	HIV

Table 1: Example of a 3-Anonymous Public Release (with raw values into bracket).

$1/K$.

Table 1 provides a toy example of a public release of 6 medical records following 3-anonymity, i.e. each public record is identical on quasi-identifiers (*Age*, *Zip* and *Gender*) with at least 2 other records. For instance, records 1, 2, 3 from Table 1 belongs to the same equivalence class and are indistinguishable one with each other. Pattern of the class is (Age=[48−62], Zip=441XX, Gender=*). Similarly, records 4, 5, 6 form the second equivalence class.

1.3 Typical Use Case

Public release supports exploration, analysis and scientific studies. The very first and popular processing of public release is then to search and filter tabular data by the way of *point queries* and *window queries*. Indeed, regular database records can be geometrically interpreted as points in a multi-dimensional space where each dimension is a column of the raw table. Point coordinates are then defined by attribute values. Transformation is obvious for numerical and ordinal variables. Categorical variables could also be equipped with a total ordering, except that without any "native" ordering, the process is driven by the application domain and background knowledge.Thus database queries are transformed into queries against a set of points.

Translated to anonymous public releases, sanitized records become *hyperrectangles* in a multi-dimensional space, where each dimension is a field in the set of quasi-identifiers. For instance, sanitized records from Table 1 are cuboids in the 3-dimensional space (`Age`, `Zipcode`, `Gender`) as shown by Figure 1. As a point query example Q_1, user would filter data to retrieve possible patient's record designated by `Age=62 AND Zip:=44120`[1] `AND Gender:=F`[2] and she would be returned the answer set $\{1, 2, 3\}$ from Table 1. Similarly, an example of a window query Q_2 would yield to searching for patient's records satisfying `Zip IN [4442X,4447X] AND Age>=50 AND Gender=*`. It would then return the set $\{4, 5, 6\}$.

To achieve such querying scenario, anonymous records are mutually disjoint spatial objects with a *rectangular extent* and window queries are *orthogonal range queries*. And any record that overlaps/lies within query region is a member of the result set. There exist many efficient algorithms and data structures [1] to compute such orthogonal range queries against the spatial representation of the anonymous database. Furthermore, since any orthogonal range query

[1]Expand the query to Zip=44120 OR Zip=4412X OR Zip=441XX OR Zip=44XXX OR Zip=4XXXX OR Zip=*.
[2]Expand the query to Gender=F OR Gender=*.

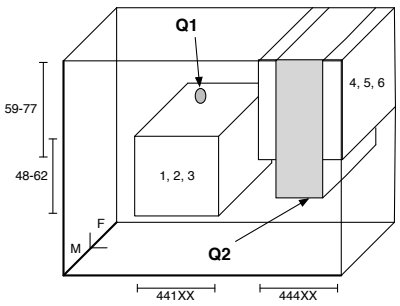

Figure 1: 3D spatial representation of the anonymous public release from Table 1 with point query Q_1 and window query Q_2.

can be *decomposed into a collection of 1-dimensional range queries*, it is then easy to manage filters on the tabular representation of the public release within a basic spreadsheet or web-client technologies as well. Query Q_2 over Table 1 gives an example of such straightforward decomposition. Those practical features are very useful in lots of iterative exploration processes that would support analysis and scientific studies. Then, we argue that the *axis-parallel rectangular coding of anonymous records is a strong requirement for a generic PPDP task*.

Other kinds of range queries are defined by the shape of query region: sphere, half-space, simplex, polytope. Sphere range queries, so-called *nearest-neighbor* queries have been extensively studied and there also exist efficient algorithms to compute such popular queries especially on rectangular objects. However, none of these range queries satisfies the decomposition property that makes anonymous releases human-friend under tabular representation.

1.4 Summary of the Requirements for PPDP

To sum-up the above discussion, we argue that every generic PPDP task should meet at least the following theoretical and practical requirements in order to be valuable for the end-user:

1. Indistinguishability principle — to ensure K-anonymity;
2. Mutually disjoint equivalence classes — to preserve quality of the anonymous public release;
3. Multidimensional point partitioning — to support point and range queries on the anonymous public release;
4. Hyperrectangular coding of equivalence classes — to allow decomposition of orthogonal range queries.

1.5 To follow-on

Section 2 reviews existing structures that support anonymization algorithms, and it presents features of the main logical structures eligible for a PPDP task. Next, Section 3 focuses on a special kind of those structures, so-called *nested hyperrectangle-based bucketed point access methods*, that have very nice features for the anonymization. Definition and searching strategies for such structures are provided. Then,

Section 4 states several open issues to address in order to accommodate and optimize the index structures to the PPDP task.

2. POINT ACCESS METHODS

Point Access Methods (PAMs) are logical structures that organize a set of point for efficient searching. We will see in this section that PAMs have features that are suitable for the anonymization problem, and as such, we argue in the following that they are the preferred data structures to support K-anonymization algorithms.

2.1 Point Partitioning for PPDP

It is worth to notice that public release with one single equivalence class described on each dimension by the all domain is obviously K-anonymous ($K \leq n$ the number of records) but it is definitely useless for the end-user. Thus, the main challenge of K-anonymization is to compute a public release where the information loss has been minimized, in the sense of a general criteria such like *certainty metric* [25], *discernibility penalty* [3] or *KL- divergence* [13] for the most popular ones. This optimization problem was proved to be $\mathcal{NP}$-hard [18].

Hence, many approximation algorithms have been proposed in the literature since the seminal work of L. Sweeney [22]. Usually, Mondrian approach [15] is thought as the baseline algorithm since it has the basic good properties we could expect from such algorithms: local recoding and multi-dimensional partitioning. Mondrian iteratively operates a binary partitioning of the data space until every block contains between K and $2K-1$ points. Actually, Mondrian builds a *kd*-tree over the raw data and publishes bounding boxes of the leaves as equivalence classes of the anonymous release. Construction has time complexity $O(N \cdot \log N)$, where N is the number of records in raw data.

Following the geometric representation of the data, Iwuchukwu et al. [12] propose to use a bulk-loading implementation of an R^+-tree, one of the most popular spatial access methods for databases, to compute the K-anonymous release. It outperforms Mondrian thanks to buffering and efficient bottom-up index construction algorithm, and it scales up to very large data sets. Furthermore, the hierarchical structure of the R^+-tree natively supports $(B^\ell K)$-anonymity for all level ℓ in the tree, with B the fanout parameter. And with an ordered leaf scan, it could support (cK)-anonymity as well, for all c in $\mathbb{N}$. Time complexity remains in $O(N \cdot \log N)$. And I/O cost for external computation is in $O(N/B \cdot \log N/B)$.

Since the R^+-tree bulk-loading algorithm is applied on a set of points rather than a set of spatial objects with an extent, it is actually a variant of a *kd-B*-tree structure where hyperrectangles have been shrinked to the minimum bounding boxes (MBB) of the subset of points in each equivalence class. Remind that a *kd-B*-tree is a bucket-oriented variant of a *kd*-tree where the fanout of each node is defined by parameter B that usually fits the disk block size. The many good features of the R^+-tree approach makes it therefore the reference algorithm for K-anonymization until now.

Many works also proposed point partitioning structures in low dimension (2-3D) for privacy preserving location-based queries [10, 9, 19, 6]. In this application domain, privacy is related to instant location of users and queries as well. Popular approaches design an anonymizer that dynamically provides a Cloaking Region to the Location-Based Service. For that purpose, Gruteser et al. [10] implements a *kd*-tree, whereas Mokbel et al. [19] uses a variant of a PR quad-tree in Casper. Ghinita et al. [9] accommodate partitioning structures from *kd*-tree and *R*-tree to hash a database of Points Of Interest (POI) and answer approximate nearest-neighbor queries in a Privacy Information Retrieval (PIR) approach. They also consider Hilbert space filling curves to map 2D points to single-dimensional data structures like B^+-trees to index POIs. Actually, they argue that their PIR approach is independent from the partitioning structure as far as it provides at most $\sqrt{N}$ buckets within up to $\sqrt{N}$ POIs each. Other work [14] focused on geo-privacy in the sense of privacy-preserving *location* data publishing. In this context, a space filling curve was also employed to order both data points and POIs on the map. Quad-trees and space filling curves do not scale for higher dimensions, and the latter cannot guarantee non overlapping bounding boxes in the worse case.

The above short review states that every approach to geo-privacy accommodates in memory and implements well-known structure for multi-dimensional point data partitioning.

K-anonymity were also studied from the *cardinality constraint clustering* point of view. On the one hand, anonymization algorithms were proposed [4, 5, 2] that achieve good quality, whereas neither they scale up in the size of the data set, nor they meet the basic orthogonal range query requirement since patterns are spheres (centers and radius) of each cluster. On the other hand, many grid clustering techniques ([24, 20] for a short excerpt) have been proposed. However, none of them are as fast and scalable as Point Access Methods (PAM) since external storage support and dedicated insert-delete-search operations are missing. Then, PAMs remain the preferred logical structures for the anonymization of very large data sets.

2.2 Comparative Analysis of PAMs

For an insight into multi-dimensional Point Access Methods, the reader is strongly invited to refer to the first chapter of [21]. In Table 2, we present a short comparison between the most popular PAMs that could be of interest for PPDP task. For the sake of simplicity, we omit the multiple extensions of each structure, available in the literature, since the main criteria of our comparison are inherent to each structure such that they remain valid whenever the extension.

Criteria are as follows:

- *bucket?* — decides whether the PAM is bucketed or not, i.e. each element of the logical structure has a parametrized size rather than a fixed-length size. Bucket PAMs are those that could be used as spatial indices for databases since the bucket size B is set to the disk page size and then, the I/O cost of such structures is controlled. Those structures are *external* or *secondary storage* structures and then, they can grow as much as the size of the data set requires to, without main memory limitations;

	bucket?	orientation	shape	grid?	done
kd-tree	No	top-down	HR	No	✓
kd-trie	No	top-down	HR	Yes	✓
BD-tree	No	top-down	NHR	No	–
BSP tree	No	top-down	CP	No	×
PR quadtree	No	top-down	HR	Yes	✓
kd-B-tree	Yes	top-down	HR	No	–
kd-B-trie	Yes	top-down	HR	Yes	–
Grid file	Yes	top-down	HR	Yes	–
R^+-tree	Yes	bottom-up	HR	No	✓
hB-tree	Yes	top-down	NHR	No	–
BV-tree	Yes	bottom-up	NHR	No	–
BANG file	Yes	top-down	NHR	Yes	–

Table 2: Comparison of index structures for multi-dimensional point data. *HR* stands for HyperRectangle, *NHR* is Nested HR, *CP* means Convex Polytope.

- *orientation* — separates PAMs into 2 categories: those that decompose the underlying space, and those that aggregate the data points. The former are *top-down* since they iteratively divide the space to build the blocks, and the latter are *bottom-up* since they operate from the data to the blocks;
- *shape* — blocks of the partitioning could have various shapes in the space. The most simple but popular one is the *hyperrectangle* (HR);
- *grid?* — decides whether pre-defined scales support the PAM or not, such that every partition line follows a grid in the space. PAMs with such feature adopt regular decomposition.
- *done* — already used into an anonymization approach? (see Section 2.1 above for a review).

The 5 first raws of Table 2 refer to in memory structures. Except the BSP tree, all of them build (nested) hyperrectangular ((N)HR) blocks, thus they meet the PPDP requirements as stated in Section 1.4. The NHR property will be discussed further in Section 3.1. The BSP tree builds convex polytopes that do not allow to decompose orthogonal range queries then it is not eligible for a PPDP task. The only BD-tree, that builds nested HRs, has not already been support of an anonymization process.

The all remaining raws are Bucketed PAMs (BPAMs) that is, indexing structures for point databases. Among them, the 4 first structures generate HR blocks, whereas the 3 last ones provide nested HRs. The only R^+-tree was used for PPDP until now.

Moreover, we argue that bottom-up spatial indexing is not systematically more efficient than top-down approaches as opposed to the conjecture from [12]. This result is given by our own experiments comparing in the same running environment R^+-tree approach (bottom-up) with the BANG file (top-down). Following usual analysis on spatial access methods, we claim that the performance is mainly dependent from the *splitting strategy*. In the BANG file, we use regular decomposition following the grid whereas the original R^+-tree grows by means of a quadratic procedure comparing pairwise distances of elements in an overflow bucket. Those strategies determine a constant factor (w.r.t. N, the number of points) in time complexity that makes the execution time slower for the R^+-tree. Hence, both top-down and bottom-up approaches deserve to be studied in the context of PPDP.

Finally, the grid-based PAMs have the ability to support background knowledge in the space decomposition process by means of dimensional scales. Consequences are multiple. First, the block splitting strategy is straightforward since scales have been pre-defined over each dimension, so that the algorithm performs very well. Second, the lowest K value is given by the density of the finest regions in the grid. Hence the user controls the privacy requirement by means of the grid resolution rather than a parameter K. Obviously, grid resolution could be adapted to match a given K value when needed.

2.3 Focus on Bucketed PAMs

Bucketed PAMs (BPAMs) are well-suited for the anonymization task. The very first reason is that BPAMs fulfill the basic requirements for PPDP as stated in Section 1.4. But BPAMs have many other nice features that could be of interest in the context of PPDP. First, since they support spatial indexing techniques in databases, they leverage 30-years research and experience in effective and efficient multi-dimensional partitioning data structures built from very large data sets. Thus, they scale up and perform very well.

Next, BPAMs natively offer basic insert-delete-search operations that straightforwardly make the anonymization process *incremental*. It then supports dynamic updates of the dataset *before* the generation of the anonymous public release, and it provides a framework to study the open issue of continuous publication.

Moreover, BPAMs require a search operation to perform at least in $O(\log n)$ to be efficient. Thus, they all develop a hierarchical structure, so-called *tree directory*, that makes possible *multi-granular anonymization* with partitioning extraction at any level in the tree. The only exception would be the Grid file that performs in $O(1)$ such like *linear hashing*, having the main drawback of a low filling rate in each block and a large and sparse directory.

3. NESTED HYPERRECTANGLES

We focus in this section on a category of BPAMs: those with nested hyperrectangle blocks. We first present their features and then we discuss about open issues such that they could effectively support PPDP tasks.

3.1 Features of NHR-based BPAMs

We argue that...

> *... NHR-based BPAMs are the most sophisticated and suitable logical structures to support PPDP tasks.*

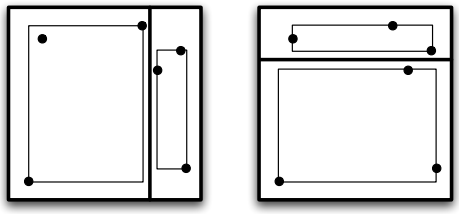

Figure 2: Low quality binary partitioning of a set of 6 points into blocks of at least 3 points, following either (a) X-axis, or (b) Y-axis.

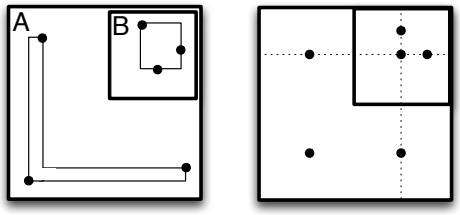

Figure 3: NHR partitioning with cardinality constraint (≥ 3 points), (a) on points from Figure 2, and (b) where HR partitioning fails.

NHR-based BPAMs operate an axis-parallel space partitioning by means of *nested hyperrectangles* rather than disjoint hyperrectangles only. This singular feature allows to improve expressive power of patterns compared to other HR-based BPAMs. For example, given a set of 6 points in a 2-dimensional space, as shown on Figure 2; assume we are trying to 3-anonymize the data set. Then, the alternative HR partitionings are those drawn on Figure 2. It also provides the MBBs of each block as the R^+-tree do. Similarly, Figure 3 shows (a) the partition obtained by a NHR-based BPAM for the same problem, and (b) a set of 6 points that can even not be partitioned with a HR-based BPAM but that can be divided within nested hyperrectangles.

Both pictures of Figure 3 show an outermost region A and a nested region B. Space spanned into $A - B$ forms one block, denoted by $[A]$, assigned to an equivalence class of the public release, whereas points that lie into B are the second block $[B]$.

Hence, NHR-based BPAMs are known to better observe clustered values into data and also to improve the filling rate of each block since there are more flexibility in the space decomposition as shown respectively on Part (a) and (b) of Figure 3.

3.2 Point and Range Queries against NHRs

Remind that one of the PPDP requirements is to provide user-friend descriptions of anonymous data set to ease point and range searching in very simple but popular environments such like spreadsheets. Remind that point queries and orthogonal range queries both have the property of being *decomposable into dimensional filters*. HR-based BPAMs are obviously tailored to fulfill such requirement. We argue that

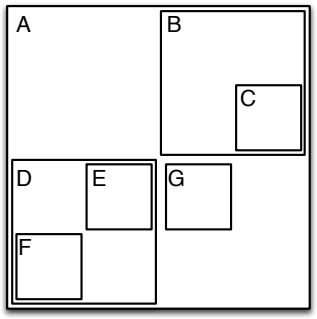

Figure 4: Example of a 2D data space partitioned with NHR-based BPAM into 7 nested regions $\{A, B, C, D, E, F, G\}$.

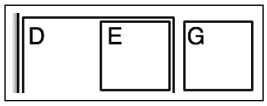

Figure 5: Example of a sub-space spanned by a range query on the partitioning of Figure 4.

anonymous public releases built with NHR-based BPAMs could also support point and orthogonal range queries, without disregarding quality and efficiency of the anonymization process.

Then, we present a first attempt to resolve point and orthogonal range queries against NHRs within a tabular representation of anonymous records. To this end, each equivalence class of the public release is encoded by its enclosing hyperrectangular region such that nested regions are allowed in the table. And the partitioning level of each region is provided in an additional column. Hence, it becomes very easy to process point queries in the anonymous table:

1. define filters on each dimension;
2. rank the intermediate result on decreasing region level;
3. keep only the records with the lowest value on the region level.

The above procedure works since the intermediate result returns nested regions only, where the innermost region is the right answer. Then, comparing levels suffices to remove false positives that are enclosing regions. For instance, a point query in block E on Figure 4 returns the intermediate result set $\{(A, 0), (D, 1), (E, 2)\}$. Then, since levels are 0, 1, 2 resp. for A, D and E, the remaining block is E and the answer of point query is the set of records from bucket $[E]$ assigned to region E.

Orthogonal range searching is slightly more difficult to manage. Indeed, if we follow the above point query process, defining range filters rather than exact match filters, then we are left with *false negatives* since enclosing regions could be partly covered by the range query. At the contrary, if we stop at step 2, then there could be *false positives* in the answer set.

Then, we propose the following methodology to manually perform orthogonal range searching in anonymous public releases. The query is first decomposed into *elementary range queries* that cover the entire query space with small cuboids that correspond to the finest resolution of regions in the public release. The resolution can be determined by means of the highest level value. Obviously, the resolution depends on the K value for a given public release. Then, each elementary range query is performed in the same way than point queries, except that filters on dimensions are ranges rather than exact matching. Finally, the answer set is the union of all the elementary range query results.

For instance, assume a range query Q that spans the subspace of A shown on Figure 5. Step 2 of point queries with range filters returns the intermediate result set $\{(A,0),(D,1),(E,2),(G,2)\}$, whereas step 3 gives $\{(E,2),(G,2)\}$. In the former result set, $(A,0)$ is a false positive, and in the later result set, $(D,1)$ is a false negative. To fix this wrong behavior, the above methodology for range searching first decomposes the query into 3 elementary queries $Q_{(1)}$, $Q_{(2)}$ and $Q_{(3)}$ that span respectively the sub-part of D, region E and region G. Values of dimensional filters are given by the examination of bounds in each column of the equivalence classes. Next, $Q_{(1)}$ is computed as a point query (with range filters) and gives the intermediate result set $\{(A,0),(D,1)\}$. Then the answer is $[D]$. The process is repeated for $Q_{(2)}$ and $Q_{(3)}$ and it returns resp. $[E]$ and $[G]$. Union of the 3 result sets is the answer to Q.

Obviously, the BPAM tree directory remains available for very large data sets and could be used as a regular database access method for any kind of range queries over the leaves of the index structure (the M-anonymous release).

4. DISCUSSION

4.1 Extension to other generalization models

K-anonymity is the very first generalization principle to achieve sanitization of data. Its main purpose is to prevent from identity disclosure. Other more sophisticated models have emerged in recent years to overcome shortcomings of K-anonymity, especially the attribute disclosure risk. Among the most popular generalization principles are ℓ-*diversity* [17] and t-*closeness* [16].

As stated in [12] for the R^+-tree, any BPAM would be able to incorporate constraints from the definition of the various existing generalization models in its anonymization process. The only accommodation would be to redefine the assignment and splitting strategies such that both resulting blocks satisfy the generalization model. For instance, to make the anonymous release ℓ-diverse, it requires that at least ℓ sensitive values are "well represented" in each equivalence class. Thus, the algorithm would incorporate checking on sensitive values in its splitting decision to only create new ℓ-diverse blocks from old ones. And it would add constraint on assignment of a new point into an existing block such that the resulting block still satisfies the ℓ-diversity, otherwise the algorithm would locally redistribute points into blocks.

Many other models of anonymization exist in the literature, such like the ε-differential privacy [7] to prevent from probabilistic attacks, but they do not use generalization to preserve privacy of records and they are far from practical approaches for most of the current anonymization requirements in real-life. For a comprehensive review of anonymization models and algorithms, the reader is invited to refer [8].

4.2 Compaction procedure

In [12], the authors propose a *compaction procedure* that simply shrinks the envelop of each block to its MBB as shown on Figure 2. Consequently, the average volume of the blocks is minimized.

The R^+-tree approach natively computes such MBBs for every block. However, top-down NHR-based BPAMs operate a decomposition of the space such that the union of all the blocks spans the entire space. Obviously, a compaction of each block would yield to a more accurate anonymous public release, and would still increase its quality. Thus, it can be considered as a straightforward improvement of top-down NHR-based BPAMs, even if computation of non hyperrectangular "MBB" such like those on Figure 3 must be carefully defined first.

5. CONCLUSIONS

In this communication, we advocated the use of Bucketed Point Access Methods for Privacy-Preserving Data Publishing tasks. We focused on the K-anonymity generalization model for data anonymization. We first reviewed the existing approaches based on multidimensional point partitioning. Then, we presented an almost comprehensive list of PAMs eligible to the PPDP task. We argued that Nested HyperRectangle-based BPAMs are the most promising structures to support PPDP. Then, we considered decomposable point and range queries against tabular representation of anonymous public releases, and we proposed a first attempt to answer such queries. Finally, we discussed about obvious extensions to various generalization models and compaction of NHRs to Minimum Bounding Boxes.

As a follow-on of that study, there could be many tracks to investigate. First, it is obvious that several instances of NHR-based BPAMs must be implemented and compared for K-anonymization. Next, extensions to other generalization models could be experimented as well. Besides, the development of a toolbox that supports search and analysis of nested hyperrectangular anonymous records would be highly valuable. Indeed, we argue that those shapes are the most sophisticated ones that could be managed by regular end-users. And finally, the definition of a compaction procedure for NHRs is also an interesting problem in the field of computational geometry.

6. REFERENCES

[1] P. K. Agarwal. Range searching. In *Handbook of Discrete and Computational Geometry*, pages 575–598. CRC Press, Inc., 1997.

[2] G. Aggarwal, R. Panigrahy, T. Feder, D. Thomas, K. Kenthapadi, S. Khuller, and A. Zhu. Achieving anonymity via clustering. *ACM Transactions on Algorithms*, 6(3):49:1–49:19, July 2010.

[3] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proc. of the 21th Int. Conf. on Data Engineering (ICDE)*, pages 217–228,

2005.

[4] J.-W. Byun, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In *Proc. of the 12th Int. Conf. on Database Systems For Advanced Applications (DASFAA)*, pages 188–200, Bangkok, Thailand, April 9-12 2007.

[5] C.-C. Chiu and C.-Y. Tsai. A k-anonymity clustering method for effective data privacy preservation. In *Proc. of the 3rd Int. Conf. on Advanced Data Mining and Applications (ADMA)*, pages 89–99, 2007.

[6] M. L. Damiani, E. Bertino, and C. Silvestri. The probe framework for the personalized cloaking of private locations. *Transactions on Data Privacy*, 3(2):123–148, 2010.

[7] C. Dwork. Differential privacy. In *Int. Colloquium on Automata, Languages and Programming (ICALP)*, volume 2, pages 1–12, Venice, Italy, July 2006. Springer Verlag.

[8] B. Fung, K. Wang, R. Chen, and P. Yu. Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys*, 42(4), June 2010.

[9] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: Anonymizers are not necessary. In *Proc. of the ACM Int. Conf. on Management Of Data (SIGMOD)*, pages 121–132, Vancouver, June 10-12 2008.

[10] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the 1st Int. Conf. on Mobile Systems, Applications and Services (MobiSys)*, pages 31–42, 2003.

[11] T. Hey, S. Tansley, and K. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*, chapter Jim Gray on eScience: A Transformed Scientific Method. Microsoft Research, 2009.

[12] T. Iwuchukwu and J. Naughton. K-anonymization as spatial indexing: Toward scalable and incremental anonymization. In *Proc. of the 33rd Int. Conf. on Very Large Data Bases (VLDB)*, pages 746–757, 2007.

[13] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *Proc. of the ACM Int. Conf. on Management Of Data (SIGMOD)*, pages 217–228, 2006.

[14] B. Krishnamachari, G. Ghinita, and P. Kalnis. Privacy-preserving publication of user locations in the proximity of sensitive sites. In *Proc. of the 20th Int. Conf. on Scientific and Statistical Database Management (SSDBM)*, LNCS, pages 95–113, Hong Kong, China, July 9-11 2008. Springer.

[15] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proc. of the 22nd Int. Conf. on Data Engineering (ICDE)*, pages 25–35, 2006.

[16] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proc. of the 23rd Int. Conf. on Data Engineering (ICDE)*, pages 106–115, 2007.

[17] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[18] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings of the 23rd ACM Symposium on Principles of Database Systems (PODS)*, France, June 14-16 2004.

[19] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *Proc. of the 32nd Int. Conf. on Very Large Data Bases (VLDB)*, pages 763–774, 2006.

[20] A. Pilevar and M. Sukumar. GCHL: A grid-clustering algorithm for high-dimensional very large spatial data bases. *Pattern Recognition Letters*, 26(7):999–1010, May 15 2005.

[21] H. Samet. *Foundations Of Multidimensional And Metric Data Structures*. Morgan Kaufmann, 2006.

[22] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[23] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, 10(5):557–570, 2002.

[24] W. Wang, J. Yang, and R. R. Muntz. STING: A statistical information grid approach to spatial data mining. In *Proc. of the 23rd Int. Conf. on Very Large Data Bases (VLDB)*, pages 186–195, 1997.

[25] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. Fu. Utility-based anonymization using local recoding. In *Proc. of the 12th ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, pages 785–790, 2006.