It's too noisy in here: using projection to improve Differential Privacy on RDF graphs

Sara Taki[1], Cédric Eichler[1], and Benjamin Nguyen[1]

INSA Centre Val de Loire, Laboratoire d'Informatique Fondamentale d'Orléans, Bourges, France
{sara.taki,cedric.eichler,benjamin.nguyen}@insa-cvl.fr

Abstract. In the last decade, adaptation of differential privacy to graph data has received growing attention. Most efforts have been dedicated to unlabeled homogeneous graphs, while labeled graphs with an underlying semantic (e.g. RDF) have been mildly addressed.
In this paper, we present a new approach based on graph projection to adapt differential privacy to RDF graphs, while reducing query sensitivity. We propose three edge-addition based graph projection methods that transform the original RDF graph into a graph with bounded degree, bounded out-degree, and bounded typed-out-degree. We demonstrate that these projections preserve neighborhood, allowing to expand the domain of any differentially private algorithm from graphs with bounded (out/typed-out) degree to any arbitrary RDF graph. Experimental and analytical evaluation through a realistic twitter use-case shows that projection can provide two orders of magnitude of utility improvement.

Keywords: Differential Privacy · RDF · SPARQL · Graph projection.

1 Introduction

RDF [11] is a standard way to model semantic (or linked) data. An RDF data set is a set of triples (subject-predicate-object) which form a labeled directed graph. The use of Linked Data is increasing, and thus privacy in such data sources is becoming an issue [3]. Indeed, directly publishing graph data may result in disclosure of sensitive information and therefore to privacy violations.

Differential privacy [4] (DP) is currently one of the most popular and prevalent definitions of privacy. In the last decade, adapting differential privacy to graphs has received growing attention. However, most efforts have been dedicated to unlabeled, homogeneous graphs, while labeled graphs with an underlying semantic have seldom been addressed. This original type of graph is our focus in this paper. It is important to note that many queries are highly sensitive to small modifications of the original graph, which means directly using differential privacy to perturb the query results is a bad option.

Contribution. In this paper, we propose a new approach based on graph projection to adapt differential privacy to edge-labeled directed graphs, i.e. RDF graphs, while reducing the sensitivity of different kinds of queries. We consider

three different privacy definitions : node privacy, outedge privacy and typed edge privacy. The main idea behind our approach is to use *graph projection* within the DP mechanisms in order to reduce the sensitivity of queries. For projections to be adequate w.r.t. the privacy definitions, we propose three edge-addition based graph projection methods that transform the original RDF graph into a graph respecting one of the following constraints: bounded degree, bounded out-degree and bounded QL-out-degree. We evaluate our contribution analytically and experimentally w.r.t. a real twitter use-case, showing significant improvement over a naive approach without projection. The rest of the paper is organized as follows. Fundamental concepts of differential privacy are introduced in Sec. 2. Sec. 3 surveys related work and introduces the neighborhood definitions associated to the considered privacy models. Our approach and contributions are described in Sec. 4. Sec. 5 presents an analysis of the approach. We finally conclude and point some future works in Sec. 6.

2 Background : Differential Privacy

This section provides core background about DP, originally introduced by Dwork in 2006 [5,4]. Due to its formal privacy guarantees, DP has emerged to be the flagship of data privacy definitions nowadays.

2.1 Definition of Differential Privacy

Intuitively, the goal of DP is to ensure that an attacker is not able to infer (beyond a certain probabilistic threshold) whether an individual contributed to the result of a query over a database. The exact protection and the notion of individuals' contributions are defined based on the concept of neighboring (or adjacent) databases. Given a distance on databases, we say that two databases are neighbors (or adjacent) if they are at distance 1. We will discuss the metrics we use to define adjacent databases in Sec. 3.1. In this section, we note d the distance on the considered space.

An algorithm is differentially private if it is likely to yield the same output on neighboring databases. The robustness of DP is quantified by a positive parameter ϵ, called privacy budget. The basic introduction of differential privacy [5,4] considered databases that are sets or arrays. In this case, each individual information corresponds to a database entry and this entry can be modified without impacting other entries. For a comprehensive overview of concepts and definitions, we refer to [4]. In what follows, we adopt the definition by [9].

Definition 1 (ϵ, δ-differential Privacy). *A randomized mechanism K: $D^n \rightarrow \mathbb{R}^k$ preserves (ϵ, δ)-differential privacy if for any pair of databases (x, y) $\in (D^n)^2$ such that d(x,y) =1, and for all sets S of possible outputs:*

$$Pr[K(x) \in S] \leq e^{\epsilon} Pr[K(y) \in S] + \delta \quad (1)$$

where the probability is taken over the randomness of K.

In what follows, we consider ϵ-DP wich is $(\epsilon, \delta) - DP$ with $\delta = 0$.

2.2 Noise Calibration

One way of achieving DP for a query q is to add to its results an appropriate amount of noise, calibrated by the global sensitivity of q. *Global sensitivity* (GS) measures the maximal variation of the query result when evaluated upon any two neighboring databases. GS depends only on q, d, and the considered space of databases.

Definition 2 (Global Sensitivity (GS) [4]). *For $f : D^n \rightarrow \mathbb{R}^k$ and all (x,y) $\in (D^n)^2$, the global sensitivity of f is*

$$\Delta f = \max_{x,y:d(x,y)=1} \parallel f(x) - f(y) \parallel_1 \tag{2}$$

where $\|\|_1$ denotes the L1 norm.

One way to satisfy DP is to add noise to the output of a query.

Theorem 1 (Laplace Mechanism [5]). *In the Laplace mechanism, in order to publish f(x) where f :$D^n \rightarrow \mathbb{R}$ and $x \in D^n$ while satisfying ϵ-DP, one publishes*

$$K(x) = f(x) + Lap(\Delta f/\epsilon) \tag{3}$$

where Lap (Δf /ϵ) represents a random draw from the Laplace distribution centered at 0 with scale Δf /ϵ. The Laplace distribution centered at μ with scale b being the distribution with probability density function

$$h(x) = \frac{1}{2b} exp(\frac{-|x-\mu|}{b}) \tag{4}$$

3 Related Work

In this paper, we propose a novel approach to construct DP-mechanisms over RDF graphs. In this section, we provide an overview of DP over graphs and privacy preserving RDF querying.

3.1 Models and distances for DP on Graphs

When using DP, the privacy model is tightly related to a distance on the considered database-space. On graph data, two distances are classically adopted: edge-DP and node-DP [7]. *In node-DP*, neighboring graphs are defined as graphs that differ by one node and all its incident edges. Node-DP represents the strongest privacy model for graphs. It protects the contribution of a node and all of its incident edges. This means that by observing the result of a node-DP mechanism over a database x, an attacker may not, with statistically significant confidence, infer whether a particular node or any of its incident edges is in x. In the case of RDF, an incident edge represents a triple involving the node as subject or object. *In edge-DP*, neighboring graphs are defined as graphs that differ by at

most one edge. Edge-DP is the weakest graph privacy model. It only protects the contribution of a single edge.

While edge-DP is usually considered weak, the sensitivity of many queries under node-DP is high and sometimes unbounded. This will degrade the utility (i.e. accuracy of the query answer), or even make it impossible to construct a DP-mechanism. In what follows, we consider other privacy models we believe to be reasonable in the context of RDF and should result in better utility.

Outedge DP [16] was introduced in the context of social networks. Its adaptation to the context of edge-labeled directed graphs is straightforward. This privacy model protects *all* the outedges of a node. In the context of RDF, this means protecting all the triples a node is the subjet of.

QL-Outedge DP [14] was introduced for edge-labeled directed graphs. It is similar to outedge privacy but considers edges' semantics by only protecting edges of a given set QL (i.e. *sensitive* labels).

Notations. An edge-labeled directed graph is a graph G = (V, E) where V is a set of vertices, E is a set of edges such that E $\subseteq$ V $\times$ L$\times$ V, with L the set of possible edge labels. We note $\mathcal{G}$ the set of such graphs.

Formal definitions for the considered distances (related to node, outedge, and QL-outedge privacy) between two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are reported in Table 1.

Notation	Definition
d_n (node)	$\lvert V\rvert + \lvert V'\rvert$ where $V = (V_2 \cup V_1) \setminus (V_2 \cap V_1)$ and $V' \subseteq (V_1 \cap V_2)$ is the smallest set such as the subgraph of G_1 induced by $V_1 \setminus (V \cup V')$ is equal to the subgraph of G_2 induced by $V_2 \setminus (V \cup V')$
d_O (outedge)	∞ if $V_1 \neq V_2$ else $\lvert V\rvert$ where $V = \{v \in V_1 \lvert \exists l \in L_1, \exists u \in V_1 : (v,l,u) \in (E_1 \cup E_2) \wedge (v,l,u) \notin (E_1 \cap E_2)\}$
d_{QL} (QL-outedge)	∞ if $V_1 \neq V_2 \vee \exists (u,v) \in (V_1 \cap V_2)^2, \exists l \in L \setminus QL$ such that $(u,l,v) \in (E_1 \cup E_2) \wedge (u,l,v) \notin (E_1 \cap E_2)$ else $\lvert V\rvert$ where $V = \{v \in V_1 \lvert \exists l \in QL, \exists u \in V_1 : (v,l,u) \in (E_1 \cup E_2) \wedge (v,l,u) \notin (E_1 \cap E_2)\}$

Table 1. Distances: notations and definitions

3.2 Applying DP on graphs

Previous studies described in [10][16] [1][13] show different approaches to work with DP in graphs. [10] present various techniques for designing node-DP algorithms for network data. The main idea is to *project* the input graph onto the set of graphs with maximum degree less than a specific threshold to bound the sensitivity of queries. It is based on a naive truncation that simply discards nodes of high degree. However, their techniques are designed for undirected labeled graphs, and not RDF, contrary to our proposed approach, which is also based on graph projection.

3.3 Privacy over RDF

Delanaux et al. [3] developed a declarative framework for anonymizing RDF graphs by using blank nodes to hide sensitive data. Anonymisation of RDF data was also studied in [12,8] where the anonymisation model is inspired by the k-anonymity model. However, k-anonymity does not provide the formal privacy guarantees that DP does.

Most of the literature related to DP on RDF datasets appears to be theoretical. In fact, to the best of our knowledge, the only work investigating DP in the context of RDF that directly provides experiments is [15]. However, they give a DP realisation via local sensitivity without the use of a smoothing function hence failing to comply with the privacy guarantees stated in [6].

4 Proposed Approach : from a subspace with low-sensitivity queries to $\mathcal{G}$

The main challenge when developing node-DP, outedge-DP, and QL-outedge-DP algorithms is that the sensitivity of many queries can be very high, or even unbounded, in $\mathcal{G}$. Consider a query that computes the maximum out-degree in a graph. Under node privacy, out-edge privacy and Ql-out-edge privacy, the global sensitivity of this query is in terms of the number of nodes in the graph, which is unbounded. Technically, many queries may have much lower sensitivity when running on a graph with bounded degree, out-degree, or QL-out-degree.

Therefore, the main idea behind our approach is *graph projection*, in order to transform the original graph G into a graph of bounded degree, out-degree, or QL-out-degree. We show that such projections can significantly reduce the sensitivity of a query and consequently the magnitude of the noise added to achieve DP. This reduction may compensate the data loss inherent to these projections, allowing them to ultimately improve utility.

In this section, we first introduce projection methods. We then introduce the necessary notations and concepts to study DP for queries within the projected space. Finally, we show how to make the whole mechanism (i.e., a projection followed by a query over the projected space) differentially private.

4.1 Proposed Projection Methods

In what follows, we propose three edge-addition based graph projection methods named T_n, T_O and T_{QL}. Projection by edge-addition was introduced by [2] for unlabeled, undirected graphs and is herein expanded to edge-labeled directed graphs. *Notations.* We note $\mathcal{G}^D$ the set of graphs with maximum degree D and $\mathcal{G}_o^D$ the set of graphs with maximum out-degree D. Finally, we note $\mathcal{G}_{QL}^D$ the set of graphs with maximum QL-out-degree D for a given QL $\subseteq L$; i.e. the set of graphs whose vertices are the source of at most D edges whose labels are in QL. Note that $\mathcal{G}_{QL}^D \subseteq \mathcal{G}_o^D \subset \mathcal{G}^D \subset \mathcal{G}$.

Projection algorithms. Projection methods T_n: $\mathcal{G} \to \mathcal{G}^D$, T_O: $\mathcal{G} \to \mathcal{G}_o^D$, T_{QL} : $\mathcal{G} \to \mathcal{G}_{QL}^D$ described in Algorithm 1 transform the original graph G into one of its sub-graphs $\tilde{G}$, such that the maximum degree, out-degree, or QL-out-degree of a node in $\tilde{G}$ is less than or equal to D.

First, the projection creates a graph with the same nodes as G but without any edges. It then tries to insert each edge of G following an *edge ordering function* –noted A– that takes a graph and outputs an ordered lists of its edges. An edge $e = (v_1, \ell, v_2)$ is successfully inserted whenever its insertion preserves the constraint, i.e. for T_n (resp. T_O, T_{QL}) inserting this edge will not raise the ***degree* of either** v_1 **or** v_2 (resp. the ***out-degree* of** v_1, the ***QL-out-degree* of** v_1) over D.

Algorithm 1: T_n, (resp. T_O; T_{QL}) : projection by edge-addition, Bound Degree (resp. Out-degree, QL-out-degree)

Input: A graph G = (V, E) $\in \mathcal{G}$, a bound D, a stable edge ordering A, (a set of labels QL $\subseteq$ L for T_{QL})
Output: An output D-degree bounded graph (resp. D-out-degree bounded graph T_O(G); D-Ql-out-degree bounded graph T_{QL}(G))

```
1  Ẽ ← ∅;
2  foreach v ∈ V do  toBound(v)←0;
3  foreach  e=(v1,l,v2) ∈ A(G) and following A's order do
4  |  if toBound(v1) < D ∧ toBound(v2) < D (resp. toBound(v1) < D; l ∈ QL
   |   ∧ toBound(v1) < D) then
5  |  |  Ẽ ← Ẽ∪ {e};
6  |  |  toBound(v1)++;
7  |  |  /* Only in Tn                                              */
8  |  |  toBound(v2)++;
9  |  end if
10 |  /* Only in TQL                                               */
11 |  if l ∉ QL then  Ẽ ← Ẽ∪ {e} ;
12 end foreach
13 return G^D = (V, Ẽ) (resp. G_o^D = (V, Ẽ), G_QL^D = (V, Ẽ))
```

Edge ordering. As seen above, the algorithm attempts to insert the edge in some predetermined order. Using a different order may produce a different result. This edge ordering must be stable in the sense that given two neighboring graphs G_1 and G_2, if two edges appear in G_1 and G_2 then their relative order must be the same in A(G_1) and A(G_2). We can construct a stable edge ordering quite easily. Indeed, as E $\subseteq$ V $\times$ L$\times$ V, a first intuition is to consider orders on the space of sources, labels, and destination (e.g. lexicographical order) and to define a total edge order by combining the three.

4.2 Privacy on Bounded (degree, Out-degree, Ql-out-degree) Graphs

Note that most concepts related to DP depend on the considered space of databases and its associated distance. Since we consider in this paper various

subspaces of $\mathcal{G}$ and various distances, this subsection introduces unambiguous notations and definitions.

Definition 3 (Restricted ϵ, δ-differential Privacy). *A randomized mechanism $K: \mathcal{G} \to S$ is $(\epsilon, \delta)_R$ differentially private over $R \subseteq \mathcal{G}$ w.r.t. a distance d over R, if for all pairs $(G_1, G_2) \in R^2$,*

$d(G_1, G_2) = 1 \implies Pr[K(G_1) \in S] \leq e^{\epsilon} Pr[K(G_2) \in S] + \delta$

By definition, the *global sensitivity* of a query is the maximum variation of its results over any neighboring graphs in the considered space.

Definition 4 (Global sensitivity on Bounded Graphs). *For any $f : \mathcal{G} \to \mathbb{R}^k$, the global sensitivity of f on $R \subseteq \mathcal{G}$, w.r.t a distance d over R is:*

$$\Delta_d^R f = \max_{(G_1,G_2) \in R^2 : d(G_1,G_2)=1} \| f(G_1) - f(G_2) \|_1 \tag{5}$$

By convention, $\Delta_d^{\mathcal{G}}$ is noted Δ_d. The sensitivity over the projected spaces can simply be seen as the sensitivity of the restriction of the original function to the projected spaces. *Considering the definitions it is trivial that for any f, R, and d, $\Delta_d^R f \leq \Delta_d f$.*

4.3 Privacy and projections

These notations having been introduced, we study in what follows the privacy guarantees of the mechanism composed by a projection followed by a query. Its sensitivity depends on the sensitivity of the projection, i.e. the maximal distance between any two neighboring graphs after projection.

Definition 5 (Global sensitivity of a projection [10]). *The global sensitivity of a projection $T: \mathcal{G} \to R$ w.r.t. a distance d over $\mathcal{G}$ and d_R over R is:*

$$\Delta_{(d,d_R)} T = \max_{(G_1,G_2) \in \mathcal{G}^2 : d(G_1,G_2)=1} d_R(T(G_1), T(G_2)) \tag{6}$$

In what follows, we assume that the same distance d is used in $\mathcal{G}$ and R, and note $\Delta_d T$ instead of $\Delta_{(d,d)} T$. The sensitivity of the composed function $f \circ T$ is bounded by the sensitivity of T times the sensitivity of f on the projected space:

Theorem 2 (Sensitivity of the composed mechanism [10]). *Given a projection $T : \mathcal{G} \to R \subseteq \mathcal{G}$, a function $f : R \to \mathbb{R}^k$, and a distance d over $\mathcal{G}$:*

$$\Delta_d(f \circ T) \leq \Delta_d^R f \times \Delta_d T \tag{7}$$

4.4 Privacy on unbounded graphs through projection

Our context involves three privacy models (node, outedge and QL-outedge) related to three distances and three projections introduced in Sec. 3.1 and 4.1, respectively. In principle, one would want to study the global sensitivity of each projection w.r.t. all distances. However, there is an obvious relation between projection and distances and we believe that a preliminary study may be restricted to (i) $\Delta_{d_n} T_n$, (ii) $\Delta_{d_O} T_O$ (iii) $\Delta_{d_{QL}} T_{QL}$. Moreover, in this paper, we focus on (ii) and (iii) for space reasons, as they seem more interesting to us in the context of RDF.

Lemma 1 (Global sensitivity of T_O). $\Delta_{d_O} T_O = 1$

We omit the proof due to space restriction. Intuitively, by removing or adding all the out-edges of some node v, v is the only impacted node in term of outdegree. Since the constraint of T_O concerns outdegree, and since the edge ordering is stable, outedges of other nodes are handled in the same way by T_O, the sole difference between the two projected graphs being the out-edges of v. Therefore, the projected graphs are still neighbors w.r.t. d_O.

Lemma 2 (Global sensitivity of T_{QL}). $\Delta_{d_{QL}} T_{QL} = 1$

The proof is similar to Lemma 1 since in the worst case scenario, outedge and QL-outedge are identical.

Hence, the global sensitivity of T_O and T_{QL} w.r.t. their related distance (d_O and d_{QL}, respectively), is 1. Therefore:

- They preserve neighborhood, i.e., the projection of two neighboring graphs through the use of T_O and T_{QL} results in two neighboring graphs (w.r.t. d_O and d_{QL}, respectively).
- According to Thm. 2, for any function f, the global sensitivity of the composed mechanism is no greater than the global sensitivity of f over the projected space (w.r.t. d_O and d_{QL}, respectively).

It directly follows that any algorithm DP on $\mathcal{G}_o^D$, or $\mathcal{G}_{QL}^D$ can be transformed into an algorithm DP on $\mathcal{G}$ without any extra privacy budget (i.e. while preserving ϵ).

Proposition 1. *Given any mechanism f whose domain is $\mathcal{G}_o^D$ (resp. $\mathcal{G}_{QL}^D$), if f is ϵ-DP w.r.t. d_O (resp. d_{QL}) then $f \circ T_O$ (resp. $f \circ T_{QL}$) is ϵ-DP on $\mathcal{G}$ w.r.t. d_O (resp. d_{QL}).*

Proof is immediate considering that T_{QL} and T_O preserve neighborhood according to Lemma 1 and 2.

5 Analytical and Experimental Evaluation

This section introduces metrics to analytically evaluate the approach and confront them to a real use-case. Analytical expectations are experimentally confirmed, demonstrating the feasibility and interest of the approach. The evaluation

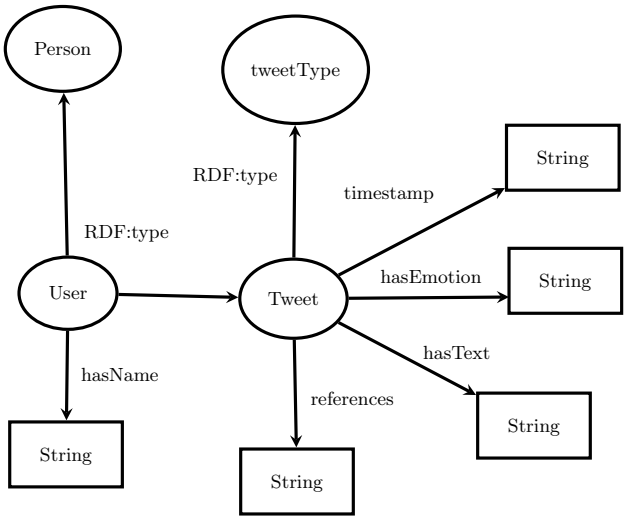

Fig. 1. RDF schema for Sentiment140

relies on the Sentiment140 dataset composed of 1.6 million tweets[1], which we have parsed and serialized in RDF/XML format. Its schema is shown in Fig. 1. Experiments are conducted using Apache Jena to run SPARQL queries and use our Java 1.8 implementation of the projection algorithms.

5.1 Metrics: Utility and Information Loss

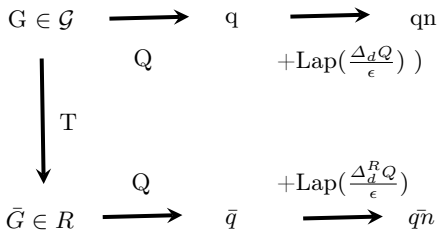

Fig. 2. Available data for evaluation : overview

Figure 2 provides an overview of the functions and values considered during our evaluation w.r.t. a query Q, a projection T, and a distance d. To simplify, we consider that $Q : \mathcal{G} \to \mathbb{R}$ and the laplacian mechanism is used to achieve DP.

We are interested in particular in evaluating: (1) the overall utility loss due to privacy, comparing $\bar{qn}$ and q; (2) the information loss due to projection, by

[1] https://www.kaggle.com/kazanova/sentiment140

comparing q and $\bar{q}$; (3) the interest of projection w.r.t. providing a DP answer without projection, by comparing qn and $\bar{qn}$. To this end, we propose the following novel metrics:

1. Expected utility loss E, which is the expected difference between $\bar{qn}$ and q. E $= \int_0^\infty xG(x)\,dx$ with $G(x)$ the probability of answering $\bar{qn}$ such as $|\bar{qn} - q| = x$:

$$E(x) = \int_0^\infty x(\frac{1}{2b}exp(\frac{-|q-x-\bar{q}|}{b}) + \frac{1}{2b}exp(\frac{-|q+x-\bar{q}|}{b})) \\ = b * exp(\frac{-(q-\bar{q})}{b}) + q - \bar{q} \tag{8}$$

   with b $= \frac{\Delta_d^R}{\epsilon}$.
2. Information loss due to projection is defined as

$$Ploss = \frac{|q-\bar{q}|}{q} \tag{9}$$

Regarding the third point –interest of projection– note that if $\Delta_d Q = \infty$ we can either consider that DP is impossible on the original space, or that qn should be considered pure noise (uniformly random over $\mathbb{R}$), which make the interest of projection immediate. In what follows, we also compare the approach with a naive one consisting in using restricted DP without projection.

5.2 Considered query and interest of the approach

We consider here a query Q over the Sentiment140 dataset that counts the number of users user "Garythetwit" has referenced. This query leverage the dataset's semantic, refers to a path of size greater than 1, and showcase several interesting properties. The SPARQL query is provided in Listing 1.1.

Listing 1.1. Query in SPARQL - # users references by Garythetwit

```
Select (count(?referencedUser) as ?c)
WHERE { <http://rdfanon.org/types#Garythetwit>
<http://rdfanon.org/types#tweeted> ?tweet.
?tweet <http://rdfanon.org/#references> ?referencedUser.}}
```

On the original dataset, Q outputs 55, $q = 55$.

Interest of the approach. It is immediate that $\Delta_{d_n} Q$ and $\Delta_{d_O} Q$ are infinite. Thus, in this case, it would not be possible to construct a DP mechanism directly from the original query without reducing its sensitivity. We will see in what follows that the approach also provides significant improvement in utility if one were to construct a restricted mechanism without projection.

If neither *tweeted* nor *references* are considered sensitive, (i.e. are in QL), $\Delta_{d_{QL}} Q$ is 0 and Q do not provide any insight to an attacker. If at least one of the two is sensitive, $\Delta_{d_{QL}} Q$ is also infinite. In what follows, we consider $QL = \{references, tweeted\}$.

5.3 Projection loss

Q considers solely *tweeted* and *references* outedges. According the schema represented in Fig. 1, a node with a *tweeted* outedge has two other outedges related to its type and name. A node with *references* outedges has 4 other outedges related to its *timestamp, emotion, text* and *query term*.

Let us first consider T_O. An immediate heuristic to reduce information loss during projection is to gives priority to *tweeted* and *references* edges over the other types. Assuming a bound D greater than the maximal number of users referenced in a tweet plus 4, it is immediate that a projection bounding the outdegree of the graph to D will not have any impact on the outdegree of tweets. If the projection gives priority to *tweeted* outedge, it leads to the same data loss as a projection bounding it to $D + 2$ while giving lowest priority to *tweeted*.

Accordingly, we propose two orders over the set of edges used during projection. o_r gives priority to edges labelled *references* and $o_{r,t}$ gives priority to those labelled *references* and *tweeted*. Apart from this, they behave like the lexicographical order of the concatenation of edges' label, source, and destination. They discriminate between two prioritized edges according to the lexicographical order of their source and destination. Regarding $o_{r,t}$, the relative order between *references* and *tweeted* edges does not matter; according to the schema these type of edges originate from different type of nodes. Note that such a use of lexicographical order for non-prioritized edge labels means that *tweeted* edges have lowest rank according to o_r. According to the previous remark, and assuming that the maximal outdegree of tweets is lower than the maximal outdegree of users, there exists a couple (D_m, D_M) such that for any D, $D_m \leq D \leq D_M$ Ploss of T_O with $o_{r,t}$ and D is equal to Ploss of T_O with o_r and $D + 2$.

Regarding T_{QL}, since edges that are neither *references* nor *tweeted* do not count during projection, there is no difference between o_r and $o_{r,t}$ and in both cases T_{QL} behaves like T_O with $o_{r,t}$.

These considerations are confirmed by our experimental evaluation reported in Table 2. Experimental bounds go from 2 to 560, meaning that we preserve at minimum up to 2 (sensitive) outedges per node and at most 560. $D = 2$ obviously leads to a inoperable database with a information loss of 1. 560 is an extremal. Indeed, the maximum out-degree of a node in the dataset is 551, therefore, a projection with $D = 560$ leads to 0 modification and 0 information loss. As expected:

1. T_O w.r.t. the order $o_{r,t}$ is equivalent to T_{QL} with $QL = \{\textit{references}, \textit{tweeted}\}$
2. T_O w.r.t. o_r leads to slightly more loss than these two projections with small bound: from 1 to 0.96 and 0.9 to 0.89 information loss with D equal to 4 and 10, respectively.

Note that Ploss with $D = 50$ is $0,27$, $\bar{q}$ in this case being 40. Indeed, the projections keep the edge between *Garythetwit* and some of its edges that do not contain any reference. An optimal projection would lead to a $\bar{q}$ between D (worst case scenario where each tweet contains a single reference) and D^2 (there exists at least D tweets with at least D references).

Table 2. Ploss for Q

Degree bound D	2	4	10	50	500	560
Ploss TO wrt o_r	1	1	0.9	0.27	0	0
Ploss TO wrt $o_{r,t}$	1	0.96	0.89	0.27	0	0
Ploss T_{QL}	1	0.96	0.89	0.27	0	0

5.4 Overall utility of the approach

Here, $\Delta^{\mathcal{G}^D_{QL}}_{d_{QL}} Q = \Delta^{\mathcal{G}^D_o}_{d_O} Q = D^2$, i.e. the sensitivity of Q w.r.t. d_{QL} (resp. d_O) restricted to the space of graphs with maximal QL-out-degree (resp. out-degree) D is D^2. Note that this is quite pessimistic and does not consider the schema of the database. Indeed, we consider that for any value of D, D tweets can reference D users each. In reality and due to character limits, the number of users referenced in a tweet is limited. Considering the database schema and constraints could lead to further reduction of the query's sensitivity over the projected space.

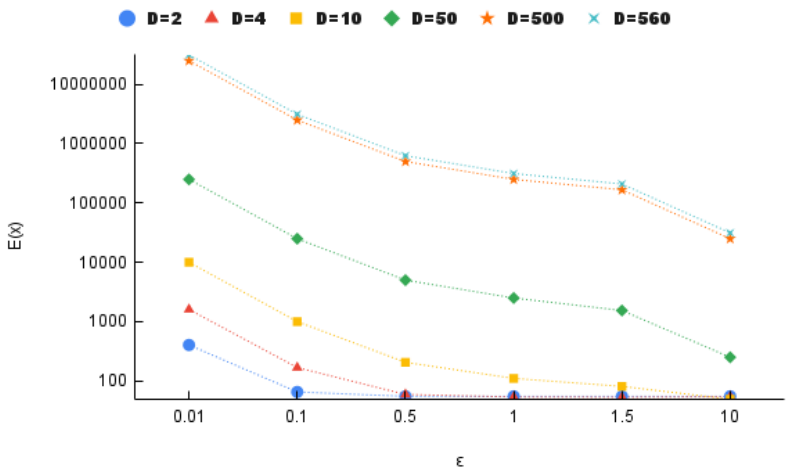

Fig. 3. Approximation of E(x) for T_{QL}/T_O w.r.t. $o_{r,t}$

We compute an approximation of E(x) by averaging the distance obtained between q and $q\bar{n}$ over 100 runs. This experimental approximation fits the analytical prevision. Since the sensitivities in the projected space are equal and since we have seen previously that the projected graphs are quite similar, we report in Fig. 3 only the analytical value for T_{QL} (which is the same as T_O w.r.t. $o_{r,t}$).

As expected, E(x) decreases while ϵ increases: utility increases as privacy guarantees weaken. More interestingly, E(x) decreases with D, meaning that *increase of information loss due to a tighter bound is compensated by the decrease in the amplitude of noise added to obtain DP guarantees.* With $\epsilon = 1$, $\frac{E(x) with D=560}{E(x) with D=50}$ is roughly 125, meaning that the expected distance between the private answer and the real value is 125 times greater with bound $D = 560$ than 50. Interestingly, as said before, $D = 560$ is an extremal case where the graph is not modified during projection. We have also seen that $\Delta_{d_O} Q = \infty$, meaning that no DP mechanism can be trivially constructed over $\mathcal{G}$. A straightforward –but somewhat weak– approach would be to construct a restricted DP mechanism over some subspace of $\mathcal{G}$, typically $\mathcal{G}_o^D$ or $\mathcal{G}_{QL}^D$ with $D = 560$. This would provide exactly the same results as our approach with $D = 560$, which provides utility several orders of magnitude worse than a regular parametrization of our approach with a bound ≤ 50.

6 Conclusion

This paper presents a new approach based on *graph projection* to adapt differential privacy to edge-labeled directed graphs –e.g. RDF graphs– while reducing the amplitude of the randomized noise.

The main idea is to use *graph projection* to reduce the sensitivity of queries. We propose three edge-addition based graph projection methods that transform an RDF graph into a graph of bounded degree, out-degree, or typed-out-degree. We show that two of these projections preserve neighborhood w.r.t. two different privacy models. Consequently, for said projections and models, the global sensitivity of the composition (query $\circ$ projection) is at most equal to the global sensitivity of the query over the projected space. Thus, we obtain a general method to expand the domain of any DP mechanism over a restricted projected space, to the space of RDF graphs. We experimentally and analytically demonstrate the feasibility and interest of the approach on a real twitter dataset w.r.t. a query with infinite sensitivity on the original space. We also show that our approach provides a utility several order of magnitude better than a naive approach relying on restricted DP without projection.

The proposed study underline the importance of two particular research directions that we plan on tackling; optimizing projection to reduce information loss and considering database schema to reduce the query sensitivity over the considered spaces. This last point is far from trivial as it implies studying DP in a space where a graph do not necessarily have neighbors and opens the possibility of the projected space to not be included in the original space.

Acknowledgements This work is supported by the French National Research Agency, under grant ANR-18-CE23-0010.

References

1. Blocki, J., Blum, A., Datta, A., Sheffet, O.: Differentially private data analysis of social networks via restricted sensitivity. In: Proceedings of the 4th conference on Innovations in Theoretical Computer Science. pp. 87–96 (2013)
2. Day, W.Y., Li, N., Lyu, M.: Publishing graph degree distribution with node differential privacy. In: Proceedings of the 2016 International Conference on Management of Data. pp. 123–138 (2016)
3. Delanaux, R., Bonifati, A., Rousset, M.C., Thion, R.: Query-based linked data anonymization. In: International Semantic Web Conference. pp. 530–546. Springer (2018)
4. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II. Lecture Notes in Computer Science, vol. 4052, pp. 1–12. Springer (2006)
5. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference. pp. 265–284. Springer (2006)
6. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science **9**(3-4), 211–407 (2014)
7. Hay, M., Li, C., Miklau, G., Jensen, D.: Accurate estimation of the degree distribution of private networks. In: 2009 Ninth IEEE International Conference on Data Mining. pp. 169–178. IEEE (2009)
8. Heitmann, B., Hermsen, F., Decker, S.: k-rdf-neighbourhood anonymity: Combining structural and attribute-based anonymisation for linked data. PrivOn@ ISWC **1951** (2017)
9. Johnson, N., Near, J.P., Song, D.: Towards practical differential privacy for sql queries. Proceedings of the VLDB Endowment **11**(5), 526–539 (2018)
10. Kasiviswanathan, S.P., Nissim, K., Raskhodnikova, S., Smith, A.: Analyzing graphs with node differential privacy. In: Theory of Cryptography Conference. pp. 457–476. Springer (2013)
11. Klyne, G., Carroll, J.J.: Resource description framework (rdf): Concepts and abstract syntax. W3C Recommendation (2004), `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`
12. Radulovic, F., García-Castro, R., Gómez-Pérez, A.: Towards the anonymisation of rdf data. In: SEKE. pp. 646–651 (2015)
13. Raskhodnikova, S., Smith, A.: Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions. arXiv preprint arXiv:1504.07912 (2015)
14. Reuben, J.: Towards a differential privacy theory for edge-labeled directed graphs. SICHERHEIT 2018 (2018)
15. Silva, R.R.C., Leal, B.C., Brito, F.T., Vidal, V.M., Machado, J.C.: A differentially private approach for querying rdf data of social networks. In: Proceedings of the 21st International Database Engineering & Applications Symposium. pp. 74–81 (2017)
16. Task, C., Clifton, C.: A guide to differential privacy theory in social network analysis. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 411–417. IEEE (2012)