Learning from Graph Structure under Differential Privacy

Rafael Pinot[1,2], Florian Yger[1], Cédric Gouy-Pailler[2], and Jamal Atif[1]

[1] Université Paris-Dauphine, PSL Research University, LAMSADE, Paris, France
[2] Institut LIST, CEA, Université Paris-Saclay, LADIS, Palaiseau, France
rafael.pinot@dauphine.fr

Abstract. This paper addresses the issue of learning from graph data under differential privacy. More precisely, we introduce a new algorithm, denoted PAMST, to compute an almost minimum spanning tree of a graph under privacy conditions. Privacy is ensured using a new setting for differential privacy in weighted graphs. It consists in assuming the graph topology $\mathcal{G} = (V, E)$ to be public and the private information to be carried by the edge weights $w : E \to \mathbb{R}$. PAMST achieves a state of the art additive approximation error of $O\left(|V|^2 \log |V|\right)$ with a $O(|V|^2)$ time complexity for fixed privacy conditions. The proposed approach theoretically, and experimentally outperforms competing methods in terms of time-complexity/accuracy trade-off. We finally illustrate the use of the proposed algorithm on a clustering application in the real world scenario of migration flows analysis.

Keywords: Differential Privacy · Minimum Spanning Tree · Clustering

1 Introduction

With pervasive uses of machine learning techniques, researchers and practitioners are observing growing concerns among individuals and organizations worrying about the security of their potentially private characteristics. Beyond primary concerns to guarantee that such information are not leaked or accidentally disclosed, a crucial issue of machine learning approaches is to ensure that such information cannot be recovered or inferred from the sole release of a statistical model learned from data. Several definitions have been introduced to characterize privacy preserving algorithms [5]. Among them, differential privacy has become the dominant standard characterization of privacy preserving algorithms by providing a formal and adaptive conception of safe data-analysis. First introduced by Dwork *et al.* [3], and further extended in [4,1,14], it states that an algorithm is differentially private if, given two "close" databases, it produces statistically indistinguishable outputs. This privacy notion is highly dependent on the notion of "closeness" one uses. Typically, in a metric space, two elements are close if the distance between them is "small enough". As Section 3 highlights,"small enough" is usually chosen according to the application at hand.

Even though it is extensively investigated in the restricted settings considering tabular databases, learning from graph databases remains challenging. Mir *et al.* [13] as well as Karwa *et al.* [9] formalized the idea of releasing statistics from a graph in a differentially private manner following the seminal work of Nissim *et al.* [16]. Several definitions of "closeness" and privacy on graphs thus appeared. The two investigated ones are edge-differential privacy [7], and node-differential privacy [10]. Conceived for the protection of the graph topology (structure), these definitions are not suitable for applications such as traffic monitoring on a static network (*e.g.* migrations as in Example 1), where the private information on the users are carried by the edge weights.

Sealfon [21] addressed this issue by providing a new formal framework for the private analysis of weighted graphs where graph topology $\mathcal{G} = (V, E)$ is public and the private information is contained in the weight function $w : E \to \mathbb{R}$. Using this framework, he was able to release an approximate minimum spanning tree with weight-approximation error of $O\left(|V| \times |E| \log |E|\right)$ for fixed privacy parameters, with time complexity $O\left(|E| + |V| \log |V|\right)$. This method, can be highly inaccurate when the graph is large, which is common in machine learning applications. A way to improve the algorithm's performances in the context of graph learning under differential privacy is to construct an iterative method which focuses on considering a *local* condition on the weight function in the process. This idea has been proposed by Gupta *et al.* [6], and extensively investigated in the framework of differentially private submodular optimization by Mitrovic *et al.* in [15]. In the latter, the issue of releasing a minimum spanning tree under privacy conditions is not directly investigated. Yet one could derive from the study of monotone submodular maximization a private version of Kruskal algorithm with an improved approximation error of $O\left(|V|^2 \log |V|\right)$. Even though the approximation error is satisfying, Kruskal algorithm in the submodular framework has an algorithmic complexity of $O(|E||V|)$ which is prohibitive when dealing with large and dense graphs.

Contributions: Our main contribution is the PAMST algorithm, designed to privately release the topology of an almost minimum spanning tree. PAMST exhibits a weight approximation error of $O\left(|V|^2 \log |V|\right)$ for fixed privacy parameters, and a time complexity of $O(|V|^2)$. This result, in contrast to previous works, enables to deal with relatively large, and dense graphs, which are frequently met in machine learning applications. Hence, PAMST outperforms former methods regarding the time-complexity/accuracy trade-off. Finally, we present a private clustering method, in order to demonstrate how to use PAMST (and previous works) in the context of unsupervised learning from graph under differential privacy. To ensure the conciseness of the body of the paper, some proofs have been pushed into the supplementary material.

2 Background on differential privacy

Let us consider some arbitrary space $\mathcal{D}$, and " $\sim$ " a closeness notion. In classical definitions of differential privacy, $\mathcal{D}$ is a set of tabular databases, where every

row characterizes an individual. In this setting two databases are said to be close if they differ by at most one individual.

Definition 1 ([3]). *A randomized algorithm $\mathcal{A}$ with domain $\mathcal{D}$ is ϵ-differentially private if for any $Z \subset Range(\mathcal{A})$ and for any $x \sim y \in \mathcal{D}$,*

$$\mathbb{P}[\mathcal{A}(x) \in Z] \leq e^{\epsilon}\mathbb{P}\left[\mathcal{A}(y) \in Z\right].$$

Where the probability space is over the coin flips of $\mathcal{A}$.

One of the first, and most used, differentially private mechanisms is the Laplace mechanism. It is based on the process of releasing a numerical query perturbed by a noise drawn from a centered Laplace distribution scaled to the sensitivity of the query, which is defined as follows.

Definition 2 ([3]). *Let us consider a numerical query $f : \mathcal{D} \to \mathbb{R}^k$, the sensitivity of f is defined as follows:*

$$\mathcal{S}_f = \max_{x \sim y \in \mathcal{D}} ||f(x) - f(y)||_1.$$

Definition 3 ([3]). *Given a numerical query $f : \mathcal{D} \to \mathbb{R}^k$, some $\epsilon > 0$, and $x \in \mathcal{D}$, the Laplace mechanism returns $f(x) + (Y_1, ..., Y_k)$, where Y_i are i.i.d. random variables drawn from Lap$(\mathcal{S}_f/\epsilon)$. Where Lap(b) denotes the Laplace distribution with scale b that is the distribution with probability density $\frac{1}{2b}\exp\left(-\frac{|x|}{b}\right)$.*

Theorem 1 ([3]). *The Laplace mechanism is ϵ-differentially private.*

Another widely used method, the Exponential mechanism, represents the typical way of answering arbitrary range queries while preserving differential privacy. Given some range of possible responses to the query $\mathcal{R}$, it is defined according to a utility function $u := \mathcal{D} \times \mathcal{R} \to \mathbb{R}$, with sensitivity $\mathcal{S}_u = \max_{r \in \mathcal{R}} \max_{x \sim y \in \mathcal{D}} |u(x, r) - u(y, r)|$. The utility function of the Exponential mechanism aims at sorting the values in $\mathcal{R}$ using the order in $\mathbb{R}$.

Definition 4 ([11]). *Given some output range $\mathcal{R}$, $\epsilon > 0$, a utility function u, and an $x \in \mathcal{D}$, the Exponential mechanism* $\mathrm{EX}(x, u, \mathcal{R}, \epsilon)$ *selects, and outputs an element r from $\mathcal{R}$ with probability proportional to* $\exp\left(\frac{\epsilon u(x,r)}{2\mathcal{S}_u}\right)$.

As the following theorem states, sampling from such a distribution preserves ϵ-differential privacy.

Theorem 2 ([11]). *For any non-empty range $\mathcal{R}$, the Exponential mechanism preserves ϵ-differential privacy.*

Any private mechanism has to achieve a trade-off between privacy and accuracy. Theorem 3 highlights this trade-off for the Exponential mechanism.

Theorem 3 ([11]). *For any $x \in \mathcal{D}$, non-empty output range $\mathcal{R}$, $\epsilon > 0$, and utility function u, the following assertions hold*

1) $\forall t \in \mathbb{R}$, with probability at most $\exp(-t)$, $u(x,r) \leq OPT_u(x) - \frac{2\mathcal{S}_u}{\epsilon}(t + \ln|\mathcal{R}|)$.
2) $\mathbb{E}[u(x,r)] \geq OPT_u(x) - \frac{2\mathcal{S}_u}{\epsilon}\ln|\mathcal{R}|$.

With $r = \mathrm{EX}(x, u, \mathcal{R}, \epsilon)$*, and* $OPT_u(x) = \max_{r\in\mathcal{R}} u(x,r)$.

The strength of differential privacy lies in its ability to comply with the followings: composition and post-processing.

Theorem 4 (Composition [2]). *For any $\epsilon > 0$ the adaptive composition of k ϵ-differentially private mechanisms is $k\epsilon$-differentially private.*

Theorem 5 (Post-Processing [4]). *Let $\mathcal{A} : \mathcal{D} \to B$ be a randomized algorithm that is ϵ-differentially private, and $h : B \to B'$ a (possibly randomized) mapping. Then $h \circ \mathcal{A}$ is ϵ-differentially private.*

In the sequel, we treat the class of simple-undirected-weighted graphs, which we simply denote weighted graphs. Section 3 presents, and discusses our new setting. Nevertheless, our work could naturally extend to more complex structures.

3 Differential privacy on graphs: a new setting

3.1 Notations

Let $G = (V, E, w)$ be a weighted graph with a vertex set V, an edge set E, and a weight function $w : E \to \mathbb{R}$. We call $\mathcal{G} = (V, E)$ the topology of the graph, and $\mathcal{W}_E$ denotes the set of all possible weight functions mapping E to the weights in $\mathbb{R}$. Finally, for any $w, w' \in \mathcal{W}_E$, we consider the following metrics:

1. ℓ_∞ metric: $d_{\mathcal{W}_E,\infty}(w, w') := \max_{e\in E} |w(e) - w'(e)|$.
2. ℓ_1 metric: $d_{\mathcal{W}_E,1}(w, w') := \sum_{e\in E} |w(e) - w'(e)|$.

3.2 Motivating example

The model of privacy presented in this section is close to the one in [21], yet we consider a different closeness notion. In fact, we choose to rely on the ℓ_∞ metric to compare graphs instead of the ℓ_1 metric. First, to fully understand the need to use a new closeness semantic, we present hereafter a motivating example in which the ℓ_∞ semantic emerges.

Example 1. A United Nation analyst wants to produce a study of the worldwide migration flows from a classical tabular database where each individual is characterized by the countries he/she traveled through (as a migrant) during the year. As illustrated in Figure 1, every row of such a dataset might not have the same length. In order to produce a clear high level study, the analyst sketches

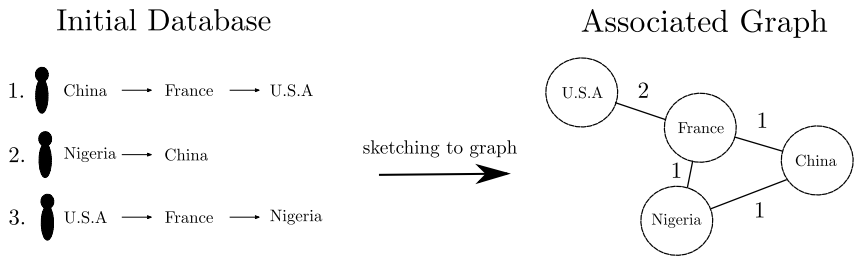


Fig. 1: Toy example of sketching a database into a graph. Each country is represented by a node, and each migration road by an edge. The weight function returns the number of individuals passing through the road.

the table as a weighted graph $G = (V, E, w)$, where countries are vertices, and edges are migration roads. The associated weight function $w : E \to \mathbb{R}$ returns for any edge, the count of individuals passing, one way or the other, through the corresponding road (Figure 1 presents a toy example of such a setting).

This framework is a typical use case in which the information, carried by the weight, must be protected to ensure the security of the individuals. Hence it makes sense to use a privacy setting were, the graph topology is public and the graph weights are private. In this kind of setting, the change of one individual in the "initial database" (see Figure 1) could change the weight of each edge in the "associated graph" by at most 1. The natural way of formalizing "each weight can be modified by at most 1" is to bound the ℓ_∞ metric by one. Hence our new setting. This example refers to the worldwide population flows, but this new setting is usable in many other application settings *e.g.* WWW traffic.

3.3 A new privacy setting

One can define differential privacy on weighted graphs either by using ℓ_∞ or ℓ_1 metrics as a closeness notion. For now, even for traffic flows monitoring (which looks a lot like Example 1), previous works use ℓ_1 distance. In this paper, thanks to Section 3.2, we claim that one should use ℓ_∞ instead. Even if our setting is different from the work in the literature, we still want to be able to compare to them (mainly [21]). We then proceed as follows: let one considers the setting presented in Section 3.2, the ℓ_1 sensitivity of the graph is $|E|$ (if every weight is shifted by 1 the final change in ℓ_1 distance is of $|E|$), while its ℓ_∞ sensitivity is 1. Therefore, we adapt the initial bound in ℓ_1 setting by simply considering the ℓ_1 sensitivity of the graph to be $|E|$. Section 4 also presents a comparison with Private Kruskal algorithm from [15] which is presented in a fully general setting. Hence, it adapts to ℓ_∞ setting without further considerations. For the sake of clarity, we restate the definitions of differential privacy, and Exponential mechanism in our graph setting.

Definition 5. *Let $\mathcal{G} = (V, E)$ be a graph topology, let $\mathcal{A}$ be a randomized algorithm taking as input a weight function $w \in \mathcal{W}_E$. $\mathcal{A}$ is ϵ-differentially private on $\mathcal{G} = (V, E)$ if for any two weight functions w, w' s.t $d_{\mathcal{W}_E,\infty}(w, w') \leq 1$, and for any set of possible outputs Z, one gets $\mathbb{P}[\mathcal{A}(w) \in Z] \leq e^{\epsilon}\mathbb{P}[\mathcal{A}(w') \in Z]$.*

Definition 6. *Given a graph topology $\mathcal{G} = (V, E)$, $\mathcal{R} \subset E$, $\epsilon > 0$, $u_{\mathcal{G}} := \mathcal{W}_E \times \mathcal{R} \to \mathbb{R}$, and $w \in \mathcal{W}_E$, the Exponential mechanism $\mathrm{EX}(\mathcal{G}, w, u_{\mathcal{G}}, \mathcal{R}, \epsilon)$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\epsilon u_{\mathcal{G}}(w,r)}{2\mathcal{S}_{u_{\mathcal{G}}}}\right)$.*

Since Section 4 presents a Prim-like algorithm, the interested reader can refer to [18] for more details. Moreover, to compare the algorithms, an additive weight-approximation error of the graph topology is used. This error is computed according to the difference between the underlying weight of the tree topology (sum of the edges weights) and the weights of the minimum spanning tree (MST), using the initial weight function.

4 Private Minimum Spanning Tree

4.1 A new algorithm and bounds on the accuracy/privacy trade-off

PAMST is a new algorithm that, given a weighted graph $G = (V, E, w)$, outputs a spanning tree topology with an almost minimum weight under differential privacy constraints. It relies on a Prim-like MST construction combined with an iterative use of the Exponential mechanism. It is thus necessary to define which utility function this mechanism relies on at each step. Let G be a weighted graph, $\mathcal{G}$ its topology, and $\mathcal{R}$ a set of edges from E. The used utility function writes as

$$\begin{aligned} u_{\mathcal{G}} : \mathcal{W}_E \times \mathcal{R} &\to \mathbb{R} \\ (w, r) &\mapsto -|w(r) - \min_{r' \in \mathcal{R}} w(r')|. \end{aligned}$$

PAMST outputs the topology of a spanning tree whose weight is almost minimal, according to the weight approximation error. The rationale behind the security guarantees of PAMST is that the Exponential mechanism, by using a utility function to define a preorder on $\mathcal{R}$, transfers the private information contained in $\mathcal{R}$ to an order in $\mathbb{R}$. The following theorems present the utility, and the security guarantees of our new method in the weighted graph setting.

Theorem 6. *Let $\mathcal{G} = (V, E)$ be the topology of a weighted graph, then $\forall \epsilon > 0$, $\mathrm{PAMST}\left(\mathcal{G}, u_{\mathcal{G}}, \cdot, \frac{\epsilon}{|V|-1}\right)$ is ϵ differentially private.*

Proof. Let $w \in \mathcal{W}_E$ be a weight function, at every step of $\mathrm{PAMST}(\mathcal{G}, u_{\mathcal{G}}, w, \epsilon)$ Theorem 2 states that $\mathrm{Ex}\left(\mathcal{G}, w, u_{\mathcal{G}}, \mathcal{R}_{S_V}, \frac{\epsilon}{|V|-1}\right)$ is $\frac{\epsilon}{|V|-1}$ differentially private. Since Algorithm 1 relies on $|V| - 1$ adaptive calls of $\frac{\epsilon}{|V|-1}$ differentially private mechanisms, Theorem 4 ensures that Algorithm 1 is ϵ-differentially private on $\mathcal{G}$.

Algorithm 1 Private Approximated Minimum Spanning Tree PAMST$(\mathcal{G}, u_{\mathcal{G}}, w, \epsilon)$

Require: A graph topology $\mathcal{G} = (V, E)$, a weight function w, a privacy degree ϵ, utility function $u_{\mathcal{G}}$.

Ensure: The topology of an approximated minimum spanning tree represented by S_E.

Pick $v \in V$ at random
$S_V \leftarrow \{v\}$
$S_E \leftarrow \emptyset$
while $S_V \neq V$ **do**
  $r = \mathrm{Ex}(\mathcal{G}, w, u_{\mathcal{G}}, \mathcal{R}_{S_V}, \frac{\epsilon}{|V|-1})$
  $S_V \leftarrow S_V \cup \{r_{\rightarrow}\}$
  $S_E \leftarrow S_E \cup \{r\}$
end while
Return S_E

Notations:
Let S be a set of nodes from G, and $\mathcal{R}_S$ the set of edges between S and $V \backslash S$. For any edge r in such a set, the incident node to r that is not in S is denoted $r_{\rightarrow}$.

Let us now introduce the theoretical bound for Algorithm 1 that highlights the privacy/accuracy trade-off of PAMST. Lemma 1 presents a worst case bound on the difference between the weight of the output tree topology and the value w^* which is the sum of the optimal choices at every step of PAMST. In the sequel, we denote $\{\mathcal{R}_1, ..., \mathcal{R}_{|V|-1}\}$ the ranges used in the successive calls of the Exponential mechanism in PAMST.

Lemma 1. *Let us consider a weighted graph $G = (\mathcal{G}, w)$, $\gamma \in (0, 1)$, and $\epsilon > 0$. Let us denote $\mathcal{T}$ the spanning tree topology returned by* PAMST*, and $w(\mathcal{T})$ its associated total weight. Then, if one denotes $w^* = \sum_{i=1}^{|V|-1} \min_{r \in \mathcal{R}_i} w(r)$, and $\Delta = w(\mathcal{T}) - w^*$, one gets with probability at least $1 - \gamma$*

$$\Delta \leq \frac{4(|V|-1)}{\epsilon} \left((|V|-1) \ln \frac{|V|-1}{\gamma} + \sum_{i=1}^{|V|-1} \ln |\mathcal{R}_i| \right).$$

Since the steps are fixed, selecting the minimal weighted edge at every step does not necessarily produce a tree. Thus Lemma 2 is introduced to compare w^* and the weight of a minimum spanning tree in G.

Lemma 2. *Let $G = (\mathcal{G}, w)$ be a weighted graph, and $\mathcal{T}^*$ an MST of G. With the above notations for w^*, one has $w^* \leq w(\mathcal{T}^*)$.*

By putting together this two results, we obtain the following worst-case bound on the reconstruction error of PAMST.

Theorem 7. *Let $G = (\mathcal{G}, w)$ be a weighted graph, $\gamma \in (0,1)$, and $\epsilon > 0$. Denoting $\mathcal{T}^*$ an MST of G, $\mathcal{T} = \mathrm{PAMST}(\mathcal{G}, u_{\mathcal{G}}, w, \epsilon)$, and $\Delta = w(\mathcal{T}) - w^*$, one gets with probability at least $1-\gamma$*

$$\Delta \leq \frac{4(|V|-1)}{\epsilon}\left((|V|-1)\ln\frac{|V|-1}{\gamma} + 2\ln\left((|V|-1)!\right)\right).$$

Proof. Since for any $\mathcal{R}_i, i \in [|V|-1]$ one has $|\mathcal{R}_i| \leq i(|V|-i)$, one gets

$$\sum_{i=1}^{|V|-1} \ln|\mathcal{R}_i| \leq \ln\left(\prod_{i=1}^{|V|-1} i(|V|-i)\right) = 2\ln\left((|V|-1)!\right). \tag{1}$$

Introducing this inequality in the result from Lemma 1, one gets with probability at least $1-\gamma$

$$\Delta \leq \frac{4(|V|-1)}{\epsilon}\left((|V|-1)\ln\frac{|V|-1}{\gamma} + 2\ln\left((|V|-1)!\right)\right). \tag{2}$$

Using Lemma 2 at the left side of the inequality, one gets the expected results.

In addition to the worst-case result, exhibited in Theorem 7, one can easily get the following expectation bound.

Theorem 8. *Let $G = (\mathcal{G}, w)$ be a weighted graph, and $\epsilon > 0$. Denoting $\mathcal{T}^*$ a MST of G, and $\mathcal{T} = \mathrm{PAMST}(\mathcal{G}, u_{\mathcal{G}}, w, \epsilon)$, one has*

$$\mathbb{E}\left[w(\mathcal{T}) - w(\mathcal{T}^*)\right] \leq \frac{8(|V|-1)^2\ln(|V|-1)}{\epsilon}.$$

Proof. For any $i \in [|V|-1]$ we denote $w_{|\mathcal{R}_i}$ the weight function restricted on $\mathcal{R}_i$. We also denote $\epsilon' := \frac{\epsilon}{|V|-1}$, $OPT_i := OPT_{u_{\mathcal{G}}}\left(w_{|\mathcal{R}_i}\right)$, $u_i := u_{\mathcal{G}}(w, r_i)$, and finally $r_i \in \mathcal{R}_i$ the edge selected at the i^{th} step of the algorithm. First, for any $i \in [|V|-1]$, one has $|\mathcal{R}_i| \leq i(|V|-i)$. Therefore, according to Theorem 3 one has

$$\mathbb{E}[u_i] \geq OPT_i - \frac{2\mathcal{S}_{u_{\mathcal{G}}}}{\epsilon'}\ln\left(i(|V|-i)\right). \tag{3}$$

Since $OPT_i = 0$ for all i, one gets by summation,

$$\mathbb{E}\left[\sum_{i\in[|V|-1]} u_i\right] \geq -\frac{4\left(|V|-1\right)\mathcal{S}_{u_{\mathcal{G}}}}{\epsilon'}\ln\left(|V|-1\right). \tag{4}$$

Replacing $\mathcal{S}_{u_{\mathcal{G}}}$ by 2, ϵ' by $\frac{\epsilon}{|V|-1}$, and using the definition of the utility function in the expectation, one gets

$$\mathbb{E}\left[w^* - w(\mathcal{T})\right] \geq -\frac{8|V-1|^2}{\epsilon}\ln\left(|V|-1\right), \tag{5}$$

where $w^* = \sum_{i=1}^{|V|-1} \min_{r\in\mathcal{R}_i} w(r)$. Finally, by inverting the inequality, and using Lemma 2, one gets the expected result.

4.2 Comparison with the State of the Art

We compare our method to the state of the art approaches, respectively Laplace [21] and Private Kruskal [15] methods. The main difference between the Laplace method and both others is the construction of the algorithm. Sealfon's method relies on a non iterative use of the Laplace mechanism while PAMST and Private Kruskal are based on an iterative use of the Exponential mechanism. The non iterative Laplace method ensures privacy by Post-processing, which leads to good results in term of time complexity, but it does not provide the best accuracy guarantees. Conversely, PAMST and Private Kruskal present better accuracy guarantees at the expense of a moderately increased algorithmic complexity. In the sequel we analyze the errors and time complexity achieved by the approaches in [21,15] and PAMST. One should note that contrarily to these works, we provide both worst-case and expectation bounds. Therefore, we compare to existing bounds *i.e.* worst-case bound for the Laplace method (Theorem 10 and 11), and expectation bound for the Private Kruskal (Remark 1). First, let us recall some results, useful for the understanding of Theorem 10.

Theorem 9 (adapted from[21]). *For any $\epsilon > 0, \gamma \in (0,1)$, $\mathcal{G} = (V, E)$, and $w : E \to \mathbb{R}$, the construction of a private minimum spanning tree based on the post-processing of the Laplace mechanism releases with probability at least $1-\gamma$ a spanning tree of weight at most $\frac{2(|V|-1)|E|}{\epsilon} \ln\left(\frac{|E|}{\gamma}\right)$ larger than optimal.*

Definition 7 ([17]). *The Lambert W-function, denoted W, also called the omega function, is the inverse function of f where for all $z \in \mathbb{C}, f(z) = ze^z$.*

Lemma 3 ([19]). *For all $k \in \mathbb{N}^*, k! \leq \sqrt{2\pi} k^{k+\frac{1}{2}} e^{-k+\frac{1}{12k}}$.*

Denoting $\mathcal{B}_{Exp,\epsilon,\gamma}$ the bound from Theorem 7, and $\mathcal{B}_{Lap,\epsilon,\gamma}$ the one from Theorem 9, the first theorem is as follows.

Theorem 10. *Let $G = (V, E, w)$ be a weighted graph. $|E|$ being the number of edges, and $|V|$ its number of nodes, one can always write $|E| = \tau(|V|-1)$ with $\tau \geq 1$. For readability, we denote $(|V|-1) = n$. Given that we evaluate the bounds of the two mechanisms according to the same parameters γ and ϵ,*

$$\text{if } \tau \geq \frac{\gamma}{n} e^{W\left(\frac{\ln(h_{n,\gamma})}{2}\right)} \text{ then } \mathcal{B}_{Exp,\epsilon,\gamma} \leq \mathcal{B}_{Lap,\epsilon,\gamma}$$

with $h_{n,\gamma} := \gamma^{-\frac{n}{\gamma}} (2\pi)^{\frac{1}{\gamma}} n^{\frac{3n+1}{\gamma}} e^{-\frac{2n}{\gamma}+\frac{1}{6n\gamma}}$.

Proof. One has that for all $\gamma \in (0,1)$ and $\epsilon > 0$

$$\mathcal{B}_{Exp,\epsilon,\gamma} \leq \mathcal{B}_{Lap,\epsilon,\gamma} \tag{6}$$

$$\Leftrightarrow \left(n \ln\left(\frac{n}{\gamma}\right) + 2\ln(n!)\right) \leq 2\tau n \ln\left(\frac{\tau n}{\gamma}\right). \tag{7}$$

Using Lemma 3, it is sufficient to show

$$\frac{1}{2}\left(n\ln\left(\frac{n}{\gamma}\right)+\ln\left(2\pi n^{2n+1}e^{-2n+\frac{1}{6n}}\right)\right)\leq\tau n\ln\left(\frac{\tau n}{\gamma}\right) \tag{8}$$

$$\Leftrightarrow\frac{1}{2}\ln\left((h_{n,\gamma})^{\gamma}\right)\leq\tau n\ln\left(\frac{\tau n}{\gamma}\right) \tag{9}$$

$$\Leftrightarrow\frac{\ln\left(h_{n,\gamma}\right)}{2}\leq\frac{\tau n}{\gamma}\ln\left(\frac{\tau n}{\gamma}\right)=g\left(\frac{\tau n}{\gamma}\right). \tag{10}$$

Since $g:=x\to x\ln(x)$ is a non-decreasing function on $\mathbb{R}^+$, solving $\frac{\ln(h_{n,\gamma})}{2}=g\left(\frac{\tau n}{\gamma}\right)$ is sufficient to find the value τ^* that for all $\tau\geq\tau^*$, satisfies the inequality. Using the Lambert W-function one gets $\tau^*=\frac{\gamma}{n}\exp\left(W\left(\frac{\ln(h_{n,\gamma})}{2}\right)\right)$. Thus one has that $\ln\left(\frac{\tau^* n}{\gamma}\right)=W\left(\ln\left(h_{n,\gamma}\right)\right)$. Finally one gets $\tau^*=\frac{\gamma}{n}\exp\left(W\left(\frac{\ln(h_{n,\gamma})}{2}\right)\right)$.

If the second result is not as tight as the first one, it has the interest of neither depending on the order of the graph nor on any privacy/certainty degree.

Theorem 11. *Let $G=(V,E,w)$ be a weighted graph. If one has $\tau\geq 6$, then $\mathcal{B}_{Exp,\epsilon,\gamma}\leq\mathcal{B}_{Lap,\epsilon,\gamma}$, regardless of the degrees of certainty and privacy γ and ϵ.*

Proof. Let us consider fixed values for $\gamma\in(0,1)$, and $\epsilon>0$. One gets

$$\mathcal{B}_{Exp,\gamma}\leq\mathcal{B}_{Lap,\gamma} \tag{11}$$

$$\Leftrightarrow\frac{2n}{\epsilon}\left(n\ln\left(\frac{n}{\gamma}\right)+2\ln\left(n!\right)\right)\leq\frac{4n^2\tau}{\epsilon}\ln\left(\frac{\tau n}{\gamma}\right). \tag{12}$$

Since $n!\leq n^n$ it suffices to find τ such that

$$\ln\left(\frac{n}{\gamma}\right)+2\ln(n)\leq 2\tau\ln\left(\frac{\tau n}{\gamma}\right) \tag{13}$$

Finally, since $\gamma\in(0,1)$, one thus easily gets $\ln\left(\frac{n}{\gamma}\right)\geq\ln(n)$. Therefore one gets the expected result.

The parameter τ in Theorems 10 and 11 should be understood as a non-normalized density, or a sparsity degree of the graph. This factor represents how far is the size of the graph from the minimal size of a connected one. One could refer to [12] for a discussion on the semantics of this parameter. This paper also studies many real-world datasets and shows that real-world networks exceed in most cases the sparsity degree exhibited in our both theorems. PAMST provably outperforms the Laplace method under weak assumptions on the graph sparseness. Private Kruskal, as well as PAMST, presents good accuracy results, as demonstrated in Theorem 12. Remark 1 discusses the linear link between the expected accuracy results of both methods.

Theorem 12 ([15]). *Let $G = (\mathcal{G}, w)$ be a weighted graph, and $\epsilon > 0$. Denoting $\mathcal{T}^*$ a MST of G, $\mathcal{T}$ the tree obtained by differentially private submodular minimization (Kruskal algorithm), one has*

$$\mathbb{E}\left(w\left(\mathcal{T}\right) - w\left(\mathcal{T}^*\right)\right) \leq \frac{2(|V|-1)^2 \ln\left(|V|-1\right)}{\epsilon} - \frac{w\left(\mathcal{T}^*\right)}{\exp(1)}$$

If we denote $\mathcal{B}_{Kruskal,\epsilon}$ the bound from Theorem 12, and $\mathcal{B}_{PAMST,\epsilon}$ the one from Theorem 8, one straightforwardly gets the following results.

Remark 1. Let $\epsilon > 0$, one has $\mathcal{B}_{PAMST,\epsilon}/4 - w(\mathcal{T}*)/\exp(1) = \mathcal{B}_{Kruskal,\epsilon}$.

Table 1: Time complexity and accuracy guarantees for = PAMST, iterative Private Kruskal mechanism, and non iterative Laplace mechanism. Two implementations of the Laplace method are investigated, respectively with Binary heap and Fibonacci heap.

Algorithm	Complexity	Worst Case	Expectation
PAMST	$O\left(\lvert V\rvert^2\right)$	$O\left(\lvert V\rvert^2 \ln \lvert V\rvert\right)$	$O\left(\lvert V\rvert^2 \ln \lvert V\rvert\right)$
Private Kruskal [15]	$O\left(\lvert V\rvert \times \lvert E\rvert\right)$	—	$O\left(\lvert V\rvert^2 \ln \lvert V\rvert\right)$
Laplace (Binary heap) [21]	$O\left(\lvert E\rvert \log \lvert V\rvert\right)$	$O\left(\lvert V\rvert \times \lvert E\rvert \ln \lvert E\rvert\right)$	—
Laplace (Fibonacci heap) [21]	$O\left(\lvert E\rvert + \lvert V\rvert \log \lvert V\rvert\right)$	$O\left(\lvert V\rvert \ln \lvert E\rvert\right)$	—

Iterative methods (PAMST and Private Kruskal) ensure privacy and accuracy, they require however more computation time. The Laplace mechanism is based on adding random noise to the edge weights before applying a deterministic MST algorithm on the perturbed graph. Since the noise simulation phase is negligible when studying the time complexity, to compare the non iterative mechanism with others methods, it suffices to compare it with the associated deterministic MST algorithm. Table 1 compares the three methods w.r.t their time complexity and accuracy guarantees. We consider two implementations of the non iterative Laplace method, namely Prim algorithm with respectively Binary heap and Fibonacci heap. As expected, the non iterative methods show better complexity in comparison to the iterative ones. This difference can be explained by the use of the priority queue. Indeed, since the Laplace method uses a deterministic algorithm, it is able to exploit priority queues-based implementation

with either Fibonnacci or Binary heaps. Such an implementation is not possible for neither PAMST nor Private Kruskal since the Exponential mechanism has to be independently computed at each step of the algorithm, and since there is no trivial way of computing a private and accurate priority queue for the whole graph. It is worth noting that PAMST presents a better time complexity compared to Private Kruskal, especially when one deals with dense graphs. In fact, in this particular case, Private Kruskal time complexity is $O(|V|^3)$ which is prohibitive when the graph is very large. PAMST represents then the state of the art for releasing a minimum spanning tree under differential privacy constraints in terms of time-complexity/accuracy trade-off.

5 Application to Clustering

Section 4 demonstrated that PAMST is a suitable mechanism to find an almost minimum spanning tree under differential privacy with reduced time complexity. This structure can be used in several unsupervised learning tasks *e.g* clustering in a graph. In the following we present a real world scenario experiment to show that PAMST-based and Private Kruskal-based clustering methods obtain similar results. Regarding the computational complexity of both algorithms, this experiment advocates for the use of PAMST in clustering applications. To confirm our theoretical findings (iterative methods are better suited to our framework than non-iterative ones) we also present experiments using the Laplace method. Finally, to be as fair as possible in the comparison protocol, we present an experiment on a small-scale real world data set.

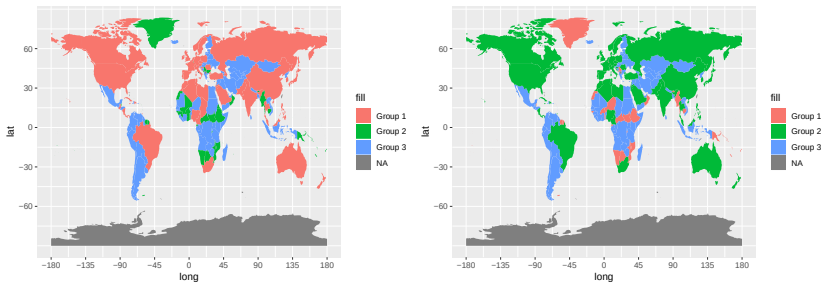
5.1 Experimental setting

United Nations Global Migration graph: To illustrate the performance of PAMST, we present in the sequel, an experiment on the "United Nations Global Migration Database"[3]. This dataset is composed of counts of migrants from 232 countries and territories worldwide in 2017. For the purpose of our experiment we sketch it into a complete graph, where nodes represent countries, and edges represent migration roads between them. Every edge is valued with the difference between migration rates (immigrants - immigrants in the year). We chose to treat this dataset as it deals with an important society concern and raises technical issues. Moreover, the graph is composed of 26792 edges and is thus tractable by all 3 methods.

Group privacy: Differential privacy is also adapted to the analysis and privacy control of a group of individuals(see [4] for more details on group differential privacy). This notion is called group privacy, and is inherited from the composition property. In practice, for any integer $k \geq 1$, an algorithm is said to be k-group ϵ-differential private if it simultaneously preserves the ϵ-differential

[3] This dataset is freely available on the following website: un.org/en/development/desa/population/migration/data/estimates2/estimates17.asp

privacy for any entity in any group of k individuals. This notion is particularly interesting when considering members from the same family. Since migration is usually a family matter, during our experiments, we chose to preserve 10-group privacy *i.e.* privacy for a group of 10 individuals.



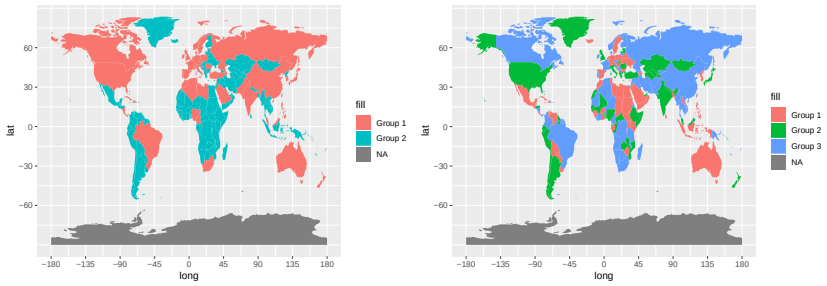
(a) PAMST, $\epsilon = 1.0$ (b) Private Kruskal, $\epsilon = 1.0$

Fig. 2: Private clustering obtained by PAMST *vs* Private Kruskal on the UN Global Migration graph. Both methods ensure 10-group privacy of parameter $\epsilon = 1$. Fig. (a) and (b) obtain similar result, and both are close to the non-private case. Group numbers, and colors are chosen arbitrarily and have no semantics.

Graph Clustering: Graph clustering [20] is a key tool for understanding the underlying structure of many data sets by locating nodes groups ruled by a specific similarity. The minimum spanning tree is known to help recognizing clusters with arbitrary shapes in tree-based clustering algorithms. It thus can be used for wider applications than community detection. In the sequel, we present a procedure for node clustering in a graph, under differential privacy conditions. It consists of two steps: 1) compute a differentially private minimum spanning tree, 2) use this tree as the input of a non-private tree-based clustering algorithm (we use the classical MSDR algorithm from [22]). This procedure results in a differentially private node clustering in a graph, thanks both to the privacy guarantees from the method used in 1) and Theorem 5 (post-processing). Such a private clustering algorithm could pave the way to mutual interaction analysis in genomics, web traffic, and migration flows analysis where privacy-preserving is not an option but a strong requirement.

5.2 Results and analysis

We apply our private clustering method to the United Nations Global Migration graph, and compare clustering results of PAMST, Private Kruskal, and Laplace method with the non-private version of MSDR algorithm (the first step is a classical minimum spanning tree algorithm). Figure 3a represents the outcome of



(a) MSDR non Private (b) Laplace, $\epsilon = 1.0$

Fig. 3: Non-private clustering Vs Laplace-based Private clustering of the UN Global Migration graph. In (a) Group 1 represents countries with a high migration rate, and Group 2 those with lower migration rate. In (b) 10-group privacy of parameter $\epsilon = 1$ is ensured, and no meaningful group semantic is recovered. Group numbers, and colors are chosen arbitrarily and should not be interpreted.

MSDR in the non-Private case. We obtain a country partition in two groups (1 and 2). According to the underlying semantic on the graph construction, group 1 can be interpreted as the set of countries with a high migration rate, and conversely, group 2 represents countries with a low migration rate. Figure 2a and 2b show that both clustering with PAMST and Private Kruskal achieve close results, while respecting the global separation, even if some "noise clusters" appear. Conversely Laplace (Figure 3b) provides a partition that does not respect the initial semantic. The results presented in Figures 3 and 2 once again confirm both that Iterative methods is more relevant than the Laplace method, and that PAMST and Private Kruskal obtain similar results. This argue in the favor of PAMST, since, as shown in the clustering application, PAMST shows results similar to Private Kruskal with a lower computational complexity.

6 Conclusion

We presented a provably and efficient differentially private algorithm for computing a minimum spanning tree of a dense graph, and demonstrated its use on a clustering task in a real world scenario. Our approach presents a significant improvement over the state of art approaches both in terms of approximation error and time complexity. It is the first approach allowing to deal efficiently with relatively large and dense graphs.

This work paves the way for trustworthy machine learning in critical application domains where datasets are massive and sensitive. Two application domains are currently under study: traffic data and population genomics. Future work will also consider optimized PAMST implementation inspired by distributed

MST computations [8]. This type of method could lead to substantial decrease in computation time when dealing with large graphs.

References

1. Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: TCC (2016)
2. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: Eurocrypt. vol. 4004, pp. 486–503. Springer (2006)
3. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography, pp. 265–284. Springer Berlin Heidelberg (2006)
4. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3-4), 211–407 (2013)
5. Fung, B., Wang, K., Cheng, R., Yu, P.: Privacy-preserving data publishing: A survey of recent developments. ACM Comput. Surv. **42**(4), 14:1–14:53 (Jun 2010)
6. Gupta, A., Ligett, K., McSherry, F., Roth, A., Talwar, K.: Differentially private combinatorial optimization. In: Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1106–1125. SODA '10, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2010)
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 (Dec 2009)
8. Karloff, H., Suri, S., Vassilvitskii, S.: A model of computation for mapreduce. In: Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 938–948. SODA '10, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2010)
9. Karwa, V., Raskhodnikova, S., Smith, A., Yaroslavtsev, G.: Private analysis of graph structure. Proceedings of the VLDB Endowment **4**(11), 1146–1157 (2011)
10. Kasiviswanathan, S.P., Nissim, K., Raskhodnikova, S., Smith, A.: Analyzing graphs with node differential privacy. In: Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography. pp. 457–476. TCC'13, Springer-Verlag, Berlin, Heidelberg (2013)
11. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: Annual IEEE Symposium on Foundations of Computer Science (FOCS). IEEE, Providence, RI (October 2007)
12. Melançon, G.: Just how dense are dense graphs in the real world? A methodological note. In: Enrico Bertini, Catherine Plaisant, G.S. (ed.) BELIV 2006: Beyond time and errors: novel evaLuation methods for Information Visualization (AVI Workshop). pp. 75–81. ACM, Venice, Italy (May 2006)
13. Mir, D., Wright, R.: A differentially private graph estimator. In: 2009 IEEE International Conference on Data Mining Workshops. pp. 122–129 (Dec 2009)
14. Mironov, I.: Renyi differential privacy. In: 30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017. pp. 263–275 (2017). https://doi.org/10.1109/CSF.2017.11
15. Mitrovic, M., Bun, M., Krause, A., Karbasi, A.: Differentially private submodular maximization: Data summarization in disguise. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 2478–2487. PMLR, International Convention Centre, Sydney, Australia (06–11 Aug 2017)
16. Nissim, K., Raskhodnikova, S., Smith, A.: Smooth sensitivity and sampling in private data analysis. In: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC. ACM Press (2007)

17. Pólya, G., Szegö, G.: Problems and Theorems in Analysis. Springer New York, New York, NY (1972)
18. Prim, R.: Shortest connection networks and some generalizations. The Bell System Technical Journal **36**(6), 1389–1401 (Nov 1957)
19. Robbins, H.: A Remark on Stirling's Formula, pp. 402–405. Springer New York, New York, NY (1985)
20. Schaeffer, S.E.: Graph clustering. Computer Science Review **1**(1), 27–64 (aug 2007)
21. Sealfon, A.: Shortest paths and distances with differential privacy. In: Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems - PODS. ACM Press (2016)
22. Zhou, Y., Grygorash, O., Hain, T.F.: Clustering With Minimum Spanning Trees. International Journal on Artificial Intelligence Tools **20**(01), 139–177 (feb 2011)