Differentially Private Vertical Federated Learning Primitives

Vincent Cohen-Addad
Google Research

Praneeth Kacham
CMU

Vahab Mirrokni
Google Research

Peilin Zhong
Google Research

Abstract

In vertical federated learning, we are given a data-set about users that is distributed across several servers. Each server owns a specific disjoint subset of attributes of all users. The goal is to perform data analysis tasks over the entire distributed data-set containing all attributes of every user. We study private algorithms for basic learning problems including pattern counting, moment estimation, k-clustering, low-rank matrix approximation and linear regression in vertical federated learning where the messages communicated between servers are differentially private. Our moment estimation and k-clustering algorithms are built on a novel pattern counting algorithm, where the goal of pattern counting is to count the number of users with a certain pattern of attributes. Our pattern counting algorithm is a generalization of the randomized response technique and we show that its approximation is near optimal. In addition, our algorithms for pattern counting and its applications also satisfy local differential privacy, which may be of independent interest.

For low-rank matrix approximation and linear regression, we adopt sketching-and-perturbation technique and show differentially private approximate algorithms for both problems in the vertical federated learning setting.

1 Introduction

Federated Learning (FL) consists of building machine learning models from datasets that are distributed across different entities while limiting the total amount of data samples exchanged amongst the entities. This allows us to build models using large sources of possibly sensitive data owned by different actors while limiting the risks regarding data privacy, data security, or data access rights [40]. Two major paradigms of Federated Learning have emerged: Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL) [66]. In HFL, different servers contain information about different set of users and the objective is to perform basic data mining or machine learning tasks over the set of all users across the servers. In VFL, all servers contain information about the same set of users but each server has a different set of attributes for each individual user. The horizontal/vertical terminology arose by treating the joint dataset as an $m \times d$ matrix where m is the number of users and d is the number of attributes.

VFL has become increasingly important as entities in different domains, that all serve the same set of users, have come to collaborate so they can train better models and understand their users better owing to the availability of increased number of attributes corresponding to each user. In this context, preserving privacy is of essential importance to assure the users that their sensitive data is not shared with other entities that has some other attributes of the same user. In some cases, sharing private data of users with other entities is regulated by privacy laws such as GDPR [22] and HIPAA [57]. More recently, the Digital Markets Act (DMA) [21] will put restrictions on data-sharing even between different entities within the same company. Hence, studying privacy-preserving algorithms for VFL is of essential industrial importance. Over the last few years, differential privacy has become the de facto standard for privacy requirements in machine learning and data mining applications. In this context, we aim to design algorithms in the VFL model that are *differentially private* (DP) [16, 18]. At a high level, DP guarantees that any single user's data cannot significantly affect the output of the algorithm. Roughly, we say that an algorithm in the VFL setting is DP if the overall historical messages (transcript) sent by each server over the entire course of the algorithm is DP with respect to the data it has. We formalize this definition of differential privacy in the VFL setting in §1.1. A similar definition has been used by [39] in the context of two servers. Recently, DP VFL algorithms were also studied by [64, 48].

Concretely, the VFL setting is modeled as follows: There are v data silos (servers) and the j-th server has an $m \times d^{(j)}$ matrix $A^{(j)}$ (or an m dimensional vector $x^{(j)}$ if $d^{(j)} = 1$). Here m denotes the number of users and $d^{(j)}$ denotes the number of attributes that the j-th server has about the users. We define $d \coloneqq \sum_j d^{(j)}$ to be the total number of attributes. We also assume that there is a central coordinator (CC) that aggregates the messages sent by different servers and computes the final solution with respect to the overall dataset $[A^{(1)}\ A^{(2)}\ \cdots\ A^{(v)}] \in \mathbb{R}^{m \times d}$. In this work, we consider the scenario where the number of servers, v, is small. This is a reasonable setting in practice when the attributes of interest are distributed over a small number of data silos/servers.

In this work, we show two general schemes for DP VFL algorithms. We call the first scheme *pattern-counting* scheme. At a high level, we propose a generalization of the randomized response [61] technique to estimate the number of users with a certain pattern of attributes across servers. To the best of our knowledge, this is the first DP VFL algorithm for the pattern counting problem. In addition, we prove that its approximation is near optimal (see §F). Our another technical contribution is to show a series of extensions including norm estimation and clustering problems under DP VFL setting by reducing these problems to our pattern counting primitive. Interestingly, our pattern counting primitive and its applications also satisfy local differential privacy, which may be of independent interest. The second scheme for DP VFL algorithms is a simple *sketching-and-perturbation* technique which was originally developed for DP numerical linear algebraic algorithms [27] in the non-FL setting. We show how to apply these techniques to obtain DP algorithms in the VFL setting for low-rank matrix approximation (a.k.a. Principal Component Analysis) and linear regression problems.

We include informal versions of our results and a high level technical overview in this extended abstract. We refer readers to the appendix for more details.

1.1 Definitions

Let $\mathcal{D}$ be the family of datasets. We define a symmetric relation $\sim$ on $\mathcal{D}$ and say that for $X, X' \in \mathcal{D}$, if $X \sim X'$, then X and X' are *neighbors*. In this paper, we consider the following definition of neighboring datasets: if datasets X and X' differ on at most one data entry, and the difference between the two data entries is at most some given bound, then we say X and X' are neighbors.

Definition 1.1 (ε-DP and (ε, δ)-DP)**.** A *randomized* mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{O}$ is (ε, δ)-differentially private ((ε, δ)-DP) if $\forall S \subseteq \mathcal{O}$ and $\forall X, X' \in \mathcal{D}$ with $X \sim X'$, $\mathbf{Pr}[\mathcal{M}(X) \in S] \leq e^{\varepsilon} \mathbf{Pr}[\mathcal{M}(X') \in S] + \delta$. If $\delta = 0$, we say the mechanism $\mathcal{M}$ is ε-DP.

Definition 1.2 (DP in VFL)**.** Consider a distributed randomized algorithm $\mathcal{A}$ over v servers and let $\mathcal{M}_{\mathcal{A}}(X^{(1)}, X^{(2)}, \cdots, X^{(v)})$ denote the transcript, i.e., the historical messages sent by all servers, when the j-th server holds the dataset $X^{(j)} \in \mathcal{D}^{(j)}$ for each $j \in [v]$. Let $\mathcal{T}$ be the set of all possible transcripts. If $\forall S \subseteq \mathcal{T}$ and $\forall j \in [v], \forall X^{(j)}, X'^{(j)} \in \mathcal{D}^{(j)}$ with $X^{(j)} \sim X'^{(j)}$, $\mathbf{Pr}[\mathcal{M}_{\mathcal{A}}(X^{(1)}, X^{(2)}, \cdots, X^{(v)}) \in S] \leq e^{\varepsilon} \mathbf{Pr}[\mathcal{M}_{\mathcal{A}}(X^{(1)}, X^{(2)}, \cdots X'^{(j)} \cdots, X^{(v)}) \in S] + \delta$, then we say $\mathcal{A}$ is (ε, δ)-DP in VFL. If $\delta = 0$, we say $\mathcal{A}$ is ε-DP in VFL.

The above definition extends the two-party differential privacy studied by [39]. Suppose a distributed algorithm is non-interactive, i.e., the messages sent by a server does not depend on the messages sent by other servers. Using the compositionality of differential privacy, if the messages sent by the j-th server is DP with respect to the dataset held by the j-th server, then the algorithm is DP in VFL. Notably, all our algorithms are non-interactive in this work.

1.2 Our Results

We give a brief overview of our DP VFL algorithms including pattern counting, moment estimation, clustering, low rank approximation and linear regression. In all results below, we present v as a constant for simplicity.

Pattern counting. For pattern counting, each server $j \in [v]$ holds a vector $x^{(j)} \in \{0, \cdots, M\}^m$. We say $x'^{(j)} \in \{0, \cdots, M\}^m$ is a neighboring dataset of $x^{(j)}$ if there is at most one user $i \in [m]$ such that $x'^{(j)}[i] \neq x^{(j)}[i]$. The goal is to develop a DP algorithm to count the number of occurrences of each possible row pattern $\rho = (\rho^{(1)}, \cdots, \rho^{(v)}) \in \{0, 1, \cdots, M\}^v$. More precisely, the number of occurrences of the pattern ρ is defined as $\mathrm{count}_\rho \coloneqq |\{\text{user } i \in [m] \mid \bigwedge_{j \in [v]} x^{(j)}[i] = \rho^{(j)}\}|$.

Theorem 1.3 (Simplified version of Theorem B.1 and Theorem B.2)**.** *There is an ε-DP VFL pattern counting algorithm which outputs $\overline{\mathrm{count}}_\rho$ for every possible pattern ρ such that with probability at least 0.99, $\forall \rho \in \{0, 1, \cdots, M\}^v, \overline{\mathrm{count}}_\rho \in \mathrm{count}_\rho \pm \sqrt{m} \cdot (1/\varepsilon)^v \cdot O_v(\log M)$.*

Using the lower bounds of [39], we can show that the $\sqrt{m}$ dependence in additive error is necessary even for $v = 2$. Obtaining lower bounds on the additive error in terms of the number of servers v is an important open question.

ℓ_p Moment estimation. In the ℓ_p moment estimation problem in VFL, each server j holds a non-negative integer vector $x^{(j)}$ and the goal is to estimate the ℓ_p moment $\|x^{(1)} + \cdots + x^{(v)}\|_p^p$ where $p \geq 1$ is some constant. We consider *event-level* DP for this problem, i.e., a neighboring dataset $x'^{(j)}$ of $x^{(j)}$ can differ from $x^{(j)}$ by at most one entry and in addition $\|x'^{(j)} - x^{(j)}\|_\infty \leq 1$. Event-level DP non-federated learning ℓ_p moment estimation algorithms have been studied by e.g., [59, 20]. The following theorem states our result for DP VFL ℓ_p moment estimation.

Theorem 1.4 (Simplified version of Theorem C.2)**.** *For any $\alpha \in (0, 0.5)$, there is an ε-DP ℓ_p moment estimation algorithm in the VFL model which achieves $(1 \pm \alpha)^p$-multiplicative error and $\sqrt{m} \cdot ((\log m)/\varepsilon\alpha)^{O_{v,p}(1)}$-additive error with probability at least 0.99.*

Similar to our pattern counting algorithm, our algorithm for ℓ_p moment estimation also satisfies both DP in vertical federated learning and local DP simultaneously. For the special case when $p = 2$, we present a simpler sketching-and-perturbation based algorithm with better approximation guarantee.

Theorem 1.5 (Simplified version of Theorem E.5)**.** *For any $\alpha \in (0, 0.5)$, there is an (ε, δ)-DP ℓ_2 moment estimation algorithm in the VFL model which achieves $(1 \pm \alpha)$-multiplicative error and $O_v\left(\frac{\log(m)\log(1/\delta)}{\alpha\varepsilon^2}\right)$-additive error with probability at least 0.99.*

Euclidean k-means clustering. Given m points $A \in \mathbb{R}^{m\times d}$ (each point is indicated by a row $A[i]$) in d-dimensional Euclidean space, k-means clustering aims to find a set of k centers $C \in \mathbb{R}^{k\times d}$ such that the clustering cost $\sum_{i\in[m]} \min_{l\in[k]} \|A[i] - C[l]\|_2^2$ is minimized. In the literature of differentially private k-clustering, it is a common assumption that an upper bound Δ is known such that every data point $A[i]$ is in a ball with radius Δ [13, 25, 52]. In the vertical federated learning model, input A is vertically partitioned into $A = [A^{(1)}\ A^{(2)}\ \cdots\ A^{(v)}]$. Any neighboring dataset $A'^{(j)}$ of $A^{(j)}$ satisfies that there is at most one row $i \in [m]$ such that $A'^{(j)}[i] \neq A^{(j)}[i]$, and $A'^{(j)}[i]$ is still in the ball with radius Δ. The goal is to develop a differentially private algorithm in the vertical federated learning model to compute a set of k centers to minimize the k-means clustering cost with respect to A.

Theorem 1.6 (Simplified version of Corollary D.4)**.** *There is an ε-DP k-means clustering algorithm in the VFL model which achieves multiplicative error $O(1)$ and additive error $O\left(\left((k/\varepsilon)d\log^{O(1)}(m) + \sqrt{m}\cdot(k/\varepsilon)^{O_v(1)}\right)\Delta^2\right)$ with probability at least 0.99.*

Our algorithm uses DP k-means algorithm in the non-federated learning as a subroutine. If we instead use local DP k-means algorithm, our algorithm can easily satisfy both DP in vertical federated learning and local DP at the same time. Note that $(k/\varepsilon)^{\Omega(1)}\Delta^2$ additive error is necessary for a DP algorithm even in the non-federated learning setting [26]. In addition $\Omega(\sqrt{m}\Delta^2)$ additive error is necessary in the local DP setting [51].

Low rank approximation and linear regression. By applying the sketching-and-perturbation technique similar to the ℓ_2 moment estimation algorithm mentioned previously, we also present DP algorithms for low rank matrix approximation and least squares linear regression in the vertical federated learning model. These algorithms are obtained by simulating non-federated learning DP algorithms of [27]. For these two problems in the vertical federated learning model, the input matrix $A \in \mathbb{R}^{m\times d}$ is partitioned into $A = [A^{(1)}\ A^{(2)}\ \cdots\ A^{(v)}]$ where $A^{(j)}$ is held by the j-th server. For the linear regression problem, there is an additional server which holds the label vector $b \in \mathbb{R}^m$. We consider the same definition of neighboring dataset as [27]. The neighboring dataset $A'^{(j)}$ of $A^{(j)}$ satisfies that there is at most one row $i \in [m]$ such that $A'^{(j)}[i] \neq A^{(j)}[i]$ and in addition $\|A'^{(j)}[i] - A^{(j)}[i]\|_2 \leq 1$. Similarly, the neighboring dataset b' of b satisfies that there is at most one entry $i \in [m]$ such that $b'[i] \neq b[i]$ and in addition $\|b' - b\|_\infty \leq 1$. The goal of rank-k matrix approximation is to find a matrix $V \in \mathbb{R}^{d\times k}$ such that $\|A - AVV^\top\|_\mathsf{F}$ is minimized. The goal of least squares linear regression is to find a coefficient vector x such that $\|Ax - b\|_2$ is minimized. Due to space constraints, we move the discussion about these results to Appendix E.

Theorem 1.7 (Simplified version of Theorem E.1)**.** *For any $\alpha \in (0, 0.5)$, there is an (ε, δ)-DP rank-k approximation algorithm in the vertical federated learning model which achieves $(1 + \alpha)$-multiplicative error and $O(\varepsilon^{-1}\sqrt{(k/\alpha^2 + \log m)\log(1/\delta)d})$-additive error.*

Theorem 1.8 (Simplified version of Theorem E.4)**.** *For any $\alpha \in (0, 0.5)$, there is an (ε, δ)-DP least squares linear regression algorithm in the vertical federated learning model which achieves $(1 + \alpha)$-multiplicative error and $O((d/\alpha^2 + \log m)\log(1/\delta)\varepsilon^{-2}(\|x^*\|_2^2 + 1))$-additive error, where x^* is the optimal solution.*

Though linear regression is also studied by [64] with a similar algorithm, they make assumptions on data distribution and analyze the distance between the output coefficient vector and the ground truth solution, while we consider the worst case approximation of the objective function.

1.3 Our Techniques

In this section, we briefly discuss the high level ideas of our algorithms. We first present our core idea for the pattern counting algorithm, and then discuss how to reduce moment estimation and clustering to the pattern counting problem.

Pattern counting. To illustrate the intuition, we consider a simple case when there are only two servers and each server holds a binary vector $x^{(1)}$ and $x^{(2)}$ respectively. The goal is to count the number of users $i \in [m]$ such that $x^{(1)}[i] = x^{(2)}[i] = 1$. An intuitive way is to send private versions of $x^{(1)}$ and $x^{(2)}$ to the central coordinator and the central coordinator estimates the number of appearances of the pattern based on the private vectors. A standard way to privately encode a binary vector is randomized response [61]: Given a binary vector $u \in \mathbb{R}^m$, we flip each entry of u with probability $1/2 - \varepsilon$, and the obtained vector q is $\Theta(\varepsilon)$-DP [18].

Given randomized responses $y^{(1)}$ and $y^{(2)}$ of $x^{(1)}$ and $x^{(2)}$ respectively, we propose the following decoding method: For $j \in \{1, 2\}$, and $i \in [m]$, let $z^{(j)}[i] = (y^{(j)}[i] - (1/2 - \varepsilon))/(2\varepsilon)$. It is easy to see that $\mathbf{E}[z^{(j)}[i]] = x^{(j)}[i]$ and thus $\mathbf{E}[z^{(1)}[i]z^{(2)}[i]] = \mathbf{E}[z^{(1)}[i]]\,\mathbf{E}[z^{(2)}[i]] = x^{(1)}[i]x^{(2)}[i]$ by the independence. Therefore $\sum_{i\in[m]} z^{(1)}[i]z^{(2)}[i]$ is an unbiased estimator of the number of users whose entries in $x^{(1)}$ and $x^{(2)}$ are both 1. In addition, we have $|z^{(j)}[i]| \le O(1/\varepsilon)$ with probability 1 and therefore using a Hoeffding bound, we obtain that $\sum_{i\in[m]} z^{(1)}[i]z^{(2)}[i]$ is a good estimate with high probability.

The above procedure can be easily extended to the case where there are $v > 2$ servers. Each server j holds a binary vector $x^{(j)}$, and the goal is to estimate the number of users whose entries in $x^{(1)}, x^{(2)}, \cdots, x^{(v)}$ are all 1. Similar to the approach described above, we can firstly compute randomized responses $y^{(1)}, y^{(2)}, \cdots, y^{(v)}$ for $x^{(1)}, x^{(2)}, \cdots, x^{(v)}$ respectively, and then derive $z^{(1)}, z^{(2)}, \cdots, z^{(v)}$ from $y^{(1)}, y^{(2)}, \cdots, y^{(v)}$ respectively. We can obtain a good estimation using $\sum_{i\in[m]} \prod_{j\in[v]} z^{(j)}[i]$. Finally, we show how to reduce the non-binary pattern counting problem to the binary pattern counting problem: Consider a pattern $\rho \in \{0, 1, \cdots, M\}^v$. For each server j, we create a binary vector $\hat{x}^{(j)}$ where $\hat{x}^{(j)}[i] = 1$ if and only if $x^{(j)}[i] = \rho^{(j)}$. Then the number of occurrences of ρ in $x^{(1)}, \cdots, x^{(v)}$ is the same as the number of users whose entries in $\hat{x}^{(1)}, \cdots, \hat{x}^{(v)}$ are all 1. Since randomized responses are local DP [18], the above procedure satisfies local DP and DP in vertical federated learning at the same time.

ℓ_p Moment estimation. We briefly describe how to use pattern counting to develop an algorithm for ℓ_p moment estimation. We assume that each server j has a vector $x^{(j)}$ with non-negative integer coordinates. For each server j, we apply Laplace mechanism to compute $y^{(j)} = x^{(j)} + \boldsymbol{L}^{(j)}$ where $\boldsymbol{L}^{(j)}$ is the vector of Laplace noise. Thus $y^{(j)}$ is differentially private. Let $x = \sum_{j\in[v]} x^{(j)}$ and $y = \sum_{i\in[m]} y^{(j)}$. For large enough entry $y[i] \ge M/2$, $y[i]^p$ is a good approximation to $x[i]^p$, where M is a threshold parameter $\sim \log(m)$. Thus, we are able to estimate the contribution of large entries to the ℓ_p moment up to a multiplicative error. For a small entry $y[i] < M/2$, we know $x[i] \le M$ and thus $(x^{(1)}[i], x^{(2)}[i], \cdots, x^{(v)}[i]) \in \{0, 1, \cdots, M\}^v$. By applying the pattern counting algorithm, we are able to estimate the number of rows $i \in [m]$ satisfying $y[i] < M/2$ and $x[i] = 0, 1, \cdots, M$ respectively. Therefore, we can estimate the contribution of small entries to the ℓ_p moment as well. Since both $y^{(j)}$ and our pattern counting are also local DP, our ℓ_p moment estimation algorithm satisfies both local DP and DP in vertical federated learning simultaneously as well.

k-Means clustering. A common framework to obtain a DP, approximate k-means solution includes two steps. The first step is to compute a good DP bicriteria solution, i.e., a set of $> k$ centers with a good k-means cost. The second step is to map each point to the closest center in the bicriteria solution. Then for each center in the bicriteria solution, we assign it a weight that is a DP approximation of number of points mapped to it. Then running a non-private weighted k-means algorithm on the centers in the bicriteria solution together with the weights provides a DP approximate k-means solution. For the first step, we first run (local) DP k-means algorithm to find an approximate k-means solution $\hat{C}^{(j)}$ for $A^{(j)}$ for each server $j \in [v]$. Then we show that the set of centers

$$\hat{C} = \left\{ \left[\hat{C}^{(1)}[i_1]\ \hat{C}^{(2)}[i_2]\ \cdots\ \hat{C}^{(v)}[i_v] \right] \ \mid\ (i_1, i_2, \cdots, i_v) \in [k]^v \right\}$$

obtained by all possible concatenations of local centers is a good bicriteria k-means solution for the entire dataset A, and each center in $\hat{C}$ can be indexed by a pattern $\rho \in [k]^v$ based on its concatenation. For each server $j \in [v]$, we create a vector $x^{(j)}$ as follows: $x^{(j)}[i] = l \in [k]$ if $\hat{C}^{(j)}[l]$ is the closest center to $A^{(j)}[i]$ among $\hat{C}^{(j)}$. Then the center in $\hat{C}$ indexed by $(x^{(1)}[i], x^{(2)}[i], \cdots, x^{(v)}[i])$ is the closest center to $A[i]$. Therefore the number of appearances of the pattern ρ in $(x^{(1)}, x^{(2)}, \cdots, x^{(v)})$ indicates the number of points in A mapped to the center indexed by ρ in $\hat{C}$. Thus the second step can be done by our pattern counting algorithm.

References

[1] Raman Arora, Jalaj Upadhyay, et al. Differentially private robust low-rank approximation. *Advances in neural information processing systems*, 31, 2018. 19

[2] Maria-Florina Balcan, Travis Dick, Yingyu Liang, Wenlong Mou, and Hongyang Zhang. Differentially private clustering in high-dimensional euclidean spaces. In *International Conference on Machine Learning*, pages 322–331. PMLR, 2017. 19

[3] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson-Lindenstrauss transform itself preserves differential privacy. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 410–419. IEEE Computer Society, 2012. doi: 10.1109/FOCS.2012.67. 19

[4] Jeremiah Blocki, Elena Grigorescu, and Tamalika Mukherjee. Differentially-private sublinear-time clustering. In *IEEE International Symposium on Information Theory (ISIT)*, pages 332–337. IEEE, 2021. 19

[5] Jeremiah Blocki, Elena Grigorescu, Tamalika Mukherjee, and Samson Zhou. How to make your approximation algorithm private: A black-box differentially-private transformation for tunable approximation algorithms of functions with low sensitivity. *arXiv preprint arXiv:2210.03831*, 2022. 19

[6] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005. 19

[7] Zhiqi Bu, Sivakanth Gopi, Janardhan Kulkarni, Yin Tat Lee, Judy Hanwen Shen, and Uthaipon Tantipongpipat. Fast and memory efficient differentially private-sgd via JL projections. In *Advances in Neural Information Processing Systems, NeurIPS 2021, December 6-14, 2021, virtual*, pages 19680–19691, 2021. 19

[8] Alisa Chang, Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Locally private k-means in one round. In *International Conference on Machine Learning*, pages 1441–1451. PMLR, 2021. 13, 15, 19, 20

[9] Weijing Chen, Guoqiang Ma, Tao Fan, Yan Kang, Qian Xu, and Qiang Yang. Secureboost+: A high performance gradient boosting tree framework for large scale vertical federated learning. *arXiv preprint arXiv:2110.10927*, 2021. 19

[10] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems*, 36(6):87–98, 2021. 19

[11] Seung Geol Choi, Dana Dachman-Soled, Mukul Kulkarni, and Arkady Yerukhimovich. Differentially-private multi-party sketching for large-scale statistics. Cryptology ePrint Archive, Paper 2020/029, 2020. URL https://eprint.iacr.org/2020/029. 19

[12] Vincent Cohen-Addad, Alessandro Epasto, Silvio Lattanzi, Vahab Mirrokni, Andres Munoz Medina, David Saulpic, Chris Schwiegelshohn, and Sergei Vassilvitskii. Scalable differentially private clustering via hierarchically separated trees. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 221–230, 2022. 13, 19

[13] Vincent Cohen-Addad, Alessandro Epasto, Vahab Mirrokni, Shyam Narayanan, and Peilin Zhong. Near-optimal private and scalable k-clustering. In *Advances in Neural Information Processing Systems*, 2022. 3, 13, 19, 21

[14] Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1621–1628, 2022. 15, 21

[15] Hu Ding, Yu Liu, Lingxiao Huang, and Jian Li. K-means clustering with distributed dimensions. In *International Conference on Machine Learning*, pages 1339–1348. PMLR, 2016. 19

[16] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008. 1

[17] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *ics*, pages 66–80, 2010. 19

[18] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. 1, 4, 9, 20

[19] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze Gauss: optimal bounds for privacy-preserving principal component analysis. In *STOC'14—Proceedings of the 2014 ACM Symposium on Theory of Computing*, pages 11–20. ACM, New York, 2014. doi: 10.1145/2591796.2591883. 19

[20] Alessandro Epasto, Jieming Mao, Andres Munoz Medina, Vahab Mirrokni, Sergei Vassilvitskii, and Peilin Zhong. Differentially private continual releases of streaming frequency moment estimations. *arXiv preprint arXiv:2301.05605*, 2023. 2, 19

[21] European Commission. Proposal for a regulation of the european parliament and of the council on contestable and fair markets in the digital sector (digital markets act). *SEC (2020) 437 final*, 2020. 1

[22] European Parliament and Council of the European Union. General data protection regulation, 2016. 1

[23] Dan Feldman, Amos Fiat, Haim Kaplan, and Kobbi Nissim. Private coresets. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 361–370. ACM, 2009. 19

[24] Dan Feldman, Chongyuan Xiang, Ruihao Zhu, and Daniela Rus. Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 3–16. IEEE, 2017. 19

[25] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. *Advances in Neural Information Processing Systems*, 33:4040–4054, 2020. 3, 13, 15, 19, 21

[26] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1106–1125. SIAM, 2010. 3, 19

[27] Moritz Hardt and Aaron Roth. Beating randomized response on incoherent matrices. In *STOC'12—Proceedings of the 2012 ACM Symposium on Theory of Computing*, pages 1255–1268. ACM, New York, 2012. doi: 10.1145/2213977.2214088. 2, 3, 16, 19

[28] Daojing He, Runmeng Du, Shanshan Zhu, Min Zhang, Kaitai Liang, and Sammy Chan. Secure logistic regression for vertical federated learning. *IEEE Internet Computing*, 26(2):61–68, 2021. 19

[29] Lingxiao Huang, Zhize Li, Jialin Sun, and Haoyu Zhao. Coresets for vertical federated learning: Regularized linear regression and k-means clustering. *arXiv preprint arXiv:2210.14664*, 2022. 19

[30] Zhiyi Huang and Jinyan Liu. Optimal differentially private algorithms for k-means clustering. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 395–408, 2018. 19

[31] Matthew Jones, Huy L Nguyen, and Thy D Nguyen. Differentially private clustering via maximum coverage. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 19

[32] Ravi Kannan, Santosh Vempala, and David Woodruff. Principal component analysis and higher correlations for distributed data. In *Conference on Learning Theory*, pages 1040–1057. PMLR, 2014. 16

[33] Michael Kapralov and Kunal Talwar. On differentially private low rank approximation. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1395–1414. SIAM, Philadelphia, PA, 2012. 19

[34] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016. 19

[35] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 19

[36] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Ann. Statist.*, 28 (5):1302–1338, 2000. ISSN 0090-5364. doi: 10.1214/aos/1015957395. 16

[37] Yang Liu, Yan Kang, Xinwei Zhang, Liping Li, Yong Cheng, Tianjian Chen, Mingyi Hong, and Qiang Yang. A communication efficient collaborative learning framework for distributed features. *arXiv preprint arXiv:1912.11187*, 2019. 19

[38] Oren Mangoubi and Nisheeth K Vishnoi. Re-analyze gauss: Bounds for private matrix approximation via dyson brownian motion. *arXiv preprint arXiv:2211.06418*, 2022. 19

[39] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil Vadhan. The limits of two-party differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science—FOCS 2010*, pages 81–90. IEEE Computer Soc., Los Alamitos, CA, 2010. 1, 2, 19

[40] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 19

[41] Darakhshan Mir, Shan Muthukrishnan, Aleksandar Nikolov, and Rebecca N Wright. Pan-private algorithms via statistics on sketches. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 37–48, 2011. 19

[42] Prashanth Mohan, Abhradeep Thakurta, Elaine Shi, Dawn Song, and David Culler. Gupt: privacy preserving data analysis made easy. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 349–360, 2012. 19

[43] Huy L. Nguyen, Anamay Chaturvedi, and Eric Z. Xu. Differentially private k-means via exponential mechanism and max cover. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9101–9108. AAAI Press, 2021. 19

[44] Kobbi Nissim and Uri Stemmer. Clustering algorithms for the centralized and local models. In *Algorithmic Learning Theory*, pages 619–653. PMLR, 2018. 13, 19

[45] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM Symposium on Theory of computing (STOC)*, pages 75–84, 2007. 19

[46] Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Locating a small cluster privately. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 413–427, 2016. 19

[47] Richard Nock, Raphaël Canyasse, Roksana Boreli, and Frank Nielsen. k-variates++: more pluses in the k-means++. In *International Conference on Machine Learning*, pages 145–154. PMLR, 2016. 19

[48] Thilina Ranbaduge and Ming Ding. Differentially private vertical federated learning. *arXiv preprint arXiv:2211.06782*, 2022. 1, 19

[49] Or Sheffet. Differentially private ordinary least squares. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3105–3114. PMLR, 2017. 19

[50] Adam D. Smith, Shuang Song, and Abhradeep Thakurta. The Flajolet-Martin sketch itself preserves differential privacy: Private counting with minimal space. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 19

[51] Uri Stemmer. Locally private k-means clustering. *The Journal of Machine Learning Research*, 22(1):7964–7993, 2021. 3, 19, 20

[52] Uri Stemmer and Haim Kaplan. Differentially private k-means with constant multiplicative error. *Advances in Neural Information Processing Systems*, 31, 2018. 3, 13, 19

[53] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *Proceedings of the sixth ACM conference on data and application security and privacy*, pages 26–37, 2016. 19

[54] Zhihua Tian, Rui Zhang, Xiaoyang Hou, Jian Liu, and Kui Ren. Federboost: Private federated learning for gbdt. *arXiv preprint arXiv:2011.02796*, 2020. 19

[55] Jalaj Upadhyay. The price of privacy for low-rank factorization. *Advances in Neural Information Processing Systems*, 31, 2018. 19

[56] Jalaj Upadhyay and Sarvagya Upadhyay. A framework for private matrix analysis. *arXiv preprint arXiv:2009.02668*, 2020. 19

[57] U.S. Department of Health & Human Services. Health Insurance Portability and Accountability Act. https://www.hhs.gov/hipaa/, 1996. 1

[58] Chang Wang, Jian Liang, Mingkai Huang, Bing Bai, Kun Bai, and Hao Li. Hybrid differentially private federated learning on vertically partitioned data. *arXiv preprint arXiv:2009.02763*, 2020. 19

[59] Lun Wang, Iosif Pinelis, and Dawn Song. Differentially private fractional frequency moments estimation with polylogarithmic space. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. 2, 19

[60] Yining Wang, Yu-Xiang Wang, and Aarti Singh. Differentially private subspace clustering. *Advances in Neural Information Processing Systems*, 28, 2015. 19

[61] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965. 2, 4, 9

[62] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Sha Wei, Fan Wu, Guihai Chen, and Thilina Ranbaduge. Vertical federated learning: Challenges, methodologies and experiments. *arXiv preprint arXiv:2202.04309*, 2022. 19

[63] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014. 18

[64] Ruihan Wu, Xin Yang, Yuanshun Yao, Jiankai Sun, Tianyi Liu, Q. Kilian Weinberger, and Chong Wang. Differentially private multi-party data release for linear regression. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*. PMLR, 2022. 1, 3, 19

[65] Kai Yang, Tao Fan, Tianjian Chen, Yuanming Shi, and Qiang Yang. A quasi-newton method based vertical federated learning framework for logistic regression. *arXiv preprint arXiv:1912.00513*, 2019. 19

[66] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019. 1, 19

[67] Jianzhe Zhao, Jiayi Wang, Zhaocheng Li, Weiting Yuan, and Stan Matwin. Vertically federated learning with correlated differential privacy. *Electronics*, 11(23):3958, 2022. 19

A Preliminaries

For an integer M, we define $[M] \coloneqq \{1, 2, \cdots, M\}$. For a vector $a \in \mathbb{R}^d$ and an index $i \in [d]$, we use $a[i]$ to denote the i-th coordinate of the vector a. If there is no ambiguity, we sometimes abuse the notation and also use a_i to denote $a[i]$. For $x, y \in \mathbb{R}$, we use $x \pm y$ to indicate the interval $[x - |y|, x + |y|]$. Let A be an $m \times d$ matrix. We use $A[i]$ to denote the i-th row of A. We sometimes abuse the notation and use A to denote a point set $\{A[1], A[2], \cdots, A[m]\}$ as well. We use $\|A\|_{\mathsf{F}} \coloneqq (\sum_i \sum_j A_{ij}^2)^{1/2}$ to denote the Frobenius norm and $\|A\|_2 \coloneqq \max_x \|Ax\|_2 / \|x\|_2$ to denote the operator norm of A. For $k \leq \min(m, d)$, we use A_k to denote the best rank k approximation of A in Frobenius norm i.e., $\|A - A_k\|_{\mathsf{F}} = \min_{\text{rank-}k\ B} \|A - B\|_{\mathsf{F}}$. By Eckart-Young-Mirsky's theorem, A_k can be obtained by truncating the Singular Value Decomposition (SVD) of A to the top k singular values. We use $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ to denote the maximum and minimum singular values of A respectively. Given a parameter $t > 0$, the Laplace distribution $\text{Lap}(t)$ parameterized by t, has the p.d.f $f(x \mid t) = (1/2t) \cdot \exp(-|x|/t)$. The above distribution has mean 0 and variance $2t^2$.

We first state a basic composition result for differential private mechanisms that we use throughout.

Theorem A.1 (Basic composition [18])**.** *Suppose $\mathcal{M} = (\mathcal{M}_1, \cdots, \mathcal{M}_k)$ is a sequence of differentially private mechanisms over the domain $\mathcal{D}$ potentially chosen* sequentially *and* adaptively*. If the mechanism $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-differentially private for $i = 1, \cdots, k$ respectively, then $\mathcal{M}$ is $(\sum_i \varepsilon_i, \sum_i \delta_i)$-differentially private.*

We will now state techniques to convert any algorithm into a differential private mechanism by adding appropriate noise to the response. Let $f : D \to \{0, 1\}$ be a binary function and the set of databases $\mathcal{D}$ be D^m. For $X \in \mathcal{D}$, define $f(X)$ to be the m-dimensional vector obtained by applying f to each element of X. For $X, X' \in \mathcal{D}$, define $X \sim X'$ if they differ in exactly one position. Given $\varepsilon < 1/4$ and $X \in \mathcal{D}$, let $\text{priv}(f(X), \varepsilon)$ be an m-dimensional random binary vector defined as

$$\mathbf{Pr}[(\text{priv}(f(X), \varepsilon))_i = 1] = \begin{cases} 1/2 + \varepsilon & \text{if } f(X)_i = 1 \\ 1/2 - \varepsilon & \text{if } f(X)_i = 0 \end{cases}.$$

Additionally, the coordinates of $\text{priv}(f(X), \varepsilon)$ are mutually independent. Now a procedure $\mathcal{M}$ that first computes $f(X)$ and then samples a random binary vector from the distribution defined by $\text{priv}(f(X), \varepsilon)$ is $O(\varepsilon)$-differentially private. This mechanism is called *randomized response* and is one of the basic ways to obtain differentially private mechanisms in a black-box way. Randomized response can be used to reveal a binary vector in a differentially private way while still being able to use some useful statistics about the original binary vector from the revealed binary vector.

Theorem A.2 ([18, 61])**.** *For $\varepsilon < 1/4$, randomized response is 8ε-differentially private.*

Remark A.3. The randomized response mechanism can be modified to accommodate all the values of $\varepsilon \geq 0$. We use this formulation for convenience in our proofs.

We now discuss mechanisms that can be used to make more general functions private in a black-box way. Let $f : \mathcal{D} \to \mathbb{R}^m$ be an arbitrary function. Define sensitivity $\Delta_{\|\cdot\|} f \coloneqq \max_{X, X' \in \mathcal{D},\, X \sim X'} \|f(X) - f(X')\|$. Here $\|\cdot\|$ is any arbitrary norm on $\mathbb{R}^m$. For convenience, we use $\Delta_1 f$ for the ℓ_1 sensitivity $\Delta_{\|\cdot\|_1} f$ and $\Delta_2 f$ for the ℓ_2 sensitivity $\Delta_{\|\cdot\|_2} f$.

Definition A.4 (Laplace and Gaussian mechanism)**.** Consider an arbitrary function $f : \mathcal{D} \to \mathbb{R}^m$. The Laplace mechanism is defined as $\mathcal{M}_L(X, f, \varepsilon) = f(X) + (Y_1, \cdots, Y_k)$. Here each random variable Y_i is sampled independently from $\text{Lap}(\Delta_1 f / \varepsilon)$. The Gaussian mechanism is defined as $\mathcal{M}_G(X, f, (\varepsilon, \delta)) = f(X) + (G_1, \cdots, G_k)$ where G_i is independently sampled from $N(0, \sigma^2)$ with $\sigma^2 = 2 \log(1.25/\delta)(\Delta_2 f)^2 / \varepsilon^2$.

Theorem A.5 ([18])**.** *For any f, ε, δ, the Laplace Mechanism $\mathcal{M}_L(\cdot, f, \varepsilon)$ is ε differentially private and the Gaussian Mechanism $\mathcal{M}_G(\cdot, f, (\varepsilon, \delta))$ is (ε, δ) differentially private.*

B Pattern Counting

In this section, we state the guarantees of our novel differentially private algorithms for the pattern counting problem in the vertical federated learning model.

Algorithm 1: Pattern Counting via Randomized Responses

Input: An $x^{(j)} \in \mathbb{R}^m$ for each server j, an upper bound $M \in \mathbb{Z}_{\geq 1}$ and a privacy parameter ε
Output: $\{\overline{\text{count}}_\rho \mid \rho \in ([M] \cup 0)^v\}$

for *each server* $j = 1, \cdots, v$ *concurrently* **do**
  for $k = 0, \cdots, M$ **do**
    For all $i \in [m]$, set $a_k^{(j)}[i] \leftarrow \mathbb{1}[x^{(j)}[i] = k]$
    $b_k^{(j)} \leftarrow \text{priv}(a_k^{(j)}, \varepsilon')$ for $\varepsilon' \leftarrow \varepsilon/16$ `// Randomized response (see § A).`
  Send $(b_0^{(j)}, \cdots, b_M^{(j)})$ to the Central Coordinator
At the Central Coordinator:
Collect the binary vectors sent by the servers
for $\rho \in ([M] \cup \{0\})^v$ **do**
  For all $i \in [m]$, set $\boldsymbol{X}_\rho[i] \leftarrow \prod_{j \in [m]} (b^{(j)}_{\rho^{(j)}}[i] - 1/2 + \varepsilon')/(2\varepsilon')$
  $\overline{\text{count}}_\rho \leftarrow \max(0, \sum_{i \in [m]} \boldsymbol{X}_\rho[i])$
return $\{\overline{\text{count}}_\rho \mid \rho \in ([M] \cup \{0\})^v\}$

Theorem B.1 (Pattern counting)**.** *Given a vector $x^{(j)} \in \mathbb{R}^m$ for each server $j = 1, \cdots, v$, a privacy parameter $\varepsilon \in (0, 0.5)$, a probability parameter $\gamma \in (0, 0.5)$ and an upper bound $M \in \mathbb{Z}_{\geq 0}$, Algorithm 1 is ε-DP in the vertical federated learning model and computes an estimate $\overline{\text{count}}_\rho$ for each $\rho \in \{0, 1, \cdots, M\}^v$. With probability $\geq 1 - \gamma$, for all $\rho \in \{0, 1, \cdots, M\}^v$, $\overline{\text{count}}_\rho \in \text{count}_\rho \pm C_v \sqrt{m} \cdot \ln(M^v/\gamma)/\varepsilon^v$ where C_v is a factor which only depends on v. The total size of messages sent by each server is $O(M \cdot m)$.*

Proof. Firstly, let us consider privacy. For each server j, we define binary vectors $a_k^{(j)}$ for $k = 0, \cdots, M$ as follows:

$$a_k^{(j)}[i] = \begin{cases} 1 & x^{(j)}[i] = k \\ 0 & \text{otherwise} \end{cases}.$$

We now note that for each server j, for neighboring vectors $x^{(j)}$ and $(x^{(j)})'$ that differ in at most one coordinate, the group of vectors $(a_0^{(j)}, \ldots, a_M^{(j)})$ and $((a_0^{(j)})', \ldots, (a_M^{(j)})')$ defined using the above definition for $x^{(j)}$ and $(x^{(j)})'$ differ in at most two coordinates. Let $\varepsilon' = \varepsilon/16$. It is clear that $\varepsilon' \leq 1/4$. Thus considering the group of vectors $(a_0^{(j)}, \ldots, a_M^{(j)})$ to be a binary vector with $m \cdot (M+1)$ coordinates and using the fact that the vectors differ in at most 2 coordinates if they are created from neighboring datasets, we obtain that releasing $b_k^{(j)} = \text{priv}(a_k^{(j)}, \varepsilon')$ for each k is $(16\varepsilon')$-differentially private and therefore is ε-DP.

Next, let us consider the communication cost. Since each server j releases $(b_0^{(j)}, \cdots, b_M^{(j)})$, where each vector is m-dimensional, the total size of the messages is at most $O(m \cdot M)$.

Finally, let us consider the approximation guarantee. We note that the coordinates of the vector $b_k^{(j)}$ are independent by definition of $\text{priv}(\cdot, \varepsilon')$. For each $i \in [m]$,

$$\mathbf{E}[b_k^{(j)}[i]] = (1/2 - \varepsilon') + (2\varepsilon')\mathbb{1}[x^{(j)}[i] = k].$$

which by linearity implies that $\mathbf{E}\left[\frac{(b_k^{(j)}[i] - 1/2 + \varepsilon')}{2\varepsilon'}\right] = \mathbb{1}[x^{(j)}[i] = k]$. Let $\rho \in \{0, 1, \cdots, M\}^v$ be an arbitrary pattern. We now have

$$\mathbf{E}\left[\prod_{j \in [v]} \frac{b^{(j)}_{\rho^{(j)}}[i] - 1/2 + \varepsilon'}{2\varepsilon'}\right] = \prod_{j \in [v]} \mathbb{1}[x^{(j)}[i] = \rho^{(j)}]$$

where we used the fact that the random variables $b^{(1)}_{\rho^{(1)}}[i], \cdots, b^{(v)}_{\rho^{(v)}}[i]$ are mutually independent. We now define the random variable

$$\boldsymbol{X}_\rho[i] \coloneqq \prod_{j \in [v]} \frac{b^{(j)}_{\rho^{(j)}}[i] - 1/2 + \varepsilon'}{2\varepsilon'}$$

and note that $|\boldsymbol{X}_\rho[i]| \le \left(\frac{1}{2\varepsilon'}\right)^v$ with probability 1 since $|b^{(j)}_{\rho^{(j)}} - 1/2 + \varepsilon'| \le 1/2 + \varepsilon'$ with probability 1. By linearity of expectation, $\mathbf{E}[\sum_{i\in[m]} \boldsymbol{X}_\rho[i]] = \text{count}_\rho$. Using the upper bound on $|\boldsymbol{X}_\rho[i]|$ and the fact that the random variables $\boldsymbol{X}_\rho[1], \cdots, \boldsymbol{X}_\rho[m]$ are mutually independent, we obtain

$$\mathbf{Pr}[|\sum_{i\in[m]} \boldsymbol{X}_\rho[i] - \text{count}_\rho| \ge t] \le 2\exp\left(-\frac{2t^2}{4m/(2\varepsilon')^{2v}}\right).$$

Picking $t = O(\sqrt{m}\ln(1/q)/(2\varepsilon')^v)$, we obtain that with probability $\ge 1-q$, $\sum_{i\in[m]} \boldsymbol{X}_\rho[i] \in \text{count}_\rho \pm O(\sqrt{m}\ln(1/q)/(2\varepsilon')^v)$. Setting $q = \gamma/(M+1)^v$, by a union bound, with probability $\ge 1-\gamma$, for all patterns $\rho \in ([M]\cup\{0\})^v$,

$$\sum_{i\in[m]} \boldsymbol{X}_\rho[i] \in \text{count}_\rho \pm O\left(\frac{\sqrt{m}\ln(M^v/\gamma)}{(2\varepsilon')^v}\right).$$

By plugging $\varepsilon' = \varepsilon/16$ into above formula, we complete the proof. □

We use sampling to reduce the communication of Algorithm 1 and get the following theorem.

Theorem B.2 (Low communication pattern counting)**.** *Given a vector $x^{(j)} \in \mathbb{R}^m$ for each server $j = 1, \cdots, v$, a privacy parameter $\varepsilon \in (0, 0.5)$, a probability parameter $\gamma \in (0, 0.5)$, an upper bound $M \in \mathbb{Z}_{\ge 0}$, and a communication-approximation trade-off parameter $p \in (\Omega(\log(1/\gamma)/m), 1]$, there is an ε-DP algorithm in the vertical federated learning model and computes an estimate $\overline{\text{count}}_\rho$ for each $\rho \in \{0, 1, \cdots, M\}^v$. With probability at least $1-\gamma$, for all $\rho \in \{0, 1, \cdots, M\}^v$, $\overline{\text{count}}_\rho \in \text{count}_\rho \pm C_v\sqrt{m/p}\cdot\ln(M^v/\gamma)/\varepsilon^v$ where C_v is a factor which only depends on v. In addition, the total size of messages sent by each server is at most $O(M\cdot pm)$.*

Proof. In Algorithm 1, each server j sends the vectors m dimensional binary vectors $b^{(j)}_0, \cdots, b^{(j)}_M$ and we argued that with probability $1-\gamma$, for each pattern ρ, the m dimensional random vector $\boldsymbol{X}_\rho[i]$ satisfies $\sum_{i=1}^m \boldsymbol{X}_\rho[i] = \text{count}_\rho \pm \sqrt{m}\cdot\ln(1/\gamma)/\varepsilon^v$. Crucially, we note that the value of $\boldsymbol{X}_\rho[i]$ depends only on the i-th coordinates of the vectors $b^{(j)}_k$ for $j = 1, \cdots, v$ and $k = 0, \cdots, M$. Let $\boldsymbol{S} \subseteq [m]$ be a random subset of coordinates sampled such that each $i \in [m]$ is in $i \in \boldsymbol{S}$ independently with probability p. With high probability, we have $|\boldsymbol{S}| = O(pm)$ and $(1/p)\,\mathbf{E}_{\boldsymbol{S}}[\sum_{i\in\boldsymbol{S}} \boldsymbol{X}_\rho[i]] = \sum_{i\in[m]} \boldsymbol{X}_\rho[i]$. Hence, sampling a few coordinates and sending the corresponding values of the vectors $b^{(j)}_k$ is sufficient to estimate $\sum_{i\in[m]} \boldsymbol{X}_\rho[i]$ and hence count_ρ. We use the fact that $|\boldsymbol{X}_\rho[i]| \le (8/\varepsilon)^v$ to obtain tail bounds on $|\sum_{i\in\boldsymbol{S}}(1/p)\boldsymbol{X}_\rho[i] - \sum_{i\in[m]} \boldsymbol{X}_\rho[i]|$.

By using Bernstein's inequality,

$$\mathbf{Pr}\left[\left|(1/p)\sum_{i\in\boldsymbol{S}} \boldsymbol{X}_\rho[i] - \sum_{i\in[m]} \boldsymbol{X}_\rho[i]\right| \ge t\right] \le 2\exp\left(-\frac{-t^2/2}{mT^2/p + Tt/3p}\right)$$

where $T = (8/\varepsilon)^v$. For $t = O(T\sqrt{(m/p)\log(1/\gamma')})$ with γ' and p satisfying $\log(1/\gamma')/p \le m$, we have $\mathbf{Pr}[|(1/p)\sum_{i\in\boldsymbol{S}} \boldsymbol{X}_\rho[i] - \sum_{i\in[m]} \boldsymbol{X}_\rho[i]| \ge O(T\sqrt{m/p}\log(1/\gamma'))] \le \gamma'$. By picking $\gamma' = \gamma/((M+1)^v)$, we can union bound over the at most $(M+1)^v$ values of ρ and obtain that with a probability $1-2\gamma$, for each $\rho \in ([M]\cup\{0\})^v$, $(1/p)\sum_{i\in\boldsymbol{S}} \boldsymbol{X}_\rho[i] \in \text{count}_\rho \pm C_v\,(1/\varepsilon)^{O(v)}\ln(1/\gamma)(\sqrt{m/p} + \sqrt{m})$. Furthermore, if $pm = \Omega(\log(1/\gamma))$, with probability at least $1-\gamma$, $|S| \le O(pm)$ and thus the total size of messages sent by a server is at most $O(pm\cdot M)$. □

C ℓ_p Moment Estimation

In this section, we consider ℓ_p moment estimation problem in vertical federated learning. Each server $j \in [v]$, has a nonnegative vector $x^{(j)}$ with integer coordinates and we want to estimate $\|\sum_{j=1}^v x^{(j)}\|_p^p$. When $p = 1$, since $x^{(1)}, x^{(2)}, \cdots, x^{(v)}$ are non-negative, $\|\sum_j x^{(j)}\|_1 = \sum_j \|x^{(j)}\|_1$. Server j computes $\|x^{(j)}\|_1$ and then sends $\|x^{(j)}\|_1 + \text{Lap}(1/\varepsilon)$ to the central coordinator. This mechanism is ε-DP as the function $\|\cdot\|_1$ has an ℓ_1 sensitivity of 1. Thus $\|\sum_j x^{(j)}\|_1$ can be estimated up to an additive error of $O(\sqrt{v}/\varepsilon)$ with probability at least 0.99 by an ε-DP algorithm in the vertical federated learning model. For $p = 2$, we present a simple sketching and perturb technique to solve the ℓ_2 moment estimation problem in Appendix E. However, the technique does not generalize to the case when $p \ne 2$. In this section, we show how to get a good ℓ_p moment estimation for general p.

For general p, we look towards an algorithm based on input perturbation. A simple differentially private algorithm that uses the Laplace mechanism is as follows: server j adds a Laplace noise vector $\boldsymbol{L}^{(j)} \in \mathbb{R}^m$ with each coordinate

Algorithm 2: ℓ_p Moment Estimation

Input: $x^{(j)} \in \mathbb{Z}^m_{\geq 0}$ for each server $j \in [v]$, privacy parameter $\varepsilon \in (0, 0.5)$, approximation parameter $\alpha \in (0, 0.5)$, and $p > 0$

Output: $\hat{C} \subset \mathbb{R}^d$

for *each server* $j = 1, \cdots, v$ *concurrently* **do**
  Compute $y^{(j)} = x^{(j)} + \boldsymbol{L}^{(j)}$ where each entry of $\boldsymbol{L}^{(j)} \in \mathbb{R}^m$ is an i.i.d. Lap$(2/\varepsilon)$
  Send $y^{(j)}$ to the central coordinator

At the Central Coordinator:

Collect $y^{(j)}$ sent by the servers and compute $y = \sum_{j\in[v]} y^{(j)}$

Let $M = \Theta(v \log(mv)/(\alpha\varepsilon))$

Compute $L = \{i \in [m] \mid y[i] \geq M/2\}$ and send L and M back to each server

For Central Coordinator and all servers:

Run pattern counting (Algorithm 1) for $x^{(1)}_{\setminus L}, \cdots, x^{(v)}_{\setminus L}$, M and privacy parameter $\varepsilon/2$

At the Central Coordinator:

Get $\overline{\text{count}}_\rho$ for every $\rho \in \{0, 1, \cdots, M\}^v$ such that $\overline{\text{count}}_\rho \in \text{count}_\rho \pm \sqrt{m} \cdot \left(\frac{1}{\varepsilon}\right)^v \cdot O_v(\ln M)$ (the output of pattern counting, Theorem B.1)

return $\sum_{i\in L} y[i]^p + \sum_{\rho\in\{0,1,\cdots,M\}^v} \overline{\text{count}}_\rho \cdot \left(\sum_{j\in[v]} \rho^{(j)}\right)^p$

of $\boldsymbol{L}^{(j)}$ being an independent Laplace random variable with parameter $1/\varepsilon$ and sends $x^{(j)} + \boldsymbol{L}^{(j)}$ to the central coordinator. Clearly, the message $y^{(j)} = x^{(j)} + \boldsymbol{L}^{(j)}$ sent by server j is ε-DP with respect to $x^{(j)}$ for $j \in [v]$. The central coordinator then computes the vector $y = \sum_{j=1}^v y^{(j)}$. We need to use the following lemma.

Lemma C.1. *With probability* ≥ 0.99, $\|\sum_{j\in[v]} \boldsymbol{L}^{(j)}\|_\infty \leq (Cv/\varepsilon)\ln(mv)$, *for some constant* C.

Proof. For $j \in [v]$ and $i \in [m]$, the coordinates $\boldsymbol{L}^{(j)}[i]$ are independent random variables drawn from Lap$(1/\varepsilon)$. For any fixed i, j, with probability $\geq 1 - 1/(100(mv)^2)$, $|\boldsymbol{L}^{(j)}[i]| \leq (C/\varepsilon)\ln(mv)$ for a large enough constant C. By a union bound, we obtain that with probability $\geq 99/100$, $\max_{i,j} |\boldsymbol{L}^{(j)}[i]| \leq (C/\varepsilon)\ln(mv)$. Thus

$$\|\sum_{j=1}^v \boldsymbol{L}^{(j)}\|_\infty = \max_i |\sum_{j=1}^v \boldsymbol{L}^{(j)}[i]| \leq (Cv/\varepsilon)\ln(mv).$$

□

Thus if $x[i] = \sum_{j=1}^v x^{(j)}[i]$ is already large enough, i.e, $\Omega((Cv/(\varepsilon\alpha))\ln(mv))$ for some multiplicative error parameter $\alpha \in (0, 0.5)$, then the amount of noise added does not distort $x[i]$ by an α fraction from the above lemma. Moreover, looking at the values of $y[i]$, the central coordinator can determine which coordinates of x are large. Let the set of coordinates that are determined as large be L. For $i \in L$, we have that $y[i] \in (1 \pm \alpha)x[i]$ and therefore the coordinator gets a $(1 \pm \alpha)^p$ approximation for $\|x_L\|_p^p$. For all $i \notin L$, we have $x[i] \leq O((v/\varepsilon\alpha)\ln(mv))$. Now, the Central Coordinator uses the pattern counting algorithm to approximately count the number of indices i with $x[i]$ value $1, 2, \cdots, O((v/\varepsilon\alpha)\ln(mv))$ respectively and uses the approximate counts to obtain an estimation for $\sum_{i\notin L}(\sum_{j=1}^v x^{(j)}[i])^p$ and overall approximation $\|x\|_p^p = \|x_L\|_p^p + \|x_{\setminus L}\|_p^p$. We have the following theorem. We describe the full procedure in Algorithm 2.

Theorem C.2 (ℓ_p Moment estimation)**.** *Suppose there are* v *servers where each server* j *has a non-negative integer vector* $x^{(j)}$ *of* m *dimensions. Given* $p > 0$, $\alpha \in (0, 0.5)$ *and* $\varepsilon \in (0, 0.5)$, *there is an* ε*-DP algorithm in the vertical federated learning model and outputs an estimation* Est *that with probability* ≥ 0.9 *satisfies* $\text{Est} \in (1 \pm \alpha)^p\|x\|_p^p \pm C'_v\sqrt{m}\varepsilon^{-(p+v)} \cdot (v\log(mv)/\alpha)^p \,\text{polylog}(\log m, 1/\varepsilon, 1/\alpha)$, *where* C'_v *is a factor that depends only on* v. *In addition, the total size of messages sent by each server is at most* $O(vm\log(mv)/(\alpha\varepsilon))$.

Proof. Let the event that $\|\sum_{j=1}^v \boldsymbol{L}^{(j)}\|_\infty \leq (Cv/\varepsilon)\ln(mv)$ be $\mathcal{E}_1$ and let $\alpha \in (0, 1/2]$ be the given *multiplicative* approximation parameter. The set L computed by Algorithm 2 is defined as

$$L \coloneqq \{i \in [m] \mid y[i] \geq (Cv/\alpha\varepsilon)\ln mv\}.$$

Let $x := \sum_{j\in[v]} x^{(j)}$. Conditioned on the event $\mathcal{E}_1$, we have for all $i \in L$ that $x[i] \geq y[i] - \|\sum_{j=1}^v \boldsymbol{L}^{(j)}\|_\infty \geq (1-\alpha)y[i]$ and $x[i] \leq y[i] + \|\sum_{j=1}^v \boldsymbol{L}^{(j)}\|_\infty \leq (1+\alpha)y[i]$. Thus for all $i \in L$, the value $y[i]$ is a good approximation for $x[i]$. We

further have that conditioned on $\mathcal{E}_1$, for all $i \notin L$, $x[i] \le y[i] + \|\sum_{j=1}^{v} \boldsymbol{L}^{(j)}\|_\infty \le (Cv/(\varepsilon\alpha))\ln(mv) + (Cv/\varepsilon)\ln(mv) \le (2Cv/(\varepsilon\alpha))\ln(mv)$.

Let x_L be the vector satisfying $\forall i \in L, x_L[i] = x[i]$ and $\forall i \notin L, x_L[i] = 0$. Let $x_{\setminus L} = x - x_L$. Thus, $\|x\|_p^p = \|x_L\|_p^p + \|x_{\setminus L}\|_p^p$. Let $M := \lceil (2Cv/(\varepsilon\alpha))\ln(mv) \rceil$. From above $\sum_{i\in L} y[i]^p$ gives a $(1\pm\alpha)^p$ approximation for $\|x_L\|_p^p$ and $\|x_{\setminus L}\|_p^p = \sum_{\ell=0}^{M} |\{i \notin L \mid x[i] = \ell\}| \cdot \ell^p$. Define $\mathrm{count}_\ell := |\{i \notin L \mid x[i] = \ell\}|$. Now approximating count_ℓ for all $0 \le \ell \le M$ suffices to approximate $\|x_{\setminus L}\|_p^p$. Let $\rho = (\rho^{(1)}, \ldots, \rho^{(v)})$: $\rho^{(j)} \in [M] \cup 0$ for all j and $\sum_{j=1}^{v} \rho^{(j)} \le M$. Given ρ, we define $\mathrm{count}_\rho := |\{i \notin L \mid \forall j \in [v] : x^{(j)}[i] = \rho^{(j)}\}|$ which implies $\mathrm{count}_\ell = \sum_{\rho\,:\,(\sum_{j=1}^{v}\rho^{(j)})=\ell} \mathrm{count}_\rho$.

Setting the failure probability $\gamma = 0.01$ and privacy parameter as ε in Theorem B.1, we obtain estimates $\overline{\mathrm{count}}_\rho$ for all the required patterns and then obtain that

$$\|x_{\setminus L}\|_p^p = \sum_\rho \mathrm{count}_\rho \cdot \left(\sum_{j=1}^{v} \rho^{(j)}\right)^p \in \sum_\rho \overline{\mathrm{count}}_\rho \cdot \left(\sum_{j=1}^{v} \rho^{(j)}\right)^p \pm \sqrt{m} \cdot \left(\frac{1}{\varepsilon}\right)^v \cdot O_v(\ln(M)) \cdot M^p.$$

Noting that $M = O(v\ln(mv)/(\varepsilon\alpha))$, we obtain

$$\|x_{\setminus L}\|_p^p = \sum_\rho \overline{\mathrm{count}}_\rho \left(\sum_{j=1}^{v} \rho^{(j)}\right)^p \pm \sqrt{m} \cdot \left(\frac{v\log(mv)}{\alpha}\right)^p \cdot \left(\frac{1}{\varepsilon}\right)^{p+v} \cdot C'_v \cdot \mathrm{polylog}(\log m, 1/\varepsilon, 1/\alpha).$$

Finally, combined with the approximation for $\|x_L\|_p^p$, we obtain an estimator Est that satisfies

$$\mathrm{Est} \in (1\pm\alpha)^p \|x\|_p^p \pm \sqrt{m} \cdot \left(\frac{v\log(mv)}{\alpha}\right)^p \cdot \left(\frac{1}{\varepsilon}\right)^{p+v} \cdot C'_v \cdot \mathrm{polylog}(\log m, 1/\varepsilon, 1/\alpha).$$

The first stage of the algorithm that computes the set of large coordinates L is ε differentially private by Laplace Mechanism. The second stage of the algorithm, that runs Algorithm 1 with privacy parameter ε is ε differential private by composition of differential privacy. Finally, the algorithm is overall 2ε differentially private and we obtain the theorem statement by rescaling ε. □

D Euclidean k-Clustering

In this section, we present a differentially private k-clustering algorithm in the vertical federated learning model. The k-clustering problem is stated as the follows. Given input $A \in \mathbb{R}^{m\times d}$ where each row $A[i]$ denotes a d-dimensional point, the goal of k-clustering problem is to find a set of k centers $C \in \mathbb{R}^{k\times d}$ such that $\mathrm{cost}_p(A, C) := \sum_{i\in[m]} \min_{j\in[k]} \|A[i] - C[j]\|_2^p$ is minimized where $p \ge 1$ is some constant. In some cases, each point in A is assigned by a non-negative weight $w(\cdot)$, and the k-clustering cost is written as $\mathrm{cost}_{w,p}(A, C) := \sum_{i\in[m]} w[i] \cdot \min_{j\in[k]} \|A[i] - C[j]\|_2^p$. Note that the problem is known as k-means for $p = 2$, and it is known as k-median for $p = 1$. In the vertical federated learning model, A is partitioned vertically over v servers, i.e., $A = [A^{(1)} \cdots A^{(v)}]$, and we want to develop a differentially private algorithm such that the central coordinator learns (approximate) centers at the end of the algorithm. Furthermore, $\|A[i]\|_2$ for each row i is known to be at most Δ, i.e., every data point is guaranteed to be in a ball with radius Δ. A neighboring dataset $A'^{(j)}$ of $A^{(j)}$ has at most one different row from $A^{(j)}$. The difference can be arbitrary but it should satisfy that each row of $A' = [A'^{(1)}\ A'^{(2)} \cdots A'^{(v)}]$ is still in the ball with radius Δ. This is a common assumption in the literature of differentially private k-clustering (see e.g., [52, 44, 8, 25, 12, 13]). In the following, we give a formal definition of the approximate solutions to k-clustering.

Definition D.1. Let $A \in \mathbb{R}^{m\times d}$, $w : [m] \to \mathbb{R}_{\ge 0}$ and $C^* \in \mathbb{R}^{k\times d}$ be the optimal k-clustering solution with respect to (A, w) for power p. If C satisfies $\mathrm{cost}_{w,p}(A, C) \le \gamma \mathrm{cost}_{w,p}(A, C^*) + \eta$ and C contains exactly k centers, we say C is a (γ, η)-approximation for ℓ_p k-clustering of (A, w). If $\eta = 0$, we say C is a γ-approximation. If $\mathrm{cost}_{w,p}(A, C)$ satisfies the approximation guarantee but C has $k' > k$ centers, then C is a (γ, η)-bicriteria approximation (or γ-bicriteria approximation).

Theorem D.2 (Bicriteria approximate k-clustering)**.** *The computation of $\hat{C}$ in Algorithm 3 is $(\varepsilon/2, \delta)$-DP in the vertical federated learning model. Furthermore, $\hat{C}$ is a $(\max(\zeta, \zeta^{-1})\gamma, \max(\zeta, 1)\eta)$-bicriteria approximation for ℓ_p k-clustering of A where $A = \left[A^{(1)}\ A^{(2)} \cdots A^{(v)}\right] \in \mathbb{R}^{m\times d}$, $\gamma = \max_{j\in[v]} \gamma^{(j)}, \eta = \sum_{j\in[v]} \eta^{(j)}$ and $\zeta = v^{p/2-1}$. The total size of messages sent by each server j for computing $\hat{C}$ is at most $O(k \cdot d^{(j)}) = O(kd)$.*

Proof. Firstly, let us consider the privacy guarantee. To compute $\hat{C}$, since each server only sends an $(\varepsilon/2,\delta)$-DP approximate k-clustering solution, the whole algorithm for computing $\hat{C}$ is $(\varepsilon/2,\delta)$-DP in the vertical federated learning.

The communication cost follows from the size of $\hat{C}^{(j)}$.

Next, let us consider the approximation guarantee. Let $C^* = \begin{bmatrix} C^{*(1)} & C^{*(2)} & \cdots & C^{*(v)} \end{bmatrix} \in \mathbb{R}^{k\times d}$ be the optimal ℓ_p k-clustering for A, where $\forall j \in [v]$, $C^{*(j)} \in \mathbb{R}^{m\times d^{(j)}}$. Let $\forall i \in [m], \phi(i) := \arg\min_{b\in[k]} \|A[i] - C^*[b]\|_2$. Then we have:

$$
\begin{aligned}
&\mathrm{cost}_p(A,\hat{C}) \\
&= \sum_{i\in[m]} \min_{b\in[|\hat{C}|]} \|A[i]-\hat{C}[b]\|_2^p = \sum_{i\in[m]} \left(\min_{b\in[|\hat{C}|]} \|A[i]-\hat{C}[b]\|_2^2\right)^{p/2} \\
&= \sum_{i\in[m]} \left(\sum_{j\in[v]} \min_{b\in[k]} \|A^{(j)}[i]-\hat{C}^{(j)}[b]\|_2^2\right)^{p/2} = \sum_{i\in[m]} \left(\sum_{j\in[v]} \left(\min_{b\in[k]} \|A^{(j)}[i]-\hat{C}^{(j)}[b]\|_2^p\right)^{2/p}\right)^{p/2} \\
&\le v^{\max(0,p/2-1)} \sum_{j\in[v]} \sum_{i\in[m]} \min_{b\in[k]} \|A^{(j)}[i]-\hat{C}^{(j)}[b]\|_2^p \\
&\le v^{\max(0,p/2-1)} \sum_{j\in[v]} \left(\eta^{(j)} + \gamma^{(j)} \sum_{i\in[m]} \|A^{(j)}[i]-C^{*(j)}[\phi(i)]\|_2^p\right) \\
&= v^{\max(0,p/2-1)} \left(\sum_{j\in[v]} \eta^{(j)} + \left(\max_{j\in[v]} \gamma^{(j)} \sum_{i\in[m]} \sum_{j\in[v]} \|A^{(j)}[i]-C^{*(j)}[\phi(i)]\|_2^p\right)\right) \\
&\le v^{\max(0,p/2-1)} \left(\sum_{j\in[v]} \eta^{(j)} + \left(\max_{j\in[v]} \gamma^{(j)} \cdot v^{\max(0,1-p/2)} \sum_{i\in[m]} \|A[i]-C^*[\phi(i)]\|_2^p\right)\right) \\
&= v^{\max(p/2-1,1-p/2)} \max_{j\in[v]} \gamma^{(j)} \cdot \mathrm{cost}_p(A,C^*) + v^{\max(0,p/2-1)} \sum_{j\in[v]} \eta^{(j)},
\end{aligned}
$$

where the first inequality follows from Holder's inequality and convexity, the second inequality follows from that $\hat{C}^{(j)}$ is a $(\gamma^{(j)},\eta^{(j)})$-approximation for $A^{(j)}$, and the third inequality follows from Holder's inequality and concavity. □

Theorem D.3 (Approximate k-clustering)**.** *Algorithm 3 is (ε,δ)-DP in the vertical federated learning model. Furthermore, the output $\bar{C}$ is a $(O_p(\max(\zeta,\zeta^{-1})\gamma), O_p(\max(\zeta,1)\eta + k^v\nu\Delta^p))$-approximation for ℓ_p k-clustering of A where $A = \begin{bmatrix} A^{(1)} & A^{(2)} & \cdots & A^{(v)} \end{bmatrix} \in \mathbb{R}^{m\times d}$, $\gamma = \max_{j\in[v]} \gamma^{(j)}, \eta = \sum_{j\in[v]} \eta^{(j)}$ and $\zeta = v^{p/2-1}$. The total size of messages sent by each server j is at most $O(kd^{(j)})$ plus the size of the messages needed for pattern counting.*

Proof. Let us first consider the privacy guarantee. Since $\forall j \in [v]$, $\hat{C}^{(j)}$ is $(\varepsilon/2,\delta)$-DP and the messages sent by server j for pattern counting is $(\varepsilon/2)$-DP, the whole algorithm is (ε,δ)-DP due to composition.

The communication cost is only from two parts: (1) sending $\hat{C}^{(j)}$, (2) running pattern counting. Thus, the total size of messages sent by server j is at most $O(kd^{(j)})$ plus the size of messages needed for pattern counting.

Next, let us consider the approximation guarantee. Let C^* be the optimal ℓ_p k-clustering solution for A. Let $\forall i \in [m], \psi(i) := \arg\min_{(i_1,i_2,\cdots,i_v)\in[k]^v} \|A[i] - \hat{C}[i_1,i_2,\cdots,i_v]\|$. Let $w[\rho] := \mathrm{count}_\rho$ for every $\rho = (i_1,i_2,\cdots,i_v) \in [k]^m$, i.e., $w[(i_1,i_2,\cdots,i_v)] = |\{i \in [m] \mid \psi(i) = (i_1,i_2,\cdots,i_v)\}|$. Let $\gamma' = \max(v^{p/2-1}, v^{1-p/2}) \cdot \max_{j\in[v]} \gamma^{(j)}, \eta' = \max(1, v^{p/2-1}) \cdot \sum_{j\in[v]} \eta^{(j)}$. We have

$$
\begin{aligned}
&\mathrm{cost}(A,\bar{C}) \\
&= \sum_{i\in[m]} \min_{b\in[k]} \|A[i]-\bar{C}[b]\|_2^p \\
&\le \sum_{i\in[m]} (\min_{b\in[k]} \|\hat{C}[\psi(i)]-\bar{C}[b]\|_2 + \|A[i]-\hat{C}[\psi(i)]\|_2)^p \\
&\le 2^{p-1}(\sum_{i\in[m]} \min_{b\in[k]} \|\hat{C}[\psi(i)]-\bar{C}[b]\|_2^p + (\gamma'\mathrm{cost}_p(A,C^*)+\eta'))
\end{aligned}
$$

$$
\begin{aligned}
&= 2^{p-1}(\sum_{l\in[k]^v} w[l]\cdot\min_{b\in[k]}\|\hat{C}[l]-\bar{C}[b]\|_2^p + (\gamma'\mathrm{cost}_p(A,C^*)+\eta')) \\
&\le 2^{p-1}(\sum_{l\in[k]^v} \hat{w}[l]\cdot\min_{b\in[k]}\|\hat{C}[l]-\bar{C}[b]\|_2^p + \sum_{l\in[k]^v}|\hat{w}[l]-w[l]|\cdot\Delta^p + (\gamma'\mathrm{cost}_p(A,C^*)+\eta')) \\
&\le O(2^p)(\sum_{l\in[k]^v} \hat{w}[l]\cdot\min_{b\in[k]}\|\hat{C}[l]-C^*[b]\|_2^p + k^v\nu\Delta^p + (\gamma'\mathrm{cost}_p(A,C^*)+\eta')) \\
&\le O(2^p)(\sum_{l\in[k]^v} w[l]\cdot\min_{b\in[k]}\|\hat{C}[l]-C^*[b]\|_2^p + k^v\nu\Delta^p + (\gamma'\mathrm{cost}_p(A,C^*)+\eta')) \\
&= O(2^p)(\sum_{i\in[m]} \min_{b\in[k]}\|\hat{C}[\psi(i)]-C^*[b]\|_2^p + k^v\nu\Delta^p + (\gamma'\mathrm{cost}_p(A,C^*)+\eta')) \\
&\le O(2^p)(\sum_{i\in[m]} (\|A[i]-\hat{C}[\psi(i)]\|_2 + \min_{b\in[k]}\|A[i]-C^*[b]\|_2)^p + k^v\nu\Delta^p + (\gamma'\mathrm{cost}_p(A,C^*)+\eta')) \\
&\le O(2^{2p})(\sum_{i\in[m]} \min_{b\in[k]}\|A[i]-C^*[b]\|_2^p + k^v\nu\Delta^p + (\gamma'\mathrm{cost}_p(A,C^*)+\eta')) \\
&\le O(2^{2p})(\mathrm{cost}_p(A,C^*) + k^v\nu\Delta^p + \gamma'\mathrm{cost}_p(A,C^*)+\eta') \\
&= O_p(\gamma')\mathrm{cost}_p(A,C^*) + O_p(\eta'+k^v\nu\Delta^p),
\end{aligned}
$$

where the first inequality follows from triangle inequality, the second inequality follows from convexity and the fact that $\hat{C}$ is a (γ',η')-bicriteria approximation for A (Theorem D.2), the second equality follows from the definition of weights w and the mapping $\psi(\cdot)$, the third inequality follows from that all points are within a ball with diameter Δ, the fourth inequality follows from $\bar{C}$ is an $O(1)$-approximation for $(\hat{C},\hat{w})$ and $\overline{\mathrm{count}}_\rho \in \mathrm{count}_\rho \pm \nu$, the fifth inequality follows from $\sum_{l\in[k]^v}|w[l]-\hat{w}[l]|\cdot\min_{b\in[k]}\|\hat{C}[l]-C^*[b]\|_2^p \le k^v\nu\Delta^p$, the third equality follows from the definition of w and ψ, the sixth inequality follows from triangle inequality, and the seventh inequality follows from convexity and the fact that $\hat{C}$ is a (γ',η')-bicriteria solution for A.

□

By plugging the pattern counting theorem (Theorem B.1), the DP approximate k-means algorithm of [25] and the non-private approximate k-means algorithm of [14] into Theorem D.3, we get the following corollary.

Corollary D.4 (DP k-means in vertical federated learning)**.** *Given partial points $A^{(j)}\in\mathbb{R}^{m\times d^{(j)}}$ for each server $j=1,\cdots,v$, a privacy parameter $\varepsilon\in(0,0.5)$, number of centers $k\ge 1$ and a communication/approximation trade-off parameter $\xi\in(\Omega(1/m),1]$, there is an ε-DP algorithm in the vertical federated learning model which outputs an $(O(1),C_v''(\varepsilon^{-1}kd\log^{O(1)}m+\sqrt{m/\xi}\,(k/\varepsilon)^v\log(k))\Delta^2)$-approximate k-means solution for $A=[A^{(1)}\ A^{(2)}\ \cdots\ A^{(v)}]\in\mathbb{R}^{m\times d}$ with probability at least 0.99, where C_v'' is a factor which only depends on v and Δ is an upper bound of the radius of the ball which contains all data points (rows) of A. The size of total messages sent by each server j is at most $O(k(d^{(j)}+\xi m))$. All local computations only need polynomial running time.*

For k-median, we obtain the following corollary using private k-means algorithm of [8] and our pattern counting algorithm.

Corollary D.5 (DP k-median in vertical federated learning)**.** *Given partial points $A^{(j)}\in\mathbb{R}^{m\times d^{(j)}}$ for each server $j=1,\cdots,v$, a privacy parameter $\varepsilon\in(0,0.5)$, number of centers $k\ge 1$ and a communication-approximation trade-off parameter $\xi\in(\Omega(1/m),1]$, there is an ε-DP algorithm in the vertical federated learning model which outputs an $\left(O(\sqrt{v}),\left(\frac{kd\log^{O(1)}m}{\varepsilon}+\sqrt{\frac{m}{\xi}}\left(\frac{k}{\varepsilon}\right)^{O(v)}\right)\Delta\right)$-approximate k-median solution for $A=[A^{(1)}\ A^{(2)}\ \cdots\ A^{(v)}]\in\mathbb{R}^{m\times d}$ with probability at least 0.99, where Δ is an upper bound of the radius of the ball which contains all data points (rows) of A. The size of total messages sent by each server j is at most $O(k(d^{(j)}+\xi m))$. All local computations only need polynomial running time.*

E Sketching and Perturbation

In this section, we show simple sketch based DP algorithms for low-rank matrix approximation, least squares linear regression, and ℓ_2 moment estimation in the vertical federated learning model. Recall our setting: There are m users and v servers. Each server $j\in[v]$ holds $A^{(j)}\in\mathbb{R}^{m\times d^{(j)}}$. The entire input $A=[A^{(1)}\ A^{(2)}\ \cdots\ A^{(v)}]\in\mathbb{R}^{m\times d}$. For each

server j, the neighbor relation is as follows: $A^{(j)} \sim A'^{(j)}$ if $A^{(j)} - A'^{(j)} = e_i x^\mathsf{T}$ for some $i \in [m]$ and $\|x\|_2 \leq \nu^{(j)}$ i.e., two datasets for server j are neighbors if they differ in at most one row by a euclidean norm at most $\nu^{(j)}$.

E.1 Low Rank Matrix Approximation

We discuss low rank approximation problem in this section. The algorithm of [27] is the most amenable to the vertical federated learning setting. They use the fact that if $\boldsymbol{G} \in \mathbb{R}^{s\times m}$ is a Gaussian matrix with $s = O(k)$, then the rowspace of $\boldsymbol{G} \cdot A$ is a good bicriteria approximation for the low rank matrix approximation problem. To obtain a differentially private mechanism, they output the matrix $\boldsymbol{G} \cdot A + \boldsymbol{N}$ where $\boldsymbol{N}$ is an $s \times m$ Gaussian matrix with entries of appropriate variance and show that the rowspace of $\boldsymbol{G} \cdot A + \boldsymbol{N}$ is also a "good" bicriteria approximation for the low rank approximation problem. We call this technique *sketch-and-perturb* and it has the desired privacy guarantees. Based on this technique, we present the final algorithm in Algorithm 4.

Theorem E.1 (Low rank approximation)**.** *Suppose there are v servers where each server j holds a submatrix $A^{(j)} \in \mathbb{R}^{m\times d^{(j)}}$. Let $A = [A^{(1)}\ A^{(2)}\ \cdots\ A^{(v)}] \in \mathbb{R}^{m\times d}$ where $d = \sum_{j\in[v]} d^{(j)}$. If $\forall j \in [v], \forall A'^{(j)} \sim A^{(j)}, \|A'^{(j)} - A^{(j)}\|_2 \leq \nu_{\max}$, then given $\varepsilon > 0, \delta \in (0,1), \alpha \in (0, 0.5), k \geq 1$, there is an (ε,δ)-DP algorithm in the vertical federated learning model which outputs $V \in \mathbb{R}^{d\times k}$ such that $\|A - AVV^\mathsf{T}\|_\mathsf{F} \leq (1+\alpha)\|A - A_k\|_\mathsf{F} + O\left(\varepsilon^{-1}\nu_{\max}\sqrt{(\alpha^{-2}k + \log m)\log(1/\delta)d}\right)$ holds with probability at least 0.8. In addition, the communication of each server is at most $O(kd/\alpha^2)$.*

Proof. We note that if $A^{(j)}$ and $A'^{(j)}$ are neighboring datasets for server j, then $\|\boldsymbol{G}(A^{(j)} - (A'^{(j)}))\|_\mathsf{F} = \|\boldsymbol{G}_{*i} \cdot (A^{(j)}[i] - A'^{(j)}[i])^\top\|_\mathsf{F}$ for some $i \in [m]$ and $\|A^{(j)}[i] - A'^{(j)}[i]\|_2 \leq \nu^{(j)}$. Thus the ℓ_2 sensitivity of $\boldsymbol{G}A^{(j)}$ is $= \nu^{(j)} \max_i \|\boldsymbol{G}_{*i}\|_2$.

Lemma E.2. *Let $\boldsymbol{G}$ be an $s \times m$ matrix with independent standard Gaussian entries. Then with probability $\geq 99/100$,*

$$\max_{i\in[m]} \|\boldsymbol{G}_{*i}\|_2 \leq \sqrt{s} + 4\sqrt{\log m}.$$

Proof. By Lemma 1 of [36], for any $i \in [m]$, $\mathbf{Pr}[\|\boldsymbol{G}_{*i}\|_2^2 \geq (\sqrt{s} + 4\sqrt{\log m})^2] \leq \exp(-2\log m)$. Hence, by a union bound, $\mathbf{Pr}[\max_{i\in[m]} \|\boldsymbol{G}_{*j}\|_2 \leq \sqrt{s} + 4\sqrt{\log m}] \geq 1 - 1/m$. □

The above lemma implies with high probability that the ℓ_2 sensitivity of $\boldsymbol{G}A^{(j)}$ is at most $\nu^{(j)} \cdot (\sqrt{s} + 4\sqrt{\log m})$. In Algorithm 4, each server first checks if the matrix $\boldsymbol{G}$ has the property that $\max_{i\in[m]} \|\boldsymbol{G}_{*i}\|_2 \leq \sqrt{s} + 4\sqrt{\log m}$. If the property does not hold, they output FAIL. If the property holds, the j-th server samples a Gaussian matrix $\boldsymbol{N}^{(j)}$ with each entry being an independent Gaussian random variable with variance $C(\nu^{(j)})^2(s + \log m)\log(1/\delta)/\varepsilon^2$ for a large enough constant C. By the properties of Gaussian mechanism, outputting $\boldsymbol{G}A^{(j)} + \boldsymbol{N}^{(j)}$ is (ε,δ)-differentially private.

Our starting point to prove the utility guarantee in Theorem E.1 is the following theorem that states the result about how good the Gaussian sketch is for computing a low rank approximation.

Theorem E.3 (Theorem 2.1 of [32])**.** *Let $\boldsymbol{G} \in \mathbb{R}^{s\times m}$ be a Gaussian matrix with entries given by standard normal random variables where entries are $O(k)$-wise independent. Let A be an arbitrary $m \times n$ matrix. If $s \geq C'k/\alpha^2$ for a large enough constant C', then with probability $\geq 9/10$, for all rank k projection matrices P,*

$$s(1-\alpha)\|A(I-P)\|_\mathsf{F}^2 \leq \|\boldsymbol{G}A(I-P)\|_\mathsf{F}^2$$

and

$$\|\boldsymbol{G}A(I-P)\|_\mathsf{F}^2 \leq s(1+\alpha)\|A(I-P)\|_\mathsf{F}^2.$$

From above, we have that each server outputting $\boldsymbol{G}A^{(i)} + \boldsymbol{N}^{(i)}$ is (ε,δ) differentially private with respect to the data on server i. Let $\boldsymbol{N} = \begin{bmatrix}\boldsymbol{N}^{(1)} & \cdots & \boldsymbol{N}^{(v)}\end{bmatrix}$ and $V \in \mathbb{R}^{d\times k}$ be an orthonormal matrix denoting the top k right singular vectors of the matrix $\boldsymbol{G}A + \boldsymbol{N}$. Note that the matrix V can be computed by the Central Coordinator after receiving the noisy sketches from the servers. Let $V_k \in \mathbb{R}^{d\times k}$ denote the top k right singular vectors of the matrix A. We now have

$$\begin{aligned}\|A(I-VV^\mathsf{T})\|_\mathsf{F} &\leq \frac{\|\boldsymbol{G}A(I-VV^\mathsf{T})\|_\mathsf{F}}{\sqrt{s(1-\alpha)}} \\ &\leq \frac{\|(\boldsymbol{G}A + \boldsymbol{N})(I-VV^\mathsf{T})\|_\mathsf{F} + \|\boldsymbol{N}(I-VV^\mathsf{T})\|_\mathsf{F}}{\sqrt{s(1-\alpha)}}\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{\|(\boldsymbol{G}A+\boldsymbol{N})(I-V_kV_k^{\mathsf{T}})\|_{\mathsf{F}}+\|\boldsymbol{N}(I-VV^{\mathsf{T}})\|_{\mathsf{F}}}{\sqrt{s(1-\alpha)}}\\
&\leq \frac{\sqrt{s(1+\alpha)}\|A(I-V_kV_k^{\mathsf{T}})\|_{\mathsf{F}}+\|\boldsymbol{N}(I-V_kV_k^{\mathsf{T}})\|_{\mathsf{F}}}{\sqrt{s(1-\alpha)}}\\
&\quad+\frac{\|\boldsymbol{N}(I-VV^{\mathsf{T}})\|_{\mathsf{F}}}{\sqrt{s(1-\alpha)}}\\
&\leq (1+O(\alpha))\|A-A_k\|_{\mathsf{F}}+\frac{2}{\sqrt{s(1-\alpha)}}\|\boldsymbol{N}\|_{\mathsf{F}}.
\end{aligned}
$$

As $\|\boldsymbol{N}\|_{\mathsf{F}}^2 \leq \sum_{j\in[v]} 2(sd^{(j)})C(\nu^{(j)})^2(s+\log m)\log(1/\delta)/\varepsilon^2$ with a probability at least 0.99 for a large enough constant C,

$$
\begin{aligned}
&\|A(I-VV^{\mathsf{T}})\|_{\mathsf{F}}\\
&\leq (1+O(\alpha))\|A-A_k\|_{\mathsf{F}}\\
&\quad+\frac{2}{\sqrt{1-\alpha}}\sqrt{\nu_{\max}^2 dC(s+\log m)\log(1/\delta)/\varepsilon^2}.
\end{aligned}
$$

Since $s=Ck/\alpha^2$, we obtain the result.

Consider the communication cost, the random seed needs $O(k^2/\alpha^2)$. The size of $O^{(j)}$ is at most $O(kd/\alpha^2)$. □

E.2 Least Squares Linear Regression

Our least squares linear regression algorithm is very similar to our low rank matrix approximation algorithm. Recall the setting for least squares linear regression: Each server $j\in[v]$ holds a matrix $A^{(j)}$, and an additional server holds a label vector $b\in\mathbb{R}^m$. The goal is to solve $\min_x \|[A^{(1)}\ \cdots\ A^{(v)}]x-b\|_2^2$. We say two label vectors b and b' are neighbors $b\sim b'$ if $b-b'=\Delta e_i$ for some $i\in[m]$ and $|\Delta|\leq\nu^{(\mathrm{label})}$. Let OPT denote the optimal value of the problem and $x^*=A^+b$ be the optimal solution. Let $\boldsymbol{G}\in\mathbb{R}^{s\times m}$ be a random Gaussian matrix and be shared with every server. The rows of $\boldsymbol{G}$ are independent, but the entries in the same row of $\boldsymbol{G}$ are $O(d)$-wise independent. The ℓ_2 sensitivity of the product with $\boldsymbol{G}$ for j-th server is bounded by $O(\sqrt{s+\log m})\nu^{(j)}$ with high probability over $\boldsymbol{G}$. By restricting to only those matrices $\boldsymbol{G}$, we obtain that for each server j revealing $\boldsymbol{G}A^{(j)}+\boldsymbol{N}^{(j)}$ is (ε,δ)-DP if each coordinate of $\boldsymbol{N}^{(j)}$ is an independent Gaussian of variance $O((\nu^{(j)})^2(s+\log m)\log(1/\delta)\varepsilon^{-2})$. Correspondingly, the "sketched-and-pertubed" version of the label vector b is given by $\boldsymbol{G}\cdot b+\boldsymbol{n}^{\mathrm{label}}$ where $\boldsymbol{n}^{(\mathrm{label})}$ is an s-dimensional Gaussian vector where each coordinate is an independent Gaussian random variable with mean 0 and variance $O(\nu^{(\mathrm{label})}(s+\log m)\log(1/\delta)\varepsilon^{-2})$. Clearly, the overall historical message sent by each server is (ε,δ)-DP. Hence, the "sketched-and-perturbed" version of the problem we solve is given by $\min_x \|[\boldsymbol{G}A^{(1)}+\boldsymbol{N}^{(1)}\ \cdots\ \boldsymbol{G}A^{(v)}+\boldsymbol{N}^{(v)}]x-(\boldsymbol{G}b+\boldsymbol{n}^{(\mathrm{label})})\|_2^2$. We show that if $s=O(d/\alpha^2)$, then the solution obtained by solving the sketched-and-perturbed problem is a $1+\alpha$ multiplicative approximation along with an additive error term that depends on $\varepsilon,\delta,\|x^*\|_2,\alpha$.

Theorem E.4 (Least squares linear regression)**.** *Suppose there are v servers where each server j holds a submatrix $A^{(j)}\in\mathbb{R}^{m\times d^{(j)}}$ and there is an additional server which holds a label vector $b\in\mathbb{R}^m$. Let $A=[A^{(1)}\ \cdots\ A^{(v)}]\in\mathbb{R}^{m\times d}$ where $d=\sum_{j\in[v]}d^{(j)}$. If $\forall j\in[v],\forall A'^{(j)}\sim A^{(j)}, \|A'^{(j)}-A^{(j)}\|_2\leq\nu_{\max}$ and $\forall b'\sim b, \|b'-b\|_2\leq\nu^{(label)}$, then given $\varepsilon>0,\delta\in(0,1),\alpha\in(0,0.5)$, there is an (ε,δ)-DP algorithm in the vertical federated learning model which outputs $\tilde{x}_G\in\mathbb{R}^d$ such that $\|A\tilde{x}_G-b\|_2^2\leq(1+\alpha)\mathrm{OPT}+O(d/\alpha^2+\log m)\log(1/\delta)\varepsilon^{-2}(\nu_{\max}^2\|x^*\|_2^2+(\nu^{(\mathrm{label})})^2)$ holds with probability at least 0.9. In addition, the communication cost of each server is at most $O(d^2/\alpha^2)$.*

Proof. The sketched-and-perturbed problem is equivalent to solving

$$
\min_x \|[\boldsymbol{G}\,\boldsymbol{G}']\begin{bmatrix}A\\ \Delta\end{bmatrix}-[\boldsymbol{G}\,\boldsymbol{G}']\begin{bmatrix}b\\ \Delta'\end{bmatrix}\|_2^2
$$

where $\boldsymbol{G}$ and $\boldsymbol{G}'$ are $s\times m$ and $s\times(d+1)$ dimensional Gaussian matrices respectively, Δ is an $(d+1)\times d$ matrix defined as

$$
\Delta=\begin{bmatrix}\Delta_1 & & \\ & \ddots & \\ & & \Delta_v\\ 0 & \cdots & 0\end{bmatrix}
$$

with Δ_j defined as

$$O(\sqrt{(\nu^{(j)})^2(s+\log m)\log(1/\delta)\varepsilon^{-2}})\cdot I_{d^{(j)}}$$

and Δ' is a $(d+1)$-dimensional vector with the first d coordinates 0 and the last coordinate being equal to $O(\sqrt{(\nu^{(\text{label})})^2(s+\log m)\log(1/\delta)\varepsilon^{-2}})$. If $s=\Omega(d/\alpha^2)$, then $\frac{1}{\sqrt{s}}[\boldsymbol{G}\,\boldsymbol{G}']$ is a $1\pm\alpha$ subspace embedding [63] for a fixed $d+1$ dimensional subspace with a probability $\geq 99/100$. Thus if $\tilde{x}_G$ is the solution to the sketched problem, then

$$\|A\tilde{x}_G-b\|_2^2\leq(1+\alpha)(\|Ax^*-b\|_2^2+\|\Delta x^*-\Delta\|_2^2).$$

As $\|\Delta\|_2^2\leq O(\nu_{\max}^2(s+\log m)\log(1/\delta)\varepsilon^{-2})$ and $\|\Delta'\|_2^2=O((\nu^{(\text{label})})^2(s+\log m)\log(1/\delta)\varepsilon^{-2})$, we obtain

$$\begin{aligned}&\|A\tilde{x}_G-b\|_2^2\\&\leq(1+\alpha)\mathrm{OPT}+\frac{C(d/\alpha^2+\log m)\log(1/\delta)\nu_{\max}^2}{\varepsilon^2}\|A^+b\|_2^2\\&\quad+\frac{C(d/\alpha^2+\log m)\log(1/\delta)(\nu^{(\text{label})})^2}{\varepsilon^2}.\end{aligned}$$

Here $\nu_{\max}=\max_{j\in[v]}\nu_j$.

Consider the communication cost, the random seed needs $O(d^2/\alpha^2)$. The size of $GA^{(j)}+N^{(j)}$ is $O(d^2/\alpha^2)$ as well. □

E.3 ℓ_2 Moment Estimation

The algorithm for ℓ_2 moment estimation is similar as above algorithms.

Theorem E.5 (ℓ_2 Moment estimation)**.** *Suppose there are v servers where each server j holds a vector $x^{(j)}\in\mathbb{R}^m$. Let $x=\sum_{j\in[v]}x^{(j)}$. If $\forall j\in[v],\forall x'^{(j)}\sim x^{(j)},\|x'^{(j)}-x^{(j)}\|_\infty\leq\nu^{(j)}$, there is an (ε,δ)-DP algorithm in the vertical federated learning model which outputs an estimation of $\|x\|_2^2$ in $(1\pm\alpha)\|x\|_2^2\pm\log(m)\log(1/\delta)\varepsilon^{-2}\alpha^{-1}v^2\nu_{\max}^2$ holds with probability at least 0.9. In addition the communication cost of each server is at most $O(1/\alpha^2)$.*

The algorithm is as follows. Since each server j holds a matrix $A^{(j)}$ with only $d^{(j)}=1$ column. We use $x^{(j)}$ to denote the vector $A^{(j)}$. Note that for any neighboring dataset $x'^{(j)}\sim x^{(j)}$, it satisfies $\|x'^{(j)}-x^{(j)}\|_\infty\leq\nu^{(j)}$. The goal of ℓ_2 moment estimation is to estimate $\|\sum_{j\in[v]}x^{(j)}\|_2^2$. Similar as before, let $\boldsymbol{G}\in\mathbb{R}^{s\times m}$ be a random Gaussian matrix and be shared with every server. The rows of $\boldsymbol{G}$ are independent, but the entries in the same row of $\boldsymbol{G}$ are $O(1)$-wise independent. The ℓ_2 sensitivity of the product with $\boldsymbol{G}$ for j-th server is bounded by $O(\sqrt{s+\log m})\nu^{(j)}$ with high probability over $\boldsymbol{G}$. By restricting to only those matrices $\boldsymbol{G}$, we obtain that for each server j revealing $\boldsymbol{G}x^{(j)}+\boldsymbol{n}^{(j)}$ is (ε,δ)-DP if each coordinate of $\boldsymbol{n}^{(j)}$ is an independent Gaussian of variance $O((\nu^{(j)})^2(s+\log m)\log(1/\delta)\varepsilon^{-2})$. Then the Central Coordinator outputs $\|\frac{1}{\sqrt{s}}\sum_{j\in[v]}(\boldsymbol{G}x^{(j)}+\boldsymbol{n}^{(j)})\|_2^2$ as an estimation of the ℓ_2 moment. It turns out that it is enough to set $s=\Theta(1/\alpha^2)$ where $(1\pm\alpha)$ is the multiplicative approximation factor.

Proof of Theorem E.5. Due to our choice of $\boldsymbol{G}$, then ℓ_2 sensitivity of $\boldsymbol{G}x^{(j)}$ is at most $O(\sqrt{s+\log(m)}\cdot\nu^{(j)})$. Thus, the algorithm is (ε,δ)-DP when $\boldsymbol{n}^{(j)}$ are Gaussian random variables with variance $C\nu_{\max}^2(s+\log m)\log(1/\delta)/\varepsilon^2$ for some sufficiently large constant C. Therefore, we have

$$\begin{aligned}\left\|\frac{1}{\sqrt{s}}(\boldsymbol{G}x+\sum_{j\in[v]}\boldsymbol{n}^{(j)})\right\|_2&\leq\frac{1}{\sqrt{s}}\|\boldsymbol{G}x\|_2+\frac{1}{\sqrt{s}}\sum_{j\in[v]}\left\|\boldsymbol{n}^{(j)}\right\|_2\\&\leq\frac{1}{\sqrt{s}}\|\boldsymbol{G}x\|_2+O(v\cdot\nu_{\max}\sqrt{1+\log(m)/s}\log(1/\delta)/\varepsilon^2)\end{aligned}$$

with probability at least 0.99. Due to Johnson–Lindenstrauss lemma with probability at least 0.99, $\frac{1}{\sqrt{s}}\|\boldsymbol{G}x\|_2\leq(1+\alpha)\|x\|_2$. Thus, with probability at least 0.98

$$\left\|\frac{1}{\sqrt{s}}(\boldsymbol{G}x+\sum_{j\in[v]}\boldsymbol{n}^{(j)})\right\|_2^2\leq(1+O(\alpha))\|x\|_2^2+O(v^2\nu_{\max}^2\log(m)\log(1/\delta)/(\varepsilon^2\alpha)).$$

Similarly, with probability at least 0.98,

$$\left\| \frac{1}{\sqrt{s}}(\boldsymbol{G}x + \sum_{j\in[v]} \boldsymbol{n}^{(j)}) \right\|_2^2 \geq (1 - O(\alpha))\|x\|_2^2 - O(v^2\nu_{\max}^2 \log(m)\log(1/\delta)/(\varepsilon^2\alpha)).$$

Finally, consider the communication. The random seeds require $O(s) = O(1/\alpha^2)$ communication. $\boldsymbol{G}x^{(j)} + \boldsymbol{n}^{(j)}$ has size $O(s) = O(1/\alpha^2)$. □

F Lower Bounds

As mentioned in the introduction, McGregor et al. [39] study the lower bounds on the additive error of differentially private algorithms in the setting with two servers. They show the following:

Theorem F.1 (§ 4.2 of [39]). *Let $x^{(1)} \in \{0,1\}^m$ be the vector at server 1 and $x^{(2)} \in \{0,1\}^m$ be the vector at server 2. Any ε DP VFL algorithm for computing $\langle x^{(1)}, x^{(2)}\rangle$ or the Hamming distance $\|x^{(1)} - x^{(2)}\|_1$ must have an additive error of $\Omega(\sqrt{m})$ with a large probability.*

Note that the Hamming distance $\|x^{(1)} - x^{(2)}\|_1$ is exactly equal to $\text{count}_{(1,0)} + \text{count}_{(0,1)}$ and hence any pattern counting algorithm can compute the Hamming distance and therefore we obtain that a pattern counting algorithm in the DP VFL setting must also have an additive error of $\Omega(\sqrt{m})$. We can similarly obtain lower bounds on the additive error for ℓ_p norm estimation in the case of two servers by reducing the Hamming distance problem to ℓ_p norm estimation as follows: define $A \coloneqq \text{count}_{(0,0)}, B \coloneqq \text{count}_{(0,1)} + \text{count}_{(1,0)}$ and $C \coloneqq \text{count}_{(1,1)}$. Define $z^{(1)} = 1 - x^{(1)}$ and $z^{(2)} = 1 - x^{(2)}$. Note that $z^{(1)}, z^{(2)}$ are also binary vectors. Now, $W \coloneqq \|x^{(1)} + x^{(2)}\|_p^p = B + C \cdot 2^p$ and $W' \coloneqq \|z^{(1)} + z^{(2)}\|_p^p = A \cdot 2^p + B$. We additionally have $A + B + C = m$. Solving for B we get, $B = (m \cdot 2^p - (W + W'))/(2^p - 2)$. Let $\widetilde{W}$ (resp. $\widetilde{W'}$) be an approximations of W (resp. W') using an $\varepsilon/2$ differential private algorithm. Then, $\widetilde{B} \coloneqq \frac{m\cdot 2^p - (\widetilde{W} + \widetilde{W'})}{2^p - 2}$ is an approximation for B. Now, if the additive error in $\widetilde{W}$ and $\widetilde{W'}$ is at most $c(2^p - 2)\sqrt{m}$, then the additive error in $\tilde{B}$ is at most $2c\sqrt{m}$ which implies that any DP VFL algorithm that approximates the ℓ_p^p value of the sum of the vectors must have an additive error of $\Omega((2^p - 2)\sqrt{m})$. Note that for $p = 1$, we do not get an $\Omega(\sqrt{m})$ lower bound as $\|x^{(1)} + x^{(2)}\|_1 = \|x^{(1)}\|_1 + \|x^{(2)}\|_1$ when both $x^{(1)}$ and $x^{(2)}$ are nonnegative. Now, server j can use the Laplace mechanism to send an approximation of $\|x^{(j)}\|_1$ to the central coordinator and therefore there is an ε DP VFL algorithm with an additive error of only $O(1/\varepsilon)$.

We note that the lower bound is only on the protocols that only have an additive error but our upper bounds for ℓ_p norm estimation additionally have a multiplicative error as well. Our result for ℓ_2 estimation using the sketch-and-perturb technique shows that algorithms that have a multiplicative error need not incur the additive error of $\Omega(\sqrt{m})$. Hence, it is still possible that for other values of p, our algorithms have a sub-optimal dependence on m and that the $\sqrt{m}$ additive error may be removable since we allow for multiplicative error as well.

G Other Related Work

The concept of federated learning is proposed by [34, 35, 40] which has received extensive attention from the machine learning community in the past fewer years. We refer readers to [66] for a survey for the concepts and applications of federated learning.

For the vertical federated learning model, there is a long line of work developing non-DP machine learning algorithms includes e.g., [29, 62, 37, 66, 65, 28, 54, 10, 9, 15].

In the non-federated learning model, there is a long line of work for DP algorithms for moment estimation [20, 41, 17, 3, 49, 11, 50, 7, 59, 5], clustering [6, 45, 23, 26, 42, 60, 46, 47, 53, 24, 2, 44, 30, 52, 25, 31, 43, 4, 12, 13], and numerical linear algebra problems [6, 27, 33, 19, 38, 56, 1, 55].

The local DP model can be regarded as a DP model for horizontal federated learning. Learning algorithms have also been heavily studied in the local DP model, including e.g., clustering [44, 52, 8, 51] and numerical linear algebra algorithms [55, 1].

Consider DP algorithms in vertical federated learning. [58, 48, 67] studied how to make the optimization process differentially private in the vertical federated learning model. [64] studied DP algorithms for linear regression in the vertical federated learning. It studied the distance between the output coefficient vector and the ground truth and had assumption on the input distribution while our algorithm studies the worst case approximation guarantee.

H Local Differential Privacy and Differential Privacy in Vertical Federated Learning

Definition H.1 (Local DP)**.** Given a dataset A containing m data points $A[1], A[2], \cdots, A[m]$, if a mechanism $\mathcal{M}$ has the form $\mathcal{M}(A) = \mathrm{Dec}(\mathrm{Enc}(A[1]), \mathrm{Enc}(A[2]), \cdots, \mathrm{Enc}(A[m]))$ where Enc is an encoding algorithm which takes only one data point and Dec is a decoding algorithm which outputs a solution based on the coding for all input points, then $\mathcal{M}$ is (ε, δ)-local DP (resp. ε-local DP) if $(\mathrm{Enc}(A[1]), \mathrm{Enc}(A[2]), \cdots, \mathrm{Enc}(A[m]))$ is (ε, δ)-DP (resp. ε-DP).

It is obvious that local DP is a stronger privacy notation than standard DP. The model of local DP can be seen as the DP model for horizontal federated learning, i.e., each device holds the entire data point of a user and the overall historical messages sent by a device must be DP with respect to its user data.

Note that since the encoder algorithm Enc in the local DP algorithm can use the entire information of a single data point at a time, local DP algorithm usually does not imply a DP algorithm in the vertical federated learning. For example, the local DP k-clustering algorithms of [8, 51] heavily relies on the location information of each data point. Thus, those algorithms cannot be implemented in the vertical federated learning.

Theorem H.2 ([18])**.** *The randomized response mechanism is ε-local DP.*

Due to above theorem, the messages sent by each server in the pattern counting algorithm that we described in Algorithm 1 is not only DP but also local DP with respect to the data held by the server. Thus, our pattern counting algorithm and its applications work for a even more restrictive DP model which is a DP model of the mixture of horizontal federated learning and vertical federated learning: Given a dataset A containing m data points $A[1], A[2], \cdots, A[m]$ where each data point $A[i]$ contains d attributes $A[i,1], A[i,2], \cdots, A[i,d]$, we want a mechanism $\mathcal{M}$ with the form

$$\mathcal{M}(A) = \mathrm{Dec}(\mathrm{Enc}(A[1,1]), \cdots, \mathrm{Enc}(A[1,d]), \cdots, \mathrm{Enc}(A[m,1]), \cdots, \mathrm{Enc}(A[m,d])))$$

where Enc is an encoding algorithm taking one attribute of one user at a time, and Dec takes all encoded messages to output a solution, and the overall message

$$(\mathrm{Enc}(A[1,1]), \cdots, \mathrm{Enc}(A[1,d]), \cdots, \mathrm{Enc}(A[m,1]), \cdots, \mathrm{Enc}(A[m,d])))$$

is DP. We give an example of potential scenarios for the above hybrid partitioning model for federated learning. Suppose there are two data repositories where the first holds some diet information of all users and the second holds some health information of all users. But the data of the first repository is collected by many untrusted food delivery apps and the data of the second repository is collected by many untrusted healthcare apps. Each app only collects a subset of attributes of a subset of users and the message sent by a user to an app must be differentially private to protect user privacy. In this case, an algorithm such as our pattern counting algorithm which is both local DP and DP in vertical federated learning would provide a way to analyze the correlation between diet habits and health conditions.

It is easy to verify that both our ℓ_p moment estimation algorithm and k-clustering algorithm also provide DP algorithms in the hybrid partitioning model for federated learning.

Algorithm 3: Approximate ℓ_p k-Clustering

Input: Partial points $A^{(j)} \in \mathbb{R}^{m \times d^{(j)}}$ for each server $j \in [v]$ where $\sum_{j \in [v]} d^{(j)} = d$, privacy parameters $\varepsilon > 0, \delta \in [0, 1)$, power $p \geq 1$ and clustering parameter $k > 0$.

Output: $\bar{C} \in \mathbb{R}^{k \times d}$

for *each server* $j = 1, \cdots, v$ *concurrently* **do**

  Compute an $(\varepsilon/2, \delta)$-DP $(\gamma^{(j)}, \eta^{(j)})$-approximation $\hat{C}^{(j)}$ for ℓ_p k-clustering of $A^{(j)}$ `// For example, run algorithms of [25] or [13]`

  Create a vector $x^{(j)} \in [k]^m$ where $\forall i \in [m], x^{(j)}[i] := \arg\min_{b \in [k]} \|A^{(j)}[i] - \hat{C}^{(j)}[b]\|_2$

For Central Coordinator and all servers, run pattern counting (Algorithm 1 or the variant mentioned in Theorem B.2) for $x^{(1)}, x^{(2)}, \cdots, x^{(v)}$ and $M = k$, with privacy parameter $\varepsilon/2$

At the Central Coordinator:

Collect the partial centers $\hat{C}^{(1)}, \hat{C}^{(2)}, \cdots, \hat{C}^{(v)}$ sent by the servers

Get $\overline{\text{count}}_\rho$ for every $\rho \in [k]^v$ such that $\overline{\text{count}}_\rho \in \text{count}_\rho \pm \nu$ (the output of pattern counting)

Compute $\hat{C} = \left\{ \left[\hat{C}^{(1)}[i_1] \ \hat{C}^{(2)}[i_2] \ \cdots \ \hat{C}^{(v)}[i_v] \right] \mid (i_1, i_2, \cdots, i_v) \in [k]^v \right\} \subset \mathbb{R}^d$

Due to the construction of $\hat{C}$, each point in $\hat{C}$ can be indexed by $\rho = (i_1, i_2, \cdots, i_v) \in [k]^v$, and we assign a weight $\hat{w}[\rho] = \max(0, \overline{\text{count}}_\rho)$

Compute an $O(1)$-approximation $\bar{C}$ for ℓ_p k-clustering of $(\hat{C}, \hat{w})$ `// For example, use [14]`

return $\bar{C}$

Algorithm 4: Distributed Differentially Private Low Rank Matrix Approximation

Input: Each server j has an $m \times d^{(j)}$ matrix $A^{(j)}$, approximation parameter α, and rank parameter $k \geq 1$, $\forall j \in [v], \forall A'^{(j)} \sim A^{(j)}$, it satisfies $\|A'^{(j)} - A^{(j)}\|_2 \leq \nu^{(j)}$

Output: A good subspace for low rank matrix approximation

$\boldsymbol{G} \in \mathbb{R}^{s \times m} \leftarrow$ A Gaussian Random matrix with $s := \Theta(k/\alpha^2)$ rows and m columns where the m entries in each row are $O(k)$-wise independent, and different rows are fully independent.;

`// If the` ℓ_2 `norms of any column of` $\boldsymbol{G}$ `is larger than` $\sqrt{s} + 4\sqrt{\log m}$`, output FAIL`

for *Each server* $j = 1, \cdots, v$ *in parallel* **do**

  $O^{(j)} \leftarrow \boldsymbol{G}A^{(j)} + \boldsymbol{N}^{(j)}$ where each entry of $\boldsymbol{N}^{(i)}$ is an independent Gaussian random variable with a variance $\Theta((\nu^{(j)})^2(\alpha^{-2}k + \log m)\log(1/\delta)/\varepsilon^2)$;

  Send $O^{(j)}$ to the Central Coordinator;

At the Central Coordinator:;

$O \leftarrow [O^{(1)} \cdots O^{(v)}]$;

$V \leftarrow$ top k right singular vectors of O;

return V;