DIFFERENTIALLY PRIVATE DISTRIBUTED PRINCIPAL COMPONENT ANALYSIS

Hafiz Imtiaz and Anand D. Sarwate

Department of Electrical and Computer Engineering, Rutgers University

ABSTRACT

Differential privacy is a cryptographically-motivated formal privacy definition that is robust against strong adversaries. The principal component analysis (PCA) algorithm is frequently used in signal processing, machine learning, and statistics pipelines. In many scenarios, private or sensitive data is distributed across different sites: in this paper we propose a differentially private distributed PCA scheme to enable collaborative dimensionality reduction. We investigate the performance of the proposed algorithm on synthetic and real datasets and show empirically that our algorithm can reach the same level of utility as the non-private PCA for some parameter choices, which indicates that it is possible to have meaningful utility while preserving privacy.

***Index Terms*—** principal component analysis, differential privacy, distributed algorithm

1. INTRODUCTION

Many machine learning and data analytics procedures involve private or sensitive data. The outputs of these algorithms can leak information which may allow potentially harmful inferences about individuals. To protect against such privacy violations, a privacy measure – differential privacy (DP), has become a popular approach during the last decade. Differential privacy [1] measures privacy risk in terms of the probability of identifying individual data points in a dataset from the results of computations (algorithms) performed on that data.

In many modern applications the data is distributed over different sites. Naturally, the number of samples held locally is small. However, most statistical estimation procedures tend to be more accurate when the number of samples is large. Therefore, we would like to exploit the data samples available across all locations/sites to estimate the desired population parameter. Note that sending the samples to a central aggregator would certainly raise privacy and communication cost concerns. To that end, we propose an algorithm to perform principal component analysis (PCA) on distributed data while satisfying differential privacy. PCA, or singular value decomposition (SVD), is one of the most widely-used stages in machine learning algorithms. SVD or PCA is used for preprocessing high-dimensional data by projecting it onto a lower dimensional subspace spanned by the singular vectors of the second-moment matrix of the data. For example, training a classifier is much faster when the data is first projected onto lower dimensions.

Related works. Several distributed PCA algorithms have been proposed [2–6] in the literature. Liang et al. [2] proposed a distributed PCA scheme where it is necessary to send both the left and right singular vectors along with corresponding singular values from each site to the aggregator. Feldman et al. [7] proposed an improvement upon this, where the sites send a $D \times R$ matrix to the aggregator. Balcan et al. [3] proposed a further improved version using fast sparse subspace embedding [8] and randomized SVD [9]. However, none of these algorithms guarantee any privacy. A privacy-preserving distributed diffusion LMS algorithm was proposed in [10]. Additionally, DP stochastic gradient descent algorithms were proposed in several works [11, 12]. To our knowledge, the only DP distributed PCA algorithm is proposed by Imtiaz et al. [13] as an intermediate stage of a DP distributed joint independent component analysis (djICA) algorithm. However, the scheme proposed to send data from one site to another in a sequential manner, which is less fault-tolerant.

In this paper, we propose a differentially private distributed PCA algorithm based on the non-private scheme of Feldman et al. [7]. We assume that there are several sites with disjoint data. There is a central node or aggregator but the aggregator is not trusted. Each site sends a "proxy" data matrix to the aggregator, which is computed satisfying differential privacy. The aggregator then computes and releases the consensus PCA subspace. We investigate the variation of performance of the proposed algorithm by varying several relevant algorithm and database parameters on synthetic and real datasets. We empirically show that meaningful utility can be achieved even while satisfying privacy.

Notation. We denote vectors, matrices and scalars with lower-case bold-faced letters (e.g. $\mathbf{x}$), upper-case bold-faced letters (e.g. $\mathbf{X}$), and unbolded letters (e.g. N), respectively. Indices are represented with lower-case regular letters and they typically run from 1 to their upper-case version (e.g. $n \in [N] \triangleq \{1, 2, \ldots, N\}$). The n-th column of a matrix $\mathbf{X}$ is denoted as $\mathbf{x}_n$. $\|\cdot\|_2$ denotes the Euclidean norm of a vector and spectral norm of a matrix. $\|\cdot\|_F$ and $\mathrm{tr}(\cdot)$ denote the Frobenius norm and the trace.

2. PROBLEM FORMULATION

Consider a system with S different sites each holding disjoint datasets and an untrusted central node or aggregator. The $D \times N_s$ data matrix in site $s \in [S]$ is denoted by $\mathbf{X}_s = [\mathbf{x}_{s,1} \ldots \mathbf{x}_{s,N_s}]$, which we interpret as containing D-dimensional features of N_s individuals. We assume that $\|\mathbf{x}_{s,n}\|_2 \leq 1 \; \forall s \in [S]$ and $\forall n \in [N_s]$. For simplicity, we further assume that the observed samples are mean-centered. The $D \times D$ sample second-moment matrix at site s is given by $\mathbf{A}_s = \frac{1}{N_s}\mathbf{X}_s\mathbf{X}_s^\top$. We denote $N = \sum_{s=1}^{S} N_s$ as the total number of samples over all sites. If we had all the samples at the central aggregator (pooled data scenario), then

The work of the authors was supported by the NSF under award CCF-1453432, by the NIH under award 1R01DA040487-01A1, and by DARPA and SSC Pacific under contract No. N66001-15-C-4070.

the data matrix would be $\mathbf{X} = [\mathbf{X}_1 \dots \mathbf{X}_S] \in \mathbb{R}^{D \times N}$. In this case, the $D \times D$ positive semi-definite second-moment matrix $\mathbf{A}$ is given by: $\mathbf{A} = \frac{1}{N}\mathbf{X}\mathbf{X}^\top$. The Schmidt approximation theorem [14] characterizes the rank-K matrix $\mathbf{A}_K$ that minimizes the difference $\|\mathbf{A} - \mathbf{A}_K\|_F$ and shows that the minimizer can be found by taking the SVD of $\mathbf{A}$: $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, where without loss of generality we assume $\mathbf{\Lambda}$ is a diagonal matrix with diagonal entries $\{\lambda_d(\mathbf{A})\}$ and $\lambda_1(\mathbf{A}) \geq \dots \geq \lambda_D(\mathbf{A}) \geq 0$. Additionally, $\mathbf{V}$ is a matrix of eigenvectors corresponding to the eigenvalues. The top-K PCA subspace of $\mathbf{A}$ is the matrix $\mathbf{V}_K(\mathbf{A}) = [\mathbf{v}_1 \dots \mathbf{v}_K]$. Given $\mathbf{V}_K(\mathbf{A})$ and the eigenvalue matrix $\mathbf{\Lambda}$, we can form an approximation $\mathbf{A}_K = \mathbf{V}_K(\mathbf{A})\mathbf{\Lambda}_K\mathbf{V}_K(\mathbf{A})^\top$ to $\mathbf{A}$, where $\mathbf{\Lambda}_K$ contains the K largest eigenvalues in $\mathbf{\Lambda}$. For a $D \times K$ matrix $\hat{\mathbf{V}}$ with orthonormal columns, the quality of $\hat{\mathbf{V}}$ in approximating $\mathbf{V}_K(\mathbf{A})$ can be measured by the *captured energy* of $\mathbf{A}$ as

$$q(\hat{\mathbf{V}}) = \mathrm{tr}(\hat{\mathbf{V}}^\top\mathbf{A}\hat{\mathbf{V}}). \tag{1}$$

The $\hat{\mathbf{V}}$, which maximizes $q(\hat{\mathbf{V}})$ has columns equal to $\{\mathbf{v}_k\}$ corresponding to the top-K eigenvectors of $\mathbf{A}$, namely $\mathbf{V}_K(\mathbf{A})$. We are interested in approximating $\mathbf{V}_K(\mathbf{A})$ in a distributed setting while guaranteeing differential privacy. One naïve approach would be to send the data matrices from the sites to the aggregator. When D and/or N_s are large, this entails a huge communication overhead. In many scenarios the local data are also private or sensitive. As the aggregator is not trusted, sending the data to the aggregator can result in a significant privacy violation. Our goals are therefore (i) to reduce the communication cost, (ii) ensure differential privacy, and (iii) provide a close approximation to the true PCA subspace $\mathbf{V}_K(\mathbf{A})$.

Differential privacy. An algorithm $\mathscr{A}(\mathbb{D})$ taking values in a set $\mathbb{T}$ provides (ϵ, δ)-differential privacy [1] if

$$\Pr(\mathscr{A}(\mathbb{D}) \in \mathbb{S}) \leq \exp(\epsilon)\Pr(\mathscr{A}(\mathbb{D}') \in \mathbb{S}) + \delta, \tag{2}$$

for all measurable $\mathbb{S} \subseteq \mathbb{T}$ and all data sets $\mathbb{D}$ and $\mathbb{D}'$ differing in a single entry (neighboring datasets). This definition essentially states that the probability of the output of an algorithm is not changed significantly if the corresponding database input is changed by just one entry. Here ϵ and δ are privacy risk parameters, where low ϵ and δ ensure more privacy. The parameter δ can be interpreted as the probability that the algorithm fails. For more details, see the recent survey [15] or monograph [16]. In our setting, two data matrices correspond to two neighboring datasets if they differ in one column. For example, $\mathbf{X}_s = [\mathbf{x}_{s,1} \dots \ \mathbf{x}_{s,N_s-1}\ \mathbf{x}_{s,N_s}]$ and $\mathbf{X'}_s = [\mathbf{x}_{s,1} \dots \ \mathbf{x}_{s,N_s-1}\ \mathbf{x'}_{s,N_s}]$ are matrices corresponding to two neighboring data sets. We observe that $\mathbf{A}_s = \frac{1}{N_s}\mathbf{X}_s\mathbf{X}_s^\top$ and $\mathbf{A'}_s = \frac{1}{N_s}\mathbf{X'}_s\mathbf{X'}_s^\top$ satisfy the condition $\|\mathbf{A}_s - \mathbf{A'}_s\|_2 \leq \frac{1}{N_s}$ [17].

3. ALGORITHM

In this section, we describe our proposed algorithm, shown in Algorithm 1, in detail. The proposed algorithm is based on the distributed PCA scheme presented in [3, 7]. As mentioned before, our data samples are distributed in S sites. Each site $s \in [S]$ contains a data matrix $\mathbf{X}_s \in \mathbb{R}^{D \times N_s}$. There is an aggregator, which is not trusted. In the pooled data scenario, we have the data matrix $\mathbf{X}$ and the sample second-moment matrix $\mathbf{A} = \frac{1}{N}\mathbf{X}\mathbf{X}^\top$. We refer to the top-$K$ PCA subspace of this sample second-moment matrix as the true (or optimal) subspace $\mathbf{V}_K(\mathbf{A})$. Our goal is to find this optimal rank-K subspace in a distributed setting, while ensuring differential privacy. At each site, we compute the sample second-moment matrix as $\mathbf{A}_s = \frac{1}{N_s}\mathbf{X}_s\mathbf{X}_s^\top$. The $\mathscr{L}_2$ sensitivity [1] of the function $f(\mathbf{A}_s) = \mathbf{A}_s$ is $\frac{1}{N_s}$ [17]. We intend to approximate $\mathbf{A}_s$ such that the approximation is (ϵ, δ) differentially private. To this end, we employ the AG algorithm [17] to compute $\hat{\mathbf{A}}_s = \mathbf{A}_s + \mathbf{E}$, where $\hat{\mathbf{A}}_s$ is the (ϵ, δ) differentially private estimate of $\mathbf{A}_s$. The noise matrix $\mathbf{E}$ is generated as described in Step 3 of Algorithm 1. Next, we compute the SVD of $\hat{\mathbf{A}}_s$ as $\hat{\mathbf{A}}_s = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$. We use $\mathbf{U}_R \in \mathbb{R}^{D \times R}$ to denote the matrix containing the top-R columns of $\mathbf{U}$. Additionally, we use $\mathbf{\Sigma}_R \in \mathbb{R}^{R \times R}$ to denote a diagonal matrix containing the corresponding top-R diagonal entries of the matrix $\mathbf{\Sigma}$. Each site then computes $\mathbf{P}_s = \mathbf{U}_R\mathbf{\Sigma}_R^{\frac{1}{2}}$ and sends it to the aggregator. At the aggregator, we compute the "proxy" sample second-moment matrix $\mathbf{A}_c$ as $\mathbf{A}_c = \frac{1}{S}\sum_{s=1}^{S}\mathbf{P}_s\mathbf{P}_s^\top$. We perform SVD on $\mathbf{A}_c$ and release the top-K eigenvector matrix $\mathbf{V}_K$, which is the (ϵ, δ) differentially private approximate to the true subspace $\mathbf{V}_K(\mathbf{A})$. Note that, we could compute the differentially-private approximation to $\mathbf{P}_s$ instead of $\mathbf{A}_s$. However, the function $f(\mathbf{A}_s) = \mathbf{P}_s$ has much higher $\mathscr{L}_2$ sensitivity than $f(\mathbf{A}_s) = \mathbf{A}_s$. Therefore, we chose to approximate $\mathbf{A}_s$ in a differentially-private way. Additionally, we opted for adding Gaussian noise instead of Laplace noise as the Laplace mechanism [16] entails $\mathscr{L}_1$ sensitivity, which dictates the additive noise variance to vary with the dimension of the matrix.

Algorithm 1 Differentially private distributed PCA (DPdisPCA)

Require: Data matrix $\mathbf{X}_s \in \mathbb{R}^{D \times N_s}$ for $s \in [S]$; privacy parameters ϵ, δ; intermediate dimension R; reduced dimension K

1: **for** $s = 1, 2, \dots, S$ **do** ▷ at the local sites
2: Compute $\mathbf{A}_s \leftarrow \frac{1}{N_s}\mathbf{X}_s\mathbf{X}_s^\top$
3: Generate $D \times D$ symmetric matrix $\mathbf{E}$ where $\{\mathbf{E}_{ij} : i \in [D], j \leq i\}$ drawn i.i.d. from $\mathscr{N}(0, \Delta^2_{\epsilon,\delta})$, where $\Delta_{\epsilon,\delta} = \frac{1}{N_s\epsilon}\sqrt{2\log(\frac{1.25}{\delta})}$, and $\mathbf{E}_{ij} = \mathbf{E}_{ji}$
4: Compute $\hat{\mathbf{A}}_s \leftarrow \mathbf{A}_s + \mathbf{E}$
5: Perform SVD $\hat{\mathbf{A}}_s = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$
6: Compute $\mathbf{P}_s \leftarrow \mathbf{U}_R\mathbf{\Sigma}_R^{\frac{1}{2}}$
7: Send $\mathbf{P}_s$ to aggregator
8: **end for**
9: Compute $\mathbf{A}_c \leftarrow \frac{1}{S}\sum_{s=1}^{S}\mathbf{P}_s\mathbf{P}_s^\top$ ▷ at the aggregator
10: Perform SVD $\mathbf{A}_c = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$
11: Release / send to sites: $\mathbf{V}_K$
12: **return** $\mathbf{V}_K$

Theorem 1 (Privacy of DPdisPCA Algorithm)**.** *Algorithm 1 computes an (ϵ, δ) differentially private approximation to the optimal subspace $\mathbf{V}_K(\mathbf{A})$.*

Proof sketch. The proof of Theorem 1 directly follows from using the Gaussian mechanism [1], the AG algorithm [17], the bound on $\|\mathbf{A}_s - \mathbf{A'}_s\|_2$ and recalling that the data samples in each site are disjoint. The computation of $\mathbf{P}_s$ at each site is (ϵ, δ) differentially private. As differential privacy is invariant under post-processing, we can combine the "proxy" data ma-

Table 1. Notation of performance measures

Algorithm / Setting	Performance Index
Pooled Data	q_{pooled}
DPdisPCA	q_{DPdisPCA}
Local Data	q_{local}
Sending $\hat{\mathbf{A}}_s$	q_{full}

trices $\mathbf{P}_s$ at the aggregator, perform the SVD and release $\mathbf{V}_K$. The released subspace $\mathbf{V}_K$ is thus the (ϵ, δ) differentially private approximate to the true subspace $\mathbf{V}_K(\mathbf{A})$. □

Communication cost. The algorithm is an one-shot approach to compute the PCA in a distributed setting. That is, the sites send $\mathbf{P}_s$ only once to the aggregator. The aggregator combines $\mathbf{P}_s$ to compute $\mathbf{A}_c$ and releases $\mathbf{V}_K$. Therefore, the communication cost is proportional to $S \times D \times R$. We note that if we send the second-moment matrix $\hat{\mathbf{A}}_s$ to the aggregator, the cost would be much higher (proportional to $S \times D^2$) because typically, $R \ll D$.

Limit on performance. Although we are adding i.i.d. Gaussian noise with a specific variance to $\mathbf{A}_s$ at each site, the truncation operation (i.e., computing the low-rank matrix $\mathbf{P}_s$) is not linear and affects the noise distribution. This means that the noise in $\mathbf{P}_s$ is neither i.i.d. nor zero-mean. Therefore, at the aggregator the estimator $\mathbf{A}_c = \frac{1}{S}\sum_{s=1}^{S} \mathbf{P}_s\mathbf{P}_s^\top$ has a larger variance to account for the non-zero mean and non-i.i.d. noise. If we instead send the $D \times D$ matrix $\hat{\mathbf{A}}_s$ from each site to the aggregator, the same estimator $\mathbf{A}_c = \frac{1}{S}\sum_{s=1}^{S} \hat{\mathbf{A}}_s$ would have much smaller variance and this would provide better performance. However, the performance is still limited by the larger sensitivity of the function $f(\mathbf{A}_s) = \mathbf{A}_s$. That is, if the aggregator was trusted, then we could add noise just once at the aggregator before releasing the PCA subspace $\mathbf{V}_K$. The variance of the noise is proportional to $\frac{1}{N^2}$, whereas the variance of the estimator $\mathbf{A}_c = \frac{1}{S}\sum_{s=1}^{S} \hat{\mathbf{A}}_s$ is proportional to $\frac{S}{N^2}$. The complete proof is omitted due to page count limitations.

4. EXPERIMENTAL RESULTS

It is apparent that the differentially private distributed PCA algorithm has a large parameter space. In particular, one might be interested to know how the performance of the algorithm varies as a function of privacy level, sample size and number of sites. Another interesting parameter is the intermediate dimension R. In this section, we present experimental results to show empirical comparison between the proposed algorithm and differentially private PCA on pooled data. We also included the performance variation of differentially private PCA on local data (i.e. data of a single site). We performed experiments on three datasets: a *synthetic* dataset ($D = 200$, $K = 50$) generated with zero mean and a pre-determined covariance matrix, the *MNIST* dataset ($D = 784$, $K = 50$) [18] (MNIST) and the *Covertype* dataset ($D = 54$, $K = 10$) [19] (COVTYPE). The MNIST consists of handwritten digits and has a training set of 60000 samples, each of size 28×28. The COVTYPE contains the forest cover type for 30×30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. We collected the dataset from the UC Irvine KDD archive [20]. For our experiments, we randomly selected 60000 samples from the COVTYPE dataset. We preprocessed the data by subtracting the mean (centering) and normalizing the samples with the maximum $\mathscr{L}_2$ norm in each dataset to enforce the condition $\|\mathbf{x}_n\|_2 \leq 1 \ \ \forall n$. We note that this preprocessing step is not differentially private. However, this step can be modified to satisfy DP at the cost of some utility. In all cases we show the average performance over 10 runs of the algorithms. As a performance measure of the produced subspace from the algorithm, we choose the captured energy defined in (1). Let us denote the captured energy in the true subspace be $q_o = \text{tr}(\mathbf{V}_K(\mathbf{A})^\top \mathbf{A}\mathbf{V}_K(\mathbf{A}))$, where $\mathbf{A}$ is the pooled-data sample second-moment matrix. In Table 1, we show the notation we use for representing the performance measures of different algorithms. For comparison, we plot the ratio of these quantities with respect to q_o.

Dependence on privacy parameter ϵ. First, we explore the trade-off between privacy and utility; i.e., between ϵ and the captured energy. We note that the standard deviation of the added noise (entries in $\mathbf{E}$) is inversely proportional to ϵ – bigger ϵ means higher privacy risk but less noise and thus better utility. We observe this in our experiments as well. In Figures 1(a)-(c), we show the variation of the ratio of the quality of different subspaces with respect to q_o for different values of ϵ. For this experiment, we kept δ, the number of samples per site N_s and the number of sites S fixed. For all the datasets, we observed that as ϵ increases (higher privacy risk), the captured energy increases. q_{DPdisPCA} would reach the optimal q_o for sufficiently large ϵ. The distributed algorithm clearly outperforms the local PCA algorithm, which is intuitive because including the information from multiple sites to estimate a population parameter always results in better performance than using the data from a single site only. Another interesting observation is that for datasets with lower dimensional samples, we can use smaller ϵ (i.e., to guarantee lower privacy risk) for the same utility. We observe that the performance index $\frac{q_{\text{full}}}{q_o}$ is better than $\frac{q_{\text{DPdisPCA}}}{q_o}$. This is due to the increased variance of the estimator at the aggregator, as explained in Section 3. The performance gap can be considered as the cost of saving on communication overhead. Additionally, the performance gap between $\frac{q_{\text{full}}}{q_o}$ and $\frac{q_{\text{pooled}}}{q_o}$ is due to the increased noise variance resulting from the higher sensitivity of $f(\mathbf{A}_s)$ at local sites.

Dependence on number of samples N_s. Intuitively, it should be easier to guarantee smaller privacy risk ϵ and higher utility $q(\cdot)$ when the number of samples is large. Figures 1(d)-(f) show how the captured energy increases as a function of sample size per site N_s. The variation with N_s reinforces the results seen earlier with variation of ϵ. For a fixed ϵ and δ, the utility increases as we increase N_s. For sufficiently large local sample size, the captured energy will reach q_o. Again, we observe a sharper increase in utility for lower-dimensional dataset.

Dependence on number of sites S. Next, we conduct an experiment where we kept the total number of samples N fixed and vary the number of sites S. ϵ and δ were also kept constant. We observe in Figures 1(g)-(i) that for all the three datasets, as we increase the number of sites (i.e., decrease the sample size N_s at each site), the performance deteriorates and then reaches a saturation point. The decrease in performance can be explained as follows: because we have smaller and smaller samples per site, the estimate of the local second-moment matrix $\mathbf{A}_s$ deviates more and more from the true sample second-moment matrix $\mathbf{A}$. Moreover, as the sensitivity of the function $f(\mathbf{A}_s) = \mathbf{A}_s$ is $\frac{1}{N_s}$, we are adding noise with higher variance when the sample size is smaller. At the presence of noise

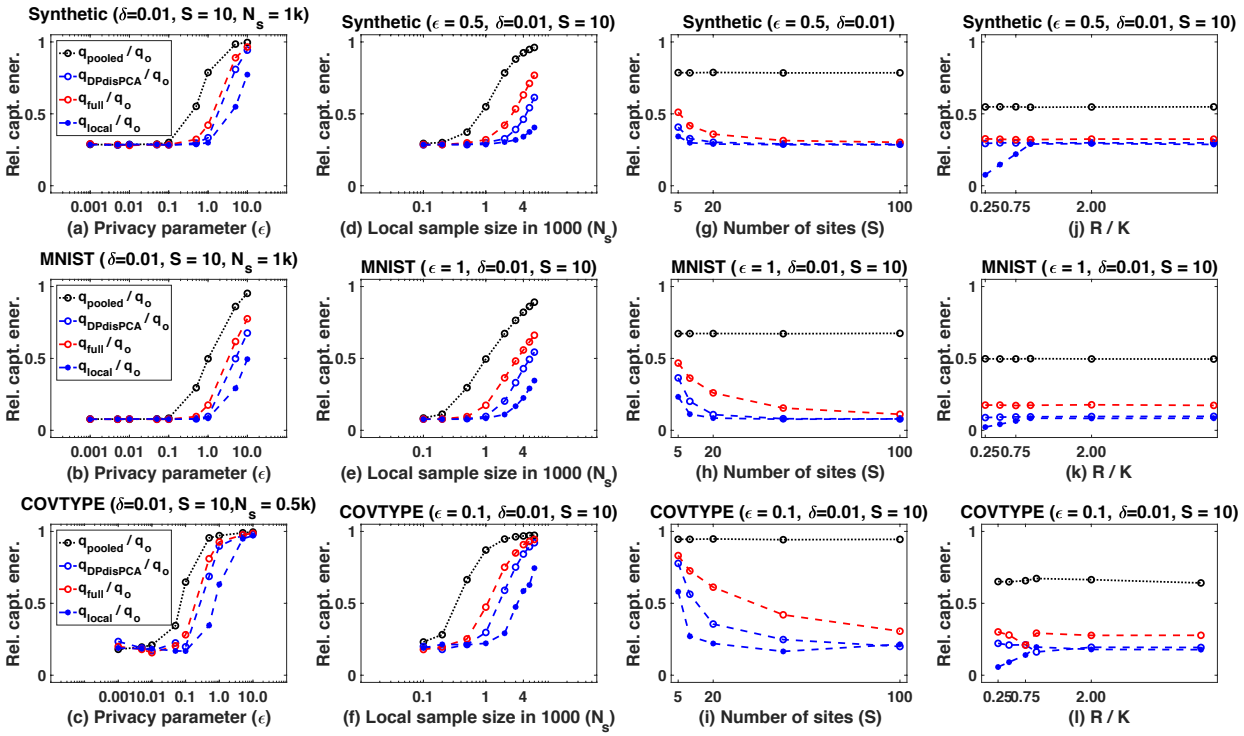

Fig. 1. Variation of the captured energy with: (a) - (c) different ϵ, (d) - (f) different number of data samples N_s, (g) - (i) different number of sites S, (j) - (l) different R for Synthetic, MNIST and COVTYPE data

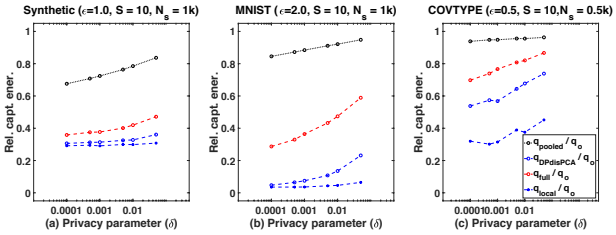

Fig. 2. Variation of the captured energy with δ for Synthetic, MNIST and COVTYPE data

with high variance, the PCA is essentially capturing only a few eigen-directions that are stronger than the noise, instead of capturing all the K directions that specify the data. Therefore, we observe the performance deterioration with increasing number of sites.

Dependence on intermediate dimension R. We explore the variation of performance with the intermediate dimension R. Intuitively, if R is too small, the matrix $\mathbf{P}_s$ that we send from each site will not represent the local second-moment matrix well. This affects the computation of $\mathbf{V}_K$. On the other hand, if we choose R to be too large, we are increasing the communication overhead (recall that each site sends a $D \times R$ matrix to the aggregator). In Figures 1(j)-(l), we show the variation of performance with the ratio $\frac{R}{K}$. We observe that initially the performance shows improvement as the ratio increases but soon it reaches saturation. This is because most of the energy of $\mathbf{A}_s$ is concentrated within a few eigen-directions. Including more directions does not necessarily increase the captured energy.

Dependence on privacy parameter δ. Finally, we explore the variation of performance with the other privacy parameter δ. Recall that δ can be considered as the probability that the algorithm releases the private information without guaranteeing privacy. We therefore want this to be as small as possible. However, lower δ results in larger noise variance. In Figure 2, we show how the performance indices vary with varying δ. We observe that if δ is not too small, the proposed algorithm can achieve good utility.

5. CONCLUSION

In this paper, we proposed a differentially private distributed PCA algorithm based on the distributed PCA scheme proposed in [7]. We showed empirically on synthetic and real datasets that it is possible to have meaningful utility while preserving privacy. We investigated the performance variation of the proposed algorithm with different parameters and showed that the utility can reach that of the non-private algorithm for some parameter choices. We also observed that for lower-dimensional datasets, we can achieve the same utility for a stricter privacy guarantee and a smaller sample size. We empirically verified the intuition that the distributed PCA always produces better results than local PCA. We investigated the cost of performance for lower communication overhead and the limit on the performance of a distributed PCA algorithm using the Gaussian mechanism. As a future work, one can try to find utility bounds in terms of privacy parameters and an aggregation mechanism that achieves better performance by accounting for the density of the truncated noise.

6. REFERENCES

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Proceedings of the Third Conference on Theory of Cryptography*, 2006, pp. 265–284.

[2] Y. Liang, M.-F. Balcan, and V. Kanchanapally, "Distributed PCA and k-Means Clustering," *Online, pages.cs.wisc.edu/∼yliang/distributedPCAandCoreset.pdf*.

[3] M.-F. Balcan, V. Kanchanapally, Y. Liang, and D. Woodruff, "Improved Distributed Principal Component Analysis," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, NIPS'14, pp. 3113–3121.

[4] Y. L. Borgne, S. Raybaud, and G. Bontempi, "Distributed Principal Component Analysis for Wireless Sensor Networks," *CoRR*, vol. abs/1003.1967, 2010.

[5] Z.-J. Bai, R. H. Chan, and F. T. Luk, *Principal Component Analysis for Distributed Data Sets with Updating*, pp. 471–483, Berlin, Heidelberg, 2005.

[6] S. V. Macua, P. Belanovic, and S. Zazo, "Consensus-based Distributed Principal Component Analysis in Wireless Sensor Networks," in *2010 IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2010, pp. 1–5.

[7] D. Feldman, M. Schmidt, and C. Sohler, "Turning Big Data into Tiny Data: Constant-size Coresets for K-means, PCA and Projective Clustering," in *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2013, SODA '13, pp. 1434–1453.

[8] K. L. Clarkson and D. P. Woodruff, "Low-Rank Approximation and Regression in Input Sparsity Time," *J. ACM*, vol. 63, no. 6, pp. 54:1–54:45, Jan. 2017.

[9] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, May 2011.

[10] I. E. K. Harrane, R. Flamary, and C. Richard, "Toward Privacy-preserving Diffusion Strategies for Adaptation and Learning over Networks," in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 1513–1517.

[11] A. Rajkumar and S. Agarwal, "A Differentially Private Stochastic Gradient Descent Algorithm for Multiparty Classification," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, La Palma, Canary Islands, 21–23 Apr 2012, vol. 22 of *Proceedings of Machine Learning Research*, pp. 933–941, PMLR.

[12] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic Gradient Descent with Differentially Private Updates," in *2013 IEEE Global Conference on Signal and Information Processing*, Dec 2013, pp. 245–248.

[13] H. Imtiaz, R. Silva, B. Baker, S. M. Plis, A. D. Sarwate, and V. Calhoun, "Privacy-preserving Source Separation for Distributed Data Using Independent Component Analysis," in *2016 Annual Conference on Information Science and Systems (CISS)*, March 2016, pp. 123–127.

[14] G. W. Stewart, "On the Early History of the Singular Value Decomposition," *SIAM Rev.*, vol. 35, no. 4, pp. 551–566, Dec. 1993.

[15] A. D. Sarwate and K. Chaudhuri, "Signal Processing and Machine Learning with Differential Privacy: Theory, Algorithms, and Challenges," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 86–94, September 2013.

[16] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[17] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze Gauss: Optimal Bounds for Privacy-preserving Principal Component Analysis," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, 2014, pp. 11–20.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[19] C. Blake, E. Keogh, and C. Merz, "UCI Repository of Machine Learning databases," 1999.

[20] M. Lichman, "UCI Machine Learning Repository," 2013.