Privacy-preserving source separation for distributed data using independent component analysis

Hafiz Imtiaz* Rogers Silva† Bradley Baker‡ Sergey M. Plis† Anand D. Sarwate* Vince Calhoun†
*Rutgers University †Mind Research Network ‡New College of Florida
hafiz.imtiaz@rutgers.edu, asarwate@ece.rutgers.edu, bradley.baker12@ncf.edu
rsilva@mrn.org, s.m.plis@gmail.com, vcalhoun@mrn.org

***Abstract*—Building good feature representations and learning hidden source models typically requires large sample sizes. In many applications, however, the size of the sample at an individual data holder may not be sufficient. One such application is neuroimaging analyses for mental health disorders – there are many individual research groups, each with a moderate number of subjects. Pooling such data can enable efficient feature learning, but privacy concerns prevent sharing the underlying data. We propose a model for private feature learning in which the data holders share differentially private views of their respective datasets to enable collaborative learning of a joint feature map. We give an example of such an algorithm for independent component analysis (ICA) – a popular blind source separation algorithm used in neuroimaging analyses. Our algorithm is a differentially private version of the recently proposed distributed joint ICA algorithm. We evaluate the performance of this method on simulated functional magnetic resonance imaging (fMRI) data.**

I. Introduction

Privacy is a central challenge in designing collaborative healthcare research systems. Researchers may wish to share data collected from local studies: such sharing can increase sample sizes leading to more robust findings that can address more complex phenomena. However, a combination of ethical, legal, and technological issues prevent them from openly sharing this data, because study subjects or patients may not wish to have their personal private health details shared without sufficient protections in place. Several recent studies have demonstrated the feasibility of identifying individuals from purportedly de-identified of anonymized data [1], [2], which only increases fears around open data sharing.

One alternative to open sharing of data is to share access to the private data via a "curator" that manages access to the data itself. In such a setting, algorithms for statistical analysis, signal processing, or machine learning must operate in a distributed manner. An algorithm can query an individual site's data through the curator. Only certain specified queries are permitted (e.g. histograms, linear functions of the data, or other simple functions), and the curator manages the privacy loss from answering these queries. The key to this quantification is using a privacy metric. One such metric is differential privacy [2], which measures privacy risk in terms of the difficulty of the hypothesis test for determining if an individual record is in the database or not. Differentially private algorithms are (almost always) randomized; the randomness creates the uncertainty in a hypothesis test that models an adversary attempting to infer the presence/absence of an individual data point in the data set. This randomness also means that the curator provides approximate answers to queries. Thus, the distributed algorithm must be robust to noise and that there is a trade-off between privacy risk (often denoted by ϵ) and accuracy (or utility) in distributed private data analysis.

One way in which larger sample sizes can help is learning more efficient feature representations for complex data. The use of a common lower-dimensional representation of the data can make further processing more efficient in terms of sample size, communication overhead, and robustness. Furthermore, such a representation can be learned once but used in many further computations. The composition property of differential privacy shows that further data-independent processing of the output of an ϵ-differentially private algorithm guarantees privacy risk no more than ϵ, so the system only needs to "pay once" for learning the feature map.

In this paper we adapt a recently-proposed algorithm for decentralized feature learning based on independent component analysis (ICA) [3]. The distributed joint ICA algorithm (djICA) can be employed to perform temporal ICA of functional magnetic resonance imaging (fMRI) data. ICA is a blind source separation algorithm; the goal of temporal ICA is to identify temporally independent components that represent activation of different neurological regions over time. Because temporal ICA operates on such high dimensional data, it requires more samples than are typically available from a single study. This is because of computational complexity and statistical sample size - the ratio of spatial to temporal dimensions often requires the aggregate temporal dimension to be similar to the voxel dimension [4]. We propose a differentially-private distributed ICA algorithm based on that of Baker et al. [4], which can be applied to decentralized data. The approach combines local computations as well as global computations to obtain both local and global parameters. We show that in some regimes, our proposed algorithm can find underlying sources in decentralized data nearly as accurately as centralized or pooled data.

II. Problem Formulation

As mentioned earlier, ICA is quite popular for blind source separation. It assumes that the observed signals are mixtures of statistically independent sources [5]. Therefore, it aims to decompose the mixed signals into the independent sources. In

978-1-4673-9457-4/16/$31.00 ©2016 IEEE

order to produce physiologically interpretable robust features, ICA has been applied to brain imaging data. Successful application of ICA on fMRI can be attributed to sparsity [6] and statistical independence between the underlying sources [5].

In this paper we consider a generative ICA model, where the independent sources $\mathbf{S} \in \mathbb{R}^{r \times N}$, composed of N observations from r statistically independent components, and a linear mixing process, defined with a mixing matrix $\mathbf{A} \in \mathbb{R}^{d \times r}$, form the observed data $\mathbf{X} \in \mathbb{R}^{d \times N}$ as a product $\mathbf{X} = \mathbf{AS}$. Many ICA algorithms propose to recover the "unmixing matrix" $\mathbf{W} = \mathbf{A}^{-1}$, assuming that $\mathbf{A}$ is invertible [4], by trying to maximize independence between rows of the product $\mathbf{WX}$. The maximal information transfer (Infomax) is a popular heuristic for estimating $\mathbf{W}$ that results in maximizing an entropy functional related to $\mathbf{WX}$. Denoting the sigmoid function as

$$g(z) = \frac{1}{1 + \exp(-z)} \tag{1}$$

we apply $g(\cdot)$ to a matrix or vector $\mathbf{Z}$ element-wise: $g(\mathbf{Z})$ is a matrix with the same size as $\mathbf{Z}$ and $(g(\mathbf{Z}))_{ij} = g(Z_{ij})$. The (differential) entropy of a random vector $\mathbf{Z}$ with joint density q is

$$h(\mathbf{Z}) = -\int q(\mathbf{Z}) \log q(\mathbf{Z}) d\mathbf{Z}. \tag{2}$$

Now, the objective of Infomax ICA can be expressed as

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}\, h(g(\mathbf{WX})). \tag{3}$$

In this paper we propose to modify a recently published [4] decentralized data ICA algorithm to ensure differential privacy [2]. An algorithm $\mathcal{A}(\mathbb{D})$ taking values in a set $\mathbb{T}$ provides (ϵ, δ)-differential privacy if

$$\Pr(\mathcal{A}(\mathbb{D}) \in \mathbb{S}) \leq \exp(\epsilon) \Pr(\mathcal{A}(\mathbb{D}') \in \mathbb{S}) + \delta, \tag{4}$$

for all measurable $\mathbb{S} \subseteq \mathbb{T}$ and all data sets $\mathbb{D}$ and $\mathbb{D}'$ differing in a single entry. 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 parameters, where low ϵ and δ ensure more privacy. It should be noted here that the parameter δ can be interpreted as the probability that the algorithm fails. Therefore, an $(\epsilon, 0)$-differentially private algorithm guarantees much stronger privacy than an (ϵ, δ)-differentially private algorithm, where $\delta > 0$. We refer to $(\epsilon, 0)$ differential privacy as ϵ-differential privacy. For more details, see the recent survey [7] or monograph [8].

In our setup, we have data distributed in different sites. We want to learn "good" features by utilizing samples from all sites while ensuring differential privacy. As mentioned before, computations are performed in local sites as well as the central site. In this scenario, two privacy concerns may arise depending on whether the central site is trusted or not. The local site can employ non-private algorithms and send the parameters to the central site if the central site is trusted. However, in the more general case, the central site is not trusted and the local sites employ differentially-private algorithms for computation of parameters.

III. Algorithm

A. Decentralized Joint ICA

The djICA algorithm is an ICA algorithm that can be applied to decentralized data [4]. A number of modified ICA algorithms exist for joining various data sets [9] together and performing simultaneous decomposition of data from a number of subjects and modalities [10]. For instance, group spatial ICA (GICA) is a noteworthy one for multi-subject analysis of task- and resting-state fMRI data [11]. It assumes that the spatial map components ($\mathbf{S}$) are similar across subjects. On the other hand, the joint ICA (jICA) [12] algorithm for multimodal data fusion assumes that the mixing process ($\mathbf{A}$) is similar over a group of subjects. However, group temporal ICA also assumes common spatial maps but pursues statistical independence of timecourses. Consequently, like jICA, the common spatial maps from temporal ICA describe a common mixing process ($\mathbf{A}$) among subjects. While very interesting, temporal ICA of fMRI is typically not investigated because of the small number of time points in each data set, which leads to unreliable estimates [4]. The decentralized jICA approach overcomes that limitation by leveraging information from data sets distributed over multiple sites.

As in djICA [4], we take a model with s sites in which site i has a collection of data matrices $\{\mathbf{X}_{i,m} \in \mathbb{R}^{d \times n_i} : m = 1, 2, \ldots, M_i\}$ consisting of a total time course of length n_i time points over d voxels for M_i individuals. Sites concatenate their local data matrices temporally to form a $d \times n_i M_i$ data matrix $\mathbf{X}_i \in \mathbb{R}^{d \times N_i}$ where $N_i = n_i M_i$. Let $N = \sum_{i=1}^{s} N_i$ be the total length. We assume a common (global) mixing matrix $\mathbf{A} \in \mathbb{R}^{d \times r}$ that generates the time courses in $\mathbf{X}_i$ from underlying sources $\mathbf{S}_i \in \mathbb{R}^{r \times N_i}$ at each site. This yields the following model:

$$\mathbf{X} = [\mathbf{AS}_1\ \mathbf{AS}_2\ \cdots \mathbf{AS}_s] \in \mathbb{R}^{d \times N}. \tag{5}$$

Decentralized joint ICA [4] uses locally computed gradients to estimate a common, global unmixing matrix $\mathbf{W} \in \mathbb{R}^{r \times d}$ corresponding to the Moore-Penrose pseudo-inverse of $\mathbf{A}$, denoted $\mathbf{A}^{+}$.

We follow a two-step distributed principal component analysis (dPCA) procedure [4] in which each site combines local PCA processing with a global PCA step after exchanging information. This is an alternative to a single step in which we compute a global PCA matrix but must send a $d \times d$ matrix between sites. Unfortunately, communicating a function of the local data may not save in the sense of differential privacy; our algorithmic contribution is to replace the PCA computations with differentially private PCA algorithms. In particular, we use a recently proposed SN algorithm [13] which is a fast and efficient algorithm for ϵ-differentially private PCA. The end result is a differentially private version of djICA.

Algorithm 1 PrivateLocalPCA

Require: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ (with n samples of dimension d, each sample has bounded norm), privacy parameter ϵ

1: $\mathbf{C} \leftarrow \mathbf{X}\mathbf{X}^\top$
2: Generate $d \times p$ matrix $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p]$ where $\mathbf{e}_i \sim \mathcal{N}(\mathbf{0}, \frac{1}{2\epsilon}\mathbf{I}_d)$ and $p = d + 1$
3: $\hat{\mathbf{C}} \leftarrow \mathbf{C} + \mathbf{E}\mathbf{E}^\top$
4: Compute the SVD $\hat{\mathbf{C}} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$.
5: Let $\mathbf{\Sigma}^{(k)} \in \mathbb{R}^{k\times k}$ contain the largest k singular values and $\mathbf{U}^{(k)} \in \mathbb{R}^{d\times k}$ the corresponding singular vectors.
6: Save $\mathbf{U}^{(k)}$ and $\mathbf{\Sigma}^{(k)}$ locally.
7: **return** $\mathbf{P} = \mathbf{U}^{(k)}\sqrt{\mathbf{\Sigma}^{(k)}}$.

B. Differentially-private dPCA algorithms

Here, we describe privacy-preserving dPCA algorithms for dimension reduction and whitening. This serves as a preprocessing step to standardize the data prior to djICA, also without communicating full data sets outside of local sites. We replace the LocalPCA and GlobalPCA algorithms in djICA by a ϵ-differentially private PCA algorithm [13]. The DP-PCA algorithm provides an ϵ-differentially private approximation to the data second-moment matrix $\mathbf{C} = \mathbf{X}\mathbf{X}^\top$. For completeness, we reproduce the method in Algorithm 1.

The second algorithm is simply the GlobalPCA algorithm of djICA with calls to LocalPCA replaced by calls to PrivateLocalPCA, and is given in Algorithm 2.

Algorithm 2 PrivateGlobalPCA

Require: s sites with data $\{\mathbf{X}_i \in \mathbb{R}^{d\times N_i} : i = 1, 2, \ldots, s\}$, intended final rank r, local rank $k \geq r$.

1: Choose a random order π for the sites.
2: $\mathbf{P}(1) = \text{PrivateLocalPCA}(\mathbf{X}_{\pi(1)}, \min\{k, \text{rank}(\mathbf{X}_{\pi(1)})\})$
3: **for all** $j = 2$ to s **do**
4: $\quad i = \pi(j)$
5: $\quad$ Send $\mathbf{P}(j-1)$ from site $\pi(j-1)$ to site $\pi(j)$
6: $\quad k' = \min\{k, \text{rank}(\mathbf{X}_i)\}$
7: $\quad \mathbf{P}' = \text{PrivateLocalPCA}(\mathbf{X}_i, k')$
8: $\quad k' = \max\{k', \text{rank}(\mathbf{P}(j-1))\}$
9: $\quad \mathbf{P}(j) = \text{PrivateLocalPCA}([\mathbf{P}'\ \mathbf{P}(j-1)], k')$
10: **end for**
11: $r' = \min\{r, \text{rank}(\mathbf{P}(s))\}$
12: $\mathbf{U} = \text{NormalizeTopColumns}(\mathbf{P}(s), r')$ ▷ At last site
13: Send $\mathbf{U}$ to sites $\pi(1), \ldots, \pi(s-1)$.
14: **for all** $i = 1$ to s **do**
15: $\quad \mathbf{X}_{i,\text{red}} = \mathbf{U}^\top\mathbf{X}_i$ ▷ The locally reduced data
16: **end for**

C. Privacy-preserving djICA

Our private version of the djICA algorithm is shown in Algorithm 3. We replace the LocalPCA procedure in djICA with PrivateLocalPCA to guarantee differential privacy for the preprocessing step. In the PrivateGlobalPCA step (Algorithm 2), each site i first computes a differentially private PCA subspace from its local data. Let $\mathbf{U}_i \in \mathbb{R}^{d\times k}$ and $\mathbf{\Sigma}_i \in \mathbb{R}^{k\times k}$ denote the top singular vectors and values from this decomposition and $\mathbf{U} \in \mathbb{R}^{d\times r}$ the common projection matrix produced by PrivateGlobalPCA. The sites receive $\mathbf{U}$ and project their local data to produce $\mathbf{X}_{i,\text{red}} \in \mathbb{R}^{r\times N_i}$. The projected data is the input to the iterative djICA algorithm that estimates the unmixing matrix $\mathbf{W}$ as described in Algorithm 3 [4].

Algorithm 3 differentially private decentralized joint ICA (djICA)

Require: data $\{\mathbf{X}_{i,\text{red}} \in \mathbb{R}^{r\times N_i} : i = 1, 2, \ldots, s\}$, where r is the same across sites, tolerance level $t = 10^{-6}$, $j = 0$, maximum iterations J, $\|\Delta_\mathbf{W}(0)\|_2^2 = t$, initial learning rate $\rho = 0.015/ln(r)$

1: Initialize $\mathbf{W} \in \mathbb{R}^{r\times r}$ ▷ for example, $\mathbf{W} = \mathbf{I}$
2: **while** $j < J$, $\|\Delta_\mathbf{W}(j)\|_2^2 \geq t$ **do**
3: $\quad$ **for all** sites $i = 1, 2, \ldots, s$ **do**
4: $\quad\quad$ Generate $\mathbf{E} \in \mathbb{R}^{r\times N_i}$ i.i.d $\sim$ Lap($\frac{\|W\|_1}{\epsilon}$)
5: $\quad\quad \mathbf{Z}_i(j) = \mathbf{W}(j-1)\mathbf{X}_i + \mathbf{b}(j-1)\mathbf{1}^\top + \mathbf{E}$
6: $\quad\quad \mathbf{Y}_i(j) = g(\mathbf{Z}_i(j))$
7: $\quad\quad \mathbf{G}_i(j) = \rho\left(\mathbf{I} + (\mathbf{1} - 2\mathbf{Y}_i(j))\mathbf{Z}_i(j)^\top\right)\mathbf{W}(j-1)$
8: $\quad\quad \mathbf{h}_i(j) = \rho\sum_{m=1}^{N_i}(\mathbf{1} - 2\mathbf{y}_{m,i}(j))$
9: $\quad\quad$ Send $\mathbf{G}_i(j)$ and $\mathbf{h}_i(j)$ to the aggregator site.
10: $\quad$ **end for**
11: $\quad$ At the aggregator site, update global variables
12: $\quad \Delta_\mathbf{W}(j) = \sum_{i=1}^{s}\mathbf{G}_i(j)$
13: $\quad \mathbf{W}(j) = \mathbf{W}_i(j-1) + \Delta_\mathbf{W}(j)$
14: $\quad \mathbf{b}(j) = \mathbf{b}(j-1) + \sum_{i=1}^{s}\mathbf{h}_i(j)$
15: $\quad$ Check upper bound and learning rate adjustment.
16: $\quad$ Send global $\mathbf{W}(j)$ and $\mathbf{b}(j)$ back to each site
17: **end while**

Even though the preprocessing is done in a differentially private manner, the djICA algorithm itself may leak information about the sites' local data since it relies on iterative message-passing between the sites and a central aggregator. We therefore modify (5) to add Laplace noise to guarantee additional privacy in the iteration. The full mixing matrix for the global data is modeled as $\mathbf{A} = (\mathbf{W}\mathbf{U}^\top)^+ \in \mathbb{R}^{d\times r}$. The algorithm iteratively updates $\mathbf{W}$ using distributed gradient descent [14]. At each iteration j the sites update locally: in lines 4 and 5, the sites adjust the local source estimates $\mathbf{Z}_i$ by the bias estimates $\mathbf{b}(j-1)\mathbf{1}^\top \in \mathbb{R}^{r\times N_i}$, followed by the sigmoid transformation $g(\cdot)$; they then calculate local gradients with respect to $\mathbf{W}_i$ and $\mathbf{y}_i$ in lines 6 and 7. Here $\mathbf{y}_{m,i}(j)$ is the m-th column of $Y_i(j)$.

After converging on a common $\mathbf{W}$, each site estimates its sources $\mathbf{S}_i$:

$$\mathbf{S}_i = \mathbf{W}\mathbf{X}_{i,\text{red}}. \tag{6}$$

D. Proposed Differentially-private djICA Algorithm

Theorem 1 (Differentially-private djICA Algorithm): Adding a matrix $\mathbf{E}$, which contains i.i.d samples from Lap($\frac{\|W\|_1}{\epsilon}$), to the variable Z in step 4 of the djICA algorithm makes

the algorithm $(J\epsilon, 0)$-differentially private, where W is the unmixing matrix.

Proof: Let us consider two neighboring data matrices $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}_M]$ and $\mathbf{X}' = [\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}'_M]$, i.e., the two matrices are same except the last (or any one) individual is swapped. We assume that individual data matrices satisfy $\|\mathbf{x}_i\|_2 \leq \frac{1}{2\sqrt{d}}$. Therefore, $\|\mathbf{x}_i\|_1 \leq \frac{1}{2}$. Now

$$\begin{aligned}\|\mathbf{X} - \mathbf{X}'\|_2 &= \|\mathbf{x}_M - \mathbf{x}'_M\|_2 \\ &\leq \|\mathbf{x}_M\|_2 + \|\mathbf{x}'_M\|_2 \\ &\leq \frac{1}{\sqrt{d}}\end{aligned}$$

Using the inequality between $\mathcal{L}_2$-norm and $\mathcal{L}_1$-norm, we have

$$\|\mathbf{X} - \mathbf{X}'\|_1 \leq \sqrt{d}\|\mathbf{X} - \mathbf{X}'\|_2 \leq \sqrt{d}\frac{1}{\sqrt{d}} = 1 \qquad (7)$$

In the PCA pre-processing step, we reduce the dimension of the data matrix $\mathbf{X}$ from $d \times N$ to $r \times N$. Let us denote this reduced dimension data matrices as $\mathbf{Y}$ and $\mathbf{Y}'$. From the definition of $\mathcal{L}_2$-norm of matrices, we can state that $\|\mathbf{Y}\|_2 = \|\mathbf{X}\|_2$. So, the relation $\|\mathbf{Y} - \mathbf{Y}'\|_1 \leq 1$ should also hold. Under these conditions, we define the following function:

$$\mathbf{Z} = f(\mathbf{Y}) = \mathbf{W}\mathbf{Y} + \mathbf{b}, \qquad (8)$$

where $\mathbf{W} \in \mathbb{R}^{r \times r}$ is the weights matrix or the unmixing matrix and $\mathbf{b} \in \mathbb{R}^{r \times N}$ is the bias estimate matrix. The $\mathcal{L}_1$ sensitivity of the function f is defined as

$$\Delta f = \max_{\|\mathbf{Y} - \mathbf{Y}'\|_1 \leq 1} \|f(\mathbf{Y}) - f(\mathbf{Y}')\|_1. \qquad (9)$$

This signifies the magnitude by which a single individual's data can change the function output in the worst case, and is the uncertainty to be introduced in order to hide the participation of a single individual. According to the Laplace mechanism, we need to add a noise matrix $\mathbf{E} \in \mathbb{R}^{r \times N}$ to $f(\mathbf{Y})$ to make it $(\epsilon, 0)$-differentially private. Here, the matrix $\mathbf{E}$ consists i.i.d. samples from a Laplace distribution with variance $2\left(\frac{\Delta f}{\epsilon}\right)^2$. Now, using (8) and (9), we have

$$\begin{aligned}\Delta f &= \max_{\|\mathbf{Y} - \mathbf{Y}'\|_1 \leq 1} \|f(\mathbf{Y}) - f(\mathbf{Y}')\|_1 \\ &= \max_{\|\mathbf{Y} - \mathbf{Y}'\|_1 \leq 1} \|\mathbf{W}(\mathbf{Y} - \mathbf{Y}')\|_1\end{aligned}$$

But we note that $\|\mathbf{W}(\mathbf{Y} - \mathbf{Y}')\|_1 \leq \|\mathbf{W}\|_1\|\mathbf{Y} - \mathbf{Y}'\|_1$. So, we have $\Delta f = \|\mathbf{W}\|_1$. Therefore, if we add the noise matrix $\mathbf{E}$ in each step at any site with i.i.d. entries from Lap$(\frac{\Delta f}{\epsilon})$, then the djICA algorithm is $(\epsilon', 0)$-differentially private, where $\epsilon' = J\epsilon$ and J is the total number of iterations required.

IV. Experimental Results

We generated synthetic data from the same model as Baker et al. [4] to test the impact of privacy on the djICA algorithm. We tested the difference between ICA with global PCA pre-processing on the pooled data and ICA with the two-stage distributed PrivateGlobalPCA algorithm. The source signals $\mathbf{S}$ were simulated using the generalized autoregressive (AR) conditional heteroscedastic (GARCH) model [15], [16]. We have utilized 1024 and 512 simulated subjects in our experiments for the distributed setting and for pooled data setting, respectively.

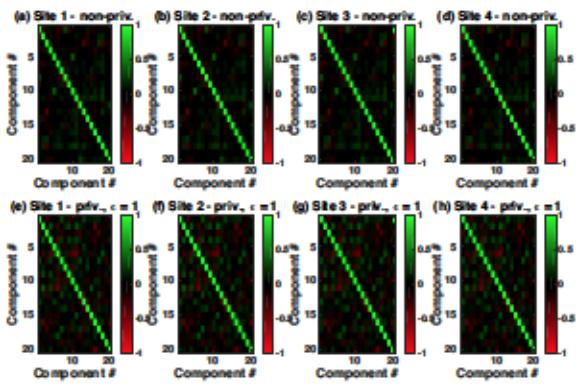

Fig. 1. Independence of components for a fixed ϵ for distributed synthetic data

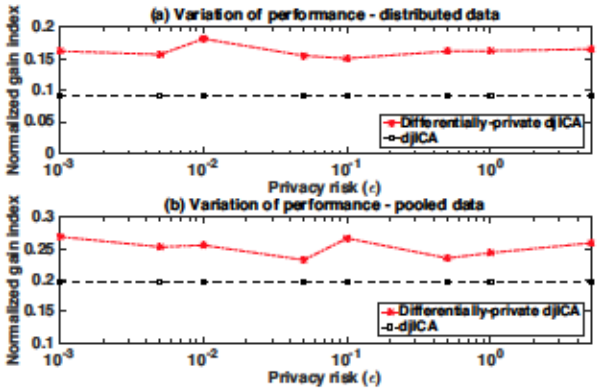

Fig. 2. Variation of normalized gain index of differentially-private djICA algorithm with different ϵ with that of djICA algorithm for (a) distributed data (1024 subjects) and (b) pooled data (512 subjects)

In order to demonstrate the quality of the estimated components, it is intuitive to show the auto-correlation plots of the estimated independent components $\mathbf{S}_{\text{est}}$. Fig. 1(a)-(d) show the auto-correlation of $\mathbf{S}_{\text{est}}$ obtained from the original djICA algorithm and Fig. 1(e)-(h) show the auto-correlation of $\mathbf{S}_{\text{est}}$ obtained from the proposed differentially-private djICA algorithm with fixed privacy ($\epsilon = 1$). As mentioned before, we used 1024 subjects equally distributed among 4 sites. We observe that for $\epsilon = 1$, the components are almost as independent as the non-private algorithm. However, with the decrease of ϵ to ensure more privacy, the independence of the components starts to reduce quite sharply. We observed similar results for the pooled data scenario.

We employ another index that quantizes the quality of the unmixing matrix. One such index is the normalized gain index [17], which varies from 0 to 1, with 0 indicating that the unmixing matrix is an identity matrix. Fig. 2(a) shows the variation of this index for differentially-private djICA

algorithm with different privacy levels (i.e. different ϵ values) for distributed data. For comparison with the non-private djICA algorithm, we included the graph of the same index in the same plot. Here, we observe that privacy has a negative impact on the normalized gain index, i.e., the performance of the ICA algorithm. This can be considered as the price one has to pay in order to ensure privacy. For the pooled data scenario, we observed similar performance degradation of the differentially-private djICA algorithm when compared with non-private djICA as shown in Figure 2(b) for the non-pooled case. Because the sample size was smaller for the pooled data simulations, the performance of the differentially-private djICA algorithm and the non-private djICA algorithm is slightly worse than their distributed counterparts.

V. Conclusions

In this paper we proposed some modifications to a recently-proposed algorithm for feature learning using decentralized data ICA. The proposed algorithm can be applied to temporal ICA of fMRI data. The djICA algorithm is capable of extracting features from distributed data almost as good as from the pooled data. We have generated synthetic data according to a popular model for our experiments. We formulated the whole system in such a way that some computations are performed in local sites and some are performed in the central site - this reduces the cost of transmitting large matrices. We have graphically demonstrated that the recovered components are statistically independent even for quite strict privacy guarantee. We have also shown the variation of privacy with a performance index, which showed empirically that the proposed differentially-private algorithm ensures good utility while preserving privacy.

Although our results in this paper are not comprehensive, they indicate the insisting on "untrusted" computation infrastructures – making each site render all messages differentially private before communicating – can have a significant impact on privacy-preserving feature learning. An alternative, left for future work, is to have the centralized aggregator be trusted – sites can communicated functions of their datas (in this case, PCA subspaces and gradients) and the aggregator can act as a differentially private curator, communicating back to the sites in a differentially private manner. Even though each site is contributing some data, the privacy guarantees they will not learn "too much" about the other sites' data. This has the added benefit of significantly reducing the amount of added noise and may salvage some of the performance loss, allowing stronger privacy protections by reducing ϵ.

Acknowledgment

This work was sponsored in part by NSF award CCF-1453432, NIH award 1R01DA040487-01A1, and DARPA and the US Navy under contract #N66001-15-C-4070.

References

[1] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "A near-optimal algorithm for differentially-private principal components," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2905–2943, Jan. 2013.

[2] 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.

[3] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287 – 314, 1994.

[4] B.T. Baker, R.F. Silva, V.D. Calhoun, A.D. Sarwate, and S.M. Plis, "Large scale collaboration with autonomy: Decentralized data ica," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, Sept 2015, pp. 1–6.

[5] V. D. Calhoun, V. K. Potluru, R. Phlypo, R. F. Silva, B. A. Pearlmutter, A. Caprihan, S. M. Plis, and T. Adalı, "Independent component analysis for brain fMRI does indeed select for maximal independence," *PLoS ONE*, vol. 8, pp. e73309, 2013.

[6] I. Daubechies, E. Roussos, S. Takerkart, M. Benharrosh, C. Golden, K. D'Ardenne, W. Richter, J. D. Cohen, and J. Haxby, "Independent component analysis for brain fMRI does not select for independence," *Proceedings of the National Academy of Sciences*, vol. 106, no. 26, pp. 10415–10422, 2009.

[7] 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.

[8] 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.

[9] J. Sui, T. Adalı, G. D. Pearlson, and V. D. Calhoun, "An ICA-based method for the identification of optimal fMRI features and components using combined group-discriminative techniques," *NeuroImage*, vol. 46, no. 1, pp. 73 – 86, 2009.

[10] J. Liu and V. Calhoun, "Parallel independent component analysis for multimodal analysis: Application to fMRI and EEG data," in *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, April 2007, pp. 1028–1031.

[11] E. A. Allen, E. B. Erhardt, E. Damaraju, W. Gruner, J. M Segall, R. F. Silva, M. Havlicek, S. Rachakonda, J. Fries, R. Kalyanam, A. M. Michael, A. Caprihan, J. A. Turner, R. Eichele, S. Adelsheim, A. D. Bryan, J. Bustillo, V. P. Clark, S. W. Feldstein Ewing, F. Filbey, C. C. Ford, K. Hutchison, R. E Jung, K. A. Kiehl, P. Kodituwakku, Y. M. Komesu, A. R. Mayer, G. D. Pearlson, J. P. Phillips, J. R. Sadek, M. Stevens, U. Teuscher, R. J. Thoma, and V. D. Calhoun, "A baseline for the multivariate comparison of resting state networks," *Frontiers in Systems Neuroscience*, vol. 5, no. 2, 2011.

[12] V.D. Calhoun, T. Adali, N.R. Giuliani, J.J. Pekar, K.A. Kiehl, and G.D. Pearlson, "Method for multimodal analysis of independent source differences in schizophrenia: Combining gray matter structural and auditory oddball functional data," *Human Brain Mapping*, vol. 27, no. 1, pp. 47 – 62, 2006.

[13] H. Imtiaz and A. D. Sarwate, "Symmetric matrix perturbation for differentially-private principal component analysis," in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, to appear 2016.

[14] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*, 1996, pp. 757–763.

[15] R. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.

[16] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J Econometrics*, vol. 31, pp. 307–327, 1986.

[17] K. Nordhausen, E. Ollila, and H. Oja, "On the performance indices of ica and blind source separation," in *Proceedings of the 12th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2011, pp. 486–490.