DISTRIBUTED DIFFERENTIALLY-PRIVATE CANONICAL CORRELATION ANALYSIS

Hafiz Imtiaz and Anand D. Sarwate

Department of Electrical and Computer Engineering, Rutgers University

ABSTRACT

We propose a distributed differentially-private canonical correlation analysis (CCA) algorithm to use on multi-view data. CCA finds a subspace for each view such that projecting the views onto these subspaces simultaneously reduces the dimension and maximizes correlation. In applications involving privacy-sensitive data, such as medical imaging, distributed privacy-preserving algorithms can let data holders maintain local control of their data while participating in joint computations with other data holders. Differential privacy is a framework for quantifying the privacy risk in such settings. However, conventional distributed differentially-private algorithms introduce more noise to guarantee a given level of privacy compared to their centralized counterparts. Our differentially-private CCA employs a noise-reduction strategy to achieve the same utility level as CCA on centralized data. Experiments on synthetic and real data show the benefit of our approach over conventional methods.

Index Terms— differential privacy, canonical correlation analysis, multi-view learning, clustering, distributed data

1. INTRODUCTION

Many signal processing and machine learning algorithms operate on private or sensitive data and can leak information about individuals present in the data set. Differential privacy [1] (DP) quantifies privacy risk in such settings: DP learning algorithms attempt to produce estimates of population properties that do not have a strong dependence on individual data points.

When private data is distributed over different locations or sites, DP algorithms can allow sites (data holders) holding a smaller number of samples to jointly learn features from the aggregate data. One example comes from neuroimaging: many research groups may study the same mental health disorder but each group may have a modest number of subjects at best. However, learning meaningful population properties or efficient feature representations from high-dimensional functional magnetic resonance imaging (fMRI) data requires a large sample size. Pooling the data at a central location may enable efficient feature learning, but privacy concerns and high communication overhead often prevent such sharing. Additionally, conventional distributed DP algorithms suffer from more noise for a given privacy level when compared to their centralized counterparts. Therefore, it is desirable to have efficient distributed privacy-preserving algorithms that provide utility close to centralized case [2, 3].

Canonical correlation analysis (CCA) [4] is a tool for characterizing linear relationships between two (or more) multidimensional variables (or "views"). The views are typically different measurements of the same physical phenomena. CCA finds the bases for each view such that the correlation matrix between the data projected onto the bases is diagonal and the correlations on the diagonal are maximized [5]. It has been used as a pre-processing step for dimensionality reduction in high-dimensional clustering, statistical analysis, medical studies and recently in machine learning, neuro-science and signal processing [6, 7, 5, 8]. The advantage of CCA over principal component analysis (PCA) or random projections [9, 10, 11, 12] is that CCA can jointly learn projection maps to improve clustering performance for *multi-view learning* [13, 14]. CCA also has applications in blind source separation, such as in fMRI analysis [15, 6, 16].

We propose a decentralized version of our centralized differentially private CCA [8] algorithm. We eliminate the excessive noise problem of conventional distributed DP algorithms by employing a correlated noise scheme. Such a system requires an honest third party to generate the correlated noise, which is feasible in trust models such as those in medical research consortia. To our knowledge, this paper proposes the first DP CCA algorithm for distributed settings. We demonstrate our approach using synthetic and real data sets to show how the utility/performance is affected by the privacy risk, number of samples and some other key parameters. Simulation results show that our distributed algorithm can achieve the same utility as the pooled data scenario satisfying (ϵ, δ)-differential privacy. For some parameter choices, our algorithm can achieve almost as much utility as the non-private algorithm, showing that meaningful privacy can (almost) come for free.

2. PROBLEM FORMULATION

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. We represent indices with lower-case regular letters and they typically run from 1 to their upper-case version (e.g. $n \in [N] \triangleq \{1, 2, \ldots, N\}$). We denote the n-th column of a matrix $\mathbf{X}$ as $\mathbf{x}_n$. Finally, we use $\|\cdot\|_2$, $\|\cdot\|_F$ and $\mathrm{tr}(\cdot)$ to denote the Euclidean norm of a vector (or spectral norm of a matrix), the Frobenius norm, and the trace, respectively.

Distributed CCA. Consider a system with S different sites, each holding disjoint data sets, and an untrusted central node or aggregator (see Fig. 1(a)). In site $s \in [S]$, the data is a pair of sample matrices $\mathbf{X}_s \in \mathbb{R}^{D_x \times N_s}$ and $\mathbf{Y}_s \in \mathbb{R}^{D_y \times N_s}$ corresponding to the two "views" of the same physical phenomena. The n-th column of $\mathbf{X}_s$ and $\mathbf{Y}_s$, denoted $\mathbf{x}_{s,n}$ and $\mathbf{y}_{s,n}$, respectively, are the n-th samples from view 1 and view 2. For simplicity, we assume that the observed samples are mean-centered. The sample size in site s is N_s and 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 data matrices would be $\mathbf{X} = [\mathbf{X}_1 \ldots \mathbf{X}_S] \in \mathbb{R}^{D_x \times N}$ and $\mathbf{Y} = [\mathbf{Y}_1 \ldots \mathbf{Y}_S] \in \mathbb{R}^{D_y \times N}$. The CCA projection vectors are defined to be the columns of the matrices $\mathbf{U} \in \mathbb{R}^{D_x \times K}$ and

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.

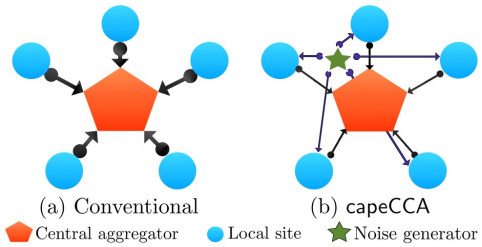

Fig. 1. Structure of the network: conventional and capeCCA

$\mathbf{V} \in \mathbb{R}^{D_y \times K}$ that solve the following problem [4, 17, 13]:

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad & \|\mathbf{U}^\top \mathbf{X} - \mathbf{V}^\top \mathbf{Y}\|_F^2 \\ \text{subject to} \quad & \frac{1}{N}\mathbf{U}^\top \mathbf{X}\mathbf{X}^\top \mathbf{U} = \mathbf{I}, \frac{1}{N}\mathbf{V}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{V} = \mathbf{I}, \\ & \frac{1}{N}\mathbf{U}^\top \mathbf{X}\mathbf{Y}^\top \mathbf{V} = \mathbf{I}, \end{aligned}$$

where $\mathbf{I}$ is the $K \times K$ identity matrix with $K \leq \min\{D_x, D_y\}$. The solution to the optimization problem [18] is given as follows: $\mathbf{U}^*$ and $\mathbf{V}^*$ contain the top-K eigenvectors of the matrices $\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}$ and $\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}$, respectively. Here, the sample covariance and cross-covariance matrices are given by $\mathbf{C}_{xx} = \frac{1}{N}\mathbf{X}\mathbf{X}^\top$, $\mathbf{C}_{yy} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^\top$ and $\mathbf{C}_{xy} = \frac{1}{N}\mathbf{X}\mathbf{Y}^\top = \mathbf{C}_{yx}^\top$. We assume that we obtain samples as $\mathbf{z}_{s,n} = \begin{bmatrix}\mathbf{x}_{s,n}^\top & \mathbf{y}_{s,n}^\top\end{bmatrix}^\top \in \mathbb{R}^D$, where $D = D_x + D_y$. We compute the $D \times D$ positive semi-definite sample covariance matrix of $\mathbf{Z} = [\mathbf{Z}_1 \ldots \mathbf{Z}_S] \in \mathbb{R}^{D \times N}$ as

$$\mathbf{C} = \frac{1}{N}\mathbf{Z}\mathbf{Z}^\top = \frac{1}{N}\sum_{s=1}^{S}\sum_{n=1}^{N_s}\mathbf{z}_{s,n}\mathbf{z}_{s,n}^\top \text{ and } \mathbf{C} = \begin{bmatrix}\mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{C}_{yy}\end{bmatrix}.$$

Without loss of generality, we can ensure that $\|\mathbf{z}_{s,n}\|_2 \leq 1$, because canonical correlations are invariant with respect to affine transformations of the variables [5]. We are interested in approximating $\mathbf{U}^*$ and $\mathbf{V}^*$ in a distributed setting while guaranteeing differential privacy. A naïve approach (non-privacy-preserving) would be to send the data matrices $\mathbf{X}_s$ and $\mathbf{Y}_s$ from the sites to the aggregator. The aggregator can then compute $\mathbf{C}$ and subsequently $\mathbf{U}^*$ and $\mathbf{V}^*$. However, when D_x, D_y and/or N_s are large, this results in a huge communication overhead. Additionally, in many scenarios, the local data are 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 to (i) ensure differential privacy, (ii) achieve the same utility as the pooled data scenario in a distributed setting and (iii) provide close approximations to the true CCA subspaces $\mathbf{U}^*$, $\mathbf{V}^*$.

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

$$\Pr(\mathscr{A}(\mathbb{D}) \in \mathbb{S}) \leq \exp(\epsilon)\Pr(\mathscr{A}(\mathbb{D}') \in \mathbb{S}) + \delta, \qquad (1)$$

for all measurable $\mathbb{S} \subseteq \mathbb{T}$ and all data sets $\mathbb{D}$ and $\mathbb{D}'$ differing in a single entry (neighboring data sets). This definition essentially states that the probability of the output of $\mathscr{A}(\mathbb{D})$ is not changed significantly if the corresponding database input is changed by one entry. Here, ϵ and δ are privacy risk parameters: lower ϵ and δ ensure more privacy. Note that δ can be interpreted as the probability that the algorithm fails. For details, see the survey [19] or monograph [20].

Conventional distributed differentially-private CCA. To solve for the optimal CCA subspaces $\mathbf{U}^*$ and $\mathbf{V}^*$, we need to compute the sample covariance matrix $\mathbf{C}$ in the distributed setting. The conventional privacy-preserving approach is for each site to use the Analyze Gauss method [21] to send an (ϵ, δ)-DP approximate $\hat{\mathbf{C}}_s = \mathbf{C}_s + \mathbf{E}_s$ of the local sample covariance matrix $\mathbf{C}_s$ to the aggregator, where $\mathbf{C}_s = \frac{1}{N_s}\mathbf{Z}_s\mathbf{Z}_s^\top$ and $\mathbf{E}_s$ is a $D \times D$ symmetric matrix with $\{[\mathbf{E}_s]_{ij} : i \in [D], j \leq i\}$ drawn i.i.d. from $\mathscr{N}(0, \tau_s^2)$. Here, the noise standard deviation is given by $\tau_s = \frac{1}{N_s\epsilon}\sqrt{2\log(\frac{1.25}{\delta})}$ using the $\mathscr{L}_2$-sensitivity [1] of $\mathbf{C}_s$: $\Delta_s = \frac{1}{N_s}$ [21] for the standard Gaussian mechanism [20]. Upon receiving the matrices $\{\hat{\mathbf{C}}_s\}$ from the sites, the aggregator computes

$$\hat{\mathbf{C}} = \frac{1}{S}\sum_{s=1}^{S}\hat{\mathbf{C}}_s = \frac{1}{S}\sum_{s=1}^{S}\mathbf{C}_s + \frac{1}{S}\sum_{s=1}^{S}\mathbf{E}_s.$$

The variance of the estimator $\hat{\mathbf{C}}$ is $S \cdot \frac{\tau_s^2}{S^2} = \frac{\tau_s^2}{S} \triangleq \tau_{\text{ag}}^2$. However, if we had all the samples in one location (centralized or pooled-data scenario), then the (ϵ, δ)-DP approximate of $\mathbf{C}$ can be computed as: $\hat{\mathbf{C}} = \frac{1}{N}\mathbf{Z}\mathbf{Z}^\top + \mathbf{E}$, where the $D \times D$ symmetric matrix $\mathbf{E}$ is generated with entries drawn i.i.d. $\sim \mathscr{N}(0, \tau_c^2)$. In this case, the noise standard deviation is given by $\tau_c = \frac{1}{N\epsilon}\sqrt{2\log(\frac{1.25}{\delta})}$, using the sensitivity of $\mathbf{C}$ in the centralized case [8] as: $\Delta_c = \frac{1}{N}$. For equal number of samples in each site, we have $\tau_c = \frac{\tau_s}{S}$. We observe the ratio: $\frac{\tau_c^2}{\tau_{\text{ag}}^2} = \frac{\tau_s^2/S^2}{\tau_s^2/S} = \frac{1}{S}$. This indicates that the conventional DP distributed scheme will always produce a sub-optimal (more noisy) estimate of the matrix $\hat{\mathbf{C}}$. As computation of the CCA subspaces is closely related to $\hat{\mathbf{C}}$, we can conclude that the conventional distributed DP scheme will have lower utility than that of a pooled data scenario. In the next section, we describe an approach to remedy this and achieve the same performance as the pooled data scenario in the distributed setting.

Algorithm 1 Distributed Differentially-private CCA (capeCCA)

Require: **0**-centered samples $\mathbf{X}_s \in \mathbb{R}^{D_x \times N_s}$ and $\mathbf{Y}_s \in \mathbb{R}^{D_y \times N_s}$ as $\mathbf{Z}_s = \begin{bmatrix}\mathbf{X}_s^\top & \mathbf{Y}_s^\top\end{bmatrix}^\top$ with $\|\mathbf{z}_{s,n}\|_2 \leq 1$ for $s \in [S]$; privacy parameters ϵ, δ; reduced dimension K

Generate $\mathbf{E}_s \in \mathbb{R}^{D \times D}$, as described in text ▷ at noise generator
Generate $\mathbf{F}_s \in \mathbb{R}^{D \times D}$, as described in text ▷ at the aggregator
for $s = 1, 2, \ldots, S$ **do** ▷ at the local sites
  Get $\mathbf{E}_s$ from the noise generator and $\mathbf{F}_s$ from the aggregator
  Generate $D \times D$ symmetric $\mathbf{G}_s$, as described in text
  Compute and send: $\hat{\mathbf{C}}_s \leftarrow \frac{1}{N_s}\mathbf{Z}_s\mathbf{Z}_s^\top + \mathbf{E}_s + \mathbf{F}_s + \mathbf{G}_s$
end for
Compute $\hat{\mathbf{C}} \leftarrow \frac{1}{S}\sum_{s=1}^{S}\left(\hat{\mathbf{C}}_s - \mathbf{F}_s\right)$ ▷ at the aggregator
Extract sub-matrices from $\hat{\mathbf{C}}$ to compute $\hat{\mathbf{U}}^*$ and $\hat{\mathbf{V}}^*$
return $\hat{\mathbf{U}}^*$ and $\hat{\mathbf{V}}^*$

3. PROPOSED DISTRIBUTED DIFFERENTIALLY-PRIVATE CCA

Our proposed method, capeCCA, is described in Algorithm 1 and exploits a particular network structure (see Fig.1(b)). This approach employs a correlated noise design to achieve the same utility of the pooled data case (i.e., $\tau_{\text{ag}} = \tau_c$) in the decentralized setting. We assume that at least two sites are honest and the rest (including the aggregator) are honest-but-curious. The honest-but-curious parties follow the protocol but may collude with an external adversary. All communications are over secure channels and that there is an honest third-party noise generator (as shown in Fig. 1(b)). The aggregator also generates noise to be sent to the sites. Recall that in the pooled

data scenario with no privacy requirements, we have the data matrices $\mathbf{X}$ and $\mathbf{Y}$. The samples are assumed to be the columns of the matrix $\mathbf{Z} = \left[\mathbf{X}^\top \mathbf{Y}^\top\right]^\top$. We can compute $\mathbf{C} = \frac{1}{N}\mathbf{Z}\mathbf{Z}^\top$, extract the sub-matrices $\mathbf{C}_{xx}$, $\mathbf{C}_{xy}$, $\mathbf{C}_{yx}$ and $\mathbf{C}_{yy}$ and compute the optimal CCA subspaces $\mathbf{U}^*$ and $\mathbf{V}^*$. In our distributed setting, we need to add noise to preserve privacy. We design the noise addition procedure in such a way that we can ensure DP for the output from each site and achieve the noise level of the pooled data scenario in the final output from the aggregator. To achieve this, we start with the noise generator. It generates the $D \times D$ matrix $\mathbf{E}_s$ with $[\mathbf{E}_s]_{ij}$ drawn i.i.d. $\sim \mathcal{N}(0, \tau_e^2)$ and $\sum_{s=1}^S \mathbf{E}_s = 0$. The aggregator generates the $D \times D$ matrix $\mathbf{F}_s$ with $[\mathbf{F}_s]_{ij}$ drawn i.i.d. $\sim \mathcal{N}(0, \tau_f^2)$. Finally, the sites generate their own symmetric $D \times D$ matrix $\mathbf{G}_s$, where $[\mathbf{G}_s]_{ij}$ are drawn i.i.d. $\sim \mathcal{N}(0, \tau_g^2)$. At each site s, we compute the sample second-moment matrix $\mathbf{C}_s = \frac{1}{N_s}\mathbf{Z}_s\mathbf{Z}_s^\top$ and release (or send to the central aggregator): $\hat{\mathbf{C}}_s = \mathbf{C}_s + \mathbf{E}_s + \mathbf{F}_s + \mathbf{G}_s$. The noise variances of $\mathbf{E}_s$, $\mathbf{F}_s$, $\mathbf{G}_s$ should ensure that the variance of the noise $\mathbf{F}_s + \mathbf{G}_s$ alone can guarantee (ϵ, δ)-DP to $\mathbf{C}_s$, since the noise terms $\mathbf{E}_s$ are correlated. Additionally, the noise variances should ensure that the variance of $\mathbf{E}_s + \mathbf{G}_s$ is sufficient to guarantee (ϵ, δ)-DP to $\mathbf{C}_s$ – as a safeguard against the untrusted aggregator, which knows $\mathbf{F}_s$ [22]. One approach is to set $\tau_e^2 = \tau_f^2 = \left(1 - \frac{1}{S}\right)\tau_s^2$ and $\tau_g^2 = \frac{1}{S}\tau_s^2$. For a given pair of (ϵ, δ), we can calculate a noise variance τ_s^2 such that adding Gaussian noise of variance τ_s^2 will guarantee (ϵ, δ)-DP. Since there are many (ϵ, δ) pairs that yield the same τ_s^2, we parameterized our method using τ_s^2. Note that the noise generator need not necessarily be a separate entity and can be considered as a common randomness, or a shared coin possessed by the sites [22]. For example, each site could generate $\hat{\mathbf{E}}_s$ and, perhaps using some secure multiparty computation protocol [23], compute $\sum_s \hat{\mathbf{E}}_s$. Each site could then use $\mathbf{E}_s \leftarrow \hat{\mathbf{E}}_s - \frac{1}{S}\sum_s \hat{\mathbf{E}}_s$ to achieve $\sum_s \mathbf{E}_s = 0$. Now, the aggregator computes

$$\hat{\mathbf{C}} = \frac{1}{S}\sum_{s=1}^{S}\left(\hat{\mathbf{C}}_s - \mathbf{F}_s\right) = \frac{1}{S}\sum_{s=1}^{S}\left(\mathbf{C}_s + \mathbf{G}_s\right), \text{ as } \sum_{s=1}^{S}\mathbf{E}_s = 0.$$

At the aggregator, the variance of the estimator is $S \cdot \frac{\tau_g^2}{S^2} = S \cdot \frac{\tau_s^2}{S^3} = \frac{\tau_s^2}{S^2} = \tau_c^2$, which is exactly the same as if all the data were present at the aggregator [22]. Next, we extract the sub-matrices $\hat{\mathbf{C}}_{xx}$, $\hat{\mathbf{C}}_{xy}$, $\hat{\mathbf{C}}_{yx}$ and $\hat{\mathbf{C}}_{yy}$ and compute the (ϵ, δ)-DP CCA subspaces $\hat{\mathbf{U}}^*$ and $\hat{\mathbf{V}}^*$. The privacy guarantee of capeCCA is given in Theorem 1. Note that capeCCA can be readily extended to incorporate unequal privacy requirements/samples sizes at each site [22]. We defer the development of a more robust model with weaker assumptions as a future work.

Theorem 1 (Privacy of capeCCA)**.** *Algorithm 1 computes an* (ϵ, δ)*-DP approximation to the optimal subspaces* $\mathbf{U}^*$ *and* $\mathbf{V}^*$*.*

Proof sketch. The proof of Theorem 1 follows from using the AG algorithm [21], the sensitivity $\Delta_s = \frac{1}{N_s}$ and recalling that the data samples in each site are disjoint. We start by showing: $\tau_e^2 + \tau_g^2 = \tau_g^2 + \tau_f^2 = \tau_s^2 = \left(\frac{1}{N_s\epsilon}\sqrt{2\log\frac{1.25}{\delta}}\right)^2$. Therefore, the computation of $\hat{\mathbf{C}}_s$ at each site is at least (ϵ, δ)-DP. As differential privacy is invariant under post-processing, we can combine the matrices $\{\hat{\mathbf{C}}_s\}$ at the aggregator while subtracting $\mathbf{F}_s$ for each site. We extract the sub-matrices $\hat{\mathbf{C}}_{xx}$, $\hat{\mathbf{C}}_{xy}$, $\hat{\mathbf{C}}_{yx}$ and $\hat{\mathbf{C}}_{yy}$ and compute the subspaces $\hat{\mathbf{U}}^*$ and $\hat{\mathbf{V}}^*$, which are (ϵ, δ)-DP approximates to the true CCA subspaces $\mathbf{U}^*$ and $\mathbf{V}^*$. □

Performance gain with correlated noise. This is the first work that proposes an algorithm for distributed DP CCA. It can be shown that as we employ the correlated noise scheme, the gain in the performance over a conventional distributed DP CCA is atleast S, even when we do not know N_s for $s \in [S]$. Moreover, in case of site drop-out, the performance of capeCCA would fall back to that of the conventional scheme [22]: the output from each site remains (ϵ, δ)-DP, irrespective of the number of dropped-out sites.

Communication cost. capeCCA is an one-shot algorithm. The noise generator and the aggregator both send one $D \times D$ matrix to the sites. Each site uses these to compute the noisy estimate of the $D \times D$ matrix $\hat{\mathbf{C}}_s$ and sends that back to the aggregator. Therefore, the total communication cost is proportional to $3SD^2$ or $O(D^2)$. This is expected as we are computing the global $D \times D$ second-moment matrix in a distributed setting for computing the CCA subspaces.

4. EXPERIMENTAL RESULTS

We consider measuring how well the output subspaces of capeCCA algorithm, $\hat{\mathbf{U}}^*$ and $\hat{\mathbf{V}}^*$, approximate the true subspaces $\mathbf{U}^*$ and $\mathbf{V}^*$ achieved from pooled non-private CCA (non − priv). We have also compared the performance of capeCCA against a conventional (but never proposed before) distributed DP CCA algorithm with no correlated noise (conv) and a centralized DP CCA [8] on local data (local). We used three data sets for our experiments: the MNIST data set [24], the University of Wisconsin X-ray Microbeam data set (XRMB) [25] and a simulated fMRI and EEG data set (fMRI+EEG) [26]. For MNIST, we chose the two views to be the top- and bottom-halves of the images, preprocessed by projecting each view onto 50-dimensional subspaces using PCA. For XRMB, we chose two speakers, JW16 and JW18, with the first view being the pellet coordinates and the second view containing acoustic features including the normalized 13-dimensional mel-frequency cepstral coefficients (MFCCs) and their first and second derivatives [13]. Each view is projected onto a 25-dimensional subspace using PCA [8]. We replicate the sample matrices p times in our experiments. For generating the fMRI+EEG data set we follow [26]. A simulated fMRI-like set of components was generated following [27] using the `simTB` toolbox [28]. For EEG features, an event-related potential (ERP)-like set of components was generated using the `EEGIFT` toolbox [29]. We mixed each set with different sets of modulation profiles to achieve the simulated fMRI and EEG signals. The modulation profiles are orthogonal within each modality and correlated across different modalities. The relation between the fMRI and EEG signals are due to these correlations. We used $K = 5$ simulated components for both fMRI and EEG signals. Each fMRI component is a 50×50 pixel image, whereas each ERP component is a 6360 time-point segment.

For the real data sets, we evaluate the quality of the subspaces produced from the algorithms by using one of the most common applications of CCA: clustering. We employ the popular K-means clustering algorithm on the reduced-dimension samples (achieved by projecting onto the CCA subspaces). We measure the performance of clustering using [8] the Caliński-Harabasz (CH) index [30, 31] for N data points and K clusters:

$$\mathsf{CH} = \frac{\frac{1}{K-1}\sum_{k=1}^{K} N_k \|\mathbf{z}_k - \mathbf{z}\|_2^2}{\frac{1}{N-K}\sum_{k=1}^{K}\sum_{n\in\mathbb{S}_k}\|\mathbf{z}_{nk} - \mathbf{z}_k\|_2^2},$$

where $\mathbf{z}_k$ is the centroid of the k-th cluster, $\mathbf{z}$ is the centroid for all of the samples, N_k denotes the size of cluster k, $\mathbb{S}_k$ is the set of indices of the members of cluster k and $\mathbf{z}_{nk}$ is the n-th point of

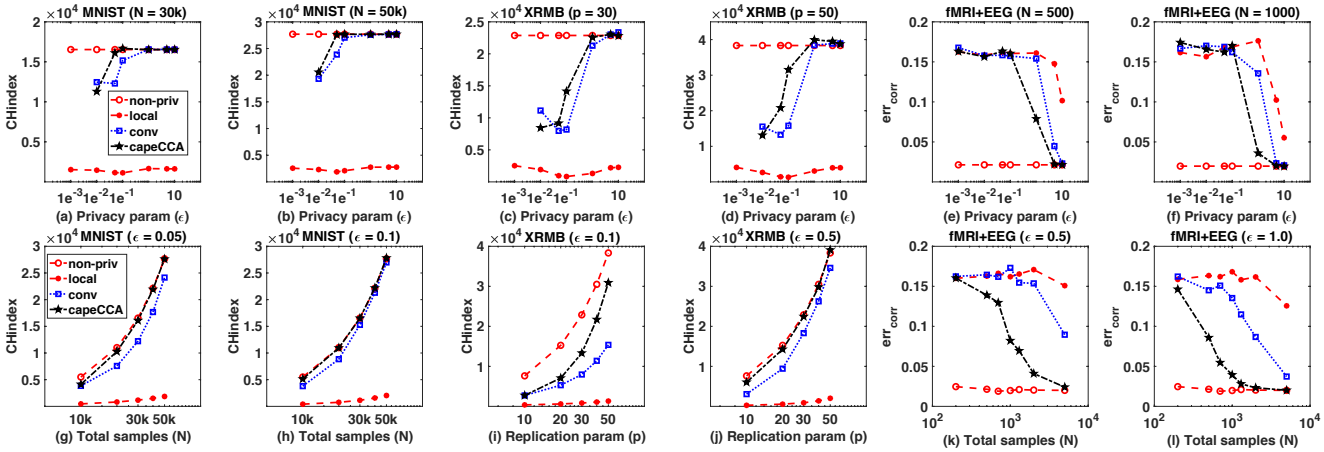

Fig. 2. Variation of performance with privacy parameter ϵ and total samples N. Fixed parameters: $\delta = 0.01$, $S = 10$.

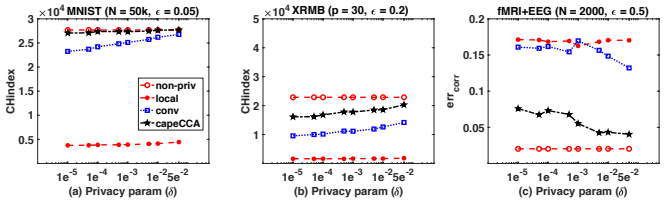

Fig. 3. Variation of performance with δ. Fixed parameter: $S = 10$.

the k-th cluster. For the fMRI+EEG data set, we are interested to see how our algorithm can estimate the correlation between the two modalities. Therefore, we use the following performance index:

$$\text{err}_{\text{corr}} = \frac{1}{K}\|\mathbf{r}^* - \hat{\mathbf{r}}^*\|_2,$$

where $\mathbf{r}^* \in \mathbb{R}^K$ and $\hat{\mathbf{r}}^* \in \mathbb{R}^K$ contain the true correlation scores and the estimated correlation scores between the corresponding modulation profiles of the two modalities. We refer the reader to Correa et al. [26] for more details. In all cases we show the average performance over 10 independent runs of the algorithms.

Privacy-utility trade-offs. First, we explore the *privacy-utility trade-off* between the privacy risk ϵ and the aforementioned performance indices. Recall that the standard deviation of the noise in capeCCA is inversely proportional to both ϵ and N_s. Larger ϵ values indicate higher privacy risk but smaller noise and therefore better utility. We observe this in our experiments as well. In Figure 2(a)-(f), we show how the CH index and err_{corr} vary with ϵ for MNIST, XRMB and fMRI+EEG data sets, while keeping δ and S fixed. For each data set, we show the performance variation with ϵ for two different sample sizes. In all cases, the proposed capeCCA approaches the performance of non $-$ priv as we increase ϵ, outperforming the conv and local. One of the reasons that capeCCA outperforms conv is the smaller noise variance at the aggregator that we can achieve due to the correlated noise scheme. Achieving better performance than local is intuitive because including the information from multiple sites to estimate a population parameter always results in a better performance than using the data from a single site only. For a particular data set, we notice that if we increase the total sample size N (and hence the sample size per site N_s), the performance of capeCCA gets even better. This is expected as the variance of the noise for capeCCA is inversely proportional to square of N_s.

Learning rates and impact of δ. To better understand the impact of sample size, we tested the algorithms on data sets of increasing size. Intuitively, it should be easier to guarantee a smaller privacy risk for the same ϵ and a higher utility (lower error) when the number of samples is large. In Figure 2(g)-(l), we show the performance of capeCCA as a function of total sample size N for synthetic and real data sets with different values of ϵ. Increasing sample size improves the performance of all algorithms. Again, we observe that even for small ϵ values, capeCCA performs nearly as well as non $-$ priv, comfortably outperforming conv and local. For a particular data set, increasing ϵ dictates even better performance. Note that for the XRMB data set, we plotted the CH index vs. sample size plot with the replication parameter p.

Finally, we investigate the variation of performance with δ. Recall that δ can be interpreted as the probability that the privacy-preserving algorithm releases the private information "out in the wild". Therefore, we want δ to be small. However, the smaller the δ is the larger the noise variance becomes, resulting in loss of utility. In Figure 3, we show the performance indices as a function of δ for the synthetic and real data sets. As expected, we observe that increasing δ results in improved performance. However, choosing a $\delta \leq \frac{1}{N}$ may not provide meaningful utility without a sufficiently large ϵ. The proposed capeCCA algorithm achieves similar performance as non $-$ priv for moderate δ values (~ 0.01).

5. CONCLUSION

In this paper, we proposed an algorithm for distributed differentially-private CCA. To our knowledge, this is the first algorithm for privacy-preserving CCA applicable in distributed settings. Our proposed algorithm achieves the same level of privacy-preserving noise variance, and therefore the same level of utility, as the pooled data scenario in a distributed setting. We achieved this by employing a correlated noise design protocol, while assuming the availability of a helper node. We empirically compared the performance of the proposed algorithm with that of the non-private, local and conventional distributed algorithms on synthetic and real data sets with varying privacy and data set parameters. We evaluated the usefulness of the produced subspaces for estimating correlation scores and clustering applications. We observed that the proposed algorithm offered very good utility even for strong privacy guarantees and matched the utility of non-private CCA for some parameter choices. Future work in this direction will be to derive novel utility bounds, as well as validation on higher-dimensional data with more than two modalities.

6. REFERENCES

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. of the Third Conf. on Theory of Cryptography*, 2006, pp. 265–284.

[2] A. D. Sarwate, S. M. Plis, J. A. Turner, M. R. Arbabshirani, and V. D. Calhoun, "Sharing privacy-sensitive access to neuroimaging and genetics data: a review and preliminary validation," *Frontiers in Neuroinformatics*, vol. 8, no. 35, 2014.

[3] S. Plis, A. D. Sarwate, D. Wood, C. Dieringer, D. Landis, C. Reed, S. R. Panta, J. A. Turner, J. M. Shoemaker, K. W. Carter, P. Thompson, K. Hutchison, and V. D. Calhoun, "COINSTAC: A privacy enabled model and prototype for leveraging and processing decentralized brain imaging data," *Frontiers in Neuroscience*, vol. 10, no. 365, 2016.

[4] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[5] M. Borga, "Canonical correlation a tutorial," 1999.

[6] F. Deleus and M. M. Van Hulle, "Functional connectivity analysis of fMRI data based on regularized multiset canonical correlation analysis," *Journal of Neuroscience Methods*, vol. 197, no. 1, pp. 143 – 157, 2011.

[7] Y. O. Li, W. Wang, T. Adali, and V. D. Calhoun, "CCA for joint blind source separation of multiple datasets with application to group fMRI analysis," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2008, pp. 1837–1840.

[8] H. Imtiaz and A. D. Sarwate, "Differentially-private canonical correlation analysis," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov 2017, pp. 283–287.

[9] S. Dasgupta, "Learning mixtures of Gaussians," *Found. of Comp. Sci.*, pp. 634 – 644, 1999.

[10] S. Vempala and G. Wang, "A spectral algorithm for learning mixture models," *J. Comput. Syst. Sci.*, vol. 68, no. 4, pp. 841–860, June 2004.

[11] D. Achlioptas and F. McSherry, "On spectral learning of mixtures of distributions," in *Int. Conf. on Computational Learning Theory*. Springer, 2005, pp. 458–469.

[12] R. Kannan, H. Salmasian, and S. Vempala, "The spectral method for general mixture models," in *Int. Conf. on Computational Learning Theory*. Springer, 2005, pp. 444–457.

[13] R. Arora and K. Livescu, "Multi-view learning with supervision for transformed bottleneck features," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2499–2503.

[14] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. of the 26th Annual Int. Conf. on Machine Learning*. 2009, ICML '09, pp. 129–136, ACM.

[15] Y. O. Li, T. Adali, W. Wang, and V. D. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3918–3929, Oct 2009.

[16] N. Correa, Yi-Ou Li, T. Adali, and V. D. Calhoun, "Examining associations between fMRI and EEG data using canonical correlation analysis," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, May 2008, pp. 1251–1254.

[17] G. H. Golub and H. Zha, "The canonical correlations of matrix pairs and their numerical computation," Tech. Rep., Stanford, CA, USA, 1992.

[18] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.

[19] 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.

[20] 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.

[21] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze Gauss: Optimal bounds for privacy-preserving principal component analysis," in *Proc. of the 46th Annual ACM Symposium on Theory of Computing*, 2014, pp. 11–20.

[22] H. Imtiaz and A. D. Sarwate, "Distributed differentially-private algorithms for matrix and tensor factorization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1449–1464, December 2018.

[23] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2017, CCS '17, pp. 1175–1191, ACM.

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[25] J. R. Westbury, "X-ray microbeam speech production database user's handbook: Madison," *WI: Waisman Center, University of Wisconsin*, 1994.

[26] N. M. Correa, Y. Li, T. Adali, and V. D. Calhoun, "Canonical Correlation Analysis for Feature-Based Fusion of Biomedical Imaging Modalities and Its Application to Detection of Associative Networks in Schizophrenia," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 6, pp. 998–1007, Dec 2008.

[27] 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.

[28] Mind Research Network, "fMRI Simulation Toolbox," .

[29] Mind Research Network, "Group ICA of EEG Toolbox," .

[30] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.

[31] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, Dec 2002.