An Online Riemannian PCA for Stochastic Canonical Correlation Analysis

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present an efficient stochastic algorithm (RSG+) for canonical correlation analysis (CCA) using a reparametrization of the projection matrices. We show how this reparametrization (into structured matrices), simple in hindsight, directly presents an opportunity to repurpose/adjust mature techniques for numerical optimization on Riemannian manifolds. Our developments nicely complement existing methods for this problem which either require $O(d^3)$ time complexity per iteration with $O(\frac{1}{\sqrt{t}})$ convergence rate (where d is the dimensionality) or only extract the top 1 component with $O(\frac{1}{t})$ convergence rate. In contrast, our algorithm offers a strict improvement for this classical problem: it achieves $O(d^2k)$ runtime complexity per iteration for extracting the top k canonical components with $O(\frac{1}{t})$ convergence rate. While the paper primarily focuses on the formulation and technical analysis of its properties, our experiments show that the empirical behavior on common datasets is quite promising. We also explore a potential application in training fair models where the label of protected attribute is missing or otherwise unavailable.

1 Introduction

Canonical correlation analysis (CCA) is a classical method for evaluating correlations between two sets of variables. It is commonly used in unsupervised multi-view learning, where the multiple views of the data may correspond to image, text, audio and so on [32, 10, 24]. Classical formulations have also been extended to leverage advances in representation learning, for example, [3] showed how the CCA can be interfaced with deep neural networks enabling modern use cases. Many results over the last few years have used CCA or its variants for problems including measuring representational similarity in deep neural networks [26], speech recognition [11], and so on.

The goal in CCA is to find linear combinations within two random variables $\mathbf{X}$ and $\mathbf{Y}$ which have maximum correlation with each other. Formally, the CCA problem is defined in the following way. Let $X \in \mathbf{R}^{N\times d_x}$ and $Y \in \mathbf{R}^{N\times d_y}$ be N samples respectively drawn from pair of random variables $\mathbf{X}$ (d_x-variate random variable) and $\mathbf{Y}$ (d_y-variate random variable), with unknown joint probability distribution. The goal is to find the projection matrices $U \in \mathbf{R}^{d_x\times k}$ and $V \in \mathbf{R}^{d_y\times k}$, with $k \leq \min\{d_x, d_y\}$, such that the correlation is maximized:

$$\max_{U,V} \; F = \text{trace}\left(U^T C_{XY} V\right) \quad \text{s.t.} \qquad U^T C_X U = I_k, V^T C_Y V = I_k \tag{1}$$

Here, $C_X = \frac{1}{N}X^T X$ and $C_Y = \frac{1}{N}Y^T Y$ are the sample covariance matrices, and $C_{XY} = \frac{1}{N}X^T Y$ denotes the sample cross-covariance.

The objective function in (1) is the expected cross-correlation in the projected space and the constraints specify that different canonical components should be decorrelated. Let us define the whitened covariance $T \coloneqq C_X^{-1/2} C_{XY} C_Y^{-1/2}$ and Φ_k (and Ψ_k) contains the top-k left (and right) singular

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

vectors of T. It is known [16] that the optimum of (1) is achieved at $U^* = C_X^{-1/2}\Phi_k$, $V^* = C_Y^{-1/2}\Psi_k$. Finally, we compute U^*, V^* by applying a k-truncated SVD to $\widetilde{T}$.

Runtime and memory considerations. The above procedure is simple but is only feasible when the data matrices are small. In most modern applications, not only are the datasets large but also the dimension d (let $d = \max\{d_x, d_y\}$) of each sample can be high, especially if representations are being learned using deep models. As a result, the computational footprint of the algorithm can be high. This has motivated the study of stochastic optimization routines for solving CCA. Observe that in contrast to the typical settings where stochastic schemes are most effective, the CCA objective does not decompose over samples in the dataset. Many efficient strategies have been proposed in the literature: for example, [14, 38] present Empirical Risk Minimization (ERM) models which optimize the empirical objective. More recently, [13, 6, 4] describe proposals that optimize the population objective. To summarize the approaches succinctly, if we are satisfied with identifying the top 1 component of CCA, effective schemes are available by utilizing either extensions of the Oja's rule [28] to the generalized eigenvalue problem [6] or the alternating SVRG algorithm [13]). Otherwise, a stochastic approach must make use of an explicit whitening operation which involves a cost of d^3 for each iteration [4].

Observation. Most approaches either directly optimize (1) or instead a reparameterized or regularized form [14, 2, 4]. Often, the search space for U and V corresponds to the entire $\mathbf{R}^{d\times k}$ (ignoring the constraints for the moment). But if the formulation could be cast in a form which involved approximately writing U and V as a product of structured matrices, we may be able to obtain specialized routines which are tailored to exploit those properties. Such a reformulation is not difficult to derive – where the matrices used to express U and V can be identified as objects that live in well studied geometric spaces. Then, utilizing the geometry of the space and borrowing relevant tools from differential geometry leads to an efficient approximate algorithm for top-k CCA which optimizes the population objective in a streaming fashion.

Contributions. (a) First, we re-parameterize the top-k CCA problem as an optimization problem on specific matrix manifolds, and show that it is equivalent to the original formulation in (1). (b) Informed by the geometry of the manifold, we derive stochastic gradient descent (SGD) algorithms for solving the re-parameterized problem with $O(d^2k)$ cost per iteration and provide convergence rate guarantees. (c) This analysis gives a direct mechanism to obtain an upper bound on the number of iterations needed to guarantee an ϵ error w.r.t. the population objective for the CCA problem. (d) The algorithm works in a streaming manner so it easily scales to large datasets and we do not need to assume access to the full dataset at the outset. (e) We present empirical evidence for both the standard CCA model and the DeepCCA setting [3], describing advantages and limitations.

2 Stochastic CCA: Reformulation, Algorithm and Analysis

Let us recall the objective function for CCA as given in (1). We denote $X \in \mathbf{R}^{N\times d_x}$ as the matrix consisting of the samples $\{\mathbf{x}_i\}$ drawn from a zero mean random variable $\mathbf{X} \sim \mathcal{X}$ and $Y \in \mathbf{R}^{N\times d_y}$ denotes the matrix consisting of samples $\{\mathbf{y}_i\}$ drawn from a zero mean random variable $\mathbf{Y} \sim \mathcal{Y}$. For notational simplicity, we assume that $d_x = d_y = d$ although the results hold for general d_x and d_y. Also recall that C_X, C_Y are the covariance matrices of $\mathbf{X}$, $\mathbf{Y}$. C_{XY} is the cross-covariance matrix between $\mathbf{X}$ and $\mathbf{Y}$. $U \in \mathbf{R}^{d\times k}$ ($V \in \mathbf{R}^{d\times k}$) is the matrix consisting of $\{\mathbf{u}_j\}$ ($\{\mathbf{v}_j\}$) , where $(\{\mathbf{u}_j\}, \{\mathbf{v}_j\})$ are the canonical directions. The constraints in (1) are called *whitening constraints*.

Reformulation: In the CCA formulation, the matrices consisting of canonical correlation directions, i.e., U and V, are unconstrained, hence the search space is the entire $\mathbf{R}^{d\times k}$. Now we reformulate the CCA objective by reparameterizing U and V. In order to do that, let us take a brief detour and recall the objective function of principal component analysis (PCA):

$$\widehat{U} = \underset{U'}{\arg\max} \quad \text{trace}(\widehat{R}) \qquad \text{subject to} \qquad \widehat{R} = U'^T C_X U'; \quad U'^T U' = I_k \tag{2}$$

Observe that by running PCA and assigning $U = \widehat{U}\widehat{R}^{-1/2}$ in (1) (analogous for V using C_Y), we can satisfy the whitening constraint. Of course, writing $U = \widehat{U}\widehat{R}^{-1/2}$ does satisfy the whitening constraint, but such a U (and V) will not maximize trace $\left(U^T C_{XY} V\right)$, objective of (1). Hence, additional work beyond the PCA solution is needed. Let us start from $\widehat{R}$ but relax the PCA solution by using an arbitrary $\widetilde{R}$ instead of diagonal $\widehat{R}$ (this will still satisfy the whitening constraint).

By writing $U = \widetilde{U}\widetilde{R}$ with $\widetilde{U}^T\widetilde{U} = I_k$ and $\widetilde{R} \in \mathbf{R}^{k\times k}$. Thus we can approximate CCA objective (we will later show how good this approximation is) as

$$\max_{\substack{\widetilde{U},\widetilde{V}\in\mathsf{St}(k,d)\\ R_u,R_v\in\mathbf{R}^{k\times k}\\ U=\widetilde{U}R_u;\ V=\widetilde{V}R_v}} \underbrace{\text{trace}\left(U^TC_{XY}V\right)}_{\widetilde{F}} + \underbrace{\text{trace}\left(\widetilde{U}^TC_X\widetilde{U}\right) + \text{trace}\left(\widetilde{V}^TC_Y\widetilde{V}\right)}_{\widetilde{F}_{\text{pca}}} \quad \text{s.t.} \quad \begin{matrix} U^TC_XU=I_k\\ V^TC_YV=I_k\end{matrix} \tag{3}$$

Here, $\mathsf{St}(k,d)$ denotes the manifold consisting of $d\times k$ (with $k \le d$) column orthonormal matrices, i.e., $\mathsf{St}(k,d) = \left\{X \in \mathbf{R}^{d\times k} | X^TX = I_k\right\}$. *Observe that in (3), we approximate the optimal U and V as a linear combination of $\widetilde{U}$ and $\widetilde{V}$ respectively. Thus, the aforementioned PCA solution can act as a feasible initial solution for (3).*

As the choice of R_u and R_v is arbitrary, we can further reparameterize these matrices by constraining them to be full rank (of rank k) and using the RQ decomposition [15] which gives us the following reformulation.

A Reformulation for CCA

$$\max_{\substack{\widetilde{U},\widetilde{V},S_u,S_v,Q_u,Q_v\\ U=\widetilde{U}S_uQ_u;\ V=\widetilde{V}S_vQ_v}} \underbrace{\text{trace}\left(U^TC_{XY}V\right)}_{\widetilde{F}} + \underbrace{\text{trace}\left(\widetilde{U}^TC_X\widetilde{U}\right) + \text{trace}\left(\widetilde{V}^TC_Y\widetilde{V}\right)}_{\widetilde{F}_{\text{pca}}} \tag{4a}$$

$$\begin{aligned} \text{subject to} \quad & U^TC_XU = I_k \\ & V^TC_YV = I_k \\ & \widetilde{U},\widetilde{V} \in \mathsf{St}(k,d);\ Q_u,Q_v \in \mathsf{SO}(k) \\ & S_u, S_v \text{ is upper triangular} \end{aligned} \tag{4b}$$

Here, $\mathsf{SO}(k)$ is the space of $k\times k$ special orthogonal matrices, i.e., $\mathsf{SO}(k) = \left\{X \in \mathbf{R}^{k\times k} | X^TX = I_k; \det(X) = 1\right\}$. Before stating formally how good the aforementioned approximation is, we first point out some interesting properties of the reformulation (4): **(a)** in the reparametrization of U and V all components are structured, hence, the search space becomes a subset of $\mathbf{R}^{k\times k}$ **(b)** we can essentially initialize with a PCA solution and then optimize (4).

Why (4) helps? First, we note that CCA seeks to maximize the total correlation under the constraint that different components are decorrelated. The difficult part in the optimization is to ensure decorrelation, which leads to a higher complexity in existing streaming CCA algorithms. On the contrary, in (4), we separate (1) into finding the PCs, $\widetilde{U},\widetilde{V}$ (by adding the variance maximization terms) and finding the linear combination (S_uQ_u and S_vQ_v) of the principal directions. Thus, here we can (almost) utilize an efficient off-the-shelf streaming PCA algorithm. We will defer describing the specific details of the optimization itself until the next sub-section. First, we will show formally why substituting (1) with (4) is sensible under some assumptions.

Why the solution of the reformulation makes sense? We start by stating some mild assumptions needed for the analysis. **Assumptions: (a)** The random variables $\mathbf{X} \sim \mathcal{N}(\mathbf{0},\Sigma_x)$ and $\mathbf{Y} \sim \mathcal{N}(\mathbf{0},\Sigma_y)$ with $\Sigma_x \preceq cI_d$ and $\Sigma_y \preceq cI_d$ for some $c > 0$. **(b)** The samples X and Y drawn from $\mathcal{X}$ and $\mathcal{Y}$ respectively have zero mean. **(c)** For a given $k \le d$, Σ_x, Σ_y have non-zero top-k eigen values.

We show how the presented solution, assuming access to an effective numerical procedure, approximates the CCA problem presented in (1). We formally state the result in the following theorem with a sketch of proof (appendix includes the full proof) by first stating the following proposition.

Definition 1. *A random variable $\mathbf{X}$ is called sub-Gaussian if the norm given by $\|\mathbf{X}\|_\star := \inf\left\{d \ge 0 | \mathbf{E}_{\mathbf{X}}\left[\exp\left(\text{trace}(X^TX)/d^2\right)\right] \le 2\right\}$ is finite. Let $U \in \mathbf{R}^{d\times k}$, then $\mathbf{X}U$ is sub-Gaussian [35].*

Proposition 1 ([31])**.** *Let $\mathbf{X}$ be a random variable which follows a sub-Gaussian distribution. Let $\widehat{X}$ be the approximation of $X \in \mathbf{R}^{N\times d}$ (samples drawn from $\mathcal{X}$) with the top-k principal vectors. Let $\widetilde{C}_X$ be the covariance of $\widehat{X}$. Also, assume that λ_i is the ith eigen value of C_X for $i = 1,\cdots,d-1$ and $\lambda_i \ge \lambda_{i+1}$ for all i. Then, the PCA reconstruction error, denoted by $\mathcal{E}_k = \|X - \widehat{X}\|$ (in the*

Frobenius norm sense) can be upper bounded as follows

$$\mathcal{E}_k \leq \min\left(\sqrt{2k}\|\Delta\|_2, \frac{2\|\Delta\|_2^2}{\lambda_k - \lambda_{k+1}}\right), \quad \Delta = C_X - \widetilde{C}_X.$$

The aforementioned proposition suggests that the error between the data matrix X and the reconstructed data matrix $\widehat{X}$ using the top-k principal vectors is bounded.

Recall from (1) and (4) that the optimal value of the true and approximated CCA objective is denoted by F and $\widetilde{F}$ respectively. The following theorem states that we can bound the error, $E = \|F - \widetilde{F}\|$ (proof in the appendix). In other words, if we start from PCA solution and can successfully optimize (4) without leaving the feasible set, we will obtain a good solution.

Theorem 1. *Using the hypothesis and assumptions above, the approximation error $E = \|F - \widetilde{F}\|$ is bounded and goes to zero while the whitening constraints in equation 4b are satisfied.*

Sketch of the Proof. Let U^* and V^* be the true solution of CCA, i.e., of (1). Let $U = \widetilde{U}S_uQ_u, V = \widetilde{V}S_vQ_v$ be the solution of (4), with $\widetilde{U}, \widetilde{V}$ be the PCA solutions of X and Y respectively. Let $\widehat{X} = X\widetilde{U}\widetilde{U}^T$ and $\widehat{Y} = Y\widetilde{V}\widetilde{V}^T$ be the reconstruction of X and Y using principal vectors. Let $S_uQ_u = \widetilde{U}^TU^*$ and $S_vQ_v = \widetilde{V}^TV^*$. Then we can write $\widetilde{F} = \text{trace}\left(U^TC_{XY}V\right)$ $= \text{trace}\left(\frac{1}{N}\left(\widehat{X}U^*\right)^T\widehat{Y}V^*\right)$. Similarly we can write $F = \text{trace}\left(\frac{1}{N}\left(XU^*\right)^TYV^*\right)$. As, $\widehat{X}$ and $\widehat{Y}$ are the approximation of X and Y respectively using principal vectors, we use proposition 1 to bound the error $\|F - \widetilde{F}\|$. Now observe that $\widehat{X}U$ can be rewritten into $X\widetilde{U}\widetilde{U}^TU$ (similar for $\widehat{Y}V$). Thus, as long as the solution S_uQ_u and S_vQ_v respectively well-approximate $\widetilde{U}^TU$ and $\widetilde{V}^TV$, $\widetilde{F}$ is a good approximation of F. □

Now, the only unresolved issue is an optimization scheme for equation 4a that keeps the constraints in equation 4b satisfied by leveraging the geometry of the structured solution space.

2.1 How to numerically optimize (4a) satisfying constraints in (4b)?

Overview. We now describe how to maximize the formulation in (4a)–(4b) with respect to $\widetilde{U}$, $\widetilde{V}$, Q_u, Q_v, S_u and S_v. We will first compute top-k principal vectors to get $\widetilde{U}$ and $\widetilde{V}$. Then, we will use a gradient update rule to solve for Q_u, Q_v, S_u and S_v to improve the objective. Since all these matrices are "structured", care must be taken to ensure that the matrices *remain on their respective manifolds* – which is where the geometry of the manifolds will offer desirable properties. We re-purpose a Riemannian stochastic gradient descent (RSGD) to do this, so call our algorithm *RSG+*. Of course, more sophisticated Riemannian optimization techniques can be substituted in. For instance, different Riemannian optimization methods are available in [1] and optimization schemes for many manifolds are offered in PyManOpt [8].

The algorithm block is presented in Algorithm 1. Recall $\widetilde{F}_{\text{pca}} = \text{trace}\left(U^TC_XU\right) + \text{trace}\left(V^TC_YV\right)$ be the contribution from the principal directions which we used to ensure the "whitening constraint". Moreover, $\widetilde{F} = \text{trace}\left(U^TC_{XY}V\right)$ be the contribution from the canonical correlation directions (*note that we use the subscript 'cca' for making CCA objective explicit*). The algorithm consists of four main blocks denoted by different colors, namely **(a)** the Red block deals with gradient calculation of the objective function where we calculate the top-k principal vectors (denoted by $\widetilde{F}_{\text{pca}}$) with respect to $\widetilde{U}$, $\widetilde{V}$; **(b)** the Green block describes calculation of the gradient corresponding to the canonical directions (denoted by $\widetilde{F}$) with respect to $\widetilde{U}$, $\widetilde{V}$, S_u, S_v, Q_u and Q_v; **(c)** the Gray block combines the gradient computation from both $\widetilde{F}_{\text{pca}}$ and $\widetilde{F}$ with respect to unknowns $\widetilde{U}$, $\widetilde{V}$, S_u, S_v, Q_u and Q_v; and finally **(d)** the Blue block performs a batch update of the canonical directions $\widetilde{F}$ using Riemannian gradient updates.

Gradient calculations. The gradient update for $\widetilde{U}, \widetilde{V}$ is divided into two parts **(a)** The (Red block) gradient updates the "principal" directions (denoted by $\nabla_{\widetilde{U}}\widetilde{F}_{\text{pca}}$ and $\nabla_{\widetilde{V}}\widetilde{F}_{\text{pca}}$), which is specifically designed to satisfy the *whitening constraint*. Since this requires updating the principal subspaces,

so, the gradient descent needs to proceed on the manifold of k-dimensional subspaces of $\mathbf{R}^d$, i.e., on the Grassmannian $\mathsf{Gr}(k, d)$. **(b)** The (green block) gradient from the objective function in (4), is denoted by $\nabla_{\widetilde{U}}\widetilde{F}$ and $\nabla_{\widetilde{V}}\widetilde{F}$. In order to ensure that the Riemannian gradient update for $\widetilde{U}$ and $\widetilde{V}$ stays on the manifold $\mathsf{St}(k, d)$, we need to make sure that the gradients, i.e., $\nabla_{\widetilde{U}}\widetilde{F}$ and $\nabla_{\widetilde{V}}\widetilde{F}$ lies in the tangent space of $\mathsf{St}(k, d)$. In order to do that, we need to first calculate the Euclidean gradient and then project on to the tangent space of $\mathsf{St}(k, d)$.

The gradient updates for Q_u, Q_v, S_u, S_v are given in the green block, denoted by $\nabla_{Q_u}\widetilde{F}$, $\nabla_{Q_v}\widetilde{F}$, $\nabla_{S_u}\widetilde{F}$ and $\nabla_{S_v}\widetilde{F}$. Note that unlike the previous step, this gradient only has components from canonical correlation computation. As before, this step requires first computing the Euclidean gradient and then projecting on to the tangent space of the underlying Riemannian manifolds involved, i.e., $\mathsf{SO}(k)$ and the space of upper triangular matrices.

Finally, we get the gradient to update the canonical directions by combining the gradients which is shown in gray block. With these gradients we can perform a batch update as shown in the blue block. A schematic diagram is given in Fig. 1.

Using convergence results presented next in Propositions 2–3, this scheme can be shown (under some assumptions) to approximately optimize the CCA objective in (1).

We can now move to the convergence properties of the algorithm. We present two results stating the asymptotic proof of convergence for top-k principal vectors and canonical directions in the algorithm.

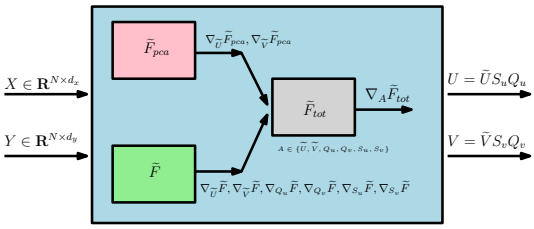


Figure 1: Schematic diagram of the proposed CCA algorithm, here $\widetilde{F}_{\text{tot}} = \widetilde{F} + \widetilde{F}_{\text{pca}}$, where $\widetilde{F}$ is the approximated objective value for CCA (as in (4))

Proposition 2 ([9]). *(Asymptotically) If the samples, X, are drawn from a Gaussian distribution, then the gradient update rule presented in Step 5 in Algorithm 1 returns an orthonormal basis – the top-k principal vectors of the covariance matrix C_X.*

Proposition 3. *([7]) Consider a connected Riemannian manifold $\mathcal{M}$ with injectivity radius bounded from below by $I > 0$. Assume that the sequence of step sizes (γ_l) satisfy the condition (a) $\sum \gamma_l^2 < \infty$ (b) $\sum \gamma_l = \infty$. Suppose $\{A_l\}$ lie in a compact set $K \subset \mathcal{M}$. We also suppose that $\exists D > 0$ such that, $g_{A_l}\left(\nabla_{A_l}\widetilde{F}, \nabla_{A_l}\widetilde{F}\right) \leq D$. Then $\nabla_{A_l}\widetilde{F} \to 0$ and $l \to \infty$.*

Notice that in our problem, the injectivity radius bound in Proposition 3 is satisfied as "I" for $\mathsf{Gr}(p, n)$, $\mathsf{St}(p, n)$ or $\mathsf{SO}(p)$ is $\pi/2\sqrt{2}, \pi/2\sqrt{2}, \pi/2$ respectively. So, in order to apply Proposition 3, we need to guarantee the step sizes satisfy the aforementioned condition. One example of the step sizes that satisfies the property is $\gamma_l = \frac{1}{l+1}$.

2.2 Convergence rate and complexity of the RSG+ algorithm

In this section, we describe the convergence rate and complexity of the algorithm proposed in Algorithm 1. Observe that the key component of Algorithm 1 is a Riemannian gradient update. Let A_t be the generic entity needed to be updated in the algorithm using the Riemannian gradient update $A_{t+1} = \mathsf{Exp}_{A_t}\left(-\gamma_t \nabla_{A_t}\widetilde{F}\right)$, where γ_t is the step size at time step t. Also assume $\{A_t\} \subset \mathcal{M}$ for a Riemannian manifold $\mathcal{M}$. The following proposition states that under certain assumptions, the Riemannian gradient update has a convergence rate of $O\left(\frac{1}{t}\right)$.

Proposition 4. *([27, 5]) Let $\{A_t\}$ lie inside a geodesic ball of radius less than the minimum of the injectivity radius and the strong convexity radius of $\mathcal{M}$. Assume $\mathcal{M}$ to be a geodesically complete Riemannian manifold with sectional curvature lower bounded by $\kappa \leq 0$. Moreover, assume that the step size $\{\gamma_t\}$ diverges and the squared step size converges. Then, the Riemannian gradient descent update given by $A_{t+1} = \mathsf{Exp}_{A_t}\left(-\gamma_t \nabla_{A_t}\widetilde{F}\right)$ with a bounded $\nabla_{A_t}\widetilde{F}$, i.e., $\|\nabla_{A_t}\widetilde{F}\| \leq C < \infty$ for some $C \geq 0$, converges in the rate of $O\left(\frac{1}{t}\right)$.*

All Riemannian manifolds we used, i.e., $\mathsf{Gr}(k, d)$, $\mathsf{St}(k, d)$ and $\mathsf{SO}(k)$ are geodesically complete, and these manifolds have non-negative sectional curvatures, i.e., lower bounded by $\kappa = 0$. Moreover

Algorithm 1: Riemannian SGD based algorithm (RSG+) to compute canonical directions

Input: $X \in \mathbf{R}^{N \times d_x}, Y \in \mathbf{R}^{N \times d_y}, k > 0$

Output: $U \in \mathbf{R}^{d_x \times k}, V \in \mathbf{R}^{d_y \times k}$

Initialize $\widetilde{U}, \widetilde{V}, Q_u, Q_v, S_u, S_v$

Partition data X, Y into batches of size B. Let j^{th} batch be denoted by X_j and Y_j

for $j \in \left\{1, \cdots, \lfloor \frac{N}{B} \rfloor\right\}$ **do**

Gradient for top-k principal vectors: calculating $\nabla_{\widetilde{U}} \widetilde{F}_{\text{pca}}, \nabla_{\widetilde{V}} \widetilde{F}_{\text{pca}}$

1. Partition X_j (Y_j) into L ($L = \lfloor \frac{B}{k} \rfloor$) blocks of size $d_x \times k$ ($d_y \times k$);
2. Let the lth block be denoted by Z_l^x (Z_l^y);
3. Orthogonalize each block and let the orthogonalized block be denoted by $\hat{Z}_l^x$ ($\hat{Z}_l^y$);
4. Let the subspace spanned by each $\hat{Z}_l^x$ (and $\hat{Z}_l^y$) be $\hat{\mathcal{Z}}_l^x \in \mathsf{Gr}(k, d_x)$ (and $\hat{\mathcal{Z}}_l^y \in \mathsf{Gr}(k, d_y)$);

$$\nabla_{\widetilde{U}} \widetilde{F}_{\text{pca}} = -\sum_l \mathsf{Exp}_{\widetilde{U}}^{-1}\left(\hat{\mathcal{Z}}_l^x\right) \quad \nabla_{\widetilde{V}} \widetilde{F}_{\text{pca}} = -\sum_l \mathsf{Exp}_{\widetilde{V}}^{-1}\left(\hat{\mathcal{Z}}_l^y\right) \tag{5}$$

Gradient from equation 4: calculating $\nabla_{\widetilde{U}} \widetilde{F}, \nabla_{\widetilde{V}} \widetilde{F}, \nabla_{Q_u} \widetilde{F}, \nabla_{Q_v} \widetilde{F}, \nabla_{S_u} \widetilde{F}, \nabla_{S_v} \widetilde{F}$

$\nabla_{\widetilde{U}} \widetilde{F} = \frac{\partial \widetilde{F}}{\partial \widetilde{U}} - \widetilde{U} \frac{\partial \widetilde{F}}{\partial \widetilde{U}}^T \widetilde{U} \quad \nabla_{\widetilde{V}} \widetilde{F} = \frac{\partial \widetilde{F}}{\partial \widetilde{V}} - \widetilde{V} \frac{\partial \widetilde{F}}{\partial \widetilde{V}}^T \widetilde{V}$

$\nabla_{Q_u} \widetilde{F} = \frac{\partial \widetilde{F}}{\partial Q_u} - \frac{\partial \widetilde{F}}{\partial Q_u}^T \quad \nabla_{Q_v} \widetilde{F} = \frac{\partial \widetilde{F}}{\partial Q_v} - \frac{\partial \widetilde{F}}{\partial Q_v}^T$

$\nabla_{S_u} \widetilde{F} = \text{Upper}\left(\frac{\partial \widetilde{F}}{\partial S_u}\right) \quad \nabla_{S_v} \widetilde{F} = \text{Upper}\left(\frac{\partial \widetilde{F}}{\partial S_v}\right)$

Here, Upper returns the upper triangular matrix of the input matrix and $\frac{\partial \widetilde{F}}{\partial \widetilde{U}}, \frac{\partial \widetilde{F}}{\partial \widetilde{V}}, \frac{\partial \widetilde{F}}{\partial Q_u}, \frac{\partial \widetilde{F}}{\partial Q_v}, \frac{\partial \widetilde{F}}{\partial S_u}, \frac{\partial \widetilde{F}}{\partial S_v}$ give the Euclidean gradients, which are provided in supplement.

Gradient to update canonical directions

$\nabla_{\widetilde{U}} \widetilde{F}_{\text{tot}} = \nabla_{\widetilde{U}} \widetilde{F}_{\text{pca}} + \nabla_{\widetilde{U}} \widetilde{F} \qquad \nabla_{\widetilde{V}} \widetilde{F}_{\text{tot}} = \nabla_{\widetilde{V}} \widetilde{F}_{\text{pca}} + \nabla_{\widetilde{V}} \widetilde{F}$;

$\nabla_X \widetilde{F}_{\text{tot}} = \nabla_X \widetilde{F}$ where, X is a generic entity: $X \in \{Q_u, Q_v, S_u, S_v\}$;

Batch update of canonical directions

$A = \mathsf{Exp}_A\left(-\gamma_j \nabla_A \widetilde{F}_{\text{tot}}\right)$ where, A is a generic entity: $A \in \{\widetilde{U}, \widetilde{V}, Q_u, Q_v, S_u, S_v\}$;

end for

$U = \widetilde{U} Q_u S_u$ and $V = \widetilde{V} Q_v S_v$;

the minimum of convexity and injectivity radius for $\mathsf{Gr}(k, d)$, $\mathsf{St}(k, d)$ and $\mathsf{SO}(k)$ are $\pi/2\sqrt{2}$. Now, as long as the Riemannian updates lie inside the geodesic ball of radius less than $\pi/2\sqrt{2}$, the convergence rate for RGD applies in our setting.

Running time. To evaluate time complexity, we must look at the main compute-heavy steps needed. The basic modules are Exp and Exp^{-1} maps for $\mathsf{St}(k, d)$, $\mathsf{Gr}(k, d)$ and $\mathsf{SO}(k)$ manifolds (see supplement). Observe that the complexity of these modules is influenced by the complexity of svd needed for the Exp map for the St and Gr manifolds. Our algorithm involves structured matrices of size $d \times k$ and $k \times k$, so any matrix operation should not exceed a cost of $O(\max(d^2 k, k^3))$, since in general $d \gg k$. Specifically, the most expensive calculation is SVD of matrices of size $d \times k$, which is $O(d^2 k)$, see [15]. All other calculations are dominated by this term.

3 Experiments

We first evaluate RSG+ for extracting top-k canonical components on three benchmark datasets and show that it performs favorably compared with [4]. Then, we show that RSG+ also fits into feature learning in DeepCCA [3], and can scale to large feature dimensions where the non-stochastic method fails. Finally we show that RSG+ can be used to improve fairness of deep neural networks without full access to labels of protected attributes during training.

3.1 CCA on Fixed Datasets

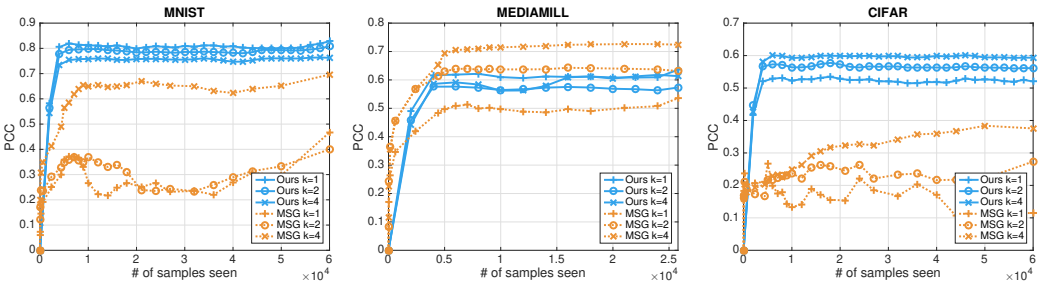
Datasets and baseline. We conduct experiments on three benchmark datasets (MNIST [22], Mediamill [33] and CIFAR-10 [21]) to evaluate the performance of RSG+ to extract top-k canonical components. To our best knowledge, [4] is the only previous work which stochastically optimizes the

population objective in a streaming fashion and can extract top-k components, so we compare our RSG+ with the matrix stochastic gradient (MSG) method proposed in [4] (There are two methods proposed in [4] and we choose MSG because it performs better in the experiments in [4]). The details regarding the three datasets and how we process them are as follows:

MNIST [22]: MNIST contains grey-scale images of size 28×28. We use its full training set containing 60K images. Every image is split into left/right half, which are used as the two views. **Mediamill** [33]: Mediamill contains around 25.8K paired features of videos and corresponding commentary of dimension $120, 101$ respectively. **CIFAR-10** [21]: CIFAR-10 contains 60K 32×32 color images. Like MNIST, we split the images into left/right half and use them as two views.

Evaluation metric. We choose to use Proportion of Correlations Captured (PCC) which is widely used [25, 14], partly due to its efficiency, especially for relatively large datasets. Let $\hat{U} \in R^{d_x \times k}, \hat{V} \in R^{d_y \times k}$ denote the estimated subspaces returned by RSG+, and $U^* \in R^{d_x \times k}, V^* \in R^{d_y \times k}$ denote the true canonical subspaces (all for top-k). The PCC is defined as PCC $= \frac{\text{TCC}(X\hat{U}, Y\hat{V})}{\text{TCC}(XU^*, YV^*)}$, where TCC is the sum of canonical correlations between two matrices.

Performance. The performance in terms of PCC as a function of # of seen samples (coming in a streaming way) are shown in Fig. 2, and our RSG+ achieves around 10 times runtime improvement from MSG (see supplement for the table). Our RSG+ captures more correlation than MSG [4] while being $5 - 10$ times faster. One case where our RSG+ underperforms [4] is when the top-k eigenvalues are dominated by the top-l eigenvalues with $l < k$ (Fig. 2b): on Mediamill dataset, the top-4 eigenvalues of the covariance matrix in view 1 are: $8.61, 2.99, 1.15, 0.37$. The first eigenvalue is dominantly large compared with the rest and our RSG+ performs better for $k = 1$ and worse than [4] for $k = 2, 4$. We also provide the runtime of RSG+ under different data dimension (set $d_x = d_y = d$) and number of total samples sampled from joint gaussian distribution in supplement.



(a) on MNIST (b) on Mediamill (c) on CIFAR

Figure 2: Performance on three datasets in terms of PCC as a function of # of seen samples.

3.2 CCA for Deep Feature Learning

Background and motivation. A deep neural network (DNN) extension of CCA was proposed by [3] and has become popular in multi-view representation learning tasks. The idea is to learn a deep neural network as the mapping from original data space to a latent space where the canonical correlations are maximized. We refer the reader to [3] for details of the task. Since deep neural networks are usually trained using SGD on mini-batches, this requires getting an estimate of the CCA objective at every iteration in a streaming fashion, thus our RSG+ can be a natural fit. We conduct experiments on a noisy version of MNIST dataset to evaluate RSG+.

Dataset. We follow [36] to construct a noisy version of MNIST: View 1 is a randomly sampled image which is first rescaled to $[0, 1]$ and then rotated by a random angle from $[-\frac{\pi}{4}, \frac{\pi}{4}]$. View 2 is randomly sampled from the same class as view 1. Then we add independent uniform noise from $[0, 1]$ to each pixel. Finally the image is truncated into $[0, 1]$ to form the view 2.

Table 1: Results of feature learning on MNIST. N/A means fails to yield a result on our hardware.

Accuracy(%)	$d = 100$	$d = 500$	$d = 1000$
DeepCCA	80.57	N/A	N/A
Ours	79.79	84.09	86.39

Implementation details. We use a simple 2-layer MLP with ReLU nonlinearity, where the hidden dimension in the middle is 512 and the output feature dimension is $d \in \{100, 500, 1000\}$. After the

network is trained on the CCA objective, we use a linear Support Vector Machine (SVM) to measure classification accuracy on output latent features. [3] uses the closed form CCA objective on the current batch directly, which costs $O(d^3)$ memory and time for every iteration.

Performance. Table 1 shows that we get similar performance when $d = 100$ and can scale to large latent dimensions $d = 1000$ while the batch method [3] encounters numerical difficulty on our GPU resources and the Pytorch [29] platform in performing an eigen-decomposition of a $d \times d$ matrix when $d = 500$, and becomes difficult if d is larger than 1000.

3.3 CCA for Fairness Applications

Background and motivation. Fairness is becoming an important issue to consider in the design of learning algorithms. A common strategy to make an algorithm fair is to remove the influence of one/more protected attributes when training the models, see [23]. Most methods assume that the labels of protected attributes are known during training but this may not always be possible. CCA enables considering a slightly different setting, where we may not have per-sample protected attributes which may be sensitive or hard to obtain for third-parties [30]. On the other hand, we assume that a model pre-trained to predict the protected attribute labels is provided. For example, if the protected attribute is gender, we only assume that a good classifier which is trained to predict gender from the samples is available rather than sample-wise gender values themselves. We next demonstrate that fairness of the model, using standard measures, can be improved via constraints on correlation values from CCA.

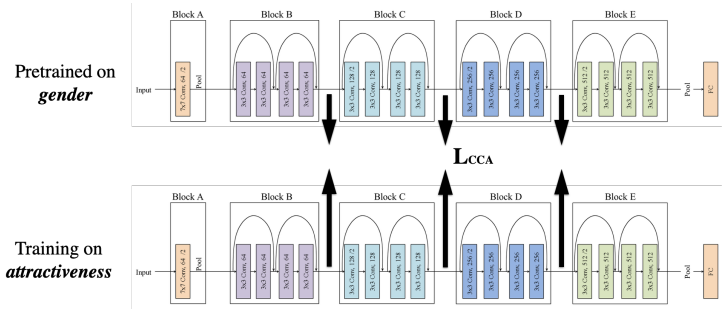


Figure 3: Training architecture for fairness experiment. The model above is the pretrained model and the model below is being trained. Use of CCA allows the two network architectures to be different.

Dataset. CelebA [37] consists of 200K celebrity face images from the internet. There are up to 40 labels, each of which is binary-valued. Here, we follow [23] to focus on the *attactiveness* attribute (which we want to train a classifier to predict) and the *gender* is treated as "protected" since it may lead to an unfair classifier according to [23].

Method. Our strategy is inspired by [26] which showed that canonical correlations can reveal the similarity in neural networks: when two networks (same architecture) are trained using different labels/schemes for example, canonical correlations can indicate how similar their features are. Our observation is the following. Consider a classifier that is trained on gender (the protected attribute), and another classifier that is trained on *attractiveness*, if the features extracted by the latter model share a high similarity with the one trained to predict gender, then it is more likely that the latter model is influenced by features in the image pertinent to gender, which will lead to an unfairly biased trained model. We show that by imposing a loss on the canonical correlation between the network being trained (but we lack per-sample protected attribute information) and a well trained classifier pre-trained on the protected attributes, we can obtain a more fair model. This may enable training fairer models in settings which would otherwise be difficult. The training architecture is shown in Fig. 3.

Table 2: Fairness results on CelebA. We applied CCA on three different layers in Resnet-18 respectively. See supplement for positions of conv $0, 1, 2$. "Ours-conv[0,1]-conv[1,2]" means stacking features from different layers to form hypercolumn features [18], which shows that our approach allows two networks to have different shape/size.

	Accuracy(%)	DEO(%)	DDP(%)
Unconstrained	76.3	22.3	4.8
Ours-conv0	76.5	17.4	**1.4**
Ours-conv1	**77.7**	**15.3**	3.2
Ours-conv2	75.9	22.0	2.8
Ours-conv[0,1]-conv[1,2]	76.0	22.1	3.9

Implementation details. To simulate the case where we only have a pretrained network on protected attributes, we train a Resnet-18 [19] on *gender* attribute, and when we train the classifier to predict *attractiveness*, we add a loss using the canonical correlations between these two networks on interme-

diate layers: $L_{\text{total}} = L_{\text{cross-entropy}} + L_{\text{CCA}}$ where the first term is the standard cross entropy term and the second term is the canonical correlation. See supplement for more details of training/evaluation.

Results. We choose two commonly used error metrics for fairness: difference in Equality of Opportunity [17] (DEO), and difference in Demographic Parity [39] (DDP). We conduct experiments by applying the canonical correlation loss on three different layers in Resnet-18. In Table 2, we can see that applying canonical correlation loss generally improves the DEO and DDP metrics (lower is better) over the standard model (trained using cross entropy loss only). Specifically, applying the loss on early layers like conv0 and conv1 gets better performance than applying at a relatively late layer like conv2. Another promising aspect of our approach is that is can easily handle the case where the protected attribute is a continuous variable (as long as a well trained regression network on the protected attribute is given) while other methods like [23, 41] need to first discretize the variable and then enforce constraints which can be much more involved.

Limitations. Our current implementation has difficulty to scale beyond $d = 10^5$ data dimension and this may be desirable for large scale DNNs. Exploring the sparsity may be one way to solve the problem and will be enabled by additional developments in modern toolboxes.

4 Related Work

Stochastic CCA: There has been much interest in designing scalable and provable algorithms for CCA: [25] proposed the first stochastic algorithm for CCA, while only local convergence is proven for non-stochastic version. [38] designed algorithm which uses alternating SVRG combined with shift-and-invert pre-conditioning, with global convergence. These stochastic methods, together with [14] [2], which reduce CCA problem to generalized eigenvalue problem and solve it by performing efficient power method, all belongs to the methods that try to solve empirical CCA problem, it can be seen as ERM approxiamtion of the original population objective, which requires solving numerical optimization of the empirical CCA objective on a fixed data set. These methods usually assume the access to the full dataset in the beginning, which is not very suitable for many practical applications where data tend to come in a streaming way. Recently, there are increasingly interest in considering population CCA problem [4] [13]. The main difficulty in population setting is we have limited knowledge about the objective unless we know the distribution of $\mathbf{X}$ and $\mathbf{Y}$. [4] handles this problem by deriving an estimation of gradient of population objecitve whose error can be properly bounded so that applying proximal gradient to a convex relexed objective will provably converge. [13] provides tightened analysis of the time complexity of the algorithm in [38], and provides sample complexity under certain distribution. The problem we are trying to solve in this work is the same as that in [4, 13]: to optimize the population objective of CCA in a streaming fashion.

Riemannian Optimization: Riemannian optimization is a generalization of standard Euclidean optimization methods to smooth manifolds, which takes the following form: Given $f : \mathcal{M} \to \mathbf{R}$, solve $\min_{x \in \mathcal{M}} f(x)$, where $\mathcal{M}$ is a Riemannian manifold. One advantage is that it provides a nice way to express many constrained optimization problems as unconstrained problems. Applications include matrix and tensor factorization [20], [34], PCA [12], CCA [40], and so on. [40] rewrites CCA formulation as Riemannian optimization on Stiefel manifold. In our work, we further explore the ability of Riemannian optimization framework, decomposing the linear space spanned by canonical vectors into products of several matrices which lie in several different Riemannian manifolds.

5 Conclusions

In this work, we presented a stochastic approach (RSG+) for the CCA model based on the observation that the solution of CCA can be decomposed into a product of matrices which lie on certain structured spaces. This affords specialized numerical schemes and makes the optimization more efficient. The optimization is based on Riemannian stochastic gradient descent and we provide a proof for its $O(\frac{1}{t})$ convergence rate with $O(d^2k)$ time complexity per iteration. In experimental evaluations, we find that our RSG+ behaves favorably relative to the baseline stochastic CCA method in capturing the correlation in the datasets. We also show the use of RSG+ in the DeepCCA setting showing feasibility when scaling to large dimensions as well as in an interesting use case in training fair models.

Potential negative societal impacts. Since CCA is a fundamental problem and not tied to specific applications, we do not see a negative societal impact of our proposed method. However, it is possible that CCA can be used to uncover relationships between measurements which can be used for discriminative purposes.

References

[1] P.-A. Absil, R. E. Mahony, and R. Sepulchre. Optimization algorithms on matrix manifolds. 2007.

[2] Z. Allen-Zhu and Y. Li. Doubly accelerated methods for faster cca and generalized eigendecomposition. In *ICML*, 2016.

[3] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255, 2013.

[4] R. Arora, T. V. Marinov, P. Mianjy, and N. Srebro. Stochastic approximation for canonical correlation analysis. In *Advances in Neural Information Processing Systems*, pages 4775–4784, 2017.

[5] G. Bécigneul and O.-E. Ganea. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018.

[6] K. Bhatia, A. Pacchiano, N. Flammarion, P. L. Bartlett, and M. I. Jordan. Gen-oja: Simple & efficient algorithm for streaming generalized eigenvector computation. In *Advances in Neural Information Processing Systems*, pages 7016–7025, 2018.

[7] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[8] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.

[9] R. Chakraborty, L. Yang, S. Hauberg, and B. Vemuri. Intrinsic grassmann averages for online linear, robust and nonlinear subspace learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[10] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pages 129–136, 2009.

[11] H. D. Couture, R. Kwitt, J. S. Marron, M. A. Troester, C. M. Perou, and M. Niethammer. Deep multi-view learning via task-optimal cca. *ArXiv*, abs/1907.07739, 2019.

[12] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20:303–353, 1998.

[13] C. Gao, D. Garber, N. Srebro, J. Wang, and W. Wang. Stochastic canonical correlation analysis. *Journal of Machine Learning Research*, 20(167):1–46, 2019.

[14] R. Ge, C. Jin, P. Netrapalli, A. Sidford, et al. Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis. In *International Conference on Machine Learning*, pages 2741–2750, 2016.

[15] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pages 134–151. Springer, 1971.

[16] G. H. Golub and H. Zha. The canonical correlations of matrix pairs and their numerical computation. 1992.

[17] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[18] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] M. Ishteva, P.-A. Absil, S. V. Huffel, and L. D. Lathauwer. Best low multilinear rank approximation of higher-order tensors, based on the riemannian trust-region scheme. *SIAM J. Matrix Anal. Appl.*, 32:115–135, 2011.

[21] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[22] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[23] V. S. Lokhande, A. K. Akash, S. N. Ravi, and V. Singh. Fairalm: Augmented lagrangian method for training fair models with little regret. *arXiv preprint arXiv:2004.01355*, 2020.

[24] Y. Luo, D. Tao, K. Ramamohanarao, C. Xu, and Y. Wen. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE transactions on Knowledge and Data Engineering*, 27(11):3111–3124, 2015.

[25] Z. Ma, Y. Lu, and D. P. Foster. Finding linear structure in large datasets with scalable canonical correlation analysis. In *ICML*, 2015.

[26] A. Morcos, M. Raghu, and S. Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, pages 5727–5736, 2018.

[27] A. Nemirovski, A. B. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optimization*, 19:1574–1609, 2009.

[28] E. Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[30] W. N. Price and I. G. Cohen. Privacy in the age of medical big data. *Nature medicine*, 25(1): 37–43, 2019.

[31] M. Reiß, M. Wahl, et al. Nonasymptotic upper bounds for the reconstruction error of pca. *Annals of Statistics*, 48(2):1098–1123, 2020.

[32] J. Rupnik and J. Shawe-Taylor. Multi-view canonical correlation analysis. In *Conference on Data Mining and Data Warehouses (SiKDD 2010)*, pages 1–4, 2010.

[33] C. G. Snoek, M. Worring, J. C. Van Gemert, J.-M. Geusebroek, and A. W. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 421–430, 2006.

[34] M. Tan, I. W.-H. Tsang, L. Wang, B. Vandereycken, and S. J. Pan. Riemannian pursuit for big matrix recovery. In *ICML*, 2014.

[35] R. Vershynin. Four lectures on probabilistic methods for data science. *The Mathematics of Data, IAS/Park City Mathematics Series*, pages 231–271, 2017.

[36] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In *International Conference on Machine Learning*, pages 1083–1092, 2015.

[37] W. Wang, R. Arora, K. Livescu, and N. Srebro. Stochastic optimization for deep cca via nonlinear orthogonal iterations. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 688–695. IEEE, 2015.

[38] W. Wang, J. Wang, D. Garber, and N. Srebro. Efficient globally convergent stochastic optimization for canonical correlation analysis. In *Advances in Neural Information Processing Systems*, pages 766–774, 2016.

[39] S. Yao and B. Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*, pages 2921–2930, 2017.

[40] F. Yger, M. Berar, G. Gasso, and A. Rakotomamonjy. Adaptive canonical correlation analysis based on matrix manifolds. In *ICML*, 2012.

[41] B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes] See experiment section (last subsection in fairness experiment).

(c) Did you discuss any potential negative societal impacts of your work? [Yes] See the last subsection of the paper.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] See section 2.

(b) Did you include complete proofs of all theoretical results? [Yes] See section 2 and supplement.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplement

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See experiment section.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] The performance we reported was averaged over several different runs.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See experiment section.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [Yes] License can be found in cited paper/website.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] In supplement.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]